

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B or equivalent) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5. If the Power of Attorney by Applicant form is not accompanied by this transmittal form or an equivalent, the Power of Attorney will not be recognized in the application.

Application Number	Unassigned
Filing Date	Unassigned
First Named Inventor	DAVID F. BRUECK
Title	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING
Art Unit	Unassigned
Examiner Name	
Attorney Docket Number	066.0141X1C7

SIGNATURE of Applicant or Patent Practitioner

Signature	/BRETT A. CARLSON/	Date	2019-01-18
Name	Brett A. Carlson	Telephone	480.385.5060
Registration Number	39,928		

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications.

☒ *Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. 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 1.11 and 1.14. This collection is estimated to take 3 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, 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.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number

Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)



I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above:

70560

OR



I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:



The address associated with the above-mentioned Customer Number

OR



The address associated with Customer Number:

OR

Firm or
Individual Name

Address

City

State

Zip

Country

Telephone

Email

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

DISH Technologies L.L.C.



Inventor or Joint Inventor (title not required below)



Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)



Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)



Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.48(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature

Max S. Gratton

Date (Optional)

7/6/2010

Name

Max S. Gratton

Title

Director & Senior Corporate Counsel of DISH Technologies L.L.C.

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.



Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. 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 1.11 and 1.14. This collection is estimated to take 3 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 1480, Alexandria, VA 22313-1480. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1480, Alexandria, VA 22313-1480.

If you need assistance in completing the form, call 1-800-PTO-8100 and select option 2.

AMENDMENTS TO THE SPECIFICATION

PLEASE AMEND PARAGRAPH 0001 OF THE SPECIFICATION AS FOLLOWS:

[0001] This application is a continuation of U.S. Patent Application Serial No. 16/004,056 filed on June 8, 2018, which is a continuation of U.S. Patent Application Serial No. 15/414,027 (now US Patent No. 9,998,516) filed on January 24, 2017, which is a continuation of U.S. Patent Application No. 14/719,122 filed on May 21, 2015, which is a continuation of U.S. Patent Application No. 14/106,051 filed on December 13, 2013 (now U.S. Patent No. 9,071,668), which is a continuation of U.S. Patent Application Serial No. 13/617,114, filed on September 14, 2012 (now US Patent No. 8,612,624), which is a continuation of U.S. Patent No. 12/906,940 filed on October 18, 2010 (now US Patent No. 8,402,156), which is a continuation of U.S. Patent Application Number 11/673,483, filed on February 9, 2007 (now US Patent No. 7,818,444), which is a continuation-in-part of Application No. 11/116,783, filed on April 28, 2005 (now U.S. Patent No. 8,868,772), which claims the benefit of U.S. Provisional Application No. 60/566,831, filed on April 31, 2004, all of which are incorporated herein by reference.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial. No.: TBD Confirmation No.: TBD
Examiner: TBD Art Unit: TBD
Filed: Herewith Customer No.: 70560
Title: APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE
CONTENT STREAMING
Attorney Docket No.: 066.0141x1C7

PRELIMINARY AMENDMENT

Mail Stop AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

*****FILED ELECTRONICALLY USING THE PTO-EFS SYSTEM***

Commissioner:

Please amend the above-referenced application prior to examination.

Amendments to the Specification begin on the following page (page 2).

Remarks begin on page 3 of this paper.

REMARKS

This preliminary amendment is being filed at the same time as the attached application to update the priority information in paragraph 0001. Please enter the foregoing amendments prior to substantive examination. No new matter has been added, and support for all of the claimed features can be found in the originally-filed application.

This application is being filed as a “Track One” application with a certification and request for prioritized examination. If any documents or fees that are required to process this application as a Track One application are not submitted herewith, ***please consider this as a petition under Rule 181 (or any other appropriate rule) for appropriate relief, and as authorization to charge Deposit Account No. 50-2091 for any fees that may be due***, including any fees to process the Track One application and any associated petitions or other documents.

Should the Examiner have any suggestions that would advance prosecution at any time during prosecution, we would welcome a telephone call to (480) 385-5069 or an email to bcarlson@lkglobal.com. Recognizing that Internet communications are not secure, we hereby authorize the USPTO to communicate with the undersigned and practitioners in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via video conferencing, instant messaging, or electronic mail. We understand that a copy of these communications will be made of record in the application file. (MPEP 502.03). Without limiting the foregoing, we further authorize the use of email for informal correspondence, including the exchange of proposed amendments or other draft documents.

If any additional documents or fees are required to consider this submittal or to otherwise prevent abandonment of this application, please consider this as a petition and request for any needed relief and as authorization to charge Deposit Account No. 50-2091 for any fees that may be due.

Respectfully submitted,
LORENZ & KOPF, LLC

Dated: January 18, 2019

By: /Brett A. Carlson /
Brett A. Carlson
Reg. No. 39,928

bcarlson@lkglobal.com

480-385-5069

LORENZ & KOPF, LLC (*formerly Ingrassia, Fisher & Lorenz, PC*)

Customer No. 70560

**CERTIFICATION AND REQUEST FOR PRIORITIZED EXAMINATION
UNDER 37 CFR 1.102(e) (Page 1 of 1)**

First Named Inventor:	DAVID F. BRUECK	Nonprovisional Application Number (if known):	066.0141X1C7
Title of Invention:	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

APPLICANT HEREBY CERTIFIES THE FOLLOWING AND REQUESTS PRIORITIZED EXAMINATION FOR THE ABOVE-IDENTIFIED APPLICATION.

1. The processing fee set forth in 37 CFR 1.17(i), the prioritized examination fee set forth in 37 CFR 1.17(c), and if not already paid, the publication fee set forth in 37 CFR 1.18(d) have been filed with the request. The basic filing fee, search fee, examination fee, and any required excess claims and application size fees are filed with the request or have been already been paid.
2. The application contains or is amended to contain no more than four independent claims and no more than thirty total claims, and no multiple dependent claims.
3. The applicable box is checked below:

I. ☒ Original Application (Track One) - Prioritized Examination under § 1.102(e)(1)

- i. (a) The application is an original nonprovisional utility application filed under 35 U.S.C. 111(a). This certification and request is being filed with the utility application via EFS-Web.
---OR---
(b) The application is an original nonprovisional plant application filed under 35 U.S.C. 111(a). This certification and request is being filed with the plant application in paper.
- ii. An executed oath or declaration under 37 CFR 1.63 is filed with the application.

II. ☐ Request for Continued Examination - Prioritized Examination under § 1.102(e)(2)

- i. A request for continued examination has been filed with, or prior to, this form.
- ii. If the application is a utility application, this certification and request is being filed via EFS-Web.
- iii. The application is an original nonprovisional utility application filed under 35 U.S.C. 111(a), or is a national stage entry under 35 U.S.C. 371.
- iv. This certification and request is being filed prior to the mailing of a first Office action responsive to the request for continued examination.
- v. No prior request for continued examination has been granted prioritized examination status under 37 CFR 1.102(e)(2).

Signature /BRETT A. CARLSON/	Date 2019-01-18
Name (Print/Typed) Brett A. Carlson	Practitioner Registration Number 39928

Note: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required in accordance with 37 CFR 1.33 and 11.18. Please see 37 CFR 1.4(d) for the form of the signature. If necessary, submit multiple forms for more than one signature, see below*.

☒ *Total of 1 forms are submitted.

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.

APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. Patent Application Number 11/673,483, filed on February 9, 2007, which is a continuation-in-part of Application No. 11/116,783, filed on April 28, 2005, which claims the benefit of U.S. Provisional Application No. 60/566,831, filed on April 30, 2004, all of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0002] The invention relates to video streaming over packet switched networks such as the Internet, and more particularly relates to adaptive-rate shifting of streaming content over such networks.

DESCRIPTION OF THE RELATED ART

[0003] The Internet is fast becoming a preferred method for distributing media files to end users. It is currently possible to download music or video to computers, cell phones, or practically any network capable device. Many portable media players are equipped with network connections and enabled to play music or videos. The music or video files (hereinafter "media files") can be stored locally on the media player or computer, or streamed or downloaded from a server.

[0004] "Streaming media" refers to technology that delivers content at a rate sufficient for presenting the media to a user in real time as the data is received. The data may be stored in memory temporarily until played and then subsequently deleted. The user has the immediate satisfaction of viewing the requested content without waiting for the media file to completely download. Unfortunately, the audio/video quality that can be received for real time presentation is constrained by the available bandwidth of the user's network connection. Streaming may be used to deliver content on demand (previously recorded) or from live broadcasts.

[0005] Alternatively, media files may be downloaded and stored on persistent storage devices, such as hard drives or optical storage, for later presentation. Downloading complete media files can take large amounts of time depending on the network connection. Once downloaded, however, the content can be viewed repeatedly anytime or anywhere. Media files prepared for downloading usually are encoded with a higher quality audio/video than can be delivered in real time. Users generally dislike this option, as they tend to want to see or hear the media file instantaneously.

[0006] Streaming offers the advantage of immediate access to the content but currently sacrifices quality compared with downloading a file of the same content. Streaming also provides the opportunity for a user to select different content for viewing on an ad hoc basis, while downloading is by definition restricted to receiving a specific content selection in its entirety or not at all. Downloading also supports rewind, fast forward, and direct seek operations, while streaming is unable to fully support these functions. Streaming is also vulnerable to network failures or congestion.

[0007] Another technology, known as “progressive downloads,” attempts to combine the strengths of the above two technologies. When a progressive download is initiated, the media file download begins, and the media player waits to begin playback until there is enough of the file downloaded that playback can begin with the hope that the remainder of the file will be completely downloaded before playback “catches up.” This waiting period before playback can be substantial depending on network conditions, and therefore is not a complete or fully acceptable solution to the problem of media presentation over a network.

[0008] Generally, three basic challenges exist with regard to data transport streaming over a network such as the Internet that has a varying amount of data loss. The first challenge is reliability. Most streaming solutions use a TCP connection, or “virtual circuit,” for transmitting data. A TCP connection provides a guaranteed delivery mechanism so that data sent from one endpoint will be delivered to the destination, even if portions are lost and retransmitted. A break in the continuity of a TCP connection can have serious consequences when the data must be delivered in real-time. When a network adapter detects delays or losses in a TCP connection, the adapter “backs off” from transmission attempts for a moment and then slowly resumes the original

transmission pace. This behavior is an attempt to alleviate the perceived congestion. Such a slowdown is detrimental to the viewing or listening experience of the user and therefore is not acceptable.

[0009] The second challenge to data transport is efficiency. Efficiency refers to how well the user's available bandwidth is used for delivery of the content stream. This measure is directly related to the reliability of the TCP connection. When the TCP connection is suffering reliability problems, a loss of bandwidth utilization results. The measure of efficiency sometimes varies suddenly, and can greatly impact the viewing experience.

[0010] The third challenge is latency. Latency is the time measure from the client's point-of-view, of the interval between when a request is issued and the response data begins to arrive. This value is affected by the network connection's reliability and efficiency, and the processing time required by the origin to prepare the response. A busy or overloaded server, for example, will take more time to process a request. As well as affecting the start time of a particular request, latency has a significant impact on the network throughput of TCP.

[0011] From the foregoing discussion, it should be apparent that a need exists for an apparatus, system, and method that alleviate the problems of reliability, efficiency, and latency. Additionally, such an apparatus, system, and method would offer instantaneous viewing along with the ability to fast forward, rewind, direct seek, and browse multiple streams. Beneficially, such an apparatus, system, and method would utilize multiple connections between a source and destination, requesting varying bitrate streams depending upon network conditions.

SUMMARY OF THE INVENTION

[0012] The present invention has been developed in response to the present state of the art, and in particular, in response to the problems and needs in the art that have not yet been fully solved by currently available content streaming systems. Accordingly, the present invention has been developed to provide an apparatus, system, and method for adaptive-rate content streaming that overcome many or all of the above-discussed shortcomings in the art.

[0013] The apparatus for adaptive-rate content streaming is provided with a logic unit containing a plurality of modules configured to functionally execute the necessary steps. These modules in the described embodiments include a receiving module configured to receive media content, a streamlet module configured to segment the media content and generate a plurality of sequential streamlets, and an encoding module configured to encode each streamlet as a separate content file.

[0014] The encoding module is further configured to generate a set of streamlets for each of the sequential streamlets. Each streamlet may comprise a portion of the media content having a predetermined length of time. The predetermined length of time may be in the range of between about .1 and 5 seconds.

[0015] In one embodiment, a set of streamlets comprises a plurality of streamlets having identical time indices, and each streamlet of the set of streamlets has a unique bitrate. The receiving module is configured to convert the media content to raw audio or raw video. The encoding module may include a master module configured to assign an encoding job to one of a plurality of host computing modules in response to an encoding job completion bid. The job completion bid may be based on a plurality of computing variables selected from a group consisting of current encoding job completion percentage, average encoding job completion time, processor speed, and physical memory capacity.

[0016] A system of the present invention is also presented for adaptive-rate content streaming. In particular, the system, in one embodiment, includes a receiving module configured to receive media content, a streamlet module configured to segment the media content and generate a plurality of sequential streamlets, each streamlet comprising a portion of the media content having a predetermined length of time, and an encoding

module configured to encode each streamlet as a separate content file and generate a set of streamlets.

[0017] The system also includes a plurality of streamlets having identical time indices and each streamlet of the set of streamlets having a unique bitrate. The encoding module comprises a master module configured to assign an encoding job to one of a plurality of host computing modules in response to an encoding job completion bid.

[0018] A method of the present invention is also presented for adaptive-rate content streaming. In one embodiment, the method includes receiving media content, segmenting the media content and generating a plurality of sequential streamlets, and encoding each streamlet as a separate content file.

[0019] The method also includes segmenting the media content into a plurality of streamlets, each streamlet comprising a portion of the media content having a predetermined length of time. In one embodiment, the method includes generating a set of streamlets comprising a plurality of streamlets having identical time indices, and each streamlet of the set of streamlets having a unique bitrate.

[0020] Furthermore, the method may include converting the media content to raw audio or raw video, and segmenting the content media into a plurality of sequential streamlets. The method further comprises assigning an encoding job to one of a plurality of host computing modules in response to an encoding job completion bid, and submitting an encoding job completion bid based on a plurality of computing variables.

[0021] Reference throughout this specification to features, advantages, or similar language does not imply that all of the features and advantages that may be realized with the present invention should be or are in any single embodiment of the invention. Rather, language referring to the features and advantages is understood to mean that a specific feature, advantage, or characteristic described in connection with an embodiment is included in at least one embodiment of the present invention. Thus, discussion of the features and advantages, and similar language, throughout this specification may, but do not necessarily, refer to the same embodiment.

[0022] Furthermore, the described features, advantages, and characteristics of the invention may be combined in any suitable manner in one or more embodiments. One skilled in the relevant art will recognize that the invention may be practiced without one

or more of the specific features or advantages of a particular embodiment. In other instances, additional features and advantages may be recognized in certain embodiments that may not be present in all embodiments of the invention.

[0023] These features and advantages of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] In order that the advantages of the invention will be readily understood, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments that are illustrated in the appended drawings.

Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings, in which:

[0025] Figure 1 is a schematic block diagram illustrating one embodiment of a system for dynamic rate shifting of streaming content in accordance with the present invention;

[0026] Figure 2a is a schematic block diagram graphically illustrating one embodiment of a media content file;

[0027] Figure 2b is a schematic block diagram illustrating one embodiment of a plurality of streams having varying degrees of quality and bandwidth;

[0028] Figure 3a is a schematic block diagram illustrating one embodiment of a stream divided into a plurality of source streamlets;

[0029] Figure 3b is a schematic block diagram illustrating one embodiment of sets of streamlets in accordance with the present invention;

[0030] Figure 4 is a schematic block diagram illustrating in greater detail one embodiment of the content module in accordance with the present invention;

[0031] Figure 5a is a schematic block diagram illustrating one embodiment of an encoder module in accordance with the present invention;

[0032] Figure 5b is a schematic block diagram illustrating one embodiment of parallel encoding of streamlets in accordance with the present invention;

[0033] Figure 6a is a schematic block diagram illustrating one embodiment of a virtual timeline in accordance with the present invention;

[0034] Figure 6b is a schematic block diagram illustrating an alternative embodiment of a VT in accordance with the present invention;

[0035] Figure 6c is a schematic block diagram illustrating one embodiment of a QMX in accordance with the present invention;

[0036] Figure 7 is a schematic block diagram graphically illustrating one embodiment of a client module in accordance with the present invention;

[0037] Figure 8 is a schematic flow chart diagram illustrating one embodiment of a method for processing content in accordance with the present invention;

[0038] Figure 9 is a schematic flow chart diagram illustrating one embodiment of a method for viewing a plurality of streamlets in accordance with the present invention; and

[0039] Figure 10 is a schematic flow chart diagram illustrating one embodiment of a method for requesting streamlets within an adaptive-rate shifting content streaming environment in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0040] Many of the functional units described in this specification have been labeled as modules, in order to more particularly emphasize their implementation independence. For example, a module may be implemented as a hardware circuit comprising custom VLSI circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices or the like.

[0041] Modules may also be implemented in software for execution by various types of processors. An identified module of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions which may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together, but may comprise disparate instructions stored in different locations which, when joined logically together, comprise the module and achieve the stated purpose for the module.

[0042] Indeed, a module of executable code may be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices. Similarly, operational data may be identified and illustrated herein within modules, and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set, or may be distributed over different locations including over different storage devices, and may exist, at least partially, merely as electronic signals on a system or network.

[0043] Reference throughout this specification to “one embodiment,” “an embodiment,” or similar language means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, appearances of the phrases “in one embodiment,” “in an embodiment,” and similar language throughout this specification may, but do not necessarily, all refer to the same embodiment.

[0044] Reference to a signal bearing medium may take any form capable of generating a signal, causing a signal to be generated, or causing execution of a program of machine-

readable instructions on a digital processing apparatus. A signal bearing medium may be embodied by a transmission line, a compact disk, digital-video disk, a magnetic tape, a Bernoulli drive, a magnetic disk, a punch card, flash memory, integrated circuits, or other digital processing apparatus memory device. In one embodiment, a computer program product including a computer useable medium having a computer readable program of computer instructions stored thereon that when executed on a computer causes the computer to carry out operations for multi-bitrate content streaming as described herein.

[0045] Furthermore, the described features, structures, or characteristics of the invention may be combined in any suitable manner in one or more embodiments. In the following description, numerous specific details are provided, such as examples of programming, software modules, user selections, network transactions, database queries, database structures, hardware modules, hardware circuits, hardware chips, etc., to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art will recognize, however, that the invention may be practiced without one or more of the specific details, or with other methods, components, materials, and so forth. In other instances, well-known structures, materials, or operations are not shown or described in detail to avoid obscuring aspects of the invention.

[0046] Figure 1 is a schematic block diagram illustrating one embodiment of a system 100 for dynamic rate shifting of streaming content in accordance with the present invention. In one embodiment, the system 100 comprises a content server 102 and an end user station 104. The content server 102 and the end user station 104 may be coupled by a data communications network. The data communications network may include the Internet 106 and connections 108 to the Internet 106. Alternatively, the content server 102 and the end user 104 may be located on a common local area network, wireless area network, cellular network, virtual local area network, or the like. The end user station 104 may comprise a personal computer (PC), an entertainment system configured to communicate over a network, or a portable electronic device configured to present content. For example, portable electronic devices may include, but are not limited to, cellular phones, portable gaming systems, and portable computing devices.

[0047] In the depicted embodiment, the system 100 also includes a publisher 110, and a web server 116. The publisher 110 may be a creator or distributor of content. For

example, if the content to be streamed were a broadcast of a television program, the publisher 110 may be a television or cable network channel such as NBC®, or MTV®. Content may be transferred over the Internet 106 to the content server 102, where the content is received by a content module 112. The content module 112 may be configured to receive, process, and store content. In one embodiment, processed content is accessed by a client module 114 configured to play the content on the end user station 104. In a further embodiment, the client module 114 is configured to receive different portions of a content stream from a plurality of locations simultaneously. For example, the client module 114 may request and receive content from any of the plurality of web servers 116.

[0048] Content from the content server 102 may be replicated to other web servers 116 or alternatively to proxy cache servers 118. Replicating may occur by deliberate forwarding from the content server 102, or by a web, cache, or proxy server outside of the content server 102 asking for content on behalf of the client module 114. In a further embodiment, content may be forwarded directly to web 116 or proxy 118 servers through direct communication channels 120 without the need to traverse the Internet 106.

[0049] Figure 2a is a schematic block diagram graphically illustrating one embodiment of a media content (hereinafter “content”) file 200. In one embodiment, the content file 200 is distributed by the publisher 110. The content file 200 may comprise a television broadcast, sports event, movie, music, concert, etc. The content file 200 may also be live or archived content. The content file 200 may comprise uncompressed video and audio, or alternatively, video or audio. Alternatively, the content file 200 may be compressed using standard or proprietary encoding schemes. Examples of encoding schemes capable of use with the present invention include, but are not limited to, DivX®, Windows Media Video®, Quicktime Sorenson 3®, On2, OGG Vorbis, MP3, or Quicktime 6.5/ MPEG-4® encoded content.

[0050] Figure 2b is a schematic block diagram illustrating one embodiment of a plurality of streams 202 having varying degrees of quality and bandwidth. In one embodiment, the plurality of streams 202 comprises a low quality stream 204, a medium quality stream 206, and a high quality stream 208. Each of the streams 204, 206, 208 is a copy of the content file 200 encoded and compressed to varying bit rates. For example, the low quality stream 204 may be encoded and compressed to a bit rate of 100 kilobits per

second (kbps), the medium quality stream 206 may be encoded and compressed to a bit rate of 200 kbps, and the high quality stream 208 may be encoded and compressed to 600 kbps.

[0051] Figure 3a is a schematic block diagram illustrating one embodiment of a stream 302 divided into a plurality of source streamlets 303. As used herein, streamlet refers to any sized portion of the content file 200. Each streamlet 303 may comprise a portion of the content contained in stream 302, encapsulated as an independent media object. The content in a streamlet 303 may have a unique time index in relation to the beginning of the content contained in stream 302. In one embodiment, the content contained in each streamlet 303 may have a duration of two seconds. For example, streamlet 0 may have a time index of 00:00 representing the beginning of content playback, and streamlet 1 may have a time index of 00:02, and so on. Alternatively, the time duration of the streamlets 304 may be any duration smaller than the entire playback duration of the content in stream 302. In a further embodiment, the streamlets 303 may be divided according to file size instead of a time index and duration.

[0052] Figure 3b is a schematic block diagram illustrating one embodiment of sets 306 of streamlets in accordance with the present invention. As used herein, the term “set” refers to a group of streamlets having identical time indices and durations but varying bitrates. In the depicted embodiment, the set 306a encompasses all streamlets having a time index of 00:00. The set 306a includes encoded streamlets 304 having low, medium, and high 204, 206, 208 bitrates. Of course each set 306 may include more than the depicted three bitrates which are given by way of example only. One skilled in the art will recognize that any number of streams having different bitrates may be generated from the original content 200.

[0053] As described above, the duration of one streamlet 304 may be approximately two seconds. Likewise each set 306 may comprise a plurality of streamlets 304 where each streamlet 304 has a playable duration of two seconds. Alternatively, the duration of the streamlet 304 may be predetermined or dynamically variable depending upon a variety of factors including, but not limited to, network congestion, system specifications, playback resolution and quality, etc. In the depicted embodiment, the content 200 may be formed

of the plurality of sets 306. The number of sets 306 may depend on the length of the content 200 and the length or duration of each streamlet 304.

[0054] Figure 4 is a schematic block diagram illustrating in greater detail one embodiment of the content module 112 in accordance with the present invention. The content module 112 may comprise a capture module 402, a streamlet module 404, an encoder module 406, a streamlet database 408, and the web server 116. In one embodiment, the capture module 402 is configured to receive the content file 200 from the publisher 110. The capture module 402 may be configured to “decompress” the content file 200. For example, if the content file 200 arrives having been encoded with one of the above described encoding schemes, the capture module 402 may convert the content file 200 into raw audio and/or video. Alternatively, the content file 200 may be transmitted by the publisher in a format 110 that does not require decompression.

[0055] The capture module 402 may comprise a capture card configured for TV and/or video capture. One example of a capture card suitable for use in the present invention is the DRC-2500 by Digital Rapids of Ontario, Canada. Alternatively, any capture card capable of capturing audio and video may be utilized with the present invention. In a further embodiment, the capture module 402 is configured to pass the content file to the streamlet module 404.

[0056] The streamlet module 404, in one embodiment, is configured to segment the content file 200 and generate source streamlets 303 that are not encoded. As used herein, the term “segment” refers to an operation to generate a streamlet of the content file 200 having a duration or size equal to or less than the duration or size of the content file 200. The streamlet module 404 may be configured to segment the content file 200 into streamlets 303 each having an equal duration. Alternatively, the streamlet module 404 may be configured to segment the content file 200 into streamlets 303 having equal file sizes.

[0057] The encoding module 406 is configured to receive the source streamlets 303 and generate the plurality of streams 202 of varying qualities. The original content file 200 from the publisher may be digital in form and may comprise content having a high bit rate such as, for example, 2mbps. The content may be transferred from the publisher 110 to the content module 112 over the Internet 106. Such transfers of data are well known in

the art and do not require further discussion herein. Alternatively, the content may comprise a captured broadcast.

[0058] In a further embodiment, the encoding module 406 is configured to generate a plurality of sets 306 of streamlets 304. The sets 306, as described above with reference to Figure 3b, may comprise streamlets having an identical time index and duration, and a unique bitrate. As with Figure 3b, the sets 306 and subsequently the plurality of streams 202 may comprise the low quality stream 204, the medium quality stream 206, and the high quality stream 208. Alternatively, the plurality of streams 202 may comprise any number of streams deemed necessary to accommodate end user bandwidth.

[0059] The encoder module 406 is further configured to encode each source streamlet 303 into the plurality of streams 202 and streamlet sets 306 and store the streamlets in the streamlet database 408. The encoding module 406 may utilize encoding schemes such as DivX®, Windows Media Video 9®, Quicktime 6.5 Sorenson 3®, or Quicktime 6.5/MPEG-4®. Alternatively, a custom encoding scheme may be employed.

[0060] The content module 112 may also include a metadata module 412 and a metadata database 414. In one embodiment, metadata comprises static searchable content information. For example, metadata includes, but is not limited to, air date of the content, title, actresses, actors, length, and episode name. Metadata is generated by the publisher 110, and may be configured to define an end user environment. In one embodiment, the publisher 100 may define an end user navigational environment for the content including menus, thumbnails, sidebars, advertising, etc. Additionally, the publisher 110 may define functions such as fast forward, rewind, pause, and play that may be used with the content file 200. The metadata module 412 is configured to receive the metadata from the publisher 110 and store the metadata in the metadata database 414. In a further embodiment, the metadata module 412 is configured to interface with the client module 114, allowing the client module 114 to search for content based upon at least one of a plurality of metadata criteria. Additionally, metadata may be generated by the content module 112 through automated process(es) or manual definition.

[0061] Once the streamlets 304 have been received and processed, the client module 114 may request streamlets 304 using HTTP from the web server 116. Using a standard protocol such as HTTP eliminates the need for network administrators to configure

firewalls to recognize and pass through network traffic for a new, specialized protocol. Additionally, since the client module 114 initiates the request, the web server 116 is only required to retrieve and serve the requested streamlet 304. In a further embodiment, the client module 114 may be configured to retrieve streamlets 304 from a plurality of web servers 116.

[0062] Each web server 116 may be located in various locations across the Internet 106. The streamlets 304 may essentially be static files. As such, no specialized media server or server-side intelligence is required for a client module 114 to retrieve streamlets 304. Streamlets 304 may be served by the web server 116 or cached by cache servers of Internet Service Providers (ISPs), or any other network infrastructure operators, and served by the cache server. Use of cache servers is well known to those skilled in the art, and will not be discussed further herein. Thus, a highly scalable solution is provided that is not hindered by massive amounts of client module 114 requests to the web server 116 at any specific location, especially the web server 116 most closely associated with or within the content module 112

[0063] Figure 5a is a schematic block diagram illustrating one embodiment of an encoder module 406 in accordance with the present invention. In one embodiment, the encoder module 406 may include a master module 502 and a plurality of host computing modules (hereinafter “host”) 504. The hosts 504 may comprise personal computers, servers, etc. In a further embodiment, the hosts 504 may be dedicated hardware, for example, cards plugged into a single computer.

[0064] The master module (hereinafter “master”) 502 is configured to receive streamlets 303 from the streamlet module 404 and stage the streamlet 303 for processing. In one embodiment, the master 502 may decompress each source streamlet 303 to produce a raw streamlet. As used herein, the term “raw streamlet” refers to a streamlet 303 that is uncompressed or lightly compressed to substantially reduce size with no significant loss in quality. A lightly compressed raw streamlet can be transmitted more quickly and to more hosts. Each host 504 is coupled with the master 502 and configured to receive a raw streamlet from the master 502 for encoding. The hosts 504, in one example, generate a plurality of streamlets 304 having identical time indices and durations, and varying bitrates. Essentially each host 504 may be configured to generate a set 306 from the raw

streamlet 503 sent from the master 502. Alternatively, each host 504 may be dedicated to producing a single bitrate in order to reduce the time required for encoding.

[0065] Upon encoding completion, the host 504 returns the set 306 to the master 502 so that the encoding module 406 may store the set 306 in the streamlet database 408. The master 502 is further configured to assign encoding jobs to the hosts 504. Each host is configured to submit an encoding job completion bid (hereinafter "bid"). The master 502 assigns encoding jobs depending on the bids from the hosts 504. Each host 504 generates a bid depending upon a plurality of computing variables which may include, but are not limited to, current encoding job completion percentage, average job completion time, processor speed and physical memory capacity.

[0066] For example, a host 504 may submit a bid that indicates that based on past performance history the host 504 would be able to complete the encoding job in 15 seconds. The master 502 is configured to select from among a plurality of bids the best bid and subsequently submit the encoding job to the host 504 with the best bid. As such, the described encoding system does not require that each host 504 have identical hardware but beneficially takes advantage of the available computing power of the hosts 504. Alternatively, the master 502 selects the host 504 based on a first come first serve basis, or some other algorithm deemed suitable for a particular encoding job.

[0067] The time required to encode one streamlet 304 is dependent upon the computing power of the host 504, and the encoding requirements of the content file 200. Examples of encoding requirements may include, but are not limited to, two or multi-pass encoding, and multiple streams of different bitrates. One benefit of the present invention is the ability to perform two-pass encoding on a live content file 200. Typically, in order to perform two-pass encoding prior art systems must wait for the content file to be completed before encoding.

[0068] The present invention, however, segments the content file 200 into source streamlets 303 and the two-pass encoding to a plurality of streams 202 may be performed on each corresponding raw streamlet without waiting for a TV show to end, for example. As such, the content module 112 is capable of streaming the streamlets over the Internet shortly after the content module 112 begins capture of the content file 200. The delay

between a live broadcast transmitted from the publisher 110 and the availability of the content depends on the computing power of the hosts 504.

[0069] Figure 5b is a schematic block diagram illustrating one embodiment of parallel encoding of streamlets in accordance with the present invention. In one example, the capture module 402 (of Figure 4) begins to capture the content file and the streamlet module 404 generates a first streamlet 303a and passes the streamlet to the encoding module 406. The encoding module 406 may take 10 seconds, for example, to generate the first set 306a of streamlets 304a (304a1, 304a2, 304a3, etc. represent streamlets 304 of different bitrates). Figure 5b illustrates the encoding process generically as block 502 to graphically illustrate the time duration required to process a raw or lightly encoded streamlet 303 as described above with reference to the encoding module 406. The encoding module 406 may simultaneously process more than one streamlet 303, and processing of streamlets will begin upon arrival of the streamlet from the capture module 402.

[0070] During the 10 seconds required to encode the first streamlet 303a, the streamlet module 404 has generated five additional 2-second streamlets 303b, 303c, 303d, 303e, 303f, for encoding and the master 502 has prepared and staged the corresponding raw streamlets. Two seconds after the first set 306a is available the next set 306b is available, and so on. As such, the content file 200 is encoded for streaming over the Internet and appears live. The 10 second delay is given herein by way of example only. Multiple hosts 504 may be added to the encoding module 406 in order to increase the processing capacity of the encoding module 406. The delay may be shortened to an almost unperceivable level by the addition of high CPU powered systems, or alternatively multiple low powered systems.

[0071] A system as described above beneficially enables multi-pass encoding of live events. Multi-pass encoding systems of the prior art require that the entire content be captured (or be complete) because in order to perform multi-pass encoding the entire content must be scanned and processed more than once. This is impossible with prior art systems because content from a live event is not complete until the event is over. As such, with prior art systems, multi-pass encoding can only be performed once the event is over. Streamlets, however, may be encoded as many times as is deemed necessary.

Because the streamlet is an encapsulated media object of 2 seconds (for example), multi-pass encoding may begin on a live event once the first streamlet is captured. Shortly after multi-pass encoding of the first streamlet 303a is finished, multi-pass encoding of the second streamlet 303b finishes, and as such multi-pass encoding is performed on a live event and appears live to a viewer.

[0072] Any specific encoding scheme applied to a streamlet may take longer to complete than the time duration of the streamlet itself, for example, a very high quality encoding of a 2-second streamlet may take 5 seconds to finish. Alternatively, the processing time required for each streamlet may be less than the time duration of a streamlet. However, because the offset parallel encoding of successive streamlets are encoded by the encoding module at regular intervals (matching the intervals at which the those streamlets are submitted to the encoding module 406, for example 2 seconds) the output timing of the encoding module 406 does not fall behind the real-time submission rate of the unencoded streamlets. Conversely, prior art encoding systems rely on the very fastest computing hardware and software because the systems must generate the output immediately in lock-step with the input. A prior art system that takes 2.1 seconds to encode 2 seconds worth of content is considered a failure. The present invention allows for slower than real-time encoding processes yet still achieves a real-time encoding effect due to the parallel offset pipes.

[0073] The parallel offset pipeline approach described with reference to Figure 5b beneficially allows for long or short encoding times without “falling behind” the live event. Additionally, arbitrarily complex encoding of streamlets to multiple profiles and optimizations only lengthens the encoding time 502 without a perceptible difference to a user because the sets 306 of streamlets 304 are encoded in a time-selective manner so that streamlets are processed at regular time intervals and transmitted at these time intervals.

[0074] Returning now to Figure 5a, as depicted, the master 502 and the hosts 504 may be located within a single local area network, or in other terms, the hosts 504 may be in close physical proximity to the master 502. Alternatively, the hosts 504 may receive encoding jobs from the master 502 over the Internet or other communications network. For example, consider a live sports event in a remote location where it would be difficult to setup multiple hosts. In this example, a master performs no encoding or alternatively

light encoding before publishing the streamlets online. The hosts 504 would then retrieve those streamlets and encode the streamlets into the multiple bitrate sets 306 as described above.

[0075] Furthermore, hosts 504 may be dynamically added or removed from the encoding module without restarting the encoding job and/or interrupting the publishing of streamlets. If a host 504 experiences a crash or some failure, its encoding work is simply reassigned to another host.

[0076] The encoding module 406, in one embodiment, may also be configured to produce streamlets that are specific to a particular playback platform. For example, for a single raw streamlet, a single host 504 may produce streamlets for different quality levels for personal computer playback, streamlets for playback on cell phones with a different, proprietary codec, a small video-only streamlet for use when playing just a thumbnail view of the stream (like in a programming guide), and a very high quality streamlet for use in archiving.

[0077] Figure 6a is a schematic block diagram illustrating one embodiment of a virtual timeline 600 in accordance with the present invention. In one embodiment, the virtual timeline 600 comprises at least one quantum media extension 602. The quantum media extension (hereinafter “QMX”) 602 describes an entire content file 200. Therefore, the virtual timeline (hereinafter “VT”) 600 may comprise a file that is configured to define a playlist for a user to view. For example, the VT may indicate that the publisher desires a user to watch a first show QMX 602a followed by QMX 602b and QMX 602c. As such, the publisher may define a broadcast schedule in a manner similar to a television station.

[0078] Figure 6b is a schematic block diagram illustrating an alternative embodiment of a VT 600 in accordance with the present invention. In the depicted embodiment, the VT 600 may include a single QMX 602 which indicates that the publisher desires the same content to be looped over and over again. For example, the publisher may wish to broadcast a never-ending infomercial on a website.

[0079] Figure 6c is a schematic block diagram illustrating one embodiment of a QMX 602 in accordance with the present invention. In one embodiment, the QMX 602 contains a multitude of information generated by the content module 112 configured to describe the content file 200. Examples of information include, but are not limited to,

start index 604, end index 606, whether the content is live 608, proprietary publisher data 610, encryption level 612, content duration 614 and bitrate values 616. The bitrate values 616 may include frame size 618, audio channel 620 information, codecs 622 used, sample rate 624, and frames parser 626.

[0080] A publisher may utilize the QVT 600 together with the QMX 602 in order to prescribe a playback order for users, or alternatively selectively edit content. For example, a publisher may indicate in the QMX 602 that audio should be muted at time index 10:42 or video should be skipped for 3 seconds at time index 18:35. As such, the publisher may selectively skip offensive content without the processing requirements of editing the content.

[0081] Figure 7 is a schematic block diagram graphically illustrating one embodiment of a client module 114 in accordance with the present invention. The client module 114 may comprise an agent controller module 702, a streamlet cache module 704, and a network controller module 706. In one embodiment, the agent controller module 702 is configured to interface with a viewer 708, and transmit streamlets 304 to the viewer 708. Alternatively, the agent controller module 702 may be configured to simply reassemble streamlets into a single file for transfer to an external device such as a portable video player.

[0082] In a further embodiment, the client module 114 may comprise a plurality of agent controller modules 702. Each agent controller module 702 may be configured to interface with one viewer 708. Alternatively, the agent controller module 702 may be configured to interface with a plurality of viewers 708. The viewer 708 may be a media player (not shown) operating on a PC or handheld electronic device.

[0083] The agent controller module 702 is configured to select a quality level of streamlets to transmit to the viewer 708. The agent controller module 702 requests lower or higher quality streams based upon continuous observation of time intervals between successive receive times of each requested streamlet. The method of requesting higher or lower quality streams will be discussed in greater detail below with reference to Figure 10.

[0084] The agent controller module 702 may be configured to receive user commands from the viewer 708. Such commands may include play, fast forward, rewind, pause, and

stop. In one embodiment, the agent controller module 702 requests streamlets 304 from the streamlet cache module 704 and arranges the received streamlets 304 in a staging module 709. The staging module 709 may be configured to arrange the streamlets 304 in order of ascending playback time. In the depicted embodiment, the streamlets 304 are numbered 0, 1, 2, 3, 4, etc. However, each streamlet 304 may be identified with a unique filename.

[0085] Additionally, the agent controller module 702 may be configured to anticipate streamlet 304 requests and pre-request streamlets 304. By pre-requesting streamlets 304, the user may fast-forward, skip randomly, or rewind through the content and experience no buffering delay. In a further embodiment, the agent controller module 702 may request the streamlets 304 that correspond to time index intervals of 30 seconds within the total play time of the content. Alternatively, the agent controller module 702 may request streamlets at any interval less than the length of the time index. This enables a “fast-start” capability with no buffering wait when starting or fast-forwarding through content file 200. In a further embodiment, the agent controller module 702 may be configured to pre-request streamlets 304 corresponding to specified index points within the content or within other content in anticipation of the end user 104 selecting new content to view. In one embodiment, the streamlet cache module 704 is configured to receive streamlet 304 requests from the agent controller module 702. Upon receiving a request, the streamlet cache module 704 first checks a streamlet cache 710 to verify if the streamlet 304 is present. In a further embodiment, the streamlet cache module 704 handles streamlet 304 requests from a plurality of agent controller modules 702. Alternatively, a streamlet cache module 704 may be provided for each agent controller module 702. If the requested streamlet 304 is not present in the streamlet cache 410, the request is passed to the network controller module 706. In order to enable fast forward and rewind capabilities, the streamlet cache module 704 is configured to store the plurality of streamlets 304 in the streamlet cache 710 for a specified time period after the streamlet 304 has been viewed. However, once the streamlets 304 have been deleted, they may be requested again from the web server 116.

[0086] The network controller module 706 may be configured to receive streamlet requests from the streamlet cache module 704 and open a connection to the web server

116 or other remote streamlet 304 database (not shown). In one embodiment, the network controller module 706 opens a TCP/IP connection to the web server 116 and generates a standard HTTP GET request for the requested streamlet 304. Upon receiving the requested streamlet 304, the network controller module 706 passes the streamlet 304 to the streamlet cache module 704 where it is stored in the streamlet cache 710. In a further embodiment, the network controller module 706 is configured to process and request a plurality of streamlets 304 simultaneously. The network controller module 706 may also be configured to request a plurality of streamlets, where each streamlet 304 is subsequently requested in multiple parts.

[0087] In a further embodiment, streamlet requests may comprise requesting pieces of any streamlet file. Splitting the streamlet 304 into smaller pieces or portions beneficially allows for an increased efficiency potential, and also eliminates problems associated with multiple full-streamlet requests sharing the bandwidth at any given moment. This is achieved by using parallel TCP/IP connections for pieces of the streamlets 304. Consequently, efficiency and network loss problems are overcome, and the streamlets arrive with more useful and predictable timing.

[0088] In one embodiment, the client module 114 is configured to use multiple TCP connections between the client module 114 and the web server 116 or web cache. The intervention of a cache may be transparent to the client or configured by the client as a forward cache. By requesting more than one streamlet 304 at a time in a manner referred to as "parallel retrieval," or more than one part of a streamlet 304 at a time, efficiency is raised significantly and latency is virtually eliminated. In a further embodiment, the client module allows a maximum of three outstanding streamlet 304 requests. The client module 114 may maintain additional open TCP connections as spares to be available should another connection fail. Streamlet 304 requests are rotated among all open connections to keep the TCP flow logic for any particular connection from falling into a slow-start or close mode. If the network controller module 706 has requested a streamlet 304 in multiple parts, with each part requested on mutually independent TCP/IP connections, the network controller module 706 reassembles the parts to present a complete streamlet 304 for use by all other components of the client module 114.

[0089] When a TCP connection fails completely, a new request may be sent on a different connection for the same streamlet 304. In a further embodiment, if a request is not being satisfied in a timely manner, a redundant request may be sent on a different connection for the same streamlet 304. If the first streamlet request's response arrives before the redundant request response, the redundant request can be aborted. If the redundant request response arrives before the first request response, the first request may be aborted.

[0090] Several streamlet 304 requests may be sent on a single TCP connection, and the responses are caused to flow back in matching order along the same connection. This eliminates all but the first request latency. Because multiple responses are always being transmitted, the processing latency of each new streamlet 304 response after the first is not a factor in performance. This technique is known in the industry as "pipelining." Pipelining offers efficiency in request-response processing by eliminating most of the effects of request latency. However, pipelining has serious vulnerabilities. Transmission delays affect all of the responses. If the single TCP connection fails, all of the outstanding requests and responses are lost. Pipelining causes a serial dependency between the requests.

[0091] Multiple TCP connections may be opened between the client module 114 and the web server 116 to achieve the latency-reduction efficiency benefits of pipelining while maintaining the independence of each streamlet 304 request. Several streamlet 304 requests may be sent concurrently, with each request being sent on a mutually distinct TCP connection. This technique is labeled "virtual pipelining" and is an innovation of the present invention. Multiple responses may be in transit concurrently, assuring that communication bandwidth between the client module 114 and the web server 116 is always being utilized. Virtual pipelining eliminates the vulnerabilities of traditional pipelining. A delay in or complete failure of one response does not affect the transmission of other responses because each response occupies an independent TCP connection. Any transmission bandwidth not in use by one of multiple responses (whether due to delays or TCP connection failure) may be utilized by other outstanding responses.

[0092] A single streamlet 304 request may be issued for an entire streamlet 304, or multiple requests may be issued, each for a different part or portion of the streamlet. If the streamlet is requested in several parts, the parts may be recombined by the client module 114 streamlet.

[0093] In order to maintain a proper balance between maximized bandwidth utilization and response time, the issuance of new streamlet requests must be timed such that the web server 116 does not transmit the response before the client module 114 has fully received a response to one of the previously outstanding streamlet requests. For example, if three streamlet 304 requests are outstanding, the client module 114 should issue the next request slightly before one of the three responses is fully received and “out of the pipe.” In other words, request timing is adjusted to keep three responses in transit. Sharing of bandwidth among four responses diminishes the net response time of the other three responses. The timing adjustment may be calculated dynamically by observation, and the request timing adjusted accordingly to maintain the proper balance of efficiency and response times.

[0094] The schematic flow chart diagrams that follow are generally set forth as logical flow chart diagrams. As such, the depicted order and labeled steps are indicative of one embodiment of the presented method. Other steps and methods may be conceived that are equivalent in function, logic, or effect to one or more steps, or portions thereof, of the illustrated method. Additionally, the format and symbols employed are provided to explain the logical steps of the method and are understood not to limit the scope of the method. Although various arrow types and line types may be employed in the flow chart diagrams, they are understood not to limit the scope of the corresponding method. Indeed, some arrows or other connectors may be used to indicate only the logical flow of the method. For instance, an arrow may indicate a waiting or monitoring period of unspecified duration between enumerated steps of the depicted method. Additionally, the order in which a particular method occurs may or may not strictly adhere to the order of the corresponding steps shown.

[0095] Figure 8 is a schematic flow chart diagram illustrating one embodiment of a method 800 for processing content in accordance with the present invention. In one embodiment the method 800 starts 802, and the content module 112 receives 804 content

from the publisher 110. Receiving content 804 may comprise receiving 804 a digital copy of the content file 200, or digitizing a physical copy of the content file 200. Alternatively, receiving 804 content may comprise capturing a radio, television, cable, or satellite broadcast. Once received 804, the streamlet module 404 generates 808 a plurality of source streamlets 303 each having a fixed duration. Alternatively, the streamlets 303 may be generated with a fixed file size.

[0096] In one embodiment, generating 808 streamlets comprises dividing the content file 200 into a plurality of two second streamlets 303. Alternatively, the streamlets may have any length less than or equal to the length of the stream 202. The encoder module 406 then encodes 810 the streamlets 303 into sets 306 of streamlets 304, in a plurality of streams 202 according to an encoding scheme. The quality may be predefined, or automatically set according to end user bandwidth, or in response to pre-designated publisher guidelines

[0097] In a further embodiment, the encoding scheme comprises a proprietary codec such as WMV9®. The encoder module 406 then stores 812 the encoded streamlets 304 in the streamlet database 408. Once stored 812, the web server 116 may then serve 814 the streamlets 304. In one embodiment, serving 814 the streamlets 304 comprises receiving streamlet requests from the client module 114, retrieving the requested streamlet 304 from the streamlet database 408, and subsequently transmitting the streamlet 304 to the client module 114. The method 800 then ends 816.

[0098] Figure 9 is a schematic flow chart diagram illustrating one embodiment of a method 900 for viewing a plurality of streamlets in accordance with the present invention. The method 900 starts and an agent controller module 702 is provided 904 and associated with a viewer 708 and provided with a staging module 709. The agent controller module 702 then requests 906 a streamlet 304 from the streamlet cache module 704. Alternatively, the agent controller module 702 may simultaneously request 906 a plurality of streamlets 304 the streamlet cache module 704. If the streamlet is stored 908 locally in the streamlet cache 710, the streamlet cache module 704 retrieves 910 the streamlet 304 and sends the streamlet to the agent controller module 702. Upon retrieving 910 or receiving a streamlet, the agent controller module 702 makes 911 a

determination of whether or not to shift to a higher or lower quality stream 202. This determination will be described below in greater detail with reference to Figure 10.

[0099] In one embodiment, the staging module 709 then arranges 912 the streamlets 304 into the proper order, and the agent controller module 702 delivers 914 the streamlets to the viewer 708. In a further embodiment, delivering 914 streamlets 304 to the end user comprises playing video and or audio streamlets on the viewer 708. If the streamlets 304 are not stored 908 locally, the streamlet request is passed to the network controller module 706. The network controller module 706 then requests 916 the streamlet 304 from the web server 116. Once the streamlet 304 is received, the network controller module 706 passes the streamlet to the streamlet cache module 704. The streamlet cache module 704 archives 918 the streamlet. Alternatively, the streamlet cache module 704 then archives 918 the streamlet and passes the streamlet to the agent controller module 702, and the method 900 then continues from operation 910 as described above.

[0100] Referring now to Figure 10, shown therein is a schematic flow chart diagram illustrating one embodiment of a method 1000 for requesting streamlets 304 within an adaptive-rate shifting content streaming environment in accordance with the present invention. The method 1000 may be used in one embodiment as the operation 911 of Figure 9. The method 1000 starts and the agent controller module 702 receives 1004 a streamlet 304 as described above with reference to Figure 9. The agent controller module 702 then monitors 1006 the receive time of the requested streamlet. In one embodiment, the agent controller module 702 monitors the time intervals Δ between successive receive times for each streamlet response. Ordering of the responses in relation to the order of their corresponding requests is not relevant.

[0101] Because network behavioral characteristics fluctuate, sometimes quite suddenly, any given Δ may vary substantially from another. In order to compensate for this fluctuation, the agent controller module 702 calculates 1008 a performance ratio r across a window of n samples for streamlets of playback length S . In one embodiment, the performance ratio r is calculated using the equation:

$$r = S \frac{n}{\sum_{i=1}^n \Delta_i}$$

[0102] Due to multiple simultaneous streamlet processing, and in order to better judge the central tendency of the performance ratio r , the agent controller module 702 may calculate a geometric mean, or alternatively an equivalent averaging algorithm, across a window of size m , and obtain a performance factor φ :

$$\varphi_{current} = \left(\prod_{j=1}^m r_j \right)^{\frac{1}{m}}$$

[0103] The policy determination about whether or not to upshift 1010 playback quality begins by comparing $\varphi_{current}$ with a trigger threshold Θ_{up} . If $\varphi_{current} \geq \Theta_{up}$, then an up shift to the next higher quality stream may be considered 1016. In one embodiment, the trigger threshold Θ_{up} is determined by a combination of factors relating to the current read ahead margin (i.e. the amount of contiguously available streamlets that have been sequentially arranged by the staging module 709 for presentation at the current playback time index), and a minimum safety margin. In one embodiment, the minimum safety margin may be 24 seconds. The smaller the read ahead margin, the larger Θ_{up} is to discourage upshifting until a larger read ahead margin may be established to withstand network disruptions. If the agent controller module 702 is able to sustain 1016 upshift quality, then the agent controller module 702 will upshift 1017 the quality and subsequently request higher quality streams. The determination of whether use of the higher quality stream is sustainable 1016 is made by comparing an estimate of the higher quality stream's performance factor, φ_{higher} , with Θ_{up} . If $\varphi_{higher} \geq \Theta_{up}$ then use of the higher quality stream is considered sustainable. If the decision of whether or not the higher stream rate is sustainable 1016 is "no," the agent controller module 702 will not attempt to upshift 1017 stream quality. If the end of the stream has been reached 1014, the method 1000 ends 1016.

[0104] If the decision on whether or not to attempt upshift 1010 is "no", a decision about whether or not to downshift 1012 is made. In one embodiment, a trigger threshold Θ_{down} is defined in a manner analogous to Θ_{up} . If $\varphi_{current} > \Theta_{down}$ then the stream quality may be adequate, and the agent controller module 702 does not downshift 1018 stream quality.

However, if $\varphi_{current} \leq \Theta_{down}$, the agent controller module 702 does downshift 1018 the stream quality. If the end of the stream has not been reached 1014, the agent controller module 702 begins to request and receive 1004 lower quality streamlets and the method 1000 starts again. Of course, the above described equations and algorithms are illustrative only, and may be replaced by alternative streamlet monitoring solutions.

[0105] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

Electronic Acknowledgement Receipt

EFS ID:	34904852
Application Number:	16252188
International Application Number:	
Confirmation Number:	3722
Title of Invention:	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING
First Named Inventor/Applicant Name:	DAVID F. BRUECK
Customer Number:	70560
Filer:	Brett Carlson/Julie Graf
Filer Authorized By:	Brett Carlson
Attorney Docket Number:	066.0141X1C7
Receipt Date:	18-JAN-2019
Filing Date:	
Time Stamp:	17:13:51
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 7320
RAM confirmation Number	012219INTEFSW17142500
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	20190118_ADS_066-0141X1C7_P2007-02-01-2-5.pdf	1823853	no	10
			567104f3ddd97754c2bea8dea67f8e46133b12e4		
Warnings:					
Information:					
2		20190118_ConAppl_066-0141X1C7_P2007-02-01-2-5.pdf	14768938	yes	37
			6b0e9213971087a0bb9b39cf2745807ad000cfd1		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Specification		1	28	
	Claims		29	36	
	Abstract		37	37	
Warnings:					
Information:					
3	Oath or Declaration filed	20190118_Declaration_066-0141X1C7_P2007-02-01-2-5.pdf	135215	no	3
			98d341969b7e66dbccf545b2f138fadf294d0db5		
Warnings:					
Information:					
4	Drawings-only black and white line drawings	20190118_Dwgs_066-0141X1C7_P2007-02-01-2-5.pdf	130400	no	11
			f83697378d2ba053dda2ad53a400c460df5ec2aa		
Warnings:					
Information:					
5	Power of Attorney	20190118_POA_066-0141X1C7_P2007-02-01-2-5.pdf	109363	no	2
			c02066475cb6e321298bf8d025d3da883ca2b403		

Warnings:					
The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing					
Information:					
6	Preliminary Amendment	20190118_PreliminaryAmendment_066-0141X1C7_P2007-02-01-2-5.pdf	152808 9ed22a56acc2a591ea5259fcf0b47b57d60c45cc	no	4
Warnings:					
Information:					
7	TrackOne Request	20190118_Track1-PrioritizedExam_066-0141X1C7_P2007-02-01-2-5.pdf	140714 19d909fc28fb7a53619d44db6af7005a8cf3323c	no	2
Warnings:					
Information:					
8	Fee Worksheet (SB06)	fee-info.pdf	42908 1c60c368602aff7d1afc80c3fc988f1657774abd	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			17304199		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
Title of Invention:		APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		
First Named Inventor/Applicant Name:		DAVID F. BRUECK		
Filer:		Brett Carlson/Julie Graf		
Attorney Docket Number:		066.0141X1C7		
Filed as Large Entity				
Filing Fees for Track I Prioritized Examination - Nonprovisional Application under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
UTILITY APPLICATION FILING	1011	1	300	300
UTILITY SEARCH FEE	1111	1	660	660
UTILITY EXAMINATION FEE	1311	1	760	760
REQUEST FOR PRIORITIZED EXAMINATION	1817	1	4000	4000
Pages:				
Claims:				
CLAIMS IN EXCESS OF 20	1202	10	100	1000
INDEPENDENT CLAIMS IN EXCESS OF 3	1201	1	460	460

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous-Filing:				
PUBL. FEE- EARLY, VOLUNTARY, OR NORMAL	1504	1	0	0
PROCESSING FEE, EXCEPT PROV. APPLS.	1830	1	140	140
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				7320

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.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	066.0141X1C7
		Application Number	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		
The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76. This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.			

Secrecy Order 37 CFR 5.2:

☐	Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)
--------------------------	---

Inventor Information:

Inventor	1	<div>Remove</div>			
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	DAVID	F.	BRUECK		
Residence Information (Select One) ☒ US Residency ☐ Non US Residency ☐ Active US Military Service					
City	SARATOGA SPRINGS	State/Province	UT	Country of Residence	US
Mailing Address of Inventor:					
Address 1	962 NORTH RATTLER ROAD				
Address 2					
City	SARATOGA SPRINGS	State/Province	UT		
Postal Code	84043	Country i	US		
Inventor	2	<div>Remove</div>			
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	MARK	B.	HURST		
Residence Information (Select One) ☒ US Residency ☐ Non US Residency ☐ Active US Military Service					
City	CEDAR HILLS	State/Province	UT	Country of Residence	US
Mailing Address of Inventor:					
Address 1	10639 NORTH AVALON				
Address 2					
City	CEDAR HILLS	State/Province	UT		
Postal Code	84062	Country i	US		
Inventor	3	<div>Remove</div>			
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	R.	DREW	MAJOR		
Residence Information (Select One) ☒ US Residency ☐ Non US Residency ☐ Active US Military Service					

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	066.0141X1C7
		Application Number	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

City	OREM	State/Province	UT	Country of Residence	US
------	------	----------------	----	----------------------	----

Mailing Address of Inventor:

Address 1	863 WEST 600 SOUTH				
Address 2					
City	OREM	State/Province	UT		
Postal Code	84058	Country	US		

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button.

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ An Address is being provided for the correspondence information of this application.

Customer Number	70560		
Email Address	docketing@lkglobal.com	<input type="button" value="Add Email"/>	<input type="button" value="Remove Email"/>

Application Information:

Title of the Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		
Attorney Docket Number	066.0141X1C7	Small Entity Status Claimed	☐
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	11	Suggested Figure for Publication (if any)	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	066.0141X1C7
		Application Number	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	☒ Customer Number	☐ US Patent Practitioner	☐ Limited Recognition (37 CFR 11.9)
Customer Number	70560		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.

When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending		Remove		
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
	Continuation of	16004056	2018-06-08		
Prior Application Status	Patented		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
16004056	Continuation of	15414025	2017-01-24	9998516	2018-06-12
Prior Application Status	Patented		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15414025	Continuation of	14719122	2015-05-21	9571551	2017-02-14
Prior Application Status	Patented		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
14719122	Continuation of	14106051	2013-12-13	9071668	2015-06-30
Prior Application Status	Patented		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
14106051	Continuation of	13617114	2012-09-14	8612624	2013-12-17

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.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number		066.0141X1C7	
		Application Number			
Title of Invention		APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING			

Prior Application Status		Patented		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13617114	Continuation of	12906940	2010-10-18	8402156	2013-03-19
Prior Application Status		Patented		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
12906940	Continuation of	11673483	2007-02-09	7818444	2010-10-19
Prior Application Status		Patented		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
11673483	Continuation in part of	11116783	2005-04-28	8868772	2014-10-21
Prior Application Status		Expired		Remove	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
11116783	Claims benefit of provisional	60566831	2004-04-30		
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the Add button.					Add

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)ⁱ the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

			Remove
Application Number	Country ⁱ	Filing Date (YYYY-MM-DD)	Access Code ^j (if applicable)
Additional Foreign Priority Data may be generated within this form by selecting the Add button.			Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

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.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	066.0141X1C7
		Application Number	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013. NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.	
---	--

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	066.0141X1C7
		Application Number	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	066.0141X1C7
		Application Number	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.			
Applicant	1		<input type="button" value="Remove"/>
<p>If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.</p>			
<input type="button" value="Clear"/>			
Assignee	Legal Representative under 35 U.S.C. 117		Joint Inventor
☐ Person to whom the inventor is obligated to assign.		☐ Person who shows sufficient proprietary interest	
If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:			
<div style="border: 1px solid black; height: 20px; width: 100%;"></div>			
Name of the Deceased or Legally Incapacitated Inventor: <div style="border: 1px solid black; width: 400px; height: 20px;"></div>			
If the Applicant is an Organization check here. ☒			
Organization Name	DISH Technologies L.L.C.		
Mailing Address Information For Applicant:			
Address 1	9601 S. Meridian Blvd.		
Address 2			
City	Englewood	State/Province	CO
Country	US	Postal Code	80112
Phone Number		Fax Number	
Email Address			
Additional Applicant Data may be generated within this form by selecting the Add button. <input type="button" value="Add"/>			

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

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.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	066.0141X1C7
		Application Number	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

Assignee	1		
Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.			
			<input type="button" value="Remove"/>
If the Assignee or Non-Applicant Assignee is an Organization check here.			☒
Organization Name	DISH Technologies L.L.C.		
Mailing Address Information For Assignee including Non-Applicant Assignee:			
Address 1	9601 S. Meridian Blvd.		
Address 2			
City	Englewood	State/Province	CO
Country ⁱ	US	Postal Code	80112
Phone Number		Fax Number	
Email Address			
Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.			<input type="button" value="Add"/>

Signature:

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). However, if this Application Data Sheet is submitted with the **INITIAL** filing of the application and either box A or B is **not** checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/BRETT A. CARLSON/		Date (YYYY-MM-DD)	2019-01-18	
First Name	Brett A.	Last Name	Carlson	Registration Number	39,928
Additional Signature may be generated within this form by selecting the Add button.				<input type="button" value="Add"/>	

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.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	066.0141X1C7
		Application Number	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

This collection of information is required by 37 CFR 1.76. 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 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet 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 records.
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.

**DECLARATION OF JOINT INVENTORS RELATING TO PATENTS AND
APPLICATIONS ENTITLED “APPARATUS, SYSTEM, AND METHOD FOR
MULTI-BITRATE CONTENT STREAMING”**

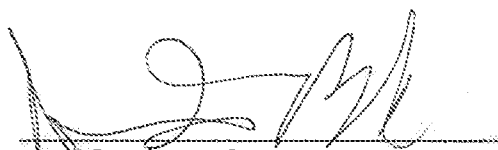
This is a joint declaration and statement made by David F. Brueck, Mark B. Hurst, and R. Drew Major. We the undersigned each hereby declare and state that, to the best of our information and belief:

1. Each of us (David F. Brueck, Mark B. Hurst, and R. Drew Major) contributed to the inventorship of the following United States Patents and the following pending application:

<u>US Patents:</u> US7818444 US8402156 US8612624 US9071668 US9571551	<u>US Patent Application</u> Serial No. 15/414,025
---	---
2. The above-identified patents and applications were made or authorized to be made by me. I believe that I am the inventor or an original joint inventor of a claimed invention in each of the applications and patents referenced above.
3. Although R. Drew Major was initially omitted as an inventor, I understand that he is now being added as a joint inventor. I agree to the change of inventorship and I have no disagreement with regard to the change in inventorship. To the best of my information and belief, Mr. Major's earlier omission was made without deceptive intent.
4. I hereby declare that all statements made herein of my own knowledge are true and that all statements on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements in this declaration are punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both, and

that willful false statements may jeopardize the validity of the application
or any patent issued thereon.

