【００７１】デコーダ９は、ページバッファ８内に一定量の分割データが記憶されると、デコード処理を開始し、その後順次デコード処理を続ける。これにより滞りなく動画像データの再生がなされる。

【００７２】次に、上記のような構成を有する動画像データ配信システム１の動作について以下に説明する。

【００７３】まず、エンコーダ２によって動画像データが分割され、分割データ及びインデックス情報が生成され、サーバ３に記憶される。

【００７４】次に、再生を求める動画像データの指定及び再生を開始させる位置が、ユーザによってビューワ４ａに入力される。

【００７５】次に、ユーザに指定された動画像データから生成されたインデックス情報５が、ビューワ４ａのインデックス情報受信部６からサーバ３に要求され、受信される。

【００７６】この受信されたインデックス情報５には、動画像データの再生時刻情報に対して分割データのファイル名が記述されている。

【００７７】分割データ受信部７では、受信されたインデックス情報５に基づいて、再生を開始したい位置以降に対応付けられている分割データがサーバ３に要求され、受信される。

【００７８】受信された分割データは、ページバッファ８の各ページに格納され、再生される時刻順に並び替えられる。

【００７９】また、ページバッファ８において一定量のバッファリングがなされると、デコーダ９によってストリーム再生が開始される。

【００８０】ページバッファ８においては、分割データが１ページ分再生される度にそのデータが破棄されてページが空けられる。

【００８１】そして、分割データ受信部７において、未受信の次の分割データが、インデックス情報５に基づいて検索され、サーバ３に要求、受信され、再生が繰り返される以上説明したように、本実施の形態に係る動画像データ配信システム１においては、動画像データがスプライスポイントにおいてファイル分割されており、その状態がインデックス情報５に記述されている。

【００８２】また、ビューワ４ａが、このインデックス情報５を要求し、続いてこのインデックス情報５に記されている好きな位置から分割データを順次要求・受信し、再生する。

【００８３】したがって、ユーザの望む位置からの分割データが順次再生されるため、動画像データのランダムアクセスが可能となる。

【００８４】また、本実施の形態においては、サーバ３の機能はファイルを保持し、またこの保持されているファイルの中から要求されたファイルを送信するのみである。したがって、この動画像データ配信システム１を導入する場合にはサーバ３に新しい機能や作用を追加する必要がない。

【００８５】ゆえに、クライアント４側のビューワ４ａを備え、サーバ３に分割データとインデックス情報５とを格納するのみで、容易にランダムアクセスを実現することができる。

【００８６】さらに、ランダムアクセスを実現させるための処理が、サーバ３ではなくビューワ４ａによって実行されるため、クライアント４が複数存在してもサーバ３の負荷の増加を防止することができる。

【００８７】なお、本実施の形態に係る動画像データ配信システム１では、エンコーダ２の機能をクライアント４に実装させ、クライアント４がネットワークを介して分割データ及びインデックス情報５をサーバ３に格納するとしてもよい。

【００８８】また、本実施の形態においては、ビューワ４ａの機能をブラウザ等の機能として実現可能であり、サーバ３はWWWサーバとすることができる。

【００８９】これにより、ブラウザで参照する動画コンテンツのランダムアクセスが可能となる。

【００９０】さらに、本実施の形態に係る動画像データ配信システム１では、クライアント４の台数を複数にすることができる。

【００９１】（第２の実施の形態）本実施の形態においては、第１の実施の形態に係る動画像データ配信システム１に備えられるビューワ４ａを変形したビューワについて説明する。

【００９２】先の第１の実施の形態においては、ビューワ４ａがサーバ３に分割データを要求してから受信するまでの時間がネットワークの回線の負荷に依存する。また、この回線の負荷は、様々な要因により時々刻々と変化する。

【００９３】先のビューワ４ａでは、伝送に要する時間や、回線の負荷による伝送速度の変化を吸収するために、動画像データの再生開始時に一定量分（しきい値を設定する）の分割データをバッファリングし、以後順次分割データを読み出している。

【００９４】この手法では、ビューワ４ａにバッファリングされる分割データの量を大きくすればその分だけ回線の負荷による遅延に対応可能となるが、その分再生開始までの待ち時間が長くなる。

【００９５】この再生開始までの待ち時間は、回線の負荷による遅延を吸収できる範囲で最小化されることが望ましい。したがって、バッファリングされる分割データの量も、回線負荷による伝送時間の変化に対応可能な範囲で最小化されることが望ましい。

【００９６】本実施の形態においては、分割データを要求してから受信するまでの時間を推定する。また、この推定された時間に基づいて、ページバッファに記憶しておく分割データの量を変更し、バッファリングされる分

Aylo Exhibit 1018
Page 1140 of 2308

割データの量を動的に最適化するビューワに関して説明する。

【００９７】図５は、本実施の形態に係るビューワの構成を示すブロック図であり、図４と同一の部分には同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【００９８】このビューワ１０は、第１の実施の形態において説明したビューワ４ａの分割データ受信部７の機能を変更した分割データ受信部１１を備え、新規の構成として時間推定部１２を付加している。また、このビューワ１０は、第１の実施の形態と同様の動画像データ配信システム１に設置可能である。

【００９９】時間推定部１２は、分割データ受信部１１がサーバ３に分割データを要求し、この分割データを受信するまでの時間（以下、「要求受信時間」という）を測定する。

【０１００】なお、この測定された要求受信時間を厳密に用いると、バッファリングの際に余裕がなくなる。したがって、時間推定部１２は多少余分を持たせるために若干多めの値を要求受信時間として推定する。例えば実際の要求受信時間が０．５秒であってとしても、要求受信時間を０．７秒と推定し、設定する。

【０１０１】この時間推定部１２において、要求受信時間は、再生開始以降には分割データが要求されて受信される度に再推定され、更新される。

【０１０２】分割データ受信部１１は、先の第１の実施の形態で説明した分割データ受信部７と同様の処理を行うが、サーバ３から分割データを読み出す際の条件が異なっている。

【０１０３】再生開始前においては、この分割データ受信部１１は、ページバッファ８内の分割データの再生時間の合計が要求受信時間を超えるまで、サーバ３から分割データを受信する。

【０１０４】再生中においては、現在再生中の分割データに関連付けされている再生時刻情報に現状の要求受信時間を加算し、この加算値に該当する再生時刻情報に関連付けされた分割データを要求し、受信する。すなわち、例えば要求受信時間が０．７秒と設定されているとすると、０．７秒後に再生されることになる分割データを要求し、受信する。

【０１０５】上記のような構成を有するビューワ１０の動作について以下に説明する。

【０１０６】図６は、本実施の形態に係るビューワ１０の再生開始前までの動作を示すフローチャートである。

【０１０７】このビューワ１０においては、まずインデックス情報５がサーバ３から受信される（ｓ１）。

【０１０８】次に、このインデックス情報５とユーザの指定内容に基づいて、再生開始位置の分割データが要求される（ｓ２）。

【０１０９】次に、要求された分割データが受信され（ｓ３）、この際に余分を持たせた要求受信時間が推定される（ｓ４）。

【０１１０】次に、インデックス情報５を参照して、次の分割データが要求され（ｓ５）、受信される（ｓ６）。

【０１１１】そして、要求受信時間分の分割データが、ページバッファ８に受信されたか否かが判定され（ｓ７）、受信されている場合には、再生が開始される。

【０１１２】受信されていない場合には、再度次の分割データが要求され、受信される。

【０１１３】図７は、本実施の形態に係るビューワ１０の再生開始後の動作を示すフローチャートである。

【０１１４】再生開始後においては、ページバッファ８に空きがあり（ｔ１）、かつ要求受信時間分の分割データがページバッファ８にない場合に（ｔ２）、次の分割データが要求される（ｔ３）。

【０１１５】なお、この間にもページバッファ８内の分割データはデコードされ再生されている（ｔ４、ｔ９）。

【０１１６】次に、要求された分割データが受信され（ｔ５）、この分割データが最後の分割データか否かが判定される（ｔ６）。

【０１１７】最後の分割データの場合にはデコード処理及び再生が継続され（ｔ７）、処理が終了される。

【０１１８】最後の分割データでない場合には、余分を持たせた要求受信時間が推定され（ｔ８）、デコード処理の実行により動画像データが再生されつつ（ｔ９）、上記の処理が繰り返される。

【０１１９】以上説明したように、本実施の形態に係るビューワ１０においては、サーバに分割データを要求してから受信するまでの時間に多少の余分を加えた時間分の分割データがページバッファ８に蓄積される。

【０１２０】したがって、必要最小限の分割データのみが先行確保されるため、再生開始までの時間を最適化することができる。

【０１２１】（第３の実施の形態）動画像データの再生中に、ネットワークの負荷により回線の伝送速度が分割データの伝送速度未満となると、動画像データがビューワにおいて途切れなく再生されない場合がある。

【０１２２】本実施の形態においては、このような場合に、再生の途中からでも高速伝送可能な分割データに切り換えて伝送を行い、これによりスムーズな再生を維持する動画像データ配信システムについて説明する。

【０１２３】本実施の形態に係る動画像データ配信システムは、図１と同様の構成を有するが、エンコーダ及びビューワに特別の機能が付加されている。

【０１２４】本システムのエンコーダは、元の動画像データが同一であり、スプライスポイントも同一であるが同一の条件における伝送速度が異なる複数種別の分割データ及びインデックス情報を生成する。

Aylo Exhibit 1018
Page 1141 of 2308

【０１２５】また、本システムのビューワは、要求受信時間に基づいて分割データの種別を選択する機能を有する。

【０１２６】以下、図１の場合と異なる機能を有するエンコーダ、及びビューワについて詳しく説明する。

【０１２７】図８は、本実施の形態に係る動画像データ配信システムに備えられるエンコーダの機能を示すブロック図である。

【０１２８】このエンコーダ１３は、エンコード時に、１つの動画像データから伝送速度の異なる複数種別の分割データを生成し、出力する。

【０１２９】なお、以下においては伝送速度の異なる２つの種別の分割データが生成される場合を例として説明するが、３つ以上の種別が生成される場合も同様である。また、この２つの種別のうち、一方を標準分割データとし、この標準分割データよりも伝送速度の速いもう一方を高速分割データとして説明する。

【０１３０】さらに、このエンコーダ１３は、生成された標準分割データのファイル名と高速分割データのファイル名を再生時刻情報に関連付けたインデックス情報１４を生成し、ファイル形式で出力する。

【０１３１】すなわち、このインデックス情報１４には、各再生時刻情報に複数列の分割データのファイル名が関連付けられており、この各列は異なる伝送速度になるようにエンコードされた分割データ列である。

【０１３２】サーバは、標準分割データ、高速分割データ、インデックス情報をファイルとして記憶し、クライアントのビューワから要求されたファイルを送信する。

【０１３３】ビューワは、第２の実施の形態に係る図５のビューワ１０と同様の構成を有するが、分割データ受信部１１に分割データ選択機能が付加されている点が異なる。

【０１３４】この分割データ選択機能は、標準分割データを要求してから受信するまでの時間が、時間推定部１２で推定された要求受信時間よりも大幅に大きい場合に、ネットワークの伝送速度が分割データの伝送速度よりも遅いと判断する。この場合、分割データ選択機能は、インデックス情報１４中の分割データの参照を、標準分割データ列から高速分割データ列に切り換える。

【０１３５】一方、高速分割データを要求してから受信するまでの時間が、時間推定部１２で推定された要求受信時間よりもある程度小さくなった場合には、ネットワークの伝送速度が分割データの伝送速度よりも速くなったと判断し、インデックス情報１４中の分割データの参照を、高速分割データから伝送速度は遅いが画質のよい標準分割データ列に切り換える。

【０１３６】図９は、このビューワにおいて再生開始後に実行される処理を示すフローチャートである。

【０１３７】このビューワでは、動画像データがデコードされ、再生されるが（ｕ１）、それと並列に要求伝送時間分の分割データがページバッファ８に格納されているか否かが判定される（ｕ２）。

【０１３８】ここで、要求伝送時間分の分割データがページバッファ８に格納されていない場合には、この要求伝送時間が大きすぎるか否かが分割データ選択機能によって判定される（ｕ３）。

【０１３９】要求伝送時間が大きすぎる場合には、インデックス情報１４の分割データ列のうち伝送速度の速い方が選択される（ｕ４）。

【０１４０】一方、要求伝送時間が大きすぎない場合には、要求伝送時間が小さすぎないか否かが判定される（ｕ５）。

【０１４１】要求伝送時間が小さすぎる場合には、インデックス情報１４の分割データ列のうち伝送速度の遅い方が選択される（ｕ６）。

【０１４２】そして、この分割データ選択機能によって選択された種別における次の分割データが、サーバに要求される（ｕ７）。

【０１４３】この間にも、動画像データのデコード処理が実行され、再生がなされている（ｕ８）。

【０１４４】要求された分割データが受信されると（ｕ９）、余分を持たせた要求受信時間が推定される（ｕ１０）。

【０１４５】そして、要求する分割データがなくなり（ｕ１１）、かつ再生する分割データもなくなると（ｕ１２）、処理が終了する。

【０１４６】要求する分割データが存在する場合には、上記の処理が繰り返される。

【０１４７】以上説明したように、本実施の形態に係る動画像データ配信システムにおいては、異なる伝送速度の分割データが生成され、分割データを要求してから受信するまでの時間が長い場合には、ビューワは伝送速度の速い分割データを読み出す。

【０１４８】また、分割データの種別の切り替えは各分割データの読み出し毎に行うことができるため、動画像データの再生開始後においても伝送速度を変更することができ、ネットワークの状況変更に柔軟に対応することができる。

【０１４９】さらに、伝送速度に余裕のある場合には画質のよい分割データに変更することができる。

【０１５０】ゆえに、クライアントのビューワで動画像データを途切れなく、良好に再生することができる。

【０１５１】なお、上記の各実施の形態において説明したビューワ、エンコーダの機能は、コンピュータに実行させることのできるプログラムとして、例えば磁気ディスク（フロッピーディスク、ハードディスク等）、光ディスク（ＣＤ－ＲＯＭ、ＤＶＤ等）、半導体メモリなどの記録媒体に書き込んで適用したり、通信媒体により伝送して各種装置に適用することも可能である。このビューワやエンコーダの機能を実現するコンピュータは、記

Aylo Exhibit 1018
Page 1142 of 2308

録媒体に記録されたプログラムを読み込み、プログラムによって動作が制御されることにより、上述した処理を実行する。

【０１５２】

【発明の効果】以上詳記したように本発明では、動画像データを分解した複数の分割データと、この各分割データとそれぞれの再生時刻情報とを関連付けたインデックス情報をサーバに備える。

【０１５３】そして、動画像データ再生装置は、まずインデックス情報を受信し、再生を開始したい位置の時間再生情報以降に関連付けられている分割データを順次読み出す。

【０１５４】これにより、サーバの構成、機能に手を加えることなく、ランダムアクセスが可能になる。

【０１５５】また、このランダムアクセスのための処理がサーバではなく動画像データ再生装置によって実行されるため、ランダムアクセス実行によるサーバの負荷の増加を防止することができる。

【０１５６】さらに、本発明では、動画像データ再生装置に蓄える分割データの量を、分割データを要求してからこの分割データを受信するまでの時間分とすることで、再生開始までの待ち時間を最適化することができる。

【０１５７】さらに、本発明では、伝送速度の異なる複数種別の分割データを用意しておき、ネットワークの状況によって受信する分割データの種別を切り換えることで、ネットワークにおける負荷の影響を受けることなくスムーズな再生が可能となる。

【図面の簡単な説明】

【図１】本発明の第１の実施の形態に係る動画像データ配信システムの構成を示すブロック図。

＊【図２】同実施の形態におけるエンコーダの機能を示すブロック図。

【図３】同実施の形態に係る動画像データ配信システムに備えられるサーバの構成を示すブロック図。

【図４】同実施の形態に係る動画像データ配信システムに備えられるビューワの機能を示ブロック図。

【図５】本発明の第２の実施の形態に係るビューワの構成を示すブロック図。

【図６】同実施の形態に係るビューワの再生開始前までの動作を示すフローチャート。

【図７】同実施の形態に係るビューワの再生開始後の動作を示すフローチャート。

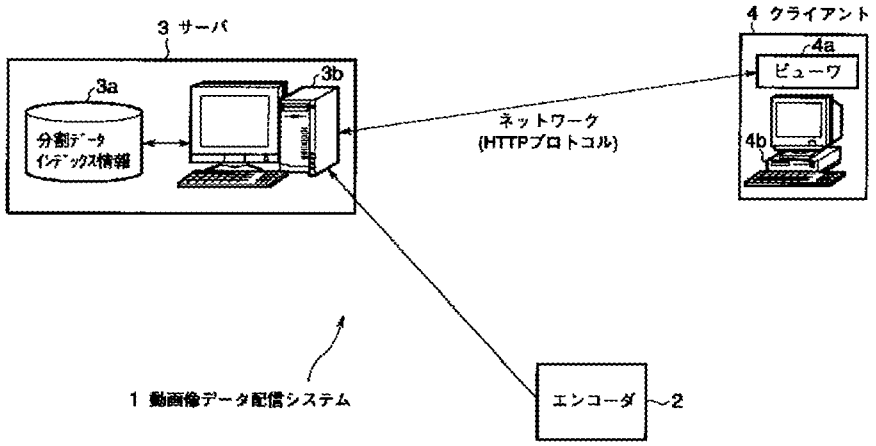
【図８】本発明の第３の実施の形態に係る動画像データ配信システムに備えられるエンコーダの機能を示すブロック図。

【図９】同実施の形態に係る動画像データ配信システムに備えられるビューワの再生開始後の動作を示すフローチャート。

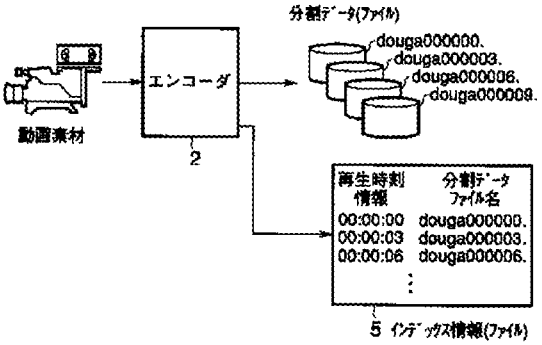
【符号の説明】

１…動画像データ配信システム
２、１３…エンコーダ
３…サーバ
４…クライアント
４ａ、１０…ビューワ
５、１４…インデックス情報
６…インデックス情報受信部
７、１１…分割データ受信部
８…ページバッファ
９…デコーダ
１２…時間推定部

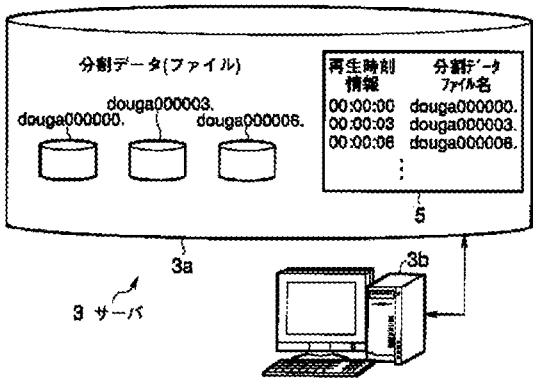
【図１】



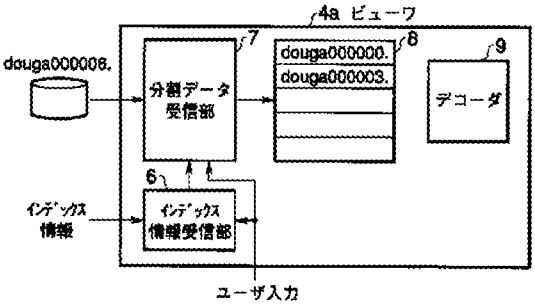
【図２】


分割データ(ファイル)
douga000000.
douga000003.
douga000006.
douga000009.
エンコーダ
動画素材
2
再生時刻情報
分割データファイル名
00:00:00 douga000000.
00:00:03 douga000003.
00:00:06 douga000006.
5 インデックス情報(ファイル)

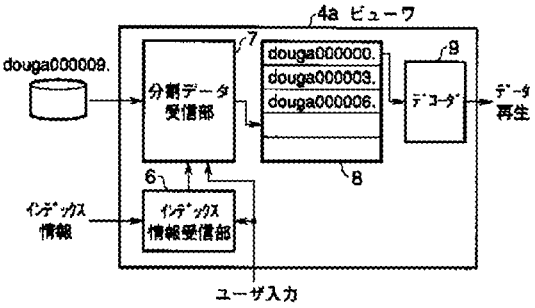
【図３】


分割データ(ファイル)
douga000000.
douga000003.
douga000006.
再生時刻情報
分割データファイル名
00:00:00 douga000000.
00:00:03 douga000003.
00:00:06 douga000006.
5
3a
3b
3 サーバ

【図４】

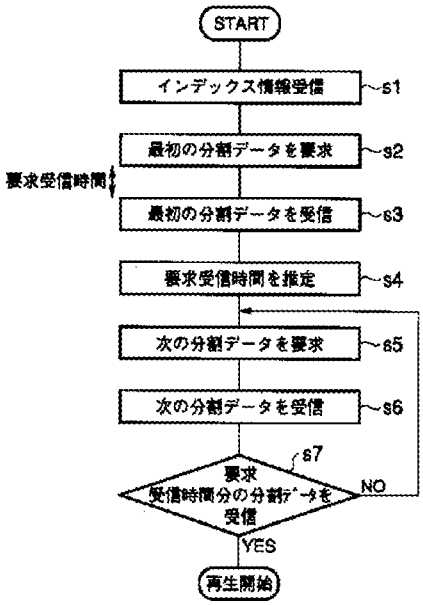

4a ビューワ
douga000006.
分割データ受信部
7
douga000000.
douga000003.
8
9
デコーダ
6
インデックス情報
インデックス情報受信部
ユーザ入力

(a)

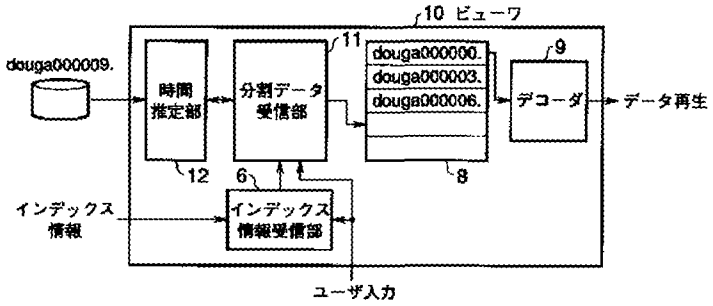

4a ビューワ
douga000009.
分割データ受信部
7
douga000000.
douga000003.
douga000006.
8
9
デコーダ
データ再生
6
インデックス情報
インデックス情報受信部
ユーザ入力

(b)

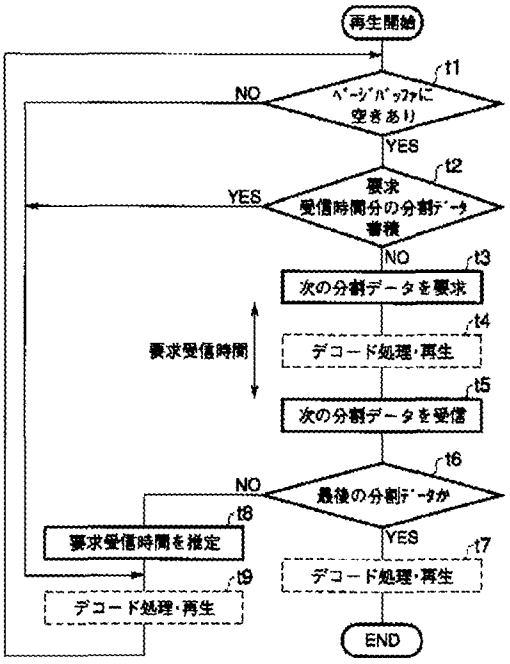
【図６】


START
インデックス情報受信
s1
最初の分割データを要求
s2
要求受信時間
最初の分割データを受信
s3
要求受信時間を推定
s4
次の分割データを要求
s5
次の分割データを受信
s6
s7
要求受信時間分の分割データを受信
NO
YES
再生開始

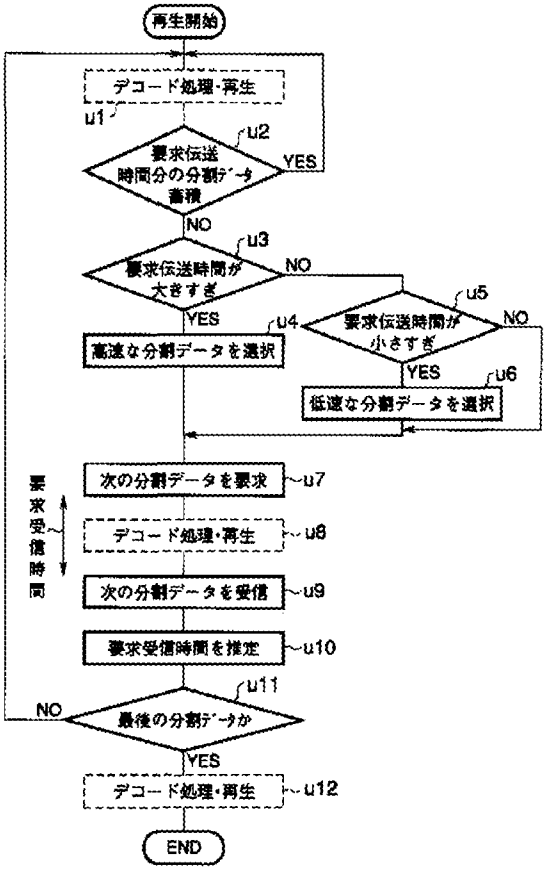
【図５】


10 ビューワ
douga000009.
時間推定部
分割データ受信部
11
douga000000.
douga000003.
douga000006.
9
デコーダ
データ再生
12
6
8
インデックス情報
インデックス情報受信部
ユーザ入力

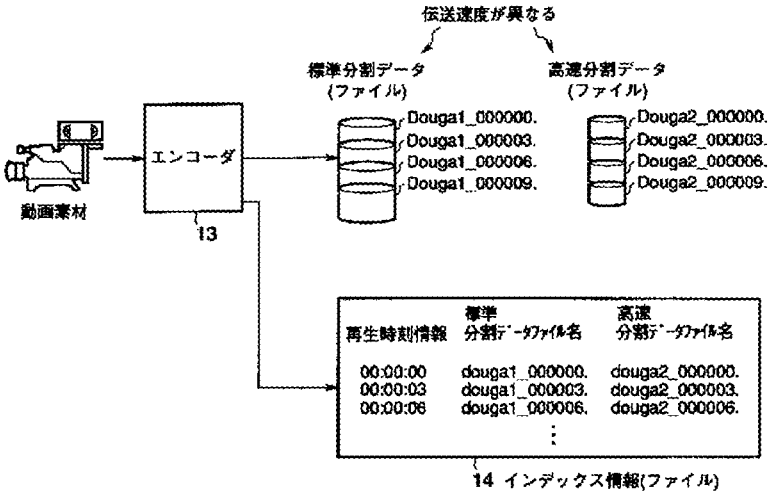
【図７】


再生開始
t1
NO
ページバッファに空きあり
YES
t2
YES
要求受信時間分の分割データ蓄積
NO
t3
次の分割データを要求
t4
デコード処理・再生
要求受信時間
t5
次の分割データを受信
t6
NO
最後の分割データか
YES
t8
要求受信時間を推定
t7
デコード処理・再生
t9
デコード処理・再生
END

【図９】


再生開始
デコード処理・再生
u1
u2
要求伝送時間分の分割データ蓄積
YES
NO
u3
要求伝送時間が大きすぎ
NO
YES
u4
高速な分割データを選択
u5
要求伝送時間が小さすぎ
NO
YES
u6
低速な分割データを選択
要求受信時間
次の分割データを要求
u7
デコード処理・再生
u8
次の分割データを受信
u9
要求受信時間を推定
u10
u11
NO
最後の分割データか
YES
デコード処理・再生
u12
END

【図８】


伝送速度が異なる
標準分割データ
(ファイル)
高速分割データ
(ファイル)
Douga1_000000.
Douga1_000003.
Douga1_000006.
Douga1_000009.
Douga2_000000.
Douga2_000003.
Douga2_000006.
Douga2_000009.
エンコーダ
動画素材
13
再生時刻情報
標準
分割データファイル名
高速
分割データファイル名
00:00:00
00:00:03
00:00:06
douga1_000000.
douga1_000003.
douga1_000006.
douga2_000000.
douga2_000003.
douga2_000006.
14 インデックス情報(ファイル)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 April 2003 (03.04.2003)

PCT

(10) International Publication Number
WO 03/027876 A1

(51) International Patent Classification[7]: G06F 15/16

(21) International Application Number: PCT/US02/30874

(22) International Filing Date:
26 September 2002 (26.09.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/325,483 26 September 2001 (26.09.2001) US

(71) Applicants and
(72) Inventors: **REYNOLDS, Jodie, Lynn** [US/US]; 728 King Way, Folsom, CA 95630 (US). **INGRAHAM, Robert, Walter** [US/US]; 127 Honey Cook Circle, Folsom, CA 95630 (US).

(74) Agents: **GALLENSON, Mavis, S.** et al.; Ladas & Parry, Suite 2100, 5670 Wilshire Boulevard, Los Angeles, CA 90036-5679 (US).

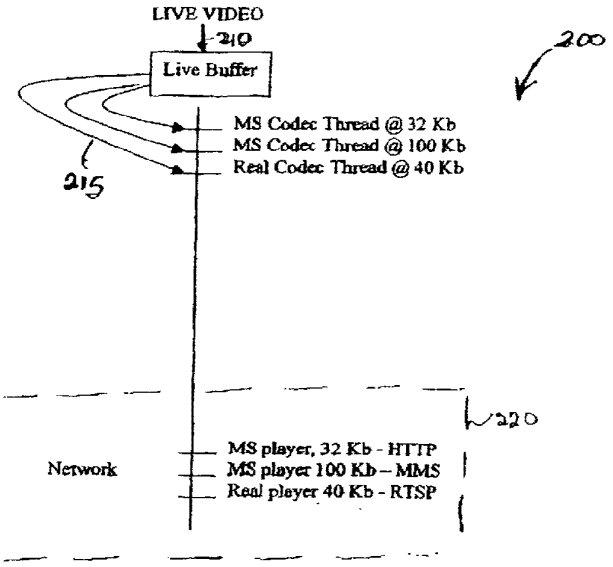
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM AND METHOD FOR COMMUNICATING MEDIA SIGNALS



(57) Abstract: A media streaming system for streaming media signals is provided. The media streaming system takes a library of separate and distinct CODECs (145) that are provided as a searchable CODEC library and used in determining specific characteristics in the media signal to identify similar sections of the signal. The media streaming system uses a computer implemented intelligence system, such as an intelligence mechanism to learn and capture the unique charateristics of a signal as the signal is being streamed (150). The media streaming system also compresses and decompresses the media signal as the signals are streamed from a source media (100) to a destination device (130, 135, 140).

WO 03/027876 A1

SYSTEM AND METHOD FOR COMMUNICATING MEDIA SIGNALS

RELATED APPLICATION

This application is a non-provisional application of "Reynolds et al. a provisional application entitled "System and Method for Communicating Media Signals" attorney docket number ID-PAT-001PR1, application number 60/325,483, filed on September 26, 2001.

FIELD

This disclosure relates to a system and method for communicating media signals between source and destination devices. More specifically, it relates to a system and method for compressing and decompressing streaming and static media signals for efficiently communicating those signals between source and destination devices using artificial intelligence mechanism.

BACKGROUND

The ability to efficiently communicate streaming and static media between remotely located devices is a significant need that has emerged exponentially with the advent of networked communications such as the Internet. This need has been recently addressed with substantial development resources on a worldwide scale.

The term "media" is herein intended to mean information that may be communicated in the form of a signal from a source device to a destination device for use by the destination device; and, where used herein, media is generally contemplated to comprise either streaming or static media signals. For the purpose of this disclosure, the term "use" as applied to a destination device's operation on media signals is intended to include playing (e.g. sounds, images, video), processing (e.g. telemetry data), or any other use or operation that is the intended purpose of the media signal.

The terms "streaming media" are herein intended to mean media signals that comprise information intended to be communicated to and used by a destination device in a temporal, streaming fashion. The term "streaming" as applied to streaming media signals is herein intended to include signals communicated and processed in a continuous manner over time, or signals that may be

signals communicated and processed in a continuous manner over time, or signals that may be communicated in a series of discrete packets, pieces, or blocks that are interrelated and may be thereafter used by the destination device in a continuous, interrelated fashion. Examples of streaming media signals for the purpose of this disclosure therefore include, without limitation, the following types of media: video, audio, audio combined with video, and data strings such as temporal telemetry. The terms "streaming media" are most typically used by reference to digitized forms of data representing the subject media.

The terms "static media" are herein intended to generally mean media that is not "streaming" as defined above. Static media signals are of the type that generally may be communicated and are intended to be used as a single packet, block, or piece. Static media therefore may include for example, without limitation the following: a discrete image, an individual and relatively temporally short video clip, a sound or sound bite, or a piece or block of information such as telemetry information. It is contemplated, however, that such a "single piece" of static media may be of sufficient magnitude to consist of a plurality of smaller pieces or sub-parts, such as for example regions or pixels of an overall image, individual frames that together form a video clip, digital bits that together comprise a sound, a group of sounds that comprise a sound bite, or bits of information that together comprise a larger block of information.

Streaming media generally includes data files that are significantly larger than static media files, and also often represent many more variables over the temporal communication of such files than experienced with most static media files. Therefore, the ability to efficiently compress streaming media for appropriate communication to destination devices for use is often a much more complex and difficult to achieve goal. Accordingly, much of this disclosure is provided by reference specifically to streaming media communication, and the present invention has been observed to provide significant benefits for such communication. However, where streaming media is specifically referenced herein with respect to this background, and further with respect to the many benefits of the present invention herein disclosed, static media is also further contemplated where appropriate according to one of ordinary skill.

Many different "type-specific" media systems have been in use for quite a long time for transmitting specific types (e.g. video, audio, image, voice, etc.) of streaming and static media signals between sources and remote destinations. Typical examples of such type-specific media systems include television transmission systems, telephone line systems, and radio transmission systems, and every television, telephone, and radio is therefore a receiving device for media. Accordingly, the needs

for efficient communication of streaming and static media touch upon many diverse communications industries, including for example the telephone, television, movie, music, and more recently interactive gaming industries.

Moreover, many medial communications systems, including the various long-standing type-specific systems, are also "format specific", wherein the subject media signals are communicated in a particular format such that the source, transmission channel, and destination device must be specifically compliant to work within that format. Examples of format specific media systems include for example encoded cable television systems that work only for certain types of media and only delivered in particular encoded formats from the cable carrier. Therefore, these systems, in hardware and software, are generally dedicated to only the type and format of media to be provided by the content provider.

Society's needs have outpaced the abilities of these dedicated, content-specific and format-specific systems. In particular, these dedicated systems are not structured to accommodate the ever increasing client demand, real-time, for specified streaming media. Still further, technology developments in the recently interconnected world has tempted the palate of society for the ability to pull, receive, push, and send multiple types of media in multiple formats using one device. Moreover, content providers need to be able to deliver many different media signals to many different types of devices in their clients' offices, living rooms, and hands. Individuals and corporations also desire to communicate with each other using various different formats and using various different respective devices.

Accordingly, a significant industry has emerged for delivering streaming and static media over the centralized network of the Internet. Content delivery companies are currently delivering a wide range of streaming media, from live horse racing and entertainment to medical telemetry and education, over the Internet, and in video and audio formats. According to one published report from DFC Intelligence, video streaming on the Internet grew 215 percent in 2000 to over 900 million total streams accessed. This includes broadband streams, which made up almost 29 percent of total accesses. This same report also estimates that as much as 15 percent of available stream inventory is now being exploited with in-stream advertising. In another report published by Internet researcher Jupiter Media Metrix, business spending alone on streaming video technology will balloon from one-hundred forty million (US$140M) US dollars in 2000 to nearly three billion (US$3B) US dollars by 2005 as companies turn to electronic interaction in communicating with employees, consumers and other businesses.

Still further, the population explosion and increasing number of people transmitting on these systems has severely impacted the available bandwidth for available information. Therefore, the ability to stream media efficiently, using limited bandwidth resources and limited available transmission speeds, is of increased societal importance.

Compression/Decompression Algorithms ("CODECS")

In view of the exponential demand for communicating the different types of media, various compression/decompression systems ("CODEC(s)") have been developed over many years, and have in particular become the recent topic of significant research and development. Specific types of CODECS and systems for managing the operation of CODECS with respect to communicating streaming and static media signals have been developed for specific types of media, including for example still-frame images such as graphics and photographs, and streaming media.

Image CODECS

Various different types of static media CODECS have been developed, and a wide variety of these CODECS are widely known and used. One specific type of static media that has been the topic of particular attention includes images (though a long series of interrelated image frames such as in video context is generally treated as streaming media due to more complex variables, e.g. size and temporal relationship between frames, that significantly impact appropriate compression/decompression needs). Examples of static media CODECing is therefore herein exemplified by reference to certain specific types of conventional image CODEC technologies and methods.

The two most common file formats for graphic images on the world wide web are known as "GIF" and "JPEG" formats, generally considered the respective standards for drawings (e.g. line art) and photographs, and are further described together with other image compression modalities for the purpose of further understanding as follows.

"JPEG" is an acronym for "Joint Photographic Experts Group", and is a graphic image file that complies with ISO standard 10918. Commonly used for photograph compression/decompression, a JPEG file is created by choosing from a range of compression qualities, or, as has also been described, by choosing from one of a suite of compression algorithms. In order to create a JPEG file, or convert

an image from another format to JPEG, the quality of image that is desired must be specified. In general, because the highest quality results in the largest file, a trade-off may then be made, as chosen by the user, between image quality and image size. The JPEG mode of compression generally includes 29 distinct coding processes although a JPEG implementer may not use them all. A JPEG image is typically given a name suffix ".jpg".

"GIF" is an acronym for "Graphics Interchange Format", and is generally considered the de facto standard form of drawing image compression/decompression for Internet communication. GIF formatting uses a compression algorithm known as the LZW algorithm, which was developed by Abraham Lempel, Jacob Ziv, and Terry Welch and made commercially available by Unisys Corporation (though in general such algorithm has been made publicly available without requiring fee-bearing licenses). More specifically, a "LZW" compression algorithm takes each input sequence of bits of a given length (e.g. 12 bits) and creates an entry in a table, sometimes called a "dictionary" or "codebook", for that particular bit pattern. The entry consists of the pattern itself and a shorter code. As input is read, any pattern that has been read before results in the substitution of the shorter code, effectively compressing the total amount of input to something smaller. Earlier approaches, known as LZ77 and LZ78, did not include the look-up table as part of the compressed file. However, the more recent LZW algorithm modality does include the table in the file, and the decoding program that decompresses the file for viewing is able to build the table itself using the algorithm as it processes the encoded input. The GIF format uses the 2D raster data type (associated with display screens using raster lines) and is encoded in binary.

Two versions of GIF formats include GIF 87a, and more recently GIF89a that allows for "animated GIF" file creation, or short sequences of images within a single GIF file that are played in sequence to present movement or change in the image (either in an endless loop or through a progression that reaches an end). GIF89A also allows for, and also for "interlaced GIF", which is a GIF image that arrives and is displayed by the receiver first as a fuzzy outline of an image that is gradually replaced by seven successive waves of bit streams that fill in the missing lines until full resolution is reached. Interlaced GIF allows, for example, a viewer using 14.4 Kbps and 28.8 Kbps modems to observe a briefer wait-time before certain information in a subject image may be processed, such as for example to make decisions (e.g. to click on the image to execute an operation such as a link).

By presenting waves of resolution filling image sequences, interlaced GIF is similar to "Progressive JPEG", which describes an image created using the JPEG suite of compression algorithms

that will "fade in" in successive waves. While the progressive JPEG is often observed to be more appealing way to deliver an image at modem connection speeds, users with faster connections may not likely notice a difference.

"PNG" or "Portable Network Graphics" format has been more recently developed for image compression and that, in time, has been publicized to replace the GIF format for Internet use (though not generally the JPEG format allowing size/quality trade-offs). This format has been developed for public consumption and development. Similar to GIF, PNG is considered a "lossless" compression format, and therefore all image information is restored when a compressed file is decompressed during viewing. However, PNG formatted files are generally intended to be from 10 to 30 percent more compressed than with a GIF format. Further aspects of PNG file formats are provided as follows: (i) color transparency may not be limited to only one color, but the degree of transparency may be controlled ("opacity"); (ii) "interlacing" of images is improved versus standard GIF; (iii) "gamma correction" is enabled, allowing for "tuning" of images in terms of color brightness required by specific display manufacturers; (iv) images can be saved using true color, palette, and gray-scale formats similar to GIF; and (v) "animation" is generally not supported, though PNG is generally considered extensible and therefore software may be layered to provide for such scriptable image animation.

"TIFF" is an acronym for "Tag Image File Format", and is a common format for exchanging raster graphics (or "bitmap") images between application programs, such as for example graphics used for scanner images. A TIFF file is usually given a name suffix of ".tif" or ".tiff", and had generally been developed in the mid-1980's with the support of Adobe Software, Microsoft, and Hewlett-Packard. TIFF files can be in any of several classes, including gray scale, color palette, or RGB full color, the descriptions and differences of which are further developed elsewhere herein this disclosure. TIFF files may also include files with JPEG, LZW, or CCITT Group 4 standard run-length image compression, which are also further described elsewhere herein. As one of the most common graphic image formats, TIFF files are typically used in desktop publishing, faxing, 3-D applications, and medical imaging applications.

Video CODECS

Video compression has been the topic of intense development for various applications, including, for example: pre-recorded video (e.g. "video-on-demand"), teleconferencing, and live video (e.g. broadcasts). "Desk-top" computers, wireless devices, conventional televisions, and high

definition televisions are examples of the different types of receiving devices that an efficient video compression system must serve.

In general, video CODEC algorithms operate on either or both of an individual, frame-by-frame basis, and/or on a "temporal compression" basis wherein each frame is the most common video compression algorithms in conventional use are based on several mathematic principles, including the following: Discrete Cosine Transforms ("DCT"), Wavelet Transforms and Pure Fractals.

"Discrete Cosine Transforms" or "DCT's" are by far the most popular transforms used for image compression applications . In general, DCT is a technique for representing waveform data as a weighted sum of cosines. The DCT is similar to the discrete Fourier transform: it transforms a signal or image from the spatial domain to the frequency domain. The DCT helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). Reasons for its popularity include not only good performance in terms of energy compaction for typical images but also the availability of several fast algorithms. DCTs are used in two international image/video compression standards, JPEG and MPEG.

"Wavelet transforms" are generally mathematical algorithms that convert signal data into a set of mathematical expressions that can then be decoded by a destination receiver device, such as for example in a manner similar to Fourier transform. Wavelets have been observed to enhance recovery of weak signals from noise, and therefore images processed in this manner can be enhanced without significant blurring or muddling of details. For this reason, wavelet signal processing has been particularly applied to X-ray and magnetic-resonance images in medical applications. In Internet communications, wavelets have been used to compress images to a greater extent than is generally possible with other conventional methods. In some cases, the wavelet-compressed image can be as small as about 25 percent the size of a similar quality image using the more familiar JPEG format, which is discussed in further detail elsewhere in this disclosure. Thus, for example, a photograph that requires 200Kb and takes a minute to download in JPEG format may require only 50Kb and take only 15 seconds to download in wavelet-compressed format. A wavelet-compressed image file is often given a name suffix ".wif", and either the receiver (e.g. Internet browser on a computer receiver) must support these format specific files, or a plug-in program will be required to read such file.

Fractal image compression is a modern technique of lossy image coding that provides several improvements over existing Fourier series compression schemes. Edge depiction is improved since, when modeled as a step function, edges require a large number of Fourier series terms to properly depict. Other advantages of fractals include fast decoding time and scale independence. Fractal

compression is based on Mandelbrot sets which take advantage of a self similar, scaling dependent, statistical feature of nature (Mandelbrot, 1983). Fractal compression and decompression involves a clustering approach to find regions which show the same characteristics as a sample region independent of rotation and scale. The fractal image compresses images as recursive equations and instructions about how to reproduce them. The equations describe the image in terms of the relationships between its components. The reduction in storage need is due to the fact that fractal compression saves equations and instructions instead of a pixel representation of the image.

"MPEG" is an acronym for Moving Picture Experts Group and has come to be used synonymously with certain evolving video and audio compression standards promulgated therefrom. In general, to use MPEG video files, a personal computer is required with sufficient processor speed, internal memory, and hard disk space to handle and play the typically large MPEG file, usually given the name suffix ".mpg". A specified MPEG viewer or client software that plays MPEG files must be available on the client system, and generally can be downloaded shareware or versions of commercial MPEG players from various sites on the Web. The modes of operation for MPEG formatted media are herein described by reference to these sequentially evolved standards as follows.

More specifically, MPEG-1 standard was designed for coding progressive video generally at a transmission rate of about 1.5Mbps. This was generally designed for the specific application for Video-CD and CD-I media. MPEG-1 audio layer-3 ("MP3") has also evolved from early MPEG work. "MPEG-2" is a standard generally designed for coding interlaced images at transmission rates above 4 Mbps, and was generally intended for use with digital TV broadcast and digital versatile disk. Though it is generally observed that many MPEG-2 players can handle MPEG-1 data as well, the opposite is not generally observed to be true and MPEG-2 encoded video is generally incompatible with MPEG-1 players. Yet another progressive standard, "MPEG-3", has also been proposed for use with high definition television ("HDTV"), though in general MPEG-3 has merged with MPEG-2 which is generally believed to meet the HDTV requirements. Finally, an "MPEG-4" standard has also been most recently developed and is intended to provide a much more ambitious standard to address speech and video synthesis, fractal geometry, and computer visualization, and has further been disclosed to incorporate artificial intelligence in order to reconstruct images.

MPEG-1 and –2 standards define techniques for compressing digital video by factors varying from 25:1 to 50:1. This compression is achieved according to these standards generally using five different compression techniques: (i) discrete cosine transform (DCT), which is a frequency-based transform; (ii) "quantization", which is a technique for losing selective information, e.g. lossy

prediction", wherein some images are predicted from the pictures immediately preceding and following the image.

Further more detailed examples of commercially available video compression technologies include: Microsoft Media Player™ (available from Microsoft Corporation), RealPlayer™ or RealSystem G2™ (commercially available from Real Networks™), Apple's QuickTime™ (commercially available from Sorenson™); and "VDO". The Microsoft Media Player™ is generally believed to apply the MPEG standard of CODEC for compression/decompression, whereas the others have been alleged to use proprietary types of CODECS. Standard compression algorithms, such as MPEG4, have made their way into the hands of developers who are building embedded systems for enterprise streaming, security, and the like.

One example of a more recent effort to provide streaming video solutions over Wireless and IP networks has been publicized by a company named Emblaze Systems (LSE:BLZ). This company has disclosed certain technology that is intended for encoding and playback of live and on-demand video messages and content on any platform: PC's, PDA's, Video cell phones and Interactive TV. Emblaze Systems is believed to be formerly GEO Interactive Media Group. The following Published International Patent Applications disclose certain streaming media compression technologies that is believed to be related to Emblaze Systems to the extent that GEO Interactive Media Group is named as "Assignee": WO9731445 to Carmel et al.; and WO9910836 to Carmel. The disclosures of these references are herein incorporated in their entirety by reference thereto.

Another company that has published CODEC technology that is intended to improve communication of streaming media for wireless applications is Packetvideo™ Corporation, more specifically intending to communicate streaming video to cellular phones. In addition, they are believed to be promoting CODEC technology that is intended to track temporal scalability and signal error resistance in order to protect video and audio streams from the hazards of the wireless environment. U.S. Patent No. 6,167,092 to Lengwehasatit discloses further examples of certain streaming media compression/decompression technology that are believed to be associated with Packetvideo as the named "Assignee" on the face of this Patent reference. The disclosure of this patent reference is herein incorporated in its entirety by reference thereto.

Another prior reference discloses CODEC technology that is intended to provide a cost effective, continuously adaptive digital video system and method for compressing color video data for moving images. The method involves capturing an analog video frame and digitizing the image into a preferred source input format for compression using a combination of unique lossy and lossless digital

Another prior reference discloses CODEC technology that is intended to provide a cost effective, continuously adaptive digital video system and method for compressing color video data for moving images. The method involves capturing an analog video frame and digitizing the image into a preferred source input format for compression using a combination of unique lossy and lossless digital compression techniques including sub-band coding, wavelet transforms, motion detection, run length coding and variable length coding. The system includes encoder and decoder (CODEC) sections, generally disclosed to use a "Huffman" encoder, for compression and decompression of visual images to provide high compression that is intended to provide good to excellent video quality. The compressed video data provides a base video layer and additional layers of video data that are multiplexed with compressed digital audio to provide a data stream that can be packetized for distribution over inter or intranets, including wireless networks over local or wide areas. The CODEC system disclosed is intended to continuously adjust the compression of the digital images frame by frame in response to comparing the available bandwidth on the data channel to the available bandwidth on the channel for the previous frame to provide an output data stream commensurate with the available bandwidth of the network transmission channel and with the receiver resource capabilities of the client users. The compression may be further adjusted by adjustment of the frame rate of the output data stream.

Further more detailed examples of CODEC systems that are intended for use at least in part for streaming video communication are disclosed in the following U.S. Patent Nos.: 6,081,295 to Adolph et al.; 6,091,777 to Guetz et al.; 6,130,911 to Lei; 6,173,069 B1 to Daly et al.; 6,263,020 B1 to Gardos et al.; 6,272,177 to Murakami et al.; and 6,272,180 B1 to Lei. The disclosures of these references are herein incorporated in their entirety by reference thereto.

Most if not all prior streaming video compression methodologies look to the extremely complex mathematical tools within such CODECS, and subtle changes to them, to carry "one size fits all" video over public and private networks of all types, from ultra-low bandwidth networks such as that found in wireless networks, to satellite communications to ultra-high speed fiber optic installations. Among the various conventional methods of compression, there are generally user-definable parameters, including tradeoffs between image size, frame rate, color depth, contrast, brightness, perceived frame quality, buffer length, etc. Further, within the algorithms themselves there are numerous non-user definable qualities and weighted calculations. It is up to the developers to set these one time for one "general" interest, and then package and ship the product.

However, while the video streaming market continues to grow rapidly, the world has not chosen one standard for compression as no one algorithm is ideal for all video sources, destinations, or transmission modalities. While a first CODEC may be best for one type of signal, or for a first portion of a signal (e.g. frame or scene comprising a series of frames), another second CODEC may be best for another type of signal, or even another second portion of the same signal. Still further, one CODEC may be best suited for compression/decompression of a particular streaming signal among send, receive, and transmission devices in a communications network; another second CODEC may be better suited than the first for the same streaming media signal but for another set of communication device parameters. For example, some video streams may deliver color to handheld devices while other video streams can take advantage of the loss of pixels in a black and white transmission to a cellular phone to increase frame rate. Required sound quality, frame rates, clarity, and buffering tolerance all decidedly impact the compression algorithm of choice for optimized video and audio delivery across multiple platforms.

In fact, certain communication device parameters may be sufficiently transient during the streaming media transmission such that an initially appropriate CODEC for an initial set of parameters may be rendered less efficient than another CODEC due to changes in those parameters during the same streamed signal transmission. Examples of such transient parameters include, without limitation: available band width in the data transmission channel, available memory or processing power in either the send or receiving devices, and dedicated display resolution/window in the receiving device (e.g. minimizing windows on a screen). These problems are compounded exponentially by a vast number of iterations of different combinations of such factors that may differentiate one CODEC from another as being most efficient for compression, decompression, and delivery of a specific streaming media signal along a particular communications device system.

As CODEC systems are "format-specific", source and destination devices must be "pre-configured" to communicate media signals between each other according to common, specific compression/decompression modalities, else transcoders must be used. However, even if conventional transcoders are used, constraints in the communication system (e.g. source, transmission channel, destination device) are not generally considered and the communication may be significantly faulty. For the purpose of further illustration, Figures 1A and 1B show two different schematic representations of conventional methods for communicating media between source 110 -120 and destination devices 130 -140. These illustrations specifically exemplify streaming video communication, though other media forms may be represented by similar systems.

Aylo Exhibit 1018
Page 1158 of 2308

It has been observed that CODEC algorithms can generally be modified for a specific application and then perform better than a similar unmodified set over that limited example. However, this generally must be done either for a series of frames, or ideally, for each individual frame. Some DCT based algorithms have as many as two billion mathematical operations occurring for each frame at higher resolutions and lower perceived quality. This is entirely too much math for average machines, even commercial servers, to perform thirty to sixty times in a single second. This is the reason for the advent of the dedicated compression board or ASIC.

Audio CODECS

In addition to society's recent interests in improving video compression, audio compression has likewise been the topic of significant efforts also for various live or pre-recorded applications, including audio broadcast, music, transmission synchronized with video, live interactive voice (e.g. telephone). Any and all of these audio compression applications must be compatible with a wide range of client-side receiver/players, such as on a multitude of handheld or desk-top devices having widely varied capabilities and operating parameters.

Conventional audio CODECS generally comprise several different types, a few of which are herein briefly summarized for the purpose of illustration.

"Code Excited Linear Prediction" or "CELP" is a type of speech compression method using waveform CODECS that use "Analysis-by-Synthesis" or "AbS" within the excitation-filter framework for waveform matching of a target signal. CELP-based CODECS have recently evolved as the prevailing technique for high quality speech compression, and has been published to transmit compressed speech of toll-quality at data rates nearing as low as about 6 kbps. However, at least one publication discloses that the quality of CELP coded speech is reduced significantly for bit rates at or below 4 kbps.

"Vocoders" are speech CODECS that are not based on the waveform coding scheme, but rather use a quantized parametric description of the target input speech to synthesize the reconstructed output speech. Vocoders have been disclosed to deliver better speech quality at low bit-rates, such as about 4 kbps, and have been developed for such applications. Low bit rate vocoders use the periodic characteristics of voiced speech and the "noise-like" characteristics of stationary unvoiced speech for speech analysis, coding and synthesis. Some early versions of vocoders (e.g. federal standard 1015 LPC-10, use a time-domain analysis and synthesis method. However, most of the more recent versions,

which at least one publication labels "harmonic coders", utilize a harmonic spectral model for voiced speech segments.

Notwithstanding the previous description of certain specific speech compression techniques, a vast number of speech CODECs and standards have been developed by industry and managed by industry and nonprofit groups. Examples of such groups include without limitation the following, and their standards are often used as reference types of CODECS: the European Telecommunications Standards Institute ("ETSI"); the Institute of Electrical and Electronics Engineers ("IEEE"); and the International Telecommunication Union Telecommunications Standards Sector ("ITU-T"), formerly the "CCITT".

One more recently disclosed method and apparatus for hybrid coding of speech, specified at 4 Kbps, encodes sppech for communication to a decoder for reproduction of the speech where the speech signal is classified into three types: (i) steady state voiced or "harmonic"; (ii) stationary unvoiced; and (iii) "transitory" or "transition" speech. A particular type of coding scheme is used for each class. Harmonic coding is used for steady state voiced speech, "noise-like" coding is used for stationary unvoiced speech, and a special coding mode is used for transition speech, designed to capture the location, the structure, and the strength of the local time events that characterize the transition portions of the speech. The compression schemes are intended to be applied to the speech signal or to the LP residual signal.

Another recently disclosed method and arrangement for adding a new speech encoding method to an existing telecommunication system is also summarized as follows. A CODEC is introduced into a speech transmitting transceiver of a digital telecommunications system in order to use a "new" CODEC and an "old" CODEC in parallel in the system. A CODEC is selected by implementing a handshaking procedure between transceivers where a speech encoding method is implemented in all transceivers and previously used in the telecommunications system concerned. The handshaking is used at the beginning of each connection. At the beginning of a phone call and after handover, the method checks whether both parties can also use the new speech encoding. The handshaking messages have been selected so that their effect on the quality of speech is minimal, and yet so that the probability of identifying the messages is maximal.

Still another relatively recent reference discloses a tunable perceptual weighting filter for tandem CODECS intended for use in speech compression. Specific filter parameters are tuned to provide improved perfromance in tandeming contexts. More specifically, the parameters used are 10th

order LPC predictor coefficients. This system is specified to use "Low-Delay Excited Linear Predictive" CODECS, or "LD-CELP".

Further more detailed examples of streaming audio communications systems using CODECS such as according to the examples just described are provided in the following US Patent references: 6,144,935 to Chen et al.; 6,161,085 to Haavisto et al.; and 6,233,550 to Gersho. The disclosures of these references are herein incorporated in their entirety by reference thereto.

Artificial Intelligence ("AI") and Neural Networks with CODECS

Various systems and methods have been recently disclosed that are intended to integrate artificial intelligence ("AI") or neural networks with the compression and decompression of streaming media signals.

The terms "artificial intelligence" are herein intended to mean the simulation of human intelligence processes by computer systems, including learning (the acquisition of information and rules for using the information), reasoning (using the rules to reach approximate or definite conclusions), and self-correction. Particular applications of AI include "expert systems", which are computer programs that simulate judgment and behavior of a human or organization that has expert knowledge and experience in a particular field. Typically, expert systems contain a knowledge base to each particular situation that is described to the program, and can be enhanced with additions to the knowledge base or to the set of rules.

The terms "neural network" are herein intended to mean a system of programs and data structures that approximates the operation of the human brain, usually involving a large number of processors operating in parallel, each with its own small sphere of knowledge and access to data in its local memory. Typically, a neural network is initially trained or fed large amounts of data and rules about data relationships, after which a program can tell the network how to behave in response to an external stimulus (e.g. input information). In making determinations, neural networks use several principles, including without limitation gradient-based training and fuzzy logic. Neural networks may be further described in terms of knowledge layers, generally with more complex networks having deeper layers. In "feedforward" neural network systems, learned relationships about data can "feed forward" to higher layers of knowledge. Neural networks can also learn temporal concepts and have been widely used in signal processing and time series analysis. Other published applications of neural

networks include oil exploration data analysis, weather prediction, the interpretation of nucleotide sequences in biology labs, and the exploration of models of thinking and consciousness.

The terms "fuzzy logic" are herein intended to mean an approach to computing based upon "degrees of truth" rather than "Boolean logic" which operates within only a true/false (or "binary", as 1 or 0) domain. Fuzzy logic was first advanced by Dr. Lotfi Zadeh of the University of California at Berkeley in the 1960's in relation to work on a problem of computer understanding of natural language, which is not easily translated into absolute Boolean logic terms. Fuzzy logic often does include the cases of 0 and 1 as extreme cases of truth, but also includes the various states of truth in between (e.g. a determination of that the state of being is at some threshold, such as 0.98, may assists in making a decision to assign a 1 with an acceptably low occurrence of error in an operation).

One example of a previously disclosed streaming media compression/decompression system intended to use with artificial intelligence through a neural network uses a Radon transform in order to compress data such as video data. Several previously disclosed AI and/or neural network systems are intended to use AI and/or neural networks for the purpose of error correction during use of certain specified lossless compression CODECS. For example, a learning system is employed to determine a difference between what was received by a receiver after compression and transmission and what is predicted to have been received at the transmission end. That difference is processed as learning to modify the tuning of the CODEC for an additional transmission.

Another example of a disclosed method and device is intended to extrapolate past signal-history data for insertion into missing data segments in order to conceal digital speech frame errors. The extrapolation method uses past-signal history that is stored in a buffer. The method is implemented with a device that is disclosed to utilize a finite-impulse response ("FIR"), multi-layer, feed-forward, artificial neural network that is trained by back-propagation for one-step extrapolation of speech compression algorithm ("SCA") parameters. Once a speech connection has been established, the speech compression algorithm device begins sending encoded speech frames. As the speech frames are received, they are decoded and converted back into speech signal voltages. During the normal decoding process, pre-processing of the required SCA parameters will occur and the results stored in the past-history buffer. If a speech frame is detected to be lost or in error, then extrapolation modules are executed and replacement SCA parameters are generated and sent as the parameters required by the SCA. In this way, the information transfer to the SCA is intended to be transparent, and the SCA processing continues as usual. This disclosure alleges that the listener will not normally notice that a

speech frame has been lost because of the smooth transition between the last-received, lost, and next-received speech frames.

Further more detailed examples of systems that are intended to use artificial intelligence and/or neural networks in systems for media compression and/or decompression, generally relating to media type-specific CODEC methods (e.g. speech, video), are variously disclosed in the following U.S. Patent References: 5,005,206 to Naillon et al.; 5,041,916 to Yoshida et al.; 5,184,218 to Gerdes; 5,369,503 to Burel et al.; 5,598,354 to Fang et al.; 5,692,098 to Kurdziel; 5,812,700 to Fang et al.; 5,872,864 to Imade et al.; 5,907,822 to Prieto, Jr.; and 6,216,267 to Mitchell. Still further examples are provided in the following Published International Patent Applications: WO 01/54285 to Rising; EPO 0372608 A1 to Naillon et al.. The disclosures of all these references cited in this paragraph are herein incorporated in their entirety by reference thereto.

Other disclosures of CODEC systems using feedback or other systems for operating CODECS for use in processing a variety of streaming media signals, but that are not believed to specifically use the labels "AI" or "neural networks", are disclosed in the following U.S. Patent Nos.: 6,072,825 to Betts et al.; 6,182,034 B1 to Malvar; 6,253,165 B1 to Malvar; 6,256,608 B1 to Malvar. The disclosures of these references are herein incorporated in their entirety by reference thereto.

Notwithstanding the significant advancements in CODEC algorithms themselves, and despite prior intended uses of AI and other feedback systems for operating CODECS in order to improve compression efficiencies in communication, there is still a need for significant improvement in the ability to efficiently provide a wide variety of streaming media signals to a wide variety of destination receiver devices over a wide variety of transmission channels with varied bandwidths and communication protocols.

There is still a need to incorporate AI and/or neural networks to apply an appropriate CODEC for communication of a streaming media signal based upon a variety of parameters, including without limitation one or more of the following: (a) the automated choosing of an appropriately optimized CODEC from a library of available CODECS of different types and operation, including in particular based upon an intelligent knowledge of the chosen CODEC's operation compared to the other CODEC's operation and/or against a standard, (b) a pre-trained and/or iteratively learned knowledge of the particular CODEC's operation within a given set of operating parameters representative of the existing situation; and (c) a tuning of the appropriate CODEC based upon an intelligent knowledge of

its operation with respect to either or both of the existing situation or a test situation with reference parameters.

In particular, there is still a need for such an intelligent CODEC system that bases an applied CODEC upon an existing situation that is defined by one or more of the following: parameters of the streaming media signal itself; parameters of the transmission channel capabilities and constraints; and parameters of the receiver device capabilities and constraints.

Still further, there is also still a need for such an intelligent CODEC system that operates based upon an intelligent knowledge with respect to all of these operations and situational parameters in order to optimize the appropriate compression, transmission, decompression, and playing of the subject streaming media signal.

Conventional Transcoders for Streaming Media

Also of recent interest in the field of streaming media communication is providing intercommunication between the wide array of "format-specific" encoding systems in present use. An existing field of various different format-specific systems and pre-encoded content has created a widely fragmented ability to process encoded content, resulting in a significant quagmire of compatibility issues between content providers and client users. If one client desires to see or hear streaming content from a particular source and that content must be put through a CODEC for compression, a compatible CODEC must be used on the client side for decompression to enjoy the signal. Unfortunately, source content is often married to only a few, and often only one, specific CODEC schemes. Therefore, if a client requests such encoded content (or if the source desires to push the encoded content to a particular client), one of two criteria must be met: (1) the client must download or otherwise possess the format-specified CODEC (decoder); or (2) the source media must be put through a "transcoder" in order to decode the source media from the first format into a second format that is compatible with the client's device/system. The term "transcoder" is herein intended to mean a system that converts a media signal from one encoded (i.e. compressed) format to another.

Various techniques for transcoding one media format into another have been previously disclosed. Figure 1C shows one illustrative example of the general process that is characteristic of many known transcoding techniques. More specifically, a request 159 is first received from a particular type of device or player for content that exists in an initial, uncompatible format. According to the specific example shown in Figure 1C, a request 159 from a Microsoft Media™ Player for Real™ Video Content is received. As the content is specifically requested, the content is decoded from the

initial format (e.g. Real-encoded format), and is then "re-encoded" into the appropriate format for the requesting player (e.g. Microsoft Media™ format). This re-encoded media is then served to the requesting client for decoding within the resident system of that player.

This conventional system has significant scalability limitations, in that simultaneous feeds on multiple channels for multiple clients must be supported by an equal number of transcoders. For example, Figure 1D shows a schematic implementation of the conventional transcoding technique just described as it manages four simultaneous stream requests from four Microsoft Media Players, wherein the requested content is initially encoded in Real™ format. The system architecture necessary to support the four encoders 151 - 154 and four decoders 155 - 158 requires significant computing resources. For example, it is believed that each encoder 151 - 154 provided in the example requires a computer having a 600 MHz (e.g. Pentium™ III) having 128 Mbytes of RAM available, or dual 400 MHz processors (e.g. Pentium II) with 256 Mb available RAM. It is further believed that each decoder 155 – 158 needs a 233 MHz machine (e.g. Pentium™ II) having 64 Mb of available RAM. So, four such streams requires the equivalent of a Quad 900 Xeon (available from Compaq, Hewlett Packard, Dell and IBM, estimated to cost at the time of this disclosure about $9K retail). This is for four simultaneous streams – society is presently demanding thousands upon thousands of simultaneous streams.

There is still a need for a transcoder system that efficiently converts multiple format-specifically encoded streaming media signals into multiple other formats using minimal computing resources and in a cost-efficient manner.

Parameters Affecting Media Communication

For the purpose of further illustrating the many variables that may impact the choice of an appropriate CODEC in order to communicate a particular streaming media signal to a desired target, the following is a brief summary of various different types of streaming video formats and processing systems. It is believed that these different systems each generally require different types of compression modalities (e.g. CODECS) in order to optimize communication and playing of streaming media signals in view of available transmission speeds and bandwidth, as well as receiver processing parameters.

Although certain specific types of communications formats and systems are further herein described in detail, the following Table 1 provides a summary of a significant cross-section of the

various different communications systems and transmission carriers currently available or disclosed in view of available speed or bandwidth.

TABLE 1: Data Rates of Various Communications Carrier systems

Technology	Speed	Physical Medium	Application
GSM mobile telephone service	9.6 to 14.4 Kbps	RF in space (wireless)	Mobile telephone for business and personal use
High-Speed Circuit-Switched Data service (HSCSD)	Up to 56 Kbps	RF in space (wireless)	Mobile telephone for business and personal use
Regular telephone service (POTS)	Up to 53 Kbps	Twisted pair	Home and small business access
Dedicated 56Kbps on Frame Relay	56 Kbps	Various	Business e-mail with fairly large file attachments
DS0	64 Kbps	All	The base signal on a channel in the set of Digital Signal levels
General Packet Radio System (GPRS)	56 to 114 Kbps	RF in space (wireless)	Mobile telephone for business and personal use
ISDN	BRI: 64 Kbps to 128 Kbps PRI: 23 (T-1) or 30 (E1) assignable 64-Kbps channels plus control channel; up to 1.544 Mbps (T-1) or 2.048 (E1)	BRI: Twisted-pair PRI: T-1 or E1 line	BRI: Faster home and small business access PRI: Medium and large enterprise access
IDSL	128 Kbps	Twisted-pair	Faster home and small business access
AppleTalk	230.4 Kbps	Twisted pair	Local area network for Apple devices; several networks can be bridged; non-Apple devices can also be connected
Enhanced Data GSM Environment (EDGE)	384 Kbps	RF in space (wireless)	Mobile telephone for business and personal use
Satellite	400 Kbps (DirecPC and others)	RF in space (wireless)	Faster home and small enterprise access
Frame relay	56 Kbps to 1.544 Mbps	Twisted-pair or coaxial cable	Large company backbone for LANs to ISP ISP to Internet infrastructure
DS1/T-1	1.544 Mbps	Twisted-pair, coaxial cable, or optical fiber	Large company to ISP ISP to Internet infrastructure
Universal Mobile Telecommunications Service (UMTS)	Up to 2 Mbps	RF in space (wireless)	Mobile telephone for business and personal use (available in 2002 or later)
E-carrier	2.048 Mbps	Twisted-pair, coaxial cable, or optical fiber	32-channel European equivalent of T-1
T-1C (DS1C)	3.152 Mbps	Twisted-pair, coaxial cable, or optical fiber	Large company to ISP ISP to Internet infrastructure
IBM Token Ring/802.5	4 Mbps (also 16 Mbps)	Twisted-pair, coaxial cable, or optical fiber	Second most commonly-used local area network after Ethernet
DS2/T-2	6.312 Mbps	Twisted-pair, coaxial cable, or optical fiber	Large company to ISP ISP to Internet infrastructure
Digital Subscriber Line (DSL)	512 Kbps to 8 Mbps	Twisted-pair (used as a digital, broadband medium)	Home, small business, and enterprise access using existing copper lines
E-2	8.448 Mbps	Twisted-pair, coaxial cable, or optical fiber	Carries four multiplexed E-1 signals
Cable modem	512 Kbps to 52 Mbps (see "Key and explanation" below)	Coaxial cable (usually uses Ethernet); in some systems, telephone used for upstream requests	Home, business, school access
Ethernet	10 Mbps	10BASE-T (twisted-pair); 10BASE-2 or -5 (coaxial cable); 10BASE-F (optical fiber)	Most popular business local area network (LAN)
IBM Token Ring/802.5	16 Mbps (also 4 Mbps)	Twisted-pair, coaxial cable, or optical fiber	Second most commonly-used local area network after Ethernet
E-3	34.368 Mbps	Twisted-pair or optical fiber	Carries 16 E-1 signals
DS3/T-3	44.736 Mbps	Coaxial cable	ISP to Internet infrastructure Smaller links within Internet infrastructure

Aylo Exhibit 1018
Page 1166 of 2308

OC-1	51.84 Mbps	Optical fiber	ISP to Internet infrastructure Smaller links within Internet infrastructure
High-Speed Serial Interface (HSSI)	Up to 53 Mbps	HSSI cable	Between router hardware and WAN lines Short-range (50 feet) interconnection between slower LAN devices and faster WAN lines
Fast Ethernet	100 Mbps	100BASE-T (twisted pair); 100BASE-T (twisted pair); 100BASE-T(optical fiber)	Workstations with 10 Mbps Ethernet cards can plug into a Fast Ethernet LAN
Fiber Distributed-Data Interface (FDDI)	100 Mbps	Optical fiber	Large, wide-range LAN usually in a large company or a larger ISP
T-3D (DS3D)	135 Mbps	Optical fiber	ISP to Internet infrastructure Smaller links within Internet infrastructure
E-4	139.264 Mbps	Optical fiber	Carries 4 E3 channels Up to 1,920 simultaneous voice conversations
OC-3/SDH	155.52 Mbps	Optical fiber	Large company backbone Internet backbone
E-5	565.148 Mbps	Optical fiber	Carries 4 E4 channels Up to 7,680 simultaneous voice conversations
OC-12/STM-4	622.08 Mbps	Optical fiber	Internet backbone
Gigabit Ethernet	1 Gbps	Optical fiber (and "copper" up to 100 meters)	Workstations/networks with 10/100 Mbps Ethernet plug into Gigabit Ethernet switches
OC-24	1.244 Gbps	Optical fiber	Internet backbone
SciNet	2.325 Gbps (15 OC-3 lines)	Optical fiber	Part of the vBNS backbone
OC-48/STM-16	2.488 Gbps	Optical fiber	Internet backbone
OC-192/STM-64	10 Gbps	Optical fiber	Backbone
OC-256	13.271 Gbps	Optical fiber	Backbone

Comments & Key for Table:

(i)The term "Kbps" as the abbreviation for "thousands of bits per second." In international English outside the U.S., the equivalent usage is "kbits s^{-1}" or "kbits/s".
(ii) Engineers use *data rate* rather than *speed*, but speed (as in "Why isn't my Web page getting here faster?") seems more meaningful for the less technically inclined.
(iii) Relative to data transmission, a related term, bandwidth or "capacity," means how wide the pipe is and how quickly the bits can be sent down the channels in the pipe. These "speeds" are aggregate speeds. That is, the data on the multiple signal channels within the carrier is usually allocated by channel for different uses or among different users.

Key: (i) "T" = T-carrier system in U.S., Canada, and Japan....(ii) "DS"= digital signal (that travels on the T-carrier or E-carrier)...(iii) "E" = Equivalent of "T" that uses all 8 bits per channel; used in countries other than U.S. Canada, and Japan....(iv) "OC" = optical carrier (Synchronous Optical Network) "STM" = Synchronous Transport Modules (see Synchronous Digital Heirarchy). (v) Only the most common technologies are shown. (vi) "Physical medium" is stated generally and doesn't specify the classes or numbers of pairs of twisted pair or whether optical fiber is single-mode or multimode. (vii) The effective distance of a technology is not shown. (viii) There are published standards for many of these technologies.

Cable modem note: The upper limit of 52 Mbps on a cable is to an ISP, not currently to an individual PC. Most of today's PCs are limited to an internal design that can accommodate no more than 10 Mbps (although the PCI bus itself carries data at a faster speed). The 52 Mbps cable channel is subdivided among individual users. Obviously, the faster the channel, the fewer channels an ISP will require and the lower the cost to support an individual user.

Internet Carrier Systems

Communication of streaming video via the Internet may take place over a variety of transmission modalities, including for example digital subscriber lines ("DSL"), "T1" lines, cable modem, plain old telephone service ("POTS") dial-up modem, and wireless carriers. While a description of the many different wireless transmission modalities is treated separately herein, a

summary of various of these other transmission modes is herein provided immediately below for the purpose of further illustration as follows.

The terms "POTS" or "plain old telephone service", or "dial-up", as applied to communications transmission channels, are herein interchangeably used. These terms are intended to mean "narrow-band" communication that generally connects end users in homes or small businesses to a telephone company office over copper wires that are wound around each other, or "twisted pair". Traditional phone service was created to let you exchange voice information with other phone users via an analog signal that represents an acoustic analog signal converted into an electrical equivalent in terms of volume (signal amplitude) and pitch (frequency of wave change). Since the telephone company's signaling is already set up for this analog wave transmission, it's easier for it to use that as the way to get information back and forth between your telephone and the telephone company. Therefore, dial-up modems are used to demodulate the analog signal and turn its values into the string of 0 and 1 values that is called digital information. Because analog transmission only uses a small portion of the available amount of information that could be transmitted over copper wires, the maximum amount of data that you can receive using ordinary modems is about 56 Kbps. The ability of your computer to receive information is constrained by the fact that the telephone company filters information that arrives as digital data, puts it into analog form for your telephone line, and requires your modem to change it back into digital. In other words, the analog transmission between your home or business and the phone company is a bandwidth bottleneck.

With "ISDN", or "Internet subscriber digital network", which some consider to be a limited precursor to DSL, incoming data rates up to about 128 Kbps may be achieved for some end user clients.

A "DSL" or "digital subscriber line" is generally defined as a "broadband" transmission carrier for communicating high-bandwidth communication over ordinary copper telephone lines. Many different types of DSL services have been disclosed, having generally varied data rates and intended applications. Though further discussion is herein provided about certain of these DSL types, the following Table 2 provides a summary of information for certain of these DSL types for the purpose of further developing an overview understanding:

TABLE 2: Types of known DSL services.

DSL Type	Description	Data Rate Downstream; Upstream	Distance Limit	Application
IDSL	ISDN Digital Subscriber Line	128 Kbps	18,000 feet on 24 gauge wire	Similar to the ISDN BRI service but data only (no voice on the same line)
CDSL	Consumer DSL from Rockwell	1 Mbps downstream; less upstream	18,000 feet on 24 gauge wire	Splitterless home and small business service; similar to DSL Lite
DSL Lite (same as G.Lite)	"Splitterless" DSL without the "truck roll"	From 1.544 Mbps to 6 Mbps downstream, depending on the subscribed service	18,000 feet on 24 gauge wire	The standard ADSL; sacrifices speed for not having to install a splitter at the user's home or business
G.Lite (same as DSL Lite)	"Splitterless" DSL without the "truck roll"	From 1.544 Mbps to 6 Mbps , depending on the subscribed service	18,000 feet on 24 gauge wire	The standard ADSL; sacrifices speed for not having to install a splitter at the user's home or business
HDSL	High bit-rate Digital Subscriber Line	1.544 Mbps duplex on two twisted-pair lines; 2.048 Mbps duplex on three twisted-pair lines	12,000 feet on 24 gauge wire	T1/E1 service between server and phone company or within a company; WAN, LAN, server access
SDSL	Symmetric DSL	1.544 Mbps duplex (U.S. and Canada); 2.048 Mbps (Europe) on a single duplex line downstream and upstream	12,000 feet on 24 gauge wire	Same as for HDSL but requiring only one line of twisted-pair
ADSL	Asymmetric Digital Subscriber Line	1.544 to 6.1 Mbps downstream; 16 to 640 Kbps upstream	1.544 Mbps at 18,000 feet; 2.048 Mbps at 16,000 feet; 6.312 Mpbs at 12,000 feet; 8.448 Mbps at 9,000 feet	Used for Internet and Web access, motion video, video on demand, remote LAN access
RADSL	Rate-Adaptive DSL from Westell	Adapted to the line, 640 Kbps to 2.2 Mbps downstream; 272 Kbps to 1.088 Mbps upstream	Not provided	Similar to ADSL
UDSL	Unidirectional DSL proposed by a company in Europe	Not known	Not known	Similar to HDSL
VDSL	Very high Digital Subscriber Line	12.9 to 52.8 Mbps downstream; 1.5 to 2.3 Mbps upstream; 1.6 Mbps to 2.3 Mbps downstream	4,500 feet at 12.96 Mbps; 3,000 feet at 25.82 Mbps; 1,000 feet at 51.84 Mbps	ATM networks; Fiber to the Neighborhood

Typically published data rates for DSL service, which may vary depending upon distance from the central office of the offering service company, includes rates up to 6.1 Mbps (theoretically published at 8.448 Mbps), which is believed to enable continuous transmission of motion video, audio, and 3-D effects. More typical individual connections provide from 512 Kbps to 1.544 Mbps downstream and about 128 Kbps upstream. A DSL line can carry both data and voice signals and the data part of the line is continuously connected. DSL has been anticipated in some publications to replace ISDN in many areas and to compete with cable modem for multimedia communication to homes and businesses. DSL operates purely within the digital domain and does not require change into analog form and back. Digital data is transmitted to destination computers directly as digital data and this allows the phone company to use a much wider bandwidth for forward transmission. Meanwhile, if

Aylo Exhibit 1018
Page 1169 of 2308

a client user chooses, the signal can be separated so that some of the bandwidth is used to transmit an analog signal so that a telephone and computer may be used on the same line and at the same time.

Most DSL technologies require that a signal splitter be installed at a home or business, requiring the expense of a phone company visit and installation. However, it is possible to manage the splitting remotely from the central office. This is known as splitterless DSL, "DSL Lite," G.Lite, or Universal ADSL (further defined below) and has recently been made a standard. Several modulation technologies are used by various kinds of DSL, although these are being standardized by the International Telecommunication Union (ITU). Different DSL modem makers are using either Discrete Multitone Technology (DMT) or Carrierless Amplitude Modulation (CAP). A third technology, known as Multiple Virtual Line (MVL), is another possibility.

A variety of parameters of DSL operation are variable and affect the effective data rates that can be achieved. DSL modems generally follow the data rate multiples established by North American and European standards. In general, the maximum range for DSL without a repeater is 5.5 km (18,000 feet). As distance decreases toward the telephone company office, the data rate increases. Another factor is the gauge of the copper wire. The heavier 24 gauge wire carries the same data rate farther than 26 gauge wire. For destination devices beyond the 5.5 kilometer range, DSL may still be provided, though only generally if the respective phone company provider has extended the local loop with optical fiber cable.

To interconnect multiple DSL users to a high-speed network as a "backbone", the telephone company uses a Digital Subscriber Line Access Multiplexer ("DSLAM"). Typically, the DSLAM connects to an asynchronous transfer mode ("ATM") network that can aggregate data transmission at gigabit data rates. At the other end of each transmission, a DSLAM demultiplexes the signals and forwards them to appropriate individual DSL connections.

"ADSL" or "Asymmetric Digital Subscriber Line" is the form of DSL that will become most familiar to home and small business users. ADSL is called "asymmetric" because most of its two-way or "duplex" bandwidth is devoted to the downstream direction, sending data to the user. Only a small portion of bandwidth is available for upstream or user-interaction messages. However, most Internet and especially graphics- or multi-media intensive Web data need lots of downstream bandwidth, but user requests and responses are small and require little upstream bandwidth. Using ADSL, up to 6.1 megabits per second of data can be sent downstream and up to 640 Kbps upstream. The high downstream bandwidth means that a telephone line may carry motion video, audio, and 3-D images to destination computers or television displays. In addition, a small portion of the downstream bandwidth

can be devoted to voice rather data, and phone conversations may be carried without requiring a separate line. Unlike a similar service over "cable" television lines, ADSL does not compete for bandwidth with neighbors in a given area. In many cases, your existing telephone lines will work with ADSL. In some areas, they may need upgrading.

"CDSL" or "Consumer DSL" is a trademarked version of DSL, to be made available by Rockwell Corporation, that is somewhat slower than ADSL (1 Mbps downstream, generally predicted to be lower upstream) but has the advantage that a "splitter" does not need to be installed at the user's end. Hardware may be required to carry CDSL by local phone companies to homes or businesses. CDSL uses its own carrier technology rather than DMT or CAP ADSL technology.

Various companies have worked with telephone companies in developing a standard and easier installation version of ADSL, called "G.Lite", that is believed to be under deployment at the time of this disclosure. "G.Lite" or "DSL Lite" (also known as "splitterless ADSL", and "Universal ADSL") is believed to be essentially a slower ADSL that doesn't require splitting of the line at the user end but manages to split it for the user remotely at the telephone company, which is believed to lower costs. G.Lite, officially ITU-T standard G-992.2, is published to provide a data rate from 1.544 Mbps to 6 Mpbs downstream and from about 128 Kbps to about 384 Kbps upstream. At least one publication has predicted G.Lite to become the most widely installed form of DSL.

"HDSL" or "High bit-rate DSL" is believed to be the earliest variation of DSL to be widely used for wideband digital transmission within a corporate site and between the telephone company and a customer. The main characteristic of HDSL is that it is symmetrical: an equal amount of bandwidth is available in both directions. For this reason, the maximum data rate is generally lower than for ADSL. HDSL can carry as much on a single wire of twisted-pair as can be carried on a T1 line in North America or an E1 line in Europe (up to about 2.32 Mbps).

"IDSL" or "ISDN DSL" is somewhat of a misnomer since it's really closer to ISDN data rates and service at about 128 Kbps than compared with the much higher rates generally associated with ADSL.

"RADSL" or "Rate-Adaptive DSL" is an ADSL technology to be made available from Westell company in which software is able to determine the rate at which signals can be transmitted on a given customer phone line and adjust the delivery rate accordingly. Westell's "FlexCap2™" version system uses RADSL to deliver from about 640 Kbps to about 2.2 Mbps downstream and from about 272 Kbps to about 1.088 Mbps upstream over an existing line.

"SDSL" or "Symmetric DSL" is similar to HDSL with a single twisted-pair line, carrying about 1.544 Mbps (U.S. and Canada) or about 2.048 Mbps (Europe) each direction on a duplex line. It's symmetric because the data rate is the same in both directions.

"UDSL" or "Unidirectional DSL" is a proposal from a European company, and is generally believed to provide a unidirectional version of HDSL.

"VDSL" or "Very high data rate DSL" is believed to be a technology under development that promises much higher data rates over relatively short distances, for example between about 51 and about 55 Mbps over lines up to about 1,000 feet or about 300 meters in length. At least one publication has predicted that VDSL may emerge somewhat after ADSL is widely deployed and co-exist with it. The transmission technology (CAP, DMT, or other) and its effectiveness in some environments is not yet determined. A number of standards organizations are working on it.

"x2/DSL" is modem from 3Com that supports 56 Kbps modem communication but is upgradeable through new software installation to ADSL when it becomes available in the user's area. At least one publication cites 3Com as describing this technology to be "the last modem you will ever need."

A "T1" transmission line is generally considered a "broadband" carrier and is defined as a type of "T-carrier" system, which is believed to have been first introduced by the Bell System in the U.S. in the 1960's as the first successful system that supported digitized voice transmission. The T-carrier system is entirely digital, using pulse code modulation and time-division multiplexing. Voice signals are typically sampled at about 8,000 times a second and each sample is digitized into an 8-bit word. With 24 channels digitized at the same time, a 192-bit frame, representing 8-bit words on each of 24 channels, is thus transmitted about 8,000 times a second. Each frame is separated from the next by a single bit, resulting in a 193-bit block. The T-1's published data rate of 1.544 Mbps generally represents the 192 bit frame, and the 1-bit signaling bit, multiplied by 8,000.

A T-1 system typically uses 4 wires and provides duplex capability, with two wires dedicated for receiving and two for sending at the same time. The T-1 digital stream includes 24, 64 Kbps channels that are multiplexed, wherein the standard 64 Kbps channel is based on the bandwidth required for a voice conversation. The four wires were originally a pair of twisted pair copper wires, but more recent systems provide coaxial cable, optical fiber, digital microwave, and other carrier technologies. The number and use of the channels may be varied from the standard guidelines.

The original transmission rate (1.544 Mbps) for T-1 lines is in common use today in Internet service provider ("ISP") connections to the Internet. Another level, the T-3 line, is published to

provide 44.736 Mbps, and is also commonly used by Internet service providers. Another commonly used service is "fractional T-1", which is the rental of some portion of the 24 channels in a T-1 line, with the other channels unused.

Display Capabilities/Constraints & Related Standards

Various different types of receiver display capabilities may also significantly impact the appropriate CODEC modality for efficiently communicating particular streaming media signals for display by the receiver. A brief summary of certain examples to illustrate such varied display parameters (e.g. resolution, clarity, color, depth, size, type/format-specific) is provided for a better understanding as follows.

One parameter that is highly variable between different types and makes of streaming media receiver devices, and therefore that may have significant impact on the appropriate CODEC to be used, is the range of colors that may be expressed by a display device, or "palette". A standard "browser-safe" palette, which may be accommodated by most software for Internet-based streaming media display, may include for example about 216 colors, though for web-based streaming media the computer display capability as well as the browser software capability must be understood.

With respect to computer display technology, a color is set for each individual pixel or addressable illumination element on the screen. Each pixel has a red, green, and blue (RGB) component. By specifying the amount of intensity for each of these components, a distinct color is given to that pixel. A "true color" display generally defines the color of a pixel on a display screen using a 24-bit value, allowing the possibility of up to 16,777,216 possible colors. The number of bits used to define a pixel's color shade is called the "bit-depth". True color is sometimes referred to as "24-bit color", though many modern color display systems offer a 32-bit color mode. An extra byte, called the "alpha channel", is typically used for control and special effects information. A "gray scale" (composed of discrete shades of gray) display setting is generally defined as having N bits of depth where N represents the saturation of black within the pixel. If N=1, the image is not called gray scale but instead monochrome, or black and white, as the bit can only be on or off and can contain no shading information.

Common computer resolutions include for example and without limitation the following:

(i) VGA or Video Graphics Array capable of displaying 640x480 pixels in 16 colors or 320x240 pixels in 256 colors in a 4:3 aspect ratio;

(ii) SVGA or Super Video Graphics Array capable of 800x600x6 bits/pixel (16 colors) or 650x480x8 bits/pixel (256 colors). SVGA was created by the Video Electronics Association (VESA); and

(iii) XGA (v1-4) or eXtended Graphics Array capable of 1024x768 pixels at 32,768 colors.

Additional standards have been added such as SXGA, defining pixel sizes above 1960x1440 and color depths of 32 bits/pixel and higher.

In the event that a larger range of colors (or palette) is used by a media signal than a particular display or browser can handle, most browsers are typically adapted to "dither" the colors, which is herein intended to mean that the browser will find colors within its palette that it can substitute for any color that is outside of its palette. To further illustrate the wide range of different system display capabilities, systems using WindowsTM (commercially available from Microsoft Corporation) and MacintoshTM (commercially available from Apple Corporation) based operating systems do not have identical palettes; within the usual 256 color palette, 216 are common to both types of browsers, whereas 40 are different and therefore require dithering by a browser operating within one of the systems if an image signal is communicated to that type of system in a format specified by the other.

Many different technologies also exist with respect to how a visual display is enabled from electronic information. The terms "VDT" or "Video Display Terminals" are generally used within the computer industry and are herein intended to be used interchangeably with simple references to "display". With respect to computer terminal use, VDT's comprise a computer output surface and projecting mechanism that shows text and graphic images to the computer user. VDT's may use a variety of specific display technologies, including for example cathode ray tubes ("CRTs"), liquid crystal displays ("LCDs"), light-emitting diodes ("LEDs"), gas plasma, or other image projection technology. The display is usually considered to include the screen or projection surface and the device that produces the information on the screen. In some computers, the display is packaged in a separate unit or "monitor", or the display may be fully integrated in a single unit with the computer processor.

With respect to LCD's in particular, this technology generally requires minimal volume and physical depth compared to other VDT's, and therefore is typically used in laptop computers and cellphone/PDA's. LCD's consume much less power than LED and gas–display VDT's because they work on the principle generally of blocking light rather than emitting it. An LCD may be either

"passive matrix" or "active matrix", which is also known as "thin film transistor" or "TFT" display. The passive matrix LCD has a grid of conductors with pixels located at each intersection in the grid. A current is sent across two conductors on the grid to control the light for any pixel. An active matrix has a transistor located at each pixel intersection, requiring less current to control the luminance of a pixel. For this reason, the current in an active matrix display can be switched on and off more frequently, improving the screen refresh time and therefore efficacy for higher speeds of streaming media (e.g. action video). Some passive matrix LCD's have dual scanning, in that they scan the grid twice with current in the same time as one scan in earlier versions; however, the active matrix is still generally considered to be the superior technology of the two. Reflective color display technology—the integration of color filters into passive-matrix display construction—is a low-power, low-cost alternative to active-matrix technology. Because they reflect ambient light, reflective LCDs deliver particularly high performance during use outside in daylight. Various different display technologies, and therefore transmission formats, have also been specifically developed for television viewing. Thus several different standards have evolved for television transmission, and their differences may significantly impact the nature and extent of compression desired (and therefore the choice of a particular CODEC) for communicating streaming media signals in television environs. These standards include in particular and without limitation: standard definition television ("SDTV"); and high definition television ("HDTV").

"SDTV" or "standard definition television" and "HDTV" or "high definition television" are the two categories of display formats for digital television ("DTV") transmissions, which are becoming the standard. These formats provide a picture quality similar to digital versatile disk ("DVD"), and are summarized relative to their similarities and differences as follows.

HDTV provides a higher quality display, with a vertical resolution display from about 720p to at least about 1080i and an aspect ratio (the width to height ratio of the screen) of generally 16:9, for a viewing experience similar to watching a movie. In comparison, SDTV has a range of lower resolutions and no defined aspect ratio. New television sets will be either HDTV-capable or SDTV-capable, with receivers that can convert the signal to their native display format. SDTV, in common with HDTV, using the MPEG-2 file compression method in a manner that generally reduces a digital signal from about 166 Mbps to about 3 Mbps. This allows broadcasters to transmit digital signals using existing cable, satellite, and terrestrial systems. MPEG-2 uses the lossy compression method, which means that the digital signal sent to the television is compressed and some data is lost, but this lost data may or may not affect how the human eye views the picture. Both the ATSC and DVB standards selected

MPEG-2 for video compression and transport. The MPEG-2 compression standard is elsewhere herein described in further detail.

Because a compressed SDTV digital signal is smaller than a compressed HDTV signal, broadcasters can transmit up to five SDTV programs simultaneously instead of just one HDTV program, otherwise known as "multicasting". Multicasting is an attractive feature because television stations can receive additional revenue from the additional advertising these extra programs provide. With today's analog television system, only one program at a time can be transmitted. Note that this use of the term "multicasting" is distinct from its use in streaming video where it involves using special addressing techniques.

When the United States decided to make the transition from analog television to DTV, the Federal Communications Commission decided to let broadcasters decide whether to broadcast SDTV or HDTV programs. Most have decided to broadcast SDTV programs in the daytime and to broadcast HDTV programs during prime time broadcasting. Both SDTV and HDTV are supported by the Digital Video Broadcasting (DTV) and Advanced Television Systems Committee (ATSC) set of standards.

HDTV as a television display technology provides picture quality similar to 35 mm. movies with sound quality similar to that of today's compact disc (further with respect to audio quality, HDTV receives, reproduces, and outputs Dolby Digital 5.1). Some television stations have begun transmitting HDTV broadcasts to users on a limited number of channels. HDTV generally uses digital rather than analog signal transmission. However, in Japan, the first analog HDTV program was broadcast on June 3, 1989. The first image to appear was the Statue of Liberty and the New York Harbor. It required a 20 Mhz channel, which is why analog HDTV broadcasting is not feasible in most countries.

HDTV provides a higher quality display than SDTV, with a vertical resolution display from 720p to 1080i. The *p* stands for *progressive scanning*, which means that each scan includes every line for a complete picture, and the *i* stands for *interlaced scanning* which means that each scan includes alternate lines for half a picture. These rates translate into a frame rate of up to 60 frames per second, twice that of conventional television. One of HDTV's most prominent features is its wider aspect ratio (the width to height ratio of the screen) of 16:9, a development based on a research-based belief that the viewer's experience is enhanced by screens that are wider. HDTV pixel numbers range from one to two million, compared to SDTV's range of 300,000 to one million. New television sets will be either HDTV-capable or SDTV-capable, with receivers that can convert the signal to their native display format.

In the United States, the FCC has assigned broadcast channels for DTV transmissions. In SDTV formats, DTV makes it possible to use the designated channels for multiple signals at current quality levels instead of single signals at HDTV levels, which would allow more programming with the same bandwidth usage. Commercial and public broadcast stations are currently deciding exactly how they will implement their use of HDTV.

Simulcast is the simultaneous transmission of the same television program in both an analog and a digital version using two different channels or frequencies. At the end of the DTV transition period, it is believed by that analog transmission will be substantially replaced such that current analog channels will be used solely for DTV. The extra channels that were used for digital broadcasting may for example then be auctioned and used for more television channels or other services such as datacasting. Simulcast is also used for the transmission of simultaneous television and Internet services, the transmission of analog and digital radio broadcasts, and the transmission of television programs in different screen formats such as the traditional format and the wide screen format. Simulcast broadcasting is used worldwide.

The transition to DTV is not an easy or inexpensive transition. For a television station to transmit DTV programming, it must build its DTV facilities, but a station must have revenue to build these facilities. Simulcast allows stations to continue receiving revenues from traditional analog programming and also gain extra revenues from the extra digital programming. Another obstacle in the transition to DTV is lack of interest among consumers. The need for special equipment is prohibiting viewers from seeing the difference between digital and analog programs, which is also slowing down public enthusiasm for DTV.

The equipment needed for operating DTV depends on whether terrestrial, cable, or satellite services are used as the transmission channel/carrier. In any event, and according to known or anticipated systems, it is generally believed that consumers will, at a minimum, have to purchase a converter to view DTV transmissions on their old television sets. In addition, consumers that use terrestrial services or antennas to receive television signals need an antenna equipped for digital signals. A consumer located in mountainous terrain in an ATSC-compliant country may not be able to receive terrestrial-based digital signals because of multipath effects. This is common even with today's analog television system. In DVB compliant countries, terrain does not affect the reception of digital signals. Satellite users are already enjoying DTV broadcasting, but a larger satellite dish might be needed to view HDTV programming.

Aylo Exhibit 1018
Page 1177 of 2308

A "set-top" box is herein defined as a device that enables a television set to become a user interface to the Internet and also enables an analog television set to receive and decode DTV broadcasts. DTV set-top boxes are sometimes called receivers. It is estimated that 35 million homes will use digital set-top boxes by the end of 2006, the estimated year ending the transition to DTV.

A typical digital set-top box contains one or more microprocessors for running the operating system, usually Linux or Windows CE, and for parsing the MPEG transport stream. A set-top box also includes RAM, an MPEG decoder chip, and more chips for audio decoding and processing. The contents of a set-top box depend on the DTV standard used. DVB-compliant set-top boxes contain parts to decode COFDM transmissions while ATSC-compliant set-top boxes contain parts to decode VSB transmissions. More sophisticated set-top boxes contain a hard drive for storing recorded television broadcasts, for storing downloaded software, and for other applications provided by the DTV service provider. Digital set-top boxes can be used for satellite and terrestrial DTV but are used mostly for cable television. A set-top box price ranges from $100 for basic features to over $1,000 for a more sophisticated box.

In the Internet realm, a set-top box often really functions as a specialized computer that can "talk to" the Internet - that is, it contains a Web browser (which is really a Hypertext Transfer Protocol client) and the Internet's main program, TCP/IP. The service to which the set-top box is attached may be through a telephone line as, for example, with Web TV or through a cable TV company like TCI.

To take advantage of Dolby Digital 5.1 channel for satellite broadcasts, a satellite receiver that provides a Dolby Digital output is necessary. For cable users, all digital set-top boxes are equipped with a Dolby Digital two-channel decoder. To use 5.1 channel sound, a 5.1 channel-compliant set-top box is needed or an external 5.1 channel decoder unit.

The most dramatic demonstration of digital television's benefits is through a high-end HDTV, because of the larger screen, wider aspect ratio and better resolution. Like most new technologies, however, HDTV is expensive. Nevertheless, less expensive digital TVs provide a markedly improved viewing experience over regular TV, and for those who choose to retain their old sets, even the addition of a set-top converter will deliver a discernibly improved picture and sound.

The FCC's schedule for transition to DTV proposes that everyone in the U.S. should have access to DTV by 2002 and that the switch to digital transmission must be completed either by 2006 or when 85% of the households in a specific area have purchased digital television sets or set-top converters.

Aylo Exhibit 1018
Page 1178 of 2308

In the early 1990s, European broadcasters, consumer equipment manufacturers, and regulatory bodies formed the European Launching Group (ELG) which launched a "DVB" or "Digital Video Broadcasting" project in order to introduce DTV throughout Europe. DVB is intended to provide an open system as opposed to a closed system. Closed systems are content-provider specific, not expandable, and optimized only for the system they were developed for. An open system, such as DVB, allows the subscriber to choose different content providers and allows integration of PCs and televisions. DVB systems are intended to be optimized for television, but as well as supporting home shopping and banking, private network broadcasting, and interactive viewing. DVB is intended to open the possibilities of providing crystal-clear television programming to television sets in buses, cars, trains, and even hand-held televisions. DVB is also promoted as being beneficial to content providers because they can offer their services anywhere DVB is supported regardless of geographic location. They can also expand their services easily and inexpensively and ensure restricted access to subscribers reducing lost revenues due to unauthorized viewing. Today, the DVB Project consists of over 220 organizations in more than 29 countries worldwide and DVB broadcast services are available in Europe, Africa, Asia, Australia, and parts of North and South America

Format-Specific Media

Various different formats for the streaming media signals themselves are also herein summarized by way of non-limiting example to also provide a further understanding of how CODECS may vary for a particular case.

"DVD" is an acronym for "digital versatile disc" is generally defined as a relatively recent optical disc technology that holds up to about 4.7 Gigabytes of information on one of its two sides, or enough for a movie about 133 minutes long on average. With two layers on each of its two sides, it may hold up to 17 Gigabytes of video, audio, or other information, compared to current CD-ROM discs of approximately the same physical size that hold about 600Mbytes (DVD holds more than about 28 times the information). DVD players are required to play DVD's, though they will also play regular CD-ROM discs. DVDs can be recorded in any of three general formats variously optimized for: (i) video (e.g. continuous movies); (ii) audio (e.g. long playing music); and (iii) or a mixture (e.g. interactive multimedia presentations). The DVD drive has a transfer rate somewhat faster than an 8-speed CD-ROM player. DVD format typically uses the MPEG-2 file and compression standard, which has about 4-times the resolution of MPEG-1 images and can be delivered at about 60 interlaced fields

per second where two fields constitute one image (MPEG-1 delvers about 30 non-interlaced frames per second. MPEG-2 and -1 standards are elsewhere herein defined in more detail. Audio quality on DVD is comparable to that of current audio compact discs.

"DVD-Video" is the name typically given for the DVD format designed for full-length movies and is a box that will work with a television set. "DVD-ROM" is a name given to the player that is believed by some to be the future replacement to CD-ROM drives in computers, as these newer drives are intended to play both regular CD-ROM discs as well as DVD-ROM discs. "DVD-RAM" is the name given to writeable versions of DVDs. "DVD-Audio" is the name typically given to players designed to replace the compact disc player.

"VHS" is an acronym for "Video Home System" and is generally defined as a magnetic videotape cartridge format, typically a half-inch wide, developed for home use with the ability to record and playback analog video and audio signals. VHS has become a popular format and the de facto standard for home movie distribution and reproduction mainly due to its pervasive presence and recordability. VHS stores signals as an analog format on a magnetic tape using technology similar to that of audiocassettes. The tapes are played back and recorded on using VHS video cassette recorders (VHS VCRs). VHS tapes store up to around two hours of video typically, although some VCRs are able to record to them at a slower speed allowing up to six or even eight hours of recording per tape.

The VHS format outputs a little over 200 lines of horizontal resolution. This compares to DVDs that output over 500 lines of horizontal resolution. Technically and perceptually, VHS is a format that has been surpassed by other formats, including for example DVD, S-VHS, Hi-8 and others. However, VHS remains a pervasive means for viewing video, and VHS tapes are still easily found across the country and around the world everywhere from movie rental stores to grocery stores making then easily accessible.

"CD" is an acronym for "compact disc" and is generally defined as small, portable, round medium for electronically recording, storing, and/or playing audio, video, text, and other information in digital form. Initially, CD's were only read-only; however, newer versions allow for recording as well (e.g. "CD-RW").

