Doc Code: PA..

Document Description: Power of Attorney

PTO/AIA/82A (07-12)

Approved for use through 11/30/2014. OMB 0651-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

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

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	066.0141X1C8

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.

[X] *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.

Doc Code: PA.
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

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 1.46(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)	7/6/2016
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 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.

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.

fuboTV Media Inc.
Exhibit 1002
Page 003

[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

fuboTV Media Inc.
Exhibit 1002
Page 004

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 form 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.

fuboTV Media Inc.
Exhibit 1002
Page 005

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

fuboTV Media Inc.
Exhibit 1002
Page 006

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

fuboTV Media Inc.
Exhibit 1002
Page 007

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.

fuboTV Media Inc.
Exhibit 1002
Page 008

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;

fuboTV Media Inc.
Exhibit 1002
Page 009

[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.

fuboTV Media Inc.
Exhibit 1002
Page 010

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-

fuboTV Media Inc.
Exhibit 1002
Page 011

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

fuboTV Media Inc.
Exhibit 1002
Page 012

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

fuboTV Media Inc.
Exhibit 1002
Page 013

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

fuboTV Media Inc.
Exhibit 1002
Page 014

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

fuboTV Media Inc.
Exhibit 1002
Page 015

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

fuboTV Media Inc.
Exhibit 1002
Page 016

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

fuboTV Media Inc.
Exhibit 1002
Page 017

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

fuboTV Media Inc.
Exhibit 1002
Page 018

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.

fuboTV Media Inc.
Exhibit 1002
Page 019

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

fuboTV Media Inc.
Exhibit 1002
Page 020

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,

fuboTV Media Inc.
Exhibit 1002
Page 021

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

fuboTV Media Inc.
Exhibit 1002
Page 022

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

fuboTV Media Inc.
Exhibit 1002
Page 023

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.

fuboTV Media Inc.
Exhibit 1002
Page 024

[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.

fuboTV Media Inc.
Exhibit 1002
Page 025

[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

fuboTV Media Inc.
Exhibit 1002
Page 026

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

fuboTV Media Inc.
Exhibit 1002
Page 027

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:

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

fuboTV Media Inc.
Exhibit 1002
Page 028

[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.

fuboTV Media Inc.
Exhibit 1002
Page 029

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.

fuboTV Media Inc.
Exhibit 1002
Page 030

Electronic Acknowledgement Receipt

EFS ID:	34906155
Application Number:	16252356
International Application Number:	
Confirmation Number:	8631
Title of Invention:	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING
First Named Inventor/Applicant Name:	DAVID B. BRUECK
Customer Number:	70560
Filer:	Brett Carlson/Julie Graf
Filer Authorized By:	Brett Carlson
Attorney Docket Number:	066.0141X1C8
Receipt Date:	18-JAN-2019
Filing Date:	
Time Stamp:	18:35:58
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	012219INTEFSW18370300
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_066-0141X1C8_P2007-02-01-2-6.pdf	1823945 c3397c648a7e7685bc741281757b92041ef42893	no	10
Warnings:					
Information:					
2		20190118_ConAppl_066-0141X1C8_P2007-02-01-2-6.pdf	14767628 178b5feea9f35412347c053df58e97f8e28d5ad6	yes	36

Multipart Description/PDF files in .zip description

Document Description	Start	End
Specification	1	28
Claims	29	35
Abstract	36	36

Warnings:					
Information:					
3	Oath or Declaration filed	20190118_Declaration_066-0141X1C8_P2007-02-01-2-6.pdf	135238 f4fcbd8227a96f820204ed6db7f76b2f6a989962	no	3
Warnings:					
Information:					
4	Power of Attorney	20190118_POA_066-0141X1C8_P2007-02-01-2-6.pdf	113855 dbe0ed23b6e7e1711bb03467ce03bd8ac2514caf	no	2
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:					

5	Preliminary Amendment	20190118_PreliminaryAmendment_066-0141X1C8_P2007-02-01-2-6.pdf	152775 9b3735415767580cb5a4181d2b692320f84817cb	no	4
Warnings:					
Information:					
6	Drawings-only black and white line drawings	20100118_Dwgs_066-0141X1C8_P2007-02-01-2-6.pdf	130400 f83697378d2ba053dda2ad53a400c460df5ec2aa	no	11
Warnings:					
Information:					
7	TrackOne Request	20190118_Track1-PrioritizedExam_066-0141X1C8_P2007-02-01-2-6.pdf	140726 5fae76a765f7069959950baf00cf6527ecab4e3a	no	2
Warnings:					
Information:					
8	Fee Worksheet (SB06)	fee-info.pdf	42903 828a4e3057adf794993d1032562755501daf5f8f0	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	17307470		

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.

New Applications Under 35 U.S.C. 111

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.

National Stage of an International Application under 35 U.S.C. 371

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.

New International Application Filed with the USPTO as a Receiving Office

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.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	066.0141X1C8
	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 Remove

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	State/Province	Country of Residence
SARATOGA SPRINGS	UT	US

Mailing Address of Inventor:

Address 1	962 NORTH RATTLER ROAD
Address 2	
City	SARATOGA SPRINGS
State/Province	UT
Postal Code	84043
Country	US

Inventor 2 Remove

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	State/Province	Country of Residence
CEDAR HILLS	UT	US

Mailing Address of Inventor:

Address 1	10639 NORTH AVALON
Address 2	
City	CEDAR HILLS
State/Province	UT
Postal Code	84062
Country	US

Inventor 3 Remove

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

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	066.0141X1C8
	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. Add

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	Add Email	Remove Email

Application Information:

Title of the Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING		
Attorney Docket Number	066.0141X1C8	**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	066.0141X1C8
	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	16252188	2019-01-18
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
16252188	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

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	066.0141X1C8
	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)
14106051	Continuation of	13617114	2012-09-14	8612624	2013-12-17
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 in part 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[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	066.0141X1C8
	Application Number	
Title of Invention	APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING	

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

☐ 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	066.0141X1C8
	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.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	066.0141X1C8
	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 — Remove

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.

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:

Name of the Deceased or Legally Incapacitated Inventor:

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. 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.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	066.0141X1C8
	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.

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. Add

Signature:

Remove

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. Add

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	066.0141X1C8
	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 CooperationTreaty.

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:

US Patents:	US Patent Application
US7818444	Serial No. 15/414,025
US8402156	
US8612624	
US9071668	
US9571551	

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

fuboTV Media Inc.
Exhibit 1002
Page 044

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

Respectfully submitted by:

David F. Brueck — Date: 27 Oct 2017

Mark B. Hurst — Date

R. Drew Major — Date

fuboTV Media Inc.
Exhibit 1002
Page 045

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

fuboTV Media Inc.
Exhibit 1002
Page 046

Doc Code: TRACK1.REQ
Document Description: TrackOne Request

PTO/SB/424 (12-11)

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):	066.0141X1C8
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.

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 B. BRUECK
Filer:	Brett Carlson/Julie Graf
Attorney Docket Number:	066.0141X1C8

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**

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 and a high quality stream, the low 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 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 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.

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 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:

fuboTV Media Inc.
Exhibit 1002
Page 051

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 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 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 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 and the high quality stream; and

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

fuboTV Media Inc.
Exhibit 1002
Page 052

8. The system of claim 7, wherein the first streamlets of the low 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 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 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 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).

fuboTV Media Inc.
Exhibit 1002
Page 053

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 and a high quality stream, each of the low 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 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 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 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 and the high quality stream, the second

fuboTV Media Inc.
Exhibit 1002
Page 054

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 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 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 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 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.

fuboTV Media Inc.
Exhibit 1002
Page 055

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 and a high quality stream, wherein the low 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 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 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;

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 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 and the high quality stream; and

sending the retrieved first streamlet from the currently selected one of the low 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 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.

fuboTV Media Inc.
Exhibit 1002
Page 056

29. The process of claim 26, wherein the first streamlets of the low 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 and a high quality stream, wherein the low 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 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 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;

selecting, by the content player device, a currently selected one of the low 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.

fuboTV Media Inc.
Exhibit 1002
Page 057

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.

fuboTV Media Inc.
Exhibit 1002
Page 058

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.

fuboTV Media Inc.
Exhibit 1002
Page 059

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.

fuboTV Media Inc.
Exhibit 1002
Page 060

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.*

fuboTV Media Inc.
Exhibit 1002
Page 061

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

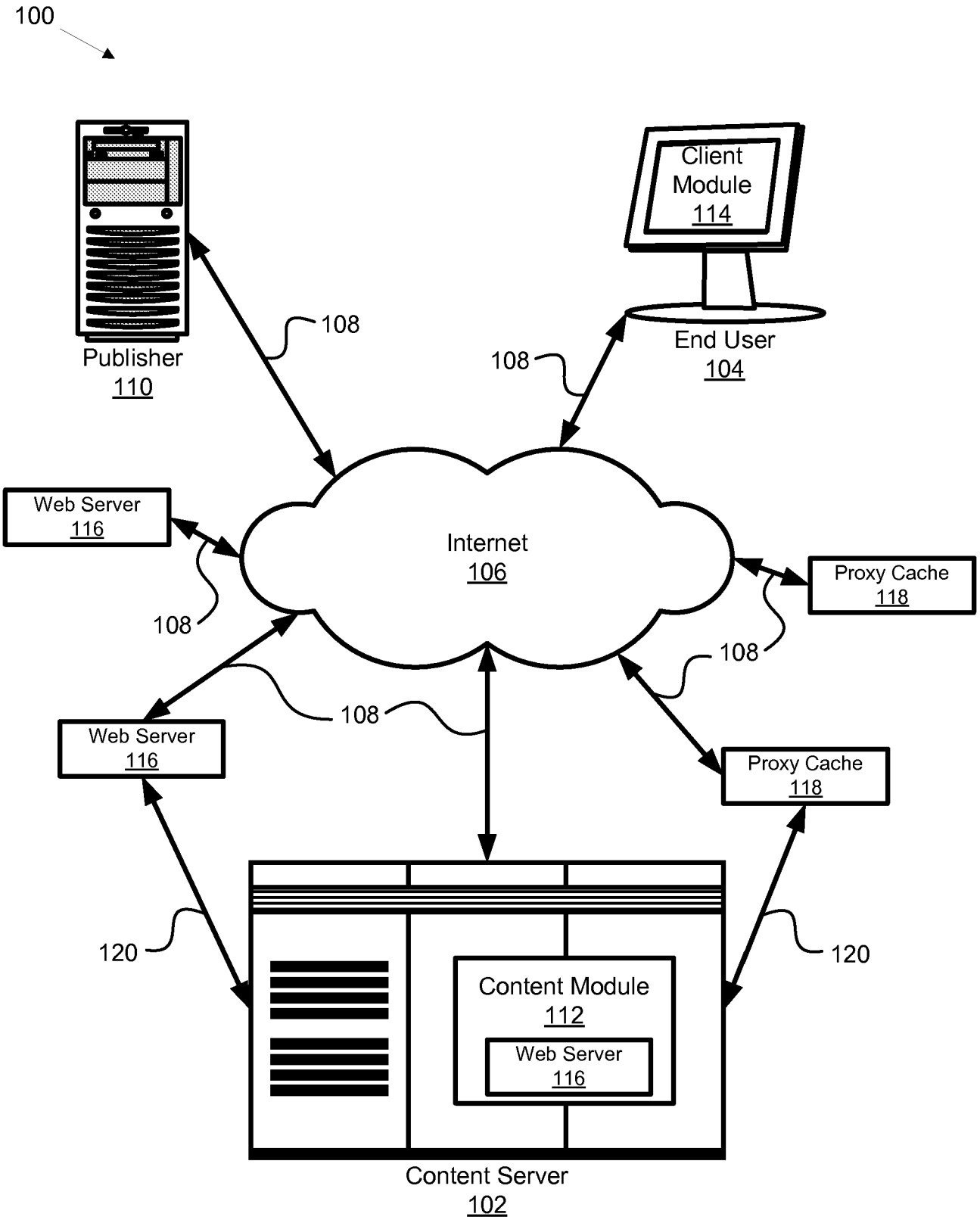

100
Publisher
110
108
Client
Module
114
End User
104
108
Web Server
116
Internet
106
Proxy Cache
118
108
108
108
Web Server
116
Proxy Cache
118
120
120
Content Module
112
Web Server
116
Content Server
102

FIG. 1

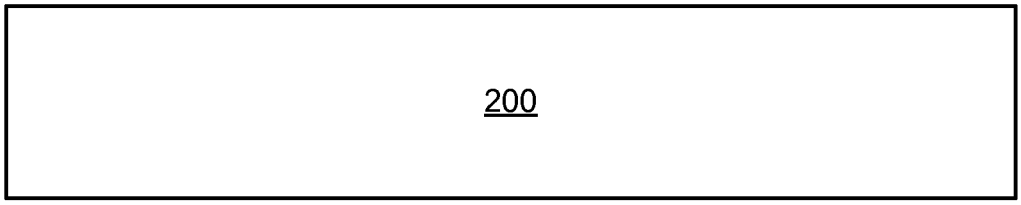


FIG. 2a

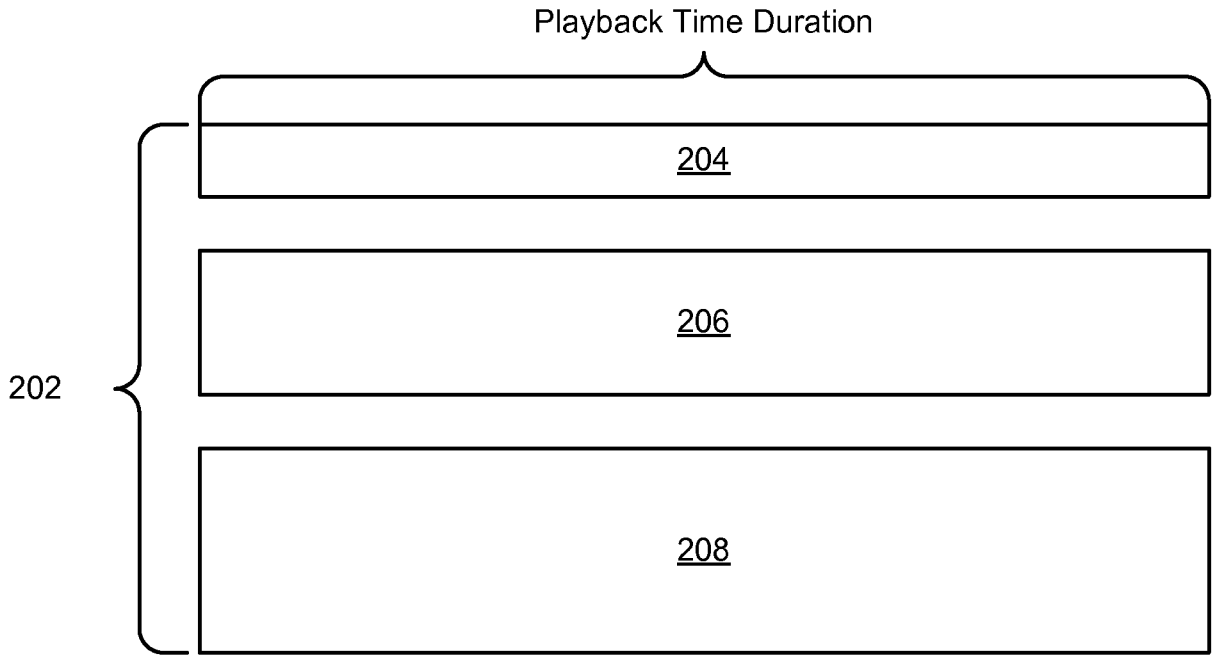


FIG. 2b

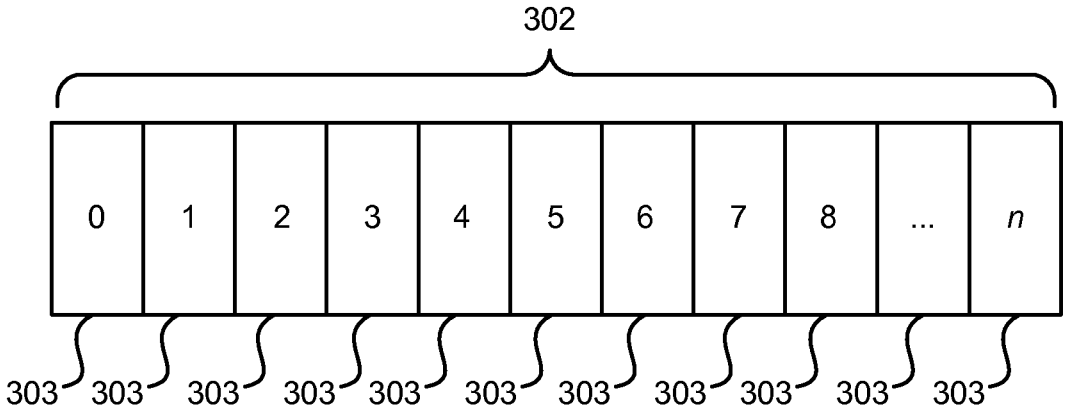

302
0
1
2
3
4
5
6
7
8
...
n
303
303
303
303
303
303
303
303
303
303
303

FIG. 3a

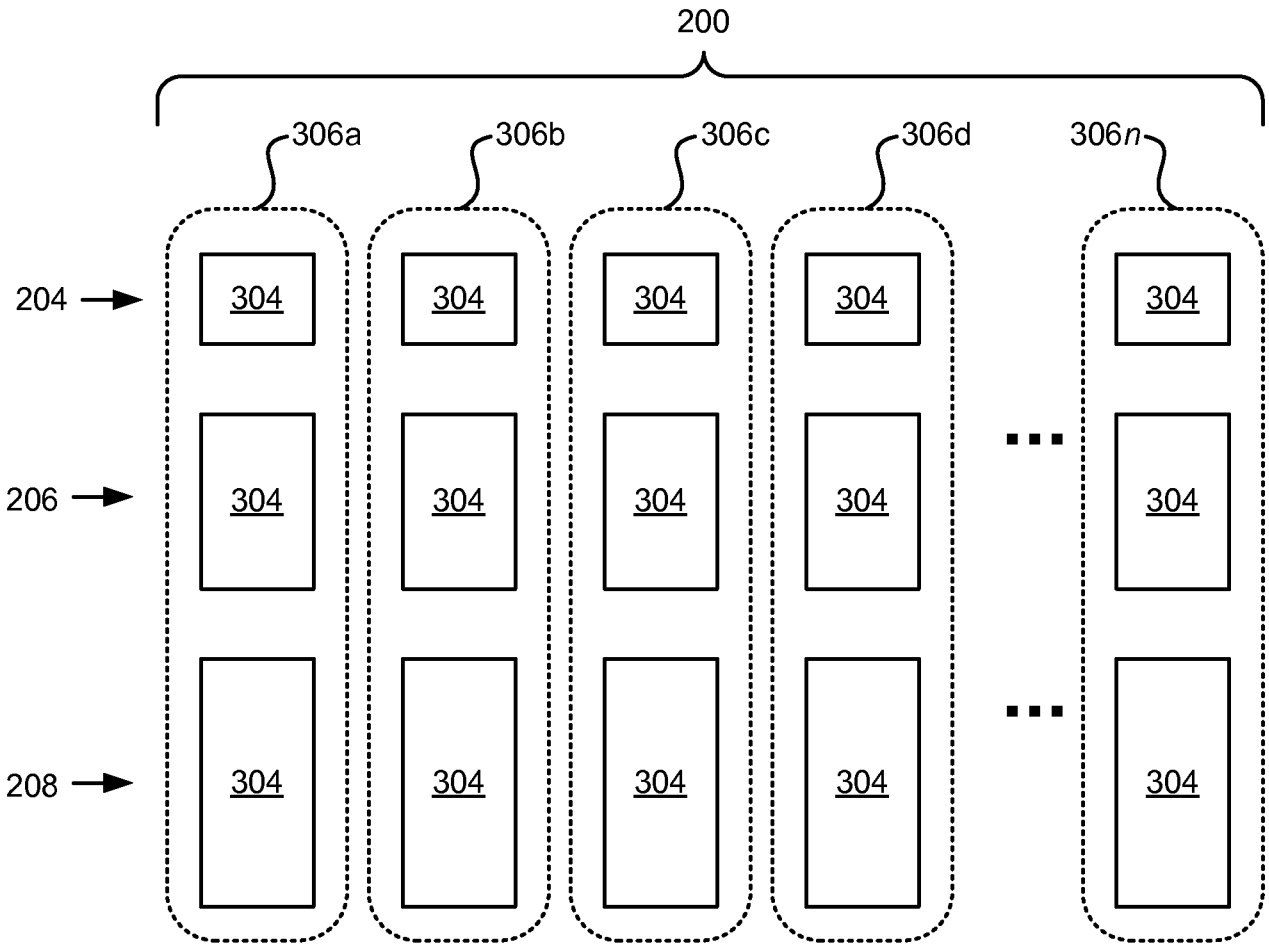

200
306a
306b
306c
306d
306n
204
206
208
304

FIG. 3b

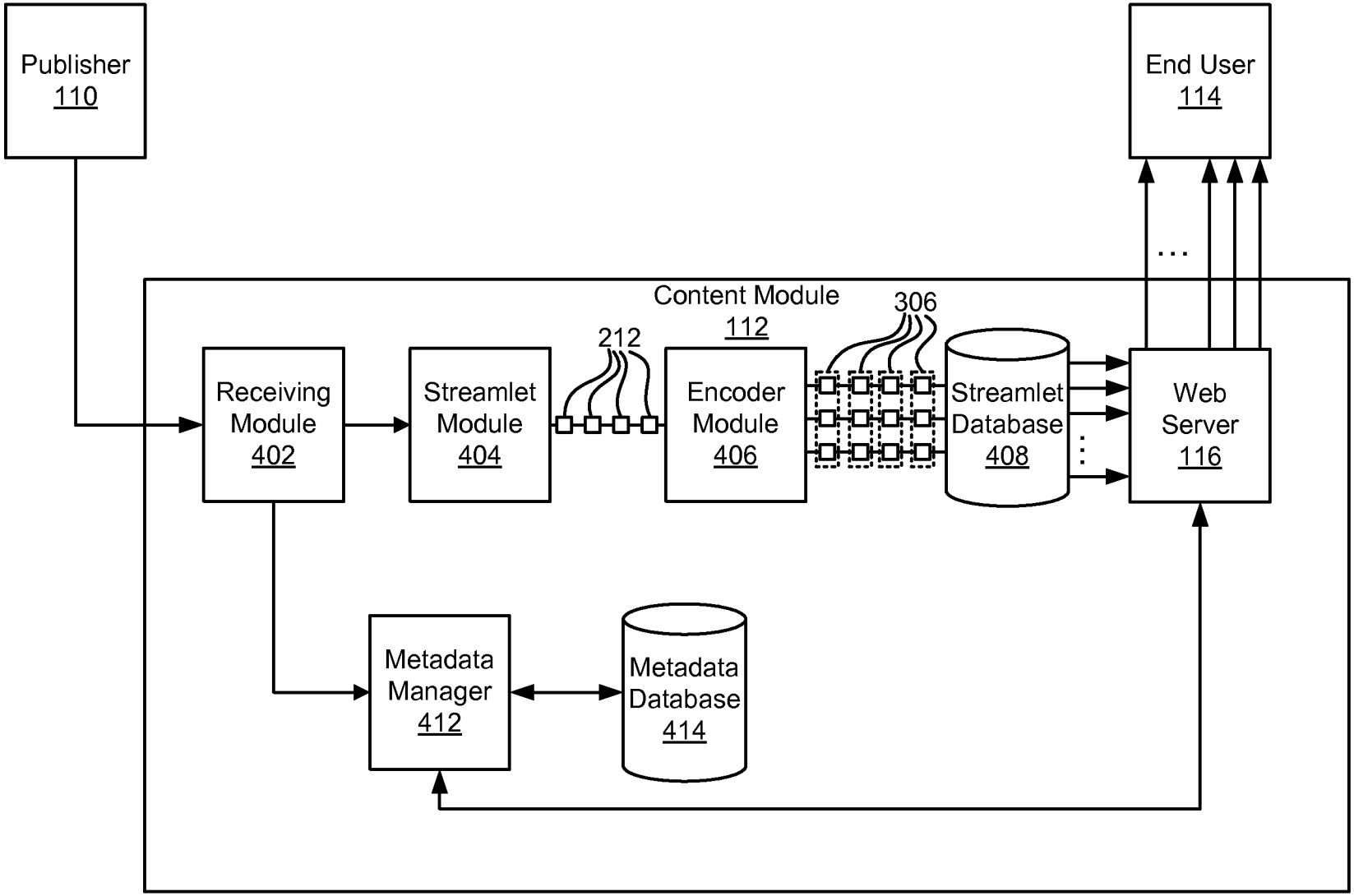

Publisher
110
End User
114
Content Module
112
306
212
Receiving
Module
402
Streamlet
Module
404
Encoder
Module
406
Streamlet
Database
408
Web
Server
116
Metadata
Manager
412
Metadata
Database
414

FIG. 4

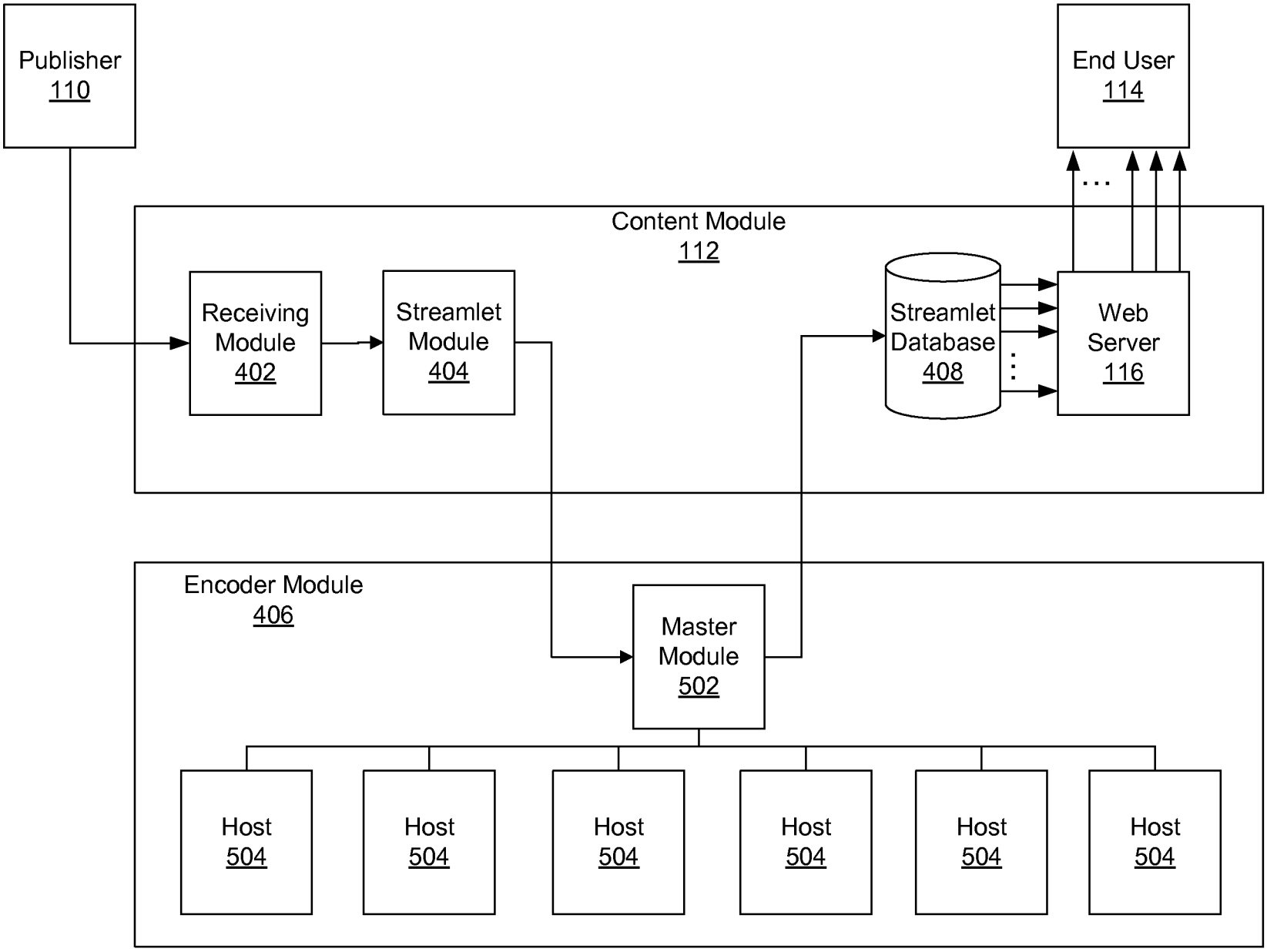

Publisher
110
End User
114
Content Module
112
Receiving
Module
402
Streamlet
Module
404
Streamlet
Database
408
Web
Server
116
Encoder Module
406
Master
Module
502
Host
504
Host
504
Host
504
Host
504
Host
504
Host
504