Respectfully submitted by:



David F. Brueck

27 Oct 2017

Date

Mark B. Hurst

Date

R. Drew Major

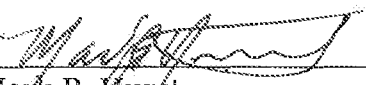
Date

1001 by fine or imprisonment of not more than five (5) years, or both, and that willful false statements may jeopardize the validity of the application or any patent issued thereon.

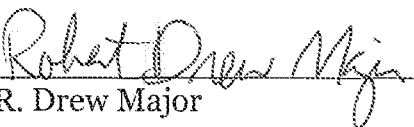
Respectfully submitted by:

David F. Brueck

Date


Mark B. Hurst

11/6/17
Date


R. Drew Major

11/6/17
Date

CLAIMS

What is claimed is:

1. A system for adaptive-rate content streaming of live event video playable on one or more end user stations over the internet, the system comprising:

at least one storage device storing live event video, the live event video encoded at a plurality of bitrates creating a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream, the low quality stream, the medium quality stream, and the high quality stream each comprising a group of streamlets encoded at a respective one of the plurality of bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video;

wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and

wherein the first streamlet of each of the groups of streamlets has the same first duration and corresponds to the same first portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the first streamlet of the low quality stream having a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream.

2. The system of claim 1, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.

3. The system of claim 2, wherein the first and second durations are different.

4. The system of claim 1, further comprising:

a plurality of web servers located at different locations across the internet, each web server configured to:

receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing a portion of the video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams;

retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and

send the retrieved first streamlet from the currently selected one of the different copies to the requesting one of the end user stations over the one or more network connections.

5. The system of claim 1, wherein each of the first streamlets has a first duration that is the range of 0.1 to 5 seconds.

6. The system of claim 1, wherein the live event is a live sports event.

7. The system of claim 1, further comprising:

a first web server configured to:

receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the live event video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the live event video;

retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and

send the retrieved first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream to the requesting one of the end user stations over the one or more network connections.

8. The system of claim 7, wherein the first streamlets of the low quality stream, the medium quality stream, and the high quality stream are available before the live event is complete.

9. The system of claim 7, wherein the streamlets of the low quality stream, the medium quality stream, and the high quality stream of the live event are available on a 10 second delay.

10. The system of claim 7, wherein the streamlets from the low quality stream, the medium quality stream, and the high quality stream of the live event, when played back, appear live to a viewer.

11. The system of claim 7, wherein the first web server is further configured to:
receive at least one virtual timeline request over the one or more internet connections from the one or more end user stations to retrieve a virtual timeline; and
send the virtual timeline to the requesting one of the end user stations over the one or more network connections.

12. The system of claim 11, wherein the virtual timeline corresponds to the currently selected one of the low quality stream, the medium quality stream, and the high quality stream.

13. The system of claim 11, wherein the virtual timeline defines a playlist for a user to view.

14. The system of claim 11, wherein the virtual timeline comprises a file that is configured to define a playlist for a user to view.

15. The system of claim 11, wherein the virtual timeline comprises at least one quantum media extension (QMX).

16. An end user station to stream a live event video over a network from a server for playback of the video, the content player device comprising:

a processor;

a digital processing apparatus memory device comprising non-transitory machine-readable instructions that, when executed, cause the processor to:

establish one or more network connections between the end user station and the server, wherein the server is configured to access at least one of a plurality of groups of streamlets;

wherein the live event video is encoded at a plurality of bitrates to create a plurality of streams including at least a low quality stream, a medium quality stream, and a high quality stream, each of the low quality stream, the medium quality stream, and the high quality stream comprising a group of streamlets encoded at the same respective bitrate, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video;

wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bit rate of no less than 600 kbps; and

wherein the first streamlets of each of the low quality stream, the medium quality stream and the high quality stream each has an equal playback duration and each of the first streamlets corresponding to the same portion of the live event video;

select a specific one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams;

place a streamlet request to the server over the one or more network connections for the first streamlet of the selected stream;

receive the requested first streamlet from the server via the one or more network connections; and
provide the received first streamlet for playback of the live event video.

17. The end user station of claim 16, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.

18. The end user station of claim 17, wherein the first and second durations are different.

19. The end user station of claim 16, wherein each of the first streamlets has a first duration that is the range of 0.1 to 5 seconds.

20. The end user station of claim 16, wherein the first streamlets of the low quality stream, the medium quality stream, and the high quality stream are available before the live event is complete.

21. The end user station of claim 16, wherein the streamlets of the low quality stream, the medium quality stream, and the high quality stream of the live event are available on a ten second delay.

22. The end user station of claim 16, wherein the streamlets from the low quality stream, the medium quality stream, and the high quality stream of the live event, when played back, appear live to a viewer.

23. The end user station of claim 16, wherein the end user station is further configured to:

request and receive a virtual timeline; and

wherein one or more streamlet requests are based on the at least one virtual timeline.

24. The end user station of claim 23, wherein the virtual timeline corresponds to the currently selected one of the low quality stream, the medium quality stream, and the high quality stream.

25. The end user station of claim 23, wherein the virtual timeline defines a playlist for a user to view.

26. A process executable by one or more servers to stream a live event video for playback by one or more end user stations, the process comprising:

storing, by the one or more servers, a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream, wherein the low quality stream, the medium quality stream, and the high quality stream each comprise a group of streamlets encoded at a respective one of the plurality of bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video;

wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and

wherein the first streamlet of each of the groups of streamlets has the same first duration and corresponds to the same first portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the first streamlet of the low quality stream having a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream;

receiving at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the live event video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the

medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the live event video;

retrieving from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and

sending the retrieved first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream to the requesting one of the end user stations over the one or more network connections.

27. The process of claim 26, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.

28. The process of claim 26, wherein the first and second durations are different.

29. The process of claim 26, wherein the first streamlets of the low quality stream, the medium quality stream, and the high quality stream are available before the live event is complete.

30 A process executable by a content player device to stream a live event video over a network from a server for playback of the video by the content player device, the process comprising:

establishing one or more network connections between the content player device and the server, wherein the server accesses a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream, wherein the low quality stream, the medium quality stream, and the high quality stream each comprise a group of streamlets encoded at a respective one of the plurality of bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video;

wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and

wherein the first streamlet of each of the groups of streamlets has the same first duration and corresponds to the same first portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the first streamlet of the low quality stream having a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream;

selecting, by the content player device, a currently selected one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the live event video;

placing a streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the live event video;

receiving the requested streamlet from the server via the one or more network connections; and

rendering, by the content player device, the received streamlet for playback of the live event video.

ABSTRACT OF THE DISCLOSURE

An apparatus for multi-bitrate content streaming includes a receiving module configured to capture media content, a streamlet module configured to segment the media content and generate a plurality of streamlets, and an encoding module configured to generate a set of streamlets. The system includes the apparatus, wherein the set of streamlets comprises a plurality of streamlets having identical time indices and durations, and each streamlet of the set of streamlets having a unique bitrate, and wherein the encoding module comprises a master module configured to assign an encoding job to one of a plurality of host computing modules in response to an encoding job completion bid. A method includes receiving media content, segmenting the media content and generating a plurality of streamlets, and generating a set of streamlets.

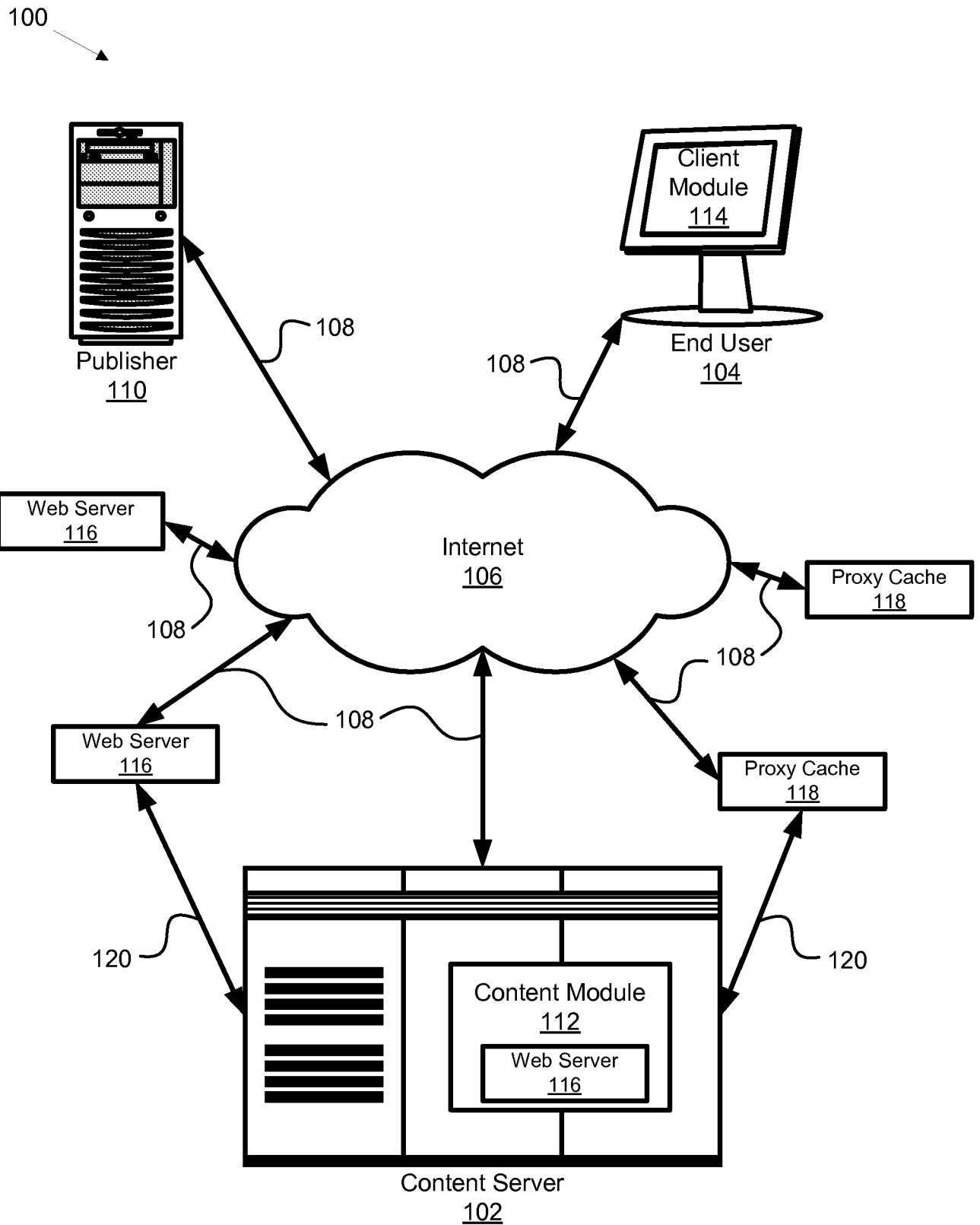


FIG. 1

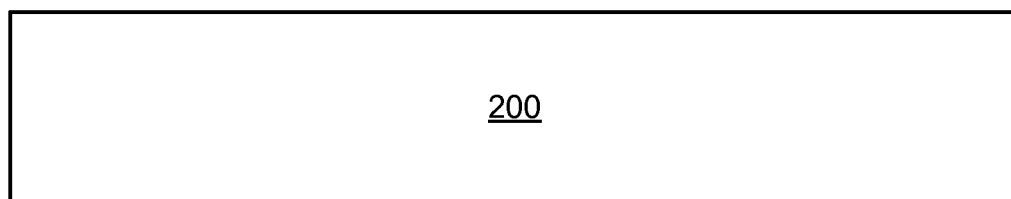


FIG. 2a

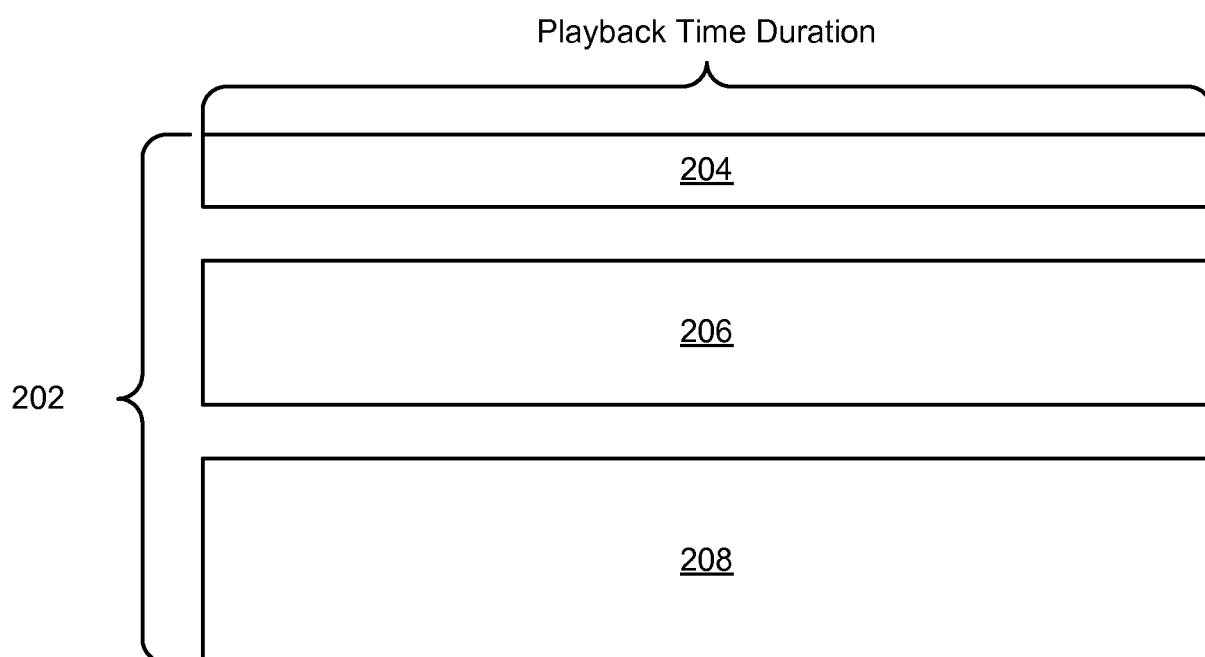


FIG. 2b

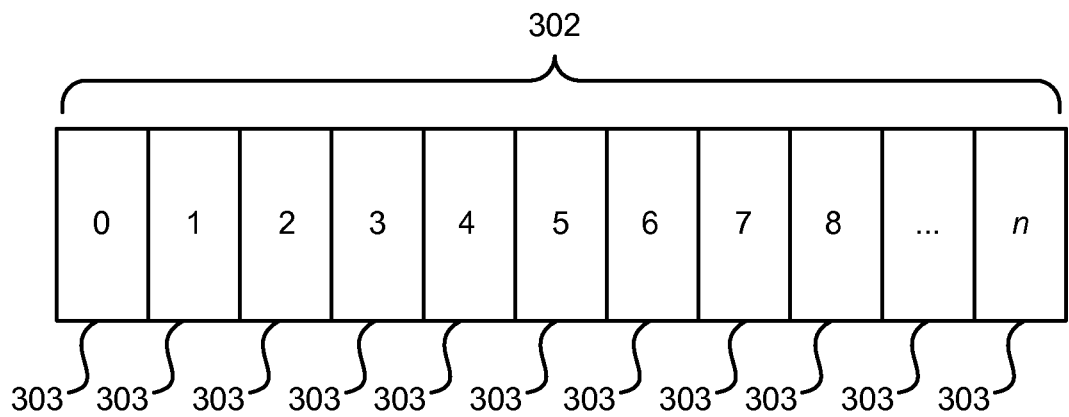


FIG. 3a

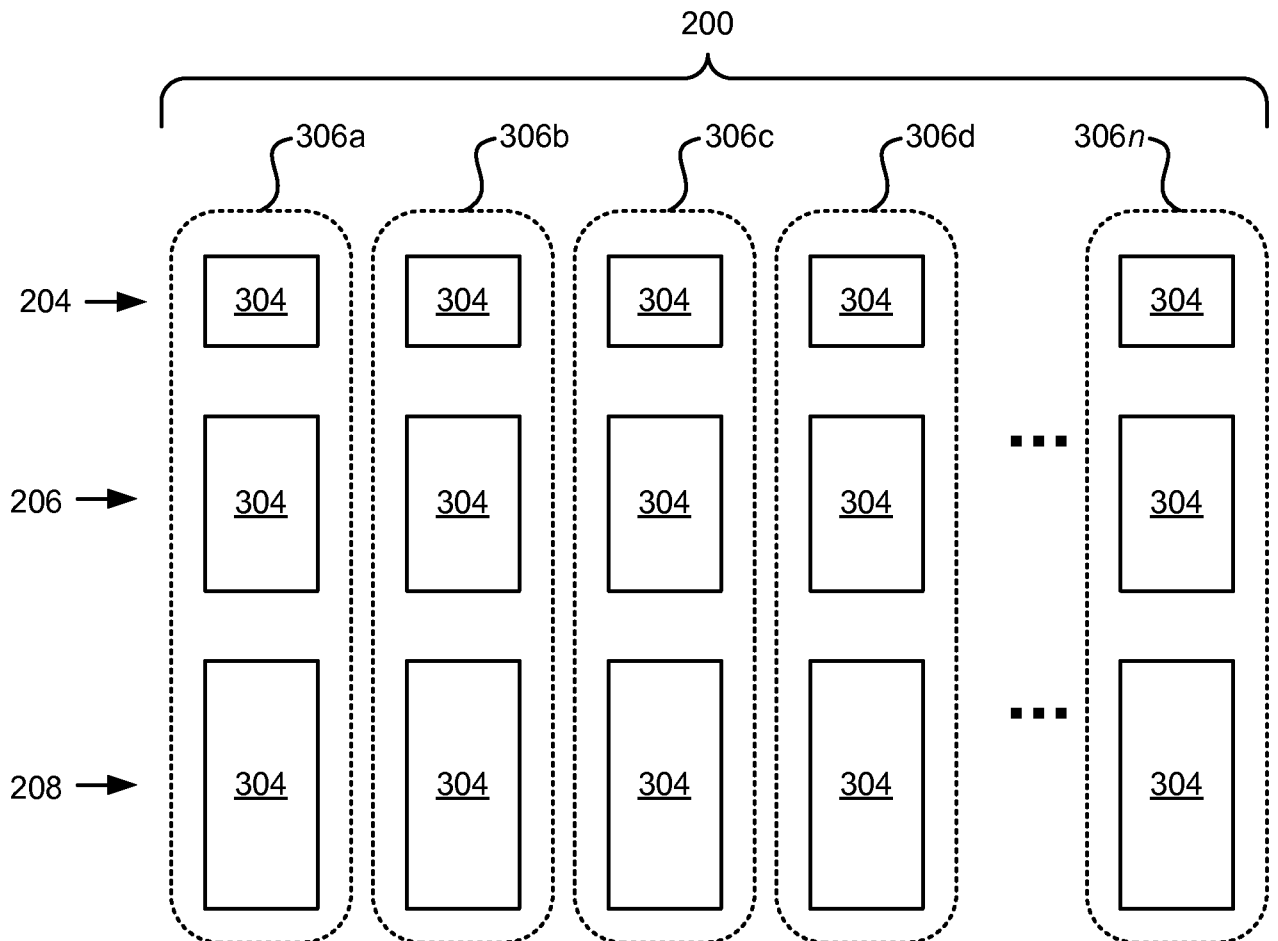


FIG. 3b

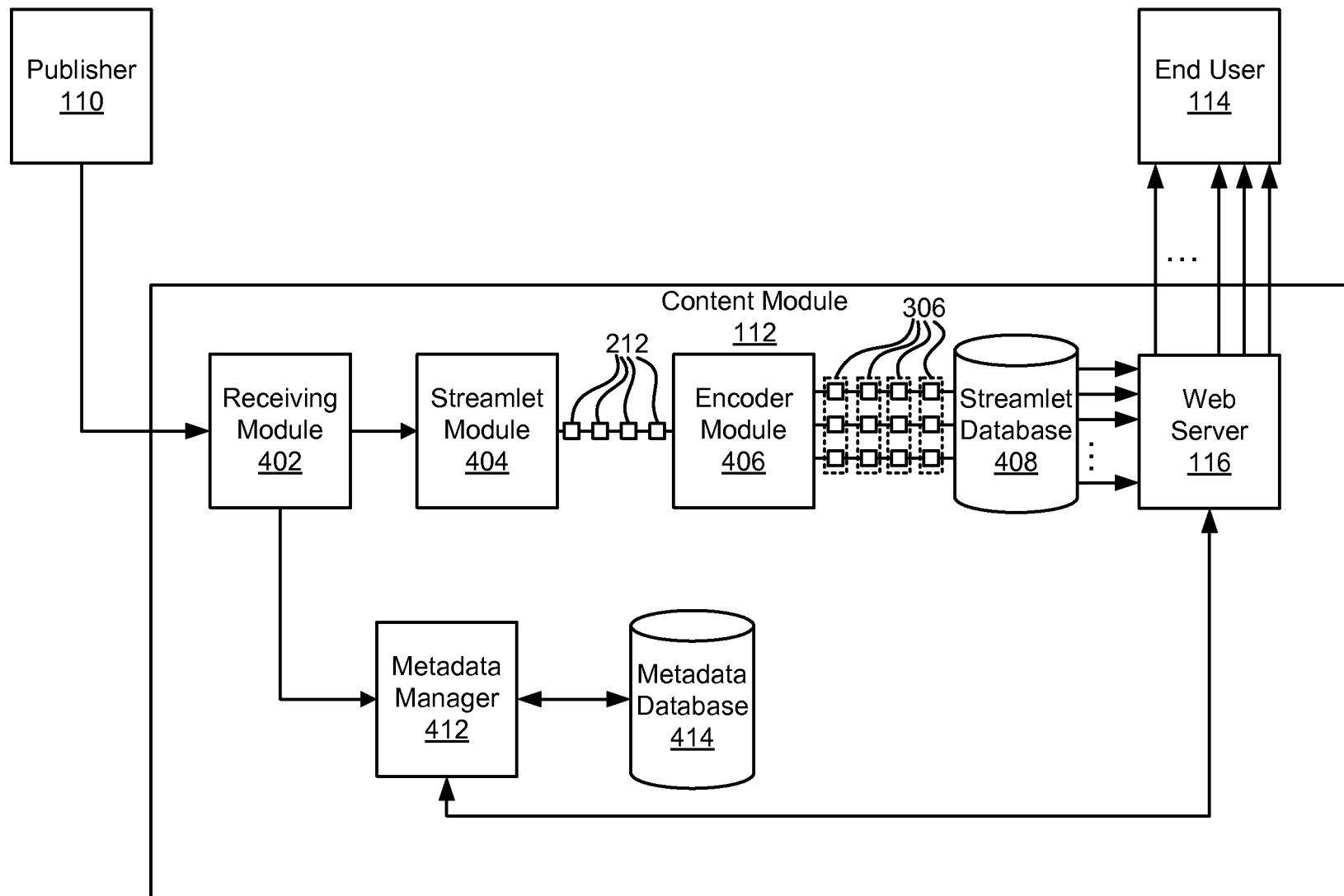


FIG. 4

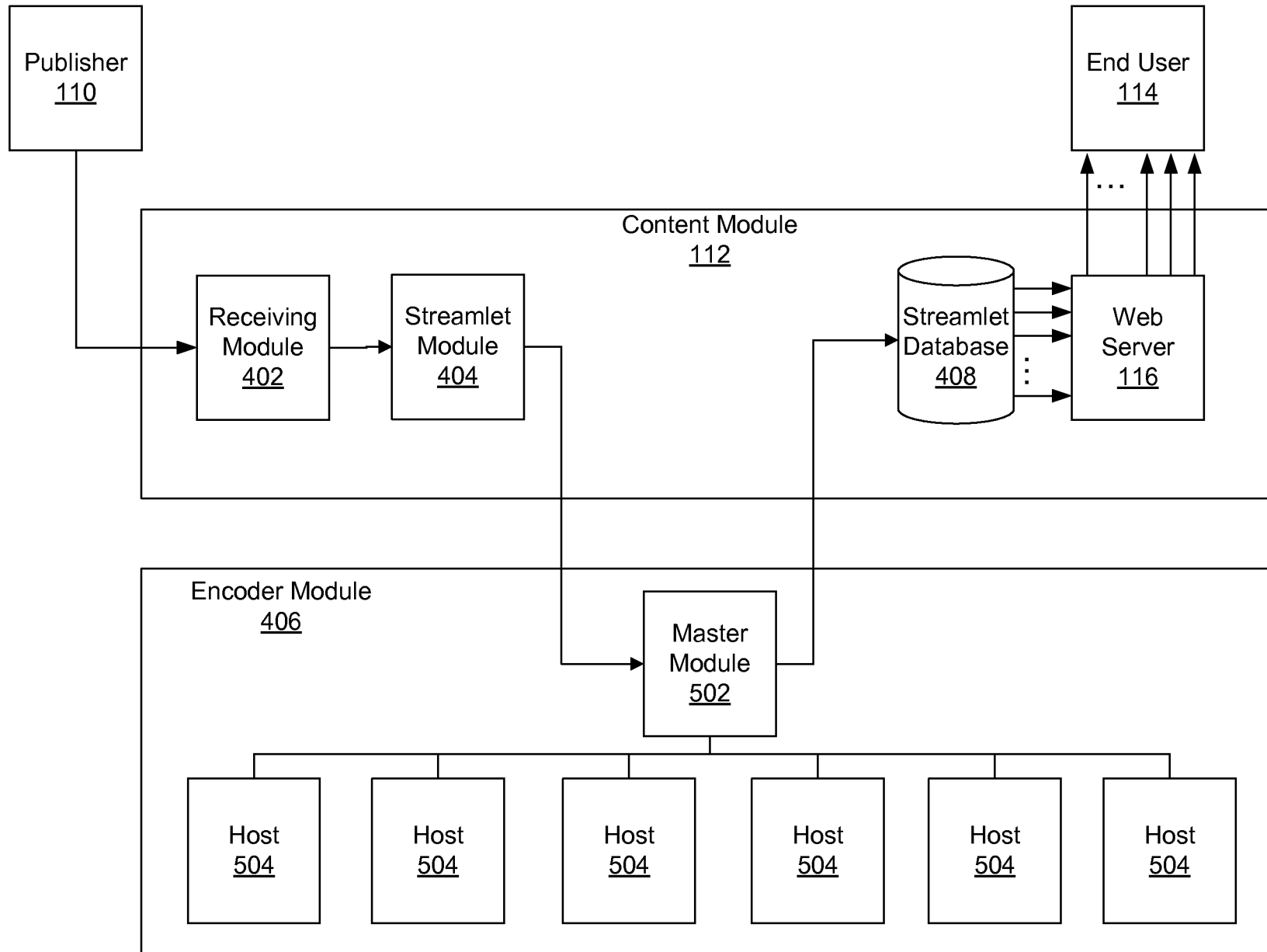


FIG. 5a

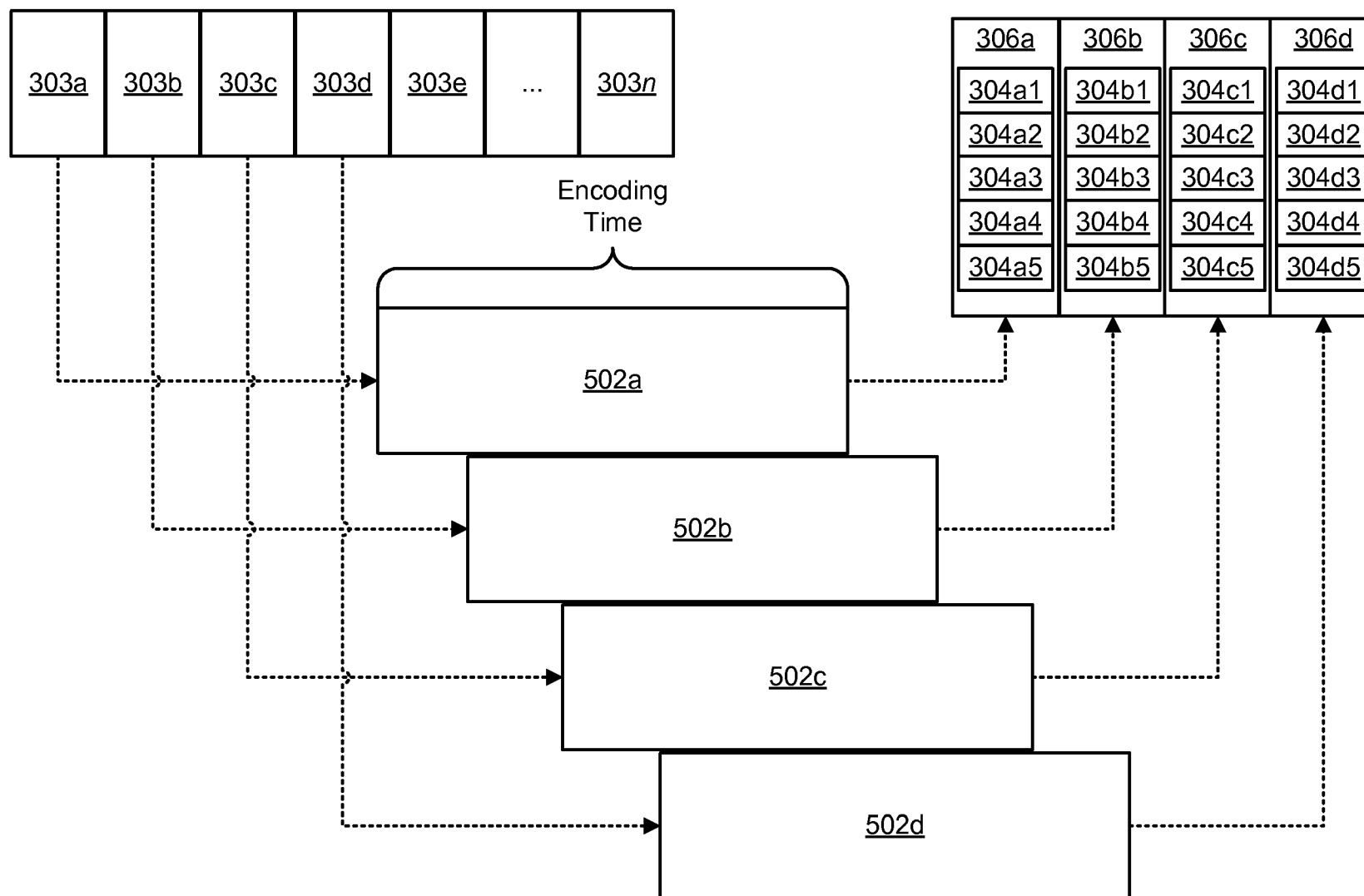


FIG. 5b

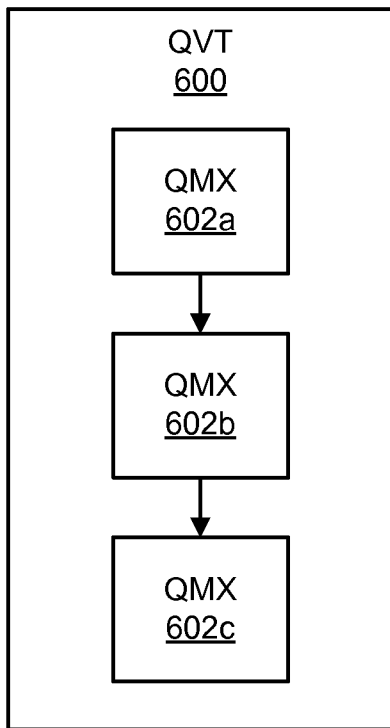


FIG. 6a

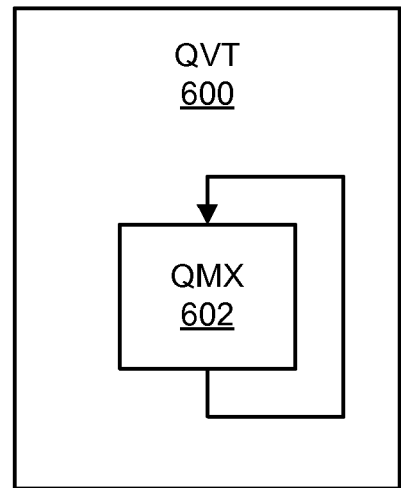


FIG. 6b

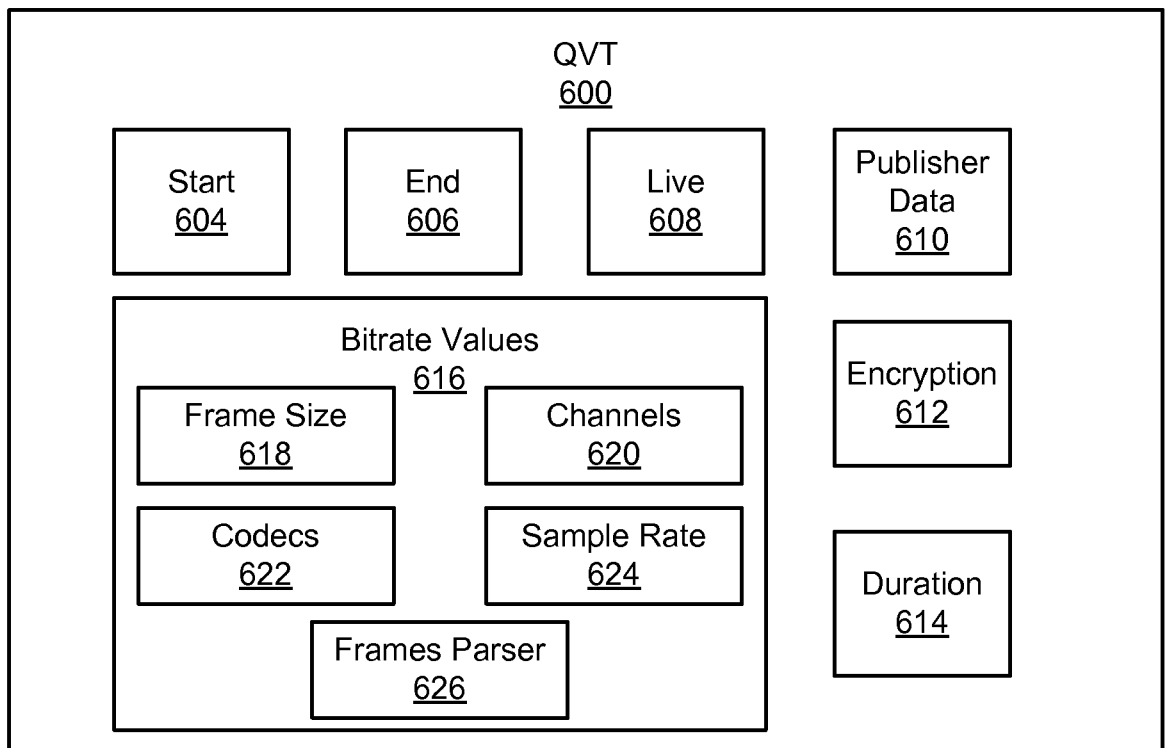


FIG. 6c

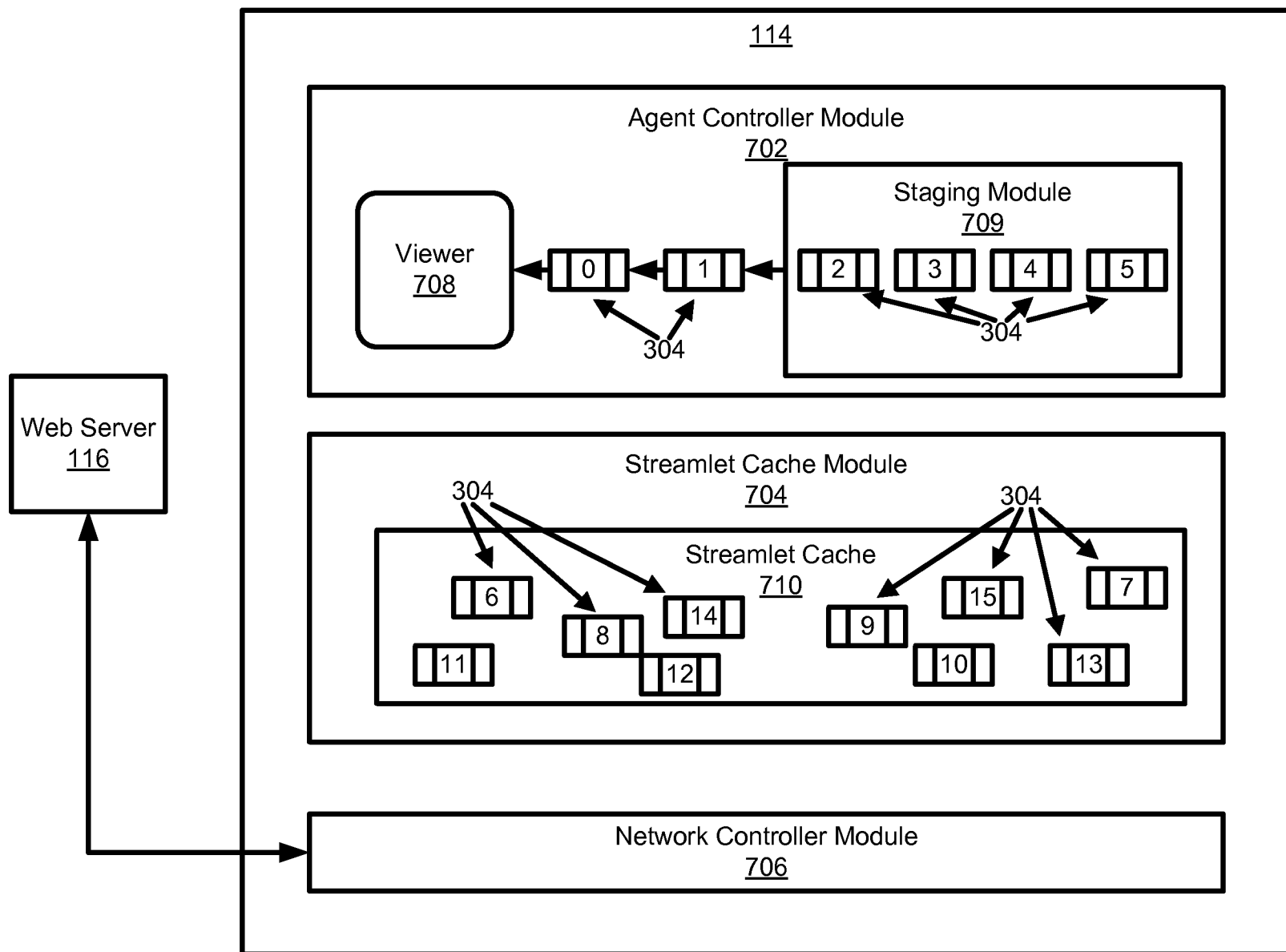


FIG. 7

800 ↘

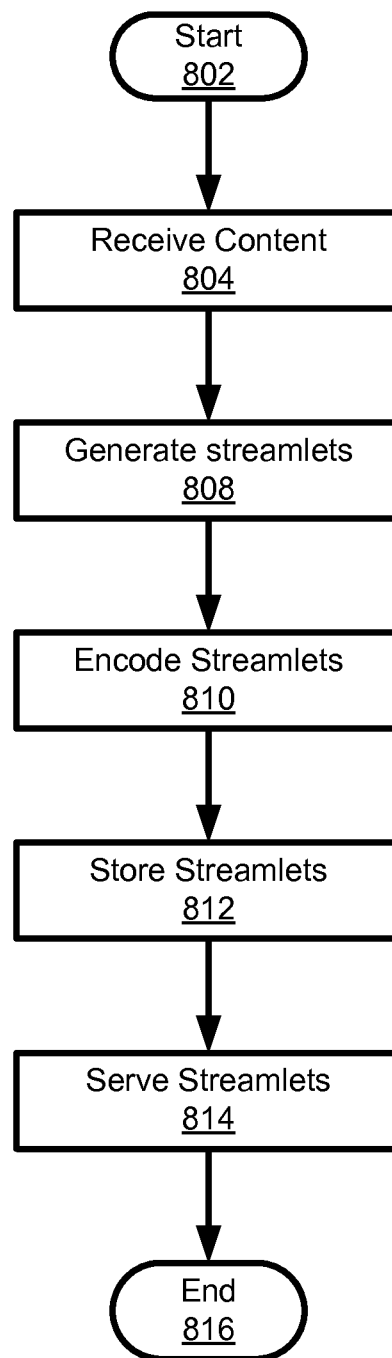


FIG. 8

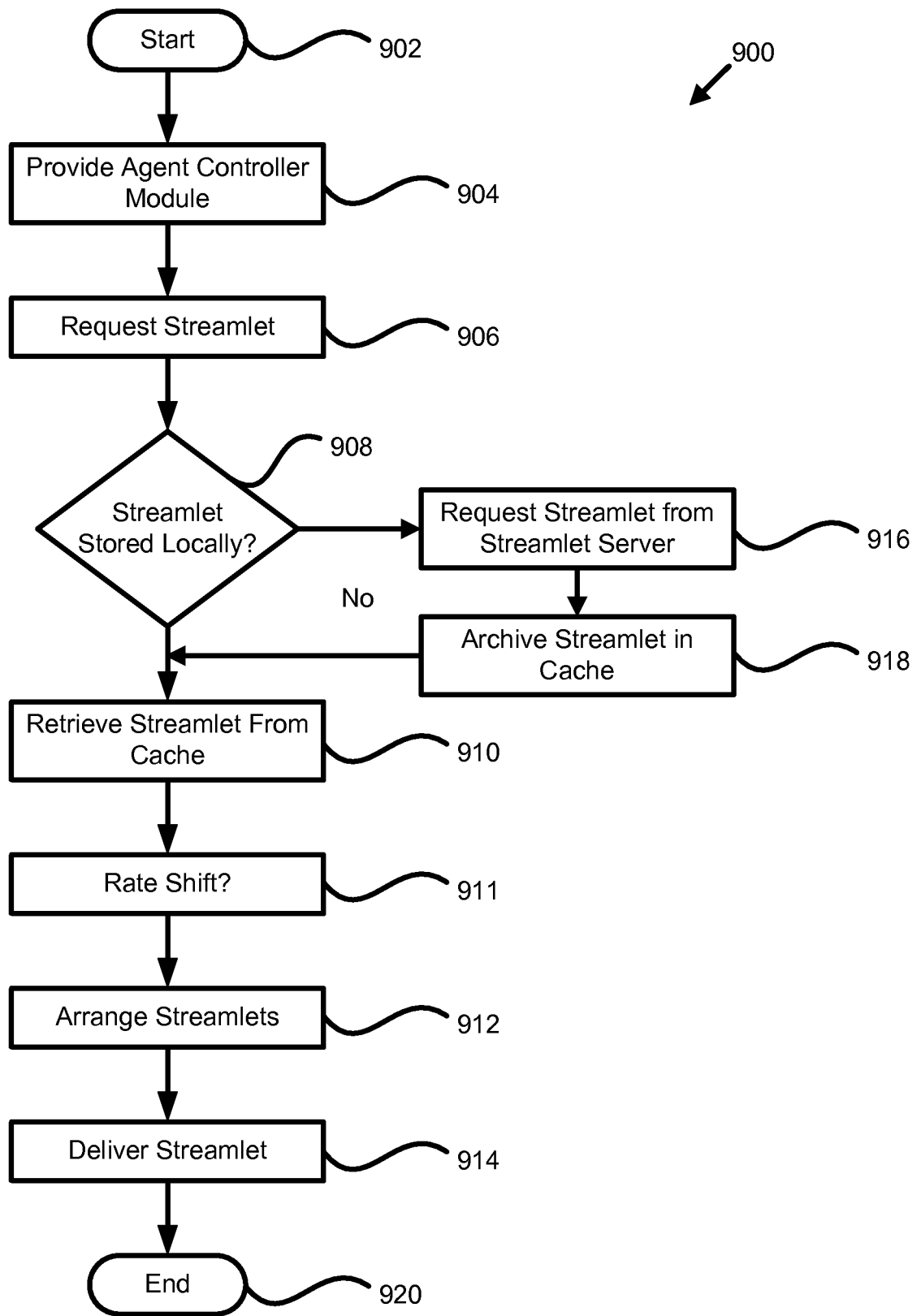


FIG. 9

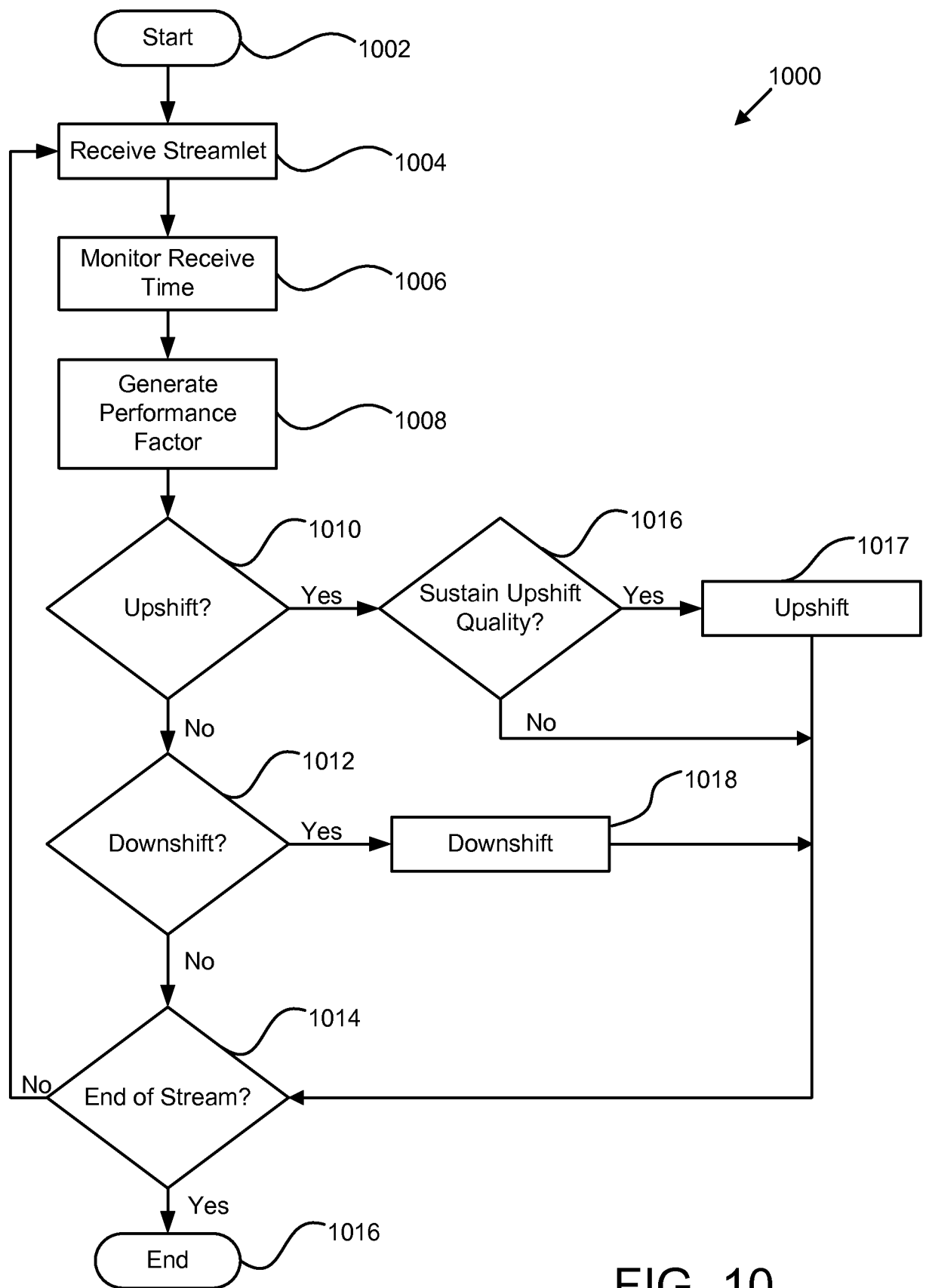


FIG. 10

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 16/252,188			
APPLICATION AS FILED - PART I									
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA		RATE(\$)	FEE(\$)	RATE(\$)	FEE(\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A		N/A	300		
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A		N/A		N/A	660		
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A		N/A	760		
TOTAL CLAIMS (37 CFR 1.16(i))	30 minus 20 =	* 10				x 100 =	1000		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	4 minus 3 =	* 1				x 460 =	460		
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						0.00		
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00		
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL		TOTAL	3180		
APPLICATION AS AMENDED - PART II									
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)		
	Total (37 CFR 1.16(i))	* Minus **	=	x =		x =			
	Independent (37 CFR 1.16(h))	* Minus ***	=	x =		x =			
	Application Size Fee (37 CFR 1.16(s))								
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
				TOTAL ADD'L FEE		TOTAL ADD'L FEE			
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)		
	Total (37 CFR 1.16(i))	* Minus **	=	x =		x =			
	Independent (37 CFR 1.16(h))	* Minus ***	=	x =		x =			
	Application Size Fee (37 CFR 1.16(s))								
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
				TOTAL ADD'L FEE		TOTAL ADD'L FEE			
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.									



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 NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
16/252,188	01/18/2019	2447	3180	066.0141X1C7	30	4

CONFIRMATION NO. 3722

FILING RECEIPT



0000000105759891

70560
LKGLOBAL (Dish Technologies L.L.C.)
7010 E. COCHISE ROAD
SCOTTSDALE, AZ 85253

Date Mailed: 02/11/2019

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

DAVID F. BRUECK, SARATOGA SPRINGS, UT;
MARK B. HURST, CEDAR HILLS, UT;
R. DREW MAJOR, OREM, UT;

Applicant(s)

DISH Technologies L.L.C., Englewood, CO

Assignment For Published Patent Application

DISH Technologies L.L.C., Englewood, CO

Power of Attorney: The patent practitioners associated with Customer Number 70560

Domestic Priority data as claimed by applicant

This application is a CON of 16/004,056 06/08/2018
which is a CON of 15/414,025 01/24/2017 PAT 9998516
which is a CON of 14/719,122 05/21/2015 PAT 9571551
which is a CON of 14/106,051 12/13/2013 PAT 9071668
which is a CON of 13/617,114 09/14/2012 PAT 8612624
which is a CON of 12/906,940 10/18/2010 PAT 8402156
which is a CON of 11/673,483 02/09/2007 PAT 7818444
which is a CIP of 11/116,783 04/28/2005 PAT 8868772
which claims benefit of 60/566,831 04/30/2004

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.) - None.

Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 02/08/2019

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 16/252,188**

Projected Publication Date: 05/23/2019

Non-Publication Request: No

Early Publication Request: No
Title

APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING

Preliminary Class

709

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor

community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.



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.
16/252,188	01/18/2019	DAVID F. BRUECK	066.0141X1C7	3722
70560	7590	02/15/2019	EXAMINER	
LKGLOBAL (Dish Technologies L.L.C.) 7010 E. COCHISE ROAD SCOTTSDALE, AZ 85253				
			ART UNIT	PAPER NUMBER
			2447	
			NOTIFICATION DATE	DELIVERY MODE
			02/15/2019	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

dish@lkglobal.com
ipdept@dish.com

<i>Decision Granting Request for Prioritized Examination (Track I)</i>	Application No. 16/252,188	Applicant(s) BRUECK et al.	
	Examiner BRIAN W BROWN	Art Unit OPET	AIA (First Inventor to File) Status No
<p>1. THE REQUEST FILED <u>18 January 2019</u> IS GRANTED .</p> <p>The above-identified application has met the requirements for prioritized examination</p> <p>A. ☒ for an original nonprovisional application (Track I).</p> <p>B. ☐ for an application undergoing continued examination (RCE).</p> <p>2. The above-identified application will undergo prioritized examination. The application will be accorded special status throughout its entire course of prosecution until one of the following occurs:</p> <p>A. filing a <u>petition for extension of time</u> to extend the time period for filing a reply;</p> <p>B. filing an <u>amendment to amend the application to contain more than four independent claims, more than thirty total claims</u>, or a multiple dependent claim;</p> <p>C. filing a <u>request for continued examination</u> ;</p> <p>D. filing a notice of appeal;</p> <p>E. filing a request for suspension of action;</p> <p>F. mailing of a notice of allowance;</p> <p>G. mailing of a final Office action;</p> <p>H. completion of examination as defined in 37 CFR 41.102; or</p> <p>I. abandonment of the application.</p> <p>Telephone inquiries with regard to this decision should be directed to BRIAN BROWN at (571)272-5338. In his/her absence, calls may be directed to Petition Help Desk at (571) 272-3282.</p>			
/BRIAN W BROWN/ Petitions Examiner, OPET			

<i>Notice of References Cited</i>	Application/Control No. 16/252,188		Applicant(s)/Patent Under Reexamination BRUECK et al.	
	Examiner CHIRAG R PATEL		Art Unit 2454	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-7369610-B2	05-2008	Xu; Jizheng	H04N21/2662	375/240.08
*	B	US-20050055425-A1	03-2005	Lango, Jason	H04L29/06027	709/219
*	C	US-20030103571-A1	06-2003	Meehan, Joseph	H04N21/234327	375/240.27
*	D	US-6496980-B1	12-2002	Tillman; Rachel E.	H04N7/17318	348/E7.071
*	E	US-20030023982-A1	01-2003	Lee, Tsu-Chang	H04N21/234327	725/116
*	F	US-5812786-A	09-1998	Seazholtz; John W.	H04M11/062	370/465
*	G	US-7934159-B1	04-2011	Rahman; Shafiq Ur	H04N21/4825	715/716
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC 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.



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.
16/252,188	01/18/2019	DAVID F. BRUECK	066.0141X1C7	3722
70560	7590	03/22/2019		
LKGLOBAL (Dish Technologies L.L.C.)			EXAMINER	
7010 E. COCHISE ROAD			PATEL, CHIRAG R	
SCOTTSDALE, AZ 85253				
			ART UNIT	PAPER NUMBER
			2454	
			NOTIFICATION DATE	DELIVERY MODE
			03/22/2019	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

dish@lkglobal.com
ipdept@dish.com

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

Double Patenting

A rejection based on double patenting of the “same invention” type finds its support in the language of 35 U.S.C. 101 which states that “whoever invents or discovers any new and useful process... may obtain a patent therefor...” (Emphasis added). Thus, the term “same invention,” in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the claims that are directed to the same invention so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

Claims 1-30 of the instant application are objected to under 37 CFR 1.75 as being a substantial duplicate of claims 1-20 of copending Application NO. 16/262356. When two claims in an application are duplicates or else are so close in content that they both cover the same thing, despite a slight difference in wording, it is proper after allowing one claim to object to the other as being a substantial duplicate of the allowed claim. See MPEP § 706.03(k).

Claims 1-30 provisionally rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1-30 of copending Application No. 16252356 (reference application). This is a provisional statutory double patenting rejection since the claims directed to the same invention have not in fact been patented.

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on nonstatutory double patenting provided the reference application or patent either is shown to be commonly owned with the examined application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. See MPEP § 717.02 for applications subject to examination under the first inventor to file provisions of the AIA as explained in MPEP § 2159. See MPEP §§ 706.02(I)(1) -

706.02(l)(3) for applications not subject to examination under the first inventor to file provisions of the AIA. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO Internet website contains terminal disclaimer forms which may be used. Please visit www.uspto.gov/patent/patents-forms. The filing date of the application in which the form is filed determines what form (e.g., PTO/SB/25, PTO/SB/26, PTO/AIA/25, or PTO/AIA/26) should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp.

Claims 1-2, 4, 7, 16-17, 26-27 and 30 are rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1 and 13-14 of U.S. Patent No. (US 9,571,551) in view of Meehan et al.— hereinafter Meehan (US 2003/0103571)

As per claims 1-2, 4 and 7, differences are that claims 1, 13, and 14 of the instant application claims a low quality, medium quality and a high quality stream and it would have been obvious for the patents of “different bit rates” to teach a low quality, a medium quality and a high quality stream. It would have been obvious for the patent to teach the limitations of “wherein the least of the low quality, medium quality and high quality stream is encode at a bitrate of no less than 600kbps” as Meehan teaches multiple enhancements layers which also equate to the medium or high quality stream

of the patent per [0064]; “Any number of enhancement layers with varying degrees of robustness may then be transmitted” and that enhancement layer is greater than 600kbps as claimed per [0065], “An enhanced layer 33 may then be mapped onto a less robust 12 Mbps or other physical layer (such as 48 Mbps or 54 Mbps)” for the motivation to optimize the quality of a video stream for wireless transmission. ([0019])

As for claim 2 claims that the second streamlet of each of the group has the same second duration in the low, medium and high quality stream and have the same bitrate is anticipated by claim 1 of the patent, state time indexes of the streamlets are the same for the different copies such that the streamlets with the same time indexes from the different copies independently yield the same portions of the single video.

As per claims 4 and 7 of the instant application is anticipated by claim 1 of the patent. Differences for claim 4 are that it would have been obvious for the web servers of the patent to be located at different locations across the network connections using the 1-way test and that the limitations of “end user station to select a higher or lower bitrate version of the live event video is anticipated by the patent which claim to “end user station selects the currently selected one of the different copies dependent upon ongoing determinations to shift the playback quality to a higher or lower quality” as the Meehan discloses multiple enhancement layers, which equate to a medium and a high quality. Examiner have shown that these claims are obvious variants of each other and not patentably distinct.

As per claim 16 of the instant application claims a low quality, medium quality and a high quality stream and it would have been obvious for the patents of “different bit rates” to teach a low quality, a medium quality and a high quality stream. It would have been obvious for claims 1, 13 and 14 of the patent to teach the limitations of “wherein the least of the low quality, medium quality and high quality stream is encode at a bitrate of no less than 600kbps” as Meehan teaches multiple enhancements layers which also equate to the medium or high quality stream of the patent per [0064]; “Any number of enhancement layers with varying degrees of robustness may then be transmitted” and that enhancement layer is greater than 600kbps as claimed per [0065], “An enhanced layer 33 may then be mapped onto a less robust 12 Mbps or other physical layer (such as 48 Mbps or 54 Mbps)” for the motivation to optimize the quality of a video stream for wireless transmission. ([0019]) It would have been obvious for the web servers of the patent to be located at different locations across the network connections using the 1-way test and that the limitations of “end user station to select a higher or lower bitrate version of the live event video is anticipated by the patent which claim to “end user station selects the currently selected one of the different copies dependent upon ongoing determinations to shift the playback quality to a higher or lower quality” and to retrieve, playback and send the streamlets for playback of live video as the Meehan discloses multiple enhancement layers, which equate to a medium and a high quality.

As per claims 26 and 30 of the instant application recites similar limitations as claim 16 and the above discussion apply.

As per claims 17 and 27 of the instant application is anticipated by claim 1 of the patent in that the second streamlet of each of the group has the same second duration in the low, medium and high quality stream and have the same bitrate is anticipated by claim 1 of the patent, state time indexes of the streamlets are the same for the different copies such that the streamlets with the same time indexes from the different copies independently yield the same portions of the single video.

Claim Rejections - 35 USC § 103

The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-3 and 5 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Xu et al. – hereinafter Xu (US 7,369,610) in view of Lango et al. – hereinafter Lango (US 2005/0055425) / Meehan (US 2003/0103571)

As per claim 1, Xu discloses a system for adaptive-rate content streaming of live event video playable on one or more end user stations over the internet, the system comprising:

at least one storage device storing event video, (Col 3 lines 47-60)the live event video encoded at a plurality of bitrates creating a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream, the low quality

stream, the medium quality stream, and the high quality stream each comprising a group of streamlets encoded at a respective one of the plurality of bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video; (Col 3 line 61- Col 4 line 24)

the first streamlet of the low quality stream having a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream. (Col 2 lines 39-51)

Xu fails to disclose live event video and wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps and wherein the first streamlet of each of the groups of streamlets has the same first duration and corresponds to the same first portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream.

Lango discloses wherein the first streamlet of each of the groups of streamlets has the same first duration and corresponds to the same first portion of the event video in the low quality stream, the medium quality stream, and the high quality stream. (Col 2 lines 39-52)

At the time of invention, it would have been obvious for the streamlets of Xu to be at different bit-rates at each enhancement layer and the same duration as taught by Lango. The motivation would have been to reduce drifting and improve the video quality. (Col 2 lines 17-25)

Meehan discloses live event video ([0039]) and wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a

bitrate of no less than 600 kbps. ([0065], “An enhanced layer 33 may then be mapped onto a less robust 12 Mbps or other physical layer (such as 48 Mbps or 54 Mbps)”

At the time of invention, it would have been obvious for the streamlets of Xu to be at different bit-rates and have the same duration at each enhancement layer as taught by Lango and for the stream to be a live event and one of the streams to be encoded at no less than 600kbps as taught by Meehan. The motivation would have been to optimize the quality of a video stream for wireless transmission. ([0019]

As per claim 2, Xu / Lango / Meehan disclose the system of claim 1. Lango discloses wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream. (Col 2 lines 39-52). Xu discloses the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream. (Col 8 line 60 – Col 9 line 2; The base layer bitstreams 304 are encoded at a base bit-rate, denoted as R.sub.B. Bit-rate refers to the number of bits used to represent a layer of a video frame.)