"Super audio compact disc" or "SACD" is a high resolution audio CD format that, together with DVD-Audio ("DVD-A"), are the two formats competing to replace the standard audio CD (though most of the industry generally is backing DVD-A, with a general exception believed to be Philips and Sony). SACD, like DVD-A, offers 5.1 channel surround sound in addition to 2-channel stereo. Both formats improve the complexity of sound by increasing the bit rate and the sample rate, and can be

played on existing CD players, although generally only at quality levels similar to those of traditional CDs. SACD uses Direct Stream Digital ("DSD") recording, which is published as being proprietary to Sony, that converts an analog waveform to a 1-bit signal for direct recording, instead of the pulse code modulation ("PCM") and filtering used by standard CDs. DSD uses lossless compression and a sampling rate of about 2.8 MHz to improve the complexity and realism of sound. SACD may also include additional information, such as text, graphics, and video clips.

Also for the purpose of further understanding, Internet-based communications also have particular protocols for communication that must be accommodated by a streaming media communications system using the Internet "superhighway". These protocols, in particular with respect to streaming media communication, are briefly summarized immediately below for the purpose of providing a more detailed understanding.

With respect to Internet communication, streaming media signals are generally communicated in digital format via data packets. The terms "packets" are herein intended to mean units of data that are routed between an origin and a destination via the Internet or any other packet-switched network. More specifically, when a file is sent, the protocol layer of the communications system (e.g. TCP layer of TCP/IP based system) divides the file into chunks of efficient size for routing. Each of these packets is separately numbered and includes the Internet address of the destination. The individual packets for a given file may travel different routs through the Internet. When they have all arrived, they are reassembled into the original file, for example by the TCP layer at the receiving end. A packet-switching scheme is an efficient way to handle transmissions on a connectionless network such as the Internet. An alternative scheme, circuit-switched, is used for networks allocated generally for voice connections. In circuit-switching, lines in the network are shared among many users as with packet-switching, but each connection generally requires the dedication of a particular path for the duration of the connection.

Wireless Communications & the WAP Gateway

Of equal importance to the contemporary age of the Internet, the age of wireless communications has significantly extended society's ability to interact outside of the fixed confines of the home and office, allowing our remote communications to break free from the umbilical cords of wires and cables. For example, in 2000, the number of mobile subscribers grew by close to 50%.

However, wireless communications systems, protocols, and enabling technologies have developed in a significantly fragmented, "format-specific" market on a world-wide scale. This is particularly true in comparing systems in wide use in the United States as compared to the rest of the world. Therefore, much effort has been expended in overcoming compatibility issues between format-specific systems and between the related wireless devices operating on different platforms. For the purpose of further understanding wireless communication as it is later related to the present invention, the following is a brief overview of significant technologies, systems, and protocols used in the wireless communication industry.

In general, the progression of wireless communications systems for cellular telephones is colloquially given the terms "1G", "2G", "2.5G", and "3G", representing respectively first generation, second generation, and so-on. Initial systems were purely analog, known as the 1G phones and systems. However, with rapid growth, available bandwidth for cellular phone use quickly eroded, giving way to digital signal processing in the 2G, which significantly widened the available bandwidth and ability for complex signal processing for advanced telecommunications. However, as demand progressed for wireless Internet access, so went the technology development from 2G phones (generally not Internet enabled), to 2.5G and 3G (progressively more enabled). As will be further developed immediately below, the systems, protocols, and enabling technologies thus have developed toward a concentrated focus in bringing the 2.5G and 3G modes to industry and consumers.

In general, there are four major digital wireless networks based upon 2G technology: time division multiple access ("TDMA"), code division multiple access ("CDMA"), Global System for Mobile communication ("GSM") and cellular digital packet data ("CDPD"). These are briefly herein described as follows.

Time division multiple access ("TDMA") is a technology used in digital cellular telephone communication that divides each cellular channel into three time slots in order to increase the amount of data that can be carried. TDMA is used by Digital-American Mobile Phone Service (D-AMPS), Global System for Mobile communication ("GSM"), and Personal Digital Cellular ("PDC). However, each of these systems implements TDMA in a somewhat different and incompatible way. An alternative multiplexing scheme to TDMA and FDMA (frequency division multiple access) is code division multiple access ("CDMA").

Code division multiple access ("CDMA") refers to any of several protocols used in 2G and 3G wireless communications. As the term implies, CDMA is a form of multiplexing that allows numerous signals to occupy a single transmission channel, optimizing the use of available bandwidth. the

technology is used in ultra-high-frequency (UHF) cellular telephone systems in the 800 MHz to 1.9 GHz bands. CDMA uses analog-to-digital conversion (ADC) in combination with spread spectrum technology. Audio input is first digitized into binary elements. The frequency of the transmitted signal is then made to vary according to a defined pattern (code), so it can be intercepted only by a receiver whose frequency response is programmed with the same code, so it follows exactly along with the transmitter frequency. There are trillions of possible frequency-sequencing codes, thus enhancing privacy and making cloning difficult. The CDMA channel is nominally 1.23 MHz wide. CDMA networks use a scheme called "soft handoff", which minimizes signal breakup as a handset passes from one cell to another. The combination of digital and spread spectrum modes supports several times as many signals per unit bandwidth as analog modes. CDMA is compatible with other cellular technologies; this allows for nationwide roaming.

The original CDMA, also known as CDMA One, was standardized in 1993 and is considered a 2G technology that is still common in cellular telephones in the U.S. One version of cdmaOne, IS-95A, is a protocol that employs a 1.25 MHz carrier and operates in RF bands at either 800 MHz or 1.9 GHz; this supports data rates of up to 14.4 Kbps. Another version, IS-95B, is capable of supporting speeds of up to 115 Kbps by bundling up to eight channels.

More recent CDMA varieties, CDMA2000 and wideband CDMA offer data speeds many times faster. CDMA2000, also known as IMT-CDMA Multi-Carrier or IS-136, is a CDMA version of the IMT-2000 standard developed by the International Telecommunications Union (ITU). The CDMA2000 standard is a 3G technology that is intended to support data communications at speeds ranging from 144Kbps to 2 Mbps. Companies that have developed versions of this standard include Ericsson and Qualcomm corporations. Wideband CDMA, or "WCDMA", is an ITU standard derived from CDMA that is also known as IMT-2000 direct spread. WCDMA is a 3G technology intended to support data rates of up to 2Mbps for local area access, or 384 Kbps for wide area access, and supports mobile/portable voice, images, data, and video communications at these speeds. WCDMA digitizes input signals and transmits the digitized output in coded, spread-spectrum mode over a 5 MHz wide carrier – a much broader range than the 200 KHz wide narrowband CDMA.

The Global System for Mobile communication ("GSM") is a digital mobile telephone system that is widely used in Europe and other parts of the world; this system uses a variation of "TDMA" (introduced immediately below) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes, compresses, and then sends data down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 Mhz

or 1800 MHz frequency band. At the time of this disclosure, GSM is generally considered the wireless telephone standard in Europe, and has been published to have over 120 million users worldwide and is available in 120 countries. At least one company in the United States, American Personal Communications (SprintTM subsidiary), is using GSM as the technology for a broadband personal communications services ("PCS"). PCS are telecommunications services that bundle voice communications, numeric and text messaging, voice-mail and various other features into one device, service contract and bill. PCS are most often carried over digital cellular links. This service is planned to have more than 400 base stations for various compact mobile handsets that are being made by manufacturers such as Ericsson, Motorola, and Nokia corporations; these devices generally include a phone, text pager, and answering machine. GSM is part of an evolution of wireless mobile telecommunications that includes High-Speed Circuit-Switched Data (HCSD), General Packet Radio System (GPRS), Enhanced Data GSM Environment (EDGE), and Universal Mobile Telecommunications Service (UMTS).

Cellular Digital Packet Data ("CDPD") is a wireless standard providing two-way, 19.2 kbps packet data transmission over existing cellular telephone channels.

Several different protocols have also been put into use for communicating over the various wireless networks. Various specific such protocols are briefly introduced as follows.

"X.25" is a packet-based protocol, principally used at the time of this disclosure in Europe and adapted as a standard by the Consultative Committee for International telegraph and Telephone (CCITT). X.25 is a commonly used network protocol that allows computers on different public networks (e.g. CompuServe, Tymnet, or TCP/IP network) to communicate through an intermediary computer at the network layer level. X.25's protocols correspond closely to the data-link and physical-layer protocols defined in the Open Systems Interconnection ("OSI").

"OSI" is a model of network architecture and a suite of protocols (a protocol stack) to implement it, developed by ISO in 1978 as a framework for international standards in heterogeneous computer network architecture. The OSI architecture is split between seven layers, from lowest to highest: (1) physical layer; (2) data link layer; (3) network layer; (4) transport layer; (5) session layer; (6) presentation layer; and (7) application layer. Each layer uses the layer immediately below it and provides a service to the layer above. In some implementations, a layer may itself be composed of sub-layers.

General Packet Radio Services ("GPRS") is a packet-based wireless communication service that promises data rates from 56 to 114 Kbps and continuous connection to the Internet for mobile phone

and computer users. The higher data rates will allow users to take part in video conferences and interact with multimedia Web sites and similar applications using mobile handheld devices as well as notebook computers. GPRS is based on Global System for Mobile ("GSM") communication and will complement existing services such as circuit-switched cellular phone connections and the Short Message Service ("SMS"). SMS is a message service offered by the GSM digital cellular telephone system. Using SMS, a short alphanumeric message (160 alphanumeric characters) can be sent to a mobile phone to be displayed there, much like in an alphanumeric pager system. The message is buffered by the GSM network until the phone becomes active.

The packet-based service of GPRS is publicized to cost users less than circuit-switched services since communication channels are being used on a shared-use, as-packets-are-needed basis rather than dedicated only to one user at a time. It is also intended to make applications available to mobile users because the faster data rate means that middleware currently needed to adapt applications to the slower speed of wireless systems will no longer be needed. As GPRS becomes widely available, mobile users of a virtual private network ("VPN") will be able to access the private network continuously rather than through a dial-up connection. GPRS is also intended to complement "Bluetooth", a standard for replacing wired connections between devices with wireless radio connections. In addition to the Internet Protocol ("IP"), GPRS supports X.25 protocol. GPRS is also believed to be an evolutionary step toward Enhanced Data GSM Environment ("EDGE") and Universal Mobile Telephone Service ("UMTS").

Universal Mobile Telecommunications Service ("UMTS") is intended to be a 3G, broadband, packet-based transmission of text, digitized voice, video, and multimedia at data rates up to 2 Mbps. UMTS is also intended to offer a consistent set of services to mobile computer and phone users no matter where they are located in the world. This service is based upon the GSM communication standard, and is endorsed by major standards bodies and manufacturers, and is the planned standard for mobile users around the world by 2002. Once UMTS is fully implemented, computer and phone users can be constantly attached to the Internet as they travel.

Enhanced digital GSM enterprise ("EDGE")service is a faster version of the Global System for Mobile (GSM) wireless service, designed to deliver data at rates up to 384 Kbps and enable the delivery of multimedia and other broadband applications to mobile phone and computer users. The EDGE standard is built on the existing GSM standard, using the same time-division multiple access (TDMA) frame structure and existing cell arrangements. EDGE is expected to be commercially

available in 2001. It is regarded as an evolutionary standard on the way to Universal Mobile Telecommunications Service (UMTS).

Wireless Application Protocol ("WAP") is a specification for a set of communication protocols to standardize the way that wireless devices such as cellular telephones and radio transceivers, can be used for Internet access, including e-mail, the World Wide Web, newsgroups, and Internet Relay Chat ("IRC"). While Internet access has been possible prior to WAP, different manufactures have used "format-specific" technologies. WAP enables devices and service systems to intercooperate.

In most recent times, much effort has been expended to merge the fields of wireless communications and the Internet in order to bridge the gap of cords, wires, and cables that had before separated the "information superhighway" from reaching people on wireless devices. Such technology merger has developed for example within the home and office network setting itself, where wireless infrared and radio frequency communications systems have been developed for interfacing equipment within a "wireless" office or home. Another substantial effort has also been underway to communicate and share information with more remote wireless devices, such as cell phones and personal digital assistants ("PDA's").

PDA's are typically small, mobile devices that may be "hand-held" and usually contain limited processors and display screens for managing, storing, and displaying telephone books, calendars, calculator, and the like. Recently available PDA's have been made "wireless enabled", either by having wireless modems embedded within the PDA itself, or by coupling to wireless modem "plug-ins" such as a cell phone. Wireless enabled PDA's are also generally "Internet enabled" with limited "browser" capability allowing the PDA to communicate with server devices over the Internet. Examples of commercially available wireless "enabled" PDA's include the Palm VII (from Palm, Inc.), and the iPAQ™ (from Compaq, Inc.). These PDA's include a Windows CE™ operating system that provides the limited browser capability and screen display for content. These phones have processing capabilities from about 33 MHz to about 220 MHz and varied screen display capabilities, such as for example 320 x 240 pixel screen displays.

Similarly, cellular phones themselves have also been recently rendered "Internet enabled", also with limited browser capability and screens to display content. Examples of "Internet enabled" cellular phones include, for example: Sanyo SCP-4000™, Motorola *i1000plus*™, among a wide range of others; this wide field represents hundreds of different processing and display capabilities.

In either the case of the PDA or the cellular phone that is "Internet-enabled", compatibility with the Internet protocols of communication must be achieved. In general, wireless communications take

place over a wireless applications protocol ("WAP"), whereas communications over the Internet proceed according to one of several different protocols, the most common being Transmission Control Protocol/Internet Protocol ("TCP/IP"). Therefore, a WAP Gateway, as shown in Figure 1E, forms a bridge between the world of the Internet (or any other IP packet network) and the wireless phone/data network, which are fundamentally different in their underlying technologies. The gateway, in essence, does the interpretation between these two distinct entities, allowing the consumer to use their cell phone or hand held computing device (e.g. PDA) to access the Internet wirelessly.

However, streaming media that is formatted for transmission to higher power computing devices such as desk-top computers having significant display capabilities is not generally compatible for receipt and viewing on these devices that have severely limited processing and display functionality. Particular "format-specific" compression schemes have been developed for use specifically with only these devices, and only specific media content may be transmitted to these devices in those formats.

There is still a need for a streaming media communications system that is adapted to transmit a wide variety of streaming media signals in appropriate formats to be played by wireless devices such as cellular phones and PDA's having unique constraints, such as, for example, limited and variable processing, memory, and display capabilities.

SUMMARY OF THE INVENTION

The present invention addresses and overcomes the various limitations, inefficiencies, resource limitations, and incompatibilities of prior known methods for streaming media communication, and is provided in various beneficial modes, aspects, embodiments, and variations as follows.

The present invention according to one embodiment is a streaming media communications system that uses a computer implemented intelligence system, such as artificial intelligence, in a network system, such as a neural network, to communicate a streaming media signal between a transmission device and at least one destination device.

The present invention according to another embodiment is a system for communicating a streaming media signal between a transmission device and a plurality of destination devices each having different media signal processing capabilities.

The present invention according to another embodiment is a streaming media communications system that is adapted to communicate a streaming media signal from a single transmission device and

at least one destination device over a plurality of different transmission channels, each having a different transmission capability or constraints with respect to communicating the streaming media signal.

The invention according to another embodiment is a neural network incorporating an artificial intelligence implementation that is adapted to be trained in an adaptive learning process with respect to the ability of a streaming media compression system's ability to compress a streaming media signal at a source into a compressed representation of the streaming media signal, transmit the compressed representation over a transmission channel to a destination device, and decompress the compressed representation into a decompressed representation of the streaming media signal which is adapted to be played by the destination device.

The invention according to another embodiment is a system for compressing streaming media signals according to a CODEC that is used at least in part based upon at least one parameter affecting communication of the streaming media signals. According to one aspect of this mode, the CODEC is used according to at least one of the following parameters: a previously learned behavior of the CODEC with respect to another reference signal, a previously learned behavior of the CODEC with respect to a prior attempt at compressing or decompressing the same streaming media signal, a comparison of the CODEC's operation with respect to the streaming media signal against a reference algorithm compression of the streaming media signal, a learned constraint of the transmission channel, and a learned constraint of the destination device. In one beneficial embodiment, the CODEC is used based upon more than one of these parameters, and in still a further beneficial variation is used based upon all of the parameters.

The invention according to another embodiment is a system for compressing streaming media signals using a CODEC library that is adapted to store multiple CODECS of different types and operations, and that is adapted to be searched and accessed by a network system, such as a neural network, in order to provide an appropriate CODEC from the CODEC library for use in compressing the input streaming media signal into a compressed representation for transmission to a destination device.

The invention according to another embodiment is a CODEC operating system that is adapted to interface with a CODEC Library and also with a neural network in order to use the neural network in a process, such as an artificial intelligence process, to choose an appropriate CODEC from the CODEC library and use the chosen CODEC for compressing the streaming media signal into a compressed representation of the streaming media signal for transmission to a destination device.

According to one aspect, the CODEC library is adapted to receive and store a new CODEC such that the new CODEC may be interfaced with the neural network in order to be chosen and applied to compress the streaming media signal as provided.

The invention according to another embodiment is a destination agent that is adapted to be stored by a destination device for use in decompressing a compressed representation of a streaming media signal. The destination agent is adapted to communicate with a remotely located, compressed streaming media transmission system in order receive and play streaming media signals therefrom. In a particularly beneficial aspect, the software agent is adapted to deliver information about the destination device to the compressed streaming media transmission system, and is also adapted to receive and decode certain encoded streaming media signals from the compressed streaming media transmission system.

The invention according to another embodiment is a system for communicating a streaming media signal having a destination agent that is adapted to be stored within a destination device for decompressing a compressed representation of a streaming media signal into a decompressed representation that may be played by the destination device.

According to one aspect of this embodiment, the destination agent has a diagnostic agent and also a decompression agent. The diagnostic agent is adapted to determine a value for at least one parameter of the destination device related to the capability for processing, storage, or display. The decompression agent is adapted to apply a CODEC decompressor to decompress the compressed representation of the streaming media signal into the decompressed representation using a CODEC based at least in part upon the value of the at least one parameter.

According to another aspect, the destination agent comprises a software agent. In one variation, the software agent is embedded within the destination device. In another variation, the software agent is adapted to be loaded onto the destination device at least in part by a remotely located source that is adapted to deliver the compressed representation of the streaming media signal to the destination device.

The invention according to another embodiment is a transcoder for trancoding streaming media signals between at least one initial format and at least one transcoded format. The transcoder includes a single thread for each of several streaming media signals.

The invention according to another embodiment is a video-on-demand streaming media system incorporating the embodiments shown in the Figures and otherwise described herein.

The invention according to another embodiment is a mobile telephone communications system incorporating the embodiments shown in the Figures and otherwise described herein.

The invention according to another embodiment is an interactive gaming system incorporating the embodiments shown in the Figures and otherwise described herein.

The invention according to another embodiment incorporates the various modes, embodiments, aspects, features, and variations herein disclosed above and elsewhere to static media, as well as to media that is stored locally after processing (e.g. compressing) and not transmitted.

BRIEF DESCRIPTION OF THE FIGURES

Figures 1A-B show schematic block diagrams representing two respective variations of prior media communications systems using conventional CODEC systems.

Figures 1C-D show schematic block diagrams representing two respective variations of prior media transcoder systems.

Figure 1E shows a schematic block flow diagram of the various interrelated components in a prior WAP gateway communications system.

Figures 2-3 show schematic block diagrams of the transcoder system of one embodiment of the present invention during two respective modes of use.

Figures 4A-5 show block flow diagrams in various detail, respectively, of a media communications system according one embodiment of the invention.

Figure 6 shows a schematic block flow diagram of various interrelated components of a "video-on-demand" streaming video communications system according one embodiment of the invention.

Figure 7 shows a schematic block flow diagram of various interrelated components of a wireless streaming video communications system according to one embodiment of the present invention.

Figure 8 shows a schematic block flow diagram of various interrelated components of a WAP gateway media communications system according one embodiment of the present invention.

Figure 9 shows a schematic block flow diagram of various interrelated components of a wireless communications system during backhauling according to one particular mode of use of the media communications system of an embodiment the present invention.

Figure 10 shows a schematic block flow diagram of various interrelated components of an interactive gaming communications system and set-top TV browsing of one embodiment of the present invention.

Aylo Exhibit 1018
Page 1190 of 2308

DETAILED DESCRIPTION OF THE EMBODIMENTS

The present invention as illustrated variously through the embodiments below (and by reference to the Figures) provides a media communications system that includes a compression system, a delivery system, and a decompression system, and in another aspect includes a transcoder system. In general, the combination of these individual sub-systems provides a capability to efficiently transcode media between multiple encoding formats, in addition to customize the compression, delivery, and decompression of randomly selected streaming media signals based upon a large array of system parameters as variables. These variables include for example, without limitation, parameters related to the following: the source video signal, the source transmitting device, the transmission modality, and the destination device. The compression, delivery, and decompression of a media signal is thus customized to be optimally efficient for a given, and changing, environment of use. As a result, a wide range of complex streaming media signals may be communicated with a level of efficiency and range of device compatibility that is significantly improved over other known systems.

Notwithstanding the benefits of the overall streaming media communication system herein described, each sub-system described also independently provides beneficially useful results for streaming media communication. The various subsystems themselves, and the various iterations of combinations of these sub-systems apparent to one of ordinary skill based at least in part upon this disclosure, are also contemplated within the scope of the invention. In addition, various aspects of the overall communication system, as well as of each sub-system described, are also contemplated as useful for other applications other than specifically for streaming media communication in particular. Therefore, where apparent to one of ordinary skill, such additional applications are further contemplated within the scope of the invention, despite the particularly useful modes applied to improved streaming media communication.

Transcoder

A video/audio transcoder 200 is provided according to the invention that enables one incoming video source 210 to be streamed across multiple formats 215 (for example MPEG4, Real Video™, and QuickTime™) from one device without human intervention. The transcoder 200 according to the present embodiment provides substantially greater functionality at a fraction of the price of other

commercially available transcoder systems. Moreover, because the system works "on-the-fly," pre-compressing of the video source 210 is significantly diminished.

More specifically, the transcoder 200 system and method according to the invention is adapted to transcode digitized media originating from any compressed or uncompressed format to be reproduced into any other compressed format -- on demand, real-time. The system 200 and method also enables efficient, simultaneous processing of multiple streams 215 of differing data from a multiplicity of different compressed or uncompressed formats into a multiplicity of different compressed formats.

The transcoder 200 of the present embodiment is herein described in an overall system by way of illustration by reference to Figure 3. As shown, a first player initially makes a connection to a server 300 that houses the transcoder 200. The player format (e.g., Microsoft Media), connection speed (e.g., 32 Kbps) and protocol (HTTP) are identified. The server 300 pulls the live or pre-encoded video into a "live buffer" or "cache" 310 and encodes it as digitized but nearly uncompressed data (e.g., AVI or MPEG2). The server 300 then loads an appropriate CODEC thread (e.g. Microsoft Media™) at the connection speed (e.g. 32 Kbps). Next, the server 300 loads a HTTP/MS player thread that serves the first client Then, a second stream is requested by a client using M/S Player at 100Kbps with MMS. The server loads the appropriate MS CODEC thread at the appropriate 100 Kbps rate. Then, the server 300 loads an MMS/MS player thread to serve the second client. Then, a third stream is requested by a client using Real Player at 40 Kbps with RTSP. The server 300 loads the appropriate Real CODEC thread at the appropriate 40 Kbps rate. Then, the server 300 loads an RTSP/Real player thread to serve the third client. Again, this illustration is exemplary, and other specific CODECS may be suitable substitutes, as well as other bit-rates, etc.

In order to provide still a further understanding of the present transcoder embodiment, Figure 3 shows the transcoder 200 by way of further example as applied to serve multiple different video streams to different clients.

In brief, the present transcoder 200 shown and described uses "thread" communications instead of "IPC" or "Inter Processor Communications" that are used according to many conventional transcoding techniques. For the purpose of this transcoder 200 description, the term "thread" is herein intended to mean an encapsulation of the flow of control in a program. Single-threaded programs are those that only execute one path through their code "at a time". Multithreaded programs may have several threads running through different code paths "simultaneously". In a typical process in which multiple threads exist, zero or more threads may actually be running at any one time. This depends on

the number of CPUs the computer on which the process is running, and also on how the threads system is implemented. While a machine or system with a number of n CPUs may be adapted to run no more than n threads in parallel, the threading operation according to the present transcoder invention may give the appearance of running many more than *n* "simultaneously" by sharing the CPUs among threads.

The transcoder 200 provides abstract APIs, and therefore the CODEC is accessed without the (much larger) native encoder overhead. Buffering 310 is created as a function of client pull for different video streams. Moreover, the transcoder 200 of the invention utilizes a network architecture – a single thread for each different connection, combining clients into same thread if they are at they are within the buffered segment of the same content. The transcoder's 200 use of threads in the manner herein shown and described is considered highly beneficial because a context switch between two threads in a single process is believed to be considerably cheaper (processing/memory/IO) than using a context switch between two processes. In addition, the fact that all data except for stack and registers are shared between threads makes them a natural vehicle for implementing tasks that can be broken down into subtasks that can be run cooperatively.

While various specific architectures may be built around the transcoder 200 embodiments just described in order to achieve particularly desired results on a case-by-case basis. However, for the purpose of further illustration, the following is an example of a more detailed system using the transcoder 200 described. The transcoder 200 is provided adapted to support a large number of simultaneous customer streams, each with differing formats. In particular, such system may support more than 5000 simultaneous streams, and in some circumstances more than 7000 simultaneous customer streams, each with differing video formats. Still further, the transcoder 200 may be implemented to convert any of a wide number of video sources to a format uniquely appropriate or required for many different individual clients each having differing needs. In one particular example, a transcoder 200 as herein described may be implemented to support such high demand simultaneously on any of the following formats: MPEG 1; MPEG 2; MPEG 4; Motion JPEG; AVI; H.261; H.263; H.263 +; RealVideo™; G-8; QuickTime™; Shockwave Flash™; Indeo Cinepak™; ASF.

It is further contemplated that the transcoder 200 may be adapted in an overall communication system to be compliant with all existing and soon anticipated fixed and mobile terminals and devices. Moreover, the transcoder 200 may be implemented to adapt output stream format variables to dynamically accommodate the channel and platform conditions of each client. Still further, the system incorporating the transcoder is adapted to support load balancing servers and routers for multi-

transcoder installations. Accordingly, it is believed that the transcoder 200 of the present invention delivers significantly greater functionality for significantly lower cost than other prior transcoding techniques and systems.

As described above, various different system architectures may incorporate the transcoder 200 of the invention without departing from the scope of the invention. However, more details of a particular architecture that is believed to suitably provide the beneficial level of support just described includes the following aspects: (i) dual P3-933 processor; (ii) any variant of Unix OS; (iii) 512 MB RAM; Redundant Firewire or Gigabit Ethernet; Redundant Power Supplies. Such system may be provided in a rack mounted configuration, or otherwise to suit a particular need.

The following aspects of the transcoder 200 of the invention should be contemplated as broadly beneficial, both independently and in various combinations as is apparent to one of ordinary skill based at least in part from this disclosure.

A system and method is provided for utilizing asynchronous software thread communication in both user and kernel space to perform efficient transcoding on multiprocessor and/or distributed computing platforms (such as clustering). It has been observed that this method is more efficient than utilizing traditional IPC methods to implement the transcoder. A shared library of CODEC algorithms is created and used to access the various CODEC algorithms, thereby incurring a lower processing overhead as well as lower memory utilization than that required by the traditional combined encoder functionality such as that used in the majority of commercial encoders. Of particular benefit, common threads may be used for multiple connections, and in fact even a single thread may be used for every individual connection using the present transcoder.

A system and method is also provided for combining multiple clients to be served by the same thread (for efficiency) whenever the same content is demanded and dynamic buffers (caches) can accommodate all of the data points demanded.

Media Compression and Delivery System

A data compression and delivery system 400 and method is also provided according to the invention for real time dynamic data signal processing for optimal reproduction of approximations of original media data over a given set of constraints. This system 400 and method is illustrated schematically by way of block flow diagrams in Figures 4A and 5. Further description of the various

beneficial features and operation of this system is provided as follows by way of exemplary embodiments generally incorporating by reference the description provided by these Figures 4A-5.

Figure 4 A ia a block diagram illustration of one embodiment of the data compression and delivery system 400 of the present invention. As shown in Figure 4A, the data compression and delivery system 400 comprises media module 405, dynamic player module 407, image processor 410, baseline snapshot module 415, classifier 417, quality of standard (QoS) module 420, network layer input module 425 and network output layer module 430. The system 400 further comprises a neural network processing module 440, timer 435, CODEC library module 445, dynamic client request module 450, ICMP module 455, device and network parameters measurement module 460 and delivery and transmit module 465.

In one embodiment, the system 400, resident at a server node(s), processes incoming uncompressed or previously compressed data. The system 400 employs neural networks 440 with artificial intelligence to monitor the incoming data to determine a plurality of key characteristics of each data segment. The system 400 correlates the incoming data characteristics with libraries 445 of pre-developed self-referencing experientially learned rules of the patterns in a scene in a sequence of frames in the input signal (e.g., a video signal) and with externally imposed constraints to optimally choose a preferred commercially available compression/decompression algorithm (e.g. CODEC) for each segment of the data. The system 400 then sets up an extensive array of usage controls, parameters and variables to optimize the chosen algorithm. Choice of algorithm and set up of parameters and variables will dynamically vary with each segment of incoming data depending upon the characteristics of the data as well as the evolving optimization process itself. The set of possible algorithms is numerous, limited only by availability and other commercial considerations. Each segment of data is encoded and compressed in the above manner and then served to a communications channel.

The compression system 400 just described is particularly useful as a streaming media compression engine, which, based upon information from the available CODEC's and the streaming media delivery system, performs frame-by-frame analysis of the incoming video using another artificially intelligent neural network 440. The system 400 then chooses the most appropriate compression format and configures the compression parameters for optimal video compression based on the best quality as measured by , in one embodiment, a selection of a peak signal to noise ratio from the underlying system environment. The result is the "optimal" video and audio service for the device and conditions present.

A more specific account of the artificial intelligence/neural network 440 aspect of this system as applied to streaming media signals is provided as follows. Initially, a library of separate and distinct CODECs are added to the system as a searchable CODEC library 445. Additional libraries of relevant reference information are also provided, including: a Network Transport Standards (NTS) library 443; and a Quality-of-Service (QoS) library 447. Then, a video (media source) is introduced either in a digitized or non-digitized format (AD conversion is used) via image processor 410. Image processor 410 then decompresses the source (if required) and employs various standard image processing algorithms used for "cleaning-up" the source image(s). The resultant source media is then passed to the baseline snapshot 415 repository where it will be used as a "perfect gold standard" for later comparison. Simultaneously, this resultant source media is also fed to the classifier 417.

The classifier 417 analyzes the source media for temporal, spatial and logical features for the purpose of creating source media sub-segments which exhibit similar combinations of temporal, spatial and logic features. "Similar" is defined to mean a contiguous sub-segment of the source media that contains common temporal, spatial and logic features that would lend themselves to a particular encoding/compression algorithm (as found in the CODEC library 445). This source media sub-segment (or, in one embodiment, a group of contiguous video and audio frames) is referred to as a "scene".

The neural network process 440 then operates upon this scene by employing CODECs from the CODEC library 445 to compress the scene. The internal configuration of each CODEC are manipulated/changed in accordance with inputs obtained from the NTS library 443, QoS library 447, Timer Process 435, Network Input Layer 425, ICMP agent 455 and the Device and Network Parameter measurement agent 460. The compressed scene is then decompressed and a comparison is made against the Baseline Snapshot 415 using a quality measurement made by the quality standard process 420. In one embodiment of the present invention, the Quality Standard Process 420 employs a peak signal noise ration (PSNR) algorithm in order to perform the comparison of the decompressed scene against the baseline snapshot of the source media. The comparison process is repeated with various CODECs from the CODEC library 445 until the Neural Network Process 440 is satisfied with the quality of the resultant compressed scene, within the constraints of the inputs received from the NTS library 443, QoS library 447, Timer process 435, Network Input Layer 425, ICMP Agent 455 and the Device and Network Parameter Measurement Agent 460. Finally, the resultant compressed scene is sent to the Network Layer Output 430 which transports the compressed scene to the Client using an appropriate Network Transport protocol and QoS algorithm.

The above process is repeated until the entire source media has been transmitted to the Client or until the process is aborted due to various possible conditions which may include: a Client request to abort, network transport failure, Client hardware failure, etc.

The NTS library 443 is a repository of network transport services that are selected and used by the Network layer output 430 to transport compressed source media to the Client and by the Network Layer Input 425 to receive information from the Client. The selection is based upon qualitative and quantitative inputs received from the Network Layer Input 425, ICMP agent 445 and the Device and Network Parameter Measurement agent 460.

The QoS library 447 is a repository of quality of service algorithms that are selected and used by the network layer output 430 to transport compressed source media to the Client. The selection is based upon qualitative and quantitative inputs received from the Network Layer Input 425, ICMP agent 455 and the Device and Network Parameter Measurement agent 460.

The ICMP agent 455 generates inputs to the neural network process 440 that dynamically provides it with the quantitative and qualitative characteristics of the transport in use between the processor and the client. In one embodiment of the present invention, the ICMP protocol is used for this purpose.

The Device and Network Parameters Measurement agent 460 generates inputs to the neural network process 440 that dynamically provides it with the qualitative and quantitative characteristics of the client's environment. In one embodiment of the present invention, these client environment characteristics include central processing unit (CPU) capacity, network interface characteristics, storage capacity and media rendering devices capabilities.

Still referring to Figure 4A, the Network Layer Input 425 provides inbound (originating from the client) network transport services. The Network Layer Output 430 provides outbound (originating from the processor) network transport services. The Timer Process 435 provides a way for the user of the invention to limit the maximum amount of time that the Neural Network Process 440 will spend in processing a given source media.

Figure 4B is a block diagram illustration of one embodiment of a CODEC selection scheme of the neural network processing module 440 of one embodiment of the present invention. The neural network processing module 440 shown in Figure 4B comprises a video frame selection module 475, CODEC parameters module 480, input layer module 485 , hidden layers 486 – 487 and output module 488. In one embodiment of the present invention, a CODEC representative signal suitable to be used as

a reference baseline signal for incoming signals to the neural network processing module 440 is generated by the neural network processing module 440. In one embodiment, the classifier 417 determines which scenes in segments of an incoming video signal represents the best scene in light of the available parameters of the underlying CODEC. A list of standards are used by the neural network processing module 440 to determine which scene in the signal represents the best scene. In one embodiment, the Neural Network Process 440 samples a number of pixels in a particular frame of video to determine changes in the number of pixels in that particular frame vis-à-vis the pre-determined parameters of the video signal. In another embodiment, significant motion changes in a particular scene in the video signal may be used as the baseline reference scene ("best scene") for subsequent incoming video.

In one embodiment of the present invention, the neural network processing module 440 takes a segment of video from the classifier 417 as an input and subsequently takes a sample of this input to derive enough information that characterizes the video signal. For example, in the scheme illustrated in Figure 4B, the Neural Network Process 440 takes a window snap-shot (e.g., a 176 X 144 pixel window) to examine. It is advantageous for the Neural Network Process 440 to look at the center of the sample window to generate enough information about the video signal. In one embodiment of the present invention, the Neural Network Process 440 uses a minimum of 8 frames to generate the requisite information about the video signal. Information from the sample window is presented with the particular CODEC parameters from parameter module 480 to the input layer 485.

The input layer 485 is coupled to a plurality of hidden layers 486 – 487 via a plurality of neurons with each connection forming either a strong or weak synoptic link from one neuron to the other. In one embodiment, each CODEC supported by the neural network processing module 440 is provided with its own neural network to process the CODEC specific parameters that come with the particular CODEC. The Neural Network Process 440 generates the "best" video signal through a round-robin like process referred to as a "bake-off" from the plurality of CODECs processed during a video sampling capture period. In processing the best video representation from incoming signals, each of the corresponding neural networks for each of the CODECS generates the best representative sample from the hidden layers 486 – 487 and feed the signal to the output module 488. In one embodiment of the present invention, the output data set of the best CODEC from each class of CODECS being processed by the Neural Network Process 440 has two possibilities. The first being the Neural Network Process 440 submitting the best results for each CODEC to the output module 488 to a "bake-off" neural network of the plurality of "best" samples for each of the plurality of CODECS which in

turn generates the winning best CODEC from the plurality of best CODECS. The bake-off neural network is smaller and faster than the neural networks that handle the processing of the CODECS.

In a second processing scheme, the Neural Network Process 440 may implement a genetic algorithm processing of the best CODECS generated by the plurality of CODECS. The genetic algorithm follows the same statistical selection approach of a marble game. Thus, instead of feeding the winning output CODEC from the various neural networks into a "bake-off" neural network, a genetic algorithm processing may be applied to feed the output module 488 from the various neural networks into a bucket and picking the best CODEC representation from a collection of scenes at the end of the source media, for example, a movie, etc. In one embodiment of the present invention, the Neural Network Process 440 uses a combination of forward and backward propagating algorithm to process the CODECS.

Referring back to Figure 4A, for the purpose of providing a further understanding of this artificial intelligence process, the following example of one particular application is provided. It is to be appreciated that the features and operation of the system provided by this exemplary application are to be considered as broadly descriptive of the neural network 440 aspect for data compression and delivery according to the invention. Other applications may be made and fall within the scope of the invention.

A video content provider installs the system of the present invention on its server. Sample videos are introduced to the system in order to perform an initial AI process as described above. A complex matrix of CODEC characterizations, e.g. for each bit rate, pattern of video, etc., is created to be drawn from later. Next, a client end-user connects to the content provider system in order to view a video M. The communication system of the invention residing on the server delivers a software agent to the client's device, thus enabling the client to connect to the communication system in order to deliver device-specific information and receive the appropriate compressed signal with decompression CODEC for playing. Next, the AI system begins loading the video M as a streaming signal into a buffer for the purpose of choosing the appropriate CODEC for each frame and compressing each frame appropriately for transmission. The time period of the buffer depends upon multiple variables, principally the processing power of the system, and may be generally for example approximately 15 seconds for systems having appropriate capability for pre-recorded but uncompressed video media. Within the buffer, each frame is compared against each CODEC according to the "types" of sequences pre-tested in matrix as depicted in the diagram.

Next, the system 400 looks at end-user parameters, e.g. screen resolution, memory available, via information received from the software agent in the client's device. The most appropriate CODEC is then chosen and configured/tuned for optimal performance by setting certain variables within the CODEC to fixed quantities (e.g. based on comparing source video vs. patterns in past, transmission channel capabilities or constraints, and destination device capabilities or constraints). The process just described is generally done frame-by-frame by the classifier 417, but the CODECS are compared for temporal compression efficiency such that the process for each frame contemplates other leading and lagging frames. Once the appropriate CODEC is chosen and tuned for each frame (or block of frames where appropriately determined automatically by the system), the delivery system reports to the client agent and delivers the tuned CODEC ahead of the corresponding frame(s) to be decompressed and played.

It is to be appreciated that the neural network 440 of this system 400 continuously learns and remembers the performance and operation of the CODECS within the CODEC library 445, and continuously uses its learning to improve the compression efficiency of the input media signal. The process of running a signal frame through the library, modifying CODEC operating parameters, comparing compression performance by the compare logic 525 (Figure 5) against reference standard compression, and running the loop again with further modifications, is an iterative 550 (Figure 5) one that generally continues to improve compression efficiency. In fact, compression with one or more CODECS in the library 445 may reach improved levels better than the reference compression algorithm(s).

Nevertheless, when time constraints 435 (Figure 4A) are present (such as in real-time push or pull demand for the streaming media content), this process must eventually be stopped at some point so that a particular frame or series of frames being processed may be compressed 575 and delivered 580 to the destination without unacceptable delay by timer 435. Then, the next frame or series may be operated upon by the neural network 440 within the CODEC operating system. These endpoints may be defined by reaching a predetermined desired result, such as for example but without limitation: (i) reaching a predetermined percentage (%) compression efficiency, such as for example as compared to the reference standard; or (ii) reaching a predetermined or imposed time limit set on the process, such as for example according to a time related to the buffer time (e.g. 15 seconds); or (iii) the earlier occurrence of either (i) or (ii). In any event, though an endpoint is reached for choosing the appropriate CODEC and performing the compression 575 and delivery 580 operations, this does not mark an endpoint for the neural network 440 training which continues. The information that is gathered through

each loop in the process is stored 550. When subsequent similar frames or system constraint parameters in an incoming frame are encountered 545 in the future, the stored information is remembered and retrieved by the neural network 440 for improving compression 575 and delivery 580 efficiency.

While many different communication protocols are contemplated, one particular embodiment which is believed to be beneficial uses a "full duplex network stack" protocol, which allows for bi-directional communication between the server and the client device. Again, while other protocols may be appropriate for a particular application, the full duplex system is preferred.

The system 400 just described addresses the difficulties encountered with previously known CODEC systems by utilizing the streaming media delivery architecture to overcome latency issues and the embedded neural network 440 to overcome speed concerns. The system 400 is then able to reconfigure the algorithms used for compression in the neural network 440, the goal being to achieve optimum results every time over any network configuration.

A wide variety of CODECS may be used within the CODEC library 445 according to the overall compression systems and methods just described, though beneficial use of any particular CODEC according to the invention contemplates such CODEC taken either alone or in combination with other CODECS. For example, an appropriate CODEC library 445 may include one or more of the following types of CODECS: (i) block CODECS (e.g. MPEG versions, such as Microsoft Media™ or QuickTime™); (ii) fractal CODECS; and (iii) wavelet CODECS (e.g. Real™). According to another aspect, an appropriate CODEC library 445 may include one or more of the following types of CODECS: (i) motion predictive CODECS; and (ii) still CODECS. Still further, the CODEC library 445 may contain one or more of the following: (i) lossy CODECS; and (ii) lossless CODECS.

In one embodiment of the present invention, all of these different types of CODECS may be represented by the CODEC library 445 according to the invention; and, more than one particular CODEC of a given type may be included in the library. Or, various combinations of these various types may be provided in order to achieve the desired ability to optimize compression of a streaming media communication over a wide range of real-time variables in the signal itself, transmission channel constraints, or destination device constraints. Still further, an additional highly beneficial aspect of the invention allows for new CODECS to be loaded into the library 445 and immediately available for use in the neural network 440 compression/delivery system 400. Nevertheless, one particular example of a CODEC library 445 which is believed to be beneficial for use in optimally communicating a wide range of anticipated streaming media signals, and of particular benefit for image signals, includes the

following specific CODECS: MPEG versions 1, 2, and 4 (e.g. Microsoft Media™ and QuickTime™); DUCK TruMotion™; ON2; Real Media™; MJPEG: H.261; H.263; H.263+; GIF; JPEG; JPEG2000; BMP; WBMP; DIVX.

The following are further examples of various aspects of the compression system and method just described that should be considered as broadly beneficial, both independently and in various combinations as is apparent to one of ordinary skill based at least in part on this disclosure. Further examples of such broad aspects are elsewhere provided in the "Summary of the Invention" as well as in the appended claims.

Use of neural networks 440 with artificial intelligence to achieve the various CODEC operations described is broadly and uniquely beneficial. In particular, a system and method is provided for pre-processing 410 of source data determined by application of learned responses to the signal quality, data content and format of the data. A system and method is provided for processing each unit (e.g. frame or block of frames) of source data by selection and application of a suitable CODEC (from a set of all available CODECS in the CODEC library 445) dependent upon observed characteristics of the source data and application of past-learned responses to compressing similar data. A system and method is provided for processing each unit of source data by setting a multiplicity of compression characteristics within a chosen compression algorithm to optimize capture and preservation of the original data integrity. Still further, each or all of the aforementioned signal processing steps is applied to each unique, sequential unit of signal data, e.g., signal clip, video frame, or individual packet as appropriate.

It is further contemplated that a CODEC management system 400 according to the invention provides a system and method for image processing that is adapted to normalize original source data/images as well as to resize and resample original data to fit the specification of the neural network processing module 440. An ability to serve any transmission or recording channel with a single system and with any source data stream is also provided. Moreover, the various systems and methods herein described, individually and beneficially in combination, are provided with compatibility to any connection or connectionless protocol, including but not limited to TCP, UDP, WTP/WDP, HTTP, etc.

The invention as herein shown and described also allows for highly beneficial applications for accelerating the learning rate of neural networks 440 while minimizing the data storage requirements to implement said networks. Different classes of data streams each have unique characteristics that require substantially greater processing by neural networks 440. For example, video data streams differ by prevalence and degree of motion, color contrast, and pattern and visibility of details. Greater

processing requires longer times to reach optimal functionality. Greater processing also requires more predictive library storage, often growing to unlimitedly large sizes. For real-time neural network processing, processing time and storage can be minimized to greatly increase functionality by providing pre-developed predictive libraries characteristic of the class of data stream.

Accordingly, the following are examples of aspects of the pre-trained neural network 440 aspects of the invention that should be appreciated as broadly beneficial, both independently and in combination (including in combination with other embodiments elsewhere herein shown and described). A system and method is provided that creates and uses artificial intelligence in a neural network 440 and pre-trains that intelligent network for use in solving a problem, which problem may be for example but not necessarily limited to streaming media compression according to a particular beneficial aspect of the invention. A system and method is also provided for subdividing the universe of problems to be solved into useful classes that may be processed according to a learned history by the intelligent network.

An intelligent streaming media delivery system and method is also provided according to the invention that manages content transmission based on end-user capabilities and transmission channel constraints, such as for example, but without limitation, available transmission speeds or bandwidth, and Internet congestion. The data compression and delivery system 400 utilizes a computer implemented intelligence process, such as an artificial intelligence process based on a neural network to analyze aspects of the connection (including without limitation differing bit rates, latencies, transmission characteristics and device limitations) to make modifications in the compression methodology and to manage Quality of Service ("QoS") 420 issues. Compressed, digital, restorable and/or decompressible data streams may be therefore delivered to a multiplicity of different local and/or remote devices via a multiplicity of transmission mediums characterized by differing capabilities. In addition, a decompression system is provided for reproducing the decompressed data at the terminal device.

In one beneficial embodiment, a terminal device establishes a link with the system resident on a server node(s). Except for software normally required to establish communications, the terminal device might not initially have resident software embedded therein associated with the present system. Upon linking the terminal device to the server node, the system transmits a software agent to the terminal device that cooperates with other software modules on the server-side that together form the overall delivery system. The software agent informs the system of the terminal device configuration and processing capacities for decompressing and displaying the data. The software agent also reports

certain relevant information to the system of the characteristics of the communication channel between the terminal and the server. Such information includes, without limitation: latency, bandwidth, and signal path integrity. Based upon terminal device configuration and real time updates of channel characteristics and capabilities, the system actively manages transmission of the compressed data stream by varying parameters such as buffer length, transmitted bit rate, and error correction. The system also feeds operating conditions to the compression system to dynamically alter encoding and compression settings to optimize delivery of the data. The delivery software agent resident on the terminal device decompresses the data stream that is composed of segment-by-segment variations in compression/decompression algorithm and settings thereof. Dependent upon the terminal device configuration, and especially for very thin clients, instructions may be refreshed on a segment-by-segment basis for each decompression algorithm and encoding setting combination. Instructions for decompressing may also be kept resident if appropriate to the terminal device.

The software agent described for transmission to and operation by the destination device is therefore also considered a highly beneficial aspect of the compression/delivery systems and methods described. By delivering the software agent to the device from the source, a wide range of existing destination devices may be used for communication according to methods that may include variable uses of one or more algorithms or other operations at the transmission source. In other words, the destination devices may not be required to be "format-specific" players as is required by much of the conventional streaming and static media communication systems. Also, by providing the destination agent with a diagnostic capability, diagnostic information may be gathered at the destination device and transmitted back to the source in a format that is compliant for use by the source in its neural network process for achieving the proper CODEC operation for a given set of circumstances.

The use of a client-side agent to supply quality of service information including client-side device data and communication channel status in real time is therefore also believed to be broadly beneficial beyond the specific applications and combinations of other aspects of the invention also herein provided. In addition, the processing of each unit of compressed, transmission-ready data to accommodate client-side device and real-time communication channel conditions is also broadly contemplated as having broad-reaching benefits. Still further, a system and method is described that provides instructions to a client-side agent to enable decompression of each sequential, uniquely compressed unit of data. Therefore, another broad benefit of the invention provides a destination device (such as from the transmission source as herein described for the particular embodiments) with a CODEC that is adapted to decompress a compressed representation of an original media signal into a

decompressed representation based upon variable parameters related to at least one of the following: aspects of the original media signal, transmission channel constraints, and destination device constraints. In another broad aspect, the destination device is adapted to use a CODEC that is chosen from a library of CODECS based upon a parameter related to an aspect of the original media signal.

The systems and methods herein described are also considered applicable to the signal processing of each unique, sequential unit of signal data, e.g., signal clip, video frame, or individual packet as appropriate. In addition, the system and its various sub-systems may also be purely software that must be loaded into each appropriate device, or it may be embedded in a host hardware component or chip, e.g. on the server side, or in certain circumstances, on the client side (e.g. various aspects of the destination agent), or for example may be stored such as in flash memory.

The various aspects of the media compression system and method just described are considered beneficial for use according to a wide range of known and soon anticipated media communication needs, including for example according to the various communications devices, communication/transmission channel formats and standards, and media types and formats elsewhere herein described (e.g. in the "Background" section above).

However, for the purpose of further understanding, Figure 6 shows a schematic view of the overall streaming media communications system 600 as specifically applied to "video-on-demand" aspects according to one embodiment of the present invention, wherein many different end users 610 – 620 at many different locations may request and receive, real-time (e.g. without substantial delay), pre-recorded video from a remote source. Further to the information provided in Figure 6, at least one specific implementation of the media communication system 600 delivers the following types of video at the following bit-rates (denotes compressed representations of original signals that are convertible by a destination device to decompressed representations having no or insubstantial loss as observed by the eye of the typical human observer): VHS-format video as low as about 250Kbs; DVD-format video at about 400 Kbps; and HDTV-format video at about 900Kbps. According to these speeds, it is believed that video-on-demand may be provided by telephone carriers over resident transmission line channels, such for example over existing DSL lines 630 - 640.

However, as available bandwidth and mass communication continue to present issues, it is believed that even greater efficiencies may be achieved resulting in delivery of compressed representations of these types of video signals at even lower bit rates. Again, as elsewhere herein described, the compression efficiencies of the invention are closely related to and improve as a function of the processing power made available to the neural network 440, and the neural network's 440

Aylo Exhibit 1018
Page 1205 of 2308

continued learning and training with respect to varied types of media. These resources may even make more remarkable compression efficiencies achievable without modification to the fundamental features of the present invention.

Therefore, the following are further examples of transmission rates for certain compressed video signals that are believed to be desirable and achievable according to one embodiment of the invention: VHS-format video as low as about 200Kbps, more preferably as low as about 150Kbps, and still more preferably as low as about 100Kbps; DVD-format video as low as about 350Kbps, more preferably as low as about 300Kbps, and still more preferably as low as about 250Kbps; and HDTV-format video as low as about 800Kbps, and still more preferably as low as about 700 Kbps.

Moreover, at least one implementation of the media communications system 400 of one embodiment of the invention delivers 20-24 frames/sec color video at a transmission rate of 7 Kbps. This is believed to enable substantial advances in communication of streaming media signals via to wireless destination devices via the WAP Gateway, as is further developed elsewhere hereunder.

It is also to be appreciated that, while video communication has been emphasized in this disclosure, other types of streaming or static media are also contemplated. For example, at least one implementation of the compression and delivery embodiments has been observed to provide substantially CD-quality sound (e.g. via compressed representations of original signals that are convertible by a destination device to decompressed representations having no or insubstantial loss as observed by the ear of the typical human observer) at a bit-rate of about 24 Kbps. At these rates, audiophile-quality sound may be delivered for playing over dial-up modems. However, with further regard to available resource commitment and extent of neural network training, it is further contemplated that the invention is adapted to deliver CD-quality sound at speeds as low as about 20Kbps, and even as low as about 15 Kbps or even 10Kbps.

Wireless Audio Communications System

It is further contemplated that the streaming media communication system of the invention has particularly useful applications within wireless audio communications networks, and in particular cellular telephony networks. Therefore, Figures 7 and 8 schematically show, with respectively increasing amounts of detail, streaming media communications systems 700 and 800 respectively specifically applied to wireless audio communications systems according to certain specific, respective embodiments of the present invention. While particular devices, system parameters, or arrangements

of communicating devices shown are believed to be beneficial in the overall application of the invention, they are not to be considered limiting and may be suitably replaced with other substitutes according to one of ordinary skill based upon this disclosure. The various wireless communications systems 700 and 800, standards, and protocols referenced elsewhere in this disclosure are thus incorporated into this section for the purpose of integration with the various aspects of compression, delivery, decompression, and transcoding according to one embodiment of the invention.

Combination of the communications system 400 of one embodiment of the present invention with the other components of a cellular communications network allows for the enhanced compression, delivery, and decompression according to the invention to manifest in an increased quality of service for wireless audio communications. Improvements in cellular communications according to the invention include, without limitation, the following examples: increasing available bandwidth, extending range of reception, and providing graceful degradation while maintaining connectivity during periods of low signal quality or reception levels.

More specifically, cellular telephony signals are characterized by relatively high degrees of variability, due for example to the roaming positions of clients, and limited cell ranges, atmospheric conditions, and significantly limited and changing available bandwidths over daily use cycles. Therefore, a self-optimizing CODEC management system according to the present invention is particularly well suited to adjust the appropriate communications and compression modalities to the changing environment. At the very least, the increase in compression efficiency and resulting decrease in bandwidth used for given signals is a valuable achievement as wireless channel traffic continues to congest.

In one particular regard, the increased compression efficiency according to the present invention is well applied to improving bandwidth issues during "soft hand-offs" between cells, as illustrated in Figure 9. During cellular phone communications, whenever a transmitter or receiver migrates between cell coverage zones, communications bandwidth requirements and resultant costs are increased by systemic requirement to "pass off" active communications between cells. The act of passing off the communication results in a "backhaul" channel from the previously active cellular transmitter to a central office for forwarding to a newly active cellular transmitter. The backhaul channel represents a significant use of bandwidth. Savings will result from increased compression. As Figure shows, such "backhauling" may include a doubling (media sent back from first cell being left and resent to second cell for transmission) or even a quadrupling (overlapping communication from both first and second cells) in the bandwidth used for communicating a particular signal.

The media communications system 400 of the present invention may recognize when backhaul is occurring, such as according to the transmission channel diagnostics provided in the software agent(s), and may respond by adjusting the degree of compression to compensate.

WAP Video Gateway

With a particular view of the rapid growth observed and predicted in the wireless or mobile Internet, embodiments of the present invention contemplate application of the intelligent compression/delivery/ decompression embodiments in combination with WAP Gateway functionality.

A system and method is therefore also provided according to the invention for encoding, compressing and transmitting complex digital media (e.g., video pictures) via bandwidth-constrained wireless communications systems utilizing Wireless Applications Protocol (WAP). In one embodiment, data is processed by the system, resident at a server node(s), employing neural networks with artificial intelligence. Sample segments of data are captured from the input stream and processed to comply with requirements unique to the class of clients. As is described in detail above, the system correlates the continuously varying digital data streams' characteristics with libraries of pre-developed experientially learned rules and with externally imposed constraints to optimally choreograph the coherence, continuity and detail of the data as ultimately received, decoded and presented at the client interface.

A gateway provided with the added functionality of the streaming media communications system herein described is shown schematically in Figure 8. According to the WAP gateway system 830 , a client agent is provided that is adapted to run on a variety of platforms, and requires no specialized hardware to decode the video streams. According to use of the streaming media delivery system of the invention elsewhere herein described, the viewer of the WAP device maintains constant communication with the system server upstream, such that the user-side client 825 may provide the encoding platform with relevant information for streaming media communication, including without limitation: available screen size, processing power, client operating system and browser version, connection speed and latency, thereby allowing the streaming media delivery system to tailor the stream to each individual client it "talks" to. Accordingly, an AI driven server 830 incorporating the AI compression as herein described may be combined with a WAP Gateway 830, combining the necessary WAP to TCP/IP protocol (or other protocol, e.g. dual server stack) translation with a Video and Audio Server 835 employing compression, delivery, and decompression systems and methods

herein described. The WAP Gateway 830 may further include a video transcoder, such as for example incorporating the transcoder systems and methods herein described. An appropriate host architecture according to this system (not shown) generally includes a rack mount system running Linux OS with a modified WAP Gateway 830 or as a software plug-in to existing servers.

This WAP gateway system 830 may be further provided in a Master/Slave relationship as another beneficial aspect of the overall streaming media delivery architecture (applicable to other delivery systems other than specifically wireless). Various content distribution networks, such as available through Akamai and Inktomi, have capitalized on the concept of improving data delivery over the Internet by using "smart caching" on servers which reside on the borders of the Internet. Such a Master/Slave relationship is maintained by the present system wherein a Master Server resides at the source of the content to be delivered and Slave Servers reside on the borders. These servers communicate "intelligently" to optimize the content delivery over the Internet and reduce latency, bandwidth and storage requirements, improving the overall quality of the video/audio stream to the end-user and decreasing the cost of media delivery to the content provider.

The WAP gateway 830 of the present invention supports continued growth in mobile communications, as large telecommunications operators are transitioning to multi-service broadband networks, and as the number of subscribers to the mobile Internet continues to expand rapidly. In particular, mobile communications is a broad class of systems and protocols, each having its own constraints and needs for interacting devices to communicate streaming media. The Gateway 830 in a particularly beneficial aspect may support a variety of "2G" systems with upgradability for upcoming "2.5G" and "3G" network technologies (numerical progression of systems generally represents progression of Internet-enabled capabilities).

The following Table 3 provides examples of known mobile communication standards, and provides certain related information used by the AI system of the present invention for optimizing communication of streaming media amongst the field of mobile destination devices as media players:

Table 3: Existing/Soon Anticipated Mobile Communications Standards

MODE	**BAUD RATE (generally)**
GSM (2G)	9.6 Kbps
CDMA	9.6 Kbps
TDMA	14.4 Kbps
CDPD	14.4 Kbps

iMODE	128 Kbps
GPRS	144 Kbps
WCDMA or CDMA2000	144 Kbps to 2 Mbps
GSM (3G)	2 Mbps

In addition, the present invention is particularly beneficial in its ability to stream a wide variety of media signals to various different types of wireless communications devices. Examples of wireless communications devices that are appropriate for use with the streaming media communications systems and methods of the invention, and which the systems and methods support interchangeably, are provided in the following Table 4:

Table 4: Examples of Internet-enabled PDA's

DEVICE	MAKE	SPEED	MEMORY	SCREEN DEPTH	SCREEN SIZE	MODEM TYPE	CONNECT SPEED
iPAQ	Compaq	206 MHz	16-64 Mb	12 b/pixel color	320x240	External (e.g. CDPD)	9600-14.4 Kbps
PalmVII	Palm	33 MHz	4 -16Mb	4 b/pixel b/w 8 b/pixel color		Internal	14.4 Kbps
Handspring	Palm	33 MHz	4 -16Mb	4 b/pixel b/w; 8 b/pixel color		External	14.4 Kbps
Blackberry	Research-In-Motion	33 MHz	4 Mb	2 b/pixel b/w		Internal	9.6-14.4 Kbps
Jornada	HP	133 MHz	16-32 Mb	18 b/pixel	320x240	External	9.6-14.4 Kbps

Casseopeia ▪▪▪▪▪▪▪▪▪	Casio	150 Mhz	16-31 Mb	12 b/pixel	320x240	External	9.6-14.4 Kbps

Various specific examples are described later below that provide observations of actual wireless Internet applications of the invention as herein described. Such examples include use of a CODEC library according to varied parameters associated with at least the following (without limitation): destination wireless communication device; transmission channel; communications protocol; and the respective streaming media signals themselves. The various particular features of the systems and methods used according to these examples are contemplated as further defining independently beneficial aspects of the invention.

Shared Interactive Environment

A system and method is also provided according to the invention for enabling real-time remote client interaction with a high-definition, multi-dimensional, multi-participant simulated environment without the need for significant client-side processing capacity. More specifically, Figure 10 shows an overall streaming media communication system as applied to shared interactive gaming according to the invention.

This system includes: (i) a proxy server; (ii) graphics rendering capabilities; (iii) a client software agent for feedback of client inputs to the game; (iv) a client software agent for supporting the delivery system of the invention; and (v) streaming from the server to the client. It is contemplated that for multiple clients, which typically represent shared interactive gaming by design, multiple components as just described are provided to support each client.

The interactive gaming embodiments contemplate implementation of data compression and delivery embodiments with devices that are also destination devices for compressed signals from other like, remotely located device systems. This arrangement is broadly beneficial, such as for example in further interactive media implementations such as video conferencing and the like. Accordingly, each remote system is both a source and a destination device, and sends and receives agents between it and other remote systems.

Destination Device

Although the communications systems of the present invention enables communication of streaming media signals to a wide variety of destination devices, a further contemplated feature of the invention provides a remote receiver to be housed as a destination device/player by client users. This set-top player may be adapted to serve at least one, though preferably multiple ones, and perhaps all, of the following: Video on Demand (VOD); Music on Demand (MOD); Interactive Gaming on Demand (IGOD); Voice Over Internet Protocol ("VoIP"), any technology providing voice telephony services over IP connections; Television Web Access; Digital Video Recording to record, pause, and playback live television; e-mail; chat; a DVD player; and other applications apparent to one of ordinary skill. All of this may be delivered to existing televisions in the comfort of users' own homes. Moreover, clients utilizing this box, or other systems interfacing with the communications system of the invention, may receive DVD quality video and surround sound over cable and DSL connections.

EXAMPLES

For the purpose of further illustrating the highly beneficial results that may be achieved according to the invention, the following are examples of specific embodiments that have been used for different types of streaming media communication, including observed results with pertinent discussion. These examples illustrate communication the same pre-recorded video over different transmission channels and to different destination devices, wherein the pre-recorded video has the following originating properties: 720lines of resolution and 32bits of color information, an originating file size of about 1.4Gigabytes.

Example 1

An "iPAQ" model 3650 hand-held PDA (commercially available from Compaq, Inc. for approximately $500 at the time of this disclosure) was provided. The PDA was interfaced with a 14.4 Kbps (max) wireless CDPD modem ("AirCard 300" wireless external modem, commercially available from Sierra Wireless for approximately $200 at the time of this disclosure) using an extension assembly (iPAQ™ PCMCIA expansion sleeve from Compaq, Inc.) with a PCMCIA card slot that couples to the wireless modem. The iPAQ™ used is generally characterized as having the following processing parameters: 206 MHz processor; 32 Mb memory; 12 b/pixel color; 240X320 screen dimensions; PocketPC™ operating system version 3.0 from Microsoft Corp. and stereo sound. The iPAQ™ was

connected to the Internet in San Francisco, California via the interfaced CDPD modem over the AT&T cellular wireless carrier system at a connection bandwidth of about 13.3Kbit/sec. A server located in San Jose, California (approximately 50mi away) was contacted by the PDA employing the http and rtsp protocols, and the PDA was used to initiate a request for a pre-recorded video having the following originating properties: 720lines of resolution and 32bits of color information, the originating file size was 1.4Gigabytes. Within about seven seconds, a compressed approximation of the pre-recorded video was received, decompressed, and displayed by the PDA on the PDA's screen. The entire video was seen at 240x320x12bpp resolution in full motion without observable delays or defects.

Example 2

A "Jornada™" model 548 hand-held PDA (commercially available from HP, Inc. for approximately $300 at the time of this disclosure) was provided. The PDA was interfaced with a 9.6 Kbps (max) wireless CDMA phone ("Motorola i85s" wireless external digital cellular phone, commercially available from Motorola authorized vendors for approximately $200 at the time of this writing) using adaptor cables (Motorola and HP RS-232 standard interface cables from Motorola and HP.) that couple the phone and PDA together to form a wireless modem. The Jornada model PDA device used is generally characterized as having the following processing parameters: 133 MHz processor; 32 Mb memory; 12 b/pixel color; 240X320 screen dimensions; PocketPC™ operating system version 3.0 from Microsoft Corp. and stereo sound. The Jornada™ was connected to the Internet in Newark, NJ via the interfaced CDMA phone/modem over the Nextel digital cellular wireless carrier system at a connection bandwidth of 8Kbit/sec. A server -located in San Jose, California (approximately 2900mi away) was contacted by the PDA employing the http and WDP protocols, and the PDA was used to initiate a request for a pre-recorded video having the following originating properties: 720lines of resolution and 32bits of color information, the originating file size was 1.4Gigabytes. Within about seven seconds, a compressed approximation of the pre-recorded video was received, decompressed, and displayed by the PDA on the PDA's screen. The entire video was seen at 176x120x8bpp in full motion without observable delays or defects.

Example 3

A "Set-top Box" model st850 book PC (commercially available from MSI, Inc. for approximately $300 at the time of this writing) was provided. The Set-top Box was interfaced with a 10 Mbps (max) ethernet/802.11 connection using CAT5 ethernet cables (Generic) that couple the Set-top Box to a broadband connection (DS3). The Set-top Box used is generally characterized as having the following processing parameters: 400 MHz processor; 64 Mb memory; 32 b/pixel color; 720 lines of screen resolution; Windows CE operating system version 2.11 from Microsoft Corp. and AC3 digital 6 channel surround-sound. The Set-top Box was connected to the Internet in Newark, NJ via the interfaced shared DS3 connection over the Alter.Net Internet Backbone at a connection bandwidth of 376Kbit/sec. A server located in San Jose, California (approximately 2900mi away) was contacted by the Set-top Box employing the http and rtsp protocols, and the Set-top Box was used to initiate a request for a pre-recorded video having the following originating properties: 720lines of resolution and 32bits of color information, the originating file size was 1.4Gigabytes. Within About nine seconds, a compressed approximation of the pre-recorded video was received, decompressed, and displayed by the Set-top Box on a commercially available reference monitor's (Sony) screen. The entire video was seen at 720linesx32bpp in full motion without observable delays or defects.

While various particular embodiments have been herein shown and described in great detail for the purpose of describing the invention, it is to be appreciated that further modifications and improvements may be made by one of ordinary skill based upon this disclosure without departing from the intended scope of the invention. For example, various possible combinations of the various embodiments that have not been specifically described may be made and still fall within the intended scope of the invention. According to another example, obvious improvements or modifications may also be made to the various embodiments and still fall within the intended scope of this invention.

What is claimed is:

1. A system for processing a media signal, comprising:

a searchable CODEC library that is adapted to store a plurality of separate and unique CODECS and also to be accessed by and searched by at least one of an artificial intelligence (AI) system and an operating system, such that each CODEC stored in the CODEC library may be individually located and accessed for use by at least one of the AI system and operating system in order to compress the media signal for transmission from a source to a destination device.

2. The system of Claim 1, wherein the CODEC library comprises at least one CODEC that is of at least one of the following three types of CODECS: block CODECS; fractal CODECS; and wavelet CODECS.

3. The system of Claim 2, wherein the CODEC library comprises:
a first CODEC of a first type on the three types, and
a second CODEC of a second type of the three types.

4. The system of Claim 3, wherein the CODEC library comprises:
a first CODEC of a first type of the three types,
a second CODEC of a second type of the three types; and
a third CODEC of a third type of the three types.

5. The system of Claim 4, wherein the CODEC library comprises:
a first CODEC that is a motion predictive CODEC; and
a second CODEC that is a still CODEC.

6. The system of Claim 5, wherein the CODEC library comprises:
a first CODEC that is a lossy CODEC; and
a second CODEC that is a lossless CODEC.

7. The system of Claim 1, further comprising:

a registry associated with the CODEC library and that is searchable in order to search and locate a particular CODEC within the CODEC library.

8. A system for communicating a media signal from a source to a destination device over a transmission channel, comprising:

a destination agent that is adapted to be used by the destination device to communicate with the source and to receive a compressed representation of the media signal from the source and to decompress the compressed representation into a decompressed representation of the media signal such that the decompressed representation may be stored or used by the destination device.

9. The system of Claim 8, wherein the destination agent comprises:

a diagnostic agent that is adapted to provide diagnostic information for transmission to a neural network associated with the source, which diagnostic information represents an ability of the destination device to receive, store, process, or display the media signal; and

a decompression agent that is adapted to decompress the compressed representation of the media signal into the decompressed representation of the signal to be stored or played by the destination device.

10. The system of Claim 9, wherein the destination agent further comprises:

a decompression agent that is adapted to receive a compressed representation of the media signal and also to receive a CODEC decompressor associated with the compressed representation of the signal from the delivery system, and that is adapted to use the CODEC decompressor to decompress the compressed representation of the signal into a decompressed representation of the signal to be stored or played by the destination device.

11. The system of Claim 10, wherein the destination agent is adapted to be stored and operated for decompressing and playing a media signal on a plurality of different destination devices each having different capabilities for decompressing, storing, or playing the media signal.

12. The system of Claim 11, wherein the destination agent is adapted to decompress a plurality of different compressed representations of the media signal by using different CODEC decompressors that are each associated with a corresponding different CODEC compressor used to compress the signal.

13. The system of Claim 12, wherein the destination agent is adapted to decompress the compressed representation of the media using a CODEC that is provided based upon at least one of the following parameters: a computer implemented previously learned behavior of the CODEC with

respect to a reference media signal, a computer implemented previously learned behavior of the CODEC with respect to a prior attempt at compressing or decompressing the same media signal, a comparison of the CODEC's operation with respect to the media signal against a reference algorithm compression of the media signal, a learned constraint of the transmission channel, and a learned constraint of the destination device.

14. The system of Claim 13, wherein the destination agent is adapted to interchangeably use a block CODEC, a fractal CODEC, or a wavelet CODEC.

15. A system for communicating a media signal from a source to a destination device over a transmission channel, comprising:

a CODEC that is adapted to compress the media signal for optimized transmission along the communications channel and decompression by the destination device, wherein the compression is done by the CODEC based upon at least one of the following parameters: a previously learned behavior of the CODEC with respect to another reference signal, a previously learned behavior of the CODEC with respect to a prior attempt at compressing or decompressing the same media signal, a comparison of the CODEC's operation with respect to the media signal against a reference algorithm compression of the media signal, a learned constraint of the transmission channel, and a learned constraint of the destination device.

16. The system of Claim 15, wherein the CODEC is adapted to compress the media signal into n different compressed representations of the media signal that are adapted to be transmitted to the destination device at n unique bit rates, respectively.

17. The system of Claim 15, further comprising:

a searchable quality of service (QoS) library associated with the CODEC and that is adapted to be used by at least one of an artificial intelligence system or an operating system in order to determine an appropriate use of the CODEC with respect to the media signal based upon at least one QoS standard stored in the QoS standards library.

18. A system for processing a media signal, comprising:

a CODEC operating system that is adapted to interface with a CODEC library and also with an artificial intelligence (AI) system in order to use the AI system to identify an appropriate CODEC from the library and to process the media signal with the CODEC.

19. The system of Claim 18, further comprising:

at least one CODEC associated with the CODEC operating system for use in processing the media signal based at least in part upon an operation by the CODEC operating system.

20. The system of Claim 19, further comprising:

a searchable CODEC library that is associated with the CODEC operating system and that is adapted to store a plurality of separate and unique CODECS, wherein the CODEC operating system is adapted to cooperate with the searchable CODEC library and also with the AI system in order to determine an appropriate CODEC stored in the CODEC library for use in processing the media signal.

21. The system of Claim 20, further comprising:

a searchable network transport standards library associated with the CODEC operating system and that is adapted to be used by at least one of an AI system or the CODEC operating system in order to determine an appropriate use of a CODEC with respect to the media signal based upon at least one network transport standard stored in the network transport standards library.

22. The system of Claim 21, further comprising:

a searchable quality of standard (QoS) library associated with the CODEC operating system and that is adapted to be used by at least one of an AI system or the operating system in order to determine an appropriate use of a CODEC with respect to the media signal based upon at least one QoS standard stored in the QoS standards library.

23. The system of Claim 22, further comprising:

a destination agent that is associated with the CODEC operating system and is adapted to be used by a destination device to communicate with a source also associated with the CODEC operating system to receive a compressed representation of the media signal from the source and to decompress the compressed representation into a decompressed representation of the media signal such that the decompressed representation may be stored or used by the destination device.

24. A system for processing a media signal, comprising:

a destination device that is adapted to communicate with a source and to receive a compressed representation of the media signal from the source, and that includes a CODEC that is adapted to decompress the compressed representation into a decompressed representation based upon at least one of the following parameters: a previously learned behavior of the CODEC with respect to another reference signal, a previously learned behavior of the CODEC with respect to a prior attempt at compressing or decompressing the same media signal, a comparison of the CODEC's operation with respect to the media signal against a reference algorithm compression of the media signal, a learned constraint of the transmission channel, and a learned constraint of the destination device.

25. The system of claim 24, wherein the media signal is adapted to be used according to an original format according to an original set of parameters, and the destination device is adapted to process and display the decompressed representation of the media signal in another format according to another set of parameters that are different than the original set of parameters, such that the destination device is incapable of processing and displaying the media signal in the original format according to the original set of parameters.

26. The system of claim 25, wherein the CODEC is adapted to compress the media signal into n different compressed representations of the media signal that are adapted to be transmitted to the destination device at n unique bit rates, respectively.

27. The system of claim 26, wherein the n different compressed representations are adapted to be transmitted at least at n substantially different transmission rates that are distinguished based upon at least one of the following parameters: different transmission system standards; different destination device standards; and different media type standards.

28. A system for processing a media signal, comprising:

a CODEC that is adapted to perform at least one of compression or decompression on the media signal; and

an artificial intelligence (AI) system that is adapted to operate in conjunction with the CODEC in order to tune the operation of the CODEC with respect to the media signal, wherein the AI system is adapted to learn a behavior of the CODEC with respect to at least one parameter affecting the CODEC's operation, and also to modify the operation of at least one of the AI system

itself or the CODEC based upon the learned behavior in order to increase the ability to provide the CODEC in an appropriate form for optimally compressing or decompressing the media signal.

29. The system of claim 28, wherein the CODEC comprises a discrete cosine transform CODEC.

30. The system of claim 29, wherein the CODEC comprises a fractal CODEC.

31. The system of claim 30, wherein the CODEC comprises a wavelet CODEC.

32. The system of claim 31, wherein the CODEC comprises a lossy CODEC.

33. The system of claim 32, wherein the CODEC comprises a substantially lossless CODEC.

34. The system of claim 33, further comprising:

a searchable CODEC library that is adapted to store a plurality of separate and unique CODECS,

wherein the CODEC is adapted to be stored in the searchable CODEC library, and the AI system is adapted to cooperate with the CODEC library in order to locate and access the CODEC so that the AI system may perform at least one operation with respect to the CODEC.

35. A system for processing a media signal, comprising:

a CODEC library that is adapted to store a plurality of separate and unique CODECS;

an artificial intelligence (AI) system associated with the CODEC library and that is adapted to perform an operation with respect to the CODEC library in order to determine an appropriate CODEC from the plurality of CODECS for use in processing the media signal;

a CODEC operating system associated with the CODEC library and also the AI system and that is adapted to control the operation of the AI system with respect to the CODEC library;

a source system that is adapted to compress the media signal based upon a CODEC from the CODEC library and based upon an operation of the AI system;

a destination system that is adapted to communicate with the source system and to receive a compressed representation of the media signal from the source system and decompress the compressed representation into a decompressed representation for use by the destination system; and

a communications system associated with the source system and also with the destination system so that the source system is adapted to transmit the compressed representation of the media signal to the destination system, and the destination system is adapted to receive the compressed representation and also to appropriately decompress the compressed representation into the decompressed representation for storage or use by the destination system.

36. The system of claim 35, further comprising:

a searchable network transport standards library associated with the AI system and also with the CODEC operating system and that is adapted to be used by the AI system in order to determine an appropriate CODEC from the CODEC library for processing the media signal based upon at least one network transport standard stored in the network transport standards library.

37. The system of claim 36, further comprising:

a searchable QoS standards library associated with the AI system and also with the CODEC operating system that is adapted to be used by the AI system in order to determine an appropriate CODEC from the CODEC library for processing the media signal based upon at least one QoS standard stored in the QoS standards library.

38. The system of claim 37, wherein the AI system is adapted to determine an appropriate use of a CODEC from the CODEC library for processing the media signal based upon at least one of the following parameters: a previously learned behavior of the CODEC with respect to another reference signal, a previously learned behavior of the CODEC with respect to a prior attempt at compressing or decompressing the same media signal, a comparison of the CODEC's operation with respect to the media signal against a reference algorithm compression of the media signal, a learned constraint of the transmission channel, and a learned constraint of the destination device.

39. A system for communicating a media signal between a source and n unique destination devices each having a unique ability to process the media signal, wherein n is an integer that is equal to or greater than one, comprising:

a compression system associated with the source and which is adapted to compress the media signal into n uniquely compressed representations of the media signal,

wherein each of the n uniquely compressed representations is adapted to be transmitted from the source to a unique one of the n destination devices, and is adapted to be decompressed by the respective destination device into a unique one of n decompressed representations of the media signal to be played by the respective destination device.

40. The system of claim 39, wherein the compression system further comprises:

a searchable CODEC library that is associated with the compression system and that is adapted to store a plurality of separate and unique CODECS, wherein the compression system is

Aylo Exhibit 1018
Page 1221 of 2308

adapted to compress the media signal into each of the n uniquely compressed representations based at least in part upon at least one CODEC stored by the CODEC library.

41. The system of claim 40, wherein the compression system further comprises:

a searchable network transport standards library associated with the compression system and that is adapted to be used by the compression system in order to determine at least one appropriate CODEC for compressing the media signal into the n uniquely compressed representations based upon at least one network transport standard stored in the network transport standards library.

42. The system of claim 41, wherein the compression system further comprises:

a searchable QoS standards library associated with the compression system and that is adapted to be used by the compression system in order to determine at least one appropriate CODEC from the CODEC library for compressing the media signal into the n uniquely compressed representations based upon at least one QoS standard stored in the QoS standards library.

43. The system of claim 42, further comprising:

a destination agent that is associated with the compression system and is adapted to be used by each of the n unique destination devices to communicate with the source, so that each of the n destination devices may receive a unique one of the n compressed representation of the media signal from the source system and to use an appropriate CODEC at least in part based upon a CODEC used by the compression system to create the respective compressed representation so that the respective destination device may appropriately decompress the respective compressed representation into an appropriate one of the decompressed representations of the media signal that may be stored or used by the respective destination device.

44. The system of claim 43, wherein the compression system is adapted to compress the media signal into each of the n different compressed representations based at least in part upon at least one of the following parameters: a previously learned behavior of at least one CODEC with respect to another reference signal, a previously learned behavior of at least one CODEC with respect to a prior attempt at compressing or decompressing the same media signal, a comparison of at least one CODEC's operation with respect to the media signal against a reference algorithm compression of the media signal, a learned constraint of at least one transmission channel, and a learned constraint of at least one destination device.

45. A system for communicating n uniquely compressed representations of a media signal from at least one source to a common receiver, comprising:

a destination device that is adapted to communicate with the at least one source and to receive each of the n uniquely compressed representations from the at least one source, wherein the destination device is further adapted to decompress each of the n uniquely compressed representations into n decompressed representations of the media signal and to play each of the n decompressed representations.

46. A system for communicating a media signal from a source to a destination device over a transmission channel, comprising:

a compression system that is adapted to receive information about a parameter of the transmission channel and to compress the media signal into a compressed representation of the media signal by using a compression technique that is based at least in part upon the parameter information.

47. A system for communicating n unique media signals between a source and a destination, comprising:

a compression system that is adapted to receive information about a parameter of the media signal and to compress the media signal into a compressed representation of the media signal by using a compression technique that is based at least in part upon the parameter information.

48. A system for compressing a media signal from a source for transmission to a destination device, comprising:

a compression system that is adapted to compress the media signal into a compressed representation of the media signal based upon at least one of a transmission channel constraint, a destination channel constraint, or a comparison of a parameter of the media signal against a prior compression operation of the compression system.

49. A system for decompressing a compressed media signal received from a source at a destination device, wherein the destination device is adapted to receive a compressed representation of the media signal from the source and also a decompression instruction from the source such that the destination device is adapted to decompress the compressed representation into a decompressed representation of the media signal based at least in part on the instruction.

Aylo Exhibit 1018
Page 1223 of 2308

50. A method for constructing a neural network for use in compressing media signals, comprising:

providing a plurality of CODECS; and

pre-learning at least one parameter associated with each CODEC's compression performance with respect to a predetermined type of media signal in a manner.....; and

remembering each pre-learned parameter in a manner such that at least a particular one of the CODECs may be chosen in the future to compress a particular new media signal based upon the pre-learned parameter of the particular CODEC.

51. The method of claim 50, further comprising:

forming a CODEC library with the plurality of CODECS;

compressing at least one predetermined type of a test media signal with each CODEC in the CODEC library;

automatically learning at least one parameter related to each CODEC's performance in compressing the at least one predetermined type of test media signal;

automatically remembering each such learned parameter in a manner such that a particular media signal to be compressed.

52. A system for communicating a media signal between a source and a destination at least in part over a wireless transmission channel, wherein a compressed representation is

a compression system that is adapted to compress the media signal into a compressed representation of the media signal that is further adapted to be transmitted to a wireless destination device over a wireless carrier; and

a neural network that uses artificial intelligence to optimize the compression of the media signal by the compression system by learning at least one of the following parameters: a constraint of the wireless destination device, a constraint of the wireless transmission carrier, and a comparison of a parameter of the media signal against a prior compression operation of the compression system.

53. A system for compressing a media signal for transmission from a wireless source to a destination device.

54. A system for compressing a media signal into a compressed representation of the media signal for transmission to a wireless destination device.

55. A system for decompressing a compressed representation of a media signal into a decompressed representation of the media signal at a destination device, wherein the compressed representation is received from a source over a wireless carrier at the destination device.

56. A system for communicating a media signal over an IP network between a source and a destination device, wherein a compressed representation of the media signal is transmitted from the source to the destination device over the IP network.

57. A system for compressing a media signal into a compressed representation of the media signal, such that the compressed representation of the media signal may be transmitted from a source to a destination device over an IP network.

58. A system for decompressing a compressed representation of a media signal into a decompressed representation of the media signal, wherein the compressed representation is received at a destination device from a source over an IP network.

59. A method for communicating a media signal over an IP network, comprising:

using artificial intelligence to determine the most appropriate CODEC from a plurality of available CODECS for compressing and decompressing the media signal, wherein the appropriate CODEC is determined based upon at least one of the following parameters: a previously learned behavior of the CODEC with respect to another reference signal, a previously learned behavior of the CODEC with respect to a prior attempt at compressing or decompressing the same media signal, a comparison of the CODEC's operation with respect to the media signal against a reference algorithm compression of the media signal, a learned constraint of the transmission channel, and a learned constraint of the destination device.

60. The method of Claim 59, wherein the system is adapted to communicate a streaming media signal over the IP network.

61. A system for communicating an audio signal between a source and a destination over a transmission channel, wherein the audio signal is compressed to form a compressed representation of the audio signal that may be transmitted over the transmission channel to the destination device

so that the destination device may decompress the compressed representation into a decompressed representation for storage or use by the destination device.

62. The system of Claim 62, wherein the system is adapted to communicate a streaming audio signal.

63. A system for compressing an audio signal into a compressed representation of the audio signal for transmission from a source to a destination device over a transmission channel.

64. A system for decompressing a compressed representation of an audio signal at a destination device for storage or use by the destination device, wherein the compressed representation is received by the destination device from a source over a transmission channel.

65. A method for communicating a streaming audio signal from a source to a destination device over a transmission channel, comprising:

delivering CD-quality music from the source to the destination device over the transmission channel at a transmission rate that is as low as about 25 Kbps.

66. The method of claim 65, wherein the CD-quality music is delivered by compressing a source audio signal into a compressed representation and delivering the compressed representation over the transmission channel to the destination device at the transmission rate, wherein the compressed representation may be decompressed by the destination device into a decompressed representation that is substantially CD-quality representation of the original audio signal that may be stored or played by the destination device.

67. A system for communicating a video signal between a source and a destination over a transmission channel, wherein the streaming video signal is compressed into a compressed representation that may be transmitted from the source to the destination device over the transmission channel and decompressed into a decompressed representation by the destination device for storage or playing by the destination device.

68. A system for communicating a streaming video signal between a source and a destination, comprising:

a media processing module;

a dynamic player module;

an image processor;

a baseline snapshot module;

a signal classifier;

a plurality of CODECS dependent neural networks processors;

a CODECS library;

a network layer input module;

a network layer output module; and

a timer.

69. The system of Claim 68, further comprising a dynamic client request processing module.

70. The system of Claim 69, further comprising a device and network parameters measurement module.

Aylo Exhibit 1018
Page 1227 of 2308

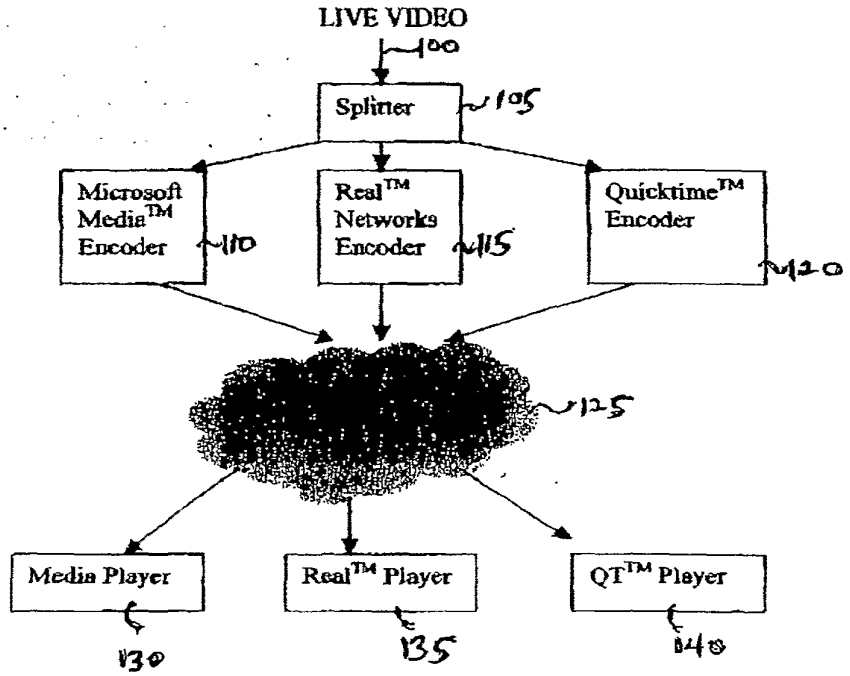


FIGURE 1A
(PRIOR ART)

Aylo Exhibit 1018
Page 1228 of 2308

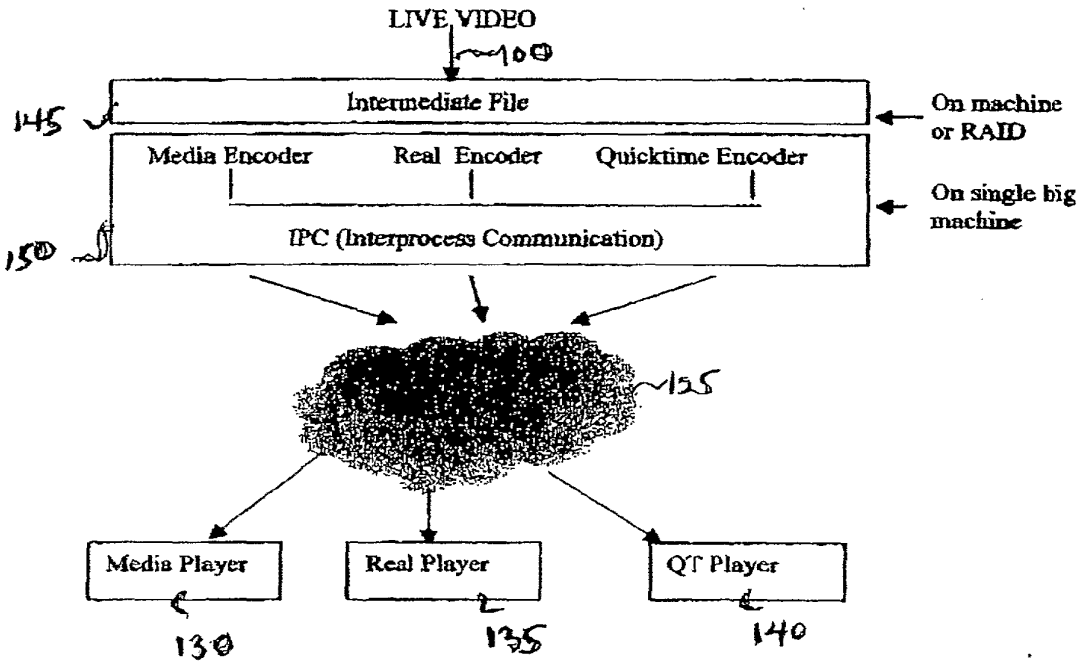


FIGURE 1B
(PRIOR ART)

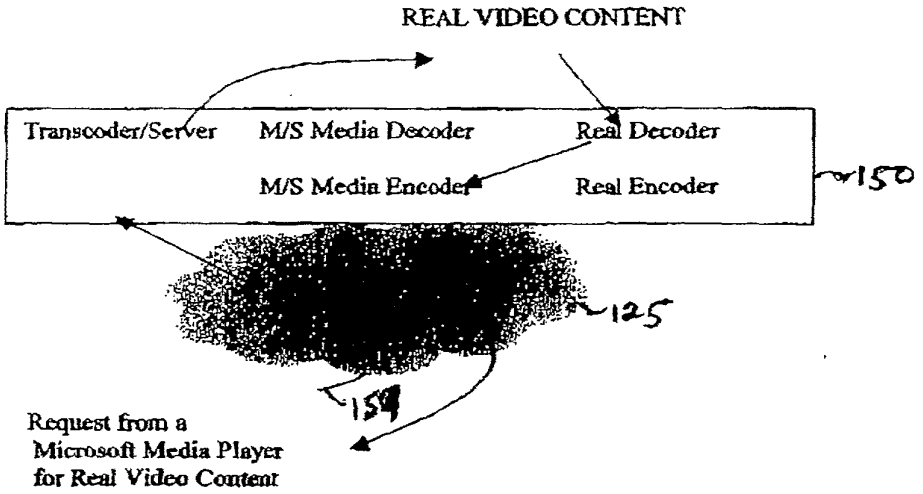


FIG. 1C
(PRIOR ART)

Aylo Exhibit 1018
Page 1230 of 2308

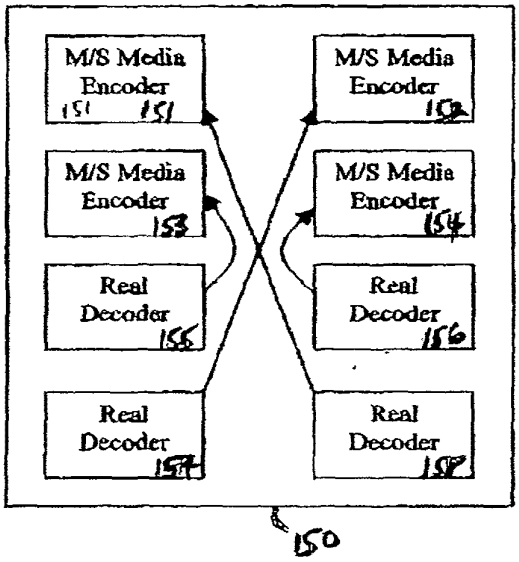


FIGURE 1D
(PRIOR ART)

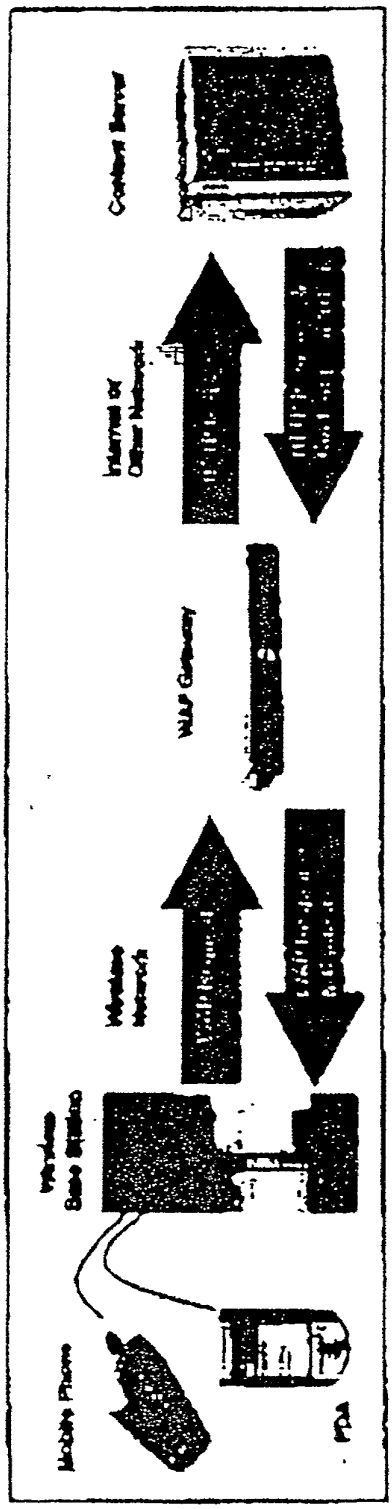


FIGURE 1E
(PRIOR ART)

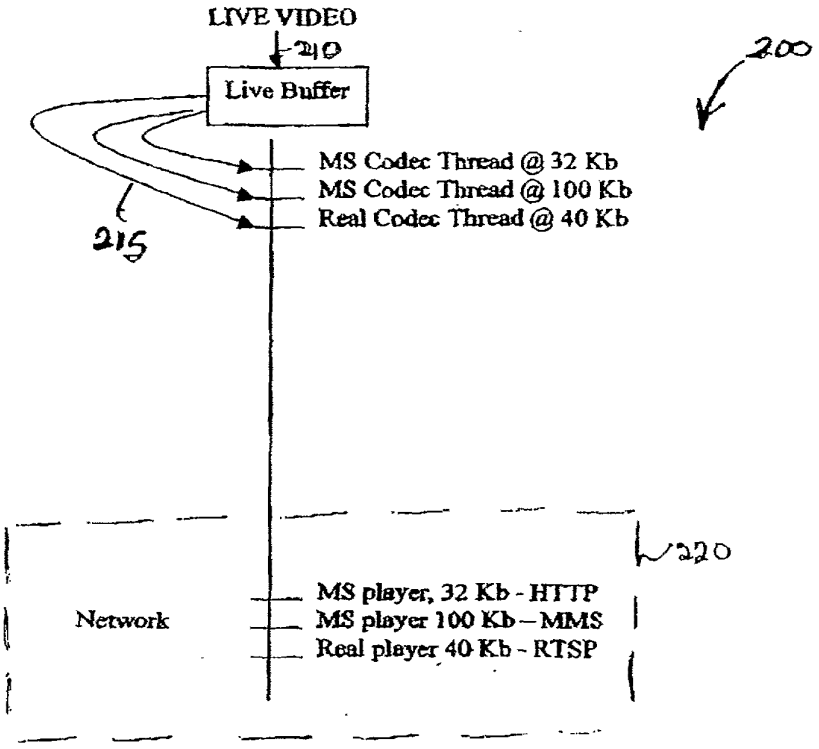

LIVE VIDEO
210
Live Buffer
200
MS Codec Thread @ 32 Kb
MS Codec Thread @ 100 Kb
Real Codec Thread @ 40 Kb
215
220
Network
MS player, 32 Kb - HTTP
MS player 100 Kb – MMS
Real player 40 Kb - RTSP

FIGURE 2

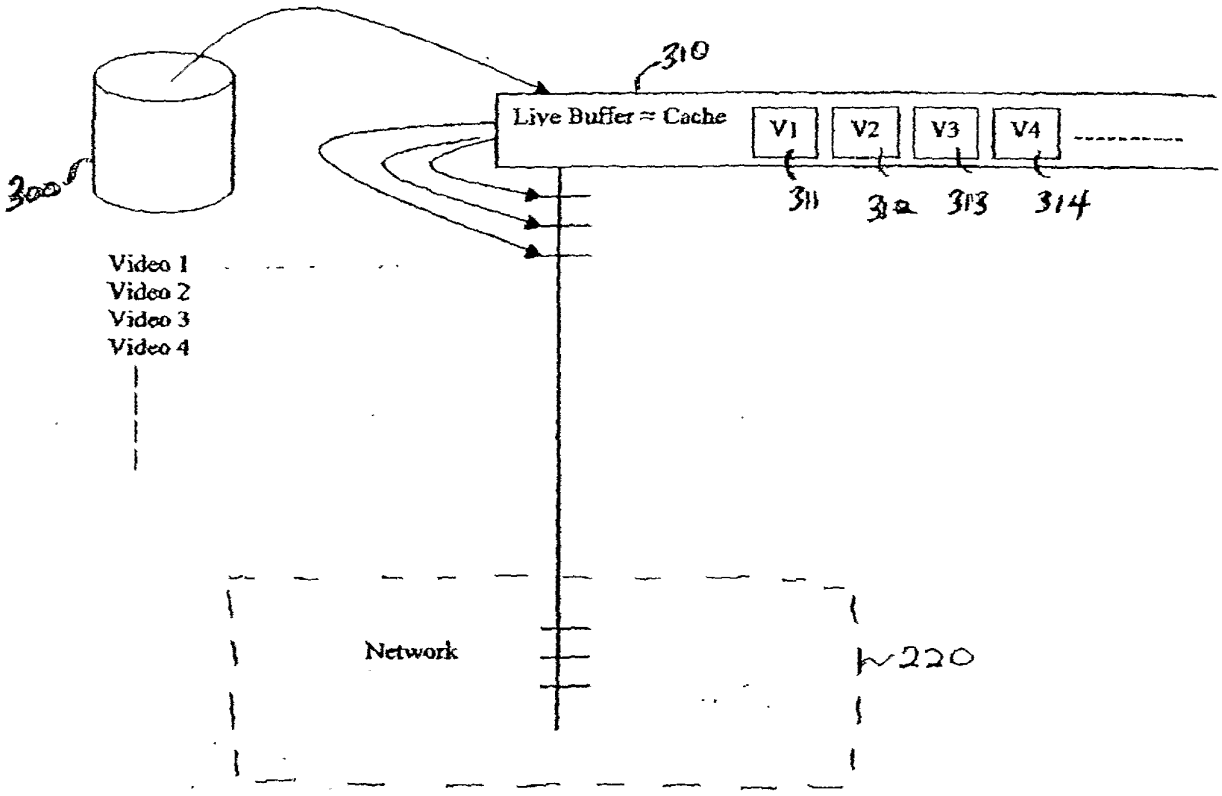
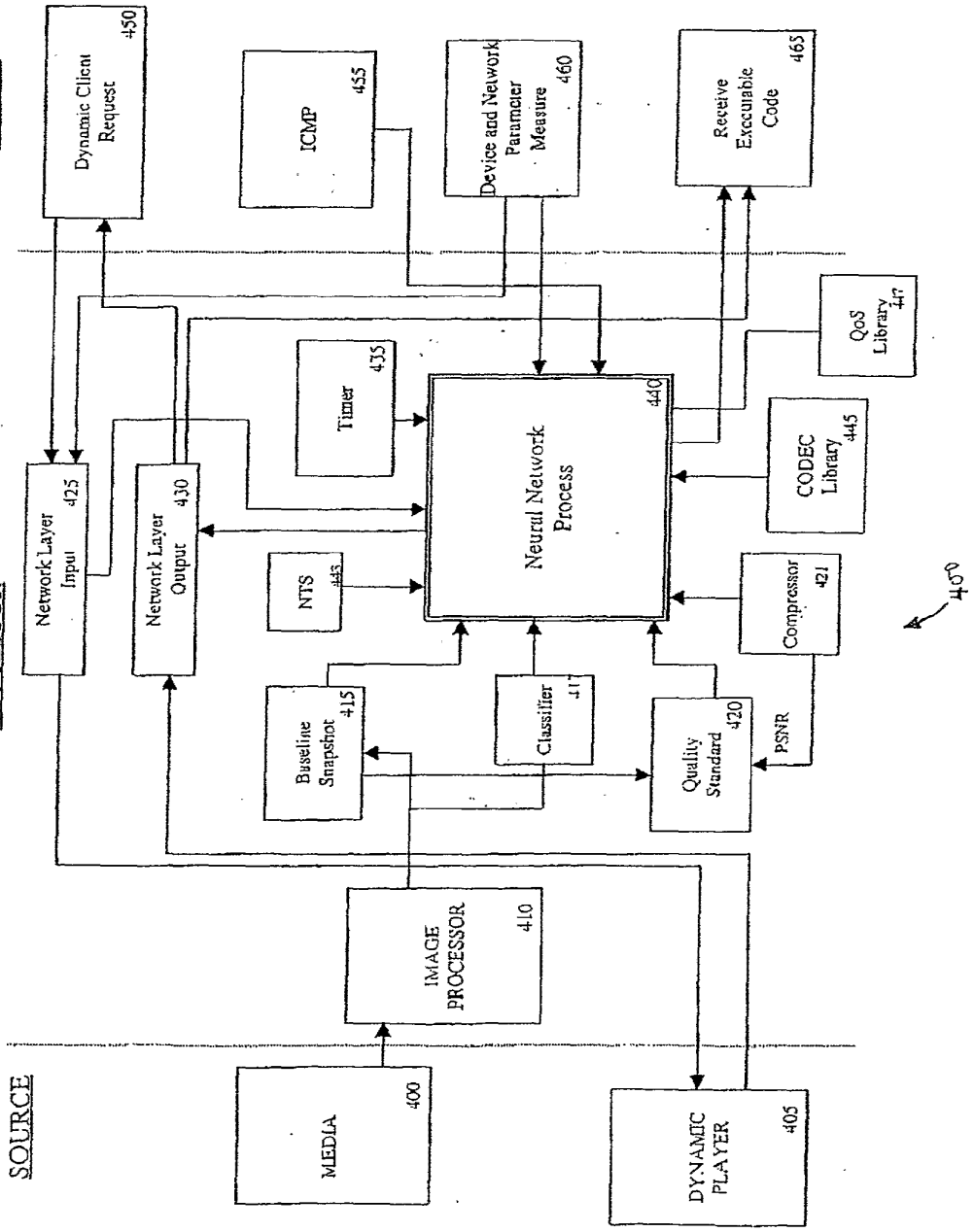