FIG. 5a

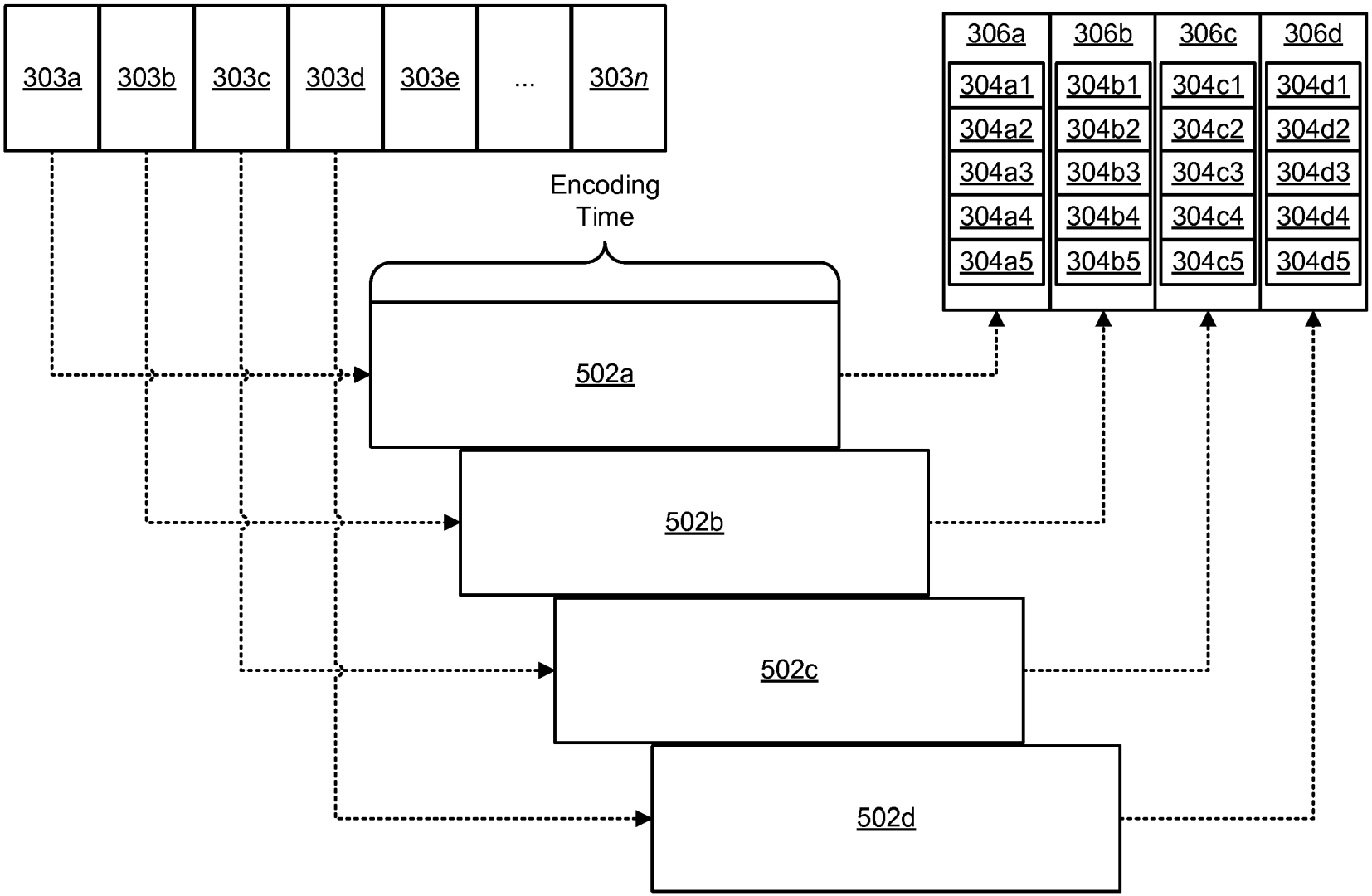


FIG. 5b

fuboTV Media Inc.
Exhibit 1002
Page 068

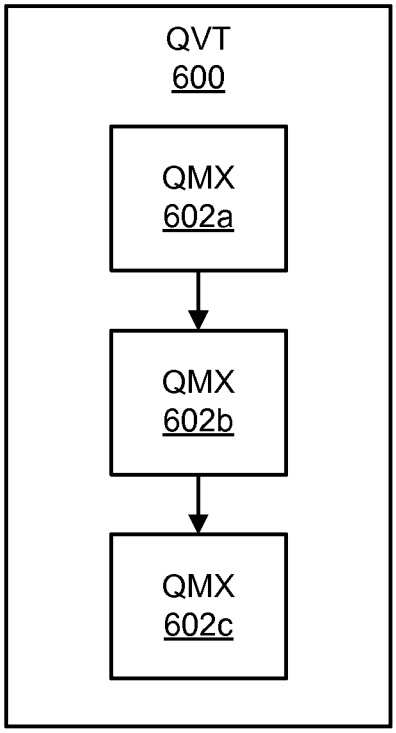


FIG. 6a

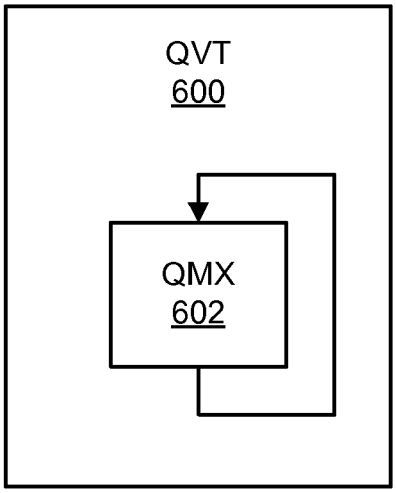


FIG. 6b

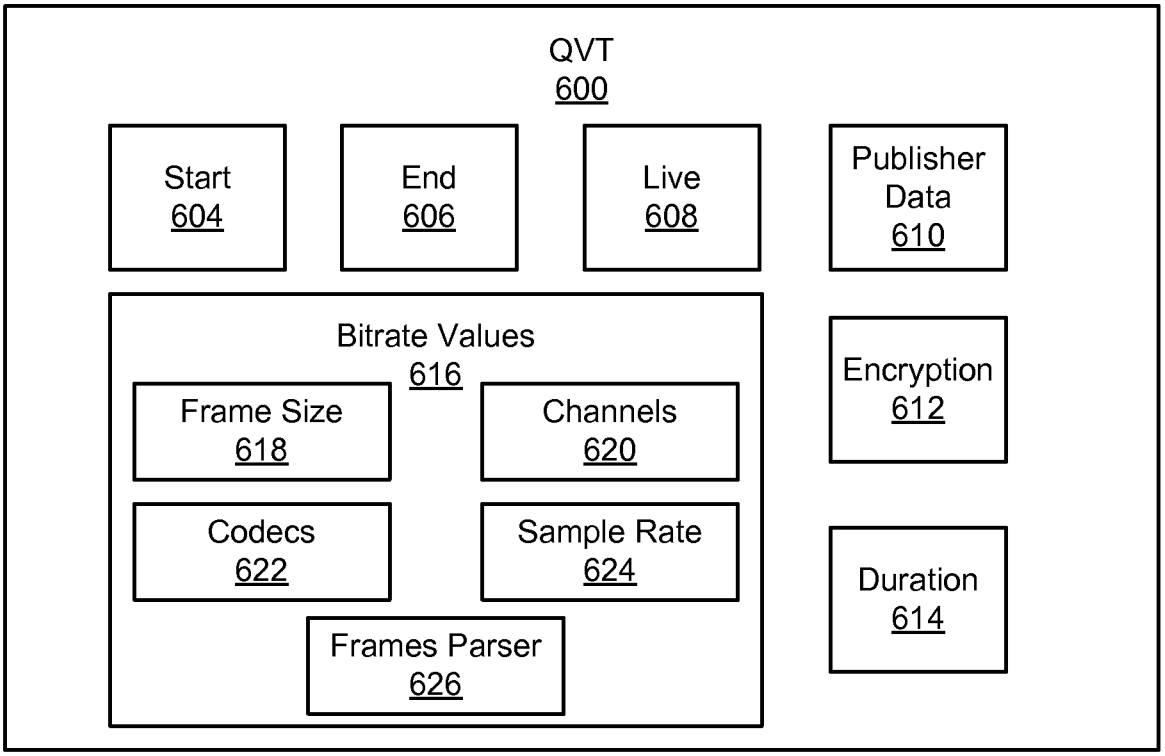


FIG. 6c

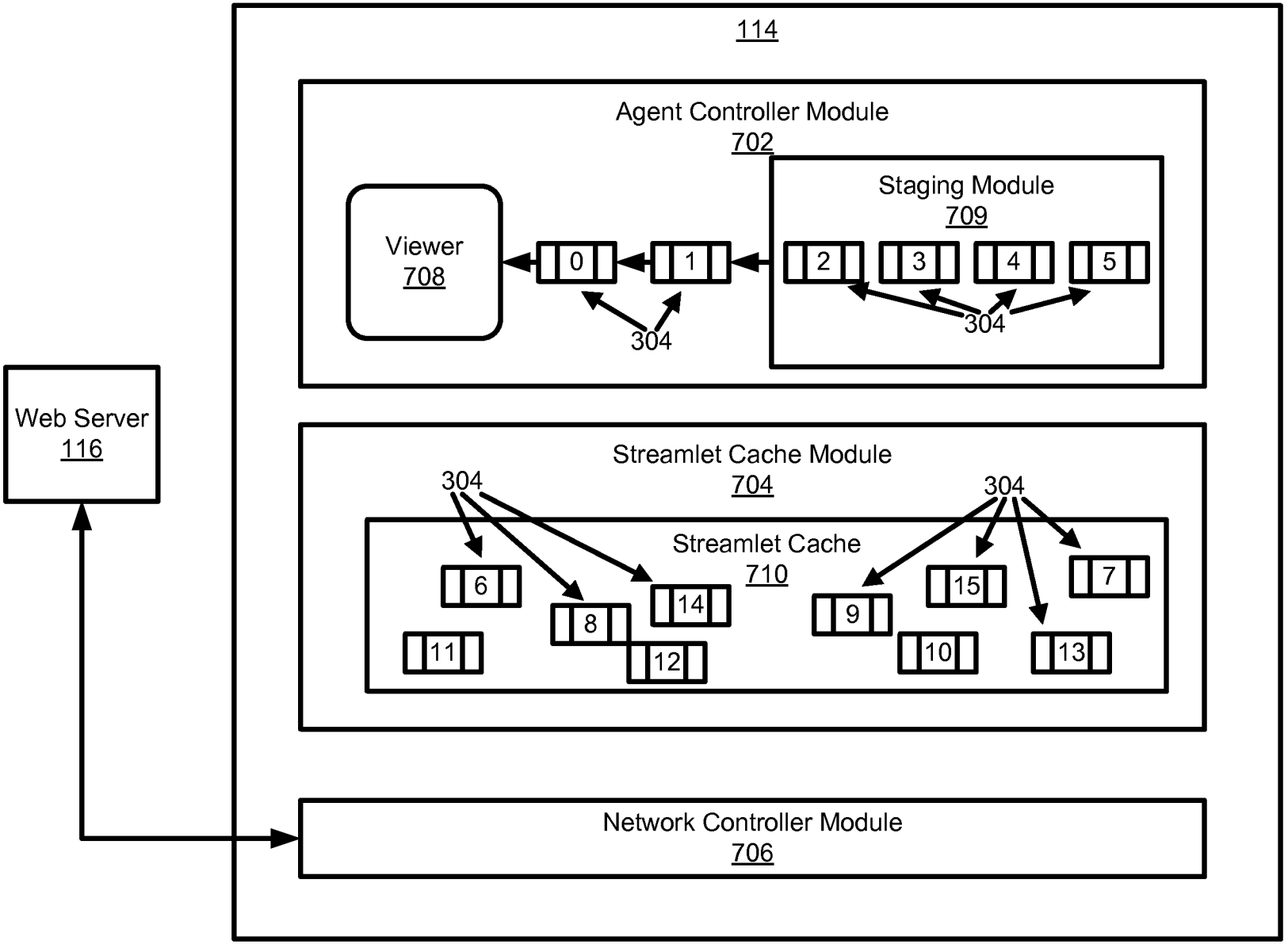

114
Agent Controller Module
702
Staging Module
709
Viewer
708
0
1
2
3
4
5
304
304
Web Server
116
Streamlet Cache Module
704
304
304
Streamlet Cache
710
6
14
8
12
11
9
15
7
10
13
Network Controller Module
706

FIG. 7

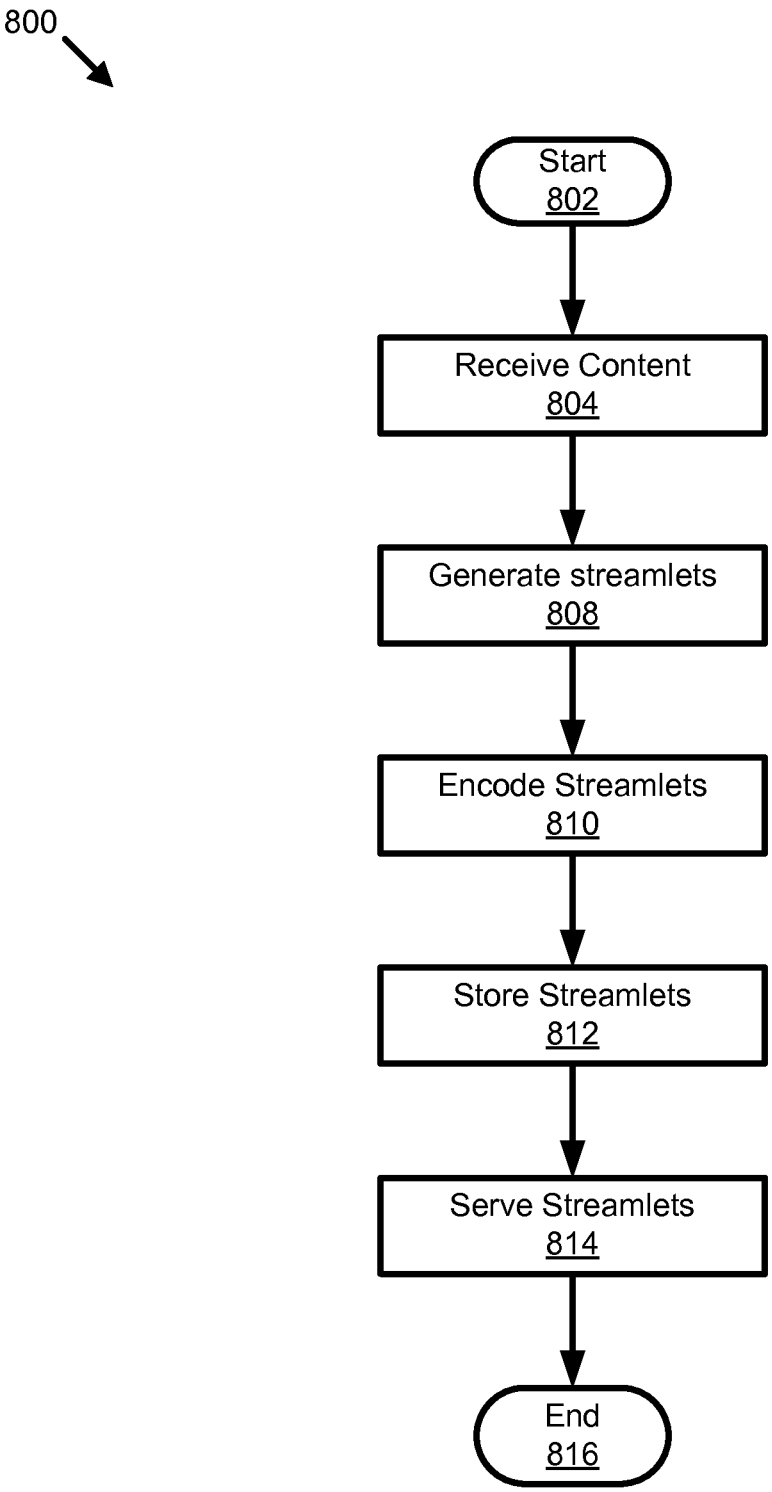

800
Start
802
Receive Content
804
Generate streamlets
808
Encode Streamlets
810
Store Streamlets
812
Serve Streamlets
814
End
816

FIG. 8

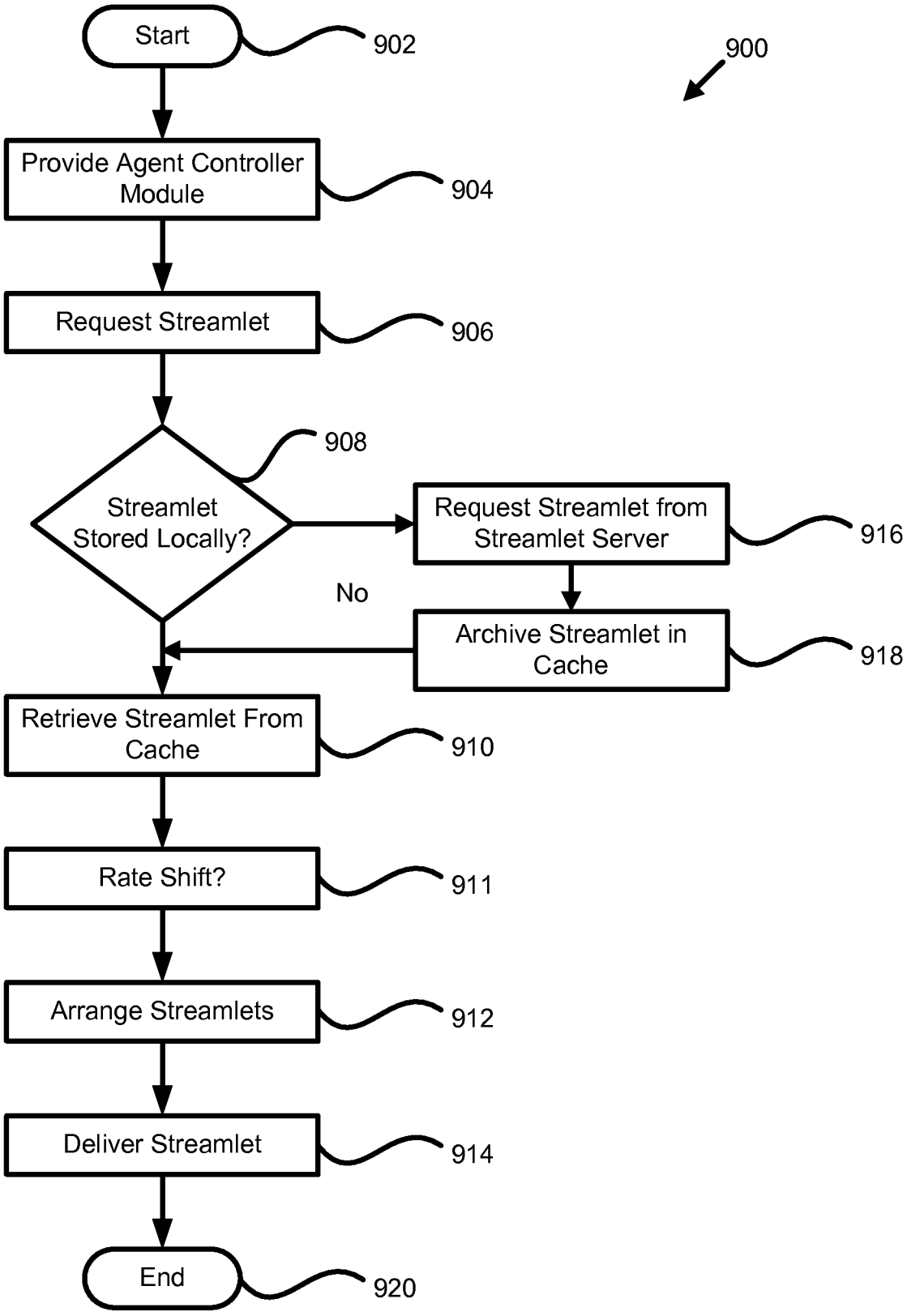

Start
902
900
Provide Agent Controller Module
904
Request Streamlet
906
908
Streamlet Stored Locally?
Request Streamlet from Streamlet Server
916
No
Archive Streamlet in Cache
918
Retrieve Streamlet From Cache
910
Rate Shift?
911
Arrange Streamlets
912
Deliver Streamlet
914
End
920

FIG. 9

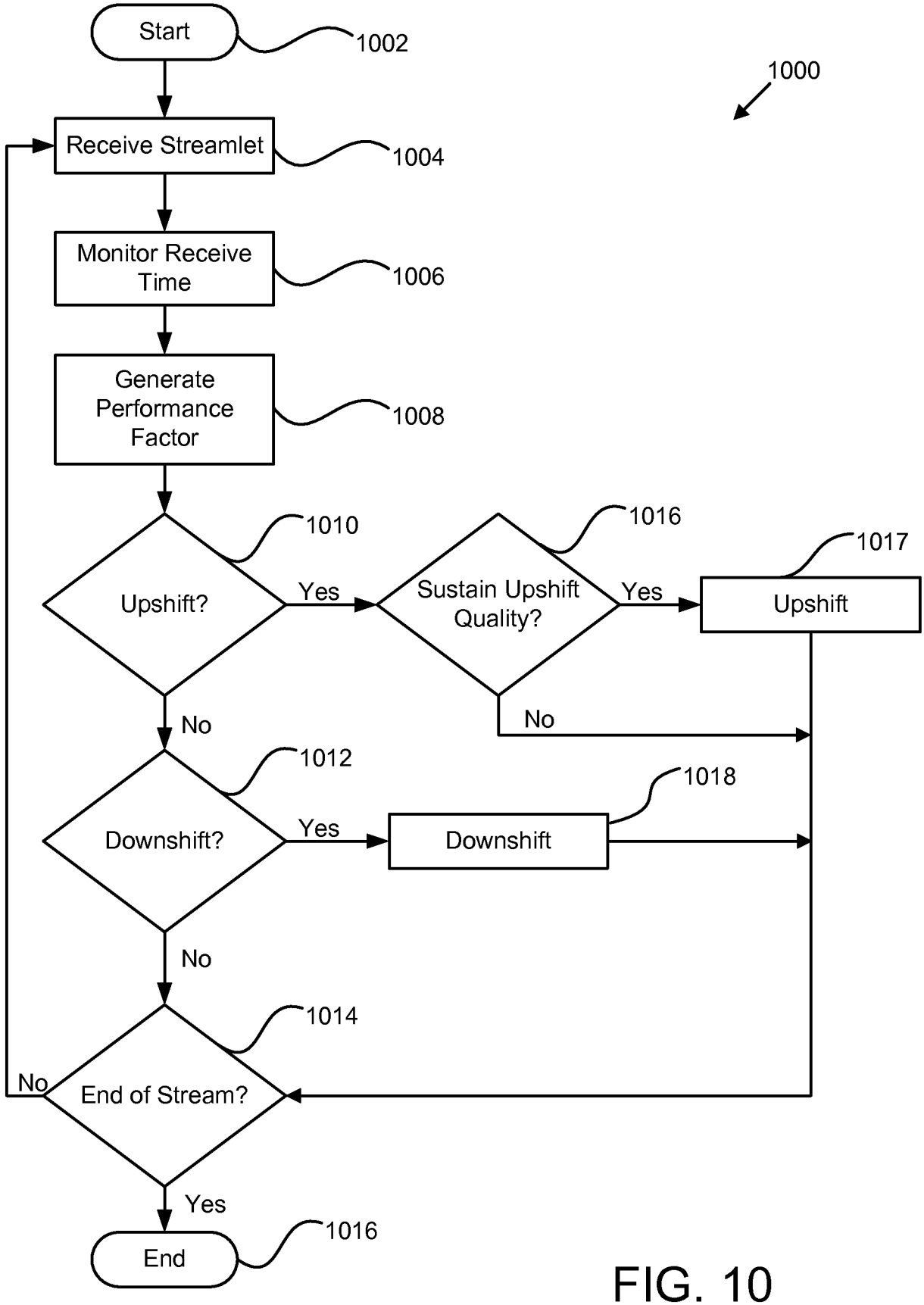

Start
1002
1000
Receive Streamlet
1004
Monitor Receive Time
1006
Generate Performance Factor
1008
1010
Upshift?
Yes
1016
Sustain Upshift Quality?
Yes
1017
Upshift
No
No
1012
Downshift?
Yes
1018
Downshift
No
1014
No
End of Stream?
Yes
1016
End

FIG. 10

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 16/252,356

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
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			OR	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) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	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
16/252,356	01/18/2019	2447	3180	066.0141X1C8	30	4

CONFIRMATION NO. 8631

FILING RECEIPT

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

OC000000105731289

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 16/252,188 *
which 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 CIP 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
(*)Data provided by applicant is not consistent with PTO records.

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.

fuboTV Media Inc.
Exhibit 1002
Page 075

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: 02/07/2019

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 16/252,356**

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

fuboTV Media Inc.
Exhibit 1002
Page 076

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 AssetsControl, 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.
16/252,356	01/18/2019	DAVID F. BRUECK	066.0141X1C8	8631

70560 7590 02/15/2019

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

EXAMINER	

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

fuboTV Media Inc.
Exhibit 1002
Page 079

Decision Granting Request for Prioritized Examination (Track I)	**Application No.** 16/252,356	**Applicant(s)** BRUECK et al.	
	Examiner BRIAN W BROWN	**Art Unit** OPET	**AIA (First Inventor to File) Status** No

1. THE REQUEST FILED 18 January 2019 IS **GRANTED** .

The above-identified application has met the requirements for prioritized examination

A. ☑ for an original nonprovisional application (Track I).
B. ☐ for an application undergoing continued examination (RCE).

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:

A. filing a **petition for extension of time** to extend the time period for filing a reply;

B. filing an **amendment to amend the application to contain more than four independent claims, more than thirty total claims** , or a multiple dependent claim;

C. filing a **request for continued examination** ;

D. filing a notice of appeal;

E. filing a request for suspension of action;

F. mailing of a notice of allowance;

G. mailing of a final Office action;

H. completion of examination as defined in 37 CFR 41.102; or

I. abandonment of the application.

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.

/BRIAN W BROWN/ Petitions Examiner, OPET	

Notice of References Cited	Application/Control No. 16/252,356	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.

fuboTV Media Inc.
Exhibit 1002
Page 081

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	16/252,356	BRUECK et al.
	Examiner	Art Unit
	CHIRAG R PATEL	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	03/19/2019								
	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	✓								

fuboTV Media Inc.
Exhibit 1002
Page 082

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.
16/252,356	01/18/2019	DAVID F. BRUECK	066.0141X1C8	8631

70560 7590 03/22/2019

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

EXAMINER
PATEL, CHIRAG R

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

Office Action Summary	Application No. 16/252,356	Applicant(s) BRUECK et al.	
	Examiner CHIRAG R PATEL	Art Unit 2454	AIA (FITF) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☑ Responsive to communication(s) filed on January 18, 2019.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL.** 2b) ☑ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☑ Claim(s) 1-30 is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6) ☐ Claim(s) _____ is/are allowed.

7) ☑ Claim(s) 1-30 is/are rejected.

8) ☐ Claim(s) _____ is/are objected to.

9) ☐ Claim(s) _____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov.**

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☑ The drawing(s) filed on 18 January 2019 is/are: a)☑ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☑ Notice of References Cited (PTO-892)

2) ☐ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date _____.

4) ☐ Other: _____.

fuboTV Media Inc.
Exhibit 1002
Page 084

Notice of Pre-AIA or AIA Status

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

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. 16/252188 (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(l)(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 negatived 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 he 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://pairdirect.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

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	16/252,356	BRUECK et al.
	Examiner	Art Unit
	CHIRAG R PATEL	2454

CPC - Searched*		
Symbol	Date	Examiner
H04N19/34, H04N19/40, HO4N21/23427	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

fuboTV Media Inc.
Exhibit 1002
Page 106

Bibliographic Data

Application No: 16/252,356

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	066.0141X1C8
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 16252188 01/18/2019

16252188 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 CIP 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**

02/07/2019

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

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
S47	1	("7934159").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2019/03/18 12:27

EAST Search History (Interference)

< This search history is empty>

3/19/2019 12:32:04 PM
C:\Users\cpatel\Documents\EAST\Workspaces\16252356.wsp

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
16/252,356	01/18/2019	DAVID F. BRUECK	066.0141X1C8

CONFIRMATION NO. 8631

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

PUBLICATION NOTICE

OC000000108421361

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

Publication No.US-2019-0158561-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 Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

fuboTV Media Inc.
Exhibit 1002
Page 114

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 16/252,356	Filing Date 01/18/2019	☐ To be Mailed

ENTITY: ☑ LARGE ☐ SMALL ☐ MICRO

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) 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), (i), 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))	* 26	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.

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.

By this response, we have amended several of the original claims (including independent claim 1), cancelled original claims 10-30 without prejudice or disclaimer, and added new claims 31-48. The new claims pertain to similar subject matter as the original claims, but are different from the claims of co-pending application Serial No. 16/262,356, as requested by the Examiner. The cancellation is made without prejudice or disclaimer, and is not made to distinguish prior art. These amendments therefore do not surrender any legal equivalents that might otherwise be available. Twenty-eight claims, including four independent claims, are pending so no fees are believed to be due at this time.

We respectfully request reconsideration of the application after entry of the foregoing amendments. **If this application is not in condition for allowance after entry of the foregoing amendments, the undersigned requests a telephone call to 480.385.5069.**

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. Each of our current independent claims recites this feature, so the prior art rejections are believed to be overcome.

We also appreciate the Examiner's clarification of the double patenting rejections. We have amended our claims by this Response, and the requested terminal disclaimers will be submitted in due course.

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.

fuboTV Media Inc.
Exhibit 1002
Page 116

Double Patenting/Terminal Disclaimers

Claims 1-30 are provisionally rejected as a double patent of co-pending application Serial No. 16/262,356[1], which was filed on the same day as this application. 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. We have amended the claims so that the claims of the two applications have somewhat different language, as requested by the Examiner in the telephonic interview.

The Office Action indicates that the rejections can be overcome with terminal disclaimers. Without consenting to the rejections, appropriate disclaimers are being submitted along with this response 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 the video program at different bit rates. Claim 1, for example, recites that the high, medium and low quality streams each

[1] The Examiner explained that the Office Action's reference to Serial No. 16/262,356 is in error.

fuboTV Media Inc.
Exhibit 1002
Page 117

comprise a group of streamlets encoded at a respective one of the plurality of different bitrates. Our other independent claims recite similar language.