As per claim 3, Xu / Lango / Meehan discloses the system of claim 2. Lango discloses wherein the first and second durations are different. ([0072])

As per claim 5, Xu / Lango / Meehan disclose the system of claim 1. Lango discloses wherein each of the first streamlets has a first duration that is the range of 0.1 to 5 seconds. ([0072])

Claims 4, 7-8, 10-12, 15-20, 22-24 and 26-30 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Xu (US 7,369,610) / Lango (US 2005/0055425) / Meehan (US 2003/0103571) further in view of Tillman et al. – hereinafter Tillman (US 6,496,980)

As per claim 4, Xu / Lango / Meehan disclose the system of claim 1. Xu discloses further comprising: a plurality of web servers located at different locations across the internet, (Col 3 lines 40-46) each web server.

Lango fails to disclose receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing a portion of the video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams; retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and send the retrieved first streamlet from the currently selected one of the different copies to the requesting one of the end user stations over the one or more network connections.

Tillman discloses receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing a portion of the video, (Col 10 line 35 – Col 11 line 15)

wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams; (Col 10 line 35 – Col 11 line 15)

retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and send the retrieved first streamlet from the currently selected one of the different copies to the requesting one of the end user stations over the one or more network connections. (Col 10 line 35 – Col 11 line 15)

At the time of invention, it would have been obvious for the streamlets of Xu to be at different bit-rates and have the same duration at each enhancement layer as taught by Lango and for the stream to be a live event and one of the streams to be encoded at no less than 600kbps as taught by Meehan and for the client to request a first streamlet from a low, medium and high quality stream version, retrieving and transmitting the streamlet over the network connections as taught by Tillman. The motivation would have been to provide a higher quality video on demand for certain portion of the stream. (Col 2 lines 28-30)

As per claim 7, Xu / Lango / Meehan disclose the system of claim 1. Tillman discloses to disclose further comprising: a first web server configured to:

receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the

first portion of the live event video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the live event video; (Col 10 line 35 – Col 11 line 15)

retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and (Col 10 line 35 – Col 11 line 15)

send the retrieved first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream to the requesting one of the end user stations over the one or more network connections. (Col 10 line 35 – Col 11 line 15)

As per claim 8, Xu / Lango / Meehan/ Tillman disclose the system of claim 7. Tillman discloses wherein the first streamlets of the low quality stream, the medium quality stream, and the high quality stream are available before the live event is complete. (Col 9 line 58 – Col 10 line 12; When the multimedia content being transmitted represents a live event, server system 32 transmits video data representing the live event on the base layer of the stream. Prior to transmission of the base layer, the server system encodes the base layer and the enhancement layers and stores the enhancement layers in the server cache 38 for future use during replay.)

As per claim 10, Xu / Lango / Meehan / Tillman disclose the method of claim 7.

Tillman discloses wherein the streamlets from the low quality stream, the medium quality stream, and the high quality stream of the live event, when played back, appear live to a viewer. (Col 10 lines 8-34; When a user of a client system desires to view streaming digital video, the client system contacts the server system to initiate the transmission of a digital video stream and/or an associated audio stream corresponding to selected multimedia content (either pre-recorded or live).

As per claim 11, Xu / Lango / Meehan / Tillman disclose the system of claim 7.

Xu discloses wherein the first web server is further configured to: receive at least one virtual timeline request over the one or more internet connections from the one or more end user stations to retrieve a virtual timeline; and send the virtual timeline to the requesting one of the end user stations over the one or more network connections. (Col 9 lines 54-65; Fig. 3: item 310)

As per claim 12, Xu / Lango / Meehan / Tillman disclose the system of claim 11.

Xu discloses wherein the virtual timeline corresponds to the currently selected one of the low quality stream, the medium quality stream, and the high quality stream. (Col 9 lines 54-65; Fig. 3: item 310)

As per claim 15, Xu / Lango / Meehan / Tillman disclose the system of claim 11.

Tillman discloses wherein the virtual timeline comprises at least one quantum media extension (QMX). (Col 7 lines 51-63; The catalog may comprise information items such

as stream information (source, title, author, copyright, etc.), format (video size, coding algorithm, etc.) and other video attributes (sequence length, optimal display size, optimal post-processing options, etc.))

As per claim 16, Xu discloses an end user station to stream a live event video over a network from a server for playback of the video, the content player device comprising: a processor;

a digital processing apparatus memory device comprising non-transitory machine-readable instructions that, when executed, cause the processor to: (Col 3 lines 47-60)

wherein the live event video is encoded at a plurality of bitrates to create a plurality of streams including at least a low quality stream, a medium quality stream, and a high quality stream, each of the low quality stream, the medium quality stream, and the high quality stream comprising a group of streamlets encoded at the same respective bitrate, (Col 3 line 61- Col 4 line 24)

Xu fails to disclose each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video; wherein the first streamlets of each of the low quality stream, the medium quality stream and the high quality stream each has an equal playback duration and each of the first streamlets corresponding to the same portion of the live event video; wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bit rate of no less than 600 kbps; establish one or more network connections between the end user station and the server, wherein the server is configured to access

at least one of a plurality of groups of streamlets; select a specific one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams; place a streamlet request to the server over the one or more network connections for the first streamlet of the selected stream; receive the requested first streamlet from the server via the one or more network connections; provide the received first streamlet for playback of the live event video.

Lango discloses each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video; (Col 2 lines 39-52)

wherein the first streamlets of each of the low quality stream, the medium quality stream and the high quality stream each has an equal playback duration and each of the first streamlets corresponding to the same portion of the live event video; (Col 2 lines 39-52)

At the time of invention, it would have been obvious for the streamlets of Xu to be at different bit-rates at each enhancement layer and the same duration as taught by Lango. The motivation would have been to reduce drifting and improve the video quality. (Col 2 lines 17-25)

Meehan discloses wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bit rate of no less than 600 kbps; and ([0065], "An enhanced layer 33 may then be mapped onto a less robust 12 Mbps or other physical layer (such as 48 Mbps or 54 Mbps)")

At the time of invention, it would have been obvious for the streamlets of Xu to be at different bit-rates at each enhancement layer and the same duration as taught by Lango and for the stream to be a live event and one of the streams to be encoded at no less than 600kbps as taught by Meehan. The motivation would have been to optimize the quality of a video stream for wireless transmission. ([0019])

Tillman discloses establish one or more network connections between the end user station and the server, wherein the server is configured to access at least one of a plurality of groups of streamlets; (Col 10 line 35 – Col 11 line 15)

select a specific one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams; (Col 10 line 35 – Col 11 line 15)

place a streamlet request to the server over the one or more network connections for the first streamlet of the selected stream; (Col 10 line 57- Col 11 line 15)

receive the requested first streamlet from the server via the one or more network connections; and (Col 10 line 57- Col 11 line 15)

provide the received first streamlet for playback of the live event video. (Col 10 line 57- Col 11 line 15)

At the time of invention, it would have been obvious for the streamlets of Xu to be at different bit-rates at each enhancement layer and the same duration as taught by Lango and for the stream to be a live event and one of the streams to be encoded at no less than 600kbps as taught by Meehan and for the client to request a first streamlet from a low, medium and high quality stream version, retrieving and transmitting the streamlet over the network connections as taught by Tillman. The motivation would

have been to provide a higher quality video on demand for certain portion of the stream.

(Col 2 lines 28-30)

As per claims 17 and 27, please see the discussion under claim 2 as similar logic applies.

As per claims 18 and 28, please see the discussion under claim 3 as similar logic applies.

As per claim 19, please see the discussion under claim 5 as similar logic applies.

As per claims 20 and 29, please see the discussion under claim 8 as similar logic applies.

As per claim 22, please see the discussion under claim 10 as similar logic applies.

As per claim 23, please see the discussion under claim 11 as similar logic applies.

As per claim 24, please see the discussion under claim 12 as similar logic applies.

As per claim 26, please see the discussion under claims 1 and 4 as similar logic applies.

As per claim 30, please see the discussion under claim 16 as similar logic applies.

Claim 6 is rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Xu (US 7,369,610) / Lango (US 2005/0055425) / Meehan (US 2003/0103571) further in view of Lee et al . – hereinafter Lee (US 2003/0023982)

As per claim 6, Xu / Lango / Meehan disclose the system of claim 1. Xu fails to disclose wherein the live event is a live sports event. Lee discloses wherein the live event is a live sports event. ([0051])

At the time of invention, it would have been obvious for the streamlets of Xu to be at different bit-rates and have the same duration at each enhancement layer as taught by Lango and for the stream to be a live event and one of the streams to be encoded at no less than 600kbps as taught by Meehan and for the live stream event of Meehan to be a sports event as taught by Lee. The motivation would have been so that the event can be seen by a plurality of devices and to satisfy the interests of the users. ([0051])

Claim 9 is rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Xu (US 7,369,610) / Lango (US 2005/0055425) / Meehan (US 2003/0103571) / Tillman

(US 6,496,980) further in view of Seazholtz et al. – hereinafter Seazholtz (US 5,812,786)

As per claim 9, Xu / Lango / Meehan / Tillman disclose the system of claim 7. Xu fails to disclose wherein the streamlets of the low quality stream, the medium quality stream, and the high quality stream of the live event are available on a 10 second delay. Seaholtz discloses wherein the streamlets of the low quality stream, the medium quality stream, and the high quality stream of the live event are available on a 10 second delay. (Col 14 lines 23-48)

At the time of invention, it would have been obvious for the streamlets of Xu to be at different bit-rates and have the same duration at each enhancement layer as taught by Lango and for the stream to be a live event and one of the streams to be encoded at no less than 600kbps as taught by Meehan and for the client to request a first streamlet from a low, medium and high quality stream version, retrieving and transmitting the streamlet over the network connections as taught by Tillman and for the delays of the streamlets to be 10 seconds as taught by Seazholtz. The motivation would have been to support bi-directional communication and multi-cast capability. (Col 2 lines 37-45)

Claims 13-14 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Xu (US 7,369,610) / Lango (US 2005/0055425) / Meehan (US 2003/0103571) / Tillman (US 6,496,980) further in view of Rahman et al. – hereinafter Rahman (US 7,934,159)

As per claim 13, Xu / Lango / Meehan / Tillman disclose the system of claim 11.

Xu fails to disclose wherein the virtual timeline defines a playlist for a user to view.

Rahman discloses wherein the virtual timeline defines a playlist for a user to view. (Col 21 line 64 – Col 22 line 30)

At the time of invention, it would have been obvious for the streamlets of Xu to be at different bit-rates and have the same duration at each enhancement layer as taught by Lango and for the stream to be a live event and one of the streams to be encoded at no less than 600kbps as taught by Meehan and for the client to request a first streamlet from a low, medium and high quality stream version, retrieving and transmitting the streamlet over the network connections as taught by Tillman and for the virtual timeline to define a playlist for the user to view as taught by Rahman. The motivation would have been to make efficient use of hardware and software resources. (Col 1 line 65 – Col 2 line 5)

As per claim 14, Xu / Lango / Meehan / Tillman disclose the system of claim 11.

Rahman discloses wherein the virtual timeline comprises a file that is configured to define a playlist for a user to view. (Col 24 lines 19-23)

Claim 21 is rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Xu (US 7,369,610) / Lango (US 2005/0055425) / Meehan (US 2003/0103571) / Tillman (US 6,496,980) further in view of Seazholtz (US 5,812,786)

As per claim 21, please see the discussion under claim 9 as similar logic applies.

Claim 25 is rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Xu (US 7,369,610) / Lango (US 2005/0055425) / Meehan (US 2003/0103571) / Tillman (US 6,496,980) further in view of Rahman (US 7,934,159)

As per claim 25, please see the discussion under claim 13 as similar logic applies.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chirag R Patel whose telephone number is (571)272-7966. The examiner can normally be reached on Monday to Friday from 8:00AM to 4:30PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess, can be reached on 571-272-3949. 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://pairedirect.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).

/Chirag R Patel/
Primary Examiner, Art Unit 2454

<i>Search Notes</i> 	Application/Control No. 16/252,188	Applicant(s)/Patent Under Reexamination BRUECK et al.
	Examiner CHIRAG R PATEL	Art Unit 2454

CPC - Searched*		
Symbol	Date	Examiner
H04N19/34, H04N19/40, H04N21/23427	03/14/2019	C.P.

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
EAST	03/14/2019	C.P.
Inventor / Double patent search	03/14/2019	C.P.

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner

--	--

Bibliographic Data

Application No: 16/252,188

Foreign Priority claimed: ☐ Yes ☒ No

35 USC 119 (a-d) conditions met: ☐ Yes ☒ No ☐ Met After Allowance

Verified and Acknowledged: /CHIRAG R PATEL/

Examiner's Signature

Initials

Title:

APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE
CONTENT STREAMING

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
01/18/2019	709	2454	066.0141X1C7
RULE			

APPLICANTS

DISH Technologies L.L.C., Englewood, CO, UNITED STATES

INVENTORS

DAVID F. BRUECK SARATOGA SPRINGS, UT, UNITED STATES

MARK B. HURST CEDAR HILLS, UT, UNITED STATES

R. DREW MAJOR OREM, UT, UNITED STATES

CONTINUING DATA

This application is a CON of 16004056 06/08/2018

16004056 is a CON of 15414025 01/24/2017 PAT 9998516

15414025 is a CON of 14719122 05/21/2015 PAT 9571551

14719122 is a CON of 14106051 12/13/2013 PAT 9071668

14106051 is a CON of 13617114 09/14/2012 PAT 8612624

13617114 is a CON of 12906940 10/18/2010 PAT 8402156

12906940 is a CON of 11673483 02/09/2007 PAT 7818444

11673483 is a CIP of 11116783 04/28/2005 PAT 8868772

11116783 has PRO of 60566831 04/30/2004

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

02/08/2019

STATE OR COUNTRY

UNITED STATES

ADDRESS

LKGLOBAL (Dish Technologies L.L.C.)

7010 E. COCHISE ROAD
SCOTTSDALE, AZ 85253
UNITED STATES

FILING FEE RECEIVED

\$7,320

<i>Index of Claims</i> 	Application/Control No. 16/252,188	Applicant(s)/Patent Under Reexamination BRUECK et al.
	Examiner CHIRAG R PATEL	Art Unit 2454

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS										
☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47										
CLAIM			DATE							
Final	Original	03/18/2019								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								
	21	✓								
	22	✓								
	23	✓								
	24	✓								
	25	✓								
	26	✓								
	27	✓								
	28	✓								
	29	✓								
	30	✓								

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	7	((("9998516") or ("9571551") or ("9071668") or ("8612624") or ("8402156") or ("7818444") or ("8868772")).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2019/03/12 11:12
S2	1	("20180295174").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2019/03/12 11:26
S3	0	("("8880721" "20030093790" "8686066" "8683066" "20080120330" "9344496" "20050262257" "9462074" "7925781" "8683066" "6757796" "20050262257" "6389473" "6708213" "20080120330" "20080120342" "7925781" "20110307545" "20030093790" "20080120342" "9344496" "8686066" "20030236904" "20110307545" "8880721" "20020087717").PN.").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2019/03/12 12:07
S4	2	((("20050055425") or ("7369610")).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2019/03/12 12:10
S5	68	encod\$3 near3 stream with "600" near kbps	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/12 12:22
S6	11587	scalable near2 video near2 cod\$3	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/12 12:27
S7	11587	scalable near2 video near2 cod\$3 AND @AD< "200400430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/12 12:28
S8	525	scalable near2 video near2 cod\$3 AND @AD< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/12 12:28
S10	282	enhancement adj layer AND bit-rate with encod\$3	US-PGPUB; USPAT; USOCR;	OR	ON	2019/03/12 12:41

			FPRS; EPO			
S11	233	enhancement adj layer AND bit-rate with encod\$3 and mpeg	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/12 12:42
S12	23	enhancement adj layer AND bit-rate with encod\$3 and mbps	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/12 13:14
S13	23	("8880721" "20030093790" "8686066" "8683066" "20080120330" "9344496" "20050262257" "9462074" "7925781" "8683066" "6757796" "20050262257" "6389473" "6708213" "20080120330" "20080120342" "7925781" "20110307545" "20030093790" "20080120342" "9344496" "8686066" "20030236904" "20110307545" "8880721" "20020087717").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/13 12:37
S14	1	("7818444").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2019/03/14 12:22
S15	0	client near3 request with enhancement near2 layer with switch\$3 with bit-rate	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 12:37
S16	9	client with scalable with video with switch\$3	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 12:40
S17	23	scalable near2 video near2 cod\$3 WITH STREAM AND @AD<"20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 12:40
S18	9	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND client near2 request with streamlet	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 12:41
S19	60	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND switch\$3 with enhancement near2 layer	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 12:41
S20	8	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND request with	US-PGPUB;	OR	ON	2019/03/14 12:42

		switch\$3 with enhancement near2 layer	USPAT; USOCR; FPRS; EPO			
S21	118	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND request with enhancement near2 layer	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 12:42
S22	25	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND request with enhancement near2 layer and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 12:44
S23	0	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND live same chunk\$3 and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 14:10
S24	5	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND live and (streamlet chunk\$3) and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 14:11
S25	1	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND live and (streamlet chunk\$3) and enhancement AND @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 14:12
S26	1	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND live NEAR STREAM and ENHANCEMENT adj layer and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 14:15
S27	4	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND real near2 time with chunk\$3 and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 14:19
S28	4	enhancement near2 layer AND live and (streamlet chunk\$3) and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 14:21
S29	36	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND timeline and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 15:23
S30	1	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND timeline and	US-PGPUB;	OR	ON	2019/03/14 15:23

		enhancement adj layer AND @ad< "20040430"	USPAT; USOCR; FPRS; EPO			
S31	8	timeline SAME enhancement adj layer AND @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 15:28
S32	0	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND VIRTUAL ADJ timeline and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 15:41
S33	9	timeline same playback AND enhancement adj layer AND @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 15:59
S34	13	timeline same playback same playlist AND @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 16:25
S35	0	timeline same playback same playlist AND scalable AND @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 16:25
S36	5	timeline same playback same playlist AND enhancement AND @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 16:25
S37	1	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND delay near2 "10" near2 second and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 19:54
S38	0	enhancement near2 layer AND live same sports and (streamlet chunk\$3) and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 20:04
S39	394	live same sports and (streamlet chunk\$3) and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 20:04
S40	0	scalab\$3 near2 video AND live same sports and (streamlet chunk\$3) and	US-PGPUB;	OR	ON	2019/03/14 20:04

		@ad< "20040430"	USPAT; USOCR; FPRS; EPO			
S41	2	scal\$3 near2 video AND live same sports and (streamlet chunk\$3) and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 20:04
S42	0	live same sports WITH EVEN SAME ENCOD\$3 and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 20:05
S43	0	live same sports WITH EVEN SAME BIT\$2RATE and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 20:05
S44	2	scal\$3 near2 video AND EVENT NEAR3 sports and (streamlet chunk\$3) and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 20:06
S45	0	scal\$3 near2 video SAME EVENT NEAR3 sports and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 20:07
S46	4	((H04N19/34 OR H04N19/40 OR H04N21/234327).CPC.) AND LIVE NEAR3 SPORTS and @ad< "20040430"	US-PGPUB; USPAT; USOCR; FPRS; EPO	OR	ON	2019/03/14 20:09

EAST Search History (I nterference)

< This search history is empty>

3/ 18/ 2019 12:22:43 PM
C:\ Users\ cpatel\ Documents\ EAST\ Workspaces\ 16252188.w sp



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 NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
16/252,188	01/18/2019	DAVID F. BRUECK	066.0141X1C7

CONFIRMATION NO. 3722

PUBLICATION NOTICE



0000000108421360

70560
LKGLOBAL (Dish Technologies L.L.C.)
7010 E. COCHISE ROAD
SCOTTSDALE, AZ 85253

Title: APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING

Publication No. US-2019-0158560-A1

Publication Date: 05/23/2019

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently <http://www.uspto.gov/patft/>.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Public Records Division. The Public Records Division can be reached by telephone at (571) 272-3150 or (800) 972-6382, by facsimile at (571) 273-3250, by mail addressed to the United States Patent and Trademark Office, Public Records Division, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently <https://portal.uspto.gov/pair/PublicPair>. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

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.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875			Application or Docket Number 16/252,188		Filing Date 01/18/2019	☐ To be Mailed
ENTITY: ☒ LARGE ☐ SMALL ☐ MICRO						
APPLICATION AS FILED - PART I						
	(Column 1)	(Column 2)				
FOR	NUMBER FILED	NUMBER EXTRA		RATE (\$)		FEE (\$)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A		
☐ SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A		N/A		
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A		
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 = *			x \$100 =		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 = *			x \$460 =		
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).					
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))						
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL		

APPLICATION AS AMENDED - PART II							
	(Column 1)		(Column 2)	(Column 3)			
AMENDMENT	06/20/2019	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(i))	* 30	Minus	** 30	= 0	x \$100 =	0
	Independent (37 CFR 1.16(h))	* 4	Minus	*** 4	= 0	x \$460 =	0
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	0
	(Column 1)		(Column 2)	(Column 3)			
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(i))	*	Minus	**	=	x \$0 =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x \$0 =	
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.						LIE	
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".						/PEARLIE A FENNELL/	
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".							
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.							

This collection of information is required by 37 CFR 1.16. 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, 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.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Electronic Acknowledgement Receipt

EFS ID:	36357779
Application Number:	16252188
International Application Number:	
Confirmation Number:	3722
Title of Invention:	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING
First Named Inventor/Applicant Name:	DAVID F. BRUECK
Customer Number:	70560
Filer:	Brett Carlson
Filer Authorized By:	
Attorney Docket Number:	066.0141X1C7
Receipt Date:	20-JUN-2019
Filing Date:	18-JAN-2019
Time Stamp:	12:23:24
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		20190620_Response_P2007-02-01-2-5_066.pdf	155745	yes	14
			01f32db2d15c1a7058b9c371d2fd1c928a63be90		

	Multipart Description/PDF files in .zip description		
	Document Description	Start	End
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Claims	2	9
	Applicant Arguments/Remarks Made in an Amendment	10	14
Warnings:			
Information:			
Total Files Size (in bytes):		155745	
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>			

AMENDMENTS TO THE CLAIMS

Please amend the claims without prejudice or disclaimer to read as follows:

1 (Currently Amended). A system for adaptive-rate content streaming of live event video playable on one or more end user stations over the internet, the system comprising:

at least one storage device storing live event video, the live event video encoded at a plurality of different bitrates creating a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream, the low quality stream, the medium quality stream, and the high quality stream each comprising a group of streamlets encoded at a respective one of the plurality of different bitrates, each group of streamlets comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video;

wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and

wherein the first streamlet of each of the groups of streamlets has the same first duration and ~~corresponds to~~ encodes the same first portion of the live event video in each of the low quality stream, the medium quality stream, and the high quality stream, and wherein the first streamlet of the low quality stream ~~having~~ encodes the same first portion of the live event video at a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream.

2 (Original). The system of claim 1, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.

3 (Original). The system of claim 2, wherein the first and second durations are different.

4 (Original). The system of claim 1, further comprising:

a plurality of web servers located at different locations across the internet, each web server configured to:

receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing a portion of the video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams;

retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and

send the retrieved first streamlet from the currently selected one of the different copies to the requesting one of the end user stations over the one or more network connections.

5 (Original). The system of claim 1, wherein each of the first streamlets has a first duration that is the range of 0.1 to 5 seconds.

6 (Original). The system of claim 1, wherein the live event is a live sports event.

7 (Original). The system of claim 1, further comprising:

a first web server configured to:

receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the live event video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the live event video;

retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and

send the retrieved first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream to the requesting one of the end user stations over the one or more network connections.

8 (Original). The system of claim 7, wherein the first streamlets of the low quality stream, the medium quality stream, and the high quality stream are available before the live event is complete.

9 (Original). The system of claim 7, wherein the streamlets of the low quality stream, the medium quality stream, and the high quality stream of the live event are available on a 10 second delay.

10 (Original). The system of claim 7, wherein the streamlets from the low quality stream, the medium quality stream, and the high quality stream of the live event, when played back, appear live to a viewer.

11 (Original). The system of claim 7, wherein the first web server is further configured to: receive at least one virtual timeline request over the one or more internet connections from the one or more end user stations to retrieve a virtual timeline; and send the virtual timeline to the requesting one of the end user stations over the one or more network connections.

12 (Original). The system of claim 11, wherein the virtual timeline corresponds to the currently selected one of the low quality stream, the medium quality stream, and the high quality stream.

13 (Original). The system of claim 11, wherein the virtual timeline defines a playlist for a user to view.

14 (Original). The system of claim 11, wherein the virtual timeline comprises a file that is configured to define a playlist for a user to view.

15 (Original). The system of claim 11, wherein the virtual timeline comprises at least one quantum media extension (QMX).

16 (Currently Amended). An end user station to stream a live event video over a network from a server for playback of the video, the content player device comprising:

a processor;

a digital processing apparatus memory device comprising non-transitory machine-readable instructions that, when executed, cause the processor to:

establish one or more network connections between the end user station and the server, wherein the server is configured to access at least one of a plurality of groups of streamlets;

wherein the live event video is encoded at a plurality of different bitrates to create a plurality of streams including at least a low quality stream, a medium quality stream, and a high quality stream, each of the low quality stream, the medium quality stream, and the high quality stream comprising a group of streamlets encoded at the same respective one of the different bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video;

wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bit rate of no less than 600 kbps; and

wherein the first streamlets of each of the low quality stream, the medium quality stream and the high quality stream each has an equal playback duration and each of the first streamlets ~~corresponding to~~ encodes the same portion of the live event video at a different one of the different bitrates;

select a specific one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams;

place a streamlet request to the server over the one or more network connections for the first streamlet of the selected stream;

receive the requested first streamlet from the server via the one or more network connections; and
provide the received first streamlet for playback of the live event video.

17 (Original). The end user station of claim 16, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.

18 (Original). The end user station of claim 17, wherein the first and second durations are different.

19 (Original). The end user station of claim 16, wherein each of the first streamlets has a first duration that is the range of 0.1 to 5 seconds.

20 (Original). The end user station of claim 16, wherein the first streamlets of the low quality stream, the medium quality stream, and the high quality stream are available before the live event is complete.

21 (Original). The end user station of claim 16, wherein the streamlets of the low quality stream, the medium quality stream, and the high quality stream of the live event are available on a ten second delay.

22 (Original). The end user station of claim 16, wherein the streamlets from the low quality stream, the medium quality stream, and the high quality stream of the live event, when played back, appear live to a viewer.

23 (Original). The end user station of claim 16, wherein the end user station is further configured to:

request and receive a virtual timeline; and

wherein one or more streamlet requests are based on the at least one virtual timeline.

24 (Original). The end user station of claim 23, wherein the virtual timeline corresponds to the currently selected one of the low quality stream, the medium quality stream, and the high quality stream.

25 (Original). The end user station of claim 23, wherein the virtual timeline defines a playlist for a user to view.

26 (Currently Amended). A process executable by one or more servers to stream a live event video for playback by one or more end user stations, the process comprising:

storing, by the one or more servers, a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream, wherein the low quality stream, the medium quality stream, and the high quality stream each comprise a group of streamlets encoded at a respective one of [[the] a plurality of different bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video;

wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and

wherein the first streamlet of each of the groups of streamlets has the same first duration and ~~corresponds to~~ encodes the same first portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the first streamlet of the low quality stream having a different one of the different bitrates than the first streamlet of the high quality stream and the first streamlet of the medium quality stream;

receiving at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the live event video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium

quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the live event video;

retrieving from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and

sending the retrieved first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream to the requesting one of the end user stations over the one or more network connections.

27 (Original). The process of claim 26, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.

28 (Original). The process of claim 26, wherein the first and second durations are different.

29 (Original). The process of claim 26, wherein the first streamlets of the low quality stream, the medium quality stream, and the high quality stream are available before the live event is complete.

30 (Currently Amended). A process executable by a content player device to stream a live event video over a network from a server for playback of the video by the content player device, the process comprising:

establishing one or more network connections between the content player device and the server, wherein the server accesses a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream, wherein the low quality stream, the medium quality stream, and the high quality stream each comprise a group of streamlets encoded at a respective one of ~~[[the]]~~ a plurality of different bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video;

wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and

wherein the first streamlet of each of the groups of streamlets has the same first duration and ~~corresponds to~~ encodes the same first portion of the live event video in the low quality stream, the medium quality stream, and the high quality stream, the first streamlet of the low quality stream having a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream;

selecting, by the content player device, a currently selected one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the live event video;

placing a streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the live event video;

receiving the requested streamlet from the server via the one or more network connections;
and

rendering, by the content player device, the received streamlet for playback of the live event video.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial. No.:	16/252,188	Conf. No.:	3722
Examiner:	Chirag R. Patel	Art Unit:	2454
Inventor:	Brueck et al.	Filed:	January 18, 2019
Docket No.:	066.0141x1C7 (P2007-02-01-2-5)	Customer No.:	70560
Title:	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

RESPONSE AND AMENDMENT
FILED UNDER 37 C.F.R § 1.111

Mail Stop AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

In response to the Office Action mailed on March 22, 2019, please amend the above-identified application as set forth below and consider the remarks that follow.

Amendments to the Claims begin on page 2 of this paper.

Remarks begin on page 10 of this paper.

REMARKS

In response to the Office Action mailed on March 22, 2019, we hereby request entry of the foregoing amendments and reconsideration of the amended application under 37 C.F.R. § 1.111. No new matter has been added, and support for the foregoing amendments may be found in the originally-filed specification, drawings and claims.

If there are any issues remaining to be resolved, the undersigned would welcome a telephone call to 480.385.5069 to discuss a supplemental response or Examiner's amendment.

Telephonic Interview

We appreciate the courtesy extended by Examiner Patel in speaking with the undersigned via telephone on June 19, 2019. We have incorporated the Examiner's suggestions in the foregoing amendments, and we believe that the issues raised in the Office Action will be resolved after entry of this response. In particular, the Examiner requested that we clarify that *each set of streamlets is generated at a different bitrate* to advance prosecution over the art of record. Our current claims clearly recite this feature.

We also appreciate the Examiner's clarification of the double patenting rejections.

If any additional discussion would be beneficial, the undersigned would welcome a follow up call at 480.385.5069 or an email to bcarlson@lkglobal.com.

Double Patenting/Terminal Disclaimers

Claims 1-30 are provisionally rejected as a double patent of co-pending application Serial No. 16/262,356. Claims 1, 6-9, 11-14, 16-18 and 21 are rejected as a "double patent" of US Patent No. 9,571,551, which is a parent to this application, in combination with another reference. The Office Action indicates that the rejections can be overcome with terminal disclaimers. Without consenting to the rejections, appropriate disclaimers are being submitted to advance prosecution.

Although payment for the terminal disclaimers will be provided with that submittal, the Office is authorized to deduct any shortfall from our USPTO Deposit Account No. 50-2091. If there are any issues with the terminal disclaimers or their

associated payment, or if any additional disclaimers are required, the Examiner is requested to contact the undersigned at 480.385.5069 or bcarlson@lkglobal.com.

Prior Art Rejections

The Office Action rejects each of our claims under Section 103. Claims 1-3 and 5 are rejected by a Section 103 combination of US Patent No. 7,369,610 (“Xu”) with US Patent Publication No. 2005/0055425 (“Lango”) and US Patent Publication No. 2003/0103571 (“Meehan”). The remaining claims are rejected by various Section 103 combinations citing Xu, Lango and Meehan in combination with additional references. Claims 4, 7-8, 10-12, 15-20, 22-24 and 26-30, for example, are rejected by the Section 103 combination of Xu, Lango, Meehan and US Patent No. 6,496,980 (“Tillman”).

As we discussed in the telephonic interview, however, even if all of the references were combined with each other, the resulting combinations would still lack the full extent of our current claim language. In particular, the combination would still lack groups of streamlets that encode the same portions of a media program at different qualities. Claim 1, for example, recites that a first streamlet of each of the groups of streamlets has the same first duration and encodes the same first portion of the live event video in each of the low quality stream, the medium quality stream, and the high quality stream at different bitrates. Our other independent claims recite similar features.

The primary reference, Xu, clearly relates to an enhancement layer scheme wherein a media program is represented by a low bandwidth “base stream” that can be used in combination with one or more separate “enhancement layers” as bandwidth allows. See Xu at Title, Abstract and FIG. 3. The Office Action correctly acknowledges (e.g., at page 8) that Xu lacks our original claim language relating to streamlets, particularly with regard to streamlets from different quality groups having the same duration and corresponding to the same portion of the live video event.

The Office Action relies upon the secondary Lango reference to supply this feature, but Lango does not relate to any sort of “streamlets” at all, much less any streamlets that encode the same portions of a video at different quality. As we discussed in the telephonic interview, Lango (at best) relates to caching media data as it passes through a server, but there is no mention of the specific features that are recited

in our claims. There is certainly no mention (or suggestion) that *the first streamlet in each of the groups of streamlets has the same first duration and encodes the same first portion of the event video in the low, medium and high quality streams*, as recited in claim 1.

Largo in general relates to caching of RTP/RTSP media content for re-distribution of previously-requested content from a cache, but there is no mention of streamlets. In rejecting several of our dependent claims that provide more detail about the durations of streamlets (e.g., claims 3 and 5), the Office Action cites to paragraph 0072 of Largo, which describes a system that breaks a received RTP/RTSP stream into “a series of chunks” of “a predetermined duration”. This mention, however, does not teach the full extent of our claim language, which relates to streamlets from different quality groups having the same duration and each encoding the same portion of a video event.

The third reference, Meehan, is only cited to teach a bitrate in excess of 600 kbps. Meehan is not alleged to teach the “streamlet” features described above, and in fact it does not do so. Even if Meehan is included in the Section 103 combination with Xu and Largo, then, the result would still fail to teach (or fairly suggest) the full extent of our claim language, particularly with regard to streamlets.

The Tillman reference cited in combination with Xu, Largo and Meehan against many of our claims, similarly fails to teach the features of our claims for which it is cited. Tillman relates to a layered encoding scheme (much like Xu) in which data is transferred over a fixed bandwidth network to a client. See Tillman’s Abstract. Tillman makes momentary mention of a client requesting an enhancement layer from a server to augment a cached base layer, but there is certainly no mention of a client selecting a streamlet from amongst different quality groups that are available. Claim 16, for example, recites that the client “select[s] a specific one of the low quality stream, the medium quality stream, and the high quality stream”. This feature is well beyond the rudimentary client requests contemplated by Tillman.

Even if Xu, Largo, Meehan and Tillman were all combined with each other, then, the result would still fail to teach (or fairly suggest) the full extent of our current claim language. Please note that the various additional references cited in combination with these references are not alleged to supply the missing “streamlet” features of the claims,

and in fact they do not do so. Indeed, many of our dependent claims recite additional features that further distinguish from the cited art. Although a discussion of these features is cumulative, particularly in view of the telephonic interview, we do not consent to any of the rejections, and we expressly reserve the right to revisit any of the Section 103 rejections at a later date, including on appeal or in a continuing application.

At present, however, we simply request reconsideration of the Section 103 rejections.

Conclusion

In view of the foregoing, it is respectfully submitted that all of the issues identified in the Office Action have all been overcome. Accordingly, we respectfully submit that the application is in condition for allowance, and such allowance is now earnestly requested.

Should the Examiner have any suggestions that would assist in prosecution, Applicant's counsel would welcome a telephone call to (480) 385-5069 or an email to bcarlson@lkglobal.com. Recognizing that Internet communications are not secure, I hereby authorize the USPTO to communicate with the undersigned and other practitioners of record in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via video conferencing, instant messaging, or electronic mail. I understand that a copy of these communications will be made of record in the application file. (MPEP 502.03). We specifically consent to the use of email to exchange proposed amendments or other draft documents.

No official fee is believed to be due in connection with this Response, other than any fees that may be addressed elsewhere in this submittal. If, however, any time extension or additional fee is required to consider this response or to otherwise prevent abandonment of this application, please consider this as a request for an extension of time and as authorization to charge Deposit Account No. 50-2091 for any fees that may be due.

Respectfully submitted,
LORENZ & KOPF, LLC

Dated: June 20, 2019

By: /BRETT A. CARLSON/
Brett A. Carlson
Reg. No. 39928

*Lorenz and Kopf, LLC
Customer No. 70560
480-385-5060*

Doc Code: DISQ.E.FILE

Document Description: Electronic Terminal Disclaimer – Approved

Application No.: 16252188

Filing Date: 18-Jan-2019

Applicant/Patent under Reexamination: BRUECK

Electronic Terminal Disclaimer filed on June 24, 2019

☒ APPROVED

This patent is subject to a terminal disclaimer

☐ DISAPPROVED

Approved/Disapproved by: Electronic Terminal Disclaimer automatically approved by EFS-Web

U.S. Patent and Trademark Office

Electronic Acknowledgement Receipt

EFS ID:	36387834
Application Number:	16252188
International Application Number:	
Confirmation Number:	3722
Title of Invention:	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING
First Named Inventor/Applicant Name:	DAVID F. BRUECK
Customer Number:	70560
Filer:	Brett Carlson/Julie Graf
Filer Authorized By:	Brett Carlson
Attorney Docket Number:	066.0141X1C7
Receipt Date:	24-JUN-2019
Filing Date:	18-JAN-2019
Time Stamp:	13:47:03
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 160
RAM confirmation Number	062519INTEFSW13470000
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Terminal Disclaimer-Filed (Electronic)	eTerminal-Disclaimer.pdf	33553	no	2
			7418dbe3152812cb0ad8cca79d07cc560e2e4d44		
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30530	no	2
			7c24e8de3b394e7f69a8b11b2d0c43597d0f1377		
Warnings:					
Information:					
Total Files Size (in bytes):			64083		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

Electronic Patent Application Fee Transmittal

Application Number:	16252188			
Filing Date:	18-Jan-2019			
Title of Invention:	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING			
First Named Inventor/Applicant Name:	DAVID F. BRUECK			
Filer:	Brett Carlson/Julie Graf			
Attorney Docket Number:	066.0141X1C7			
Filed as Large Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
STATUTORY OR TERMINAL DISCLAIMER	1814	1	160	160
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				160

Doc Code: DIST.E.FILE Document Description: Electronic Terminal Disclaimer - Filed		PTO/SB/26 U.S. Patent and Trademark Office Department of Commerce
Electronic Petition Request	TERMINAL DISCLAIMER TO OBTAIN A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT	
Application Number	16252188	
Filing Date	18-Jan-2019	
First Named Inventor	DAVID BRUECK	
Attorney Docket Number	066.0141X1C7	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING	
☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action ☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.		
Owner	Percent Interest	
DISH Technologies L.L.C.	100%	
<p>The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)</p> <p>9571551</p> <p>as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.</p> <p>In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:</p> <ul style="list-style-type: none"> - expires for failure to pay a maintenance fee; - is held unenforceable; - is found invalid by a court of competent jurisdiction; - is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321; - has all claims canceled by a reexamination certificate; - is reissued; or - is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer. <p>☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.</p>		

☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.	
Applicant claims the following fee status:	
☐ Small Entity ☐ Micro Entity ☒ Regular Undiscounted	
I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.	
THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES	
I certify, in accordance with 37 CFR 1.4(d)(4) that I am:	
☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application Registration Number <u>39928</u>	
☐ A sole inventor	
☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application	
☐ A joint inventor; all of whom are signing this request	
Signature	/BRETT A. CARLSON/
Name	/BRETT A. CARLSON/

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
 Form PTO/SB/96 may be used for making this certification. See MPEP § 324.



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.
16/252,188	01/18/2019	DAVID F. BRUECK	066.0141X1C7	3722
70560	7590	06/26/2019		
LKGLOBAL (Dish Technologies L.L.C.)			EXAMINER	
7010 E. COCHISE ROAD			PATEL, CHIRAG R	
SCOTTSDALE, AZ 85253				
			ART UNIT	PAPER NUMBER
			2454	
			NOTIFICATION DATE	DELIVERY MODE
			06/26/2019	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

dish@lkglobal.com
ipdept@dish.com

CONFIDENTIAL AND PRIVILEGED – WORK PRODUCT

Serial. No.:	16/252,188	Conf. No.:	3722
Examiner:	Chirag R. Patel	Art Unit:	2454
Inventor:	Brueck et al.	Filed:	January 18, 2019
Docket No.:	066.0141x1C7 (P2007-02-01-2-5)	Customer No.:	70560
Title:	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		

Proposed Amendment – DRAFT For Discussion Only

1 (Currently Amended). A system for adaptive-rate content streaming of live event video playable on one or more end user stations over the internet, the system comprising:

at least one storage device storing live event video, the live event video encoded at a plurality of bitrates creating a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream, the low quality stream, the medium quality stream, and the high quality stream each comprising a group of streamlets encoded at a respective one of the plurality of bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the live event video;

wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and

wherein the first streamlet of each of the groups of streamlets has the same first duration and ~~corresponds to~~ encodes the same first portion of the live event video in each of the low quality stream, the medium quality stream, and the high quality stream, and wherein the first streamlet of the low quality stream having encodes the same first portion of the live event video at a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream.

<i>Applicant-Initiated Interview Summary</i>	Application No. 16/252,188	Applicant(s) BRUECK et al.		
	Examiner CHIRAG R PATEL	Art Unit 2454	AIA (First Inventor to File) Status No	Page 1 of 2

All participants (applicant, applicants representative, PTO personnel):

1. CHIRAG R PATEL (Primary Examiner); Telephonic
2. Brett Carlson (Attorney); Telephonic

Date of Interview: 20 June 2019

Claims Discussed: 1

Prior Art Discussed: Xu, Lango

Amendment proposed: proposed amendments to independent claims which relate to the concepts of encoding the first portion of stream into plurality of streamlets. An agreement with respect to claim language was not reached. Examiner will review references and/or update search.

Issues Discussed:

Proposed Amendments:
discussed potential amendment to the independent claims.

Attachment(s): Proposed Amendments,

/CHIRAG R PATEL/ Primary Examiner, Art Unit 2454	
---	--

Applicant is reminded that a complete written statement as to the substance of the interview must be made of record in the application file. It is the applicants responsibility to provide the written statement, unless the interview was initiated by the Examiner and the Examiner has indicated that a written summary will be provided. See MPEP 713.04

Please further see:

MPEP 713.04

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews, paragraph (b)

37 CFR § 1.2 Business to be transacted in writing

U.S. Patent and Trademark Office
PTOL-413/413b (Rev. 01/01/2015)

Interview Summary

Paper No. 20190620

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	16252188
Filing Date	2019-01-18
First Named Inventor	David F. Brueck
Art Unit	2454
Examiner Name	Chirag R. Patel
Attorney Docket Number	066.0141X1C7

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	8521836	B2	2013-08-27	Kewalramani et al.	
	2	7558869	B2	2009-07-07	Leon et al.	
	3	7707303	B2	2010-04-27	Albers et al.	
	4	7925781	B1	2011-04-12	Chan et al.	
	5	8880721	B2	2014-11-04	Hurst et al.	
	6	9344496	B2	2016-05-17	Hurst et al.	
	7	8683066	B2	2014-03-25	Hurst et al.	
	8	8686066	B2	2014-04-01	Kwampian et al.	

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16252188
Filing Date	2019-01-18
First Named Inventor	David F. Brueck
Art Unit	2454
Examiner Name	Chirag R. Patel
Attorney Docket Number	066.0141X1C7

9	6708213	B1	2004-03-16	Bommaiah et al.
10	6757796	B1	2004-06-29	Hofmann
11	9462074	B2	2016-10-04	Guo 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	20050033855	A1	2005-02-10	Moradi et al.	
	2	20030093790	A1	2003-05-15	Logan et al.	
	3	20030195977	A1	2003-10-16	Liu et al.	
	4	20040199655	A1	2004-10-07	Davies et al.	
	5	20040267956	A1	2004-12-30	Leon et al.	
	6	20040220926	A1	2004-11-04	Lamkin et al.	

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16252188
Filing Date	2019-01-18
First Named Inventor	David F. Brueck
Art Unit	2454
Examiner Name	Chirag R. Patel
Attorney Docket Number	066.0141X1C7

7	20060010003	A1	2006-01-12	Kruse	
8	20060277564	A1	2006-12-07	Jarman	
9	20150058496	A1	2015-02-26	Hurst et al.	
10	20100262711	A1	2010-10-14	Bouazizi	
11	20050251832	A1	2005-11-10	Chiueh	
12	20040221088	A1	2004-11-04	Lisitsa et al.	
13	20040179032	A1	2004-09-16	Huang	
14	20020161908	A1	2002-10-31	Benitez et al.	
15	20030021282	A1	2003-01-30	Hospodor	
16	20080120330	A1	2008-05-22	Reed et al.	
17	20080120342	A1	2008-05-22	Reed et al.	

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16252188
Filing Date	2019-01-18
First Named Inventor	David F. Brueck
Art Unit	2454
Examiner Name	Chirag R. Patel
Attorney Docket Number	066.0141X1C7

18	20110307545	A1	2011-12-15	BOUAZIZI
19	20050262257	A1	2005-11-24	Major et al.
20	20030236904	A1	2003-12-25	Walpole 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							

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	Krasic et al., QUALITY-ADAPTIVE MEDIA STREAMING BY PRIORITY DROP, Oregon Graduate Institute, 2001.	×
	2	Krasic et al., QoS SCALABILITY FOR STREAMED MEDIA DELIVERY, Oregon Graduate Institute School of Science & Engineering Technical Report CSE 99-011, September 1999.	☒
	3	Huang et al., ADAPTIVE LIVE VIDEO STREAMING BY PRIORITY DROP, Portland State University PDXScholar, July 21, 2003.	☒

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16252188
Filing Date	2019-01-18
First Named Inventor	David F. Brueck
Art Unit	2454
Examiner Name	Chirag R. Patel
Attorney Docket Number	066.0141X1C7

4	Walpole et al, A PLAYER FOR ADAPCTIVE MPEG VIDEO STREAMING OVER THE INTERNET, Oregon Graduate Institute of Science and Technology, October 25, 2012.	☒
5	ALBANESE, ANDREW ET AL. "Priority Encoding Transmission", TR-94-039, August 1994, 36 pgs, International Computer Science Institute, Berkeley, CA.	☒
6	BIRNEY, BILL "Intelligent Streaming", May 2003, Microsoft.	☒
7	GOYAL, VIVEK K. "Multiple Description Coding: Compression Meets the Network," September 2001, pgs. 74-93, IEEE Signal Processing Magazine.	☒
8	ON2 TECHNOLOGIES, INC. "TrueMotion VP7 Video Codec" White Paper, Document Version 1.0, January 10, 2005.	☒
9	PATHAN, AL-MUKADDIM ET AL. "A Taxonomy and Survey of Content Delivery Networks" Australia, February 2007, available at http://www.gridbus.org/reports/CDN-Taxonomy.pdf .	☒
10	PURI, ROHIT ET AL. "Multiple Description Source Coding Using Forward Error Correction Codes," October 1999, 5 pgs., Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA.	☒
11	WICKER, STEPHEN B. "Error Control Systems for Digital Communication and Storage," Prentice-Hall, Inc., New Jersey, USA, 1995, parts 1-6.	☒
12	LIU, JIANGCHUAN ET AL. "Opportunities and Challenged of Peer-to-Peer Internet Video Broadcast," School of Computing Science, Simon Fraser University, British Columbia, Canada.	☒
13	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 .	☒
14	MOVE NETWORKS, INC. "The Next Generation Video Publishing System," April 11, 2007; http://www.movenetworks.com/wp-content/uploads/move-networks-publishing-system.pdf .	☒

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16252188
Filing Date	2019-01-18
First Named Inventor	David F. Brueck
Art Unit	2454
Examiner Name	Chirag R. Patel
Attorney Docket Number	066.0141X1C7

15	YOSHIMURA, TAKESHI ET AL. "Mobile Streaming Media CDN Enabled by Dynamic SMIL", NTT DoCoMo, Multimedia Laboratories and Hewlett-Packard Laboratories, dated May 7-11, 2002, ACM 1-58113-449-5/02/0005; http://www2002.org/CDROM/refereed/515/ .	☒
16	Nguyen, T. et al., Multiple Sender Distributed Video Streaming, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 6, no. 2, April 1, 2004, pages 315-326, XP011109142, ISSN: 1520-9210, DOI: 10.1109/TMM.2003.822790.	☒

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	16252188
Filing Date	2019-01-18
First Named Inventor	David F. Brueck
Art Unit	2454
Examiner Name	Chirag R. Patel
Attorney Docket Number	066.0141X1C7

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.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

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)	2019-06-27
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 records.
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.

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. 70560. 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.

Electronic Acknowledgement Receipt

EFS ID:	36433290
Application Number:	16252188
International Application Number:	
Confirmation Number:	3722
Title of Invention:	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING
First Named Inventor/Applicant Name:	DAVID F. BRUECK
Customer Number:	70560
Filer:	Brett Carlson/Julie Graf
Filer Authorized By:	Brett Carlson
Attorney Docket Number:	066.0141X1C7
Receipt Date:	27-JUN-2019
Filing Date:	18-JAN-2019
Time Stamp:	15:11:13
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Form (SB08)	20190627_IDS-2_066-0141X1C7.pdf	531141 b91ad259f3be6c33466f3c3abbccd18458b b9b07	no	8

Warnings:

Information:					
2	Transmittal Letter	20190627_IDSCert_066-0141X1C7.pdf	227439	no	1
			efab76b4c7fe4a8492ded8b6ad9b2fa7442d2629		
Warnings:					
Information:					
3	Non Patent Literature	Huang_AdaptiveLiveVideoStreaming.pdf	2524324	no	11
			d11b5abc7e3870ffc0fd6176396674bdbbcd8c7a		
Warnings:					
Information:					
4	Non Patent Literature	Krasic_QoSScalabilityForStreamedMediaDelivery.pdf	232474	no	10
			63f916fa208fe954d7b15a25ec0c3c3ec7c3704b		
Warnings:					
Information:					
5	Non Patent Literature	Krasic_Quality-AdaptiveMedia.pdf	288035	no	10
			2c5a1fb17fd9783462bcea4b14497ec76424909b		
Warnings:					
Information:					
6	Non Patent Literature	NPL_Albanese_PriorityEncoding.pdf	1436896	no	36
			40ea8d1dc311303a58036aa6803271aebec3986		
Warnings:					
Information:					
7	Non Patent Literature	NPL_Birney_IntelligentStreaming.pdf	631194	no	7
			0d23a6e65ccfb6f0463367759c3f890f0a55cd6cc		
Warnings:					
Information:					
8	Non Patent Literature	NPL_Clement_MoveClosesFunding.pdf	172934	no	2
			c65a317075d6d88f73613f212b13b4e09646ea79		
Warnings:					
Information:					

9	Non Patent Literature	NPL_Goyal_CodingCompression.pdf	4353461	no	20
			78149b1c9bf070e64d21f2a37ba8f1d66dbd45ff		
Warnings:					
Information:					
10	Non Patent Literature	NPL_Liu_PeerToPeerInternetVideo.pdf	1125709	no	12
			1dbfd7983420ecb8f68943b0539a4d344045f5fb		
Warnings:					
Information:					
11	Non Patent Literature	NPL_MoveNetworks_NextGeneration.pdf	1128862	no	21
			446af274461f93a7d60548a003fcc17a6c5e4a0f		
Warnings:					
Information:					
12	Non Patent Literature	NPL_Nguyen_MultipleSenderDistVideo.pdf	1053500	no	12
			af974412420b4c476c2c59e0e76443e17141b1bb		
Warnings:					
Information:					
13	Non Patent Literature	NPL_On2Technologies.pdf	887186	no	13
			cc8df69d3dd6cc62a111dc6d213c583e0683bc58		
Warnings:					
Information:					
14	Non Patent Literature	NPL_Pathan_ContentDelivery.pdf	3819974	no	44
			bd517be17fd62cd8c811c76a32b0a45da2230af0		
Warnings:					
Information:					
15	Non Patent Literature	NPL_Puri_SourceCoding.pdf	395947	no	5
			bdd0ea033a2ab94cb2477b4e4b363bde71280a6e		
Warnings:					
Information:					

16	Non Patent Literature	NPL_Wicker_ErrorControlSys_Part1.pdf	4017078	no	85
			05a44138df970b54e4654a8e826e77ee3111a734		
Warnings:					
Information:					
17	Non Patent Literature	NPL_Wicker_Part2.pdf	4003752	no	81
			9af8bc8ca3ee2444a65208263b785ffdf6f0df42		
Warnings:					
Information:					
18	Non Patent Literature	NPL_Wicker_Part3.pdf	5167479	no	89
			ca519c14ff8af13af6d790858696235a433b2fcc		
Warnings:					
Information:					
19	Non Patent Literature	NPL_Wicker_Part4.pdf	8345160	no	95
			4e2a64c21ea5974a4a96928ea79904151b31b1a3		
Warnings:					
Information:					
20	Non Patent Literature	NPL_Wicker_Part5.pdf	16458768	no	91
			9c07a14f8208b38c97d27b7d16e139b240d12c4f		
Warnings:					
Information:					
21	Non Patent Literature	NPL_Wicker_Part6.pdf	12015062	no	68
			7d72bb382aa4c68c0f1ed8169dadd6689aaf1fe2		
Warnings:					
Information:					
22	Non Patent Literature	NPL_Yoshimura_MobileStreamingMedia.pdf	378918	no	20
			aac54a2603ed446586676ed16c15022af1acc1eb		
Warnings:					
Information:					

23	Non Patent Literature	Walpole_APlayerForAdaptiveMPEG.pdf	1119882	no	12
			ddb4cc0ec5db2c359797f0d721cff194e6c1b91d		
Warnings:					
Information:					
Total Files Size (in bytes):				70315175	
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					



(11) **EP 0 919 952 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
02.06.1999 Bulletin 1999/22

(51) Int. Cl.⁶: **G06T 9/00**

(21) Application number: **98109423.8**

(22) Date of filing: **25.05.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
 Designated Extension States:
AL LT LV MK RO SI

• **Brunetton, Sylvain**
1009 Pully (CH)

(74) Representative:
Ravenel, Thierry Gérard Louis et al
I C B,
Ingénieurs Conseils en Brevets SA,
7, rue des Sors
2074 Marin (CH)

(30) Priority: **28.11.1997 US 66856 P**

(71) Applicant:
ECOLE POLYTECHNIQUE FEDERALE DE
LAUSANNE
1015 Lausanne (CH)

Remarks:

A request for correction of the claims numbering has been filed pursuant to Rule 88 EPC. A decision on the request will be taken during the proceedings before the Examining Division (Guidelines for Examination in the EPO, A-V, 3.).

(54) **Method for coding/decoding of a digital signal**

(57) The invention concerns a method for primary processing a digital signal comprising the steps of :

- providing a digital signal to a primary processing unit ;
- primary processing said signal with said primary processing unit according to a plurality of primary processing algorithms to provide primary processed output signals,
- determining statistics of use of at least one of said primary processing algorithms,
- providing a digital statistics signal representative of said statistics for each of said primary processed output signals
- associating to each of said primary processed output signals its own statistics signal ;
- determining a complexity primary processing information signal based on said statistic signal.

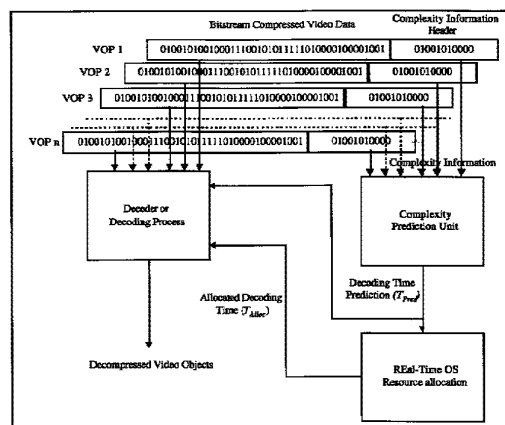


Figure 12

EP 0 919 952 A1

Description**DEFINITION OF THE INVENTION**

[0001] The present invention is a processing apparatus which aims at controlling and guaranteeing the real-time performance of data processing. The invention is based on the principle of collecting statistics of the algorithms used to process the data, on transmitting such statistics when necessary, and of associating a complexity information to such statistics. The real-time processing control can be done generating a processing time prediction signal which is a function of the complexity information, and use such prediction signal to provide processing resources which guarantee real-time processing or enables the definition of an associated degraded processing that guarantees real-time processing. Although the described invention is of general application, this document describes its application to video coding and decoding and suggests the straightforward extensions to audio coding-decoding, and to composition and rendering of media objects.

1. FIELD OF THE INVENTION

[0002] The implementation of software based video/audio decoders is an advantageous solution over traditional dedicated hardware systems. The main reason of this fact is the possibility of decoding several different video/audio compression standards, at different levels and profiles, by just changing the software while using the same hardware platform, main memory, and interfaces. Another favorable feature is the flexibility of the decoder able to cope with new developments and new tools included into future versions of the standards by just updating the decoding software. However, simultaneously guaranteeing real-time performance, needed when processing video/audio bit-streams, and an efficient use of the processing power, remains a very difficult task.

[0003] The present invention reports the main reasons of such difficulties and presents a new technique able to predict the decoding time of compressed video/audio bit-streams without the need of the actual decoding. This can be obtained if appropriate statistical information about the actual encoding processing is included and transmitted with the compressed video/audio data. Moreover, the present invention reports how such results can be used to implement new efficient processing resource allocation strategies. New possible schemes of such intelligent interactions between the encoder and the real-time OS are also proposed. An example of implementation for the MPEG-4 VM6 video algorithm based on these developments is presented. This implementation enables, at the same time, to guarantee real time performances even when the available processing resources are lower than the one theoretically necessary, and it yields optimal (minimal) image quality degradation. Such results are obtained by using the prediction of the decoding-resource needs to implement Computational Graceful Degradation techniques. The invention only reports simulation results for the MPEG-4 VM6.1 video compression algorithm, but the techniques and concepts presented here are of general application. Indeed, these techniques can be easily applied to audio coding, synthetic audio reproduction and to any future or existing compression and composition method. More precisely, they can be used for any video compression standard that is capable of transmitting in an appropriate form the statistical coding information (i.e. MPEG-1 and MPEG-2 using an appropriate transport system), and they can be implemented in any processor-based platform.

[0004] The present description is organized as follows. Section 2 discusses the problems and utility of defining and using worst case and average case complexities, justifying the need of complexity prediction. Section 3 presents the modeling of decoding complexity and the general principles of the prediction techniques. Section 4 presents the background of the invention. Section 5 summarizes the main concepts of the invention while section 6 summarizes the content of the diagrams and drawings illustrating the details and main characteristics of the invention.

[0005] In section 7, experimental results of the MPEG-4 VM6 texture coding algorithm are reported. Section 8 presents the definition of the proposed intelligent interactions between real-time OS-schedulers and video decoders. Section 9 presents the basic principles and the definition of the developed Computational Graceful Degradation (CGD) techniques, and section 10 reports overall experimental results of the control of decoding complexity using complexity prediction joint with the application of CDG techniques. Finally, section 11 concludes the description of the invention, summarizing the achievements of the invention by declaring the corresponding claims.

2. WORST-CASE AND AVERAGE-CASE DECODING COMPLEXITY

[0006] A simple solution to the problem of the variability of decoding complexity is to implement software/hardware solutions that always guarantee worst case complexity decoding. Such solutions are, in fact, the ones implicitly implemented in most of the dedicated hardware decoders. Dedicated hardware platforms, in general, are designed to guarantee worst case complexity decoding. The various hardware modules implementing the different decoding procedures are synchronized with the input/output data rate. These dedicated hardware modules can always handle any input bit-stream, not exceeding a maximum rate, and containing compressed sequences of appropriate maximum image reso-

lutions. This can be always guaranteed, irrespectively from the coding options selected to produce the compressed bit-stream.

[0007] In contrast, in the case of programmable platforms it is necessary to define what is a worst case decoding complexity. Let us consider video coding even if the same concept can be extended to natural and synthetic audio and to other form of multimedia processing. In the video compressing field one can assume that the worst case decoding complexity is represented by the decoding of each bit/macroblock, using the coding options allowed by the considered standard that requires the highest decoding time. The coding algorithms/modes of H.263 can be approximately considered as a superset of the algorithms/modes of the various video compression standards (H.261, MPEG-1 and MPEG-2). The complete results and details of this worst-case/average-case study are omitted here for brevity. The conclusions resulting from this analysis can be summarized as follows :

1) Decoding functions can be divided into two classes. For one class a worst case can be easily defined. Let us denote this class as A. To this class belong function such as IDCT, inverse quantisation, motion compensated prediction in its various form (forward, backward, bi-directional, overlapped ...). For the other class denoted as class B, complexity is extremely data dependent and cannot be easily characterized by the simple parameters of an hypothetical MPEG-2 like complexity profile. To this class belong operations such as VLD, VLC coding, Syntax-based Arithmetic Coding (SAC) and in general all parsing operations. The difficulty of defining a worst case is evident considering for instance SAC for which the difference between a theoretical worst-case and experimental cases is about two orders of magnitude.