310
Live Buffer ≈ Cache
V1
V2
V3
V4
311
312
313
314
300
Video 1
Video 2
Video 3
Video 4
Network
220

FIGURE 3

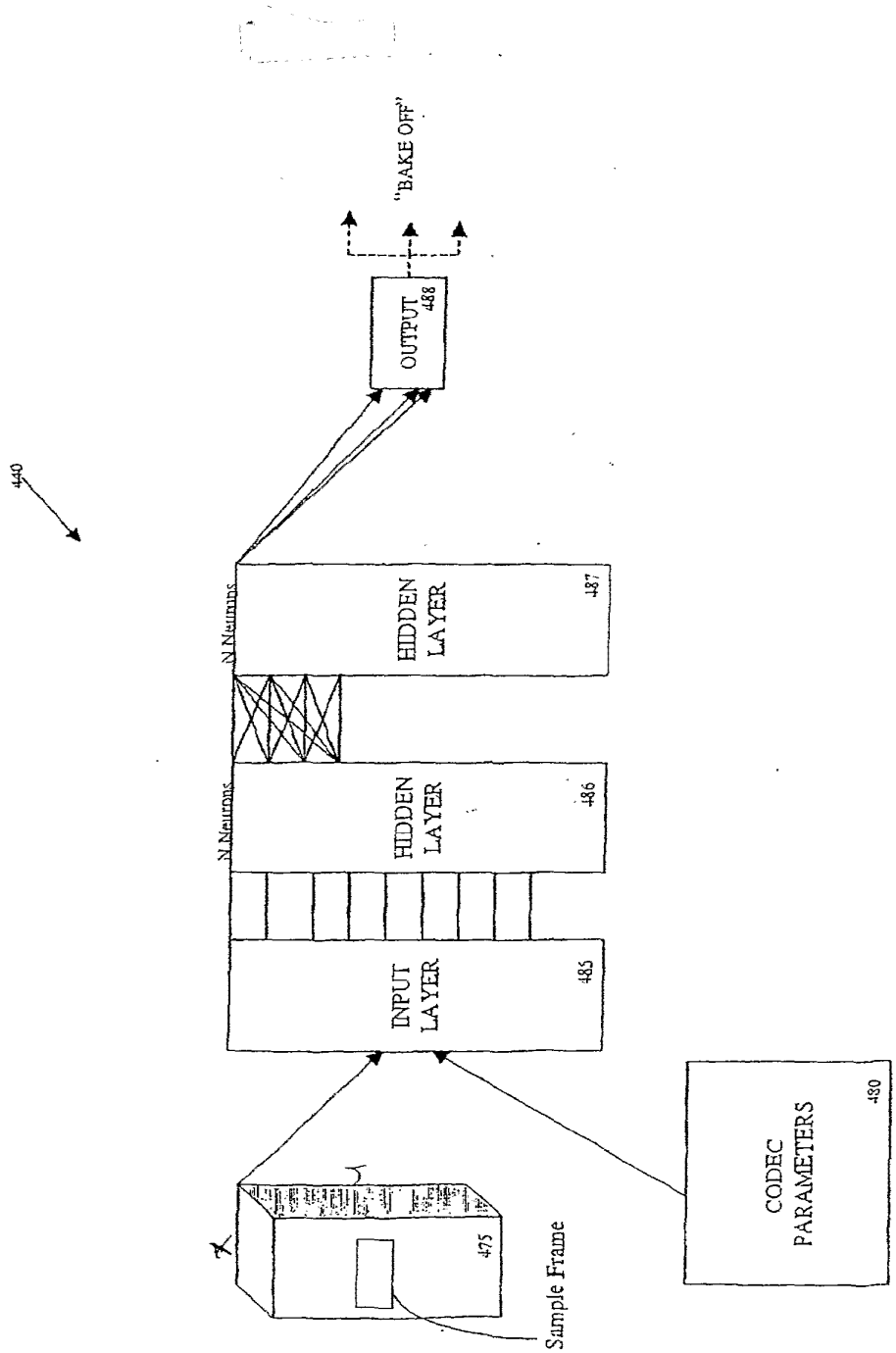
Aylo Exhibit 1018
Page 1235 of 2308

FIGURE 4A



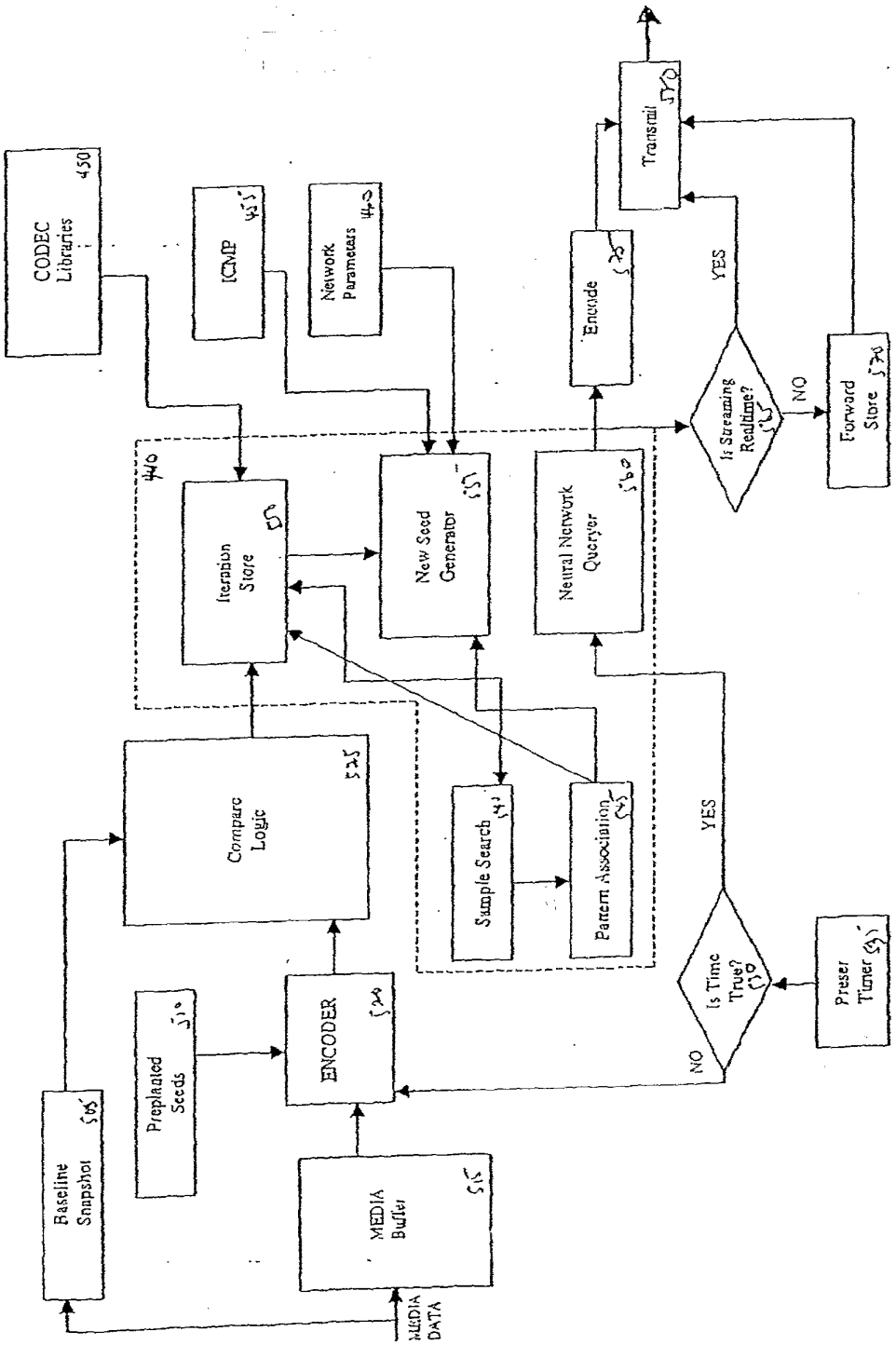
Aylo Exhibit 1018
Page 1236 of 2308

FIGURE 4B



Aylo Exhibit 1018
Page 1237 of 2308

FIGURE 5



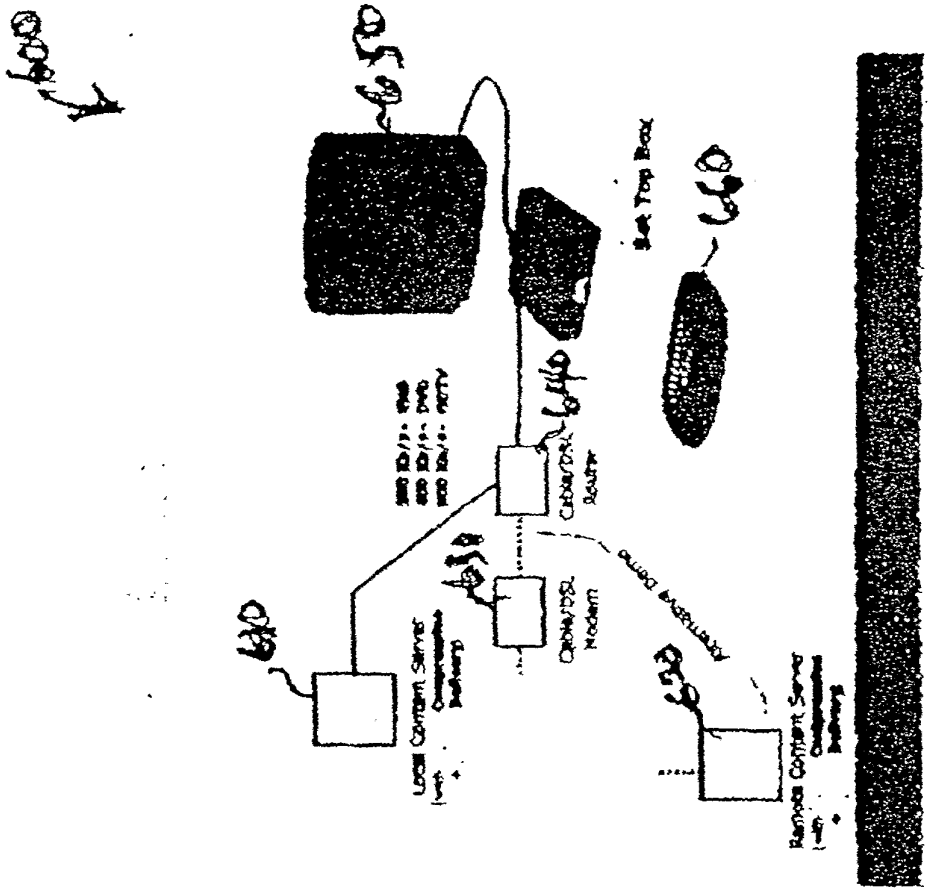


FIGURE 6

Aylo Exhibit 1018
Page 1238 of 2308

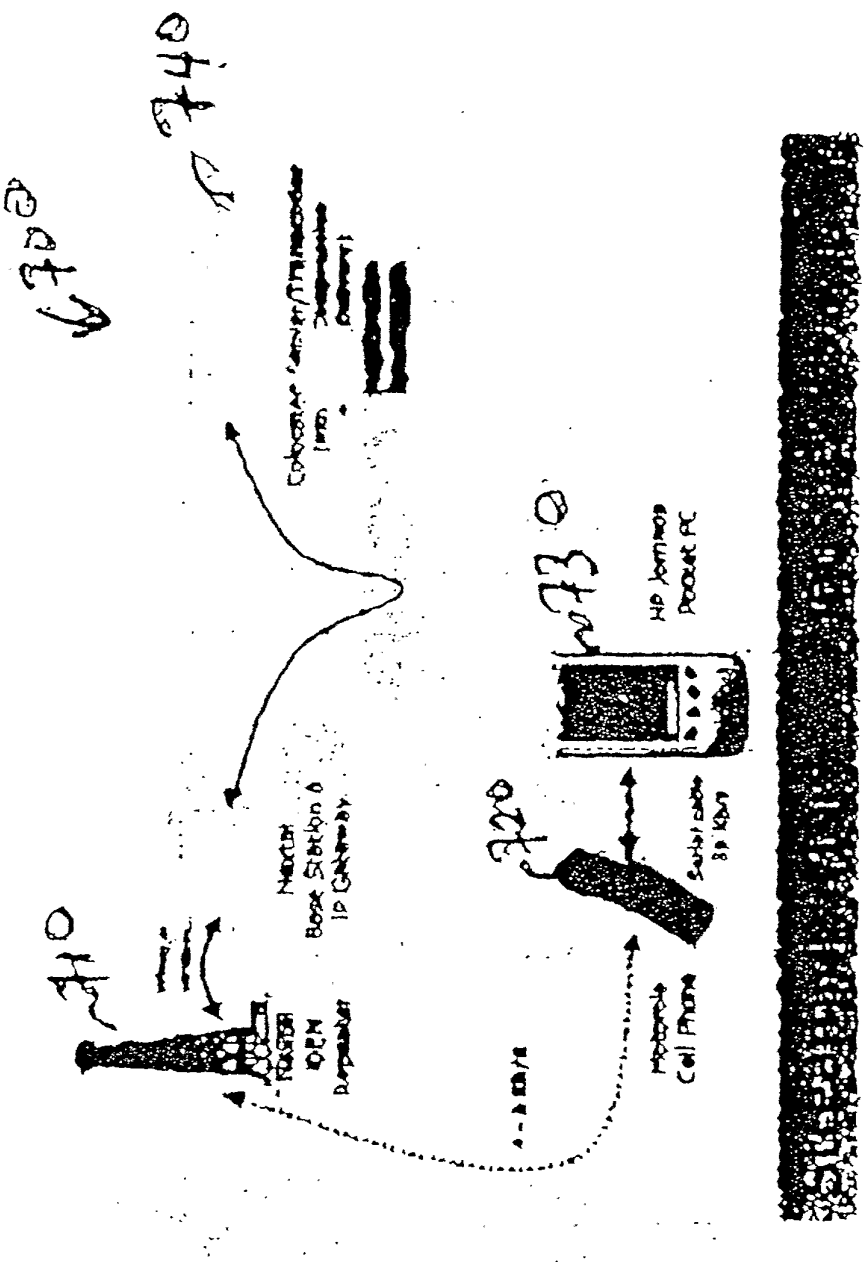


FIGURE 7

Aylo Exhibit 1018
Page 1239 of 2308

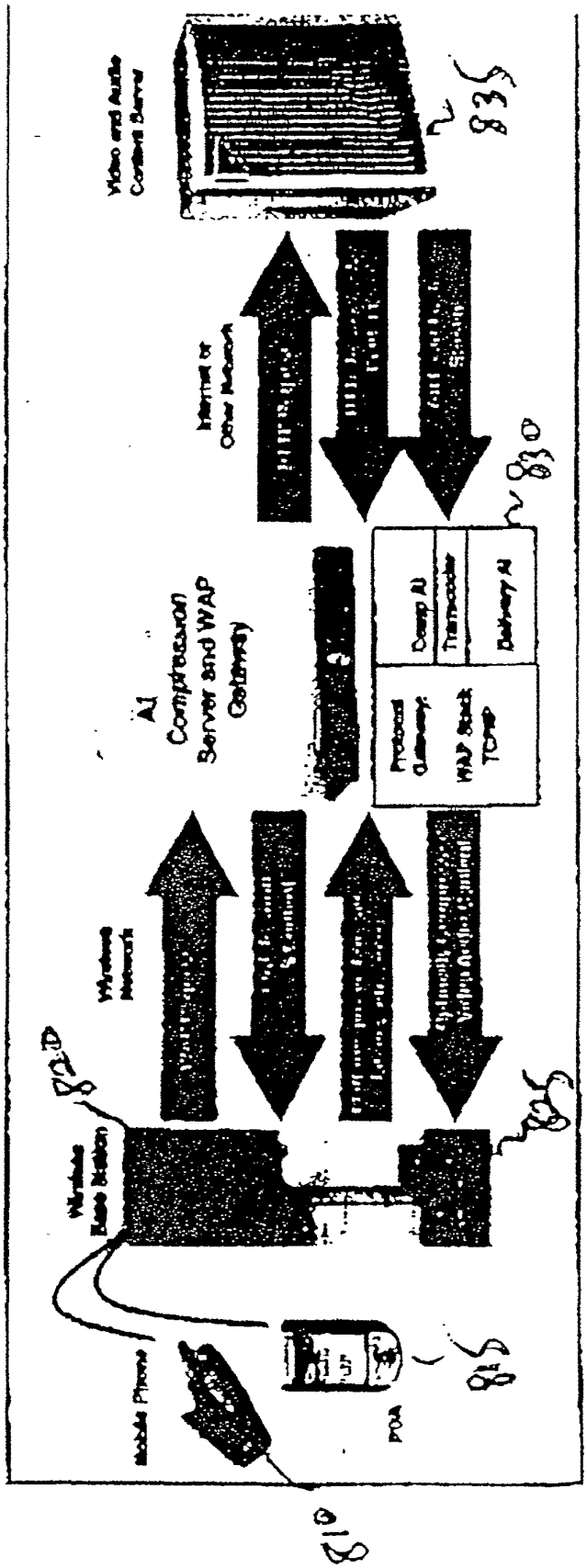
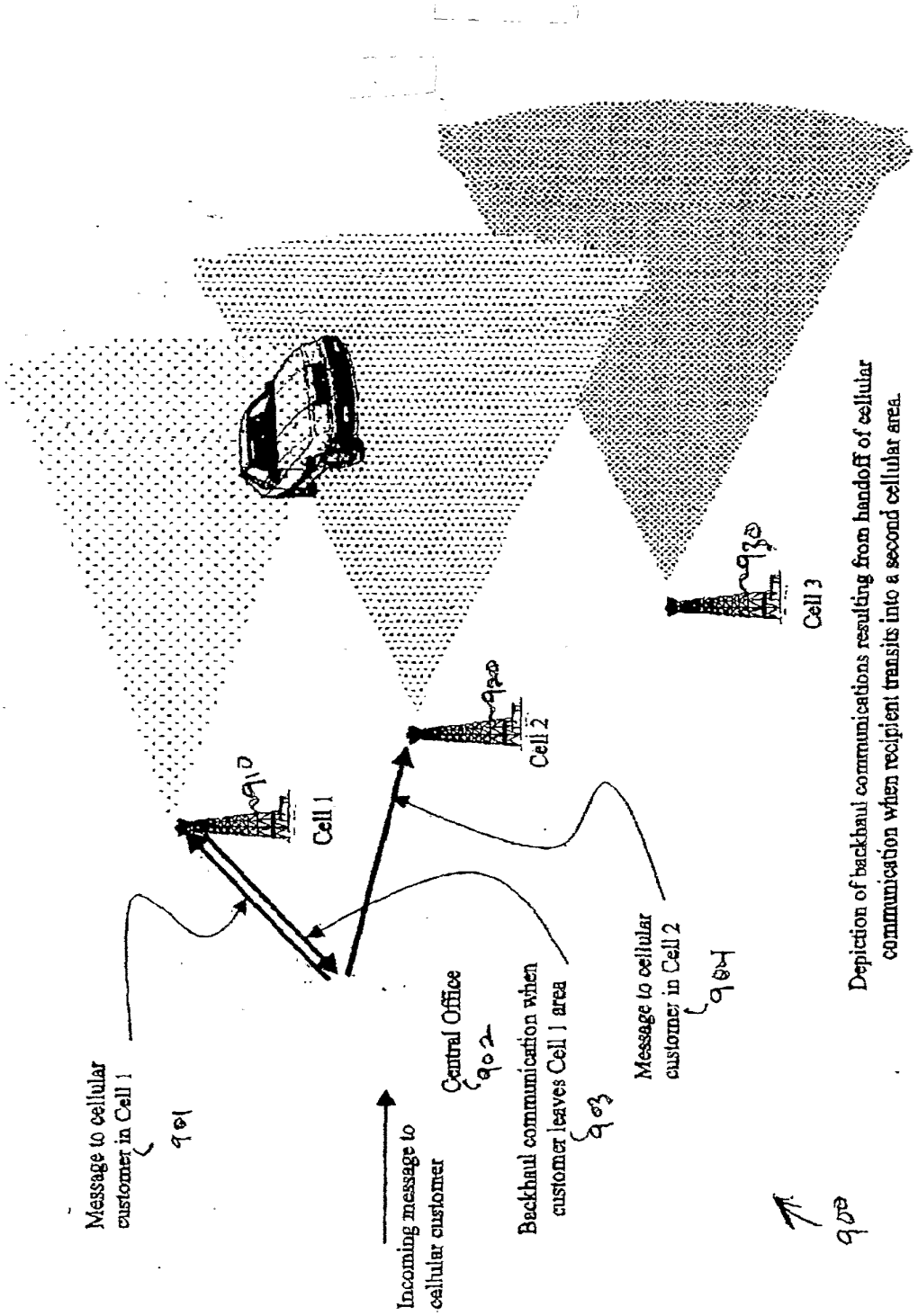
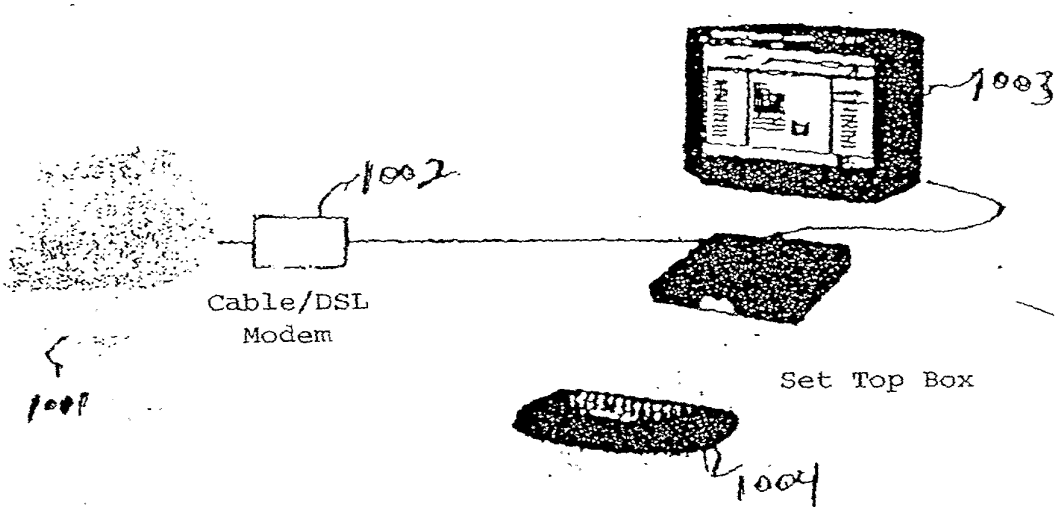

Video and Audio Content Server
835
Internet or Other Network
830
AI Compression Server and WAP Gateway
Deep AI Translator
Delivery AI
Protocol Gateway
WAP Stack TCP/IP
Wireless Network
825
820
Wireless Base Station
Mobile Phone
PDA
815
810
800

FIGURE 8

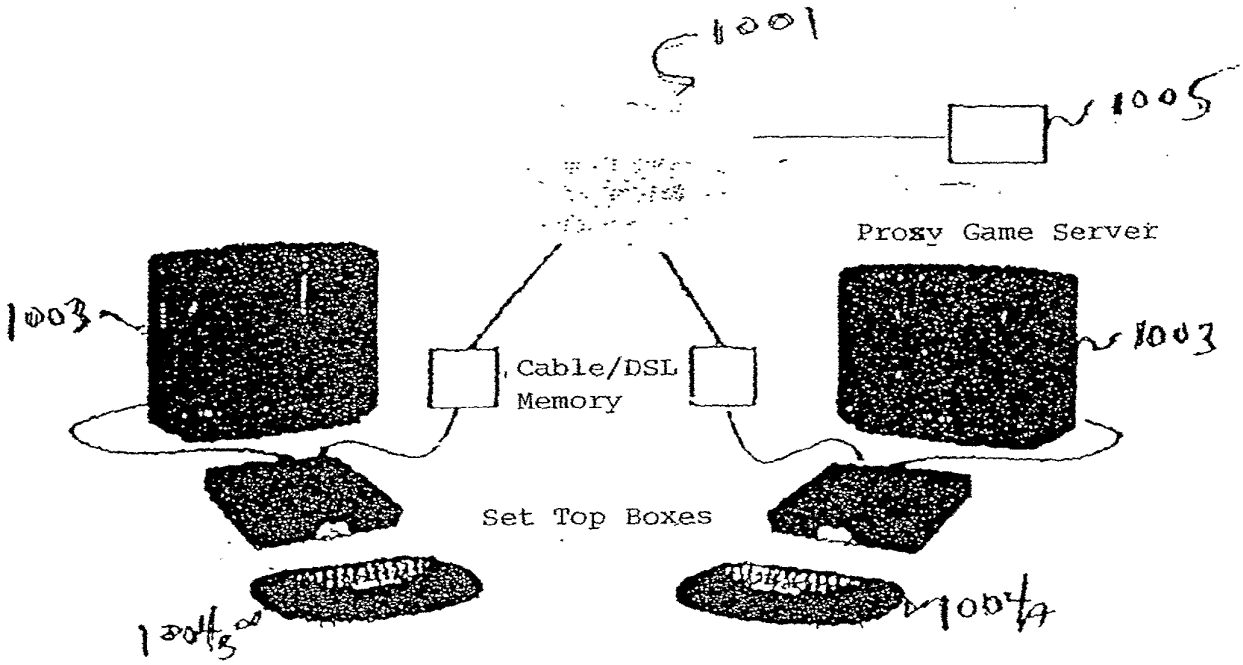
FIG. 9



Aylo Exhibit 1018
Page 1241 of 2308



Browsing via the Set Top Box



Interactive Gaming

Fig. 10

Aylo Exhibit 1018
Page 1242 of 2308

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US02/30874

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 15/16
US CL : 709/231,232,217-219,250,220; 704/243

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 709/231,232,217-219; 345/335,327

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6,195,692 B1 (HSU) 27 Febuary 2001, column 5 lines 4-19,	1-69
Y	US 6,243,676 B1 (WITTEMAN) 5 June 2001, column 7 line 55-col 8 line 56,	1-69
Y	US 6,115,755 A (KRISHAN) 5 September 2000, column 3 lines 19-45, column 4 lines 54-67	1-69
Y	US 6,085,236 A (LEA) 4 July 2000, column 18 lines 9-20, col 26 line 59-column 27 line 4,	1-69

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 December 2002 (10.12.2002)	15 JAN 2003
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Mark Powell Telephone No. 703-305-9703

Form PCT/ISA/210 (second sheet) (July 1998)

IDS CERTIFICATION STATEMENTS

I. THIS IDS IS BEING FILED UNDER 37 C.F.R. 1.97(b): (check one box)

☐ Within three months of the filing date of a national application other than a continued prosecution application under 1.53(d).

☐ Within three months of the date of entry of the national stage as set forth in 1.491 in an international application.

☐ Before the mailing date of a first Office Action on the merits. In the event that a first Office Action on the merits has been mailed, please consider this IDS under 37 C.F.R. 1.97(c) and see the statement under 1.97(e) **or** charge deposit account 50-2091 the fee specified by 37 C.F.R. 1.17(p) if such statement is not selected in Section IV.

☐ Before the mailing of a first Office Action after the filing of a request for continued examination under 1.114.

II. THIS IDS IS BEING FILED UNDER 37 C.F.R. 1.97(c):

☒ After the period specified in 37 C.F.R. 1.97(b) and before the mailing date of any of a final action under 1.113, a notice of allowance under 1.311 or an action that otherwise closes prosecution in the application, and accompanied by:

- ☐ a statement specified in 1.97(e) as subsequently provided in this IDS; or
- ▩ the fee set forth in 1.17(p).

III. THIS IDS IS BEING FILED UNDER 37 C.F.R. 1.97(d):

☐ After the period specified in 37 C.F.R. 1.97(c) and on or after the mailing date of any of a final action under 1.113, a notice of allowance under 1.311 or an action that otherwise closes prosecution in the application, but before or simultaneous with payment of the issue fee, and accompanied by a statement specified in 1.97(e) as subsequently provided in this IDS **and** the fee set forth in 1.17(p).

IV. Statement under 37 C.F.R. 1.97(e) (check only one box)

The undersigned hereby states that:

☐ Each item of information contained in the IDS was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this IDS.

☐ No item of information contained in the IDS was cited in a communication from a foreign patent office in a counterpart foreign application, and, to knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the IDS was known to any individual designated in 37 C.F.R. 1.56(c) more than three months prior to the filing of this IDS.

V. Statement under 37 C.F.R. §1.704(d):

☐ The undersigned hereby states that each item of information contained in the information disclosure statement: (i) was first cited in any communication from a patent office in a counterpart foreign or international application or from the Office, and this communication was not received by any individual designated in § 1.56(c) more than thirty days prior to the filing of the information disclosure statement; or (ii) is a communication that was issued by a patent office in a counterpart foreign or international application or by the Office, and this communication was not received by any individual designated in § 1.56(c) more than thirty days prior to the filing of the information disclosure statement..

VI. CONCLUSION

It is respectfully submitted that the claims presently on file patently distinguish the present invention from each of the references cited in this IDS. The references cited in this IDS are being cited only in the interest of candor and without any admission that they constitute statutory prior art or contain matter which anticipates the invention or which would render the same obvious, either singly or in a combination, to a person of ordinary skill in the art. Furthermore the filing of this IDS should not be construed to be an admission that the information cited in this IDS is, or is considered to be, material to patentability as defined in 1.56(b). In addition, the filing of this IDS should not be construed as a representation that a search has been made.

If the Examiner has any questions concerning this IDS, the Examiner is requested to contact the undersigned at the telephone number associated with Customer No. 110240. If it is determined that this IDS has been filed under the wrong rule, the PTO is requested to consider this IDS under the proper rule, with a petition if necessary, and charge the appropriate fee to Deposit Account No. 50-2091.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 April 2004 (29.04.2004)

PCT

(10) International Publication Number
WO 2004/036824 A1

(51) International Patent Classification[7]: H04L 9/06

(21) International Application Number: PCT/FI2003/000752

(22) International Filing Date: 10 October 2003 (10.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20021820 14 October 2002 (14.10.2002) FI

(71) Applicant *(for all designated States except US)*: **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **CURCIO, Igor, D.D.** [IT/FI]; Hatanpään valtatie 12 C 57, FIN-33100 Tampere (FI). **LUNDAN, Miikka** [FI/FI]; Männikönkatu 4 A 7, FIN-33820 Tampere (FI).

(74) Agent: **FORSSÉN & SALOMAA OY**; Eerikinkatu 2, FIN-00100 Helsinki (FI).

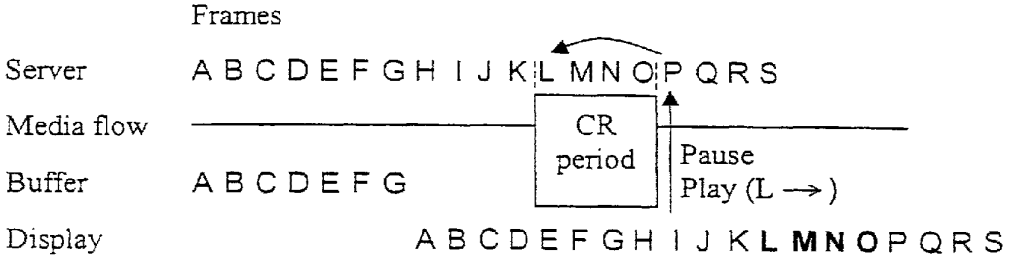
(81) Designated States *(national)*: AE, AG, AL, AM, AT (utility model), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, EG, ES, FI (utility model), FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT (utility model), PT, RO, RU, SC, SD, SE, SG, SK (utility model), SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: STREAMING MEDIA



(57) Abstract: The invention relates to a method for streaming of media from a streaming server (111) to a mobile client device (101) over an air-interface, wherein, after a cell reselection, the method comprises requesting the streaming server (111) to resend streaming media which the mobile client device (101) was not able to receive due to the cell reselection.

WO 2004/036824 A1

STREAMING MEDIA

FIELD OF THE INVENTION

The invention relates to streaming of media from a streaming server to a mobile client device over an air-interface.

BACKGROUND OF THE INVENTION

A packet switched streaming service PSS is currently being standardized for mobile environment by 3GPP (3rd Generation Partnership Project). An example of a communications system capable of streaming media (streaming video and/or audio) is shown in Figure 1. The system comprises a streaming server 111 which is coupled to an IP-network (Internet Protocol) 104. The IP-network 104 may be, for example, the Internet or a service provider operator's intranet (an intranet network belonging to the operator's domain). The IP-network 104 is coupled to a core network 103 of a mobile communications network via a G_i interface. The mobile communications network also has a radio access network (RAN) 102 coupled to the core network 103. The radio access network 102 provides mobile communications devices 101 with access to the mobile communications network over an air-interface. Said access may be provided either by circuit switched means (circuit switched voice or data call) or packet switched means or both. In the following, GPRS (General Packet Radio Service) is used as an example of a packet switched means to communicate over the air-interface. Concerning GPRS, the generic term radio access network (RAN) is considered to comprise base (transceiver) stations (BS) and base station controllers (BSC).

In streaming media a sequence of 'moving images' (i.e. video) or sound (i.e. audio) or a sequence of 'moving images' together with sound (i.e. multimedia) is sent in a compressed form from the streaming server 111 to a mobile communications device 101 (hereinafter referred to as a client device 101). In contrast to the

technique in which an entire media file has to be arrived at the client before it can be played, the streaming technique enables sending of media (video and/or audio) from the streaming server 111 to the client device 101 in a continuous manner and playing of the media as it arrives at the client.

When media is streamed in cellular mobile communications systems such as the system presented in Figure 1, new problems specific to the mobile environment arise. These problems are mostly due to the different restrictions of mobile systems. One such problem arises in a cell reselection (i.e. handover) situation presented in Figure 2.

In Figure 2, a first base station BS1 of the radio access network 102 serves the area of a first cell 201 whereas a second base station BS2 of the radio access network 102 serves the area of a second cell 202. Before cell reselection, the client device 101 is served by the first base station BS1, i.e. the client device 101 has an active radio link with the first base station BS1. This situation is shown on the left side of the Figure 2. After the cell reselection, the client device 101 is served by the second base station BS2. The radio link with the first base station BS1, shown by a dotted line, is dropped. This situation is shown on the right side of the Figure 2.

The cell reselection (CR) can be divided in three periods in time:

a) Pre-CR period
b) CR period
c) Post-CR period.

During the pre-CR period, received signal quality in the first cell weakens and the client device 101 starts cell reselection signaling. The client device 101 is able to receive streaming media via the first base station BS1 during that period. During the CR period, the actual cell reselection is performed. The client device 101 can

not receive streaming media during that period. During the post-CR period, the client device 101 is able to receive streaming media via the second base station BS2.

The cell reselection may cause a long break in service. For example, a cell reselection (CR period) in a case where GPRS is used as radio bearer may take up to 30-40 seconds. This will affect an ongoing streaming session. For example, part of the packets carrying streaming media may be lost and/or freezing of the played media may occur at the client device 101. If the streaming media comprises a video stream, the freezing means that a still image will appear and stay for a period of time on a client device display before the streaming media is again received and can be played again. If the streaming media comprises an audio stream, the freezing means that no sound is played back (i.e. silence) until the CR period is over.

Accordingly, there is a need to find appropriate means for reducing the effects of cell reselection on an ongoing streaming session.

SUMMARY OF THE INVENTION

According to a first aspect of the invention, there is provided a method for streaming of media from a streaming server to a mobile client device over an air-interface, wherein the method comprises:
requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection.

A request requesting the streaming server to send the streaming media may be sent from the mobile client device to the streaming server before or after the mentioned cell reselection.

The mentioned request is not limited to requesting to send only the media which

the mobile client device is (or was) not able to receive due to the cell reselection (during a cell reselection period). In an embodiment, the request may be construed such that what is actually requested is: resending of a portion of a media content not received as well as sending any remaining portion of the media content.

The term media is considered to mean either video or audio or another media, such as still image, or any combination thereof, i.e multimedia.

Preferably, a resending request is sent in response to the cell reselection. The re-sending request is preferably an application layer request sent between a mobile client device application and a streaming server application.

In a preferred embodiment, the streaming server is requested to send at an increased speed for a determined period of time in order to increase at the client device a degree of fullness of a temporary store, such as a buffer, in which the streaming media is temporarily stored before playback. Preferably, this is done by by requesting the streaming server to switch from sending a higher bit rate media stream to sending a lower bit rate media stream at an increased speed. The higher and lower bit rate streams are preferably provided by a multi-rate codec.

According to a second aspect of the invention, there is provided a mobile client device for receiving streaming media from a streaming server over an air-interface, the mobile client device comprising:
means for requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection.

Preferably, the air-interface couples said mobile client device to a mobile communications network. The mobile client device preferably comprises a cellular mobile phone.

According to a third aspect of the invention, there is provided a streaming server

for sending streaming media to a mobile client device over an air-interface, the streaming server comprising:
means for receiving a request requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection; and
means for acting upon the received request.

According to a fourth aspect of the invention, there is provided a system comprising a streaming server and a mobile client device, for streaming of media from the streaming server to the mobile client device over an air-interface, the system comprising, at the mobile client device:
means for requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection, the system further comprising, at the streaming server:
means for receiving the request; and
means for acting upon the received request.

According to a fifth aspect of the invention, there is provided a computer program executable in a mobile client device, the computer program comprising:
program code for causing the mobile client device to request the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection.

According to a sixth aspect of the invention, there is provided a computer program executable in a streaming server, the computer program comprising:
program code for causing the streaming server to receive a request requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection; and
program code for acting upon the received request.

Dependent claims contain preferable embodiments of the invention. The subject

matter contained in dependent claims relating to a particular aspect of the invention is also applicable to other aspects of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described by way of example with reference to the accompanying drawings in which:

Figure 1 shows a communications system capable of streaming media;

Figure 2 shows a cell reselection situation in the communications system of Figure 1;

Figure 3 shows a client device buffer in a situation in which the client device buffer is shorter than the time that cell reselection takes;

Figure 4 shows a client device buffer in a situation in which the client device buffer is longer than the time that cell reselection takes;

Figure 5 shows another illustration of the situation which is behind Figure 4;

Figure 6 shows preferred action in a preferred embodiment of the invention;

Figure 7 illustrates a preferred embodiment of the invention;

Figure 8 illustrates a preferred embodiment of the invention;

Figure 9 shows a client device according to a preferred embodiment of the invention;

Figure 10 shows a streaming server according to a preferred embodiment of the

invention; and

Figure 11 illustrates three ways of sending messages according to preferred embodiments of the invention.

DETAILED DESCRIPTION

The system shown in Figure 1 can also be used in the preferred embodiments of the invention (the same applies to Figure 2). Accordingly, the system comprises a streaming server 111 which is coupled to an IP-network 104. The IP-network 104 may be, for example, the Internet or a service provider operator's intranet. The IP-network 104 is coupled to a core network 103 of a mobile communications network via a G_i interface. The mobile communications network also has a radio access network (RAN) 102 coupled to the core network 103. The radio access network 102 provides mobile communications devices 101 with access to the mobile communications network over an air-interface. Said access may be provided either by circuit switched means (circuit switched voice or data call) or packet switched means or both. In the following, GPRS (General Packet Radio Service) is used as an example of a packet switched means to communicate over the air-interface.

In order to better understand the preferred embodiments of the invention, let us consider the following example.

In this example, a mobile communications device 101 (again referred to as a client device 101) has established a streaming session with the streaming server 111 when being served by the first base station BS1. In this example and in the following description, mostly a video streaming session is considered. However, a corresponding consideration applies to an audio streaming session as well as to a multimedia (e.g. video with audio) streaming session.

RTSP (Real Time Streaming Protocol) protocol is used in the streaming session

setup. Once the session has been established, the streaming itself may be performed (i.e. media flow may be sent) according to RTP (Real time Transport Protocol) or another protocol. However, if it is desired to make a change in the established session, this will again be done by using RTSP.

At the beginning of the established streaming session, the received streaming media (media flow), transmitted in packets (or frames), is buffered (i.e. temporarily stored) in a buffer at the client device (hereinafter referred to as a client device buffer). When the client device buffer is full, playing (playback) of the streaming media is started at one end of the buffer while the buffer is continuously filled at the other end with received streaming media. Therefore, the buffer should remain nearly full during normal operation of the system.

The client device now undergoes a cell reselection from the first cell 201 to the second cell 202 (Fig. 2). It is to be noted that, although the radio access network (RAN) 102 is illustrated in Figure 2 by a single cloud only, it may comprise different radio access networks. It may comprise a GPRS (GPRS-only) RAN, a 3G (third generation) RAN or a combination thereof. The first cell 201 (served by the base station BS1) and the second cell 202 (served by the base station BS2) may therefore be, for example, cells of the GPRS RAN or 3G RAN. Accordingly, the cell reselection (handover) can occur, for example, between base stations belonging to GPRS RAN, between base stations belonging to 3G RAN or between a base station belonging to GPRS RAN and a base station belonging to 3G RAN.

The client device 101 can not receive streaming media during the CR period. Therefore, when the CR period begins the client device buffer starts emptying because the buffer content is further played while the continuous filling is stopped. Depending on the size of the client device buffer, two different cases can be identified:

i) The client device buffer (in time) is shorter than the time that cell reselection

takes, i.e. the time to empty a full client device buffer is shorter than the CR period,

ii) The client device buffer (in time) is longer than the time that cell reselection takes, i.e. the time to empty a full client device buffer is longer than the CR period.

In the first case (Figure 3), the buffer will become totally empty during the CR period. Since the buffer becomes empty, the client device 101 has no media to play anymore. Therefore, the client device 101 starts rebuffering. Rebuffering can be started during the post-CR period. In order to get the rebuffering started the client device 101 can use an RTSP PAUSE/PLAY method. In this method, the client device 101 first sends an RTSP PAUSE message to the streaming server 111. The PAUSE message will cause the server 111 to pause sending of the media flow. Subsequently, the client device 101 sends an RTSP PLAY message which requests the streaming server 111 to start resending the streaming media from the last received frame onwards. Rebuffering starts when the client again starts receiving the streaming media. When the client device buffer is full, playing of the streaming media is restarted from the last received frame onwards. In the meantime, the last received frame is shown as a still image on a client device display. However, in spite of the drawback of showing a still image, no packets are effectively lost since the media flow which is not received at the client device during the CR period is resent from the server 111 to the client device 101 after the CR period.

In the second case (Figure 4), the whole content of the buffer has not been played during the CR period. Therefore, the buffer only becomes partially empty during the CR period, i.e. there is still streaming media (DATA) in the buffer when the CR period ends. During the post-CR period (after the CR period) playing of the streaming media stored in the buffer before the CR period (i.e. DATA*) is further continued at one end of the buffer while the other end is filled with the streaming

media received after the CR period (i.e. DATA**). If nothing is done, DATA* will eventually end and playing is continued with the first frame of data DATA**. No still image is shown. However, the first frame of DATA** is not the frame that should be played right after the last frame of DATA*. Since the client device 101 could not receive streaming media during the CR period it has not even received the frame that should be showed after the last frame of DATA*. There exists a temporal gap in the buffered media. As a result, the client device 101 will skip showing the sequence of streaming media (video) which could not be received due to the cell reselection Concerning an audio stream, the same effect occurs, i.e. the client device 101 will skip playing sound which could not be received.

The second case is further illustrated with the aid of Figure 5. It is to be noted that Figure 5 only shows a very simplified case. The time is considered to flow from left to right in Figure 5. The streaming media sequence that the server sends comprises video frames A-S. The frames are received at the client device and buffered before playing. The playing of the media starting from frame A starts when the buffer is full (i.e. when the frames A-G have been stored in the buffer), the frames (moving video picture) are subsequently shown on a client device display. The buffering of the frames following frame G is not shown in Figure 5 for clarity reasons.

The media flow from the server to the client device is not possible during the CR period. Therefore, the client device does not receive the frames L-O. Otherwise, the media flow is perfectly received at the client device. Now that the the CR period is shorter than the buffer size, the showing of the frame (frame K) last received before the CR period will be followed by the showing of the frame (frame P) first received after the CR period. Video frames L-O are not shown at all. Concerning an audio stream, the same effect occurs, i.e. the user hears a pause and a temporal gap in the sound.

There exists two type of frames: intra frames and inter frames. The intra frames

contain all necessary information of an image whereas inter-frames only contain changes or predicted changes compared to a previous image. The previous image may be an intra frame or an inter frame. Therefore, if the frame P is an intra frame the shown picture only warps ahead in time from frame K to frame P. In this case the frames L-O are not simply shown to the user. But if the frame P is an inter frame severe distortion in the shown (moving) picture is likely to occur before the next intra frame is received and played.

A preferred embodiment of the invention as shown in Figure 6 concentrates on the above mentioned problem relating to the second case. In this embodiment, the buffer is longer than the duration of the CR period. Therefore, the buffer does not become totally empty during the CR period but only partially empty. It is to be noted that Figure 6 only shows a very simplified case. The time is considered to flow from left to right in Figure 6. The streaming media sequence that the server sends comprises frames A-S. The frames are received at the client device and buffered before playing. The playing of the media starting from frame A starts when the buffer is full, the frames (moving video picture) are subsequently shown on a client device display. The buffering of the frames following frame G is not shown in Figure 6 for clarity reasons. The media flow from the server to the client device is not possible during the CR period. Therefore, the client device does not receive the frames L-O. Otherwise, the media flow is perfectly received at the client device.

When the CR period is over the client device knows exactly what is the last frame that it has received before the CR period (i.e. during the pre-CR period). In this embodiment it is frame K. The client device then requests, right after the CR period has been ended, the streaming server to start resending the streaming media from the last received frame onwards. Upon receiving the request the streaming server starts resending the media flow. The first frame to be sent is in this embodiment frame L. Any frames sent and received after the CR period but before the resending has started (possibly some frames starting from frame P) are

ignored by the client device. When the streaming media (of frames A-K) stored in the buffer before the CR period ends, then the playing is continued with the first frame (frame L) of the resent streaming media received after the CR period. Therefore, no discontinuity in the shown video picture should appear to the user. In Figure 6, the resent video frames L-O which would have otherwise failed to have been received at the client device are shown in bold font. Correspondingly, concerning an audio stream, the streaming server is requested to start resending the audio stream at a point at which reception has been stopped in the client device. No discontinuity in playback should appear.

In practice, the resending request can be performed by the RTSP PAUSE/PLAY method. In this method, the client device first sends, after the CR period has been ended, an RTSP PAUSE message to the streaming server. The PAUSE message will cause the server to pause sending of the media flow. However, the playing of the streaming media received at the client device is not paused unless the buffer gets empty (which should not be the case). Subsequently, the client device sends an RTSP PLAY message to the streaming server 111. The PLAY message contains information on the starting point of the resending. At the end of the CR period the client device knows the time of the last received frame. Based on that, the client device determines the starting point before sending the PLAY message. The PLAY message causes the server to start resending.

An example of the PAUSE message is as follows:

```
PAUSE rtsp://example.com/foo RTSP/1.0
CSeq: 6
Session: 354832
```

The PAUSE message informs the server that a change is coming. An example of the subsequently sent PLAY message is as follows:

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 7
Session: 354832
Range: npt=28.00-

The message field 'Range' tells the starting point of resending. In this example, the starting point is 28 seconds from the start of the streaming media sequence. In relation to the embodiment shown in Figure 6, this point would be exactly the time of frame L in the streaming media sequence.

In the embodiment just described, the frames (packets) that were lost during the CR period will be resent from the server to the client devide. Also, since the buffer size is longer than the duration of the CR period, no interruption in the shown video picture should appear and the user experience is maximized.

However, as the case is such that during normal streaming the degree of fullness of the buffer does not change, since the buffer is emptied (played) at the same rate as it is filled, this may cause an additional problem. In the embodiment just described, after the cell reselection, the client device buffer stays emptier than before, for the reasons as described in the foregoing. Therefore if, for example, a new cell reselection is to be performed in a near future, the emptier buffer may cause the same drawback as discussed in connection with the first case above (i.e. the case in which the client device buffer was shorter than the time that cell reselection takes).

A preferred embodiment of the invention concentrates on the above identified problem. In this embodiment, to ensure smooth behaviour of the shown video picture (correspondingly: played sound), the client device buffer is filled after the CR period, for a period of time, at a higher rate than it is emptied (played). This period may be called a filling period. When the filling period is over the buffer is full again and normal streaming in which the buffer is filled at the same rate as

emptied is resumed.

It is to be noted that normally raising the degree of fullness of the buffer would require the playing to be paused. In the present embodiment, playing is not paused but the degree of fullness of the buffer can still be raised due to the clever buffer management in which the buffer is filled, during playing, at a rate higher than the playing rate.

To raise the degree of fullness of the buffer without pausing the playback, the client device, in this embodiment, requests the server to switch to sending a lower bit rate streaming media sequence, but to use in the actual sending the same transmission bit rate (hereinafter referred to as an original transmission bit rate) as before. In order to reach the original transmission bit rate the client device requests the server to speed up the transmission of the lower bit rate sequence by a speeding factor. Speeding up the transmission bit rate causes more data to be written in the buffer than read out from the buffer. Therefore, the degree of fullness of the buffer increases as desired.

In other words, the server is asked to switch from sending the original sequence encoded at a first bit rate to sending a new corresponding sequence encoded at a second bit rate lower than the first bit rate and to increase the transmission bit rate of the new sequence in order to reach the original transmission bit rate (bandwidth). It is to be noted that the bit rate at which a media stream is encoded (and will be decoded) is a different concept than the transmission bit rate. The bit rate at which the media stream has been encoded has an effect on the picture quality. If the media stream has been encoded at a higher bit rate, this means that more bits have been used in encoding compared to encoding at a lower bit rate. This typically results in a better picture quality. The transmission bit rate, on the other hand, is the bit rate at which the media stream is actually sent and it depends on the available bandwidth.

The request to switch to sending a lower bit rate sequence and to speed up the transmission bit rate can be delivered by using the RTSP PAUSE/PLAY method. When the client device desires to fill the client device buffer during the post-CR period, it first sends a PAUSE message, corresponding to the one presented already in the preceding description, to the server to indicate that a change is coming. Subsequently it generates a PLAY message and sends it to the server.

An example of such a PLAY message is as follows:

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 7
Session: 354832
Range: npt=28.00-40.00
Bandwidth: 20000
Speed: 1.5

This message contains two optional message fields, namely 'Bandwidth' and 'Speed' to be understood at the client device and streaming server. These fields are already specified as optional fields by the IETF (Internet Engineering Task Force) in the standard RFC 2326 (Real Time Streaming Protocol).

The message field 'Bandwidth' tells the server to change into sending a lower bit rate sequence (here: a sequence whose bit rate is 20 kbps) and the message field 'Speed' tells the server to speed up the sending by a speeding factor (here: 1.5). The message field 'Range' tells the starting point and stopping point in the streaming media sequence in time units (calculated from the beginning of the streaming media sequence(s)).

The PLAY message shown would suit for an example in which the server initially sends a 30 kbps bit rate sequence at an original transmission bit rate of 30 kbps and in which the server, in order to fill the buffer, is desired to switch into sending

a sequence whose bit rate is 20 kbps and to speed up the sending at a speeding factor 1.5 to reach the original transmission bit rate of 30 kbps during a 12 seconds filling period (i.e. between the time instants of 28 and 40 seconds of the media sequence).

If the client knows the possible bit rate options (they are typically communicated from the server to the client during session setup, for example, with the aid of a 200 OK response to an RTSP DESCRIBE message) it can calculate the speeding factor needed to reach the original bitrate using the following formula:

$$speeding\ factor = \frac{original\ transmission\ bit\ rate}{new\ transmission\ bit\ rate}$$

When the buffer is full, the client device sends another RTSP PAUSE and PLAY message pair to the server. An example of the PLAY message is as follows:

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 67
Session: 354832
Range: npt=40.00-
Bandwidth: 30000
Speed: 1.0

This exemplary message requests the server to start sending the (original) 30 kbps bit rate media sequence (Bandwidth: 30000) at the original transmission bit rate 30 kbps (Speed: 1.0) starting from the time instant 40 seconds of the media sequence (Range: npt=40.00-).

The client device can calculate the length of the buffer filling period using the following formulas:

$$filling\ period = \frac{LowSeqTime}{speeding\ factor}$$

wherein

$$LowSeqTime = (BufferSize - BufferData) \cdot \frac{original\ transmission\ bit\ rate}{original\ transmission\ bit\ rate - new\ transmission\ bit\ rate}$$

In these formulas, *LowSeqTime* indicates the duration of the lower bit rate sequence playing time in the client device, *BufferSize* indicates the size of the buffer in seconds and *BufferData* indicates the data left in the buffer in seconds.

The filling of the buffer is illustrated in Figure 7. It is to be noted that Figure 7 only shows a very simplified case. All delays, for example, have been ignored. The time is considered to flow from left to right in Figure 7. It can be seen that the buffer starts emptying when the CR period starts. Upon receiving the first PLAY message the server switches to sending a media sequence having a lower bit rate (here: 20 kbps) and speeds up the sending by a speeding factor (here again: 1.5). Since the speeding factor is 1.5, the buffer filling takes twice the time that the emptying period lasted. For example, if the emptying period lasted for 25 seconds, the filling period should be 50 seconds with this speeding factor. When the buffer is full the second PLAY message is sent and the server switches to sending the original media sequence having a higher bit rate (here: 30 kbps) and continues sending at the original transmission bit rate.

In the foregoing, it has been described that, for example, a video stream typically comprises both intra and inter frames, wherein the intra frames are 'independent frames' containing all necessary information of an image whereas inter-frames only contain changes or predicted changes compared to a previous image. In the following preferred embodiment of the invention, the timing of the switch from the lower bit rate sequence to the original bit rate sequence is more closely de-

scribed with this respect.

In this embodiment the purpose is to time, whenever possible, the switch from the lower bit rate sequence to the original bit rate sequence to take place at the point of an intra frame of the original bit rate sequence (or, more generally, at the point of an intra frame of the sequence to which the sending is switched). In this way, prediction errors and/or jumps in media (frame) flow can be avoided. Figure 8 illustrates this embodiment. It is to be noted that Figure 8 only shows a simplified case. The time is considered to flow from left to right in Figure 8. In this embodiment, the first switch from the original bit rate sequence to the lower bit rate sequence is effected at point 28 seconds with the aid of a first PAUSE/PLAY message pair.

As illustrated in Figure 8, the position of intra frames in the original bit rate sequence may differ from the ones of the lower bit rate sequence. However, based on:

- the time required to totally fill the buffer, i.e. '*filling period*',
- the time distance of two adjacent intra frames in the original sequence, i.e. an intra frame rate, '$IFrameTime_{Original}$', and
- the time at which the first switch from the original bit rate sequence to the lower bit rate sequence is effected, i.e. '*SwitchTime*',

it can be calculated a suitable point in time for swiching back to the original bit rate sequence. The parameter '$IFrameTime_{Original}$' can be calculated in the client device from the original bit rate sequence before switching to the lower bit rate sequence. The calculation of the suitable point in time for swiching back to the original bit rate sequence can be performed using the following formula:

$$SeqChangeTime = IFrameTime_{Original} * \left\lfloor \frac{SwitchTime + filling\ period}{IFrameTime_{Original}} \right\rfloor$$

In this formula, '*SeqChangeTime*' indicates the point in time for swiching back to the original bit rate sequence and the brackets indicate a *floor*-function (or *trunc*-function) which truncates the fractional part of the value computed in the brackets. For example, if the time required to totally fill the buffer, i.e. '*filling period*' is 16 seconds, the time at which the first switch is effected, i.e. '*SwitchTime*' is 28 seconds and the time difference between two adjacent intra frames in the original sequence, i.e. '$IFrameTime_{Original}$' is 5 seconds, then the point in time for swiching back to the original bit rate sequence is 40 seconds (*SeqChangeTime*=5*floor((28+16)/5) = 40 s). Accordingly, the second PAUSE/PLAY message pair is sent at point 40 seconds, as indicated in Figure 8. In this embodiment, the starting point and stopping point to be placed in the message field 'Range' of the PLAY message are 28 seconds and 40 seconds, respectively.

It is to be noted that, due to the intra frame rate, it might not in all cases be possible to totally fill the buffer. For example, in the example just described, it would have taken 4 seconds more to totally fill the buffer. However, the formula gives the closest suitable intra frame to the buffer filling period.

In another embodiment, it may well be that, at the point of the last received frame before the CR period, there is not an intra frame in the lower bit rate media sequence. In this embodiment, the starting point of the resending request is adjusted by a necessary amount of frames (or time) backwards such that there will be an intra frame in the lower bit rate sequence at the starting point. In this case, a set of last received frames before the CR period, which belong to a time period subsequent to the starting point, are ignored at the client device in order to guarantee a continuous playback.

In another embodiment of the invention, the switches between two different bit rate sequences are performed without the timing method(s) just described. In this embodiment, the first switching point (from higher to lower bit rate sequence) is directly determined by the last received frame and the second switching point (from lower to higher bit rate sequence) is directly determined by the the time required to totally fill the buffer (*'filling period'*). The switching points can therefore end up being at the point of an intra or inter frame. Should a switching point end up being at the point of an inter frame (for example a P-frame), a small prediction error may occur in the played media, however, in this embodiment, the buffer can be totally filled.

Another way to perform switching is to use so called "switch frames". These are frames which contain "difference" information between corresponding frames in different bit rate sequences. In this embodiment, a bridge and switch between the two sequences is performed with the aid of these frames using the difference information.

Figure 9 shows a client device 101 according to a preferred embodiment of the invention. The client device may be a mobile station of a cellular radio telephone network. The client device 101 comprises a processing unit MCU, a radio frequency part RF, and a user interface UI. The radio frequency part RF and the user interface UI are coupled to the processing unit MCU. The user interface UI typically comprises a display, one or more speaker and a keyboard (not shown) with the aid of which a user can use the device 101.

The processing unit MCU comprises a processor (not shown), a memory 210 and computer software. The software has been stored in the memory 210. The aforementioned client device buffer 240 is also comprised in the memory 210. The processor controls, in accordance with the software, the operation of the client device 101, such as receiving of the streaming media sent from the server 111 and sending of requests to the server 111 via the radio frequency part RF, reading and

writing of received streaming media (video and/or audio) in the buffer 240, and presenting of the received streaming video on the display and audio on the one or more speaker of the user interface UI. A suitable size of the buffer (in time) can be, for example, 1.5 or 2 times the maximum (or average) CR period time.

The software comprises a streaming client software application 220 (hereinafter referred to as client software 220), a protocol stack 230 for implementing the necessary protocol layers such as an RTP layer, an RTSP layer, an SDP (Session Description Protocol) layer, a TCP layer (Transmission Control Protocol), an IP layer and, below the IP-layer, lower protocol layers. In addition, the software comprises as a part of the client software 220 a media player for playing the received media.

The processor generates, based on the client software 220 the aforementioned PAUSE and PLAY messages and sends them to the server 111 via the radio frequency part RF. The processor also performs, based on the client software 220, the necessary calculations relating to the speeding factor, relating to the buffer filling period and relating to the suitable point in time for swiching from a lower bit rate sequence back to the original bit rate sequence.

The generation and sending of the resending request (PLAY message) and other appropriate action is triggered by a cell reselection event occurred in the mobile client device 101. The event can be detected by the client software 220 by reception of an asynchronous message from an API (Application Programming Interface) provided by the lower layers of the protocol stack 230. Alternatively or in addition, the event can be detected by monitoring the buffer level, i.e. the degree of fullness of the client device buffer 240. In this case, if the buffer 240 does not receive data for a certain amount X of time (depending on the implementation, the parameter X can be defined as a constant value, and it is a threshold for the client software 220 to understand that the cell reselection event has occurred), and if the client device 101 afterwards starts to receive data after a certain variable amount

Y of time (wherein Y > X, and Y is the real duration of the CR period), then the client can trigger the action described in this description.

Figure 10 shows a streaming server 111 according to a preferred embodiment of the invention. The streaming server 111 comprises a processing unit CPU, a first memory 310, an IP network interface 350, and a second memory 360. The first memory 310, the IP network interface 350, and the second memory 360 are coupled to the processing unit CPU.

The processing unit CPU controls, in accordance with computer software stored in the first memory 310, the operation of the streaming server 111, such as processing of requests received from the client device 101 and the sending of pre-recorded media streams, stored in the second memory (disk) 360, to the client device 101 via the IP network interface 350.

The software comprises a streaming server software application 320 (hereinafter referred to as server software 320), a protocol stack 330 for implementing the necessary protocol layers such as an RTP layer, an RTSP layer, an SDP layer, a TCP layer, an IP layer and lower protocol layers.

The PAUSE and PLAY messages sent from the client device 101 are received via the IP network interface 350. A processor (not shown) of the processing unit CPU processes the messages according to the server software 320 and the protocol stack 330 and takes appropriate action.

The present invention provides means for reducing the effects of cell reselection on an ongoing streaming session. It is to be noted that, according to preferable embodiments of the invention, a request for resending (e.g. a PLAY message) is sent on the application layer, i.e. between the client software application 220 and the server software application 320. Preferably RTSP over TCP (Transmission Control Protocol) or RTSP over another reliable protocol is used in transferring

application layer requests from the client device to the streaming server. The reception of the messages at the streaming server can therefore basically be guaranteed.

Although it has been described that the cell reselection would be performed between two base stations, it should be noted that a cell reselection can be performed also between two sectors of one and the same base station. Also, it should be noted that depending on the implemention, it may well be that separate messages (PAUSE messages) pausing the sending of streaming media are not needed. In an alternative embodiment, both to stop sending streaming media and to start resending streaming media is caused by a single appropriate message.

Further, in relation to the buffer filling embodiment, it has been described in the foregoing that when the server switches to sending a lower bit rate sequence, the original transmission bit rate would be maintained. However, in an alternative embodiment of the invention, a transmission bit rate higher than the original transmission bit rate is used during the filling period in order to fill the buffer more quickly. In this embodiment, it is assumed that a larger bandwidth can be requested by the mobile client device and that a larger bandwidth is actually available in the radio access network.

Further, in relation to the buffer filling embodiment, the bandwidth information of the sequence can be sent to the streaming server 111 also by other means than the RTSP field 'Bandwidth' (for example, by requesting a specific sequence encoded at a bitrate known to the client device 101). In this case, the field 'Bandwidth' is not used, the fields 'Speed' and 'Range' are recalculated according to the actual known sequence bit rate.

It is possible that the network (air-interface) bandwidth changes during the buffer filling period. If the client device 101 supports bandwidth adaptation that involves bit stream switching, the client device 101 should pause the buffer filling (with an

RTSP PAUSE message) before sending a bandwidth adaptation message to the streaming server 111. After the bandwidth adaptation operations are over, the client device can start the buffer filling again and recalculate the 'Bandwidth', 'Speed' and 'Range' values according to the bit rate and timing information of the new media stream.

Also, in relation to the buffer filling embodiment, it should be noted that, in an alternative embodiment, the second PAUSE/PLAY messages are not sent at all but the switch back to sending the original media sequence at the original transmission bit rate will be performed automatically by the server based on the stopping point information contained in the first PLAY message.

The resending request mentioned in the various embodiments of the invention may, in certain cases, actually be a sending request. One such case is considered as an alternative embodiment of the invention. In this embodiment, the client device (101) knowing beforehand that a cell reselection is going to occur in a very near future sends a PAUSE message to the streaming server (111) before the beginning of the cell reselection period (i.e. during the pre-CR period). Sending of the PAUSE message is triggered by a cell reselection initiation event notified to the client software 220 by means of the lower layer API. The PAUSE message causes the streaming server 111 to stop sending the streaming media. When the CR-period is over, the client device 101 then sends a PLAY message causing the streaming server to start sending at the point at which sending was stopped before the CR period. Playing of the streaming media is not stopped at the client device 101 in between, and if the client device buffer 240 has been selected to be longer in time than the time that cell reselection takes, the buffer 240 does not become totally empty during cell reselection and the user does not experience any jumps or interruption in playback. The PLAY message may contain a request to send at an increased speed in order to increase the degree of fullness of the buffer 240.

In yet another embodiment, the mobile client device 101 sends, before the

CR period, to the streaming server 111 an appropriate message requesting the streaming server 111 to stop sending the streaming media and to start sending again at a suitable point in time after the CR period. Sending of the message is triggered, during the pre-CR period, by a cell reselection initiation event notified to the client software 220 by means of the lower layer API. The mobile client device 101, knowing approximately the time that the cell reselection is going to take, estimates a suitable point in time at which it is again able to receive data. It inserts this information to the message so as to prevent the streaming server 111 from starting to send again too early.

An example of the above mentioned message is as follows:

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 7
Session: 354832
Range: npt=28.00-40.00; **time=19970123T153600Z**
Bandwidth: 20000
Speed: 1.5

This is an RTSP PLAY message having a header field 'Time'. The value of that field schedules the start of the "future" sending of the streaming media.

Figure 11 illustrates (especially relating to the buffer filling embodiments) the following three ways in which messages may be sent:

Case 1: The first PAUSE and PLAY messages are sent after the CR period. The second PAUSE and PLAY messages are sent when the buffer 240 is full.

Case 2: The first PAUSE message is sent before the CR period. The associated PLAY message is sent after the CR period. The second PAUSE and PLAY messages are sent when the buffer 240 is full.

Case 3: The first PAUSE message is sent before the CR period. The associated PLAY message is sent after the PAUSE message but before the CR period. The second PAUSE and PLAY messages are sent when the buffer 240 is full.

The sending of the second PAUSE message is not necessary, if the first PLAY message contains a closed 'Range' field.

Particular implementations and embodiments of the invention have been described. It is clear to a person skilled in the art that the invention is not restricted to details of the embodiments presented above (e.g. message names and message field names), but that it can be implemented in other embodiments using equivalent means without deviating from the characteristics of the invention. The scope of the invention is only restricted by the attached patent claims.

CLAIMS:

1. A method for streaming of media from a streaming server (111) to a mobile client device (101) over an air-interface, wherein the method comprises:

requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection.

2. A method according to claim 1, wherein the streaming server is provided with a starting point at which to start sending.

3. A method according to claim 1, wherein streaming server (111) sends the streaming media which the mobile client device (101) is not able to receive due to said cell reselection as well as a remaining portion of streaming media in response to the request.

4. A method according to claim 1, wherein the cell reselection comprises a cell reselection period during which the mobile client device (101) is not able to receive streaming media, the method comprising:

sending from the mobile client device (101) to the streaming server (111), after the cell reselection period, a resending request which requests the streaming server (111) to resend streaming media which the mobile client device (101) was not able to receive during the cell reselection period.

5. A method according to claim 4, wherein the resending request is generated according to RTSP protocol (Real Time Streaming Protocol).

6. A method according to claim 4, wherein the resending request is implemented by an RTSP PAUSE/PLAY message pair.

7. A method according to claim 1, wherein the streaming media is temporarily stored in a temporary store (240), such as a buffer, at the client device (101)

before playing.

8. A method according to claim 7, wherein the temporary store (240) has a size longer in time than a cell reselection period.

9. A method according to claim 7, wherein the streaming server is requested to send streaming media at a rate higher than the playing rate of that media so as to increase a degree of fullness of the temporary store (240).

10. A method according to claim 9, wherein a bandwidth or desired transmission bit rate with speeding factor is communicated to the streaming server (111) in a request.

11. A method according to claim 9, wherein the streaming media is stored at the mobile client device (101) at a rate higher than the playing rate.

12. A method according to claim 9, wherein the streaming server (111) is subsequently requested to resume an original configuration.

13. A method according to claim 7, wherein a degree of fullness of the temporary store (240) decreases during the cell reselection, and the streaming server is requested to send the not received streaming media although the temporary store (240) has not become totally empty, and said requesting is performed without pausing playback at the mobile client device (101).

14. A method according to claim 1, wherein the streaming server has a set of media streams available for transmission in which the same media content has been encoded at different bit rates.

15. A method according to claim 14, wherein information on the available set of media streams is beforehand communicated to the mobile client device (101)

in a streaming session setup.

16. A method according to claim 15, wherein the streaming server (111) is requested to switch from sending a higher bit rate media stream to sending a lower bit rate media stream at an increased speed.

17. A method according to any preceding claim, wherein the streaming media comprise one of the following: a video stream, an audio stream, another stream of single media, a multimedia stream.

18. A method according to any preceding claim, wherein the streaming server (111) sends streaming media to the mobile client device (101) via a mobile communications network.

19. A method according to any preceding claim, wherein the mobile communications network comprises a mobile packet radio network, such as a GPRS (General Packet Radio Service) network.

20. A method according to any preceding claim, wherein said cell reselection is performed between two base stations (BS1, BS2) which are selected from a group comprising: base stations belonging to a GPRS system, base stations belonging to a third generation mobile communications system.

21. A mobile client device (101) for receiving streaming media from a streaming server (111) over an air-interface, the mobile client device (101) comprising:

means (220, 230, MCU) for requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection.

22. A streaming server (111) for sending streaming media to a mobile client device (101) over an air-interface, the streaming server (111) comprising:

means (350) for receiving a request requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection; and

means (320, 330, CPU) for acting upon the received request.

23. A system comprising a streaming server (111) and a mobile client device (101), for streaming of media from the streaming server (111) to the mobile client device (101) over an air-interface, the system comprising, at the mobile client device (101):

means (220, 230, MCU) for requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection, the system further comprising, at the streaming server (111):

means (350) for receiving the request; and

means (320, 330, CPU) for acting upon the received request.

24. A computer program executable in a mobile client device (101), the computer program comprising:

program code for causing the mobile client device (101) to request the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection.

25. A computer program executable in a streaming server (111), the computer program comprising:

program code for causing the streaming server (111) to receive a request requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection; and

program code for acting upon the received request.

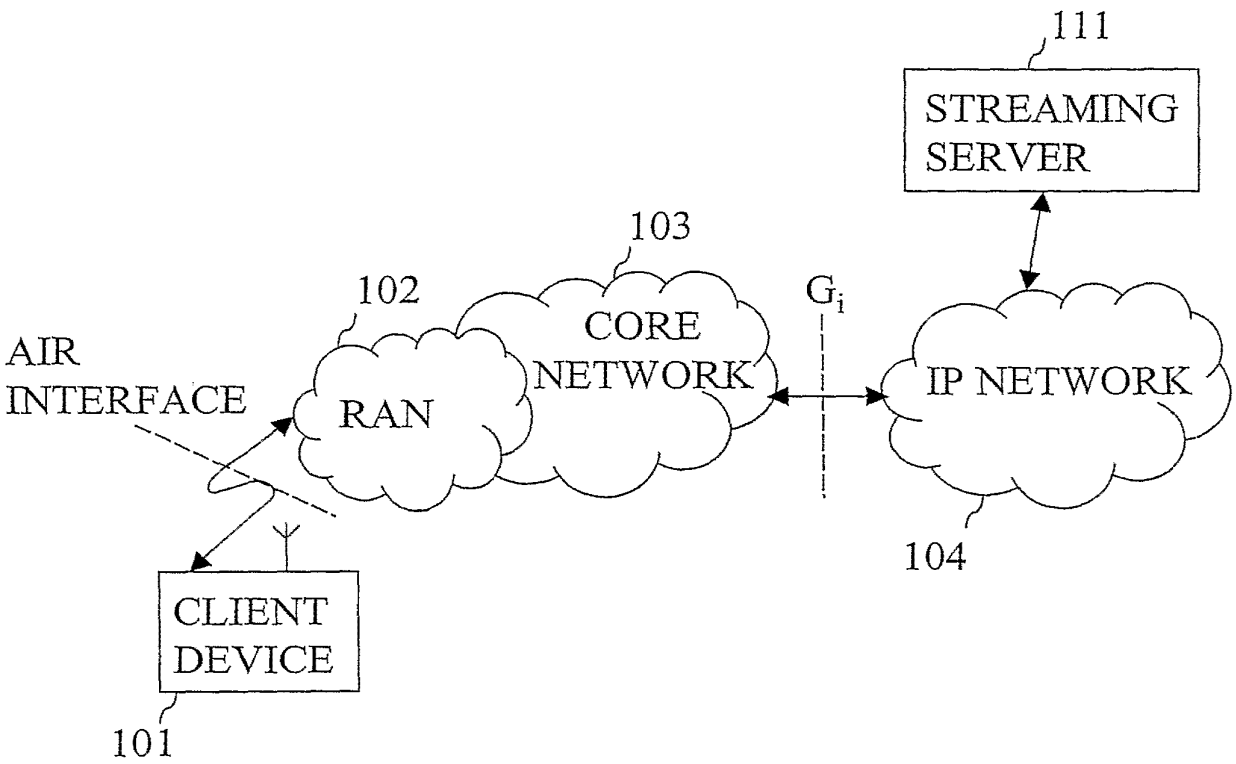
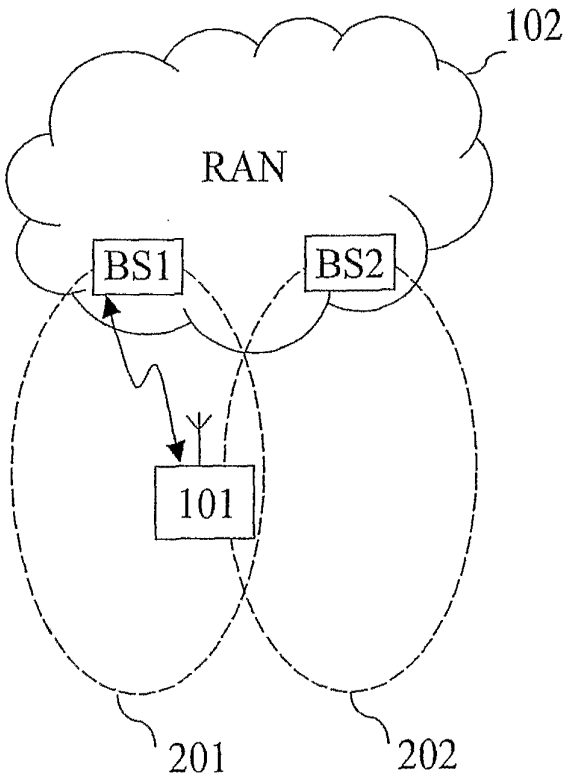

111
STREAMING SERVER
103
102
CORE NETWORK
G_i
AIR INTERFACE
RAN
IP NETWORK
104
CLIENT DEVICE
101

FIG. 1

BEFORE CELL RESELECTION:



AFTER CELL RESELECTION:

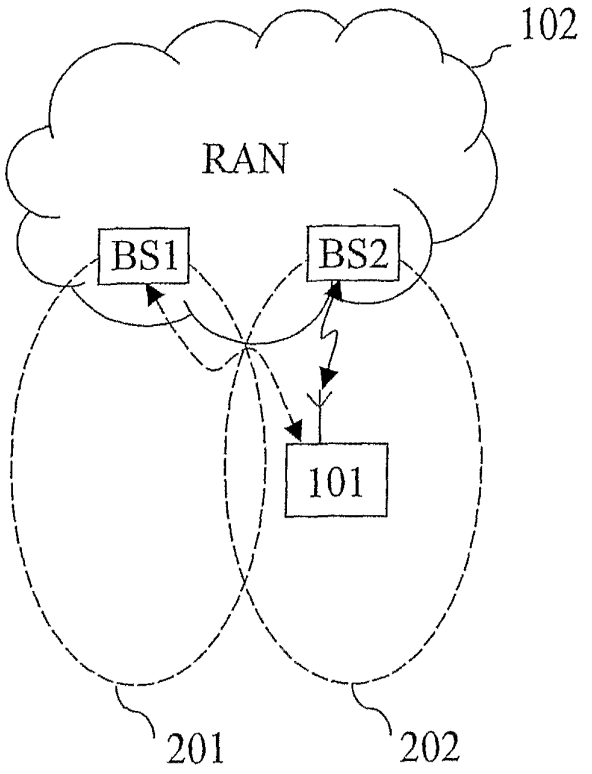


FIG. 2

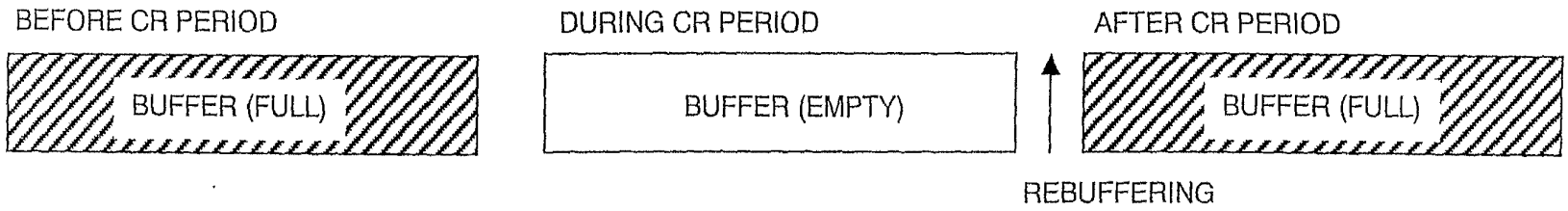

BEFORE CR PERIOD
BUFFER (FULL)
DURING CR PERIOD
BUFFER (EMPTY)
AFTER CR PERIOD
BUFFER (FULL)
REBUFFERING

FIG. 3

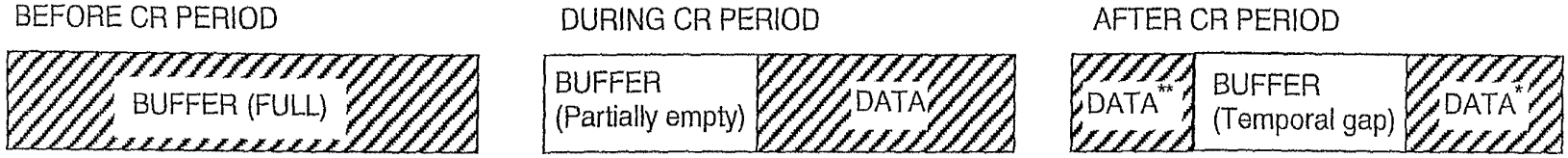

BEFORE CR PERIOD
BUFFER (FULL)
DURING CR PERIOD
BUFFER
(Partially empty)
DATA
AFTER CR PERIOD
DATA**
BUFFER
(Temporal gap)
DATA*

FIG. 4

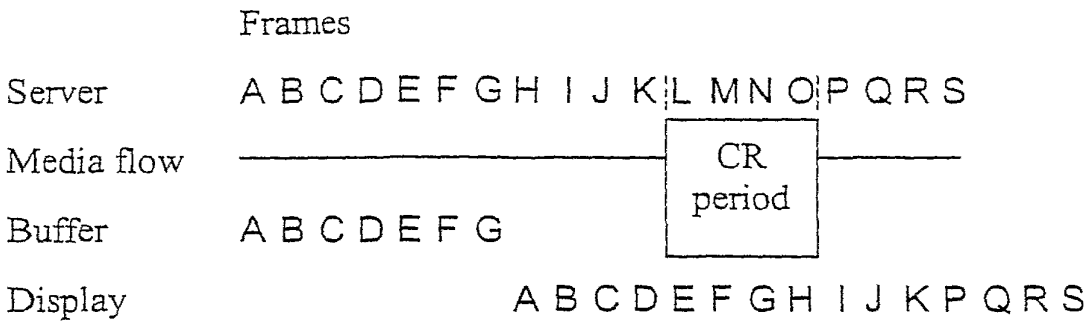

Frames
Server
A B C D E F G H I J K L M N O P Q R S
Media flow
CR period
Buffer
A B C D E F G
Display
A B C D E F G H I J K P Q R S

FIG. 5

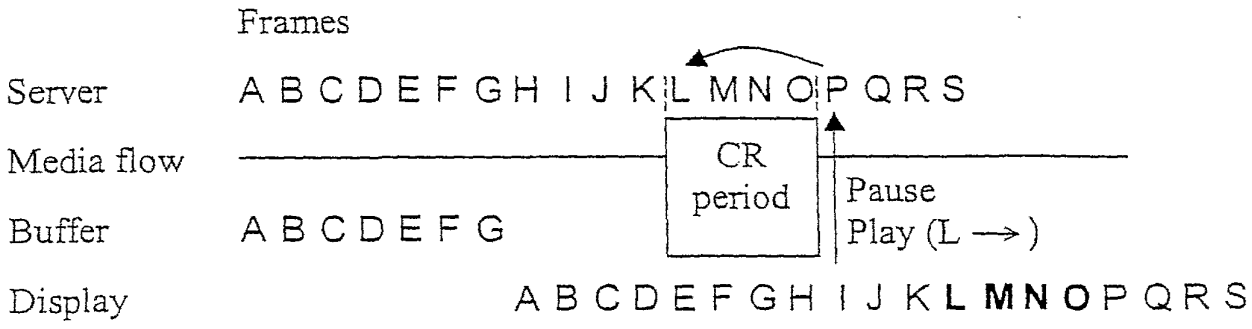

Frames
Server
A B C D E F G H I J K L M N O P Q R S
Media flow
CR period
Pause
Play (L →)
Buffer
A B C D E F G
Display
A B C D E F G H I J K L M N O P Q R S

FIG. 6

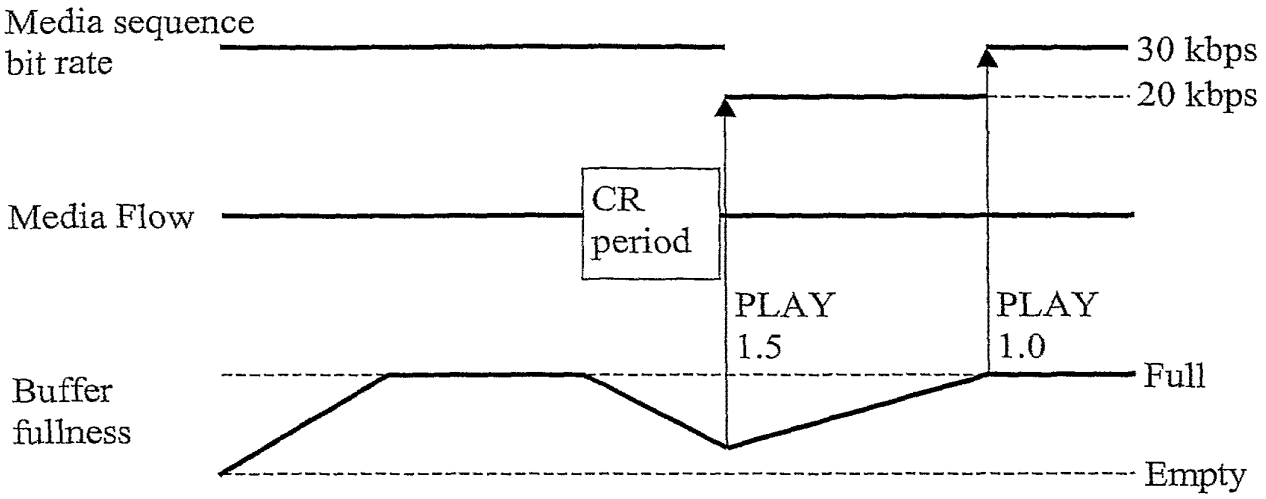

Media sequence bit rate
30 kbps
20 kbps
Media Flow
CR period
PLAY 1.5
PLAY 1.0
Buffer fullness
Full
Empty

FIG. 7

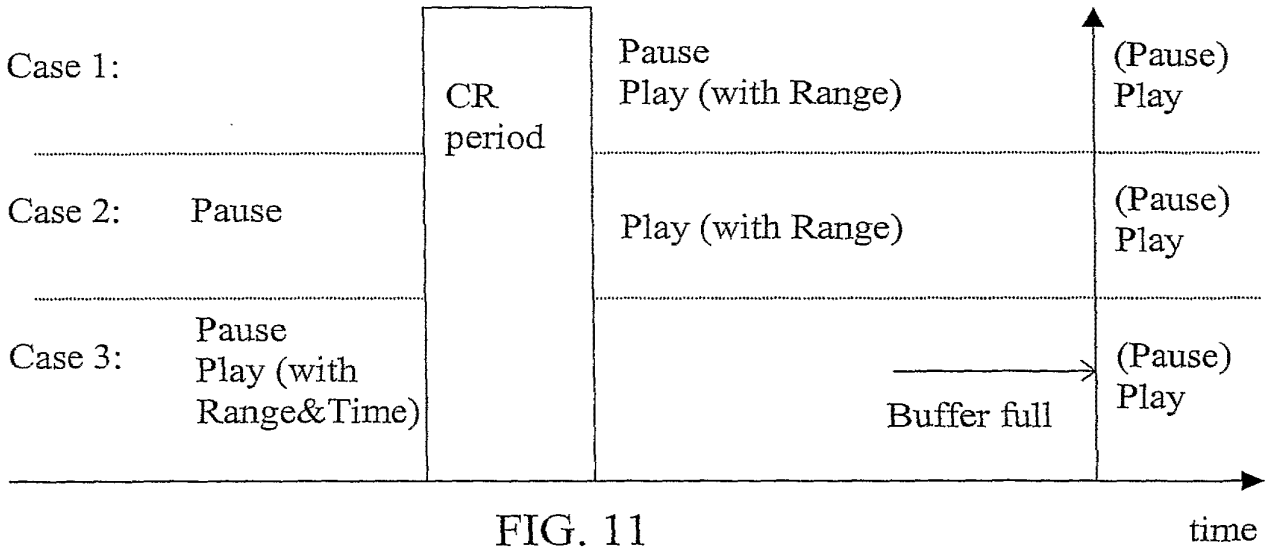

Case 1:
CR period
Pause
Play (with Range)
(Pause) Play
Case 2:
Pause
Play (with Range)
(Pause) Play
Case 3:
Pause
Play (with Range&Time)
Buffer full
(Pause) Play
time

FIG. 11

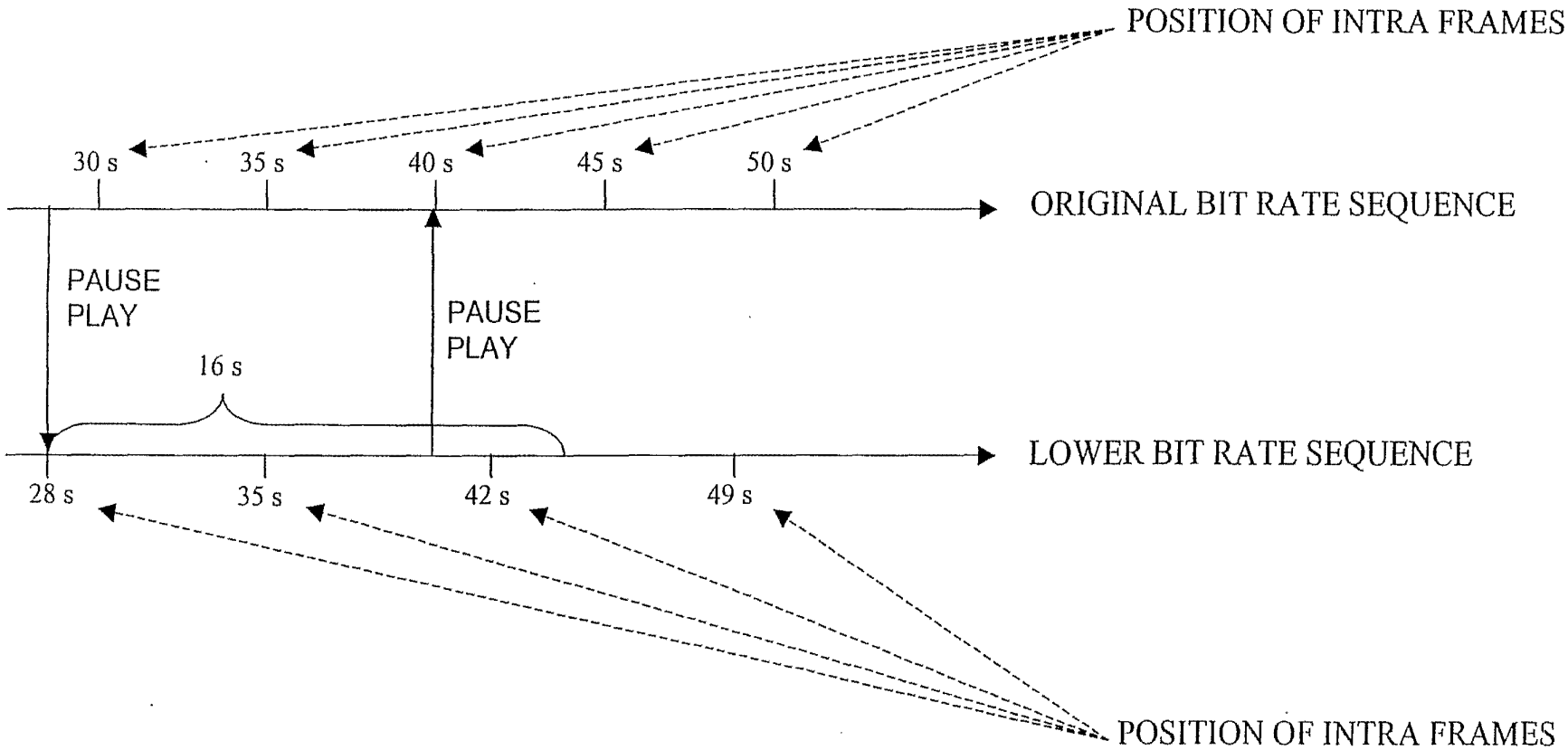

POSITION OF INTRA FRAMES
30 s
35 s
40 s
45 s
50 s
ORIGINAL BIT RATE SEQUENCE
PAUSE
PLAY
PAUSE
PLAY
16 s
LOWER BIT RATE SEQUENCE
28 s
35 s
42 s
49 s
POSITION OF INTRA FRAMES

FIG. 8

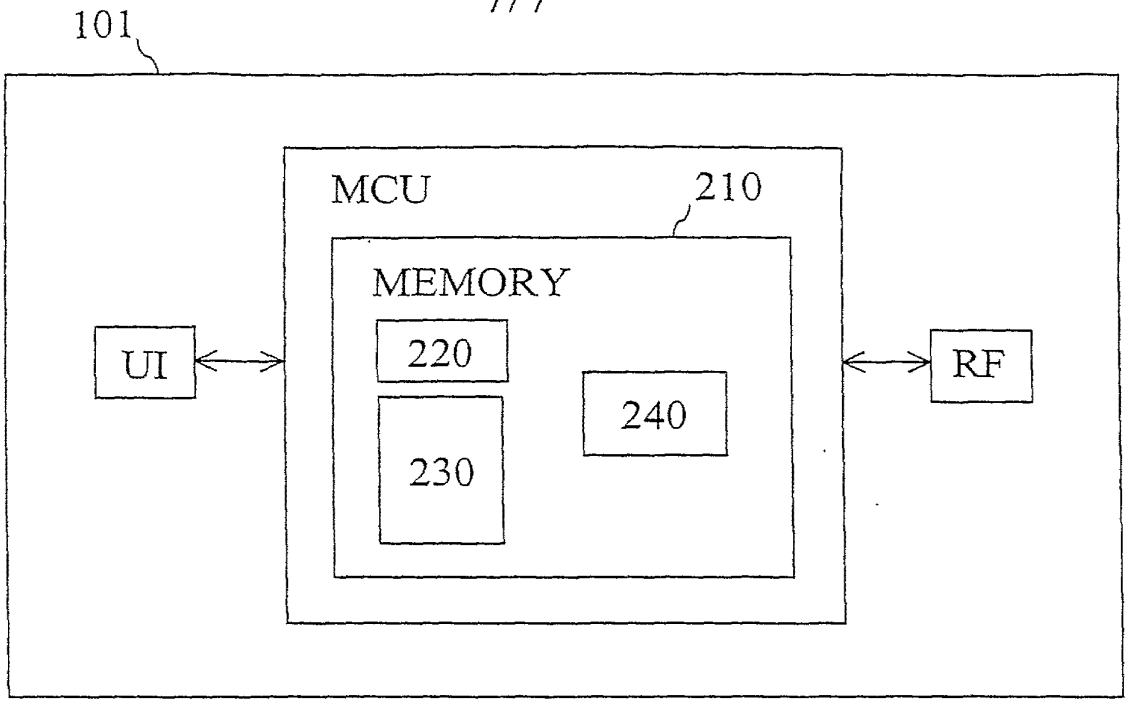

101
MCU
210
MEMORY
220
230
240
UI
RF

FIG. 9

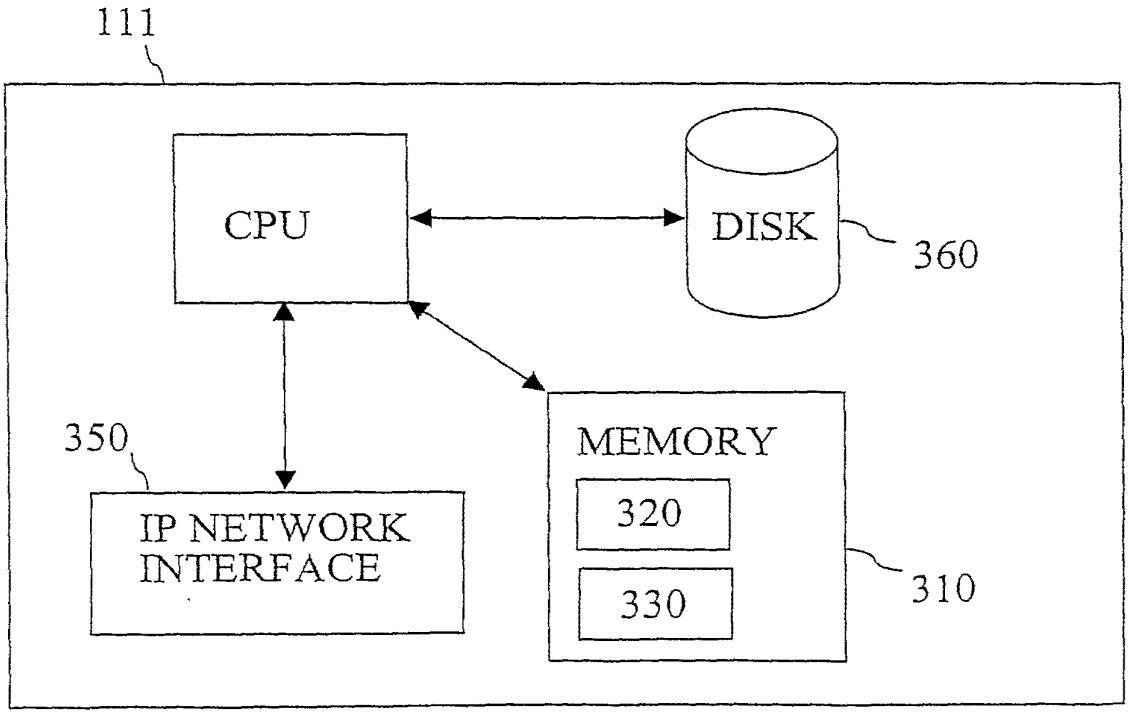

111
CPU
DISK
360
350
IP NETWORK
INTERFACE
MEMORY
320
330
310

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 03/00752

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04L 9/06
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04L, H04Q, H04B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,A	US 2003009576 A1 (APOSTOLOPOULOS, J.G. ET AL), 9 January 2003 (09.01.03), paragraphs [0009] - [0011]; figures 8-11	1-25
A	3GPP TS 26.233 V5.0.0: 3rd Generation Partnership project; Technical Specification Group Services and System Aspects; Transparent end-to-end packet switched streaming service (PSS); General description (Release 5), March 2002, pages 1-12, see section 4, "Usage scenarios" and section 5, "General service architecture".	1-25

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 December 2003	16 -12- 2003
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Fredrik Blomqvist /OGU Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 03/00752

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 0130090 A2 (TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)), 26 April 2001 (26.04.01), page 2, line 29 - page 4, line 19 -- --------	1-25

Form PCT/ISA/210 (continuation of second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

Information on patent family members

06/09/03

International application No.
PCT/FI 03/00752

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003009576 A1	09/01/03	WO 03005761 A	16/01/03
WO 0130090 A2	26/04/01	AU 1065601 A	30/04/01
		EP 1224829 A	24/07/02

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	11116783
Filing Date	2005-04-28
First Named Inventor	R. Drew Major
Art Unit	2444
Examiner Name	Donabed, Ninos J.
Attorney Docket Number	066.0141 (P2004-04-01.1)

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	7523181	B2	2009-04-21	Swildens et al.	
	2	6976090	B2	2005-12-13	Ben-Shaul et al.	
	3	7240100	B1	2007-07-03	Wein et al.	
	4	6850965	B2	2005-02-01	Allen	
	5	7334044	B1	2008-02-19	Allen	
	6	7760801	B2	2010-07-20	Ghanbari et al.	
	7	7099954	B2	2006-08-29	Li et al.	
	8	7054911	B1	2006-05-30	Lango et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	9	7349358	B2	2008-03-25	Hennessey et al.	
	10	7797439	B2	2010-09-14	Cherkasova et al.	
	11	7477688	B1	2009-01-13	Zhang et al.	
	12	7054774	B2	2006-05-30	Batterberry et al.	
	13	6760772	B2	2004-07-06	Zou et al.	
	14	5168356		1992-12-01	Acampora et al.	
	15	7546355	B2	2009-06-09	Kalnitsky	
	16	7174385	B2	2007-02-06	Li	
	17	7376747	B2	2008-05-20	Hartop	
	18	7719985	B2	2010-05-18	Lee et al.	
	19	8683066	B2	2014-03-25	Hurst et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	20	5768527		1998-06-16	Zhu et al.	
	21	7274740	B2	2007-09-25	van Beek et al.	
	22	6185736	B1	2001-02-06	Ueno	
	23	6845107	B1	2005-01-18	Kitazawa et al.	
	24	6874015	B2	2005-03-29	Kaminsky et al.	
	25	7093001	B2	2006-08-15	Yang et al.	
	26	7599307	B2	2009-10-06	Seckni et al.	
	27	7788395	B2	2010-08-31	Bowra et al.	
	28	7536469	B2	2009-05-19	Chou et al.	
	29	7412531	B2	2008-08-12	Lango et al.	
	30	7295520	B2	2007-11-13	Lee et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	31	6795863	B1	2004-09-21	Doty, Jr.	
	32	7609652	B2	2009-10-27	Kellerer et al.	
	33	7325073	B2	2008-01-29	Shao et al.	
	34	6374289	B2	2002-04-16	Delaney et al.	
	35	7194549	B1	2007-03-20	Lee et al.	
	36	7054365	B2	2006-05-30	Kim et al.	
	37	6122660		2000-09-19	Baransky et al.	
	38	7391717	B2	2008-06-24	Klemets et al.	
	39	7260640	B1	2007-08-21	Kramer et al.	
	40	7310678	B2	2007-12-18	Gunaseelan et al.	
	41	7593333	B2	2009-09-22	Li et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	42	6618752	B1	2003-09-09	Moore et al.	
	43	6721723	B1	2004-04-13	Gibson et al.	
	44	6859839	B1	2005-02-22	Zahorjan et al.	
	45	7779135	B2	2010-08-17	Hudson et al.	
	46	7369610		2008-05-06	Xu et al.	
	47	7330908		2008-02-12	Jungck	
	48	7577750		2009-08-18	Shen et al.	
	49	7328243		2008-02-05	Yeager et al.	
	50	7116894		2006-10-03	Chatterton	
	51	7653735		2010-01-26	Mandato et al.	
	52	8135852	B2	2012-03-13	Nilsson et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	53	5841432		1998-11-24	Carmel et al.	
	54	5404446		1995-04-04	Bowater et al.	
	55	6449719	B1	2002-09-10	Baker	
	56	7817985	B2	2010-10-19	Moon	
	57	8209429	B2	2012-06-26	Jacobs et al.	
	58	7733830	B2	2010-06-08	Curcio et al.	
	59	7660906	B1	2010-02-09	Armour	
	60	8036265	B1	2011-10-11	Reynolds et al.	
	61	5267334		1993-11-30	Normille et al.	
	62	8370514	B2	2013-02-05	Hurst et al.	
	63	8612624	B2	2013-12-17	Frueck et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	64	6654790	B2	2003-11-25	Ogle et al.	
	65	6675199	B1	2004-01-06	Mohammed et al.	
	66	6697072	B2	2004-02-24	Russell et al.	
	67	6792449	B2	2004-09-14	Colville et al.	
	68	6801947	B1	2004-10-05	Li	
	69	6885471	B1	2005-04-26	Minowa et al.	
	70	7031700	B1	2006-04-18	Weaver et al.	
	71	7046805	B2	2006-05-16	Fitzhardinge et al.	
	72	7111044	B2	2006-09-19	Lee	
	73	7124164	B1	2006-10-17	Chemtob	
	74	7176957	B2	2007-02-13	Ivashin et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	75	7177642	B2	2007-02-13	Sanchez Herrero et al.	
	76	7190670	B2	2007-03-13	Varsa et al.	
	77	7313236	B2	2007-12-25	Amini et al.	
	78	7529541	B2	2009-05-05	Cho et al.	
	79	6552227	B2	2003-04-22	Mendelovici et al.	
	80	7555464	B2	2009-06-30	Candelore	
	81	7558472	B2	2009-07-07	Locket et al.	
	82	7631039	B2	2009-12-08	Eisenberg	
	83	7761609	B1	2010-07-20	Srinivasan et al.	
	84	7873040	B2	2011-01-18	Karlsgodt	
	85	7818444	B2	2010-10-19	Brueck et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	86	8402156	B2	2013-03-19	Brueck et al	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20040083283	A1	2004-04-29	Sundaram et al.	
	2	20080219151	A1	2008-09-11	Ma et al.	
	3	20020152318	A1	2002-10-17	Menon et al.	
	4	20060080718	A1	2006-04-13	Gray et al.	
	5	20050120107	A1	2005-06-02	Kagan et al.	
	6	20040103444	A1	2004-05-27	Weinberg et al.	
	7	20050188051	A1	2005-08-25	Sneh	
	8	20030154239	A1	2003-08-14	Davis et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	9	20080133766	A1	2008-06-05	Luo	
	10	20070079325	A1	2007-04-05	de Heer	
	11	20030014684	A1	2003-01-16	Kashyap	
	12	20060206246	A1	2006-09-14	Walker	
	13	20050076136	A1	2005-04-07	Cho et al.	
	14	20030018966	A1	2003-01-23	Cook et al.	
	15	20040030599	A1	2004-02-12	Sie et al.	
	16	20050262257	A1	2005-11-24	Major et al.	
	17	20030140159	A1	2003-07-24	Campbell et al.	
	18	20010047423	A1	2001-11-29	Shao et al.	
	19	20030021166	A1	2003-01-30	Soloff	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	20	20020176418	A1	2002-11-28	Hunt et al.	
	21	20020169926	A1	2002-11-14	Pinckney, III et al.	
	22	20020178330	A1	2002-11-28	Schlowsky-Fischer et al.	
	23	20050084166	A1	2005-04-21	Bonch et al.	
	24	20020091840	A1	2002-07-11	Pulier et al.	
	25	20040117427	A1	2004-06-17	Allen et al.	
	26	20040030547	A1	2004-02-12	Leaning et al.	
	27	20060165166	A1	2006-07-27	Chou et al.	
	28	20020097750	A1	2002-07-25	Gunaseelan et al.	
	29	20020152317	A1	2002-10-17	Wang et al.	
	30	20080037527	A1	2008-02-14	Chan et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	31	20040168052	A1	2004-08-26	Clisham et al.	
	32	20050066063	A1	2005-03-24	Grigorovitch et al.	
	33	20050204046	A1	2005-09-15	Watanabe	
	34	20060059223	A1	2006-03-16	Klemets et al.	
	35	20060168295	A1	2006-07-27	Batterberry et al.	
	36	20080281803	A1	2008-11-13	Gentric	
	37	20050185578	A1	2005-08-25	Padmanabhan et al.	
	38	20070280255	A1	2007-12-06	Tsang et al.	
	39	20100098103	A1	2010-04-22	Xiong et al.	
	40	20090210549	A1	2009-08-20	Hudson et al.	
	41	20090055547	A1	2009-02-26	Hudson et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	42	20080205291	A1	2008-08-28	Li et al.	
	43	20070094405	A1	2007-04-26	Zhang	
	44	20060184688	A1	2006-08-17	Ganguly et al.	
	45	20080263180	A1	2008-10-23	Hurst et al.	
	46	20090055471	A1	2009-02-26	Kozat et al.	
	47	20040260701	A1	2004-12-23	Lehikoinen et al.	
	48	20060168290	A1	2006-07-27	Doron	
	49	20070030833	A1	2007-02-08	Pirzada et al.	
	50	20070067480	A1	2007-03-22	Beek et al.	
	51	20060133809	A1	2006-06-22	Chow et al.	
	52	20040049780	A1	2004-03-11	Gee	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	53	20030065803	A1	2003-04-03	Heuvelman	
	54	20020161911	A1	2002-10-31	Pinckney, III et al.	
	55	20060236219	A1	2006-10-19	Grigorovitch et al.	
	56	20080046939	A1	2008-02-21	Lu et al.	
	57	20060130118	A1	2006-06-15	Damm	
	58	20070204310	A1	2007-08-30	Hua et al.	
	59	20020176418	A1	2002-11-28	Hunt et al.	
	60	20050223087	A1	2005-10-06	Van Der Stok	
	61	20040010613	A1	2004-01-15	Apostolopoulos et al.	
	62	20030204602	A1	2003-10-30	Hudson et al.	
	63	20010013128	A1	2001-08-09	Hagai et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	64	20050055425		2005-03-10	Lango et al.	
	65	20050015509		2005-01-20	Sitaraman, Ramesh K.	
	66	20020131496		2002-09-19	Vasudevan et al.	
	67	20020161898		2002-10-31	Hartop et al.	
	68	20040220926		2004-11-04	Lamkin et al.	
	69	20100158101	A1	2010-06-24	Wu et al.	
	70	20090132721	A1	2009-05-21	Soroushian et al.	
	71	20090132599	A1	2009-05-21	Soroushian et al.	
	72	20090055417	A1	2009-02-26	Hannuksela	
	73	20090049186	A1	2009-02-19	Agnihotri et al.	
	74	20090006538	A1	2009-01-01	Risney, JR. et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	75	20060277564	A1	2006-12-07	Jarman	
	76	20040153458	A1	2004-08-05	Noble et al.	
	77	20020087634	A1	2002-07-04	Ogle et al.	
	78	20020118809	A1	2002-08-29	Eisenberg	
	79	20020122491	A1	2002-09-05	Karczewicz et al.	
	80	20020133547	A1	2002-09-19	Lin	
	81	20020136406	A1	2002-09-26	Fitzhardinge et al.	
	82	20020138619	A1	2002-09-26	Ramaley et al.	
	83	20020146102	A1	2002-10-10	Lang	
	84	20020178138	A1	2002-11-28	Ender et al.	
	85	20020184391	A1	2002-12-05	Phillips	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	86	20020194608	A1	2002-12-19	Goldhor	
	87	20030007464	A1	2003-01-09	Balani	
	88	20030037103	A1	2003-02-20	Salmi et al.	
	89	20030078972	A1	2003-04-24	Tapissier et al.	
	90	20030220972	A1	2003-11-27	Montet et al.	
	91	20040098748	A1	2004-05-20	Bo et al.	
	92	20050009520	A1	2005-01-13	Herrero et al.	
	93	20050024487	A1	2005-02-03	Chen	
	94	20050050152	A1	2005-03-03	Penner et al.	
	95	20050066063	A1	2005-03-24	Grigorovitch et al.	
	96	20050204385	A1	2005-09-15	Sull et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	97	20050254508	A1	2005-11-17	Aksu et al.	
	98	20060047779	A1	2006-03-02	Deshpande	
	99	20060218264	A1	2006-09-28	Ogawa et al.	
	100	20060242315	A1	2006-10-26	Nichols	
	101	20060270404	A1	2006-11-30	Tuohino et al.	
	102	20060282540	A1	2006-12-14	Tanimoto	
	103	20060288099	A1	2006-12-21	Jefferson et al.	
	104	20070037599	A1	2007-02-15	Tillet et al.	
	105	20070078768	A1	2007-04-05	Dawson	
	106	20080060029	A1	2008-03-06	Park et al.	
	107	20080091838	A1	2008-04-17	Miceli	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	108	20140207966	A1	2014-07-24	Hurst et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1	2006010113	WO	A2	2006-01-26	Network Foundation Technologies, LLC		☐
	2	200192752	JP		2001-04-06	Hitachi Information Systems Ltd.		☒
	3	2003009581	WO	A1	2003-01-30	British Telecommunications Public Limited Company		☐
	4	2004036824	WO	A1	2004-04-29	Nokia corporation		☐
	5	2004054930	JP		2004-02-19	Microsoft Corporation		☒
	6	2001067264	WO	A1	2001-09-13	Streamcenter, Inc.		☐
	7	2003003760	WO	A2	2003-01-09	Burst.com		☐
	8	2003027876	WO	A1	2003-04-03	Reynolds et al.		☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	9	2004025405	WO	A2	2004-03-25	Thomson Licensing S.A.		☐
	10	1298931	EP	A2	2003-02-04	Oplayo		☐
	11	1202487	EP	A2	2001-10-31	Kabushiki Kaisha Toshiba		☐
	12	1395014	EP	A1	2002-08-27	Matsushita Electric Industrial Co.		☐
	13	2367219	GB	A	2000-09-20	Vintage Global		☐
	14	2011004225	JP	A	2011-01-06	NEC Infronta Corp.		☒
	15	2000201343	JP	A	2000-07-18	Toshiba Corp.		☒

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	USPTO "International Search Report" mailed December 12, 2008; International Appln. No. PCT/US2008/061035, filed April 21, 2008.	☐
	2	AUSTRALIAN GOVERNMENT "Examiner's First Report" dated October 17, 2011; Australian Patent Appln. No. 2011213730.	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	3	KOREAN INTELLECTUAL PROPERTY OFFICE "Official Notice of Preliminary Rejection" issued July 28, 2011; Korean Patent Appln. No. 10-2006-7025274.	☒
	4	JAPAN PATENT OFFICE "Notice of Rejection Ground" mailed April 26, 2011; Japanese Patent Appln. No. 2007-511070.	☒
	5	FUJISAWA, HIROSHI ET AL. "Implementation of Efficient Access Mechanism for Multiple Mirror-Servers" IPSJ SIG Technical Report, vol. 2004, no. 9 (2004-DPS-116), January 30, 2004, Information Processing Society of Japan, pp. 37-42	☒
	6	LIU, JIANGCHUAN ET AL. “Opportunities and Challenged of Peer-to-Peer Internet Video Broadcast,” School of Computing Science, Simon Fraser University, British Columbia, Canada	☐
	7	USPTO INTERNATIONAL SEARCHING AUTHORITY “International Search Report and Written Opinion,” mailed November 5, 2008; International Appln. No. PCT/US2008/009281, filed August 1, 2008.	☐
	8	ZHANG, XINYAN ET AL. "CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming" IEEE 2005.	☐
	9	GUO, YANG "DirectStream: A Directory-Based Peer-To-Peer Video Streaming Service" LexisNexis, Elsevier B.V. 2007.	☐
	10	LIU, JIANGCHUAN ET AL. "Adaptive Video Multicast Over the Internet" IEEE Computer Society, 2003.	☐
	11	REJAIE, REZA ET AL. "Architectural Considerations for Playback of Quality Adaptive Video Over the Internet" University of Southern California, Information Sciences Institute, 1998.)	☐
	12	ROY, SUMIT ET AL. "A System Architecture for Managing Mobile Streaming Media Services" Streaming Media Systems Group, Hewlett-Packard Laboratories, 2003.	☐
	13	XU, DONGYAN ET AL. "On Peer-to-Peer Media Streaming" Department of Computer Sciences, Purdue University, 2002.	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	14	KOZAMERNIK, FRANC "Media Streaming Over the Internet - An Over of Delivery Technologies" EBU Technical Review, October 2002.	☐
	15	LIENHART, RAINER ET AL. "Challenges in Distributed Video Management and Delivery" Intel Corporation, EECS Dept., UC Berkeley, 2000-2002.	☐
	16	JAPAN PATENT OFFICE "Final Office Action" mailed February 28, 2012 in Patent Application No. 2007-511070 filed on October 26, 2006.	☒
	17	JAPAN PATENT OFFICE "Interrogation" mailed November 06, 2012 in Patent Application No. 2007-511070 filed on October 26, 2006.	☒
	18	CANADIAN INTELLECTUAL PROPERTY OFFICE "Office Action" mailed September 09, 2013 in Patent Application No. 2,564,861 filed on October 30, 2006.	☐
	19	USPTO "Office Action" mailed September 13, 2013 in Patent Application No. 13/757,571 filed on February 01, 2013.	☐
	20	USPTO "Notice of Allowance" mailed June 24, 2014 in Patent Application No. 13/757.571, filed on February 01, 2013.	☐
	21	EUROPEAN PATENT OFFICE "Extended Search Report" dated July 10, 2014 in Patent Application No. 12154559.4 filed on September 20, 2002.	☐
	22	NGUYEN, Thinh, "Multiple Sender Distributed Video Streaming" in IEEE Transactions on Multimedia, Volume 6, No. 2, Published April 2, 2004.	☐
	23	WEBLIO, The Meaning of Performance Factor --- English-Japanese Weblio Dictionary, [online], February 24, 2012; retrieved from the internet - URL:http://ejje, weblio.jp/content/performance+factor	☐
	24	MASATO TSURU ET AL., Recent Evolution of the Internet Measurement and Inference Techniques, IEICE Technical Report, vol. 103, no. 123 (IN2003-16 to 23), IEICE, June 12, 2003, pp. 37 to 42, ISSN: 0913-05685.	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11116783
	Filing Date	2005-04-28
	First Named Inventor	R. Drew Major
	Art Unit	2444
	Examiner Name	Donabed, Ninos J.
	Attorney Docket Number	066.0141 (P2004-04-01.1)

	25	TAKESHI YOSHIMURA ET AL., Mobile Streaming Media CDN Enabled by Dynamic SMIL, WWW2002, May 7-11, 2002; retrieved from the Internet at http://www2002.org/CDROM/refereed/515/	☐
	26	CANADIAN INTELLECTUAL PROPERTY OFFICE, Office Action, mailed October 15, 2012 for Patent Application No. 2,564,861.	☐
	27	CLEMENT, B., Move Networks Closes $11.3 Million on First Round VC Funding, Page One PR, Move Networks, Inc. Press Releases, February 7, 2007, http://www.move.tv/press/press20070201.html.	☐
	28	MOVE NETWORKS, INC., The Next Generation Video Publishing System, April 11, 2007; http://www.movenetworks.com/wp-content/uploads/move-networks-publishing-system.pdf.	☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	11116783
Filing Date	2005-04-28
First Named Inventor	R. Drew Major
Art Unit	2444
Examiner Name	Donabed, Ninos J.
Attorney Docket Number	066.0141 (P2004-04-01.1)

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☒ See attached certification statement.