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 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 high, medium and low quality streams each comprise a group of streamlets encoded at a respective one of the plurality of different bitrates, 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 Lango, 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 but encoded at different bit rates.

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 Lango, 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, Lango and Meehan against many of our claims, similarly fails to teach the features of our claims for which it is cited.

fuboTV Media Inc.
Exhibit 1002
Page 118

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.

Even if Xu, Lango, 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.

fuboTV Media Inc.
Exhibit 1002
Page 119

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

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial. No.:	16/252,356	Conf. No.:	8631
Examiner:	Chirag R. Patel	Art Unit:	2454
Inventor:	Brueck et al.	Filed:	January 18, 2019
Docket No.:	066.0141x1C8 (P2007-02-01-2-6)	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.

fuboTV Media Inc.
Exhibit 1002
Page 121

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 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~~;

wherein the amount of data in each streamlet corresponding to the same part of the live event video in the low quality stream, the medium quality stream, and the high quality stream varies according to the different bitrates; and

the plurality of streamlets in the low quality stream, the medium quality stream, and the high quality stream having a duration that is the same as each other.

2 (Currently Amended). 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~~ wherein the low quality stream is encoded at a bit rate of no

fuboTV Media Inc.
Exhibit 1002
Page 122

greater than 100 kbps, and the medium quality stream is encoded at a bit rate between 100 kbps and 600 kbps.

3 (Currently Amended). The system of ~~claim 2, wherein the first and second durations are different~~ claim 1 wherein the streamlets in each of the high quality stream, the medium quality stream and the low quality stream are each encoded at a different one of the plurality of different bitrates.

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

fuboTV Media Inc.
Exhibit 1002
Page 123

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.

Claims 10-30 (Cancelled).

31 (New). A content player device to stream a 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 client module and the server, wherein the server is configured to access at least one of a plurality of groups of streamlets;

wherein the video is encoded at a plurality of different bitrates to create a

fuboTV Media Inc.
Exhibit 1002
Page 124

plurality of streams including at least a low quality stream, a medium quality stream, and a high quality stream, wherein each of the low quality stream, the medium quality stream, and the high quality stream comprises a streamlet that encodes the same portion of the video at a different one of the plurality of different bitrates;

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

wherein the streamlet encoding the same portion of the video in the low quality stream has an equal playback duration as the streamlet encoding the same portion of the video in the high quality stream;

select a specific one of the streams based upon a determination by the client module 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 selected stream;

receive the requested streamlets from the server via the one or more network connections; and

provide the received streamlets for playback of the video.

32 (New). The content player device of claim 31 wherein each streamlet of the plurality of streamlets in the low quality stream, the medium quality stream, and the high quality stream has a duration that is the same as each other;

33 (New). The content player device of Claim 31, wherein the low quality stream is encoded at a bit rate of no greater than 100 kbps and the medium quality stream is encoded at a bit rate between 100 kbps and 600 kbps.

34 (New). The content player device of claim 31, wherein each streamlet of the plurality of streamlets in the low quality stream, the medium quality stream, and the high quality stream has a duration that is the range of 0.1 to 5 seconds.

fuboTV Media Inc.
Exhibit 1002
Page 125

35 (New). The content player device of claim 31, wherein the video is a video of a live event.

36 (New). The content player device of claim 35, wherein the streamlets of the low quality stream, the medium quality stream, and the high quality stream are available before the live event is complete.

37 (New). The content player device of claim 36, 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.

38 (New). The content player device of claim 37, 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.

39 (New). 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 configured to store live event video, the live event video encoded at a plurality of different bit rates creating a plurality of streams including at least a low quality stream and a high quality stream;

the low quality stream and the high quality stream each encoding the same portion of the live event video with a streamlet that is encoded a different one of the different bit rates;

wherein the plurality of streamlets in the low quality stream and the plurality of streamlets in the high quality stream have durations that are equal to each other.

40 (New). The system of claim 39, wherein the streamlets in each of the low quality stream and the high quality stream corresponding to the same portion of the live event video have equal durations.

41 (New). The system of claim 39 wherein the plurality of streams further comprise a medium quality stream encoded at a bit rate higher than the low quality stream and lower than the

fuboTV Media Inc.
Exhibit 1002
Page 126

high quality stream.

42 (New). The system of claim 41 wherein the low quality stream is encoded at a bit rate of no less than 100 kbps, the high quality stream is encoded at a bit rate of no less than 600 kbps, and the medium quality stream is encoded at a bit rate between 100 kbps and 600 kbps.

43 (New). The system of claim 39, 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 a 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 one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the client module to select a higher or lower bitrate version of the streams;

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

send the retrieved 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.

44 (New). The system of claim 39, 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 a 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 one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the client module to select a higher or lower bitrate version of the streams;

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

fuboTV Media Inc.
Exhibit 1002
Page 127

send the retrieved 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.

45 (New). The system of claim 42, wherein all of the streamlets in each of the low quality stream, the medium quality stream and the high quality stream have equal durations.

46 (New). The system of claim 42, wherein the streamlets of the low quality stream, the medium quality stream, and the high quality stream are available on a ten second delay before the live event is complete, 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.

47 (New). A content player device to stream a 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 client module and the server, wherein the server is configured to access at least one of a plurality of groups of streamlets;

wherein the video is encoded at a plurality of different bitrates to create a plurality of streams including at least a low quality stream and a high quality stream,

the low quality stream and the high quality stream each representing the same portion of the video with a streamlets encoded at a different one of the plurality of different bitrates; and

wherein the streamlet representing the same portion of the video in the low quality stream and the streamlet representing the same portion of the video in the high quality stream have durations equal to each other;

select a specific one of the streams based upon a determination by the client module 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

fuboTV Media Inc.
Exhibit 1002
Page 128

for the selected stream;

receive the requested streamlets from the server via the one or more network connections; and

provide the received streamlets for playback of the video.

48 (New). The content player device of claim 47, wherein the plurality of streamlets in the low quality stream have a duration equal to the duration of the plurality of streamlets in the high quality stream.

fuboTV Media Inc.
Exhibit 1002
Page 129

Electronic Acknowledgement Receipt

EFS ID:	36357908
Application Number:	16252356
International Application Number:	
Confirmation Number:	8631
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:	066.0141X1C8
Receipt Date:	20-JUN-2019
Filing Date:	18-JAN-2019
Time Stamp:	12:30:22
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_P2007-02-01-2-6_066.pdf	178188 1210f9e424a35026b909c5817df2b8bb289dcc97	yes	14

	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):	178188	

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.

New Applications Under 35 U.S.C. 111

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.

National Stage of an International Application under 35 U.S.C. 371

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.

New International Application Filed with the USPTO as a Receiving Office

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.

Electronic Patent Application Fee Transmittal

Application Number:	16252356
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:	066.0141X1C8

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: DISQ.E.FILE
Document Description: Electronic Terminal Disclaimer – Approved

Application No.: 16252356

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

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 OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT**
Application Number	16252356
Filing Date	18-Jan-2019
First Named Inventor	DAVID BRUECK
Attorney Docket Number	066.0141X1C8
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 LLC	100%

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)

9571551

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.

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:
- 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.

◉ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.

○ 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 39928

○ 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.

Electronic Acknowledgement Receipt

EFS ID:	36387725
Application Number:	16252356
International Application Number:	
Confirmation Number:	8631
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:	066.0141X1C8
Receipt Date:	24-JUN-2019
Filing Date:	18-JAN-2019
Time Stamp:	13:42:02
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	062419INTEFSW13415900
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 a46577dc721f5f65b42690a677f405c0be3c2fb2	no	2
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30600 def5b89fc571f9e73b8176de410138e14a0bc93d	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	64153		

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.

New Applications Under 35 U.S.C. 111

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.

National Stage of an International Application under 35 U.S.C. 371

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.

New International Application Filed with the USPTO as a Receiving Office

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.

Applicant-Initiated Interview Summary	Application No. 16/252,356	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: 19 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 amendments 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. 20190621

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.

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.
16/252,356	01/18/2019	DAVID F. BRUECK	066.0141X1C8	8631

70560 7590 06/26/2019

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

EXAMINER	
PATEL, CHIRAG R	

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.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 January 2006 (26.01.2006)

PCT

(10) International Publication Number
WO 2006/010113 A2

(51) International Patent Classification:
G06F 15/173 (2006.01)

(21) International Application Number:
PCT/US2005/024517

(22) International Filing Date: 11 July 2005 (11.07.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/586,952 9 July 2004 (09.07.2004) US

(71) Applicant *(for all designated States except US)*: **NETWORK FOUNDATION TECHNOLOGIES, LLC** [US/US]; 2215 Cooktown Road, Ruston, LA 71270 (US).

(72) Inventors: **O'NEAL, Mike**; 2215 Cooktown Road, Ruston, LA 71270 (US). **TALTON, John P.**; 2215 Cooktown Road, Ruston, LA 71270 (US).

(74) Agents: **LANDA, Adam B.** et al.; Greenberg Traurig, MetLife Building, 200 Park Avenue, New York, NY 10166 (US).

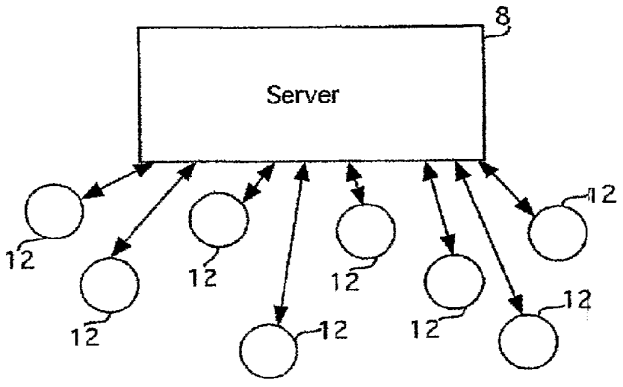
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK,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

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

Published:
— *without international search report and to be republished upon receipt of that report*

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

(54) Title: SYSTEMS FOR DISTRIBUTING DATA OVER A COMPUTER NETWORK AND METHODS FOR ARRANGING NODES FOR DISTRIBUTION OF DATA OVER A COMPUTER NETWORK



PRIOR ART

(57) Abstract: Various embodiments of the present invention relate to a system for distributing data (e.g., content data) over a computer network and a method of arranging receiver nodes in a computer network such that the capacity of a server is effectively increased (e.g., the capacity of a server may be effectively multiplied many times over; the capacity of the server may be effectively increased exponentially). In one embodiment the present invention may take advantage of the excess capacity many receiver nodes possess, and may use such receiver nodes as repeaters. The distribution system may include node(s) having database(s) which indicate ancestor(s) and/or descendant(s) of the node so that reconfiguration of the distribution network may be accomplished without burdening the system's primary server. An embodiment of the present invention may include a process for configuring a computer information distribution network having a primary server node and user nodes docked in a cascaded relationship, and reconfiguring the network in the event that a user node departs from the network. In one example (which example is intended to be illustrative and not restrictive), the process may include the steps of providing a new user node (or connection requesting user node) with a connection address list of nodes within the network, having the new user node (or connection requesting user node) go to (or attempt to go to) the node at the top of the connection address list, determine whether that node is still part of the distribution network, and connect thereto if it is, and if it is not, to go to (or attempt to go to) the next node on the connection address list. In another example (which example is intended to be illustrative and not restrictive), when a user node departs from the distribution network, a propagation signal may be transmitted to the nodes below it in the network, causing them to move up in the network in a predetermined order. In another example (which example is intended to be illustrative and not restrictive), the present invention may provide a decentralized approach which provides, to each new user node (or connection requesting user node) a path back to the root server.

WO 2006/010113 A2

SYSTEMS FOR DISTRIBUTING DATA OVER A COMPUTER NETWORK AND METHODS FOR ARRANGING NODES FOR DISTRIBUTION OF DATA OVER A COMPUTER NETWORK

FIELD OF THE INVENTION

This patent application is related to and claims the benefit of the filing date of U.S. Provisional Patent Application Serial No. 60/586,952 for "Systems for Distributing Data Over A Computer Network And Methods For Arranging Nodes For Distribution Of Data Over A Computer Network", filed July 9, 2004.

Various embodiments of the present invention relate to systems for distributing data (e.g., content data) over a computer network and methods of arranging nodes for distribution of data (e.g., content data) over a computer network. In one example (which example is intended to be illustrative and not restrictive), the systems and methods of the present invention may be applied to the distribution of streaming audiovisual data over the Internet.

For the purposes of the present application the term "node" (e.g., as used in the phrase a first node communicates with a second node) is intended to refer to a computer system (or sometimes just "computer") configured to send and/or receive data over a computer network (e.g., a computer network including the Internet, a local-area-network, a wide-area-network, a wireless network). Since a "node" in a computer network would not exist but for a computer system being available on such node, the terms "node" and "computer system" (or sometimes just "computer") may be used interchangeably (i.e., the term "node" should be understood to include an available computer system (or sometime just "computer").

Further, for the purposes of the present application the term "upstream" (e.g., as used in the phrase a first computer system sends data via an upstream connection to a second computer system) is intended to refer to the communication path between a first computer system and the Internet when the first computer system is sending data to a second computer system via the Internet.

Further still, for the purposes of the present application the term "downstream" (e.g., as used in the phrase a first computer system receives data via a downstream connection from a second computer system) is intended to refer to the communication path between a first computer system and the Internet when the first computer system is receiving data from a second computer system via the Internet.

Further still, for the purposes of the present application the term "uptree" (e.g., as

used in the phrase a first node sends data via an uptree connection to a second node) is intended to refer to the network topology communication path between a first node and a second node when a first node is sending data to a second node which is higher up on the network topology tree (that is, closer to the root server).

Further still, for the purposes of the present application the term "downtree" (e.g., as used in the phrase a first node sends data via a downtree connection to a second node) is intended to refer to the network topology communication path between a first node and a second node when a first node is sending data to a second node which is lower down on the network topology tree (that is, closer to the leaves).

Further still, for the purposes of the present application the term "docked" (e.g., as used in the phrase a child node docks with a parent node) is intended to refer to forming a connection between nodes via which data may flow in at least one direction (e.g., at least uptree or downtree).

Further still, for the purposes of the present application the term "apparent available capacity" (e.g., as used in the phrase apparent available capacity to transmit content data) is intended to refer to a nominal capacity (e.g., a node's presumed capacity based upon a design specification but not necessarily actually tested in the network).

BACKGROUND OF THE INVENTION

In a computer network such as the Internet, each node in the network has an address. A computer system resident at a particular address may have sufficient bandwidth or capacity to receive data from, and to transmit data to, many other computer systems at other addresses. An example of such a computer system is a server, many commercial versions of which can simultaneously exchange data with thousands of other computer systems.

A computer system at another location may have only sufficient bandwidth to effectively exchange data with only one other computer system. An example of such a system is an end user's personal computer connected to the Internet by a very low speed dialup modem. However, even typical personal computers connected to the Internet by higher speed dialup modems may have sufficient bandwidth that they may exchange data essentially simultaneously in an effective manner with several other computer systems. Moreover, an end user's personal computer system may have even greater bandwidth when connected to the Internet by ISDN lines, DSL (e.g., ADSL) lines, cable modems, T1 lines or even higher capacity links. As discussed more fully below, various embodiments of the present invention

fuboTV Media Inc.
Exhibit 1002
Page 145

may take advantage of the availability of such higher capacity end user systems (e.g., those computer systems capable of essentially simultaneously exchanging data with multiple computer systems).

In a typical situation, as shown in FIG. 1, a content provider distributes its data by making the data available on a server node 8 simultaneously to a plurality of users at user nodes 12 (of note, the terms "server", "root server" and "primary server" may be used interchangeably throughout the present application to refer to the same device (i.e., the highest level parent node in a given network). The double-headed arrows show the two-way communication between each end user's computer system and the server. Essentially the content provider's server transmits a separate stream of signals to each receiver node. To accommodate additional users, the content provider would typically either add equipment to increase capacity or it would engage a mirror site to accomplish essentially the same result as adding equipment. The capacities of the end user computers is of virtually no consequence in such a system.

Another system for distributing data is the Napster™ music file exchange system provided by Napster, Inc. of Redwood City, Calif. A schematic of the Napster™ music file exchange system (as its operation is presently understood) is illustrated in FIG. 2.

More particularly, it is believed that in the Napster™ system a copy of the music data is not kept on the server. The server 9 instead maintains a database relating to the various music files on the computers of users who are logged onto the server 9. When a first user 12a sees that a desired music file is available from a second logged on user 12b, the first user causes his computer to query the server 9 for the second user's node address and a connection is made between the first and second user's computers through which the first user's computer notifies the second user's computer of the desired file and the second user's computer responds by transmitting a copy of the desired music file directly to the first user's computer. It is further believed that a first user attempting to download a particular file from a second user must start completely over again if the second user cancels its transmission or goes off line during the data transfer.

In another area, engineers have developed what is known as "streaming media." In summary, streaming media is a series of packets (e.g., of compressed data), each packet representing moving images and/or audio.

To help understand streaming media in more detail, it is helpful to review the traditional Internet distribution method. Each node (whether it is a server node or a user node) in a computer network has a unique identification (sometimes referred to as an "IP" address)

fuboTV Media Inc.
Exhibit 1002
Page 146

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.

fuboTV Media Inc.
Exhibit 1002
Page 147

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

fuboTV Media Inc.
Exhibit 1002
Page 148

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;

fuboTV Media Inc.
Exhibit 1002
Page 149

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;

fuboTV Media Inc.
Exhibit 1002
Page 150

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.

fuboTV Media Inc.
Exhibit 1002
Page 151

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.

fuboTV Media Inc.
Exhibit 1002
Page 152

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.).

fuboTV Media Inc.
Exhibit 1002
Page 153

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

fuboTV Media Inc.
Exhibit 1002
Page 154

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

fuboTV Media Inc.
Exhibit 1002
Page 155

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

fuboTV Media Inc.
Exhibit 1002
Page 156

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

fuboTV Media Inc.
Exhibit 1002
Page 157

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

fuboTV Media Inc.
Exhibit 1002
Page 158

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

fuboTV Media Inc.
Exhibit 1002
Page 159

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.

fuboTV Media Inc.
Exhibit 1002
Page 160

Referring now to a "Distribution Network Construction" example (which example is intended to be illustrative and not restrictive), it is noted that under this example the length of the distribution chains (i.e., the number of levels of user nodes linked through each other to the root server) is aimed to be kept as small as possible.

Assuming that all new user nodes have the same re-transmission capability (or disregarding the different re-transmission capabilities which different new user nodes may have), the user nodes would be distributed in this example through the first level until all the direct connection slots to the server were filled. Then as new user nodes sought connection to the distribution network, they would be assigned to connections with the first level nodes 12 until all the slots available on the first level nodes were filled. This procedure would be repeated with respect to each level as more and more new user nodes attempted to connect to the distribution network. In other words, the server, acting as an instructing node, would perform a Server's Connection Instruction Routine in which one step is determining whether there is room on the server for a new user node and, if so, the server would instruct the new user node to connect directly to the server. If there were no room on the server, then the server would perform the step of consulting its network topology database and devising a connection address list having the fewest number of connection links to a prospective parent user node. After performing the Server's Connection Instruction Routine, the server would either allow the new user node to dock directly with the server or send the new user node on its way to the first prospective parent user node on the connection address list.

In this example (which example is intended to be illustrative and not restrictive), a partially occupied potential parent node in a particular level (i.e., a prospective parent node already having a child but still having an available slot for an additional child node) may be preferred over unoccupied (i.e., childless) potential parent nodes on the same level. This may help to keep the number of interior nodes (i.e., the number of nodes re-transmitting data) to a minimum.

Alternatively, in another example (which example is intended to be illustrative and not restrictive), nodes that have zero children may be preferred over nodes that already have one child. While it is true that this may increase the number of repeat nodes, what the inventors have found is that by filling the frontier in an "interlaced" fashion, connecting nodes build up their buffers more quickly (allowing a reduction in the incidence of stream interruptions).

Referring now to FIGS. 12A-12C, these FIGS. illustrate, in this example, what happens when a partially occupied parent node is preferred over a parent childless node as a

fuboTV Media Inc.
Exhibit 1002
Page 161

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

fuboTV Media Inc.
Exhibit 1002
Page 162

(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

fuboTV Media Inc.
Exhibit 1002
Page 163

may be assigned more than two child nodes directly connected thereto, bandwidth ratings greater than 2.0 may be assigned under this example to Internet connections having greater bandwidth than T1 connections.

- In another example (which example is intended to be illustrative and not restrictive), other bandwidth networks (e.g., 50 Kbit/sec and 100 Kbit/sec) may be supported.
- In another example, the bandwidth needed for a node to be "repeat capable" may depend on the particular network to which it is attached (upon initial connection the connecting node may first test its upstream bandwidth and then report that bandwidth to the network server; the server may determine whether the node is "repeat capable" or not.
- In another example (which example is intended to be illustrative and not restrictive), a node may be deemed "repeat capable" if the node is not firewalled (e.g., port 1138 is open) and the node has sufficient bandwidth to retransmit two copies of the incoming stream.
- A third factor is performance. A user node's performance rating may, in one example, be zero (0.0) if it is disconnected as a result of a Grandparent's Complaint Response Routine (discussed below in connection with FIG. 31). Otherwise, the user node's performance rating, in this example, may one (1.0).

Further, in this example (which example is intended to be illustrative and not restrictive), a user node's utility rating may determined by multiplying connection time by performance rating by bandwidth rating. That is, in this example:

- Utility Rating = Time x Performance Rating x Bandwidth Rating.

Referring once again to FIG. 13, it is seen that in this example information regarding a user node's time connected to the network, bandwidth rating, performance rating, utility rating and potential re-transmission rating (discussed below) may be stored in the user node's elapsed time buffer 137, bandwidth rating buffer 138, performance rating buffer 139, utility rating buffer 121 and potential re-transmission rating buffer 122, respectively.

In another example (which example is intended to be illustrative and not restrictive), nodes entering the network may be judged to be either "repeat capable" or "non-repeat capable" (wherein non-repeat capable nodes may be called "bumpable" nodes). Repeat

fuboTV Media Inc.
Exhibit 1002
Page 164

capability may be based on: (1) Upstream bandwidth (e.g., tested at initial connection); and (2) the firewall status, opened or closed, of a node. In one specific example, if a node is either firewalled or has upstream bandwidth less than (approx.) 2.5 times the video streaming rate, that node will be deemed bumpable. All nodes joining the network, whether bumpable or non-bumpable, may be placed as close to the server as possible. However, for the purpose of placement (as described for example in the Universal Connection Address Routine of Figure 20), a repeat capable node can bump a bumpable node when being placed in the network.

Referring now to FIG. 17 a flow diagram associated with this example illustrating the Server's Connection Instruction Routine is shown. This routine need not necessarily rely on the new user node's utility rating. This routine may rely on the potential re-transmission rating of a user node, which is arrived at by multiplying the user node's performance rating by its bandwidth rating. That is, in this example:

- Potential Re-transmission Rating = Performance Rating x Bandwidth Rating.

Thus, in this example, the server would want to put those user nodes with the highest potential re-transmission rating as close to the server in a distribution chain as possible (i.e., as high uptree as possible) because they have the greatest likelihood of being able to re-transmit one or more copies of content data. On the other hand, a potential re-transmission rating of zero in this example indicates that a user node has no ability to (or little expected reliability in) re-transmitting even one copy of content data (the server in this example would want to put a user node with a zero rating as far as reasonably possible from the server in a distribution chain (i.e., as low downtree as possible)). For the purpose of ease of discussion, in the flow diagram example of FIG. 17 illustrating the Server's Connection Instruction Routine, the server is concerned about whether the potential re-transmission rating is zero (i.e., either one or both of the performance rating and bandwidth rating is zero) or greater than zero (i.e., both the performance rating and the bandwidth rating are greater than zero).

In step 171 the server node interrogates the new user node (or connection requesting node) for its bandwidth rating and performance rating. If the new user node is really new to the distribution network, or if it has returned to the server because all of the user node's ancestor nodes have disappeared from the network, the new user node's performance memory will contain, in this example, a performance rating of 1.0 (e.g., the default rating). However, if the new user node has been dispatched to the server for a new connection to the network

fuboTV Media Inc.
Exhibit 1002
Page 165

because the new user node had failed to provide content data to one of its child nodes, then, in one example, its performance memory will contain a performance rating of zero.

In step 172, the server determines, in this example, whether the potential re-transmission rate of the connection requesting node is greater than zero (i.e., whether both the bandwidth rating and the performance rating are greater than zero, or, if only the bandwidth rating is considered, whether the bandwidth rating is greater than zero (i.e., the connection requesting node is a high-bandwidth node)). If the answer is "yes," then the server goes to step 173 in which the server determines whether it has room for the new user node. If the answer to the query in step 173 is "yes," then the server goes to step 174 in which it instructs the new user node to connect directly to the server. Then the server goes to step 102 in the Server's Connection Routine (see FIG. 10).

If the answer to the query in step 173 is "no" (i.e., the server does not have the capacity to serve the new user node directly), then the server goes to step 175. In step 175, the server, acting as an instructing node, consults its network topology database and devises a connection address list having the fewest number of connection links to a prospective parent node. That is, the server checks its information regarding the nodes in the level closest to the server (i.e., the highest uptree) to determine whether there are any potential parent nodes with space available for the new user node. If there are no potential parent nodes with space available, then the database is checked regarding nodes in the level one link further from the server, and so on until a level is found having at least one potential parent node with space available for the new user node. That is, the server determines which parent node with unused capacity for a child node is closest to the server (i.e., in the highest level, with the first level being the highest), and devises the connection address list from such prospective parent node to the server. The server then goes to step 102 (shown in FIG. 10).

In another embodiment of the invention, the server could skip step 172 and go directly to step 173 (or it could go to step 172 and, if the answer to the query in step 172 is "no" (i.e., either one or both of the bandwidth rating and the performance rating are zero, or, if only the bandwidth rating is considered, whether the bandwidth rating is zero (i.e., the connection requesting node is a low-bandwidth node)), the server could go to step 175). However, this could prevent the server from loading up the highest levels of the distribution chains with nodes capable of re-transmitting at least one acceptable copy of the content data (in contrast to if the server does perform step 172 and does goes to step 176 if the answer to the query in step 172 is "no.").

fuboTV Media Inc.
Exhibit 1002
Page 166

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

fuboTV Media Inc.
Exhibit 1002
Page 167

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

fuboTV Media Inc.
Exhibit 1002
Page 168

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.

fuboTV Media Inc.
Exhibit 1002
Page 169

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

fuboTV Media Inc.
Exhibit 1002
Page 170

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

fuboTV Media Inc.
Exhibit 1002
Page 171

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

fuboTV Media Inc.
Exhibit 1002
Page 172

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

fuboTV Media Inc.
Exhibit 1002
Page 173

prospective (or recommended) parent node for node N from the PRPL and an alternate recommended parent node from the SRPL (if either the PRPL or SRPL is empty, the instructing node will make a "nil" selection from that list – the instructing node knows from step 208 that at least one of the lists will not be empty). The instructing node then goes to step 211 in which it determines whether the alternate recommended parent node is closer to the server (i.e., higher uptree) than the recommended parent node derived from the PRPL. If the alternate recommended parent node is on the same level as, or on a lower level than the recommended parent node derived from the PRPL (or if the selection from the SRPL is nil), then the answer to the query in step 211 is no.

In such event, the instructing node goes to step 212 in which it consults its topology database and devises a connection address list from the recommended parent node back to the server, and provides such connection address list to node N (if the instructing node is a user node, then the connection address list leads back to the server through the instructing node).

As in step 205 discussed above, at this point, node N performs the Prospective Child Node's Connection Routine discussed above in connection with FIG. 14, and in step 212, the instructing node inserts node N into its topology database as a child of the recommended parent node.

The instructing node moves to step 213 in which it adds node N to an appropriate position on the PRPL. It does this because, as a result of step 202, it knows that the node N is capable of re-transmitting content data to its own child nodes.

The instructing node then goes to step 214 in which it checks its topology database to determine whether the recommended parent, with node N presumptively docked to it as a child node, is fully occupied (in the example here of a binary tree network, whether it has two child nodes). If the answer is no, then the instructing node has finished this routine. If the answer is yes, then the instructing node goes to step 215 in which it removes the recommended parent from the PRPL because it is now deemed to not be an available prospective parent node.

Then the instructing node goes to step 216 in which it consults its topology database to determine whether any of the recommended parent node's children is a low-bandwidth node (in this example, knowing that node N is not a low-bandwidth node and knowing that the recommended parent node has two child nodes, the question is whether the child node other than node N is a low-bandwidth node.) If the answer is no (i.e., all the recommended parent node's children are high-bandwidth nodes), then the instructing node has finished the routine.

fuboTV Media Inc.
Exhibit 1002
Page 174

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

fuboTV Media Inc.
Exhibit 1002
Page 175

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

fuboTV Media Inc.
Exhibit 1002
Page 176

- E (a third level node with available capacity)
  - J (the fourth level node with available capacity having the highest utility rating)
- K (a fourth level node with available capacity)

The SRPL in this example would be as follows:

- SRPL
  - C (the fully occupied second level node with a low-bandwidth child node)
  - D (the fully occupied third level node with a low-bandwidth child node)
  - I (the fully occupied fourth level node with a low-bandwidth child node)

If node N is a low-bandwidth node, node A will give, in this example, node N the following as its new connection address list:

- E-B-A-S.

Since node E would not have two child nodes, it would remain on the PRPL.

If node N is a high-bandwidth node, node A will compare (step 211 of FIG. 20B) the first node on the PRPL (the recommended parent node) with the first node on the SRPL (the alternate recommended parent node). Here, node C, the first node on the SRPL is a higher level node than node E, the first node on the PRPL. So, node A will send a disconnect signal (step 218 of FIG. 20B) to node F, node C's low-bandwidth child node. Then it will provide node N with the following new connection address list and add node N to the PRPL (step 219 of FIG. 20B):

- C-A-S.