2) A strict worst case complexity analysis, of both class A and B functions, leads to unrealistic results. In other words, the computational resources necessary to handle the strict worst-case complexity decoding is much higher than the resources used in average.

3) For class A functions, the actual decoding complexity depends on the actual coding options and reports a large range of variability.

4) For class B functions, the actual decoding complexity reports a dependence on coding options, but presents a reasonably narrow range of variability which is much lower than the range theoretical worst case-average case.

[0008] Worst case analysis, therefore, is not adequate to define a useful decoding complexity reference. This fact is true not only for operations which are strongly data dependent (class B), but in general for all functions even those that have a very well defined worst case such as all class A functions. From these results it is clear that guaranteeing a worst case decoding implies guaranteeing the decoding of pathologically complex sequences that in practice have an extremely low probability of occurrence. Moreover, such a guarantee would imply the allocation of a processing power up to two orders of magnitude higher than the one actually needed for the decoding of the standard test sequences. The processor in practice would work in normal conditions exploiting only a small fraction of the theoretically available processing power. Such solution to the problem of guaranteeing real-time performance is not economical and, obviously, not useful in practical applications.

[0009] Another possibility for the solution of the problem of guaranteeing real-time performance could be to consider as reference an average case decoding complexity. Before being able to successfully use such average complexity, a few questions need to be answered. Is it meaningful to define an average complexity? How can it be done? Is the average complexity sequence dependent? How can it be converted on different platforms? Can the average complexity accurately be described by parameters such as image resolution and bit-rate? Are average results characterized by sufficiently narrow variance bounds so as they can be used a priori without the need of knowing some information about the actual complexity of the incoming sequence?

[0010] One will here briefly summarize the main results and experiments that answer the mentioned questions. So as to try to define an average case for the complexity that have been considered in previously, the H.263 standard uses four typical test sequences over a wide bit-rate range from 16 kbit/sec up to 1024 kbit/sec. The sequences in CIF format are : Akiyo (300 frames), Coastguard (300 frames), Foreman (300 frames) and Mobile & Calendar (300 frames), and all these test sequences have been evaluated at fifteen different bit-rate values. Although a rigorous statistical analysis would have required a much higher number of sequences and specific statistical confidence tests, it is believed that the number of frames (1200) and the different contents of the test images are well representative of typical image sequence contents. One have considered separately the operations that have unrealistic worst case complexities (class B) from the one with a well-defined worst case behavior (class A). The experiments, for the class B operations, yield figures that are not strongly bit-rate dependent. Variance bounds for each frame are relatively narrow and these values do not report relevant variations depending on the test sequence. Such results confirm also the fact that it is not necessary to worry about strict worst case complexity. In the other case of class A, the complexity of the operations reports a relatively regular statistical behavior at very low bit-rates (< 16 kb/s), but complexity reports dependence on the bit-rate, on the image content and in particular on the used encoding options.

[0011] Figure 1 reports one example of H.263 complexity using different coding options for the same sequence (Akiyo

CIF) and same bit-rate. On this figure, the decoding time or decoding complexity (expressed in μsec) is given versus the frame number for the sequence Akyio CIF encoded with H.263 at 128 Kb/sec using different coding options. Top curve options : advanced prediction modes (APM), syntax-based arithmetic coding and unrestricted motion vectors (UMV) ; middle curve options : APM and UMV ; bottom curve options : UMV. The same sequence presents, at the same bit-rate, average-decoding complexities that range up to a factor 4 (from about 0.05 to 0.2 sec per frame). Local fluctuations for a relatively static sequence such as Akyio are also considerably large. The decoder is the Telenor H.263 software release running on a Sun Ultrasparc platform.

[0012] Figure 2 reports the same experiment for the sequence Coastguard. The range of the complexity figures considering both average and local values is very large as can be easily noticed. On this figure, decoding time or decoding complexity (expressed in msec) is given versus frame number for the sequence Coastguard encoded with H.263 at 128 Kb/sec using different coding options. Top curve : advanced prediction modes (APM), syntactic arithmetic coding and unrestricted motion vectors (UMV) ; middle curve APM and UMV ; bottom curve : UMV. The same sequence at the same bit-rate has average decoding complexities that range within a factor 4 from about 0.05 to 0.2 sec per frame. The decoder is the Telenor H.263 software release on a Sun Ultrasparc platform.

[0013] Although far from the theoretical worst case complexities, the variation range is less than one order of magnitude, experimental results reports behaviors for which it is difficult to be able to predict the average complexity values only relying on resolution and bit-rate. Image content and coding options (i.e. intra or inter prediction modes, advanced prediction modes, half pixel vector prediction and so on) play a fundamental role. As it will be reported in the next sections, the situation of the complexity variations for MPEG-4 sequences is much higher due to the presence of different sized VOPs arbitrarily shaped objects, of static and dynamic sprites, and the large variety of coding modes.

[0014] From a pure mathematical point of view an average complexity value can always be defined considering a large data-base of encoded sequences and using the more probable occurrence of encoding options. Conversely, such a theoretical average value is not useful at all for the aim of the present invention that is the efficient scheduling of each decoding task and the efficient use of the available processing power. Even the local average values of the actual sequence often are not stationary, and their variations around the mathematical average one are so unpredictable that no useful scheduling policy can be realized.

[0015] Therefore, the consequences of this study are that straightforward solutions to the problem of real-time resource allocation policy are of two kinds. The first is a sustained over-sizing of the processing resources. The second is to accept interruptions of the audio/video output stream when the needed decoding resources exceed the ones available on the system. Obviously, both are not desirable being respectively not efficient and economical, or they provide a quality of the services which may easily results perceptually unacceptable.

[0016] The ideal solution would be to develop techniques that are able to accurately predict such unknown behaviors and, therefore, make feasible resource allocation policies that can guarantee real-time and minimize QoS degradation.

[0017] The description of such techniques and the presentation of the results are the subjects of the next sections.

[0018] As discussed above, the key issue for the implementation of any real-time application on a software-based platform is to be able to understand when the required decoding processing time exceeds the available processing time without the need of performing the actual decoding. This is particularly important, for instance, in MPEG-4 systems in which a number of bit-streams containing a variable number of VOPs and sprites need to be decoded respecting real-time constraints, but it is also important for existing compression standards (MPEG-1, MPEG-2). The available processing time is known, in general, because it is allocated by the OS-scheduler to each decoding process. The problem is how to estimate the necessary decoding processing requirements without the need of an actual decoding of each of the incoming VOP bit-streams. Decoding each bit-stream using a priori allocated processing could result in exceeding the time slot and, therefore, in missing real-time requirements. Real-time requirements are generally defined implicitly by the frame/sampling rate of video audio bit-streams or can be easily extracted by an appropriate relative time information associated to each video frame or audio object (also denoted as time stamp). Such extraction of the real-time constraints is trivial and is not discussed here. The consequence of missing real-time constraints is the introduction of sudden interruptions on the video content. A complete analysis of the conditions which lead to miss real time-constraints for the various OS scheduling algorithm (pre-emptive multi-threading, etc.) is out of the scope of this document and it is not discussed here. The reader can refer to the included reference S. Battista, F. Gasperoni, B. Klaiber, M. Mattavelli, L. Pautet, S. Tardieu, Concurrent over Sequential Modeling for MSDL, ISO/IEC JTC1/SC29/WG11, Doc. M1525, Maceio, Brazil, November 1996 incorporated herein by reference.

3. BACKGROUND OF THE INVENTION

[0019] Without any doubt, the recent trend in multimedia system implementation technology is towards moving from dedicated real-time hardware to software/processor based platforms. The processing power of the last on the market processors, and of the newly announced video/audio-processor families, enables the software implementation of video/audio decoding of various compression standards and different resolution levels. Unfortunately, simply relying on

the average processing power does not guarantee to respect real-time constraints of video/audio decoding ; furthermore it does not solve the problem of an efficient processing resource allocation. The fundamental reason of these facts is that the complexity of the video/audio decoding process is a fluctuating function of the time, as described in M. Mattavelli and S. Brunetton, A statistical study of MPEG-4 VM texture decoding complexity, ISO/IEC JTC1/SC29/WG11, Doc. M924, Tampere Finland, July 1996 incorporated herein by reference.

[0020] The term complexity here simply means the time needed by the decoding process. Other more abstract measures of complexity such as the amount and type of executed algorithmic elementary operations, the executed processor cycles, the amount of memory transfers to and from the external frame memories, and their combinations could be used as more precise and specific complexity measures. In the context of the present invention however, one is interested on the final results of these complexity factors. Without any loss of generality, one can consider, for any specific target software-hardware implementation, the actual processor decoding time. Indeed all the above mentioned measures of complexity can be converted for each specific platform into a single complexity measure given by the execution time.

[0021] Even by considering a specific processor platform or a specific compression standard with a relative level and profile, the complexity of video decoding is a variable and unpredictable quantity, as described in ISO/IEC MPEG2 incorporated herein by reference. The actual figures of decoding complexity depend in a complex way on many different factors. One can mention, for instance, the actual bit-rate, the image content, the options used for encoding, the Group Of Picture (GOP) structure, or the Group Of Video Object Planes (noted as VOP) structure (GOV) which can change on the fly, the state of the buffer that controls the quantization parameters and many others, as described in M. Mattavelli and S. Brunetton, Scalability of MPEG-4 VM based on the computational power of texture decoding, ISO/IEC JTC1/SC29/WG11, Doc. M925, Tampere, Finland, July 1996 incorporated herein by reference.

[0022] The complexity variability range increases with the richness of the standard. Richness means the number of implemented coding algorithms or coding modes and the number of possible coding options having different complexities. For instance, the richness of H.263 is much higher than H.261, but it is lower than the new MPEG-4 standard under development where multiple Video Object Planes (VOPs) of different sizes and characteristics need to be decoded.

[0023] In this description, Video Object (VO) means a temporal sequence of two-dimensional or three-dimensional arbitrarily shaped Video Object Planes (VOP). In case of rectangular or square shapes, these video object planes, also called hereinafter video planes, are denoted by the classical name of frames. An audio object means a stream of packets containing digital audio samples, or audio events in mono or multichannel format which can be associated to a VO. One or more audio packets can be associated to one VOP or video frame. By three-dimensional arbitrarily shaped video object planes we mean the projection or view of any three-dimensional natural or synthetic 3-dimensional object or 3-dimensional model.

In summary, the processor decoding time of each incoming frame or VOP for any given compression standard is highly variable and impossible to predict just relying on resolution and bit-rate information. Therefore, an efficient resource allocation policy (process scheduling) which aims at guaranteeing real-time performances is very difficult in these conditions. Scheduling policies that release the thread to the decoding process and receive it back after the task is completed are clearly not adequate. The same conclusion is true for a priori allocated processing time intervals. If more processing time than the one actually allocated is needed, real-time constraints are already violated by the video application. The implementation of emergency measures such as skipping frames or portions of images, on one hand might succeed in recovering the correct timing, on the other could have catastrophic effects on the Quality of Services (QoS) of the video application.

4. SUMMARY OF THE INVENTION

[0024] It is an object of the present invention to control and to guarantee real-time performance of data processing. It is another object of the present invention to provide a processing apparatus that associates a complexity information to the processed data.

[0025] It is another object of the present invention to provide a coding-decoding apparatus that further includes the associated complexity information to be transmitted to the decoder.

[0026] It is another object of the present invention to provide a system that, on the basis of said complexity information, provides a complexity prediction signal to be used to allocate processing resources to optimize real-time data processing.

[0027] It is another object of the present invention to provide a processing system that supervises the processing using an allocated resource signal, a predicted complexity signal, a complexity information signal and provide a processing that satisfy specified real-time constraints. According to the present invention, there is provided a processing apparatus comprising a method for processing a digital signal including the steps of : providing a digital signal, processing said signal using a plurality of processing algorithms, determining statistics of use of processing algorithms, providing a digital statistical signal representative of the statistics, associating to each of the processed signal its own

associated statistics and determining a complexity information signal based on said statistic signal.

[0028] According to the present invention, there is also provided a coding apparatus that further inserts the said complexity information signal into a transmission layer containing or not the associated data and transmits the said complexity information to a decoder apparatus.

[0029] According to the present invention, there is also provided a processing apparatus that generates from the complexity information signal a processing prediction signal which is used to allocate processing resources in order to optimize the real-time processing of the associated data.

[0030] According to the present invention, there is also provided a processing apparatus that receives the processing prediction signal, the allocated processing resources and the complexity information signal, and uses the said signals to generate processing directives so as the data is processed according to the system real-time constraints.

[0031] Although the described invention is of general application, the present description describes its application to the preferred embodiment of the video coding and decoding, and suggests the straightforward extensions to audio coding-decoding and to composition and rendering of media objects.

[0032] The technique provided by the present invention is capable of predicting with an excellent degree of approximation the complexity of video or audio decoding, without the need of the actual decoding of the incoming bit-stream. These results are possible when an appropriate coding information can be included in the incoming bit-streams. The experiments demonstrate that the degree of approximation achievable is excellent considering the range of variability of the actual decoding complexity of video or audio compression standards, and that such degree is more than adequate to enable an efficient processor resource allocation. One application of the proposed technique to the MPEG-4 VM6 video coding algorithm is also demonstrated. Real-time video performances are guaranteed using the predicted decoding complexity to implement Computational Graceful Degradation (CGD) features whenever the video decoding complexity exceeds the available processing resources. By means of the proposed technique, it can be guaranteed at the same time the satisfaction of the real-time constraints and the aim of minimizing the unavoidable degradation of the QoS due to temporary lack of processing resources or to peaks of the intrinsic video decoding complexity. The same approach can obviously be used for audio decoders.

[0033] It is clear that the approach is not limited to the coding/decoding process for compression purposes, but that it can be applied to the process of composing and rendering media objects (i.e. video, audio, text, graphics, two-dimensional and three-dimensional graphic primitives and models) to the display output. The term composition means the action of putting these objects together in the same representation space. The action of transforming the media objects from a common representation space to a rendering device (speakers and viewing window) is called rendering. An illustrative example of the terminal environment for the handling of multimedia object is reported in Figure 19. More specific possible examples of the decoding, composition and rendering processes are reported in Figure 20 and 21. Figure 19 reports the processing stages in an audiovisual terminal illustrating inputs outputs and processing of audiovisual objects. Figure 20 reports an example of a decoder which provides uncompressed audiovisual objects to the audio/video compositor. Figure 21 reports an illustrative example of composition of audiovisual objects.

[0034] Additional objectives and advantages of the invention will be set forth in the description that follows, and in parts will be obvious from the description, or may be learned by practice of the invention. The objectives and advantages of the invention may be realized and obtained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

5. BRIEF DESCRIPTION OF THE DRAWINGS

[0035] The accompanying drawings constitute a part of the specification and illustrate the principles of the invention.

- Figures 1 and 2 report a graphical representation of the decoding complexity of H.263 video illustrating the needs and background of the invention for the video decoding embodiments ;
- Figure 3 reports an example of the modeling of processing algorithms. It includes the association of processing statistic to each state associated to the execution of a specific algorithm ;
- Figure 4 reports the extension of the model of figure 3 for a more complete and accurate model of the algorithmic complexity ;
- Figure 5 reports one preferred embodiment of the invention for the MPEG-4 video decoding. It illustrates a complexity model with the associated statistics for MPEG-4 texture decoding ;
- Figure 6 extends the model of figure 5 including additional coding modes and coding parameters plus the associated statistics ;
- Figures 7 and 8 report simulation results of a preferred embodiment of the invention. By associating a complexity measure to the statistics of the data processing (video decoding in the preferred embodiment), it is possible to predict the processing complexity of data ; actual decoding time, predicted decoding time and prediction error are represented.

- Figure 9 reports the measured decoding time and Decoding Time Prediction (expressed in μsec) versus the sequence VOP number of the MPEG-4 VM6 for the sequence Coastguard (Background, Small boat, Big Boat and Water VOPs) at 64 Kb/s ; these results being obtained on a specific decoder.
- Figure 10 reports the measured decoding time and Decoding Prediction Time (expressed in μsec) versus the sequence VOP number of the MPEG-4 VM6 for the sequence Coastguard (Background, Small boat, Big Boat and Water VOPs) at 128 Kb/s ; these results being obtained on a specific decoder.
- Figure 11 reports the measured decoding time and Decoding Prediction Time (expressed in μsec) versus the sequence VOP number of the MPEG-4 VM6 for the sequence Coastguard (Background, Small boat, Big Boat and Water VOPs) at 128 Kb/s ; these results being obtained on a specific decoder.
- Figure 12 reports a block diagram of part of the invention. The statistics carrying the complexity information and associated to each data to be processed are read by a prediction unit which generates a prediction signal sent to the OS scheduler which allocates resources as function of the prediction signal. The figure is reporting one example of preferred embodiment for the video decoding of multiple VOP ;
- Figure 13 reports a block diagram of part of the invention relative to the construction of the complexity information header at the encoder side. Statistics of the coding and/or decoding algorithms/modes/parameters are generated at the encoder side and then inserted in a bit-stream to be transmitted to the decoder ;
- Figure 14 reports a block diagram of part of the invention relative to the interaction between the decoder supervisor and the decoder using various signals in order to guarantee real-time decoding of the data ;
- Figure 15 reports the diagram of a complexity model including associated degradation algorithms ;
- Figures 16, 17, 18 report examples of results of a preferred embodiment of the invention. The video decoding for the MPEG-4 texture decoding is provided imposing a maximum constant decoding time (complexity) for each VOP. Degradation results are also reported ;
- Figure 19 reports an example of a possible preferred embodiment of the invention illustrating the various interactions and processing related to the handling of media objects (audio, video, text, graphic, 2-D and 3D models) in a multimedia terminal ;
- Figure 20 reports an example of a portion of multimedia terminal block diagram that is another possible embodiment of the invention, and
- Figure 21 reports a drawing illustrating the composition and rendering processes, both being possible preferred embodiments of the invention.

6. PREDICTING DECODING COMPLEXITY BY MEANS OF ENCODING STATISTIC INFORMATION

6.1 A model of decoding complexity for data dependent processing

[0036] To achieve the objectives of the present invention, one need a simple model of data processing complexity that preferably has the following properties :

- 1) it is only related to the processing of the incoming data,
- 2) it is as compact as possible so that its description requires the minimum amount of information (number of bits),
- 3) it is representative in terms of processing complexity,
- 4) it is not related to the any specific software or hardware implementation of the given processing,
- 5) it can be easily mapped on any software/hardware platform implementation,
- 6) it is sufficiently accurate to yield useful predictions of the processing execution time.

[0037] A way of describing the operations of processing incoming data can be to consider the sequence of ensembles of operations as appropriate distinct algorithms. If the sequence of algorithms is not dependent explicitly on the actual content of the data to be processed or by the intermediate results of the processing, the complexity of the processing can be easily calculated a priori and the complexity-modeling problem is trivial.

[0038] In general, if the sequence of operations is explicitly data dependent (i.e. the processing instructions are contained in the data themselves) and depends also on intermediate results of the processing, the problem of modeling algorithmic complexity is much more difficult.

[0039] It is well known that any processing can be modeled by a finite-state-machine for which next state depends on the input, current state and current output (the well-known machine of Mealy). This formulation is rigorous and general. In theory it could also be used to measure the processing complexity. If one assume to associate a complexity measure to each state transition, one could measure the global complexity of the processing by monitoring the state of the machine at each step, and adding up the complexity related to each state transition. In practice, such model is not useful for two main reasons. The state machine description of usual video decoding would result extremely complex. The global complexity associated to the sequence of state transition would result too abstract to make feasible the mapping

on different implementations than the canonical state machine. In practice, it would be extremely difficult to derive a complexity expressed as total execution time for different specific platforms.

[0040] A more useful model is to represent the decoding process by a flux diagram for which each data state corresponds to the execution of a specific algorithm f . Such representation is more naturally related to the usual representations of the execution of software programs. Let us associate to each data state a measure of the occurrence of such state, which is incremented each time the data processing falls in each state. Let us suppose to monitor the execution of the decoding process and, for the occurrence of each algorithm execution, to increment the associated measure. The result of this model is a representation of the data processing given by a sequence of measures associated to each algorithm composing the complexity model. In general, any encoder knows which algorithm sequence is needed for the decoding, since the encoding and the coding syntax define without ambiguities the decoding algorithms. In particular, video decoders have to implement a decoder, to build, reference images, therefore they are able indeed to generate the described complexity representation.

[0041] This modeling approach has the nice feature that admits different levels of complexity resolution. One can consider only a very limited number of main algorithms, or partitioning into a large set of sub-algorithms or any intermediate representation. The cost of the representation, obviously, depends on the number of algorithms that compose the model. The accuracy of the complexity representation depends on the number and type of algorithms, in other words on their relevance in terms of complexity. A simple example of a complexity model is reported in figure 3 which reports a flow graph for the modeling of a processing composed by three algorithms f and associated occurrence complexity measures c . In this model, the processing is represented by tree algorithms f and by the associated counters c . Figure 4 reports a flow graph of the model of Figure 3 for which the complexity modeling of algorithms f_1 and f_3 have been designed by partitioning f_1 and f_3 in two new algorithms and their associated complexity models, the model is more accurate since the algorithms 1 and 3 are partitioned into other algorithms associated to new algorithm occurrence measures. It must be noted that what are considered as coding modes associated to specific coding options in the usual coding language, are from the point of view of the invention distinct algorithms, since different complexities correspond to different coding modes. It must be also noted that the executions of the same algorithm using different parameters can be considered as the executions of distinct algorithms associated to the corresponding parameters.

[0042] The mapping of the complexity into any software/hardware implementation is straightforward. By knowing the complexity (execution time) of each algorithm on a specific platform and by using the information of the actual occurrence measure of each algorithms in the decoding process, one can easily calculate the total execution time (what is denoted as complexity) for the processing of an incoming bit-stream.

[0043] The key issue of this approach is to be able to find the representation in terms of minimal set of algorithms and sub-algorithms (tools) that being representative of the processing complexity, yields the best approximation of the execution time. This is particularly evident for very complex processing task such as video decoding. The importance of a minimal set is to keep as compact as possible the cost of the complexity representation and, therefore, make possible its separate transmission at a low cost.

[0044] Another feature of the proposed model is that it can easily take into account parallel software or hardware implementations. No hypothesis has been done on the internal implementation of each algorithms. In fact, they can be implemented at any level of hardware or software parallelism. The case for which different algorithms of the model have parallel implementations can also be treated by simplifying the model and just ignoring some counters associated to algorithms implemented in parallel. Simple practical examples of these cases are given below after the complete definition of the modeling-prediction complexity problem.

6.2 A technique to predict video decoding complexity : Decoding Complexity Prediction by Coding Statistics (DCPCS)

[0045] Let us consider a specific software/hardware implementation of a video decoder, running on a specific hardware platform. Since no hypothesis is made on the architecture of the software or on the architecture of the hardware, the approach doesn't lose any generality.

[0046] Let us consider that the true complexity C^* (execution time) for the decoding of a given input data (i.e. a single frame or a MPEG-4 VOP) is a function P^* , specific for the software/hardware implementation at hand, of the sequence of all the various processing operations p_i , $i=1, \dots, L$ needed to decode the incoming data :

$$C^* = P^*(p_1, p_2, \dots, p_L). \quad (1)$$

[0047] The p_i describe the sequence of operations that define the decoding process. They are, for instance, the parsing of the bit-stream, the sequence of operations for decoding the in various form Intra, Inter, Not-Coded, Shape, Padding, the inverse quantisation, the IDCT, the motion compensation in its various form OBMC, simple prediction, 4 vector prediction, the parameter that characterize the processing of each prediction mode (integer pel vector prediction, half pel vector prediction), the number of bitwise parsing operations, the number of non-zero DCT coefficients and so on.

[0048] The actual expression of the function P^* is certainly extremely difficult to obtain. On the other hand, the objective of the present invention is to avoid the necessity to know all the parameters that define the complexity, but just to be able to predict the true complexity C^* with a good degree of approximation. The cost of the description of the exact complexity function P^* in terms of all coding parameters would be, indeed, prohibitive for the purposes of the present invention.

[0049] Therefore, let us leave the problem of finding the exact expression of the complexity function P^* and try to find a sufficiently accurate approximation F of the function P^* which provides a prediction C , using the model proposed in the previous section. One is therefore limited to consider a number I of algorithms f_i , with $i=1, \dots, I$ with $I \ll L$, and the associated model counters c_i :

$$C = F(c_1 \cdot f_1, c_2 \cdot f_2, \dots, c_i \cdot f_i, \dots, c_I \cdot f_I) = c_1 \cdot w_1 + c_2 \cdot w_2 + \dots + c_i \cdot w_i + \dots + c_I \cdot w_I \approx C^* \quad (2)$$

where the coefficients w_i related to each c_i are complexity weight corresponding to the model algorithm f_i which accounts for the specific software and hardware architecture at hand.

[0050] Let us suppose now, that the values of the vector c for the incoming data (frame or VOP) are known. If the vector w of the w_i weights is known, one can predict the complexity of the incoming VOP bit-stream by a simple linear combination.

6.3 A method to estimate the algorithms weight coefficients w

[0051] The values of the vector w can be obtained by direct measurements on the platform, if the complexity values (execution time) of each function are accessible. In general, they can be also indirectly estimated by means of a simple linear regression applied on a set of measures of the true complexities C_k^* , with $k=1 \dots K$, obtained by K samples of data processing (i.e. K frames or K VOPs taken from different sequences) on the selected platform:

$$\begin{cases} C_1^* = w_1 \cdot c_{11} + w_2 \cdot c_{12} + \dots + w_i \cdot c_{1i} + \dots + w_I \cdot c_{1I} \\ C_2^* = w_1 \cdot c_{21} + w_2 \cdot c_{22} + \dots + w_i \cdot c_{2i} + \dots + w_I \cdot c_{2I} \\ \dots\dots\dots \\ C_k^* = \dots \\ \dots\dots\dots \\ C_K^* = w_1 \cdot c_{K1} + w_2 \cdot c_{K2} + \dots + w_i \cdot c_{Ki} + \dots + w_I \cdot c_{KI} \end{cases} \quad (3)$$

[0052] In matrix form the estimated vector of weights w can be expressed by the classical least square minimization expression:

$$\hat{w} = (C^T C)^{-1} C^T C^* \quad (4)$$

where T indicates the transpose matrix, and -1 indicates the inverse matrix.

[0053] Obviously, the linear regression and the measures of the C_k complexities can be made off-line since they have as objective the estimation of the vector w which is characteristic of a specific software/hardware decoder implementation and do not depend on the incoming data.

[0054] Once the vector of the w parameters is estimated, the evaluation of the actual complexity C of the incoming bit-stream can be simply done by calculating the linear combination of the complexity function. In practice, this can be done by reading the actual c_i counter values written in an appropriate form in the incoming data, and calculating the value of $F(c)$. This fact implies, for instance, that video compression standards should be able to insert complexity information in appropriate headers of frames or VOPs, or transmit them as accompanying information. Figure 7 reports the measured decoding time and Decoding Time Prediction (expressed in msec) versus the sequence VOP number of the texture decoding of MPEG-4 VM6 for the sequence Akyio (Background and Akyio VOPs) at different bit-rates. The decoder is the Momusys VM6 on a Sun Ultrasparc. The absolute value of the Decoding Time Prediction error is

reported in the bottom curve near the 0 value.

6.4 Definition of the complexity models for the texture decoding of MPEG-4 VM6 VOPs

[0055] In the previous section, a model for decoding complexity has been developed, aiming at being at the same time related only to the processing and apt to be mapped on any platform implementation. These ideal requirements have been expressed in six points. A last requirement needs to be verified. Does the model and the associated complexity representation provide complexity (time) predictions sufficiently compact but accurate enough for the aims of the present invention? There is described in this paragraph an example of application of the modeling to the texture decoding of MPEG-4 VM6 VOPsm, excluding the processing of shape coding and the padding of macroblocks. This exclusion is just motivated by the fact that at the time of the modeling and simulations, one have investigated only the stable coding modules of the MPEG-4 VM. The same modeling principles can be applied to those and to all new MPEG-4 coding algorithms and modes.

[0056] Let us partition the decoding of each MPEG-4 VOP in the following seven algorithms which represent all the texture coding process of any VOP including luminance and chrominance (Intra, Inter, Skip, Advanced Prediction Mode (ADM), Normal Prediction Mode (NPM), DCT coefficient processing (DCT_coeff), and DCT line processing (DCT_line)). Let us associate to each coding-decoding algorithm a statistic measure defined as follows. For Intra, Inter and Skip, it counts each time a macroblock is coded by using the Intra, Inter or Non-Coded coding mode. For ADM and NPM the statistic measure indicates the number of blocks 8x8 predicted using advanced prediction modes (OBMC or four vectors for macroblock) or normal prediction modes. For DCT_coeff, the measure counts the total number of DCT coefficients per VOP, and DCT_line the total number of executed DCT 8x1 per VOP. All these counters can be easily inserted in any MPEG-4 compliant or any other video/audio encoder. Figure 13 reports a schematic diagram of the process of counting coding algorithms/modes/parameters and provides a statistic description of the decoding process according to any complexity model. This diagram represents the generation of the Complexity Information Header at the encoder side. The encoder increments the counters of the coding algorithms/modes/parameters that are used to encode the video frame or video object to generate the associated statistics signal (known as complexity information). At the end of the encoding of each video frame or video object the statistics of the used coding algorithms/modes/parameters is inserted in the Complexity Information Header. Figure 1 reports a simplified flow graph of the MPEG-4 VM6 texture decoding algorithms. Note that, in the proposed technique, it is not necessary to explicitly model the actual decoder flow graph. The above graph represents the schematic flow diagram of a possible simple VOP texture decoding model of MPEG-4. It has to be noticed that by estimating the complexity weights using the total VOP decoding measures as explained in the previous paragraph, it is not even necessary to explicitly specify all transitions in the decoding flow diagram model.

[0057] More accurate complexity models could be designed by introducing new algorithms and the associated measures of occurrences. An example of a more accurate model including the partitioning of APM and NPM into predicted or bi-directional modes and into half-pel motion compensation modes is reported in Figure 6 which represents an example of extension of the MPEG-4 texture decoding complexity model including the different mode prediction and half-pel motion compensation functions.

[0058] A reduction to simpler decoding models when many counters of different algorithms are available is straightforward. Some counters can be ignored if, in a specific software/hardware implementation, they do not implies any or any relevant complexity difference. For instance, if the implementation does not yields different processing time if a macroblock is predicted by a full pel or half pel vector, or if there is no difference between APM or NPM, all these counters can be ignored or can be combined. The specific action depends on the architecture and on the complexity features of the software/hardware decoding platform used.

[0059] As reported in these model examples, a meaningful partitioning of the algorithms in terms of complexity can be easily done only referring to the processing of the data and without the need of specifying any software or hardware implementation. Only in the phase of estimating the weights w_i and to perform the actual prediction it is necessary to refer to a specific hardware/software implementation.

[0060] The two simple models presented here are also capable of taking into account parallel implementations which are implemented in each algorithm such as any packed arithmetic implementation or higher levels of parallelism. Parallel implementations at levels of macroblock decoding are not likely to occur in MPEG-4 systems. Decoders need to respect a given sequential order in decoding the macroblocks that is intrinsic in the syntax of the video data flow. Moreover, for multimedia processor implementations, the ones that are particularly considered in this description, parallelism is likely to be implemented at much lower levels than macroblock decoding. Higher levels are indeed very unlikely to appear.

[0061] Platforms that include also some hardware module accelerators, can be taken into account for the model. If one have, for instance, the hardware implementation of a 8x8 DCT module, one can simply discard the DCT_coeff and DCT_line counters and simply use the combination of Intra and Inter counters to predict the DCT decoding complexity.

Similar measures can be taken for other hardware accelerator modules.

[0062] The present complexity modeling approach can be applied to the audio/video decoding, composition and rendering of media objects or to any real-time processing of data.

7. EXPERIMENTAL RESULTS OF VIDEO DECODING COMPLEXITY PREDICTION

[0063] This section reports simulation results of the prediction of VOP decoding complexity for the MPEG-4 VM6 compression standard. Counters corresponding to the algorithms defined by the model of Figure 5 have been inserted in the Momusys VM6 encoder implementation to generate the mentioned statistic measures. These integer figures have been converted into percentage values referred to the actual number of VOP macroblocks, normalized to 8 bit values and have been supposed to be transmitted as a VOP complexity header for each VOP. The cost of this information transmission results of a few tenths of bits and can be considered negligible if compared to the total video bit-rate. Obviously, any other representation of the statistical coding information which has a sufficient precision can be employed. A schematic diagram of this decoding time (complexity) prediction signal generation process is reported in figure 14 which represents a schematic diagram of VOP video decompression interactions between the Video Decoder and the Video Decoder Supervisor when the Allocated Decoding Time (T_{Alloc}) is lower than the Decoding Time Prediction (T_{Pred}). The Target Decoding Time and Decoder Time Reference are used to verify that the Decompressed Video elements are made available at the desired time. The Target Decoding Time is derived implicitly by the frame or sampling rate or by the information associated to each VOP or video frame (associated time stamps).

[0064] Measures of the texture decoding time of each VOP for different video sequence have been performed on a Sun UltraSparc platform. The decoder is the Momusys VM6 decoder.

[0065] The estimation of the weight w vector has been obtained by a linear regression applied on the measured decoding time of each VOP using the expression of equation (4). To demonstrate the robustness of the proposed technique, the complexity prediction results have been generated using a vector w , estimated using the complexity measurements of all the test sequences, except the one for which the prediction is calculated. For instance, to predict a sequence A, the weights estimated on sequences B and C have been used, and so on respectively for the other test sequences. Table 1 reports the results of the different estimates of the vector w for the sequences Akyio, Coastguard and Bream. As can be easily noticed, complexity (time) estimates from different sequences do not differ relevantly from one to the other. Disposing of a relatively large set of sequences with the associated measurements, one can easily perform off-line estimates of decoder weights w characteristic of the decoder at hand.

[0066] Figure 7 reports the measured VOP decoding time, the VOP decoding time prediction and the time prediction errors (in absolute value) for the two VOPs of the sequence Akyio. Figure 8 represents the measured decoding time and Decoding Time Prediction (expressed in msec) versus the sequence VOP number of the MPEG-4 VM6 for the sequence Bream (Background and Fish VOP) at different bit-rates. The decoder is the Momusys VM6 on a Sun Ultrasparc. The absolute value of the Decoding Time Prediction error is reported in the bottom curve near the 0 value. The results have been obtained at different bit-rates and all measures are expressed in msec. The same experiment results are reported for the sequences Bream and Coastguard. In figures 9, 10 and 11, as can be easily noticed, while the measured actual decoding time has strong variations for each VOP, the decoding time prediction is very accurate and the two curves for most of the samples are difficult to distinguish one from the other. In these experiments, the error of the prediction is lower than 5% for each VOP and in average is around 1-2%.

W (msec/f)	Akiyo	Bream	Coastguard
Intra	38.68	38.84	38.15
Inter	30.95	29.60	29.56
Skip	5.55	5.53	5.51
DCT_line	2.15	0.37	1.24
DCT_coeff	1.47	6.58	6.18
APM	226.47	230.23	228.70
NPM	36.30	37.60	36.01

Table 1. Estimated complexity weights vector w for the texture decoding complexity model of Figure 5 for different sets of measures. The complexity relevance for each weight is given by the product of the algorithm count and by the weight itself.

[0067] The excellent results demonstrate that a relatively simple complexity model is able to predict the decoding time (complexity) with a very good degree of approximation, when appropriate coding statistics are available. The degree of approximation resulting from the proposed method, indeed, is more than adequate to enable an intelligent and efficient resource allocation strategy.

[0068] The description and discussion of the new possible schemes of interaction between the real-time OS and the video decoder are the subject of next sections.

[0069] Another important conclusion is the fact that a very complex process such as MPEG-4 texture video decoding can be described by a suitable high level coding statistic information which is not directly related to any software or hardware implementation. The vector of coding mode counter values is, in fact, a compact coding parameter and coding mode information of the compressed bit-stream. Moreover, the obtained results reports that it is not necessary to define a very detailed model of the software/hardware decoder platform. Very complex decoder features such as cash strategy or memory accesses which could have a non-strictly deterministic behavior, can be taken into account in a statistical sense when one estimate the value of the weight vector characteristic of the decoder platform used.

8. NEW SCHEMES OF DECODER-OS SCHEDULER INTERACTIONS

[0070] New schemes of interaction between the decoder and the OS scheduler of the multimedia processor are possible when decoding time prediction is feasible. These schemes of interaction can be designed in order to provide the best resource allocation policy able to guarantee real-time performances.

[0071] In Figure 12, an example of decoding scheme employing decoding time prediction is reported. This figure represents a decoding scheme example in which the Complexity Information of each VOP is processed by the Complexity Prediction Unit which provides the Decoding Time Prediction signal to the decoder and to the real-time OS/scheduler that allocates the necessary resources (Allocated Decoding Time) to the decoder or decoding processes. The Decoding Time Prediction unit reads the Complexity Information Header carried in each VOP/frame bit-stream header or transmitted as side information in any format, produces the decoding time prediction for each frame/VOP, and provides this information to the real-time OS scheduler. In this way, the OS scheduler can allocate the necessary resources (i.e. Allocated Decoding Time) for the decoding of each or all the VOPs respecting the associated real-time constraints. In case of non real-time encoding, the complexity information can be compressed and transmitted before the real-time transmission and real-time decoding of the video/audio bit-streams. The usefulness of such scheme is evident considering that many VOPs of unknown complexity (decoding time) need to be decoded within an associated time interval

T_c . This scheme is able to guarantee real-time performances if the available video processing resources T_{Alloc} (i.e. Allocated Decoding Time) are sufficient for the decoding of all N incoming VOPs within the given time interval. Formally, the following relation has to be satisfied :

$$T_c \geq T_{Alloc} \geq \sum_{n=1}^N T_n(t) \quad (5)$$

where T_n is the decoding time of the n^{th} VOP, with $n=1, \dots, N$, where N being the VO and VOP index and t being the VOP sampling time.

[0072] This simple procedure scheme is applicable in two cases. The first is when all the VOPs belongs to the same sampling time t (implicitly defined by the sampling rate or also carried by time stamps associated to each VOP). In this case, there is no additional delay introduced by the decoder. The second case occurs when the VOPs can also belong to sequential sampling periods, i.e. M sampling intervals of the reference time t denoted as Dt (also referred as different time stamps). In this case, an additional delay has to be considered and an output buffer needs to be added to the decoder :

$$T_c \geq T_{Alloc} \geq \sum_{n=1}^N \sum_{m=1}^M T_n(t+m\Delta t) \quad (6)$$

[0073] This second scheme enables to smooth out computational peaks on a larger number of VOPs and on a larger time interval than a single image sampling period Dt . In other words, expression (6) might be satisfied even when expression (5) is not for some values of the sampling time t . Obviously, this result is obtained at the cost of an additional delay and an additional output buffer.

[0074] When the available processing resources (i.e. time) are not sufficient to decode all VOPs, this simple strategy cannot guarantee the respect of the real-time constraints. In such cases, other form of interactions and new decoding techniques are needed.

[0075] The main idea is to reduce the complexity of the decoding in order to satisfy relation (5) and (6) accepting some degradation to the resulting image quality. These techniques are called here Computational Graceful Degradation (CGD).

[0076] A schematic diagram of the construction of the complexity information header at the encoder side is reported in Figure 13. The encoder, during the encoding process, increments the registers (counters) corresponding to a set of encoding algorithms/modes/parameters used for the actual VOP or frame compression. At the end of the process, the counters contains a compact encoding/decoding statistical information that is written in the Complexity Information Header.

9. COMPUTATIONAL GRACEFUL DEGRADATION (CGD)

[0077] The main idea of CGD is to make available results of poorer quality when the results of the desirable quality cannot be obtained in real time. When the complexity of the task exceeds the available processing power, real-time services, possibly of degraded quality, are nevertheless provided respecting all real-time constraints. This approach makes the guarantee of the correct scheduling of tasks possible, even when the computational system load fluctuates and relation (5) and (6) cannot be satisfied.

[0078] The concept of graceful system degradation using imprecise results has been previously introduced in the field of real-time computing with applications to control system and automation. In these contexts, imprecise results are defined as intermediate results with the property that the more recent and intermediate result is the closer one that approximates the final result (i.e. the idea only works in monotonically improving computations). For more details, the reader can refer to W.A. Halang, A.D. Stoyenko, Constructing predictable real time systems, Kluwer Academic Publisher, 1991 ; K. Lin, S. Natarjan, J. Liu, Imprecise results : Utilizing partial computations in real-time systems, Proceedings of IEEE Real-Time System Symposium, December 1987 ; J. Liu, K. Lin, S. Natarjan, Scheduling real time periodic jobs using imprecise results, Proceedings of the IEEE Real-Time System Symposium, December 1987 incorporated herein by reference.

[0079] In these cases the problems to be solved are more concerned with fault tolerance, I/O overloads, stability and reactivity to unexpected inputs, and the corresponding techniques are not suitable for purposes of real-time decoding of compressed video/audio bit-streams.

[0080] In contrast, the aim of the present invention is to be able to provide the guarantee of real-time video decoding

with possible degradations of the image quality, i.e. reductions of the peak signal to noise ratio (PSNR), without losing parts of the image or parts of the audio content. Such alternative is certainly much more desirable than other straightforward options such as cutting image portions or simply skipping frames. These latter options, as can be easily understood, can have catastrophic consequences on prediction based coding systems. In H.263 systems, for instance, skipping the decoding of a single frame or a portion of it implies the skipping of all future incoming frames (or portion of them) before the next incoming Intra frame. The time interval of such skipping operation might cover many seconds, and the information content loss might be very relevant. Such abrupt interruptions, which are perceptually very annoying, might be caused not only by the simple fluctuation of the decoding complexity load as discussed in the previous sections, but also by the fluctuations of the load of other non-video/audio processing tasks which reduce the available processing resources (time). Moreover, such events are in general out of the control of the video/audio decoder system and cannot be easily forecasted. Considering these facts, the importance of techniques able to filter out complexity decoding peak loads or able to reduce the average decoding computation load without causing perceptually annoying consequences is evident. Skipping B-VOP could be, in some cases, a viable option for video accepting the introduction of motion judder in the sequence reproduction. Unfortunately, the presence of B-frames or B-VOP cannot be guaranteed in any bit-stream and just at the right time for which a reduction of the complexity is needed to respect the real time constraints. For audio applications, any skip of part of the content has perceptually a very annoying result.

[0081] In contrast with what could be commonly thought, also the classical multi-resolution coding features such as spatial or temporal scalabilities are not really useful for the aims of the present invention. In general, they are used to optimize the transmission bandwidth usage, enabling the compatibility of services at different bit-rates and spatial resolutions, but they must be embedded in the original bit-stream at the encoder side. If they are not embedded, the changes of resolution at the decoder side can result more computationally expansive than the simple decoding at the given resolution.

[0082] Another important reason for the introduction of CGD is given by the definition of profiles, levels and conformance tests for video compression standards based on software/processor platforms. For the MPEG-4 standard, it is expected to have a range of applications that will be much broader than the one of MPEG-1 and MPEG-2. Two possible scenarios can be identified. The first is the definition of a limited number of profiles and levels based on computational power. The second is the possibility of a continuum of decoders of different computational power. The first scenario faces very difficult tasks such as the definition of decoder conformance testing, and at the end it may result in a jungle of too many different levels and profiles.

[0083] The second scenario is completely open to platforms of different computational power. It requires pure functional decoder conformance testing and is able to provide the maximum quality of the service that is achievable by the capabilities of each platform.

[0084] The second scenario, although much more attractive than the first, poses new questions and problems have not yet been studied in video and audio coding. It is mandatory that all decoders can decode all MPEG-4 services but at different levels of quality. It means that a scalability-based on computational power is needed. This new type of scalability should enable a graceful degradation of the video quality when less than theoretically necessary computational power is available at the decoding side. In contrast with MPEG-2 scalabilities that are discrete, embedded in the bit-stream and designed for specific applications, this computational power based scalability should be continuous and implemented in all MPEG-4 bit-streams.

[0085] From these arguments, the usefulness of CGD and its important features and applications are clear. The last key point that has left CGD as a good idea but of difficult usage and implementation for video coding was that it was not clear how to implement it efficiently. Now that the present invention has developed a technique able to predict with a good degree of approximation the required decoding time of each decoding process, one has the key for useful and efficient implementation of CGD techniques.

9.1 Principles of decoding using CGD techniques

[0086] Let us consider a complexity model such as the one of Figure 3 composed by a set of I algorithms f_i , with $i = 1, \dots, I$ and suppose to be able to design for all, or only for a subset S^{CGD} of the functions f_i composing the model, new functions f_i^{CGD} which have the following three properties :

- 1) the execution of a function f_i^{CGD} yields results that are an approximation of the ones produced by the original function f_i ,
- 2) the execution of f_i^{CGD} does not prevent the exact execution of any subsequent decoding functions,
- 3) the weight w_i^{CGD} associated to f_i^{CGD} in the considered decoder implementation is lower than the corresponding w_i associated to f_i (i.e. $w_i^{CGD} < w_i$).

[0087] For a schematic representation of a complexity model which includes new degraded functions, one can refer

to Figure 15 which represents a schematic diagram of a complexity model including CGD algorithms, CGD counters (b_i) and new state transitions generated by the new CGD functions. In this example $f_1, f_3 \in S^{CGD}$, and $f_2 \in S^{Orig}$. In other words, for video decoding for instance, the execution of any f_i^{CGD} might produces PSNR degradations in the reconstruction of some image parts, but not prevents from correctly decoding the rest of the frame or VOP.

[0088] Let us partition the functions forming the complexity model into two disjoint sets, S^{Orig} and S^{CGD} respectively, for the ones that do not admit a CGD version and for the ones for which it is possible to design a CGD version.

[0089] Let us consider for simplicity an image sequence composed by a sequence of single frames or single VOPs (i.e. $N=1$) and suppose that expression (5) is not satisfied for some values of the decoder reference time t :

$$T(t) > T_{Alloc} \quad (7)$$

where $T(t)$ is the decoding time of the VOP at hand at the sampling time t , and to define as processing deficit T_{Def} the time for which the decoding complexity exceeds the Allocated Decoding Time :

$$T(t) - T_{Alloc} = T_{Def} \quad (8)$$

[0090] Since the decoding time can be approximated by its prediction (Decoding Time Prediction) it can be expressed by :

$$T(t) \approx T_{Pred}(t) = \sum_{i=1}^I c_i w_i \quad (9)$$

where the c_i are the counters of the complexity model and the w_i are the corresponding complexity weights.

[0091] The lower limit of the decoding time executing all CGD algorithms f_i^{CGD} at the place of f_i for any $i \in S^{CGD}$, is given by :

$$T_{CGD} = \sum_{i \in S_{CGD}} c_i w_i^{CGD} + \sum_{i \in S_{Orig}} c_i w_i \quad (10)$$

[0092] If $T_{CGD} < T_{Alloc} < T_{Pred}$ it is therefore possible to guarantee the real-time constraints by executing the f_i^{CGD} algorithms and accepting approximate decoding results. The difference $T_{Pred} - T_{CGD}$ can also be expressed as :

$$T_{Pred} - T_{CGD} = \sum_{i \in S_{CGD}} c_i (w_i - w_i^{CGD}) \quad (11)$$

[0093] If $T_{CGD} < T_{Alloc} < T_{Pred}$, it is therefore possible to find for each c_i with $i \in S^{CGD}$, a couple of integer coefficients a_i, b_i , for which :

$$a_i + b_i = c_i \quad (12)$$

and denoting as $T_{Process}$ any decoding time interval which respects the real time constraints (obtained by the implicit sampling rate or expressed using appropriate time stamps associated to each VOP or video frames, or audio packet), one can express the desired process decoding time as any decoding time value that satisfy the expression :

$$T_{process} = \sum_{i \in S_{CGD}} (a_i w_i^{CGD} + b_i w_i) + \sum_{i \in S_{Orig}} c_i w_i \leq T_{Alloc} \quad (13)$$

[0094] Obviously for any Allocated Decoding Time the aim is to respect the real-time constraints with the lowest degradation in the video/audio content. Associating a degradation measure to each w_i^{CGD} it is possible to minimize the degradation for a given TAlloc. Therefore, the aim is to find the set $S_{a,b}$ of couple of coefficients a_i, b_i , with $i \in S^{CGD}$, that satisfy expression (12) and minimize the video or audio degradations expressed as difference with the non degraded output :

$$S_{a,b} = \underset{i \in S_{CGD}}{\operatorname{argmin}} (a, b) : (PSNR - PSNR(a_i, b_i)) \quad (14)$$

[0095] The determination of the set $S_{a,b}$ can be done by means of any suitable optimization method or any suitable optimization heuristic by the Video Decoder Supervisor, schematically represented in Figure 14. In summary, the Video Decoder Supervisor receives from the OS the Allocated Decoding Time signal (T_{Alloc}), the decoding time prediction signal (TPred) from the Complexity Prediction unit, and the Complexity Information signal represented by the actual values of the coding/decoding statistic (set of c_i coefficients) for the VO. Since the Video Decoder Supervisor knows the values of the set of w_i and w_i^{CGD} it can determine, by a suitable standard optimization technique, the corresponding set of coefficients a_i, b_i such that expression (12) is satisfied. Having defined the set of a_i, b_i coefficients the CGD decoding is partially defined. In other words, the set of coefficients defines for each $i \in S^{CGD}$ how many degraded algorithms (f_j^{CGD}) at the place of original algorithms (f_j) need to be executed.

[0096] Since the determination of the degradation coefficients is done on the basis of the decoding time prediction, the video decoder can generate a Decoding Status signal reporting the status of the decoding (i.e. the fraction of the bit-stream or the fraction of the VO already decoded). The signal representing the Decoding Status of the decoding process reports to the Video Decoder Supervisor an information which can be used to correct the Computational Graceful Degradation Directive signal (i.e. the setting of the a_i, b_i coefficients in this example) if the Current Decoding Time and the Current Decoding Time Prediction signal are not conformant. Current Predicted Decoding Time means the Decoding Time Prediction calculated by using relation (13), only counting the occurrences of the decoding algorithms actually used to decode the fraction of the decoded bit-stream at the current reference time (time for which the decoding status signal is generated). The term not conformant means that the actual Decoding Time is greater than the Decoding Time Prediction of an amount of time which overcomes statistical variations which are functions of the complexity information, thus reporting possible errors in the Decoding Time Prediction signal.

[0097] The described interactions between OS, the Decoder and Video Decoder Supervisor can be easily extended for the case of sequences containing multiple VO at the same sampling time (associated time stamps), or belonging to a defined time interval containing several sampling periods (several associated time stamps). Expressions (10, 11, 13, 14) can be easily modified including the appropriate summation over all VOP. Therefore, in the case of N VOP at the same sampling time t , one have :

$$T_N(t) = \sum_{n=1}^N T_n(t) > T_{Alloc} \quad (15)$$

where $T_N(t)$ is the total decoding time of the N VOP.

$$T_N(t) \approx T_{Pred,N} = \sum_{n=1}^N T_{Pred,n}(t) = \sum_{n=1}^N \sum_{i=1}^I c_{i,n} w_i \quad (16)$$

$$T_{CGD,N} = \sum_{n=1}^N \left(\sum_{i \in S_{CGD}} c_{i,n} w_i^{CGD} + \sum_{i \in S_{Orig}} c_{i,n} w_i \right) \quad (17)$$

$$T_{Pred,N} - T_{CGD,N} = \sum_{n=1}^N \sum_{i \in S_{CGD}} c_{i,n} (w_i - w_i^{CGD}) \quad (18)$$

$$a_{i,n} + b_{i,n} = c_{i,n} \quad (19)$$

$$T_{Process,N} = \sum_{n=1}^N \left(\sum_{i \in S_{CGD}} (a_{i,n} w_i^{CGD} + b_{i,n} w_i) + \sum_{i \in S_{Orig}} c_{i,n} w_i \right) \leq T_{Alloc} \quad (20)$$

$$S_{a,b,N} = \operatorname{argmin}_{\{a,b\} \in S_{CGD}} \left(\sum_{n=1}^N (PSNR - PSNR(a_{i,n}, b_{i,n})) \right) \quad (21)$$

or :

$$S_{a,b,N} = \operatorname{argmin}_{\{a,b\} \in S_{CGD}} \left(\sum_{n=1}^N \alpha_n (PSNR - PSNR(a_{i,n}, b_{i,n})) \right) \quad (22)$$

where a_n is a semantic degradation coefficient which might be available in the coded bitstreams.

[0098] For N Video Objects and M sampling time intervals Dt (denoted also as time stamps) one have :

$$T_{N,M} = \sum_{m=1}^M T_N(t+m\Delta t) = \sum_{m=1}^M \sum_{n=1}^N T_n(t+m\Delta t) > T_{Alloc} \quad (23)$$

$$T_{N,M} \equiv T_{Pred,N,M} = \sum_{m=1}^M \sum_{n=1}^N T_{Predn}(t+m\Delta t) = \sum_{m=1}^M \sum_{n=1}^N \sum_{i=1}^I c_{i,n,m} w_i \quad (24)$$

$$T_{CGD,N,M} = \sum_{m=1}^M \sum_{n=1}^N \left(\sum_{i \in S_{CGD}} c_{i,n,m} w_i^{CGD} + \sum_{i \in S_{Orig}} c_{i,n,m} w_i \right) \quad (25)$$

$$T_{Pred,N,M} - T_{CGD,N,M} = \sum_{m=1}^M \sum_{n=1}^N \sum_{i \in S_{CGD}} c_{i,n,m} (w_i - w_i^{CGD}) \quad (26)$$

$$a_{i,n,m} + b_{i,n,m} = c_{i,n,m} \quad (27)$$

$$T_{Process,N,M} = \sum_{m=1}^M \sum_{n=1}^N \left(\sum_{i \in S_{CGD}} (a_{i,n,m} w_i^{CGD} + b_{i,n,m} w_i) + \sum_{i \in S_{Orig}} c_{i,n,m} w_i \right) \leq T_{Alloc} \quad (28)$$

$$S_{a,b,N,M} = \operatorname{argmin}_{(a,b)_{i \in S_{\text{COD}}}} \left(\sum_{m=1}^M \sum_{n=1}^N (PSNR - PSNR(a_{i,n,m}, b_{i,n,m})) \right) \quad (29)$$

or

$$S_{a,b,N,M} = \operatorname{argmin}_{(a,b)_{i \in S_{\text{COD}}}} \left(\sum_{m=1}^M \sum_{n=1}^N \alpha_n (PSNR - PSNR(a_{i,n,m}, b_{i,n,m})) \right) \quad (30)$$

9.2 Extension of the techniques to video/audio composition

[0099] The methods aiming at predicting the decoding time (i.e. generating a Decoding Time prediction Signal) of video frames or video objects, so as to respect the real-time constraints specified implicitly by the audio/video rates or by the timing information carried by the associated time stamps, can be easily extended to the video/audio composition processes. Composition process means the action of putting decompressed video or audio objects together in the same representation space. In a similar way of the decoding process, one can generate a Composition Time Prediction for each composition process according to the interactions described in Figure 12. Such Composition Time Prediction is a function of the occurrence of composition algorithms, composition modes and composition parameters (associated composition statistics). Differently from the decoding process, composition algorithms, modes and parameters can be also provided by an interactive user interface. In a similar way the Composition Allocated Time can be generated as function of the Composition Time Prediction. The interactions between the compositor and the OS can be described as in Figure 12 where the Decoder is substituted by the Compositor.

[0100] The technique of associating Decoding Degradation Algorithms and decoding by using degradation algorithms described in section 6.1 and Figure 14 can be applied by substituting respectively the Allocated Decoding Time by the Decoding Time Prediction, the Target Decoding Time and the Decoder Reference Time by the Allocated Composition Time, the Composition Time Prediction by the Target Composition Time and the Compositor Reference Time. All relations and equations reported in section 6.1 referring to the decoding process can be referred to the composition process.

9.3 Examples of possible degradation modes

[0101] The decoding operations of a video compression algorithm can be divided into two main categories :

- 1) operations that cannot be simplified since they may prevent the extraction and usage of relevant coding information,
- 2) operations that can be simplified, possibly leading to imprecise or degraded video results.

[0102] To the first category typically belong all bit-stream parsing operations, entropy decoding (i.e. VLD and arithmetic coding), and all operations related to synchronization and to the syntax of the decoding process. For these operations, no degradation mode definition is possible, otherwise it would imply skipping image portions or entire frames since either the data or the coding modes and coding parameters are lost.

[0103] To the second category, for instance, belong operations such as the IDCT, the prediction with and without motion compensation in its various forms (PB/B-frame prediction, overlapped block motion compensation (OBMC), half pixel interpolation and so on, padding, shape coding. Simplifications of such operations applied to pixels or transform coefficients are always possible but at the expenses of imprecise results. The challenge can be expressed as follows : given a specific compression standard, software and processor architecture, implement it and find the f_i^{CGD} which provides the maximum complexity reduction ($w_i - w_i^{\text{CGD}}$) versus the minimum image/audio quality degradation (usually measured in PSNR). Many are the algorithms that can be simplified. We report here only two examples that illustrate the idea.

9.3.1 IDCT simplified modes

[0104] IDCT on a 8x8 block can be decomposed into 16 IDCT operation on a 8x1 block, or on IDCT considering only

the lowest 4x4 or 2x2 coefficients. Skipping higher order 8x1 IDCT or enabling processing only 2x2 or 4x4 coefficients constitute some alternatives which are less complex than the standard 8x8 IDCT. Obviously, these less demanding alternatives yield a filtering of the high frequency possibly contained in the higher order coefficients.

9.3.2 Simplified prediction modes

[0105] Many prediction modes can be simplified at the expenses of some PSNR degradations. In particular, the most complex such as the OBMC with or without half-pel motion compensation can yield relevant complexity decreases of both computational complexity and memory bandwidth by implementing simpler algorithms.

10. EXAMPLES OF RESULTS FOR THE MPEG-4 VIDEO COMPRESSION STANDARD

[0106] This section reports simulation results of the decoding of VOP using the complexity model and decoding time prediction results described in Section 4. The MPEG-4 VM6 algorithm in the Momusys VM6 implementation modified adding degradation functions to the advanced prediction modes has been used for the simulations. A fixed amount of processing time is allocated for each VOP composing the sequences. In case the decoding time prediction (T_{Pred}) for each VOP exceeds the fixed allocated time (T_{Alloc}), an appropriate number of advanced prediction modes are executed using degraded functions in order to satisfy relation (13).

[0107] Figures 16, 17, 18 report the decoding results at 64, 128 and 256 kb/s for the four VO (Small Boat, Big Boat, background and Water) of the sequence Coastguard. On the left, the diagrams report the decoding time of each VO expressed in msec versus the VOP number. Two curves are reported : original decoding and decoding using degradation function in order to satisfy the decoding time limits. The PSNR image degradations corresponding to the execution of degraded functions are reported on the right.

[0108] Figure 16 reports on the left standard decoding time and CGD decoding time (in msec) for the four VO of the sequence Coastguard at 64 kb/s. Each VO has a upper bound for the CGD decoding time. On the right the PSNR loss of the CGD decoding with reference to the standard decoding.

[0109] Figure 17 reports on the left standard decoding time and CGD decoding time (in msec) for the four VO of the sequence Coastguard at 128 kb/s. Each VOP has a upper bound for the CGD decoding time. On the right the PSNR loss of the CGD decoding with reference to the standard decoding.

[0110] Figure 18 reports on the left standard decoding time and CGD decoding time (in msec) for the four VO of the sequence Coastguard at 256 kb/s. Each VO has a upper bound for the CGD decoding time. On the right the PSNR loss of the CGD decoding with reference to the standard decoding.

[0111] The implementation of CGD base on the decoding time prediction generation technique enables, within the discussed limits, the a priori setting of the decoding time guaranteeing the possibility of a scheduling of each decoding task that satisfy real time constraints, even if the available resources (Allocated Decoding Time) are inferior to the theoretically necessary ones. Moreover, the degradations introduced are visually acceptable and they can be minimized according to expression (14) by suitable techniques. The results obtained with a PSNR degraded are certainly much more desirable than introducing sudden interruptions and image content losses to the video sequence, and enables an efficient resource allocation policy.

Claims

1. Method for primary processing a digital signal comprising the steps of :

- providing a digital signal to a primary processing unit ;
- primary processing said signal with said primary processing unit according to a plurality of primary processing algorithms to provide primary processed output signals,
- determining statistics of use of at least one of said primary processing algorithms,
- providing a digital statistics signal representative of said statistics for each of said primary processed output signals
- associating to each of said primary processed output signals its own statistics signal ;
- determining a complexity primary processing information signal based on said statistic signal.

2. Digital primary processed output signal, generated by a primary processing unit using primary processing algorithms and being intended to be transmitted to a receiving device wherein said primary processed output signal comprises data bit streams, each of said data bit streams comprising a main bit string representative of said primary processed output signal and a secondary bit string representative of the statistics of use of said primary processing algorithms associated to said main bit string.