☐ Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☐ None

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/BRETT A. CARLSON/	Date (YYYY-MM-DD)	2014-07-29
Name/Print	Brett A. Carlson	Registration Number	39928

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 January 2006 (26.01.2006)

PCT

(10) International Publication Number
WO 2006/010113 A2

(51) International Patent Classification:
G06F 15/173 (2006.01)

(21) International Application Number:
PCT/US2005/024517

(22) International Filing Date: 11 July 2005 (11.07.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/586,952 9 July 2004 (09.07.2004) US

(71) Applicant *(for all designated States except US)*: **NETWORK FOUNDATION TECHNOLOGIES, LLC** [US/US]; 2215 Cooktown Road, Ruston, LA 71270 (US).

(72) Inventors: **O'NEAL, Mike**; 2215 Cooktown Road, Ruston, LA 71270 (US). **TALTON, John P.**; 2215 Cooktown Road, Ruston, LA 71270 (US).

(74) Agents: **LANDA, Adam B.** et al.; Greenberg Traurig, MetLife Building, 200 Park Avenue, New York, NY 10166 (US).

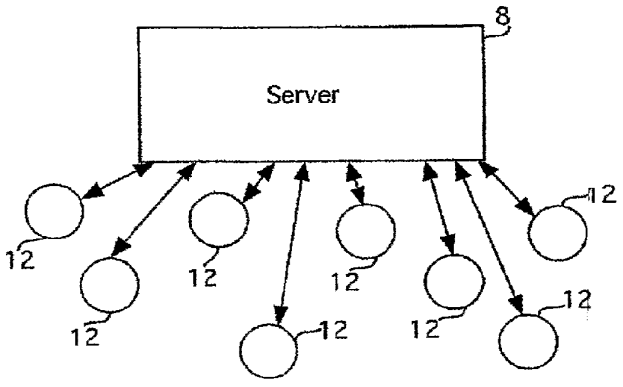
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEMS FOR DISTRIBUTING DATA OVER A COMPUTER NETWORK AND METHODS FOR ARRANGING NODES FOR DISTRIBUTION OF DATA OVER A COMPUTER NETWORK



PRIOR ART

(57) Abstract: Various embodiments of the present invention relate to a system for distributing data (e.g., content data) over a computer network and a method of arranging receiver nodes in a computer network such that the capacity of a server is effectively increased (e.g., the capacity of a server may be effectively multiplied many times over; the capacity of the server may be effectively increased exponentially). In one embodiment the present invention may take advantage of the excess capacity many receiver nodes possess, and may use such receiver nodes as repeaters. The distribution system may include node(s) having database(s) which indicate ancestor(s) and/or descendant(s) of the node so that reconfiguration of the distribution network may be accomplished without burdening the system's primary server. An embodiment of the present invention may include a process for configuring a computer information distribution network having a primary server node and user nodes docked in a cascaded relationship, and reconfiguring the network in the event that a user node departs from the network. In one example (which example is intended to be illustrative and not restrictive), the process may include the steps of providing a new user node (or connection requesting user node) with a connection address list of nodes within the network, having the new user node (or connection requesting user node) go to (or attempt to go to) the node at the top of the connection address list, determine whether that node is still part of the distribution network, and connect thereto if it is, and if it is not, to go to (or attempt to go to) the next node on the connection address list. In another example (which example is intended to be illustrative and not restrictive), when a user node departs from the distribution network, a propagation signal may be transmitted to the nodes below it in the network, causing them to move up in the network in a predetermined order. In another example (which example is intended to be illustrative and not restrictive), the present invention may provide a decentralized approach which provides, to each new user node (or connection requesting user node) a path back to the root server.

WO 2006/010113 A2

SYSTEMS FOR DISTRIBUTING DATA OVER A COMPUTER NETWORK AND METHODS FOR ARRANGING NODES FOR DISTRIBUTION OF DATA OVER A COMPUTER NETWORK

FIELD OF THE INVENTION

This patent application is related to and claims the benefit of the filing date of U.S. Provisional Patent Application Serial No. 60/586,952 for "Systems for Distributing Data Over A Computer Network And Methods For Arranging Nodes For Distribution Of Data Over A Computer Network", filed July 9, 2004.

Various embodiments of the present invention relate to systems for distributing data (e.g., content data) over a computer network and methods of arranging nodes for distribution of data (e.g., content data) over a computer network. In one example (which example is intended to be illustrative and not restrictive), the systems and methods of the present invention may be applied to the distribution of streaming audiovisual data over the Internet.

For the purposes of the present application the term "node" (e.g., as used in the phrase a first node communicates with a second node) is intended to refer to a computer system (or sometimes just "computer") configured to send and/or receive data over a computer network (e.g., a computer network including the Internet, a local-area-network, a wide-area-network, a wireless network). Since a "node" in a computer network would not exist but for a computer system being available on such node, the terms "node" and "computer system" (or sometimes just "computer") may be used interchangeably (i.e., the term "node" should be understood to include an available computer system (or sometime just "computer").

Further, for the purposes of the present application the term "upstream" (e.g., as used in the phrase a first computer system sends data via an upstream connection to a second computer system) is intended to refer to the communication path between a first computer system and the Internet when the first computer system is sending data to a second computer system via the Internet.

Further still, for the purposes of the present application the term "downstream" (e.g., as used in the phrase a first computer system receives data via a downstream connection from a second computer system) is intended to refer to the communication path between a first computer system and the Internet when the first computer system is receiving data from a second computer system via the Internet.

Further still, for the purposes of the present application the term "uptree" (e.g., as

used in the phrase a first node sends data via an uptree connection to a second node) is intended to refer to the network topology communication path between a first node and a second node when a first node is sending data to a second node which is higher up on the network topology tree (that is, closer to the root server).

Further still, for the purposes of the present application the term "downtree" (e.g., as used in the phrase a first node sends data via a downtree connection to a second node) is intended to refer to the network topology communication path between a first node and a second node when a first node is sending data to a second node which is lower down on the network topology tree (that is, closer to the leaves).

Further still, for the purposes of the present application the term "docked" (e.g., as used in the phrase a child node docks with a parent node) is intended to refer to forming a connection between nodes via which data may flow in at least one direction (e.g., at least uptree or downtree).

Further still, for the purposes of the present application the term "apparent available capacity" (e.g., as used in the phrase apparent available capacity to transmit content data) is intended to refer to a nominal capacity (e.g., a node's presumed capacity based upon a design specification but not necessarily actually tested in the network).

BACKGROUND OF THE INVENTION

In a computer network such as the Internet, each node in the network has an address. A computer system resident at a particular address may have sufficient bandwidth or capacity to receive data from, and to transmit data to, many other computer systems at other addresses. An example of such a computer system is a server, many commercial versions of which can simultaneously exchange data with thousands of other computer systems.

A computer system at another location may have only sufficient bandwidth to effectively exchange data with only one other computer system. An example of such a system is an end user's personal computer connected to the Internet by a very low speed dialup modem. However, even typical personal computers connected to the Internet by higher speed dialup modems may have sufficient bandwidth that they may exchange data essentially simultaneously in an effective manner with several other computer systems. Moreover, an end user's personal computer system may have even greater bandwidth when connected to the Internet by ISDN lines, DSL (e.g., ADSL) lines, cable modems, T1 lines or even higher capacity links. As discussed more fully below, various embodiments of the present invention

may take advantage of the availability of such higher capacity end user systems (e.g., those computer systems capable of essentially simultaneously exchanging data with multiple computer systems).

In a typical situation, as shown in FIG. 1, a content provider distributes its data by making the data available on a server node 8 simultaneously to a plurality of users at user nodes 12 (of note, the terms "server", "root server" and "primary server" may be used interchangeably throughout the present application to refer to the same device (i.e., the highest level parent node in a given network). The double-headed arrows show the two-way communication between each end user's computer system and the server. Essentially the content provider's server transmits a separate stream of signals to each receiver node. To accommodate additional users, the content provider would typically either add equipment to increase capacity or it would engage a mirror site to accomplish essentially the same result as adding equipment. The capacities of the end user computers is of virtually no consequence in such a system.

Another system for distributing data is the Napster™ music file exchange system provided by Napster, Inc. of Redwood City, Calif. A schematic of the Napster™ music file exchange system (as its operation is presently understood) is illustrated in FIG. 2.

More particularly, it is believed that in the Napster™ system a copy of the music data is not kept on the server. The server 9 instead maintains a database relating to the various music files on the computers of users who are logged onto the server 9. When a first user 12a sees that a desired music file is available from a second logged on user 12b, the first user causes his computer to query the server 9 for the second user's node address and a connection is made between the first and second user's computers through which the first user's computer notifies the second user's computer of the desired file and the second user's computer responds by transmitting a copy of the desired music file directly to the first user's computer. It is further believed that a first user attempting to download a particular file from a second user must start completely over again if the second user cancels its transmission or goes off line during the data transfer.

In another area, engineers have developed what is known as "streaming media." In summary, streaming media is a series of packets (e.g., of compressed data), each packet representing moving images and/or audio.

To help understand streaming media in more detail, it is helpful to review the traditional Internet distribution method. Each node (whether it is a server node or a user node) in a computer network has a unique identification (sometimes referred to as an "IP" address)

associated with it. On the Internet, the unique address may be referred to as a Uniform Resource Locator ("URL"). A user desiring to obtain data from a particular server enters that server's URL into the user's browser program. The browser program causes a connection request signal to be sent over the Internet to the server. If the server has the capacity to accept the connection, the connection is made between the server and the user node (files requested by the user are typically transmitted by the server in full to the user node and the browser program may store the files in buffer memory and display the content on the user's computer system monitor - some files may be more permanently stored in the computer system's memory for later viewing or playing.) The connection with the server is typically terminated once the files have been received at the user node (or the connection may be terminated a short time thereafter). Either way, the connection is usually of a very short time duration.

With streaming media, the contact between the server and user nodes is essentially continuous. When a connection between a server node and user node is made and streaming media is requested, the server sends streaming media packets of data to the user node. A streaming media player installed on the user's computer system (e.g., software, such as RealMedia™ from RealNetworks, Inc. of Seattle, Wash.,) causes the data to be stored in buffer memory. The player decompresses the data and begins playing the moving images and/or audio represented by the streaming media data on the user's computer system. As the data from a packet is played, the buffer containing that packet is emptied and becomes available to receive a new packet of data. As a result, the memory assets of a user's computer are not overly taxed. Continuous action content, such as, for example, the display of recorded motion picture films, videos or television shows may be distributed and played in essentially "real time," and live events, such as, for example, concerts, football games, court trials, and political debates may be transmitted and viewed essentially "live" (with only the brief delays needed for compression of the data being made available on the server, transmission from the server to the user node, and decompression and play on the user's computer system preventing a user from seeing the event at the exact same moment in time as a person actually at the event). And, when the systems are working as designed, the server node and user node may stay connected to each other until all the packets of data representing the content have been transmitted.

SUMMARY OF THE INVENTION

Various embodiments of the present invention relate to a system for distributing data (e.g., content data) over a computer network and a method of arranging receiver nodes in a computer network such that the capacity of a server is effectively increased (e.g., the capacity of a server may be effectively multiplied many times over; the capacity of the server may be effectively increased exponentially). In one embodiment the present invention may take advantage of the excess capacity many receiver nodes possess, and may use such receiver nodes as repeaters. The distribution system may include node(s) having database(s) which indicate ancestor(s) and/or descendant(s) of the node so that reconfiguration of the distribution network may be accomplished without burdening the system's primary server. An embodiment of the present invention may include a process for configuring a computer information distribution network having a primary server node and user nodes docked (i.e., connected) in a cascaded relationship, and reconfiguring the network in the event that a user node departs from the network. In one example (which example is intended to be illustrative and not restrictive), the process may include the steps of providing a new user node (or connection requesting user node) with a connection address list of nodes within the network, having the new user node (or connection requesting user node) go to (or attempt to go to) the node at the top of the connection address list, determine whether that node is still part of the distribution network, and connect thereto if it is, and if it is not, to go to (or attempt to go to) the next node on the connection address list. In another example (which example is intended to be illustrative and not restrictive), when a user node departs from the distribution network, a propagation signal may be transmitted to the nodes below it in the network, causing them to move up in the network in a predetermined order. In another example (which example is intended to be illustrative and not restrictive), the present invention may provide a decentralized approach which provides, to each new user node (or connection requesting user node) a path back to the root server.

Of note, various embodiments of the present invention may be applied to the transmission (e.g., the "appointment" transmission) of audiovisual content such as, for example (which example is intended to be illustrative and not restrictive), live or pre-recorded concerts, football games, court trials, political debates, motion picture films, videos or television shows. Such transmission of audiovisual content may be made with acceptable levels of quality, wherein "acceptable levels of quality" may vary depending upon the end users and the type of transmission. Further, such transmission of audiovisual content may be

carried out using streaming media transmissions intended to reach large audiences, in much the way that television shows transmitted over television cable and broadcast media reach large audiences (in this regard, the present invention may enable a server to transmit streaming media to the large number of users which would be essentially simultaneously logging on to view a particular audiovisual presentation under this television-type "large audience" transmission).

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic drawing of a prior art computer information distribution network;

FIG. 2 is a schematic drawing of another prior art computer information distribution network;

FIG. 3 is a schematic drawing of an embodiment of a computer information distribution network formed pursuant to the present invention;

FIG. 4 is a schematic drawing of another embodiment of a computer information distribution network formed pursuant to the present invention;

FIG. 5 is a schematic drawing of another embodiment of a computer information distribution network formed pursuant to the present invention;

FIG. 6 is a schematic drawing of a particular topology of a computer information distribution network formed pursuant to an embodiment of the present invention;

FIG. 7 is a schematic drawing of a particular topology of the computer information distribution network formed pursuant to an embodiment of the present invention as shown in FIG. 6 (after the occurrence of an event);

FIG. 8 is a schematic drawing of another topology of a computer information distribution network formed pursuant to an embodiment of the present invention;

FIG. 9 is a schematic drawing of the topology of the computer information distribution network formed pursuant to an embodiment of the present invention as shown in FIG. 8 (after the occurrence of an event);

FIG. 10 is a flow diagram of an embodiment of the present invention showing a Server Connection Routine which may be performed when a prospective child node seeks to join the distribution network;

FIG. 11 is a flow diagram of an embodiment of the present invention showing a Prospective Child Node's Request to Server for a Connection Routine;

FIGS. 12A-12E are schematic drawings showing varying topologies of the computer information distribution network formed pursuant to an embodiment of the present invention under several circumstances;

FIG. 13 is a block diagram of an embodiment of the present invention showing the memory blocks into which the software used may partition a user node's memory;

FIG. 14 is a flow diagram of an embodiment of the present invention showing a Prospective Child Node's Connection Routine - the routine which a new user node (or connection requesting user node) may go through in attempting to connect to a distribution chain (or tree) after receiving a connection address list;

FIG. 15 is a flow diagram of an embodiment of the present invention showing a Prospective Child Node's Connection Routine With Return to Server Subroutine;

FIG. 16 is a flow diagram of an embodiment of the present invention illustrating a Prospective Parent Node's Connection Routine;

FIG. 17 is a flow diagram of an embodiment of the present invention illustrating a Server's Connection Instruction Routine;

FIG. 18 is a flow diagram of an embodiment of the present invention illustrating a Fully Occupied Parent's Connection Instruction Routine;

FIG. 19 is a flow diagram of an embodiment of the present invention illustrating a Multi-Node Selection Subroutine;

FIGS. 20A and 20B are together a flow diagram of an embodiment of the present invention illustrating a Universal Connection Address Routine;

FIG. 21 is a schematic drawing of a topology of a computer information distribution network of an embodiment of the present invention before a new node will be added using the Universal Connection Address Routine;

FIG. 22 is a schematic drawing of a topology of the computer information distribution network of an embodiment of the present invention when a new node is added using the Universal Connection Address Routine;

FIG. 23 is a schematic drawing of a topology of a computer information distribution network of an embodiment of the present invention before a reconfiguration event;

FIG. 24 is a schematic drawing of a topology of a computer information distribution network of an embodiment of the present invention shown in FIG. 23 after a reconfiguration event;

FIG. 25 is a schematic drawing of a topology of a computer information distribution network of an embodiment of the present invention, slightly different from the topology shown in FIG. 23, before a reconfiguration event;

FIG. 26 is a schematic drawing of a topology of a computer information distribution network of an embodiment of the present invention shown in FIG. 25 after a reconfiguration event;

FIG. 27 is a flow diagram of an embodiment of the present invention showing a Child Node's Propagation Routine;

FIG. 28 is a schematic drawing of another topology of a computer information distribution network of an embodiment of the present invention before a reconfiguration event;

FIG. 29 is a schematic drawing of a topology of a computer information distribution network of an embodiment of the present invention shown in FIG. 28 after a reconfiguration event;

FIG. 30 is a schematic drawing of another topology of a computer information distribution network of an embodiment of the present invention before a "complaint" regarding communications;

FIG. 31 is a flow diagram of an embodiment of the present invention showing a Grandparent's Complaint Response Routine.

FIGS. 32, 33A-33C, 34A-34C, 35A, 35B and 36 relate to "Registration Process" examples according to the present invention;

FIGS. 37 and 38 relate to a "Firewall Request / Firewall Status" example according to the present invention;

FIGS. 39A, 39B and 40 relate to a "Propagate Path / Down" example according to the present invention;

FIGS. 41A, 41B and 42 relate to a "Propagate NTM / Up" example according to the present invention;

FIGS. 43 and 44 relate to a "Banned List" example according to the present invention;

FIG. 45 relates to a "Node Interlacing" example according to the present invention; and

FIGS. 46A-46C relate to various distribution network topology examples according to the present invention.

Among those benefits and improvements that have been disclosed, other objects and advantages of this invention will become apparent from the following description taken in conjunction with the accompanying figures. The figures constitute a part of this specification and include illustrative embodiments of the present invention and illustrate various objects and features thereof (of note, the same reference numeral may be used to identify identical elements throughout the figures).

DETAILED DESCRIPTION OF THE INVENTION

Detailed embodiments of the present invention are disclosed herein; however, it is to be understood that the disclosed embodiments are merely illustrative of the invention that may be embodied in various forms. In addition, each of the examples given in connection with the various embodiments of the invention are intended to be illustrative, and not restrictive. Further, the figures are not necessarily to scale, some features may be exaggerated to show details of particular components. Therefore, specific structural and functional details disclosed herein are not to be interpreted as limiting, but merely as a representative basis for teaching one skilled in the art to variously employ the present invention.

Referring now to FIG. 3, an illustration of a linear propagation arrangement 10 of a computer network according to an embodiment of the present invention is shown. In this embodiment, the primary server node (or simply, server) 11 provides content data (e.g., streaming media) to user nodes 12 connected directly to it (sometimes referred to as "first level user nodes"). Each first level user node 12 has a second level user node 13 connected to it and each second level user node 13 has a third level user node 14 connected to it. The computer system at each first level user node 12 passes a copy of the content data received from server node 11 to the computer system at the second level user node 13 attached to such first level user node 12. The computer system at each second level user node 13 in turn passes the content data onto the computer system at the fourth level user node 14 attached to it.

As more fully discussed below, under this embodiment the computer systems at the server and user nodes may have distribution software installed in them which enables the nodes to be arranged as shown and for the computer systems to receive and re-transmit data.

The cascadingly connected arrangement of nodes (i.e., first level nodes are connected to the server, second level nodes are connected to first level nodes, third level nodes are connected to second level nodes and so on) shown in FIG. 3 takes advantage of the bandwidth available in certain nodes to essentially simultaneously receive and transmit data.

In a linear propagation arrangement, the effective distribution capacity of a server is in essence multiplied by the number of levels of nodes linked together. In the example of FIG. 3 (which example is intended to be illustrative and not restrictive), the distribution capacity of the server node is increased from 8 user nodes to 24 in just three levels.

Of note, many user nodes may have at least sufficient bandwidth (e.g., upstream bandwidth as well as downstream bandwidth) to receive data from one node and to re-transmit streams of data essentially simultaneously to two or more other nodes. This capacity could be used in another embodiment to set up a computer network with a cascadingly connected exponential propagation arrangement 16 (as shown in FIG. 4). As the name implies, an exponential propagation arrangement effectively increases the distribution capacity of a server exponentially. For example (which example is intended to be illustrative and not restrictive), with just three levels of user nodes, each having the capacity to retransmit two data streams, the distribution capacity of the server in FIG. 4 is increased from 8 user nodes to 56.

In another embodiment a distribution network may also be set up as a cascadingly connected hybrid linear/exponential arrangement 18, such as shown in FIG. 5.

Of note, the effective distribution capacity grows more quickly in a hybrid linear/exponential arrangement with each new level than does the distribution capacity in a linear propagation arrangement, and less quickly than does the distribution capacity in a pure exponential arrangement. However, any of these arrangements allows a server's distribution capacity to be greatly increased (e.g., with little or no additional investment in equipment for the server). Further, any other desired hybrid arrangement may be used.

Referring again to FIGS. 1-5, all of the connections between nodes are illustrated with arrows on each end. This is intended to signify that data flows in both directions between connected nodes. For example (which example is intended to be illustrative and not restrictive), a user node connected to a server may transmit data to the server indicating the identity of the user node, what the user node wants and/or other data, while the server node may transmits data confirming its identity and containing information and other content to the user node. In the remaining drawings such arrows may not be shown for the sake of simplicity. Further, it is noted that the bulk of data which is transmitted may be content (e.g., web pages, music files, streaming media), and such content will be understood to flow downtree from node to node in the drawings (i.e., in the direction from the root server to a first level node, to a second level node, to a third level node, etc.).

As can be understood from the above, an exponential propagation arrangement may create the most distribution capacity. However, while many (and perhaps even most) user nodes may have sufficient bandwidth to re-transmit acceptable quality multiple copies of the data, a number of user nodes may not have sufficient bandwidth to re-transmit more than one copy of the data with acceptable quality, and some user nodes may not have sufficient bandwidth to re-transmit even a single acceptable copy. Under these conditions, a distribution system employing the present invention may be configured, for example, as a hybrid propagation network. In such a system, a personal computer acting as a server node may reach many (e.g., hundreds, thousands or more) user nodes, even if the server node itself has capacity to transmit content data directly to only one other node.

In the following discussion it will be assumed that the system employing the present invention will take advantage of the capability of many user nodes to simultaneously re-transmit up to two copies of data (e.g., content data). However, it should be understood that the invention could take advantage of user nodes capable of simultaneous re-transmission of even higher numbers of copies.

In one embodiment the length of the distribution chains (i.e., the number of levels of user nodes linked through each other to the server) may be kept as small as possible to reduce the probability that the nodes at the ends of the chains will suffer a discontinuity of service. Thus, under this embodiment, in order to maximize the number of user nodes which may be connected to the server while trying to keep the length of the distribution chains as small as possible, the user nodes having the greatest bandwidth may be placed as far up uptree as possible (i.e., be placed as close as possible to the server, where "close" refers to network node level and not necessarily to geographical proximity).

In this regard, if an unreliable user node is placed near or at the end of a chain, few or no other user nodes would be affected by the unreliable user node's leaving the network. In contrast, if an unreliable user node were placed at or near the beginning of a chain (i.e., be far uptree), many other nodes would be affected by the unreliable node's departure. Because various embodiments of the invention may be used for appointment viewing of streaming media, in which an essentially continuous connection to an audiovisual content source may essentially be a necessity, it may be important (e.g., for this steaming media use) that the most reliable user nodes be placed high up in the chain (i.e., high uptree).

A user node may be deemed unreliable for any of a number of reasons. One example (which example is intended to be illustrative and not restrictive), is that the user node is connected to the Internet by lines having intermittent failures. Another example (which

example is intended to be illustrative and not restrictive), is that the user is merely sampling the content available on the network and intentionally disconnects the user node from the distribution network after discerning that he or she has no interest in the content.

Thus, under an embodiment of the present invention the user nodes may be positioned in the most advantageous positions, taking into account the dynamic nature of the distribution network, in which many user nodes may enter and leave the distribution network throughout the server's transmission of a streaming media show. In addition, the invention may help to preserve the viewing experience of users at user nodes positioned even at the end of a distribution chain (the server and/or user nodes may be enabled to perform the operations required to set up and maintain the distribution network by having data distribution software installed therein).

Referring now to a "User Node Arrival" example (which example is intended to be illustrative and not restrictive), it can be seen from FIGS. 3-5 that the distribution chains can be viewed as a plurality of family trees, each family tree being rooted to the server 11 through one of the first level nodes 12 (each node in the distribution network may have distribution software loaded in it which enables the node to perform the functions described below; before any new node may join the distribution network, such new node may also have such software loaded in it).

Of note, in the bulk of the examples discussed below, the network topologies have a branch factor of two (i.e., no user node is assigned more than two child nodes to be connected directly to it). A network topology with a branch factor of two may be referred to as a binary tree topology. It should be understood, of course, that the teachings set forth herein may be extended to network topologies having other branch factors, for example (which example is intended to be illustrative and not restrictive), branch factors of three, four or more.

In any case, FIG. 6 is a schematic drawing of a distribution network topology useful for describing the current "User Node Arrival" example. This FIG. 6 shows a distribution chain or family tree rooted to the server 11 through user node A, a first level user node 12 (the dashed lines represent connections from the server to other first level nodes or from node A to another user node). User node A could be thought of as a child node of the server and as a parent node for other user nodes connected directly to it. User node B, a second level user node 13, could be thought of as A's child. User nodes C and D, third level user nodes 14, may be thought of as B's children and A's grandchildren (and each other's siblings). User nodes E and F, fourth level user nodes 15, may be thought of as C's children (and each other's

siblings). User node G, also a fourth level user node 15, may be thought of as D's child. And user nodes E, F and G may be thought of as B's grandchildren and A's great grandchildren.

In the present example, whenever a new user node (or connection requesting user node) 19, such as node X in FIG. 6, seeks connection to the distribution network, it will first make a temporary connection to the server node (or a connection server, not shown) in order to begin the process for connecting to the distribution system. The server (or connection server) will discern from the user node a bandwidth rating (discussed below) appropriate to that node and, depending upon the available capacity of the server and any existing distribution chains, the server will either assign the new user node to a spot directly connected to the server or will provide the new user node with a connection path through a tree to the server.

FIG. 10 is a flow diagram associated with this example showing a Server's Connection Routine which is performed when a prospective child node seeks to join the distribution network. In step 101 the server performs the Server's Connection Instruction Routine (discussed below), in which the server determines what connection instructions to give to the new user node (or connection requesting user node). The server then goes to step 102 where it determines whether, as a result of the Server's Connection Instruction Routine, the prospective child node is being instructed to dock with the server. If the prospective child node is being instructed to dock with the server, then the server goes to step 103 in which the server would allow the new user node to dock with it, and the server would begin transmitting data (e.g., streaming media) directly to the new user node.

Of note, as referred to above, two different servers could be used - one to perform the server's connection routine and the other to transmit data (e.g., streaming media) - since both servers would be performing server functions, they will hereinafter sometimes be considered a single server for purposes of the description herein.

Of further note, while the prospective child node may be a new user node or a connection requesting user node, to simplify the following discussion such prospective child node may simply be referred to as a "a new user node".

In any case, if the new user node is not being instructed to dock directly with the server, then the server goes to step 104 in which it provides the new user node with an address connection list and disconnects the new user node from the server.

Essentially contemporaneously with the performance by the server of the Server's Connection Routine, the new user node is performing the Prospective Child Node's Request to Server for a Connection Routine. FIG. 11 is a flow diagram associated with this example

illustrating the Prospective Child Node's Request to Server for a Connection Routine. Upon making the temporary connection to the server, the new user node goes to step 111 in which it provides bandwidth rating and performance rating information to the server. It then proceeds to step 112 in which it receives connection instructions from the server. Then the new user node proceeds to step 113 to determine whether it has been instructed to dock directly with the server. If the answer is "yes," then the new user node proceeds to step 114 in which it erases any information which may have been in its ancestor database and, if the distribution software has a Return to Server Subroutine in it, resets the connection attempt counter to zero. The new user node then proceeds to step 115 in which it docks with the server and begins receiving data (e.g., streaming media). If the new user node is not being instructed to dock directly with the server, then the new user node goes to step 116 in which it receives the new connection address list from the server and loads such list into the user node's ancestor database and begins the Prospective Child Node's Connection Routine (discussed below). If the distribution software has a Return to Server Subroutine in it, the connection attempt counter is reset to one in step 116.

In the examples discussed in connection with FIGS. 6-9 and 12A-12C (which examples are intended to be illustrative and not restrictive), the server has determined that the new user node X will not be allowed to connect directly to the server. Also, for the purposes of these examples, all of the user nodes are presumed to be equally capable of simultaneously re-transmitting two copies of the content data and that the tree rooted through node A is the most appropriate tree through which node X should be connected to the server. In one embodiment the server may rely on chain length in determining to which particular user node, already in the distribution network, that node X should connect.

Referring again to the example of FIG. 6, the topology of the distribution network based on the most recent information available to the server at the moment that node X seeks to join the distribution network is shown. Path D-B-A-S (where "S" is the server) represents the shortest available path from an end of a chain back to the server (e.g., from a node level point of view and not necessarily from a geographical proximity point of view), and the server 11 gives that path information, or connection address list, to node X during step 104 of FIG. 10 (and node X receives such list during step 116 of FIG. 11). That is, node X will be given a connection address list with the URLs and/or IP addresses of each of nodes D, B, A and S. The distribution software in node X causes the path information to be stored in the ancestor portion (or ancestor database) 132 of node X's topology database 131 shown in FIG. 13. The ancestor database may include an ancestor list, a list of node X's ancestors' addresses

from node X's parent back to the server node. (FIG. 13 shows an example block diagram (which example is intended to be illustrative and not restrictive) showing the memory blocks into which the software used in connection with the present invention may partition a user node's memory 130.) Node X then attempts to contact node D first, the user node most distant from server 11 in the path. Note that when the server provides the D-B-A-S connection address to node X, the server is giving what it "believes" to be the complete path information going all the way back to the server. That is, subsequent to the most recent generation of the topological data, node D may have departed from the network, as may have one or more of its ancestors, resulting in a reconfiguration (discussed below) of at least a portion of the tree of which D was a part.

Referring now to FIG. 14 is a flow diagram associated with this example showing the Prospective Child Node's Connection Routine, the routine which a new user node, here node X, will go through in attempting to connect to a distribution chain (or tree) after receiving a connection address list (which node X has stored in the ancestor portion of its topology database) from the server or from a prospective parent node or during a reconfiguration event. In step 141, node X attempts to contact the first node on the connection address list. The first node, and only node, on the connection address list could be the server itself. Here node D is the first node on the list. Node X then proceeds to step 142 and determines whether the first node on the connection address list is still on-line and still part of the distribution network (in one example, if no response is received within a predetermined period of time, from the first node on the connection address list, the answer to the query in step 142 will be deemed to be no). If node D is on-line and still part of the distribution network, node X proceeds to step 143 in which node X inquires whether node D has room for node X. This inquiry may need to be made because the distribution network may have gone through a reconfiguration event resulting in node D's not having sufficient capacity to provide a copy of the content data to node X. If node D has room for node X, then node X proceeds to step 144 in which it connects (or docks) with node D and begins receiving content data from it. This is depicted in FIG. 7. Note that node X is now one of several level four nodes 15.

On the other hand, in step 142, if node D is not on-line (e.g., no response is received from node D within a predetermined period of time) or if node D is on-line but is no longer part of the distribution system (e.g., subsequent to the server's obtaining its topology data the user of the system at node D either caused his or her computer system to go off-line or to leave the distribution system, or there was a line failure causing the computer system to go off line), as depicted in FIG. 8, then node X goes to step 145 in which it deletes the first

address from the connection address list in node X's ancestor database (and, in one example, and for a purpose which will become clear when discussing reconfiguration events below, sets its reconfiguration flag buffer 136 to the lowest propagation rating). At this time node B, in the present example, becomes the first node on the connection address list. Then node X goes back to step 141 and repeats the routine described above. Note that because of node D's leaving the distribution network, a reconfiguration event was triggered which resulted in node G changing from a fourth level node 15 to a third level node 14.

In step 143, if the prospective parent node has no room for the new node (e.g., the capacity of the prospective parent node is fully occupied), the new node goes to step 146, in which it receives a new connection address list from the prospective parent. The prospective parent may then perform a Fully Occupied Parent's Connection Instruction Routine, discussed below in connection with FIG. 18, wherein it creates the new connection address list based on topology data obtained from its progeny. That new list may include the path back to the server through the prospective parent just in case, as discussed above, there are user node departures along the path.

Of note, by having a prospective parent node prepare the new connection address list as described in this example, the burden on the server is reduced and is distributed among the user nodes.

Still referring to the example depicted in FIG. 8, when node X gets to step 143, node B will respond that it has no room and node X will proceed to step 146. When node X performs step 146, the new connection address list it receives from node B will be G-B-A-S. Then node X proceeds to step 141 and repeats the routine from that point on. When the routine is performed on the topology shown in FIG. 8, step 143 will result in node G responding that it has room for node X. Node X will then perform step 144 and be connected to the distribution network through node G as shown in FIG. 9. Here, node X becomes a fourth level node 15.

The distribution software may include a Return to Server Subroutine, comprised of steps 151, 152 and 153 as shown in FIG. 15, as part of the Prospective Child Node's Connection Routine. This subroutine reduces the risk that a prospective child node would enter into an endless cycle of fruitless attempts to dock with nodes in a particular tree. If the answer to the query in step 143 is "no," then node X goes to step 151 in which it increments by one the connection attempt counter 135 in node X's memory. Then node X goes to step 152 in which it determines whether the connection attempt limit has been reached. In one example (which example is intended to be illustrative and not restrictive), the limit may be

preset at any number greater than one and may depend upon what the designer of a particular distribution network determines would be a reasonable amount of time for a node to attempt to make a connection on a particular tree, or a branch of a tree, before that node should be given an opportunity to obtain a completely new connection address list directly from the server. If the connection attempt limit has not been reached, then node X proceeds with step 146 as discussed above. If the connection attempt limit has been reached, then node X goes to step 153, in which it goes back to the server and begins the connection routine again as discussed above in connection with FIG. 6. If docking with a parent node is successful, then after step 144 node X performs step 154 in which the connection attempt counter is set to zero.

Referring now to FIG. 16, a flow diagram associated with this example illustrating the Prospective Parent Node's Connection Routine is shown. Using the example illustrated in FIGS. 6, 8 and 9, when node X queries node B in step 142 during the Prospective Child Node's Connection Routine, node B may begin performing the Prospective Parent Node's Connection Routine. In step 161, in response to node X's query, node B determines whether it is part of the distribution system node X seeks to join. If node B were not part of the distribution network, it would respond in the negative to node X's query and node B would be finished with the Prospective Parent Node's Connection Routine. In the example of FIG. 8, the answer is "yes" and node B proceeds to step 162.

In step 162 node B determines whether it has room for a new node. If the answer were "yes," node B would proceed to step 163 where it would allow node X to dock with it, and node B would begin transmitting to node X data (e.g., streaming media) originating from the server. In the example illustrated in FIG. 8, the answer is "no," and node B, acting as an instructing node, goes to step 164 where it performs the Fully Occupied Parent's Connection Instruction Routine (discussed below) and provides the prospective new child node (here node X) with a new connection address list. As noted above, the new connection address list may include the path back to the server through the node B (the prospective parent in this example) in the event that there are user node departures along the path, which departures may include node B.

After performing step 164 the temporary connection between node B and node X is terminated, and node X is sent on its way. In the example of FIGS. 8 and 9, the new connection address list is G-B-A-S. When node X approaches node G, node G performs the Prospective Parent Node's Connection Routine discussed above. In the example illustrated in FIGS. 8 and 9, node G allows node X to dock with it.

Referring now to a "Distribution Network Construction" example (which example is intended to be illustrative and not restrictive), it is noted that under this example the length of the distribution chains (i.e., the number of levels of user nodes linked through each other to the root server) is aimed to be kept as small as possible.

Assuming that all new user nodes have the same re-transmission capability (or disregarding the different re-transmission capabilities which different new user nodes may have), the user nodes would be distributed in this example through the first level until all the direct connection slots to the server were filled. Then as new user nodes sought connection to the distribution network, they would be assigned to connections with the first level nodes 12 until all the slots available on the first level nodes were filled. This procedure would be repeated with respect to each level as more and more new user nodes attempted to connect to the distribution network. In other words, the server, acting as an instructing node, would perform a Server's Connection Instruction Routine in which one step is determining whether there is room on the server for a new user node and, if so, the server would instruct the new user node to connect directly to the server. If there were no room on the server, then the server would perform the step of consulting its network topology database and devising a connection address list having the fewest number of connection links to a prospective parent user node. After performing the Server's Connection Instruction Routine, the server would either allow the new user node to dock directly with the server or send the new user node on its way to the first prospective parent user node on the connection address list.

In this example (which example is intended to be illustrative and not restrictive), a partially occupied potential parent node in a particular level (i.e., a prospective parent node already having a child but still having an available slot for an additional child node) may be preferred over unoccupied (i.e., childless) potential parent nodes on the same level. This may help to keep the number of interior nodes (i.e., the number of nodes re-transmitting data) to a minimum.

Alternatively, in another example (which example is intended to be illustrative and not restrictive), nodes that have zero children may be preferred over nodes that already have one child. While it is true that this may increase the number of repeat nodes, what the inventors have found is that by filling the frontier in an "interlaced" fashion, connecting nodes build up their buffers more quickly (allowing a reduction in the incidence of stream interruptions).

Referring now to FIGS. 12A-12C, these FIGS. illustrate, in this example, what happens when a partially occupied parent node is preferred over a parent childless node as a

destination address for a new user node (assuming all other factors are equal). FIG. 12A is a schematic diagram associated with this example showing a topology wherein nodes C and D, both third level nodes 14, have remaining capacity for one or more child nodes. Server 11 has the choice of sending new user node X to either node C, as shown in FIG. 12B, or node D, as shown in FIG. 12C, without increasing the length of the longest chain in the distribution network. However, user nodes are free to leave the distribution network at any time. With the topology shown in FIG. 12C, if either of nodes C or D leaves the network, there would be an effect on a child node. With the topology shown in FIG. 12B, there is a significant chance that if one of nodes C and D leaves the network, there would be no impact on any child nodes. That is, if it is node D that leaves the network, there is no child which would be made an orphan.

Of note, in discussing the examples illustrated in FIGS. 6-9 and 12A-12C, it has been presumed that all of the user nodes are equally capable of simultaneously re-transmitting two copies of the content data. However, the user nodes actually joining a distribution network will have various capabilities for re-transmitting data. Some will have no reliable capability because of hardware limitations. Other user nodes will have high nominal capability because of their hardware and type of Internet connection, but may have low reliability because of poor line conditions or the vagaries of the desires of the humans actually using the user nodes. For example, a user node having a T1 or faster connection to the Internet has a significantly large bandwidth, but is not reliable if the user (or viewer) is merely sampling the content available on the distribution network.

As noted above, in one example the user nodes having the greatest reliable capability should be placed as high up in a distribution chain as possible (i.e., as far uptree as possible) because they would have the ability to support the greatest number of child nodes, grandchild nodes and so on.

Thus, to differentiate the reliable capabilities of user nodes, a number of factors may be considered. Examples of such factors (which examples are intended to be illustrative and not restrictive), include following:

- One is time, that is, the number of seconds (or other units of time) since the node made its most recent connection to the network. Everything else being equal, a user node which has been continuously connected to the distribution network for a long period of time may be considered under this example to be likely more reliable

(either because of line conditions or user interest) than a user node which has been continuously connected to the network for a short period of time.

- Another factor is bandwidth rating, which may be determined, for example, by actual testing of the user node when it first attempts to connect to the server or a parent node (e.g., at initial connection time) or determined by the nominal bandwidth as determined by the type of connection made by the user node to the Internet. In one example (which example is intended to be illustrative and not restrictive), we will describe ratings based on nominal bandwidth as follows:

    o A user node with a 56 Kbits/sec dialup modem connection to the Internet is essentially useless for re-transmission of content data because of its small bandwidth. In this example, such a node is assigned a bandwidth rating of zero (0.0).

    o A user node with a cable modem or DSL (e.g., ADSL) connection to the Internet may be given, in this example, a bandwidth rating of one (1.0), because it is a common type of node and has a nominal upstream transmission bandwidth of 128 Kbits/sec, which is large enough to potentially re-transmit two acceptable quality copies of the content data it receives (assuming a 50Kbit/sec content data stream). Such capability fits well into a binary tree topology.

    o A full rate ISDN modem connection nominally has an upstream bandwidth of 56 Kbits/sec and a downstream bandwidth of 56 Kbits/sec, which would potentially support acceptable quality re-transmission of a single copy of the content data stream (assuming a 50Kbit/sec content data stream). For this reason, a user node with a full rate ISDN modem connection to the Internet may be given, in this example, a bandwidth rating of one-half (0.5), or half the rating of a user node connected to the Internet by a DSL (e.g., ADSL) or cable modem connection.

    o User nodes with T1 or greater links to the Internet should be able to support more (e.g. at least twice as many) streams as DSL (e.g., ADSL) or cable modems, and therefore may be given, in this example, a bandwidth rating of two (2.0). In the event that in a distribution network parent nodes

may be assigned more than two child nodes directly connected thereto, bandwidth ratings greater than 2.0 may be assigned under this example to Internet connections having greater bandwidth than T1 connections.

- In another example (which example is intended to be illustrative and not restrictive), other bandwidth networks (e.g., 50 Kbit/sec and 100 Kbit/sec) may be supported.
- In another example, the bandwidth needed for a node to be "repeat capable" may depend on the particular network to which it is attached (upon initial connection the connecting node may first test its upstream bandwidth and then report that bandwidth to the network server; the server may determine whether the node is "repeat capable" or not.
- In another example (which example is intended to be illustrative and not restrictive), a node may be deemed "repeat capable" if the node is not firewalled (e.g., port 1138 is open) and the node has sufficient bandwidth to retransmit two copies of the incoming stream.
- A third factor is performance. A user node's performance rating may, in one example, be zero (0.0) if it is disconnected as a result of a Grandparent's Complaint Response Routine (discussed below in connection with FIG. 31). Otherwise, the user node's performance rating, in this example, may one (1.0).

Further, in this example (which example is intended to be illustrative and not restrictive), a user node's utility rating may determined by multiplying connection time by performance rating by bandwidth rating. That is, in this example:

- Utility Rating = Time x Performance Rating x Bandwidth Rating.

Referring once again to FIG. 13, it is seen that in this example information regarding a user node's time connected to the network, bandwidth rating, performance rating, utility rating and potential re-transmission rating (discussed below) may be stored in the user node's elapsed time buffer 137, bandwidth rating buffer 138, performance rating buffer 139, utility rating buffer 121 and potential re-transmission rating buffer 122, respectively.

In another example (which example is intended to be illustrative and not restrictive), nodes entering the network may be judged to be either "repeat capable" or "non-repeat capable" (wherein non-repeat capable nodes may be called "bumpable" nodes). Repeat

capability may be based on: (1) Upstream bandwidth (e.g., tested at initial connection); and (2) the firewall status, opened or closed, of a node. In one specific example, if a node is either firewalled or has upstream bandwidth less than (approx.) 2.5 times the video streaming rate, that node will be deemed bumpable. All nodes joining the network, whether bumpable or non-bumpable, may be placed as close to the server as possible. However, for the purpose of placement (as described for example in the Universal Connection Address Routine of Figure 20), a repeat capable node can bump a bumpable node when being placed in the network.

Referring now to FIG. 17 a flow diagram associated with this example illustrating the Server's Connection Instruction Routine is shown. This routine need not necessarily rely on the new user node's utility rating. This routine may rely on the potential re-transmission rating of a user node, which is arrived at by multiplying the user node's performance rating by its bandwidth rating. That is, in this example:

- Potential Re-transmission Rating = Performance Rating x Bandwidth Rating.

Thus, in this example, the server would want to put those user nodes with the highest potential re-transmission rating as close to the server in a distribution chain as possible (i.e., as high uptree as possible) because they have the greatest likelihood of being able to re-transmit one or more copies of content data. On the other hand, a potential re-transmission rating of zero in this example indicates that a user node has no ability to (or little expected reliability in) re-transmitting even one copy of content data (the server in this example would want to put a user node with a zero rating as far as reasonably possible from the server in a distribution chain (i.e., as low downtree as possible)). For the purpose of ease of discussion, in the flow diagram example of FIG. 17 illustrating the Server's Connection Instruction Routine, the server is concerned about whether the potential re-transmission rating is zero (i.e., either one or both of the performance rating and bandwidth rating is zero) or greater than zero (i.e., both the performance rating and the bandwidth rating are greater than zero).

In step 171 the server node interrogates the new user node (or connection requesting node) for its bandwidth rating and performance rating. If the new user node is really new to the distribution network, or if it has returned to the server because all of the user node's ancestor nodes have disappeared from the network, the new user node's performance memory will contain, in this example, a performance rating of 1.0 (e.g., the default rating). However, if the new user node has been dispatched to the server for a new connection to the network

because the new user node had failed to provide content data to one of its child nodes, then, in one example, its performance memory will contain a performance rating of zero.

In step 172, the server determines, in this example, whether the potential re-transmission rate of the connection requesting node is greater than zero (i.e., whether both the bandwidth rating and the performance rating are greater than zero, or, if only the bandwidth rating is considered, whether the bandwidth rating is greater than zero (i.e., the connection requesting node is a high-bandwidth node)). If the answer is "yes," then the server goes to step 173 in which the server determines whether it has room for the new user node. If the answer to the query in step 173 is "yes," then the server goes to step 174 in which it instructs the new user node to connect directly to the server. Then the server goes to step 102 in the Server's Connection Routine (see FIG. 10).

If the answer to the query in step 173 is "no" (i.e., the server does not have the capacity to serve the new user node directly), then the server goes to step 175. In step 175, the server, acting as an instructing node, consults its network topology database and devises a connection address list having the fewest number of connection links to a prospective parent node. That is, the server checks its information regarding the nodes in the level closest to the server (i.e., the highest uptree) to determine whether there are any potential parent nodes with space available for the new user node. If there are no potential parent nodes with space available, then the database is checked regarding nodes in the level one link further from the server, and so on until a level is found having at least one potential parent node with space available for the new user node. That is, the server determines which parent node with unused capacity for a child node is closest to the server (i.e., in the highest level, with the first level being the highest), and devises the connection address list from such prospective parent node to the server. The server then goes to step 102 (shown in FIG. 10).

In another embodiment of the invention, the server could skip step 172 and go directly to step 173 (or it could go to step 172 and, if the answer to the query in step 172 is "no" (i.e., either one or both of the bandwidth rating and the performance rating are zero, or, if only the bandwidth rating is considered, whether the bandwidth rating is zero (i.e., the connection requesting node is a low-bandwidth node)), the server could go to step 175). However, this could prevent the server from loading up the highest levels of the distribution chains with nodes capable of re-transmitting at least one acceptable copy of the content data (in contrast to if the server does perform step 172 and does goes to step 176 if the answer to the query in step 172 is "no.").

In step 176 the server consults its network topology database and devises a connection address list having the greatest number of connection links to a prospective parent node. That is, the server checks its information regarding the nodes in the level furthest from the server (i.e., the lowest downtree) to determine whether there are any potential parent nodes with space available for the new user node. If there are no potential parent nodes with space available, then the database is checked regarding nodes in the level one link closer to the server, and so on until a level is found having at least one potential parent node with space available for the new user node. In this manner, user nodes having limited or no reliable re-transmission capability may be started off as far from the server as possible and will have a reduced effect on the overall capacity of the distribution network.

As indicated above, and as discussed more fully below, one or more reconfiguration events may have transpired since the server's topology database was last updated. As a result, the first prospective parent node which is actually present on the distribution network for the new user node to contact may not have room for the new user node. By way of example (which example is intended to be illustrative and not restrictive), when node X tries to join the distribution network having the topology shown in FIG. 12A, the server provides node X with the following connection address list: C-B-A-S. If node C had disappeared from the network between the last update of the server's topology database and node X's attempting to contact node C, then node E, by virtue of a reconfiguration event, would be connected, in this example, to node B as shown in FIG. 12D. Then node X, in performing the Prospective Child Node's Connection Routine discussed in connection with FIGS. 14 and 15, would contact node B. Node B, in the Prospective Parent Node's Connection Routine, discussed in connection with FIG. 16, would have to answer the query of step 162 in the negative and go to step 164, in which it performs the Fully Occupied Parent's Connection Instruction Routine.

Referring now to FIG. 18, a flow diagram associated with this example illustrating that routine is shown. It is similar to the Server's Connection Instruction Routine. Since the fully occupied parent node has already determined that it has no room for the new user node (or connection requesting user node), the Fully Occupied Parent's Connection Instruction Routine does not need to include a step in which a determination is made regarding whether there is room for the new user node (or connection requesting node). In the Fully Occupied Parent's Connection Instruction Routine the fully occupied parent node, acting as an instructing node, first performs step 181 in which it interrogates the new user node for its bandwidth rating and performance rating. In step 182, the fully occupied parent node determines whether the potential re-transmission rate of the new user node is greater than

zero. (If only the bandwidth rating is considered, then it determines whether the bandwidth rating is greater than zero (i.e., the connection requesting node is a high-bandwidth node).) If the answer is "yes," then the fully occupied parent node goes to step 183 in which the fully occupied parent node consults its topology database, which contains the latest information available to that node regarding the links from the fully occupied parent node back to the server and regarding the fully occupied parent node's own progeny (i.e., its children, grandchildren etc.,) and devises a new connection address list having the fewest number of connection links to a new prospective parent node. That is, the fully occupied parent node checks its information regarding the nodes in the level closest to the fully occupied parent node (but not closer to the server) to determine whether there are any potential parent nodes with space available for the new user node. If there are no potential parent nodes with space available, then the database is checked regarding nodes in the level one link further from the fully occupied parent node (and further from the server), and so on until a level is found having at least one potential parent node with space available for the new user node. That is, the fully occupied parent node determines which new prospective parent node with unused capacity for a child node is closest to the fully occupied parent node, and devises the connection address list from such new prospective parent node through the fully occupied parent node and on to the server. The fully occupied parent node then goes to step 184 in which it provides the new user node with the new connection address list and disconnects the new user node from the fully occupied parent.

If the answer to the query in step 182 is no (i.e., either (i) one of the bandwidth rating and performance rating is zero or (ii) if only the bandwidth rating is considered, the bandwidth rating is zero (i.e., the connection requesting node is a low-bandwidth node)), the fully occupied parent node goes to step 185. In step 185 the fully occupied parent node consults its network topology database and devises a connection address list having the greatest number of connection links to a new prospective parent node. That is, the fully occupied parent node checks its information regarding the nodes in the level furthest from it (and farther from the server than it is) to determine whether there are any potential parent nodes with space available for the new user node. If there are no potential parent nodes with space available, then the database is checked regarding nodes in the level one link closer to the fully occupied parent node, and so on until a level is found having at least one potential parent node with space available for the new user node. As discussed above, this helps assure that new user nodes having limited or no reliable re-transmission capability are started off as far from the server as possible. After the fully occupied parent node devises the connection

address list from such new prospective parent node through the fully occupied parent node and on to the server, the fully occupied parent node then goes to step 184 where it performs as discussed above.

In another embodiment of the invention, the distribution software could be designed such that a fully occupied parent performs an abbreviated Fully Occupied Parent's Connection Instruction Routine, in which steps 181, 182 and 185 are not performed. That is, it could be presumed that the server has done the major portion of the work needed to determine where the new user node should be placed and that the fully occupied parent user node need only redirect the new user node to the closest available new prospective parent. In such event, only steps 183 and 184 would be performed.

In the example discussed above in which node C had disappeared from the network when new user node X had been given, by the server, connection address C-B-A-S, and in which node B is a fully occupied parent node as shown in FIG. 12D, node B would appear to have the choice of devising either connection address list D-B-A-S or E-B-A-S regardless of whether the full or abbreviated Fully Occupied Parent's Connection Instruction Routine were performed.

In this regard, to determine which connection address list to choose when there are two or more prospective parent nodes in a particular level having space for a new user node, the distribution software could have an additional subroutine as part of steps 175, 176, 183 and 185. An example of this subroutine, called the Multi-Node Selection Subroutine is illustrated in FIG. 19.

In step 191 the server or fully occupied parent node deciding where to send a new user node determines whether any of the potential new parent nodes is partially occupied. As discussed earlier, in one example a partially occupied potential parent node may be preferred over an unoccupied potential parent node. In this case, if any of the potential parent nodes is partially occupied, then the server or fully occupied parent node goes to step 192. In step 192 the partially occupied prospective parent node with the highest utility rating is selected as the new prospective parent node. If there were only a single partially occupied potential parent node, then that node is selected.

If in step 191 it is determined that there are no partially occupied potential parent nodes, then the server or fully occupied parent node goes to step 193. In step 193 the unoccupied prospective parent node with the highest utility rating is selected as the new prospective parent node.

In another example (which example is intended to be illustrative and not restrictive),the software engineer could have step 193 follow an affirmative response to the query in step 191 and step 192 follow a negative response; in such event, unoccupied prospective parent nodes would be selected ahead of partially occupied prospective parent nodes (all other things being equal, it may be advantageous to select nodes that have zero children over nodes that already have one child; while it is true that this may increase the number of repeat nodes, what the inventors have found is that by filling the frontier in an "interlaced" fashion, connecting nodes build up their buffers more quickly (allowing a reduction in the incidence of stream interruptions).

After whichever of steps 192 and 193 is completed, the server or fully occupied parent node returns to the step from which it entered the Multi-Node Selection Subroutine (i.e., step 175, 176, 183 or 185), and completes that step.

So, in the example shown in FIG. 12D, where node C left the network thereby leaving node B with the chore of devising a new connection address list for node X, node B would perform the Fully Occupied Parent's Connection Instruction Routine. Regardless of the bandwidth and performance ratings of node X, node B would be choosing between nodes D and E in the third level. In step 191 node B would determine that neither D nor E is partially occupied, and therefore node B would go to step 193. Assuming that nodes E and D have equal bandwidth and performance ratings and that node D was connected to the network longer than node E, node D would be selected because it would have the higher utility rating since it was connected to the network longer than node E. Node B would then go to step 194 and then return to the step from which it entered the Multi-Node Selection Subroutine. When node B returns to step 183 or 185, it completes that step and moves on to step 184. In that step, node B provides new user node X with new connection address list D-B-A-S and node X connects to the distribution network as shown in FIG. 12E.

Referring now to an "Alternative Distribution Network Construction" example (which example is intended to be illustrative and not restrictive), it is noted that under this example the length of the distribution chains (i.e., the number of levels of user nodes linked through each other to the server) is aimed to be kept as small as possible.

Also under this example, user nodes having the highest bandwidth capabilities are aimed to be kept closer to the server (e.g., in order to allow the greatest expansion of the distribution system). However, it is possible that zero bandwidth rated nodes may nevertheless appear relatively far uptree (thereby stunting the growth of that chain). The following method may be used in constructing the distribution network both by servers and

by prospective parents which are actually completely occupied, either of which may be thought of as an instructing node (that is, software enabling the routines discussed below could be installed on servers and user nodes alike).

More particularly, in the distribution system of this example, each child node reports directly to (or is tested by) its parent node with respect to information relating to the child node's bandwidth rating, performance rating, potential re-transmission rating and utility rating. In turn, each parent node reports all the information it has obtained regarding its child nodes on to its own parent node (a parent node also reports to each of its child nodes the address list from that parent back to the server, which list forms what may be referred to as the ancestor portion of the topology database – in addition, a parent node reports to each of its child nodes the addresses of their siblings). The reports occur during what may be referred to as a utility rating event. Utility rating events may occur, for example, on a predetermined schedule (e.g., utility rating events may occur as frequently as every few seconds). As a result of all the reporting, each node stores in its topology database the topology of the tree (including all its branches) extending below that node, and the server stores in its topology database the topology of all the trees extending below it. This may be referred to as the descendant portion (or descendant database) 133 of the topology database (see FIG. 13). The descendant database of a particular node may include a descendant list, a list of the addresses of all the nodes cascadingly connected below that particular node. Included in the topology database information may be the utility ratings of the nodes below the node in which that particular topology database resides.

After each reporting event in this example, each parent node (including the server), acting as an instructing node, devises two lists of prospective (or recommended) parent nodes. In this example, the first list, or Primary Recommended Parent List ("PRPL"), stored in the Primary Recommended Parent List buffer 123 (see FIG. 13), lists all the nodes in the descendant portion of that node's topology database which have bandwidth available to support another child node (in one example (which example is intended to be illustrative and not restrictive), in a binary tree system, all nodes in the descendant portion of the topology database having (i) a bandwidth rating of at least one and (ii) less than two child nodes would be listed). They would be listed with those node's which are closest to the node in which that particular topology database resides at the top of the list, and those nodes which are in the same level would be ranked such that the node with the highest utility rating would be listed first, the node with the second highest utility rating would be listed second and so on. By way of example (which example is intended to be illustrative and not restrictive), the PRPL of a

second level node would list a third level node with available bandwidth ahead of a fourth level node with available bandwidth even if the fourth level node's utility rating were higher than that of the third level node.

The second list in this example, or Secondary Recommended Parent List ("SRPL"), stored in the Secondary Recommended Parent List buffer 124 (see FIG. 13), lists all the nodes in the descendant portion of that node's topology database which have the ability to re-transmit content data to child nodes but are fully occupied, and at least one of its child nodes is incapable of re-transmitting content data to another child node (in one example (which example is intended to be illustrative and not restrictive), in a binary tree system, all nodes in the descendant portion of the topology database having (i) a bandwidth rating of at least one and (ii) at least one child node having a bandwidth rating less than one (i.e., being incapable of re-transmitting content data to two child nodes) would be listed). Like the nodes in the PRPL, the nodes in the SRPL would be listed with those node's which are closest to the node in which that particular topology database resides at the top of the list, and those nodes which are in the same level would be ranked such that the node with the highest utility rating would be listed first, the node with the second highest utility rating would be listed second and so on.

Of note, in this example the SRPL lists those parent nodes having the growth of their branches (i.e., their further progeny) blocked or partially blocked by a low-bandwidth child node. This may lead to an unbalanced growth of the distribution system, and a limitation on the total capacity of the system.

Of further note, to the extent that a node (including a server) has room for another child node or is the parent of a low bandwidth node, it may be listed on its own PRPL or SRPL.

Referring now to FIGS. 20A and 20B, an example flow diagram illustrating what may be referred to as the Universal Connection Address Routine is shown. As indicated above, a server or a fully occupied prospective parent node receiving a connection request may be referred to as an "instructing node." When a new node (here node N) approaches an instructing node, the instructing node performs step 201, it receives a connection request. It then goes to step 202 in which it interrogates or tests node N to determine whether it is a low-bandwidth node ("low-bandwidth" may mean, for example, either or both: (1) a node that is incapable of broadcasting two copies of its incoming stream on to its children; and (2) a node that is incapable of rebroadcasting its stream because it is firewalled). In the binary tree system of one example, a low-bandwidth node is a node with a bandwidth rating less than

one. If node N is a low bandwidth node, the instructing node proceeds to step 203 in which the instructing node determines whether there are any available prospective parent nodes which are not fully occupied. Sometimes the distribution network may be fully occupied. If it is, the instructing node's PRPL will be empty. If it is empty, the response to the query in step 203 would be yes. Then the instructing node goes to step 209 in which the new node is sent back to the server to start the connection process from the beginning. If the response to the query in step 203 is no, then the instructing node goes to step 204 in which it selects a prospective (or recommended) parent node for node N. The instructing node then moves on to step 205 in which it consults its topology database and devises a connection address list from the recommended parent node back to the server, and provides such connection address list to node N. (If the instructing node is a user node, then the connection address list leads back to the server through the instructing node.) At this point, node N performs the Prospective Child Node's Connection Routine discussed above in connection with FIG. 14.

As part of step 205, the instructing node inserts node N into its topology database as a child of the recommended parent node. It does this because other new nodes may seek connection instructions prior to the next utility rating event (i.e., before reports providing updated information regarding the topology of the distribution network are received by the instructing node), and such new nodes should not be sent to a prospective parent which the instructing node could know is fully occupied. In this regard, the instructing node then goes to step 206 in which it checks its topology database to determine whether the recommended parent, with node N presumptively docked to it as a child node, is fully occupied (in the example here of a binary tree network, whether it has two child nodes). If the answer is no, then the instructing node has finished this routine. If the answer is yes, then the instructing node goes to step 207 in which it removes the recommended parent from the PRPL and inserts it into the SRPL, and the instructing node has finished the routine.

If the answer to the query in step 202 is no (e.g., node N is not a low-bandwidth node; in the binary tree network example it is a high-bandwidth node capable of re-transmitting content data to two child nodes), the instructing node moves on to step 208. There it determines whether both the PRPL and SRPL are empty (which may occur under certain circumstances, such as, for example, when the number of levels in a distribution system is capped and at least all the nodes on all but the last level are fully occupied with high-bandwidth nodes). If so, the instructing node goes to step 209 in which the new node is sent back to the server to start the connection process from the beginning. If the response to the query in step 208 is no, then the instructing node goes to step 210 in which it selects a

prospective (or recommended) parent node for node N from the PRPL and an alternate recommended parent node from the SRPL (if either the PRPL or SRPL is empty, the instructing node will make a "nil" selection from that list – the instructing node knows from step 208 that at least one of the lists will not be empty). The instructing node then goes to step 211 in which it determines whether the alternate recommended parent node is closer to the server (i.e., higher uptree) than the recommended parent node derived from the PRPL. If the alternate recommended parent node is on the same level as, or on a lower level than the recommended parent node derived from the PRPL (or if the selection from the SRPL is nil), then the answer to the query in step 211 is no.

In such event, the instructing node goes to step 212 in which it consults its topology database and devises a connection address list from the recommended parent node back to the server, and provides such connection address list to node N (if the instructing node is a user node, then the connection address list leads back to the server through the instructing node).

As in step 205 discussed above, at this point, node N performs the Prospective Child Node's Connection Routine discussed above in connection with FIG. 14, and in step 212, the instructing node inserts node N into its topology database as a child of the recommended parent node.

The instructing node moves to step 213 in which it adds node N to an appropriate position on the PRPL. It does this because, as a result of step 202, it knows that the node N is capable of re-transmitting content data to its own child nodes.

The instructing node then goes to step 214 in which it checks its topology database to determine whether the recommended parent, with node N presumptively docked to it as a child node, is fully occupied (in the example here of a binary tree network, whether it has two child nodes). If the answer is no, then the instructing node has finished this routine. If the answer is yes, then the instructing node goes to step 215 in which it removes the recommended parent from the PRPL because it is now deemed to not be an available prospective parent node.

Then the instructing node goes to step 216 in which it consults its topology database to determine whether any of the recommended parent node's children is a low-bandwidth node (in this example, knowing that node N is not a low-bandwidth node and knowing that the recommended parent node has two child nodes, the question is whether the child node other than node N is a low-bandwidth node.) If the answer is no (i.e., all the recommended parent node's children are high-bandwidth nodes), then the instructing node has finished the routine.

If the answer is yes, then the growth of the recommended parent node's line of progeny is partially blocked by a low-bandwidth child node. The instructing node moves on to step 217 in which it adds the recommended parent to the SRPL.

If the answer to the query in step 211 is yes (i.e., the alternate recommended parent (selected from the SRPL) is: (i) on a higher level than the recommended parent (selected from the PRPL) or (ii) the selection from the PRPL is nil), then the instructing node moves to step 218. In that step the instructing node consults its topology database to determine: (i) which of the alternate recommended parent node's child nodes is a low-bandwidth node or (ii) if they both are low-bandwidth nodes, which of the child nodes has been connected to the system a shorter period of time. In one example (which example is intended to be illustrative and not restrictive), the instructing node may send a disconnect signal to that child node with instructions to return to the server to start the connection process from the beginning (as a new user node (or connection requesting user node)). In another example (which example is intended to be illustrative and not restrictive), the bumped node may climb its path to its grandparent which gives it a new connection path – rather than returning to the root server (doing so typically means that the incoming node that bumped the non-repeat-capable node ends up being that node's new parent).

The instructing node moves on to step 219 in which it consults its topology database and devises a connection address list from the alternate recommended parent node back to the server, and provides such connection address list to node N (if the instructing node is a user node, then the connection address list leads back to the server through the instructing node).

As in steps 205 and 212 discussed above, at this point, node N performs the Prospective Child Node's Connection Routine discussed above in connection with FIG. 14, and in step 219, the instructing node inserts node N into its topology database as a child of the alternate recommended parent node.

The instructing node moves to step 220 in which it adds node N to an appropriate position on the PRPL. It does this because, as a result of step 202, it knows that the node N is capable of re-transmitting content data to its own child nodes.

The instructing node then goes to step 221 in which it checks its topology database to determine whether all the child nodes of the alternate recommended parent are high-bandwidth nodes. If the answer is no, then the instructing node has finished this routine. If the answer is yes, then the instructing node goes to step 222 in which it removes the alternate recommended parent from the SRPL because it is now deemed to not be an available

alternative prospective parent node since the growth of its progeny line is not even partially blocked by one of its own children. At this point the instructing node has finished the routine.

With the routines discussed in connection with the above examples, the distribution network will be built with each new node assigned to the shortest tree (or chain), and those with the fewest number of links between it and the server. However, low-bandwidth nodes, which would tend to block the balanced growth of the distribution network, would be displaced by high-bandwidth nodes and moved to the edges of the network where they would have reduced effect on the growth of the network.

Referring now to FIG. 21, this FIG. may be used as an example to illustrate how the Universal Connection Address Routine works. It shows primary server 11 with node A as first level node 12, its child nodes B and C as second level nodes 13, their child nodes D, E, F and G as third level nodes 14, node D and G's child nodes H, I, J and K as fourth level nodes 15, and node I's child nodes L and M as fifth level nodes 17. Assume for this example that all available docking links directly to the server 11 are occupied by high-bandwidth nodes, and that all trees are at full capacity other than that rooted to the server 11 through first level node A. The utility rating for each node is set forth in FIG. 21 under its letter designation. Low-bandwidth nodes (in this example, those nodes having a bandwidth rating less than one) are shown with a bandwidth rating of "Ω," which indicates in this example that such nodes will not have any child nodes assigned to them. Nodes A, B, C, D, E, G, I, J and K are high-bandwidth nodes (or repeater nodes (i.e., they are capable of re-transmitting the content data they receive from their respective parent nodes to their respective child nodes, if any)).

A new user node (or other connection requesting user node), here referred to as "node N", may be directed to node A as a result of various reconfiguration events. When node A receives a connection request from node N, node A will, in this example, either consult both of its PRPL and SRPL buffers, if node N is a high-bandwidth node, or consult just its PRPL buffer (if node N is a low-bandwidth node).

Nodes F, H, L, and M will not appear on any list in this example since they are low-bandwidth nodes. Nodes B, C, D, G, I and A itself will not appear on the PRPL in this example since these nodes are fully occupied. However, nodes C, D, and I will appear on the SRPL in this example because they each have at least one low bandwidth child node.

Using the rules discussed above (i.e., ranking prospective parent nodes by level and within levels by utility rating) the PRPL would be as follows in this example:

- PRPL

- E (a third level node with available capacity)
  - J (the fourth level node with available capacity having the highest utility rating)
- K (a fourth level node with available capacity)

The SRPL in this example would be as follows:

- SRPL
  - C (the fully occupied second level node with a low-bandwidth child node)
  - D (the fully occupied third level node with a low-bandwidth child node)
  - I (the fully occupied fourth level node with a low-bandwidth child node)

If node N is a low-bandwidth node, node A will give, in this example, node N the following as its new connection address list:

- E-B-A-S.

Since node E would not have two child nodes, it would remain on the PRPL.

If node N is a high-bandwidth node, node A will compare (step 211 of FIG. 20B) the first node on the PRPL (the recommended parent node) with the first node on the SRPL (the alternate recommended parent node). Here, node C, the first node on the SRPL is a higher level node than node E, the first node on the PRPL. So, node A will send a disconnect signal (step 218 of FIG. 20B) to node F, node C's low-bandwidth child node. Then it will provide node N with the following new connection address list and add node N to the PRPL (step 219 of FIG. 20B):

- C-A-S.

Since node C would now have two high-bandwidth child nodes (nodes N and G), node C would be removed from the SRPL (step 222 of FIG. 20B).

Further, in this example wherein node N is a high-bandwidth node, when node F returns to the server 11 (or, in another example, goes to Node A) for a new connection, the

server (or Node A) will also use the Universal Connection Address Routine (Node F may go to Node A rather than the server because (in one example) when a node is "kicked" it may climb its internal connection path rather than jumping directly to the root server). Since node F is a low-bandwidth node, the server (or Node A) will give node F the following as its new connection address list:

- E-B-A-S.

FIG. 22 illustrates the new topology. As can be seen, absent intervening events, low-bandwidth node F will end up moving down from the third level to the fourth level and the bandwidth capacity of the third level will increase from six child nodes to eight (its maximum in this binary tree example). The potential bandwidth capacity of the fourth level would also increase, from ten child nodes to twelve.

Referring now to a "Distribution Network Reconfiguration" example (which example is intended to be illustrative and not restrictive), it is noted that under this example the user nodes are free to exit the distribution network at will. And, of course, user nodes may exit the distribution network due to equipment failures between the user node and the communication network (the Internet in the examples discussed herein). This was first discussed in connection with FIGS. 10A-E.

Under this example the present invention may handle the reconfiguration required by a user node departure by having certain user nodes in the same tree as the departing node perform a Propagation Routine. The results of the propagation routine of this example are illustrated in FIGS. 23 and 24. There a tree rooted to the server through first level user node P is illustrated. Node P has two child nodes, second level nodes Q and X. Through node X, P has two grandchild nodes, third level nodes A and B. Based on the relative utility ratings of nodes Q and X, P has sent signals to its children instructing them to set the color of the flag represented by the information in their respective reconfiguration flag buffers to "green" and "red," respectively. The use of colors as designators is merely discretionary. In reality they represent relative ratings of siblings for purposes of determining their roles in a reconfiguration event, and may be referred to as "propagation ratings." In addition, instead of colors, numbers (e.g., 1 and 2) could be used, as well as any of many other nomenclature schemes to describe the grades of propagation ratings. The "red" "green" ratings discussed herein shall be deemed representative of all such schemes. The number of grades of propagation ratings assigned by a parent node may be equal to the number of children each

parent node has. For example, for a branch factor "n" system, the maximum number of grades of propagation ratings may be "n." Since the distribution network in the examples discussed herein is a binary tree distribution network, a parent node will be required to assign at most up to two grades of propagation ratings.

In the example shown in FIG. 23, node P, during the most recent utility rating measurement event, discerned that node Q has a higher utility rating than node X, and therefore P has assigned node Q a green rating, represented by the solid-lined circle surrounding the letter Q in FIG. 23. P has assigned node X a red rating, represented by the dashed-lined circle surrounding the letter X in FIG. 23. In a similar manner, node X has assigned green and red ratings to third level nodes A and B, respectively. In this example (which example is intended to be illustrative and not restrictive), at the same time that a parent node assigns a propagation rating to a child node, it may also provide to such child node the address of the child node's sibling, if there is any. A child node may store information about its sibling (or siblings) in the sibling portion (or sibling database) 134 of its topology database (see FIG. 13). The sibling database includes a sibling list, a list of the addresses of a node's siblings (the data relating to the siblings' addresses may also contain information regarding the propagation ratings of the siblings (e.g., in the event that the distribution system has a branch factor greater than two)). In the example shown in FIG. 23, nodes Q and X know that they are each other's siblings and nodes A and B know that they are each other's siblings.

In another example (which example is intended to be illustrative and not restrictive), nodes do not store information about their siblings. In other words, in this example, a node generally does not know who its sibling is. Instead, cross connections between a red node and its green sibling are implemented via a "priority join" message sent to a red child that includes the IP address of the green sibling as the first entry in the connection path list.

In the event that node X were to leave the distribution network, nodes A and B would of course stop receiving content data from node X. Nodes A and B would consult their topology databases for the address of their grandparent, and each would interrogate node P for instructions. Since node X is out of the distribution network, node P would send out a reconfiguration event signal (sometimes referred to as a "propagation signal") to the nodes which had been node X's children (i.e., the nodes which were node P's grandchildren through node X) and node P would send an upgrade propagation rating signal to its remaining child, here node Q.

In response to the upgrade propagation rating signal, node P's remaining child, node Q, would set its reconfiguration flag buffer 136 (see FIG. 13) to the next highest available propagation rating grade. Since in the example illustrated in FIG. 23, node Q's reconfiguration flag buffer already is set for the highest propagation rating grade (here green), node Q could either do nothing or reset its propagation rating in response to the upgrade propagation rating signal. The result would be the same, node Q's propagation rating would remain green. If node Q's propagation rating were red, then it would set its propagation rating to green in response to the upgrade propagation rating signal. Node Q would do nothing else in response to the upgrade propagation rating signal (note that as a matter of design choice, the software engineer could have the parent of the departed node (here node P is the parent of departed node X) send no upgrade propagation rating signal to its remaining child node (here node Q) if its remaining child node already is at the highest propagation rating).

In this example, the recipients of the propagation signal (i.e., the children of the missing node) would respond thereto as follows. First they would check their own respective propagation ratings. If the propagation signal's recipient has the highest propagation rating grade (here green), it would reset its propagation rating to the lowest rating grade (here red); re-transmit the propagation signal to its own child nodes (if there are any); disconnect the child node which did not have the highest propagation rating prior to its receipt of the propagation signal if the propagation signal's recipient with the green rating has more than one child node; allow its sibling to dock with it as a child node for the purpose of transmitting content data to that child node; and dock (or remained docked), for purposes of receiving content data, with the node sending the propagation signal to it (in systems having a branch factor of more than two, the propagation signal recipient whose propagation rating had been the highest would disconnect its child nodes which did not have the highest propagation rating, prior to the receipt of the propagation signal just sent, and it would allow its (i.e., the formerly green node's) siblings to dock with it as child nodes).

If the propagation signal's recipient has other than the highest propagation rating (i.e., just prior to the receipt of the propagation signal), it would upgrade its propagation rating to the next highest rating grade; dock with its sibling which had the highest rating grade; and begin receiving content data from it. If the propagation signal's recipient has other than the highest propagation rating, it does not re-transmit the propagation signal to its own child nodes.

In the example illustrated in FIG. 23, nodes A and B receive the propagation signal from node P. Since node A has the highest propagation rating grade, here green, it: (i) sets its

reconfiguration flag buffer 136 (see FIG. 13) so that it has the lowest propagation rating grade, here red; and (ii) docks with node P (becoming a second level node 13) to begin receiving content data from node P. Node B changes its propagation rating from red to the next higher propagation rating (and since this example is a binary tree (or branch factor two) system, the next higher rating is the highest, green) and docks with node A to receive content data. The resulting topology is shown in FIG. 24. Note that node B remains a third level node 14.

FIGS. 25 and 26 illustrate what happens in this example when the departing node is green. In FIG. 25, node X is the departing node and it is green. When node P sends the upgrade propagation rating signal to node Q, it changes its propagation rating from red to green. In other respects the reconfiguration event proceeds essentially as described in the paragraph immediately above, and results in the topology shown in FIG. 26 (which is the same as the topology in FIG. 24).

FIG. 27 is a flow diagram associated with an example showing the Child Node's Propagation Routine, the routine which is followed by a child node upon receiving a propagation signal. Of note, this example is provided for illustration only, and is not restrictive, and other examples are of course possible (e.g., the color of a node may assigned by its parent such that nodes assign and know the color of their children but not of themselves).

In any case, still referring to FIG. 27, the child node first performs step 271 wherein it determines whether it has received such a signal. If it has not, then it does nothing. If it has received a propagation signal, it proceeds to step 272 wherein it checks its propagation rating grade in reconfiguration flag buffer 136 (see FIG. 13) . If its propagation rating is at the highest grade (i.e., it is a "green" node in this example), then it proceeds to step 273 where it sets its reconfiguration flag buffer to the lowest propagation rating grade. It then proceeds to step 274 in which it re-transmits the propagation signal to its own child nodes, and which results in the undocking of all its child nodes except for the one with the highest propagation rating. The child node (i.e., the child node referred to in the second sentence of this paragraph) then performs step 275 in which it determines whether it is already docked with the node which sent the propagation signal. If it is not (i.e., it received the propagation signal from its grandparent), then it proceeds to step 276. In that step it: (i) devises a new connection address list, which is the ancestor portion of its topology database with the current nominal parent node removed, resulting in the grandparent node becoming the first node on the connection address list; and (ii) then performs the Prospective Child Node's Connection

Routine (i.e., it goes to step 141 discussed above in connection with FIGS. 14 and 15). The Prospective Child Node's Connection Routine is performed because some likelihood exists that even the grandparent node may have departed from the distribution system between the moment it sent out the propagation signal and the moment that its highest rated grandchild from its missing child attempted to dock with it.

If the answer to the query in step 275 is in the affirmative (i.e., the child node is receiving a propagation signal which has been re-transmitted by its parent), then the child node does nothing more (or as illustrated in FIG. 27, it performs step 277 which is to remain docked to its parent node).

If the answer to the query in step 272 is in the negative (i.e., it does not have the highest propagation rating (it is a "red" node in this example)), then it proceeds to step 278 in which it: (i) sets its reconfiguration flag buffer so that its propagation rating is the next higher grade (in this binary tree system the rating is upgraded from red to green); (ii) undocks from the parent with which it was docked before receiving the propagation signal (if it was docked with a parent before receiving such signal); and (iii) either (a) waits a predetermined period of time during which the node's sibling which was green prior to their receipt of the propagation signal should have re-transmitted the propagation signal to its own child nodes (thereby causing any red child nodes to undock from it) or (b) confirms that the node's sibling which was green prior to their receipt of the propagation signal actual did re-transmit the propagation signal to its child nodes (if it has any). Then the child node performs step 279 in which it: (i) devises a new connection address list, which is the ancestor portion of its topology database with its sibling node having the highest propagation rating placed in front of the previous parent as the first node on the connection address list; and (ii) then performs the Prospective Child Node's Connection Routine (i.e., it goes to step 141 discussed above in connection with FIGS. 14 and 15). The Prospective Child Node's Connection Routine is performed because some likelihood exists that the sibling may have departed from the distribution system between the moment that the propagation signal had been sent to the child node and the moment that the child node attempted to dock with its sibling.

In another example (which example is intended to be illustrative and not restrictive), part (iii) of step 278 may be handled using the "priority join" mechanism. More particularly, in this example, a priority join from a node X cannot be refused by a target node T – it must accept (dock) the incoming node. If the target node T already had two children G & R, the target node T instructs its red child R to connect (via "priority join") to the green node G. The node T then disconnects from its red child R immediately prior to accepting the incoming

node X. The bumped node R then sends a priority join to its own green sibling A as the target. These actions cause the reconfiguration events to cascade until the edge of the tree is reached (of note, the algorithm described in this paragraph may have essentially the same effect as illustrated in Figures 28 and 29 – under certain circumstances the algorithm described in this paragraph may be somewhat more reliable).

FIG. 28 illustrates an example distribution system comprising node P as a first level node 12; nodes Q and X as second level nodes 13; nodes A and B as third level nodes 14; nodes C, D, E an F as fourth level nodes 15; nodes G and H as fifth level nodes 17; nodes I and J as sixth level nodes 20; nodes K and L as seventh level nodes 21 and nodes M and N as eighth level nodes 22. Nodes Q, B, C E, H, I, L and M have red propagation ratings, as symbolized by the dashed circles, and the remaining nodes have green propagation ratings as symbolized by the solid circles. As a result of node X's departing from the system, node P sends out an upgrade propagation rating signal to node Q and a propagation signal to the children of node P's departed child node X (i.e., nodes A and B).

In another example (which example is intended to be illustrative and not restrictive), during a graceful depart of node x, node x may send out messages to its parent and children immediately prior to its departure – rather than the parent of node x initiating these messages as described above.

In any case, continuing the example of FIG. 28, node Q changes its propagation rating to green. Further, in this example nodes A and B begin the Child Node's Propagation Routine. With respect to node A, it answers the query of step 271 in the affirmative. Since its rating is green, it also answers the query of step 272 in the affirmative. In step 273 node A changes the setting of its reconfiguration flag buffer to show a red propagation rating. In step 274 it re-transmits the propagation signal to its child nodes C and D. It answers the query of step 275 in the negative because it is not docked for purposes of receiving content data with node P. Node A then goes to step 276 wherein it consults the ancestor portion of its topology database and creates a new connection address list starting with its grandparent. The new connection address list is P-S. Then node A performs the Prospective Child Node's Connection Routine, starting with step 141. (See FIG. 14.) Assuming that no other intervening events have occurred, node A will successfully dock with node P.

Node B answers the query of step 271 in the affirmative and, because its propagation rating was red when it received the propagation signal, it answers the query of step 272 in the negative. From there it goes to step 278 in which it changes its propagation rating to green. Then in step 279 node B consults the ancestor portion of its topology database and creates a

new connection address list with its sibling, node A, placed in front of node X as the first address on the new connection address list. The new connection address list is A-X-P-S. Then node B performs the Prospective Child Node's Connection Routine, starting with step 141. Assuming that no other intervening events have occurred, node B will successfully dock with node A (if node A has disappeared, node B would attempt to dock with node X, but since it is not there, it would move on to node P.)

Node A, which had the higher propagation rating of the two child nodes of departed node X, moves up one level to become a second level node 13. Since it is a "new" entry to the second level, its initial utility rating and its propagation rating will be lower than that of node Q. As time goes by, at subsequent utility rating measurement events node A's utility rating (and hence its propagation rating) may become higher than that of node Q.

Node B, which had the lower propagation rating of the two child nodes of departed node X, does not re-transmit the propagation signal to its child nodes (nodes E and F), and they will follow B wherever it goes in the distribution system. In the example discussed here, node B becomes the child of node A while remaining third level node 14. At least initially, it will be node A's child with the higher utility rating and propagation rating.

As a result of node A's performing step 274, its child nodes C and D (which were fourth level nodes 15 in FIG. 28 when node X was still in the system) receive a propagation signal. Since node C has a red propagation rating, it, like node B, will remain in its level, change its propagation rating to green and dock with its sibling, with the result being that node D becomes its parent.

Since node D had a green propagation rating when it received the propagation signal, it answers the queries of steps 271 and 272 in the affirmative and changes its propagation rating to red in step 274. It answers the query of step 275 in the affirmative, and remains docked with node A. As a result, node D moves up a level and becomes an third level node 14 (with, at least until the next utility rating event, a lower utility rating and propagation rating than its new sibling, node B).

This process proceeds down the tree, with each child of a node which moves up a level doing one of the following under this example: (i) if it is a node having a green propagation rating, it remains docked with its parent, thereby itself moving up a level, changes its propagation rating to red and re-transmits the propagation signal to its child nodes; or (ii) if it is a node having a red propagation rating, it docks with the node which was its sibling, thereby staying in the same level, and changes its propagation rating to green.

The resulting topology after the reconfiguration event caused by node X's departure is illustrated in FIG. 29.

Of note, it is believed that reconfiguration using the Child Node's Propagation Routine of this example when a node departs the system does not necessarily result, in the long run, in significantly more inter-node reconnection events than any other reconfiguration method (and may result in fewer reconnection events). Moreover, it is believed that reconfiguration using the Child Node's Propagation Routine of this example helps assure that the more reliable nodes are promoted up the distribution network even if many reconfiguration events occur close in time to each other in a particular tree.

Referring now to a specific example of pseudo-code (which example is intended to be illustrative and not restrictive), it is noted that a propagation routine may be set forth as follows (wherein node X is the node departing from the distribution system):

```
Departs(X)
Begin
  If (X.Parent.GreenChild == X)  /* the departing node is green*/
  {
    X.Parent.GreenChild := X.Parent.RedChild
    X.Parent.RedChild := X.GreenChild
  }
  Else /* the departing node is already red */
  {
    X.Parent.RedChild :=X.GreenChild
  }
  Propagate(X.GreenChild, X.RedChild)
End
```

Referring now to another specific example of pseudo-code (which example is intended to be illustrative and not restrictive), it is noted that a propagation routine may be set forth as follows (wherein node promotion during the network reconfiguration event is precipitated by node X's departure):

```
Propagate (X, RedSibling)
Begin
  OriginalGreenChild := X.GreenChild
```

```
OriginalRedChild := X.RedChild
X.RedChild := X.GreenChild
X.GreenChild := RedSibling
If (OriginalGreenChild <> null)
{
  Propagate(OriginalGreenChild, OriginalRedChild)
}
End
```

Referring now to a "Malfunctioning Nodes" example (which example describes a voting system between grandchildren and grandparents about a potentially malfunctioning parent and is intended to be illustrative and not restrictive), it is noted that what happens when a node leaves the distribution network was discussed in the above example (e.g., when a node leaves the network, its parent will, at the next utility rating event, report that it has room for a new node, except when the parent has been contacted about the missing node by its grandchild node or nodes, in which event, a reconfiguration event will proceed as described above). Sometimes, however, a node intends to remain in the network, but there is a failure of communication.

In this regard, each node's topology database may include information regarding that node's ancestors and descendants. In the event that a child node stops receiving signals from its parent node, the child may, in one example, send a "complaint" message to its grandparent. The grandparent may check whether the parent is still there. If it is not, then the grandparent may send a propagation signal to the child nodes of the missing parent.

Further, if the grandparent detects that the parent node is actually still there, then the grandparent node may send a disconnect signal to the parent node (e.g., sending it back to the server to begin the connection process again). This may occur, for example, when one of the two following conditions is exists: (i) the child node is the only child node of the parent, or (ii) the child and its sibling(s) are complaining to the grandparent.

In this example, the grandparent would also send a propagation signal to the child nodes of the disconnected parent, and a reconfiguration event would occur.

However, if the child node has siblings and they are not sending complaint signals to the grandparent, then the grandparent may assume that the problem is with the complaining child node. Thus, grandparent may send a disconnect signal to the complaining child node (e.g., sending it back to the server to begin the connection process again). If the complaining

child node had its own child nodes, they would contact the departed child node's parent to complain, starting a reconfiguration event (in another example (which example is intended to be illustrative and not restrictive), the grandparent would send out signals to the malfunctioning parent (instructing it to depart) and to the children (instructing the green child to climb its path to its grandparent and instructing the red child to cross connect to the green child).

The foregoing example (which example is intended to be illustrative and not restrictive) can be described in connection with FIG. 30 (which depicts a tree in a binary system having a primary server 11, in which node A, a first level node 12, has two child nodes B and B' (which are second level nodes 13); node B has two child nodes C and C' (which are third level nodes 14); node C has two child nodes D and D' (which are fourth level nodes 15); and node D has two child nodes E and E' (which are fifth level nodes 17)) and FIG. 31 (which is a flow diagram showing an example Grandparent's Complaint Response Routine).

Who does node C complain to in this example when node C no longer gets satisfactory service from node B? Node C complains to its grandparent, node A. If node C does not hear back from node A within a predefined amount of time (e.g., if node A has left the network), node C will then exit the network and immediately reconnect as a new node by going to the primary server S for assignment.

What does node A do in response to a complaint from node C in this example? When node A receives a complaint from grandchild node C about node C's parent node B (i.e., node A's child node B), node A will choose to either remove its child node B or its grandchild node C. A will make this determination based on whether node C alone, or both node C and its sibling node C' are experiencing problems with node B, together with the knowledge of whether node A is continuing to receive satisfactory service reports from node B. If node A is continuing to get satisfactory service reports from node B and there is no indication that node C' (the sibling of node C) is experiencing problems with node B, then node A will assign the "fault" to node C and issue a disconnect order for its removal. At this point the green child of node C (i.e., node D or D') will move up a level connecting to parent B, and the red child of node C (the other of node D or D') will connect as a child of its former sibling. The reconfiguration event will then propagate as discussed above.

If, on the other hand, node A is not getting satisfactory service reports in this example from node B and/or a complaint from node C's sibling arrives within a `window` of node C's complaint, then node A will assign the "fault" to node B and issue a disconnect order for its

removal. At this point the green child of node B (i.e., node C or C') will move up a level connecting to grandparent node A, and the red child of node B (the other of node C or C'), will connect as a child of its former sibling. The reconfiguration event will then propagate as discussed above.

An exception to the above is the case where node C is the only child of node B. Under these circumstances node B may be disconnected in this example by node A based solely on the recommendation of node C. In another example, the system will not disconnect a node with one child based solely on the complaint of that child (in this case there may be insufficient data to accurately assign blame - disconnecting the node being complained against and promoting the complainer may have the potential to promote a node that is experiencing problems).

Referring now to FIG. 31 (showing a flowchart associated with this example), node A receives a complaint from node C about node B in step 311. Node A then goes to step 312 in which it checks its topology database to determine whether node C is the only child of node B. If the answer is no (as shown in FIG. 30), then node A goes to step 313 in which node A determines whether there is a communication problem between it and node B. If the answer is no, then node A proceeds to step 314 in which it determines whether it has received a similar complaint about node B from node C' within a predetermined period of time of node A's having received the complaint from node C. If the answer to that query is no, then node A goes to step 315 in which it sends a disconnect signal to node C. At that point node A has completed the Grandparent's Complaint Response Routine (in one example a propagation signal does not necessarily have to sent by node A to node C's child nodes - if any exist, they will complain to node B which, upon discerning that node C is no longer in the distribution network, will send a propagation signal to such complaining nodes).