Since node C would now have two high-bandwidth child nodes (nodes N and G), node C would be removed from the SRPL (step 222 of FIG. 20B).

Further, in this example wherein node N is a high-bandwidth node, when node F returns to the server 11 (or, in another example, goes to Node A) for a new connection, the

fuboTV Media Inc.
Exhibit 1002
Page 177

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

fuboTV Media Inc.
Exhibit 1002
Page 178

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.

fuboTV Media Inc.
Exhibit 1002
Page 179

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

fuboTV Media Inc.
Exhibit 1002
Page 180

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

fuboTV Media Inc.
Exhibit 1002
Page 181

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

fuboTV Media Inc.
Exhibit 1002
Page 182

node X. The bumped node R then sends a priority join to its own green sibling A as the target. These actions cause the reconfiguration events to cascade until the edge of the tree is reached (of note, the algorithm described in this paragraph may have essentially the same effect as illustrated in Figures 28 and 29 – under certain circumstances the algorithm described in this paragraph may be somewhat more reliable).

FIG. 28 illustrates an example distribution system comprising node P as a first level node 12; nodes Q and X as second level nodes 13; nodes A and B as third level nodes 14; nodes C, D, E an F as fourth level nodes 15; nodes G and H as fifth level nodes 17; nodes I and J as sixth level nodes 20; nodes K and L as seventh level nodes 21 and nodes M and N as eighth level nodes 22. Nodes Q, B, C E, H, I, L and M have red propagation ratings, as symbolized by the dashed circles, and the remaining nodes have green propagation ratings as symbolized by the solid circles. As a result of node X's departing from the system, node P sends out an upgrade propagation rating signal to node Q and a propagation signal to the children of node P's departed child node X (i.e., nodes A and B).

In another example (which example is intended to be illustrative and not restrictive), during a graceful depart of node x, node x may send out messages to its parent and children immediately prior to its departure – rather than the parent of node x initiating these messages as described above.

In any case, continuing the example of FIG. 28, node Q changes its propagation rating to green. Further, in this example nodes A and B begin the Child Node's Propagation Routine. With respect to node A, it answers the query of step 271 in the affirmative. Since its rating is green, it also answers the query of step 272 in the affirmative. In step 273 node A changes the setting of its reconfiguration flag buffer to show a red propagation rating. In step 274 it re-transmits the propagation signal to its child nodes C and D. It answers the query of step 275 in the negative because it is not docked for purposes of receiving content data with node P. Node A then goes to step 276 wherein it consults the ancestor portion of its topology database and creates a new connection address list starting with its grandparent. The new connection address list is P-S. Then node A performs the Prospective Child Node's Connection Routine, starting with step 141. (See FIG. 14.) Assuming that no other intervening events have occurred, node A will successfully dock with node P.

Node B answers the query of step 271 in the affirmative and, because its propagation rating was red when it received the propagation signal, it answers the query of step 272 in the negative. From there it goes to step 278 in which it changes its propagation rating to green. Then in step 279 node B consults the ancestor portion of its topology database and creates a

fuboTV Media Inc.
Exhibit 1002
Page 183

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.

fuboTV Media Inc.
Exhibit 1002
Page 184

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
```


fuboTV Media Inc.
Exhibit 1002
Page 185