3. Method for secondary processing a secondary digital output signal according to claim 2 comprising the steps of :

- (a) reading said secondary bit string ;
- (b) determining, from the secondary bit string, primary processing algorithms to be used to secondary process said primary output signal associated to said main bit string, for generating a complexity primary processed information signal for each signal ;
- (c) allocating a secondary processing time and/or operating system resources as a function of said complexity primary processed information signal for secondary processing of said primary output signal ;
- (d) sending main bit string to secondary processing means ;
- (e) secondary processing said main bit string using said allocated secondary processing time and/or operating system resources, according to a secondary process and
- (f) generating a secondary processed signal.

4. Method for coding a digital signal comprising the steps of :

- providing an uncompressed digital signal including audio packets, video frames, video planes, or audio/video objects to an encoder ;
- compressing said signal with said encoder according to a compression syntax, said encoder using a plurality of coding algorithms to provide a compressed signal including coded audio packets, video frames, video planes or audio/video objects,
- determining statistics of use of at least one of decoding algorithms necessary to decode each of said coded audio packets, video frames, video planes or audio/video objects from said associated coding algorithms, and/or determining statistic of use of at least one of coding algorithms,
- providing a digital statistics signal representative of said statistics for each of said coded audio packets, video frames, video planes or audio/video objects,
- associating to each of said coded audio packets, video frames, video planes or audio/video objects its own statistics signal.

5. Coding method according to claim 3, wherein the step of determining statistics consists in determining the statistics of use of some of decoding algorithms, wherein to each decoding algorithm is associated at least one decoding mode and/or decoding parameter and wherein the statistics signal is based on the decoding modes and/or decoding parameters.

6. Coding method according to claim 3 or 4, wherein the step of determining statistics further consists in determining the statistics of use of some of coding algorithms when said coding algorithms and the compression syntax allow the determination of corresponding decoding algorithms necessary to decode each of said coded audio packets, video frames, video planes or audio/video objects wherein to each coding algorithm is associated at least one coding mode and/or coding parameter and wherein the statistics signal is based on the coding modes and/or coding parameters necessary to decode when using said coding algorithms.

7. Coding method according to claim 4, wherein the statistic signal is generated by juxtaposing digital words each representing the occurrences of use of a decoding mode and/or decoding parameter necessary to decode said associated audio packet, video frame, video plane or audio/video object.

8. Coding method according to claim 5 and/or 6, wherein the statistic signal is generated by juxtaposing digital words each representing the occurrences of use of a coding mode and/or coding parameter necessary to decode said associated audio packet, video frame, video plane or audio/video object.

9. Coding method according to any of claims 3 to 7, wherein the step of associating consists in inserting said statistics signal into said compressed signal for each associated coded audio packet, video frame, video plane or audio/video object.

10. Coding method according to any of claims 3 to 8, wherein the step of determining statistics consists in computing the occurrence of use of said coding and/or decoding algorithms for determining a complexity information for each coded audio packet, video frame, video plane or audio/video object.

11. Digital compressed signal including coded audio packets, video frames, video planes or audio/video objects, said signal being generated by an encoder using a compression syntax and coding algorithms and being intended to be

transmitted to a receiving device wherein said signal is decoded according to decoding algorithms, and wherein said signal comprises at least a layer of coded audio packets, video frames, video planes or audio/video objects bit streams, each of said bit streams comprising a main bit string representative of the coded audio packets, video frames, video planes or audio/video objects and a secondary bit string representative of the statistics of use of said coding and/or decoding algorithms associated to said main bit string.

12. Digital compressed signal, according to claim 10, wherein said secondary bit string is inserted in said layer of said bit streams.

13. Digital compressed signal, according to claim 10, further comprising a plurality of parallel layers of coded audio packets, video frames, video planes, or audio/video objects bit streams, wherein said main bit string is inserted or transmitted in a layer different to said second bit string.

14. Digital compressed signal, according to claim 11 or 12, wherein said secondary bit string is inserted or transmitted prior to said main bit string.

15. Digital compressed signal, according to any of claims 10 to 13, wherein to each decoding algorithm is associated at least one decoding mode and/or decoding parameter and wherein said secondary bit string consists of words each representing the occurrences of use of the decoding modes and/or decoding parameters.

16. Digital compressed signal, according to claim 10 or 14, wherein to each coding algorithm is further associated at least one coding mode and/or coding parameter and wherein said secondary bit string consists of words each representing the occurrences of use of the coding modes and/or coding parameters, said coding algorithms and the compression syntax allowing the determination of corresponding decoding algorithms necessary to decode each of said coded audio packets, video frames, video planes or audio/video objects.

17. Method for decoding a digital compressed signal according to any of claims 10 to 15 comprising the steps of :

- (a) reading said secondary bit string ;
- (b) determining, from the secondary bit string, decoding algorithms to be used to decode said coded audio packets, video frames, video planes or audio/video objects, associated to said main bit string, for generating a complexity information signal for each coded audio packet, video frame, video plane or audio/video object ;
- (c) allocating a decoding time and/or operating system resources as a function of said complexity information signal for the decoding process of said compressed audio packet, video frames, video planes or audio/video objects ;
- (d) sending main bit string to decoding means ;
- (e) decoding said main bit string using said allocated decoding time and/or operating system resources, according to a decoding process and
- (f) generating a decompressed signal including decoded audio packets, video frames, video planes or audio/video objects.

18. Method for decoding a digital compressed signal according to claim 16 wherein the step determining decoding algorithms is also based on the occurrence of use of the decoding modes and/or decoding parameters necessary to decode the said audio packets, video frames, video planes or audio/video objects.

19. Method for decoding a digital compressed signal according to claim 16 or 17 wherein the step determining decoding algorithms is further based on the occurrence of use of the coding modes and/or coding parameters used by said coding algorithms, wherein said coding algorithms and the compression syntax allow the determination of corresponding decoding algorithms necessary to decode each of said coded audio packets, video frames, video planes or audio/video objects

20. Decoding method according to any of claims 16 to 18 further comprising the step of determining a decoding time prediction signal as a function of said complexity information signal ; said decoding time prediction signal being used for the step of allocating said decoding time and/or operating system resources for optimizing said decoding process and or decoding processes.

21. Decoding method according to claim 19 wherein the step of decoding the main bit string consists in

- generating an allocated decoding time signal,
- sending said decoding time prediction signal and allocated decoding time signal to decoder supervisor means ;
- generating and sending a degradation directive signal to said decoding means, as a function of said decoding time prediction signal, allocated decoding time, so as to define the use of number and type of decoding algorithms and degradation algorithms to be used in the decoding process of said main bit string.

22. Decoding method according to claim 20 wherein the step of decoding the main bit string further consists in :

- generating a decoder time reference signal and a target decoding time signal,
- sending said decoder time reference signal and said target decoding time signal to decoder supervisor means ;
- generating and sending said degradation directive signal to said decoding means while also taking into account said decoder time reference signal and said target decoding time signal.

23. Decoding method according to claim 20 or 21 wherein the step of decoding the main bit string further consists in :

- generating a decoding status signal from said decoding means
- sending said decoding status signal to said decoder supervisor means from the said decoding means ;
- generating and sending said degradation directive signal to said decoding means while also taking into account said decoding status signal.

24. Decoding method according to claim 16 wherein steps (a) to (c) are carried out prior to step (e).

25. Decoding method according to any of claims 20 to 23, further comprising the step of sending said complexity information signal to said decoder supervisor means.

26. Decoding method according to any of claims 20 to 24 wherein the step of generating said degradation directive signal consists in associating at least one degradation algorithm to at least one decoding algorithm, said decoding algorithm and said degradation algorithm being associated to a complexity coefficient/weight representative of the execution time of said decoding algorithm and said degradation algorithm.

27. Decoding method according to claim 25 wherein said complexity coefficients/weights are dependent on the decoder used for implementing said decoding method.

28. Decoding method according to claim 25 or 26, wherein said decoding time prediction is a function of the occurrence of use of said decoding algorithm and said complexity coefficients/weights according to the following equation :

$$T(t) = T_{Pred}(t) = \sum_{i=1}^I c_i w_i \quad (9)$$

where T is the time necessary to decode a coded audio packet, video frame, video plane or audio/video object, T_{Pred} is the decoding time prediction of said audio packets, video frames, video planes or audio/video objects, c_i is the occurrence of said decoding algorithm, and w_i is the complexity coefficient/weight.

29. Decoding method according to claim 27, wherein when the decoding time prediction is a function of the occurrence of use of said decoding algorithms and degradation algorithms, said degradation algorithms being associated to a complexity coefficient/weight.

30. Decoding method according to claim 27 or 28, wherein when the decoding time prediction is greater than the allocated decoding time, one uses a degradation method according to which one selects the degradation algorithm instead of its associated decoding algorithm according to equation :

$$T_{process} = \sum_{i \in S_{CGD}} (a_i w_i^{CGD} + b_i w_i) + \sum_{i \in S_{Orig}} c_i w_i \leq T_{Alloc} \quad (13)$$

where, T_{Process} is the prediction time necessary to decode a audio packets, video frames, video planes or audio/video objects, said decoding time prediction taking account the degradation, c_i is the occurrence of decoding algorithm, a_i is the number the selected degradation algorithms, w_i^{CGD} is the complexity coefficient/weight of the degradation algorithm, b_i is the number of the non selected degradation algorithms, and w_i is the complexity coefficient/weight, where $a_i + b_i = c_i$, so as to reach a T_{Process} lesser than or equal to said allocated decoding time.

31. 30. Decoding method according to claim 28 or 29, wherein said decoding status signal is also used for monitoring the current decoding time of a coded audio packets, video frames, video planes or audio/video objects and make sure that said current decoding time is in conformity with the associated decoding time prediction.

32. Decoding method according to claim 30, wherein if said current decoding time is not in conformity with the associated predicted decoding time, then said decoder supervisor means uses at least one further degradation algorithm defined in claim 29.

33. Decoding method according to claim 31, wherein if said current decoding time is not in conformity with the associated decoding time prediction, then said decoder supervisor means further uses said complexity information signal.

34. Decoding method according to any of claims 29 to 32 for a plurality of audio packets, video frames, or video planes, some of them belonging to different audio/video objects, wherein when the decoding time prediction is the sum of the decoding time prediction of each of said audio packets, video frames or video planes of said plurality, said decoding time prediction being defined in claim 28.

35. Coding system for an uncompressed digital signal including audio packets, video frames, video planes or audio/video objects, comprising :

- an audio/video encoder receiving said audio packets, video frames, video planes or audio/video objects on one input terminal, said encoder being intended to code said signal according to a compression syntax by using a plurality of coding algorithms, and to provide, on one output terminal, a compressed signal including coded audio packets, video frames, video planes or audio/video objects intended to be decompressed according to associated decoding algorithms ;
- means for determining statistic of use of at least one of said coding algorithms, and/or means for determining statistic of use of at least one of said decoding algorithms ;
- and means for generating a digital statistic signal representative of said statistics for each of said coded audio packets, video frames, video planes or audio/video objects.

36. Coding system according to claim 27, wherein said statistics determining means comprise computing means for counting the occurrence of use of coding modes and/or coding parameters used by the encoder for encoding said signal, when using said coding algorithms.

37. Coding system according to claim 34 or 35, wherein said statistics determining means further comprise computing means for counting the occurrence of use of decoding modes and/or decoding parameters from said coding algorithms used by the encoder for encoding said signal.

38. Coding system according to any of claim 34 to 36, further comprising means for generating a compressed output signal including coded audio packets, video frames, video planes or audio/video objects wherein each of said coded audio packet, video frame, video plane or audio/video object is associated with its statistics signal.

39. Decoding system for a digital signal including coded audio packets, video frames, video planes or audio/video objects according to any of claims 5 to 15, comprising :

- a decoder receiving said main bit string, representative of a coded audio packets, video frames, video planes or audio/video objects, on a first input terminal and providing, on an output terminal, a decompressed signal including decoded audio packets, video frames, video planes or audio/video objects,
- resource allocation means ;
- means for complexity prediction receiving, on one input terminal, said secondary bit string, and providing to resource allocation means a decoding time prediction signal which is determined as a function of the information contained in said secondary bit string, said resource allocation means providing to the decoder information representative of the allocated decoding time.

40. Decoding system according to claim 38 wherein said resource allocation means comprises an operating system or a layer of an operating system.
- 5 41. Decoding system according to claim 38 or 39 wherein said decoding time prediction signal is also provided to a second input terminal of said decoder.
42. Decoding system according to any of claims 38 to 40, wherein
- said decoder comprises decoder supervisor means and audio/video decoding means,
 - 10 - said decoder supervisor means receiving, on a first input terminal, said allocated decoding time signal and, on a second input terminal, said predicted decoding time signal and providing on one output terminal a degradation directive signal,
 - said decoding means receiving on a first input terminal said main bit string and on a second input terminal said degradation directive signal and providing on an output terminal said decompressed signal including decoded audio packets, video frames, video planes or audio/video objects.
 - 15
43. Decoding system according to claim 41 wherein said decoder further receive a complexity information signal on third input terminal.
- 20 44. Decoding system according to claim 41 wherein said decoder further receive a decoder time reference signal and a target decoding time signal respectively on fourth and fifth input.
45. Decoding system according to any of claims 41 to 43 wherein said decoding means are arranged to provide a decoding status signal to a sixth input of the decoder supervisor means.
- 25 46. Decoding system according to claim 44 wherein said complexity information signal is applied to said decoder supervisor means.
47. Method for composing a digital signal comprising the steps of :
- 30
- providing an uncompressed video element including video frames and/or video planes and/or video objects, textures, texts, 2 and 3 dimensional graphic primitives, 2 and 3 dimensional models to a composing device ;
 - composing said video element with said composing device according to a composition syntax and/or an interactivity information, said composing device using a plurality of composing algorithms necessary to compose each of said video elements and provide a representation in a common visual representation space of the said multiple elements,
 - 35 - determining statistics of use of at least one of said composing algorithms for generating a complexity information
 - associating to each of said video elements its own complexity information.
 - 40
48. Composition method according to claim 46 further consisting in :
- providing an uncompressed digital audio element including audio packets, audio objects, synthetic audio objects
 - 45 - composing said audio element with said composing device according to a composition syntax, said audio composing device using a plurality of composing algorithms necessary to compose each of said audio elements and provide a representation in a common audio representation space
 - determining statistics of use of at least one of said algorithms for generating a complexity information.
 - associating to each of said audio element its own complexity information.
 - 50
49. Composition method according to claim 46 or 47, wherein to each composition algorithm is associated at least one composition mode and/or composition parameter and wherein the statistics is based on the occurrence of use of composition modes and/or composition parameters.
- 55 50. Method for composing a digital signal according to any of claims 46 to 48 comprising the steps of :
- (a) reading the complexity information associated to at least one of said video element and/or audio element
 - (c) allocating a composition time and/or operating system resources as a function of said complexity information

tion signal for the composition process of said associated video element and/or audio element ;
 (d) sending said video element and/or audio element to composition means ;
 (e) compose in a common representation space said video element and/or audio element using said allocated
 5 composition time and/or operating system resources, according to a composition process, and
 (f) generating a common representation space including said video element and/or audio element according to
 the composition syntax.

51. Method for composing digital video/audio elements according to claim 49 wherein the step determining composition
 10 algorithms is also based on the occurrence of use of the composition modes and/or composition parameters
 necessary to compose the said composition elements in a common composition space.

52. Composition method according to any of claims 48 to 50 further comprising the step of determining a composition
 time prediction signal as a function of said complexity information ; said composition time prediction signal being
 15 used for the step of allocating said composition time and/or operating system resources for optimizing said composition
 process and or composition processes.

53. Composition method according to claim 50 wherein the step of composing the audio/video elements consists in

- generating an allocated composition time signal,
- 20 - sending said composition time prediction signal and allocated composition time signal to composition supervisor
 means ;
- generating and sending a degradation directive signal to said composition means, as a function of said composition
 time prediction signal, allocated composition time, so as to define the use of number and type of composition
 25 algorithms and degradation algorithms to be used in the composition process of said video/audio
 composition elements.

54. Composition method according to claim 52 wherein the step of composing the video/audio elements further consists
 in :

- generating a compositor time reference signal and a target composition time signal,
- 30 - sending said compositor time reference signal and said target composition time signal to composition supervisor
 means ;
- generating and sending said degradation directive signal to said composition means while also taking into
 account said compositor time reference signal and said target composition time signal.

55. Composition method according to claim 52 or 53 wherein the step of composing the audio/video elements further
 consists in :

- generating a compositor status signal from said composition means
- 40 - sending said compositor status signal to said composition supervisor means from the said composition means
 ;
- generating and sending said degradation directive signal to said means while also taking into account said
 compositor status signal.

56. Composition method according to claim 48 wherein steps (a) to (c) are carried out prior to step (e).

57. Composition method according to any of claims 51 to 55, further comprising the step of sending said complexity
 information signal to said compositor supervisor means.

58. Decoding method according to any of claims 53 to 56 wherein the step of generating said degradation directive signal
 50 consists in associating at least a degradation algorithm to at least one composition algorithm, said composition
 algorithm and said degradation algorithm being associated to a complexity coefficient/weight representative of the
 execution time of said composition algorithm and said degradation algorithm.

59. Composition method according to claim 57 wherein said complexity coefficients/weights are dependent on the
 compositor used for implementing said composition method.

60. Composition method according to claim 57 or 58, wherein said composition time prediction signal is a function of

the occurrence of use of said composition algorithms and said complexity coefficients/weights according to the following equation :

$$T(t) = T_{Pred}(t) = \sum_{i=1}^I c_i w_i \quad (9),$$

where T is the time necessary to compose a audio/video element, T_{Pred} is the composition time prediction of said audio/video element, c_i is the occurrence of said composition algorithm, and w_i is the complexity coefficient/weight.

61. Composition method according to claim 59, wherein when the composition time prediction is a function of the occurrence of use of said composition algorithms and degradation algorithms, said degradation algorithms being associated to a complexity coefficient/weight.

62. Composition method according to claim 59 or 60, wherein when the composition time prediction is greater than the allocated composition time, one uses a degradation method according to which one selects the degradation algorithm instead of its associated composition algorithm according to equation :

$$T_{process} = \sum_{i \in S_{CGD}} (a_i w_i^{CGD} + b_i w_i) + \sum_{i \in S_{Orig}} c_i w_i \leq T_{Alloc} \quad (13)$$

where, $T_{Process}$ is the prediction time necessary to compose a audio/video element, said composition time prediction taking account the degradation, c_i is the occurrence of composition algorithm, a_i is the number the selected degradation algorithms, w_i^{CGD} is the complexity coefficient/weight of the degradation algorithm, b_i is the number of the non selected degradation algorithms, and w_i is the complexity coefficient/weight, where $a_i + b_i = c_i$, so as to reach a $T_{Process}$ lesser than or equal to said allocated composition time.

63. Composition method according to claim 60 or 61, wherein said composition status signal is also used for monitoring the current composition time of a composition video/audio element and make sure that said current composition time is in conformity with the associated composition time prediction.

64. Composition method according to claim 62, wherein if said current composition time is not in conformity with the associated composition time prediction, then said composition supervisor means uses at least one further degradation algorithm defined in claim 61.

65. Composition method according to claim 63 wherein if said current composition time is not in conformity with the associated composition time prediction, then said decoder supervisor means further uses said complexity information signal.

66. Composition method according to any of claims 51 to 63 for a plurality of audio/video composition elements wherein when the composition time prediction is the sum of the composition prediction time of each of said audio/video composition elements, said composition time prediction being defined in claim 50.

67. Composition system for an uncompressed digital video element including video frames and/or video planes and/or video objects, textures, texts, 2 and 3 dimensional graphic primitives, 2 and 3 dimensional models, comprising :

- a video compositor receiving said video element on one input terminal, said compositor being intended to compose said video element using a composition syntax using a plurality of composing algorithms, and to provide, on one output terminal, a representation in a common video representation space said video element ;
- means for determining statistic of use of at least one of said composition algorithms,
- and means for generating a digital statistic signal representative of said statistics for each of said composed video element.

68. Composition system for an uncompressed digital audio element including audio packets, audio objects, synthetic audio objects, comprising :

- a audio compositor receiving said audio element on one input terminal, said compositor being intended to compose said audio element using a composition syntax using a plurality of composing algorithms, and to provide, on one output terminal, a representation in a common audio representation space said audio element ;
 - means for determining statistic of use of at least one of said composition algorithms,
 - and means for generating a digital statistic signal representative of said statistics for each of said composed audio element.
69. Composition system for an audio-visual object comprising a composition system according to claim 66 combined with a composition system according to claim 67.