If the response to any of the queries in steps 312, 313 and 314 is yes in this example, then node A proceeds to step 316 in which it disconnects node B. Node A then proceeds to step 317 in which it sends a propagation signal to its grandchild nodes (here nodes C and C') and a reconfiguration event will occur as described above.

The method of this example described above takes a "2 out of 3" approach on the question of node removal. In this regard, it is believed that many interior nodes in a binary tree distribution network constructed pursuant the present invention will have direct connections to three other nodes: their parent, their green child, and their red child. The connections between a node and its three "neighbor" nodes can be thought of as three separate communications channels. In this example a node is removed from the network

when there are indications of failure or inadequate performance on two of these three channels. When there are indications that two of the channels are working normally, the complaining node is presumed to be "unhealthy" and is removed.

In the case where an interior node has only a single child, the "complaint" of that child to its grandparent is sufficient, in this example, to remove the parent node - even when the node to be removed is communicating perfectly well with it's own parent. In other words, given the communication chain A-B-C where node A is the parent of node B and node B is the parent of node C, and given that node C has no siblings, then a complaint of node C to node A will cause node A to remove node B in this example regardless of the fact that node A and node B are not having communication problems (this course of action may help ensure that a potentially unreliable node does not move up the hierarchy of nodes, even at the cost of occasionally sending a "healthy" node back to the edge of the distribution chain based on uncorroborated complaints of its child). Of course, as mentioned above, there are examples where the system will not disconnect a node with one child based solely on the complaint of that child.

In another example of voting (which example is intended to be illustrative and not restrictive), instead of (or in conjunction with) nodes voting on the behavior of other nodes, nodes may continually grade their own performance and if they detect that there is insufficient bandwidth on two or more of their links (links to their parent and their children) the node will perform a graceful depart from the network (initiating a reconfiguration as described elsewhere) and then the node will either climb its path to an ancestor node (or go directly to the server) to request a connection path to the edge of the tree. This system allows nodes to "step aside" when they detect connection / bandwidth related issues.

Referring now to a "Stepped Delay of Playing Content Data" example (which example is intended to be illustrative and not restrictive) it is noted that in order to have all the user nodes experience the playing of the content data at approximately the same time, regardless of the level in which the node resides, the content data buffer 125 shown in FIG. 13 of a user node may be larger the higher uptree it is in a distribution chain (i.e., the closer it is to the root server). The larger the buffer, the greater the amount of time between a particular event (or from the node's receiving the content data), and the node's actually playing the content data in its content data buffer. To the extent that the content data buffer of a node is sized to vary the time that the playing of the content data is started, the content data buffer is a delay. Alternatively, the delay may be a timer or a delay buffer 126 as shown in

FIG. 13, or a combination of elements. The delay's purpose may be to help assure that all users experience the play of the content data approximately simultaneously.

For a distribution network having n levels of user nodes cascadingly connected to a server node, where n is a number greater than one (and therefore the distribution network comprises the server node and first through nth level user nodes), the period of time (i.e., the delay time) between:

(i) the moment content data is received by a node or, from a predetermined moment such as, for example, a reporting event or utility rating event (the occurrence of which may be based on time periods as measured by a clock outside of the distribution system and would occur approximately simultaneously for all nodes); and

(ii) the playing of the content data by the node;

is greater for a node the higher uptree the node is on the distribution chain. That is, the delays in first level nodes create greater delay times than do the delays in second level nodes, the delays in second level nodes create greater delay times than the delays in third level nodes and so on. Described mathematically, in an n level system, where x is a number from and including 2 to and including n, the delay time created by a delay in an (x-1) level node is greater than the delay time created by a delay in an x level node.

By way of example (which example is intended to be illustrative and not restrictive), first level nodes could have a built in effective delay time in playing content data of one-hundred-twenty seconds, second level nodes could have a built in effective delay time of one-hundred-five seconds, third level nodes could have a built in effective delay time of ninety seconds, fourth level nodes could have a built in effective delay time of seventy-five seconds, fifth level nodes could have a built in effective delay time of sixty seconds, sixth level nodes could have a built in effective delay time of forty-five seconds, seventh level nodes could have a built in effective delay time of thirty seconds, and eighth level nodes could have a built in effective delay time of fifteen seconds.

The delays may take advantage of the fact that transmission of the packets forming the content data from one node to another may take less time than the actual playing of such packets. This may allow for essentially simultaneous playing of the content data throughout all the levels of the distribution network. The delays may also allow nodes to handle reconnection and reconfiguration events without any significant loss of content data.

In another example (which example is intended to be illustrative and not restrictive), the system may not endeavor to have buffers closer to the server contain more data than buffers of nodes that are further away. Instead, every node may have a buffer that can hold,

for example, approximately two minutes worth of A/V content. The system may endeavor to fill every node's buffer as rapidly as possible, regardless of the node's distance from the server, in order to give each node the maximum amount of time possible to deal with network reconfiguration and uptree node failure events.

Of note, nodes may receive a "current play time index" when they connect to the server. This value may be used to keep all nodes at approximately the same point in the stream – to have all nodes playing approximately the same moment of video at the same time – by telling them, essentially, begin playing A/V content at this time index. The play time index may be, for example, approximately two minutes behind "real time" – allowing for nodes to buffer locally up to two minutes of content (the present invention may permit drift between nodes within the buffer (e.g., two minute buffer) or the present invention may reduce or eliminate this drift to keep all nodes playing closer to "lock step").

Turning now to a general description of an example process for Node X to enter a network including server Node S, Node X must contact Node S to initiate the process. For illustrative purposes, in the following example (which example is intended to be illustrative and not restrictive) each node in the network may have no more than two connections, and the root server (i.e., the highest node in the tree) shall be referred to as Node S.

The process of this example begins with Node X making a request of Node S to join the network.

If Node S has an open connection (e.g., including one or more connected bumpable non-repeat-capable nodes which may be replaced due to an incoming connection request from a repeat-capable node), Node S can accept Node X's request to join the network, and Node X becomes a child of Node S.

If, on the other hand, Node S does not have an open connection, Node S may reject Node X's request, and may provide to Node X a node-by-node path from Node S to Node P, which Node S understands to be in the network, and under which, as determined by Node S, Node X may be able to join the network.

The process of this example continues with Node X making a request of Node P to join the network.

If Node P does not respond – regardless of why Node P does not respond – Node X may request connection from the next higher node in the path. For the purpose of this illustration, thereafter the next higher node is now considered Node P. Node X may make such a request of successive Node Ps, until Node X receives a response.

If a particular Node P has an open connection, that Node P can accept Node X's request to join the network, and Node X becomes a child of that particular Node P.

If, on the other hand, a particular Node P does not have an open connection, that particular Node P may reject Node X's request, and may provide to Node X a node-by-node path from that particular Node P to Node P', which the instructing Node P understands to be in the network, and under which, as determined by the instructing Node P, Node X may be able to join the network.

If Node X receives a further rejection, Node X may then treat the Node P' as a Node P and continue the process.

In general, under this example the process is continued until Node X finds a connection.

Referring now more specifically to a "Registration Process" example involving two arbitrary Nodes X and N (neither of which is necessarily a server), FIG. 32 shows Node X sending a request to Node N for acceptance into the distribution network tree (of which Node N is part).

In this example, Node N (which may be thought of as an instructing node) may generally return one of three codes (or messages): Accept, Path, or Reject. FIGS. 33A-33C show example messaging between Node X and Node N under this example. More particularly:

- FIG. 33A shows Node X sending a Register message to Node N and Node N responding to the Register message with an Accept message
- FIG. 33B shows Node X sending a Register message to Node N and Node N responding to the Register message with a Path message (e.g., identifying a path from Node N to a new instructing node)
- FIG. 33C shows Node X sending a Register message to Node N and Node N responding to the Register message with a Reject message (i.e., rejecting Node X's request for a connection to Node N).

In another example, the registration process may be extended to include some system maintenance or configuration activities. Such maintenance or configuration activities may include "flood testing" (discussed in detail below), identification of a bad protocol, and/or Firewall Status (discussed in detail below).

In this regard, FIG. 34A shows the messaging (in connection with registration with a flood test) between Node X, a node requesting to join the network, Node S, a server node

within the network, and Node R (e.g., a random (or predetermined) node that may, but not need be, within the network).

Further, FIG. 34B shows the messaging (in connection with registration with a bad protocol version) between Node X and Node S, a server node within the network. Of note, such a bad protocol message may be sent by the server to a connection requesting node to inform that node that it is running an obsolete version of the client software and must upgrade to the current version before being allowed to join the network.

Further still, FIG. 34C shows the messaging (in connection with registration with standard response) between Node X and Node S, a server node within the network (wherein such standard response includes, for example, Accept / Reject / Path).

Referring now in more detail to the "flood test" example shown in FIG. 34A, this example process begins, as above, with Node X making a request to Node S to join the network. Node S then sends Node X a Meta message (such a Meta message may be sent by the server to a connection requesting node to inform that node of certain characteristics of the network such as, for example: the streaming bandwidth rate used by the network, the server's public key for authenticating that secure packets originated from this server, and any other information the architects of the network might deem necessary or desirable for nodes to know when connecting to the network). Node S may also require that Node X undergo performance or flood testing before being accepted or placed within the network. In such an event, Node S responds to Node X's request to join the network with an instruction that Node X undergo a flood test (e.g., of Node X's upstream channel). Such an instruction may be accompanied by an address of Node R, the illustrative flood test target. Note that any node within or without the network may conduct the flood test. One purpose of the flood test may be to permit Node S to be aware of Node X's upstream capability before determining where (or if) to assign Node X in the network.

Where Node S requires that Node X undergo performance or flood testing before being accepted or placed within the network, Node S provides a message so informing Node X, shown as "DoFlood" in FIG 34A. The DoFlood message may provide to Node X the address of the flood test target, but the flood test target could be Node S itself, or a predetermined flood test target.

In response to the DoFlood message, Node X sends an InitFlood message to the flood test target. In one embodiment, Node X would await an acknowledgement (not shown) of its InitFlood message, but in another embodiment, awaiting an acknowledgement is not

necessary. Node X would then begin transmitting FloodData messages, as shown in FIG 34A.

Since, in this example, FloodData messages are transmitted for a purpose of determining Node X's upstream bandwidth, in one embodiment, Node X would transmit an amount of FloodData to the flood test target as shown in FIG 34A. In various embodiments, the maximum amount of FloodData may be limited to a fixed number of FloodData messages, to a fixed amount of data transmitted within FloodData messages, or to transmission of FloodData or FloodData messages over a fixed period of time.

To protect the flood test target from being inundated with FloodData (e.g., where Node X has a very high upstream bandwidth capability), Node X may cap (e.g., be so directed by Node S) the FloodData transmission to, e.g., 100 Kbits/sec, 200 Kbits/sec, or some other data rate that would be a desired data transmission rate for a node on the particular network of interest.

Thus, by receiving the FloodData messages, the flood test target can determine, for example, an upstream capability rating of Node X (provided that the flood test target has the same or more bandwidth in the downstream channel compared to the upstream capability of Node X.

In another example, both the target and the sender will know the bandwidth of the sender at the conclusion of the flood test. More particularly, the present invention may use TCP/IP (rather than UDP), and all packets sent may receive an acknowledgement. In this example, Node X, the node sending out the flood packets, sends out the next flood packet only after receipt of the first packet has been acknowledged by the target of the flood test. Thus, the number of flood data packets Node X is able to send per unit time (along with the size of those packets) may allow Node X to calculate its upstream bandwidth.

In another example, the target of the "flood test" may be any node including the server but excluding any firewalled or low-bandwidth nodes.

After the "flood test", Node X may then connect with Node S (or be instructed to go to another node if Node S can not/will not accept Node X). Of note, if Node X fails the "flood test", Node X may be sent far downtree or may be refused entry into the network entirely. Further, if node N is accepted under these circumstances it will be deemed a "bumpable" non-repeat-capable node.

Referring now in more detail to the "Firewall Status" example shown in FIG. 34C, this example process begins, as above, with Node X making a request to Node S to join the network. Node S then send Node X a Meta message. A firewall status message is a message

sent over a new connection from (in the case of Figure 34C) Node S to Node X. If Node X receives this message, over a communications channel initiated by Node S, Node X will know that it is not firewalled. Failure to receive a firewall status message when a node is expecting such a message allows that node to conclude that it is in fact firewalled. Thus, if Node X is expecting to receive a firewall status message over new connection initiated by some other node and that message does not arrive, Node X can conclude (with a high degree of likelihood of being correct) that it is firewalled. Also, since (in one example) the present invention uses the TCP/IP messaging protocol the sender, Node S, will also know whether the firewall status message was successfully received by Node X and thus Node S will also know whether Node X is or is not firewalled (one or more other nodes may also receive data regarding the firewall status).

After receiving the firewall status report (not shown) Node S will then respond with a standard response (accept, path, reject) as discussed above.

To now give a specific example of the use of both the "flood test" and the "firewall status" mechanisms during registration of a new node, FIG. 35A shows a state of the network before Node X joins the network and FIG. 35B shows a state of the network after Node X joins the network. Further, FIG. 36 shows the messaging among Node X, Node S, Node B and Node R. More particularly:

- Node X sends a Register message to Node S and Node S responds with a Meta message (providing Node X with basic information about the network) and a DoFlood message (which requests Node X to determine its bandwidth and report back to Node S when Node X has that information). Node S knows to send a "DoFlood" to Node X because the initial Register message sent by Node X to Node S contains an indication that Node X does not yet know its current upstream bandwidth capacity. The DoFlood message will contain the address of a target node (Node R), selected by Node S, that Node X should flood against.
- Node X sends an InitFlood message to Node R and then "floods" Node R with data for some period of time (e.g., 15 seconds). At the conclusion of this test Node X will be able to compute its upstream bandwidth based on the number and size of data packets it was able to successfully transmit to Node R.

- Node X sends another Register message to Node S and Node S responds with a Meta message. Contained within this second Register message is the result of Node X's "flood test" which gives an indication of Node X's upstream bandwidth capacity.
- Node S now tests whether external nodes can initiate communication with Node X by sending Node X a Firewall Status message over a new connection. If this message is successfully received by Node X (and acknowledgement received by Node S), both Nodes X and S will know that Node X is not firewalled (on a network communication port).
- Node S sends a Path message to Node X. This message will include a list of nodes that X should attempt connection with, beginning with the preferred target node (e.g., Node B). As described elsewhere in the application, the choice of target node, and hence the contents of the path, may be influenced by both the current configuration of the network and whether or not Node X is repeat-capable (based on the results of the "flood test" and "firewall test").
- Node X sends a Register message to Node B and Node B responds with an Accept message.

Referring now to a "Firewall Request / Firewall Status" example involving two arbitrary Nodes A and B (neither of which is necessarily a server), FIG. 37 shows a portion of a network including Node A and Node B. In this example (as shown in FIG. 38), a Firewall Request message is sent by Node B to request that the recipient of the message, Node A, test Node B's Firewall Status. The receiving Node A may send (e.g., over a new connection), the Firewall Status message to Node B. If Node B timely receives and interprets the Firewall Status message it can be certain that it is a non-firewalled node. If Node B does not timely receive a Firewall Status message from Node A then Node B may assume that it is firewalled.

In another example (which example is intended to be illustrative and not restrictive), if Node A fails to connect to Node B or to send the Firewall Status message no further action is needed on the part of Node A.

Node B may also augment its assumption about its Firewall Status by using a three strikes (i.e., three attempts) method before considering itself to be firewalled (of course, three strikes is provided as an example only, and any desired number of "strikes" or attempts may be used).

In another example (which example is intended to be illustrative and not restrictive), the multiple attempts may be carried out within a predetermined time "window" (after which receiving no Firewall Status message Node B may consider itself firewalled).

In another example (which example is intended to be illustrative and not restrictive), Firewall Requests may be sent out by all nodes (e.g., at regular time intervals) in order to maintain a current and accurate picture of their Firewall Status.

In another example (which example is intended to be illustrative and not restrictive), Firewall Requests may be sent out by certain node(s) (e.g., at regular time intervals) in order to maintain a current and accurate picture of their Firewall Status.

In another example (which example is intended to be illustrative and not restrictive), the Firewall Status message may be sent by the server on initial registration (e.g., without the need for a Firewall Request message and both the sender (the server) and the receiving node (the node that is registering) may use the successful delivery of the Firewall Status message as an indicator of the connecting node's Firewall Status.

Referring now to a "Propagate Path / Down" example (which example is intended to be illustrative and not restrictive), FIG. 39A shows a portion of a network including Node A and Node B. Further, FIG. 39B shows an example of nodes in a tree sending Propagate Path / Down messages downtree to their children (e.g., at regular time intervals).

In one example (which example is intended to be illustrative and not restrictive), the Propagate Path / Down message may include topology information containing a path back to the root server and the time interval for sending such message may be calculated on a node-by-node and/or level-by-level basis relative to the amount of data to be sent in the Propagate Path / Down message (wherein the amount of data in the Propagate Path / Down message is related to a particular node's (or level's) downtree position relative to the root server). Note, that although this time interval may not be very important for Propagate Path / Down, since the amount of data for each successive downtree node (or level) may increase relatively slowly (e.g., linearly), the time period may be of critical importance for Propagate NTM / Up (which is discussed below), since the amount of data for each successive uptree node (or level) may increase relatively quickly (e.g., exponentially). Of note, the acronym "NTM" refers to the Network Topology Model.

In any case, FIG. 40 shows the messaging between Node A and Node B in connection with a Propagate Path / Down message (i.e., Node A sends a Propagate Path message downtree to Node B).

Referring now to a "Propagate NTM / Up" example (which example is intended to be illustrative and not restrictive), FIG. 41A shows a portion of a network including Node A and Node B. Further, FIG. 41B shows an example of nodes in a tree sending Propagate NTM / Up messages uptree to their parents (e.g., at regular time intervals).

In one example (which example is intended to be illustrative and not restrictive), the Propagate NTM / Up message may include information containing the topology of the subtree(s) below a particular node and the time interval for sending such message may be calculated on a node-by-node and/or level-by-level basis relative to the amount of data being sent in the Propagate NTM / Up message (wherein the amount of data in the Propagate NTM / Up message is related to a particular node's (or level's) uptree position relative to tree height (e.g., the height of a particular node above its lowest leaf level)). Note that in this example (which example is intended to be illustrative and not restrictive), the NTM data that the node sends up is rooted at itself and contains both its children.

In any case, FIG. 42 shows the messaging between Node A and Node B in connection with a Propagate NTM / Up message (i.e., Node B sends a Propagate NTM message uptree to Node A).

In another embodiment, the following process may be employed when a node enters the network. More particularly, when the server adds a node X to the system (gives it a connection path to node Y) the server will add node X to its network topology model as a child of node Y. In other words, the server will assume that the connection will take place normally with no errors. The reason why doing so may be critical to the functioning of the system in this embodiment is that if the server failed to account for the fact that node X is now (probably) a child of node Y, using the "keep the tree balanced" approach embodied in the Universal Connection Address Routine, all new nodes connecting to the system would be assigned as children of node Y up until the point where a propagate up arrives indicating that node X was successfully added to the tree. If the tree is large and node Y is many levels removed from the server it could be some time (e.g., a minute or more, in very large trees several minutes) before confirmation arrives that node X truly connected to node Y as its child.

Now, in addition to simply placing node X into the server's NTM (Network Topology Model) and being done with the matter, in this embodiment the system must account for the fact that it may take a number of propagate NTM/up cycles for the true status of node X to be reported back to the server. Thus, for some number of propagate cycles (or some amount of time) the server may compare the newly propagated (updated) NTM it is receiving from its

children to the list of "recently added nodes". If a recently added node is not in the propagated NTM it will be re-added to the NTM as a child of node Y. At some point node X will no longer be considered "recently added" and the server will cease adding node X back into its NTM. In other words, the server will assume that, for whatever reason, node X never successfully connected to the network.

In another embodiment, the following process may be employed when a node gracefully exits the system. More particularly, when a node (node X) leaves the network by performing a graceful exit (e.g., the user selects "exit" or clicks the "x" or issues some other command to close the application) the node will send signals to its parent and child nodes (if any) immediately prior to actually shutting down. Upon receipt of such a "depart" message, the parent of the departing node (node P) will modify its internal NTM (Network Topology Model) to a configuration that corresponds to the predicted shape of the network after the departing node exits. In this embodiment, node P will assume that its NTM is accurate and that the entire cascade of reconfiguration events initiated by the departure of node X will happen normally. As such, node P will now expect the former green child of node X (node XG -- if such a child exists) to connect – essentially taking node X's place in the network.

At this point, node P's NTM represents a prediction of the (anticipated) state of the network rather than the actual reported configuration. In order to indicate that node P does not (yet) have a physical connection to node XG, node P will mark node XG as "soft" in its NTM – meaning node XG is expected to take up this position in the network but has not yet physically done so. Node P will report this "predicted" NTM (along with node XG's status as a soft node) up during its propagate NTM/Up cycles, until node P physically accepts a child connection in node XG's "reserved" spot and receives an actual report of the shape of the downtree NTM from that child, via propagate NTM/ups.

Of note, the above actions may be taken in order to have the propagated NTM's reflect, as accurately as possible, the topology of the network.

Of further note, because of the distributed nature of a network according to the present invention, a complete and accurate picture of the current shape of the network at any instance in time can never be guaranteed. In fact, though the root server has the "widest" view of the network, without the heuristics described herein, that view would be the "most dated" of any node in the network. Thus, heuristics such as those just described may be vital to producing NTM's that are as close to complete and accurate as possible (e.g., so that the Universal Connection Address Routine can position incoming nodes so as to keep the distribution tree as nearly balanced as possible).

Referring now to a "Banned List" example (which example is intended to be illustrative and not restrictive), it is noted that such a "banned list" (or "banished list") may be used for keeping certain nodes off of the network, removing certain nodes from the network, and/or keeping certain nodes as far downtree as possible (e.g., in the case of a binary tree network keeping certain nodes as "red" nodes).

More particularly, FIG. 43 shows a portion of a network including Node A, Node B, Node C and Node D and FIG. 44 shows the messaging between Node A, Node B, Node C and Node D in connection with Banned List messages (i.e., in this FIG. 44, Node A sends a Banned List message to Node B and Node B sends a Banned List message to Nodes C and D).

In one example (which example is intended to be illustrative and not restrictive), nodes receiving this message should re-transmit the message downtree to all children in an unaltered fashion.

In another example (which example is intended to be illustrative and not restrictive), the Banned List messages may include IP lists and/or URL's.

In another example (which example is intended to be illustrative and not restrictive), the Banned List messages may originate at the server.

In another example (which example is intended to be illustrative and not restrictive), Banned List messages may be sent and/or originated by any desired node(s).

In another example (which example is intended to be illustrative and not restrictive), Banned List messages and may be sent using a Secure Packet format.

In another example (which example is intended to be illustrative and not restrictive), node(s) may store the content of the message (e.g., the banned list itself) for future use.

In another example (which example is intended to be illustrative and not restrictive), the server (or any other desired node(s)) may send out and/or originate the Banned List messages at regular time intervals.

Referring now to a "Bumpable Node" embodiment, it is noted that this "bumpable node" mechanism may be used to "bump" (or send downtree) a particular node (e.g., when a new node is added). More particularly, in one example (which example is intended to be illustrative and not restrictive), a server may instruct Node X to connect with Node P and Node P may know to "bump" its child, Node C, by: (a) instructing Node C to disconnect from Node P; (b) allowing Node X to connect to Node P; and (c) instructing Node C to re-connect as a child of Node X. Of note, such a "bumping" operation may be carried out, for example (which example is intended to be illustrative and not restrictive), because the

bumped node is firewalled and/or because the bumped node has failed an upstream and/or downstream bandwidth test (which test may be carried out by the server or by any other node(s)).

Of note, as used in this application, the terms "bumpable nodes" or "bumpable non-repeat-capable nodes" or "non-repeat-capable nodes" are nodes which either: (a) have insufficient bandwidth to act as repeaters in the network or (b) are firewalled.

Of further note, in one example the more inclusive "is bumpable" test may replace the "is low bandwidth" test in the Universal Connection Address Routine of Figure 20.

Referring now to a "Load Manager" embodiment, it is noted that this "load manager" mechanism may be used to rate limit (i.e., slow or stop) the addition of new nodes to the network. Such rate limiting may be useful because the larger the network grows, the more nodes can join per unit of time.

In one example (which example is intended to be illustrative and not restrictive), the rate limited addition of new nodes may be carried out by a server (or connection server) and may be based upon the length of time that it takes for the buffers associated with one or more nodes (or levels) of the existing network to fill (or reach a certain value). In a more specific example, the rate limited addition of new nodes may be based upon the length of time that it takes for the buffers associated with potential parents of the new nodes to fill (or reach a certain value).

In another example (which example is intended to be illustrative and not restrictive), the rate limited addition of new nodes may be carried out by any desired node(s) (e.g., one or more user nodes and/or a combination of one or more user nodes and a server (or connection server) node).

Referring now to a "Node Interlacing" embodiment, it is noted that this "node interlacing" mechanism may be used to add new nodes to the network by interlacing the new nodes on a level-by-level basis as shown in FIG. 45.

More particularly, as seen in this FIG. 45, new nodes may be added in the order indicated (as can be seen below, this example (which example is intended to be illustrative and not restrictive) relates to a binary tree network). That is, Node 1 may first be added as a child of the server followed by Node 2 as a child of the server – this will fully populate the first level; Node 3 may next be added as a child of Node 1, followed by Node 4 as a child of Node 2, followed by Node 5 as a child of Node 1, followed by Node 6 as a child of Node 2 – this will fully populate the second level; Node 7 may next be added as a child of Node 3, followed by Node 8 as a child of Node 5, followed by Node 9 as a child of Node 4, followed

by Node 10 as a child of Node 6, followed by Node 11 as a child of Node 3, followed by Node 12 as a child of Node 5, followed by Node 13 as a child of Node 4, followed by Node 14 as a child of Node 6 – this will fully populate the third level, etc.

Of note, such node interlacing may be used to efficiently fill (or bring to a certain value) the buffers associated with one or more nodes (or levels) of the existing network while new nodes are added.

Of further note, such node interlacing may help prevent adding a second child to a parent until the first child's buffer is full (or brought to a certain value).

Of still further note, such node interlacing may help to: (a) prevent taxing the parent's downtree channel; and/or (b) prevent two children being left with partially filled buffers in the event that their parent departs.

Of still further note, such node interlacing may be used, for example, in conjunction with or as an alternative to the load balancing mechanism discussed above.

In another embodiment the present invention may be used when a plurality of computer systems are connected (e.g., to the Internet) by one or more of an ISDN line, a DSL (e.g., ADSL) line, a cable modem, a T1 line and/or an even higher capacity link.

In another embodiment a distribution network for the distribution of data from a server node to a plurality of user nodes connected to the server node, wherein each of the user nodes has associated therewith a buffer is provided, which distribution network comprises: a mechanism for rate limiting the addition of new user nodes to the distribution network, wherein the rate limited addition of new user nodes to the distribution network is based at least in part upon the length of time that it takes for at least one buffer associated with at least one user node to reach a predetermined percentage of capacity, wherein the predetermined percentage of capacity is greater than zero.

In another embodiment a distribution network for the distribution of data from a server node to a plurality of user nodes connected to the server node, wherein each of the user nodes is in one of a plurality of levels located downtree from the server node is provided, which distribution network comprises: a mechanism for adding new user nodes to the distribution network in an interlaced manner, wherein the interlacing provides that each sibling user node in a given level of the distribution network is associated with just one child before any siblings in the given level are associated with two children.

The following embodiments will now describe a number of additional example algorithms which may be utilized in whole or in part to implement the present invention:

In one embodiment a process for docking a connection requesting user node with a distribution network is provided, said process including the following steps:

(a) having an instructing node receive a connection request from said connection requesting user node;

(b) forming an instructing node topology database indicating, at a point in time:

(i) which, if any, user nodes are docked with said instructing node as child nodes; which, if any, user nodes are docked with each of said child nodes as grandchild nodes of said instructing node; which, if any, user nodes are docked with each of said grandchild nodes as great-grandchild nodes of said server node; and so on; and

(ii) said docked user nodes respective bandwidth capacities;

(c) selecting from a group of nodes including the instructing node and said docked user nodes a recommended parent node for said connection requesting user node, wherein said recommended parent node has apparent available capacity to transmit content data to said connection requesting user node and is at least as close (i.e., in terms of network topology) to said instructing node as any other docked user node having apparent available capacity to transmit content data to said connection requesting user node (of note, placement may also or alternatively depend, as described elsewhere in the present application, in whole or in part on whether the connection requesting node is repeat-capable or not (i.e., whether it can support children)); and

(d) providing a connection address list to said connection requesting user node, said connection address list listing each node in said instructing node's topology database from said recommended parent node back to said instructing node.

In one example, the process may further comprise the following steps:

(e) having said connection requesting user node go to (or attempt to go to) the node at the top of said connection address list;

(f) having said connection requesting user node determine whether the node at the top of said connection address list is part of the distribution network;

(g) if the node at top of said connection address list is not part of the distribution network, deleting such node from said connection address list and repeating steps (e) and (f) with respect to the next node at the top of said connection address list;

(h) if the node at the top of said connection address list is part of said distribution network, having said connection requesting user node dock with said node at the top of said connection address list.

In another example, step (d) may further comprise the following step:

(i) if said instructing node is not a server node, including in said connection address list a list of nodes cascadingly connected with each other from the instructing node back to said server node.

In another example, the process may further comprise the following steps:

(e) having said connection requesting user node go to (or attempt to go to) the node at the top of said connection address list;

(f) having said connection requesting user node determine whether the node at the top of said connection address list is part of the distribution network;

(g) if the node at the top of said connection address list is not part of the distribution network, deleting such node from said connection address list and repeating steps (e) and (f) with respect to the next node at the top of said connection address list;

(h) if the node at the top of said connection address list is part of said distribution network, determining whether said node at the top of said connection address list has available capacity to transmit content data to said connection requesting user node; and

(i) if the node at the top of said connection address list is part of said distribution network and has available capacity to transmit content data to said connection requesting user node, having said connection requesting user node dock with the node at the top of said connection address list.

In another example, the process may further comprise the following steps:

(j) if the node at the top of connection address list is part of said distribution network and said node at the top of said connection address list does not have available capacity to transmit content data to said connection requesting user node, repeating steps (a)-(d), with said node at the top of connection address list being said instructing node and a new connection address list being the product of steps (a)-(d).

In another example, the process may further comprise the steps of counting the times that step (j) is performed and, when step (j) is performed a predetermined number of times, having said connection requesting user node repeat steps (a)-(d) with said primary server being the instructing node.

In another embodiment a process for docking a connection requesting user node with a distribution network is provided, said process including the following steps:

(a) having an instructing node receive a connection request from said connection requesting user node;

(b) forming an instructing node topology database indicating, at a point in time:

(i) which, if any, user nodes are docked with said instructing node as child nodes; which, if any, user nodes are docked with each of said child nodes as grandchild nodes of said instructing node; which, if any, user nodes are docked with each of said grandchild nodes as great-grandchild nodes of said server node; and so on; and

(ii) said docked user nodes' respective bandwidth capacities;

(c) assigning utility ratings to each of the docked user nodes, said utility ratings being a function of at least the bandwidth capacity of the docked user node and the elapsed time that the docked user node has been docked with the distribution network;

(d) selecting from a group of nodes including the instructing node and said docked user nodes a recommended parent node for said connection requesting user node, wherein said recommended parent node has apparent available capacity to transmit content data to said connection requesting user node and is at least as close (i.e., in terms of network topology) to said instructing node as any other docked user node having apparent available capacity to transmit content data to said connection requesting user node, and when the closest (i.e., in terms of network topology) docked user node with apparent available capacity to transmit content data to said connection requesting user node is equidistant (i.e., in terms of network topology) to said instructing node with at least one other docked user node with apparent available capacity to transmit content data to said connection requesting user node, said recommended parent node is the docked user node having the highest utility rating among such equidistant (i.e., in terms of network topology) docked user nodes with apparent available capacity; and

(e) providing a connection address list to said connection requesting user node, said connection address list listing each node in said instructing node's topology database from said recommended parent node back to said instructing node and, if said instructing node is not a primary server node, back to said primary server node.

In another embodiment a process for docking a connection requesting user node with a distribution network is provided, said process including the following steps:

(a) having an instructing node receive a connection request from said connection requesting user node;

(b) forming an instructing node topology database indicating, at a point in time:

(i) which, if any, user nodes are docked with said instructing node as child nodes; which, if any, user nodes are docked with each of said child nodes as grandchild nodes of said instructing node; which, if any, user nodes are docked with

each of said grandchild nodes as great-grandchild nodes of said instructing node; and so on; and

(ii) said docked user nodes' respective bandwidth capacities, including a designation of whether said respective docked user node's bandwidth capacity is below a predetermined threshold (e.g., said respective docked user node is a low-bandwidth node) or at least said predetermined threshold (e.g., said respective docked user node is a high-bandwidth node);

(c) forming a primary recommended parent node list ("PRPL") comprised of the instructing node (if it has available capacity to transmit content data to said connection requesting user node) plus those docked user nodes having apparent available capacity to transmit content data to said connection requesting user node, with said instructing node being placed first on the PRPL (if it is on the PRPL) and said docked user nodes having apparent available capacity to transmit content date to said connection requesting user node being ranked with those docked nodes which are closer (i.e., in terms of network topology) to the instructing node being ranked higher than those docked nodes which are further away (i.e., in terms of network topology) and with equidistant (i.e., in terms of network topology) docked user nodes being ranked such that those docked nodes which have the higher utility ratings being ranked higher than docked nodes having lower utility ratings;

(d) forming a secondary recommended parent node list ("SRPL") comprised of the instructing node (if it has no available capacity to transmit content data to said connection requesting user node but does have at least one low-bandwidth node docked directly to it) plus said docked user nodes which (i) are high-bandwidth nodes with no available capacity to transmit content data to said connection requesting user node and (ii) have at least one low-bandwidth node docked directly to it, with said instructing node being placed first on the SRPL (if it is on the SRPL) and said docked user nodes on the SRPL being ranked with those docked nodes which are closer (i.e., in terms of network topology) to the instructing node being ranked higher than those docked nodes which are further away (i.e., in terms of network topology) and with equidistant (i.e., in terms of network topology) docked user nodes being ranked such that those docked nodes which have the higher utility ratings being ranked higher than docked nodes having lower utility ratings;

(e) determining whether the connection requesting user node is a low-bandwidth node;

(f) if the connection requesting user node is a low-bandwidth node:

(i) selecting the highest ranked node on the PRPL as a recommended parent node; and

(ii) providing a connection address list to said connection requesting user node, said connection address list listing each node in said instructing node's topology database from said recommended parent node back to said instructing node and, if said instructing node is not a primary server node, back to said primary server node; and

(g) if the connection requesting user node is a high-bandwidth node:

(i) selecting the highest ranked node on the PRPL as a recommended parent node;

(ii) selecting the highest ranked node on the SRPL as an alternate recommended parent node;

(iii) determining whether the alternate recommended parent node is closer (i.e., in terms of network topology) to the server node than the recommended parent node;

(iv) if the alternate recommended parent node is not closer (i.e., in terms of network topology) to the server node than the recommended parent node, providing a connection address list to said connection requesting user node, said connection address list listing each node in said instructing node's topology database from said recommended parent node back to said instructing node and, if said instructing node is not the primary server node, back to said primary server node;

(v) if the alternate recommended parent node is closer (i.e., in terms of network topology) to the server node than the recommended parent node, (1) forcing the disconnection of a low-bandwidth node from the alternate recommended parent node and (2) providing a connection address list to said connection requesting user node, said connection address list listing each node in said instructing node's topology database from said alternate recommended parent node back to said instructing node and, if said instructing node is not the primary server node, back to said primary server node.

In one example, step (f)(ii) may include the following steps:

(1) adding said connection requesting user node to the topology database as a child of the recommended parent node; and

(2) if said recommended parent node would have no apparent available capacity to transmit content data to an additional node with said connection requesting user node docked with it, deleting the recommended parent node from the PRPL and adding it to the SRPL.

In another example, step (g)(iv) may include the following steps:

(1) adding said connection requesting user node to the topology database as a child of the recommended parent node;

(2) if said recommended parent node would have no apparent available capacity to transmit content data to an additional node with said connection requesting user node docked with it, deleting the recommended parent node from the PRPL; and

(3) if said recommended parent node is deleted from the PRPL but has at least one low-bandwidth node docked directly to it, adding the recommended parent node to the SRPL.

In another example, step (g)(v) may include the following steps:

(3) adding said connection requesting user node to the topology database as a child of the alternate recommended parent node; and

4) if said alternate recommended parent node would have no apparent available capacity to transmit content data to an additional node with said connection requesting user node docked with it, deleting the alternate recommended parent node from the SRPL.

In another example, the process may further comprise the following steps:

(h) having said connection requesting user node go to (or attempt to go to) the node at the top of said connection address list;

(i) having said connection requesting user node determine whether the node at the top of said connection address list is part of the distribution network;

(j) if the node at top of connection address list is not part of the distribution network, deleting such node from connection address list and repeating steps (h) and (i) with respect to the next node at the top of connection address list; and

(k) if the node at top of connection address list is part of said distribution network, having said connection requesting user node dock with said node at the top of connection address list.

In another example, the process may further comprise the following steps:

(h) having said connection requesting user node go to (or attempt to go to) the node at the top of connection address list;

(i) having said connection requesting user node determine whether the node at the top of said connection address list is part of the distribution network;

(j) if the node at top of connection address list is not part of the distribution network, deleting such node from connection address list and repeating steps (h) and (i) with respect to the next node at the top of connection address list;

(k) if the node at the top of connection address list is part of said distribution network, determining whether said node at the top of connection address list has available capacity to transmit content data to said connection requesting user node; and

(l) if the node at the top of said connection address list is part of said distribution network and has available capacity to transmit content data to said connection requesting user node, having said connection requesting user node dock with the node at the top of said connection address list.

In another example, the process may further comprise the following steps:

(m) if the node at the top of said connection address list is part of said distribution network and the node at the top of said connection address list does not have available capacity to transmit content data to said connection requesting user node, repeating steps (a)-(g), with the node at the top of said connection address list being said instructing node and a new connection address list being the product of steps (a)-(g).

In another example, the process may further comprise the steps of counting the times that step (m) is performed and, when step (m) is performed a predetermined number of times, having said connection requesting user node repeat steps (a)-(g) with said primary server being the instructing node.

In another embodiment a process for docking a connection requesting user node with a distribution network is provided, said process including the following steps:

(a) having an instructing node receive a connection request from said connection requesting user node;

(b) determining whether said connection requesting user node's bandwidth capacity is below a predetermined threshold (e.g., said connection requesting user node is a low-bandwidth node) or at least said predetermined threshold (e.g., said connection requesting user node is a high-bandwidth node);

(c) forming an instructing node topology database indicating, at a point in time:

(i) which, if any, user nodes are docked with said instructing node as child nodes; which, if any, user nodes are docked with each of said child nodes as grandchild nodes of said instructing node; which, if any, user nodes are docked with each of said grandchild nodes as great-grandchild nodes of said server node; and so on; and

(ii) said docked user nodes' respective bandwidth capacities;

(d) if said connection requesting user node is a high-bandwidth user node, selecting from a group of nodes including the instructing node and said docked user nodes a recommended parent node for said connection requesting user node, wherein said recommended parent node has apparent available capacity to transmit content data to said connection requesting user node and is at least as close (i.e., in terms of network topology) to said instructing node as any other docked user node having apparent available capacity to transmit content data to said connection requesting user node;

(e) if said connection requesting user node is a low-bandwidth user node, selecting from a group of nodes including the instructing node and said docked user nodes a recommended parent node for said connection requesting user node, wherein said recommended parent node has apparent available capacity to transmit content data to said connection requesting user node and is at least as far (i.e., in terms of network topology) from said instructing node as any other docked user node having apparent available capacity to transmit content data to said connection requesting user node; and

(f) providing a connection address list to said connection requesting user node, said connection address list listing each node in said instructing node's topology database from said recommended parent node back to said instructing node.

In one example, step (f) may further comprise the following step: (i) if said instructing node is not a server node, including in said connection address list a list of nodes cascadingly connected with each other from the instructing node back to said server node.

In another embodiment a process for connecting a connection requesting user node to a computer information distribution network having a primary server node and user nodes docked therewith in a cascaded relationship is provided, said process further comprising the following steps:

(a) providing said connection requesting user node with a connection address list which sets forth a list of user nodes docked in series with each other back to said primary server;

(b) having said connection requesting user node go to (or attempt to go to) the node at top of said connection address list;

(c) having said connection requesting user node determine whether the node at the top of said connection address list is part of the distribution network;

(d) if the node at the top of said connection address list is not part of the distribution network, deleting such node from said connection address list and repeating steps (b) and (c) with respect to the next node at the top of connection address list; and

(e) if the node at the top of said connection address list is part of said distribution network, having said connection requesting user node dock with the node at the top of said connection address list.

In another embodiment a process for connecting a connection requesting user node to a computer information distribution network having a primary server node and user nodes docked therewith in a cascaded relationship is provided, said process further comprising the following steps:

(a) providing said connection requesting user node with a connection address list which sets forth a list of user nodes docked in series with each other back to said primary server;

(b) having said connection requesting user node go to (or attempt to go to) the node at the top of said connection address list;

(c) having said connection requesting user node determine whether the node at the top of said connection address list is part of the distribution network;

(d) if the node at the top of said connection address list is not part of the distribution network, deleting such node from connection address list and repeating steps (b) and (c) with respect to the next node at the top of said connection address list;

(e) if the node at the top of said connection address list is part of said distribution network, determining whether the node at the top of said connection address list has available capacity to transmit content data to said connection requesting user node; and

(f) if the node at the top of said connection address list is part of said distribution network and has available capacity to transmit content data to said connection requesting user node, having said connection requesting user node dock with the node at the top of said connection address list.

In another embodiment relating to a computer information distribution network comprising a primary server node and user nodes docked therewith in a cascaded relationship, wherein each user node may have no more than a predetermined maximum number of child nodes docked with it, a process for reconfiguring the network in the event of a user node's departure therefrom is provided, said process including the following steps:

(a) providing information at a point in time to a node in the distribution network as follows:

(i) the node's propagation rating if it is a user node having a parent node which is a user node, wherein a propagation rating is one of a predetermined number of grades ranging from highest to second highest to third highest and so on, with the number of grades being equal to said predetermined maximum number;

(ii) an ancestor list setting forth the node's ancestor nodes' addresses (if it has any ancestor nodes) back to the primary server node, with the node's parent node being atop the ancestor list;

(iii) a sibling list of the node's sibling nodes' addresses (if it has any sibling nodes) and their respective propagation ratings (of note, in another example (which example is intended to be illustrative and not restrictive), nodes may not have data about their sibling (i.e., nodes do not know who their sibling is); instead, the mechanism used to get a red node to connect to its green sibling may be for the parent of those two nodes (or a proxy for that parent) to send the red child: (1) a connection path with that red child's green sibling at the top of the path; and (2) a command to "priority join" to that green child - when the green child node accepts the incoming "priority join" connection request from its former red sibling, the green node will kick its own red child (if such a child exists) instructing that red child to join as a child of its own green sibling (these actions will cause a chain reaction of reconfigurations which will cascade until the edge of the tree is reached)); and

(iv) a child list of the node's child nodes' addresses (if it has any child nodes) and their respective propagation ratings;

(b) having a first node send a propagation signal to each former child node of a departed node, wherein said first node is an ancestor of said former child node of said departed node;

(c) with respect to each child node receiving the propagation signal which did not have the highest propagation rating prior to its receiving the propagation signal (hereinafter referred to as a "red node"), upon its receiving the propagation signal:

(i) setting the propagation rating of the red node to the next higher grade above the propagation rating of that red node before it received the propagation signal;

(ii) having the red node undock from its parent node (if it was docked with its parent node before it received the propagation signal); and

(iii) having the red node dock with its sibling node which had the highest propagation rating prior to the red node's receiving the propagation signal;

(d) with respect to each child node receiving the propagation signal which had the highest propagation rating prior to its receiving a propagation signal (hereinafter referred to as a "green node"), upon its receiving the propagation signal:

(i) setting the propagation rating of that green node to the lowest grade;

(ii) having the green node retransmit the propagation signal to its child nodes (if it has any); and

(iii) determining whether the green node is docked with the node from which the green node received the propagation signal and if it is, having the green node remain docked with said node from which the green node received the propagation signal, and if it is not, having the green node dock with said node from which the green node received the propagation signal; and

(e) repeating steps (c) and (d) with respect to each user node which receives a retransmitted propagation signal.

In another embodiment relating to a computer information distribution network comprising a primary server node and user nodes docked therewith in a cascaded relationship, wherein each user node may have no more than a predetermined maximum number of child nodes docked with it, a process for reconfiguring the network in the event of a user node's departure therefrom is provided, said process including the following steps:

(a) providing information at a point in time to a node in the distribution network as follows:

(i) the node's propagation rating if it is a user node having a parent node which is a user node, wherein a propagation rating is one of a predetermined number of grades ranging from highest to second highest to third highest and so on, with the number of grades being equal to said predetermined maximum number;

(ii) an ancestor list setting forth the node's ancestor nodes' addresses (if it has any ancestor nodes) back to the primary server node, with the node's parent node being atop the ancestor list;

(iii) a sibling list of the node's sibling nodes' addresses (if it has any sibling nodes) and their respective propagation ratings; and

(iv) a child list of the node's child nodes' addresses (if it has any child nodes) and their respective propagation ratings;

(b) having a first node send a propagation signal to each former child node of a departed node, wherein said first node is an ancestor of said former child node of said departed node;

(c) with respect to each child node receiving the propagation signal which did not have the highest propagation rating prior to its receiving the propagation signal (hereinafter referred to as a "red node"), upon its receiving the propagation signal:

(i) setting the propagation rating of the red node to the next higher grade above the propagation rating of that red node before it received the propagation signal;

(ii) having the red node undock from its parent node (if it was docked with its parent node before it received the propagation signal);

(iii) devising a first connection address list comprising the address of the red node's sibling node which had the had the highest propagation rating prior to the red node's receiving the propagation signal followed by the red node's ancestor list;

(iv) having said red node go to (or attempt to go to) the node at the top of said first connection address list;

(v) having said red node determine whether the node at the top of said first connection address list is part of the distribution network;

(vi) if the node at the top of said first connection address list is not part of the distribution network, deleting such node from said first connection address list and repeating steps (c)(iv) and (c)(v) with respect to the next node at the top of said first connection address list;

(vii) if the node at the top of said first connection address list is part of said distribution network, determining whether the node at the top of said first connection address list has available capacity to transmit content data to said red node; and

(viii) if the node at the top of said first connection address list is part of said distribution network and has available capacity to transmit content data to said red node, having said red node dock with the node at the top of said first connection address list (with regard to these steps (vii) and (viii), in another example (which example is intended to be illustrative and not restrictive), when a red node connects to a green node as part of a network reconfiguration event, it does not matter whether there is "available capacity" or not for the green node to accept its incoming red sibling – the green node has no choice but to accept the node (these actions are facilitated in this example by the red node issuing a "priority join" to the green node; if the green node does not have "available capacity" it has to kick its own red child to make room for the incoming green node));

(d) with respect to each child node receiving the propagation signal which had the highest propagation rating prior to its receiving a propagation signal (hereinafter referred to as a "green node"), upon its receiving the propagation signal:

(i) setting the propagation rating of that green node to the lowest grade;

(ii) having the green node retransmit the propagation signal to its child nodes (if it has any);

(iii) determining whether the green node is docked with the node from which the green node received the propagation signal and if it is, having the green node remain docked with said node from which the green node received the propagation signal, and if it is not, devising a second connection address list starting with said node from which the green node received the propagation signal and followed by that portion of said green node's ancestor list extending back to the primary server from said node from which the green node received the propagation signal;

(iv) having said green node go to (or attempt to go to) the node at the top of said second connection address list;

(v) having said green node determine whether the node at the top of said second connection address list is part of the distribution network;

(vi) if the node at the top of said second connection address list is not part of the distribution network, deleting such node from said second connection address list and repeating steps (d)(iv) and (d)(v) with respect to the next node at the top of said second connection address list;

(vii) if the node at the top of said second connection address list is part of said distribution network, determining whether said node at the top of said second connection address list has available capacity to transmit content data to said green node; and

(viii) if the node at the top of said second connection address list is part of said distribution network and has available capacity to transmit content data to said green node, having said green node dock with the node at the top of said second connection address list; and

(e) repeating steps (c) and (d) with respect to each user node which receives a retransmitted propagation signal.

In one example, step (c)(iii) may include the following additional step: (1) waiting a predetermined period of time so as to allow the red node's sibling with the highest

propagation rating prior to the red node's receipt of the propagation signal (hereinafter the "green node") to retransmit the propagation signal to the green node's child nodes (if any).

In another embodiment relating to a computer information distribution network comprising an instructing node having a fist user node docked with said instructing node to receive content data therefrom and a second user node docked with said first user node to receive content data therefrom, a process for handling a purported communication interruption between said first user node and said second user node is provided, said process including the following steps:

(a) having said instructing node receive a first complaint about said first user node from said second user node;

(b) if the second user node is the only user node docked with said first user node at a point in time, if said instructing node receives another complaint about the first user node from another user node within a predetermined period of time from when said instructing node received said first complaint or if said instructing node experiences a communication problem between it and said first user node during said predetermined period of time: (i) disconnecting said first user node from said instructing node; and

(c) if said instructing node has not experienced a communication problem between it and said first user node during said predetermined period of time and if a plurality of user nodes are docked with said first user node at said point in time and said instructing node has not received another complaint about said first user node from a user node other than said second user node within said predetermined period of time: (i) disconnecting said second user node from said first user node.

In one example, step (b) may include the following step: (ii) sending a signal to the user nodes docked with said first user node indicating the departure of said first user node from said network.

In another example, each user node may have no more than a predetermined maximum number of child nodes docked with it, and wherein step (b) may include the following steps

(ii) providing information at said point in time to each of said second user node, the other user nodes which had been docked with said first user node (if any) and at least one of any user nodes cascadingly connected in the distribution network under said first user node prior to said first user node's being disconnected from said instructing node (if any) as follows:

(1) the node's propagation rating, wherein a propagation rating is one of a predetermined number of grades ranging from highest to second highest to third highest and so on, with the number of grades being equal to said predetermined maximum number;

(2) an ancestor list setting forth the node's ancestor nodes' addresses (if it has any ancestor nodes) back to the instructing node, with the node's parent node being atop the ancestor list;

(3) a sibling list of the node's sibling nodes' addresses (if it has any sibling nodes) and their respective propagation ratings; and

(4) a child list of the node's child nodes' addresses (if it has any child nodes) and their respective propagation ratings;

(iii) having said instructing node send a propagation signal to each of said second user node and the other user nodes which had been docked with said first user node (if any);

(iv) with respect to each user node receiving the propagation signal which did not have the highest propagation rating prior to its receiving the propagation signal (hereinafter referred to as a "red node"), upon its receiving the propagation signal:

(1) setting the propagation rating of the red node to the next higher grade above the propagation rating of that red node before it received the propagation signal;

(2) having the red node undock from its parent node (if it was docked with its parent node before it received the propagation signal); and

(3) having the red node dock with its sibling node which had the highest propagation rating prior to the red node's receiving the propagation signal;

(v) with respect to each user node receiving the propagation signal which had the highest propagation rating prior to its receiving a propagation signal (hereinafter referred to as a "green node"), upon its receiving the propagation signal:

(1) setting the propagation rating of that green node to the lowest grade;

(2) having the green node retransmit the propagation signal to its child nodes (if it has any); and

(3) determining whether the green node is docked with the node from which the green node received the propagation signal and if it is, having the green node remain docked with said node from which the green node received the propagation signal, and if it is not, having the green node dock with said node from which the green node received the propagation signal; and

(vi) repeating steps (iv) and (v) with respect to each user node which receives a retransmitted propagation signal.

In another embodiment a distribution network for the distribution of content data from a server node to user nodes, wherein said user nodes are connected to said server and each other in cascaded relationship is provided, wherein:

(a) at least one of said user nodes is a repeater node connected directly to said server node, wherein said repeater node retransmits content data received by it to a user node docked to it for purpose of receiving content data from said repeater node (hereinafter referred to as a "child node"); and

(b) wherein each repeater node has the ability to provide to a user node which is attempting to dock with said repeater node connection address instructions.

In one example, each repeater node may include a descendant database indicating:

(i) which child nodes, if any, at a point in time, are docked with it so as to receive content data from said repeater node, and

(ii) which user nodes, if any, at said point in time, are purportedly docked with each of said child nodes.

In another example, the connection address instructions may refer said user node which is attempting to dock with said repeater node to a node in said descendant database.

In another example, each user node may include an ancestor database indicating to which node said user node is docked so that said user node may receive content data therefrom (hereinafter referred to as a "parent node"), and to which node, if any, at said point in time, said parent node is docked so that it may receive content data therefrom.

In another example, if said parent node of said user node departs from said distribution network, said user node may contact another node on its ancestor database.

In another example, each child node of a repeater node may include a sibling database indicating which user nodes, if any, are also child nodes of said repeater node.

In another embodiment a distribution network for the distribution of content data from a server node to user nodes is provided, wherein n levels of user nodes are cascadingly connected to said server node, wherein n is a number greater than one, wherein each user node includes a delay which causes the playing of content data by such node to be delayed a period of time (hereinafter "delay time") from a point in time, and wherein delays in higher level nodes create greater delay times than do delays in lower level nodes.

In another embodiment a distribution network for the distribution of content data from a server node to user nodes is provided, wherein n levels of user nodes are cascadingly

connected to said server node, wherein n is a number greater than one, wherein each user node includes a delay which causes the playing of content data by such node to be delayed a period of time (hereinafter "delay time") from a point in time, wherein x is a number from and including 2 to and including n, and wherein delays in (x-1) level nodes create greater delay times than do delays in x level nodes.

In one example, said point in time is an event experienced approximately simultaneously by substantially all user nodes.

In another example, said point in time is when said node receives content data.

While a number of embodiments and examples of the present invention have been described, it is understood that these embodiments and examples are illustrative only, and not restrictive, and that many modifications may become apparent to those of ordinary skill in the art. For example, any desired branch factor may be used (*see*, e.g., FIG. 46A showing a branch factor of 1 (i.e., each parent node may have only one child node); FIG. 46B showing a branch factor of 2 (i.e., each parent node may have one or two children nodes); and FIG. 46C showing a branch factor of 3 (i.e., each parent node may have one, two or three children nodes)). Of course, other branch factors (e.g., 4, 5, 6, etc) may be used and, as discussed above, the distribution network may be a hybrid network utilizing subtrees having different branch factors). Further still, a modified Utility Rating may be determined as follows: Utility Rating = Time (of connection) x Bandwidth Rating. Further still, a node ID may be based upon: (a) an IP address (external/internal/DHCP); (b) a URL; and/or (c) a Port. Further still, the root server may propagate "its" time upon assignment of a connection and the user node may get a "delta" (i.e., difference) from its clock to use (in the event of a reset the user node may re-acquire the time from the server (or from another time-providing source)). Further still, the various time intervals discussed herein may be predetermined time intervals (e.g., .5 sec., 1 sec, 1 min.) and/or the time intervals may be based upon some varying criteria (e.g., flow of data in the network, number of nodes in the network, number of levels in the network, position of a given node in the network). Further still, partial propagation of data (e.g., ancestor data and/or descendent data) from one or more nodes may be permitted. Further still (as discussed above), one or more of the steps described herein may be omitted (and the various steps which are performed may not necessarily need to be carried out in the order described herein (which description was intended to represent a number of examples, and not be restrictive)). Further still, the root server and each of the user nodes may utilize essentially identical software for carrying out the network configuration, reconfiguration and data transfer operations described herein or the root server may have dedicated server software

while the user nodes have different, dedicated user node software. Further still, the entire distribution network (e.g., the root server and all of the user nodes) may have the same branch factor (e.g., a branch factor of 2) or the user nodes may have a first branch factor (e.g., a branch factor of 2) and the root server may have a second branch factor (e.g., the root server may be capable of acting as a parent to more user nodes than 2). Further still, heuristics may be employed to drive network configuration, reconfiguration and/or data flow (e.g., based upon historic network configuration, reconfiguration and/or data flow).

Aylo Exhibit 1018
Page 1388 of 2308

CLAIMS

1. A method for propagating a network topology model of a binary tree topology network through the network, the network including a plurality of nodes, including a root node and at least one leaf node, operatively networked together to provide a communications path for data, each of the plurality of nodes except the root node being docked downtree of a parent node, the parent node being one of the plurality of nodes, and each of the plurality of nodes adapted to be docked uptree of a first child node and a second child node, each of the at least one leaf nodes having none of the plurality of nodes docked downtree of and therewith, the method comprising the steps of:

each of the at least one leaf nodes providing a propagate network topology model message containing the address of the providing leaf node to the parent node docked uptree of and with the providing leaf node;

for each of the plurality of nodes receiving the propagate network topology model message from each child node docked downtree of and therewith:

forming a network topology model representing one or more paths from the receiving node to each one of the plurality of nodes downtree of the receiving node;

storing in a memory accessible to the receiving node, the network topology model so formed;

providing to the parent node docked uptree of and with the receiving node a propagate network topology model message containing the network topology model so formed,

thereby propagating a network topology model through the network.

2. The method of claim 1, wherein the communications path for data comprises a downtree communications path.

3. The method of claim 1, wherein the communications path for data comprises a downtree communications path for data and an uptree communications path for data.

4. The method of claim 1, wherein the step of each of the at least one leaf nodes providing a propagate network topology model message is carried out a plurality of times.

5. The method of claim 1, wherein the step of each of the at least one leaf nodes providing a propagate network topology model message is carried out periodically.

6. The method of claim 1, wherein a first node being one of the plurality of nodes, but not a leaf node, receives a register message from a second node that is not part of the network, and wherein the first node is docked uptree of a first child node and a second child node, each of the first child node and the second child node being one of the plurality of nodes, the method further comprising the steps of:

searching the network topology model stored in the memory of the first node for a path to an open node, the open node being one of the plurality of nodes that the network topology model shows as not being docked uptree of and with two child nodes;

sending a path message to the second node containing at least an address of the open node.

7. The method of claim 6, wherein the path message comprises a path from the open node to the first node.

8. The method of claim 6, wherein the searching step comprising finding the open node such that a connection to the open node keeps the binary topology network balanced.

9. The method of claim 6, wherein step of searching further comprises the steps of:

searching consecutively further down hierarchical levels of the binary tree represented in the network topology model stored in the memory of the first node:

first to locate a path representing a node not being docked uptree of and with any child nodes, and if no such path is located,

then to locate a path representing a node not being docked uptree of and with two child nodes, and

repeating the searching step until one such path is located; and

designating the located path as a path to the open node.

10. A method for propagating a network topology model of a binary tree topology network through the network, the network including a plurality of nodes, including a root node and at least one leaf node, operatively networked together to provide a communications path for data, each of the plurality of nodes except the root node being docked downtree of a parent node, the parent node being one of the plurality of nodes, and each of the plurality of nodes adapted to be docked uptree of a first child node and a second child node, each of the at least one leaf nodes having none of the plurality of nodes docked downtree of and therewith, the method comprising the steps of:

providing a propagate path message from the root node to each of the plurality of nodes that are docked downtree of and with the root node, the propagate path message indicating an address of the root node;

for each of the plurality of nodes receiving the propagate path message from the root node or from another one of the plurality of nodes:

storing in a memory accessible to the receiving node, the path to the root node stored in the propagate path message;

if the receiving node has at least one of the plurality of nodes docked downtree of and therewith, adding to the propagate path message an address of the receiving node; and

providing the propagate path message to each of the plurality of nodes docked downtree of and with the receiving node, the propagate path message a path to the root node;

thereby propagating a root path of a binary tree topology network through the network.

11. The method of claim 10, wherein the communications path for data comprises a downtree communications path for data.

12. The method of claim 10, wherein the communications path for data comprises a downtree communications path for data and an uptree communications path for data.

13. The method of claim 10, wherein the step of providing a propagate path message is carried out a plurality of times.

14. The method of claim 10, wherein the step of providing a propagate path message is carried out periodically.

15. A method for propagating a network topology model of a binary tree topology network through the network, the network including a plurality of nodes, including a root node and at least one leaf node, operatively networked together to provide a communications path for data, each of the plurality of nodes except the root node being docked downtree of a parent node, the parent node being one of the plurality of nodes, and each of the plurality of nodes adapted to be docked uptree of a first child node and a second child node, each of the at least one leaf nodes having none of the plurality of nodes docked downtree of and therewith, the method comprising the steps of:

each of the at least one leaf nodes, providing a propagate network topology model message containing the address of the providing leaf node to the parent node docked uptree of and with the providing leaf node;

for each of the plurality of nodes receiving the propagate network topology model message from each child node docked downtree of and therewith:

forming a network topology model comprising a path from the receiving node to each one of the plurality of nodes downtree of the receiving node;

storing in a memory accessible to the receiving node, the network topology model so formed;

providing to the parent node docked uptree of and with the receiving node a propagate network topology model message containing the network topology model so formed,

providing a propagate path message from the root node to each of the plurality of nodes that are docked downtree of and with the root node, the propagate path message indicating an address of the root node;

for each of the plurality of nodes receiving propagate path message from the root node or from another one of the plurality of nodes:

storing in a memory accessible to the receiving node, the path to the root node stored in the propagate path message;

if the receiving node has at least one of the plurality of nodes docked downtree of and therewith, adding to the propagate path message an address of the receiving node; and

providing the propagate path message to each of the plurality of nodes docked downtree of and with the root node.

16. The method of claim 15, wherein the communications path for data comprises a downtree communications path for data.

17. The method of claim 15, wherein the communications path for data comprises a downtree communications path for data and an uptree communications path for data.

18. The method of claim 15, wherein the step of providing a propagate network topology model message is carried out a plurality of times.

19. The method of claim 15, wherein the step of providing a propagate network topology model message is carried out periodically.

20. The method of claim 15, wherein the step of providing a propagate path message is carried out a plurality of times.

21. The method of claim 15, wherein the step of providing a propagate path message is carried out periodically.

Aylo Exhibit 1018
Page 1393 of 2308

22. The method of claim 15, wherein a first node being one of the plurality of nodes, but not a leaf node, receives a register message from a second node that is not part of the network, and wherein the first node is docked uptree of a first child node and a second child node, each of the first child node and the second child node being one of the plurality of nodes, the method further comprising the steps of:

searching the network topology model stored in the memory of the first node for a path to an open node, the open node being one of the plurality of nodes that the network topology model shows as not being docked uptree of and with two child nodes;

sending a path message to the second node containing at least an address of the open node.

23. The method of claim 22, wherein the path message comprises a path from the open node to the first node.

24. The method of claim 22, wherein the path message comprises a path from the open node to the root node.

25. The method of claim 22, wherein the searching step comprising finding the open node such that a connection to the open node keeps the binary topology network balanced.

26. The method of claim 22, wherein step of searching further comprises the steps of:

searching consecutively further down hierarchical levels of the binary tree represented in the network topology model stored in the memory of the first node:

first to locate a path representing a node not being docked uptree of and with any child nodes, and if no such path is located,

then to locate a path representing a node not being docked uptree of and with two child nodes, and

repeating the searching step until one such path is located; and

designating the located path as a path to the open node.

27. A method for propagating a network topology model of a binary tree topology network through the network, the network including a plurality of nodes, including a root node and at least one leaf node, operatively networked together to provide a communications path for data, each of the plurality of nodes except the root node being docked downtree of a parent node, the parent node being one of the plurality of nodes, and each of the plurality of nodes adapted to be docked uptree of a first child node and a second child node, each of the at least one leaf nodes having none of the plurality of nodes docked downtree of and therewith, the method comprising the steps of:

providing a banned list message from the root node to each of the plurality of nodes that are docked downtree of and with the root node, the banned list message comprising an identifier for each of a plurality of banned nodes;

for each of the plurality of nodes receiving the banned list message from the root node or from another one of the plurality of nodes:

storing in a memory accessible to the receiving node, the identifiers for each of the plurality of banned nodes; and

providing the banned list message to each of the plurality of nodes docked downtree of and with the receiving node;

thereby propagating the list of banned nodes through the network.

28. The method of claim 27, wherein the communications path for data comprises a downtree communications path for data.

29. The method of claim 27, wherein the communications path for data comprises a downtree communications path for data and an uptree communications path for data.

30. The method of claim27, further comprising the step of rejecting a register message from a new node that is not part of the network if the node is identified on the banned list.

31. A method for propagating a network topology model of a binary tree topology network through the network, the network including a plurality of nodes, including a root node and at least one leaf node, operatively networked together to provide a communications path for data, each of the plurality of nodes except the root node being docked downtree of a parent node, the parent node being one of the plurality of nodes, and each of the plurality of nodes adapted to be docked uptree of a first child node and a second child node, each of the at least one leaf nodes having none of the plurality of nodes docked downtree of and therewith, the method comprising the steps of:

the first node sending a request firewall status message over a first channel to one of the plurality of nodes that is docked with the first node;

the one of the plurality of nodes receiving the request firewall status message sending at least one response message to the first node over a separate channel; and

the first node determining its firewall status based upon whether it receives the response message.

32. The method of claim 31, wherein the communications path for data comprises a downtree communications path for data.

33. The method of claim 31, wherein the communications path for data comprises a downtree communications path for data and an uptree communications path for data.

34. The method of claim 31, wherein the step of sending at least one response message comprises the steps of:

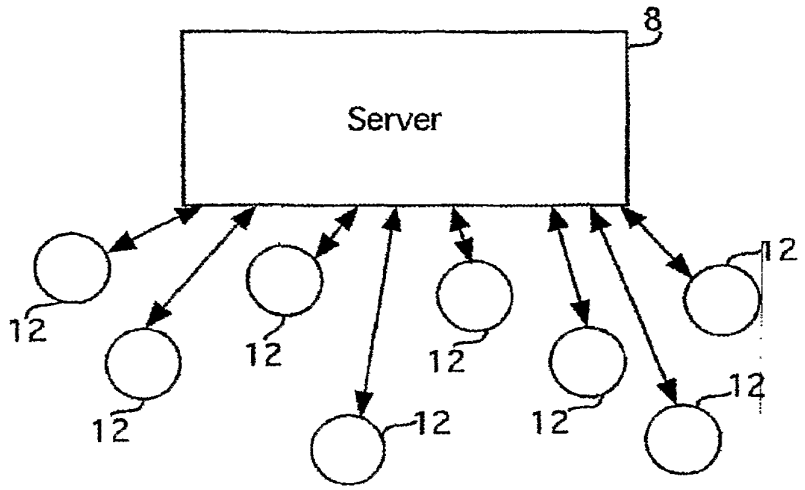
sending a first response message to the first node over a separate channel;

waiting a predefined period for an acknowledgement of the first response message, and if no acknowledgement is received, sending a second response message to the first node over the separate channel;

waiting a predefined period for an acknowledgement of the second response message, and if no acknowledgement is received, sending a third response message to the first node over the separate channel; and

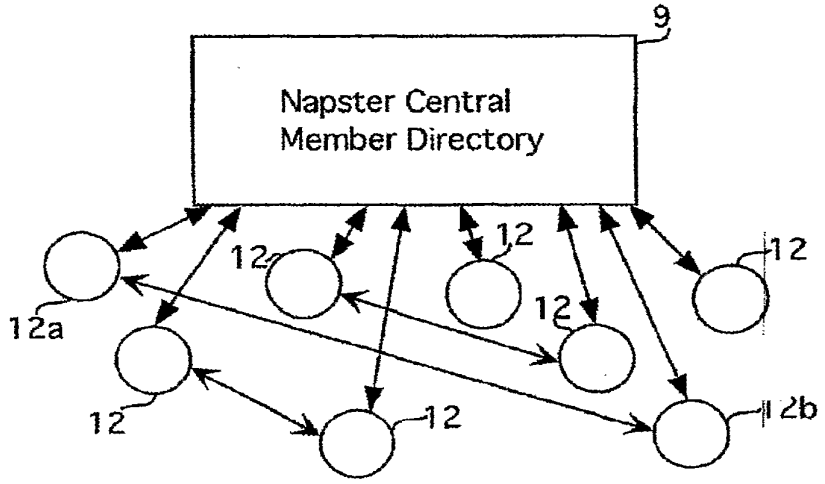
waiting a predefined period for an acknowledgement of the third response message, and if no acknowledgement is received, sending a firewall status on message to the first node over the first channel.

Aylo Exhibit 1018
Page 1397 of 2308



PRIOR ART

Fig. 1



PRIOR ART

Fig. 2

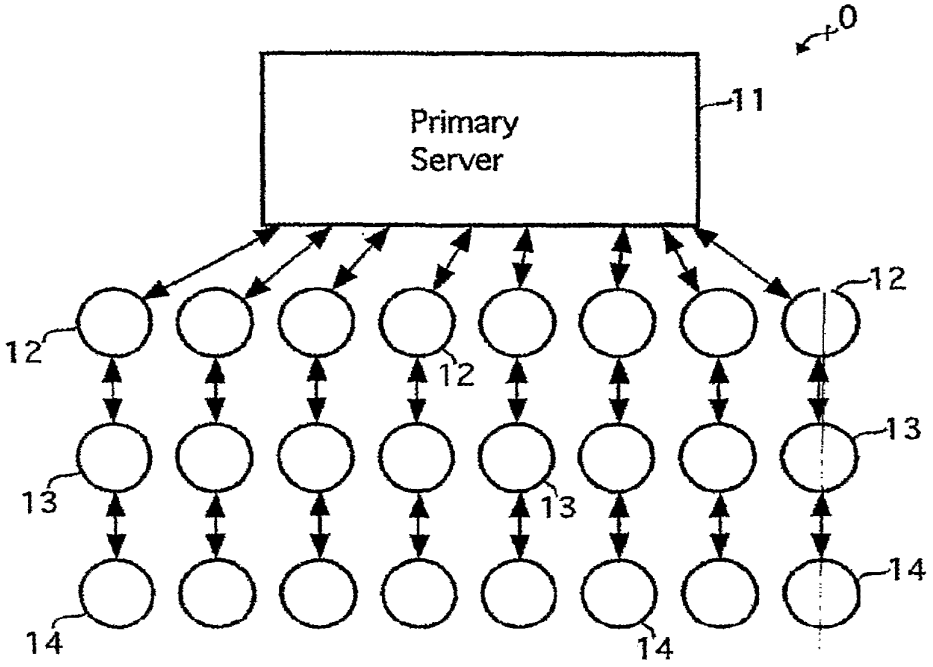


Fig. 3

"Start" [illegible]

[illegible]

Corresponds to [illegible] application [illegible]

31

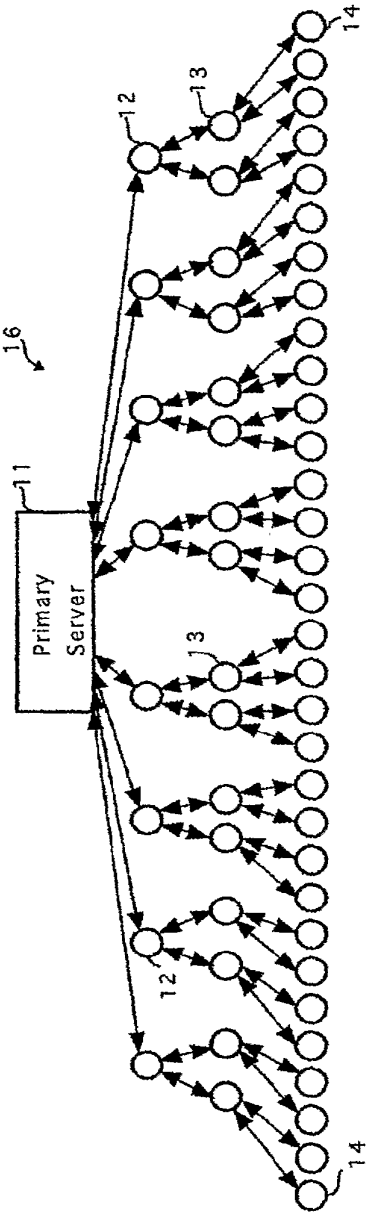


Fig. 4

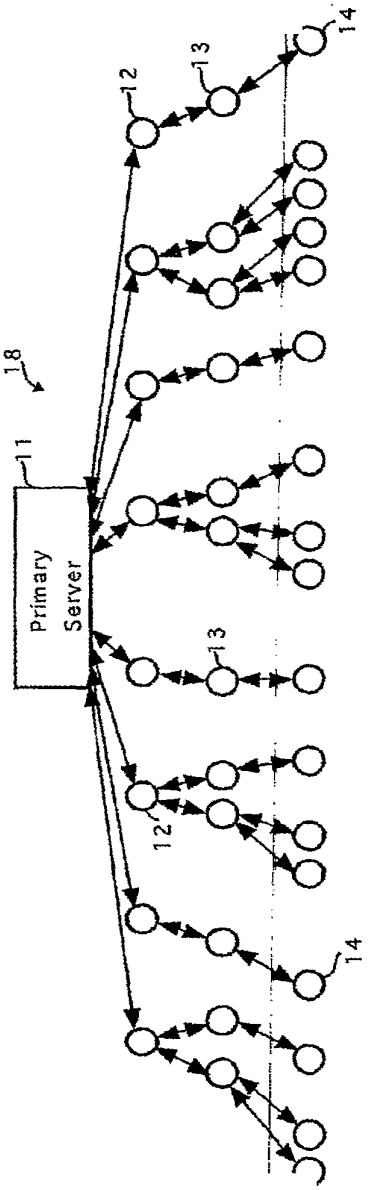


Fig. 5

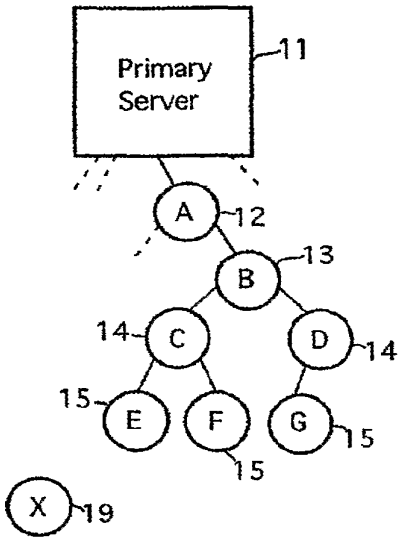


Fig. 6

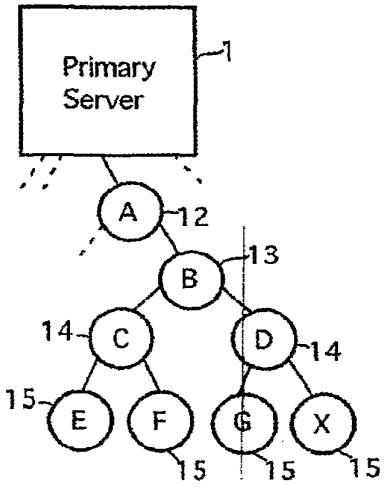


Fig. 7

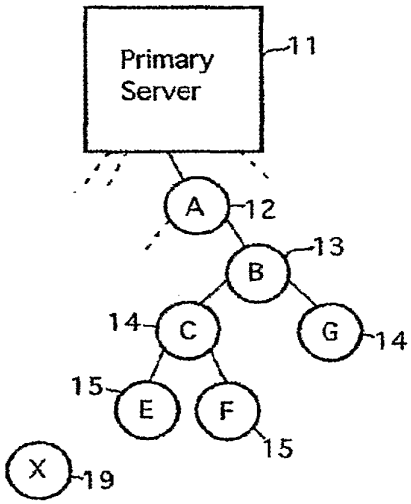


Fig. 8

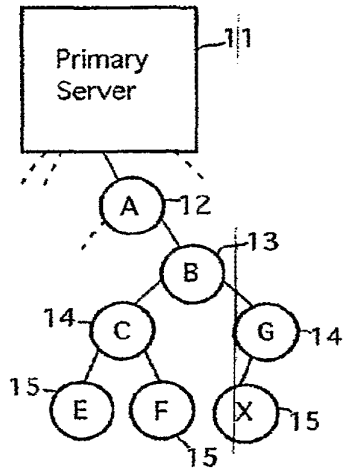


Fig. 9

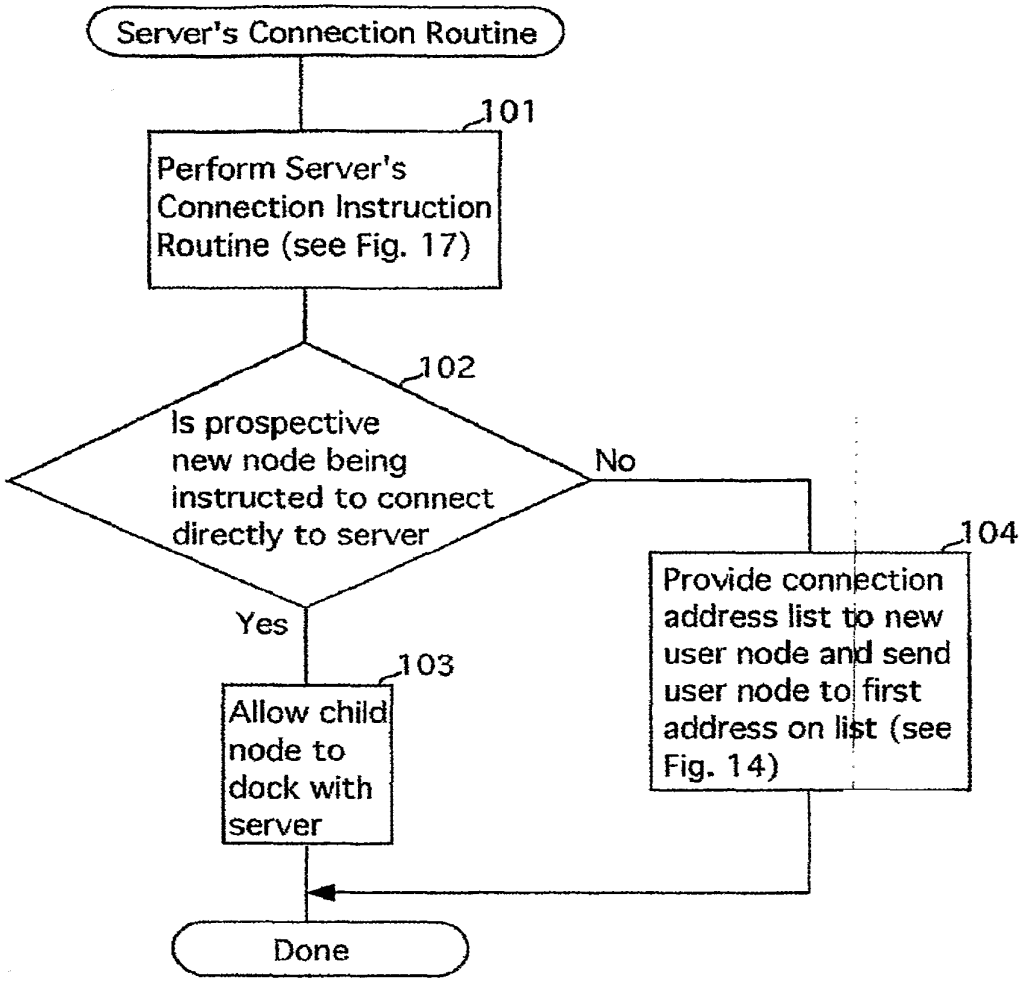

Server's Connection Routine
101
Perform Server's Connection Instruction Routine (see Fig. 17)
102
Is prospective new node being instructed to connect directly to server
No
Yes
103
Allow child node to dock with server
104
Provide connection address list to new user node and send user node to first address on list (see Fig. 14)
Done

Fig. 10

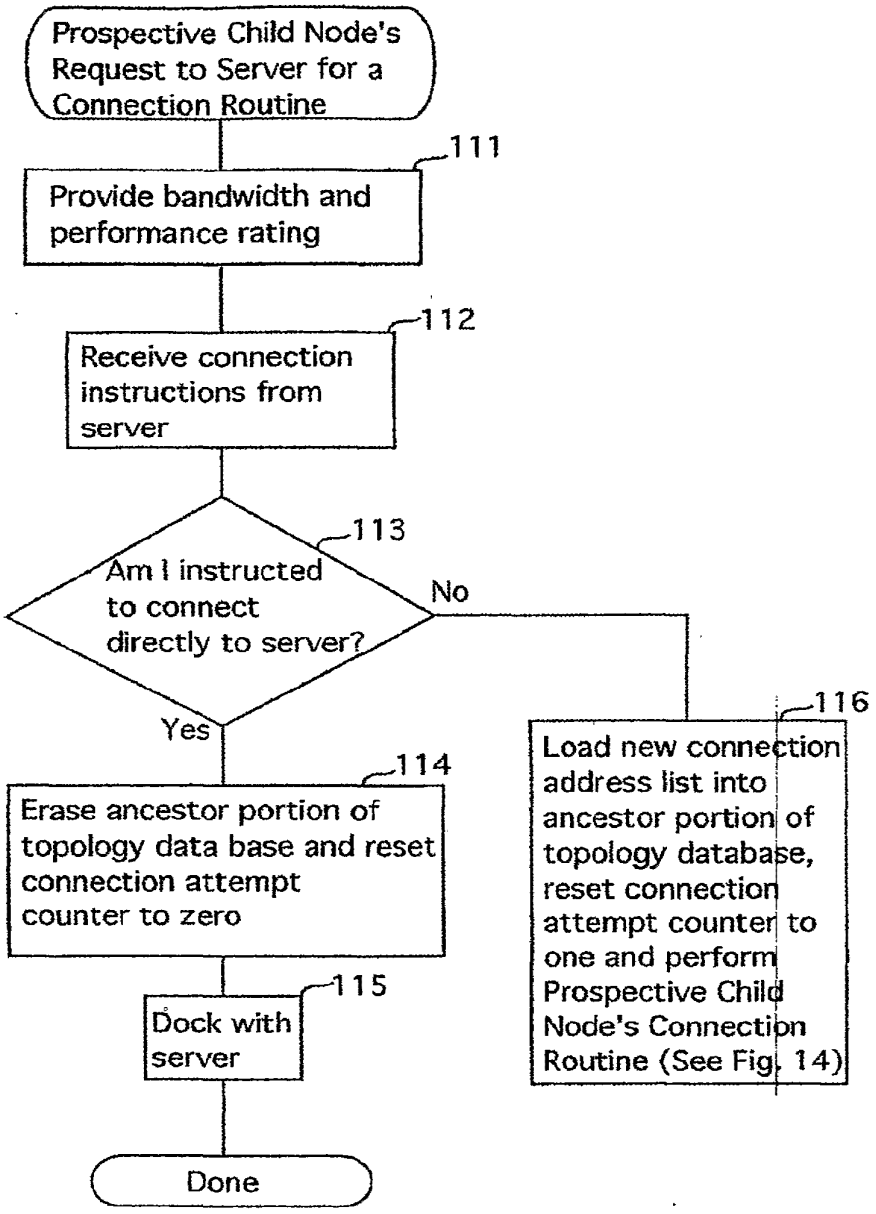

Prospective Child Node's Request to Server for a Connection Routine
111
Provide bandwidth and performance rating
112
Receive connection instructions from server
113
Am I instructed to connect directly to server?
No
Yes
114
Erase ancestor portion of topology data base and reset connection attempt counter to zero
115
Dock with server
Done
116
Load new connection address list into ancestor portion of topology database, reset connection attempt counter to one and perform Prospective Child Node's Connection Routine (See Fig. 14)

Fig. 11

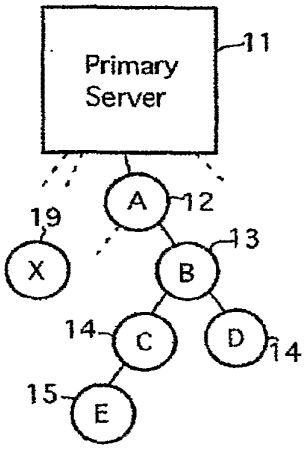


Fig. 12A

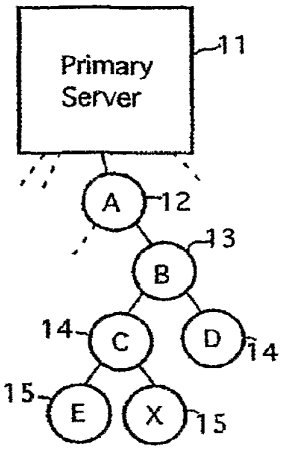


Fig. 12B

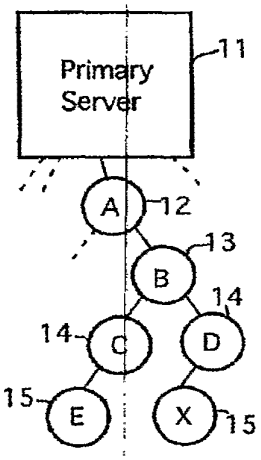


Fig. 12C

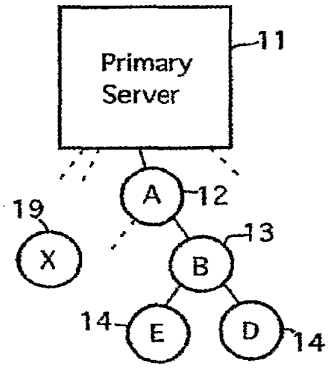


Fig. 12D

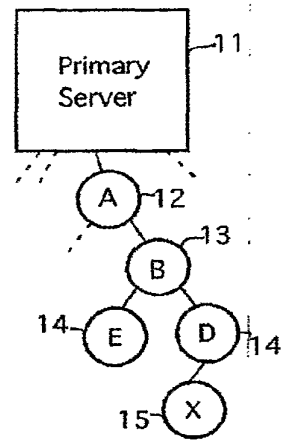


Fig. 12E

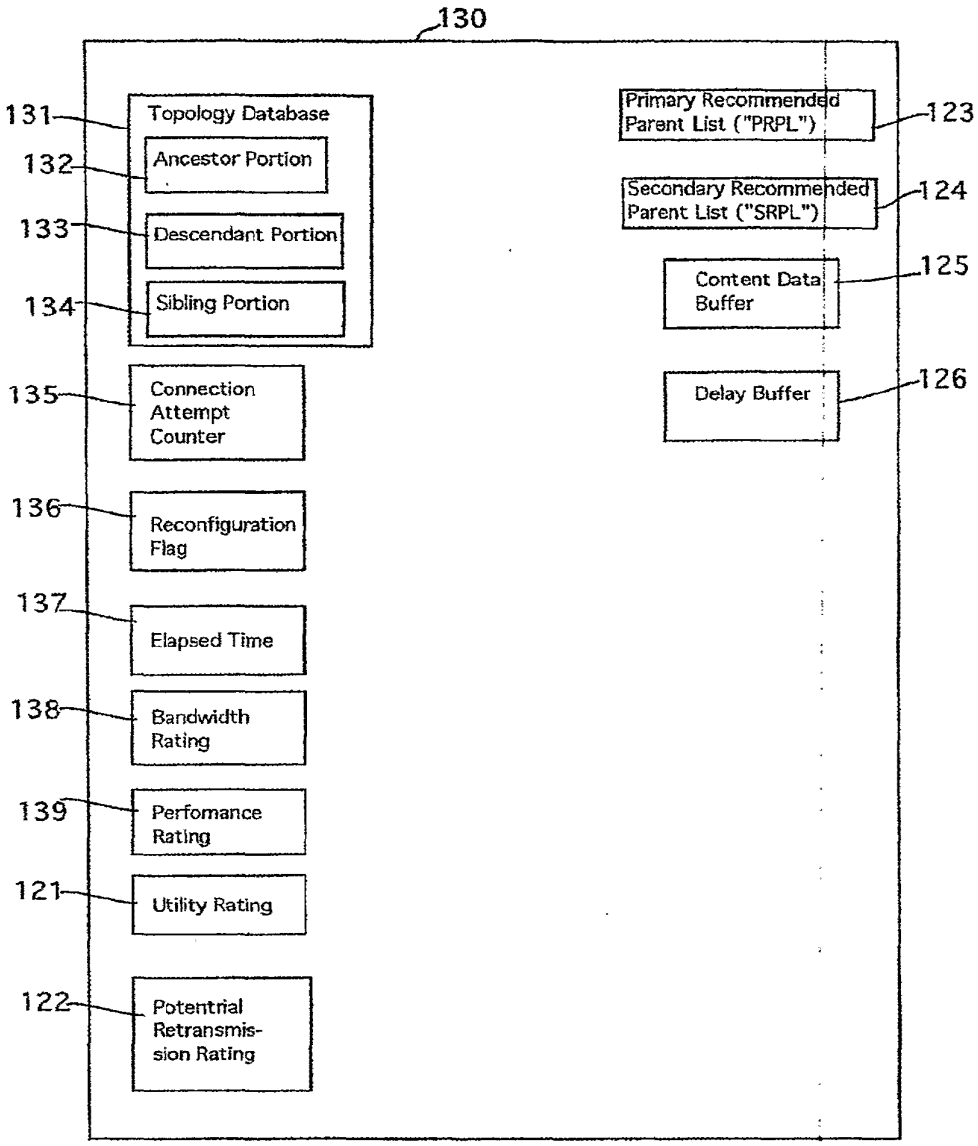

130
131
Topology Database
132
Ancestor Portion
133
Descendant Portion
134
Sibling Portion
135
Connection Attempt Counter
136
Reconfiguration Flag
137
Elapsed Time
138
Bandwidth Rating
139
Perfomance Rating
121
Utility Rating
122
Potentrial Retransmis-sion Rating
Primary Recommended Parent List ("PRPL")
123
Secondary Recommended Parent List ("SRPL")
124
Content Data Buffer
125
Delay Buffer
126

Fig. 13

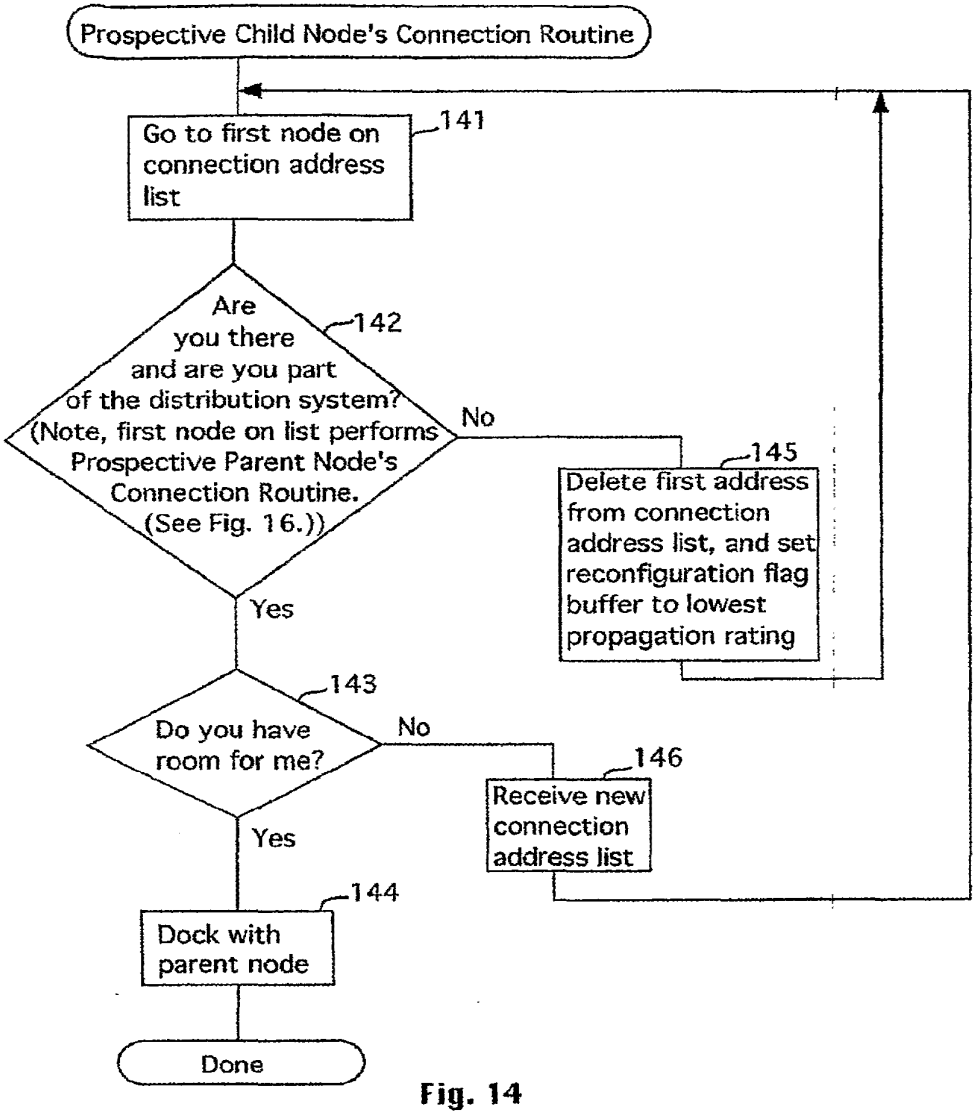

Prospective Child Node's Connection Routine
Go to first node on connection address list
141
Are you there and are you part of the distribution system? (Note, first node on list performs Prospective Parent Node's Connection Routine. (See Fig. 16.))
142
No
145
Delete first address from connection address list, and set reconfiguration flag buffer to lowest propagation rating
Yes
Do you have room for me?
143
No
146
Receive new connection address list
Yes
Dock with parent node
144
Done

Fig. 14

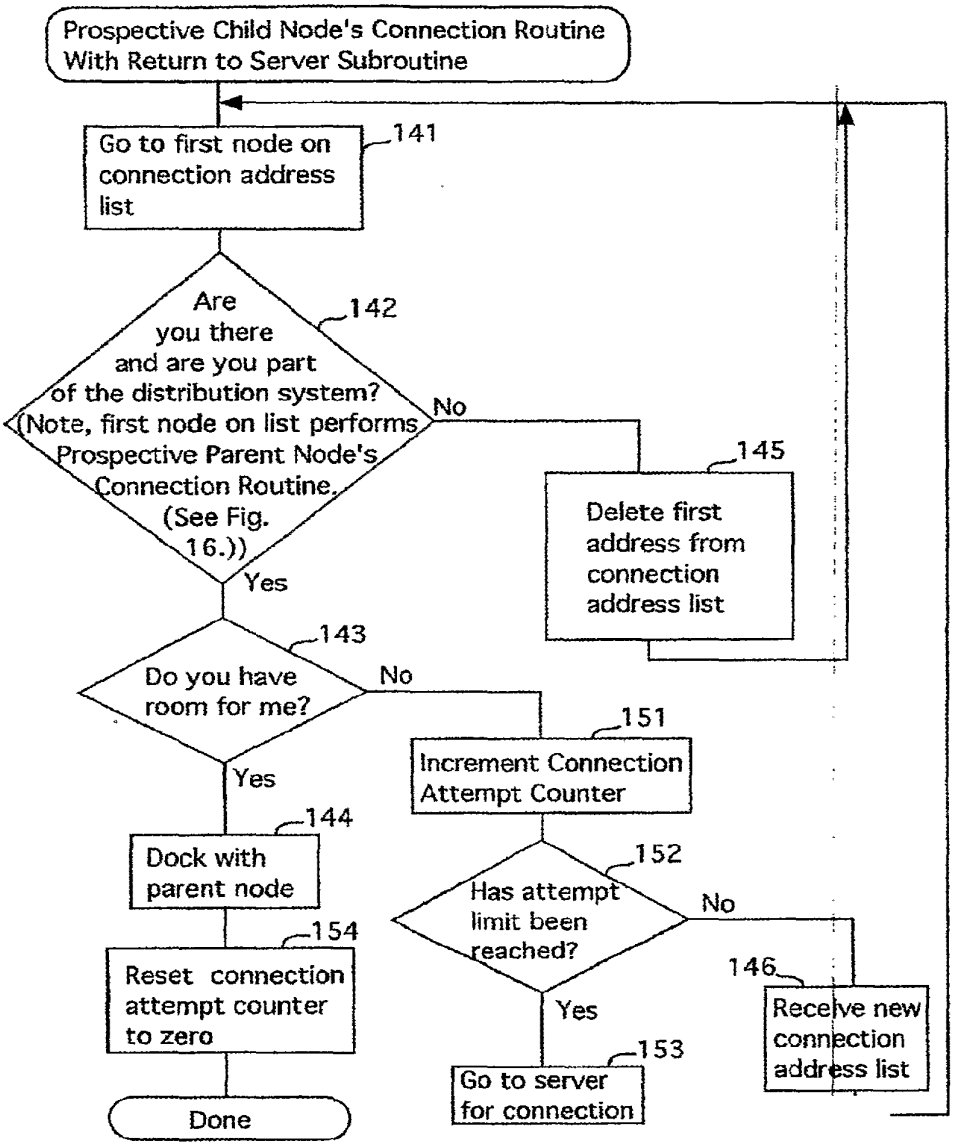

Prospective Child Node's Connection Routine With Return to Server Subroutine
Go to first node on connection address list
141
Are you there and are you part of the distribution system? (Note, first node on list performs Prospective Parent Node's Connection Routine. (See Fig. 16.))
142
No
Yes
Delete first address from connection address list
145
Do you have room for me?
143
No
Yes
Increment Connection Attempt Counter
151
Dock with parent node
144
Has attempt limit been reached?
152
No
Yes
Reset connection attempt counter to zero
154
Receive new connection address list
146
Go to server for connection
153
Done

Fig. 15

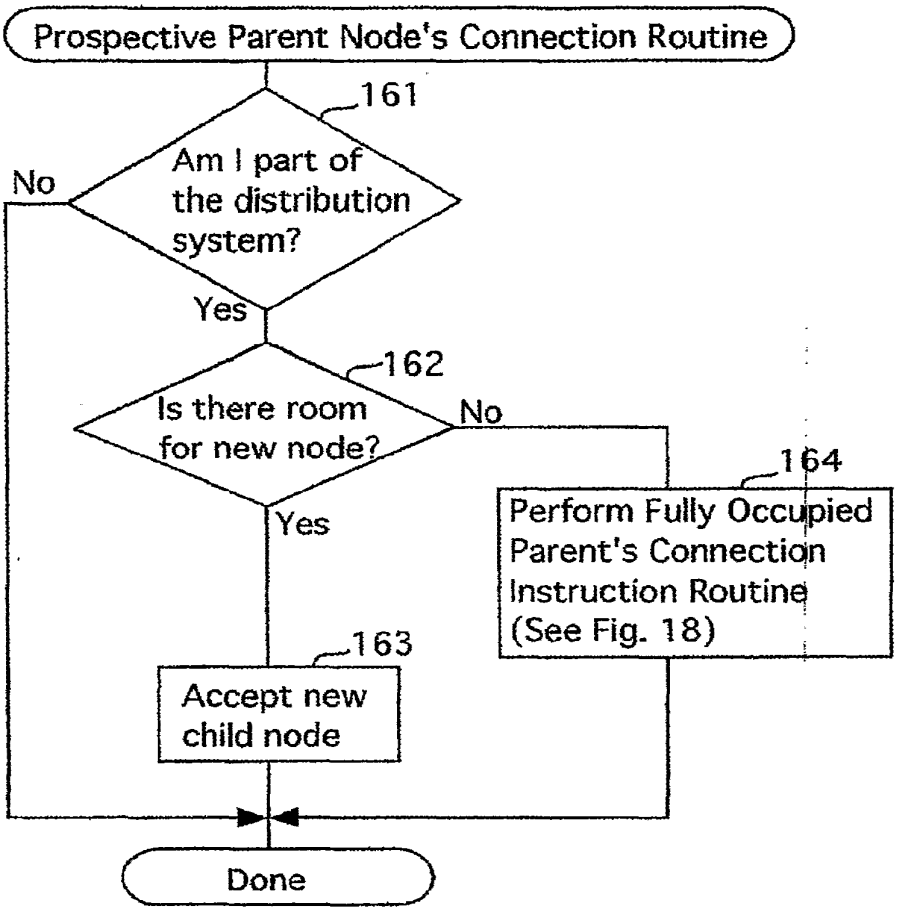

Prospective Parent Node's Connection Routine
161
Am I part of the distribution system?
No
Yes
162
Is there room for new node?
No
Yes
164
Perform Fully Occupied Parent's Connection Instruction Routine (See Fig. 18)
163
Accept new child node
Done

Fig. 16

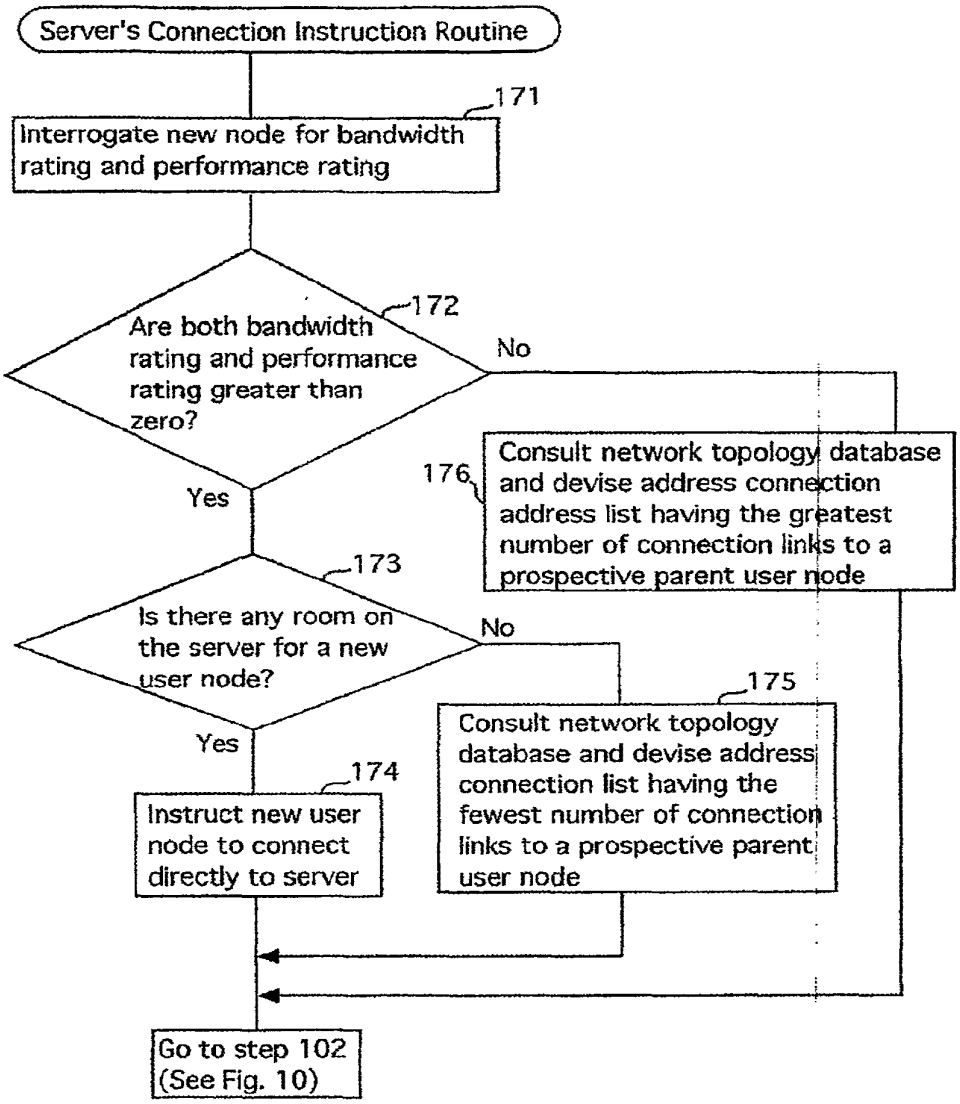

Server's Connection Instruction Routine
171
Interrogate new node for bandwidth rating and performance rating
172
Are both bandwidth rating and performance rating greater than zero?
No
Yes
176
Consult network topology database and devise address connection address list having the greatest number of connection links to a prospective parent user node
173
Is there any room on the server for a new user node?
No
Yes
175
Consult network topology database and devise address connection list having the fewest number of connection links to a prospective parent user node
174
Instruct new user node to connect directly to server
Go to step 102 (See Fig. 10)