```
OriginalRedChild := X.RedChild
X.RedChild := X.GreenChild
X.GreenChild := RedSibling
If (OriginalGreenChild <> null)
{
  Propagate(OriginalGreenChild, OriginalRedChild)
}
End
```

Referring now to a "Malfunctioning Nodes" example (which example describes a voting system between grandchildren and grandparents about a potentially malfunctioning parent and is intended to be illustrative and not restrictive), it is noted that what happens when a node leaves the distribution network was discussed in the above example (e.g., when a node leaves the network, its parent will, at the next utility rating event, report that it has room for a new node, except when the parent has been contacted about the missing node by its grandchild node or nodes, in which event, a reconfiguration event will proceed as described above). Sometimes, however, a node intends to remain in the network, but there is a failure of communication.

In this regard, each node's topology database may include information regarding that node's ancestors and descendants. In the event that a child node stops receiving signals from its parent node, the child may, in one example, send a "complaint" message to its grandparent. The grandparent may check whether the parent is still there. If it is not, then the grandparent may send a propagation signal to the child nodes of the missing parent.

Further, if the grandparent detects that the parent node is actually still there, then the grandparent node may send a disconnect signal to the parent node (e.g., sending it back to the server to begin the connection process again). This may occur, for example, when one of the two following conditions is exists: (i) the child node is the only child node of the parent, or (ii) the child and its sibling(s) are complaining to the grandparent.

In this example, the grandparent would also send a propagation signal to the child nodes of the disconnected parent, and a reconfiguration event would occur.

However, if the child node has siblings and they are not sending complaint signals to the grandparent, then the grandparent may assume that the problem is with the complaining child node. Thus, grandparent may send a disconnect signal to the complaining child node (e.g., sending it back to the server to begin the connection process again). If the complaining

fuboTV Media Inc.
Exhibit 1002
Page 186

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

fuboTV Media Inc.
Exhibit 1002
Page 187

removal. At this point the green child of node B (i.e., node C or C') will move up a level connecting to grandparent node A, and the red child of node B (the other of node C or C'), will connect as a child of its former sibling. The reconfiguration event will then propagate as discussed above.

An exception to the above is the case where node C is the only child of node B. Under these circumstances node B may be disconnected in this example by node A based solely on the recommendation of node C. In another example, the system will not disconnect a node with one child based solely on the complaint of that child (in this case there may be insufficient data to accurately assign blame - disconnecting the node being complained against and promoting the complainer may have the potential to promote a node that is experiencing problems).

Referring now to FIG. 31 (showing a flowchart associated with this example), node A receives a complaint from node C about node B in step 311. Node A then goes to step 312 in which it checks its topology database to determine whether node C is the only child of node B. If the answer is no (as shown in FIG. 30), then node A goes to step 313 in which node A determines whether there is a communication problem between it and node B. If the answer is no, then node A proceeds to step 314 in which it determines whether it has received a similar complaint about node B from node C' within a predetermined period of time of node A's having received the complaint from node C. If the answer to that query is no, then node A goes to step 315 in which it sends a disconnect signal to node C. At that point node A has completed the Grandparent's Complaint Response Routine (in one example a propagation signal does not necessarily have to sent by node A to node C's child nodes - if any exist, they will complain to node B which, upon discerning that node C is no longer in the distribution network, will send a propagation signal to such complaining nodes).

If the response to any of the queries in steps 312, 313 and 314 is yes in this example, then node A proceeds to step 316 in which it disconnects node B. Node A then proceeds to step 317 in which it sends a propagation signal to its grandchild nodes (here nodes C and C') and a reconfiguration event will occur as described above.

The method of this example described above takes a "2 out of 3" approach on the question of node removal. In this regard, it is believed that many interior nodes in a binary tree distribution network constructed pursuant the present invention will have direct connections to three other nodes: their parent, their green child, and their red child. The connections between a node and its three "neighbor" nodes can be thought of as three separate communications channels. In this example a node is removed from the network

fuboTV Media Inc.
Exhibit 1002
Page 188

when there are indications of failure or inadequate performance on two of these three channels. When there are indications that two of the channels are working normally, the complaining node is presumed to be "unhealthy" and is removed.

In the case where an interior node has only a single child, the "complaint" of that child to its grandparent is sufficient, in this example, to remove the parent node - even when the node to be removed is communicating perfectly well with it's own parent. In other words, given the communication chain A-B-C where node A is the parent of node B and node B is the parent of node C, and given that node C has no siblings, then a complaint of node C to node A will cause node A to remove node B in this example regardless of the fact that node A and node B are not having communication problems (this course of action may help ensure that a potentially unreliable node does not move up the hierarchy of nodes, even at the cost of occasionally sending a "healthy" node back to the edge of the distribution chain based on uncorroborated complaints of its child). Of course, as mentioned above, there are examples where the system will not disconnect a node with one child based solely on the complaint of that child.

In another example of voting (which example is intended to be illustrative and not restrictive), instead of (or in conjunction with) nodes voting on the behavior of other nodes, nodes may continually grade their own performance and if they detect that there is insufficient bandwidth on two or more of their links (links to their parent and their children) the node will perform a graceful depart from the network (initiating a reconfiguration as described elsewhere) and then the node will either climb its path to an ancestor node (or go directly to the server) to request a connection path to the edge of the tree. This system allows nodes to "step aside" when they detect connection / bandwidth related issues.

Referring now to a "Stepped Delay of Playing Content Data" example (which example is intended to be illustrative and not restrictive) it is noted that in order to have all the user nodes experience the playing of the content data at approximately the same time, regardless of the level in which the node resides, the content data buffer 125 shown in FIG. 13 of a user node may be larger the higher uptree it is in a distribution chain (i.e., the closer it is to the root server). The larger the buffer, the greater the amount of time between a particular event (or from the node's receiving the content data), and the node's actually playing the content data in its content data buffer. To the extent that the content data buffer of a node is sized to vary the time that the playing of the content data is started, the content data buffer is a delay. Alternatively, the delay may be a timer or a delay buffer 126 as shown in

fuboTV Media Inc.
Exhibit 1002
Page 189

FIG. 13, or a combination of elements. The delay's purpose may be to help assure that all users experience the play of the content data approximately simultaneously.

For a distribution network having n levels of user nodes cascadingly connected to a server node, where n is a number greater than one (and therefore the distribution network comprises the server node and first through nth level user nodes), the period of time (i.e., the delay time) between:

(i) the moment content data is received by a node or, from a predetermined moment such as, for example, a reporting event or utility rating event (the occurrence of which may be based on time periods as measured by a clock outside of the distribution system and would occur approximately simultaneously for all nodes); and

(ii) the playing of the content data by the node;

is greater for a node the higher uptree the node is on the distribution chain. That is, the delays in first level nodes create greater delay times than do the delays in second level nodes, the delays in second level nodes create greater delay times than the delays in third level nodes and so on. Described mathematically, in an n level system, where x is a number from and including 2 to and including n, the delay time created by a delay in an (x-1) level node is greater than the delay time created by a delay in an x level node.

By way of example (which example is intended to be illustrative and not restrictive), first level nodes could have a built in effective delay time in playing content data of one-hundred-twenty seconds, second level nodes could have a built in effective delay time of one-hundred-five seconds, third level nodes could have a built in effective delay time of ninety seconds, fourth level nodes could have a built in effective delay time of seventy-five seconds, fifth level nodes could have a built in effective delay time of sixty seconds, sixth level nodes could have a built in effective delay time of forty-five seconds, seventh level nodes could have a built in effective delay time of thirty seconds, and eighth level nodes could have a built in effective delay time of fifteen seconds.

The delays may take advantage of the fact that transmission of the packets forming the content data from one node to another may take less time than the actual playing of such packets. This may allow for essentially simultaneous playing of the content data throughout all the levels of the distribution network. The delays may also allow nodes to handle reconnection and reconfiguration events without any significant loss of content data.

In another example (which example is intended to be illustrative and not restrictive), the system may not endeavor to have buffers closer to the server contain more data than buffers of nodes that are further away. Instead, every node may have a buffer that can hold,

fuboTV Media Inc.
Exhibit 1002
Page 190

for example, approximately two minutes worth of A/V content. The system may endeavor to fill every node's buffer as rapidly as possible, regardless of the node's distance from the server, in order to give each node the maximum amount of time possible to deal with network reconfiguration and uptree node failure events.

Of note, nodes may receive a "current play time index" when they connect to the server. This value may be used to keep all nodes at approximately the same point in the stream – to have all nodes playing approximately the same moment of video at the same time – by telling them, essentially, begin playing A/V content at this time index. The play time index may be, for example, approximately two minutes behind "real time" – allowing for nodes to buffer locally up to two minutes of content (the present invention may permit drift between nodes within the buffer (e.g., two minute buffer) or the present invention may reduce or eliminate this drift to keep all nodes playing closer to "lock step").

Turning now to a general description of an example process for Node X to enter a network including server Node S, Node X must contact Node S to initiate the process. For illustrative purposes, in the following example (which example is intended to be illustrative and not restrictive) each node in the network may have no more than two connections, and the root server (i.e., the highest node in the tree) shall be referred to as Node S.

The process of this example begins with Node X making a request of Node S to join the network.

If Node S has an open connection (e.g., including one or more connected bumpable non-repeat-capable nodes which may be replaced due to an incoming connection request from a repeat-capable node), Node S can accept Node X's request to join the network, and Node X becomes a child of Node S.

If, on the other hand, Node S does not have an open connection, Node S may reject Node X's request, and may provide to Node X a node-by-node path from Node S to Node P, which Node S understands to be in the network, and under which, as determined by Node S, Node X may be able to join the network.

The process of this example continues with Node X making a request of Node P to join the network.

If Node P does not respond – regardless of why Node P does not respond – Node X may request connection from the next higher node in the path. For the purpose of this illustration, thereafter the next higher node is now considered Node P. Node X may make such a request of successive Node Ps, until Node X receives a response.

fuboTV Media Inc.
Exhibit 1002
Page 191

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

fuboTV Media Inc.
Exhibit 1002
Page 192

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

fuboTV Media Inc.
Exhibit 1002
Page 193

necessary. Node X would then begin transmitting FloodData messages, as shown in FIG 34A.

Since, in this example, FloodData messages are transmitted for a purpose of determining Node X's upstream bandwidth, in one embodiment, Node X would transmit an amount of FloodData to the flood test target as shown in FIG 34A. In various embodiments, the maximum amount of FloodData may be limited to a fixed number of FloodData messages, to a fixed amount of data transmitted within FloodData messages, or to transmission of FloodData or FloodData messages over a fixed period of time.

To protect the flood test target from being inundated with FloodData (e.g., where Node X has a very high upstream bandwidth capability), Node X may cap (e.g., be so directed by Node S) the FloodData transmission to, e.g., 100 Kbits/sec, 200 Kbits/sec, or some other data rate that would be a desired data transmission rate for a node on the particular network of interest.

Thus, by receiving the FloodData messages, the flood test target can determine, for example, an upstream capability rating of Node X (provided that the flood test target has the same or more bandwidth in the downstream channel compared to the upstream capability of Node X.

In another example, both the target and the sender will know the bandwidth of the sender at the conclusion of the flood test. More particularly, the present invention may use TCP/IP (rather than UDP), and all packets sent may receive an acknowledgement. In this example, Node X, the node sending out the flood packets, sends out the next flood packet only after receipt of the first packet has been acknowledged by the target of the flood test. Thus, the number of flood data packets Node X is able to send per unit time (along with the size of those packets) may allow Node X to calculate its upstream bandwidth.

In another example, the target of the "flood test" may be any node including the server but excluding any firewalled or low-bandwidth nodes.

After the "flood test", Node X may then connect with Node S (or be instructed to go to another node if Node S can not/will not accept Node X). Of note, if Node X fails the "flood test", Node X may be sent far downtree or may be refused entry into the network entirely. Further, if node N is accepted under these circumstances it will be deemed a "bumpable" non-repeat-capable node.

Referring now in more detail to the "Firewall Status" example shown in FIG. 34C, this example process begins, as above, with Node X making a request to Node S to join the network. Node S then send Node X a Meta message. A firewall status message is a message

fuboTV Media Inc.
Exhibit 1002
Page 194

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.

fuboTV Media Inc.
Exhibit 1002
Page 195

- 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).

fuboTV Media Inc.
Exhibit 1002
Page 196

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).

fuboTV Media Inc.
Exhibit 1002
Page 197

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

fuboTV Media Inc.
Exhibit 1002
Page 198

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).

fuboTV Media Inc.
Exhibit 1002
Page 199

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

fuboTV Media Inc.
Exhibit 1002
Page 200

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

fuboTV Media Inc.
Exhibit 1002
Page 201

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:

fuboTV Media Inc.
Exhibit 1002
Page 202

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.

fuboTV Media Inc.
Exhibit 1002
Page 203

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:

fuboTV Media Inc.
Exhibit 1002
Page 204

(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

fuboTV Media Inc.
Exhibit 1002
Page 205

each of said grandchild nodes as great-grandchild nodes of said instructing node; and so on; and

(ii) said docked user nodes' respective bandwidth capacities, including a designation of whether said respective docked user node's bandwidth capacity is below a predetermined threshold (e.g., said respective docked user node is a low-bandwidth node) or at least said predetermined threshold (e.g., said respective docked user node is a high-bandwidth node);

(c) forming a primary recommended parent node list ("PRPL") comprised of the instructing node (if it has available capacity to transmit content data to said connection requesting user node) plus those docked user nodes having apparent available capacity to transmit content data to said connection requesting user node, with said instructing node being placed first on the PRPL (if it is on the PRPL) and said docked user nodes having apparent available capacity to transmit content date to said connection requesting user node being ranked with those docked nodes which are closer (i.e., in terms of network topology) to the instructing node being ranked higher than those docked nodes which are further away (i.e., in terms of network topology) and with equidistant (i.e., in terms of network topology) docked user nodes being ranked such that those docked nodes which have the higher utility ratings being ranked higher than docked nodes having lower utility ratings;

(d) forming a secondary recommended parent node list ("SRPL") comprised of the instructing node (if it has no available capacity to transmit content data to said connection requesting user node but does have at least one low-bandwidth node docked directly to it) plus said docked user nodes which (i) are high-bandwidth nodes with no available capacity to transmit content data to said connection requesting user node and (ii) have at least one low-bandwidth node docked directly to it, with said instructing node being placed first on the SRPL (if it is on the SRPL) and said docked user nodes on the SRPL being ranked with those docked nodes which are closer (i.e., in terms of network topology) to the instructing node being ranked higher than those docked nodes which are further away (i.e., in terms of network topology) and with equidistant (i.e., in terms of network topology) docked user nodes being ranked such that those docked nodes which have the higher utility ratings being ranked higher than docked nodes having lower utility ratings;

(e) determining whether the connection requesting user node is a low-bandwidth node;

(f) if the connection requesting user node is a low-bandwidth node:

fuboTV Media Inc.
Exhibit 1002
Page 206

(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

fuboTV Media Inc.
Exhibit 1002
Page 207

(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;

fuboTV Media Inc.
Exhibit 1002
Page 208

(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

fuboTV Media Inc.
Exhibit 1002
Page 209

(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;

fuboTV Media Inc.
Exhibit 1002
Page 210

(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:

fuboTV Media Inc.
Exhibit 1002
Page 211

(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;

fuboTV Media Inc.
Exhibit 1002
Page 212

(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;

fuboTV Media Inc.
Exhibit 1002
Page 213

(c) with respect to each child node receiving the propagation signal which did not have the highest propagation rating prior to its receiving the propagation signal (hereinafter referred to as a "red node"), upon its receiving the propagation signal:

(i) setting the propagation rating of the red node to the next higher grade above the propagation rating of that red node before it received the propagation signal;

(ii) having the red node undock from its parent node (if it was docked with its parent node before it received the propagation signal);

(iii) devising a first connection address list comprising the address of the red node's sibling node which had the had the highest propagation rating prior to the red node's receiving the propagation signal followed by the red node's ancestor list;

(iv) having said red node go to (or attempt to go to) the node at the top of said first connection address list;

(v) having said red node determine whether the node at the top of said first connection address list is part of the distribution network;

(vi) if the node at the top of said first connection address list is not part of the distribution network, deleting such node from said first connection address list and repeating steps (c)(iv) and (c)(v) with respect to the next node at the top of said first connection address list;