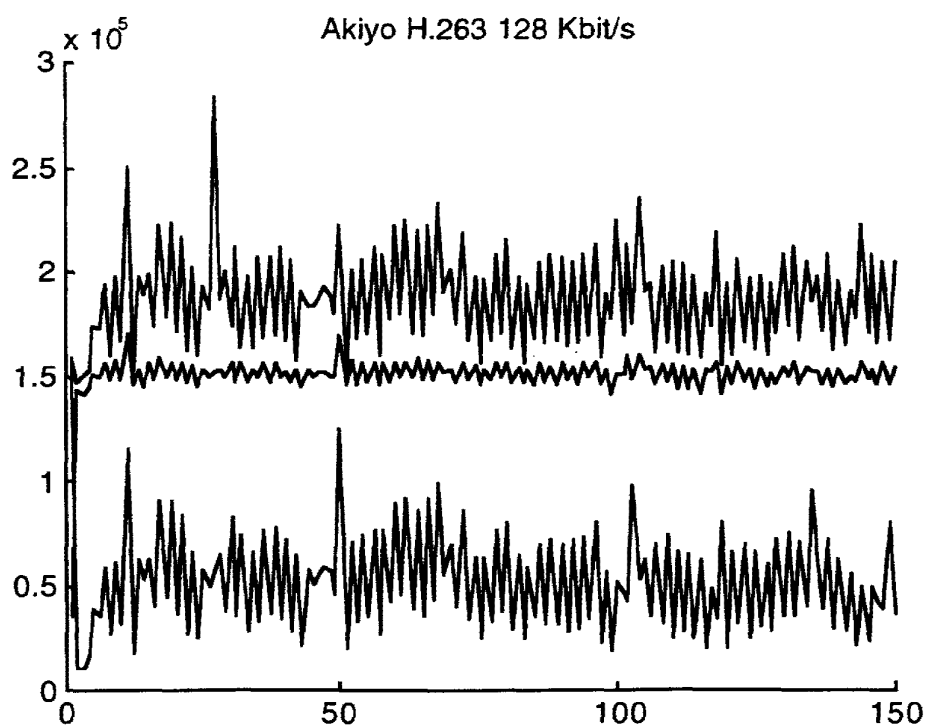


Figure 1

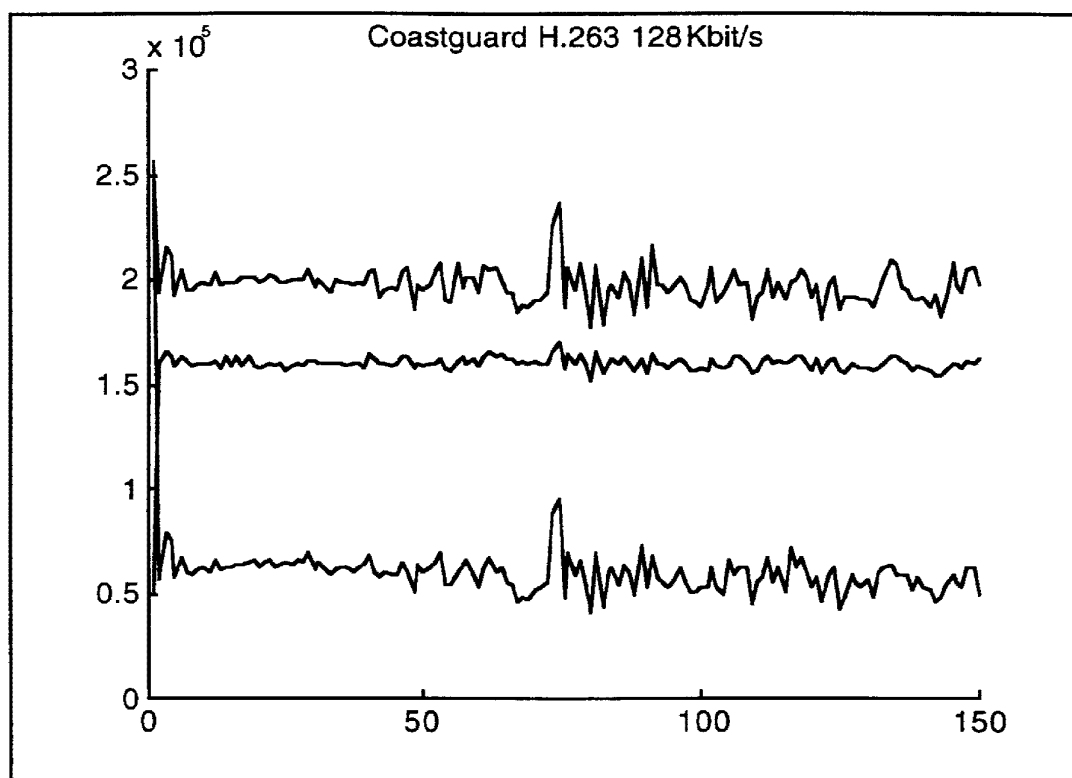


Figure 2

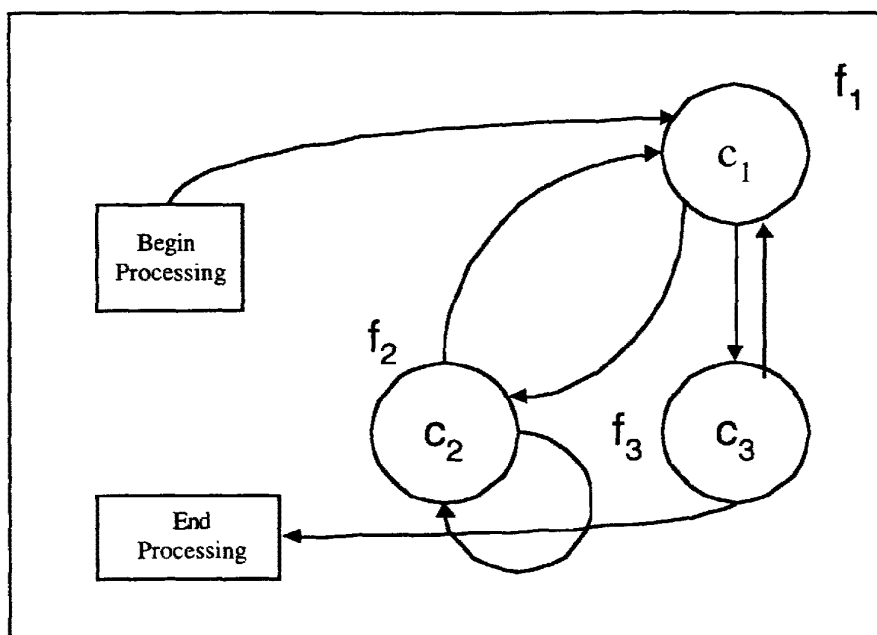


Figure 3

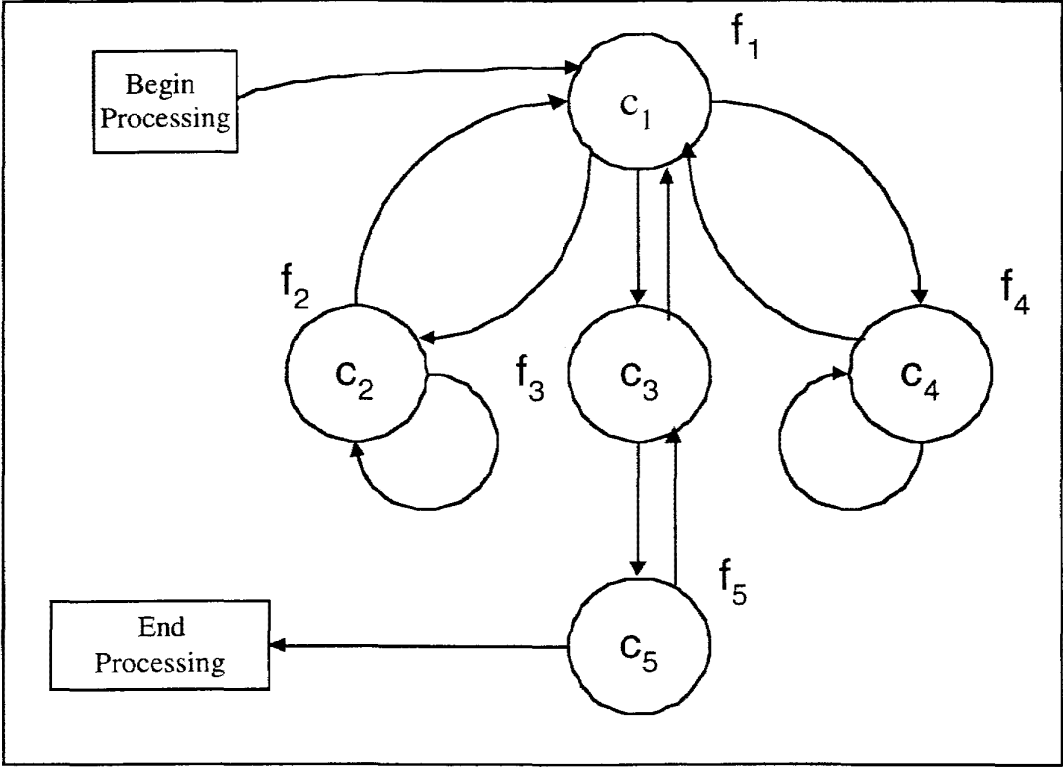


Figure 4

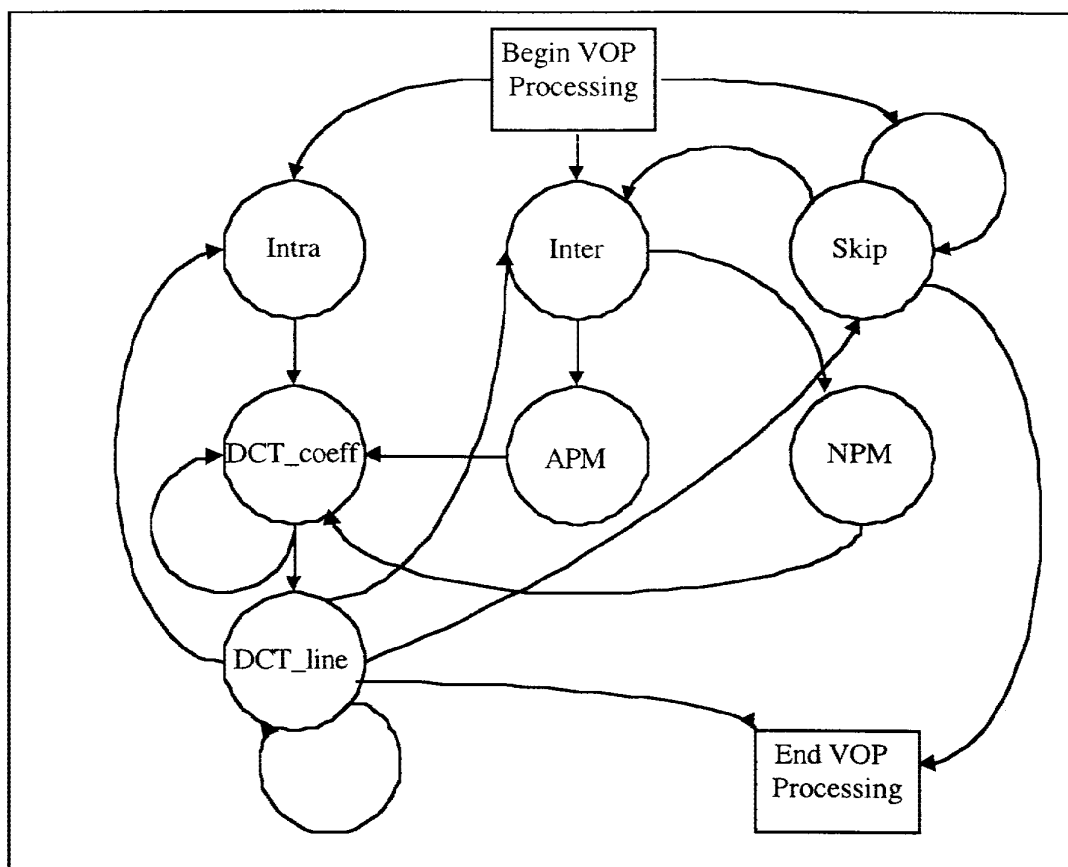


Figure 5

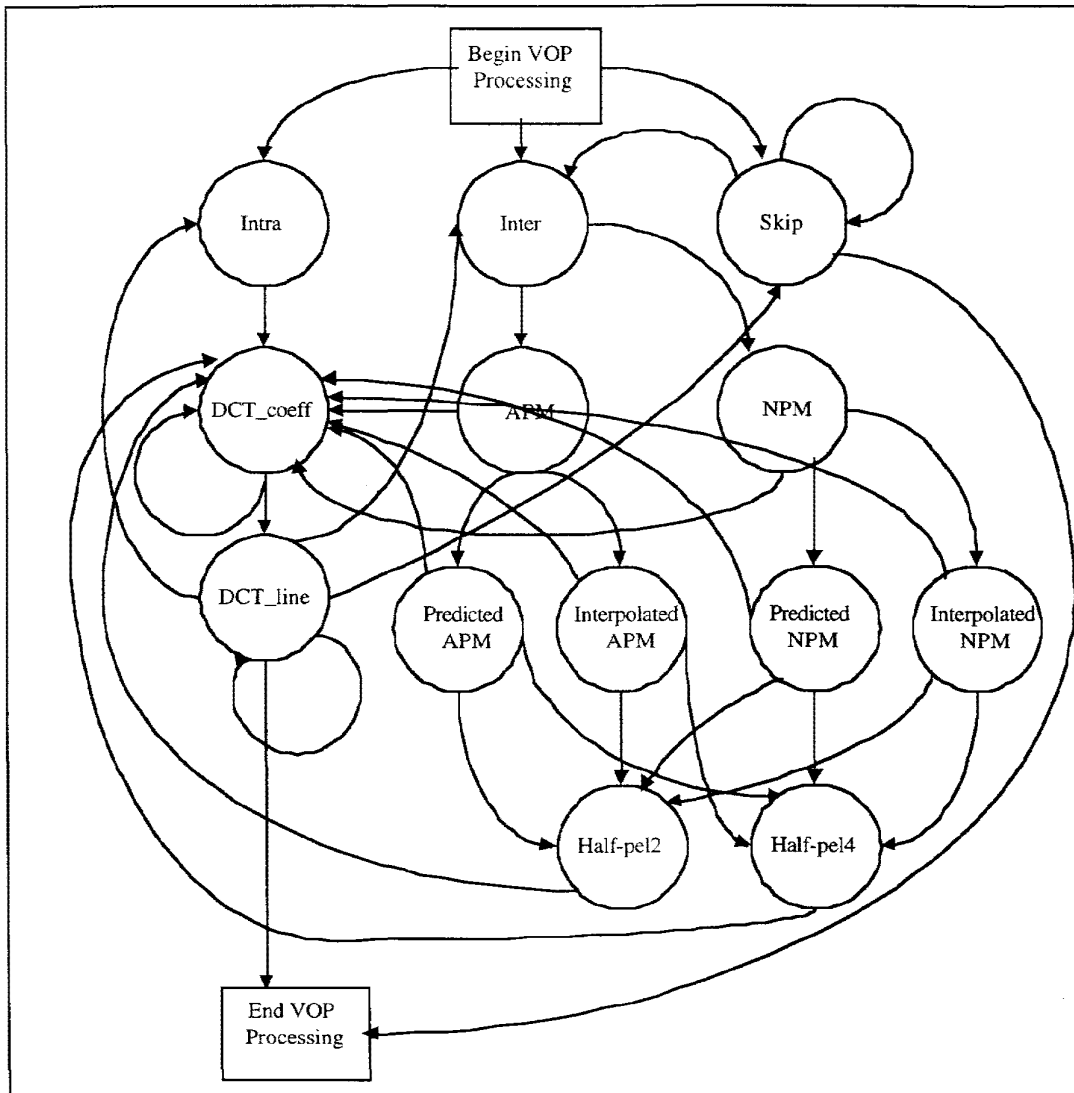


Figure 6

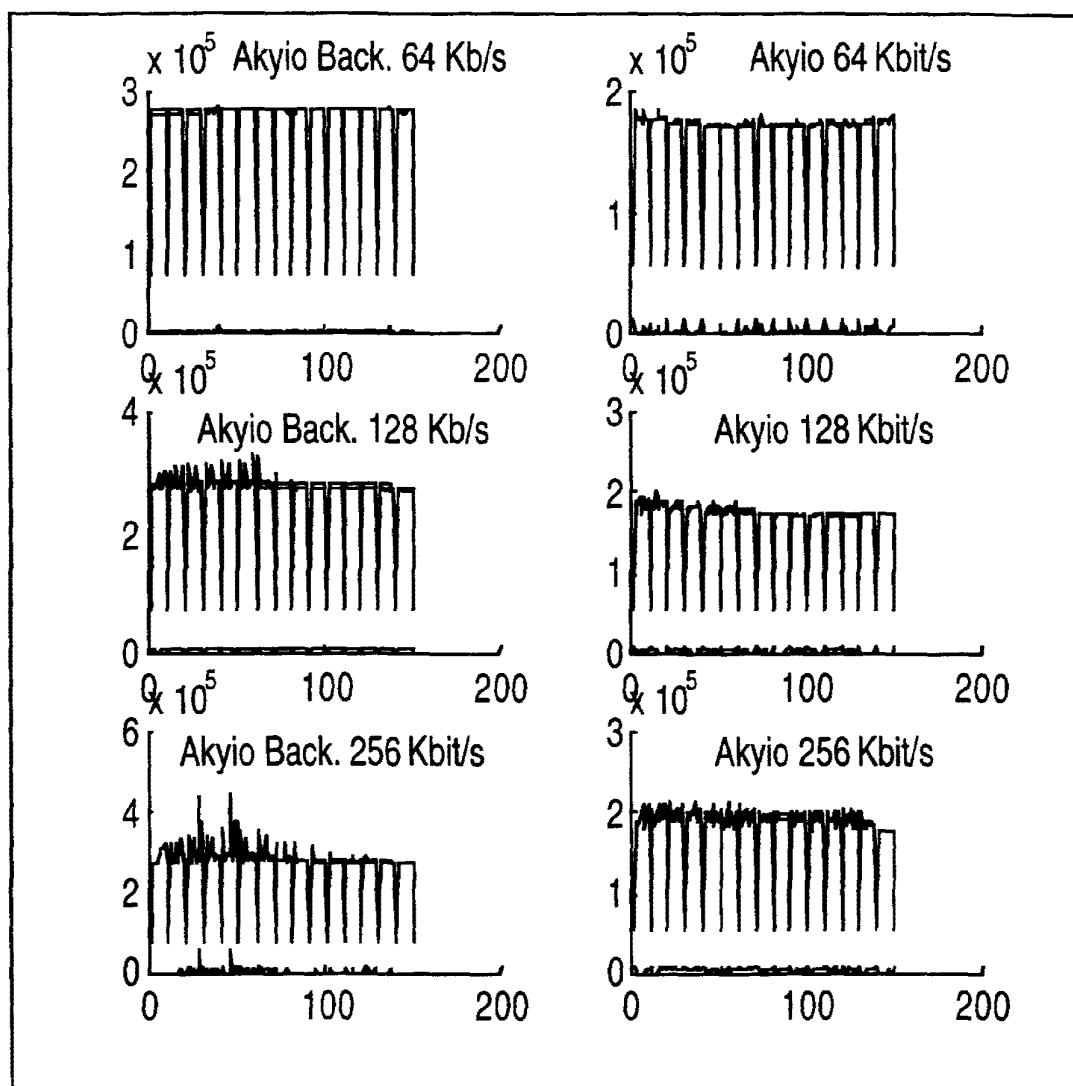


Figure 7

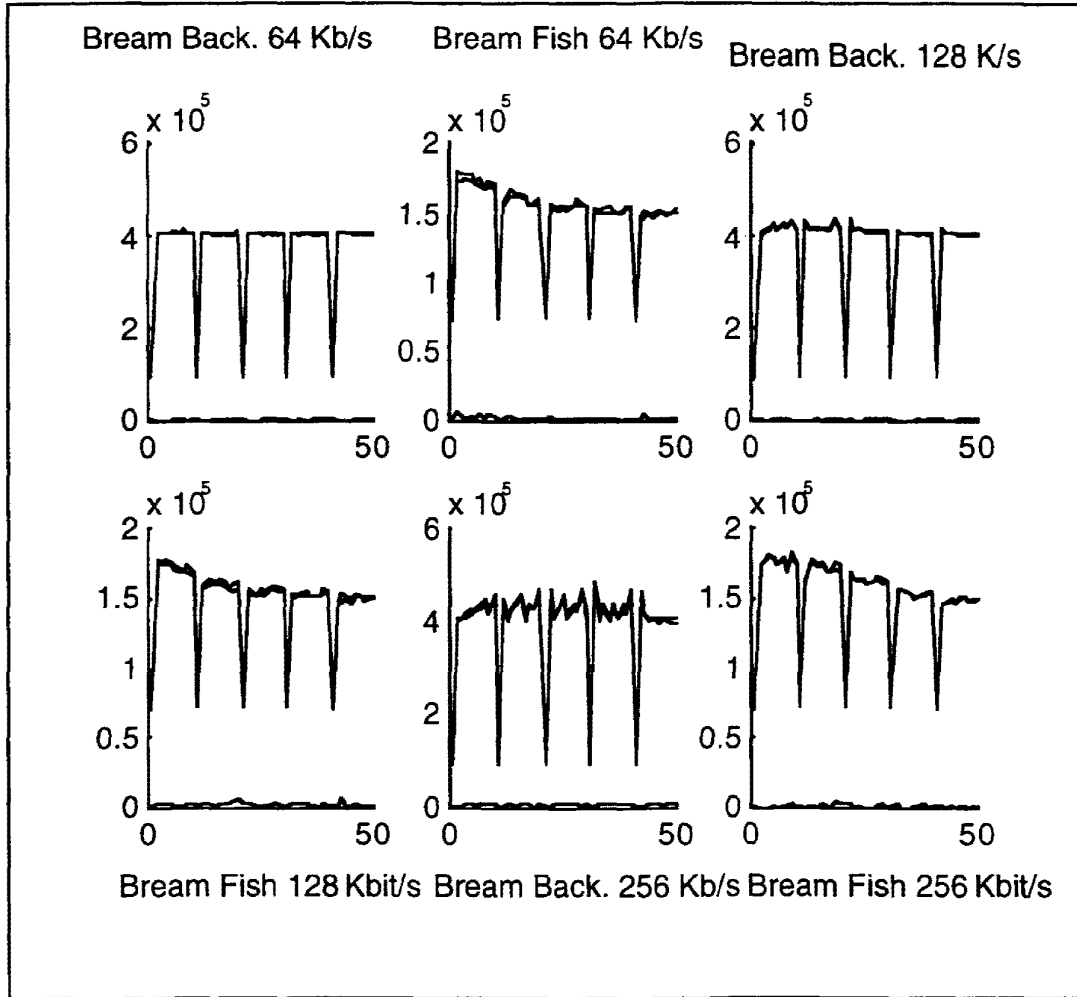


Figure 8

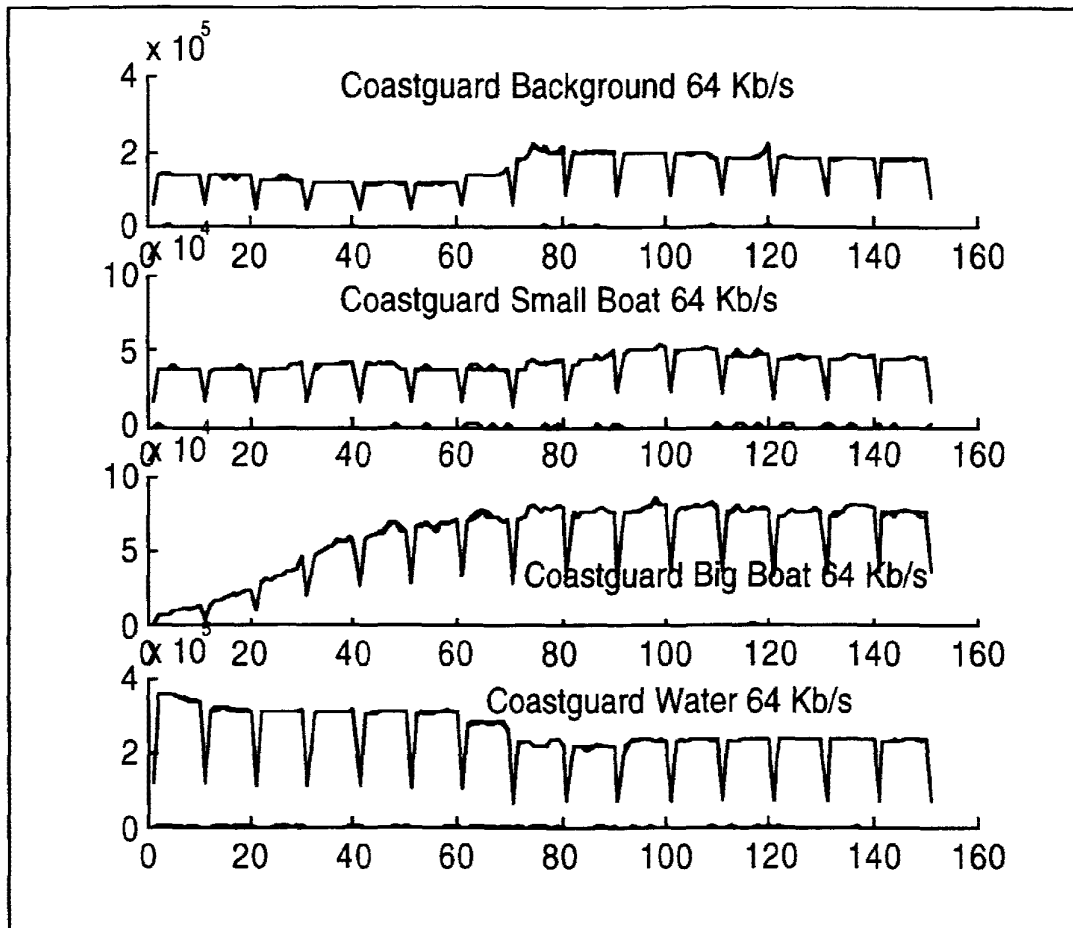


Figure 9

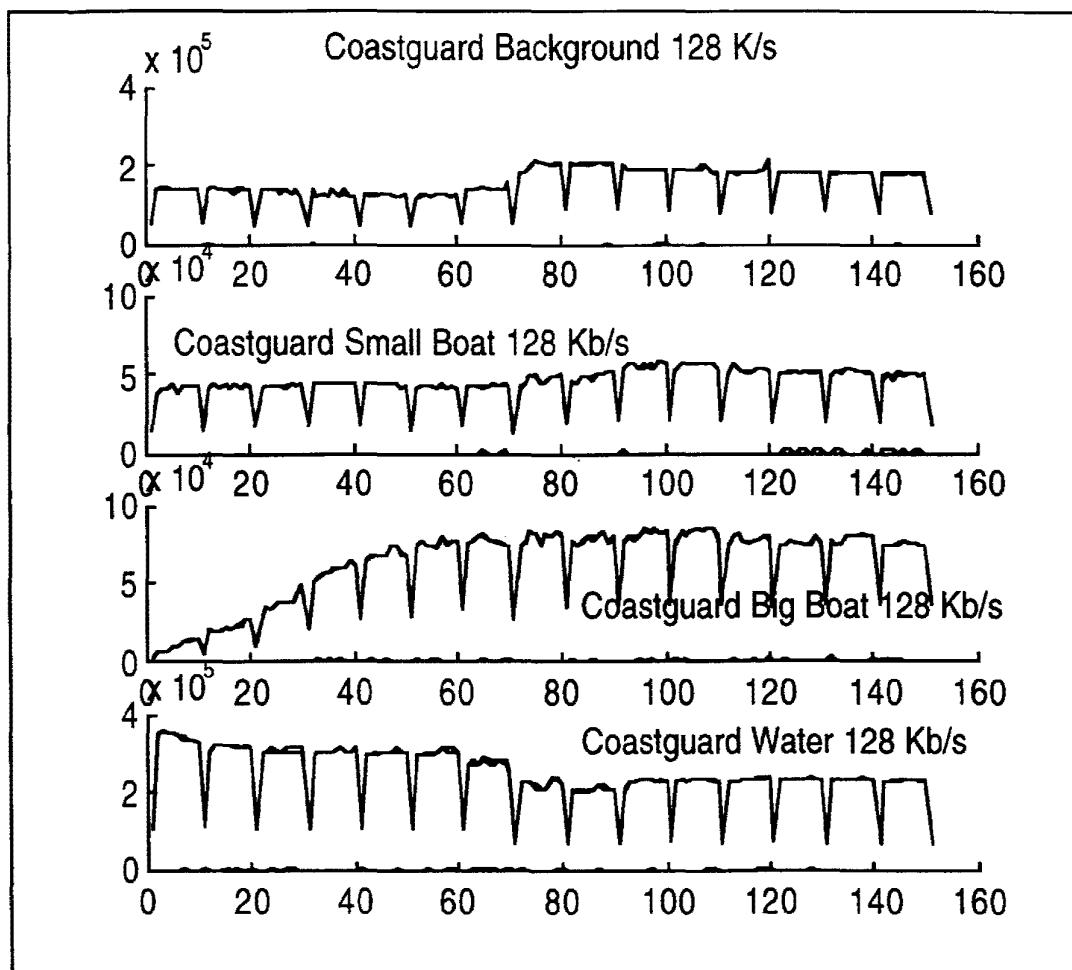


Figure 10

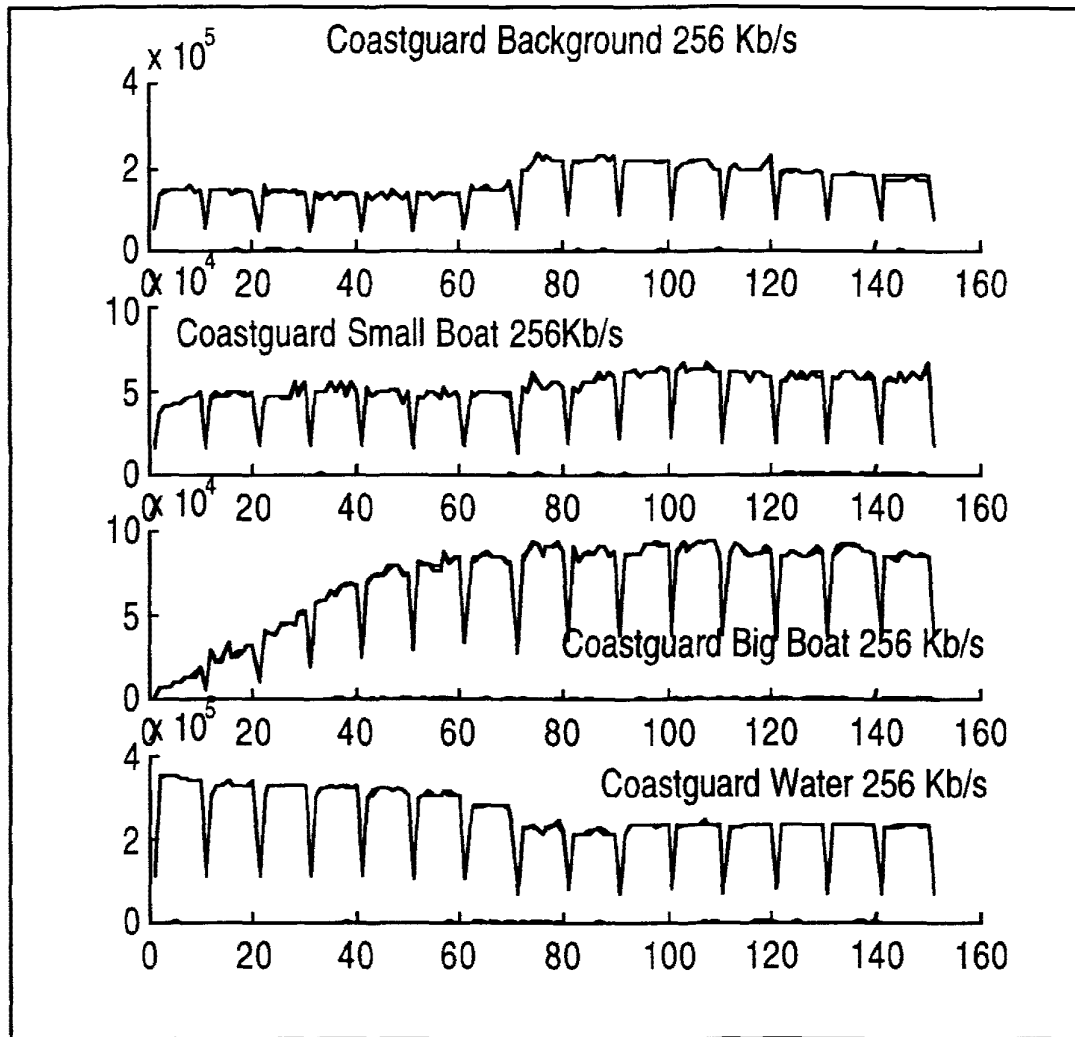


Figure 11

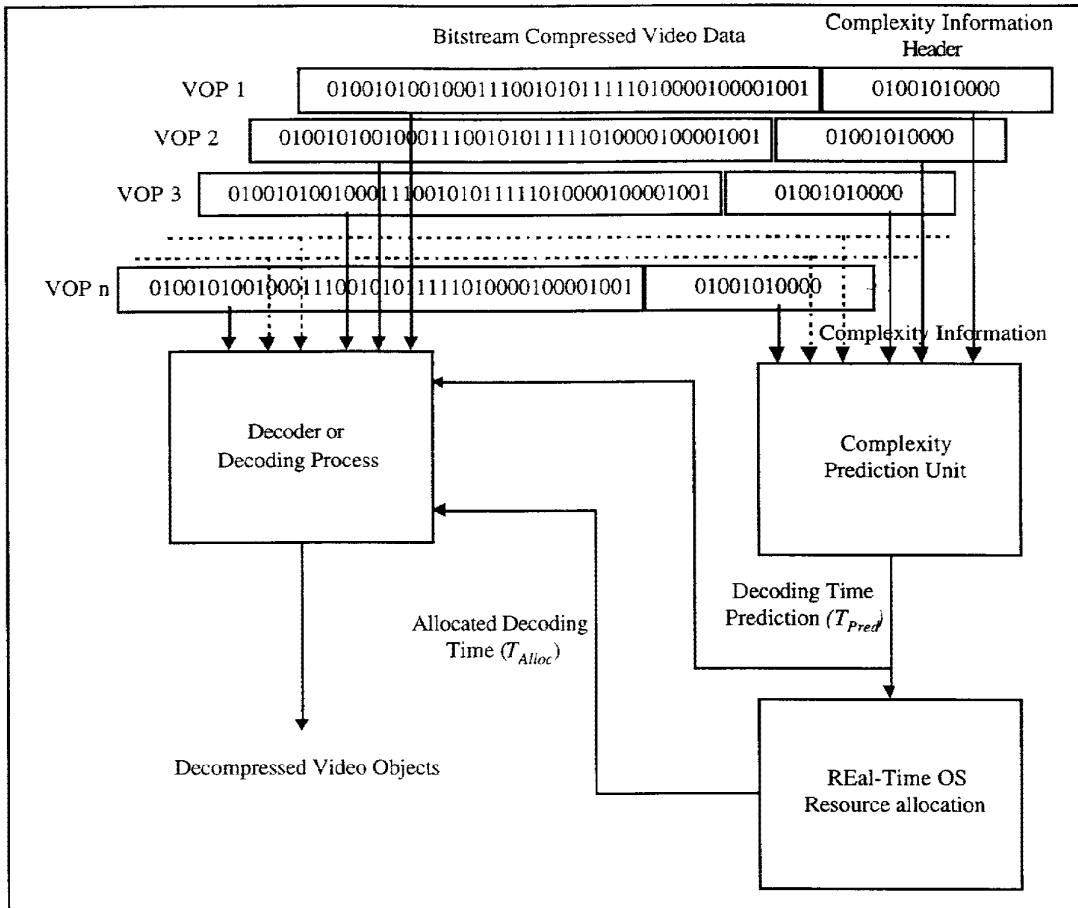


Figure 12

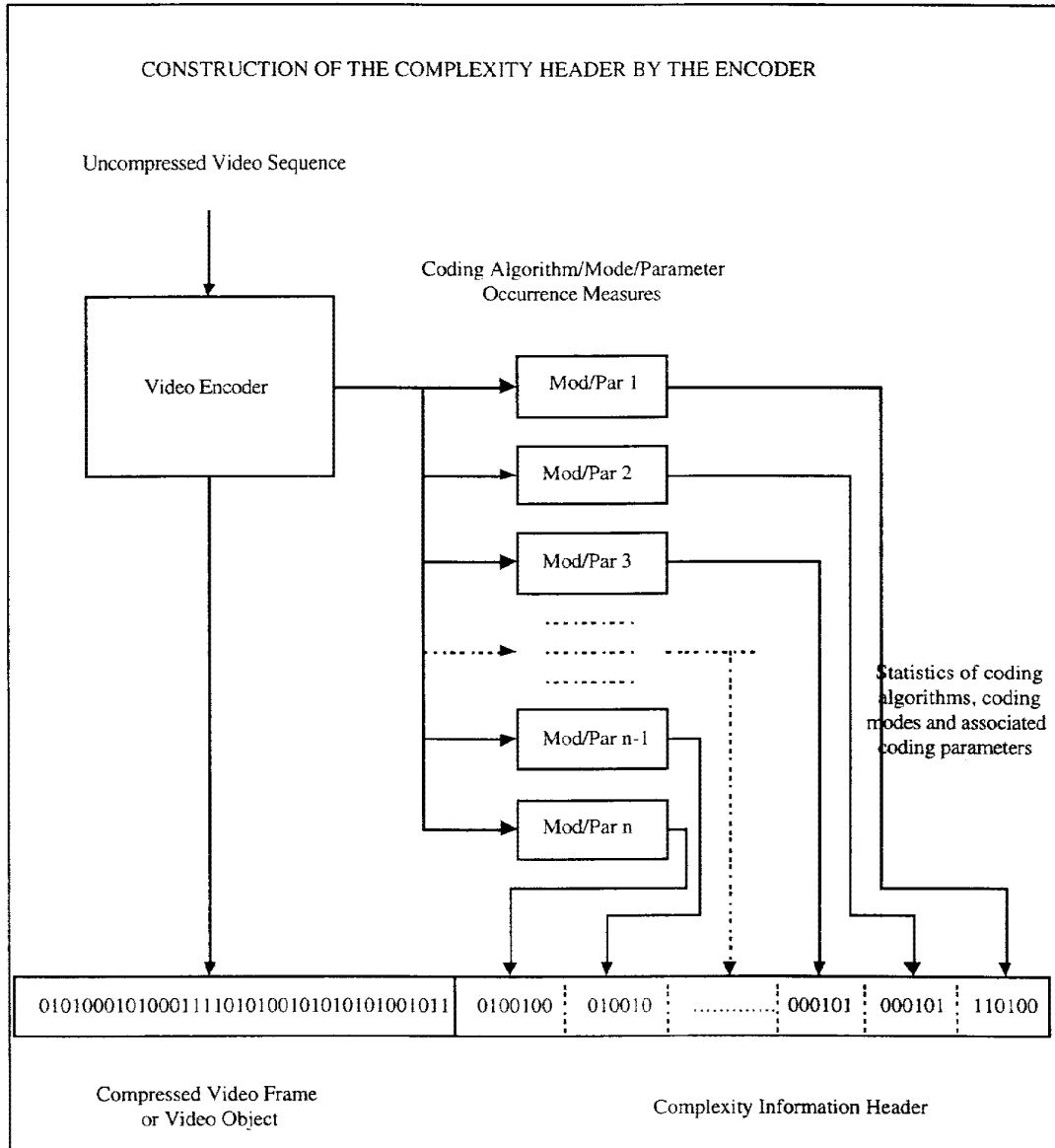


Figure 13

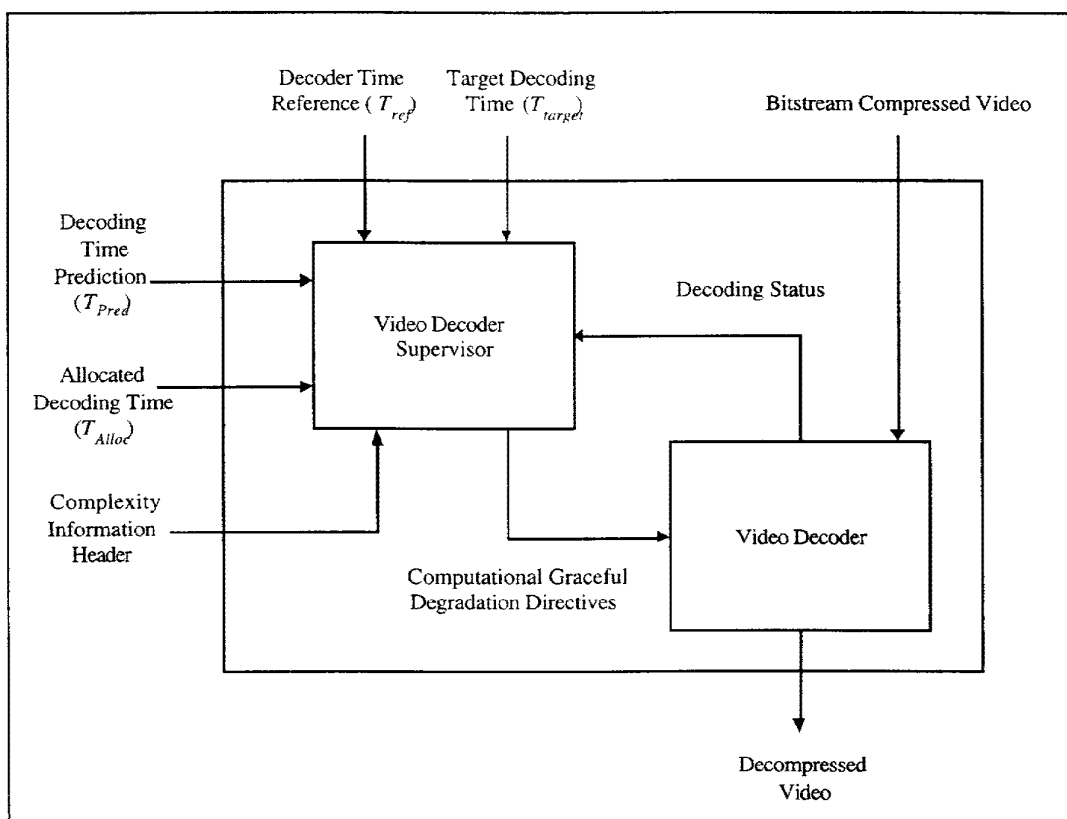


Figure 14

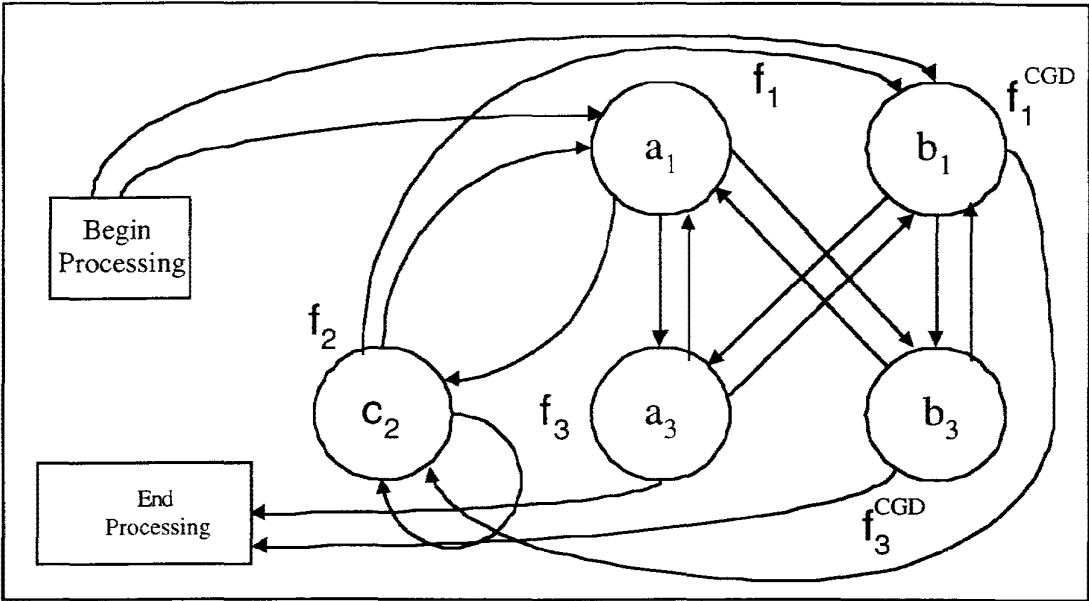


Figure 15

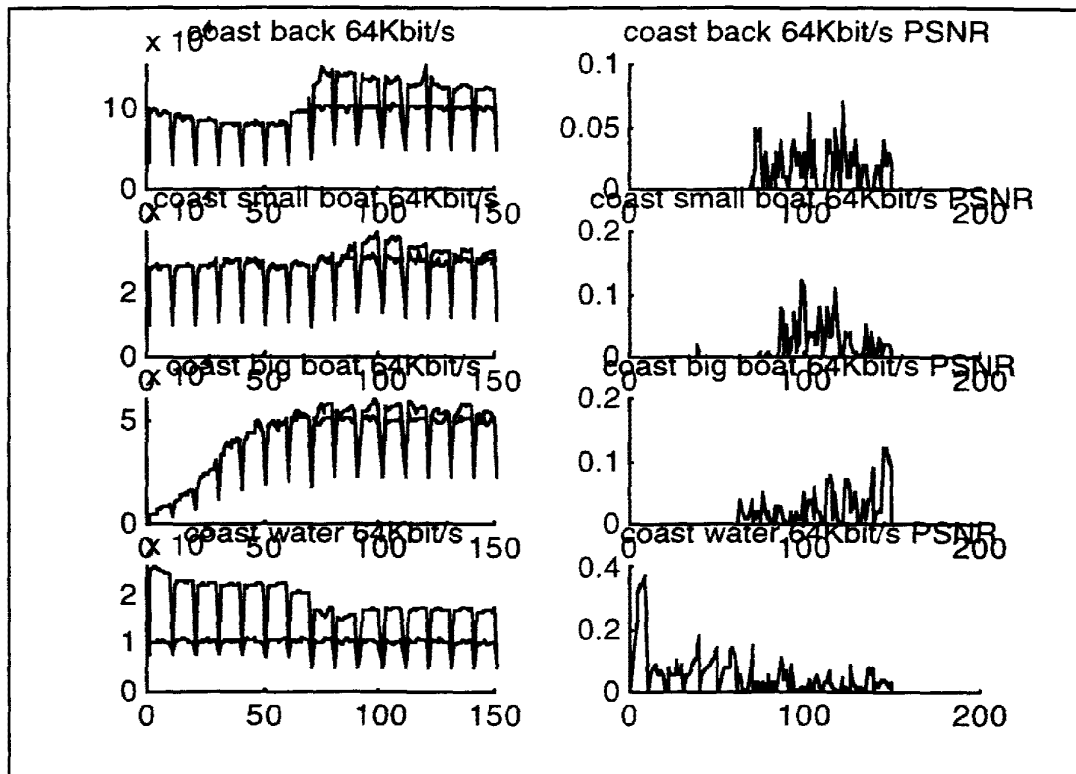


Figure 16

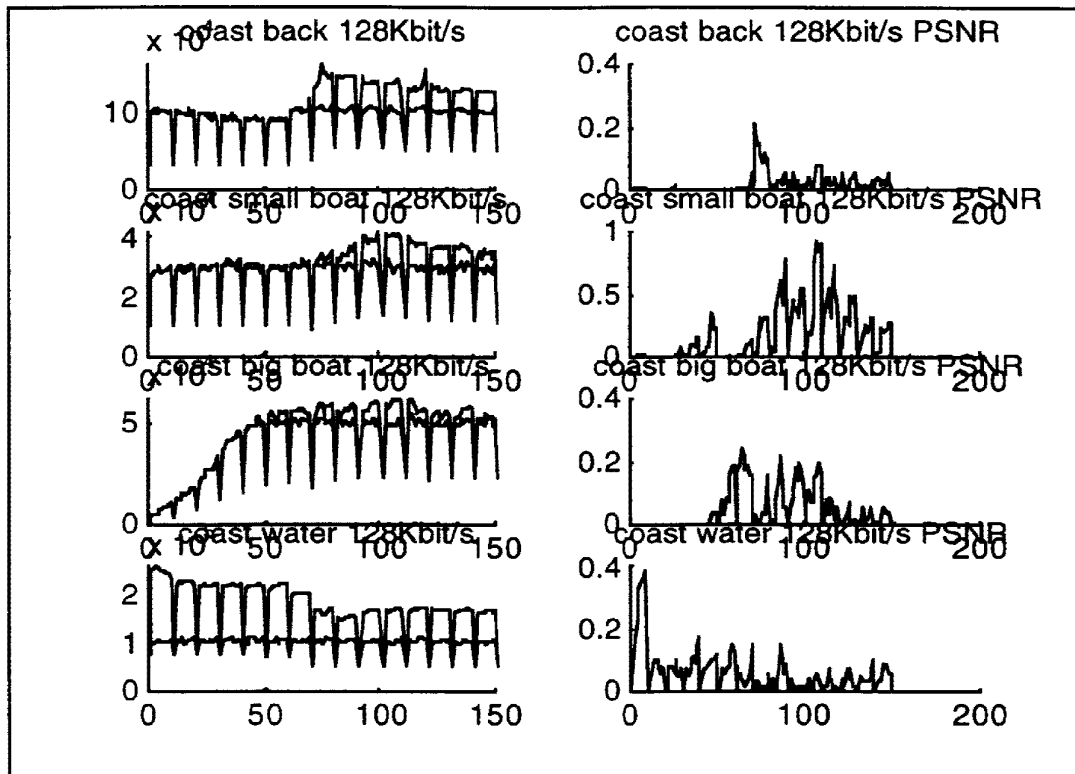


Figure 17

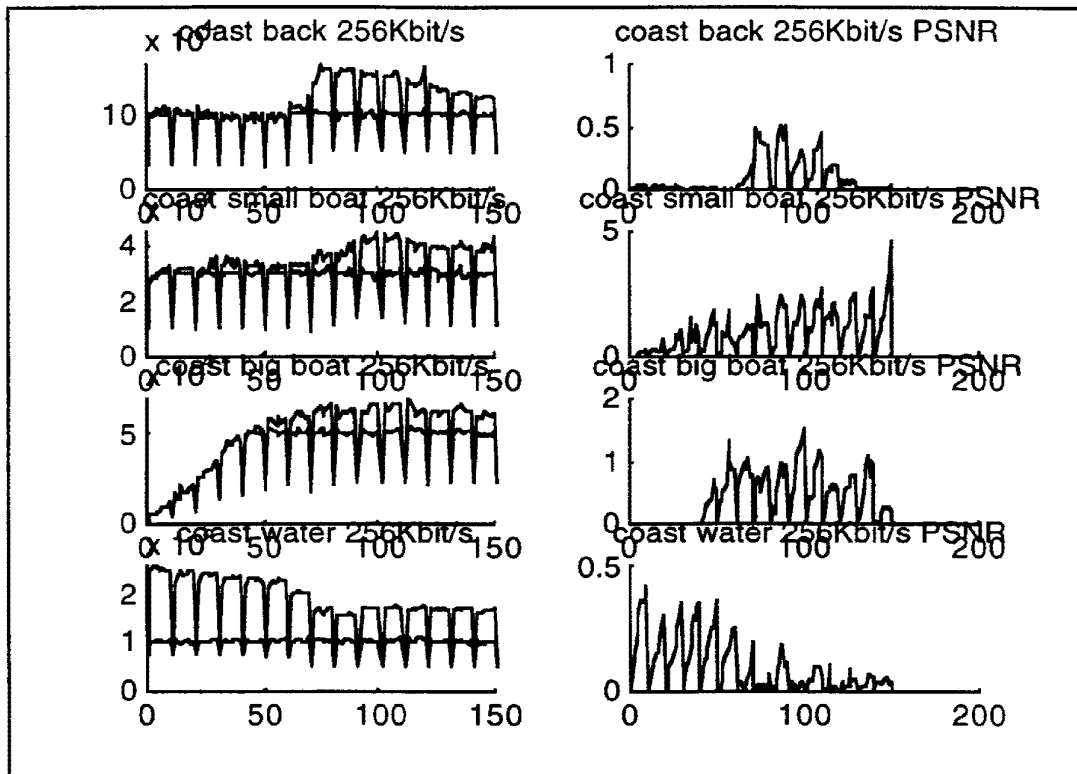


Figure 18

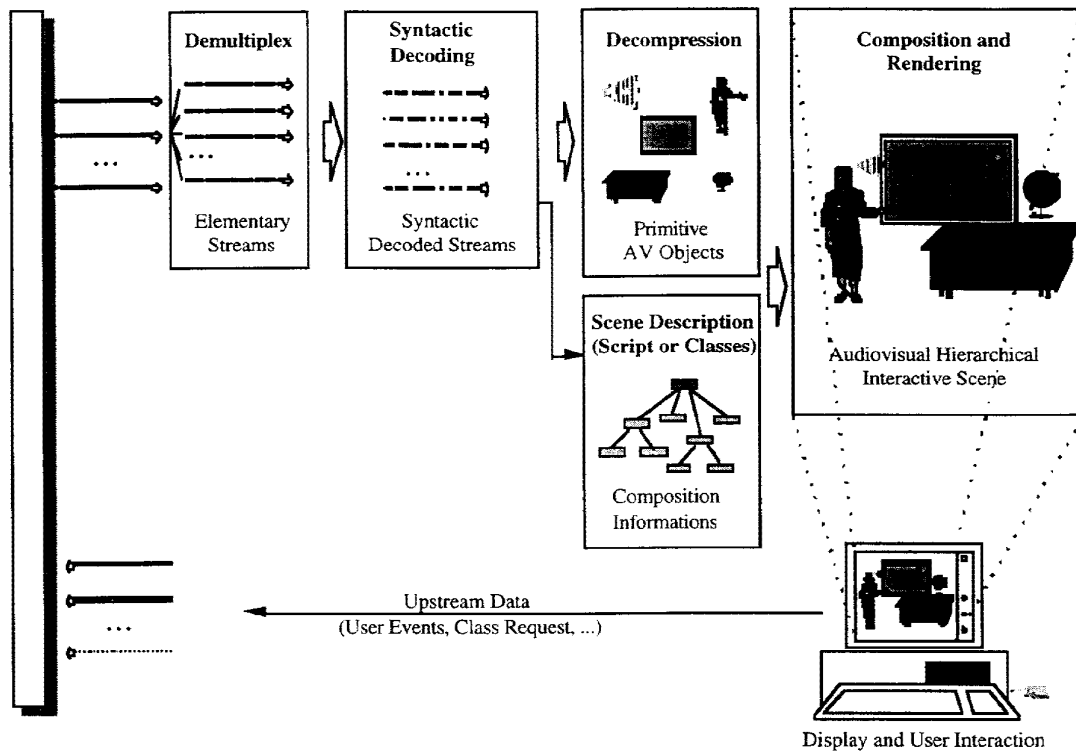


Figure 19

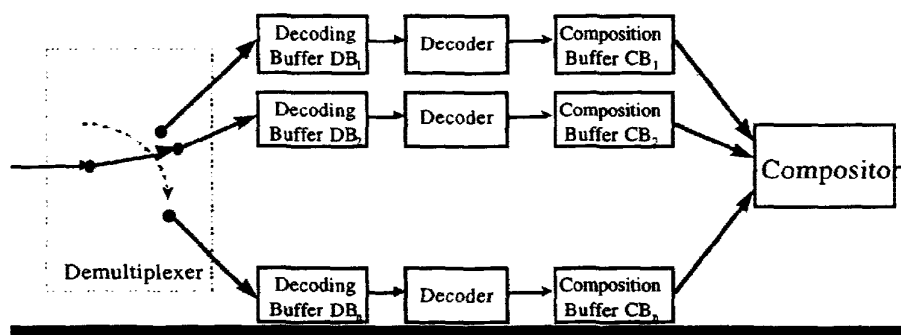


Figure 20

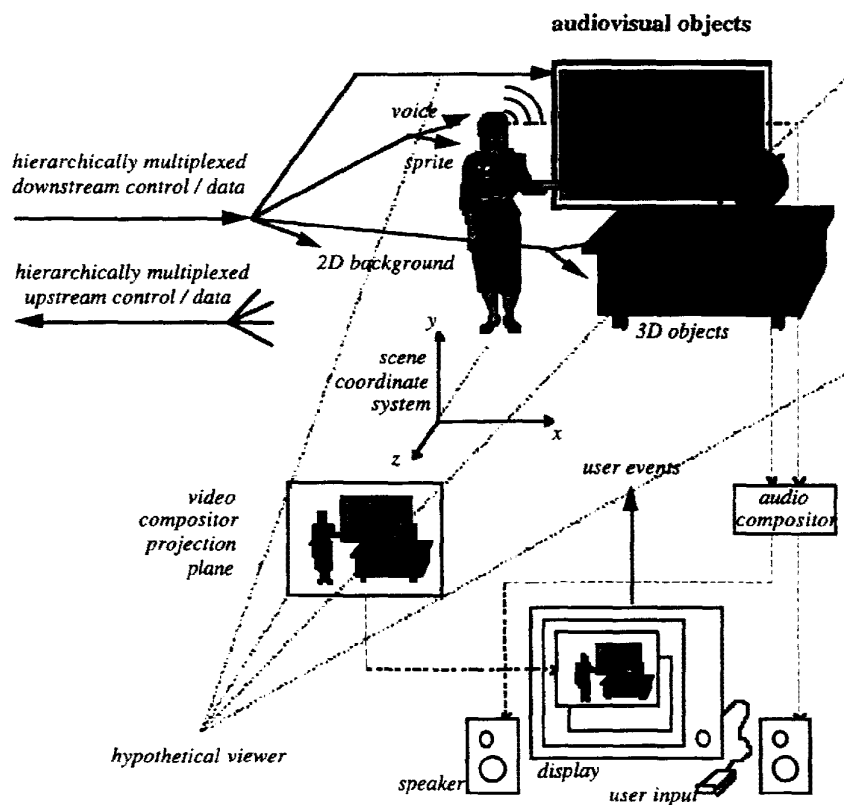


Figure 21



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 98 10 9423

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.6)
E	EP 0 866 426 A (ALSTHOM CGE ALCATEL) 23 September 1998 * abstract; claims 1,5,6 *	1-68	G06T9/00
A	EP 0 512 174 A (SEMAPHORE INC) 11 November 1992 * page 2, line 40 - line 43 * * page 3, line 32 - line 36 *	1-68	
A	MATTAVELLI M ET AL: "Computational graceful degradation for video sequence decoding" PROCEEDINGS. INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (CAT. NO.97CB36144), PROCEEDINGS OF INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, SANTA BARBARA, CA, USA, 26-29 OCT. 1997, pages 330-333 vol.1, XP002080307 ISBN 0-8186-8183-7, 1997, Los Alamitos, CA, USA, IEEE Comput. Soc, USA		
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)
			G06T
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 12 October 1998	Examiner Pierfederici, A
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 02 (P04C01)

(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.

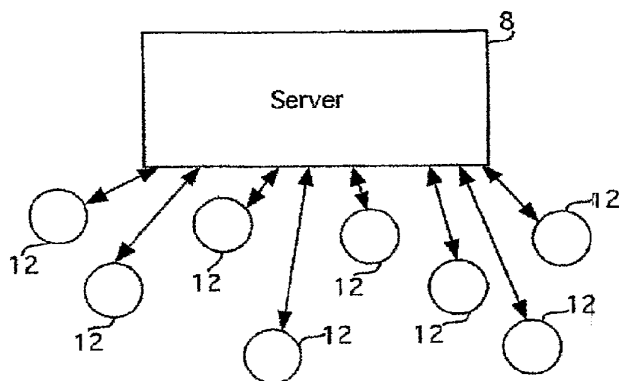
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, 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 NapsterTM music file exchange system provided by Napster, Inc. of Redwood City, Calif. A schematic of the NapsterTM music file exchange system (as its operation is presently understood) is illustrated in FIG. 2.

More particularly, it is believed that in the NapsterTM 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 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:

- $\text{Utility Rating} = \text{Time} \times \text{Performance Rating} \times \text{Bandwidth Rating}.$

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:

- $\text{Potential Re-transmission Rating} = \text{Performance Rating} \times \text{Bandwidth Rating}.$

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 upree) 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 go 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 up tree) 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

- o E (a third level node with available capacity)
 - o J (the fourth level node with available capacity having the highest utility rating)
- o K (a fourth level node with available capacity)

The SRPL in this example would be as follows:

- SRPL
 - o C (the fully occupied second level node with a low-bandwidth child node)
 - o D (the fully occupied third level node with a low-bandwidth child node)
 - o 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 and 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 retransmits 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 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 be 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 to 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 its 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 up the tree 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 n th 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 up the tree 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 uptime 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 dntree 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 data 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 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 first 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: $\text{Utility Rating} = \text{Time (of connection)} \times \text{Bandwidth Rating}$. 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).

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.

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 claim 27, 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:

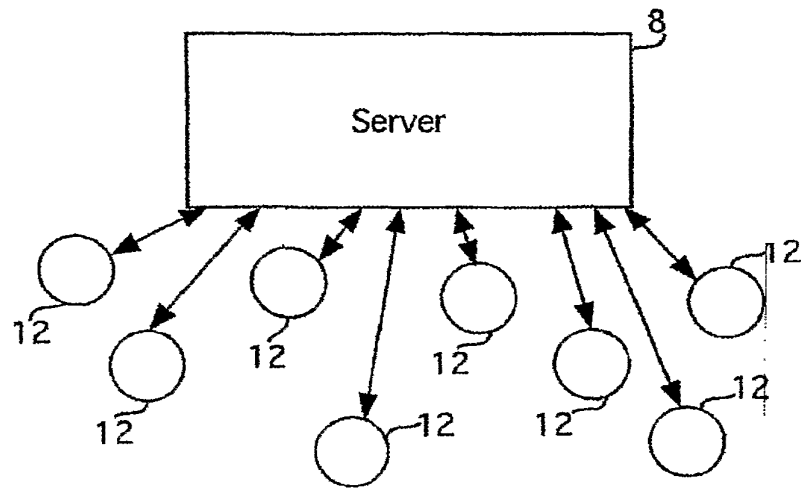
sending a first response message to the first node over a separate channel;

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;

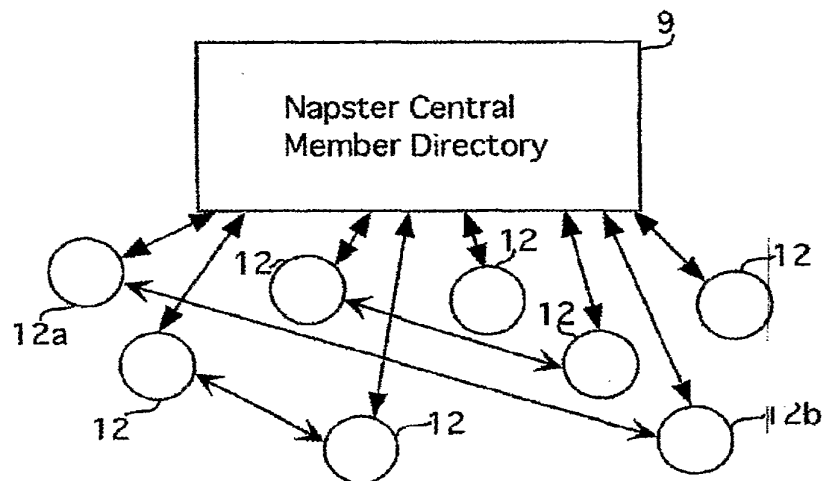
waiting a predefined period for an acknowledgement of the second response message, and if no acknowledgement is received, sending a third response message to the first node over the separate channel; and

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.

1/31



PRIOR ART

Fig. 1

PRIOR ART

Fig. 2

1 of 31

2/31

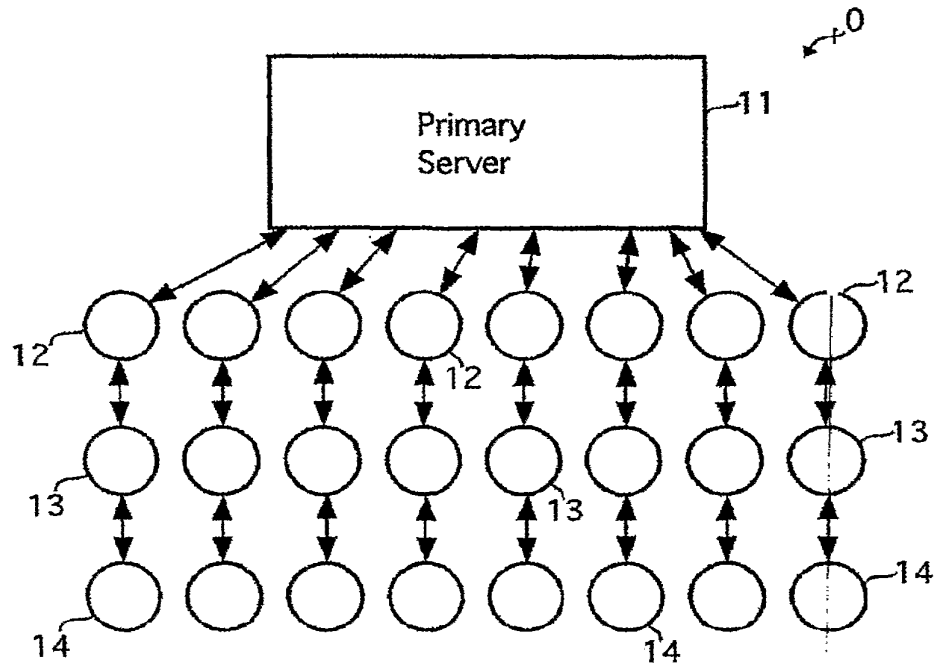


Fig. 3

Start-up
 corresponds to the
 application of the
 server

31

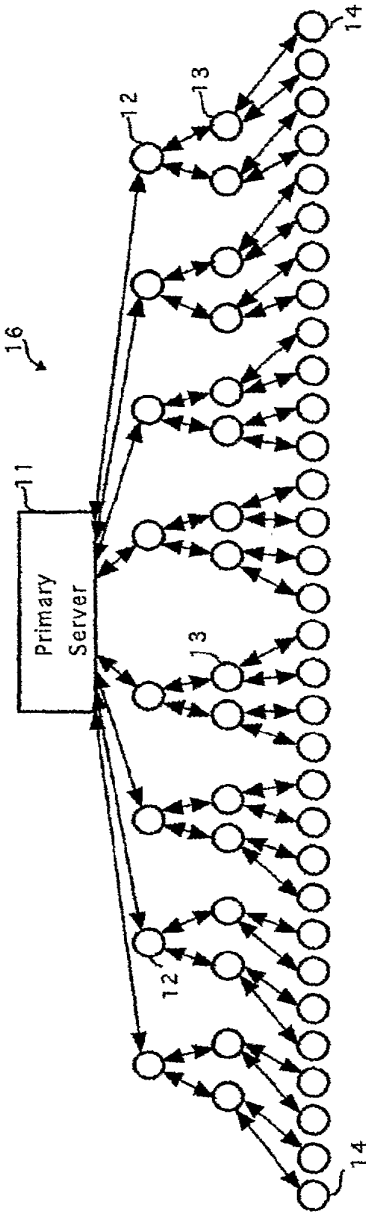


Fig. 4

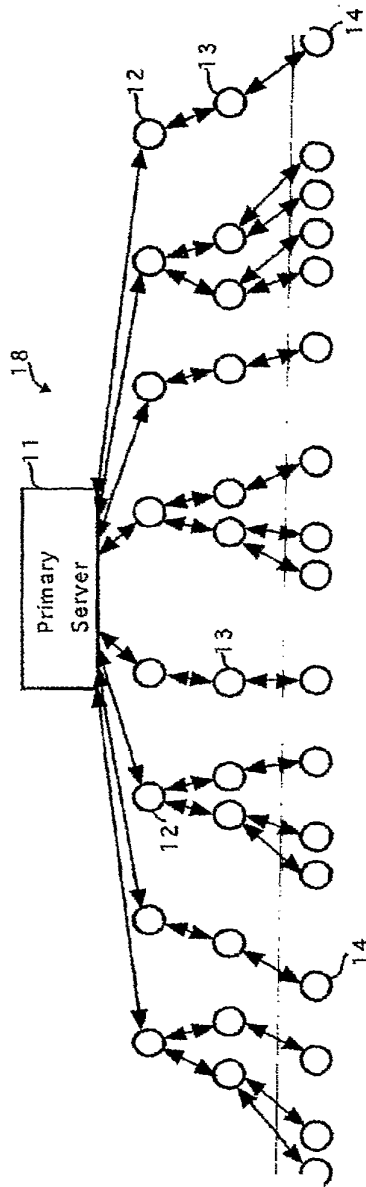


Fig. 5

4/31

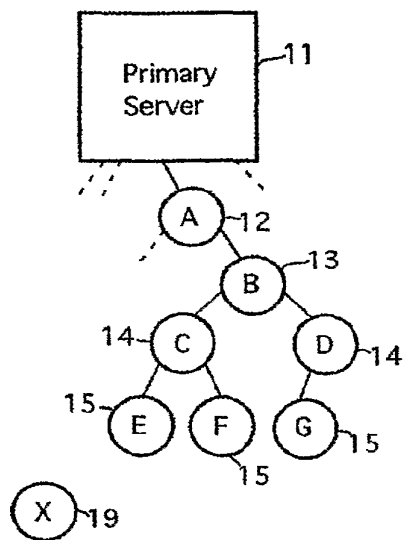


Fig. 6

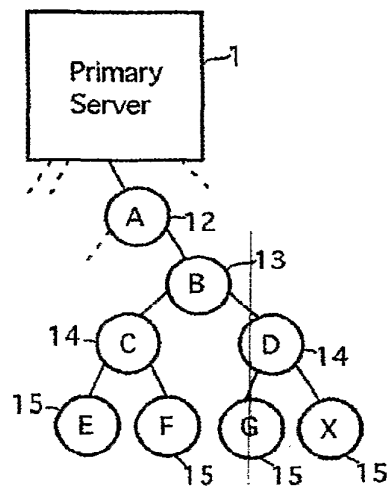


Fig. 7

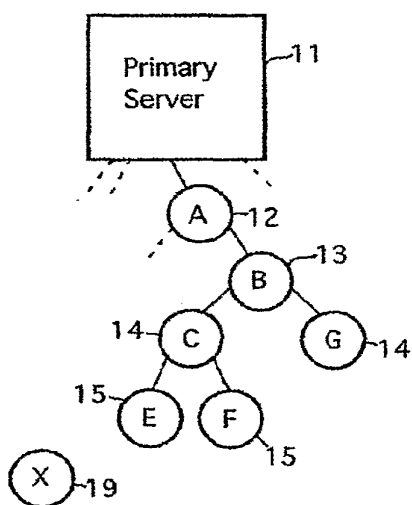


Fig. 8

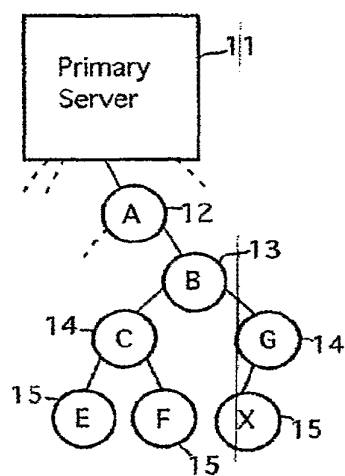
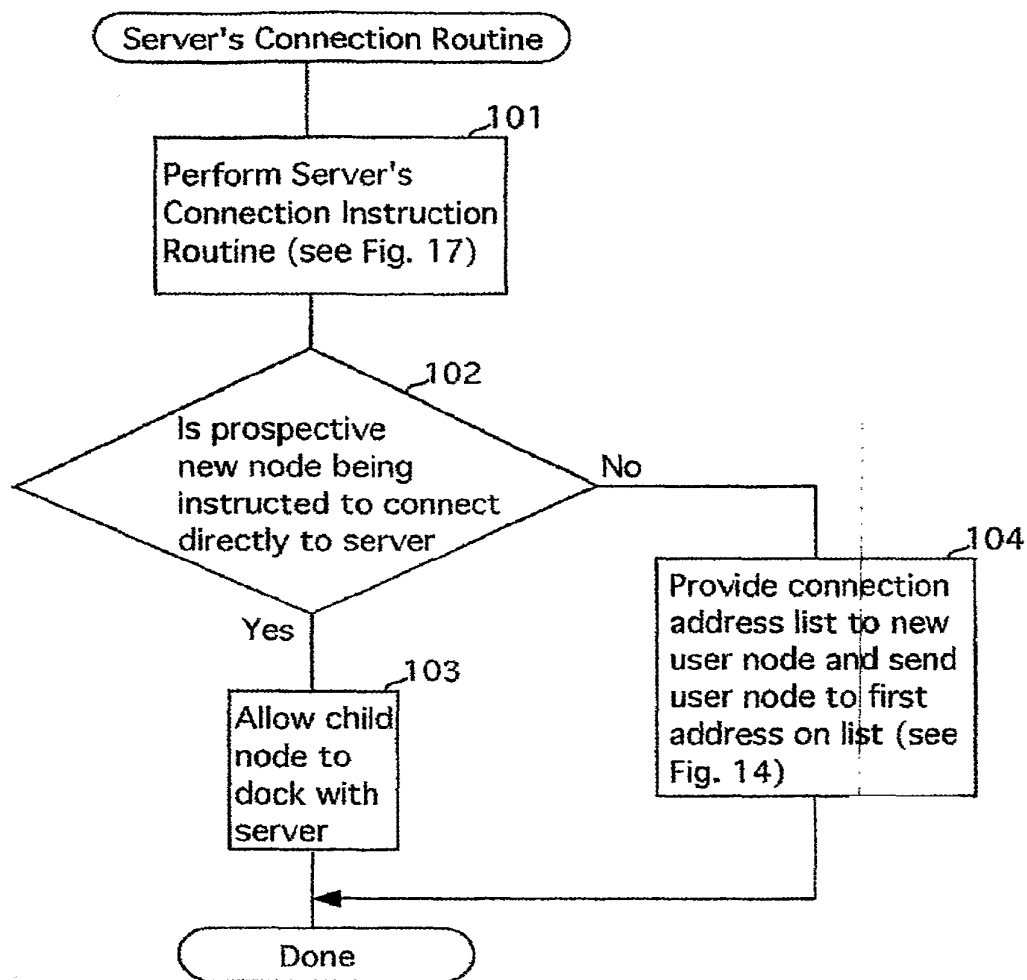


Fig. 9

5/31

**Fig. 10**

6/31

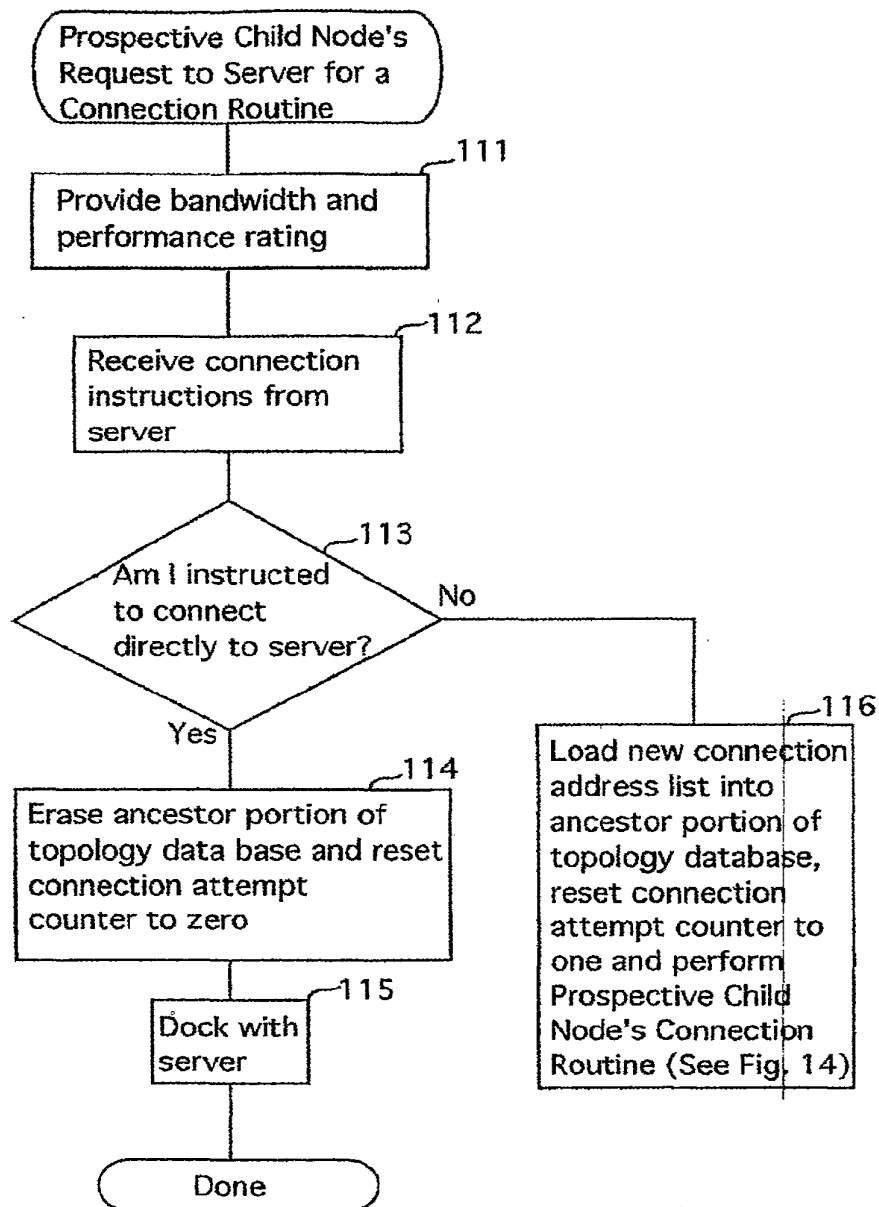
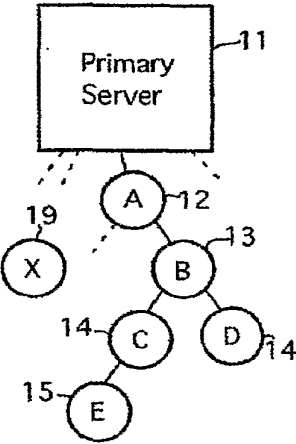
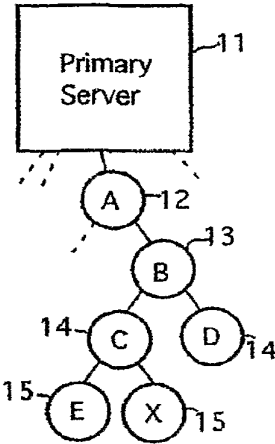
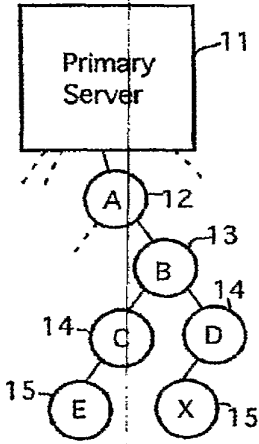
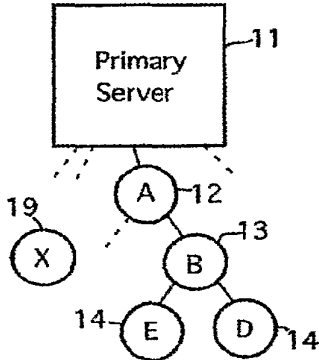
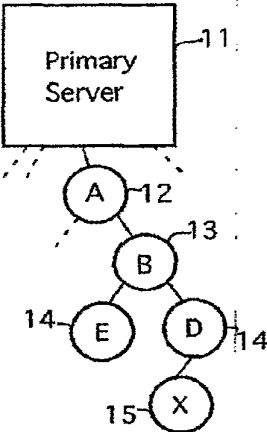


Fig. 11

6 of 31

7/31



8/31

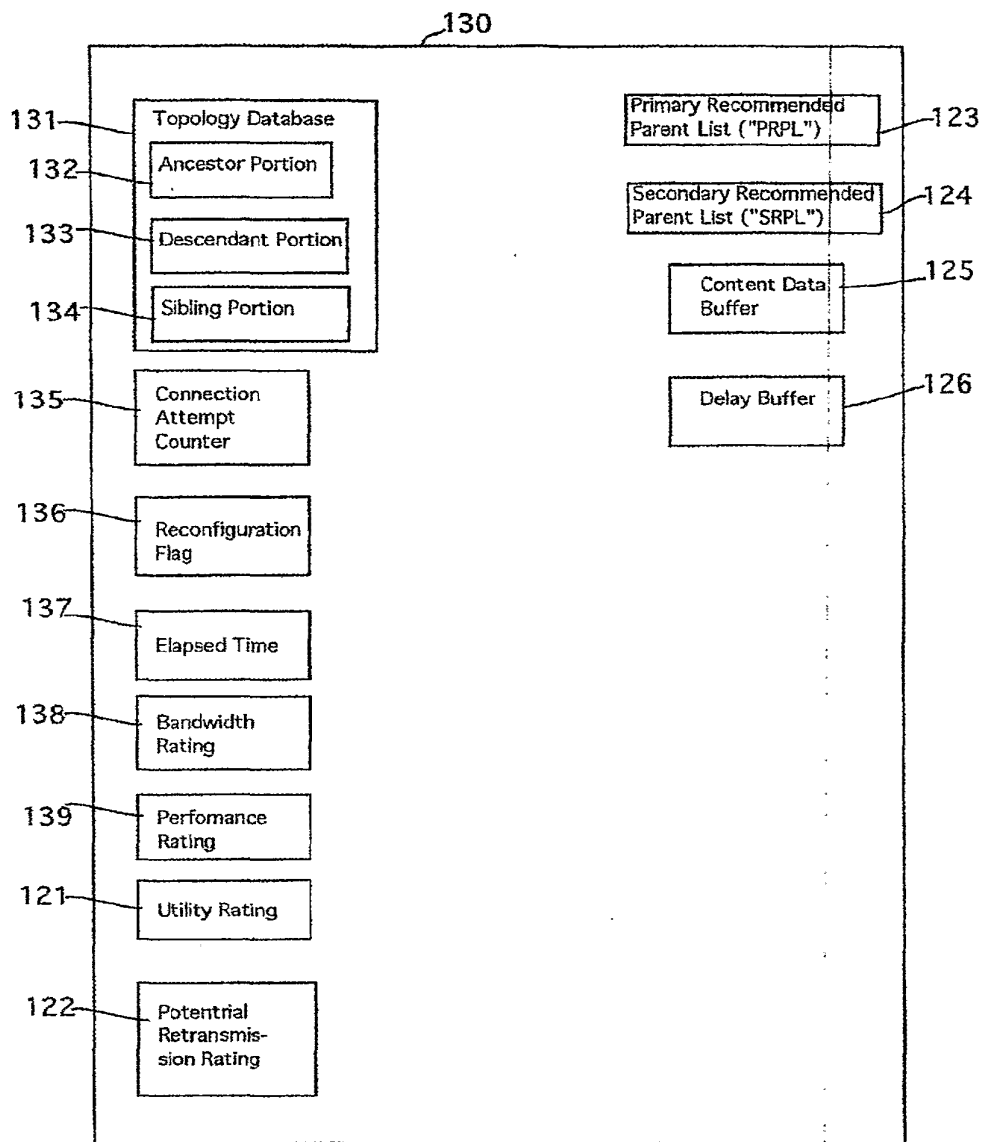


Fig. 13

9/31

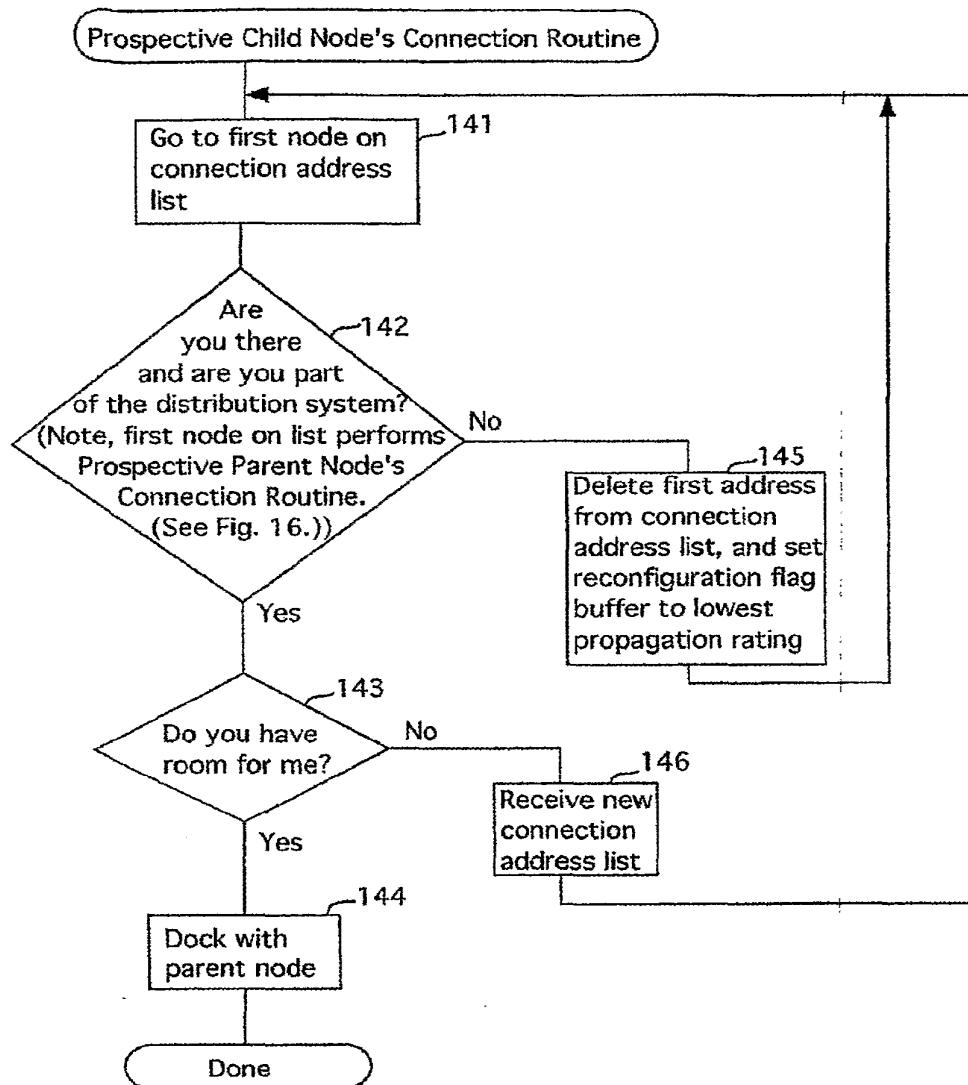


Fig. 14

10/31

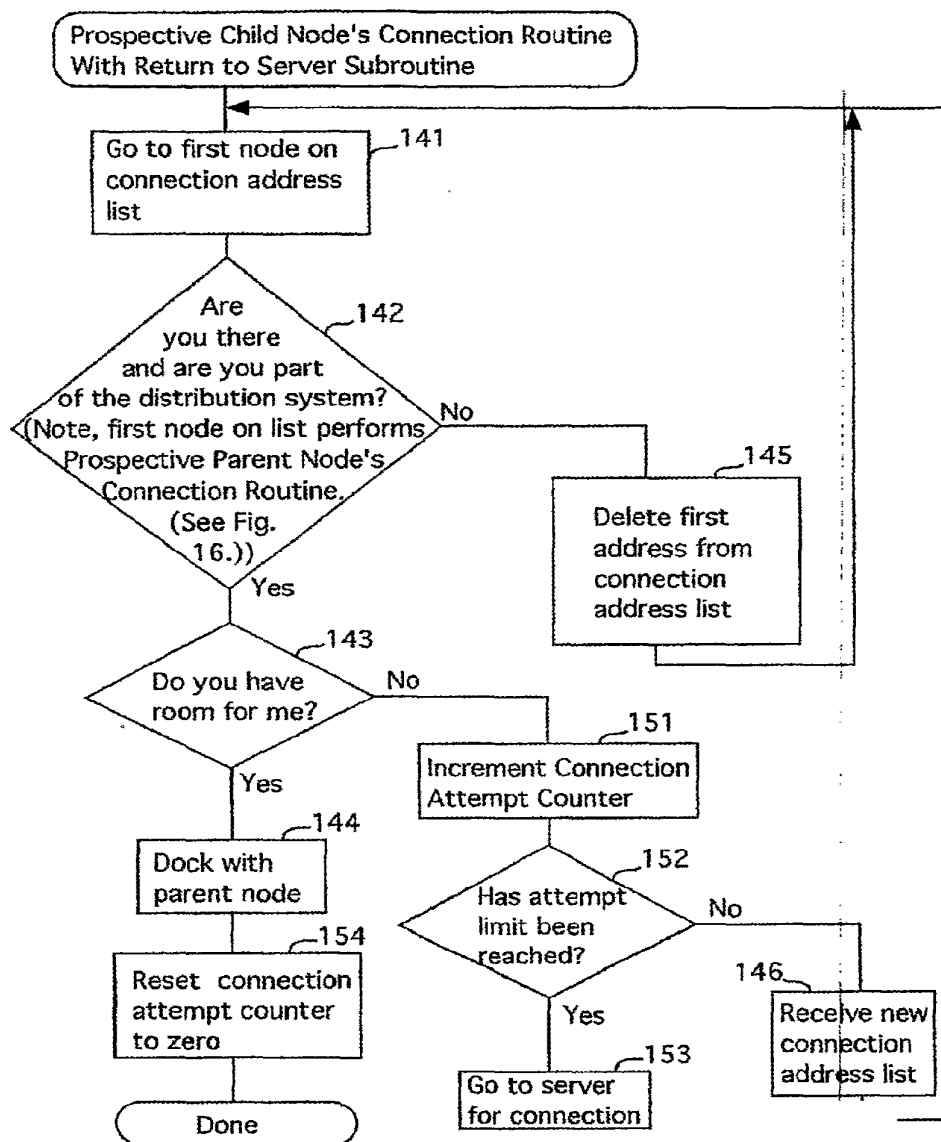
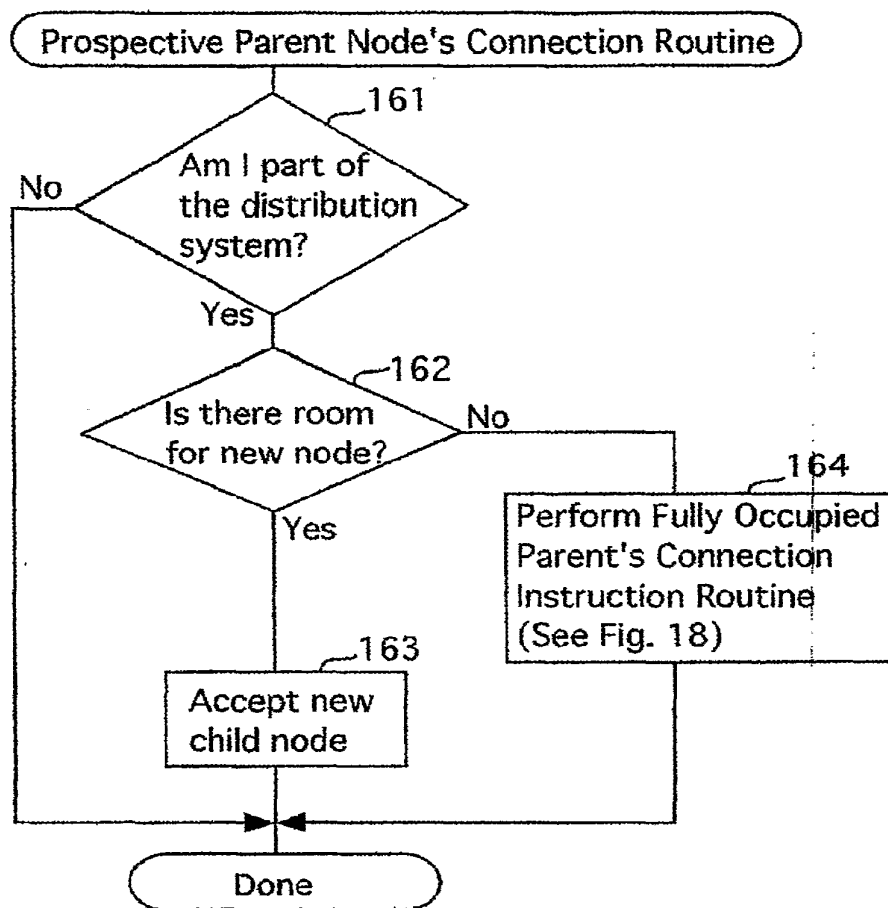


Fig. 15

11/31

**Fig. 16**

12/31

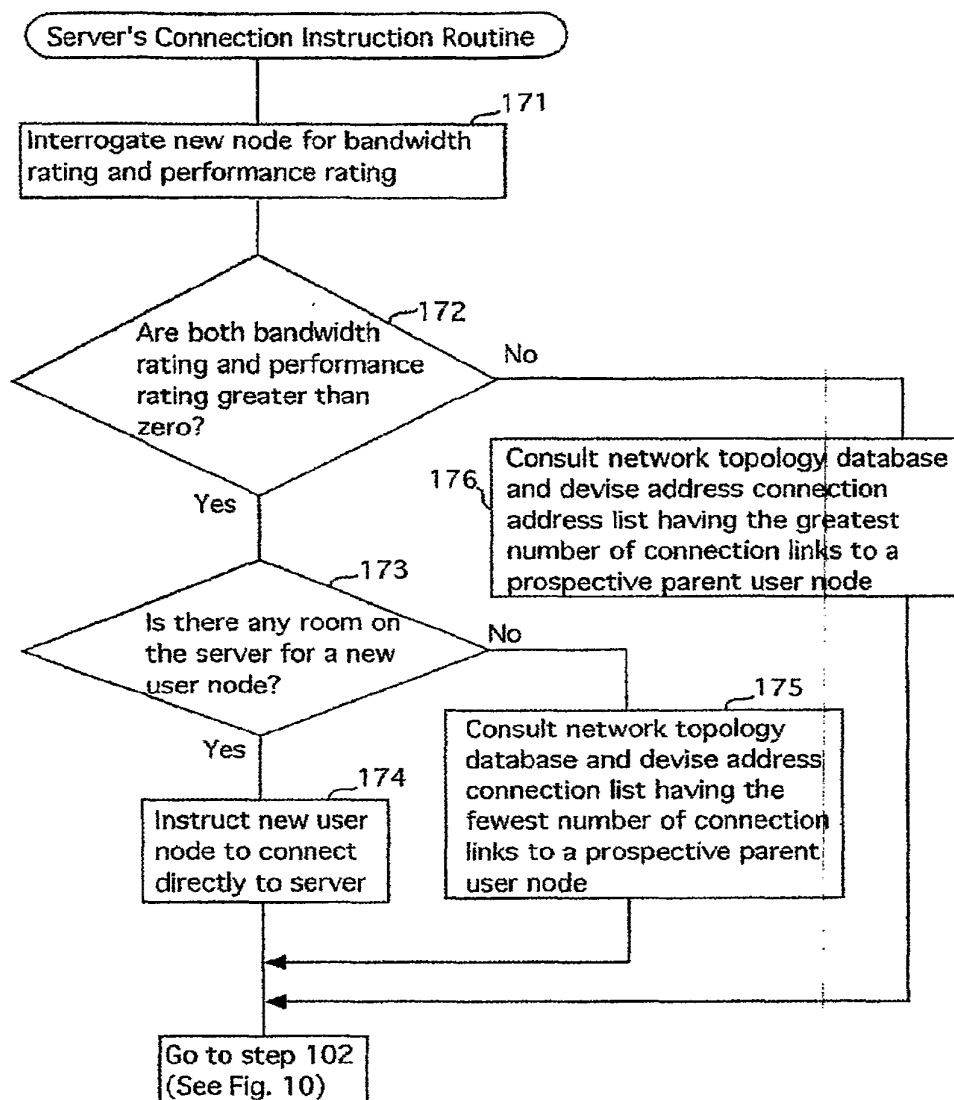


Fig. 17

12 of 31

13/31

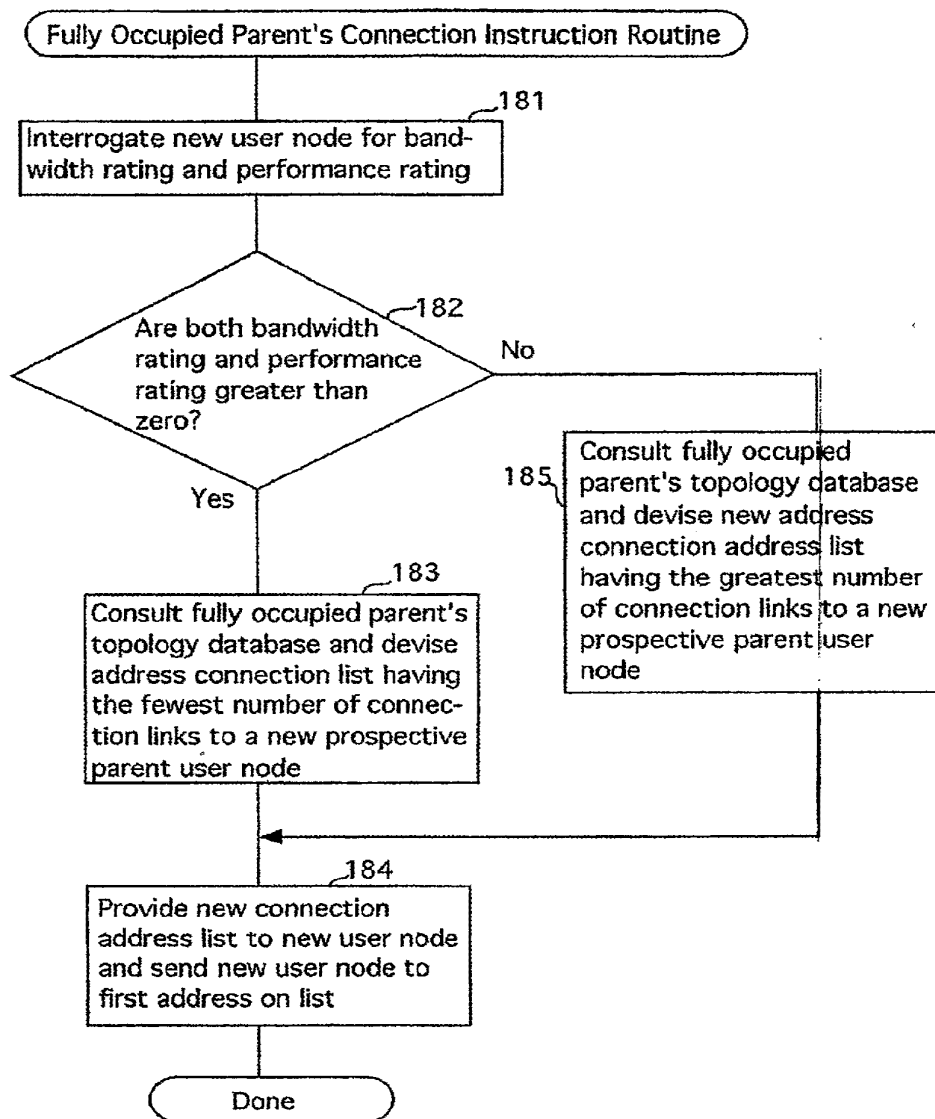
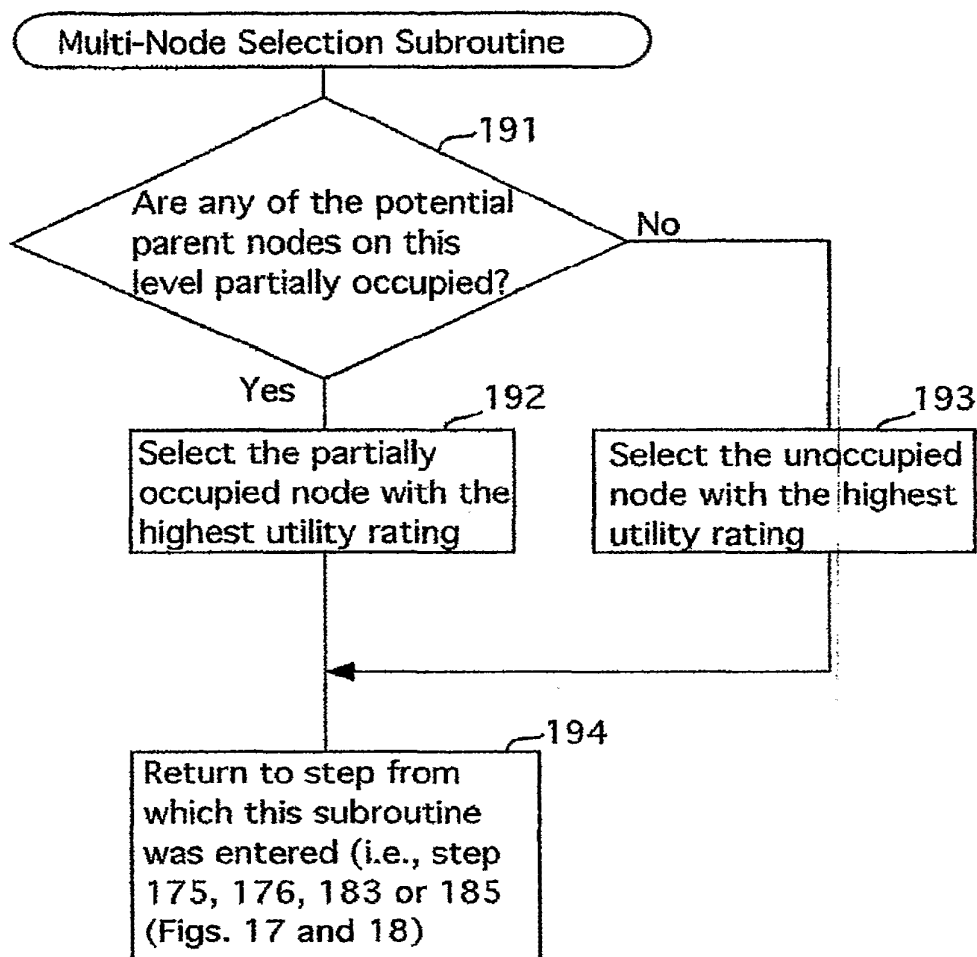