Fig. 17

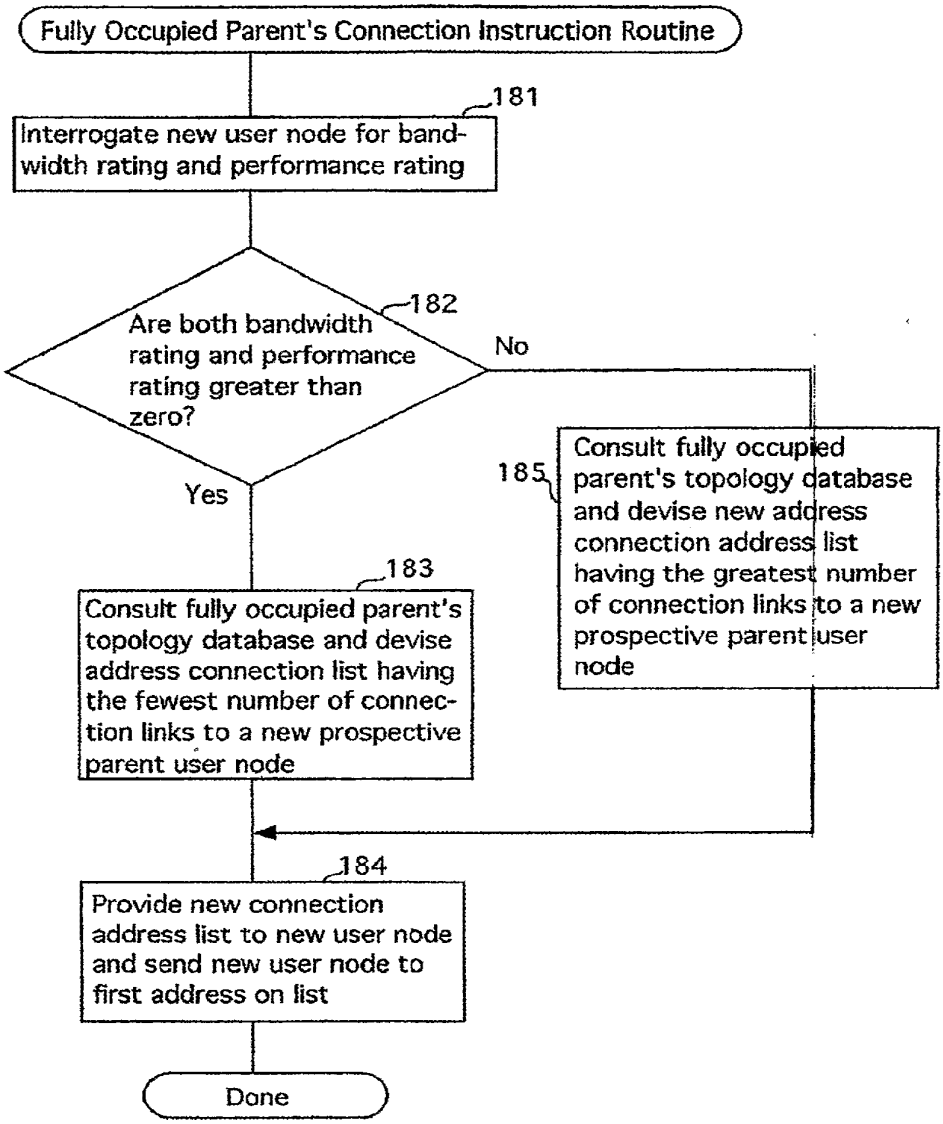

Fully Occupied Parent's Connection Instruction Routine
181
Interrogate new user node for bandwidth rating and performance rating
182
Are both bandwidth rating and performance rating greater than zero?
No
Yes
185
Consult fully occupied parent's topology database and devise new address connection address list having the greatest number of connection links to a new prospective parent user node
183
Consult fully occupied parent's topology database and devise address connection list having the fewest number of connection links to a new prospective parent user node
184
Provide new connection address list to new user node and send new user node to first address on list
Done

Fig. 18

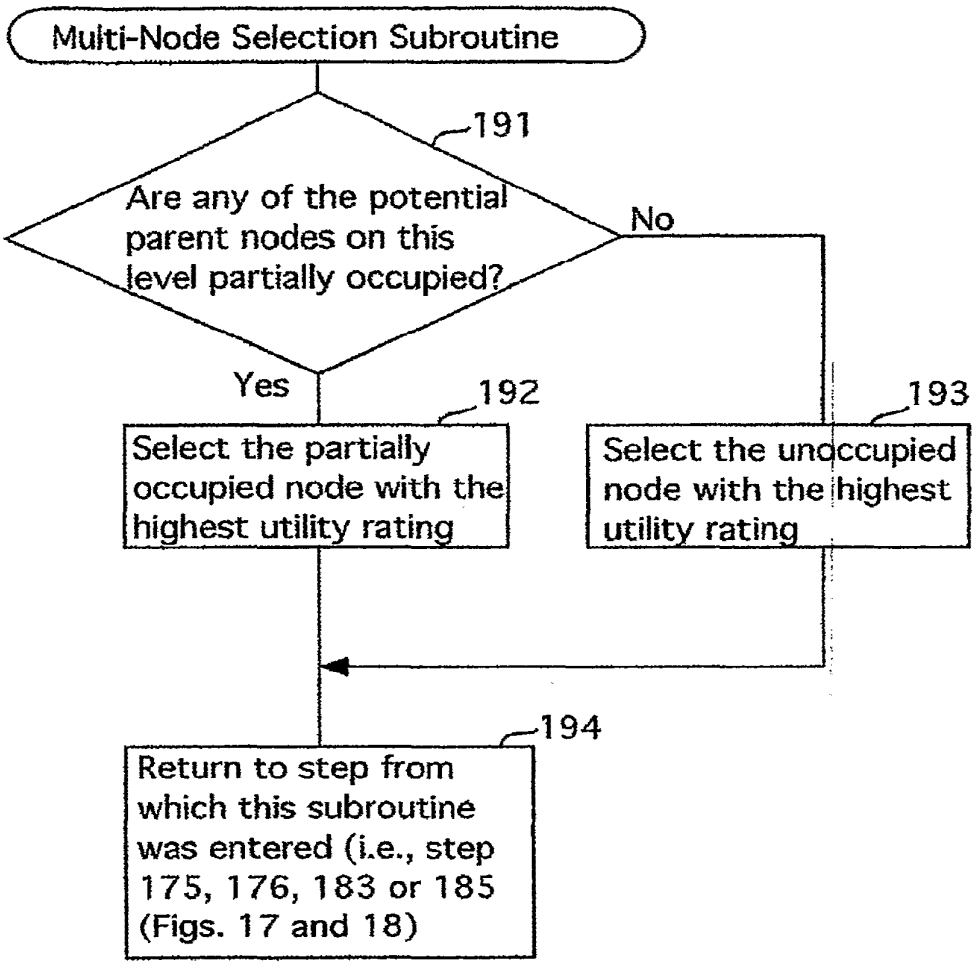

Multi-Node Selection Subroutine
191
Are any of the potential parent nodes on this level partially occupied?
No
Yes
192
Select the partially occupied node with the highest utility rating
193
Select the unoccupied node with the highest utility rating
194
Return to step from which this subroutine was entered (i.e., step 175, 176, 183 or 185 (Figs. 17 and 18)

Fig. 19

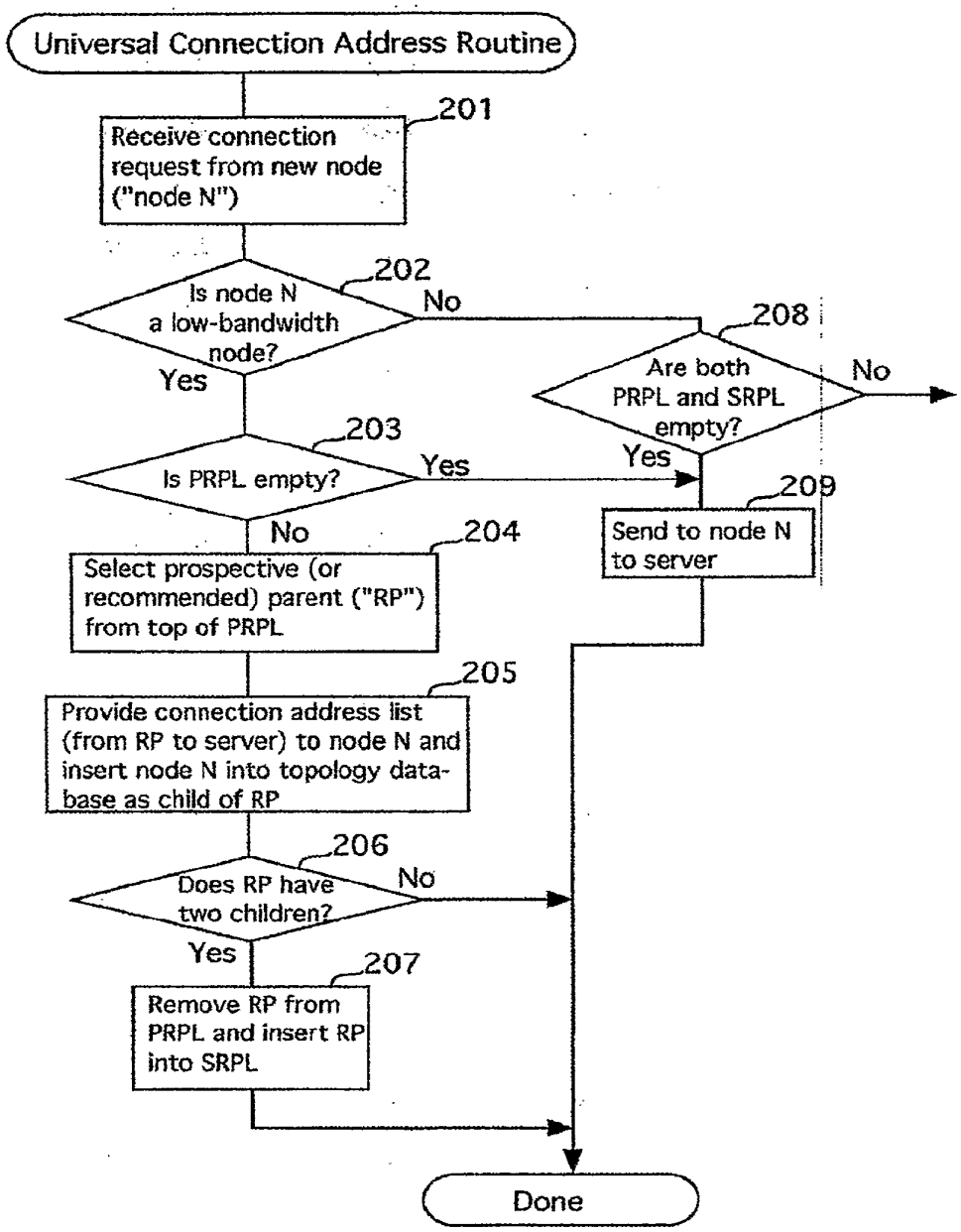

Universal Connection Address Routine
201
Receive connection request from new node ("node N")
202
Is node N a low-bandwidth node?
No
Yes
208
Are both PRPL and SRPL empty?
No
Yes
203
Is PRPL empty?
Yes
No
209
Send to node N to server
204
Select prospective (or recommended) parent ("RP") from top of PRPL
205
Provide connection address list (from RP to server) to node N and insert node N into topology data-base as child of RP
206
Does RP have two children?
No
Yes
207
Remove RP from PRPL and insert RP into SRPL
Done

Fig. 20A

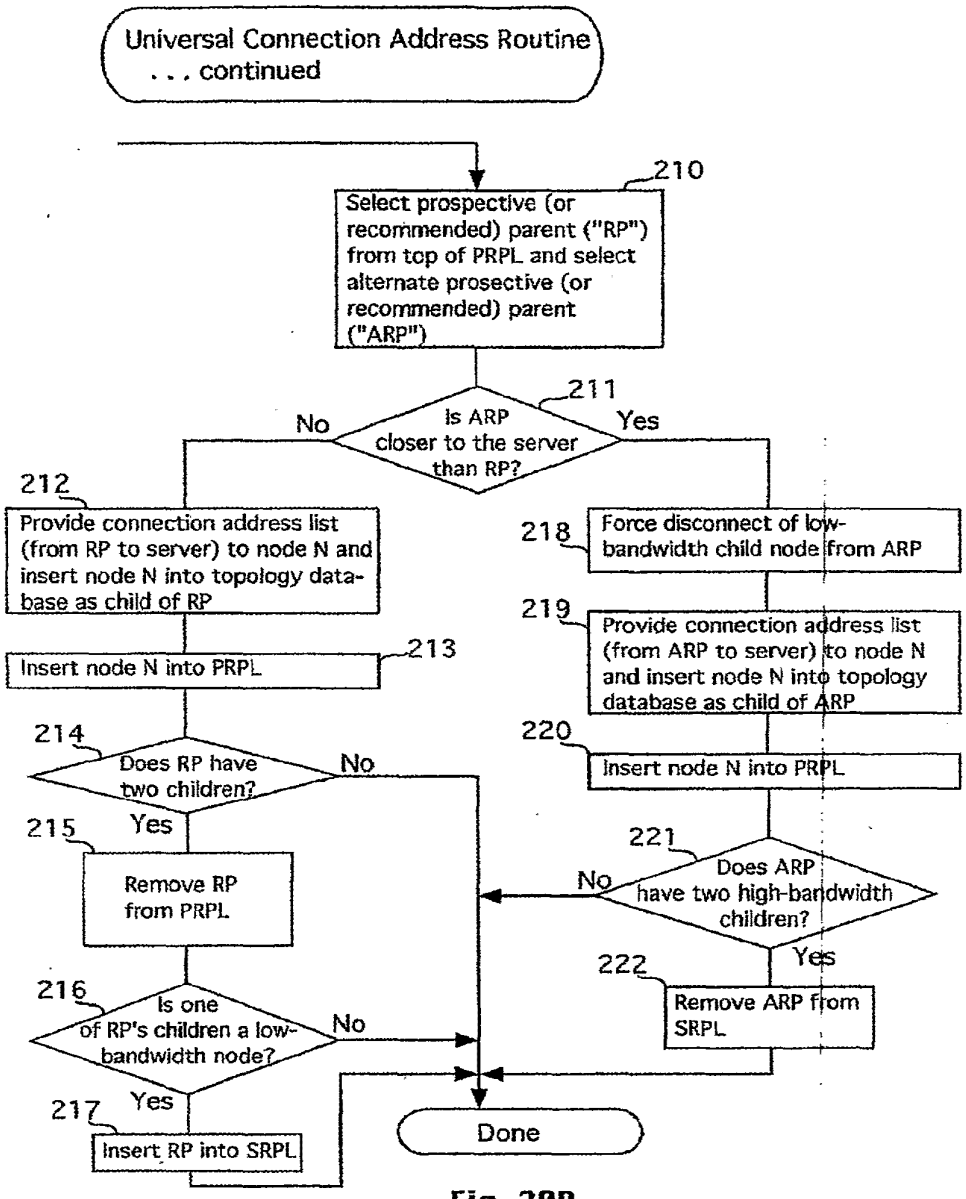

Universal Connection Address Routine
. . . continued
210
Select prospective (or recommended) parent ("RP") from top of PRPL and select alternate prosective (or recommended) parent ("ARP")
211
Is ARP closer to the server than RP?
No
Yes
212
Provide connection address list (from RP to server) to node N and insert node N into topology data-base as child of RP
213
Insert node N into PRPL
214
Does RP have two children?
No
Yes
215
Remove RP from PRPL
216
Is one of RP's children a low-bandwidth node?
No
Yes
217
Insert RP into SRPL
218
Force disconnect of low-bandwidth child node from ARP
219
Provide connection address list (from ARP to server) to node N and insert node N into topology database as child of ARP
220
Insert node N into PRPL
221
Does ARP have two high-bandwidth children?
No
Yes
222
Remove ARP from SRPL
Done

Fig. 20B

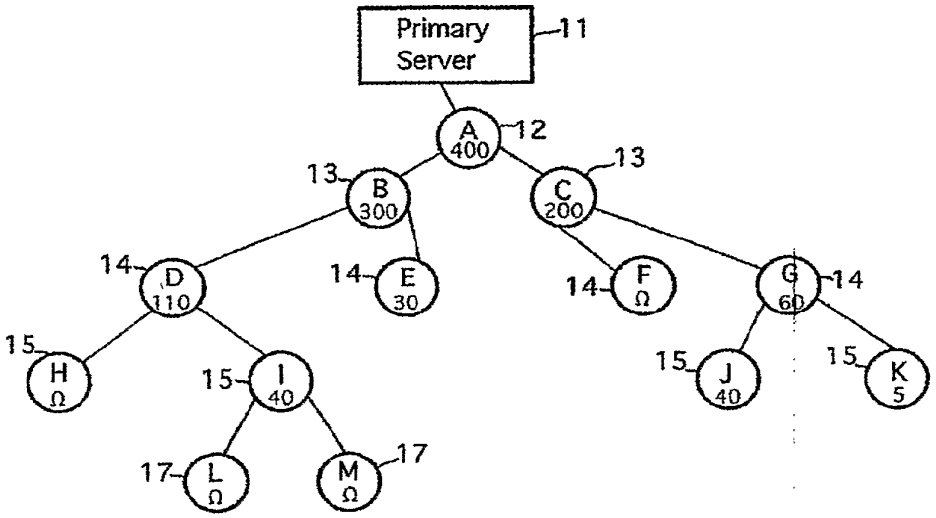

Primary Server
11
A
400
12
B
300
13
C
200
13
D
110
14
E
30
14
F
Ω
14
G
60
14
H
Ω
15
I
40
15
J
40
15
K
5
15
L
Ω
17
M
Ω
17

Fig. 21

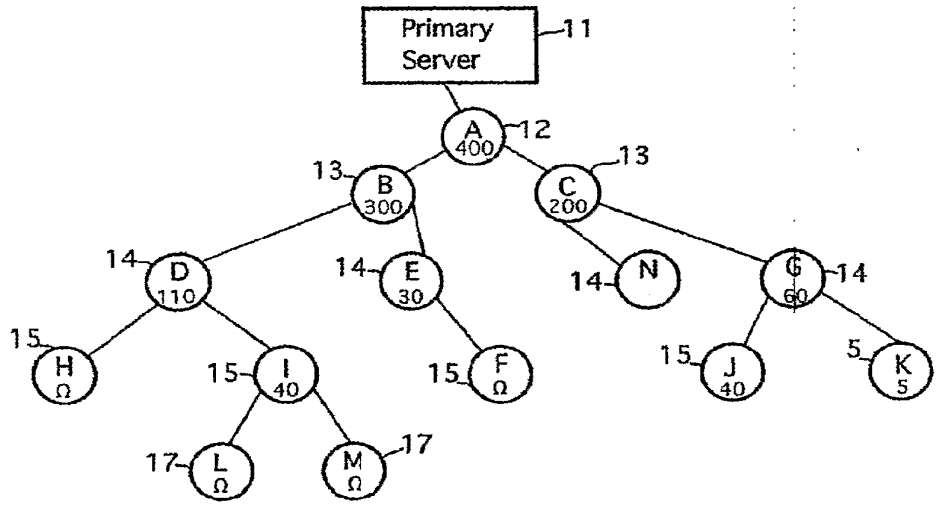

Primary Server
11
A
400
12
B
300
13
C
200
13
D
110
14
E
30
14
N
14
G
60
14
H
Ω
15
I
40
15
F
Ω
15
J
40
15
K
5
5
L
Ω
17
M
Ω
17

Fig. 22

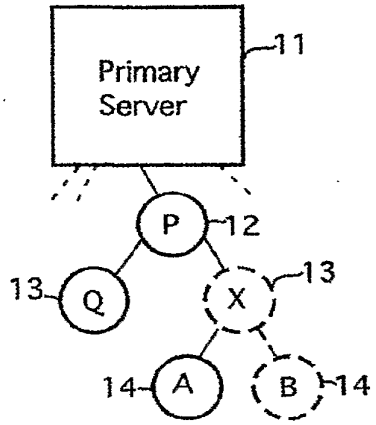


Fig. 23

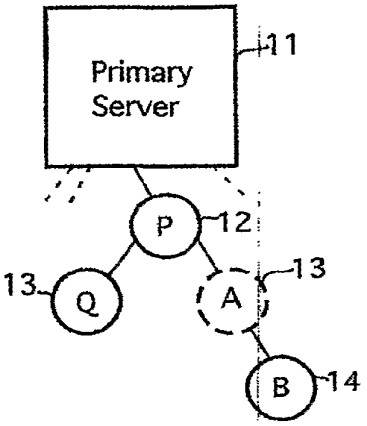


Fig. 24

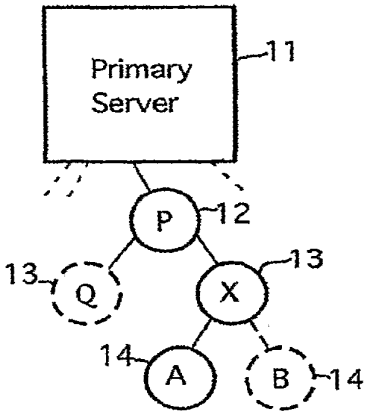


Fig. 25

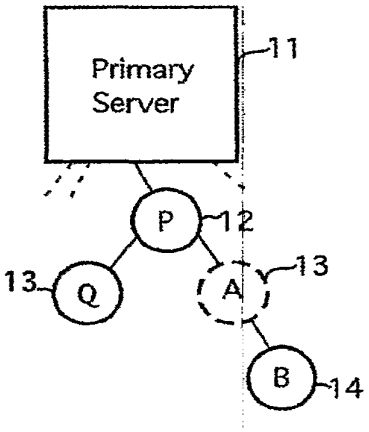


Fig. 26

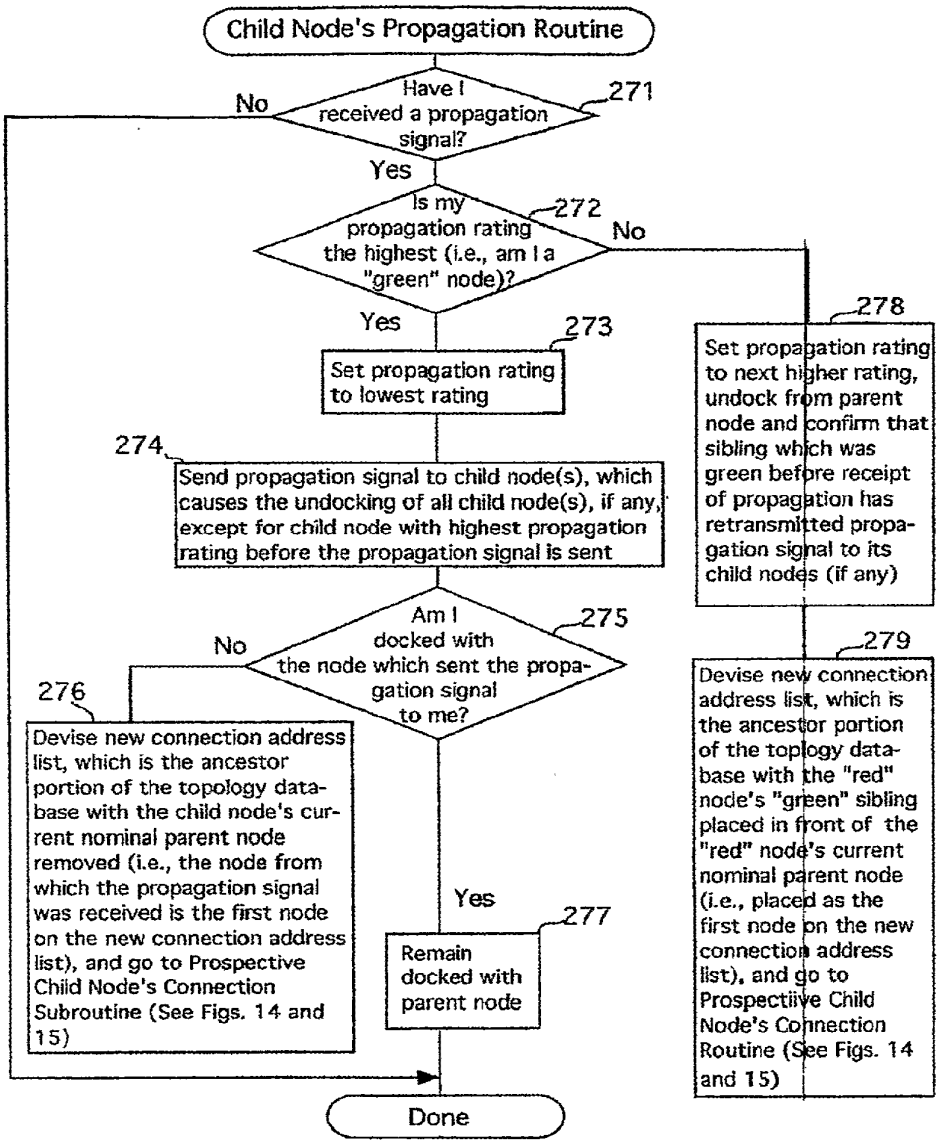

Child Node's Propagation Routine
Have I received a propagation signal?
271
No
Yes
Is my propagation rating the highest (i.e., am I a "green" node)?
272
No
Yes
273
Set propagation rating to lowest rating
274
Send propagation signal to child node(s), which causes the undocking of all child node(s), if any, except for child node with highest propagation rating before the propagation signal is sent
Am I docked with the node which sent the propagation signal to me?
275
No
Yes
276
Devise new connection address list, which is the ancestor portion of the topology database with the child node's current nominal parent node removed (i.e., the node from which the propagation signal was received is the first node on the new connection address list), and go to Prospective Child Node's Connection Subroutine (See Figs. 14 and 15)
277
Remain docked with parent node
278
Set propagation rating to next higher rating, undock from parent node and confirm that sibling which was green before receipt of propagation has retransmitted propagation signal to its child nodes (if any)
279
Devise new connection address list, which is the ancestor portion of the topology database with the "red" node's "green" sibling placed in front of the "red" node's current nominal parent node (i.e., placed as the first node on the new connection address list), and go to Prospective Child Node's Connection Routine (See Figs. 14 and 15)
Done

Fig. 27

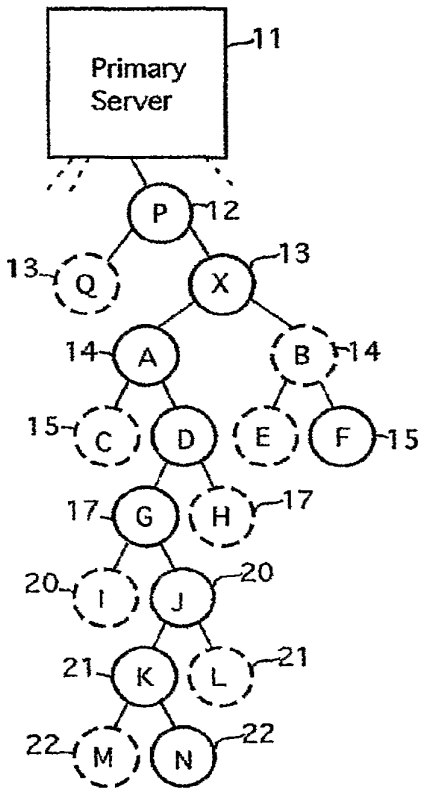

Primary Server
11
P
12
13
Q
X
13
14
A
B
14
15
C
D
E
F
15
17
G
H
17
20
I
J
20
21
K
L
21
22
M
N
22

Fig. 28

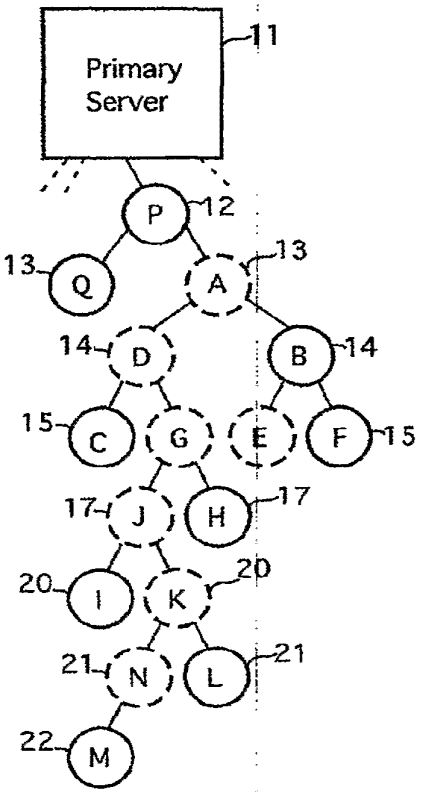

Primary Server
11
P
12
13
Q
A
13
14
D
B
14
15
C
G
E
F
15
17
J
H
17
20
I
K
20
21
N
L
21
22
M

Fig. 29

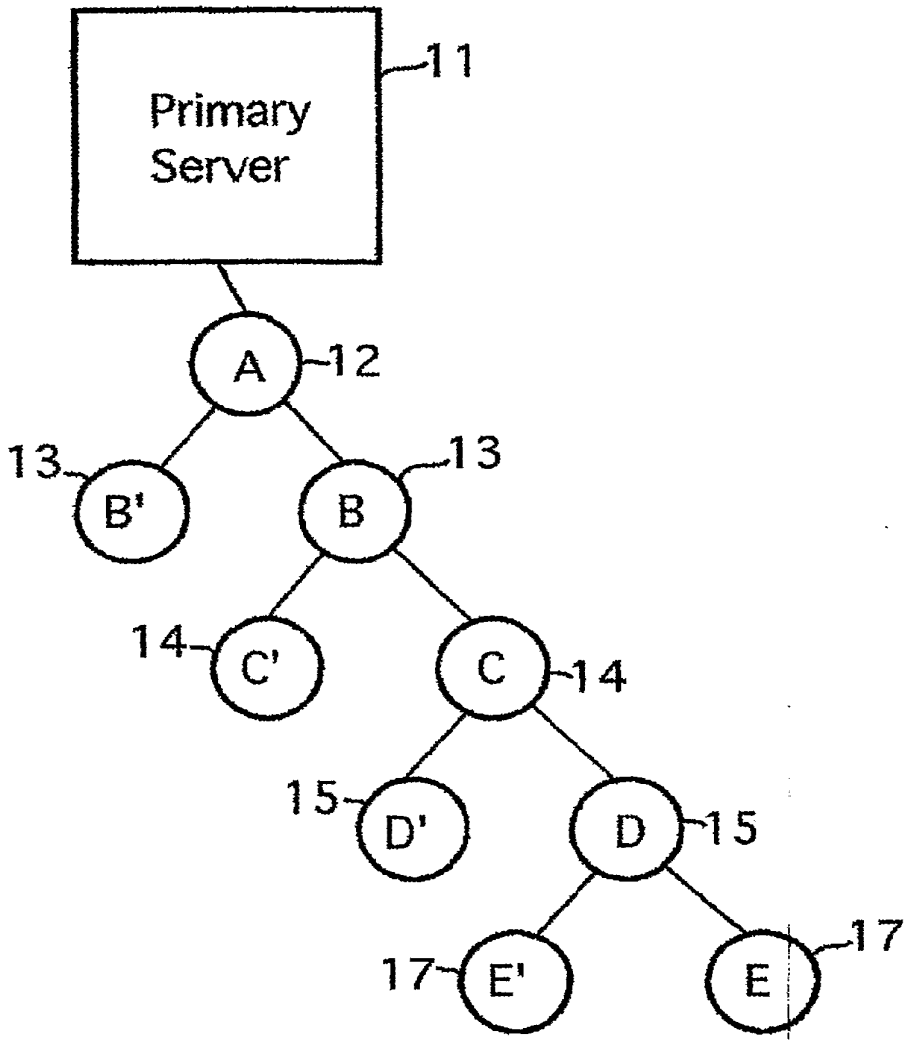


Fig. 30

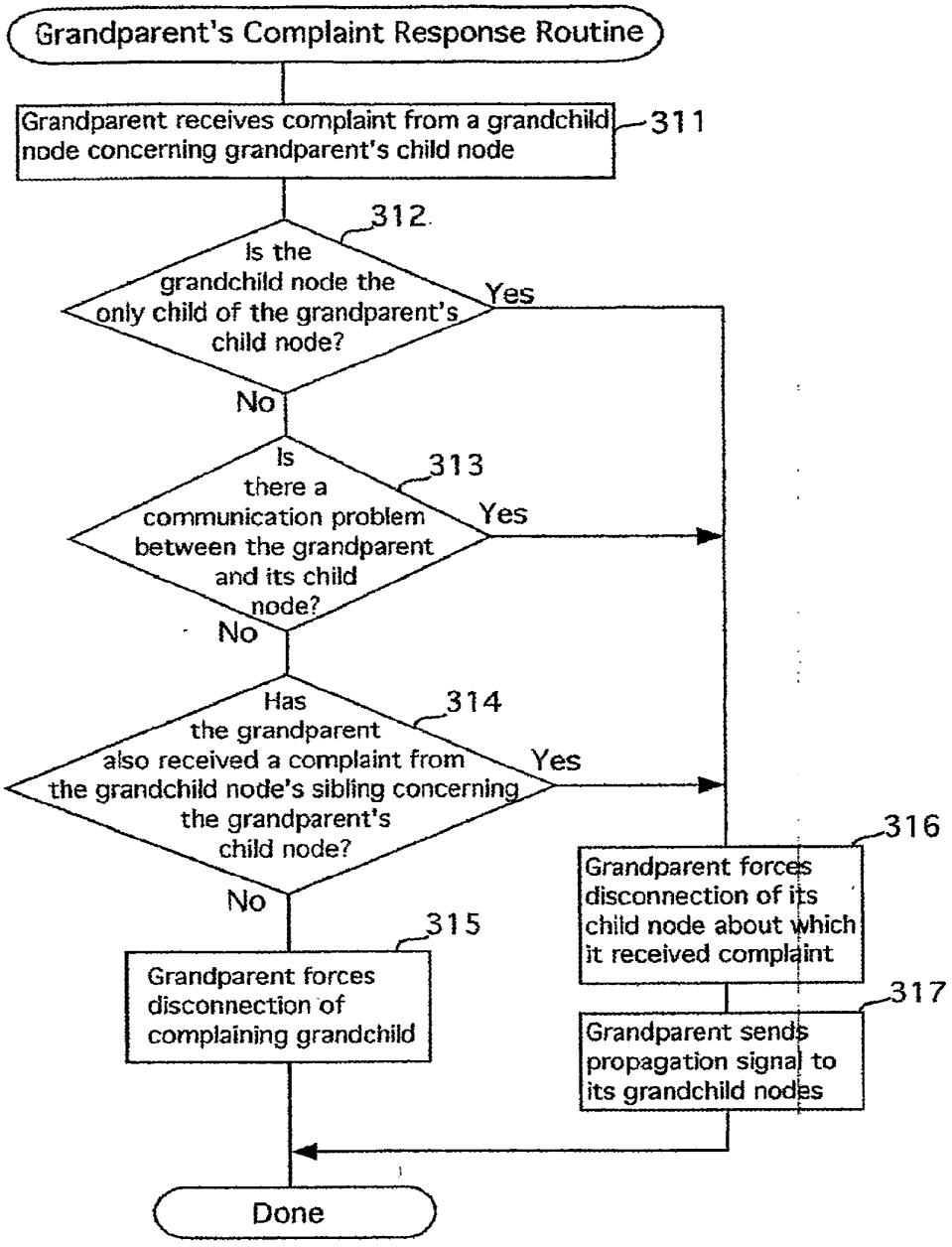

Grandparent's Complaint Response Routine
Grandparent receives complaint from a grandchild node concerning grandparent's child node
311
312
Is the grandchild node the only child of the grandparent's child node?
Yes
No
313
Is there a communication problem between the grandparent and its child node?
Yes
No
314
Has the grandparent also received a complaint from the grandchild node's sibling concerning the grandparent's child node?
Yes
No
315
Grandparent forces disconnection of complaining grandchild
316
Grandparent forces disconnection of its child node about which it received complaint
317
Grandparent sends propagation signal to its grandchild nodes
Done

Fig. 31

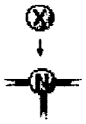


Fig. 32

Fig. 33A

Fig. 33B

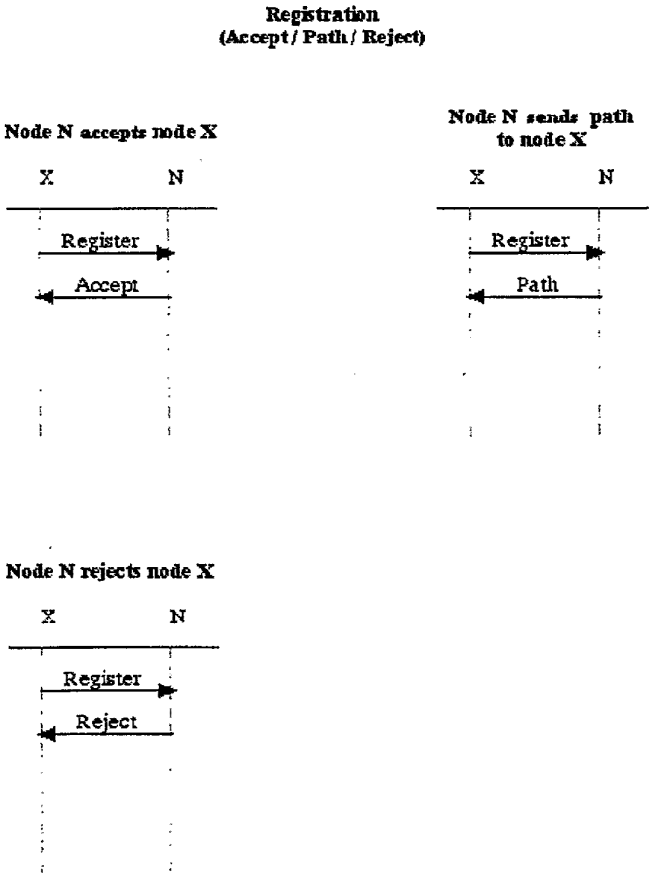


Fig. 33C

Fig. 34A

Fig. 34B

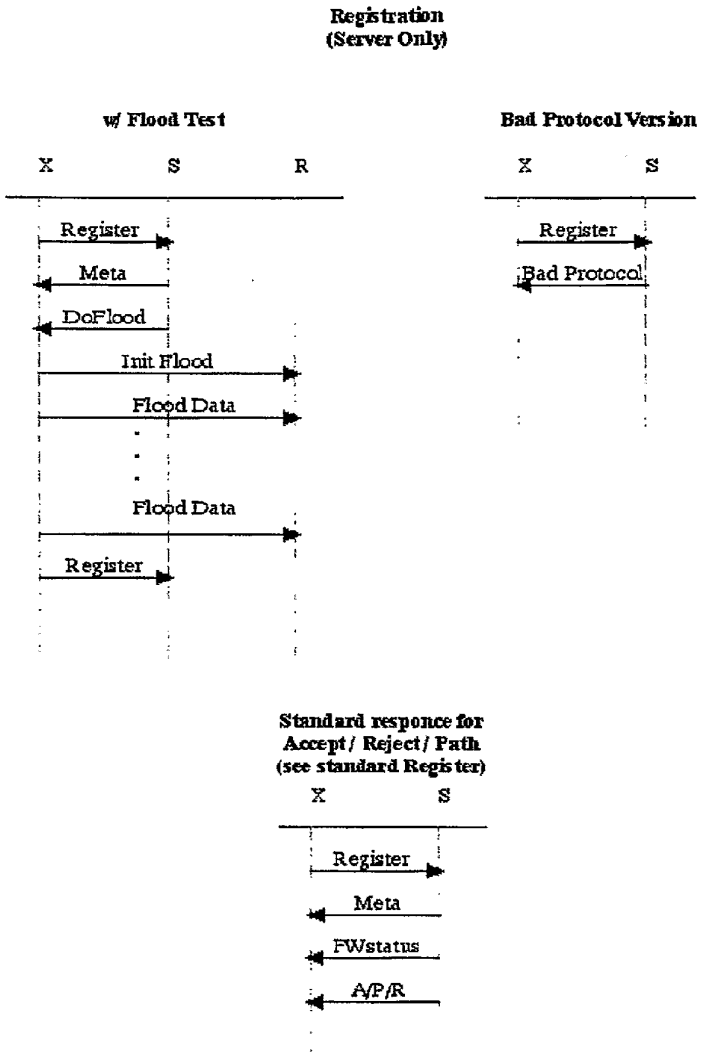


Fig. 34C

Fig. 35A

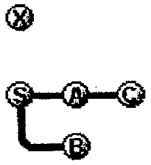
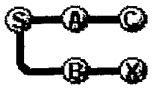


Fig. 35B



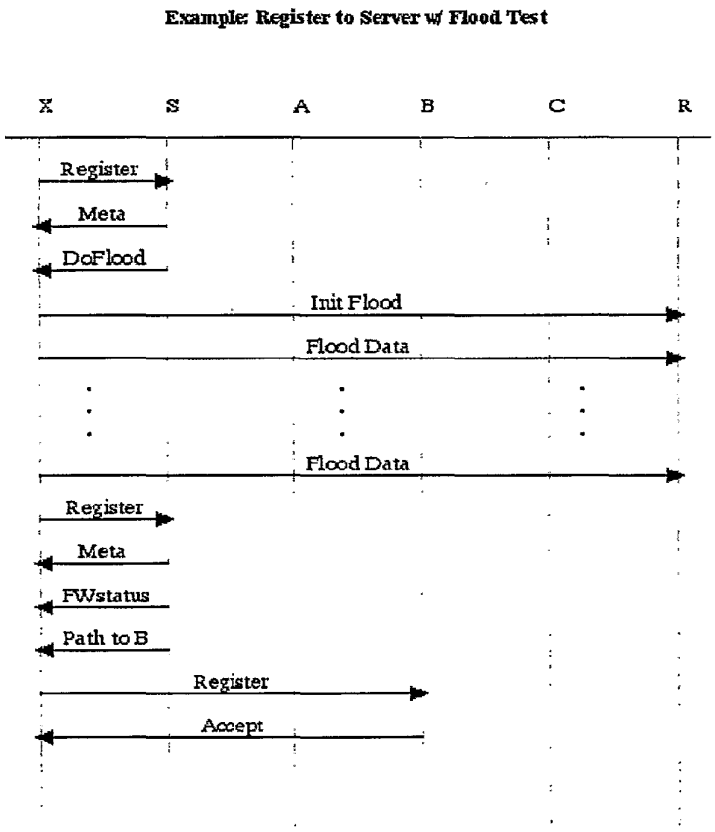


Fig. 36

Fig. 37



Fig. 38

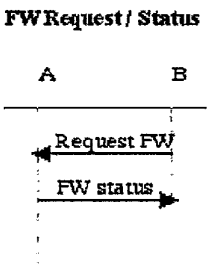


Fig. 39A



Fig. 39B

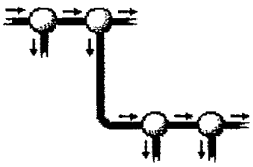
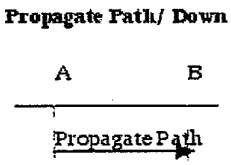


Fig. 40



28/31

Fig. 41A



Fig. 41B

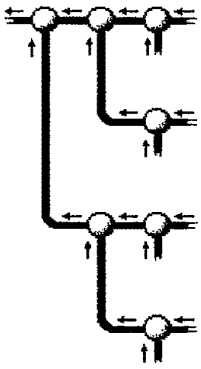


Fig. 42

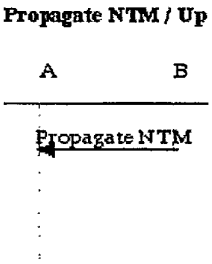


Fig. 43

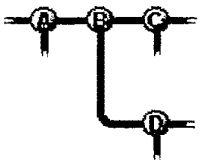


Fig. 44

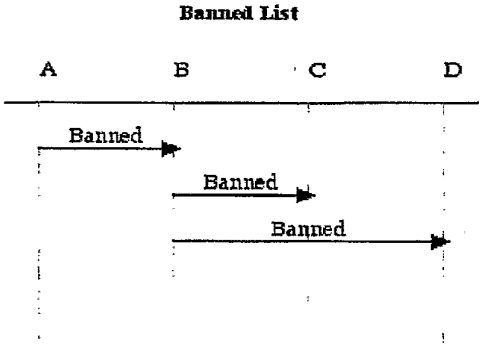


Fig. 43

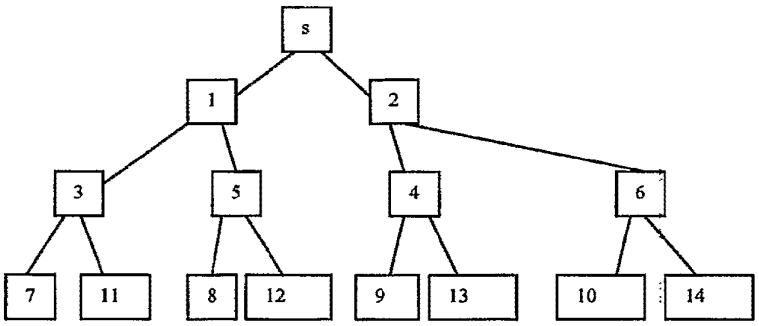


Fig. 46A

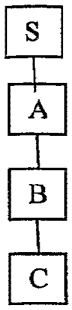


Fig. 46B

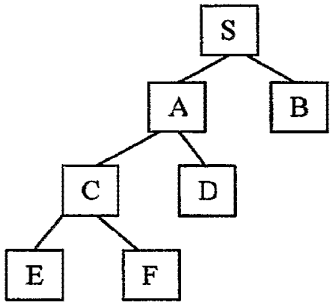
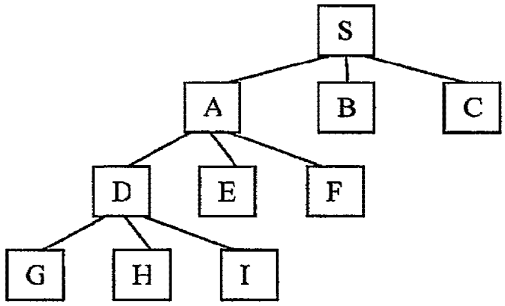


Fig. 46C



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-092752
(43)Date of publication of application : 06.04.2001

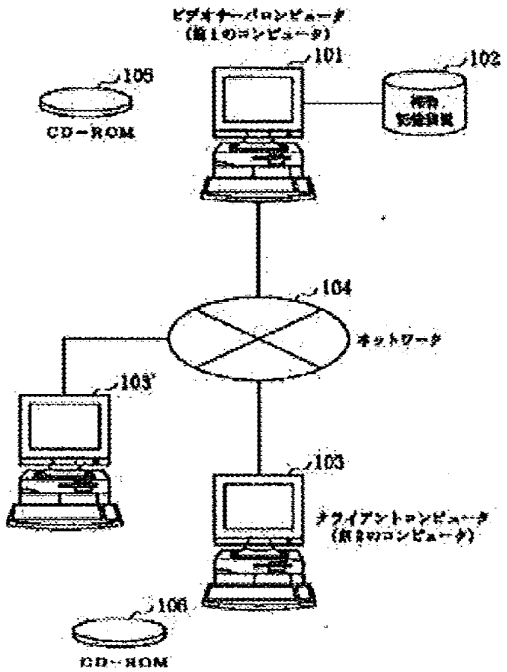
(51)Int.Cl. G06F 13/00
G06F 12/00
H04L 12/56
H04N 7/24

(21)Application number : 11-271422
(22)Date of filing : 24.09.1999

(71)Applicant : HITACHI INFORMATION SYSTEMS LTD
(72)Inventor : SHIOMI YOSHIHIRO
AIKAWA NOBUYUKI
NIGO MASAKI

(54) SYSTEM FOR DISTRIBUTING PICTURE DATA AND RECORDING MEDIUM USED FOR THE SAME

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a picture data distribution system capable of efficiently distributing picture data with appropriate quality in accordance with changes of the network connection environment of a client computer from moment to moment and a recording medium on which its program is recorded.
SOLUTION: This system has a 1st computer (image server computer 101) capable of distributing the picture data and one or more 2nd computers (client computer 103) receiving the picture data, and the 1st computer 101 receives a line situation notification message from the 2nd computer 103 and transmits the picture data (picture data extracted among a plurality of pieces of picture data which are preliminarily stored and have different compressibility or picture data compressed with optimum compressibility every time) which are subjected to compressibility being optimum to the communication speed of a line to which the 2nd computer is connected on the basis of the message.



Aylo Exhibit 1018
Page 1429 of 2308

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－92752
（P2001－92752A）
(43)公開日　平成13年４月６日(2001.4.6)

(51)Int.Cl.[7]		識別記号	FI		テーマコード(参考)	
Ｇ０６Ｆ	13/00	３５４	Ｇ０６Ｆ	13/00	３５４Ｄ	５Ｂ０８２
	12/00	５４５		12/00	５４５Ｍ	５Ｂ０８９
Ｈ０４Ｌ	12/56		Ｈ０４Ｌ	11/20	１０２Ａ	５Ｃ０５９
Ｈ０４Ｎ	7/24		Ｈ０４Ｎ	7/13	Ｚ	５Ｋ０３０

審査請求　未請求　請求項の数５　ＯＬ　（全 11 頁）

(21)出願番号　特願平11－271422

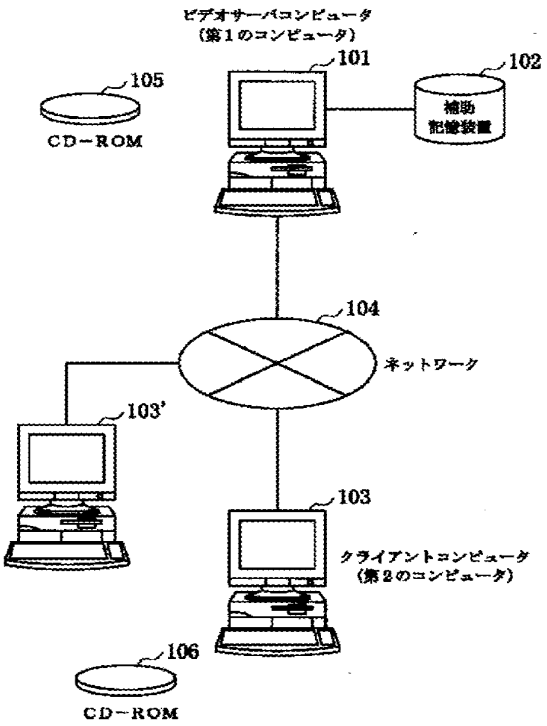
(22)出願日　平成11年９月24日(1999.9.24)

(71)出願人　000152985
株式会社日立情報システムズ
東京都渋谷区道玄坂１丁目16番５号
(72)発明者　塩見　芳弘
東京都渋谷区道玄坂一丁目16番５号　株式会社日立情報システムズ内
(72)発明者　相川　伸之
東京都渋谷区道玄坂一丁目16番５号　株式会社日立情報システムズ内
(74)代理人　100077274
弁理士　磯村　雅俊　（外１名）

最終頁に続く

(54)【発明の名称】　画像データ配信システムおよびそれに用いる記録媒体

(57)【要約】
【課題】　クライアントコンピュータのネットワーク接続環境の時々刻々の変化に応じて、適切な品質の画像データを効率よく配信することが可能な画像データ配信システムおよびそのためのプログラムを記録した記録媒体の提供。
【解決手段】　画像データを配信可能な第１のコンピュータ（画像サーバコンピュータ１０１）とその画像データを受信する１以上の第２のコンピュータ（クライアントコンピュータ１０３）を有し、第１のコンピュータ１０１は、第２のコンピュータ１０３から回線状況通知メッセージを受信し、そのメッセージに基づいて、第２のコンピュータが接続されている回線の通信速度に最適な圧縮を施した画像データ（予め記憶している圧縮率の異なる複数の画像データから抽出した画像データかまたはその都度最適な圧縮率で圧縮した画像データ）を送信する。



1

【特許請求の範囲】

【請求項１】　画像データをストリーミング配信可能な第１のコンピュータとその画像データを受信する１以上の第２のコンピュータを有する画像データ配信システムにおいて、

前記第１のコンピュータは、

前記第２のコンピュータから回線状況通知メッセージに基づいて、該第２のコンピュータが接続されている回線の通信速度に最適な圧縮を施した画像データを送信する最適画像配信手段を有することを特徴とする画像データ配信システム。

【請求項２】　前記最適画像配信手段は、

送信時間を計測するタイマと、

圧縮率の異なる複数の画像データおよび該画像データ毎に該画像データの内容を識別可能な内部識別アドレスと送信の際の送信タイミングを規定するタイムスタンプとの対応データを保持する記憶手段と、

前記第２のコンピュータからの回線状況通知メッセージに基づいて前記記憶手段から最適な圧縮率の画像データを抽出するとともに、該抽出した画像データの、前記タイマで計測された送信時間に対応する前記内部識別アドレス以降の画像データ内容を配信する手段とを有することを特徴とする請求項１記載の画像データ配信システム。

【請求項３】　前記最適画像送出手段は、

１つの画像データを保持する記憶手段と、

前記第２のコンピュータから所定の時間間隔で受信する回線状況通知メッセージに基づいて、前記記憶手段に保持されている１つの画像データから、該第２のコンピュータが接続されている回線に最適な圧縮率の画像データを動的に作成して配信する手段とを有することを特徴とする請求項１記載の画像データ配信システム。

【請求項４】　前記第１のコンピュータは、画像サーバコンピュータであり、前記第２のコンピュータは、クライアントコンピュータであることを特徴とする請求項１または請求項２記載の画像データ配信システム。

【請求項５】　請求項１または２記載の第１のコンピュータにおける各手段および第２のコンピュータにおける各手段を実現する処理をプログラムコード化して記憶したことを特徴とするコンピュータで読み取り可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ネットワーク上でビデオストリーミングデータを送信する場合におけるデータ作成および配信システムに関し、特にデータを要求するクライアントコンピュータのネットワーク接続環境の動的な変化に対応して最適なビデオデータ配信を実現するビデオデータ配信システムおよびそのためのプログラムを記録した記録媒体に関する。

2

【０００２】

【従来の技術】ネットワーク上における一般的なビデオデータ配信においては、ビデオサーバコンピュータはクライアントコンピュータのデータ送信要求に対し該当するデータのストリーミング配信を行なっている。通常のビデオデータは非常にサイズが大きいため、回線上に大きな帯域幅を必要としていた。従って、従来は、限られたネットワーク内において回線の帯域幅を考慮し適切な圧縮を行うことにより、ビデオデータ配信が行われていた。

【０００３】しかし、近年のインターネット技術の普及に伴い、インターネットを利用したビデオ配信システムが多数考案されている。インターネットを利用した場合、インターネットに接続され世界中に点在する膨大な数のコンピュータがクライアントコンピュータとなり得る。そのため、クライアントコンピュータのネットワーク接続環境はクライアントコンピュータ毎に異なることとなる。

【０００４】従って、例えば高品質（低圧縮率）のビデオデータを配信した場合、帯域幅の狭い通信回線に接続しているクライアントコンピュータは、配信された全てのビデオデータを受信しきれなくなる。この理由により、インターネット上でビデオ配信を行なう場合には、できるだけ多くのクライアントコンピュータで受信可能となるように、帯域幅の狭い通信回線に合わせた低品質（高圧縮率）のビデオデータを配信するようにしていた。この環境においては、高品質のビデオデータの受信可能なクライアントコンピュータであっても品質の低いビデオデータを受信することに甘んじなければならなかった。

【０００５】この問題点に関するものとして、ビデオサーバが、予め記憶装置に格納してある中間形態の動画データを事前に高品質および低品質の複数のデータを準備し、さらにビデオデータを要求するクライアントコンピュータが接続されている回線の通信速度を取得する手段を具備し、取得した通信速度により最適なデータを配信する仕組みが考案されている。

【０００６】

【発明が解決しようとする課題】しかしながら、一般に、クライアントコンピュータのネットワーク接続環境すなわち帯域幅は回線の混雑の度合等により刻々と変化するものであるが、上記従来技術においては、動画データの配信要求のあったクライアント端末のネットワーク性能（伝送速度）を検知し、それに応じた圧縮率の動画データを送信するものであり、ネットワーク性能の時々刻々の変化に応じて送信ビデオデータの品質（データ圧縮率）を変化させるものとはなっていない。

【０００７】本発明の目的は、クライアントコンピュータのネットワーク接続環境すなわち帯域幅の時々刻々の変化に応じて、適切な品質の画像データを効率よく配信

Aylo Exhibit 1018
Page 1431 of 2308

することが可能な画像データ配信システムおよびそのためのプログラムを記録した記録媒体を提供することである。

【０００８】

【課題を解決するための手段】本発明は、上記目的を達成するために、画像データをストリーミング配信可能な第１のコンピュータ（好ましくは画像サーバコンピュータ）とその画像データを受信する１以上の第２のコンピュータ（好ましくはクライアントコンピュータ）を有する画像データ配信システムにおいて、第１のコンピュータは、第２のコンピュータから回線状況通知メッセージに基づいて、第２のコンピュータが接続されている回線の通信速度に最適な圧縮を施した画像データを送信する最適画像配信手段を有することを特徴としている。

【０００９】なお、前記最適画像配信手段は、（ａ）送信時間を計測するタイマと、圧縮率の異なる複数の画像データおよび該画像データ毎に該画像データの内容を識別可能な内部識別アドレスと送信の際の送信タイミングを規定するタイムスタンプとの対応データを保持する記憶手段と、第２のコンピュータからの回線状況通知メッセージに基づいて記憶手段から最適な圧縮率の画像データを抽出するとともに、該抽出した画像データの、タイマで計測された送信時間に対応する内部識別アドレス以降の画像データ内容を配信する手段とを有するか、あるいは、（ｂ）１つの画像データを保持する記憶手段と、第２のコンピュータから所定の時間間隔で受信する回線状況通知メッセージに基づいて、記憶手段に保持されている１つの画像データから、第２のコンピュータが接続されている回線に最適な圧縮率の画像データを動的に作成して配信する手段とを有している。

【００１０】また、本発明の記録媒体は、前記第１のコンピュータ（画像サーバコンピュータ）における各手段および第２のコンピュータ（クライアントコンピュータ）における各手段を実現する処理をプログラムコード化してコンピュータ読み取り可能なＣＤ－ＲＯＭ，ＤＶＤ，ＦＤなどの記録媒体に記録したものである。

【００１１】

【発明の実施の形態】以下、本発明の好ましい実施の形態を説明する。画像データを配信する機能を有するネットワークに接続されている画像配信サーバを具備し、該画像配信サーバは画像データを格納可能な記憶装置を具備し、画像配信サーバにアクセス可能な不特定多数のクライアントコンピュータを具備する構成とする。

【００１２】配信するビデオデータ作成において、ビデオサーバコンピュータは、オリジナルビデオデータを基に、高品質から低品質まで複数のビデオデータを記憶装置に作成し、個々のビデオデータに対し一定時間間隔のタイムスタンプを付加する機能を具備する。

【００１３】また、ビデオデータを受信するクライアントコンピュータでは、ビデオデータ受信中に受信パケットの状態を監視し、損失パケットが一定の基準を超えた場合、クライアント側の通信回線に十分な帯域が確保されていないと判定し、また、一定の基準時間の間に損失パケットが無い場合は、クライアント側の通信回線に十分な帯域が確保されていると判定し、その判定結果を配信中のビデオサーバコンピュータへ送信する手段を具備する。

【００１４】上述の判定結果を受信したビデオサーバは、その判定結果に適切な品質のビデオデータファイルを選択し、現在配信中のビデオデータファイルの切り換えを行う。その際、切り換え前後の２つのビデオデータに付加されているタイムスタンプを参照し、配信を中止する時点でのタイムスタンプと同様のタイムスタンプを持つ時点より切り換え後のビデオデータの配信を開始する機能を具備する。

【００１５】また、上記目的を達成するために、本発明は、ビデオデータを配信する機能を有するネットワークに接続されているビデオサーバコンピュータを具備し、さらにその記憶装置に記憶されているオリジナルビデオデータを送信時に所定の圧縮率で圧縮しながら配信する機能を具備する構成とする。

【００１６】ビデオデータを受信する際において、クライアントコンピュータは、ビデオデータ受信中に受信パケットの状態を監視し、損失パケットが一定の基準を超えた場合クライアント側の通信回線に十分な帯域が確保されていないと判定し、また、一定の基準の時間、損失パケットが無い場合、クライアント側の通信回線に十分な帯域が確保されていると判定し、その判定結果を配信中のビデオサーバコンピュータへ送信する手段を特徴とする機能を具備する。

【００１７】さらに、上述の判定結果を受信したビデオサーバコンピュータは、その判定結果に従い、動的に圧縮率に変更し、その変更された圧縮率で圧縮されたビデオデータを配信する機能を具備する。

【００１８】上記構成を採用することにより、配信されるビデオデータは、受信するクライアントコンピュータのネットワーク接続環境に従い配信するビデオデータの品質を動的に変化させることができる。これにより、クライアントコンピュータは常に最適化されたビデオデータを見掛け上切れ目なく受信することが可能となる。

【００１９】また、本発明の記録媒体は、上記の如きクライアントコンピュータが行う処理やビデオサーバコンピュータが行う処理をプログラムコード化してコンピュータ読み取り可能なＣＤ－ＲＯＭ，ＤＶＤ，ＦＤなどの記録媒体に記録したものである。

【００２０】（実施例の説明）以下、本発明の実施例として、２つの実施例を説明する。図１は、第１の実施例および第２の実施例における共通のハードウエア構成を示すブロック図である。本発明は、圧縮して配信することが可能な全てのデータ（特に画像データ）に対して適

用可能であるが、本実施例では、ビデオデータ（動画データ）を例にとって説明する。

【００２１】図１において、１０１はビデオデータの作成および配信を行なうビデオサーバコンピュータ、１０２はビデオデータを格納する補助記憶装置、１０３および１０３’はクライアントコンピュータ、１０４はビデオサーバコンピュータ１０１とクライアントコンピュータ１０３が接続されるネットワーク環境である。以下の説明では、１つのサーバコンピュータと１つ以上のクライアントコンピュータからなるクライアント・サーバシステムの場合を用いて説明するが、第１のコンピュータから第２のコンピュータへの２つのコンピュータ間でのデータの配信にも適用可能であることはいうまでもない。

【００２２】（第１の実施例）まず、第１の実施例について図２～図６を用いて説明する。図２は、記憶装置に格納する圧縮ビデオデータファイルのファイル構造の説明図である。図２において、２０３は圧縮ビデオデータファイルのヘッダ情報、２０４は本実施例で取り上げる画像データファイル、２０５はＥＯＦ（ＥＯＦ：End of File）を示す。また、２０１は本発明の特徴である圧縮ビデオデータファイルのタイムスタンプであり、２０２は、タイムスタンプ２０１と対応する圧縮ビデオデータファイルの内部識別アドレスである。なお、本実施例で取り上げる圧縮率の異なる他の画像ファイル（圧縮画像ファイルＢおよび圧縮画像ファイルＣ）構造も図２と同様である。内部識別アドレス２０２は、ファイル毎にタイムスタンプ１０２に対応して設けられるものであるが、ファイルの品質（圧縮率）が異なればそれに伴って異なる値を有する。

【００２３】図３は、同一画像ファイルから生成された圧縮率の異なる３種類の圧縮ビデオデータファイルのヘッダ情報の説明図である。同図（ａ）は低圧縮画像ファイルＡのヘッダ情報を、同図（ｂ）は中圧縮画像ファイルＢのヘッダ情報を、同図（ｃ）は高圧縮画像ファイルＣのヘッダ情報を示している。３０１はタイムスタンプ情報、３０２は、インデックスに対応する図２に示すファイルの内部識別アドレスである。

【００２４】図４は、圧縮率の異なる３つの画像ファイルとタイムスタンプと内部識別アドレスの関係を説明するための図である。同図（ａ）は、低圧縮画像ファイルＡとタイムスタンプと内部識別アドレスの関係を、同図（ｂ）は中圧縮画像ファイルＢとタイムスタンプと内部識別アドレスの関係を、同図（ｃ）は高圧縮画像ファイルＣとタイムスタンプと内部識別アドレスの関係を、それぞれ示している。本例では、説明を簡単にするために、同図（ａ）に示す低圧縮画像ファイルＡを１／２に圧縮したものを中圧縮ファイルＢとして同図（ｂ）に示し、さらにこれを１／２に圧縮したものを高圧縮画像ファイルＣとして同図（ｃ）に示すとともに、タイムスタンプ，内部識別アドレスと圧縮画像ファイルＡ～Ｃの内容の関係がわかりやすいように画像ファイル中に「い」、「ろ」、「は」・・・などを便宜的に示した。

【００２５】図５は、ビデオサーバコンピュータ１０１に保持される、通信速度と該通信速度に最適化された圧縮画像ファイルとの対応図である。図５における通信速度を表わす記号（ｍ、ｎ）の大小関係は、ｍ＜ｎとする。同図に示すように、通信速度が低速の場合は（～ｍ）、低品質である高圧縮画像ファイルＣを、通信速度が高速の場合は（ｎ～）、高品質である低圧縮画像ファイルＡを配信するようにする。

【００２６】図６および図７は、第１の実施例の動作を説明するためのフローチャートである。まず、ビデオデータを配信するビデオサーバコンピュータ１０１の動作を、図６を用いて詳細に説明する。

【００２７】図６において、クライアントコンピュータ１０３からのデータ要求に従い、該当する記憶装置内のビデオデータ（システムには初期配信ファイルが設定済とする。本実施例では、初期配信ファイルは「中圧縮画像ファイルＢ」とする）の配信を開始し、かつ配信時間の計測を開始する（ステップ６０１）。

【００２８】クライアントコンピュータ１０３から低品質データの受信要求（後述する図７のステップ７０６参照）があった場合（ステップ６０３）、図５の対応図に従い、現在送信中のファイル（ファイルＢ）より１段階品質の劣る「高圧縮画像ファイルＣ」を送信ファイルとして確定する（ステップ６０４、６０５）。

【００２９】次に現在時点の配信時間を測定する（ステップ６０６）。本実施例では、この時の測定時間が４ｍｓであったと仮定し、図３に示すヘッダ情報より、「ファイルＢ」の４ｍＳの開始内部識別アドレスは「iiii」と確定する（ステップ６０７）。すなわち、画像「い」、「ろ」は配信済みとする。ビデオサーバコンピュータ１０１は即座に配信中の「中圧縮画像ファイルＢ」の配信を中止し、内部識別アドレス「iiii」から「高圧縮画像ファイルＣ」の画像「は」、「に」、・・・の配信を開始する（ステップ６０８）。

【００３０】逆に、クライアントコンピュータ１０３より高品質データの受信要求（後述する図７のステップ７０７参照）があった場合（ステップ６０３：Ｙ）、図５の対応図に従い、現在配信中の「中圧縮画像ファイルＢ」より１段階品質の高い「低圧縮画像ファイルＡ」を次送信ファイルとして確定する（ステップ６０４、６０５）。以下の動作は同様である。すなわち、ビデオサーバコンピュータ１０１は即座に配信中の「中圧縮画像ファイルＢ」の配信を中止し、内部識別アドレス「iiii」から「低圧縮画像ファイルＡ」の画像「は」、「に」、・・・の配信を開始する（ステップ６０８）。

【００３１】なお、現時点での配信データが「高圧縮画像ファイルＣ」であった場合にさらに低品質の高圧縮の

Aylo Exhibit 1018
Page 1433 of 2308

画像ファイルが要求された場合、図５より１段階品質の劣るビデオデータは存在しないことがわかる。また、現時点での配信データが「低圧縮画像ファイルＡ」であった場合にさらに高品質の低圧縮画像ファイルが要求された場合にも、図５より１段階品質の高いビデオデータは存在しないことがわかる。このような場合は、データの存在しない旨のメッセージを送信し、現時点でのデータ配信を続行する（ステップ６０９）。

【００３２】また、ビデオサーバコンピュータ１０１は、ビデオデータの終端すなわち図２に示す「ＥＯＦ」まで配信を行った場合（ステップ６０２：Ｙ）、データの配信を終える。

【００３３】次に、ビデオデータを受信するクライアントコンピュータ１０３の動作を、図７を用いて詳細に説明する。図７において、先ず、ビデオサーバコンピュータ１０１に対するビデオデータ要求により、ビデオサーバから送信されたデータの受信を開始し、同時に損失パケット監視タイマをリセットする（ステップ７０１）。

【００３４】損失パケット数がシステムに既設定済の数を超過した場合（ステップ７０２：Ｙ）、クライアントコンピュータ側の通信回線に十分な帯域が確保されていないと判断し、ビデオサーバに低品質データの要求メッセージを送信する（ステップ７０６）。損失パケット数がシステムに既設定済の数を超過しない場合（ステップ７０２：Ｎ）、損失パケットの発生を常に監視し（ステップ７０３）、損失パケットが発生した場合（ステップ７０３：Ｙ）、損失パケット監視タイマをリセットする（ステップ７０４）。

【００３５】この損失パケット監視タイマが、システムに既設定済の一定時刻を超過した場合、すなわち損失パケットが一定時間発生しなかった場合（ステップ７０３：Ｎ，７０５：Ｙ）、クライアントコンピュータ１０３側の通信回線に十分な帯域が確保されており現在受信中のビデオデータよりも高品質なデータ（低圧縮のデータ）の受信が可能と判断し、ビデオサーバコンピュータ１０１に高品質データの要求メッセージを送信する（ステップ７０７）。ステップ７０８では、ビデオサーバコンピュータ１０１からの配信データに従い今までと異なる品質の配信データを受信し、同時に損失パケット監視タイマをリセットする。

【００３６】ここで、図６のステップ６０９に示すデータ無しのメッセージを受信した場合、そのメッセージに従い、低品質データ無しの場合はステップ７０６の低品質データ要求メッセージ送信を、また、高品質データ無しの場合はステップ７０７の高品質データ要求メッセージ送信を中止する。なお、クライアントコンピュータ１０３はビデオデータの終端をしめす「ＥＯＦ」まで受信した場合（ステップ７０９：Ｙ）、受信動作を終了する。

【００３７】最後に、ビデオデータフォーマットについて詳述する。本実施例における配信ビデオデータのフォーマットは、図２に示すようにヘッダ情報部分（２０３）とデータ部分（２０４）とから構成され、ヘッダ部分には、図３に示すような、タイムスタンプ毎のアドレスが保持されている。ビデオサーバによりビデオデータを圧縮する際に、ヘッダ部分を生成し付加する。なお、図２および図３における２０１と３０１、２０２と３０２は同値である。

【００３８】以上、第１の実施例は、オリジナルなビデオデータを基に高品質（低圧縮）から低品質（高圧縮）まで、品質の異なる複数のビデオデータをビデオサーバの記憶装置に記憶しておき、クライアントコンピュータからの配信要求に対し、クライアントコンピュータにおける受信状態に応じて、最適な品質のビデオデータを動的に切替えながら配信するものである。

【００３９】（第２の実施例）次に説明する第２の実施例は、ビデオサーバに保持するビデオデータは、オリジナルなもの１種類のみとし、サーバから配信するビデオデータの圧縮を、クライアントコンピュータにおける受信状態に応じて動的に最適化するものである。

【００４０】以下、本第２実施例のビデオサーバ側の処理を、図８を用いて詳細に説明する。同図において、まず、クライアントコンピュータからのデータ要求に従い、該当する記憶装置内のオリジナルビデオデータをビデオサーバに取り出し（ステップ８０１）、圧縮を行いながら（システムには初期圧縮レートが設定済とする。）送信する（ステップ８０３）。クライアントコンピュータ１０３より低品質データ要求または高品質データ要求のメッセージ受信があった場合（ステップ８０４）、圧縮レートを変更する（ステップ８０５）。

【００４１】なお、ビデオサーバコンピュータ１０１はビデオデータの終端まで配信した場合（ステップ８０２）、配信動作を終了する。

【００４２】ビデオデータを受信するクライアントコンピュータの動作は、第１の実施例と同様のため詳細な説明は省略する。

【００４３】なお、上記の如き、ビデオサーバコンピュータが行う各処理手順およびクライアントコンピュータが行う各処理手順を、ＣＤ－ＲＯＭ，ＤＶＤ，あるいはＦＤなどの記録媒体に一緒にまたは別々に記録して市場に流通させることができる。利用者は、この記録媒体をビデオサーバコンピュータとクライアントコンピュータにインストールすることにより、本発明を簡単に利用することが可能になる。

【００４４】

【発明の効果】以上詳細に説明したように、本発明によれば、ビデオサーバにより配信されるビデオデータは、受信するクライアントコンピュータのネットワーク接続環境に従い動的に変化させることが実現できる。これにより、クライアントコンピュータは常に最適化されたビ

Aylo Exhibit 1018
Page 1434 of 2308

9

デオデータを見掛け上切れ目なく受信することが可能となる。

【図面の簡単な説明】

【図１】本発明の第１の実施例および第２の実施例における共通のハードウエア構成を示すブロック図である。

【図２】記憶装置に格納する圧縮ビデオデータファイルのファイル構造の説明図である。

【図３】同一画像ファイルから生成された圧縮率の異なる３種類の圧縮ビデオデータファイルのヘッダ情報の説明図である。

【図４】圧縮率の異なる３つの画像ファイルとタイムスタンプと内部識別アドレスの関係を説明するための図である。

【図５】ビデオサーバコンピュータに保持される、通信速度と該通信速度に最適化された圧縮画像ファイルとの対応図である。

【図６】第１の実施例におけるビデオサーバコンピュータの動作を説明するためのフローチャートである。

【図７】第１の実施例におけるクライアントコンピュータの動作を説明するためのフローチャートである。

【図８】第２の実施例におけるビデオサーバコンピュータの動作を説明するためのフローチャートである。

10

【符号の説明】

１０１：ビデオデータの作成および配信を行なうビデオサーバコンピュータ（第１のコンピュータ）

１０２：ビデオデータを格納する記憶装置

１０３，１０３’：ネットワークと接続されるクライアントコンピュータ（第２のコンピュータ）

１０４：ビデオサーバコンピュータとクライアントコンピュータが接続されるネットワーク環境

１０５：プログラム記録媒体（ＣＤ－ＲＯＭ）

１０６：プログラム記録媒体（ＣＤ－ＲＯＭ）

２０１：圧縮ファイルのタイムスタンプ

２０２：圧縮ファイルのタイムスタンプに対応した内部識別アドレス

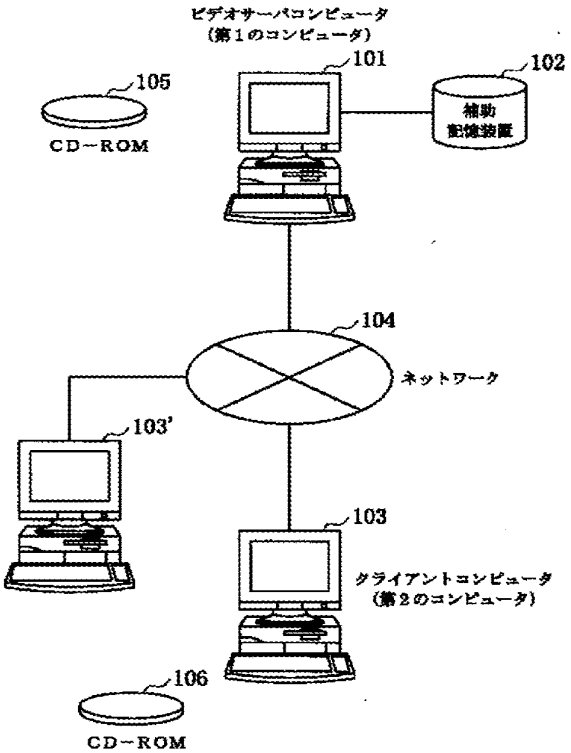
２０３：圧縮ファイルのファイルのヘッダ情報

２０４：本実施例で取り上げるファイルＡのファイルフォーマット

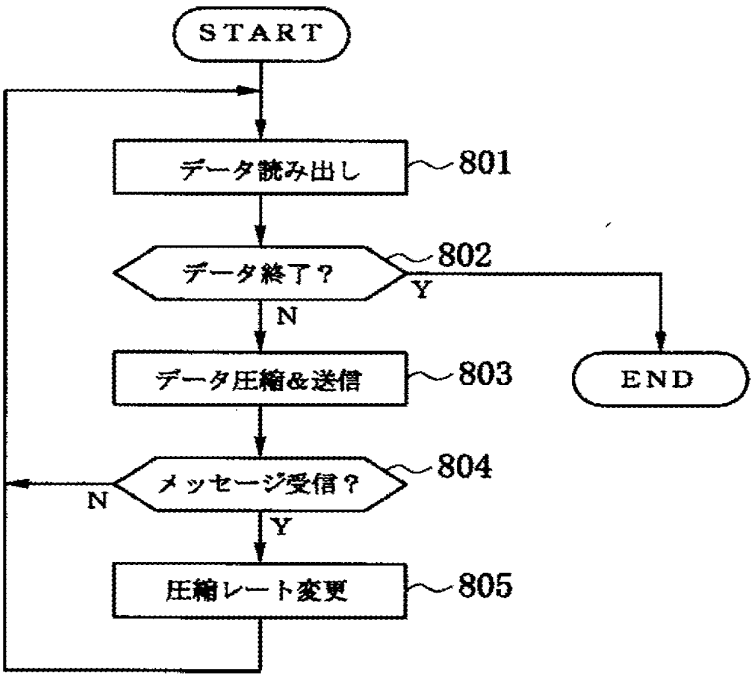
３０１：圧縮ファイルヘッダのタイムスタンプ情報

３０２：圧縮ファイルヘッダのタイムスタンプ情報に対応する内部識別アドレス

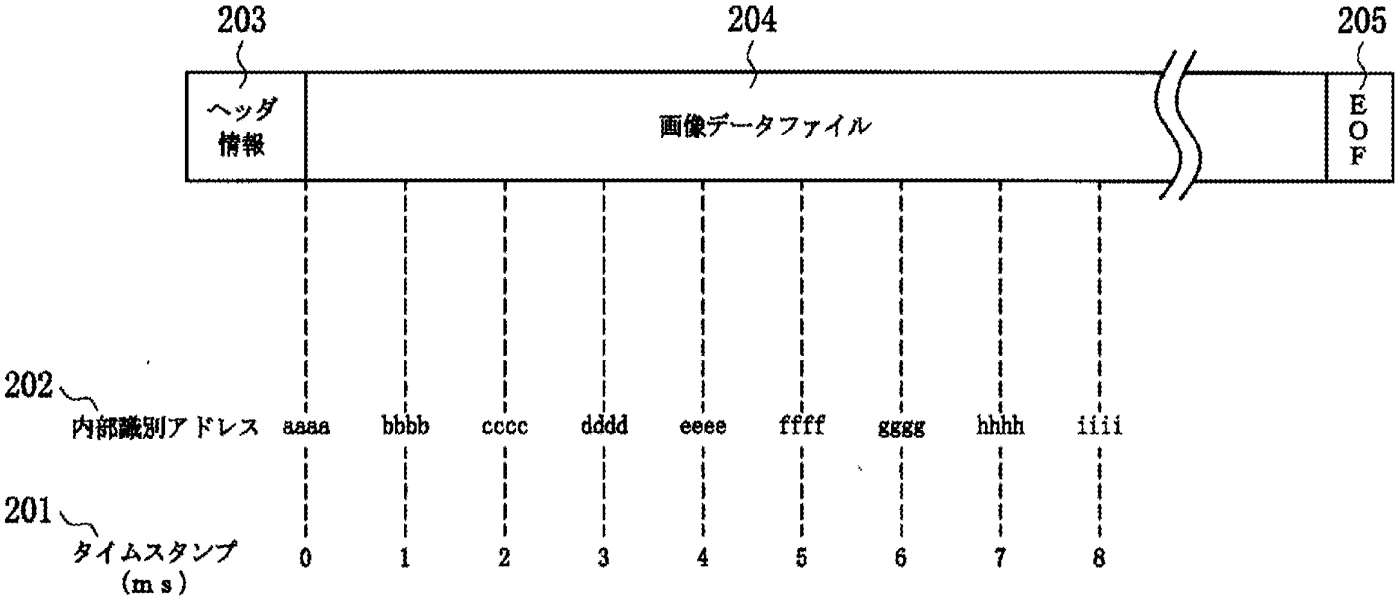
【図１】



【図８】



【図２】



特開２００１－９２７５２

Aylo Exhibit 1018
Page 1436 of 2308

【図３】

ヘッダ情報の構造

301　302

タイムスタンプ	内部識別アドレス
0	a a a a
1	b b b b
2	c c c c
3	d d d d
4	e e e e
5	f f f f
6	g g g g
7	h h h h
8	i i i i
:	:

低圧縮
画像ファイル（A）

（a）

301　302

タイムスタンプ	内部識別アドレス
0	a a a a
1	c c c c
2	e e e e
3	g g g g
4	i i i i
5	k k k k
6	m m m m
7	o o o o
8	q q q q
:	:

中圧縮
画像ファイル（B）

（b）

301　302

タイムスタンプ	内部識別アドレス
0	a a a a
1	e e e e
2	i i i i
3	m m m m
4	q q q q
5	u u u u
6	y y y y
7	:
8	:
:	:

高圧縮
画像ファイル（C）

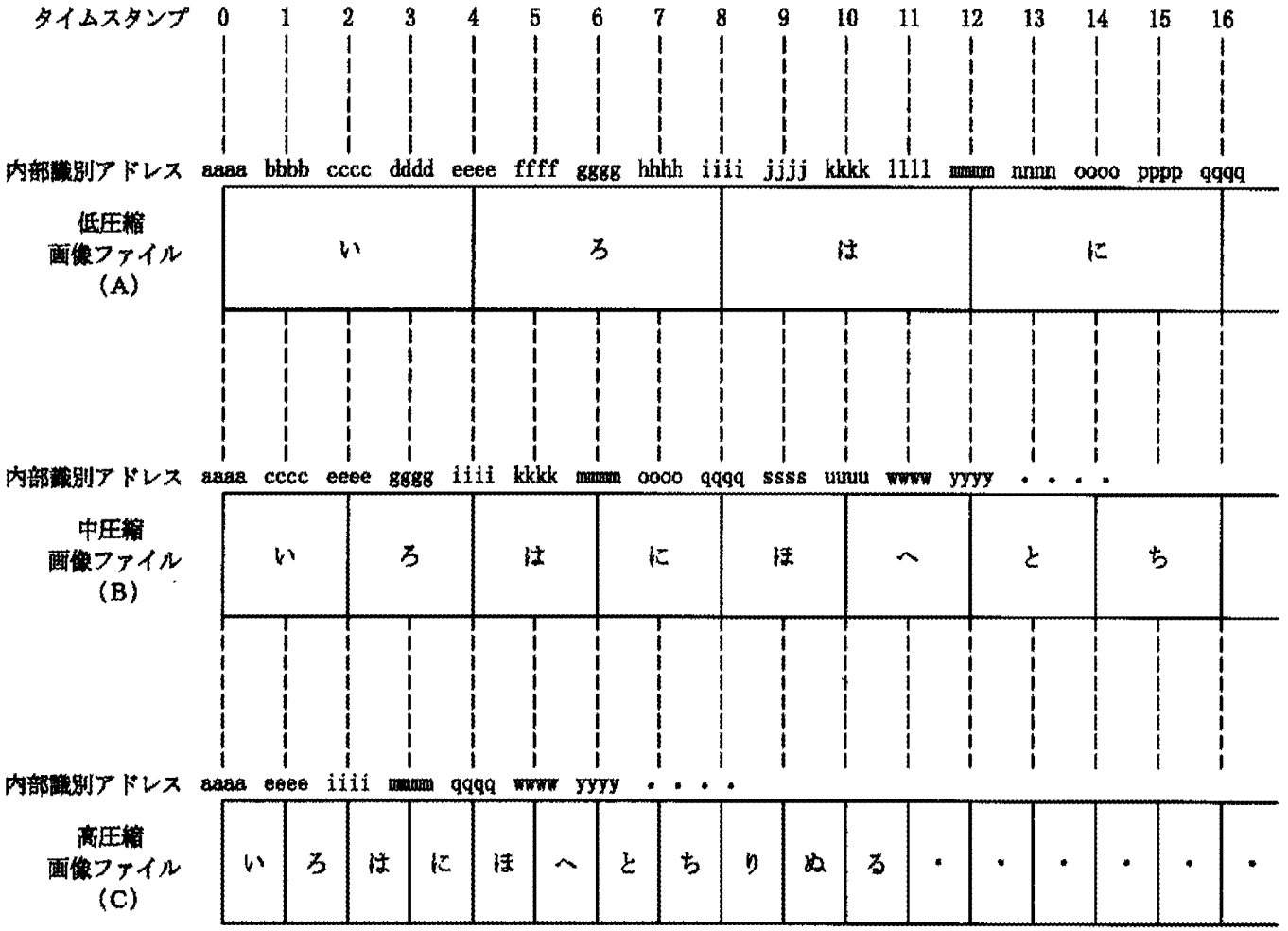
（c）

【図５】

通信速度と対応圧縮ファイル

項番	通信速度（ｂｐｓ）	ファイル名
1	～m	圧縮画像ファイルC
2	m～n	圧縮画像ファイルB
3	n～	圧縮画像ファイルA

通信速度の大小関係　m＜n



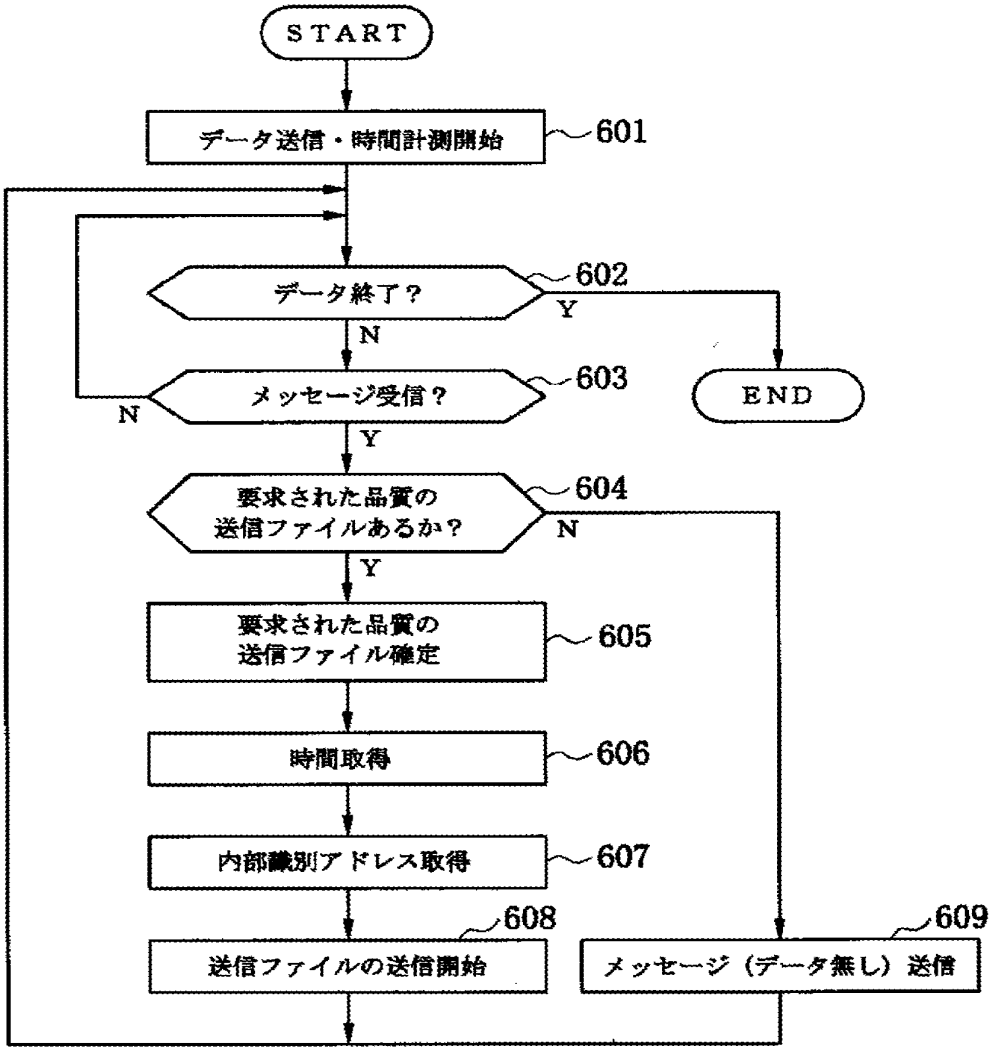
【図４】

特開２００１－９２７５２

Aylo Exhibit 1018
Page 1438 of 2308

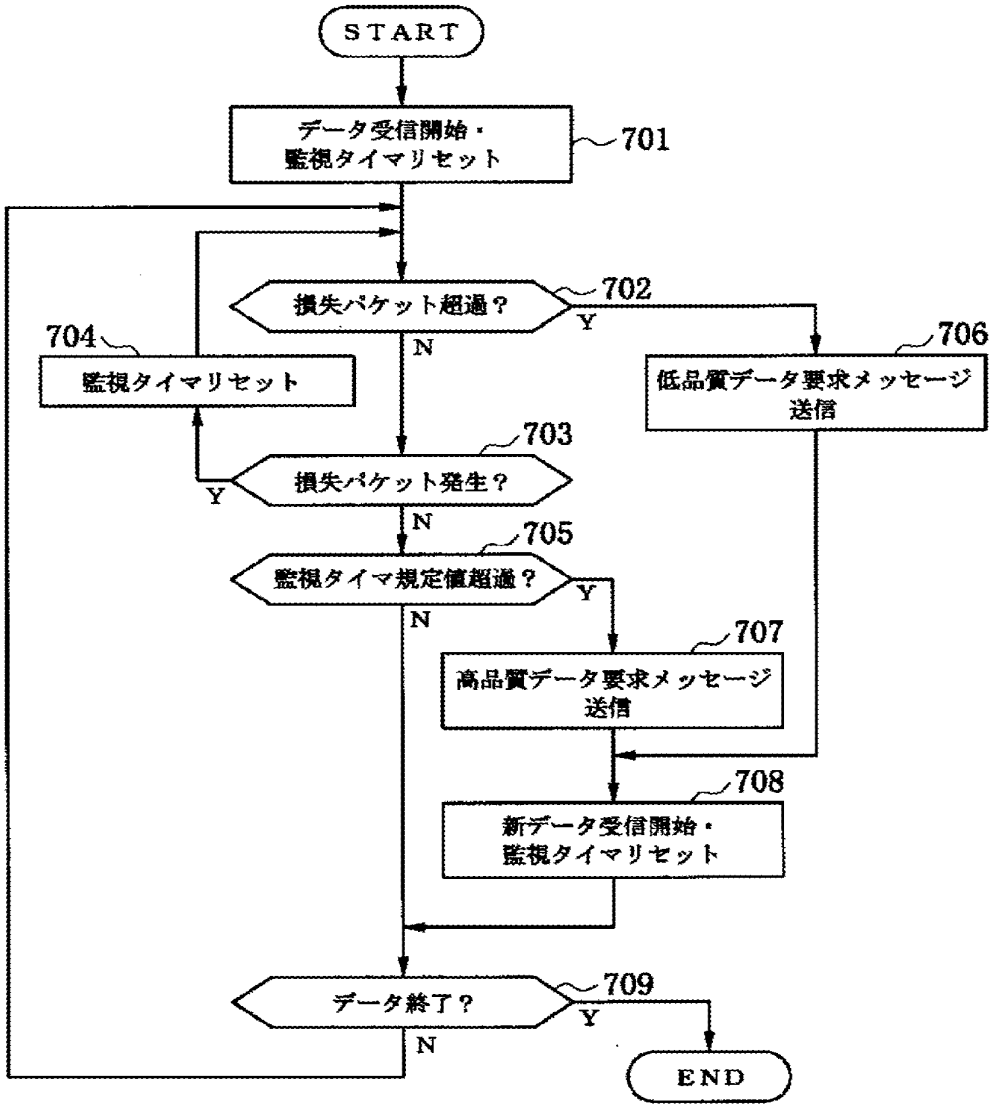
【図６】

（ビデオサーバコンピュータの処理）



【図７】

(クライアントコンピュータの処理)



フロントページの続き

(72)発明者　二郷　正樹
東京都渋谷区道玄坂一丁目16番５号　株式
会社日立情報システムズ内

Ｆターム(参考)　5B082 GA01 HA05
5B089 GA11 JA33 JB03 JB22 KA05
KA12 KB06 KB11 KC28 KC53
KC59 KC60 KH28
5C059 KK34 SS09 TA60 TA71 TC21
UA02 UA38
5K030 GA08 HB02 LA07 LC09 LD13
MB13

Aylo Exhibit 1018
Page 1440 of 2308

PLUS Search Results for S/N 11116783, Searched Thu Jul 31 10:22:47 EDT 2014

The Patent Linguistics Utility System (PLUS) is a USPTO automated search system for U.S. Patents from 1971 to the present PLUS is a query-by-example search system which produces a list of patents that are most closely related linguistically to the application searched. This search was prepared by the staff of the Scientific and Technical Information Center, SIRA.

20050262257 99
7818444 90
8402156 90
8612624 90
8683066 90
20080195743 90
20090043906 90
20100106847 90
20100107206 90
20110035507 90
20130013800 90
20140101329 90
20140207966 90
20110188567 83
4541854 75
6574618 74
6757894 74
7051315 74
7197570 74
20010034736 74
20010037399 74
20020087717 74
20020087963 74
20120210216 71
20130236158 71
20140189765 71
3554451 59
3640308 59
4136720 59
4157250 59
4479909 59
4505724 59
4961535 59
6802978 59
8027587 59
8027587 59
20030085176 59
20030235523 59
20040255605 59
20060242903 59
20120311174 59
3800053 56
4004594 56
4346524 56
4452256 56
4932424 56
5638834 56
5970626 56
6918113 55
6959320 55

Intelligent streaming
B Birney - Microsoft Corp, 2003
Cited by 29 Related articles Cite Save

IIS smooth streaming technical overview
A Zambelli - Microsoft Corporation, 2009 - bogotobogo.com
... 3 Windows Media Stream Thinning, MBR and **Intelligent Streaming** .....
3 The ... Web. A Brief History of Multiple Bit Rate **Streaming** Windows Media Stream Thinning, MBR and **Intelligent Streaming** The ...
Cited by 157 Related articles All 6 versions Cite Save More

Intelligent streaming framework
RS Lisitsa, DA Sather, C Hagiu - US Patent 6,766,407, 2004 - Google Patents
The present invention analyzes a user's or client application's requirements and searches for a solution based on the end-to-end latency requirements (requested or derived), data formats, control protocols, timing and synchronization, local **streaming**, and resource ...
Cited by 23 Related articles All 3 versions Cite Save

Measurements of the congestion responsiveness of windows streaming media
J Nichols, M Claypool, R Kinicki, M Li - ... support for digital audio and video, 2004 - dl.acm.org
... WSM supports **Intelligent Streaming** [1], an application level means of reducing the **streaming** bitrate. **Intelligent Streaming** lowers the bitrate of the stream in mid-playout during network congestion and then switch back to a higher bitrate when congestion abates. ...
Cited by 35 Related articles All 8 versions Cite Save

Intelligent network streaming and execution system for conventionally coded applications
ME Benitez, AM Holler, LV Shah, DT Arai... - US Patent ..., 2006 - Google Patents
An **intelligent** network **streaming** and execution system for conventionally coded applications provides a system that partitions an application program into page segments by observing the manner in which the application program is conventionally installed. A minimal portion ...
Cited by 74 Related articles All 3 versions Cite Save

Data transmitting apparatus, data receiving apparatus, data transmitting manner, and data receiving manner
T Watanabe - US Patent App. 10/515,257, 2003 - Google Patents
... For example, methods of changing a bandwidth according to a density of communication include SureStream (SureStream is a registered trademark of Real Networks, Inc.), **Intelligent Streaming** (**Intelligent Streaming** is a registered trademark of Microsoft Corporation), and "Data ...
Cited by 37 Related articles All 2 versions Cite Save

Method and apparatus for generating and marketing video e-mail and an intelligent video streaming server
A Jalali, A Moradi - US Patent App. 10/634,733, 2003 - Google Patents
A method and apparatus for generating and sending video emails containing video and graphics, and optionally text, animation, sound, attachments and links. The graphics are produced as a skin on the computer screen and the video plays in a defined location on ...
Cited by 37 Related articles All 2 versions Cite Save

smt-SPRINTS: software precomputation with intelligent streaming for resource-constrained SMTs
T Wang, CD Antonopoulos... - Euro-Par 2005 Parallel ..., 2005 - Springer
Abstract We present SPRINTS, a source-level speculative precomputation framework for scientific applications running on SMTs with two execution contexts. Our framework targets memory-bound applications and reduces memory latency by prefetching long streams of ...
Cited by 4 Related articles All 10 versions Cite Save

CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming
X Zhang, J Liu, B Li, TSP Yum - INFOCOM 2005. 24th Annual ..., 2005 - ieeexplore.ieee.org
... We propose a scalable membership and partnership management algorithm together with an **intelligent** scheduling algorithm, which enable efficient and continuous **streaming** of medium- to high-bandwidth con- tents with low control overhead. ...
Cited by 1924 Related articles All 31 versions Cite Save

Extended intelligent video streaming system
A Jalali, A Moradi - US Patent App. 11/087,363, 2005 - Google Patents
A system for transmitting and displaying large media files uses an online **streaming** service to upload full-length movies and other video and audio files to a wide array of viewers. The

system includes a robust, scalable, and fast upload technique that allows files of any size ...
Cited by 15 Related articles All 2 versions Cite Save

Gigavoxels: Ray-guided **streaming** for efficient and detailed voxel rendering

C Crassin, F Neyret, S Lefebvre... - Proceedings of the 2009 ..., 2009 - dl.acm.org
... Volume data can require large amounts of memory, thus limiting the scene's extent and resolution of details. The fact that the scene can no longer be held entirely in memory implies the need for **intelligent** data **streaming** from larger, but slower memory units. ...
Cited by 147 Related articles All 24 versions Cite Save

System for capability based multimedia **streaming** over a network

A Sahai, RK Gupta, J Kothari - US Patent 6,594,699, 2003 - Google Patents
... greater than 32 MB of memory. Currently there is no way to distinguish between different clients, their hardware and software capabilities, and use that information to do **intelligent streaming** of multimedia data by the server. ...
Cited by 181 Related articles All 2 versions Cite Save

Intelligent streaming framework

C Hagiu, RS Lisitsa, DA Sather - US Patent 7,073,011, 2006 - Google Patents
The present invention analyzes a user's or client application's requirements and searches for a solution based on the end-to-end latency requirements (requested or derived), data formats, control protocols, timing and synchronization, local **streaming**, and resource ...
Cited by 3 Related articles All 3 versions Cite Save

Delving into internet **streaming** media delivery: a quality and resource utilization perspective

L Guo, E Tan, S Chen, Z Xiao, O Spatscheck... - Proceedings of the 6th ..., 2006 - dl.acm.org
... MBR encoding also enables dynamic stream switch among streams of different rates encoded in the object during a user session, in order to adapt to the current bandwidth, which is called **Intelligent Streaming** [4] in the Windows media service and SureStream [7] in the ...
Cited by 89 Related articles All 21 versions Cite Save

End-to-end security in the presence of **intelligent** data adapting proxies: The case of authenticating transcoded **streaming** media

C Gentry, A Hevia, R Jain, T Kawahara... - Selected Areas in ..., 2005 - ieeexplore.ieee.org
Abstract—We consider the problem of maintaining end-to-end security in the presence of **intelligent** proxies that may adaptively modify data being transmitted across a network. The video coding community considers this problem in the context of transcoding media ...
Cited by 19 Related articles All 4 versions Cite Save

... School of Computer Science, Beijing Institute of Technology, Beijing 100081, China); Survey of Key Technologies of the Distribution System for **Streaming** ...