(vii) if the node at the top of said first connection address list is part of said distribution network, determining whether the node at the top of said first connection address list has available capacity to transmit content data to said red node; and

(viii) if the node at the top of said first connection address list is part of said distribution network and has available capacity to transmit content data to said red node, having said red node dock with the node at the top of said first connection address list (with regard to these steps (vii) and (viii), in another example (which example is intended to be illustrative and not restrictive), when a red node connects to a green node as part of a network reconfiguration event, it does not matter whether there is "available capacity" or not for the green node to accept its incoming red sibling – the green node has no choice but to accept the node (these actions are facilitated in this example by the red node issuing a "priority join" to the green node; if the green node does not have "available capacity" it has to kick its own red child to make room for the incoming green node));

fuboTV Media Inc.
Exhibit 1002
Page 214

(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

fuboTV Media Inc.
Exhibit 1002
Page 215

propagation rating prior to the red node's receipt of the propagation signal (hereinafter the "green node") to retransmit the propagation signal to the green node's child nodes (if any).

In another embodiment relating to a computer information distribution network comprising an instructing node having a fist user node docked with said instructing node to receive content data therefrom and a second user node docked with said first user node to receive content data therefrom, a process for handling a purported communication interruption between said first user node and said second user node is provided, said process including the following steps:

(a) having said instructing node receive a first complaint about said first user node from said second user node;

(b) if the second user node is the only user node docked with said first user node at a point in time, if said instructing node receives another complaint about the first user node from another user node within a predetermined period of time from when said instructing node received said first complaint or if said instructing node experiences a communication problem between it and said first user node during said predetermined period of time: (i) disconnecting said first user node from said instructing node; and

(c) if said instructing node has not experienced a communication problem between it and said first user node during said predetermined period of time and if a plurality of user nodes are docked with said first user node at said point in time and said instructing node has not received another complaint about said first user node from a user node other than said second user node within said predetermined period of time: (i) disconnecting said second user node from said first user node.

In one example, step (b) may include the following step: (ii) sending a signal to the user nodes docked with said first user node indicating the departure of said first user node from said network.

In another example, each user node may have no more than a predetermined maximum number of child nodes docked with it, and wherein step (b) may include the following steps

(ii) providing information at said point in time to each of said second user node, the other user nodes which had been docked with said first user node (if any) and at least one of any user nodes cascadingly connected in the distribution network under said first user node prior to said first user node's being disconnected from said instructing node (if any) as follows:

fuboTV Media Inc.
Exhibit 1002
Page 216

(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

fuboTV Media Inc.
Exhibit 1002
Page 217

(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

fuboTV Media Inc.
Exhibit 1002
Page 218

connected to said server node, wherein n is a number greater than one, wherein each user node includes a delay which causes the playing of content data by such node to be delayed a period of time (hereinafter "delay time") from a point in time, wherein x is a number from and including 2 to and including n, and wherein delays in (x-1) level nodes create greater delay times than do delays in x level nodes.

In one example, said point in time is an event experienced approximately simultaneously by substantially all user nodes.

In another example, said point in time is when said node receives content data.

While a number of embodiments and examples of the present invention have been described, it is understood that these embodiments and examples are illustrative only, and not restrictive, and that many modifications may become apparent to those of ordinary skill in the art. For example, any desired branch factor may be used (*see*, e.g., FIG. 46A showing a branch factor of 1 (i.e., each parent node may have only one child node); FIG. 46B showing a branch factor of 2 (i.e., each parent node may have one or two children nodes); and FIG. 46C showing a branch factor of 3 (i.e., each parent node may have one, two or three children nodes)). Of course, other branch factors (e.g., 4, 5, 6, etc) may be used and, as discussed above, the distribution network may be a hybrid network utilizing subtrees having different branch factors). Further still, a modified Utility Rating may be determined as follows: Utility Rating = Time (of connection) x Bandwidth Rating. Further still, a node ID may be based upon: (a) an IP address (external/internal/DHCP); (b) a URL; and/or (c) a Port. Further still, the root server may propagate "its" time upon assignment of a connection and the user node may get a "delta" (i.e., difference) from its clock to use (in the event of a reset the user node may re-acquire the time from the server (or from another time-providing source)). Further still, the various time intervals discussed herein may be predetermined time intervals (e.g., .5 sec., 1 sec, 1 min.) and/or the time intervals may be based upon some varying criteria (e.g., flow of data in the network, number of nodes in the network, number of levels in the network, position of a given node in the network). Further still, partial propagation of data (e.g., ancestor data and/or descendent data) from one or more nodes may be permitted. Further still (as discussed above), one or more of the steps described herein may be omitted (and the various steps which are performed may not necessarily need to be carried out in the order described herein (which description was intended to represent a number of examples, and not be restrictive)). Further still, the root server and each of the user nodes may utilize essentially identical software for carrying out the network configuration, reconfiguration and data transfer operations described herein or the root server may have dedicated server software

fuboTV Media Inc.
Exhibit 1002
Page 219

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).

fuboTV Media Inc.
Exhibit 1002
Page 220

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.

fuboTV Media Inc.
Exhibit 1002
Page 221

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,

fuboTV Media Inc.
Exhibit 1002
Page 222

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.

fuboTV Media Inc.
Exhibit 1002
Page 223

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;

fuboTV Media Inc.
Exhibit 1002
Page 224

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.

fuboTV Media Inc.
Exhibit 1002
Page 225

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

fuboTV Media Inc.
Exhibit 1002
Page 226

repeating the searching step until one such path is located; and

designating the located path as a path to the open node.

27. A method for propagating a network topology model of a binary tree topology network through the network, the network including a plurality of nodes, including a root node and at least one leaf node, operatively networked together to provide a communications path for data, each of the plurality of nodes except the root node being docked downtree of a parent node, the parent node being one of the plurality of nodes, and each of the plurality of nodes adapted to be docked uptree of a first child node and a second child node, each of the at least one leaf nodes having none of the plurality of nodes docked downtree of and therewith, the method comprising the steps of:

providing a banned list message from the root node to each of the plurality of nodes that are docked downtree of and with the root node, the banned list message comprising an identifier for each of a plurality of banned nodes;

for each of the plurality of nodes receiving the banned list message from the root node or from another one of the plurality of nodes:

storing in a memory accessible to the receiving node, the identifiers for each of the plurality of banned nodes; and

providing the banned list message to each of the plurality of nodes docked downtree of and with the receiving node;

thereby propagating the list of banned nodes through the network.

28. The method of claim 27, wherein the communications path for data comprises a downtree communications path for data.

29. The method of claim 27, wherein the communications path for data comprises a downtree communications path for data and an uptree communications path for data.

30. The method of claim27, further comprising the step of rejecting a register message from a new node that is not part of the network if the node is identified on the banned list.

fuboTV Media Inc.
Exhibit 1002
Page 227

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;

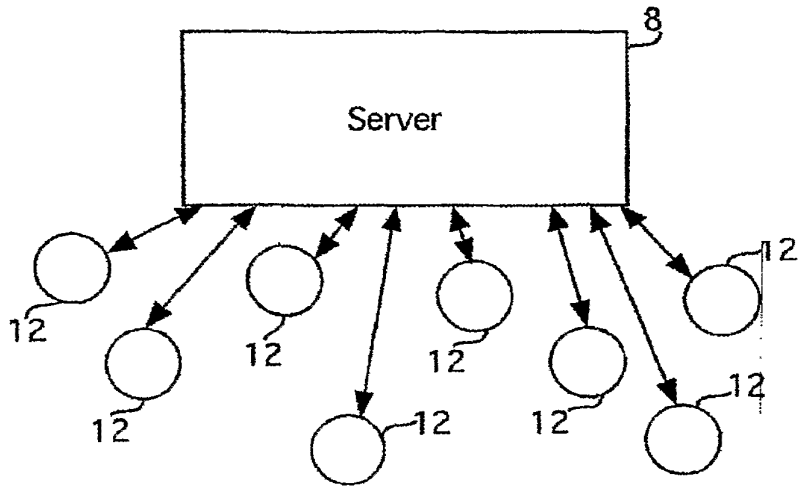
waiting a predefined period for an acknowledgement of the first response message, and if no acknowledgement is received, sending a second response message to the first node over the separate channel;

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

fuboTV Media Inc.
Exhibit 1002
Page 228

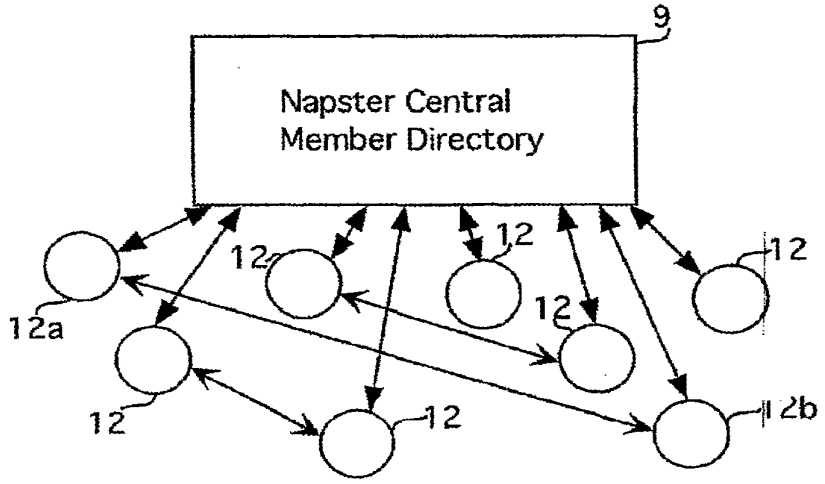
waiting a predefined period for an acknowledgement of the third response message, and if no acknowledgement is received, sending a firewall status on message to the first node over the first channel.

fuboTV Media Inc.
Exhibit 1002
Page 229



PRIOR ART

Fig. 1



PRIOR ART

Fig. 2

fuboTV Media Inc.
Exhibit 1002
Page 230

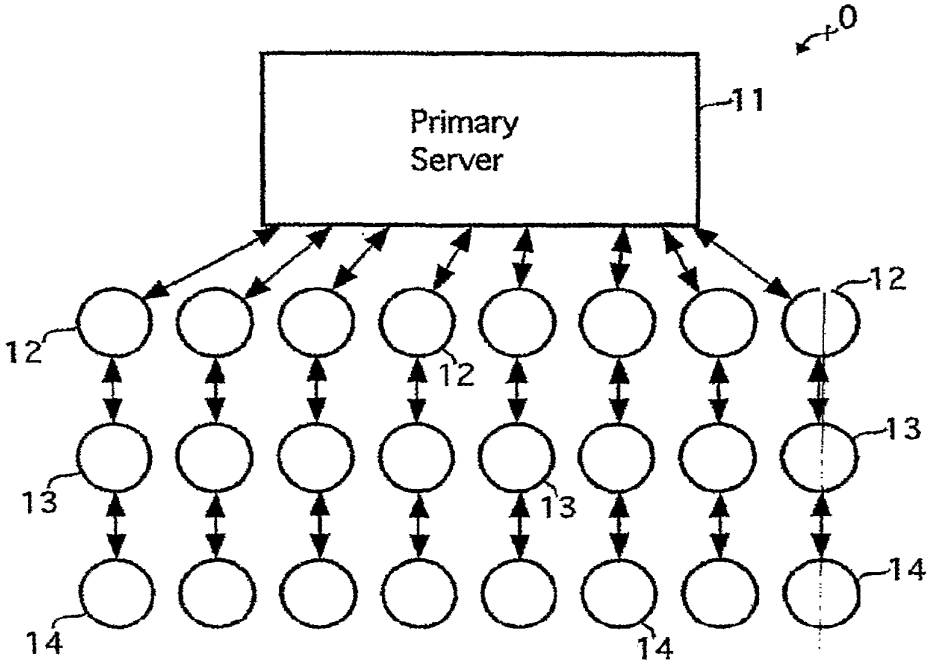


Fig. 3

"Start" [illegible]

[illegible]

Corresponds to [illegible] application [illegible]

31

fuboTV Media Inc.
Exhibit 1002
Page 231

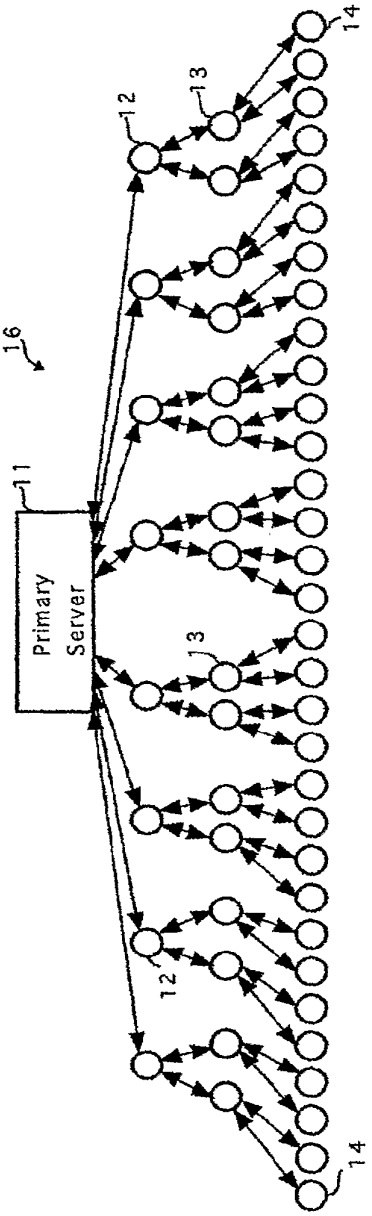


Fig. 4

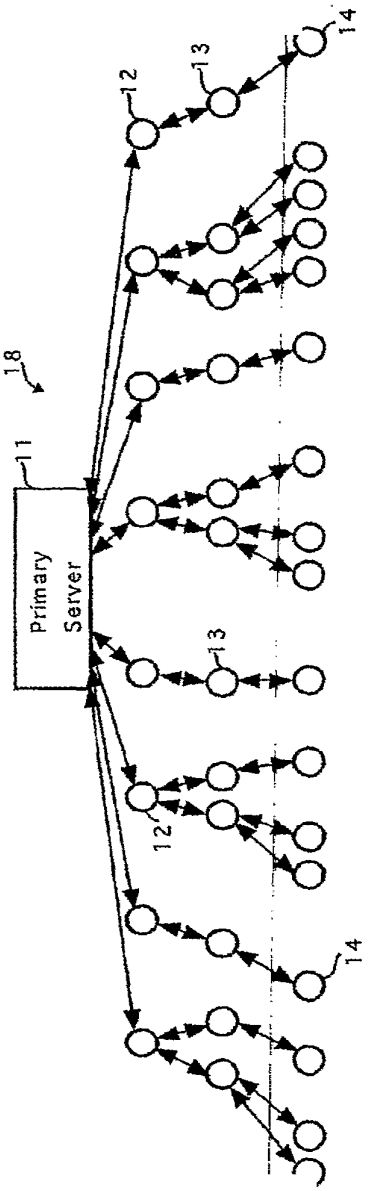


Fig. 5

fuboTV Media Inc.
Exhibit 1002
Page 232

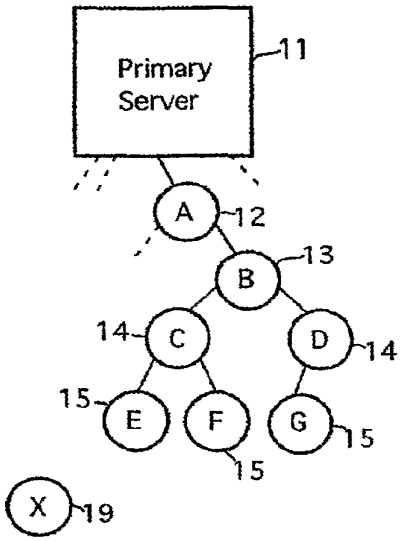


Fig. 6

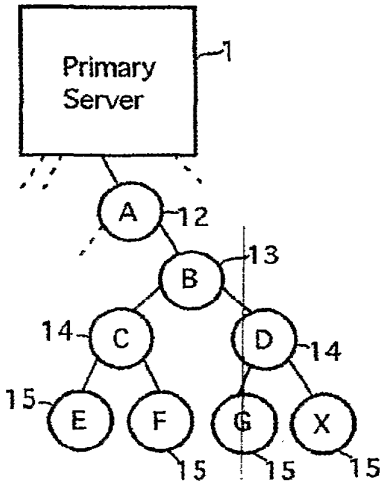


Fig. 7

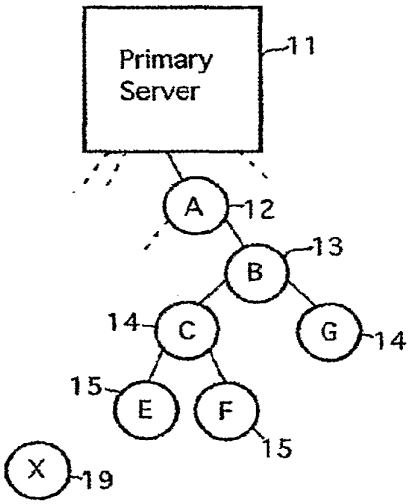


Fig. 8

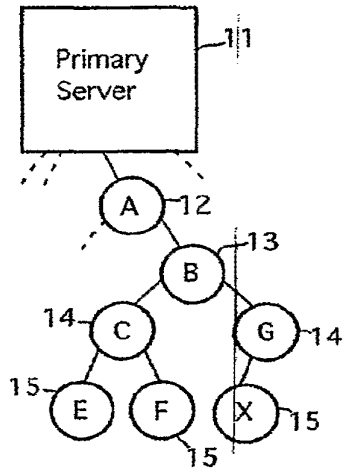


Fig. 9

fuboTV Media Inc.
Exhibit 1002
Page 233

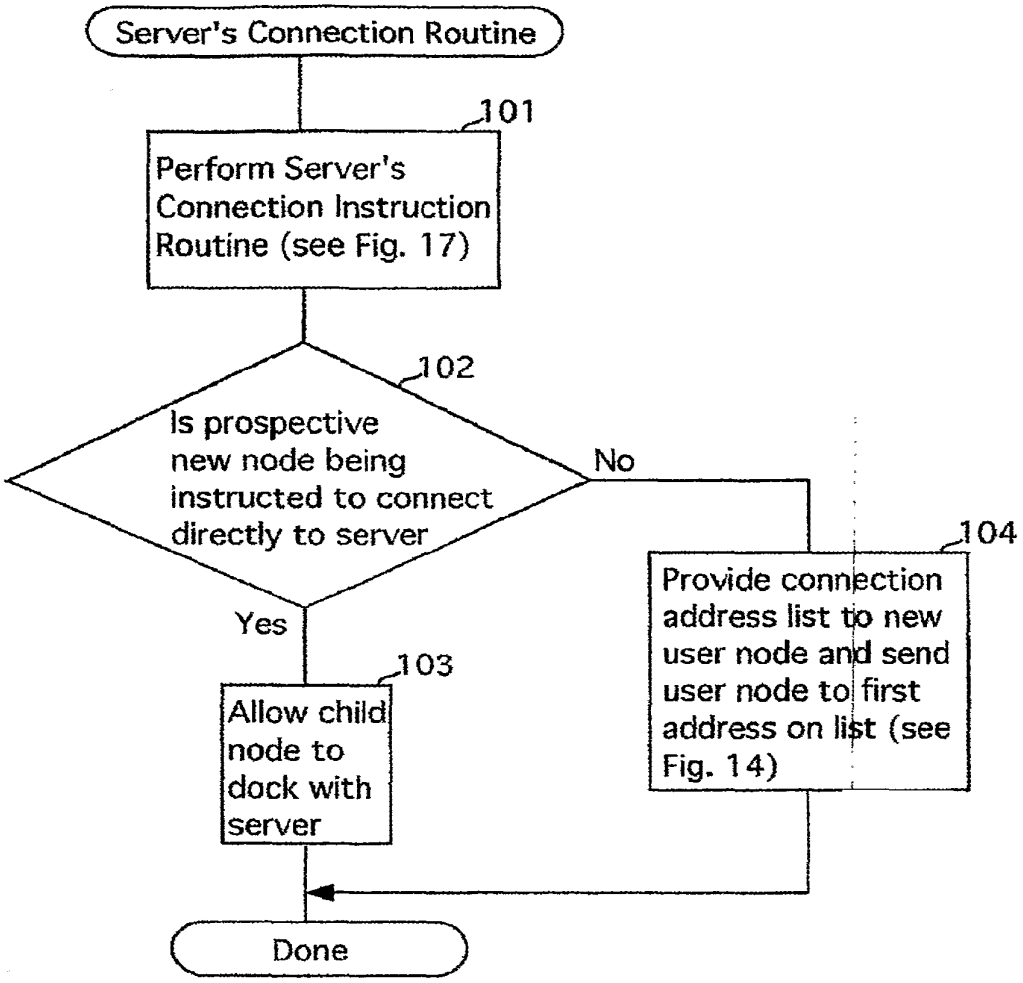


Fig. 10

fuboTV Media Inc.
Exhibit 1002
Page 234

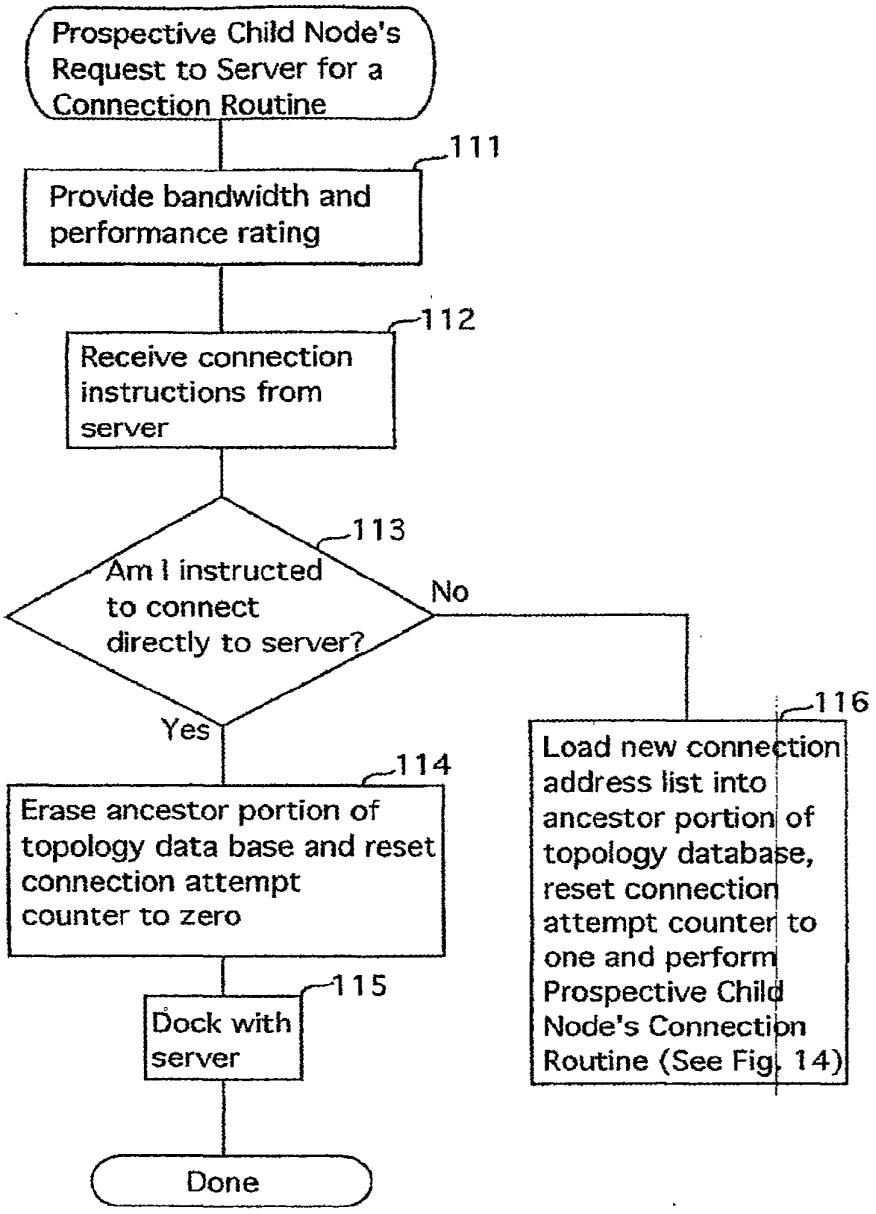


Fig. 11

fuboTV Media Inc.
Exhibit 1002
Page 235

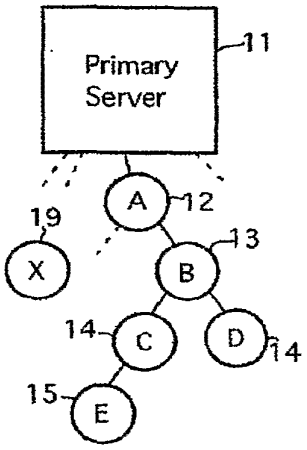


Fig. 12A

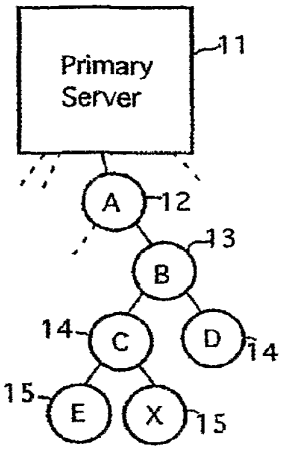


Fig. 12B

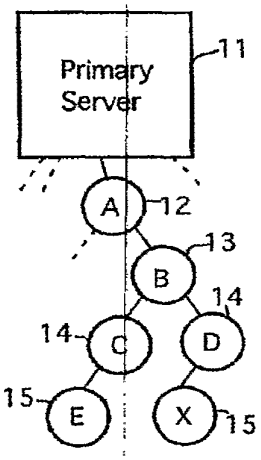


Fig. 12C

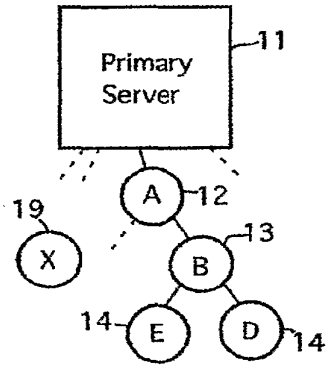


Fig. 12D

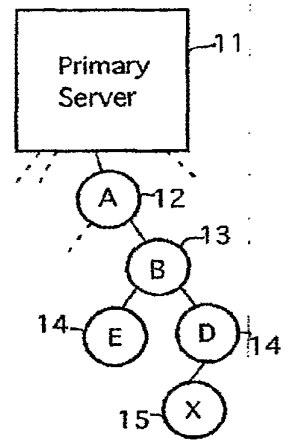


Fig. 12E

fuboTV Media Inc.
Exhibit 1002
Page 236

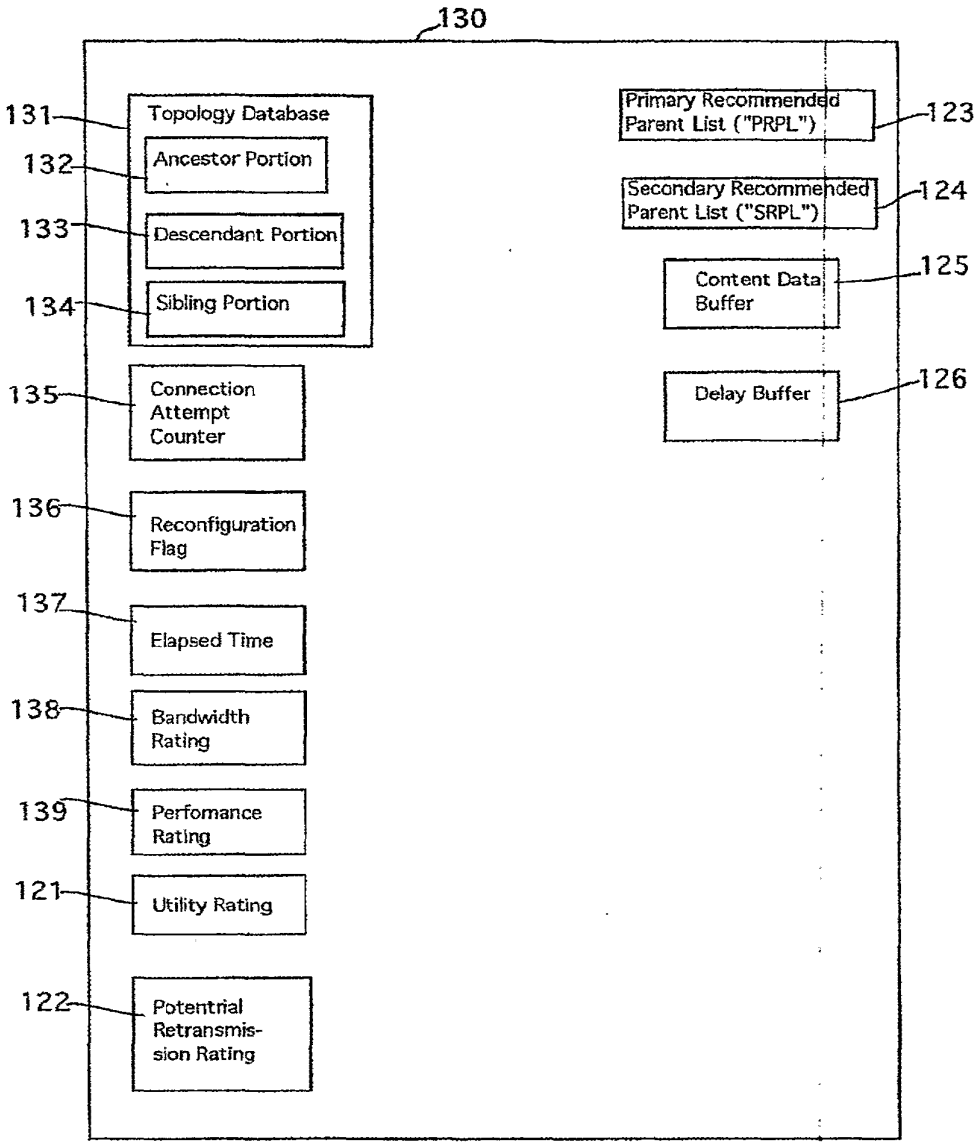


Fig. 13

fuboTV Media Inc.
Exhibit 1002
Page 237

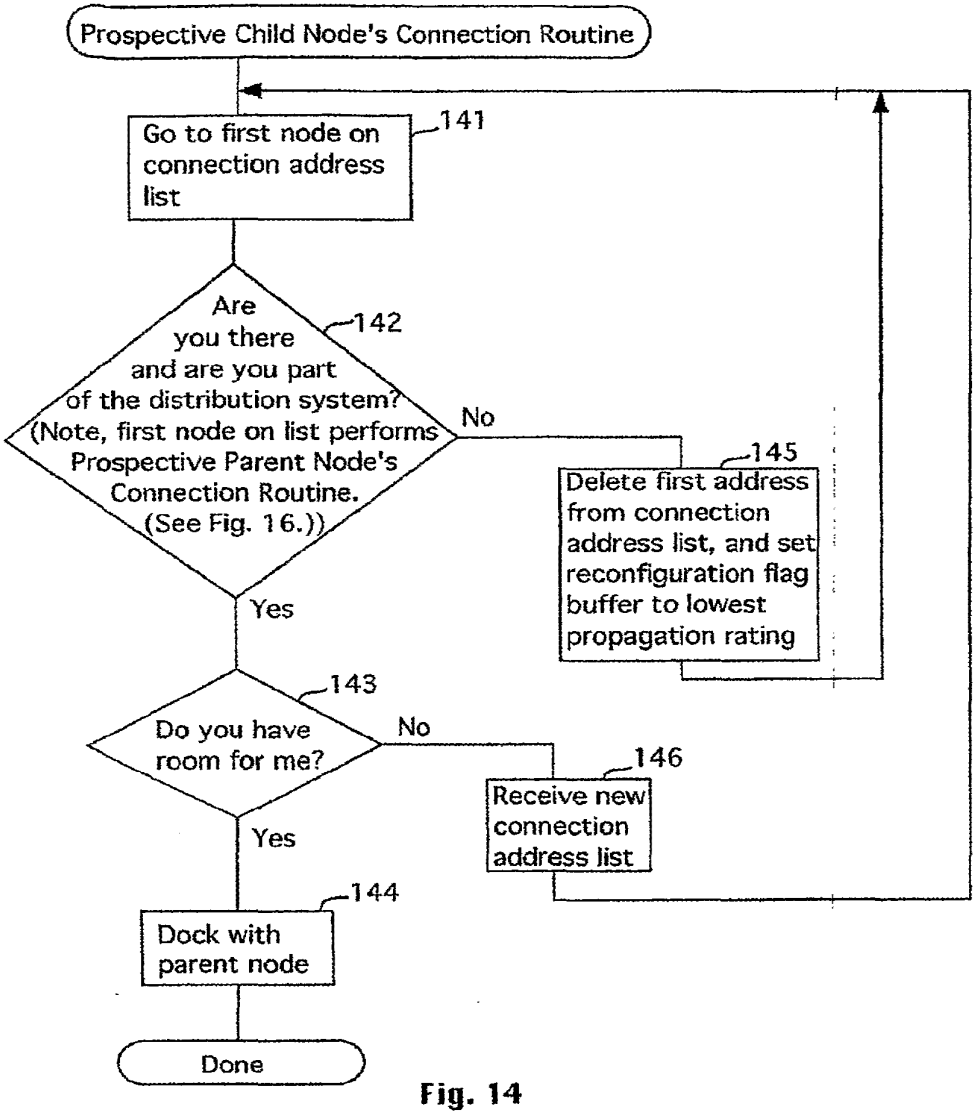


Fig. 14

fuboTV Media Inc.
Exhibit 1002
Page 238

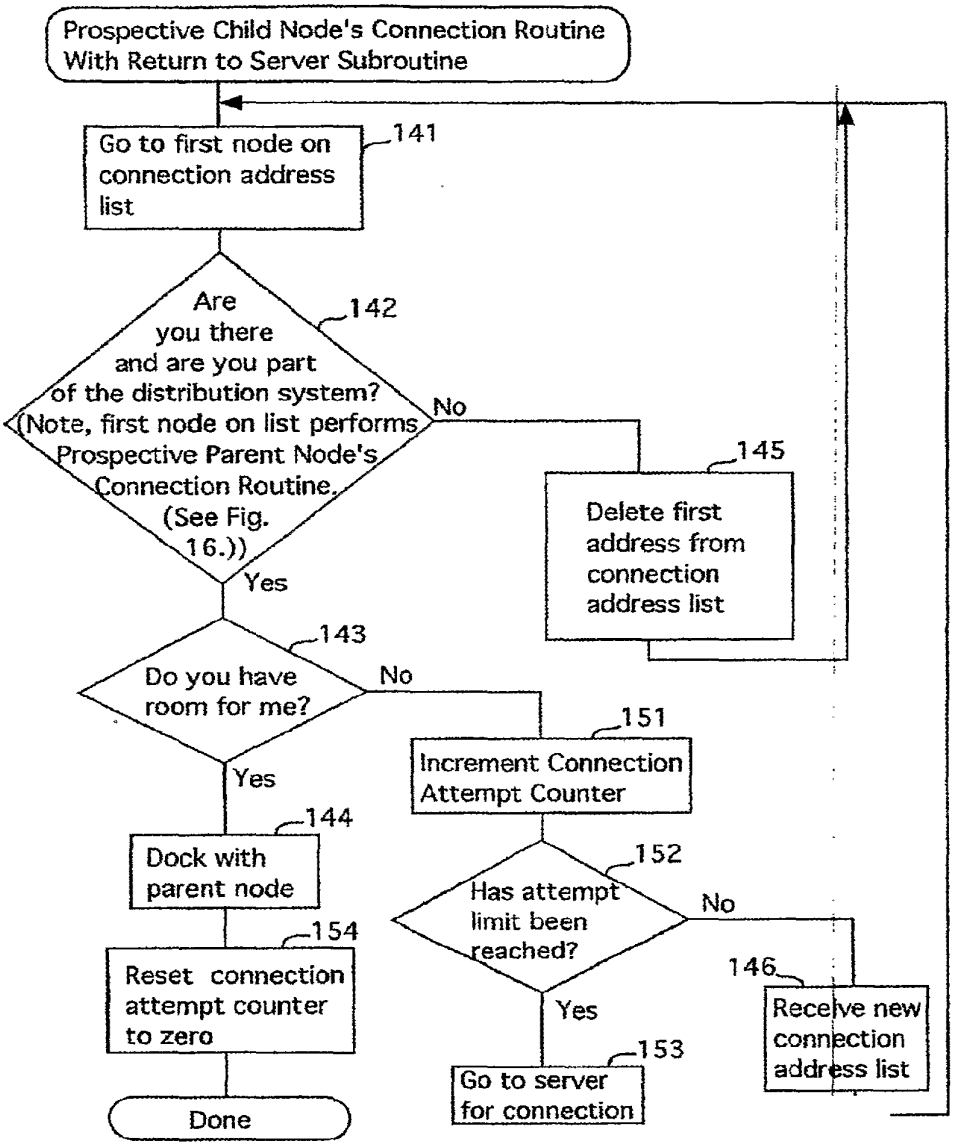


Fig. 15

fuboTV Media Inc.
Exhibit 1002
Page 239

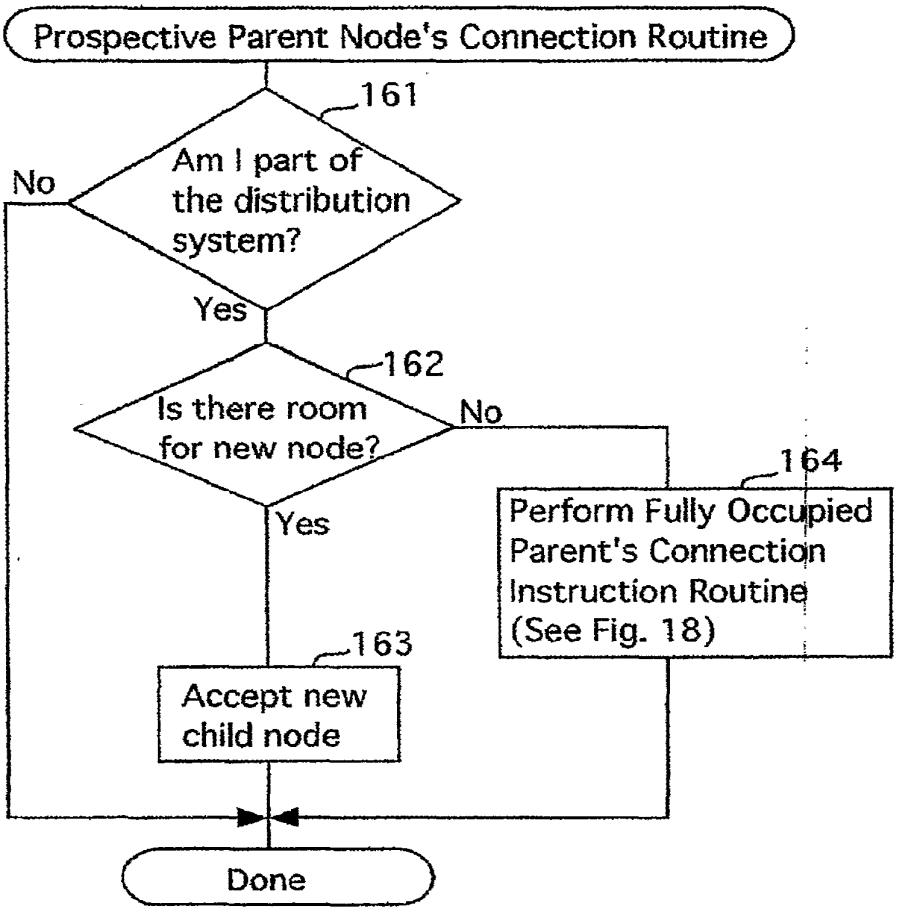


Fig. 16

fuboTV Media Inc.
Exhibit 1002
Page 240

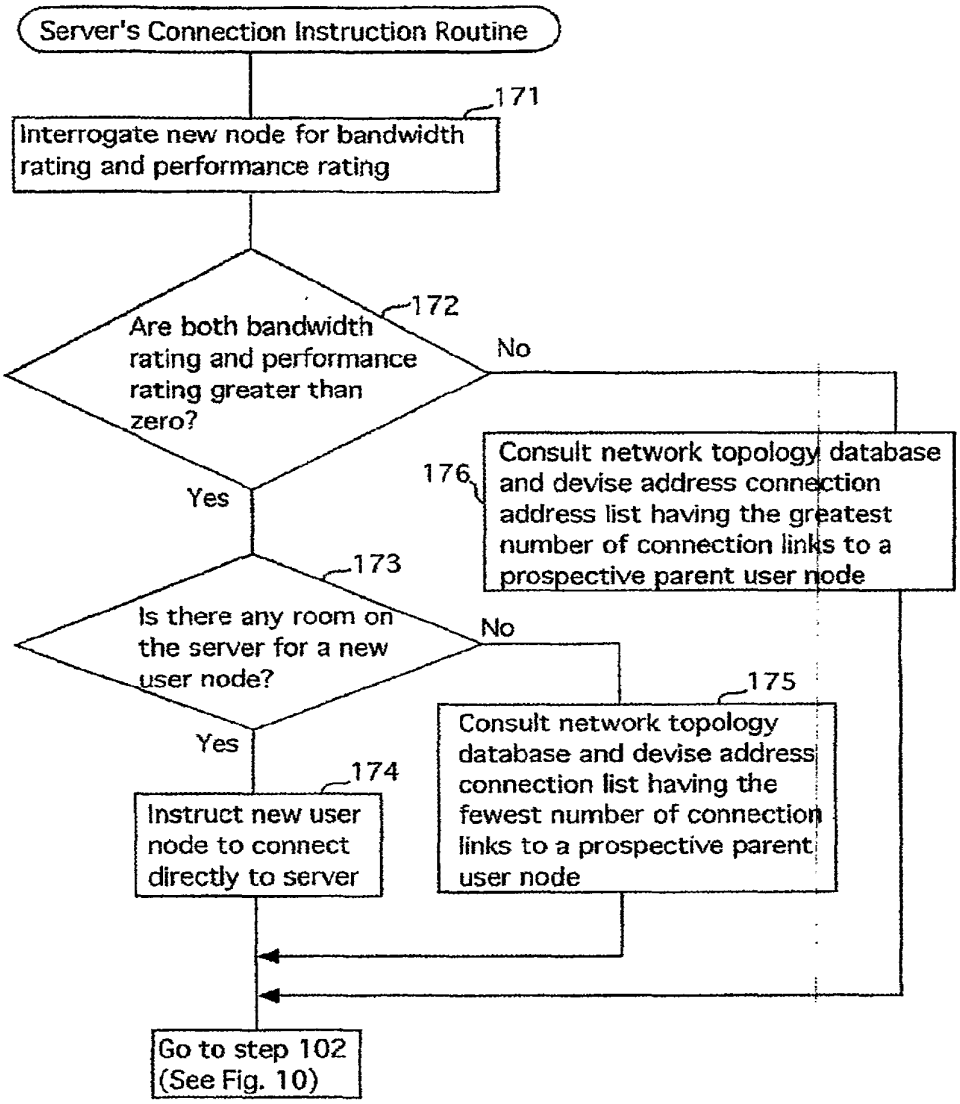


Fig. 17

fuboTV Media Inc.
Exhibit 1002
Page 241

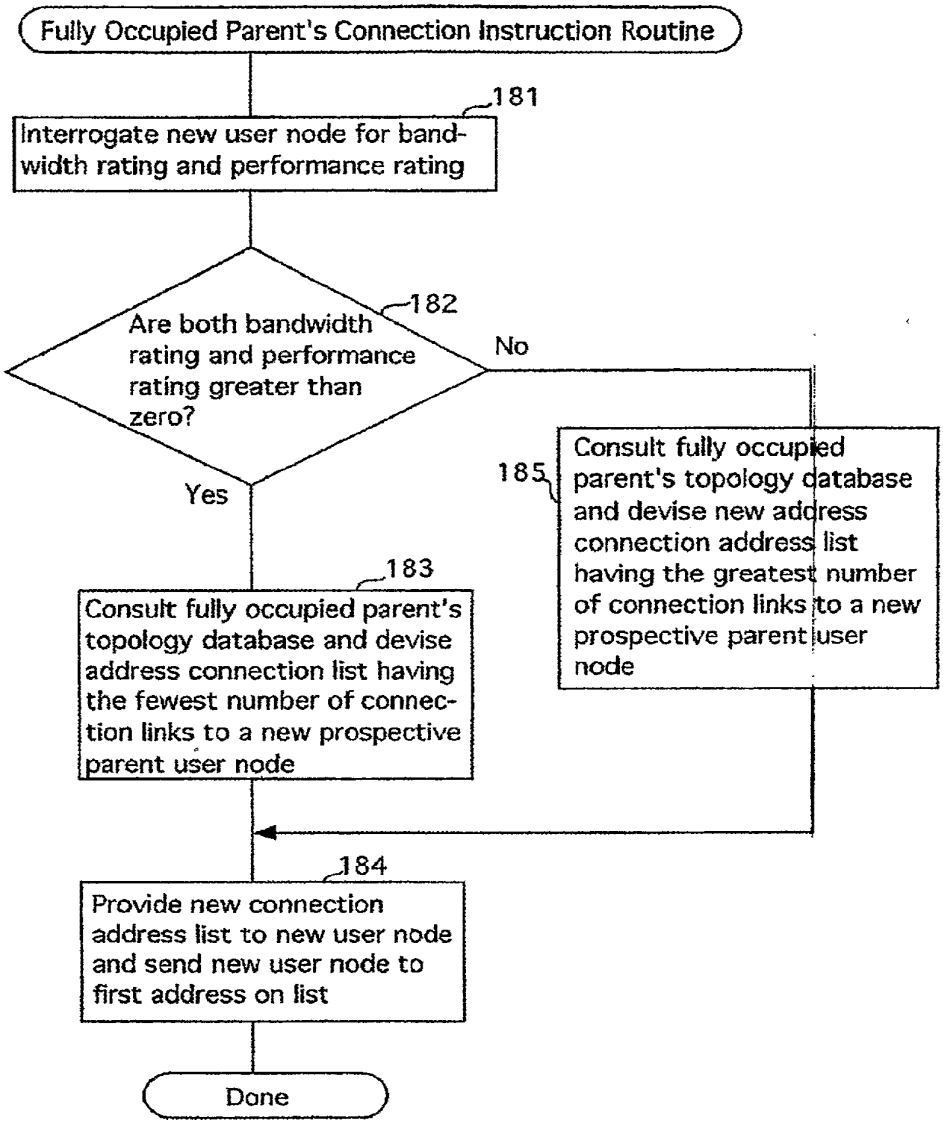


Fig. 18

fuboTV Media Inc.
Exhibit 1002
Page 242

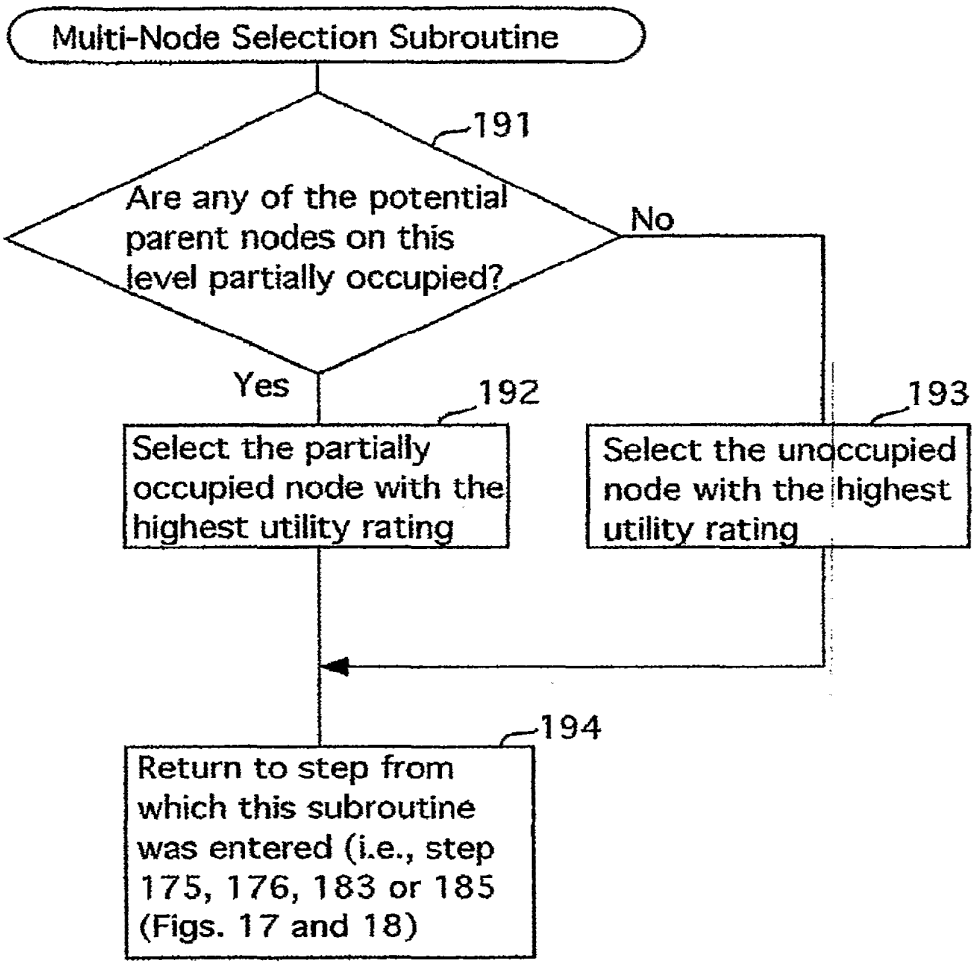


Fig. 19

fuboTV Media Inc.
Exhibit 1002
Page 243

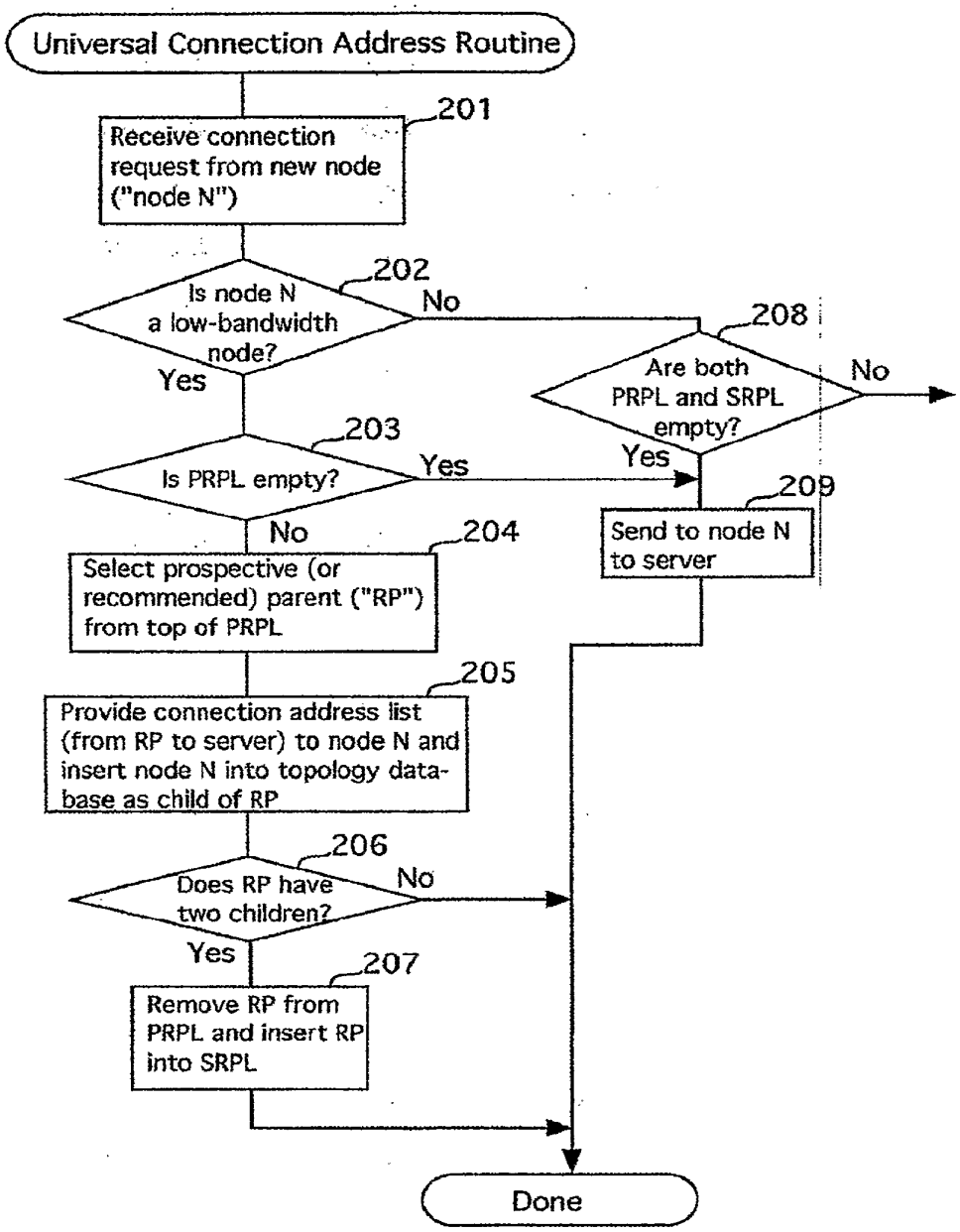


Fig. 20A

fuboTV Media Inc.
Exhibit 1002
Page 244

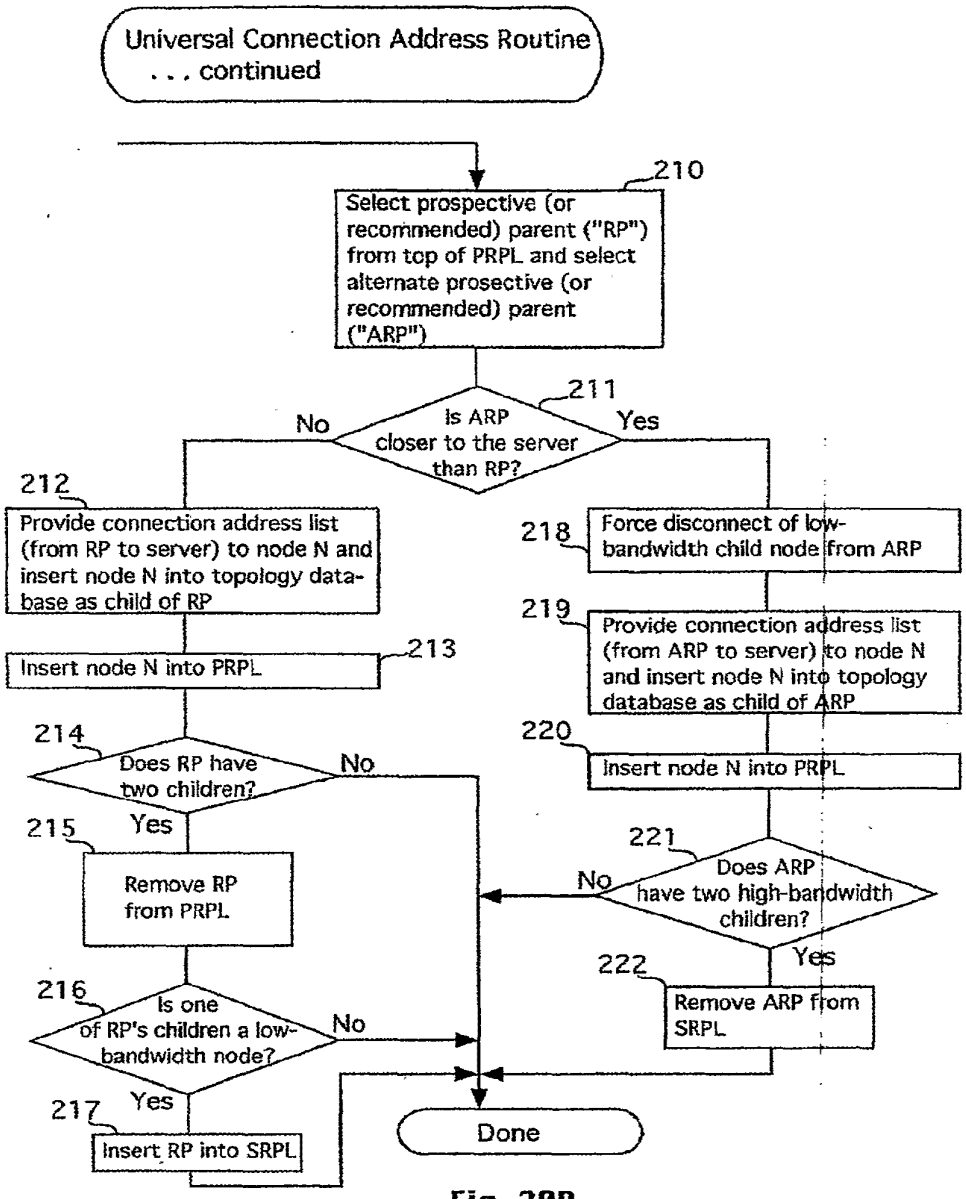


Fig. 20B

fuboTV Media Inc.
Exhibit 1002
Page 245

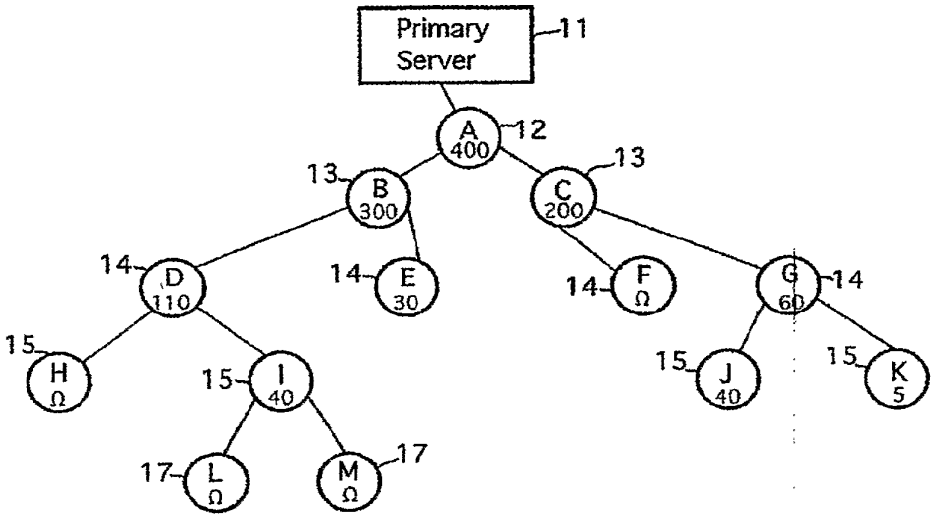


Fig. 21

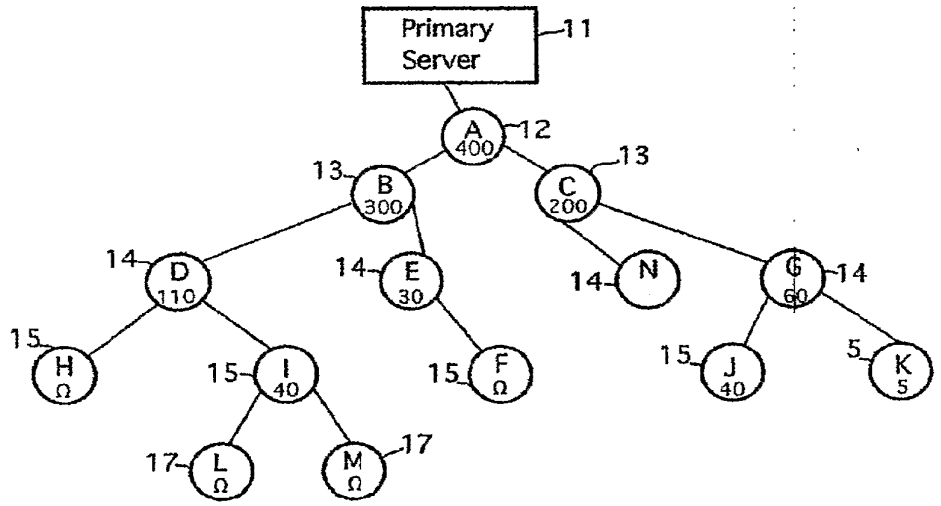


Fig. 22

fuboTV Media Inc.
Exhibit 1002
Page 246

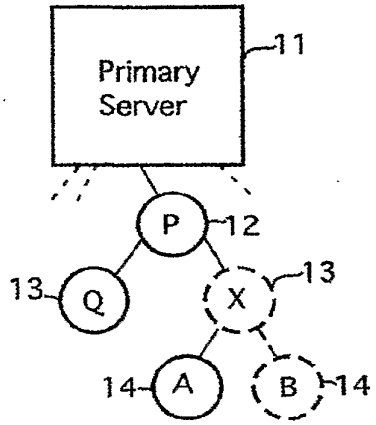


Fig. 23

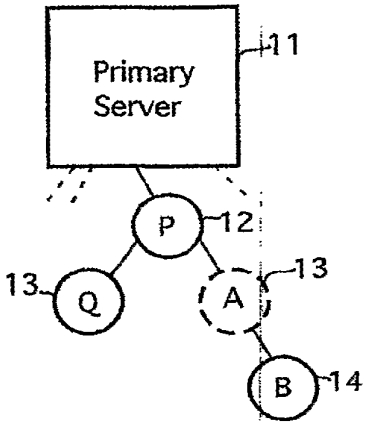


Fig. 24

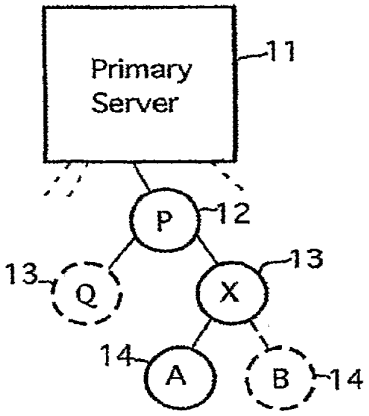


Fig. 25

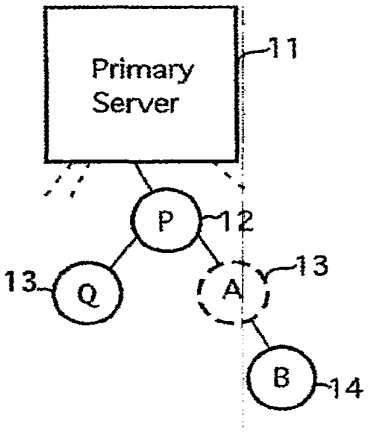


Fig. 26

fuboTV Media Inc.
Exhibit 1002
Page 247

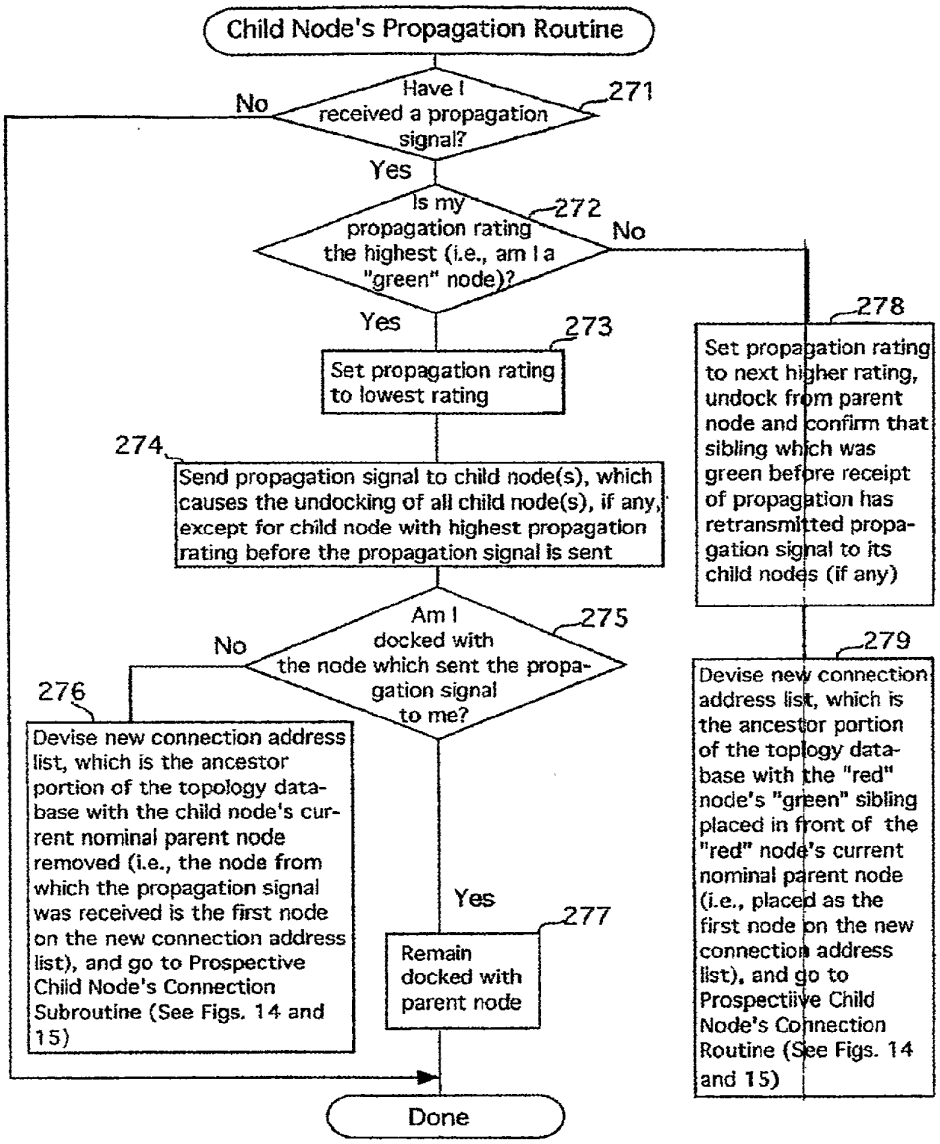


Fig. 27

fuboTV Media Inc.
Exhibit 1002
Page 248

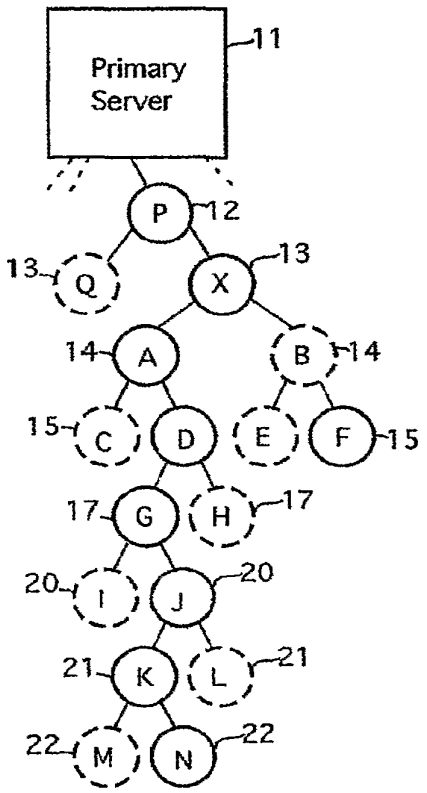


Fig. 28

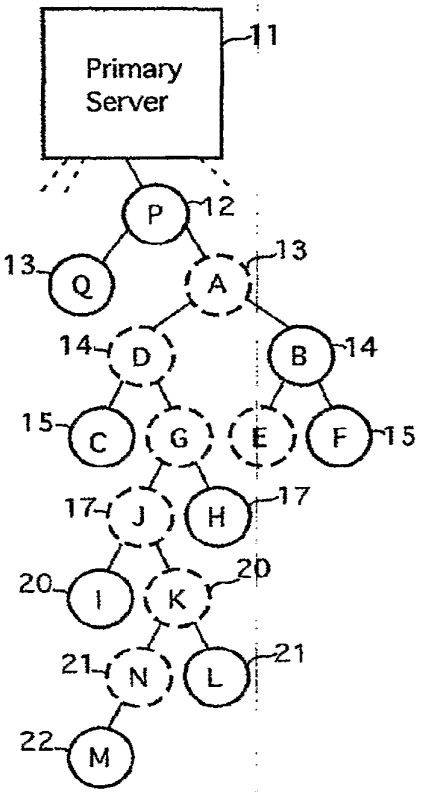


Fig. 29

fuboTV Media Inc.
Exhibit 1002
Page 249

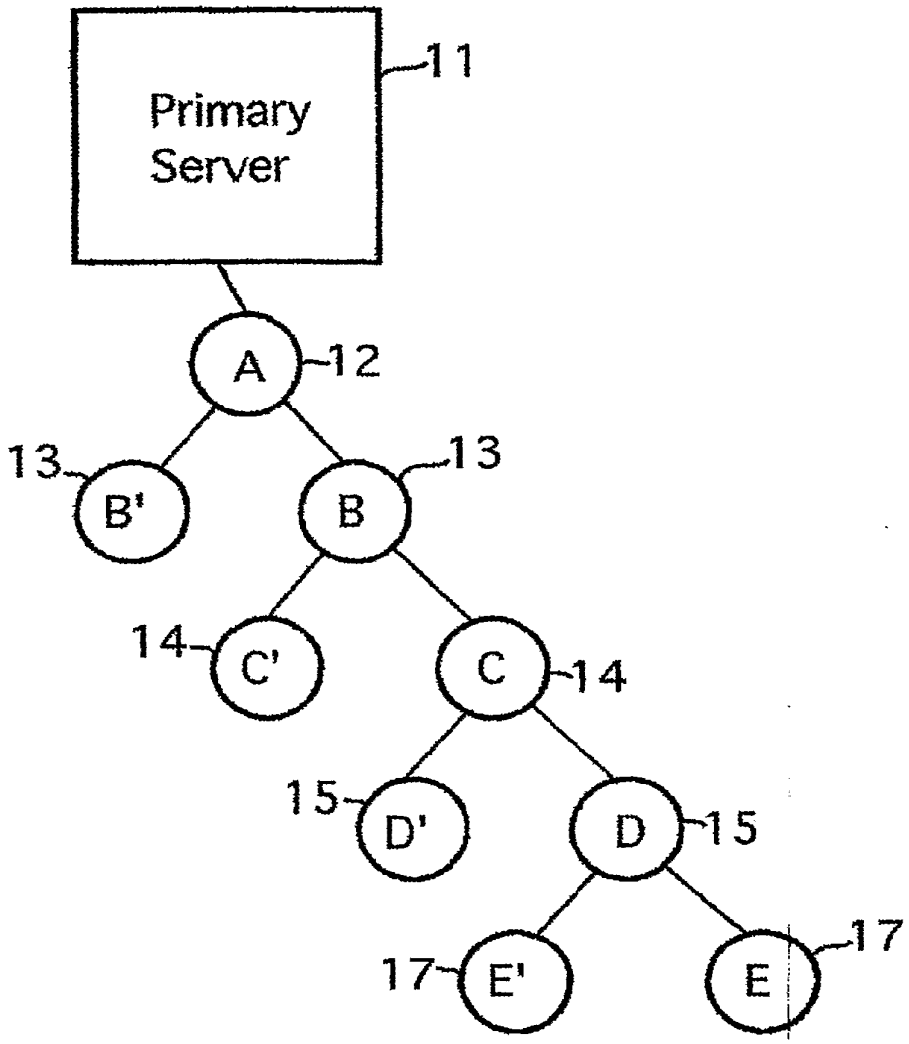


Fig. 30

fuboTV Media Inc.
Exhibit 1002
Page 250

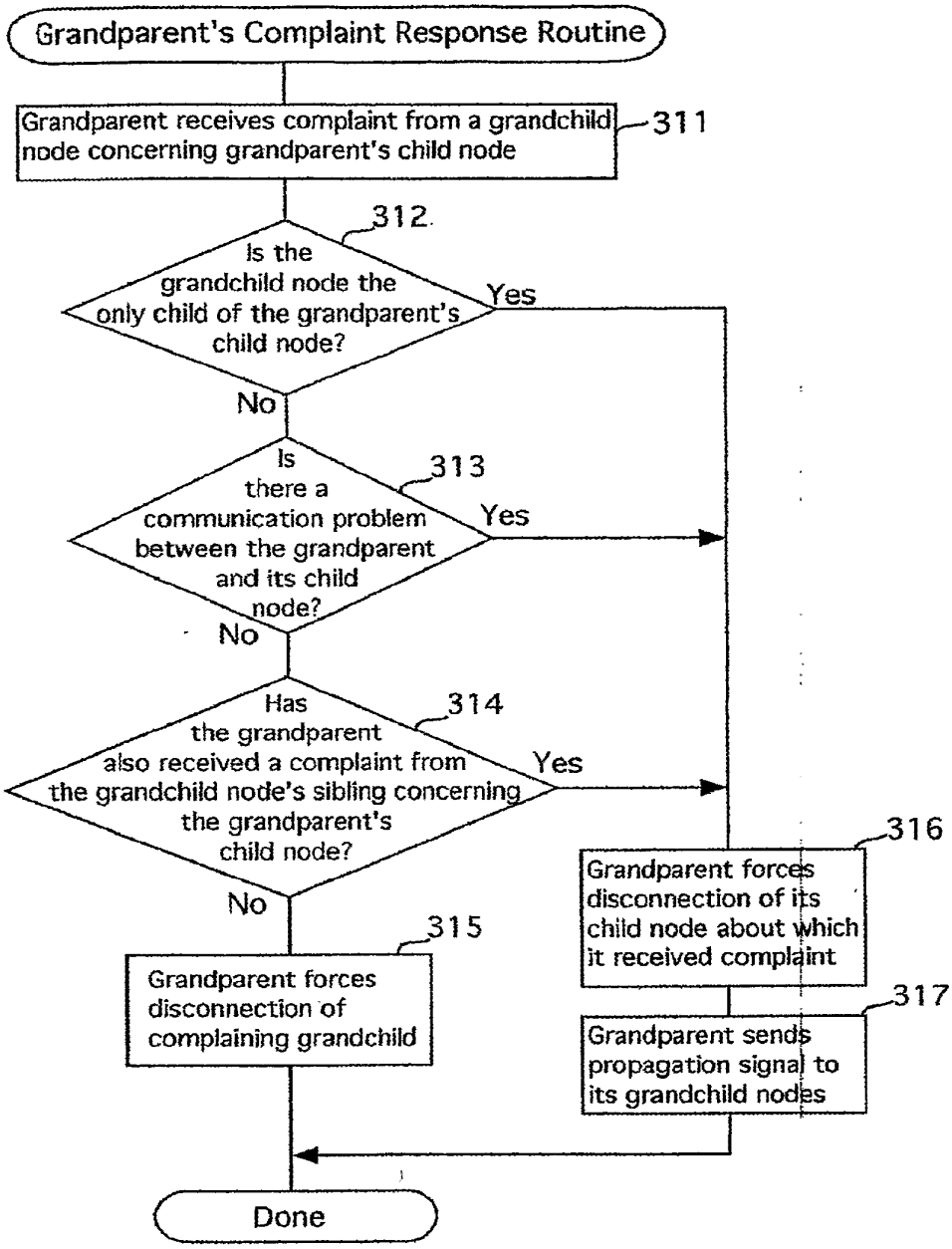


Fig. 31

fuboTV Media Inc.
Exhibit 1002
Page 251

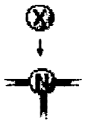


Fig. 32

Fig. 33A

Fig. 33B

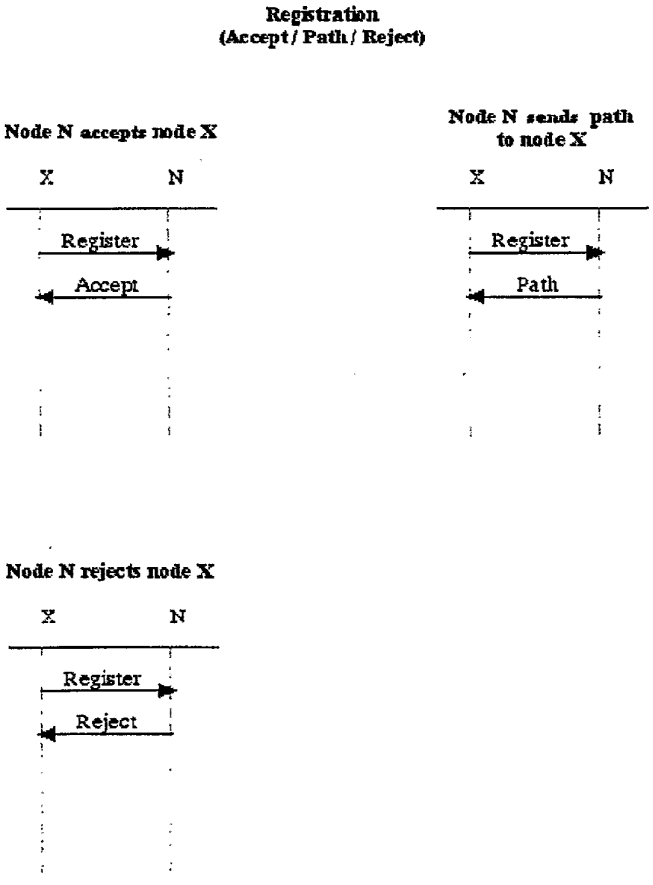


Fig. 33C

fuboTV Media Inc.
Exhibit 1002
Page 252

Fig. 34A Fig. 34B

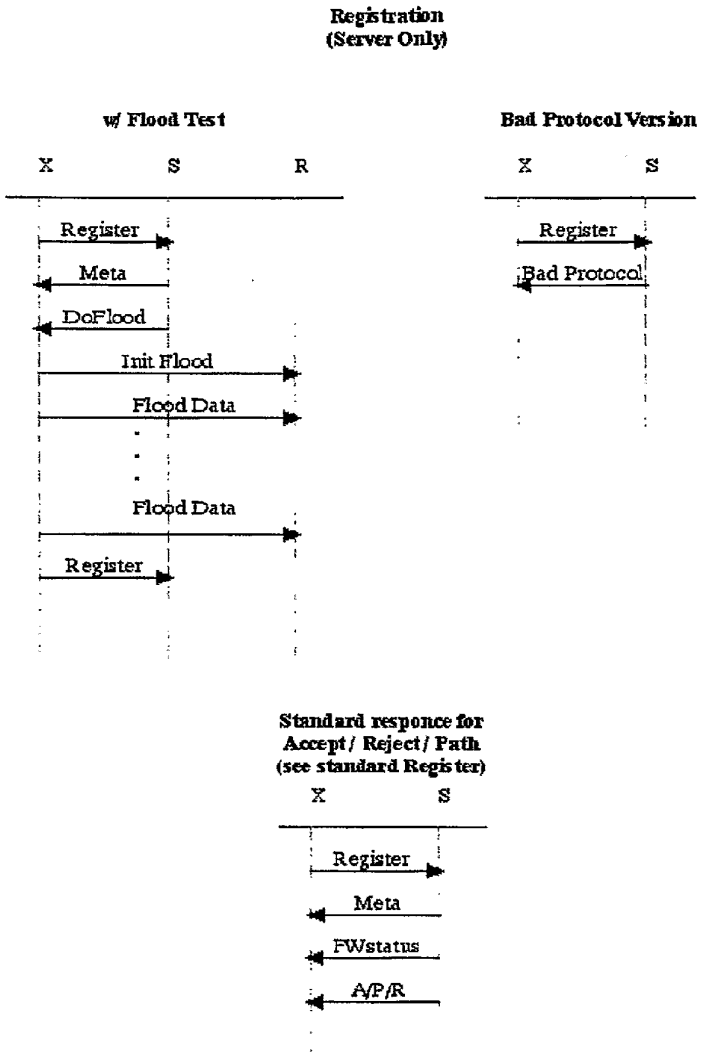


Fig. 34C

fuboTV Media Inc.
Exhibit 1002
Page 253

Fig. 35A

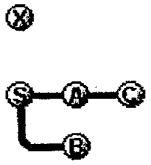
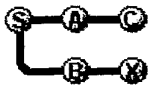


Fig. 35B



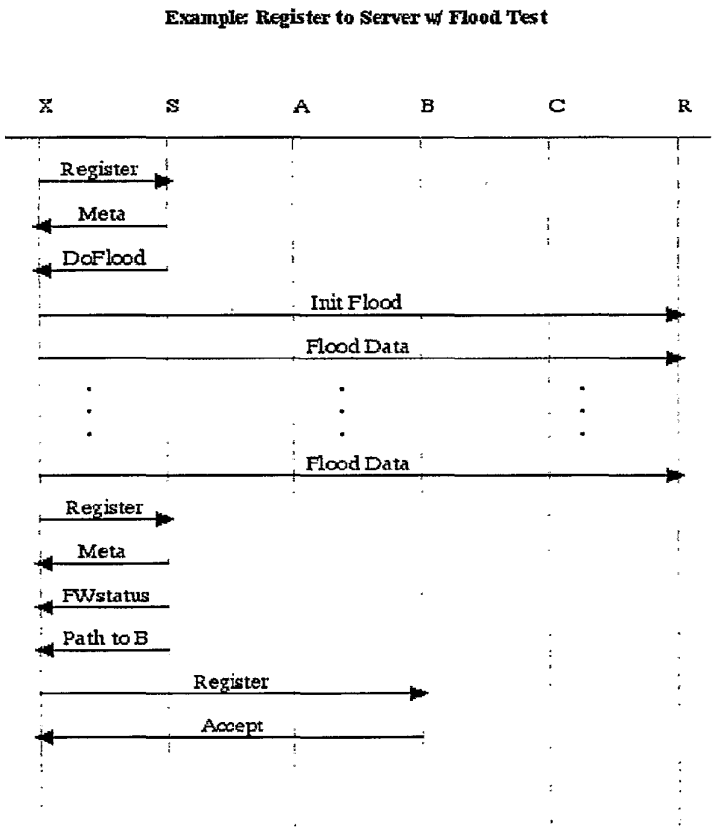


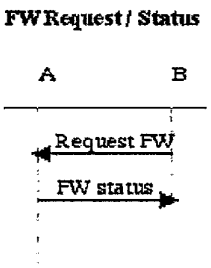
Fig. 36

fuboTV Media Inc.
Exhibit 1002
Page 254

Fig. 37



Fig. 38



fuboTV Media Inc.
Exhibit 1002
Page 255

Fig. 39A



Fig. 39B

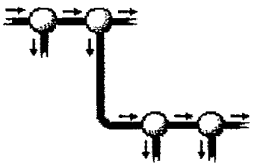
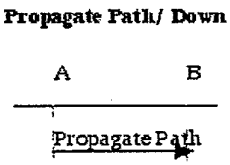


Fig. 40



fuboTV Media Inc.
Exhibit 1002
Page 256

28/31

Fig. 41A



Fig. 41B

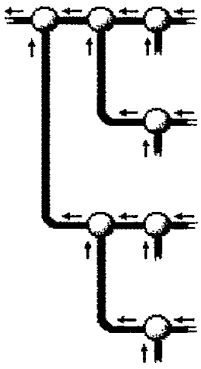
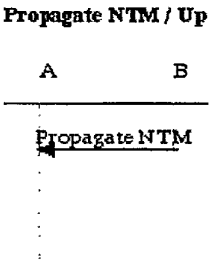


Fig. 42



fuboTV Media Inc.
Exhibit 1002
Page 257

Fig. 43

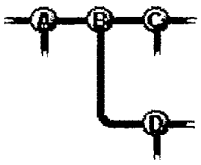
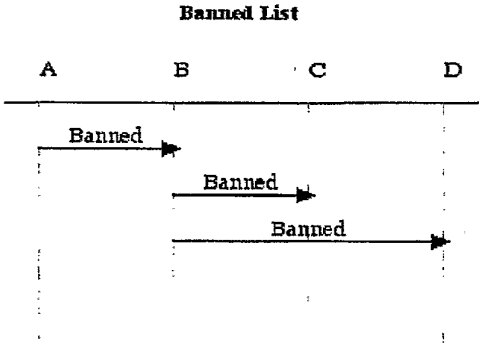
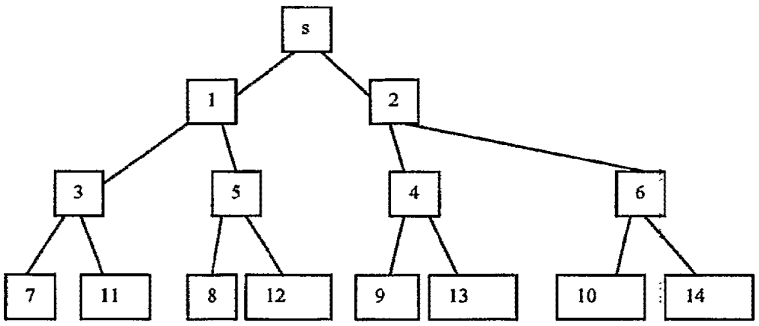


Fig. 44



fuboTV Media Inc.
Exhibit 1002
Page 258

Fig. 43



fuboTV Media Inc.
Exhibit 1002
Page 259

Fig. 46A

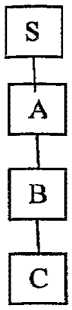


Fig. 46B

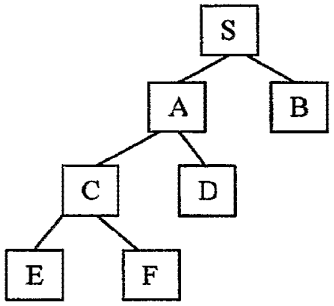
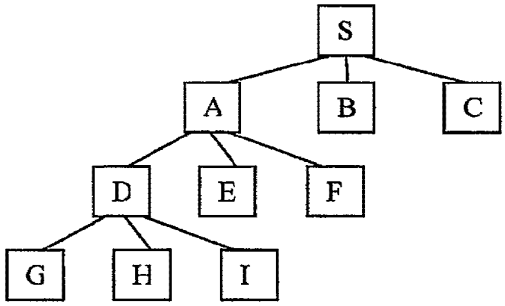


Fig. 46C



fuboTV Media Inc.
Exhibit 1002
Page 260

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 670 256 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.06.2006 Bulletin 2006/24

(51) Int Cl.:
H04N 7/26 *(2006.01)*

(21) Application number: **05111156.5**

(22) Date of filing: **23.11.2005**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IS IT LI LT LU LV MC NL PL PT RO SE SI SK TR
Designated Extension States:
AL BA HR MK YU

(30) Priority: **10.12.2004 US 10114**
10.12.2004 US 10113
10.12.2004 US 10040

(71) Applicant: **Microsoft Corporation**
Redmond WA 98052 (US)

(72) Inventors:
- **Klemets, Anders E.,**
  c/o Microsoft Corporation
  Redmond, WA 98052 (US)
- **Huang, Cheng,**
  c/o Microsoft Corporation
  Redmond, WA 98052 (US)
- **Chou, Philip A.,**
  c/o Microsoft Corporation
  Redmond, WA 98052 (US)

(74) Representative: **Grünecker, Kinkeldey,**
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München (DE)

(54) A system and process for controlling the coding bit rate of streaming media data

(57) A system and process for controlling the coding bit rate of streaming media data is presented. This coding bit rate control involves dynamically adjusting the coding bit rate to control client buffer duration to prevent the buffer from underflowing, while keeping the average coding bit rate close to the average transmission bit rate of the network (an thus maximizing the quality of the data playback). Using the theory of optimal linear quadratic control, the client buffer duration is kept as close as possible to a target level while still keeping the coding bit rate (and hence the quality) as constant as possible. In addition, a leaky bucket model is incorporated into the control loop so that the changes in buffer duration due to natural variation in the instantaneous coding bit rate are not mistaken for changes in buffer duration due to network congestion.

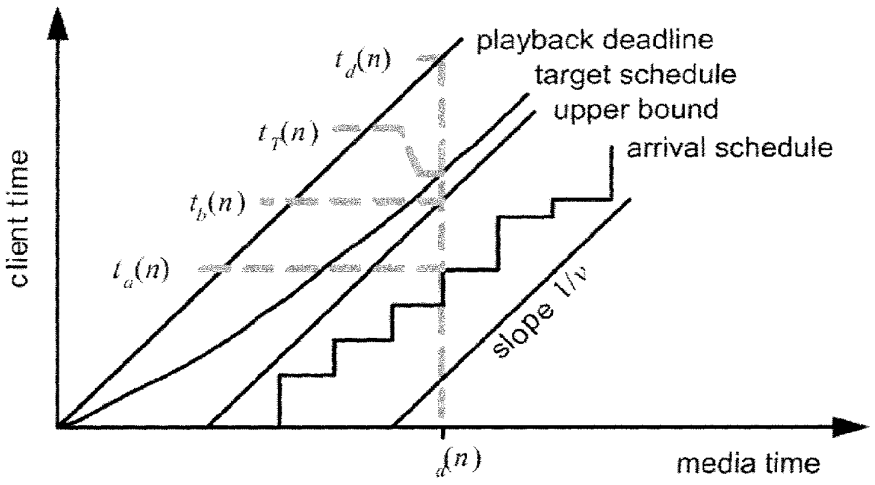


FIG. 5

EP 1 670 256 A2

Printed by Jouve, 75001 PARIS (FR)

fuboTV Media Inc.
Exhibit 1002
Page 261

Description

BACKGROUND

Technical Field:

[0001] The invention is related to controlling the coding bit rate of streaming media, and more particularly to a system and process for controlling the coding bit rate of streaming media data that provides fast startup, continuous playback, and maximal quality and smoothness over the entire streaming session.

Background Art:

[0002] Perhaps the major technical problem in streaming media on demand over the Internet is the need to adapt to changing network conditions. As competing communication processes begin and end, the available bandwidth, packet loss and packet delay all fluctuate. Network outages lasting many seconds can and do occur. Resource reservation and quality of service support can help, but even they cannot guarantee that network resources will be stable. If the network path contains a wireless link, for example, its capacity may be occasionally reduced by interference. Thus it is necessary for commercial-grade streaming media systems to be robust to hostile network conditions. Moreover, such robustness cannot be achieved solely by aggressive (nonreactive) transmission. Even constant bit rate transmission with re-transmissions for every packet loss cannot achieve a throughput higher than the channel capacity. Some degree of adaptivity to the network is therefore required.

[0003] End users expect that a good streaming media system will exhibit the following behavior: content played back on demand will start with low delay; once started, it will play back continuously (without stalling) unless interrupted by the user; and it will play back with the highest possible quality given the average communication bandwidth available. To meet these expectations in the face of changing network conditions, buffering of the content at the client before decoding and playback is required.

[0004] Buffering at the client serves several distinct but simultaneous purposes. First, it allows the client to compensate for short-term variations in packet transmission delay (i.e., "jitter"). Second, it gives the client time to perform packet loss recovery if needed. Third, it allows the client to continue playing back the content during lapses in network bandwidth. And finally, it allows the content to be coded with variable bit rate, which can dramatically improve overall quality. Note that even so-called constant bit rate (CBR) coded content is actually coded with variable bit rate within the constraints of a decoding buffer of a given size. The larger the decoding buffer size, the better the quality. The required decoding buffering is part of the larger client buffer.

[0005] The size of the client buffer can be expressed as the number of seconds of content in the buffer, called the buffer *duration.* The buffer duration tends to increase as content enters the buffer and tends to decrease as content leaves the buffer. Content leaves the buffer when it is played out, at a rate of v seconds of content per second of real time, where v is the *playback speed* (typically 1 for normal playback, but possibly more than 1 for high speed playback or less than 1 for low speed playback). Content enters the buffer when it arrives at the client over the network, at a rate of r_a/r_c seconds of content per second of real time, where r_a is the *arrival rate,* or average number of bits that arrive at the client per second of real time, and r_c is the *coding bit rate,* or the average number of bits needed to encode one second of content. Thus the buffer duration can be increased by increasing r_a, decreasing r_c, and/or decreasing v (and vice versa for decreasing the buffer duration). Although the buffer duration can be momentarily controlled by changing r_a or changing v, these quantities are generally not possible to control freely for long periods of time. The arrival rate r_a on average is determined by the network capacity, while the playback speed v on average is determined by user preference. Thus if the network capacity drops dramatically for a sustained period, reducing the coding bit rate r_c is the only appropriate way to prevent a *rebuffering event* in which playback stops (v=0) while the buffer refills.

[0006] Thus, adaptivity to changing network conditions requires not only a buffer, but also some means to adjust the coding bit rate r_c of the content. This can be done by stream switching in combination with multi bit rate (MBR) coding or coarse grained or fine grained scalable coding. Today's commercial streaming media systems [1] rely on MBR coding as well as *thinning,* which is a form of coarse grained scalability. In MBR coding, semantically identical content is encoded into alternative bit streams at different coding bit rates and stored in the same media file at the server, allowing the content to be streamed at different levels of quality corresponding to the coding bit rates r_c, possibly using bit stream switching [2]. In coarse grained scalable coding (such as MPEG-2/4 temporal or SNR scalability) the content is encoded into several sub-streams or *layers,* so that the coding bit rate r_c can be changed in large deltas by adding or dropping (at possibly restricted times) one layer of content at a time. Thinning is a special case of coarse grained scalability in which dependent video frames (P and B frames) are dropped before independent video frames (I frames), which are in turn are dropped before audio frames. Future commercial systems may support fine grained scalability (FGS) as well. Fine grained scalable coding (such as 3D SPIHT [6], MPEG-4 FGS [7], or EAC [8]) allows the coding bit rate r_c to change

fuboTV Media Inc.
Exhibit 1002
Page 262