Fig. 18

13 of 31

14/31

**Fig. 19**

15/31

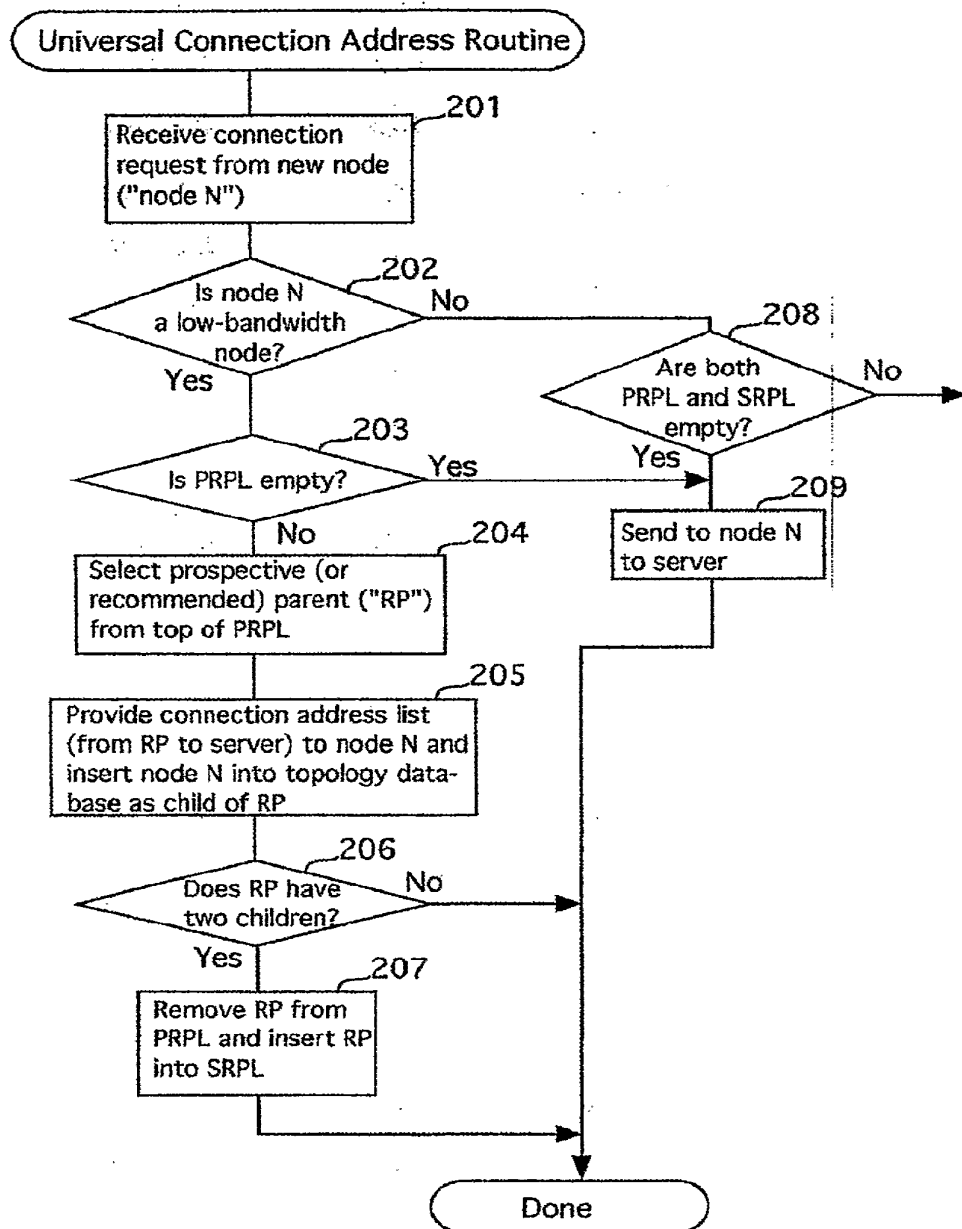


Fig. 20A

15 of 31

16/31

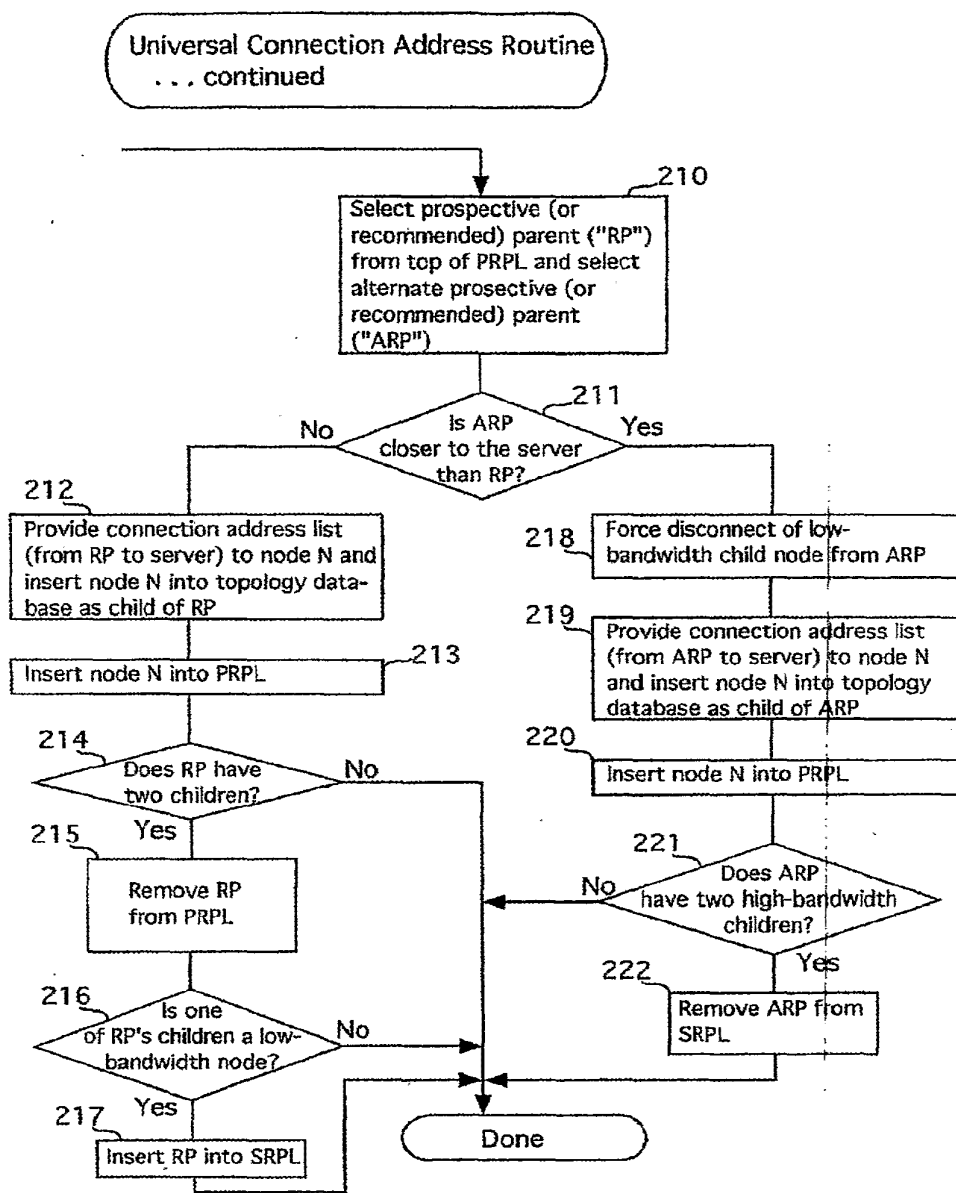


Fig. 20B

17/31

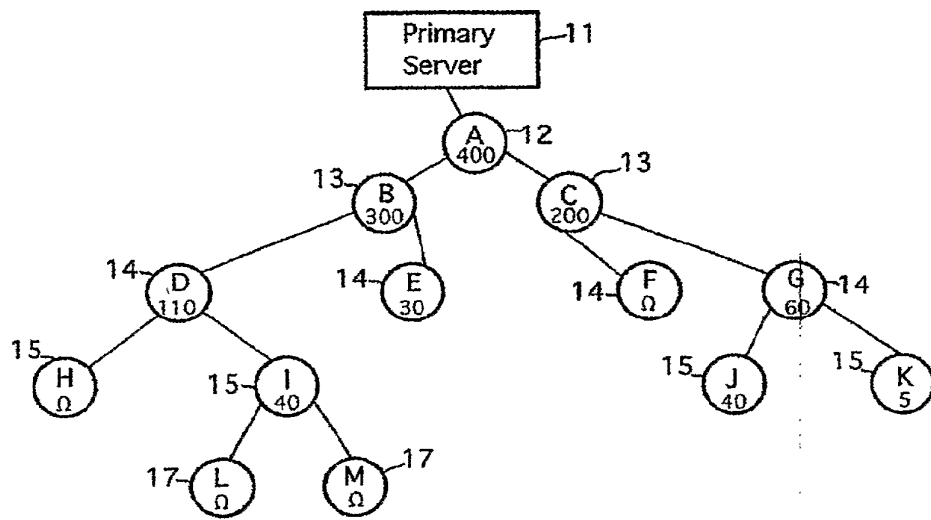


Fig. 21

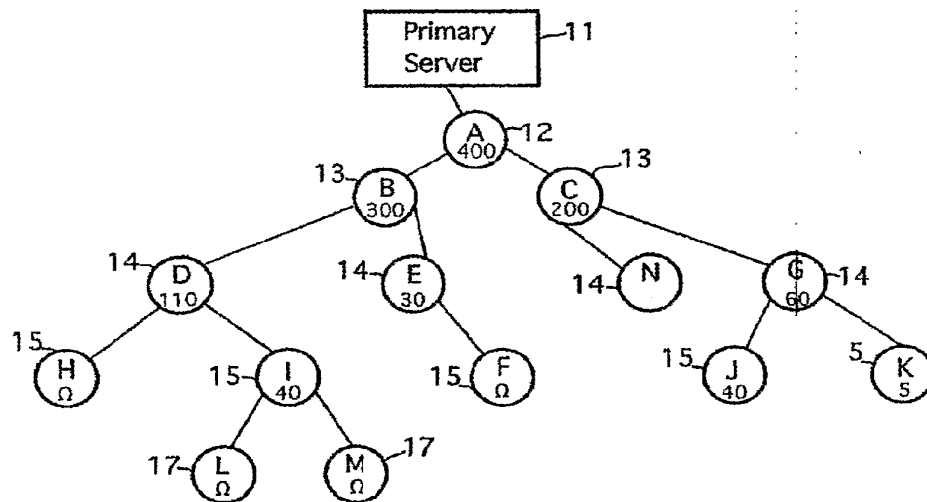


Fig. 22

17 of 31

18/31

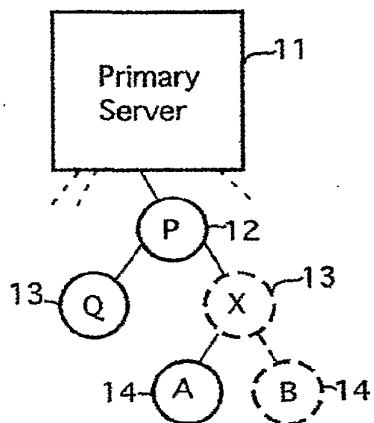


Fig. 23

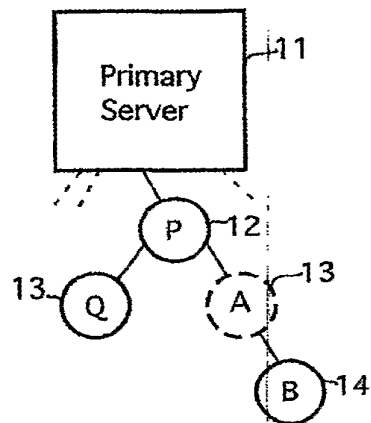


Fig. 24

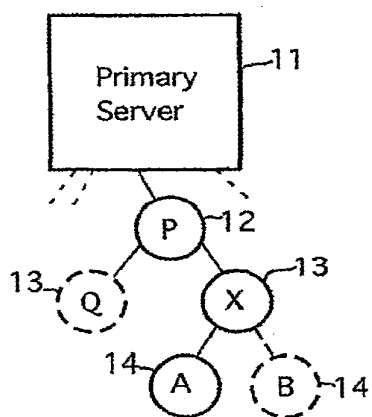


Fig. 25

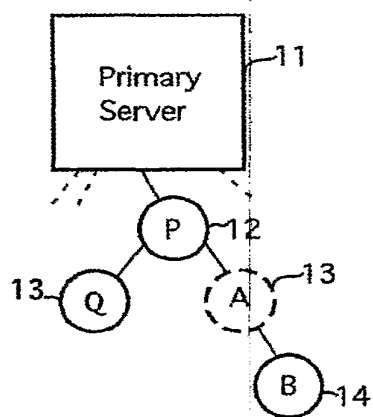


Fig. 26

18 of 31

19/31

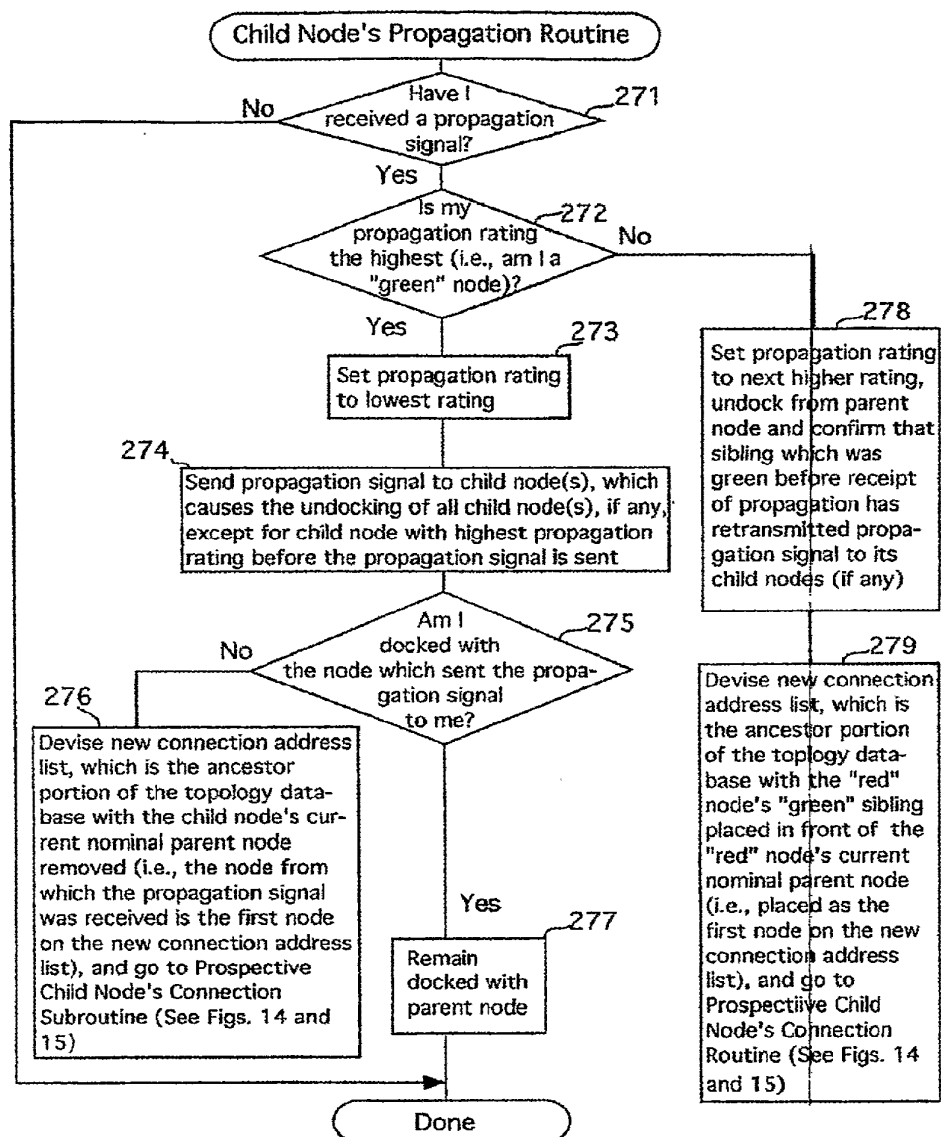


Fig. 27

20/31

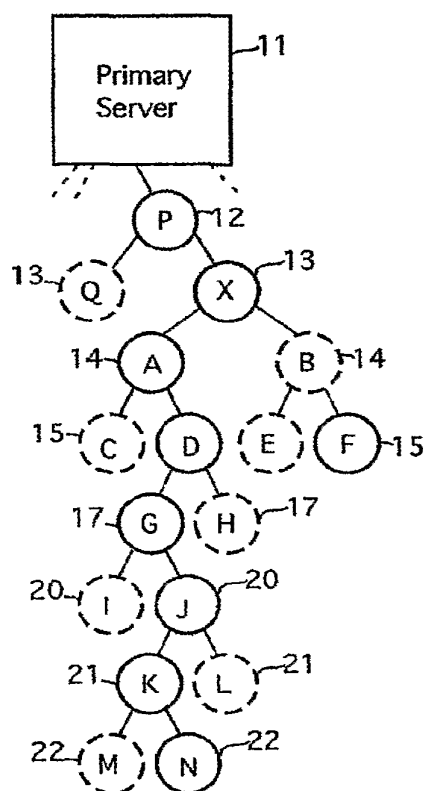


Fig. 28

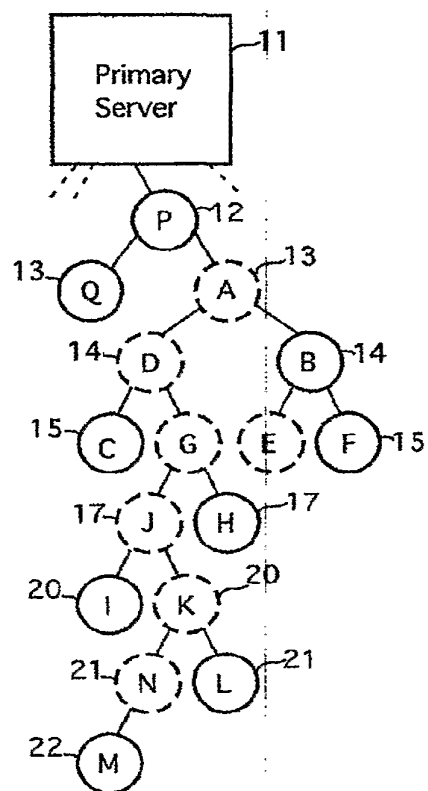
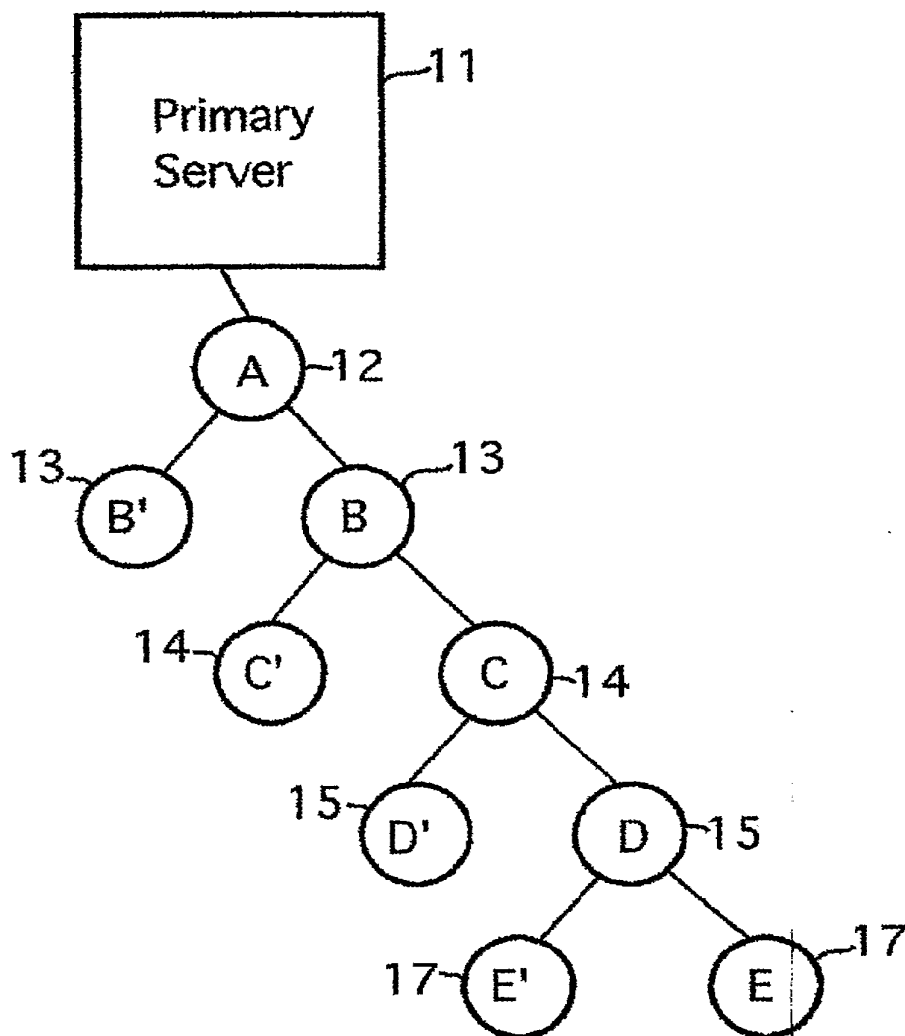


Fig. 29

21/31

**Fig. 30**

21 of 31

22/31

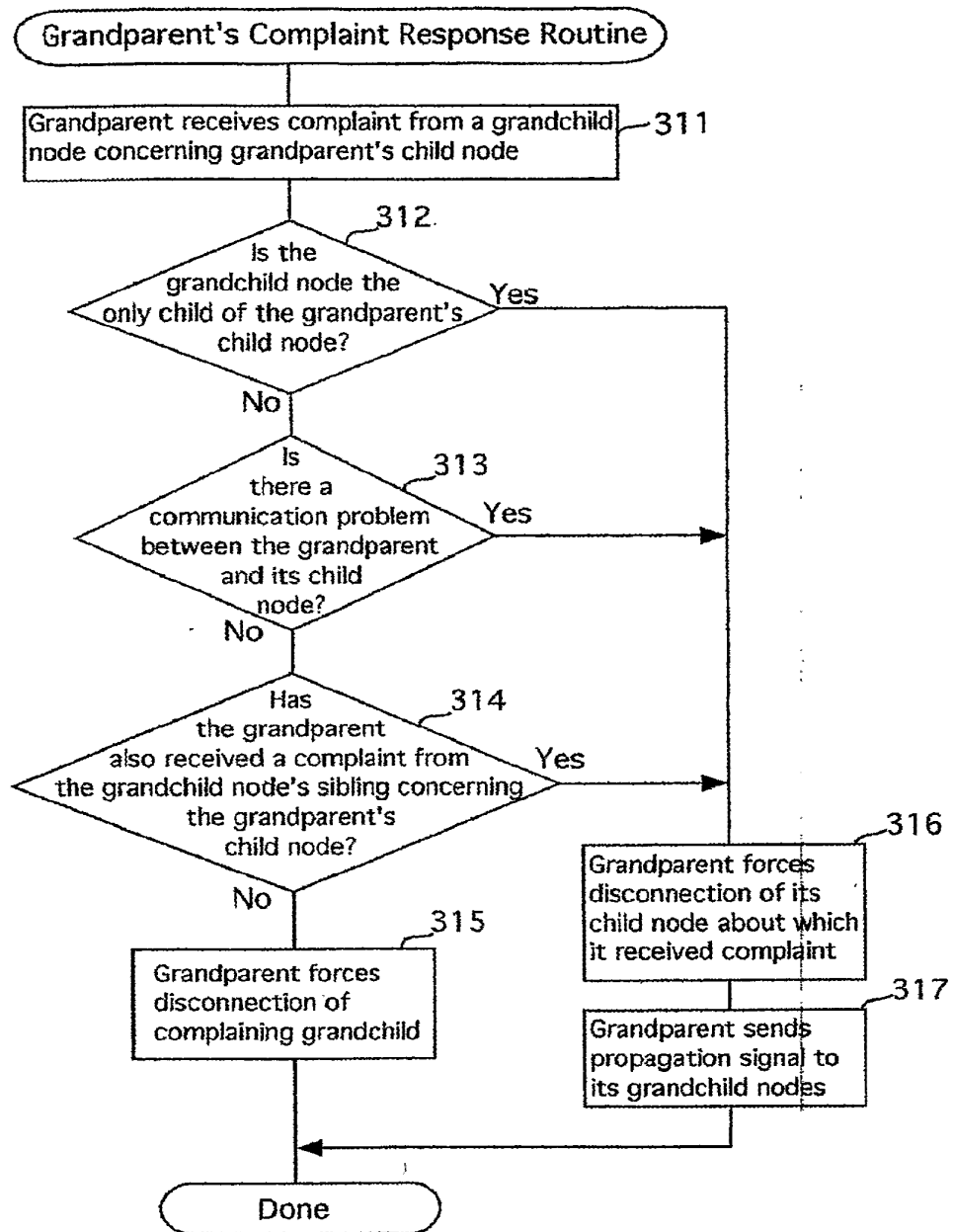


Fig. 31

22 of 31

23/31



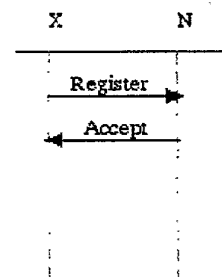
Fig. 32

Fig. 33A

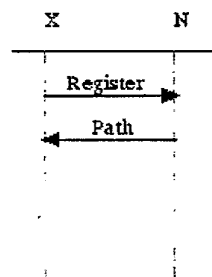
Fig. 33B

**Registration
(Accept / Path / Reject)**

Node N accepts node X



Node N sends path to node X



Node N rejects node X

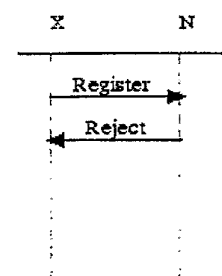
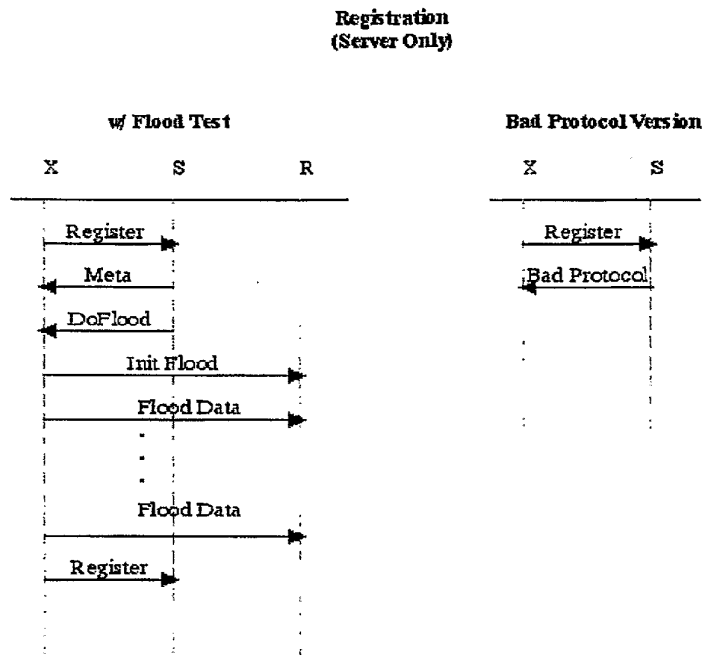


Fig. 33C

24/31

Fig. 34A

Fig. 34B



**Standard response for
Accept / Reject / Path
(see standard Register)**

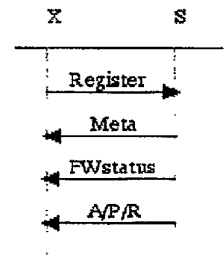


Fig. 34C

25/31

Fig. 35A

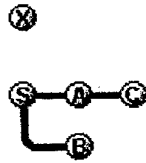
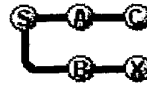


Fig. 35B



Example: Register to Server w/ Flood Test

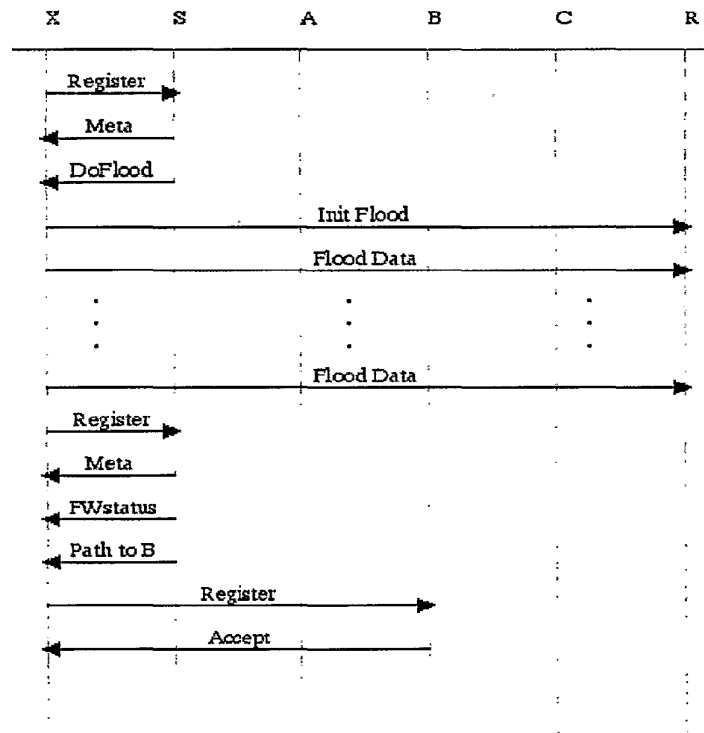


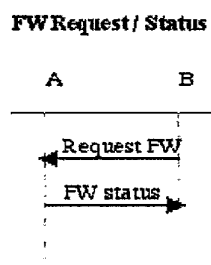
Fig. 36

26/31

Fig. 37



Fig. 38



27/31

Fig. 39A



Fig. 39B

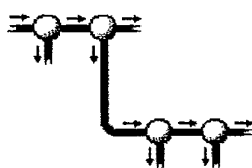
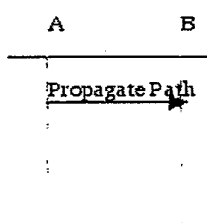


Fig. 40

Propagate Path/ Down



28/31

Fig. 41A



Fig. 41B

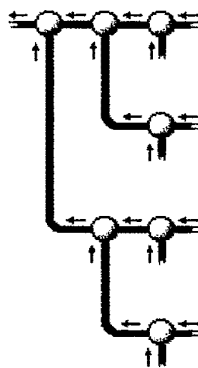


Fig. 42

Propagate NTM / Up

A	B
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

Propagate NTM

•
•
•
•
•
•
•

29/31

Fig. 43

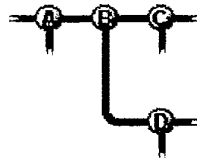
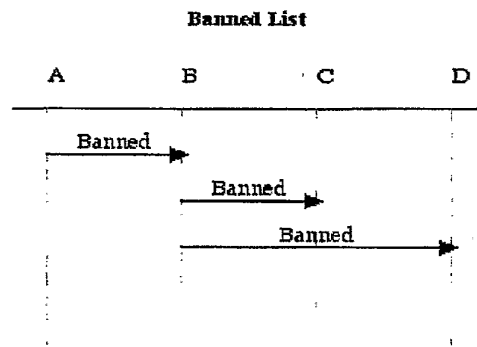
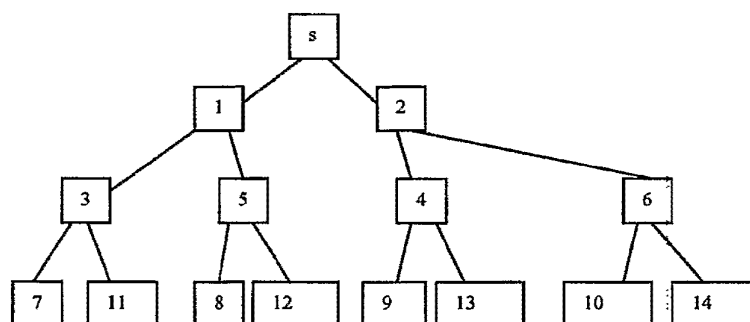


Fig. 44



30/31

FIG. 43



31/31

FIG. 40A



Fig. 46B

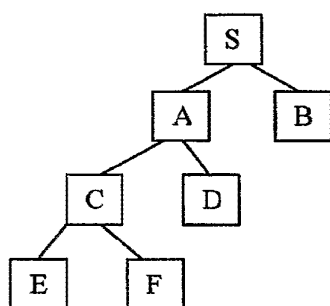
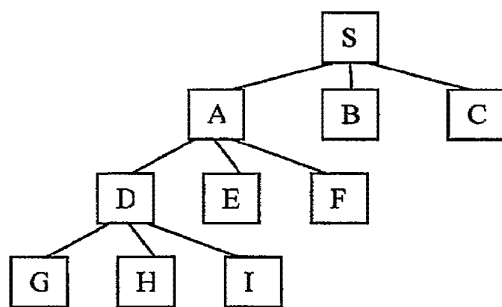


Fig. 46C



31 of 31

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. 70560. 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.

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-201343

(43)Date of publication of application : 18.07.2000

(51)Int.Cl.

H04N 7/173

(21)Application number : 11-000497

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 05.01.1999

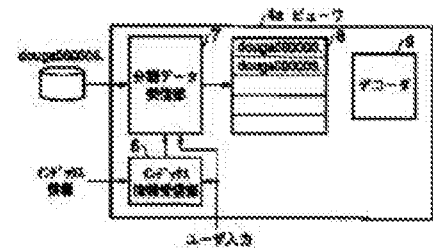
(72)Inventor : CHIJITANI MASAhide

(54) MOVING IMAGE DATA REPRODUCING DEVICE, MOVING IMAGE DATA DIVIDING DEVICE AND RECORDING MEDIUM READABLE BY COMPUTER IN WHICH PROGRAM IS RECORDED

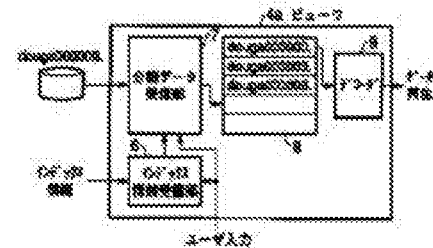
(57)Abstract:

PROBLEM TO BE SOLVED: To enable random access to moving image data without changing a server function.

SOLUTION: A moving image data reproducing device with an index information receiving means 6 to receive index information to be generated by relating plural pieces of divided data to be obtained by deviding the moving image data to its reproduction time information by requesting the information to a server, a divided data receiving means 7 to receive the divided data related to reproduction time information at the reproduction time information to which starting of reproducing is instructed and afterward by successively requesting the information to the server by referring to the index information received by the index information receiving means 6, a storage means 9 to store the divided data received by the divided data receiving means 7 and a reproducing means to successively reproduce the divided data stored in the storage means 9 according to the index information received by the index information receiving means 6 is provided.



(a)



(b)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-201343
(P2000-201343A)

(43) 公開日 平成12年7月18日 (2000.7.18)

(51) Int.Cl.⁷

H 0 4 N 7/173

識別記号

6 3 0

F I

H 0 4 N 7/173

テマコード* (参考)

6 3 0 5 C 0 6 4

審査請求 未請求 請求項の数13 ○L (全 13 頁)

(21) 出願番号 特願平11-497

(22) 出願日 平成11年1月5日 (1999.1.5)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 千々谷 眞英

東京都府中市東芝町1番地 株式会社東芝
府中工場内

(74) 代理人 100058479

弁理士 鈴江 武彦 (外6名)

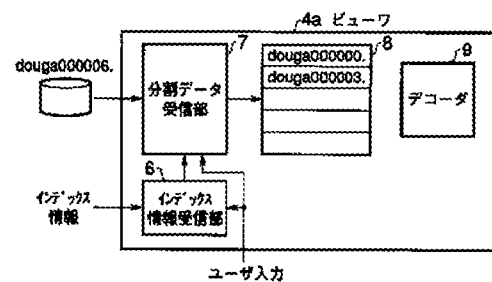
Fターム(参考) 5C064 BA07 BB10 BC16 BC20 BC27
BD16

(54) 【発明の名称】 動画像データ再生装置及び動画像データ分割装置並びにプログラムを記録したコンピュータ読み取り可能な記録媒体

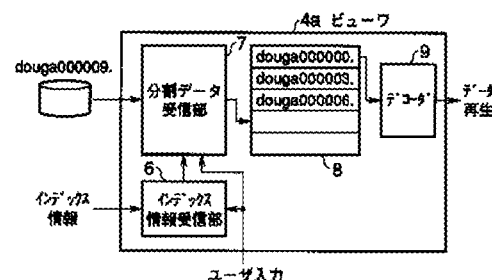
(57) 【要約】

【課題】 サーバの機能を変更することなく、動画像データのランダムアクセスを可能にする。

【解決手段】 動画像データが分割されて得られる複数の分割データとその再生時刻情報とを関連付けて生成されるインデックス情報を、サーバに要求して受信するインデックス情報受信手段と、当該インデックス情報受信手段に受信されたインデックス情報を参照し、再生開始を指示された再生時刻情報以降の再生時刻情報に関連付けられた分割データをサーバに順次要求して受信する分割データ受信手段と、当該分割データ受信手段に受信された分割データを記憶する記憶手段と、インデックス情報受信手段に受信されたインデックス情報にしたがって、記憶手段に記憶されている分割データを順次再生させる再生手段とを具備した動画像データ再生装置を提供する。



(a)



(b)

【特許請求の範囲】

【請求項1】 動画像データが分割されて得られる複数の分割データとそれぞれの再生時刻情報とを関連付けて生成されるインデックス情報を、サーバに要求して受信するインデックス情報受信手段と、

当該インデックス情報受信手段に受信されたインデックス情報にしたがって、再生開始を指示された再生時刻情報及びその後の再生時刻情報に関連付けられた分割データを、サーバに順次要求して受信する分割データ受信手段と、

当該分割データ受信手段によって受信された分割データを記憶する記憶手段と、

前記記憶手段に記憶されている分割データを順次再生させる再生手段とを具備したことを特徴とする動画像データ再生装置。

【請求項2】 請求項1記載の動画像データ再生装置において、

前記分割データの送信をサーバに要求してから、この分割データを受信するまでの時間を推定する時間推定手段を付加し、

前記分割データ受信手段は、当該時間推定手段によって推定された時間とほぼ同じ時間分の分割データが前記記憶手段に記憶されていない場合に、前記インデックス情報にしたがってサーバに分割データを要求し、

前記再生手段は、前記時間推定手段によって推定された時間とほぼ同じ時間分の分割データが前記記憶手段に記憶された場合に、この記憶手段に記憶されている分割データの再生を開始することを特徴とする動画像データ再生装置。

【請求項3】 動画像データが分割されて得られる複数の分割データについて伝送速度が異なる複数の種別が生成されている場合に、この複数種別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て生成されるインデックス情報を、サーバに要求して受信するインデックス情報受信手段と、

前記分割データの送信をサーバに要求してから、この分割データを受信するまでの時間を推定する時間推定手段と、

当該時間推定手段によって推定された時間に基づいて、要求する分割データの種別を選択する分割データ選択手段と、

前記インデックス情報受信手段に受信されたインデックス情報にしたがって、再生開始を指示された再生時刻情報及びその後の再生時刻情報に関連付けられ、かつ前記分割データ選択手段によって選択されている種別の分割データを、サーバに順次要求して受信する分割データ受信手段と、

当該分割データ受信手段によって受信された分割データを記憶する記憶手段と、

前記記憶手段に記憶されている分割データを順次再生さ

せる再生手段とを具備したことを特徴とする動画像データ再生装置。

【請求項4】 請求項3記載の動画像データ再生装置において、

前記分割データ受信手段は、前記時間推定手段に推定された時間とほぼ同じ時間分の分割データが前記記憶手段に記憶されていない場合に、前記データ選択手段によって選択されている種別の分割データを、前記インデックス情報にしたがって前記サーバに要求し、

10 前記再生手段は、前記時間推定手段によって推定された時間とほぼ同じ時間分の分割データが前記記憶手段に記憶された場合に、この記憶手段に記憶されている分割データの再生を開始することを特徴とする動画像データ再生装置。

【請求項5】 動画像データを複数に分割して複数の分割データを生成し、かつ前記複数の分割データとこの分割データの再生時刻情報とを関連付けたインデックス情報を生成する動画像データ分割装置。

20 【請求項6】 動画像データが分割されて得られる複数の分割データについて伝送速度が異なる複数の種別を生成し、この複数の種別の各分割データをそれぞれの再生時刻情報に関連付けたインデックス情報を生成する動画像データ分割装置。

【請求項7】 コンピュータに、動画像データが分割されて得られる複数の分割データとそれぞれの再生時刻情報とを関連付けて生成されるインデックス情報を、サーバに要求して受信するインデックス情報受信機能と、

当該インデックス情報受信機能によって受信されたインデックス情報にしたがって、再生開始を指示された再生時刻情報及びその後の再生時刻情報に関連付けられた分割データを、サーバに順次要求して受信する分割データ受信機能と、

当該分割データ受信機能によって受信された分割データを記憶する記憶機能と、

前記記憶機能によって記憶されている分割データを順次再生させる再生機能とを実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項8】 請求項7記載のコンピュータ読み取り可能な記録媒体において、

前記分割データの送信をサーバに要求してから、この分割データを受信するまでの時間を推定する時間推定機能を付加し、

前記分割データ受信機能は、当該時間推定機能によって推定された時間とほぼ同じ時間分の分割データが前記記憶機能によって記憶されていない場合に、前記インデックス情報にしたがってサーバに分割データを要求し、

前記再生機能は、前記時間推定機能によって推定された時間とほぼ同じ時間分の分割データが前記記憶機能によって記憶された場合に、この記憶機能によって記憶され

ている分割データの再生を開始することを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項9】 コンピュータに、
動画データが分割されて得られる複数の分割データについて伝送速度が異なる複数の種別が生成されている場合に、この複数種別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て生成されるインデックス情報をサーバに要求して受信するインデックス情報受信機能と、
前記分割データの送信をサーバに要求してから、この分割データを受信するまでの時間を推定する時間推定機能と、

当該時間推定機能によって推定された時間に基づいて、要求する分割データの種別を選択する分割データ選択機能と、

前記インデックス情報受信機能によって受信されたインデックス情報にしたがって、再生開始を指示された再生時刻情報及びその後の再生時刻情報に関連付けられ、かつ前記分割データ選択機能によって選択されている種別の分割データを、サーバに順次要求して受信する分割データ受信機能と、

当該分割データ受信機能によって受信された分割データを記憶する記憶機能と、

前記記憶機能によって記憶されている分割データを順次再生させる再生機能とを実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項10】 請求項9記載のコンピュータ読み取り可能な記録媒体において、

前記分割データ受信機能は、前記時間推定機能によって推定された時間とほぼ同じ時間分の分割データが前記記憶機能に記憶されていない場合に、前記データ選択機能によって選択されている種別の分割データを、前記インデックス情報にしたがって前記サーバに要求し、
前記再生機能は、前記時間推定機能によって推定された時間とほぼ同じ時間分の分割データが前記記憶機能によって記憶された場合に、この記憶機能によって記憶されている分割データの再生を開始することを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項11】 コンピュータに、
動画データを複数の分割して複数の分割データを生成し、かつ前記複数の分割データとこの分割データの再生時刻情報とを関連付けたインデックス情報を生成する機能を実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項12】 コンピュータに、
動画データが分割されて得られる複数の分割データについて伝送速度が異なる複数の種別を生成し、この複数の種別の各分割データをそれぞれの再生時刻情報に関連付けたインデックス情報を生成する機能を実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項13】 サーバに記憶されている動画データをクライアントに配信する動画データ配信システムにおいて、

前記サーバは、

前記動画データが分割されて得られる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とを保持し、

前記クライアントは、

前記インデックス情報を前記サーバに要求して受信するインデックス情報受信手段と、

当該インデックス情報受信手段に受信されたインデックス情報にしたがって、再生開始を指示された再生時刻情報及びその後の再生時刻情報に関連付けられた分割データを前記サーバに順次要求して受信する分割データ受信手段と、

当該分割データ受信手段によって受信された分割データを記憶する記憶手段と、

前記記憶手段に記憶されている分割データを順次再生させる再生手段とを備えることを特徴とする動画データ配信システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、サーバの構成及び機能を変更することなく、動画データのランダムアクセスを可能にするための動画データ再生装置及び動画データ分割装置並びにプログラムを記録したコンピュータ読み取り可能な記録媒体に関する。

【0002】

【従来の技術】従来の動画データ配信システムは、主にクライアントとWWWサーバとから構成される。この従来の動画データ配信システムでは、まずクライアントからWWWサーバに再生を求める動画データ（動画コンテンツ）の伝送要求が送信される。

【0003】WWWサーバは各種データを保持しており、このデータの中から要求された動画データを選択し送信する。

【0004】送信された動画データは、伝送要求を送信したクライアントに受信され、ストリーム再生がなされる。

【0005】このような従来の動画データ配信システムにおいて、クライアント及びWWWサーバがインターネットを介して接続されている場合には、一般的にHTTPプロトコルにより動画データの伝送がなされる。

【0006】ここで、例えば動画データの再生を求めるユーザの中には、動画データの途中位置からの再生（以下、「ランダムアクセス」という）を求める者もある。

【0007】従来の動画データ配信システムによってランダムアクセスを実現するためには、ランダムアクセスを可能とする専用のHTTPサーバをこの動画デー

タ配信システムに適用する必要がある。

【0008】また、別の手法として、ランダムアクセスを可能とする専用のプログラムであるCGI(Common Gateway Interface)をWWWサーバに実装させる必要がある。

【0009】この専用のHTTPサーバや、CGIを実装したWWWサーバは、動画データを再生するクライアントから独自の要求を受信し、動画データを指定された位置からHTTPプロトコルによって送信する機能を持つ。

【0010】

【発明が解決しようとする課題】しかしながら、上記のように、ランダムアクセスを可能とするために専用のHTTPサーバを用いたり、サーバにCGIを実装する方法においては、従来の動画データ配信システムのWWWサーバに手を加える必要があり、現状のインフラのままでは実現することができない。

【0011】したがって、ランダムアクセスの実現、導入が困難であり、作業が複雑である。

【0012】また、この従来の方法では、ランダムアクセスのための処理が主にサーバでなされるため、サーバの負荷が増加するという問題が生じる。特に、動画データ配信システムが複数のクライアントを有する場合には、その傾向が強くなる。

【0013】本発明は、以上のような実状に鑑みてなされたもので、サーバの構成や機能に手を加えることなく動画データのランダムアクセスを可能とするための動画データ再生装置及び動画データ分割装置並びにプログラムを記録したコンピュータ読み取り可能な記録媒体を提供することを目的とする。

【0014】

【課題を解決するための手段】本発明の骨子は、動画データを複数の分割して分割データとし、この分割データにそれぞれの再生時間を示す再生時刻情報を関連付けてインデックス情報とし、このインデックス情報にしたがって分割データを順次再生する点にある。

【0015】以下、本発明を実現するにあたって講じた具体的手段について説明する。

【0016】第1の発明は、動画データが分割されて得られる複数の分割データとそれぞれの再生時刻情報とを関連付けて生成されるインデックス情報をサーバに要求して受信するインデックス情報受信手段と、当該インデックス情報受信手段に受信されたインデックス情報にしたがって、再生開始を指示された再生時刻情報及びその後の再生時刻情報に関連付けられた分割データをサーバに順次要求して受信する分割データ受信手段と、当該分割データ受信手段によって受信された分割データを記憶する記憶手段と、インデックス情報受信手段に受信されたインデックス情報の再生時刻情報にしたがって、記憶手段に記憶されている分割データを順次再生させる再

生手段とを具備した動画データ再生装置である。

【0017】この第1の発明の動画データ再生装置においては、再生開始を指示された再生時刻情報以降の分割データが順次受信されストリーム再生される。

【0018】また、この動画データ再生装置を適用する場合には、サーバは自己が保持するデータのうち要求されたデータを送信する機能を有していればよい。

【0019】したがって、第1の発明においては、サーバに新しい構成や機能を付加することなく容易にランダムアクセスが実現される。

【0020】また、ランダムアクセスを実現させるための処理が、サーバではなく動画データ再生装置(クライアント)側によって実行されるため、サーバの負荷の増加を防止することができる。

【0021】第2の発明は、第1の発明の動画データ再生装置において、分割データの送信をサーバに要求してから、この分割データを受信するまでの時間を推定する時間推定手段を付加し、分割データ受信手段は、当該時間推定手段によって推定された時間とほぼ同じ時間分の分割データが記憶手段に記憶されていない場合に、前記インデックス情報にしたがってサーバに分割データを要求し、再生手段は、時間推定手段によって推定された時間とほぼ同じ時間分の分割データが記憶手段に記憶された場合に、この記憶手段に記憶されている分割データの順次再生を開始する動画データ再生装置である。

【0022】この第2の発明の動画データ再生装置においては、分割データを要求してから受信するまでの間に再生されると推測される分割データが、記憶手段に保持される。

【0023】したがって、この第2の発明においては、第1の発明と同様の作用効果に加えて、必要最低限の量の分割データが格納されしだい、動画データの再生がなされるため、再生が開始されるまでの時間を最適化することができる。

【0024】これにより、再生開始までの時間を短縮させることができる。

【0025】また、次の分割データが要求されてから受信されるまでの間に必要な分割データは確保済みであるため、動画データの再生が途切れることを防止することができる。

【0026】さらに、分割データをサーバに要求してから受信するまでの時間が逐次測定され、この時間にしたがって再生可能な状態で記憶しておく分割データの量が決定される。

【0027】ゆえに、回線の動的な負荷変動にも対応することができ、適切な量の分割データが確保される。

【0028】したがって、スムーズなストリーム再生が可能となる。

【0029】第3の発明は、動画データが分割されて得られる複数の分割データについて伝送速度が異なる複

数の種別が生成されている場合に、この複数種別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て生成されるインデックス情報をサーバに要求して受信するインデックス情報受信手段と、分割データの送信をサーバに要求してから、この分割データを受信するまでの時間を推定する時間推定手段と、当該時間推定手段によって推定された時間に基づいて、要求する分割データの種別を選択する分割データ選択手段と、インデックス情報受信手段に受信されたインデックス情報にしたがって、再生開始を指示された再生時刻情報及びその後の再生時刻情報に関連付けられ、かつ分割データ選択手段によって選択されている種別の分割データをサーバに順次要求して受信する分割データ受信手段と、当該分割データ受信手段によって受信された分割データを記憶する記憶手段と、インデックス情報受信手段に受信されたインデックス情報の再生時刻情報にしたがって、記憶手段に記憶されている分割データを順次再生させる再生手段とを具備した動画データ再生装置である。

【0030】この第3の発明の動画データ再生装置においては、伝送速度の異なる複数種別の分割データのうち、回線の負荷変動を考慮して適した伝送速度の分割データが選択され受信される。

【0031】したがって、回線の伝送速度が遅い場合には、高速伝送可能な分割データにより伝送がなされるため、再生すべきデータが未伝送となることを防止することができる。

【0032】また、第1の発明と同様の作用により、サーバを変更することなく、ランダムアクセスが可能である。

【0033】第4の発明は、第3の発明の動画データ再生装置において、分割データ受信手段は、時間推定手段に推定された時間とほぼ同じ時間分の分割データが記憶手段に記憶されていない場合に、データ選択手段によって選択されている種別の分割データを、インデックス情報にしたがってサーバに要求し、再生手段は、時間推定手段によって推定された時間とほぼ同じ時間分の分割データが記憶手段に記憶された場合に、この記憶手段に記憶されている分割データの順次再生を開始する動画データ再生装置である。

【0034】この第4の発明の動画データ再生装置においては、第3の発明と同様の作用効果に加えて、分割データをサーバに要求してから受信するまでの間の再生に必要と推測される分の分割データが記憶手段に保持される。

【0035】したがって、先の第2の発明と同様の効果を得ることができる。

【0036】第5の発明は、動画データを複数に分割して複数の分割データを生成し、かつ前記複数の分割データとこの分割データの再生時刻情報とを関連付けたインデックス情報を生成する動画データ分割装置であ

る。

【0037】この第5の発明の動画データ分割装置を用いて複数の分割データ及びインデックス情報を作成してサーバに備え、第1又は第2の発明の動画データ再生装置を用いることで、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となる。

【0038】また、この動画データ分割装置を用いることで、複数の分割データ及びインデックス情報を容易に作成できる。

【0039】第6の発明は、動画データが分割されて得られる複数の分割データについて伝送速度が異なる複数の種別を生成し、この複数の種別の各分割データをそれぞれの再生時刻情報に関連付けたインデックス情報を生成する動画データ分割装置である。

【0040】この第6の発明の動画データ分割装置を用いて複数種別の各分割データ及びインデックス情報を作成してサーバに備え、第3又は第4の発明の動画データ再生装置を用いることで、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となり、伝送にかかる時間を調節することができる。

【0041】また、この動画データ分割装置を用いることで、複数種別の分割データ及びインデックス情報を容易に作成できる。

【0042】第7乃至第12の発明は、それぞれ上記第1乃至第6の発明と同様の機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0043】この第7乃至第12の発明のようなプログラムを記録した記録媒体を用いることによって、上述した機能を有していない装置や、計算機においても、簡単に上述した機能を付加することができる。

【0044】第13の発明は、サーバに記憶されている動画データをクライアントに配信する動画データ配信システムにおいて、サーバは、動画データが分割されて得られる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とを保持し、クライアントは、インデックス情報をサーバに要求して受信するインデックス情報受信手段と、当該インデックス情報受信手段に受信されたインデックス情報にしたがって、再生開始を指示された再生時刻情報及びその後の再生時刻情報に関連付けられた分割データをサーバに順次要求して受信する分割データ受信手段と、当該分割データ受信手段によって受信された分割データを記憶する記憶手段と、インデックス情報受信手段に受信されたインデックス情報の再生時刻情報にしたがって、記憶手段に記憶されている分割データを順次再生させる再生手段とを備える動画データ配信システムである。

【0045】この第13の発明の動画データ配信システムにおいては、クライアントが先の第1の発明の動画

像データ再生装置と同様に機能する。

【0046】これにより、第1の発明と同様の作用効果を得ることができる。

【0047】なお、第2乃至第12の発明も、この第13の発明と同様に、動画像データ配信システムに適用可能である。

【0048】

【発明の実施の形態】以下、図面を参照しながら本発明の実施の形態について説明する。

【0049】(第1の実施の形態)本実施の形態においては、動画像データ(動画コンテンツ)を複数に分割した分割データと、この複数の分割データとそれぞれの再生時間情報を関連付けたインデックス情報とを用いることで、サーバを変更することなく動画像データのランダムアクセスを可能とする動画像データ配信システムについて説明する。

【0050】図1は、本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である。

【0051】この動画像データ配信システム1は、動画像データから分割データとインデックス情報とを生成するエンコーダ2と、内部の記憶装置3aに各種データをファイル形式で記憶するサーバ3と、動画像データ再生用のビューワ4aを有するクライアント4とから構成されている。

【0052】この動画像データ配信システム1では、動画像データがスプライスポイントでファイル分割されており、ビューワ4aがインデックス情報に基づいて好きな位置の分割データをサーバ3に要求し、受信する。

【0053】以下、この動画像データ配信システム1の各構成要素について詳しく説明する。

【0054】図2は、本実施の形態に係る動画像データ配信システム1に備えられるエンコーダ2の機能を示すブロック図である。

【0055】このエンコーダ2は、動画エンコード時に、ビデオ等の動画素材からなる動画像データ(動画コンテンツ)を所定のスプライスポイント毎に分割して複数の分割データを生成する。また生成した各分割データには異なるファイル名を付けて出力する。この図2では、分割データのファイル名の例として「doga000000」、「doga000003」、「doga000006」等が示されている。

【0056】なお、この各分割データは、ビューワ4a内の受信ページバッファの1ページ内に収まるサイズで生成される。

【0057】さらに、このエンコーダ2は、各分割データのファイル名と、それぞれの再生時刻の情報とを関係付けてインデックス情報5を生成し、ファイル形式で出力する。

【0058】図3は、本実施の形態に係る動画像データ配信システム1に備えられるサーバ3の構成を示すブ

ック図である。

【0059】サーバ3は、主に記憶装置3aと、サーバ本体3bとから構成される。

【0060】サーバ本体3bは、記憶装置3aに様々なデータを蓄積する機能と、クライアント4からの要求にしたがって記憶装置3aの内容を読み出して送信する機能を有する。

【0061】記憶装置3aは、各種のデータを記憶し、ここではエンコーダ2によって生成された複数の分割データ及びインデックス情報5も記憶している。

【0062】クライアント4は、図1で示すように、主にクライアント本体4bと、動画像データ再生装置として機能するビューワ4aとから構成されており、クライアント本体4b上でビューワ4aが稼働する。

【0063】図4は、本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の機能を示すブロック図である。

【0064】なお、図4(a)は再生開始前のバッファリング中の状態を示し、3ページ分の分割データがページバッファに格納されるまでの状態を示している。

【0065】一方、図4(b)は再生中の状態を示しており、1ページ分の分割データがデコーダに読み出されると、再びサーバ3から分割データが読み出される状態を示している。

【0066】このビューワ4aは、動画像データを再生する装置であり、インデックス情報受信部6、分割データ受信部7、ページバッファ8、デコーダ9から構成される。

【0067】インデックス情報受信部6は、再生を求める動画像データの指定をユーザから入力する。また、指定された動画像データに関するインデックス情報5をサーバ3に要求し、この要求の応答としてサーバ3からインデックス情報5を受信して保持する。

【0068】分割データ受信部7は、動画像データの再生開始位置となる再生時刻情報をユーザから入力する。また、インデックス情報受信部6のインデックス情報5を参照し、ユーザから入力された再生時刻情報以降の再生時刻情報に関連付けられているファイルの送信をサーバ3に要求する。この要求は、ページバッファ8に格納された分割データの量が一定量(ここでは3つとする)になるまで順次出力する。

【0069】そして、この分割データ受信部7は、この要求の応答として分割データを順次受信し、ページバッファ8に出力する。

【0070】ページバッファ8は、複数のページからなり、各ページ毎に一つの分割データをバッファリングする。また、各ページの内容を再生時刻順に並び替えて記憶する。さらに、デコーダ9によって読み出されたページ内の分割データを破棄し、次に入力される分割データを新規に確保する。

【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フ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、ビューフ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リング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、ビューフ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側のビューフ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ではなくビューフ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、ビューフ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フ4aを変形したビューフについて説明する。

【0092】先の第1の実施の形態においては、ビューフ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】先のビューフ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、ビューフ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

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な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類の分割データ及びインデックス情報を生成する。

【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、要求伝送時間が小さすぎない場合には、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

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すブロック図。

【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。

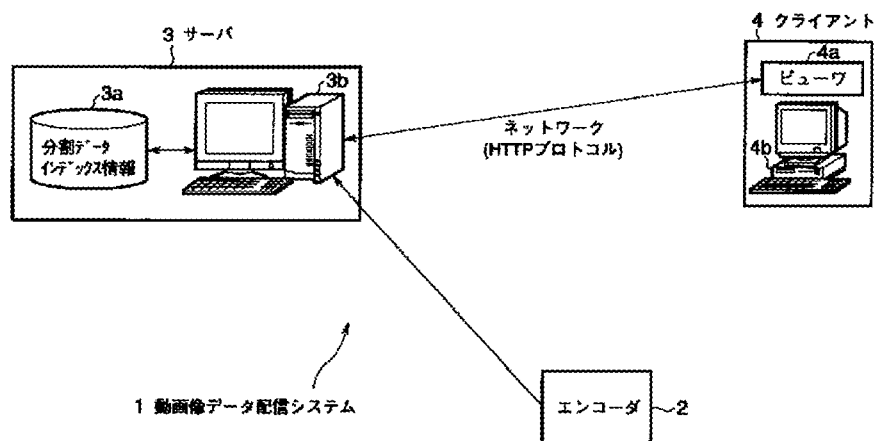
【図8】本発明の第3の実施の形態に係る動画像データ配信システムに備えられるエンコーダの機能を示すブロック図。

【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。

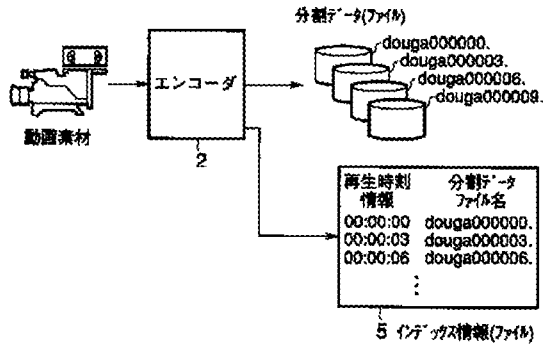
【符号の説明】

- 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

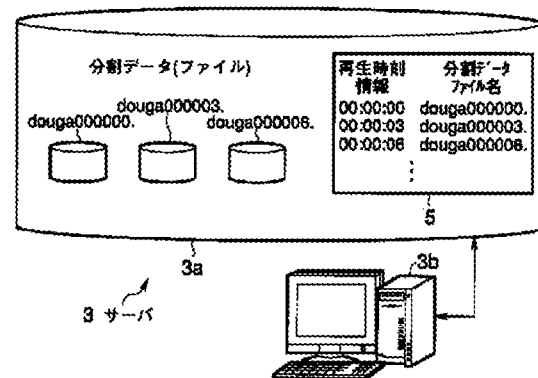
【図1】



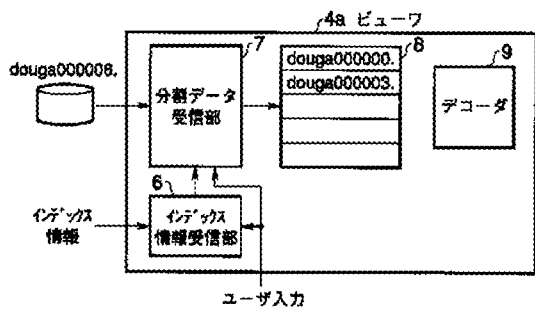
【図2】



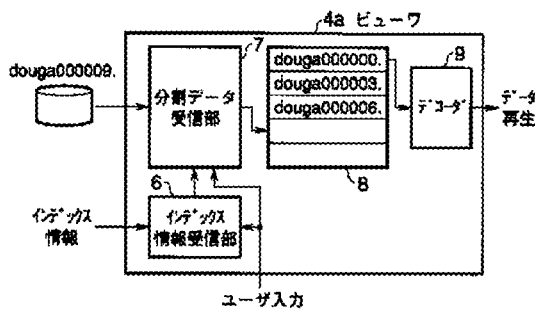
【図3】



【図4】

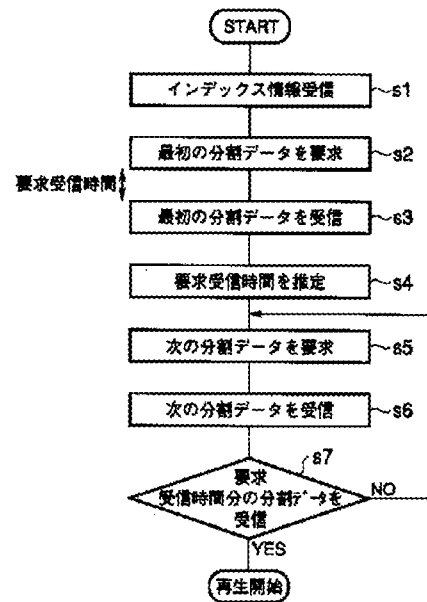


(a)

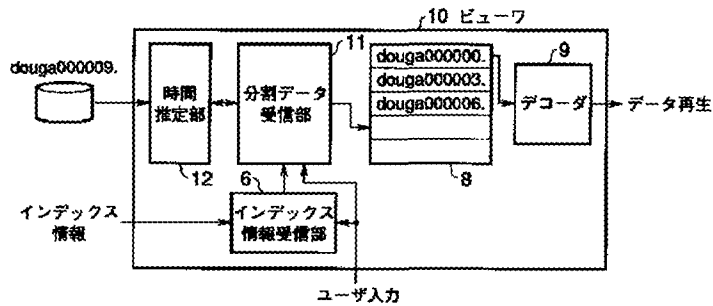


(b)

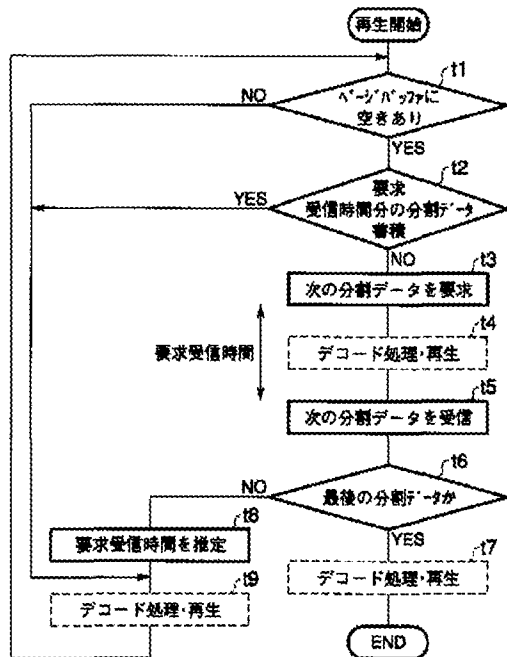
【図6】



【図5】



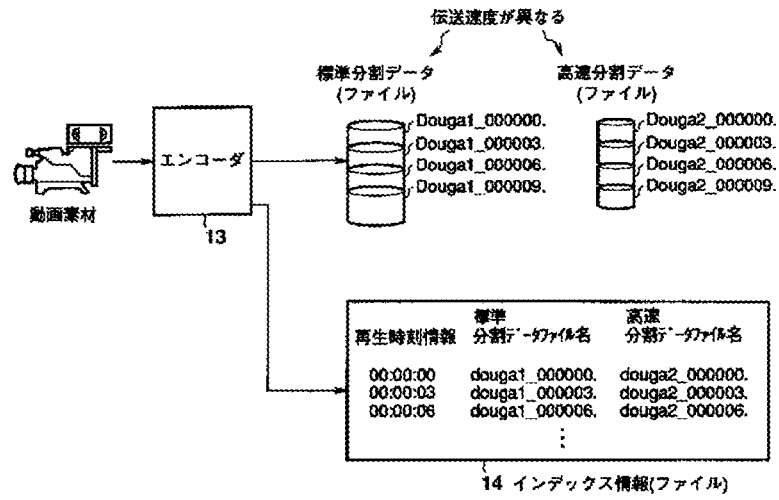
【図7】



【図9】



【図8】



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2011-004225

(43)Date of publication of application : 06.01.2011

(51)Int.Cl.