Y Ge - Acta Electronica Sinica, 2009
Cited by 17 Related articles Cite Save

Enhanced soft-handoff scheme for real-time **streaming** services in **intelligent** transportation systems based on CDMA

Y Chung, DH Cho - **Intelligent** Transportation Systems, IEEE ..., 2006 - ieeexplore.ieee.org
Abstract—Code division multiple access (CDMA)-based radio-on-fiber road–vehicle communication systems were proposed to support real-time **streaming** services in **intelligent** transportation systems (ITSs). Soft handoff should be provided for real-time **streaming** ...
Cited by 14 Related articles All 10 versions Cite Save

Comparative Analysis and Tests of **Intelligent Streaming** Video on Demand for Next Generation Networks: Two Colombian Study Cases

J Zabala, M Diaz, E Gaona... - ICNS 2011, The Seventh ..., 2011 - thinkmind.org
Abstract: his document proposes to evaluate the transmission of Video on Demand using **Intelligent Streaming** and Next Generation Networks concepts. **Intelligent Streaming** is an adaptive methodology to take advantage of bandwidth and the network resources, the ...
Cited by 2 Related articles Cite Save

Intelligent control for adaptive video **streaming**

V Menkovski, A Liotta - Consumer Electronics (ICCE), 2013 ..., 2013 - ieeexplore.ieee.org
Abstract—We present an autonomous learning agent for adaptive video **streaming** in best effort networks. The agent learns an optimal control strategy in regards to the delivered quality of experience without the need for implementation of a complex heuristics.

Cited by 8 Related articles Cite Save

An **intelligent streaming** media video service system

C Huang, J Li, H Shi - ..., '02. Proceedings. 2002 IEEE Region 10 ..., 2002 - ieeexplore.ieee.org

Abstract: With tlie developiiieiit o F iiiultiiiiedia and nehvork technologies~ tlie iietwork video services have being becoiiie more and iiiore important. The iiitelligent strcaniiiig iiiedia senices system, based 011 tlie coiiibiiiatioii between **streaming** media aiid MPEG **...**

Cited by 2 Related articles Cite Save

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	4705	(upconvert$3 or upshift$3) with (stream$3 or (video with strem$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L2	4705	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L3	0	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) same ((multiple or plurality or many or three or two) near3 copi$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L4	16	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) same ((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L5	253	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) and ("709"/$$$.ccls. or "725"/$$$.ccls.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L6	244	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) and ("709"/$$$.ccls. or "725"/$$$.ccls.) and network	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L7	23030	((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)) same (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L8	756102	((multiple or plurality or many or	US-PGPUB;	OR	ON	2014/07/31

		three or two) near5 (version or format$4 or copies or cop$4))	USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			00:31
L9	23030	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L10	2345	L9 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L11	2345	L9 and "709"/$$$.ccls. and (stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L12	738	L9 and "709"/231.ccls. and (stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L13	8	("20020174434" | "20030067872" | "20040030797").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L14	44	("20020174434" | "20030067872" | "20040030797").PN. or ("20020073167" | "20050123058" | "20070024705" | "20080028428" | "20080056373" | "6091775" | "6091777" | "6510553" | "6731600" | "6732183" | "6968387" | "7075986" | "7349976").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L15	21	L14 and (Stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L16	5	L14 and (Stream$3 or (video with stream$3)) and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2014/07/31 00:31

			DERWENT; IBM_TDB			
L17	260	L9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L18	260	L9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" and (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L19	0	L9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" same (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L20	260	L9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" and (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L21	8168	L9 and (stream$3 or video) and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L22	191	L9 and @ad<"20040430" and (upshift$3 or upconvert$3 or downshift$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L23	10	L22 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L24	1	L22 and "709"/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L25	1	L22 and "709"/230-234.ccls.	US-PGPUB; USPAT; USOCR;	OR	ON	2014/07/31 00:31

			FPRS; EPO; JPO; DERWENT; IBM_TDB			
L26	20509	"709"/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L27	42	"709"/230-234.ccls. and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with strem$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L28	42	"709"/230-234.ccls. and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L29	18	(real near2 network).as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L30	202	(sure-stream or sure adj stream or surestream)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L31	45	(sure-stream or surestream)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L32	4	"7054948".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L33	1	"7054948".pn. and surestream	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L34	11	"7406527".pn.	US-PGPUB;	OR	ON	2014/07/31

			USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			00:31
L36	2259	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L37	2259	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L39	107	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L40	0	L39 and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L41	0	L39 and (upconvert$3 or upshift$3 or downshift$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L42	140	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L43	0	((dynamically near3 chang$4) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L44	4	((dynamically near3 chang$4) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2014/07/31 00:31

		stream$3)) same server	DERWENT; IBM_TDB			
L45	23	((dynamically) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L46	0	((dynamical) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L47	46	((dynamic$5) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L48	23	L47 not L45	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L49	927	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L50	166	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L51	88	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4)) and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L52	32	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2014/07/31 00:31

		(@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4)) and "709"/230-234.ccls.	DERWENT; IBM_TDB			
L53	20509	"709""/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L54	3562	(multi-rate and (stream$3 or (video with stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L55	49	(multi-rate and (stream$3 or (video with stream$3))).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L56	8	("20020174434" | "20030067872" | "20040030797").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L57	44	("20020174434" | "20030067872" | "20040030797").PN. or ("20020073167" | "20050123058" | "20070024705" | "20080028428" | "20080056373" | "6091775" | "6091777" | "6510553" | "6731600" | "6732183" | "6968387" | "7075986" | "7349976").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L58	21	L57 and (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L59	1	L57 and (stream$3 or (video with stream$3)) and ((plurality or many or multiple) near5 (version or format or cop$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L60	4	L57 and (stream$3 or (video with stream$3)) and ((plurality or many or multiple) near5 (version or format or cop$3 or quality or (high and low)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2014/07/31 00:31

			IBM_TDB			
L61	1166673	(plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L62	1235	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)) with ((automatic$4 or dynamic$5) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L63	138	L62 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L64	68	L62 and "709"/$$$.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L65	5	L62 and "709"/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L66	739	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)) with ((automatic$4) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L67	740	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5 or (bit adj2 rate or bit-rate))) with ((automatic$4) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L68	5	L67 and "709"/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L69	23672	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near8 automat$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO;	OR	ON	2014/07/31 00:31

			JPO; DERWENT; IBM_TDB			
L70	13352	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L71	73	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L72	0	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/231.ccls. and (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L73	3	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/230-234.ccls. and (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L74	14464	709/231-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L75	6072	709/231-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L76	851	L75 and ((multiple or plurality or many or three) near5 files)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L77	733	L75 and ((multiple or plurality or many or three) near3 files)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L78	276	L75 and ((multiple or plurality or many or three) near3 copies)	US-PGPUB; USPAT;	OR	ON	2014/07/31 00:31

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
L79	1	L75 and ((multiple or plurality or many or three) near3 copies) near5 (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L80	18	L75 and ((multiple or plurality or many or three) near3 (copies or files)) near5 (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L81	1	L75 and ((multiple or plurality or many or three) near3 (copies))with (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L82	1	L75 and ((multiple or plurality or many or three) near3 (copies)) with (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L83	276	L75 and ((multiple or plurality or many or three) near3 (copies))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L84	1	L75 and ((multiple or plurality or many or three) near3 (copies)) near5 (bit adj3 rate)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L85	8	@ad<"20050408" and ((multiple or plurality or many or three) near3 (copies)) near5 (bit adj3 rate)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L86	898	(dynamic near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31

L87	315	(dynamic near5 stream$3).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L88	16	(dynamic near5 stream$3).ti. and @ad<"20050408" and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L89	3	"5832300".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L90	0	(dautomatic$4 near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L91	727	(automatic$4 near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L92	358	@ad<"20050408" and L91	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L93	0	(automatic$4 near5 stream$3 with (upshif$3 or downshif$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L94	0	(automatic$4 near5 stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L95	0	(automatic$4 with stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2014/07/31 00:31

			DERWENT; IBM_TDB			
L96	5	(stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L97	0	(stream$3 with (video with stream$3)).ti. and (automatic$4 near4 switch$3 near4 different near4 quality)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L98	0	(stream$3 with (video with stream$3)) and (automatic$4 near4 switch$3 near4 different near4 quality)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L99	2150270	(dynamic$3 and switch$3 and stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L100	523542	(dynamic$3 and switch$3 and stream$3 or video).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L101	1	(dynamic$3 and switch$3 and stream$3 and video).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L102	1	(dynamic$3 and switch$3 and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L103	1	(dynamic$3 and (switch$3 or shivt$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L104	13	(automatic$4 and (switch$3 or shivt$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR;	OR	ON	2014/07/31 00:31

			FPRS; EPO; JPO; DERWENT; IBM_TDB			
L105	14	(automatic$4 and (switch$3 or shift$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L106	241	(automatic$4 and (switch$3 or shift$3 or upshift$3) and (stream$3 and video))	DERWENT	OR	ON	2014/07/31 00:31
L107	9	(automatic$4 near5 (switch$3 or shift$3 or upshift$3) near5 (stream$3 and video))	DERWENT	OR	ON	2014/07/31 00:31
L108	2	(automatic$4 near5 (copies or files or versions) near5 (stream$3 and video))	DERWENT	OR	ON	2014/07/31 00:31
L109	29	(multiple or many or three or plurality) near5 ((files or version) near5 (quality or bit-rate or bit adj rate))	DERWENT	OR	ON	2014/07/31 00:31
L110	945	(multiple or many or three or plurality) near5 ((files or version) near5 (quality or bit-rate or bit adj rate))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L111	117	L110 and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L112	221423	(multiple or many or three or plurality) near5 ((files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L113	2932	L112 and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L115	28	L112 and 709/230-234.ccls. and (upshift$3 or downshift$3 or upcovert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L116	12	L112 and 709/230-234.ccls. and	US-PGPUB;	OR	ON	2014/07/31

		((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality))	USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			00:31
L117	17	709/230-234.ccls. and ((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality with (video or stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L118	27	((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality with (video or stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L119	100080	microsoft.as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L120	1724	microsoft.as. and (stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L121	753	microsoft.as. and (stream$3).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L122	7	microsoft.as. and (stream$3 with quality).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L123	0	(automatica$3 near5 switch$3 near5 (different near5 qualit$5) near5 (files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L124	0	(automatic$5 near5 switch$3 near5 (different near5 qualit$5) near5 (files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2014/07/31 00:31

			IBM_TDB			
L125	0	(automatic$5 near5 switch$3 near5 ((different near5 qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L126	1	((automatic$5 near5 (switch$3 or upshift$3 or downshift$3 or upconvert$3)) with ((different near5 qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L127	6	((automatic$5 near5 (switch$3 or upshift$3 or downshift$3 or upconvert$3)) with ((qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L128	2	"20050262257".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L129	0	"20050262257".pn. and (ration or average)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L130	1	"20050262257".pn. and (ratio or average)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L131	0	"20050262257".pn. and (sub-parts or subparts)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L132	1	"20050262257".pn. and (safe$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L133	2	"2002174434".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO;	OR	ON	2014/07/31 00:31

			JPO; DERWENT; IBM_TDB			
L134	2	"20020174434".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L135	2	"20020144276".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L136	11	("20020078209" | "20040117459" | "20050021616" | "20060031523").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L137	1	("20020078209" | "20040117459" | "20050021616" | "20060031523").PN. and clip$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L138	1	"20020188745".pn. and play$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L139	0	"20020188745".pn. and tcp	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L140	0	"20020188745".pn. and http	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L141	1	"20020188745".pn. and web	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L142	2	"20020188745".pn. and internet	US-PGPUB; USPAT;	OR	ON	2014/07/31 00:31

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
L143	8046629	((plurality or mutiple or two or threee or many) near5 tcp mear5 connection)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L144	8046958	((plurality or multiple or two or threee or many) near5 tcp mear5 connection)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L145	4601	(((plurality or multiple or two or threee or many) near5 (tcp near5 connection)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L146	119	(((plurality or multiple or two or threee or many) near5 (tcp near5 connection))) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L147	64	(((plurality or multiple) near5 (tcp near5 connection))) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L148	0	"20020188745".pn. and cache	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L149	0	"20020188745".pn. and operator	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L150	1	"20020188745".pn. and separate	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31

L151	1	"20050262257".pn. and (cache)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L152	1	"20050262257".pn. and (saf$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L153	103	((R) near2 (Major)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2014/07/31 00:32
L154	19	((Mark) near2 (Hurst)).INV.	US-PGPUB	OR	ON	2014/07/31 00:32
L155	57	(709/246.ccls.) and L145	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:33
S1	899	(upconvert$3 or upshift$3) with (stream$3 or (video with strem$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:04
S2	899	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:04
S3	0	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) same ((multiple or plurality or many or three or two) near3 copi$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:05
S4	2	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) same ((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:06
S5	85	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) and ("709"/$$$.ccls. or "725"/$$$.ccls.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:06

S6	81	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) and ("709"/$$$.ccls. or "725"/$$$.ccls.) and network	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:06
S7	12707	((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)) same (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:08
S8	441696	((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:08
S9	12707	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:08
S10	1044	S9 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:09
S11	1044	S9 and "709"/$$$.ccls. and (stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:09
S12	299	S9 and "709"/231.ccls. and (stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:09
S13	6	("20020174434" | "20030067872" | "20040030797").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:12
S14	32	("20020174434" | "20030067872" | "20040030797").PN. or ("20020073167" | "20050123058" | "20070024705" | "20080028428" | "20080056373" | "6091775" |	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2009/02/09 17:12

		"6091777" | "6510553" | "6731600" | "6732183" | "6968387" | "7075986" | "7349976").PN.	DERWENT; IBM_TDB			
S15	21	S14 and (Stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:13
S16	5	S14 and (Stream$3 or (video with stream$3)) and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:13
S17	229	S9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:15
S18	229	S9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" and (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:15
S19	0	S9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" same (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:16
S20	229	S9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" and (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:16
S21	7262	S9 and (stream$3 or video) and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:18
S22	186	S9 and @ad<"20040430" and (upshift$3 or upconvert$3 or downshift$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:19
S23	10	S22 and "709"/$$$.ccls.	US-PGPUB; USPAT;	OR	ON	2009/02/09 17:19

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S24	1	S22 and "709"/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:20
S25	1	S22 and "709"/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:20
S26	7913	"709"/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:20
S27	7	"709"/230-234.ccls. and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with strem$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:21
S28	7	"709"/230-234.ccls. and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:22
S29	14	(real near2 network).as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:32
S30	142	(sure-stream or sure adj stream or surestream)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:35
S31	26	(sure-stream or surestream)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:35

S32	2	"7054948".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:37
S33	1	"7054948".pn. and surestream	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:37
S34	9	"7406527".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:01
S35	855	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:02
S36	855	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:03
S37	855	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:03
S38	855	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/231.ccls. and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:03
S39	94	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:04
S40	0	S39 and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2009/02/09 18:08

			DERWENT; IBM_TDB			
S41	0	S39 and (upconvert$3 or upshift$3 or downshift$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:08
S42	121	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:25
S43	0	((dynamically near3 chang$4) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:30
S44	0	((dynamically near3 chang$4) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:31
S45	2	((dynamically) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:33
S46	0	((dynamical) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:34
S47	5	((dynamic$5) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:34
S48	3	S47 not S45	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:36
S49	782	(((multiple or plurality or many or three or two) near5 (version or	US-PGPUB; USPAT;	OR	ON	2009/02/09 19:45

		format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430")	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S50	146	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:56
S51	78	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4)) and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:58
S52	29	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4)) and "709"/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 20:01
S53	7913	"709""/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 20:02
S54	1660	(multi-rate and (stream$3 or (video with stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 20:04
S55	11	(multi-rate and (stream$3 or (video with stream$3))).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 20:04
S56	6	("20020174434" | "20030067872" | "20040030797").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:44

S57	32	("20020174434" | "20030067872" | "20040030797").PN. or ("20020073167" | "20050123058" | "20070024705" | "20080028428" | "20080056373" | "6091775" | "6091777" | "6510553" | "6731600" | "6732183" | "6968387" | "7075986" | "7349976").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:45
S58	21	S57 and (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:45
S59	1	S57 and (stream$3 or (video with stream$3)) and ((plurality or many or multiple) near5 (version or format or cop$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:46
S60	4	S57 and (stream$3 or (video with stream$3)) and ((plurality or many or multiple) near5 (version or format or cop$3 or quality or (high and low)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:47
S61	670510	(plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:51
S62	662	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)) with ((automatic$4 or dynamic$5) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:52
S64	72	S62 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:02
S65	62	S62 and "709"/$$$.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:05
S66	5	S62 and "709"/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO;	OR	ON	2009/02/11 15:06

			JPO; DERWENT; IBM_TDB			
S67	424	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)) with ((automatic$4) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:07
S68	425	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5 or (bit adj2 rate or bit-rate))) with ((automatic$4) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:08
S69	5	S68 and "709"/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:09
S70	12589	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near8 automat$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:10
S71	6937	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:10
S72	26	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:11
S73	0	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/231.ccls. and (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:11
S74	2	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/230-234.ccls. and (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:12
S75	5236	709/231-234.ccls.	US-PGPUB; USPAT;	OR	ON	2009/02/11 15:12

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S76	4485	709/231-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:12
S77	691	S76 and ((multiple or plurality or many or three) near5 files)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:13
S78	591	S76 and ((multiple or plurality or many or three) near3 files)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:14
S79	233	S76 and ((multiple or plurality or many or three) near3 copies)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:14
S80	1	S76 and ((multiple or plurality or many or three) near3 copies) near5 (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:15
S81	9	S76 and ((multiple or plurality or many or three) near3 (copies or files)) near5 (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:25
S82	1	S76 and ((multiple or plurality or many or three) near3 (copies))with (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:27
S83	1	S76 and ((multiple or plurality or many or three) near3 (copies)) with (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:27

S84	233	S76 and ((multiple or plurality or many or three) near3 (copies))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:28
S85	1	S76 and ((multiple or plurality or many or three) near3 (copies)) near5 (bit adj3 rate)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:28
S86	6	@ad<"20050408" and ((multiple or plurality or many or three) near3 (copies)) near5 (bit adj3 rate)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:29
S87	251	(dynamic near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:32
S88	169	(dynamic near5 stream$3).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:34
S89	10	(dynamic near5 stream$3).ti. and @ad<"20050408" and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:34
S90	2	"5832300".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:40
S91	0	(dautomatic$4 near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:41
S92	326	(automatic$4 near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2009/02/11 15:41

			DERWENT; IBM_TDB			
S93	241	@ad<"20050408" and S92	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:42
S94	0	(automatic$4 near5 stream$3 with (upshif$3 or downshif$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:43
S95	0	(automatic$4 near5 stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:43
S96	0	(automatic$4 with stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:43
S97	3	(stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:43
S98	0	(stream$3 with (video with stream$3)).ti. and (automatic$4 near4 switch$3 near4 different near4 quality)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:52
S99	0	(stream$3 with (video with stream$3)) and (automatic$4 near4 switch$3 near4 different near4 quality)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:52
S100	1081374	(dynamic$3 and switch$3 and stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:57
S101	271819	(dynamic$3 and switch$3 and stream$3 or video).ti.	US-PGPUB; USPAT; USOCR;	OR	ON	2009/02/11 15:58

			FPRS; EPO; JPO; DERWENT; IBM_TDB			
S102	0	(dynamic$3 and switch$3 and stream$3 and video).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:58
S103	0	(dynamic$3 and switch$3 and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:59
S104	0	(dynamic$3 and (switch$3 or shivt$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:59
S105	5	(automatic$4 and (switch$3 or shivt$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:59
S106	6	(automatic$4 and (switch$3 or shift$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:59
S107	108	(automatic$4 and (switch$3 or shift$3 or upshift$3) and (stream$3 and video))	DERWENT	OR	ON	2009/02/11 16:00
S108	6	(automatic$4 near5 (switch$3 or shift$3 or upshift$3) near5 (stream$3 and video))	DERWENT	OR	ON	2009/02/11 16:02
S109	1	(automatic$4 near5 (copies or files or versions) near5 (stream$3 and video))	DERWENT	OR	ON	2009/02/11 16:02
S110	11	(multiple or many or three or plurality) near5 ((files or version) near5 (quality or bit-rate or bit adj rate))	DERWENT	OR	ON	2009/02/11 16:03
S111	365	(multiple or many or three or plurality) near5 ((files or version) near5 (quality or bit-rate or bit adj rate))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:04
S112	37	S111 and 709/230-234.ccls.	US-PGPUB;	OR	ON	2009/02/11

			USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			16:04
S113	108249	(multiple or many or three or plurality) near5 ((files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:10
S114	1240	S113 and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:11
S115	1038	S113 and 709/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:11
S116	11	S113 and 709/230-234.ccls. and (upshift$3 or downshift$3 or upcovert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:12
S117	3	S113 and 709/230-234.ccls. and ((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:14
S118	3	709/230-234.ccls. and ((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality with (video or stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:14
S119	7	((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality with (video or stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:15
S120	50419	microsoft.as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2009/02/11 16:15

			IBM_TDB			
S121	981	microsoft.as. and (stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:16
S122	650	microsoft.as. and (stream$3).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:16
S123	6	microsoft.as. and (stream$3 with quality).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:17
S124	0	(automatica$3 near5 switch$3 near5 (different near5 qualit$5) near5 (files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:19
S125	0	(automatic$5 near5 switch$3 near5 (different near5 qualit$5) near5 (files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:20
S126	0	(automatic$5 near5 switch$3 near5 ((different near5 qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:20
S127	0	((automatic$5 near5 (switch$3 or upshift$3 or downshift$3 or upconvert$3)) with ((different near5 qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:20
S128	6	((automatic$5 near5 (switch$3 or upshift$3 or downshift$3 or upconvert$3)) with ((qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:22
S129	2	"20050262257".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO;	OR	ON	2009/02/13 17:34

			JPO; DERWENT; IBM_TDB			
S130	0	"20050262257".pn. and (ration or average)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:38
S131	1	"20050262257".pn. and (ratio or average)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:39
S132	0	"20050262257".pn. and (sub-parts or subparts)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:39
S133	1	"20050262257".pn. and (safe$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:43
S134	2	"2002174434".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:49
S135	2	"20020174434".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:49
S136	2	"20020144276".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:51
S137	8	("20020078209" | "20040117459" | "20050021616" | "20060031523").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 18:33
S138	1	("20020078209" | "20040117459" | "20050021616" | "20060031523").PN.	US-PGPUB; USPAT;	OR	ON	2009/02/13 18:33

		and clip$4	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S139	1	"20020188745".pn. and play$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 18:37
S140	0	"20020188745".pn. and tcp	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:00
S141	0	"20020188745".pn. and http	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:00
S142	1	"20020188745".pn. and web	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:00
S143	2	"20020188745".pn. and internet	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:01
S144	4765975	((plurality or mutiple or two or threee or many) near5 tcp mear5 connection)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:03
S145	4766104	((plurality or multiple or two or threee or many) near5 tcp mear5 connection)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:03
S146	2131	(((plurality or multiple or two or threee or many) near5 (tcp near5 connection)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:03

S147	30	(((plurality or multiple or two or threee or many) near5 (tcp near5 connection))) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:04
S148	12	(((plurality or multiple) near5 (tcp near5 connection))) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:07
S149	0	"20020188745".pn. and cache	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:22
S150	0	"20020188745".pn. and operator	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:22
S151	1	"20020188745".pn. and separate	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:25
S152	1	"20050262257".pn. and (cache)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:25
S153	1	"20050262257".pn. and (saf$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:36

EAST Search History (Interference)

<This search history is empty>

7/31/2014 12:45:06 AM
C:\Users\ndonabed\Desktop\Workspaces\11116783_amendment.wsp

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	4705	(upconvert$3 or upshift$3) with (stream$3 or (video with strem$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L2	4705	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L3	0	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) same ((multiple or plurality or many or three or two) near3 copi$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L4	16	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) same ((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L5	253	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) and ("709"/$$$.ccls. or "725"/$$$.ccls.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L6	244	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) and ("709"/$$$.ccls. or "725"/$$$.ccls.) and network	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L7	23030	((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)) same (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L8	756102	((multiple or plurality or many or	US-PGPUB;	OR	ON	2014/07/31

		three or two) near5 (version or format$4 or copies or cop$4))	USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			00:31
L9	23030	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L10	2345	L9 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L11	2345	L9 and "709"/$$$.ccls. and (stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L12	738	L9 and "709"/231.ccls. and (stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L13	8	("20020174434" | "20030067872" | "20040030797").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L14	44	("20020174434" | "20030067872" | "20040030797").PN. or ("20020073167" | "20050123058" | "20070024705" | "20080028428" | "20080056373" | "6091775" | "6091777" | "6510553" | "6731600" | "6732183" | "6968387" | "7075986" | "7349976").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L15	21	L14 and (Stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L16	5	L14 and (Stream$3 or (video with stream$3)) and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2014/07/31 00:31

			DERWENT; IBM_TDB			
L17	260	L9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L18	260	L9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" and (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L19	0	L9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" same (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L20	260	L9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" and (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L21	8168	L9 and (stream$3 or video) and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L22	191	L9 and @ad<"20040430" and (upshift$3 or upconvert$3 or downshift$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L23	10	L22 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L24	1	L22 and "709"/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L25	1	L22 and "709"/230-234.ccls.	US-PGPUB; USPAT; USOCR;	OR	ON	2014/07/31 00:31

			FPRS; EPO; JPO; DERWENT; IBM_TDB			
L26	20509	"709"/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L27	42	"709"/230-234.ccls. and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with strem$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L28	42	"709"/230-234.ccls. and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L29	18	(real near2 network).as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L30	202	(sure-stream or sure adj stream or surestream)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L31	45	(sure-stream or surestream)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L32	4	"7054948".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L33	1	"7054948".pn. and surestream	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L34	11	"7406527".pn.	US-PGPUB;	OR	ON	2014/07/31

			USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			00:31
L35	2259	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L36	2259	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L37	2259	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L38	2259	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/231.ccls. and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L39	107	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L40	0	L39 and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L41	0	L39 and (upconvert$3 or upshift$3 or downshift$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L42	140	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (709/231.ccls. and	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2014/07/31 00:31

		@ad<"20040430")	IBM_TDB			
L43	0	((dynamically near3 chang$4) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L44	4	((dynamically near3 chang$4) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L45	23	((dynamically) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L46	0	((dynamical) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L47	46	((dynamic$5) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L48	23	L47 not L45	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L49	927	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L50	166	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L51	88	(((multiple or plurality or many or	US-PGPUB;	OR	ON	2014/07/31

		three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4)) and "709"/$$$.ccls.	USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			00:31
L52	32	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4)) and "709"/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L53	20509	"709""/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L54	3562	(multi-rate and (stream$3 or (video with stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L55	49	(multi-rate and (stream$3 or (video with stream$3))).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L56	8	("20020174434" | "20030067872" | "20040030797").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L57	44	("20020174434" | "20030067872" | "20040030797").PN. or ("20020073167" | "20050123058" | "20070024705" | "20080028428" | "20080056373" | "6091775" | "6091777" | "6510553" | "6731600" | "6732183" | "6968387" | "7075986" | "7349976").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L58	21	L57 and (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31

L59	1	L57 and (stream$3 or (video with stream$3)) and ((plurality or many or multiple) near5 (version or format or cop$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L60	4	L57 and (stream$3 or (video with stream$3)) and ((plurality or many or multiple) near5 (version or format or cop$3 or quality or (high and low)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L61	1166673	(plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L62	1235	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)) with ((automatic$4 or dynamic$5) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L63	138	L62 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L64	68	L62 and "709"/$$$.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L65	5	L62 and "709"/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L66	739	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)) with ((automatic$4) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L67	740	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5 or (bit adj2 rate or bit-rate))) with ((automatic$4) near5 (switch$3 or	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2014/07/31 00:31

		chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	DERWENT; IBM_TDB			
L68	5	L67 and "709"/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L69	23672	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near8 automat$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L70	13352	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L71	73	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L72	0	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/231.ccls. and (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L73	3	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/230-234.ccls. and (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L74	14464	709/231-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L75	6072	709/231-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L76	851	L75 and ((multiple or plurality or many or three) near5 files)	US-PGPUB; USPAT; USOCR;	OR	ON	2014/07/31 00:31

			FPRS; EPO; JPO; DERWENT; IBM_TDB			
L77	733	L75 and ((multiple or plurality or many or three) near3 files)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L78	276	L75 and ((multiple or plurality or many or three) near3 copies)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L79	1	L75 and ((multiple or plurality or many or three) near3 copies) near5 (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L80	18	L75 and ((multiple or plurality or many or three) near3 (copies or files)) near5 (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L81	1	L75 and ((multiple or plurality or many or three) near3 (copies))with (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L82	1	L75 and ((multiple or plurality or many or three) near3 (copies)) with (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L83	276	L75 and ((multiple or plurality or many or three) near3 (copies))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L84	1	L75 and ((multiple or plurality or many or three) near3 (copies)) near5 (bit adj3 rate)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L85	8	@ad<"20050408" and ((multiple or	US-PGPUB;	OR	ON	2014/07/31

		plurality or many or three) near3 (copies)) near5 (bit adj3 rate)	USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			00:31
L86	898	(dynamic near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L87	315	(dynamic near5 stream$3).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L88	16	(dynamic near5 stream$3).ti. and @ad<"20050408" and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L89	3	"5832300".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L90	0	(dautomatic$4 near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L91	727	(automatic$4 near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L92	358	@ad<"20050408" and L91	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L93	0	(automatic$4 near5 stream$3 with (upshif$3 or downshif$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2014/07/31 00:31

			IBM_TDB			
L94	0	(automatic$4 near5 stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L95	0	(automatic$4 with stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L96	5	(stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L97	0	(stream$3 with (video with stream$3)).ti. and (automatic$4 near4 switch$3 near4 different near4 quality)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L98	0	(stream$3 with (video with stream$3)) and (automatic$4 near4 switch$3 near4 different near4 quality)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L99	2150270	(dynamic$3 and switch$3 and stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L100	523542	(dynamic$3 and switch$3 and stream$3 or video).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L101	1	(dynamic$3 and switch$3 and stream$3 and video).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L102	1	(dynamic$3 and switch$3 and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO;	OR	ON	2014/07/31 00:31

			JPO; DERWENT; IBM_TDB			
L103	1	(dynamic$3 and (switch$3 or shivt$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L104	13	(automatic$4 and (switch$3 or shivt$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L105	14	(automatic$4 and (switch$3 or shift$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L106	241	(automatic$4 and (switch$3 or shift$3 or upshift$3) and (stream$3 and video))	DERWENT	OR	ON	2014/07/31 00:31
L107	9	(automatic$4 near5 (switch$3 or shift$3 or upshift$3) near5 (stream$3 and video))	DERWENT	OR	ON	2014/07/31 00:31
L108	2	(automatic$4 near5 (copies or files or versions) near5 (stream$3 and video))	DERWENT	OR	ON	2014/07/31 00:31
L109	29	(multiple or many or three or plurality) near5 ((files or version) near5 (quality or bit-rate or bit adj rate))	DERWENT	OR	ON	2014/07/31 00:31
L110	945	(multiple or many or three or plurality) near5 ((files or version) near5 (quality or bit-rate or bit adj rate))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L111	117	L110 and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L112	221423	(multiple or many or three or plurality) near5 ((files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L113	2932	L112 and 709/230-234.ccls.	US-PGPUB; USPAT;	OR	ON	2014/07/31 00:31

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
L114	1281	L112 and 709/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L115	28	L112 and 709/230-234.ccls. and (upshift$3 or downshift$3 or upcovert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L116	12	L112 and 709/230-234.ccls. and ((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L117	17	709/230-234.ccls. and ((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality with (video or stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L118	27	((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality with (video or stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L119	100080	microsoft.as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L120	1724	microsoft.as. and (stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L121	753	microsoft.as. and (stream$3).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31

L122	7	microsoft.as. and (stream$3 with quality).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L123	0	(automatica$3 near5 switch$3 near5 (different near5 qualit$5) near5 (files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L124	0	(automatic$5 near5 switch$3 near5 (different near5 qualit$5) near5 (files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L125	0	(automatic$5 near5 switch$3 near5 ((different near5 qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L126	1	((automatic$5 near5 (switch$3 or upshift$3 or downshift$3 or upconvert$3)) with ((different near5 qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L127	6	((automatic$5 near5 (switch$3 or upshift$3 or downshift$3 or upconvert$3)) with ((qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L128	2	"20050262257".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L129	0	"20050262257".pn. and (ration or average)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L130	1	"20050262257".pn. and (ratio or average)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2014/07/31 00:31

			DERWENT; IBM_TDB			
L131	0	"20050262257".pn. and (sub-parts or subparts)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L132	1	"20050262257".pn. and (safe$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L133	2	"2002174434".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L134	2	"20020174434".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L135	2	"20020144276".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L136	11	("20020078209" | "20040117459" | "20050021616" | "20060031523").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L137	1	("20020078209" | "20040117459" | "20050021616" | "20060031523").PN. and clip$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L138	1	"20020188745".pn. and play$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L139	0	"20020188745".pn. and tcp	US-PGPUB; USPAT; USOCR;	OR	ON	2014/07/31 00:31

			FPRS; EPO; JPO; DERWENT; IBM_TDB			
L140	0	"20020188745".pn. and http	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L141	1	"20020188745".pn. and web	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L142	2	"20020188745".pn. and internet	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L143	8046629	((plurality or mutiple or two or threee or many) near5 tcp mear5 connection)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L144	8046958	((plurality or multiple or two or threee or many) near5 tcp mear5 connection)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L145	4601	(((plurality or multiple or two or threee or many) near5 (tcp near5 connection)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L146	119	(((plurality or multiple or two or threee or many) near5 (tcp near5 connection))) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L147	64	(((plurality or multiple) near5 (tcp near5 connection))) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L148	0	"20020188745".pn. and cache	US-PGPUB;	OR	ON	2014/07/31

			USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			00:31
L149	0	"20020188745".pn. and operator	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L150	1	"20020188745".pn. and separate	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L151	1	"20050262257".pn. and (cache)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L152	1	"20050262257".pn. and (saf$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:31
L153	103	((R) near2 (Major)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2014/07/31 00:32
L154	19	((Mark) near2 (Hurst)).INV.	US-PGPUB	OR	ON	2014/07/31 00:32
L155	57	(709/246.ccls.) and L145	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 00:33
L156	67	((Mark) near2 (Hurst)).INV.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 01:14
L157	1	((Mark) near2 (Hurst)).INV. and tcp.clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/07/31 01:15

L158	103	((R) near2 (Major)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2014/07/31 01:15
S1	899	(upconvert$3 or upshift$3) with (stream$3 or (video with strem$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:04
S2	899	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:04
S3	0	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) same ((multiple or plurality or many or three or two) near3 copi$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:05
S4	2	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) same ((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:06
S5	85	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) and ("709"/$$$.ccls. or "725"/$$$.ccls.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:06
S6	81	(upconvert$3 or upshift$3) with (stream$3 or (video with stream$3)) and ("709"/$$$.ccls. or "725"/$$$.ccls.) and network	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:06
S7	12707	((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)) same (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:08
S8	441696	((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:08
S9	12707	(((multiple or plurality or many or three or two) near5 (version or	US-PGPUB; USPAT;	OR	ON	2009/02/09 17:08

		format$4 or copies or cop$4))) same (stream$3 or (video with stream$3))	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S10	1044	S9 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:09
S11	1044	S9 and "709"/$$$.ccls. and (stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:09
S12	299	S9 and "709"/231.ccls. and (stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:09
S13	6	("20020174434" | "20030067872" | "20040030797").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:12
S14	32	("20020174434" | "20030067872" | "20040030797").PN. or ("20020073167" | "20050123058" | "20070024705" | "20080028428" | "20080056373" | "6091775" | "6091777" | "6510553" | "6731600" | "6732183" | "6968387" | "7075986" | "7349976").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:12
S15	21	S14 and (Stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:13
S16	5	S14 and (Stream$3 or (video with stream$3)) and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:13
S17	229	S9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2009/02/09 17:15

			IBM_TDB			
S18	229	S9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" and (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:15
S19	0	S9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" same (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:16
S20	229	S9 and "709"/231.ccls. and (stream$3 or video) and @ad<"20040430" and (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:16
S21	7262	S9 and (stream$3 or video) and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:18
S22	186	S9 and @ad<"20040430" and (upshift$3 or upconvert$3 or downshift$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:19
S23	10	S22 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:19
S24	1	S22 and "709"/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:20
S25	1	S22 and "709"/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:20
S26	7913	"709"/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO;	OR	ON	2009/02/09 17:20

			JPO; DERWENT; IBM_TDB			
S27	7	"709"/230-234.ccls. and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with strem$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:21
S28	7	"709"/230-234.ccls. and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:22
S29	14	(real near2 network).as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:32
S30	142	(sure-stream or sure adj stream or surestream)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:35
S31	26	(sure-stream or surestream)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:35
S32	2	"7054948".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:37
S33	1	"7054948".pn. and surestream	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 17:37
S34	9	"7406527".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:01
S35	855	(((multiple or plurality or many or three or two) near5 (version or	US-PGPUB; USPAT;	OR	ON	2009/02/09 18:02

		format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S36	855	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:03
S37	855	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:03
S38	855	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and709/231.ccls. and @ad<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:03
S39	94	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:04
S40	0	S39 and (upconvert$3 or upshift$3 or downshift$3) with (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:08
S41	0	S39 and (upconvert$3 or upshift$3 or downshift$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 18:08
S42	121	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (709/231.ccls. and @ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:25
S43	0	((dynamically near3 chang$4) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server and (709/231.ccls. and	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:30

		@ad<"20040430")				
S44	0	((dynamically near3 chang$4) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:31
S45	2	((dynamically) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:33
S46	0	((dynamical) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:34
S47	5	((dynamic$5) with (((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4)))) same3 (stream$3 or (video with stream$3)) same server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:34
S48	3	S47 not S45	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:36
S49	782	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:45
S50	146	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:56
S51	78	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4)) and	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 19:58

		"709"/$$$.ccls.				
S52	29	(((multiple or plurality or many or three or two) near5 (version or format$4 or copies or cop$4))) same3 (stream$3 or (video with stream$3)) same server and (@ad<"20040430") and ((dynamic$5 or automatical$5) near3 (upshift$3 or downshift$3 or upconvert$3 or downcovnert$3 or chan$4)) and "709"/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 20:01
S53	7913	"709""/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 20:02
S54	1660	(multi-rate and (stream$3 or (video with stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 20:04
S55	11	(multi-rate and (stream$3 or (video with stream$3))).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/09 20:04
S56	6	("20020174434" | "20030067872" | "20040030797").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:44
S57	32	("20020174434" | "20030067872" | "20040030797").PN. or ("20020073167" | "20050123058" | "20070024705" | "20080028428" | "20080056373" | "6091775" | "6091777" | "6510553" | "6731600" | "6732183" | "6968387" | "7075986" | "7349976").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:45
S58	21	S57 and (stream$3 or (video with stream$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:45
S59	1	S57 and (stream$3 or (video with stream$3)) and ((plurality or many or multiple) near5 (version or format or cop$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:46

S60	4	S57 and (stream$3 or (video with stream$3)) and ((plurality or many or multiple) near5 (version or format or cop$3 or quality or (high and low)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:47
S61	670510	(plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:51
S62	662	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)) with ((automatic$4 or dynamic$5) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 14:52
S64	72	S62 and "709"/$$$.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:02
S65	62	S62 and "709"/$$$.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:05
S66	5	S62 and "709"/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:06
S67	424	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5)) with ((automatic$4) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:07
S68	425	((plurality or many or three or two or multiple) near5 (file or cop$3 or version or format$3 or compress$5 or (bit adj2 rate or bit-rate))) with ((automatic$4) near5 (switch$3 or chang$3 or upshift$3 or downshift$3 or upconvert$3 or downconvert$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:08
S69	5	S68 and "709"/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2009/02/11 15:09

			DERWENT; IBM_TDB			
S70	12589	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near8 automat$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:10
S71	6937	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:10
S72	26	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:11
S73	0	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/231.ccls. and (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:11
S74	2	(stream$3 or (video with stream$3)) and ((plurality or many or multiple) near4 automat$5) and 709/230-234.ccls. and (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:12
S75	5236	709/231-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:12
S76	4485	709/231-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:12
S77	691	S76 and ((multiple or plurality or many or three) near5 files)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:13
S78	591	S76 and ((multiple or plurality or many or three) near3 files)	US-PGPUB; USPAT; USOCR;	OR	ON	2009/02/11 15:14

			FPRS; EPO; JPO; DERWENT; IBM_TDB			
S79	233	S76 and ((multiple or plurality or many or three) near3 copies)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:14
S80	1	S76 and ((multiple or plurality or many or three) near3 copies) near5 (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:15
S81	9	S76 and ((multiple or plurality or many or three) near3 (copies or files)) near5 (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:25
S82	1	S76 and ((multiple or plurality or many or three) near3 (copies))with (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:27
S83	1	S76 and ((multiple or plurality or many or three) near3 (copies)) with (encod$3 or bit-rate or qualilty)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:27
S84	233	S76 and ((multiple or plurality or many or three) near3 (copies))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:28
S85	1	S76 and ((multiple or plurality or many or three) near3 (copies)) near5 (bit adj3 rate)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:28
S86	6	@ad<"20050408" and ((multiple or plurality or many or three) near3 (copies)) near5 (bit adj3 rate)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:29
S87	251	(dynamic near5 stream$3).ti.	US-PGPUB;	OR	ON	2009/02/11

			USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			15:32
S88	169	(dynamic near5 stream$3).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:34
S89	10	(dynamic near5 stream$3).ti. and @ad<"20050408" and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:34
S90	2	"5832300".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:40
S91	0	(dautomatic$4 near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:41
S92	326	(automatic$4 near5 stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:41
S93	241	@ad<"20050408" and S92	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:42
S94	0	(automatic$4 near5 stream$3 with (upshif$3 or downshif$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:43
S95	0	(automatic$4 near5 stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2009/02/11 15:43

			IBM_TDB			
S96	0	(automatic$4 with stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:43
S97	3	(stream$3 with (upshift$3 or downshift$3 or upconvert$3 or downconvert$3)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:43
S98	0	(stream$3 with (video with stream$3)).ti. and (automatic$4 near4 switch$3 near4 different near4 quality)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:52
S99	0	(stream$3 with (video with stream$3)) and (automatic$4 near4 switch$3 near4 different near4 quality)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:52
S100	1081374	(dynamic$3 and switch$3 and stream$3 or video)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:57
S101	271819	(dynamic$3 and switch$3 and stream$3 or video).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:58
S102	0	(dynamic$3 and switch$3 and stream$3 and video).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:58
S103	0	(dynamic$3 and switch$3 and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:59
S104	0	(dynamic$3 and (switch$3 or shivt$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO;	OR	ON	2009/02/11 15:59

			JPO; DERWENT; IBM_TDB			
S105	5	(automatic$4 and (switch$3 or shivt$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:59
S106	6	(automatic$4 and (switch$3 or shift$3 or upshift$3) and (stream$3 and video)).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 15:59
S107	108	(automatic$4 and (switch$3 or shift$3 or upshift$3) and (stream$3 and video))	DERWENT	OR	ON	2009/02/11 16:00
S108	6	(automatic$4 near5 (switch$3 or shift$3 or upshift$3) near5 (stream$3 and video))	DERWENT	OR	ON	2009/02/11 16:02
S109	1	(automatic$4 near5 (copies or files or versions) near5 (stream$3 and video))	DERWENT	OR	ON	2009/02/11 16:02
S110	11	(multiple or many or three or plurality) near5 ((files or version) near5 (quality or bit-rate or bit adj rate))	DERWENT	OR	ON	2009/02/11 16:03
S111	365	(multiple or many or three or plurality) near5 ((files or version) near5 (quality or bit-rate or bit adj rate))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:04
S112	37	S111 and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:04
S113	108249	(multiple or many or three or plurality) near5 ((files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:10
S114	1240	S113 and 709/230-234.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:11
S115	1038	S113 and 709/230-234.ccls. and @ad<"20050408"	US-PGPUB; USPAT;	OR	ON	2009/02/11 16:11

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S116	11	S113 and 709/230-234.ccls. and (upshift$3 or downshift$3 or upcovert$3 or downconvert$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:12
S117	3	S113 and 709/230-234.ccls. and ((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:14
S118	3	709/230-234.ccls. and ((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality with (video or stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:14
S119	7	((upshift$3 or downshift$3 or upcovert$3 or downconvert$3) with (quality with (video or stream$3)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:15
S120	50419	microsoft.as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:15
S121	981	microsoft.as. and (stream$3).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:16
S122	650	microsoft.as. and (stream$3).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:16
S123	6	microsoft.as. and (stream$3 with quality).ti. and @ad<"20050408"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:17

S124	0	(automatica$3 near5 switch$3 near5 (different near5 qualit$5) near5 (files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:19
S125	0	(automatic$5 near5 switch$3 near5 (different near5 qualit$5) near5 (files or version))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:20
S126	0	(automatic$5 near5 switch$3 near5 ((different near5 qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:20
S127	0	((automatic$5 near5 (switch$3 or upshift$3 or downshift$3 or upconvert$3)) with ((different near5 qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:20
S128	6	((automatic$5 near5 (switch$3 or upshift$3 or downshift$3 or upconvert$3)) with ((qualit$5) near5 (files or version)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/11 16:22
S129	2	"20050262257".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:34
S130	0	"20050262257".pn. and (ration or average)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:38
S131	1	"20050262257".pn. and (ratio or average)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:39
S132	0	"20050262257".pn. and (sub-parts or subparts)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2009/02/13 17:39

			DERWENT; IBM_TDB			
S133	1	"20050262257".pn. and (safe$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:43
S134	2	"2002174434".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:49
S135	2	"20020174434".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:49
S136	2	"20020144276".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 17:51
S137	8	("20020078209" | "20040117459" | "20050021616" | "20060031523").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 18:33
S138	1	("20020078209" | "20040117459" | "20050021616" | "20060031523").PN. and clip$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 18:33
S139	1	"20020188745".pn. and play$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 18:37
S140	0	"20020188745".pn. and tcp	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:00
S141	0	"20020188745".pn. and http	US-PGPUB; USPAT; USOCR;	OR	ON	2009/02/13 19:00

			FPRS; EPO; JPO; DERWENT; IBM_TDB			
S142	1	"20020188745".pn. and web	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:00
S143	2	"20020188745".pn. and internet	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:01
S144	4765975	((plurality or mutiple or two or threee or many) near5 tcp mear5 connection)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:03
S145	4766104	((plurality or multiple or two or threee or many) near5 tcp mear5 connection)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:03
S146	2131	(((plurality or multiple or two or threee or many) near5 (tcp near5 connection)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:03
S147	30	(((plurality or multiple or two or threee or many) near5 (tcp near5 connection))) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:04
S148	12	(((plurality or multiple) near5 (tcp near5 connection))) and 709/231.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:07
S149	0	"20020188745".pn. and cache	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:22
S150	0	"20020188745".pn. and operator	US-PGPUB;	OR	ON	2009/02/13

			USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			19:22
S151	1	"20020188745".pn. and separate	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:25
S152	1	"20050262257".pn. and (cache)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:25
S153	1	"20050262257".pn. and (saf$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/02/13 19:36

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L161	0	(media player video playback multipl tcp connection subsequent).clm.	US-PGPUB; USPAT; UPAD	AND	ON	2014/07/31 01:19
L162	1	(media player video playback multiple tcp connection subsequent).clm.	US-PGPUB; USPAT; UPAD	AND	ON	2014/07/31 01:19
L163	1	(media player video playback multiple tcp connection subsequent).clm.	US-PGPUB; USPAT; UPAD	AND	ON	2014/07/31 01:19
L164	0	(media player video playback multiple tcp connection subsequent generat$5).clm.	US-PGPUB; USPAT; UPAD	AND	ON	2014/07/31 01:20

7/ 31/ 2014 1:20:27 AM
C:\ Users\ ndonabed\ Documents\ EAST\ Workspaces\ 11116783_amendment.wsp

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

110240 7590 08/12/2014
Ingrassia Fisher & Lorenz, P.C. (DISH Digital)
7010 East Cochise Road
Scottsdale, AZ 85253

EXAMINER
DONABED, NINOS

ART UNIT	PAPER NUMBER
2444	

DATE MAILED: 08/12/2014

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/116,783	04/28/2005	R. Drew Major	066.0141 (P2004-04-01.1)	5535

TITLE OF INVENTION: APPARATUS, SYSTEM, AND METHOD FOR ADAPTIVE-RATE SHIFTING OF STREAMING CONTENT

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0	$0	$480	11/12/2014

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

110240 7590 08/12/2014
Ingrassia Fisher & Lorenz, P.C. (DISH Digital)
7010 East Cochise Road
Scottsdale, AZ 85253

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/116,783	04/28/2005	R. Drew Major	066.0141 (P2004-04-01.1)	5535

TITLE OF INVENTION: APPARATUS, SYSTEM, AND METHOD FOR ADAPTIVE-RATE SHIFTING OF STREAMING CONTENT

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0	$0	$480	11/12/2014

EXAMINER	ART UNIT	CLASS-SUBCLASS
DONABED, NINOS	2444	709-231000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:
- ❑ Issue Fee
- ❑ Publication Fee (No small entity discount permitted)
- ❑ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ❑ A check is enclosed.
- ❑ Payment by credit card. Form PTO-2038 is attached.
- ❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ❑ Applicant certifying micro entity status. See 37 CFR 1.29 — NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.
- ❑ Applicant asserting small entity status. See 37 CFR 1.27 — NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.
- ❑ Applicant changing to regular undiscounted fee status. — NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

PTOL-85 Part B (10-13) Approved for use through 10/31/2013. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/116,783	04/28/2005	R. Drew Major	066.0141 (P2004-04-01.1)	5535

110240 7590 08/12/2014

Ingrassia Fisher & Lorenz, P.C. (DISH Digital)
7010 East Cochise Road
Scottsdale, AZ 85253

EXAMINER	
DONABED, NINOS	

ART UNIT	PAPER NUMBER
2444	

DATE MAILED: 08/12/2014

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 02/11)

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No. 11/116,783	Applicant(s) MAJOR ET AL.	
	Examiner NINOS DONABED	Art Unit 2444	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *board descision dated 7/23/2014*.
   ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on_____.
2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
3. ☒ The allowed claim(s) is/are *20,21,23-26,35-38 and 40-50*. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov .
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   Certified copies:

   a) ☐ All b) ☐ Some *c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
4. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
5. ☐ Examiner's Amendment/Comment
6. ☒ Examiner's Statement of Reasons for Allowance
7. ☐ Other _____.

/NINOS DONABED/
Primary Examiner, Art Unit 2444

1. The present application is being examined under the pre-AIA first to invent provisions.

Information Disclosure Statement

2. The information disclosure statement filed 7/29/2014 fails to comply with the provisions of 37 CFR 1.97, 1.98 and MPEP § 609 because prosecution is closed. It has been placed in the application file, but the information referred to therein has not been considered as to the merits. Applicant is advised that the date of any re-submission of any item of information contained in this information disclosure statement or the submission of any missing element(s) will be the date of submission for purposes of determining compliance with the requirements based on the time of filing the statement, including all certification requirements for statements under 37 CFR 1.97(e). See MPEP § 609.05(a).

REASONS FOR ALLOWANCE

The following is an examiner's statement of reasons for allowance: Claims 20,21,23-26,35-38 and 40-50 are allowable over the prior art of record: the closest prior art of record (Birney "Intelligent Streaming") does not teach or suggest in detail "streaming by a media player operating on an end user station a video from a set of one or more servers, wherein each of a plurality of different copies of the video encoded at different bit rates is stored as multiple files on the set of servers, wherein each of the multiple files yields a different portion of the video on playback, wherein the multiple files across the different copies yield the same portions of the video on playback, each of said files having a time index such that the files whose playback is the same portion of

the video for each of the different copies have the same time index in relation to the beginning of the video, and wherein the streaming comprises: requesting by the media player a plurality of sequential ones of the files of one of the copies from the set of servers over a plurality of Transmission Control Protocol (TCP) connections based on the time indexes; automatically requesting by the media player from the set of servers over the plurality of TCP connections subsequent portions of the video by requesting for each such portion one of the files from one of the copies dependent upon successive determinations by the media player to shift the playback quality to a higher or lower quality one of the different copies, said automatically requesting including, repeatedly generating a set of one or more factors indicative of the current ability to sustain the streaming of the video using the files from different ones of the copies, wherein the set of one or more factors relate to the performance of the network; and making the successive determinations to shift the playback quality based on at least one of the set of factors to achieve continuous playback of the video using the files of the highest quality one of the copies determined sustainable at that time; and presenting the video by playing back with the media player on the end user station the requested files in order of ascending playback time" in combination with all the elements of the independent claim as argued by the Applicant.

Birney teaches multiple copies of the same stream being stored on a server at different bit rates with the same time index used to switch between the stream based on network bandwidth limitations. Whereas, stated above, Applicant's claimed invention states "streaming by a media player operating on an end user station a video from a set

of one or more servers, wherein each of a plurality of different copies of the video encoded at different bit rates is stored as multiple files on the set of servers, wherein each of the multiple files yields a different portion of the video on playback, wherein the multiple files across the different copies yield the same portions of the video on playback, each of said files having a time index such that the files whose playback is the same portion of the video for each of the different copies have the same time index in relation to the beginning of the video, and wherein the streaming comprises: requesting by the media player a plurality of sequential ones of the files of one of the copies from the set of servers over a plurality of Transmission Control Protocol (TCP) connections based on the time indexes; automatically requesting by the media player from the set of servers over the plurality of TCP connections subsequent portions of the video by requesting for each such portion one of the files from one of the copies dependent upon successive determinations by the media player to shift the playback quality to a higher or lower quality one of the different copies, said automatically requesting including, repeatedly generating a set of one or more factors indicative of the current ability to sustain the streaming of the video using the files from different ones of the copies, wherein the set of one or more factors relate to the performance of the network; and making the successive determinations to shift the playback quality based on at least one of the set of factors to achieve continuous playback of the video using the files of the highest quality one of the copies determined sustainable at that time; and presenting the video by playing back with the media player on the end user station the requested files in order of ascending playback time." So as indicated in the above statements.

Applicant's arguments have been considered persuasive, in light of the claims limitation as well as the enabling portions of the specification.

Examiner also notes the board decision and reversal on 7/23/2014.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled “Comments on Statement of Reasons for Allowance.”

Any inquiry concerning this communication or earlier communications from the examiner should be directed to NINOS DONABED whose telephone number is (571)272-8757. The examiner can normally be reached on Monday-Friday, 7:30 AM-5:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Peter-Anthony Pappas can be reached on (571) 272-7646. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/NINOS DONABED/
Primary Examiner, Art Unit 2444

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11116783	MAJOR ET AL.
	Examiner	Art Unit
	NINOS DONABED	2444

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
709	246	05/07/2008	ND

SEARCH NOTES		
Search Notes	Date	Examiner
See EAST Search History	05/07/2008	ND
Conducted double patenting search	05/07/2008	ND
Consulted with SPE William Vaughn	11/13/2008	ND
See updated EAST search	12/13/2009	ND
Consulted with SPE William Vaughn regarding Final	02/12/2009	ND
Consulted with SPE William Vaughn regarding Advisory	03/29/2009	ND
Consulted with SPE William Vaughn regarding Advisory	5/7/2009	ND
Consulted with Primary Paul Kang	02/10/2009	ND
See updated east search	7/30/2009	ND
Consulted with SPe William Vaughn regarding search	7/30/2009	ND
Consulted SPE William Vaughn regarding Final	2/4/2010	nd
Consulted SPE William Vaughn regarding Appeal	1/27/2011	nd
Consulted with Qas Mark Radtke regarding IDS	7/30/2014	nd
Conducted 101 claim analysis	7/31/2014	nd
Conducted 112 2nd paragraph analysis	7/31/2014	nd
Conducted 112 6th paragraph claim analysis	7/31/2014	nd
Conducted double patenting search (EAST, eDAN, PALM) *Inventor **Assignee	7/31/2014	nd
East Search	7/31/2014	nd

SEARCH NOTES		
Search Notes	**Date**	**Examiner**
Plus Serch	7/31/2014	nd
NPL Seach (Google, Google Scholar)	7/31/2014	nd
709/246,231 + texted searched1	7/31/2014	nd

INTERFERENCE SEARCH			
US Class/ CPC Symbol	**US Subclass / CPC Group**	**Date**	**Examiner**
	Conducted Interference Search - See EAST Search History	7/31/2014	nd

Notice of References Cited	Application/Control No. 11/116,783	Applicant(s)/Patent Under Reexamination MAJOR ET AL.	
	Examiner NINOS DONABED	Art Unit 2444	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-2005/0033855	02-2005	Moradi et al.	709/231
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11116783	MAJOR ET AL.
	Examiner	**Art Unit**
	NINOS DONABED	2444

CPC										
Symbol									**Type**	**Version**
H04N		21	/	845					I	2013-01-01
H04N		21	/	84					F	2013-01-01
H04N		21	/	64792					I	2013-01-01
H04N		21	/	64769					I	2013-01-01
H04N		21	/	643					I	2013-01-01
H04N		21	/	25808					I	2013-01-01
H04N		21	/	2662					I	2013-01-01
			/							
			/							
			/							
			/							
			/							
			/							
			/							
			/							

CPC Combination Sets										
Symbol									**Type**	**Set**	**Ranking**	**Version**
			/									
			/									

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	21	
/NINOS DONABED/ Primary Examiner.Art Unit 2444	7/31/2014	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	20	7

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11116783	MAJOR ET AL.
	Examiner	**Art Unit**
	NINOS DONABED	2444

US ORIGINAL CLASSIFICATION		INTERNATIONAL CLASSIFICATION	
CLASS	SUBCLASS	CLAIMED	NON-CLAIMED
709	231	G 0 6 F 15 / 16 (2006.01.01)	

CROSS REFERENCE(S)					
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
709	246				

NONE	Total Claims Allowed:	
(Assistant Examiner) (Date)	21	
/NINOS DONABED/ Primary Examiner.Art Unit 2444 7/31/2014	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner) (Date)	20	7

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11116783	MAJOR ET AL.
	Examiner	**Art Unit**
	NINOS DONABED	2444

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
-	1	-	17	-	33	20	49								
-	2	-	18	-	34	21	50								
-	3	-	19	7	35										
-	4	1	20	8	36										
-	5	2	21	9	37										
-	6	-	22	10	38										
-	7	3	23	-	39										
-	8	4	24	11	40										
-	9	5	25	12	41										
-	10	6	26	13	42										
-	11	-	27	14	43										
-	12	-	28	15	44										
-	13	-	29	16	45										
-	14	-	30	17	46										
-	15	-	31	18	47										
-	16	-	32	19	48										

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	21	
/NINOS DONABED/ Primary Examiner.Art Unit 2444	7/31/2014	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	20	7

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 April 2004 (29.04.2004)

PCT

(10) International Publication Number
WO 2004/036824 A1

(51) International Patent Classification[7]: H04L 9/06

(21) International Application Number: PCT/FI2003/000752

(22) International Filing Date: 10 October 2003 (10.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20021820 14 October 2002 (14.10.2002) FI

(71) Applicant *(for all designated States except US)*: **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **CURCIO, Igor, D.D.** [IT/FI]; Hatanpään valtatie 12 C 57, FIN-33100 Tampere (FI). **LUNDAN, Miikka** [FI/FI]; Männikönkatu 4 A 7, FIN-33820 Tampere (FI).

(74) Agent: **FORSSÉN & SALOMAA OY**; Eerikinkatu 2, FIN-00100 Helsinki (FI).

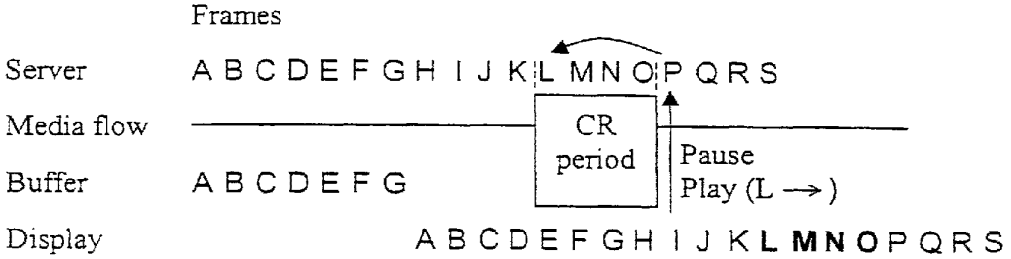
(81) Designated States *(national)*: AE, AG, AL, AM, AT (utility model), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, EG, ES, FI (utility model), FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT (utility model), PT, RO, RU, SC, SD, SE, SG, SK (utility model), SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: STREAMING MEDIA



(57) Abstract: The invention relates to a method for streaming of media from a streaming server (111) to a mobile client device (101) over an air-interface, wherein, after a cell reselection, the method comprises requesting the streaming server (111) to resend streaming media which the mobile client device (101) was not able to receive due to the cell reselection.

WO 2004/036824 A1

STREAMING MEDIA

FIELD OF THE INVENTION

The invention relates to streaming of media from a streaming server to a mobile client device over an air-interface.

BACKGROUND OF THE INVENTION

A packet switched streaming service PSS is currently being standardized for mobile environment by 3GPP (3rd Generation Partnership Project). An example of a communications system capable of streaming media (streaming video and/or audio) is shown in Figure 1. The system comprises a streaming server 111 which is coupled to an IP-network (Internet Protocol) 104. The IP-network 104 may be, for example, the Internet or a service provider operator's intranet (an intranet network belonging to the operator's domain). The IP-network 104 is coupled to a core network 103 of a mobile communications network via a G_i interface. The mobile communications network also has a radio access network (RAN) 102 coupled to the core network 103. The radio access network 102 provides mobile communications devices 101 with access to the mobile communications network over an air-interface. Said access may be provided either by circuit switched means (circuit switched voice or data call) or packet switched means or both. In the following, GPRS (General Packet Radio Service) is used as an example of a packet switched means to communicate over the air-interface. Concerning GPRS, the generic term radio access network (RAN) is considered to comprise base (transceiver) stations (BS) and base station controllers (BSC).

In streaming media a sequence of 'moving images' (i.e. video) or sound (i.e. audio) or a sequence of 'moving images' together with sound (i.e. multimedia) is sent in a compressed form from the streaming server 111 to a mobile communications device 101 (hereinafter referred to as a client device 101). In contrast to the

technique in which an entire media file has to be arrived at the client before it can be played, the streaming technique enables sending of media (video and/or audio) from the streaming server 111 to the client device 101 in a continuous manner and playing of the media as it arrives at the client.

When media is streamed in cellular mobile communications systems such as the system presented in Figure 1, new problems specific to the mobile environment arise. These problems are mostly due to the different restrictions of mobile systems. One such problem arises in a cell reselection (i.e. handover) situation presented in Figure 2.

In Figure 2, a first base station BS1 of the radio access network 102 serves the area of a first cell 201 whereas a second base station BS2 of the radio access network 102 serves the area of a second cell 202. Before cell reselection, the client device 101 is served by the first base station BS1, i.e. the client device 101 has an active radio link with the first base station BS1. This situation is shown on the left side of the Figure 2. After the cell reselection, the client device 101 is served by the second base station BS2. The radio link with the first base station BS1, shown by a dotted line, is dropped. This situation is shown on the right side of the Figure 2.

The cell reselection (CR) can be divided in three periods in time:

a) Pre-CR period
b) CR period
c) Post-CR period.

During the pre-CR period, received signal quality in the first cell weakens and the client device 101 starts cell reselection signaling. The client device 101 is able to receive streaming media via the first base station BS1 during that period. During the CR period, the actual cell reselection is performed. The client device 101 can

not receive streaming media during that period. During the post-CR period, the client device 101 is able to receive streaming media via the second base station BS2.

The cell reselection may cause a long break in service. For example, a cell reselection (CR period) in a case where GPRS is used as radio bearer may take up to 30-40 seconds. This will affect an ongoing streaming session. For example, part of the packets carrying streaming media may be lost and/or freezing of the played media may occur at the client device 101. If the streaming media comprises a video stream, the freezing means that a still image will appear and stay for a period of time on a client device display before the streaming media is again received and can be played again. If the streaming media comprises an audio stream, the freezing means that no sound is played back (i.e. silence) until the CR period is over.

Accordingly, there is a need to find appropriate means for reducing the effects of cell reselection on an ongoing streaming session.

SUMMARY OF THE INVENTION

According to a first aspect of the invention, there is provided a method for streaming of media from a streaming server to a mobile client device over an air-interface, wherein the method comprises:
requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection.

A request requesting the streaming server to send the streaming media may be sent from the mobile client device to the streaming server before or after the mentioned cell reselection.

The mentioned request is not limited to requesting to send only the media which

the mobile client device is (or was) not able to receive due to the cell reselection (during a cell reselection period). In an embodiment, the request may be construed such that what is actually requested is: resending of a portion of a media content not received as well as sending any remaining portion of the media content.

The term media is considered to mean either video or audio or another media, such as still image, or any combination thereof, i.e multimedia.

Preferably, a resending request is sent in response to the cell reselection. The re-sending request is preferably an application layer request sent between a mobile client device application and a streaming server application.

In a preferred embodiment, the streaming server is requested to send at an increased speed for a determined period of time in order to increase at the client device a degree of fullness of a temporary store, such as a buffer, in which the streaming media is temporarily stored before playback. Preferably, this is done by by requesting the streaming server to switch from sending a higher bit rate media stream to sending a lower bit rate media stream at an increased speed. The higher and lower bit rate streams are preferably provided by a multi-rate codec.

According to a second aspect of the invention, there is provided a mobile client device for receiving streaming media from a streaming server over an air-interface, the mobile client device comprising:
means for requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection.

Preferably, the air-interface couples said mobile client device to a mobile communications network. The mobile client device preferably comprises a cellular mobile phone.

According to a third aspect of the invention, there is provided a streaming server

for sending streaming media to a mobile client device over an air-interface, the streaming server comprising:
means for receiving a request requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection; and
means for acting upon the received request.

According to a fourth aspect of the invention, there is provided a system comprising a streaming server and a mobile client device, for streaming of media from the streaming server to the mobile client device over an air-interface, the system comprising, at the mobile client device:
means for requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection, the system further comprising, at the streaming server:
means for receiving the request; and
means for acting upon the received request.

According to a fifth aspect of the invention, there is provided a computer program executable in a mobile client device, the computer program comprising:
program code for causing the mobile client device to request the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection.

According to a sixth aspect of the invention, there is provided a computer program executable in a streaming server, the computer program comprising:
program code for causing the streaming server to receive a request requesting the streaming server to send streaming media which the mobile client device is not able to receive due to a cell reselection; and
program code for acting upon the received request.

Dependent claims contain preferable embodiments of the invention. The subject

matter contained in dependent claims relating to a particular aspect of the invention is also applicable to other aspects of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described by way of example with reference to the accompanying drawings in which:

Figure 1 shows a communications system capable of streaming media;

Figure 2 shows a cell reselection situation in the communications system of Figure 1;

Figure 3 shows a client device buffer in a situation in which the client device buffer is shorter than the time that cell reselection takes;

Figure 4 shows a client device buffer in a situation in which the client device buffer is longer than the time that cell reselection takes;

Figure 5 shows another illustration of the situation which is behind Figure 4;

Figure 6 shows preferred action in a preferred embodiment of the invention;

Figure 7 illustrates a preferred embodiment of the invention;

Figure 8 illustrates a preferred embodiment of the invention;

Figure 9 shows a client device according to a preferred embodiment of the invention;

Figure 10 shows a streaming server according to a preferred embodiment of the

invention; and

Figure 11 illustrates three ways of sending messages according to preferred embodiments of the invention.

DETAILED DESCRIPTION

The system shown in Figure 1 can also be used in the preferred embodiments of the invention (the same applies to Figure 2). Accordingly, the system comprises a streaming server 111 which is coupled to an IP-network 104. The IP-network 104 may be, for example, the Internet or a service provider operator's intranet. The IP-network 104 is coupled to a core network 103 of a mobile communications network via a G_i interface. The mobile communications network also has a radio access network (RAN) 102 coupled to the core network 103. The radio access network 102 provides mobile communications devices 101 with access to the mobile communications network over an air-interface. Said access may be provided either by circuit switched means (circuit switched voice or data call) or packet switched means or both. In the following, GPRS (General Packet Radio Service) is used as an example of a packet switched means to communicate over the air-interface.

In order to better understand the preferred embodiments of the invention, let us consider the following example.

In this example, a mobile communications device 101 (again referred to as a client device 101) has established a streaming session with the streaming server 111 when being served by the first base station BS1. In this example and in the following description, mostly a video streaming session is considered. However, a corresponding consideration applies to an audio streaming session as well as to a multimedia (e.g. video with audio) streaming session.

RTSP (Real Time Streaming Protocol) protocol is used in the streaming session

setup. Once the session has been established, the streaming itself may be performed (i.e. media flow may be sent) according to RTP (Real time Transport Protocol) or another protocol. However, if it is desired to make a change in the established session, this will again be done by using RTSP.

At the beginning of the established streaming session, the received streaming media (media flow), transmitted in packets (or frames), is buffered (i.e. temporarily stored) in a buffer at the client device (hereinafter referred to as a client device buffer). When the client device buffer is full, playing (playback) of the streaming media is started at one end of the buffer while the buffer is continuously filled at the other end with received streaming media. Therefore, the buffer should remain nearly full during normal operation of the system.

The client device now undergoes a cell reselection from the first cell 201 to the second cell 202 (Fig. 2). It is to be noted that, although the radio access network (RAN) 102 is illustrated in Figure 2 by a single cloud only, it may comprise different radio access networks. It may comprise a GPRS (GPRS-only) RAN, a 3G (third generation) RAN or a combination thereof. The first cell 201 (served by the base station BS1) and the second cell 202 (served by the base station BS2) may therefore be, for example, cells of the GPRS RAN or 3G RAN. Accordingly, the cell reselection (handover) can occur, for example, between base stations belonging to GPRS RAN, between base stations belonging to 3G RAN or between a base station belonging to GPRS RAN and a base station belonging to 3G RAN.

The client device 101 can not receive streaming media during the CR period. Therefore, when the CR period begins the client device buffer starts emptying because the buffer content is further played while the continuous filling is stopped. Depending on the size of the client device buffer, two different cases can be identified:

i) The client device buffer (in time) is shorter than the time that cell reselection

takes, i.e. the time to empty a full client device buffer is shorter than the CR period,

ii) The client device buffer (in time) is longer than the time that cell reselection takes, i.e. the time to empty a full client device buffer is longer than the CR period.

In the first case (Figure 3), the buffer will become totally empty during the CR period. Since the buffer becomes empty, the client device 101 has no media to play anymore. Therefore, the client device 101 starts rebuffering. Rebuffering can be started during the post-CR period. In order to get the rebuffering started the client device 101 can use an RTSP PAUSE/PLAY method. In this method, the client device 101 first sends an RTSP PAUSE message to the streaming server 111. The PAUSE message will cause the server 111 to pause sending of the media flow. Subsequently, the client device 101 sends an RTSP PLAY message which requests the streaming server 111 to start resending the streaming media from the last received frame onwards. Rebuffering starts when the client again starts receiving the streaming media. When the client device buffer is full, playing of the streaming media is restarted from the last received frame onwards. In the meantime, the last received frame is shown as a still image on a client device display. However, in spite of the drawback of showing a still image, no packets are effectively lost since the media flow which is not received at the client device during the CR period is resent from the server 111 to the client device 101 after the CR period.

In the second case (Figure 4), the whole content of the buffer has not been played during the CR period. Therefore, the buffer only becomes partially empty during the CR period, i.e. there is still streaming media (DATA) in the buffer when the CR period ends. During the post-CR period (after the CR period) playing of the streaming media stored in the buffer before the CR period (i.e. DATA*) is further continued at one end of the buffer while the other end is filled with the streaming

media received after the CR period (i.e. DATA**). If nothing is done, DATA* will eventually end and playing is continued with the first frame of data DATA**. No still image is shown. However, the first frame of DATA** is not the frame that should be played right after the last frame of DATA*. Since the client device 101 could not receive streaming media during the CR period it has not even received the frame that should be showed after the last frame of DATA*. There exists a temporal gap in the buffered media. As a result, the client device 101 will skip showing the sequence of streaming media (video) which could not be received due to the cell reselection Concerning an audio stream, the same effect occurs, i.e. the client device 101 will skip playing sound which could not be received.

The second case is further illustrated with the aid of Figure 5. It is to be noted that Figure 5 only shows a very simplified case. The time is considered to flow from left to right in Figure 5. The streaming media sequence that the server sends comprises video frames A-S. The frames are received at the client device and buffered before playing. The playing of the media starting from frame A starts when the buffer is full (i.e. when the frames A-G have been stored in the buffer), the frames (moving video picture) are subsequently shown on a client device display. The buffering of the frames following frame G is not shown in Figure 5 for clarity reasons.

The media flow from the server to the client device is not possible during the CR period. Therefore, the client device does not receive the frames L-O. Otherwise, the media flow is perfectly received at the client device. Now that the the CR period is shorter than the buffer size, the showing of the frame (frame K) last received before the CR period will be followed by the showing of the frame (frame P) first received after the CR period. Video frames L-O are not shown at all. Concerning an audio stream, the same effect occurs, i.e. the user hears a pause and a temporal gap in the sound.

There exists two type of frames: intra frames and inter frames. The intra frames

contain all necessary information of an image whereas inter-frames only contain changes or predicted changes compared to a previous image. The previous image may be an intra frame or an inter frame. Therefore, if the frame P is an intra frame the shown picture only warps ahead in time from frame K to frame P. In this case the frames L-O are not simply shown to the user. But if the frame P is an inter frame severe distortion in the shown (moving) picture is likely to occur before the next intra frame is received and played.

A preferred embodiment of the invention as shown in Figure 6 concentrates on the above mentioned problem relating to the second case. In this embodiment, the buffer is longer than the duration of the CR period. Therefore, the buffer does not become totally empty during the CR period but only partially empty. It is to be noted that Figure 6 only shows a very simplified case. The time is considered to flow from left to right in Figure 6. The streaming media sequence that the server sends comprises frames A-S. The frames are received at the client device and buffered before playing. The playing of the media starting from frame A starts when the buffer is full, the frames (moving video picture) are subsequently shown on a client device display. The buffering of the frames following frame G is not shown in Figure 6 for clarity reasons. The media flow from the server to the client device is not possible during the CR period. Therefore, the client device does not receive the frames L-O. Otherwise, the media flow is perfectly received at the client device.

When the CR period is over the client device knows exactly what is the last frame that it has received before the CR period (i.e. during the pre-CR period). In this embodiment it is frame K. The client device then requests, right after the CR period has been ended, the streaming server to start resending the streaming media from the last received frame onwards. Upon receiving the request the streaming server starts resending the media flow. The first frame to be sent is in this embodiment frame L. Any frames sent and received after the CR period but before the resending has started (possibly some frames starting from frame P) are

ignored by the client device. When the streaming media (of frames A-K) stored in the buffer before the CR period ends, then the playing is continued with the first frame (frame L) of the resent streaming media received after the CR period. Therefore, no discontinuity in the shown video picture should appear to the user. In Figure 6, the resent video frames L-O which would have otherwise failed to have been received at the client device are shown in bold font. Correspondingly, concerning an audio stream, the streaming server is requested to start resending the audio stream at a point at which reception has been stopped in the client device. No discontinuity in playback should appear.

In practice, the resending request can be performed by the RTSP PAUSE/PLAY method. In this method, the client device first sends, after the CR period has been ended, an RTSP PAUSE message to the streaming server. The PAUSE message will cause the server to pause sending of the media flow. However, the playing of the streaming media received at the client device is not paused unless the buffer gets empty (which should not be the case). Subsequently, the client device sends an RTSP PLAY message to the streaming server 111. The PLAY message contains information on the starting point of the resending. At the end of the CR period the client device knows the time of the last received frame. Based on that, the client device determines the starting point before sending the PLAY message. The PLAY message causes the server to start resending.

An example of the PAUSE message is as follows:

```
PAUSE rtsp://example.com/foo RTSP/1.0
CSeq: 6
Session: 354832
```

The PAUSE message informs the server that a change is coming. An example of the subsequently sent PLAY message is as follows:

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 7
Session: 354832
Range: npt=28.00-

The message field 'Range' tells the starting point of resending. In this example, the starting point is 28 seconds from the start of the streaming media sequence. In relation to the embodiment shown in Figure 6, this point would be exactly the time of frame L in the streaming media sequence.

In the embodiment just described, the frames (packets) that were lost during the CR period will be resent from the server to the client devide. Also, since the buffer size is longer than the duration of the CR period, no interruption in the shown video picture should appear and the user experience is maximized.

However, as the case is such that during normal streaming the degree of fullness of the buffer does not change, since the buffer is emptied (played) at the same rate as it is filled, this may cause an additional problem. In the embodiment just described, after the cell reselection, the client device buffer stays emptier than before, for the reasons as described in the foregoing. Therefore if, for example, a new cell reselection is to be performed in a near future, the emptier buffer may cause the same drawback as discussed in connection with the first case above (i.e. the case in which the client device buffer was shorter than the time that cell reselection takes).

A preferred embodiment of the invention concentrates on the above identified problem. In this embodiment, to ensure smooth behaviour of the shown video picture (correspondingly: played sound), the client device buffer is filled after the CR period, for a period of time, at a higher rate than it is emptied (played). This period may be called a filling period. When the filling period is over the buffer is full again and normal streaming in which the buffer is filled at the same rate as

emptied is resumed.

It is to be noted that normally raising the degree of fullness of the buffer would require the playing to be paused. In the present embodiment, playing is not paused but the degree of fullness of the buffer can still be raised due to the clever buffer management in which the buffer is filled, during playing, at a rate higher than the playing rate.

To raise the degree of fullness of the buffer without pausing the playback, the client device, in this embodiment, requests the server to switch to sending a lower bit rate streaming media sequence, but to use in the actual sending the same transmission bit rate (hereinafter referred to as an original transmission bit rate) as before. In order to reach the original transmission bit rate the client device requests the server to speed up the transmission of the lower bit rate sequence by a speeding factor. Speeding up the transmission bit rate causes more data to be written in the buffer than read out from the buffer. Therefore, the degree of fullness of the buffer increases as desired.

In other words, the server is asked to switch from sending the original sequence encoded at a first bit rate to sending a new corresponding sequence encoded at a second bit rate lower than the first bit rate and to increase the transmission bit rate of the new sequence in order to reach the original transmission bit rate (bandwidth). It is to be noted that the bit rate at which a media stream is encoded (and will be decoded) is a different concept than the transmission bit rate. The bit rate at which the media stream has been encoded has an effect on the picture quality. If the media stream has been encoded at a higher bit rate, this means that more bits have been used in encoding compared to encoding at a lower bit rate. This typically results in a better picture quality. The transmission bit rate, on the other hand, is the bit rate at which the media stream is actually sent and it depends on the available bandwidth.

The request to switch to sending a lower bit rate sequence and to speed up the transmission bit rate can be delivered by using the RTSP PAUSE/PLAY method. When the client device desires to fill the client device buffer during the post-CR period, it first sends a PAUSE message, corresponding to the one presented already in the preceding description, to the server to indicate that a change is coming. Subsequently it generates a PLAY message and sends it to the server.

An example of such a PLAY message is as follows:

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 7
Session: 354832
Range: npt=28.00-40.00
Bandwidth: 20000
Speed: 1.5

This message contains two optional message fields, namely 'Bandwidth' and 'Speed' to be understood at the client device and streaming server. These fields are already specified as optional fields by the IETF (Internet Engineering Task Force) in the standard RFC 2326 (Real Time Streaming Protocol).

The message field 'Bandwidth' tells the server to change into sending a lower bit rate sequence (here: a sequence whose bit rate is 20 kbps) and the message field 'Speed' tells the server to speed up the sending by a speeding factor (here: 1.5). The message field 'Range' tells the starting point and stopping point in the streaming media sequence in time units (calculated from the beginning of the streaming media sequence(s)).

The PLAY message shown would suit for an example in which the server initially sends a 30 kbps bit rate sequence at an original transmission bit rate of 30 kbps and in which the server, in order to fill the buffer, is desired to switch into sending

a sequence whose bit rate is 20 kbps and to speed up the sending at a speeding factor 1.5 to reach the original transmission bit rate of 30 kbps during a 12 seconds filling period (i.e. between the time instants of 28 and 40 seconds of the media sequence).

If the client knows the possible bit rate options (they are typically communicated from the server to the client during session setup, for example, with the aid of a 200 OK response to an RTSP DESCRIBE message) it can calculate the speeding factor needed to reach the original bitrate using the following formula:

$$speeding\ factor = \frac{original\ transmission\ bit\ rate}{new\ transmission\ bit\ rate}$$

When the buffer is full, the client device sends another RTSP PAUSE and PLAY message pair to the server. An example of the PLAY message is as follows:

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 67
Session: 354832
Range: npt=40.00-
Bandwidth: 30000
Speed: 1.0

This exemplary message requests the server to start sending the (original) 30 kbps bit rate media sequence (Bandwidth: 30000) at the original transmission bit rate 30 kbps (Speed: 1.0) starting from the time instant 40 seconds of the media sequence (Range: npt=40.00-).

The client device can calculate the length of the buffer filling period using the following formulas:

$$filling\ period = \frac{LowSeqTime}{speeding\ factor}$$

wherein

$$LowSeqTime = (BufferSize - BufferData) \cdot \frac{original\ transmission\ bit\ rate}{original\ transmission\ bit\ rate - new\ transmission\ bit\ rate}$$

In these formulas, *LowSeqTime* indicates the duration of the lower bit rate sequence playing time in the client device, *BufferSize* indicates the size of the buffer in seconds and *BufferData* indicates the data left in the buffer in seconds.

The filling of the buffer is illustrated in Figure 7. It is to be noted that Figure 7 only shows a very simplified case. All delays, for example, have been ignored. The time is considered to flow from left to right in Figure 7. It can be seen that the buffer starts emptying when the CR period starts. Upon receiving the first PLAY message the server switches to sending a media sequence having a lower bit rate (here: 20 kbps) and speeds up the sending by a speeding factor (here again: 1.5). Since the speeding factor is 1.5, the buffer filling takes twice the time that the emptying period lasted. For example, if the emptying period lasted for 25 seconds, the filling period should be 50 seconds with this speeding factor. When the buffer is full the second PLAY message is sent and the server switches to sending the original media sequence having a higher bit rate (here: 30 kbps) and continues sending at the original transmission bit rate.

In the foregoing, it has been described that, for example, a video stream typically comprises both intra and inter frames, wherein the intra frames are 'independent frames' containing all necessary information of an image whereas inter-frames only contain changes or predicted changes compared to a previous image. In the following preferred embodiment of the invention, the timing of the switch from the lower bit rate sequence to the original bit rate sequence is more closely de-

scribed with this respect.

In this embodiment the purpose is to time, whenever possible, the switch from the lower bit rate sequence to the original bit rate sequence to take place at the point of an intra frame of the original bit rate sequence (or, more generally, at the point of an intra frame of the sequence to which the sending is switched). In this way, prediction errors and/or jumps in media (frame) flow can be avoided. Figure 8 illustrates this embodiment. It is to be noted that Figure 8 only shows a simplified case. The time is considered to flow from left to right in Figure 8. In this embodiment, the first switch from the original bit rate sequence to the lower bit rate sequence is effected at point 28 seconds with the aid of a first PAUSE/PLAY message pair.

As illustrated in Figure 8, the position of intra frames in the original bit rate sequence may differ from the ones of the lower bit rate sequence. However, based on:

- the time required to totally fill the buffer, i.e. '*filling period*',
- the time distance of two adjacent intra frames in the original sequence, i.e. an intra frame rate, '$IFrameTime_{Original}$', and
- the time at which the first switch from the original bit rate sequence to the lower bit rate sequence is effected, i.e. '*SwitchTime*',

it can be calculated a suitable point in time for swiching back to the original bit rate sequence. The parameter '$IFrameTime_{Original}$' can be calculated in the client device from the original bit rate sequence before switching to the lower bit rate sequence. The calculation of the suitable point in time for swiching back to the original bit rate sequence can be performed using the following formula:

$$SeqChangeTime = IFrameTime_{Original} * \left\lfloor \frac{SwitchTime + filling\ period}{IFrameTime_{Original}} \right\rfloor$$

In this formula, '*SeqChangeTime*' indicates the point in time for swiching back to the original bit rate sequence and the brackets indicate a *floor*-function (or *trunc*-function) which truncates the fractional part of the value computed in the brackets. For example, if the time required to totally fill the buffer, i.e. '*filling period*' is 16 seconds, the time at which the first switch is effected, i.e. '*SwitchTime*' is 28 seconds and the time difference between two adjacent intra frames in the original sequence, i.e. '$IFrameTime_{Original}$' is 5 seconds, then the point in time for swiching back to the original bit rate sequence is 40 seconds (*SeqChangeTime*=5*floor((28+16)/5) = 40 s). Accordingly, the second PAUSE/PLAY message pair is sent at point 40 seconds, as indicated in Figure 8. In this embodiment, the starting point and stopping point to be placed in the message field 'Range' of the PLAY message are 28 seconds and 40 seconds, respectively.

It is to be noted that, due to the intra frame rate, it might not in all cases be possible to totally fill the buffer. For example, in the example just described, it would have taken 4 seconds more to totally fill the buffer. However, the formula gives the closest suitable intra frame to the buffer filling period.

In another embodiment, it may well be that, at the point of the last received frame before the CR period, there is not an intra frame in the lower bit rate media sequence. In this embodiment, the starting point of the resending request is adjusted by a necessary amount of frames (or time) backwards such that there will be an intra frame in the lower bit rate sequence at the starting point. In this case, a set of last received frames before the CR period, which belong to a time period subsequent to the starting point, are ignored at the client device in order to guarantee a continuous playback.

In another embodiment of the invention, the switches between two different bit rate sequences are performed without the timing method(s) just described. In this embodiment, the first switching point (from higher to lower bit rate sequence) is directly determined by the last received frame and the second switching point (from lower to higher bit rate sequence) is directly determined by the the time required to totally fill the buffer (*'filling period'*). The switching points can therefore end up being at the point of an intra or inter frame. Should a switching point end up being at the point of an inter frame (for example a P-frame), a small prediction error may occur in the played media, however, in this embodiment, the buffer can be totally filled.

Another way to perform switching is to use so called "switch frames". These are frames which contain "difference" information between corresponding frames in different bit rate sequences. In this embodiment, a bridge and switch between the two sequences is performed with the aid of these frames using the difference information.

Figure 9 shows a client device 101 according to a preferred embodiment of the invention. The client device may be a mobile station of a cellular radio telephone network. The client device 101 comprises a processing unit MCU, a radio frequency part RF, and a user interface UI. The radio frequency part RF and the user interface UI are coupled to the processing unit MCU. The user interface UI typically comprises a display, one or more speaker and a keyboard (not shown) with the aid of which a user can use the device 101.

The processing unit MCU comprises a processor (not shown), a memory 210 and computer software. The software has been stored in the memory 210. The aforementioned client device buffer 240 is also comprised in the memory 210. The processor controls, in accordance with the software, the operation of the client device 101, such as receiving of the streaming media sent from the server 111 and sending of requests to the server 111 via the radio frequency part RF, reading and

writing of received streaming media (video and/or audio) in the buffer 240, and presenting of the received streaming video on the display and audio on the one or more speaker of the user interface UI. A suitable size of the buffer (in time) can be, for example, 1.5 or 2 times the maximum (or average) CR period time.

The software comprises a streaming client software application 220 (hereinafter referred to as client software 220), a protocol stack 230 for implementing the necessary protocol layers such as an RTP layer, an RTSP layer, an SDP (Session Description Protocol) layer, a TCP layer (Transmission Control Protocol), an IP layer and, below the IP-layer, lower protocol layers. In addition, the software comprises as a part of the client software 220 a media player for playing the received media.

The processor generates, based on the client software 220 the aforementioned PAUSE and PLAY messages and sends them to the server 111 via the radio frequency part RF. The processor also performs, based on the client software 220, the necessary calculations relating to the speeding factor, relating to the buffer filling period and relating to the suitable point in time for swiching from a lower bit rate sequence back to the original bit rate sequence.

The generation and sending of the resending request (PLAY message) and other appropriate action is triggered by a cell reselection event occurred in the mobile client device 101. The event can be detected by the client software 220 by reception of an asynchronous message from an API (Application Programming Interface) provided by the lower layers of the protocol stack 230. Alternatively or in addition, the event can be detected by monitoring the buffer level, i.e. the degree of fullness of the client device buffer 240. In this case, if the buffer 240 does not receive data for a certain amount X of time (depending on the implementation, the parameter X can be defined as a constant value, and it is a threshold for the client software 220 to understand that the cell reselection event has occurred), and if the client device 101 afterwards starts to receive data after a certain variable amount

Y of time (wherein Y > X, and Y is the real duration of the CR period), then the client can trigger the action described in this description.

Figure 10 shows a streaming server 111 according to a preferred embodiment of the invention. The streaming server 111 comprises a processing unit CPU, a first memory 310, an IP network interface 350, and a second memory 360. The first memory 310, the IP network interface 350, and the second memory 360 are coupled to the processing unit CPU.

The processing unit CPU controls, in accordance with computer software stored in the first memory 310, the operation of the streaming server 111, such as processing of requests received from the client device 101 and the sending of pre-recorded media streams, stored in the second memory (disk) 360, to the client device 101 via the IP network interface 350.

The software comprises a streaming server software application 320 (hereinafter referred to as server software 320), a protocol stack 330 for implementing the necessary protocol layers such as an RTP layer, an RTSP layer, an SDP layer, a TCP layer, an IP layer and lower protocol layers.

The PAUSE and PLAY messages sent from the client device 101 are received via the IP network interface 350. A processor (not shown) of the processing unit CPU processes the messages according to the server software 320 and the protocol stack 330 and takes appropriate action.

The present invention provides means for reducing the effects of cell reselection on an ongoing streaming session. It is to be noted that, according to preferable embodiments of the invention, a request for resending (e.g. a PLAY message) is sent on the application layer, i.e. between the client software application 220 and the server software application 320. Preferably RTSP over TCP (Transmission Control Protocol) or RTSP over another reliable protocol is used in transferring

application layer requests from the client device to the streaming server. The reception of the messages at the streaming server can therefore basically be guaranteed.

Although it has been described that the cell reselection would be performed between two base stations, it should be noted that a cell reselection can be performed also between two sectors of one and the same base station. Also, it should be noted that depending on the implemention, it may well be that separate messages (PAUSE messages) pausing the sending of streaming media are not needed. In an alternative embodiment, both to stop sending streaming media and to start resending streaming media is caused by a single appropriate message.

Further, in relation to the buffer filling embodiment, it has been described in the foregoing that when the server switches to sending a lower bit rate sequence, the original transmission bit rate would be maintained. However, in an alternative embodiment of the invention, a transmission bit rate higher than the original transmission bit rate is used during the filling period in order to fill the buffer more quickly. In this embodiment, it is assumed that a larger bandwidth can be requested by the mobile client device and that a larger bandwidth is actually available in the radio access network.

Further, in relation to the buffer filling embodiment, the bandwidth information of the sequence can be sent to the streaming server 111 also by other means than the RTSP field 'Bandwidth' (for example, by requesting a specific sequence encoded at a bitrate known to the client device 101). In this case, the field 'Bandwidth' is not used, the fields 'Speed' and 'Range' are recalculated according to the actual known sequence bit rate.

It is possible that the network (air-interface) bandwidth changes during the buffer filling period. If the client device 101 supports bandwidth adaptation that involves bit stream switching, the client device 101 should pause the buffer filling (with an

RTSP PAUSE message) before sending a bandwidth adaptation message to the streaming server 111. After the bandwidth adaptation operations are over, the client device can start the buffer filling again and recalculate the 'Bandwidth', 'Speed' and 'Range' values according to the bit rate and timing information of the new media stream.

Also, in relation to the buffer filling embodiment, it should be noted that, in an alternative embodiment, the second PAUSE/PLAY messages are not sent at all but the switch back to sending the original media sequence at the original transmission bit rate will be performed automatically by the server based on the stopping point information contained in the first PLAY message.

The resending request mentioned in the various embodiments of the invention may, in certain cases, actually be a sending request. One such case is considered as an alternative embodiment of the invention. In this embodiment, the client device (101) knowing beforehand that a cell reselection is going to occur in a very near future sends a PAUSE message to the streaming server (111) before the beginning of the cell reselection period (i.e. during the pre-CR period). Sending of the PAUSE message is triggered by a cell reselection initiation event notified to the client software 220 by means of the lower layer API. The PAUSE message causes the streaming server 111 to stop sending the streaming media. When the CR-period is over, the client device 101 then sends a PLAY message causing the streaming server to start sending at the point at which sending was stopped before the CR period. Playing of the streaming media is not stopped at the client device 101 in between, and if the client device buffer 240 has been selected to be longer in time than the time that cell reselection takes, the buffer 240 does not become totally empty during cell reselection and the user does not experience any jumps or interruption in playback. The PLAY message may contain a request to send at an increased speed in order to increase the degree of fullness of the buffer 240.

In yet another embodiment, the mobile client device 101 sends, before the

CR period, to the streaming server 111 an appropriate message requesting the streaming server 111 to stop sending the streaming media and to start sending again at a suitable point in time after the CR period. Sending of the message is triggered, during the pre-CR period, by a cell reselection initiation event notified to the client software 220 by means of the lower layer API. The mobile client device 101, knowing approximately the time that the cell reselection is going to take, estimates a suitable point in time at which it is again able to receive data. It inserts this information to the message so as to prevent the streaming server 111 from starting to send again too early.

An example of the above mentioned message is as follows:

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 7
Session: 354832
Range: npt=28.00-40.00; **time=19970123T153600Z**
Bandwidth: 20000
Speed: 1.5

This is an RTSP PLAY message having a header field 'Time'. The value of that field schedules the start of the "future" sending of the streaming media.

Figure 11 illustrates (especially relating to the buffer filling embodiments) the following three ways in which messages may be sent:

Case 1: The first PAUSE and PLAY messages are sent after the CR period. The second PAUSE and PLAY messages are sent when the buffer 240 is full.

Case 2: The first PAUSE message is sent before the CR period. The associated PLAY message is sent after the CR period. The second PAUSE and PLAY messages are sent when the buffer 240 is full.

Case 3: The first PAUSE message is sent before the CR period. The associated PLAY message is sent after the PAUSE message but before the CR period. The second PAUSE and PLAY messages are sent when the buffer 240 is full.

The sending of the second PAUSE message is not necessary, if the first PLAY message contains a closed 'Range' field.

Particular implementations and embodiments of the invention have been described. It is clear to a person skilled in the art that the invention is not restricted to details of the embodiments presented above (e.g. message names and message field names), but that it can be implemented in other embodiments using equivalent means without deviating from the characteristics of the invention. The scope of the invention is only restricted by the attached patent claims.

CLAIMS:

1. A method for streaming of media from a streaming server (111) to a mobile client device (101) over an air-interface, wherein the method comprises:

requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection.

2. A method according to claim 1, wherein the streaming server is provided with a starting point at which to start sending.

3. A method according to claim 1, wherein streaming server (111) sends the streaming media which the mobile client device (101) is not able to receive due to said cell reselection as well as a remaining portion of streaming media in response to the request.

4. A method according to claim 1, wherein the cell reselection comprises a cell reselection period during which the mobile client device (101) is not able to receive streaming media, the method comprising:

sending from the mobile client device (101) to the streaming server (111), after the cell reselection period, a resending request which requests the streaming server (111) to resend streaming media which the mobile client device (101) was not able to receive during the cell reselection period.

5. A method according to claim 4, wherein the resending request is generated according to RTSP protocol (Real Time Streaming Protocol).

6. A method according to claim 4, wherein the resending request is implemented by an RTSP PAUSE/PLAY message pair.

7. A method according to claim 1, wherein the streaming media is temporarily stored in a temporary store (240), such as a buffer, at the client device (101)

before playing.

8. A method according to claim 7, wherein the temporary store (240) has a size longer in time than a cell reselection period.

9. A method according to claim 7, wherein the streaming server is requested to send streaming media at a rate higher than the playing rate of that media so as to increase a degree of fullness of the temporary store (240).

10. A method according to claim 9, wherein a bandwidth or desired transmission bit rate with speeding factor is communicated to the streaming server (111) in a request.

11. A method according to claim 9, wherein the streaming media is stored at the mobile client device (101) at a rate higher than the playing rate.

12. A method according to claim 9, wherein the streaming server (111) is subsequently requested to resume an original configuration.

13. A method according to claim 7, wherein a degree of fullness of the temporary store (240) decreases during the cell reselection, and the streaming server is requested to send the not received streaming media although the temporary store (240) has not become totally empty, and said requesting is performed without pausing playback at the mobile client device (101).

14. A method according to claim 1, wherein the streaming server has a set of media streams available for transmission in which the same media content has been encoded at different bit rates.

15. A method according to claim 14, wherein information on the available set of media streams is beforehand communicated to the mobile client device (101)

in a streaming session setup.

16. A method according to claim 15, wherein the streaming server (111) is requested to switch from sending a higher bit rate media stream to sending a lower bit rate media stream at an increased speed.

17. A method according to any preceding claim, wherein the streaming media comprise one of the following: a video stream, an audio stream, another stream of single media, a multimedia stream.

18. A method according to any preceding claim, wherein the streaming server (111) sends streaming media to the mobile client device (101) via a mobile communications network.

19. A method according to any preceding claim, wherein the mobile communications network comprises a mobile packet radio network, such as a GPRS (General Packet Radio Service) network.

20. A method according to any preceding claim, wherein said cell reselection is performed between two base stations (BS1, BS2) which are selected from a group comprising: base stations belonging to a GPRS system, base stations belonging to a third generation mobile communications system.

21. A mobile client device (101) for receiving streaming media from a streaming server (111) over an air-interface, the mobile client device (101) comprising:

means (220, 230, MCU) for requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection.

22. A streaming server (111) for sending streaming media to a mobile client device (101) over an air-interface, the streaming server (111) comprising:

means (350) for receiving a request requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection; and

means (320, 330, CPU) for acting upon the received request.

23. A system comprising a streaming server (111) and a mobile client device (101), for streaming of media from the streaming server (111) to the mobile client device (101) over an air-interface, the system comprising, at the mobile client device (101):

means (220, 230, MCU) for requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection, the system further comprising, at the streaming server (111):

means (350) for receiving the request; and

means (320, 330, CPU) for acting upon the received request.

24. A computer program executable in a mobile client device (101), the computer program comprising:

program code for causing the mobile client device (101) to request the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection.

25. A computer program executable in a streaming server (111), the computer program comprising:

program code for causing the streaming server (111) to receive a request requesting the streaming server (111) to send streaming media which the mobile client device (101) is not able to receive due to a cell reselection; and

program code for acting upon the received request.