at any time in deltas sometimes as small as one byte per presentation. FGS coding offers great flexibility in adapting to variable network conditions, and can demonstrably improve quality under such conditions.

[0007] Some examples of existing technology that adjusts the coding bit rate r_c of the content in an attempt to adapt to changing network conditions includes de Cuetos and Ross [9], which decouples the transmission rate and the coding bit rate. They assume that the transmission rate is determined by the network transport protocol (TCP or TFRC). Based on this, they develop a heuristic real time algorithm for adaptive coding bit rate control and compare its performance to an optimal offline coding bit rate control policy if the transmission rate is given prior to streaming. The work of Rejaie, Handley and Estrin [4] proposes a scheme for transmitting layered video in the context of unicast congestion control, which basically includes two mechanisms. One mechanism is a coarse-grained mechanism for adding and dropping layers (changing the overall coding bit rate and quality). The other is a fine-grained interlayer bandwidth allocation mechanism to manage the receiver buffer (not changing the overall coding bit rate or quality). A potential issue with this approach is that it changes the coding bit rate by adding or dropping one (presumably coarse) layer at a time. If the layers are fine-grained, as in the case of FGS coded media, then adding or dropping one (fine-grained) layer at a time typically cannot provide a prompt enough change in coding bit rate. Moreover, since the adding and dropping mechanism is rather empirical, the mechanism may simply not be suitable for FGS media. The work of Q. Zhang, Zhu and Y-Q. Zhang [5] proposes a resource allocation scheme to adapt the coding bit rate to estimated network bandwidth. The novelty of their approach is that they consider minimizing the distortion (or equivalently maximizing the quality) of all applications, such as file-transfers and web browsing in addition to audio/video streaming. However, their optimization process does not include the smoothness of individual streams and might lead to potential quality fluctuations.

[0008] However, even with buffering and the ability to adjust the coding bit rate, existing technologies for streaming media on demand over the Internet suffer from two problems:

1. Playback often stalls during network congestion. That is, during playback of high bit rate content, if the network bit rate drops below the content bit rate, the client buffer runs out of content and playback stops while the client rebuffers (known as a "rebuffering" event).
2. Start-up delay is often too long (about 5 seconds).

There are existing solutions to both of these problems, but they do not always work well. One solution to the first problem is to stream the content encoded at a coding bit rate that is low relative to the average bit rate transmitted over the network (the *transmission bit rate*). This will enable the buffer to build up over time. With such a large reserve of unplayed information on the client, temporary network congestion will not affect playback. However, this solution has two problems. First, the coding bit rate of the content is not as high as the average transmission bit rate of the network and hence the quality is lower than it could be. Second, the buffer can grow nearly as large as the streamed file itself. This may demand too many resources on the client device.

[0009] Another solution to the first problem is to try to maintain the client buffer at a constant level (typically about 10 seconds), while switching between different coding bit rates for the same content, trying to match the transmission bit rate of the network. However, rebuffering events are still commonly observed in practice, because choosing the right time to switch streams is difficult. One reason that it is difficult is that there are natural variations in the instantaneous coding bit rate of the content, even in so-called constant bit rate encodings, which can confuse the client buffer management algorithm.

[0010] The second problem above (long start-up delay) also has multiple solutions. One solution is to fill up the client buffer quickly, with a quick initial transmission rate burst. With the client buffer full, playback can safely begin. However, this solution has several problems. First, it is only applicable when there is sufficient "headroom" in the network to increase the transmission bit rate for a few seconds. Thus it is usually not applicable for modem connections, for example. Second, it stresses the network, causing other applications in the network to back off. It has been shown that during the burst period, there can be as much as 80% packet loss, causing all TCP connections sharing the same bottleneck to back off. Third, by implication, if there is headroom in the network for bursting, then the streaming application may not be using the full bandwidth available to it during the remainder of the file, meaning that quality is lower than it should be.

[0011] Another solution to the second problem is to play back the content slower than real time, allowing playback to begin while the client buffer builds up. This is an innovative solution, but has the obvious temporal distortion.

[0012] A final solution to the second problem is to lower temporarily the coding bit rate of the content below the transmission bit rate of the network, allowing playback to begin while the client buffer builds up. This is a solution proposed by Chou et al. in [13].

[0013] The system and process of the present invention resolve the problems of the existing techniques and provide fast startup, continuous playback, and maximal quality and smoothness over the entire streaming session.

[0014] It is noted that in the preceding paragraphs, as well as in the remainder of this specification, the description refers to various individual publications identified by a numeric designator contained within a pair of brackets. For example, such a reference may be identified by reciting, "reference [1]" or simply "[1]". A listing of references including

fuboTV Media Inc.
Exhibit 1002
Page 263

the publications corresponding to each designator can be found at the end of the Detailed Description section.

SUMMARY

[0015] The present invention is directed toward a system and process for controlling a coding bit rate of streaming media data being transmitted to a client from a server over a computer network. In general, this coding bit rate control involves dynamically adjusting the coding bit rate of the streaming media data to control the client buffer duration. The purpose of this is to prevent the client buffer from underflowing, while keeping the average coding bit rate close to the average transmission bit rate of the network (an thus maximizing the quality of the playback of the data). The problem of coding bit rate control is formulated as a standard problem in linear quadratic optimal control, in which the client buffer duration is controlled as closely as possible to a target level. The smoothness of the average coding bit rate over consecutive frames is also considered when deciding whether to change the coding bit rate as part of the optimal control process. This yields a higher and more stable quality as network conditions change. In addition, the natural variation in the instantaneous coding bit rate that occurs for a given average coding bit rate is explicitly take into consideration. This is accomplished by incorporating the leaky bucket model into the control loop so that the changes in buffer duration due to natural variation in the instantaneous coding bit rate are not mistaken for changes in buffer duration due to network congestion. It is noted that in the present system and process, it is not the actual fullness of the client buffer that is controlled, but an upper bound on the time of arrival of bits into the client buffer, to a target level. The upper bound is based on the leaky bucket model of the coding bit rate.

[0016] Preventing the client buffer from underflowing, while keeping the average coding bit rate close to the average transmission bit rate of the network, meets two of the aforementioned user expectations. Namely, if the buffer never underflows, it allows for continuous playback. In addition, keeping the average coding bit rate close to the average transmission bit rate of the network means the data is played back with the highest possible quality given the average communication bandwidth available. This leaves the remaining issue of startup delay. This issue is resolved in the present system and process by controlling the size of the client buffer over time. More particularly, the aforementioned client buffer target level is adjusted to start small, and then grow slowly over time. If the buffer is initially small, it allows for shorter startup delay. In addition, as the buffer is eventually allowed to grow large, it enhances the robustness of the system as well as creating high, nearly constant quality. Thus, client buffer management is a key element affecting the performance of streaming media systems.

[0017] More particularly, the present system and process involves a server generating a streaming media data stream that exhibits one of a number of coding bit rates supported by the server. Initially, the server chooses the coding bit rate during a startup period. However, after the startup period, the client provides coding bit rate requests to the server. In response, the server transmits the streaming media data at the most appropriate supported coding bit rate closest to the rate requested by the client. The coding bit rates requested by the client are those estimated to provide a high quality playback of the streaming media data while still keeping a decoder buffer of the client used to receive streaming media data from the server filled to a desired duration level.

[0018] The client computes the coding bit rate that will provide the desired results on an ongoing basis using the aforementioned linear quadratic optimal control technique. This coding bit rate computation involves determining on a frame-by-frame basis a coding bit rate that reduces a difference between an estimated latest anticipated arrival time of the frame under consideration and a prescribed target arrival time, while at the same time reducing the change in the coding bit rate to a prescribed degree. In one embodiment involving the aforementioned leaky bucket model, the coding bit rate computations are based on parameters indicative of the state of the encoder buffer of the server. These parameters are computed by the server and provided to the client along with the streaming media data.

[0019] More particularly, the server first computes a set of parameters defining an initial state of the encoder buffer as it would exist if a data stream corresponding to a supported coding bit rate where streamed therefrom. A separate set of parameters is computed for each of the supported coding bit rates, and are provided to the client in a preamble to the streaming media data. These parameters include the coding bit rate associated with the data stream, the size of the encoder buffer employed with the coding bit rate of the data stream, and a value indicative of the initial encoder buffer fullness exhibited at the coding bit rate of the data stream. It is noted that the size of the encoder buffer employed with each supported coding bit rate varies and is chosen so as to be the minimum size buffer that will still contain the data stream at any point in the streaming process given the coding bit rate and the initial encoder buffer fullness.

[0020] In addition to the initial encoder buffer parameters, the server also computes an upper bound gap for each frame of the streaming media data generated for each coding bit rate supported by the server. This upper bound gap is defined as the number of data bits that the server's encoder buffer can contain over the bits currently contained therein after a just-generated frame is fully input into the buffer. More particularly, the upper bound gap for each frame is computed as the difference between the encoder buffer size and the last computed value for an encoder buffer fullness after insertion of the just-generated frame. The encoder buffer fullness value after insertion of the just-generated frame is computed as the sum of the last computed value of the encoder buffer fullness value prior to insertion of the just-

fuboTV Media Inc.
Exhibit 1002
Page 264

generated frame and the size of that just-generated frame. The encoder buffer fullness value prior to insertion of the just-generated frame is computed as either zero, or the difference between the value computed for the encoder buffer fullness after the insertion of the frame generated immediately before the just-generated frame and the coding bit rate associated with that prior frame divided by the instantaneous frame rate associated with said prior frame, whichever is larger. The instantaneous frame rate is equal to the reciprocal of, the time the next frame is scheduled to be encoded less the time the just-generated frame was encoded. In one embodiment, the server provides the upper bound gap computed for the first frame of a sequence of frames generated after a change in the coding rate to the client along with an indication of the coding bit rate associated with the sequence of frames as part of the data associated with the first frame of the sequence. In another embodiment, the server provides the gap value with every frame generated.

[0021] The client employs the encoder buffer parameters and the upper bound gap values to determine on a frame-by-frame basis the coding bit rate that will reduce a difference between the estimated latest anticipated arrival time of a frame under consideration and a prescribed target arrival time of that frame, in order to keep the client buffer at approximately the desired duration level. In cases where the upper bound gap is only provided with the first frame after a change in the coding bit rate, the client estimates the upper bound gap for the frames that do not include a gap value. It is also noted that given the aforementioned initial encoder buffer conditions, it is possible for the client to estimate the upper bound gap values for every frame received. Thus, an alternate embodiment where the server is not configured to compute and provide the gap values on an ongoing basis, the client computer can still compute them on its own.

[0022] When a frame with a new average coding bit rate arrives at the client there is a shift in the upper bound gap. This shift can be on the order of seconds and hence, rather than being negligible, can be confusing to the controller. One solution to this is to introduce a simultaneous shift in the control target schedule. To this end, the server also computes a shift value for each frame representing the first frame after a coding bit rate change. This shift value represents the difference between the upper bound gap that would be associated with the frame generated immediately before this first frame had it been encoded at the new coding bit rate and the upper bound gap actually associated with the frame as encoded at the previous coding bit rate. The shift value is provided to the client along with the first frame after a change in the coding bit rate. The client shifts the currently scheduled target arrival time associated with the just received "first" frame by the shift value provided, and the currently scheduled target arrival times for future frames are shifted such that they collectively approach, over a period of time, and eventually coincide with, the previous target arrival times for those frames. In this way the adverse effects of an upper bound gap shift when the coding bit rate is changed are mitigated and the target arrival time values are eventually brought back in line.

[0023] In regard to the target arrival time for each frame, these are chosen so as to make the amount of time between the target time of a frame and its playback time large enough, after the startup period, that network jitter, delays and throughput changes which may cause the actual arrival time of the frame to be after its target arrival time do not result in the frame arriving after its scheduled playback time. During the startup period, the target arrival times are chosen so as to make the target arrival time closer to the playback time to assist in reducing startup delay. In one embodiment this is accomplished by setting the target arrival time for each frame using a logarithmic target schedule. In another embodiment this is accomplished by setting the target arrival time for each frame using a two-piece linear target schedule where the difference between the target arrival time and the playback time for a frame arriving during the startup period increases linearly to a prescribed amount of time that is large enough to account for the network jitter, delays and throughput changes and after which the difference remains substantially constant.

[0024] It is noted that the startup delay can be further minimized by beginning playback of the first frame of the streaming media data when the client's decoder buffer has the minimum amount of data therein that is needed to ensure an underflow condition does not occur at the initial coding bit rate. Still further, the initial coding bit rate can be set to a level that is less than the anticipated arrival rate of the streaming media data at the client since the minimum amount of data needed to ensure an underflow condition does not occur, decreases as the arrival rate increases relative to the coding bit rate. Thus, in the startup period before the client request changes in the coding bit rate, the server provides the streaming media data at the initial coding bit rate. In one embodiment, the initial coding bit rate is set to approximately one-half the anticipated arrival rate, where the anticipated arrival rate is estimated to be the anticipated transmission rate from the server at the current bandwidth available on the network. It is noted that the startup period can be defined as the period of time prior to the first instance of an estimated latest arrival time of a frame under consideration computed for the initial coding bit rate being earlier than a target arrival time for that frame.

[0025] In regard to the client identifying the coding bit rate for each frame that also reduces the change in the coding bit rate to a prescribed degree, it is noted that the prescribed degree can vary depending on whether the coding bit rate increased. If it did increase, then the prescribed degree to which any future change in the coding bit rate is minimized is made greater in comparison to the prescribed degree in cases where the last change in the coding bit rate decreased the rate. Still further, it is also desirable to minimize quality variations due to large or frequent changes to the coding bit rate. In order to stabilize the code bit rate changes, the following actions can be taken. First, in the case where the client identifies a new coding bit rate that represents an increase, a new coding bit rate would be requested from the server only if the new rate does not exceed the current moving average arrival rate of the frames. In another embodiment, the

fuboTV Media Inc.
Exhibit 1002
Page 265

new coding bit rate representing an increase would only be requested from the server, even if it exceeds the current moving average arrival rate of the frames, if the current client buffer duration exceeds the desired duration level by an amount that it is estimated will not be expended at the higher coding bit rate prior to the passing of a prescribed period of time (e.g., 60 seconds). However, in a case where the client identifies a new coding bit rate that represents a decrease, a new coding bit rate would be requested from the server even if it represents a significant departure from the immediately previous coding bit rate. This is because there is little risk in the client buffer underflowing when the coding rate is decreased.

[0026] In regard to the aforementioned moving average arrival rate of the frames, a new procedure for computing this rate is employed with the present system and process. More particularly, the moving average arrival rate is computed on a packet-by-packet basis, by calculating the product of the moving average arrival rate computed for the immediately preceding packet to a currently received packet and a fractional weighting factor, added to the product of the instantaneous arrival rate of the currently received packet and one minus the fractional weighting factor. In this computation, the fractional weighting factor is not a constant as in the past, but instead is based on the inter-arrival gaps between received packets. More particularly, the fractional weighting factor $\beta(k)$ for a currently received packet k is computed as $\frac{e^{-\alpha[t(k)-t(k-1)]}-e^{-\alpha[t(k)-t(0)]}}{1-e^{-\alpha[t(k)-t(0)]}}$ where α is the reciprocal of a prescribed time constant, $t(k)$ is the actual arrival time of the current packet, $t(k\text{-}1)$ is the actual arrival time of the packet received immediately prior to the current packet, and $t(0)$ is the arrival time of the first packet of the streaming media data. In this case, the instantaneous arrival rate $r_a(k)$ of the current packet k is computed as $\frac{b(k)}{t_a(k)-t_a(k-1)}$, where $b(k)$ is the size of the current packet. If $b(k)$ is expressed in bits and $t_a(k) - t_a(k - 1)$ is expressed in seconds, then the moving average arrival rate represents the arrival rate of the streaming media data bit bits per second. It is noted that a similar procedure can be employed for any arbitrary units of rate of the streaming media.