H04W 48/20 (2009.01)

H04W 84/12 (2009.01)

H04W 48/16 (2009.01)

(21)Application number : 2009-146288

(71)Applicant : NEC INFRONTIA CORP

(22)Date of filing : 19.06.2009

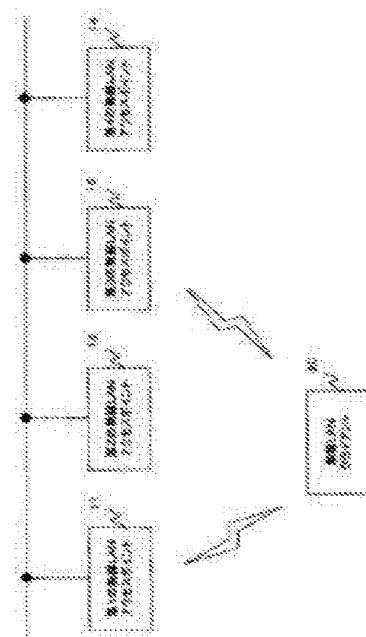
(72)Inventor : HODOSHIMA TAKESHI

(54) **SYSTEM, METHOD AND PROGRAM OF HIGH-SPEED ROAMING OF WIRELESS LAN**

(57)Abstract:

PROBLEM TO BE SOLVED: To attain high-speed roaming in a wireless LAN network.

SOLUTION: In a LAN system including: a plurality of wireless LAN access points; and a wireless LAN client to be connected to the wireless LAN access points, the wireless LAN access points are the ones existing on the same local area network, used channels and BSSIDS of the wireless LAN access points set to the same ESSID are recognized as wireless LAN access point information, held as data, and the wireless LAN client utilizes the wireless LAN access point information when the above connection is performed.



(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2011-4225

(P2011-4225A)

(43) 公開日 平成23年1月6日(2011.1.6)

(51) Int. Cl.		F I		テーマコード (参考)
HO4W 48/20	(2009.01)	HO4Q	7/00 416	5K067
HO4W 84/12	(2009.01)	HO4Q	7/00 630	
HO4W 48/16	(2009.01)	HO4Q	7/00 404	

審査請求 未請求 請求項の数 15 O L (全 11 頁)

(21) 出願番号	特願2009-146288 (P2009-146288)	(71) 出願人	000227205 NECインフロンティア株式会社 神奈川県川崎市高津区北見方2丁目6番1号
(22) 出願日	平成21年6月19日(2009.6.19)	(74) 代理人	100130029 弁理士 永井 道雄
		(74) 代理人	100166338 弁理士 関口 正夫
		(74) 代理人	100152054 弁理士 仲野 孝雅
		(72) 発明者	程島 健 神奈川県川崎市高津区北見方2丁目6番1号 NECインフロンティア株式会社内
		Fターム(参考)	5K067 AA15 EE02 EE10 HH22 HH23 JJ72

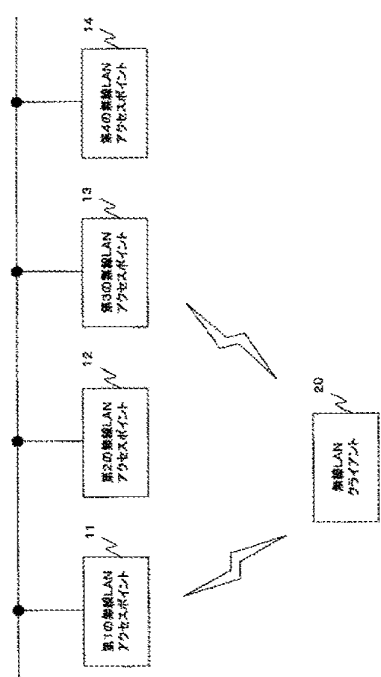
(54) 【発明の名称】 無線LANの高速ローミングシステム、高速ローミング方法及び高速ローミングプログラム

(57) 【要約】

【課題】無線LANネットワークにおいて高速ローミングを実現する。

【解決手段】複数の無線LANアクセスポイントと、当該無線LANアクセスポイントと接続される無線LANクライアントと、を有する無線LANシステムにおいて、前記無線LANアクセスポイントは同一のローカルエリアネットワーク上に存在する無線LANアクセスポイントであって同一のESSIDに設定されている無線LANアクセスポイントの使用チャンネル及びBSSIDを同一ネットワークの無線LANアクセスポイント情報として認識しデータとして保持し、前記無線LANクライアントは前記接続を行う際に前記無線LANアクセスポイント情報を利用する。

【選択図】図1



【特許請求の範囲】**【請求項1】**

複数の無線LANアクセスポイントと、当該無線LANアクセスポイントと接続される無線LANクライアントと、を有する無線LANシステムにおける接続方法において、

前記無線LANアクセスポイントは同一のローカルエリアネットワーク上に存在する無線LANアクセスポイントであって同一のESSIDに設定されている無線LANアクセスポイントの使用チャンネル及びBSSIDを同一ネットワークの無線LANアクセスポイント情報として認識しデータとして保持し、前記無線LANクライアントは前記接続を行う際に前記無線LANアクセスポイント情報を利用することを特徴とする接続方法。

【請求項2】

前記無線LANアクセスポイントは、前記無線LANクライアントと接続された際に、自らが保持している前記無線LANアクセスポイント情報を無線LANクライアントに受け渡すことを特徴とする請求項1に記載の接続方法。

【請求項3】

前記無線LANクライアントは最初に前記無線LANアクセスポイントに接続を試みる際は前記ローカルエリアネットワーク上に存在する前記無線LANアクセスポイントの全チャンネルをスキャンし、当該スキャンをした中から最良の条件の無線LANアクセスポイントを選択し接続し、前記無線LANクライアントが前記無線LANアクセスポイントと圏外から再接続を試みる際、或いは、ローミングする際には、前記受け取った無線LANアクセスポイント情報を基に、最低限のチャンネルをスキャンし、当該スキャンをしたチャンネルの中から最良の無線LANアクセスポイントを選択し接続することを特徴とする請求項1又は2に記載の接続方法。

【請求項4】

前記最低限のチャンネルのスキャンは、前記無線LANクライアントが、前記無線LANアクセスポイント情報の中から自分の接続可能なESSIDを持つ無線LANアクセスポイントのチャンネルを抽出し、当該抽出したチャンネルのみを前記最低限のチャンネルと判断することにより行われることを特徴とする請求項1乃至3の何れか1項に記載の接続方法。

【請求項5】

複数の無線LANアクセスポイントと接続される無線LANクライアントが行う接続方法であって、

前記無線LANアクセスポイントに最初に接続を試みる際は前記ローカルエリアネットワーク上に存在する前記無線LANアクセスポイントの全チャンネルをスキャンし、当該スキャンをした中から最良の条件の無線LANアクセスポイントを選択し接続し、接続された際に前記無線LANアクセスポイントから、当該無線LANアクセスポイントと同一のローカルエリアネットワーク上に存在する無線LANアクセスポイントであって同一のESSIDに設定されている無線LANアクセスポイントの使用チャンネル及びBSSIDを同一ネットワークの無線LANアクセスポイント情報として受け取り、前記無線LANクライアントが前記無線LANアクセスポイントと圏外から再接続を試みる際、或いは、ローミングする際には、前記受け取った無線LANアクセスポイント情報を基に、最低限のチャンネルをスキャンし、その中から最良の無線LANアクセスポイントを選択し接続することを特徴とする接続方法。

【請求項6】

複数の無線LANアクセスポイントと、当該無線LANアクセスポイントと接続される無線LANクライアントと、を有する無線LAN接続システムにおいて、

前記無線LANアクセスポイントは同一のローカルエリアネットワーク上に存在する無線LANアクセスポイントであって同一のESSIDに設定されている無線LANアクセスポイントの使用チャンネル及びBSSIDを同一ネットワークの無線LANアクセスポイント情報として認識しデータとして保持し、前記無線LANクライアントは前記接続を

行う際に前記無線LANアクセスポイント情報を利用することを特徴とする接続システム

【請求項7】

前記無線LANアクセスポイントは、前記無線LANクライアントと接続された際に、自らが保持している前記無線LANアクセスポイント情報を無線LANクライアントに受け渡すことを特徴とする請求項6に記載の接続システム。

【請求項8】

前記無線LANクライアントは最初に前記無線LANアクセスポイントに接続を試みる際は前記ローカルエリアネットワーク上に存在する前記無線LANアクセスポイントの全チャンネルをスキャンし、当該スキャンをした中から最良の条件の無線LANアクセスポイントを選択し接続し、前記無線LANクライアントが前記無線LANアクセスポイントと圏外から再接続を試みる際、或いは、ローミングする際には、前記受け取った無線LANアクセスポイント情報を基に、最低限のチャンネルをスキャンし、当該スキャンをしたチャンネルの中から最良の無線LANアクセスポイントを選択し接続することを特徴とする請求項6又は7に記載の接続システム。

【請求項9】

前記最低限のチャンネルのスキャンは、前記無線LANクライアントが、前記無線LANアクセスポイント情報の中から自分の接続可能なESSIDを持つ無線LANアクセスポイントのチャンネルを抽出し、当該抽出したチャンネルのみを前記最低限のチャンネルと判断することにより行われることを特徴とする請求項6乃至8の何れか1項に記載の接続システム。

【請求項10】

複数の無線LANアクセスポイントと接続される無線LANクライアントにおいて、
前記無線LANアクセスポイントに最初に接続を試みる際は前記ローカルエリアネットワーク上に存在する前記無線LANアクセスポイントの全チャンネルをスキャンし、当該スキャンをした中から最良の条件の無線LANアクセスポイントを選択し接続し、接続された際に前記無線LANアクセスポイントから、当該無線LANアクセスポイントと同一のローカルエリアネットワーク上に存在する無線LANアクセスポイントであって同一のESSIDに設定されている無線LANアクセスポイントの使用チャンネル及びBSSIDを同一ネットワークの無線LANアクセスポイント情報として受け取り、前記無線LANクライアントが前記無線LANアクセスポイントと圏外から再接続を試みる際、或いは、ローミングする際には、前記受け取った無線LANアクセスポイント情報を基に、最低限のチャンネルをスキャンし、その中から最良の無線LANアクセスポイントを選択し接続することを特徴とする無線LANクライアント。

【請求項11】

複数の無線LANアクセスポイントと、当該無線LANアクセスポイントと接続される無線LANクライアントと、を有するシステムに搭載される無線LAN接続プログラムにおいて、

前記無線LANアクセスポイントは同一のローカルエリアネットワーク上に存在する無線LANアクセスポイントであって同一のESSIDに設定されている無線LANアクセスポイントの使用チャンネル及びBSSIDを同一ネットワークの無線LANアクセスポイント情報として認識しデータとして保持し、前記無線LANクライアントは前記接続を行う際に前記無線LANアクセスポイント情報を利用するシステムとしてコンピュータを機能させることを特徴とする接続プログラム。

【請求項12】

前記無線LANアクセスポイントは、前記無線LANクライアントと接続された際に、自らが保持している前記無線LANアクセスポイント情報を無線LANクライアントに受け渡すことを特徴とする請求項11に記載の接続プログラム。

【請求項13】

前記無線LANクライアントは最初に前記無線LANアクセスポイントに接続を試みる

際は前記ローカルエリアネットワーク上に存在する前記無線LANアクセスポイントの全チャンネルをスキャンし、当該スキャンをした中から最良の条件の無線LANアクセスポイントを選択し接続し、前記無線LANクライアントが前記無線LANアクセスポイントと圏外から再接続を試みる際、或いは、ローミングする際には、前記受け取った無線LANアクセスポイント情報を基に、最低限のチャンネルをスキャンし、当該スキャンをしたチャンネルの中から最良の無線LANアクセスポイントを選択し接続することを特徴とする請求項11又は12に記載の接続プログラム。

【請求項14】

前記最低限のチャンネルのスキャンは、前記無線LANクライアントが、前記無線LANアクセスポイント情報の中から自分の接続可能なESSIDを持つ無線LANアクセスポイントのチャンネルを抽出し、当該抽出したチャンネルのみを前記最低限のチャンネルと判断することにより行われることを特徴とする請求項11乃至13の何れか1項に記載の接続プログラム。

【請求項15】

複数の無線LANアクセスポイントと接続される無線LANクライアントに搭載される接続プログラムにおいて、

前記無線LANアクセスポイントに最初に接続を試みる際は前記ローカルエリアネットワーク上に存在する前記無線LANアクセスポイントの全チャンネルをスキャンし、当該スキャンをした中から最良の条件の無線LANアクセスポイントを選択し接続し、接続された際に前記無線LANアクセスポイントから、当該無線LANアクセスポイントと同一のローカルエリアネットワーク上に存在する無線LANアクセスポイントであって同一のESSIDに設定されている無線LANアクセスポイントの使用チャンネル及びBSSIDを同一ネットワークの無線LANアクセスポイント情報として受け取り、前記無線LANクライアントが前記無線LANアクセスポイントと圏外から再接続を試みる際、或いは、ローミングする際には、前記受け取った無線LANアクセスポイント情報を基に、最低限のチャンネルをスキャンし、その中から最良の無線LANアクセスポイントを選択し接続する無線LANクライアントとしてコンピュータを機能させることを特徴とする接続プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、無線LANに関し、より詳細には無線LANのローミング又は再接続に関する。

【背景技術】

【0002】

近年無線LANの機器の低価格化等の影響もあり、無線LANが一般に広く普及している。また、無線LANを構築する際に1つのローカルエリアネットワーク内に多数の無線LANアクセスポイントを数設するといったことも一般的に行われている。

【0003】

このような環境下において無線LANクライアントは、1つのローカルエリアネットワーク内の全ての無線LANアクセスポイントを対象として、接続可能な無線LANアクセスポイントを検索する。

【0004】

そして、無線LANクライアントは、検索の結果に基づいて最良と判断した無線LANアクセスポイントに対して接続動作をする（例えば特許文献1参照）。

【図面の簡単な説明】

【0016】

【図1】本発明の実施形態における無線LANネットワークの基本的構成の例を表す図である。

【図2】本発明の実施形態における各機器の設定の例を表す図である。

【図3】本発明の実施形態における無線LANアクセスポイント情報の例を表す図である

。

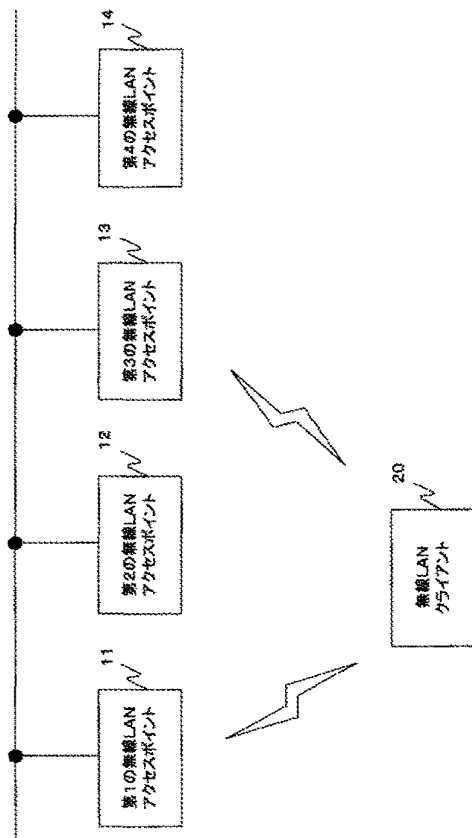
【図4】本発明の実施形態における無線LANクライアントの無線LANアクセスポイント接続動作を表すフローチャートである。

【符号の説明】

【0045】

- 1 1 第1の無線LANアクセスポイント
- 1 2 第2の無線LANアクセスポイント
- 1 3 第3の無線LANアクセスポイント
- 1 4 第4の無線LANアクセスポイント
- 2 0 無線LANクライアント

【図1】



各機器の設定			
SYSTEM NAME	BSSID	ESSID	CH
AP11	11111-11111	Office1	1
AP12	22222-22222	Office1	4
AP13	33333-33333	Office1	7
AP14	44444-44444	Office2	14
CL20	-	Office1	-

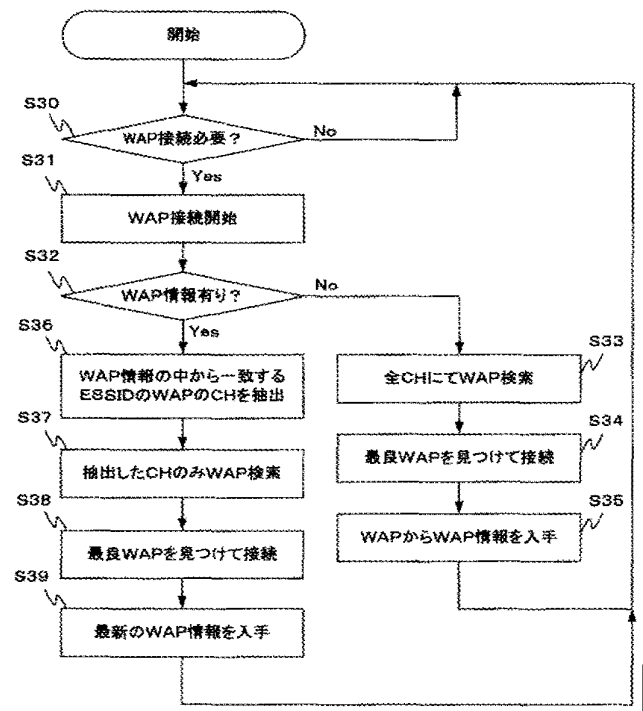
無線LANアクセスポイント情報			
SYSTEM NAME	BSSID	ESSID	CH
AP11	11111-11111	Office1	1
AP12	22222-22222	Office1	4
AP13	33333-33333	Office1	7
AP14	44444-44444	Office2	14

【※3】

(8)

特開2011-4225 (P2011-4225A)

【図4】



(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 1 298 931 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.04.2003 Bulletin 2003/14

(51) Int Cl.7: **H04N 7/24**

(21) Application number: **02396145.1**

(22) Date of filing: **20.09.2002**

(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

(72) Inventor: **Ala-Honkola, Pekka**
00180 Helsinki (FI)

(74) Representative: **Savela, Antti-Jussi**
Patent Agency Compantent Ltd.
Hämeentie 29, 4th. Fl.
00500 Helsinki (FI)

(30) Priority: **20.09.2001 FI 20011856**

(71) Applicant: **Oplayo Oy**
00180 Helsinki (FI)

(54) Adaptive media stream

(57) This invention relates to media streams transmitted in a network, such as the Internet. The idea is to change the source of the content of a media stream when an available transmission speed rate changes.

Each source contains essentially the same information (such as video and voice), but the suitability of each source for transmission has been adjusted to a certain speed rate.

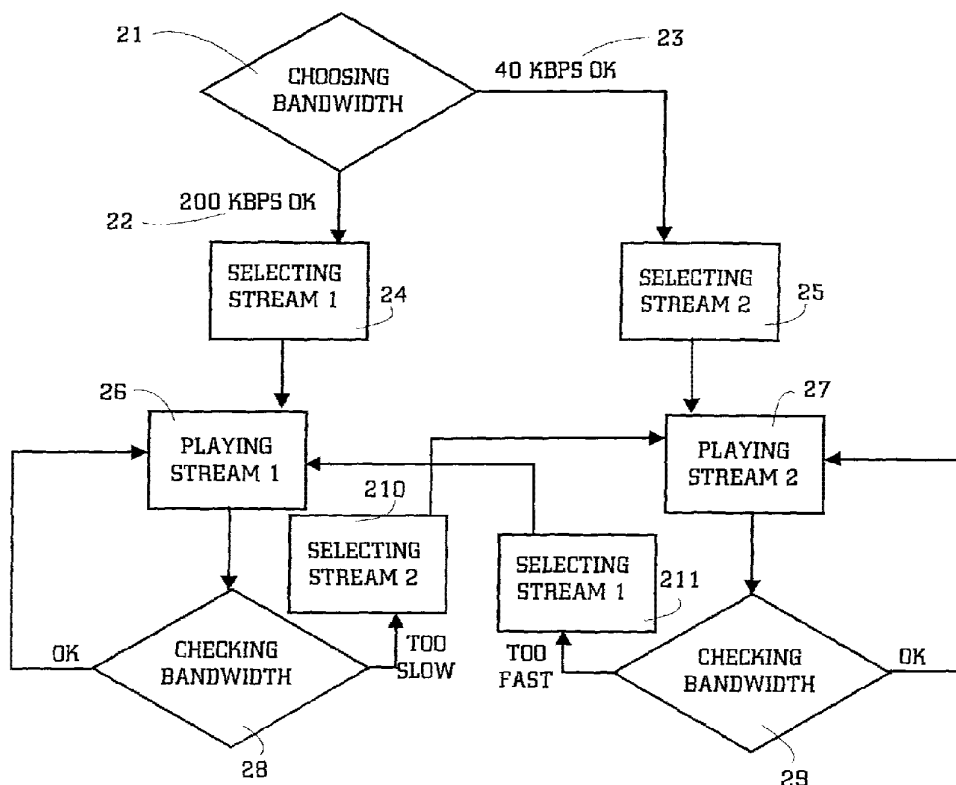


FIG. 2

EP 1 298 931 A2

Description

FIELD OF THE INVENTION

[0001] This invention relates to media streams transmitted in a network, such as the Internet. Especially, the invention relates to media streams concerning video pictures and voice.

BACKGROUND OF THE INVENTION

[0002] When media streams that contain continuous data, for example video pictures and voice, are transmitted in a network, such as the Internet, the speeds of the media streams have to be adjusted with the speed of the transmission connection available at the time. Often, in the case of video pictures, the adjustment is made by changing the visual quality of the pictures. The speed of a transmission connection in the Internet depends strongly on a user's Internet terminal. Some terminals support technically higher speed rates than others (usually older terminals). Thus, speed adjusted media streams must be offered for a variety of terminals.

[0003] Transmission speed may also vary with time. For example, when congestion happens in the network, the transmission speed may decrease considerably. After the congestion, the transmission speed returns to the normal speed rate.

[0004] Nowadays, most of the content of the Internet is in WWW servers, which use the HTTP protocol for transmitting desired contents to users. Originally, the HTTP protocol was not designed for continuous media streams (video, voice). The problem of the known solutions is that when a media stream has to be adjusted for a changed transmission condition, i.e. to be an adaptive media stream, a special software or protocol must be used, instead of the HTTP protocol. Real Server® (RealNetworks Inc, Seattle, Washington), Quicktime® Server (Apple Computers, Cupertino California), and Windows Media Server® (Microsoft, Redmond, Washington) are commercial server software, which support adaptive media streams. These software packages use special protocols, such as RTP (Real-Time Transport Protocol) and RTSP (Real Time Streaming Protocol), which are designed for the adaptive media streams.

[0005] However, using these software packages usually requires a user to invest and maintain parallel devices and software in addition to the WWW server that is used to serve other Internet content, such as web pages. User communities such as corporations require support staff which must be educated to use these systems. For a service provider this means in practice that a special media stream server must be maintained. The goal of this invention is to eliminate these drawbacks. This is achieved in a way described in the claims.

SUMMARY OF THE INVENTION

[0006] The following definitions should be kept in mind when reading these specifications:

Connection: A transport layer virtual circuit established between two programs for the purpose of communication. A connection can have varying transmission speed rates.

HTTP transport: An application layer transmission of data from a file on a server computer to a user terminal using HTTP 1.1 protocol. One HTTP 1.1 connection can consist of one or several HTTP 1.1 transports.

Stream: A continuous transmission of data over for example HTTP transport

[0007] An important aspect of the invention is to change the source of an adaptive media stream when an available connection speed rate changes. Each source contains essentially the same information (such as video and voice), but the suitability of each source for transmission has been adjusted to a certain connection speed rate. The coding of the source information can be different among the sources, which affects the speed rate. The picture information can be black-and-white instead of a color picture. The picture size can be smaller or larger. The video can be with or without voice. When the source is changed, also a new HTTP transmission for the new source is started for a new stream containing the information of the new source. In other words, the adaptive media stream, i.e. the adaptive stream, comprises at least one media stream for transmitting information.

[0008] In an arrangement according to the invention a source file is formed. The source file contains a plurality of sources intended for different media streams, according to such parameters as bandwidth, computing capability of the receiving terminal, etc. At the beginning of the transmission the optimal transmission speed rate at the time is chosen according to an information transfer condition. The source in the source file, which corresponds to the available speed rate, is selected to be the source of the adaptive media stream. When the available transmission speed rate changes, either to be faster or slower, a command for changing the source of the adaptive media stream, is sent from the user's terminal to the service provider's server, i.e. the source server, wherein the file of the service is. Preferably commands are sent using an HTTP protocol. When the available transmission speed rate changes again, a new command for changing the source is sent from the user's terminal to the source server. The commands are sent when it is possible to use a higher speed rate or when a lower speed rate must be used.

[0009] In one aspect, the invention thus provides a method for providing an adaptive media stream between a sending terminal and a user's terminal wherein

the adaptive media stream comprises at least one media stream and the method comprises the steps of choosing a transmission speed rate for the media stream, selecting a suitable data source for the media stream among sources containing essentially the same information content, each source intended for a certain information transfer condition and playing the media stream. During such playing, checking the suitability of the actual transmission speed rate, and continuing playing the media stream if the transmission speed is still suitable or selecting a new data source, which is suitable at the moment as a response for the selection of the new data source, playing a new media stream. Those selection actions continue until the played stream stops.

[0010] Optionally, the step of choosing comprises the phase of reading a header field of a file, which contains at least two data sources, the header indicating the starting points of the sources and to which transmission speed rate each source is indented. Yet another option calls for the selecting step to include the phase of starting to read from the starting point of the selected source.

[0011] Other aspect of the invention provide for the step of monitoring to include the phase of monitoring a filling degree of a buffer for incoming data in the user's terminal, e.g. if the buffer is becoming empty or full, then selecting to continue to play the present media stream or to start playing another media stream. The information of the selection of the new data source may be transmitted from the user's terminal to the sending terminal using a GET method of an HTTP protocol with an optional RANGE field. When a new stream is played, an optional phases of terminating the old media stream after starting to play the new media stream and creating the new media stream before playing the new media stream is also contemplated.

[0012] If a new stream is selected, according to another aspect of the invention, the step of reading timing marks in said adaptive media stream for defining the right point to start playing the new media stream is also performed. Further, the step of selecting the initial speed may comprise the phase of reading a header field of files, where each contains one data source, the header indicating the starting points of the source and to which transmission speed rate the source is indented.

[0013] An arrangement for providing an adaptive media stream between a sending terminal and a user's terminal, is also contemplated, wherein the adaptive media stream comprises at least one media stream and the arrangement comprises:

- a) Means in the client terminal for choosing a transmission speed rate for the media stream,
- b) A data structure in the sending terminal comprising at least two data sources containing essentially the same information, each source intended for a certain transmission speed rate,
- c) Means in the client terminal and the sending terminal for selecting a suitable data source from

among the sources for the media stream,

d) Means in the user's terminal for checking the suitability of the transmission speed rate when the media stream is played,

e) Means in the user's terminal for reselecting a suitable data source from among the sources for a new media stream.

[0014] An arrangement as described above may further comprise means for playing the data structure in a way that the playing is started from the right point of the data structure. Preferably, means a) comprises means for reading a message from the user's terminal, which defines alternative transmission speed rates. Further preferably, each source comprises timing marks, which indicate the corresponding point in the other sources. More preferably, the arrangement further comprises a GET method of an HTTP protocol with an optional RANGE field for transmitting the information of the reselection of the suitable data source from the user's terminal to the sending terminal. In another preferable embodiment, means d) comprises means for monitoring a filling degree of a buffer for incoming data in the user's terminal. It also preferable that the data structure is a file containing the data sources. Alternatively, the data structure may be a number of files, each of them containing one data source.

[0015] Means c) may further comprise means for reading a header of the data structure, which indicates the starting points of the sources and to which transmission speed rate each source is intended. Means c) may further comprise means for reading headers of the files, which indicate the starting point of the source and to which transmission speed rate the source is intended.

[0016] In yet another aspect, the invention teaches a method for storing adaptive media stream data comprising the steps of storing a plurality of media streams, each representing a substantially similar information content, and each adapted for a different set of information transfer conditions; and providing a plurality of pointers into a first of said media streams, said pointers associated with a source location in one of said media streams and pointing to at least one target location in a second of said media streams; wherein the information content of said target location is substantially a continuation of the information content prior to said source location.

[0017] Preferably, said plurality of media streams is contained in a single file. Said pointers are preferably embedded within said adaptive media stream data, and comprise target locations pointer comprises a byte offset into said single file. Alternatively, said target location pointer comprises an offset into said second source stream, within said single file. Another alternative teaches that said target locations pointers comprise a pointer to another file containing said second media stream or a portion thereof. Optionally in this case, said target location pointer further comprises a byte offset into said

other file.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] In the following the invention is described in more detail by means of Figs. 1 - 3 in the attached drawings where.

- FIG. 1 illustrates an example of the structure of a file according to the invention,
 FIG. 2 illustrates an example of a flow chart describing the function of the inventive arrangement,
 FIG. 3 illustrates an example of an arrangement according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0019] FIG. 1 describes an example of a file structure according to the invention. The file is in the service provider's server (source server) from where users can stream it. For example the file contains a video in two different formats. At the beginning of the file there exists a header 1, which contains starting bytes for two different media sources - the sources for the media stream. Source 1, called stream 1 contains video data in a resolution of 320*240, for a speed rate 200 kbps (kilobit per second) 2. The starting byte 4 for that video data is byte 2000 in the file. Source 2, called stream 2 contains video data in a resolution of 160*120 optimized for a speed rate 40 kbps 3. The starting byte of that video data 5 in the file is at byte 800000. If the network and a user's terminal support the transmission speed rate 200 kbps then stream 1 is sent to the user - otherwise stream 2 is sent. In this embodiment, all data feeds are assumed to support communication at a speed of at least 40 kbps. It should be noted that stream 1 and 2 are sources i.e. the parts of the file structures in this example, and a media stream is a continuous transmission of data of a source over a transmission protocol. The transmission protocol used can be any protocol, in this text the HTTP protocol is used.

[0020] Stream 1 is sent, i.e. streamed, to the user. Stream 1 (and the media stream for it as well) contains timing marks 6, 7, 8, 9. The timing marks are embedded in stream 1 at a various locations, and correlate the information in the two separate resolution streams. Thus the timing marks of stream 1 in the example of Fig. 1 indicate the byte of the file where the stream is running at the time and the corresponding byte 6A, 7A, 8A, 9A, in stream 2. If for some reason, the network can not transmit stream 1 with 200 kbps, or the user's terminal can not receive at this speed rate, the transmission of the video is changed to stream 2. For example, when the need to change the speed rate to a lower one is noticed and the most recently received (or alternatively the next) timing mark is, let's say, the mark 7 indicating time 1000 ms, it tells the corresponding byte 805000 in stream 2 to where the source of the transmission of the

video transfers 11.

[0021] Similarly, stream 2 (and the media stream for it) contains timing marks 10, 11, 12, 13, which point to the location of corresponding information in stream 1, as shown by the corresponding byte 10A, 11A, 12A, 13A, in stream 1. When the capability of transmitting 200 kbps is noticed during the transmission at 40 kbps rate, the transmission of the video is changed to stream 1. For example, when the availability of the higher speed rate is noticed and the next timing mark is the mark 13 indicating time 3000 ms, the corresponding byte 77000, located in stream 1, points to the location of corresponding information in stream 1 so the source of transmission is shifted to stream 1, using data stored in location 9 onwards.

[0022] FIG. 2 shows an example of a flow chart, which illustrates a function and method of the invention. FIG. 3 shows an example of the inventive arrangement. When a user visits a home page of a service provider, he/she may find something he wants to order, such as a video show. The video player client running in the user terminal sends 34 a request message from the terminal 33 to the service provider's server 31 through the network 32, such as the Internet, to get the show using a GET method, which is one of the methods defined in the HTTP1.1 protocol. Let the request be:

GET http://www.serviceprovider.com/video/music/song1.video. HTTP/1.1

[0023] Often, the connection speed rate depends on the available (and optional) connection and processing speeds of the terminal. At the beginning, the first stream which is sent works like a test stream. The test stream, i.e. the first stream preferably forms the first part of the file structure, which preferably contains data for the lowest bandwidth. Thus the first stream over the HTTP transport also contains a list of alternative speed rates. The player in the client terminal chooses 21 one of the supported speed rate from the list. (It should be noted that the server can be informed of the capability of a terminal or the available bandwidth using other techniques and the specific method of learning the link capabilities is a matter of technical choice to one skilled in the art.) In this example, the user's terminal supports speed rates of 200 kbps and 40 kbps. Let's assume that the desired file 36 is the file in FIG. 1. If the speed rate of 200 kbps 22 is available, stream 1 is selected 24. Otherwise, the speed rate of 40 kbps 23 is chosen and stream 2 is selected 25. (Notice that the higher speed rate means the higher bandwidth.)

[0024] In the selection of the speed rate of 200 kbps the server starts to play 26 stream 1 beginning at byte (2000) of the file. The data and the timing marks are sent 35 to the user's terminal during the play. The terminal checks 28 frequently that the currently available bandwidth is sufficient for the selected stream. If the bandwidth is sufficient, playing stream 1 is continued. If the bandwidth becomes unavailable for stream 1, stream 2 is selected 210 and stream 2 starts to play 27, 311 be-

ginning at the byte of the file defined by the latest timing mark. The bandwidth of stream 2 is checked **29** similarly as the bandwidth of stream 1, and when the bandwidth is again available for the speed rate of 200 kbps, stream 1 is selected **211**.

[0025] In the selection of the speed rate of 40 kbps the server starts to play **25** stream 2 from the right byte (800000) of the file. The data and the timing marks are sent **35** to the user's terminal during the play. The terminal checks **29** frequently that the bandwidth is okay for the selected stream. If the bandwidth is still restricted to 40 kbps, playing stream 2 is continued. If the bandwidth rise to a level sufficient to support 200 kbps transmission, stream 1 is selected **211** and stream 1 starts to play **26**, and is sent **311** to the terminal **33** beginning at the byte of the file defined by the latest timing mark. The bandwidth of stream 1 is checked **28** similarly as the bandwidth of stream 2, and when the bandwidth is again too slow for the speed rate of 200 kbps, stream 2 is selected **210**.

[0026] The checking of the bandwidth of the streams can be made, for example, measuring a filling degree of a buffer for incoming data connection in the user's terminal. When the buffer is becoming empty, the required bandwidth for current stream is too high compared to what is available through the network connection, and the speed rate should be dropped. When the buffer is becoming full, bandwidth for current stream is lower than what is available through the network connection, and the speed rate can be raised. After the measurement, the suitable stream source is selected **210**, **211** and a GET command with an additional field RANGE is sent **310** from the user's terminal to the server for changing the stream. For example, let the GET command be GET <http://www.serviceprovider.com/video/music/song3video> RANGE: bytes= 805000- when the transmission speed rate changes from 200 kbps to 40 kbps.

[0027] It is worth noting that preferably, when a stream is changed, a new HTTP transport (and naturally a new media stream) is started and the old media stream is terminated after the new media stream starts playing. From the view of the user the desired file, such as a video show, seems to continue without any break because the new stream starts playing from the byte indicated by the timing mark, thus the information continuity is maintained, despite the media stream switch. So, when the source is changed, also a new HTTP transport is created for the new stream, i.e. media stream, instance and for the new source. Thus the adaptive media stream, i.e. the adaptive stream, is made up of one media stream or several consecutive media streams. Since the consecutive media streams carry essentially the same data, a user gets a visual impression of continuous shows. In other words, the adaptive media stream comprises at least one media stream for transmitting information. The HTTP 1.1 protocol provides a feature called Persistent Connections. Persistent Connection means that several requests (such as GET), can be sent via one TCP con-

nection. Thus one connection may carry several consecutive streams. However, it should be noted that several TCP connections may be needed or used for transmitting consecutive streams.

[0028] Furthermore, it may be said that the adaptive media stream comprises several (or one if the speed rate remains the same during the whole transmission) instances (separate streams) of the same object (the adaptive media stream containing information such as video). Naturally, there are more ways to model the adaptive media stream, but common for these models are that transmission of information of the adaptive stream, (the stream of a continuous data, such as video over one or several connections) happens in one or several consecutive media streams over a transport protocol, each media stream adapted for certain conditions, and certain links or pointers periodically connect or cross connect the media streams in essentially content coherent fashion. Thus varying transmission conditions may dictate how many suitable media streams are sent forming together the adaptive stream.

[0029] For creating the inventive arrangement an initial selection module **39** (i.e. means) for choosing an original transmission speed rate and selecting a suitable source in a service provider's server is needed. In a user's terminal another estimation module **37** (i.e. means for checking transmission rate) is needed, for checking the suitability of the transmission speed rate as reselection module **38** (i.e. means) for reselecting a suitable source are needed.

[0030] Although, the processing and network connection speed of a user's terminal is mostly the decisive element when selecting a suitable transmission speed rate, it may happen that the network itself becomes congested. In congestion, the available speed rate decreases and the transmission speed rate must be lowered. On the other hand, if the network allows a higher speed, the transmission speed rate may be increased.

[0031] Using the inventive arrangement, an adaptive media stream can be served from with a common HTTP server. Therefore, there is no need for the use of a special media stream server, or parallel devices and software. Remarkable cost savings are achieved, since usually service providers and users support the HTTP protocol. Also the reliability of the system increases because of fewer sources of faults.

[0032] It is clear that inventive arrangement can be implemented in many ways. For example, the time period between two timing marks can be different than in the example of this text. It can be any suitable period depending on the features of a solution, such as 2 or 4 seconds. The format of the inventive file can be any suitable format as well, such as MPEG or another.

[0033] The inventive file may comprise more than two different streams (containing the same content), from which the most suitable stream is selected at any given time. Therefore, timing marks must contain a number of references for the corresponding byte in the other data

sources.

[0034] All the sources do not have to be in the same file, but each source can be a separate file. However, the use of this kind of structure is not as efficient when attempting to use as little processing power as possible compared to the use of one file structure. In the structure of several files, timing marks must indicate the right file with the corresponding byte.

[0035] Although, a sending terminal have been described to be a server in the examples above, it is clear that the sending terminal can be another terminal capable of sending information.

[0036] Similarly other mechanisms are known and suitable to measure and select the most appropriate source. Such methods may be server based or user terminal based, or even utilize intermediate network nodes to estimate network congestion and bandwidth, and similar measurements that are indicative of total node to node capacity, as well as specific node capabilities, and the selection of sources (and media streams) may be carried out in various ways that will provide the best end-user experience according to varying conditions.

[0037] According to the above-mentioned matters, it is evident that the invention is not restricted to the solutions described in this text, but it can be modified for various solutions, in the scope of the inventive idea.

Claims

1. A method for providing an adaptive media stream between a sending terminal and a user's terminal **characterized in that** the adaptive media stream comprises at least one media stream and the method comprises the steps of:

- a) choosing a transmission speed rate for the media stream,
- b) selecting a suitable data source for the media stream among sources containing essentially the same information, each source intended for a certain transmission speed rate,
- c) playing the media stream,
- d) checking the suitability of the transmission speed rate,
- e) either continuing playing the media stream if the transmission speed is still suitable or,
- f) selecting a new data source, which is suitable at the moment,
- g) as a response for the selection of the new data source, playing a new media stream,
- h) repeating steps d) to g) until the stream, which is played, stops.

2. A method according to claim 1 **characterized in that** step b) comprises the phase of reading a header field of a file, which contains at least two data sources, the header indicating the starting points of

the sources and to which transmission speed rate each source is intended.

3. A method according to claim 2 **characterized in that** step c) comprises the phase of starting to read from the starting point of the selected source.
4. A method according to claim 1 or 3 **characterized in that** step d) comprises the phase of monitoring a filling degree of a buffer for incoming data in the user's terminal.
5. A method according to claim 4 **characterized in that** if the buffer is becoming empty or full, then step f) is chosen, else step e) is chosen.
6. A method according to claim 1 or 5 **characterized in that** the information of the selection of the new data source is transmitted from the user's terminal to the sending terminal using a GET method of an HTTP protocol with an optional RANGE field.
7. A method according to claim 1 **characterized in that** step g) comprises phases of terminating the old media stream after starting to play the new media stream and creating the new media stream before playing the new media stream.
8. A method according to claim 1 or 7, **characterized in that** step g) further comprises the phase of reading timing marks in the data source for defining the right point to start playing the new media stream.
9. A method according to claim 1 **characterized in that** step b) comprises the phase of reading a header field of files, where each contains one data source, the header indicating the starting points of the source and to which transmission speed rate the source is intended.
10. An arrangement for providing an adaptive media stream between a sending terminal and a user's terminal, **characterized in that** the adaptive media stream comprises at least one media stream and the arrangement comprises:
 - a) means in the sending terminal for choosing a transmission speed rate for the media stream,
 - b) a data structure in the sending terminal comprising at least two data sources containing essentially the same information, each source intended for a certain transmission speed rate,
 - c) means in the sending terminal for selecting a suitable data source from among the sources for the media stream,
 - d) means in the user's terminal for checking the suitability of the transmission speed rate when the media stream is played,

e) means in the user's terminal for reselecting a suitable data source from among the sources for a new media stream.

11. An arrangement according to claim 10, **characterized in that** the arrangement further comprises means for playing the data structure in a way that the playing is started from the right point of the data structure. 5
- 10
12. An arrangement according to claim 11, **characterized in that** means a) comprises means for reading a message from the user's terminal, which defines alternative transmission speed rates. 15
13. An arrangement according to claim 11, **characterized in that** each source comprises timing marks, which indicate the corresponding point in the other sources. 20
14. An arrangement according to claim 13, **characterized in that** the arrangement further comprises a GET method of an HTTP protocol with an optional RANGE field for transmitting the information of the reselection of the suitable data source from the user's terminal to the sending terminal. 25
15. An arrangement according to claim 11, **characterized in that** means d) comprises means for monitoring a filling degree of a buffer for incoming data in the user's terminal. 30
16. An arrangement according to claim 11, **characterized in that** the data structure is a file containing the data sources. 35
17. An arrangement according to claim 16, **characterized in that** means c) comprises means for reading a header of the data structure, which indicates the starting points of the sources and to which transmission speed rate each source is intended. 40
18. An arrangement according to claim 11, **characterized in that** the data structure is a number of files, each of them containing one data source. 45
19. An arrangement according to claim 18, **characterized in that** means c) comprises means for reading headers of the files, which indicate the starting point of the source and to which transmission speed rate the source is intended. 50

55

BYTE		
0	HEADER	1
2	STREAMS	
3	1. 320*240 200KBPS 2000	
	2. 160*120 40 KBPS 800000	
2000	STREAM 1	
2100	TIME 10 MS, STREAM 2 @ 800100	6A
.	DATA	
27000	TIME 1000 MS, STREAM 2 @ 805000	7A
.	DATA	
52000	TIME 2000 MS, STREAM 2 @ 810000	8A
.	DATA	
77000	TIME 3000 MS, STREAM 2 @ 815000	9A
.	DATA	
800000	STREAM 2	
800100	TIME 10 MS, STREAM 1 @ 2100	10A
.	DATA	
805000	TIME 1000 MS, STREAM 1 @ 27000	11A
.	DATA	
810000	TIME 2000 MS, STREAM 1 @ 52000	12A
.	DATA	
815000	TIME 3000 MS, STREAM 1 @ 77000	13A
.	DATA	

FIG. 1

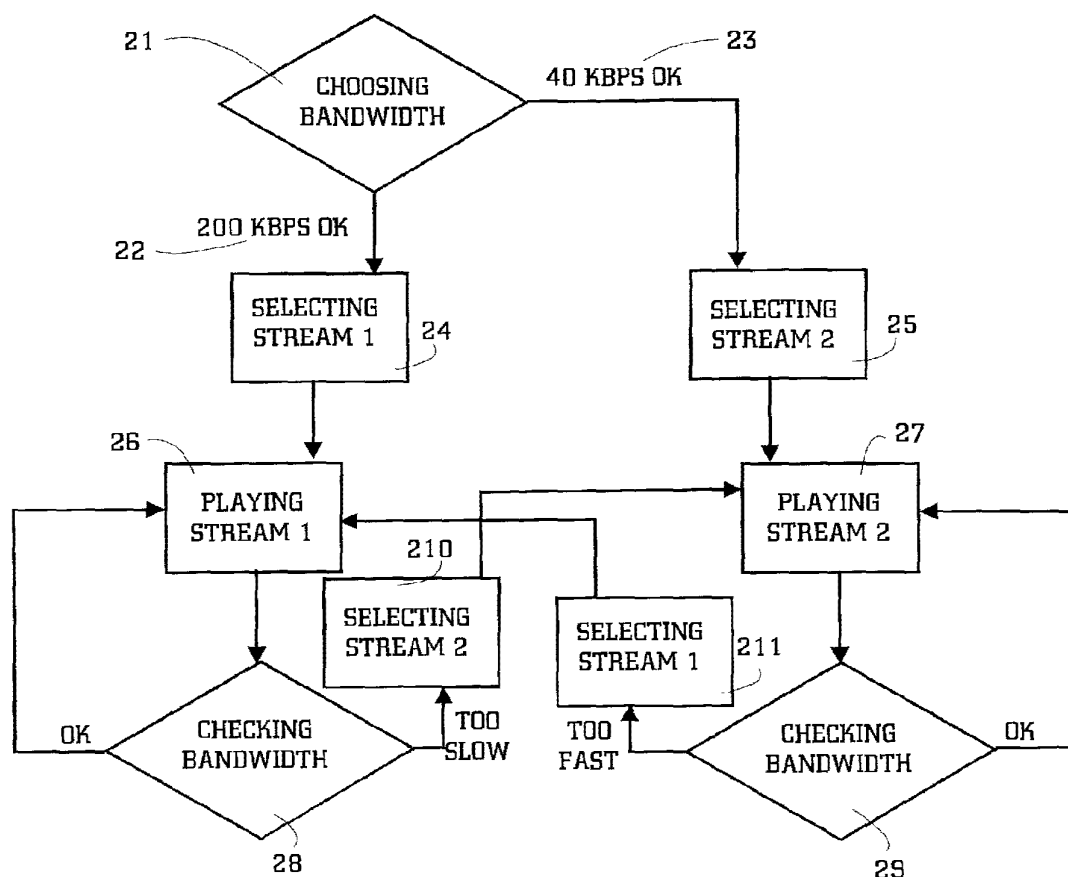


FIG. 2

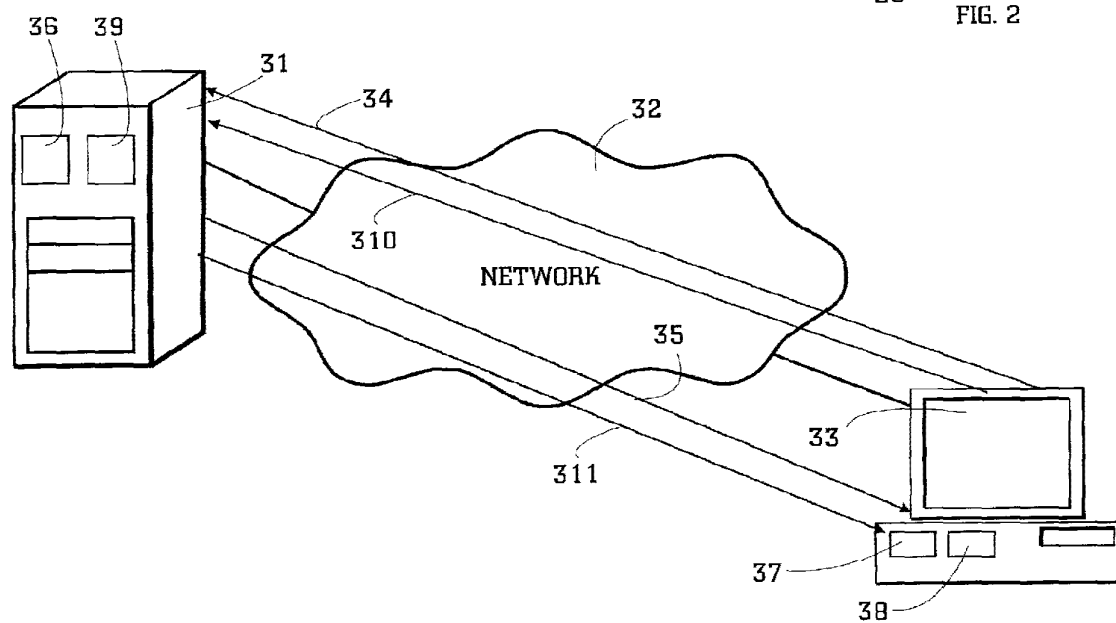


FIG. 3

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
25 March 2004 (25.03.2004)

PCT

(10) International Publication Number
WO 2004/025405 A2

(51) International Patent Classification⁷: **G06F**
(21) International Application Number:
PCT/US2003/028275

(22) International Filing Date:
10 September 2003 (10.09.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/409,645 10 September 2002 (10.09.2002) US

(71) Applicant (for all designated States except US): **THOMSON LICENSING S.A.** [FR/FR]; 46, Quai A. Le Gallo, F-92648 Boulogne (FR).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **BOYCE, Jill, MacDonald** [US/US]; 3 Brandywine Court, Manalapan, NJ 07726 (US). **RAMASWAMY, Kumar** [IN/US]; 71 Sayre Drive, Princeton, NJ 08540 (US).

(74) Agents: **TRIPOLI, Joseph, S.** et al.; Thomson Licensing Inc., Suite #200, Two Independence Way, Princeton, NJ 08540 (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, EG, 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, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

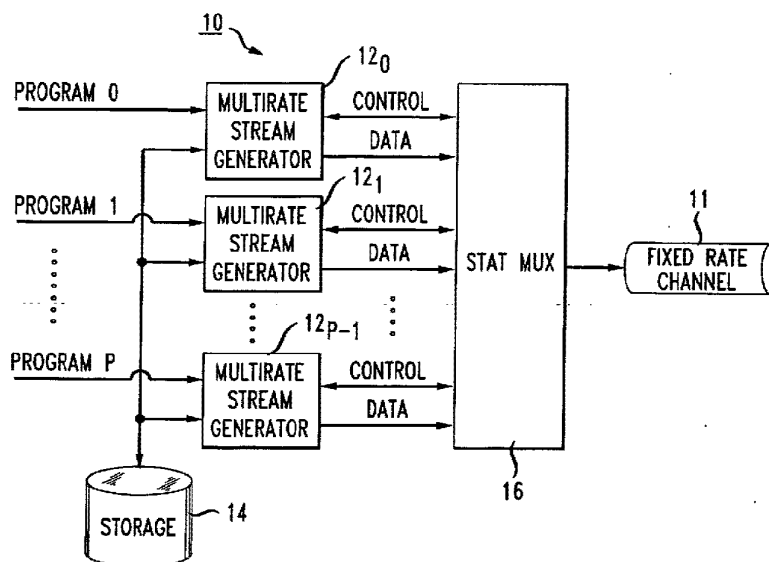
(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ,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

Published:

— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: VIDEO ON DEMAND SERVER SYSTEM AND METHOD



(57) Abstract: A Video-on-Demand Server architecture (100, 200, 300) transmits a plurality of pre-coded programs having different bit rates across a fixed bandwidth channel (11). For each program, a generator (120-12P-1, 110, 210) generates a plurality of different bit rate representations for each program. Each generator also provides control information at each of a plurality of successive time windows T for each bit rate representation. The control information provides a bit rate and a quality measure during each time window T. The control information enables a statistical multiplexer (16, 160, 260) to select a bit rate representation for each program during each time window T to maximize the quality of the selected representations while not exceeding the total available channel capacity.

WO 2004/025405 A2



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 ON DEMAND SERVER SYSTEM AND METHOD

TECHNICAL FIELD

5 This invention relates to a technique for efficiently providing video on demand while maintaining bandwidth constraints of a fixed channel.

BACKGROUND ART

10 Present day Video-on-Demand (VoD) systems typically transmit Constant Bit Rate (CBR) video. Transmitting CBR video does not achieve the same efficiency as transmitting variable bit rate (VBR) video for equivalent long-term average bit rates. Transmitting multiple pre-recorded VBR-encoded video programs over a constant bandwidth channel has problems since the sum of the instantaneous bitrates required by each program can sometimes exceed the
15 total available bitrate. On the other hand, broadcast applications employing real-time encoders use statistical multiplexing to exploit the instantaneous bitrate variations between multiple programs being carried on the same carrier or transponder. The encoders operate in real time to enable the system to constrain the total combined bit rate for all programs so as not to exceed the channel capacity over a given time window. Each video program has an associated complexity
20 measure. A central controller dynamically adjusts the bit rate allocated to each video program based upon the relative complexities.

 There now exist devices known as "video transraters" that operate to reduce the bit rate of video streams within a common compression standard, such as MPEG 2 for example. Present day transraters often experience difficulties when converting between constant and variable bit
25 rates. Such transraters also can experience difficulties when trying to alter the bit rate for bit streams that have many scene changes, or a large number of I-Pictures.

 Scalable video encoding, which permits dividing a video signal into a base layer and one or more enhancement layers, can also address bit rate issues. Several methods of scalable video encoding exist, including spatial, SNR, temporal, data partitioning, fine grain scalability (FGS),
30 frequency scalability. The MPEG-2 and MPEG-4 video compression standards include several scalability methods. Using scalable encoding requires both the transmitter and receiver have the same ability to implement different coding algorithms, thus introducing additional complexity.

- 2 -

Thus, there exists a need for a technique for managing variable bit rate video on demand that obviates the disadvantages of the prior art.

BRIEF SUMMARY OF THE INVENTION

5

Briefly, in accordance with present principles, there is provided a method for transmitting a plurality of pre-coded programs having different bit rates across a fixed bandwidth channel. For each program, at least two, and preferably, a plurality of different bit rate representations are generated. Control information is provided at each of a plurality of successive time windows for each bit rate representation. The control information provides a bit rate and a quality measure during each time window. The control information enables selection during each time window of a bit rate representation for each program to maximize the quality of the selected representations while not exceeding the total available channel capacity in that time window.

15 BRIEF SUMMARY OF THE DRAWING

FIGURE 1 depicts a block schematic of first preferred embodiment of a Video-on-Demand server architecture in accordance with the present principles;

FIGURE 2 depicts a block schematic of second preferred embodiment of a Video-on-Demand server architecture in accordance with the present principles; and

FIGURE 3 depicts a block schematic of third preferred embodiment of a Video-on-Demand server architecture in accordance with the present principles.

DETAILED DESCRIPTION

25

Before proceeding to describe the technique of the present principles, the following terminology will prove helpful:

T is the time interval over which the system optimization is being contemplated

C is the total channel capacity available in time frame T

P is the total number of programs

$p \in (0, P-1)$, is the index of a particular program

$N[p]$ is the total number of representations of program p