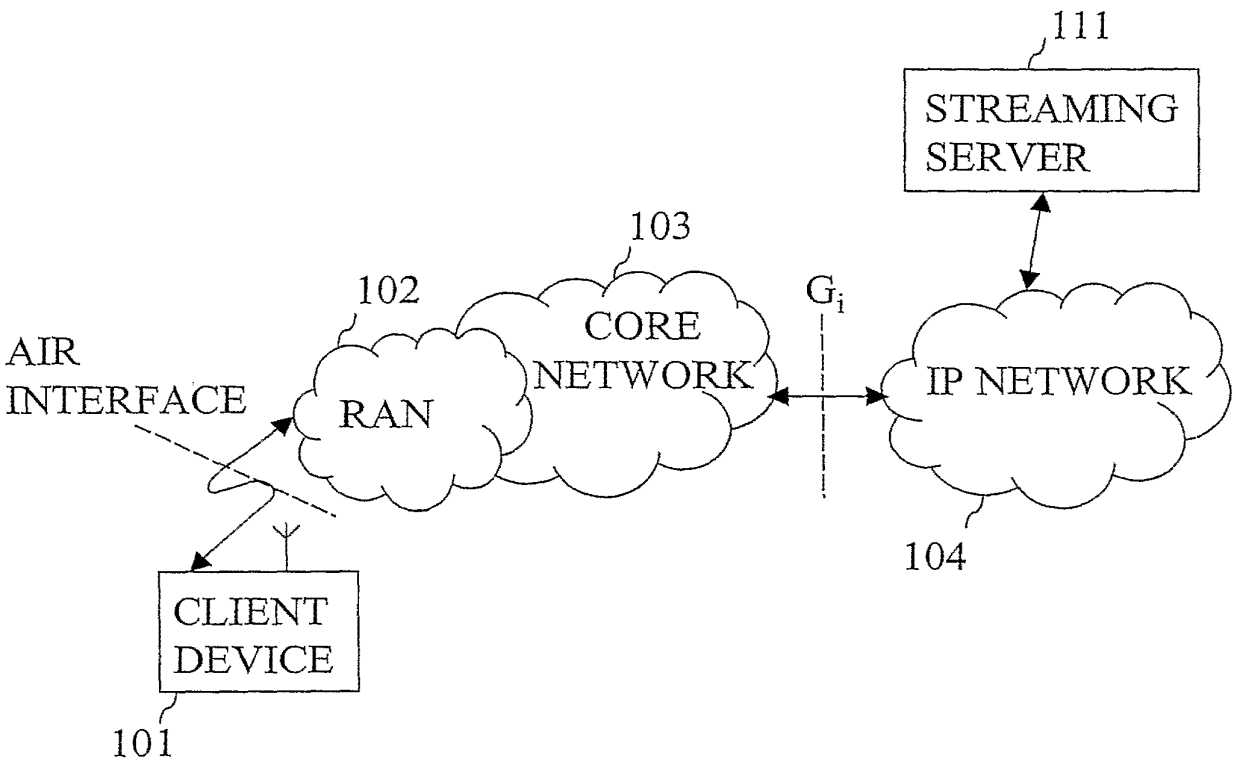
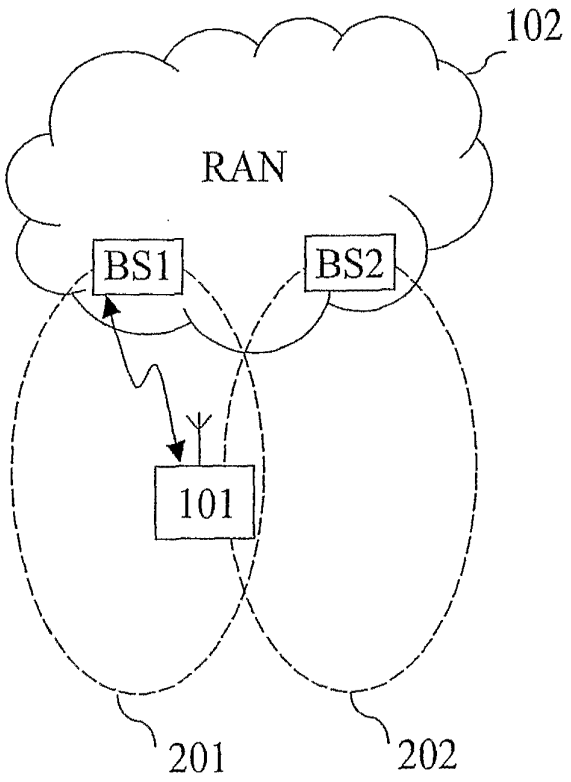

111
STREAMING SERVER
103
102
CORE NETWORK
G_i
AIR INTERFACE
RAN
IP NETWORK
104
CLIENT DEVICE
101

FIG. 1

BEFORE CELL RESELECTION:



AFTER CELL RESELECTION:

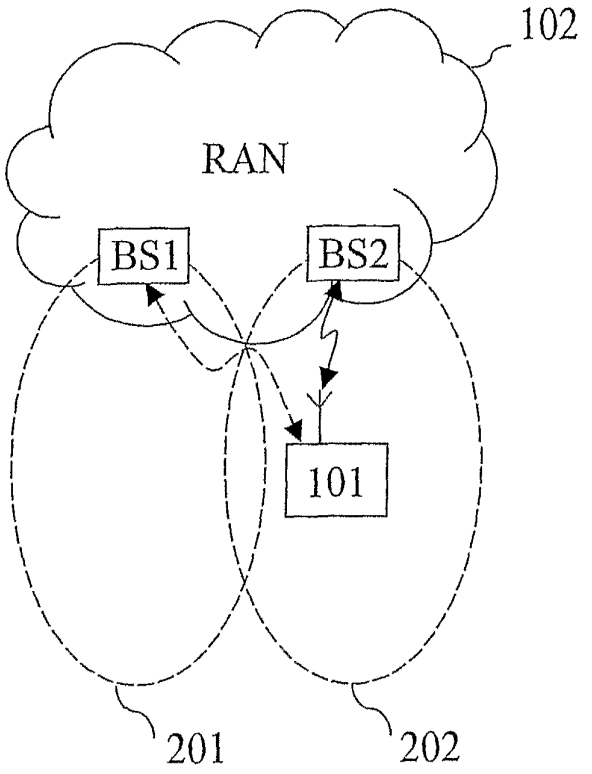


FIG. 2

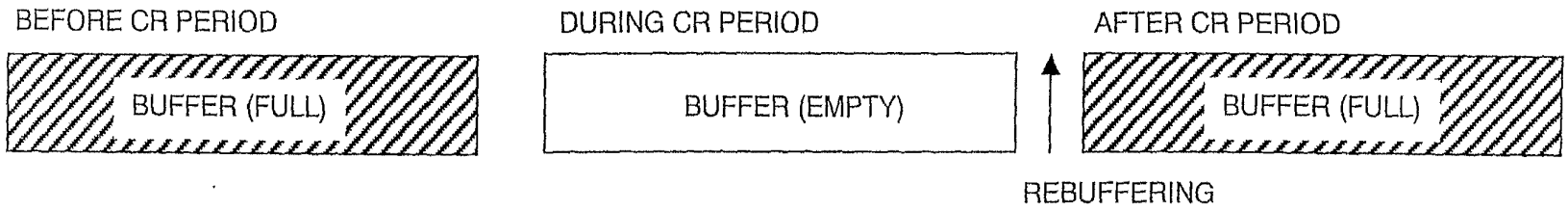

BEFORE CR PERIOD
BUFFER (FULL)
DURING CR PERIOD
BUFFER (EMPTY)
AFTER CR PERIOD
BUFFER (FULL)
REBUFFERING

FIG. 3

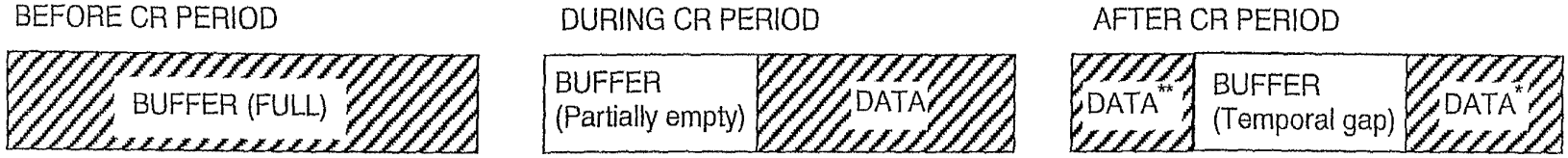

BEFORE CR PERIOD
BUFFER (FULL)
DURING CR PERIOD
BUFFER
(Partially empty)
DATA
AFTER CR PERIOD
DATA**
BUFFER
(Temporal gap)
DATA*

FIG. 4

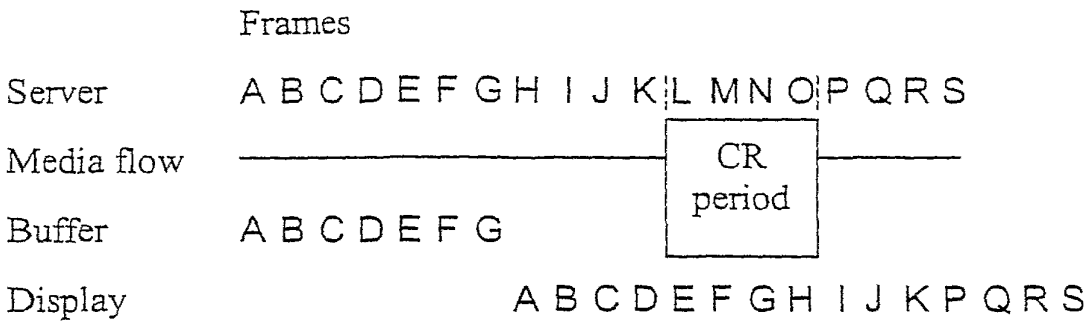

Frames
Server
A B C D E F G H I J K L M N O P Q R S
Media flow
CR period
Buffer
A B C D E F G
Display
A B C D E F G H I J K P Q R S

FIG. 5

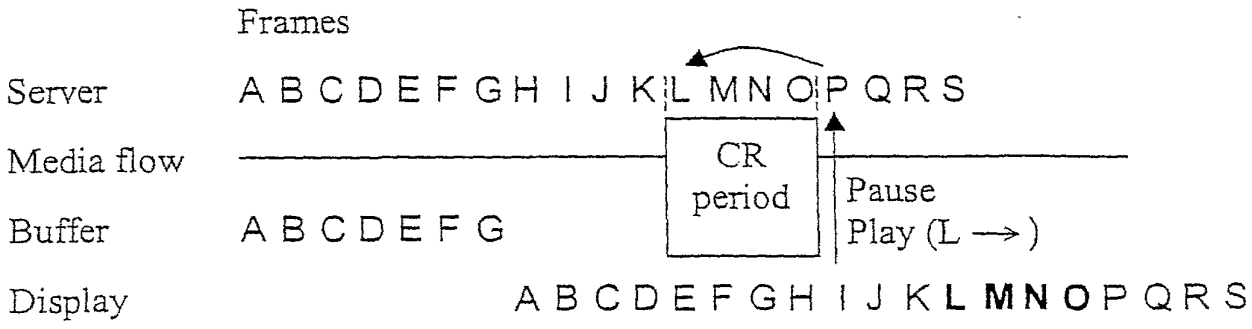

Frames
Server
A B C D E F G H I J K L M N O P Q R S
Media flow
CR period
Pause
Play (L →)
Buffer
A B C D E F G
Display
A B C D E F G H I J K L M N O P Q R S

FIG. 6

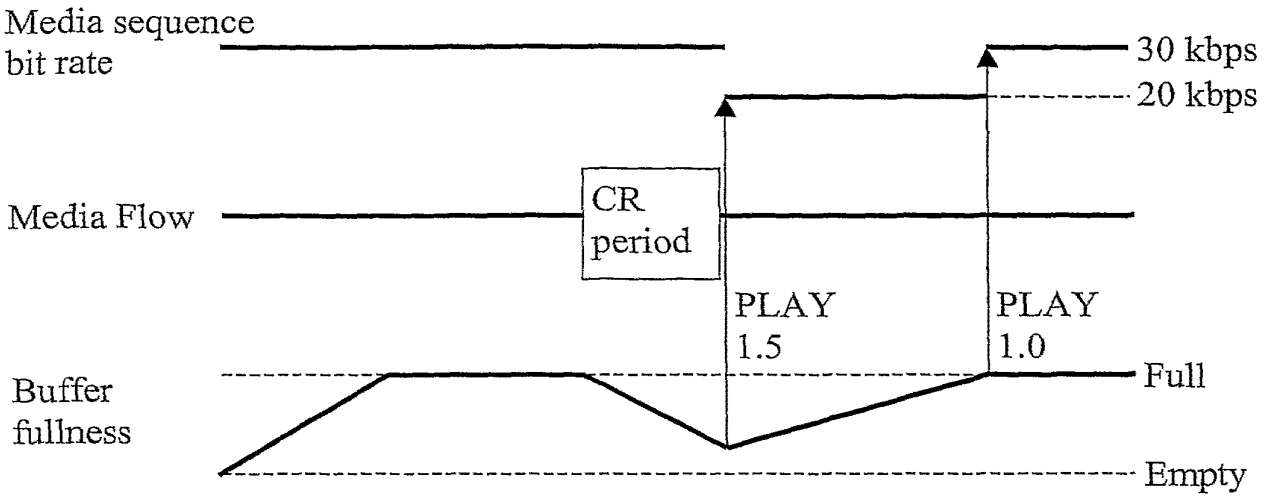

Media sequence bit rate
30 kbps
20 kbps
Media Flow
CR period
PLAY 1.5
PLAY 1.0
Buffer fullness
Full
Empty

FIG. 7

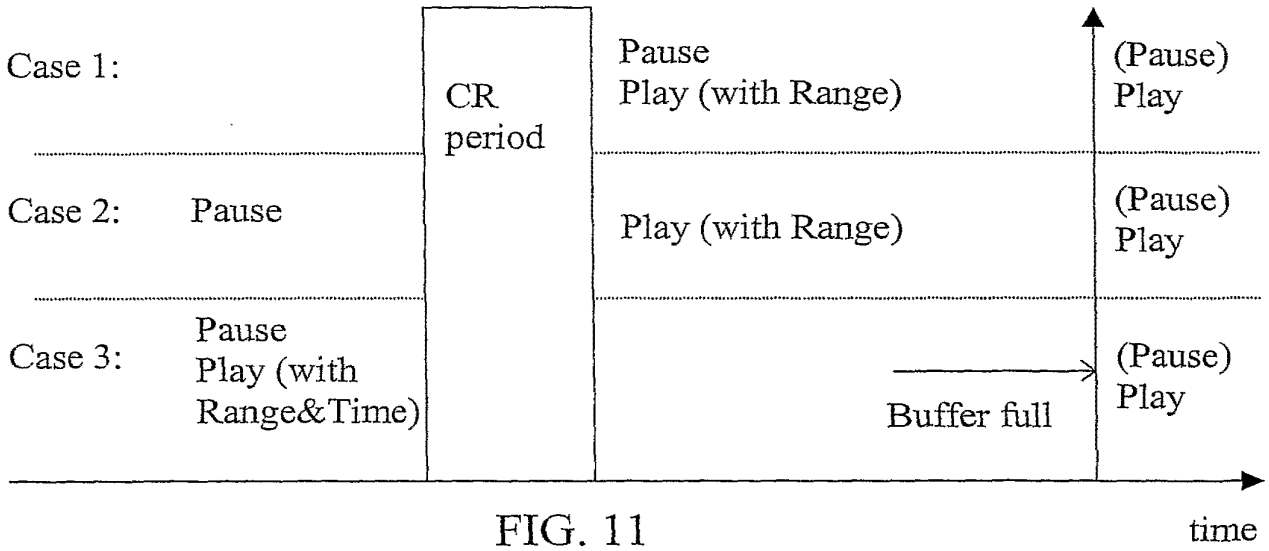

Case 1:
CR period
Pause
Play (with Range)
(Pause) Play
Case 2:
Pause
Play (with Range)
(Pause) Play
Case 3:
Pause
Play (with Range&Time)
Buffer full
(Pause) Play
time

FIG. 11

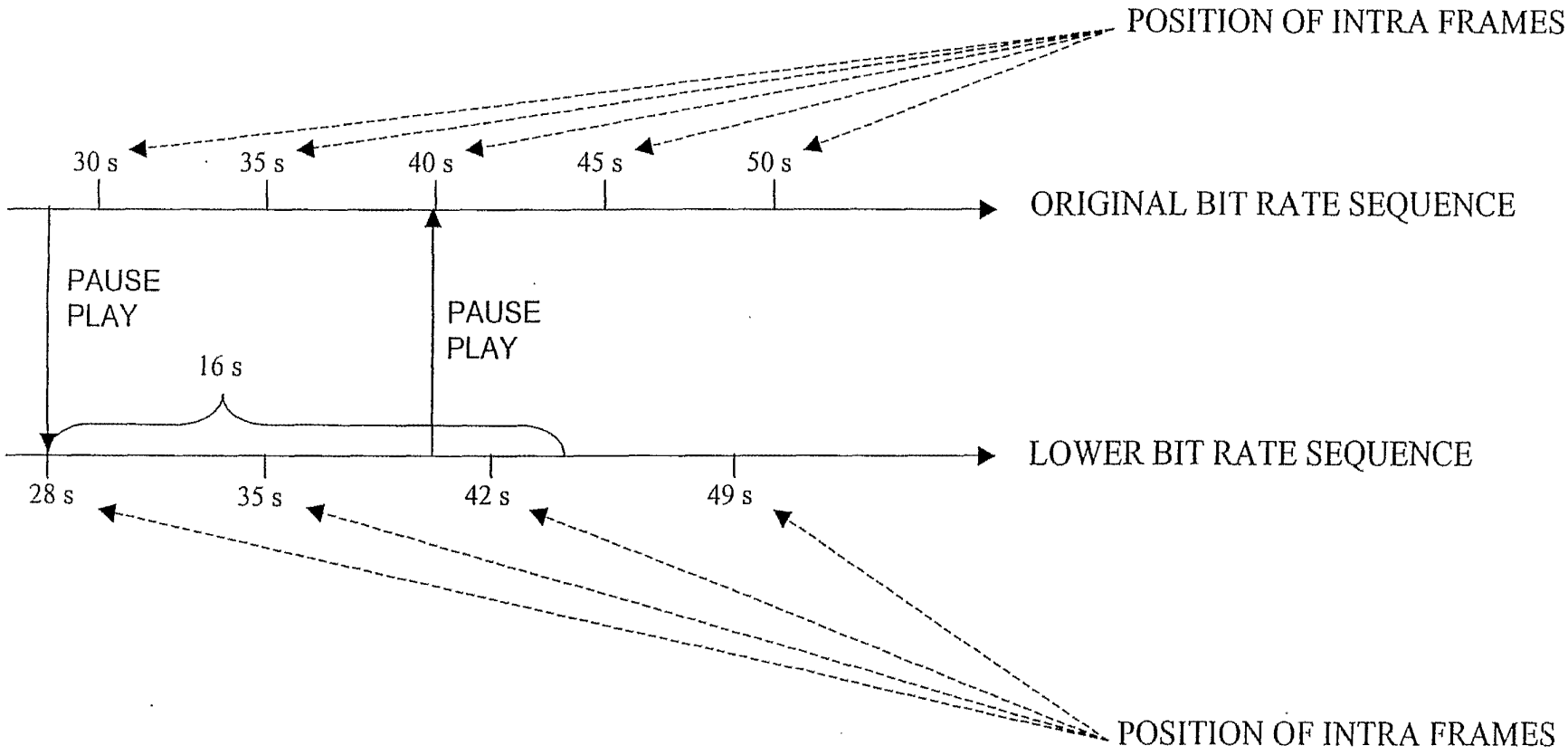

POSITION OF INTRA FRAMES
30 s
35 s
40 s
45 s
50 s
ORIGINAL BIT RATE SEQUENCE
PAUSE
PLAY
PAUSE
PLAY
16 s
LOWER BIT RATE SEQUENCE
28 s
35 s
42 s
49 s
POSITION OF INTRA FRAMES

FIG. 8

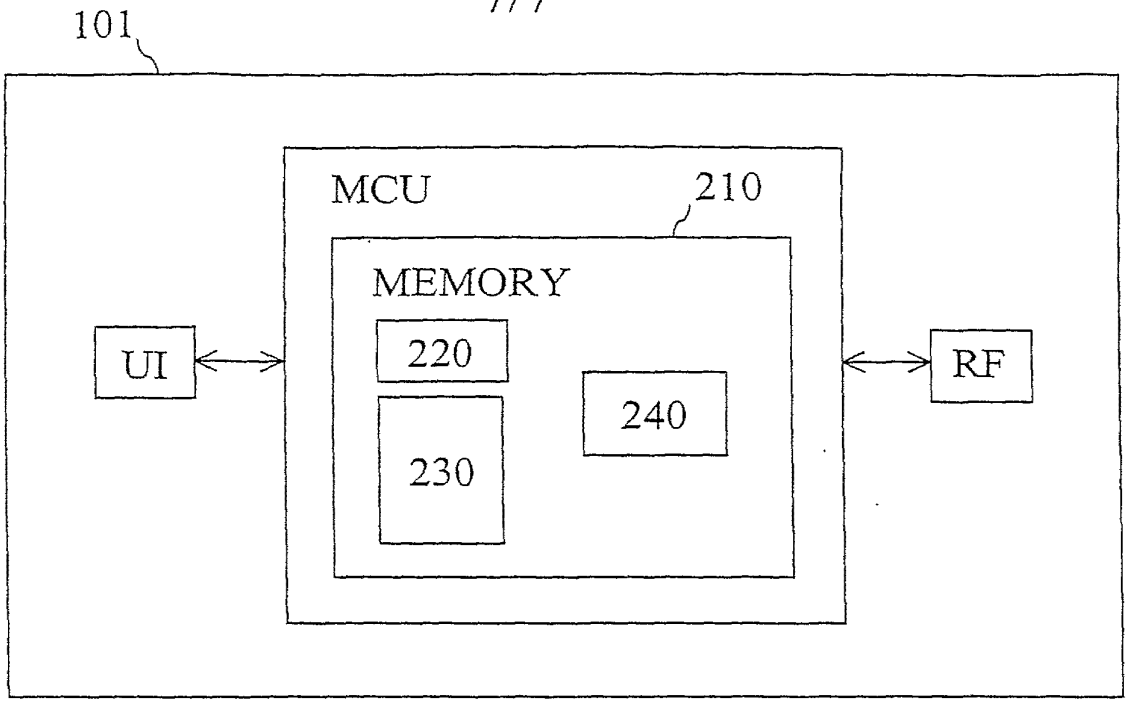

101
MCU
210
MEMORY
220
230
240
UI
RF

FIG. 9

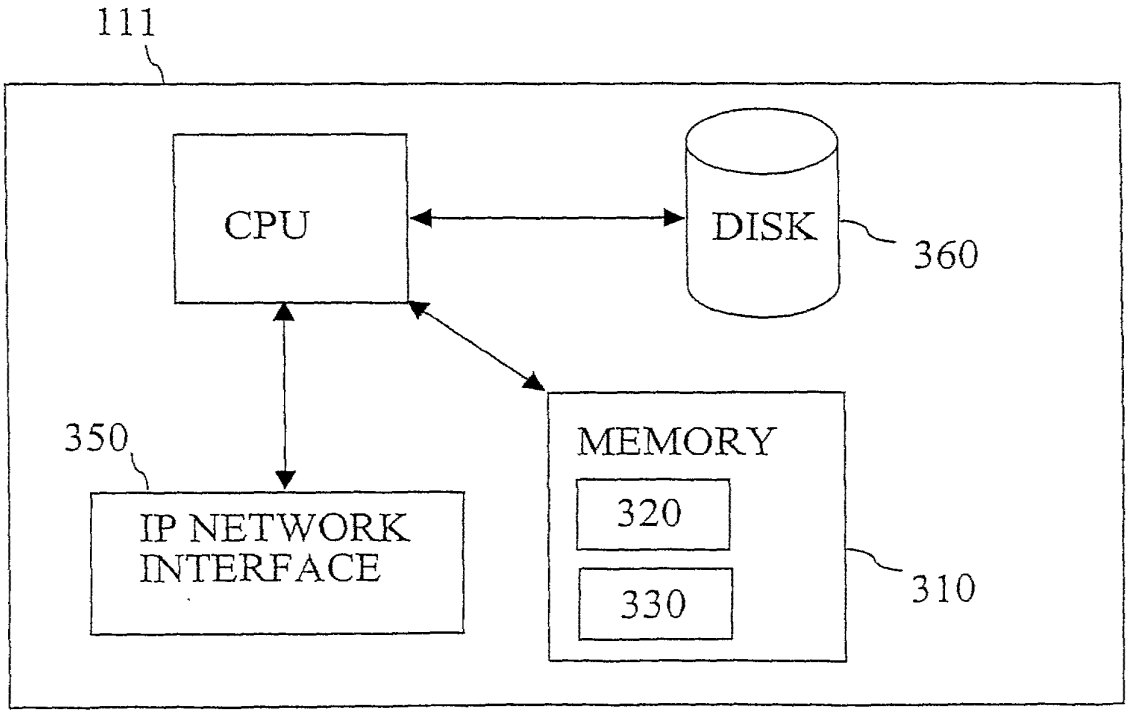

111
CPU
DISK
360
350
IP NETWORK INTERFACE
MEMORY
320
330
310

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 03/00752

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04L 9/06
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04L, H04Q, H04B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,A	US 2003009576 A1 (APOSTOLOPOULOS, J.G. ET AL), 9 January 2003 (09.01.03), paragraphs [0009] - [0011]; figures 8-11	1-25
A	3GPP TS 26.233 V5.0.0: 3rd Generation Partnership project; Technical Specification Group Services and System Aspects; Transparent end-to-end packet switched streaming service (PSS); General description (Release 5), March 2002, pages 1-12, see section 4, "Usage scenarios" and section 5, "General service architecture".	1-25

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 December 2003	16 -12- 2003
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Fredrik Blomqvist /OGU Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 03/00752

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 0130090 A2 (TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)), 26 April 2001 (26.04.01), page 2, line 29 - page 4, line 19 -- --------	1-25

Form PCT/ISA/210 (continuation of second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

Information on patent family members

06/09/03

International application No.
PCT/FI 03/00752

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003009576 A1	09/01/03	WO 03005761 A	16/01/03
WO 0130090 A2	26/04/01	AU 1065601 A	30/04/01
		EP 1224829 A	24/07/02

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 395 014 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
03.03.2004 Bulletin 2004/10

(51) Int Cl.[7]: **H04L 29/06**, H04L 29/08

(21) Application number: **02019052.6**

(22) Date of filing: **27.08.2002**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SK TR
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant: **MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.**
Kadoma-city, Osaka 571-8501 (JP)

(72) Inventors:
- **Rey, José Luis**
  64287 Darmstadt (DE)
- **Burmeister, Carsten**
  22081 Hamburg (DE)
- **Hakenberg, Rolf**
  64295 Darmstadt (DE)

(74) Representative: **Grünecker, Kinkeldey,**
Stockmair & Schwanhäusser Anwaltssozietät
Maximilianstrasse 58
80538 München (DE)

(54) A method of transmitting data streams with data segments of variable length

(57) A method of transmitting data streams between a sending host and a receiving host using a transport protocol comprising the steps of making available data streams at the sending host in the form of mono-rate or multi-rate content streams having the same content but different encoding rates, establishing a connection between the sending host and the receiving host, transmitting data streams with a selected encoded rate to the receiving host, and separating the data streams into data segments of variable length at the sending host as required during transmission.

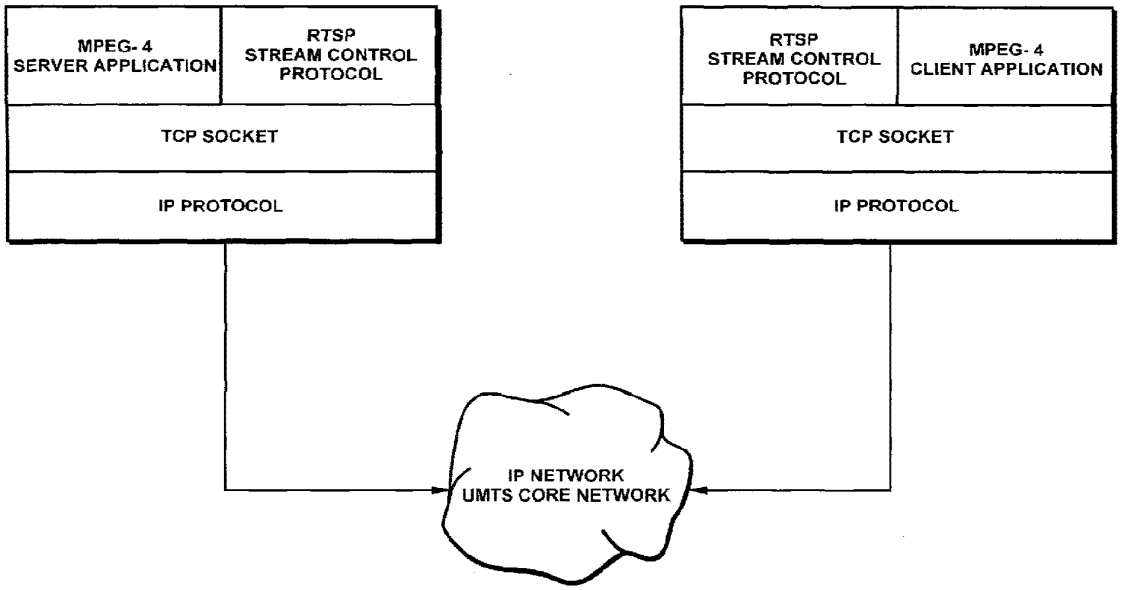


Fig. 1

Printed by Jouve, 75001 PARIS (FR)

EP 1 395 014 A1

Description

[0001] The present invention relates to a method of transmitting data streams between a sending host and a receiving host using a transport protocol such as TCP.

[0002] In network communications between a sending host and a receiving host, the transmission of data packets is mostly done on a best-effort basis. In the communication link, the available bandwidth and the current link conditions vary and hence, the end hosts must be provided with mechanisms to probe and adjust themselves to the current link conditions.

[0003] With the development of 3rd generation networks, so-called streaming applications are gaining importance. Streaming applications are understood as a continuous stream of data, e.g. video or audio data which is typically sensitive to time delays. In order that the user will be able to receive multimedia data streams on a portable receiver, for example a mobile phone, several requirements on the transport mechanism will be imposed in order that the delivered data meets with the resulting time constraints. These constraints are difficult to meet in a best-effort environment, which is subject to bandwidth variations and delays.

[0004] To cope with the problem of bandwidth adaption, it is common praxis to make several data streams with different encoding rates but same content available at the sending host. According to the available bandwidth, it is then switched between the data streams in order to adjust the rate to the available bandwidth. In other words, multi-rate content streams at the sending host are transmitted and switching operations between streams are performed depending on the available bandwidth. The current stream is forwarded to the receiving host in one piece and consequently, for every switching operation, the ongoing connection must be torn down and a new one must be set up. As a result, this operation is inefficient and decreases the data throughput.

[0005] Alternatively, it is possible to transmit only one stream of data to the receiving host and to discard data packets based on priority and time constraints. However, this leads to a loss of data and is therefore not an optimal solution.

[0006] The object subject to the present invention is to minimize the latency of the switching operation and to ensure fast recovery from an incorrect switching decision.

[0007] This object is solved according to the present invention with a method as set forth in claim 1. Preferred embodiments of the method are subject to various dependent claims.

[0008] The method of the present invention supposes the existence of multi-rate content data streams at the sending host. With the method of the present invention, it is possible to quickly switch between data streams that are variably segmented at the application layer as required during transmission. Instead of using data streams of fixed segment length, the data segments given to the socket are made smaller before, during and after the switching. For each content several video streams encoded with different rates are stored at the video server. Under normal sending conditions, i.e. the transmission has reached some equilibrium where the bandwidth is enough and the losses are not dominating, the video content is passed to the sending TCP socket in segments of the same size (irregardless of the stream encoding rate, the segment size is a function of the link, not of the encoding rate). Only when the server is notified (or knows by other means) that a stream switch might be needed, then it reduces the size of these segments.

[0009] Further, with the method of the present invention, it is neither necessary to modify the originally encoded data at the sending host or to modify the transport protocol nor to introduce additional control messages, making the method easy to implement.

[0010] Preferably, the switching is performed between data streams of different rate and the variable segmentation of the data streams is performed around the switching instant. Hence, a quick stream switching from the old to the new data stream can be performed. Further, preferably, the segmentation during, before or after the switching instance is made finer, such that the switching operation is more resilient to an incorrect decision and enables a quicker, further switching operation.

[0011] Advantageously, the step of switching between streams involves a feedback message (e.g. an RTSP command) i.e. communication between hosts when a switching operation is needed and when to make the necessary preparations therefor, for example, to start with finer segmentation.

[0012] It is furthermore of advantage if the feedback message includes at least one of the parameters, which comprise the length of data segment, an identifier of the data to be sent or a stream control command which makes the operation smoother and less susceptible to data losses.

[0013] Preferably, the length of the data segments is gradually increased after a switching instant to reduce interface communication, i.e. between application and TCP socket.

[0014] The present invention will be more fully understood from the following detailed description with reference to the accompanying drawings which show:

Figure 1: the basic architecture of a sending host and a receiving host connected by a communications network in which the method subject to the present invention can be performed,

Figure 2: a block diagram of a server architecture as the sending host in which the method of the present invention is performed, and

Aylo Exhibit 1018
Page 1574 of 2308

Figure 3: a block diagram of a client architecture.

[0015] Figure 1 illustrates the basic architecture of a communications network for connecting a video server as a sending host to a video client acting as a receiving host. The communications network is, for example, a UMTS core network, an external network, such as the internet or an internal intranet over which the server and the client communicate by means of a transport protocol. As transport protocol TCP is selected as an example, but it is clear to a skilled person that any other transport protocol for transmitting data can be employed.

[0016] In a more general way, it will now be explained how TCP handles the exchange of data between the server and the client.

[0017] First of all, the video server needs to transfer its data to the protocol and tell it to send same to the client. To progressively read and send data, the protocol uses a kind of buffer with added functionalities, a so-called TCP socket. Generally, sockets are unequivocally identified by the IP addresses of the correspondents, the protocols and ports which are used. Once a connection between two sockets is established, the communication can be controlled by use of interface calls, which, for example may comprise "create a socket", "bind a socket to a transport address", "accept" or "close a connection", "send", "receive" or "write" data between the sender and the receiver.

[0018] Wide-spread socket implementations are UNIX BSD 4.2 and Windows Winsock 2.0, where equivalent interface calls are implemented.

[0019] Each interface call has parameters, e.g. for the "send" call, parameters of relevance are the socket descriptor, the identifier of the data to be sent and the length of the data stream. Once a data stream is passed in one call, no single bit of this data stream can be modified. The only way to abort transfer of this data stream is to break down the connection.

[0020] Nevertheless, in order to close the connection, the application must explicitly issue a "close" call. Interface calls can be queued as, for example, in FTP. When asked for a file, the FTP server usually issues two interface calls for this transmission, namely a "send" call indicating the socket, the identifier of the data stream and the length thereof and a "close" call identifying the socket. The two interface calls are put into the queue and when the "send" call is completed, the connection is irreversibly broken down because the "close" call was next in the queue.

[0021] Alternatively, for example, in case of TELNET, the connection is established and the socket is set to wait for incoming data, i.e. commands typed on the keyboard.

[0022] Of course, the application can be configured such that the server only issues a send call and waits for the client to close. However, this is uncommon.

[0023] Additionally, it is assumed that the TCP protocol provides a means for asynchronously signaling certain information about events in the socket. Often in the specification, the information will be an error message. In other cases, there will be information relating to the completion of processing a SEND or RECEIVE or another interface call.

[0024] Referring again to Figure 1, the video server provides, e.g. an MPEG 4 compressed data stream, i. e. a given video content, which is made available to the TCP socket in the form of mono-rate or multi-rate streams. Multi-rate streams are defined for the purpose of this and the following description, as one or several data streams with the same video content at different encoding rates. The MPEG data streams are forwarded to the TCP socket. Subsequently, by means of an IP protocol, data is transmitted to the video client. A stream control protocol, e.g. RTSP, should be used to perform such operating as PLAY, PAUSE or STOP the transmission.

[0025] With reference to Figure 2, which shows the basic architecture of a video server 200, at the MPEG server application 210, video content is made available in a buffer 211 at different coding rates. A switch 212, which receives stream control RTSP commands from RTSP block 220, such as "Play", "Pause", "Tear down" or "Change rate", selects one of the data streams which is forwarded to the TCP socket 230 in form of an MPEG file.

[0026] It is clear to a skilled person that other stream control protocols than RTSP could be used at the server and the client. The stream control protocol typically instructs to perform play, forwarding, rewinding, pause or stop functions. Additionally, if multi-rate content is available, switch to a higher or lower encoding rate can be instructed.

[0027] Once a server has received a request for a file containing a chosen content encoded at a desired rate, the server application forwards each of the segments, in which the file was previously segmented, one after another to the TCP socket by means of consecutive interface calls, e.g. consecutive SEND calls. Unlike in current FTP applications, the connection is not torn down after each data transmission process is completed. The data connection remains open, so that there is no need to build up a new connection every time a segment is transmitted. Following this, the TCP socket forms with these segments packet data units (PDUs) for transmission to the client, as it finds most appropriate

[0028] Upon establishing a connection, the TCP socket selects a segment size for the data streams which the server finds most appropriate for the link under consideration of the channel bandwidth and starts transmitting.

[0029] Initially, the size of the segments is made small before, during and after a switching operation in order to avoid a large waiting time before a further switch operation can be performed in case the previous switch decision was incorrect. Subsequently, the size of the data segments can be increased gradually throughout the connection time.

Aylo Exhibit 1018
Page 1575 of 2308

[0030] Eventually, when there is a change in the available link bandwidth, the video client shall indicate a possible switch operation by means of, for example, an RTSP message and the server will start doing finer segmentation. Hence, the server switches between data streams of different encoding rates. This is done based on initial information and subsequently controlled by feedback from the client. Alternatively, this could be determined by the server, which would, however, impose processing requirements on the server. Also a combination of control mechanisms at the server and the client is possible. How the server or the client exactly makes the decision to perform a switching operation or indicates a possible anticipated switch is out of the scope of the present invention.

[0031] The basic architecture of a TCP streaming client 300 is illustrated in Figure 3. The client 330 typically consists of a TCP socket 310, an application layer buffer 320 and a buffer manager 330 for the application layer buffer.

[0032] The client 300 processes the data segments of variable length as they come. In response thereto, it sends ACKs and performs flow control in order to prevent an overflow of the TCP socket 310 while extracting the maximum bit rate possible from the link. To do this, the TCP socket indicates to the server, with each ACK, how much data it can receive until the socket is full using a window field in the header of the TCP protocol. If needed, the TCP client signals a zero-window telling the server to enter a "Persist" mode wherein no data is sent until the client allows to do so while the connection is maintained. The Persist mode and the signaling of a window field are standard functionalities of the TCP protocol. When the TCP circuit has capacity for more data, it signals this to the server. Streaming control commands are signalled to the server by means of RTSP commands.

[0033] The application layer buffer 320, also called "play-out buffer" receives data from the socket and transmits same to the MPEG client application 340. The manager 330 for the application layer buffer 320 receives a frame request from the MPEG client application and issues a socket request upon considering the state of the application layer buffer. The buffer manager 320 therefore insures that the bit rate with which the TCP socket 310 can deliver the new data to the application layer buffer 320 matches the bit rate at which the application layer buffer is emptied to the MPEG client application 340. The exact control mechanism of the buffer manager is out of the scope of the present invention.

[0034] Finally, from the MPEG client application, the data is transferred to a display 350 for visualization for the user.

Claims

1. A method of transmitting data streams between a sending host and a receiving host using a transport protocol comprising the steps of:

   making available data streams at the sending host in form of mono-rate or multi-rate content streams having same content but different encoding rates,

   establishing a connection between the sending host and the receiving host for the duration of transmission,

   transmitting data streams of the same content with a selected encoding rate to the receiving host, and

   separating the data streams in data segments of variable length at the sending host as required during the established connection.

2. The method according to claim 1, further comprising the steps of switching between data streams of different rate and performing the variable segmentation of the data streams around the switching instant.

3. The method according to claim 1 or 2, wherein the data streams are finer segmented during, before or after the switching instant.

4. The method according to claim 2, wherein the step of switching between streams involves a feedback message.

5. The method according to claim 4, wherein the feedback message includes signalling at least one of the parameters which comprise data segment length, an identifier of the data to be sent or a stream control command.

6. The method according to one of claims 1-5, wherein the receiving host communicates to the sending host when a stream switching is needed.

7. The method according to one of claims 1-6, wherein the sending host determines the timing of the switching instant dependent on the available bandwidth of the communication link.

8. The method according to one of claims 1-7, wherein the length of the data segments is gradually increased after a switching instant.

9. The method according to one of claims 1-8, wherein the received data streams at the receiving host are temporarily stored in an internal buffer.

10. The method according to one of claims 1-9, wherein the receiving host issues stream control commands

Aylo Exhibit 1018
Page 1576 of 2308

to the sending host depending on the state of the internal buffer.

Aylo Exhibit 1018
Page 1577 of 2308

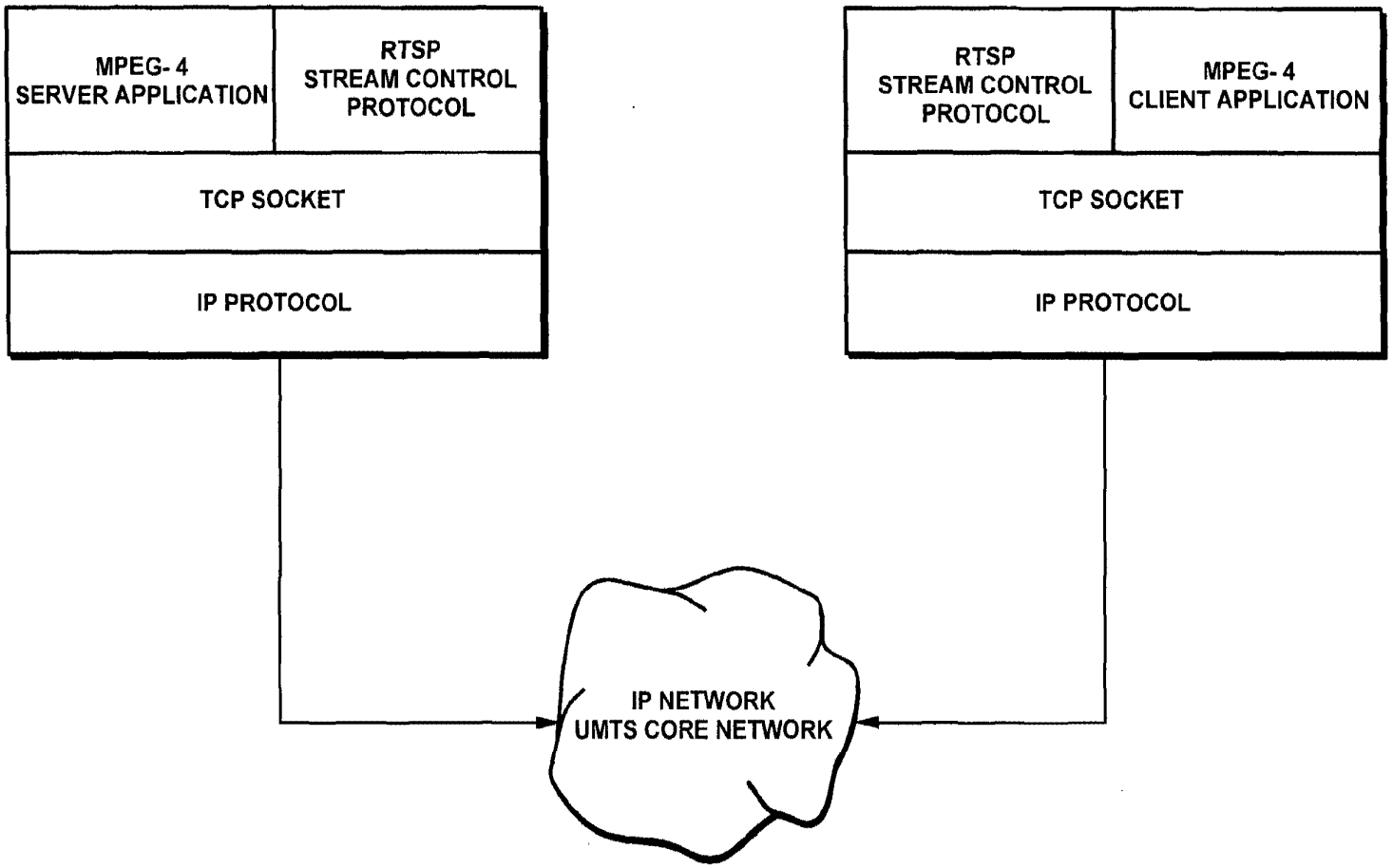


Fig. 1

Aylo Exhibit 1018
Page 1578 of 2308

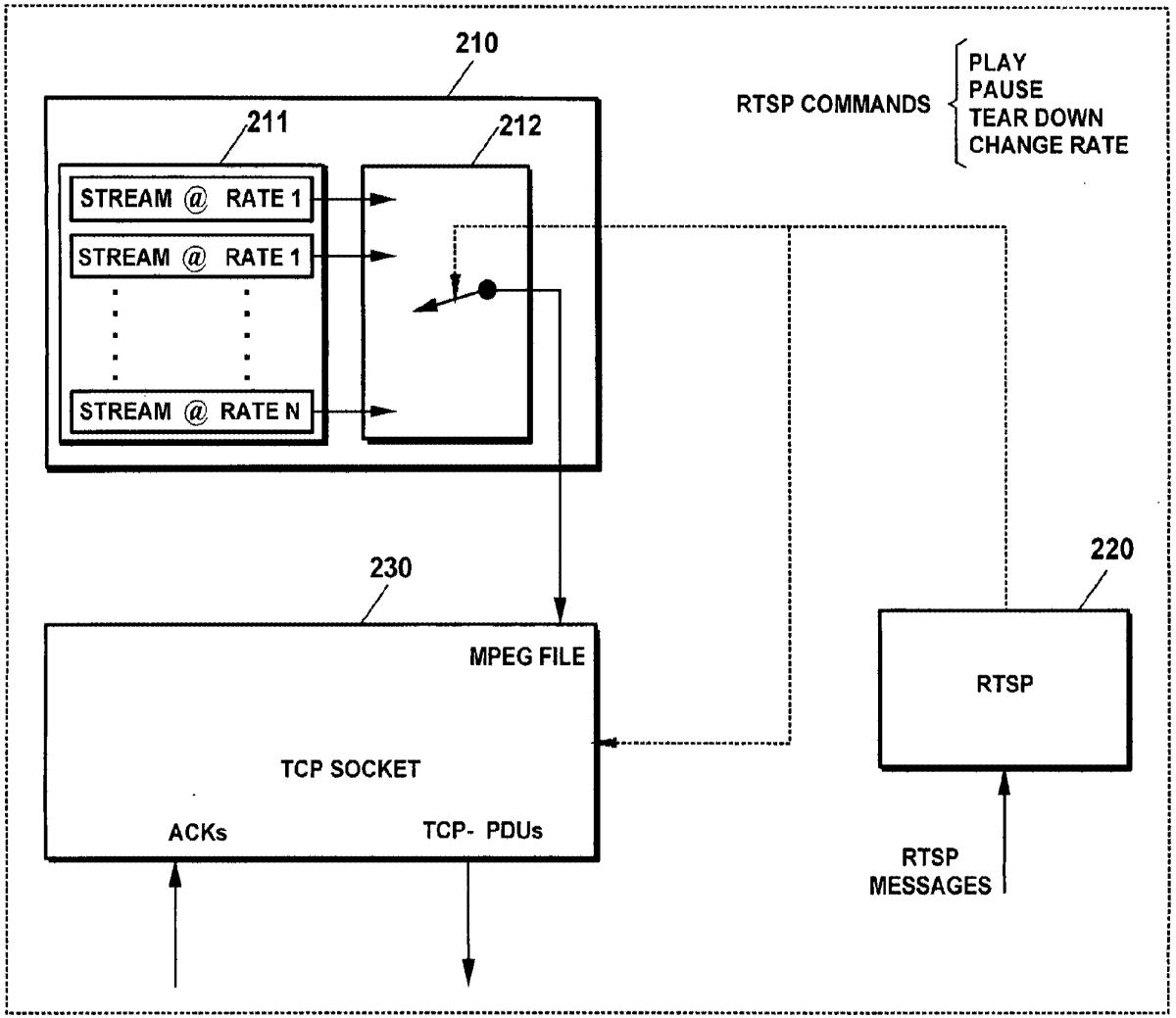


Fig. 2

Aylo Exhibit 1018
Page 1579 of 2308

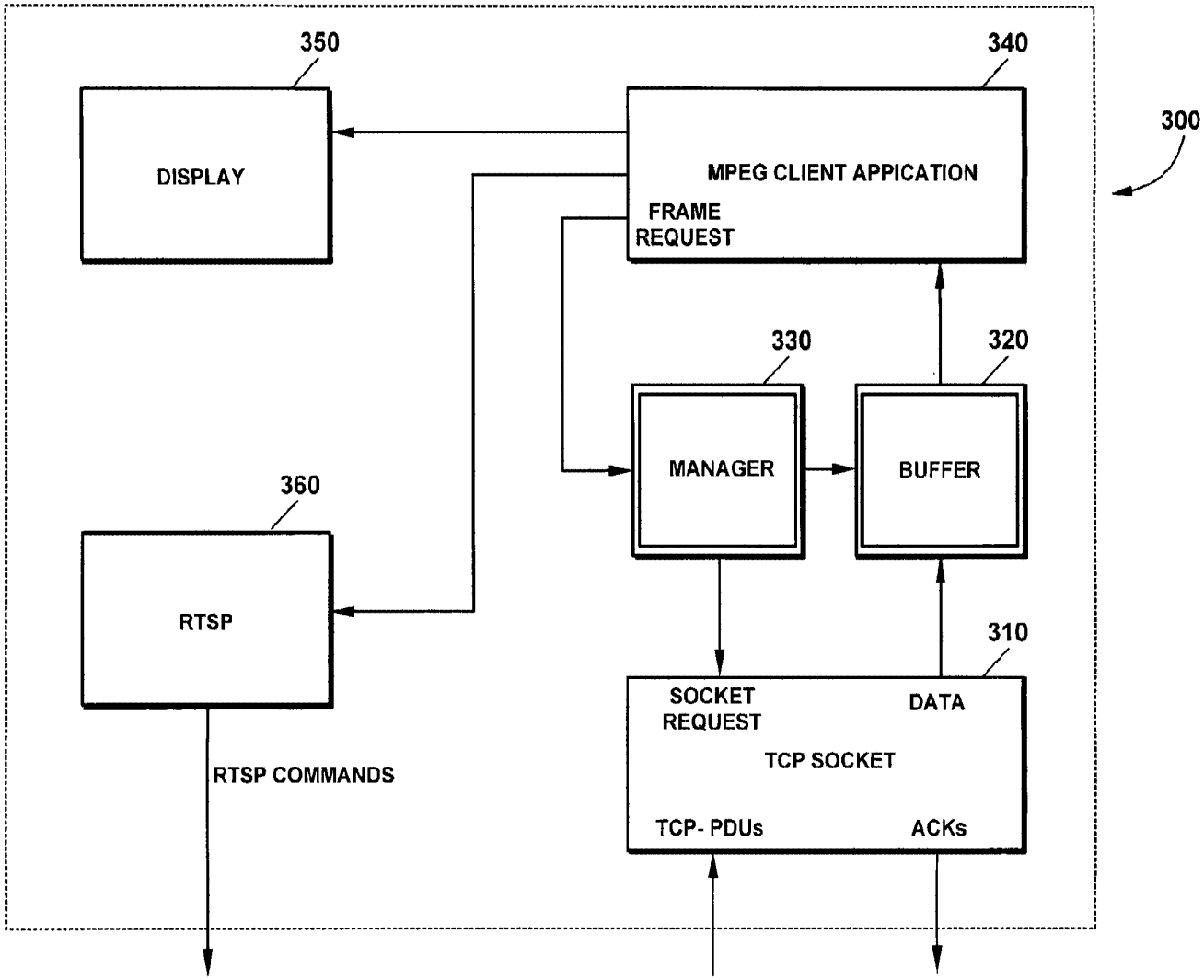

350
340
300
DISPLAY
MPEG CLIENT APPICATION
FRAME
REQUEST
330
320
MANAGER
BUFFER
360
RTSP
310
SOCKET
REQUEST
DATA
TCP SOCKET
RTSP COMMANDS
TCP- PDUs
ACKs

Fig. 3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 01 9052

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
Y	CONKLIN G J ET AL: "VIDEO CODING FOR STREAMING MEDIA DELIVERY ON THE INTERNET" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE INC. NEW YORK, US, vol. 11, no. 3, March 2001 (2001-03), pages 269-281, XP001093477 ISSN: 1051-8215 * page 271, left-hand column, line 35 - line 44; figures 1,2 * * page 270, left-hand column, line 11 - line 20 * ---	1-10	H04L29/06 H04L29/08
Y	EP 0 786 885 A (NOKIA TELECOMMUNICATIONS OY) 30 July 1997 (1997-07-30) * abstract * * page 2, column 2, line 14 - line 34 * ---	1-10	
X	WO 02 054284 A (LEISTAD GEIRR I ;DANKWORTH HARALD (NO); FAST SEARCH & TRANSFER ASA) 11 July 2002 (2002-07-11) * page 2, line 31 - page 3, line 7 * * page 4, line 6 - page 5, line 6; figure 1 * * page 9, line 18 - page 10, line 15 * * page 11, line 3 - line 28; figure 3 * --- -/--	1	
			TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04L

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	7 January 2003	Michael, T

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Aylo Exhibit 1018
Page 1581 of 2308

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 01 9052

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	HUI J Y ET AL: "CLIENT-SERVER SYNCHRONIZATION AND BUFFERING FOR VARIABLE RATE MULTIMEDIA RETRIEVALS" IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE INC. NEW YORK, US, vol. 14, no. 1, 1996, pages 226-237, XP000548824 ISSN: 0733-8716 * page 228, left-hand column, line 15 - line 33 * * page 228, right-hand column, line 14 - line 32 * ---	10	
A	JONGWON KIM ET AL: "TCP-friendly Internet video streaming employing variable frame rate encoding and interpolation" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, vol. 10, no. 7, October 2000 (2000-10), pages 1164-1177, XP002226167 ISSN: 1051-8215 * page 1164, right-hand column, line 25 - line 40; figure 1 * ---	7	TECHNICAL FIELDS SEARCHED (Int.Cl.7)
A	WO 02 052859 A (DURRANT SIMON ;LAKE JULIAN (GB); MULROY PAT (GB); PA CONSULTING SE) 4 July 2002 (2002-07-04) * page 3, line 12 - line 30 * -----	7	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	7 January 2003	Michael, T

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Aylo Exhibit 1018
Page 1582 of 2308

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 02 01 9052

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

07-01-2003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0786885 A	30-07-1997	FI 960382 A	27-07-1997
		EP 0786885 A2	30-07-1997
		US 5914960 A	22-06-1999
WO 02054284 A	11-07-2002	NO 20010056 A	05-07-2002
		WO 02054284 A1	11-07-2002
WO 02052859 A	04-07-2002	WO 02052859 A1	04-07-2002

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

Aylo Exhibit 1018
Page 1583 of 2308

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 January 2003 (30.01.2003)

PCT

(10) International Publication Number
WO 03/009581 A1

(51) International Patent Classification[7]: H04N 5/00, 7/24

(21) International Application Number: PCT/GB02/03307

(22) International Filing Date: 19 July 2002 (19.07.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
01306206.2 19 July 2001 (19.07.2001) EP

(71) Applicant *(for all designated States except US)*: **BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY** [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **JACOBS, Richard, James** [GB/GB]; 15 Through Duncans, Woodbridge, Suffolk IP12 4EA (GB). **WALKER, Matthew, David** [GB/GB]; 9 Thirlmere Court, Felixstowe, Suffolk IP11 9SN (GB). **MIRAS, Dimitrios** [GR/GR]; 12 Proti Parodos Avras, 24100 Kalamata (GR).

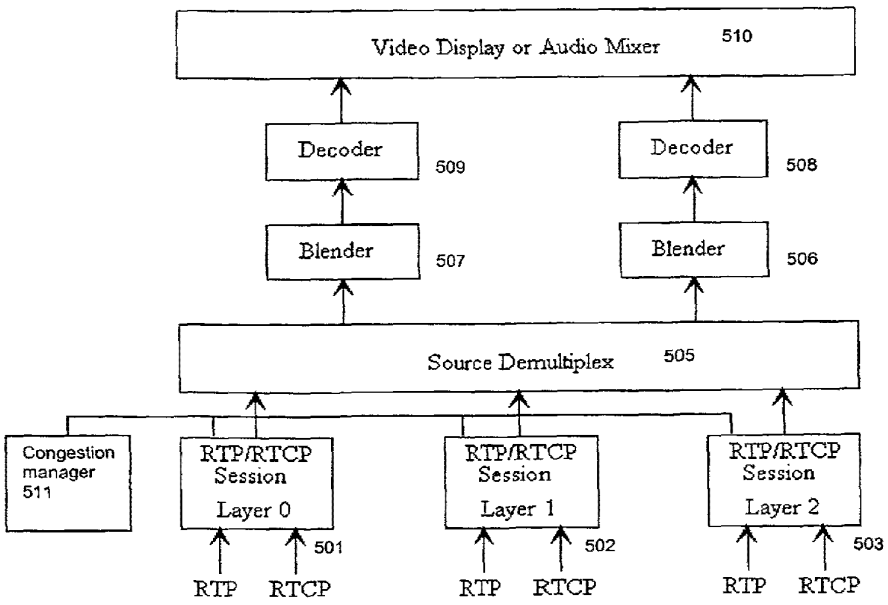
(74) Agent: **WILLIAMSON, Simeon, Paul**; 0T Group Legal, Intellectual Property Department, Holborn Centre 8th Floor, 120 Holborn, London EC1N 2TE (GB).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: VIDEO STREAM SWITCHING



(57) Abstract: A technique is described for providing a client on a packet based network with a stream of encoded video data. The system is able to maximise the bit-rate of the video stream by adapting to fluctuations in network capacity. The technique is characterised in that adaptation of the bit-rate of the transmitted encoded video data is timed to occur upon a scene change in the video sequence. In this way the interruption to the viewer when the perceived quality of the video sequence increases or decreases is minimised as it is 'hidden' in the scene change. The technique is described as applied to hierarchically encoded video data but equally may be applied to other encoding techniques which adapt to network conditions.

WO 03/009581 A1

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Video Stream Switching

The invention is in the field of video streaming over packet networks and in particular concerns the adaptive transmission of data in response to network congestion.

In recent years, the Internet has experienced a proliferation in transmission of real-time multimedia, mainly in the form of streamed audio-visual content, either delivered live or from pre-recorded sources. Furthermore, traditional forms of multimedia such as streaming and conferencing are being followed by applications with richer content such as Internet multi-channel TV and complex immersive environments. This increase in traffic will put strains on the network and it is therefore desirable that application programs are designed to respond to congestion if stability of the network is to be maintained. It is desirable that network conditions are monitored and output bit-rates adjusted to the available bandwidth.

There is a more immediate advantage to the user in having an adaptive output bit-rate which is that the highest possible transmission bit-rate that the network will allow is used and therefore the user will always receive the best possible image quality. There are a number of known approaches to adaptive quality video streaming, one of which is hierarchical coding. In this technique the original video data is encoded into a number of discreet streams called layers, where the first layer consists of basic data of a relatively poor quality and where successive layers represent more detailed information so that layers can be added to increase the image quality or layers can be taken away, depending on the available bandwidth.

As the bit-rate available to a session is subject to significant variations in the number of layers that are transmitted, quality fluctuations occur in the decoded image as layers are added or dropped. When layers are added or dropped frequently the fluctuations in quality may become disturbing to a viewer.

In a first embodiment of the present invention there is provided a method of operating a multimedia server, said method comprising:

providing a stream of video data representing a video sequence to an output of the multimedia server, wherein the output of the media server is connected to a packet based network,

measuring a property of the video data in order to determine the occurrence of a scene change in the video sequence,

detecting the available bandwidth on the network,

varying the bit-rate of the stream of video data,

wherein the method is characterised in that

variation in the bit-rate of the video data is controlled to occur in response to variations in the capacity of the network (203) and preferentially with a scene change in the video sequence.

The term preferentially used herein is meant to indicate that in certain circumstances (discussed in greater detail below) it may not be practical to wait for a change of scene to occur in the video sequence before varying the bit-rate of the video stream, and in such circumstances the bit-rate will be changed at points in the video stream which do not correspond to a change of scene.

The term scene change (or change of scene) is intended to refer to a sudden change in a video sequence within the space of a one or a very few frames such as typically occurs at a change of scene, whether or not there has been an actual change of scene.

Embodiments of the present invention will now be described, by way of example only, with reference to the following figures, where;

Figure 1 is a schematic diagram of the content-based inter-stream session bandwidth sharing architecture.

Figure 2 shows the order of packets in a layered coding system.

Figure 3 is a diagram of a media server.

Figure 4 is a diagram of the network interface of the media server shown in Figure 3.

Figure 5 is a diagram of a client.

Figure 6 is a diagram of a client according to a first embodiment of the present invention.

The client/ server arrangement for a known hierarchical streaming technology is shown in Figure 1. A media server 202 is provided which has access to compressed audiovisual data 201, which may be fed 'live' from an outside broadcast or may be pre-compressed and stored in a database. The data source 201 may be on the same premises as the media server 202 and linked via an intranet. The media server 202 runs on a suitable server computer and which has access to the Internet 203.

A video viewer, hitherto referred to as the client 204, running on a PC suitably configured to have access to the Internet 203, may connect to the media server 202 via the Internet 203 and thus the client 204 is able to access content. A suitable PC terminal is used.

Layered video compression is achieved with the 1998 version of H.263 but equally may be any other codec, such as MPEG4. Each layer in the hierarchy is coded in such a way as to allow the quality of individual pictures to be enhanced and their resolution to be increased, and additional pictures to be included to increase the overall picture rate, as explained with reference to Figure 2. Figure 2 illustrates a typical dependency between pictures in an H.263 scalable layered coder, with the boxes representing the frames for each layer and arrows showing the dependency between frames. The lowest row shows original, un-coded frames. The next row shows the lowest layer (Layer 0) of the hierarchy which is coded at half the frame rate of Layer 1. Frames in Layer 0 are predicted from the previously encoded frame, as in conventional video compression. Frames in Layer 1 may be predicted from the previously encoded frame in Layer 1 and, if present, the temporally simultaneous Layer 0 encoded frame. Frames in Layer 2 may be predicted from the previously encoded frame in Layer 2 and, if present, the temporally simultaneous Layer 1 or Layer 0 encoded frame. The H.263 specification allows for 15 layers; in the present embodiment server and client software is not limited in the number of layers that can be used, but in this case a database has been generated with video streams containing four layers.

Figure 3 shows the architecture of media server 201. In the present embodiment the audiovisual source is a database 301 of compressed video data and the media server 201 is responsible for reading the compressed data from a database, packetising and distributing it. The data is distributed according to the

Real-time Transport Protocol (RTP) which provides end-to-end network transport functions suitable for applications transmitting real-time data such as audio, video or simulation data. RTP does not address resource reservation and does not guarantee quality-of-service for real-time services. The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery and to provide minimal control and identification functionality. RTP and RTCP are designed to be independent of the underlying transport and network layers. The media server 201 has four major components: database reader 302, video RTP packetiser 303, audio RTP packetiser 305 and a network interface 304. Video and audio data are stored as files of compressed bits in the database 301. The database reader 302 retrieves and synchronises the compressed audio and video from the database 301.

The audio and video data is then sent to a RTP packetiser 303,305. Audio and video information is transmitted over the IP network using the User Datagram Protocol (UDP) and Real-time Transport Protocol (RTP) packetisation. UDP provides a checksum to detect transmission errors, but does not guarantee data delivery: packets may be lost, duplicated or re-ordered. RTP provides end-to-end delivery services, such as payload type identification, sequence numbering, time-stamping and delivery monitoring. RTP packetisers 303,305 attach the RTP Header and, in the case of video packets, the H.263 Payload Header which provides some protection from packet loss. The Payload Header contains information specific to the video stream, such as motion vector predictors, which is obtained by decoding the compressed bit stream.

The rate at which data is read from the database 301 is controlled from the network interface 304, which is illustrated in greater detail in Figure 4. With reference to Figure 4 the arrangement of the network interface 304 for the video data will now be described. A packet numbering module 401 assigns a number to each packet of data. These packet numbers are used by the client to assemble packets into the sequence required for decoding. The number of bits produced by compressing a picture is in general not constant, even when the transmission rate is constant. The first picture of a new scene will usually produce a larger number of bits than a picture where there is little movement or detail. The encoding process will use a control strategy together with an output data buffer to smooth these variations in preparation for transmission at a

constant bit rate. In the case of layered coding, each layer has its own control strategy and buffer as it has its own constant bit rate constraint to meet. The delay across each of these buffers will not be constant and will not be the same for each layer. Delay variation between the layers is therefore introduced at the source of the data. Transmission across a network may cause this delay variation to increase. A packet numbering scheme is therefore needed to ensure that a client can arrange the packets it receives into the order required for decoding. This scheme is required to handle the cases of the client not receiving all of the layers and of packet loss in the layers that it does receive.

Each layer is transmitted as an independent RTP Session on a separate IP address by a session handler 403, 405, 407, 409. The rate at which data is transmitted is controlled by the Transfer Rate Control module 402, which counts Layer 0 bytes to ensure that the correct number are transmitted in a given period of time. The transmission rate of the other layers is smoothed and locked to the rate of Layer 0 using First-In First-Out (FIFO) buffer elements 404, 406, 408.

The client 204 of the known hierarchical streaming technology will now be described with reference to Figure 5. Each RTP/RTCP Session associated with each layer of encoded data has a session handler 501, 502, 503, 504 at the client which is responsible for receiving RTP packets from the network.

The packets are then passed to a source demultiplex module 505. This receives packets from all layers and all sources, demultiplexes them by source, and routes them to a blender module 506, 507 for that source.

The blender module 506, 507 receives packets from all layers from one source in the order that they were received from the network. This may not be the order required for decoding because of packet inter-arrival jitter or packet loss. The blender module 506, 507 uses the packet numbers in the RTP headers to arrange the packets from each layer in order and then combines the packets from all layers together. The output of the blender module 506, 507 is a single stream of packets which are in the correct order for decoding. This is the same order as they come out of the packet numbering module 401 of the media server 202.

Packets are then sent to the decoder 508, 509 where the packets are decoded into 20ms blocks of audio samples or to video pictures. In the case of pictures, these are rendered to a window on the display 508.

Also provided is a congestion manager 511 to which the session handlers 501-504 report packet loss. If packets are being consistently lost, indicating network congestion, the congestion manager 511 will instruct the session handler responsible for the highest layer of compressed data to terminate the RTP/RTCP Session. Periodically the congestion manager 511 will instruct an experimental joining of a layer via the appropriate session handler to test whether there is available bandwidth in the network. If this experiment is successful, i.e. that substantially all of the packets of each layer are getting reaching the client, then the new layer will be adopted. In this way the maximum available bandwidth is employed.

It possible for the congestion manager 509 to instruct layers to be dropped and restored rapidly as the network bandwidth fluctuates. This could be annoying for a viewer and so a way of 'hiding' the shift is employed. The layered encoding method described above is adapted so that if a layer is to be dropped or added, the changeover preferentially occurs during a scene-change in the video data. A video scene is typically perceived as the number of consecutive frames within a video sequence that do not show significant changes in the video content. Within a video scene or shot, the camera action may be fixed or may exhibit a number of relatively uniform changes like panning, zooming, tracking etc. Scene changes may be recognised as abrupt transitions of the camera action or gradual transitions. In order to identify the scenes within a video sequence the assumption is that the levels of motion energy as well as those of luminance and colour do not change much between successive frames within a single scene. Techniques of scene boundary identification include pixel differencing, motion vector and block matching techniques. A very sudden change in the content of temporally adjacent frames will thus appear as a change of scene; clearly it is conceivable that such a change might not actually relate to a scene change at all, but, for example, may instead be the consequence of a large foreground object coming rapidly into view. However, such an abrupt change will nonetheless be a good place to add or remove a

session layer and thus the term scene change is meant to cover such large changes in frame content from one frame to another.

Figure 6 illustrates a system which detects congestion in the network and also imposes the extra condition of adding or dropping a layer upon a scene change. The server 601 comprises the components of the media server 202 of Figure 1 and the unit 603 comprises the components of the client 204 also shown in Figure 1. Video data from the audiovisual source is also passed to a content analyser 604 which segments the video sequence into scenes. The segmented video is then passed to an objective quality assessment module (OQAM) 605 which assesses the contribution to the perceived quality of a scene that each component layer provides.

These results are then passed to an inter layer adaptation module (ILAM) 607. The function of the ILAM 607 is to continuously calculate the number of layers that maximise the perceived quality for the session. The ILAM 607 also receives input from a sender congestion manager 608 which reports on the bandwidth available to the session on the network 203. The sender congestion manager 608 receives feedback from a client congestion manager 609 on the number of packets that have been received. If this matches the number of packets that were sent then the bandwidth is known to be the current transmission rate. If packets are being lost then the bandwidth available to the session is less than the transmission rate, in which case the sender congestion manager 608 informs the ILAM 607 that a layer should be dropped.

In order to select which layer should be dropped the ILAM 607 couples the bandwidth required by a layer with its contribution to the quality of the complete image, as calculated by the OQAM 605. The ILAM 607 performs an exhaustive search on all of the bandwidth/quality values. When the ILAM 607 has selected which layer is to be dropped from a particular scene, the timing of the drop is preferentially set to coincide with the transmission of that particular scene. In this way the drop in quality occurs with the scene change and is thus much less noticeable to a viewer than if the quality change had occurred during a scene. In the case that no packets have been lost for a preset period of time the sender congestion manager 608 will request the ILAM 607 to add in a layer to test

whether all of the available bandwidth is being employed. If no packets are lost during this experiment then the newly added layer is maintained. This process of experimentation with adding layers is continued until a significant proportion of packets are lost, in which instance the system can be confident that all of the available bandwidth is being employed. Again, the timing of the introduction of a layer is set to occur preferentially as the scene changes.

Under a given bit-rate allocation the levels of perceived quality do not change considerably within a scene, but scene cuts cause considerable changes in perceived quality, especially when the content features (spatial and motion energy) change a lot between subsequent scenes. As a consequence there will also be a significant difference in the corresponding quality scores for those successive scenes which may justify a rescheduling of the number of layers in the stream.

The invention is not limited in use to hierarchical encoding schemes. The invention is suitable for use in any encoding technique where the adaptation of transmission bit-rate to accommodate fluctuations in network capacity occurs. For instance the invention may be applied to a transcoding system where encoded data streams are transcoded from a high bit-rate to a low bit-rate or from a low bit-rate to a high bit-rate. The present invention would be suitable to reduce the impact for the viewer as the output bit-rate shifts in response to network conditions by timing the transition to occur upon a scene change in the encoded video sequence. Another example of an adaptive video streaming technique to which the invention may be applied is where multiple independent video streams of different bit-rates are transmitted. In this case the client chooses which stream to accept based on session bandwidth. The client may switch from one stream to another as bandwidth fluctuates; the present invention ensures that the switch is timed to coincide with a scene change in the encoded video stream.

Naturally, there may be times when a change of scene is such a long way away that it is advantageous to switch from one bit-rate to another other than during a change of scene. For example, consider the case where multiple independent video streams at different bit-rates are available for transmission by a media server to a client as, for example, described in co-pending European patent

application No. 00310594.7 the contents of which are hereby incorporated herein by way of reference. In such a case, the server may be capable of transmitting a first stream at a bit-rate of 500kbit/s and a second, higher quality stream at a bit-rate of 1500kbit/s. The client may initially request that the server transmit the first stream at a transmission rate of 1000kbit/s. If the network is not congested and all of the packets transmitted are successfully received by the client, the receive buffer at the client will start to fill with data at a rate of 500kbit/s, since the client will only be removing data from the buffer at a rate of 500kbit/s. After say 10 seconds of filling up the buffer at this rate, the client will have a buffer of ten seconds worth of data at which point it may decide it can attempt to receive the higher bit-rate second stream of video data from the server and thus sends an appropriate request to the server to this effect. If, however, the server is aware that the client has a receive buffer of 5 Mbytes size, it knows that it may continue sending data from the first stream at the rate of 1000kbit/s for at least another 150 seconds before the receive buffer overflows, causing problems for the client. Therefore, the server may attempt to wait for a specified period to see if a scene change occurs during this interval. If so, a switch to the second higher bit-rate signal is not made until the change of scene occurs. Of course, if there is no change of scene within the determined period, the server switches to the higher rate anyway. In this example, a period of only ten seconds is deemed appropriate as the waiting time.

Note that instead of simply waiting to see if a change of scene occurs in the specified period and switching at the end of the period if no such change of scene is detected, an alternative method would be to pre-analyse the video to be sent (clearly this only applies to pre-stored video data and not live video) and to note when changes of scene occur. In such a case, the server, upon receipt of a request from the client to switch streams, could search to see if a suitable change of scene will occur within the predetermined period and if not to switch immediately to the new bit stream.

In the present example, upon switching to the higher rate bit stream, the client may have requested a transmission rate of 1500kbit/s corresponding to the rate at which the data will be drawn from the receive buffer by the client. In such a case, the buffer size of 10 seconds should remain constant so long as all of the

transmitted packets are successfully received by the client. However, in the event of congestion on the network, a proportion of packets may fail to arrive at the client. In such a case, the server will be warned of this via the RTCP. If the congestion is sufficiently severe, the server may deduce that the buffer is in danger of emptying which would cause a break in the video displayed by the client to occur. To prevent this, the server may switch back to the lower bit rate stream. Via the notification of how many packets are being lost, the server can deduce how long it will be before the buffer is emptied. For this time, the server can wait to see if a change of scene occurs, and if so, the new stream will be switched to at that point. Note that it would also be possible for the client to simply request that the new stream be switched to upon detecting that its receive buffer is emptying at an unsustainable rate.

Note that the amount of data in the buffer is actually of less significance than the rate of change of the amount of data in the buffer. Thus it is preferably this quantity which either the server or the client measures in order to determine whether to change from one bit stream to another (or whether to add or drop a layer in the first example).

CLAIMS

1. A method of operating a multimedia server (202), said method comprising;

providing a stream of video data representing a video sequence to an output of the multimedia server (202), wherein the output of the media server (202) is connected to a packet based network (203),

measuring a property of the video data in order to determine the occurrence of a scene change in the video sequence,

detecting the available bandwidth on the network (203),

varying the bit-rate of the stream of video data,

wherein the method is characterised in that

variation in the bit-rate of the video data is controlled to occur in response to variations in the capacity of the network (203) and preferentially with a scene change in the video sequence.

2. A method of operating a multimedia server (202) in accordance with claim 1 wherein upon detecting a decrease in the available bandwidth the bit-rate of the video data is decreased.

3. A method of operating a multimedia server (202) in accordance with claim 1 wherein the bit-rate of the video data is periodically increased in order to ensure that all of the available bandwidth is being used.

4. A method of operating a multimedia server 202 in accordance with claims 1 to 3 wherein said stream of video data is comprised of a plurality of hierarchically encoded layers.

5. A method of operating a multimedia server (202) in accordance with claim 4 wherein the bit-rate of the video data is varied by adding or dropping layers from the stream of data transferred to the output port.

6. A method of operating a multimedia server (202) in accordance with claims 4 or 5 wherein the objective quality contribution of each layer to the

overall quality of the video sequence is measured in order to determine which of the layers is to be added or dropped from the stream of data transferred to the output port.

7. A method of operating a multimedia server 202 in accordance with claims 1 to 3 wherein said stream of video data is comprised of a plurality of independently encoded flows each of which encoded at different bit-rates.

8. A method of operating a multimedia server (202) in accordance with claim 7 wherein the bit-rate of the video data is varied by switching between independently encoded flows.

9. A method of operating a multimedia server 202 in accordance with any of the preceding claims in which the source 201 of video data is stored in encoded form in a database.

10. A method of operating a multimedia server 202 in accordance with any of claims 1 to 8 in which the source 201 of video data is encoded in real-time.

11. A method of operating a multimedia server 202 in accordance with any of the preceding claims wherein, upon detecting a variation in the capacity of the network such as to render it appropriate for a change in bit rate to occur, the server waits for a predetermined time to attempt to detect a change of scene, and if such a change of scene is detected within the predetermined time, the change of bit rate is caused to occur at the time of the change of scene, but if a change of scene is not detected within the predetermined time, then the change of bit rate is caused to occur at the expiry of the predetermined time.

12. A method of operating a multimedia server 202 in accordance with any of the preceding claims wherein, upon detecting a variation in the capacity of the network such as to render it appropriate for a change in bit rate to occur, the server determines if there is going to be a change of scene within a predetermined time, and if it is determined that there will be such a change of scene within the predetermined time, the change of bit rate is caused to occur at the time of the change of scene, but if it is determined that there is not going to

be a change of scene within the predetermined time, then the change of bit rate is caused to occur substantially immediately after having made such a determination.

13. The method of claim 12 further including the step of, prior to providing a stream of video data to the output of the media server, processing the video data to determine the positions within the video data in which changes of scenes occur and making this information available for permitting whether a scene change will occur within a predetermined period at any point along the stream of video data.

14. A multimedia server (202) comprising;

a reader (302) for reading video data representing a video sequence from a source (301),

a scene change detector (604) for detecting changes of scene in the video sequence,

a bit-rate controller (607) for controlling the rate at which the video data is transferred from the reader to an output port, wherein the output port is capable of communicating with a client (204) on a packet network (203);

the media server further comprising means (608) for detecting available bandwidth between the multimedia server (202) and client (204) on the packet network,

characterised in that

the bit-rate controller (607) is arranged to vary the bit-rate of the video data transferred to the output port in dependence on the detected bandwidth preferentially in correlation with a scene change detected by the scene change detector (604).

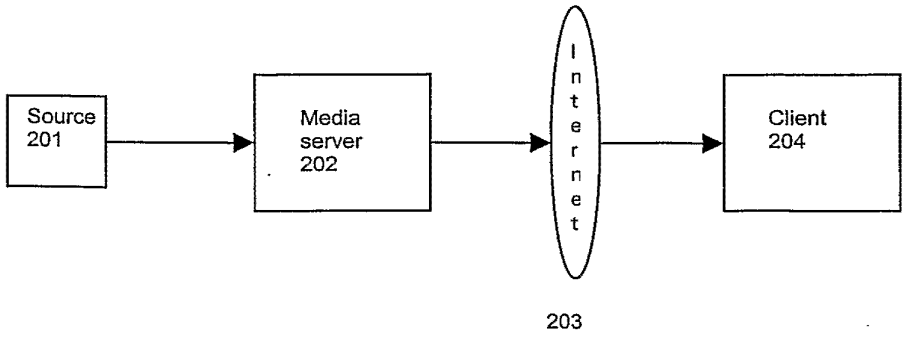


Figure 1

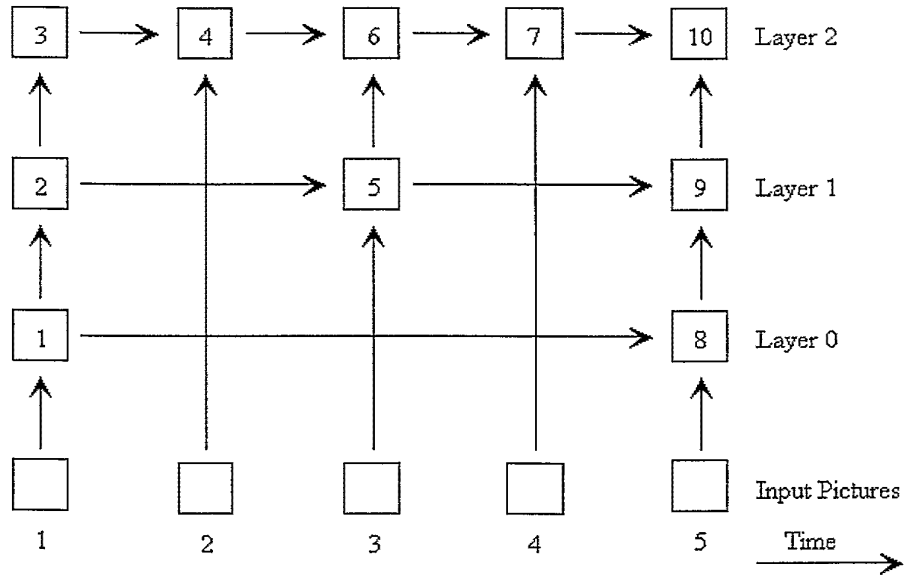


Figure 2

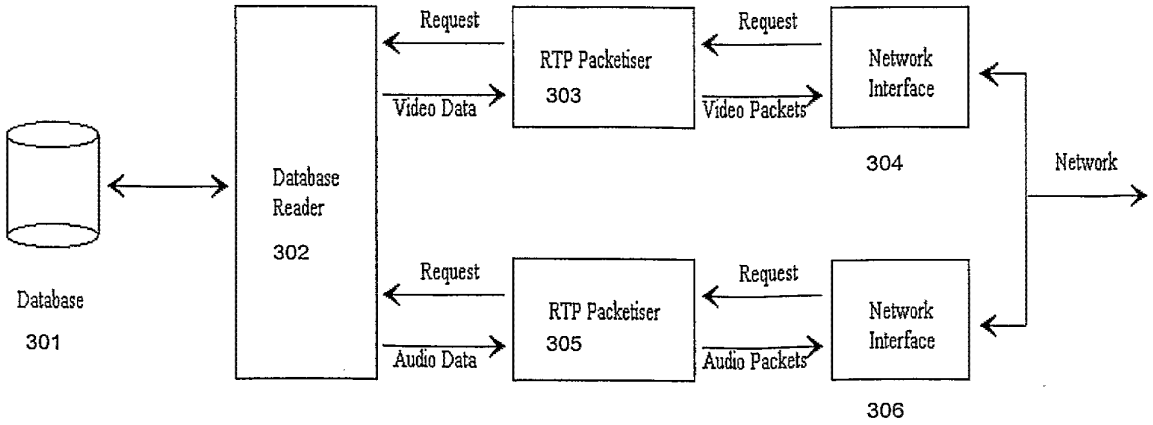


Figure 3

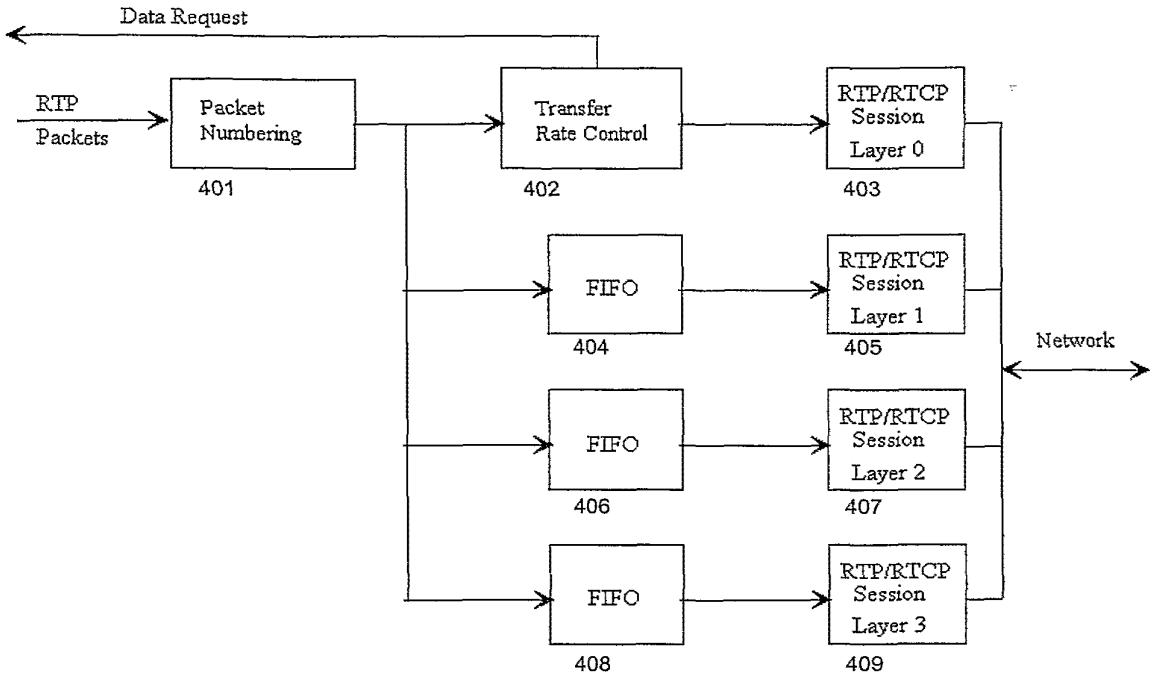


Figure 4

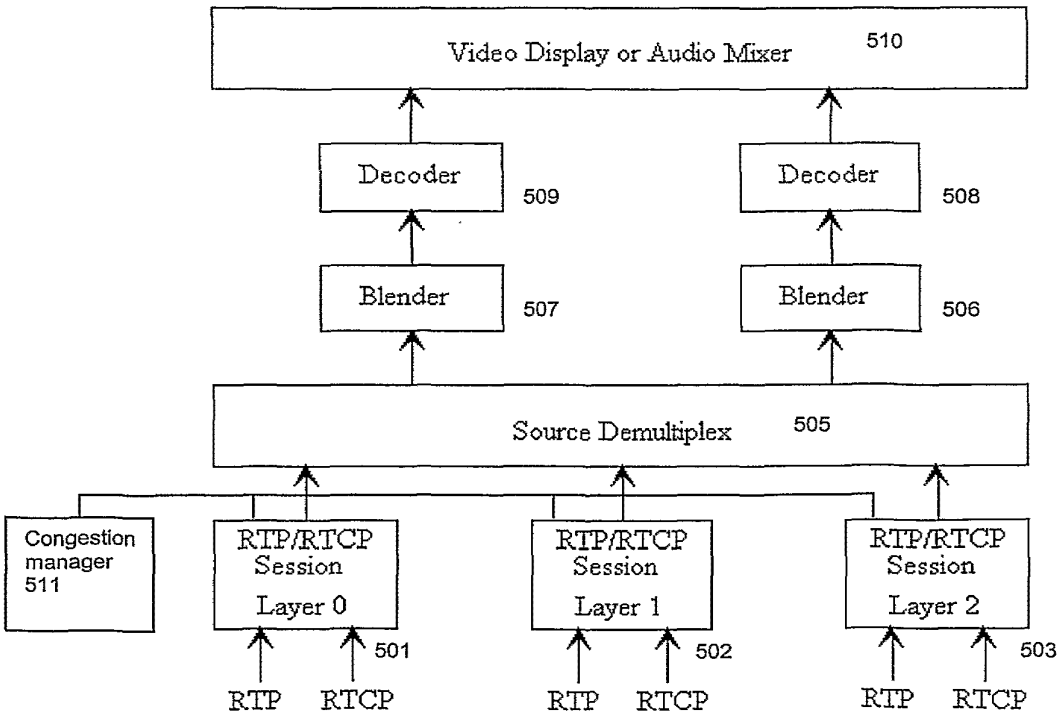


Figure 5

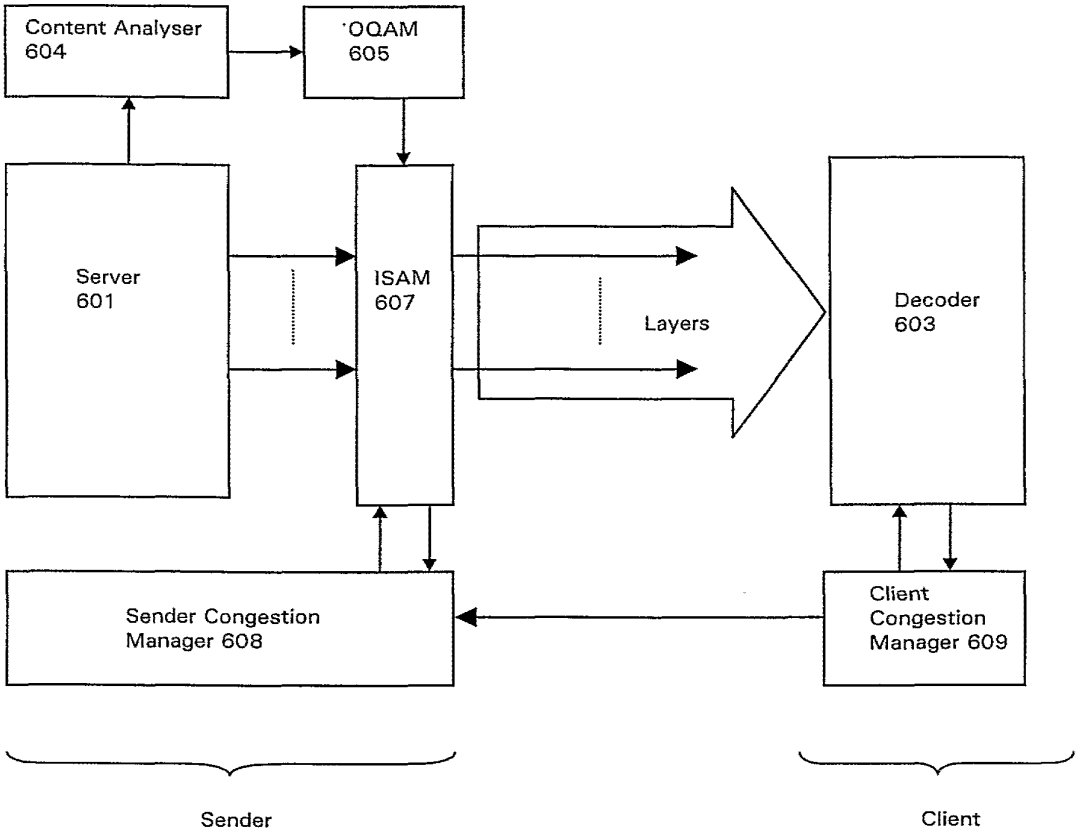

Content Analyser
604
OQAM
605
Server
601
ISAM
607
Layers
Decoder
603
Sender Congestion
Manager 608
Client
Congestion
Manager 609
Sender
Client

Figure 6

INTERNATIONAL SEARCH REPORT

ational Application No
PCT/GB 02/03307

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N5/00 H04N7/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, PAJ, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 014 694 A (AHARONI AMIR ET AL) 11 January 2000 (2000-01-11)	1-10,14
Y	abstract column 2, line 29 - line 65 column 3, line 62 -column 4, line 34 column 8, line 54 -column 9, line 35 column 10, line 34 - line 49 column 14, line 34 -column 15, line 6	11-13
X	ANONYMOUS: "Method to Deliver Scalable Video across a Distributed Computer System" IBM TECHNICAL DISCLOSURE BULLETIN, IBM CORP. NEW YORK, US, vol. 5, no. 37, 1 May 1994 (1994-05-01), pages 251-256, XP002079392 ISSN: 0018-8689	1-10,14
A	the whole document	11-13

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 September 2002	23/09/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Fantini, F

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6 038 000 A (HURST JR ROBERT NORMAN) 14 March 2000 (2000-03-14)	11-13
A	column 2, line 52 -column 3, line 22 column 4, line 20 - line 49 column 6, line 9 - line 45 column 7, line 1 - line 24 column 13, line 46 -column 14, line 18 column 20, line 13 - line 30	1,14
A	EP 1 079 631 A (SONY UK LTD) 28 February 2001 (2001-02-28) page 3, line 1 - line 4	1-14
A	US 5 990 945 A (DENTENEER THEODORUS J J ET AL) 23 November 1999 (1999-11-23) abstract column 3, line 13 - line 35 column 4, line 42 - line 44 column 5, line 37 - line 39	11-13

INTERNATIONAL SEARCH REPORT

Information on patent family members

ıtional Application No
PCT/GB 02/03307

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 6014694	A	11-01-2000	AU	7928998 A	19-01-1999
			EP	0992160 A1	12-04-2000
			WO	9900984 A1	07-01-1999
			JP	2002511216 T	09-04-2002
US 6038000	A	14-03-2000	US	6137834 A	24-10-2000
			EP	0954924 A1	10-11-1999
			JP	2001509354 T	10-07-2001
			WO	9832281 A1	23-07-1998
			EP	0901718 A1	17-03-1999
			JP	2002503402 T	29-01-2002
			KR	2000016034 A	25-03-2000
			WO	9745965 A1	04-12-1997
EP 1079631	A	28-02-2001	GB	2353653 A	28-02-2001
			EP	1079631 A1	28-02-2001
			JP	2001119305 A	27-04-2001
US 5990945	A	23-11-1999	CN	1199532 A	18-11-1998
			EP	0860087 A1	26-08-1998
			WO	9809442 A1	05-03-1998
			JP	11514825 T	14-12-1999

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 January 2003 (09.01.2003)

PCT

(10) International Publication Number
WO 03/003760 A2

(51) International Patent Classification[7]: H04Q 3/00

(21) International Application Number: PCT/US02/16590

(22) International Filing Date: 25 June 2002 (25.06.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/893,364 26 June 2001 (26.06.2001) US

(71) Applicant: **BURST.COM** [US/US]; 613 Fourth Street, Suite 201, Santa Rosa, CA 95404 (US).

(72) Inventor: **ALLEN, Arthur, Douglas**; 1322 Isabelle Avenue, Mountain View, CA 94040 (US).

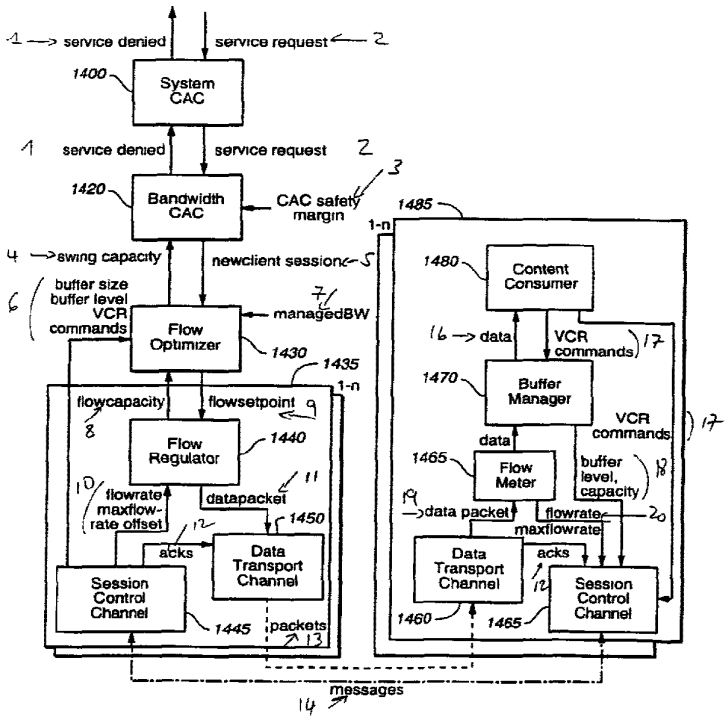
(74) Agents: **SWISS, Gerald, F.** et al.; BURNS, DOANE, SWECKER & MATHIS, L.L.P., P.O. BOX 1404, Alexandria, VA 22313-1404 (US).

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD FOR CONNECTION ACCEPTANCE CONTROL AND RAPID DETERMINATION OF OPTIMAL MULTI-MEDIA CONTENT DELIVERY OVER NETWORKS



(57) Abstract: A method of bandwidth allocation for delivery of stored digital content from at least one server device to at least one client device by way of a network is disclosed. The method begins by prescribing a control variable which represents a target flow rate from the server device to each client device. Next, time-varying constraints on the flow rate of the content are determined. A cost function of the control variable for each client is determined. The cost function corresponds to a maximized value of the control variable. Finally, bandwidth is prescribed to each client based upon the value of the control variable maximized by the cost function. In this respect, the method achieves optimal allocation of bandwidth between the server and the respective clients.

WO 03/003760 A2

METHOD FOR CONNECTION ACCEPTANCE CONTROL AND RAPID DETERMINATION OF OPTIMAL MULTI-MEDIA CONTENT DELIVERY OVER NETWORKS

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] This invention relates to the field of delivery of multimedia content over a variety of networks. More specifically, it pertains to multimedia servers which service many clients simultaneously for the delivery of multimedia content which is used and played back at each client. It addresses methods for determining optimal delivery rates to each client and methods for determining whether new clients may be accepted without diminishing the quality of service to existing clients.

Status of the Prior Art

[0002] In the history of multimedia program delivery, some in the industry have long advocated the use of large client-side buffers and faster-than-real-time content delivery over a network as offering the best of all worlds: a jitter-free viewing experience and a cost-effective utilization of the network resources at hand. Few systems, however, go very far in addressing how to schedule clients or a method for accepting new clients. Real-time systems, often known as streaming systems, can schedule new clients in a very simple manner -- if sufficient bandwidth remains for the added real-time stream, then the client may be accepted. However, such systems do not maximize the number of simultaneous clients. On the other hand, faster than real-time delivery, sometimes known as store-and-forward systems, opens up the possibility for more flexible scheduling procedures to control and optimize the number of simultaneous clients while ensuring a high level of quality of service.

[0003] The methods for such call acceptance and flow modulation that have been proposed in the prior art have been largely ad-hoc and also incomplete. These have been ad-hoc in the sense that there has been no guiding rationale for their selection from among many possible and potentially superior alternatives. The methods have also been incomplete insofar as they did not address the question of whether any given incoming request for service should be accepted or denied. Video-on-demand systems, or more generally, any system in which a multimedia server is designed to serve multiple clients over a network to deliver bounded content, can benefit from the use of such flow modulation techniques and call acceptance procedures.

Optimal Content Flow Modulation

[0004] One time-honored way of designing methods of the class required here is to re-cast the problem to be solved as an optimization problem, in which one seeks to maximize a designated value function moment-by-moment, subject to a set of real-world operational constraints which will typically vary over time. Accordingly, given a set of clients and associated sessions, an optimal delivery procedure continuously establishes content flow rates from the content server to each of its clients so as to maximize aggregate value according to the governing value function.

[0005] This approach holds several advantages: 1) optimization problems are well understood, and are tractable by a large and diverse collection of computational methods; 2) if it exists, a global solution that is obtained is arguably optimal by construction, and thus superior or equal to all other.

[0006] The present invention teaches the method of optimizing two particular value functions:

1) total data delivered (maximize throughput).

2) total delivery charges (maximize charges).

[0007] The first value function does not distinguish one customer from another and will deliver as much data as possible from server to clients irrespective of the characteristics of the latter. The second value function favors the service of high paying customers. It can easily be seen that the first function is a special case of the second one whereby all clients are charged equally.

[0008] As will be seen in this disclosure, optimizing for these functions and identifying the necessary constraints requires a new and unique perspective that is specifically designed for the multimedia environment. Moreover, the disclosed methods are specifically designed to account for and accommodate real-world scenarios of today's networks. Consequently many variations of the method are presented to accommodate various scenarios.

SUMMARY OF THE INVENTION

Call/Connection Acceptance Control (CAC)

[0009] A CAC procedure is responsible for deciding whether a candidate for service can be accommodated without jeopardizing sessions already in progress at the present time or at some time in the future; failing that it must decide whether a service request should be queued for a time or rejected.

Flow Modulation

[0010] Flow modulation methods are those portions of the system which manage the communication and data flow between the server and the clients. Collectively, these methods provide the multimedia data to the client and provide the server with the information about the state of the transmission, playback, user status and network status. These parameters are further used by the present invention in the CAC procedures. In fact, as will be shown, the proposed CAC procedures are tightly integrated with the flow modulation methods.

Adaptation to Variations in Network Capacity

[0011] Operational constraints may change over time. For instance, one might elect to vary the total bandwidth available for multimedia content delivery according to the time of day. Alternatively, exogenous data flows on the network may cause unexpected disturbances by usurping available bandwidth and impeding the delivery of data along established session channels. The content delivery strategy of the present invention includes the ability to adapt to scheduled as well as unexpected disturbances so as to minimize unwanted disruptions of services.

Burst Transmissions Provide the Opportunity to Adapt

[0012] The present invention, due to it's faster-than-realtime transmissions (also know as burst transmissions), which are realized by use of high-bandwidth networks and large client cache or intermediate storage, provides an opportunity to adapt to changing network conditions. In contrast real-time (streaming) systems are essentially designed for worst-case scenarios: each client must be assumed to constantly use the complete real-time playback bandwidth. Such a system is unable to adapt to any derivation from this scenario. For example, take the simple case where the total server bandwidth is 100% utilized by all clients playing back the streaming video. Should any network condition change, such as a temporary

decrease in available bandwidth over the network, then one or more clients' playback is interrupted, and the system can not recover from such a condition until the bandwidth is regained. Even worse if a single client presses pause either that unused bandwidth must remain reserved and no more clients can be accepted, or that paused client is pushed out in order to service the new client. In essence little or no CAC procedure may be implemented.

[0013] In contrast the present invention burst transmits portions of a program and immediately 'gets ahead of itself', thus allowing headroom for a myriad of methods to intelligently handle new clients, client interactivity and possible network fluctuations.

[0014] Methods are taught for optimally determining the flow rate to each client. Methods are also taught for accepting or rejecting new clients; these call-acceptance methods are tightly coupled with said flow rate modulation methods. A series of constraint expressions are presented which govern the methods for determining the flow rates and acceptance of new clients. Linear programming techniques are used to optimally solve these expressions. Various embodiments are presented including scenarios for multiple-rate tariffs, and time-of-day bandwidth variations.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] These as well as other features of the present invention will become more apparent upon reference to the drawings wherein:

Figure 1 depicts the flow of control and/or data between the different stations of a content delivery session;

Figure 2 illustrates the Entity Data Model;

Figure 3 geometrically illustrates the linear programming problem statement;

Figure 4a geometrically illustrates an expansion of the flow optimization problem statement;

Figure 4b geometrically illustrates the method for rapid determination of flow rates to maximize flow;

Figure 5 illustrates a method for implementing flow modulation to maximize flow;

Figure 6 illustrates a method for implementing flow modulation for maximized charges;

Figure 7 illustrates typical content flow;

Figure 8 illustrates typical server swing capacity;

Figure 9 illustrates a method for call-acceptance and control (CAC);

Figure 10 illustrates planned constraints on maximum flow;

Figure 11 illustrates a method for call-acceptance and control (CAC) with scheduled flow changes;

Figure 12 illustrates stratification of services;

Figure 13 illustrates a method for call-acceptance and control (CAC) for maximal charge;

Figure 14 is a system block diagram for a system implementing the present optimization of client bandwidth;

Figure 15 is a flowchart showing an algebraic method for calculating the optimization of client bandwidth;

Figure 16 is a flowchart showing an algebraic method for calculating the optimization of client costs; and

Figure 17 illustrates optimized viewing of a live event.

DETAILED DESCRIPTION OF THE INVENTION

Data & Control Flows (Figure 1)

[0016] Figure 1 depicts the flow of control and/or data between the different stations of a content delivery session. As shown a client attempts a connection 100 and manifests itself to the Content Selection subsystem by means of a low bandwidth control channel (not shown). Next the client is authenticated and a selection is made 110, typically with the aid of browser software. If the client is not authenticated, it is dismissed from the system 120. If the client has been authenticated and a program selected for viewing then the rate of service is set at this point 130, perhaps according to the selection that was made, or some contractual stipulation. The client is now placed on the service queue of the CAC subsystem 140. A client that is made to wait too long will eventually balk 150. Assuming this does not occur, the CAC subsystem 140 will eventually allocate a channel to the client and open a session 160. Control now devolves upon the Content Flow Modulator (not shown) which starts the flow of content from server to client 170. Subsequent capacity changes, whether predictable or not, may force an abrupt termination of a session in progress 180. Otherwise the session runs to completion 190.

Entity Data Model (Figure 2, listing, table)

[0017] The entities entering into our discussion are depicted in Figure 2. Client 200 maintains certain data associated with this entity; as shown but not labeled, which includes without limitation, status, id and costOfService. The other entities also each include unlabeled but depicted data. The diagram further depicts the relationship between each entity. As shown, client 200 is assigned a session 240. Client 200 employs a channel 210. Client 200 selects contentSelection 230. Session 240 delivers content through channel 210. Server 220 modulates channel 210. Server 200 contains contentSelection 210. Server 220 accepts, defers or denies client 200 and contentSelection 230 is associated with session 240.

[0018] Furthermore Figure 2 depicts the various one-to-many relationships. Each client 200 employs one channel 210. Client 200 may or may not receive one of channel 210, as depicted by the 0/1 notation. Similarly, client 200 may or may not receive a session 240. However, whenever client 200 does receive a session 240, it will always receive a channel 210 since channel 210 and session 240 are allocated as a pair. One or more (N) of client 200 may select one of contentSelection 230. And server 220 contains one or more (N) of contentSelection 230. Each one of contentSelection 230 is associated with 0-N of session 240. Each session 240 delivers content through one of channel 210. And server 220 modulates one or more (N) of channel 210.

[0019] A more detailed list of each entity of Fgure 2, and each one's associated description, data elements and function calls is listed below. This listing closely resembles that of object-oriented programming. As such, 'methods' represent the ability to obtain or modify data, while 'attributes' represent data which is directly associated with that particular entity. The listing also includes information relating to one embodiment wherein software programming specifics are disclosed, such as