[0027] As mentioned previously, the server generates data streams that exhibit one of a number of coding bit rates supported by the server, and transmits data streams at the most appropriate supported coding bit rate closest to the rate requested by the client. The number of available coding bit rates will depend on how the data is encoded. For example, if a fine grain scalable coding scheme is used theoretically there could be a large number of rates available (although for practical reasons the number is more likely to be lower, e.g., 50). However, if a coarse grain scalable coding scheme or a multiple bit rate coding scheme is employed, there could be a more limited number of coding bit rates available from the server. Thus, in some cases an optimum coding bit rate identified by the client may not be available from the server. In addition, even if there is a matching coding bit rate available, the upper bound gap may be such that switching to that rate would risk a client buffer underflow. Given this, the present system and process includes techniques for determining the most appropriate coding bit rate, from the available rates, in view of the optimum coding bit rate identified by the client. It is noted that the available rates could be provided to the client from the server in a preamble to the streaming media data. In such a case, the client itself would perform the analysis and request the resulting supported coding bit rate from the server. However, in an alternate embodiment, the client would request the optimum coding bit rate it identified, and the server would perform the analysis to determine which supported rate is most appropriate. The server would then provide a data stream exhibiting the selected coding bit rate. In either case, the analysis involves finding a supported coding bit rate that is equal to, or if none are equal, the closest smaller rate to, the optimum coding bit rate identified by the client. Whenever the supported coding bit rate found is lower than the coding bit rate associated with the last generated frame of the streaming media data, all future frames (or those beginning with a frame specified by the client) are generated at that supported rate. However, when the supported coding bit rate found is higher than the coding bit rate associated with the last generated frame, it is determined if a difference between the upper bound gap associated with the last generated frame had it been encoded at the supported coding bit rate found and the upper bound gap associated with that frame encoded at the current coding bit rate, is less than or equal to a maximum allowable difference value. If the difference is less than the maximum allowable difference value, the affected future frames are generated at the supported rate found. But, if the difference is not less than the maximum allowable difference value, the next lower supported coding bit rate is found and the foregoing actions are repeated to ultimately identify the appropriate rate.

[0028] The maximum allowable difference value is computed by the client, and if the server is doing the analysis, the client provides the value along with the request for data at the new coding bit rate. The client chooses the maximum allowable difference value such that the latest anticipated arrival time associated with a frame provided by the server immediately prior to a frame under consideration by the client, had it been coded at the requested coding bit rate, is no more than a prescribed fraction of the way from that frame's target arrival time to its playback deadline (e.g., 1/3 of the way).

[0029] In addition to the just described benefits, other advantages of the present invention will become apparent from

fuboTV Media Inc.
Exhibit 1002
Page 266

the detailed description which follows hereinafter when taken in conjunction with the drawing Fig.s which accompany it.

DESCRIPTION OF THE DRAWINGS

[0030] The specific features, aspects, and advantages of the present invention will become better understood with regard to the following description, appended claims, and accompanying drawings where:

FIG. 1 is a diagram depicting a general purpose computing device constituting an exemplary system for implementing the present invention.

FIG. 2 is block diagram of a simplified streaming media communication pipeline.

FIG. 3 is a graph showing schedules at which bits in the coded bit stream pass points A, B, C and D in the communication pipeline of Fig. 2 in terms of bits vs. media time.

FIG. 4 is a graph showing a buffer tube containing a coding schedule in terms of bits vs. media time.

FIG. 5 is a graph showing an arrival schedule and its upper bound in client time vs. media time, where the upper bound is controlled to a target schedule that is increasingly in advance of the playback deadline to provide greater robustness over time.

FIG. 6 is a graph showing a target arrival schedule design.

FIGS. 7(a)-(b) are graphs showing two different target arrival schedules. Fig. 7(a) shows a logarithmic schedule and Fig. 7(b) shows a two-piece linear schedule.

FIG. 8 is a graph showing buffer tubes for various transmission rates.

FIG. 9 is a graph illustrating exponential averaging.

FIG. 10 is a graph showing buffer tube changes and control target adjustments.

FIGS. 11A-C show a flow chart diagramming an embodiment of the coding bit rate control process according to the present invention.

FIG. 12 is a timeline illustrating the conservative limit used in the conservative coding bit rate up-switching procedure.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0031] In the following description of the preferred embodiments of the present invention, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

I. The Computing Environment

[0032] Before providing a description of the preferred embodiments of the present invention, a brief, general description of a suitable computing environment in which portions of the invention may be implemented will be described. Fig. 1 illustrates an example of a suitable computing system environment 100. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

[0033] The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

fuboTV Media Inc.
Exhibit 1002
Page 267

[0034] The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

[0035] With reference to Fig. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[0036] Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer readable media.

[0037] The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Fig. 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

[0038] The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, Fig. 1 illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[0039] The drives and their associated computer storage media discussed above and illustrated in Fig. 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In Fig. 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 110 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus 121, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the

fuboTV Media Inc.
Exhibit 1002
Page 268

system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 195. A camera 192 (such as a digital/electronic still or video camera, or film/photographic scanner) capable of capturing a sequence of images 193 can also be included as an input device to the personal computer 110. Further, while just one camera is depicted, multiple cameras could be included as input devices to the personal computer 110. The images 193 from the one or more cameras are input into the computer 110 via an appropriate camera interface 194. This interface 194 is connected to the system bus 121, thereby allowing the images to be routed to and stored in the RAM 132, or one of the other data storage devices associated with the computer 110. However, it is noted that image data can be input into the computer 110 from any of the aforementioned computer-readable media as well, without requiring the use of the camera 192.

[0040] The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in Fig. 1. The logical connections depicted in Fig. 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0041] When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Fig. 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0042] The exemplary operating environment having now been discussed, the remaining parts of this description section will be devoted to a description of the program modules embodying the invention.

II. Problem Formulation

A. Temporal Coordinate Systems

[0043] It will pay to distinguish between the temporal coordinate systems, or clocks, used to express time. Herein, *media time* refers to the clock running on the device used to capture and timestamp the original content, while *client time* refers to the clock running on the client used to play back the content. It is assumed that media time is real time (i.e., one second of media time elapses in one second of real time) at the time of media capture, while client time is real time at the time of media playback. The symbol τ is used to express media time and the symbol t to express client time, with subscripts and other arguments to indicate corresponding events. For example, $\tau_d(0),\tau_d(1),\tau_d(2),...$ is used to express the playback deadlines of frames 0,1,2,... in media time, while $t_d(0),t_d(1),t_d(2),..$ is used to express the playback deadlines of frames 0,1,2,... at the client. Content may be played back at a rate v times real time. Thus the conversion from media time to client time can be expressed as

$$t = t_0 + \frac{\tau - \tau_0}{v}, \qquad (1)$$

where t_0 and τ_0 represent the time of a common initial event, such as the playback of frame 0 (or the playback of the first frame after a seek or rebuffering event) in media and client coordinate systems, respectively.

B. Leaky Bucket Model

[0044] For the moment imagine a scenario in which both the encoder 200 and the decoder 202 run in real time over an isochronous communication channel of a network 204. In this case, to match the instantaneous coding bit rate to the instantaneous channel rate, an *encoder buffer* 206 is required between the encoder 200 and the network 204 and a *decoder buffer* 208 is required between the network 204 and the decoder 202 , as illustrated in Fig. 2. A *schedule* is the sequence of times at which successive bits in the coded bit stream pass a given point in the communication pipeline.

fuboTV Media Inc.
Exhibit 1002
Page 269

The graph in Fig. 3 illustrates the schedules of bits passing the points A, B, C, and D in Fig. 2. Schedule A is the schedule at which captured frames are instantaneously encoded and put into the encoder buffer. This schedule is a staircase in which the nth step rises by $b(n)$ bits at time $\tau(n)$, where $\tau(n)$ is the time at which frame n is encoded, and $b(n)$ is the number of bits in the resulting encoding. Schedules B and C are the schedules at which bits respectively enter and leave the communication channel. The slope of these schedules is R bits per second, where R is the communication rate of the channel. Schedule D is the schedule at which frames are removed from the decoder buffer and instantaneously decoded for presentation. Note that Schedule D is simply a shift of Schedule A. Note also that Schedule B is a lower bound to Schedule A, while Schedule C is an upper bound to Schedule D. Indeed, the gap between Schedules A and B represents, at any point in time, the size in bits of the encoder buffer, while the gap between Schedules C and D likewise represents the size of the decoder buffer. The encoder and decoder buffer sizes are complementary. Thus the coding schedule (either A or D) can be contained within a *buffer tube,* as illustrated in the graph of Fig. 4, having slope R , height B , and initial offset F^d from the top of the tube (or equivalently initial offset $F^e = B - F^d$ from the bottom of the tube). It can be seen that $D = F^d/R$ is the *startup delay* between the time that the first bit arrives at the receiver and the first frame is decoded. Thus it is of interest to minimize F^d for a given R.

[0045] A *leaky bucket* is a metaphor for the encoder buffer. The encoder dumps $b(n)$ bits into the leaky bucket at time $\tau(n)$, and the bits leak out at rate R. In general it is possible for the leak rate R to be high enough so that the bucket occasionally empties. Thus the encoder buffer fullness $F^e(n)$ immediately before frame n is added to the bucket and the encoder buffer fullness $B^e(n)$ immediately after frame n is added to the bucket evolve from an initial encoder buffer fullness $F^e(0) = F^e$ according to the dynamical system

$$B^e(n) = F^e(n) + b(n), \tag{2}$$

$$F^e(n+1) = \max\{0, B^e(n) - R/f(n)\}, \tag{3}$$

where

$$f(n) = \frac{1}{\tau(n+1) - \tau(n)} \tag{4}$$

is the instantaneous frame rate, for n=0,1,2,... If R is sufficiently low, then the bucket will never run dry (underflow), but if R is too low the bucket will eventually overflow. The largest R such that the buffer will never run dry is taken to be the average coding bit rate r_c of the bit stream. This is made more precise in the following two paragraphs.

[0046] A leaky bucket with size B, rate R, and initial fullness F^e is said to *contain* a stream having a schedule characterized by the steps $\{(b(n),\tau(n))\}$ if $B^e(n) \le B$ for all n. The minimum bucket size needed to contain the stream given leak rate R and initial fullness F^e is defined as

$$B^e_{\min}(R, F^e) = \min_n B^e(n), \tag{5}$$

while the corresponding initial decoder buffer fullness is defined as

$$F^d_{\min}(R, F^e) = B^e_{\min}(R, F^e) - F^e. \tag{6}$$

The minimum of each of these over F^e is denoted as

fuboTV Media Inc.
Exhibit 1002
Page 270

$$B^e_{\min}(R) = \min_{F^e} B^e_{\min}(R, F^e), \tag{7}$$

$$F^d_{\min}(R) = \min_{F^e} F^d_{\min}(R, F^e). \tag{8}$$

It is shown in [10, Proposition 2] that remarkably; these are each minimized by the same value of F^e, which is hence equal to

$$F^e_{\min}(R) = B^e_{\min}(R) - F^d_{\min}(R). \tag{9}$$

Thus the bit stream with schedule $\{(b(n),\tau(n))\}$, for each bit rate R there is a unique leaky bucket that contains the stream and that has the minimum buffer size B as well as the minimum startup delay $D = F^d/R$. These parameters can be computed with the above equations.

[0047] For sufficiently low leak rates R, the leaky bucket does not underflow, when beginning with initial fullness $F^e = F^e_{\min}(R)$. Tthe maximum such rate R may be used as the average coding bit rate r_c of a bit stream with coding schedule $\{(b(n),\tau(n))\}$.

[0048] Leak rates R greater than r_c can also be employed. It is shown in [10] that both $B^e_{\min}(R)$ and $F^d_{\min}(R)$ are decreasing, piecewise linear, and convex in R. Hence if the transmission rate R is greater than the average coding bit rate r_c, the startup delay $D = F^d_{\min}(R)/R$ can be reduced compared to $D = F^d_{\min}(r_c)/R$. This fact will be used in Section IV-A.

[0049] A leaky bucket with leak rate $R=r_c$, size $B = B^e_{\min}(r_c)$ and initial decoder buffer fullness $F^d = F^d_{\min}(r_c)$ thus corresponds to a straight buffer tube bounding the coding schedule as in Fig. 4. Each stream in the media file has a coding schedule; thus each stream corresponds to a straight buffer tube with slope equal to the average coding bit rate r_c of the stream. The size B of the buffer tube and its offset F^e (or F^d) relative to the coding schedule can be either computed by the above formula for a variable bit rate (VBR) stream (such as a constant-quality sub-stream of a scalable stream), or obtained from the size B and initial state F^e of the actual encoder buffer used to encode the stream if it is a constant bit rate (CBR) stream.

[0050] In the following the gap $g(n)$ at frame n between the buffer tube *upper bound* and the coding schedule is considered, as depicted in Fig. 4. Note that the decoder buffer fullness $F^d(n) = B - F^e(n)$ can also be expressed

$$F^d(n) = b(n) + g(n) = g(n-1) + \frac{r_c(n)}{f(n)}, \tag{10}$$

where $r_c(n)$ is the coding bit rate of the buffer tube, now taking into account that different frames may lie in different buffer tubes with different coding bit rates as coding bit rate control is applied and streams are switched.

C. Rate Control Model

[0051] Assume for the moment that bits arrive at the client at a constant rate r_a. Then frame n (having size $b(n)$) arrives at the client $b(n)/r_a$ seconds after frame n-1. Indeed, the index of a bit is proportional to its arrival time. Dividing the vertical scale of the schedules in Fig. 4 by r_a, the schedules are obtained in terms of client time, rather than bits, as shown in the graph of Fig. 5. The coding schedule divided by r_a becomes the *arrival schedule,* which provides for each n the time $t_a(n)$ of arrival of frame n at the client. The buffer tube upper bound (in bits) divided by r_a becomes the buffer tube upper bound (in time), which provides for each n the time $t_b(n)$ by which frame n is guaranteed to arrive. In the

fuboTV Media Inc.
Exhibit 1002
Page 271

same plot the *playback deadline* is shown, which is the time $t_d(n)$ at which frame n is scheduled to be played (after instantaneous decoding). Thus the gap between a frame's arrival time and its playback deadline is the client buffer duration at the time of the frame arrival. This must be non-negative to allow continuous playback.

[0052] In reality the arrival rate is not constant. If $t_a(n\text{-}1)$ and $t_a(n)$ are the arrival times of frames n and n-1 respectively, then

$$r_a(n) = \frac{b(n)}{t_a(n) - t_a(n-1)} \tag{11}$$

can be defined as the *instantaneous arrival rate* at frame n. In practice the average arrival rate at frame n is estimated by a moving average $\tilde{r}_a(n)$ of previous values of $r_a(n)$, as detailed in Section IV-C. Hence using Eq. (11) the arrival time of frame n can be expressed in terms of the arrival time of frame n-1 as

$$t_a(n) = t_a(n-1) + \frac{b(n)}{r_a(n)} \tag{12}$$

$$= t_a(n-1) + \frac{b(n)}{\tilde{r}_a(n)} + v(n), \tag{13}$$

where the $v(n)$ term is an error term that captures the effect of using the slowly moving average $\tilde{r}_a(n)$. From Eq. (10), however, it can be seen that

$$b(n) = \frac{r_c(n)}{f(n)} + g(n-1) - g(n), \tag{14}$$

whence (substituting Eq. (14) into Eq. (13)) yields

$$t_a(n) = t_a(n-1) + \frac{r_c(n)}{f(n)\tilde{r}_a(n)} + \frac{g(n-1)}{\tilde{r}_a(n)} - \frac{g(n)}{\tilde{r}_a(n)} + v(n). \tag{15}$$

Now defining the buffer tube upper bound (in time) of frame n as

$$t_b(n) = t_a(n) + \frac{g(n)}{\tilde{r}_a(n)}, \tag{16}$$

so that

fuboTV Media Inc.
Exhibit 1002
Page 272

$$t_b(n) - t_b(n-1) = t_a(n) - t_a(n-1) + \frac{g(n)}{\tilde{r}_a(n)} - \frac{g(n-1)}{\tilde{r}_a(n-1)}, \qquad (17)$$

the following update equation is obtained:

$$t_b(n) = t_b(n-1) + \frac{r_c(n)}{f(n)\tilde{r}_a(n)} + w(n-1), \qquad (18)$$

where

$$w(n-1) = \frac{g(n-1)}{\tilde{r}_a(n)} - \frac{g(n-1)}{\tilde{r}_a(n-1)} + v(n). \qquad (19)$$

is again an error term that captures variations around a locally constant arrival rate.

[0053] Using Eq. (16), the client can compute $t_b(n\text{-}1)$ from the measured arrival time $t_a(n - 1)$, the estimated arrival rate $\tilde{r}_a(n - 1)$, and $g(n - 1)$ (which can be transmitted to the client along with the data in frame n-1 or computed at the client as described in Section V-E). Then using Eq. (18), the client can control the coding bit rate $r_c(n)$ so that $t_b(n)$ reaches a desired value, assuming the frame rate and arrival rate remain roughly constant. From this perspective, Eq. (18) can be regarded as the state transition equation of a feedback control system and it is thus possible to use a control-theoretic approach to regulate the coding bit rate.

D. Control Objective

[0054] With the state transition equation defined in Eq. (18), uninterrupted playback can be achieved by regulating the coding bit rate so that the client buffer does not underflow. To introduce a margin of safety that increases over time, a *target schedule* is introduced, as illustrated in Fig. 5, whose distance from the playback deadline grows slowly over time. By regulating the coding bit rate, it is attempted to control the buffer tube upper bound so that it tracks the target schedule. If the buffer tube upper bound is close to the target schedule, then the arrival times of all frames will certainly be earlier than their playback deadlines and thus uninterrupted playback will be ensured. Note that controlling the actual arrival times (rather than their upper bounds) to the target would result in an approximately constant number of bits per frame, which would in turn result in very poor quality overall. By taking the leaky bucket model into account, it is possible to establish a control that allows the instantaneous coding bit rate to fluctuate naturally according to the encoding complexity of the content, within previously established bounds for a given average coding bit rate.

[0055] Although controlling the upper bound to the target schedule is our primary goal, it is also desirable to minimize quality variations due to large or frequent changes to the coding bit rate. This can be achieved by introducing into the cost function a penalty for relative coding bit rate differences.

[0056] Letting $t_T(n)$ denote the target for frame n, the following cost function is used to reflect both of concerns:

$$I = \sum_{n=0}^{N} \left((t_b(n) - t_T(n))^2 + \sigma \left(\frac{r_c(n+1) - r_c(n)}{\tilde{r}_a(n)} \right)^2 \right), \qquad (20)$$

where the first term penalizes the deviation of the buffer tube upper bound from the target schedule and the second term penalizes the relative coding bit rate difference between successive frames. N is the control window size and σ is a Lagrange multiplier or weighting parameter to balance the two terms.

fuboTV Media Inc.
Exhibit 1002
Page 273

III. Optimal Control Solution

[0057] Before presenting the optimal control solution, the design rational of the target schedule will be described.

A. Target Schedule Design

[0058] Fig. 6 is a graph showing an illustrative target schedule. The gap between the playback deadline and the target schedule is the desired client buffer duration (in client time). If the gap is small at the beginning of streaming, then it allows a small startup delay, while if the gap grows slowly over time, it gradually increases the receiver's ability to counter jitter, delays, and throughput changes.

[0059] The slope of the target schedule relates the average coding bit rate to the average arrival rate. Let $t_T(n)$ be the target for frame n. As illustrated in Fig. 6, the slope of the target schedule at frame n is

$$s(n) = \frac{t_T(n+1) - t_T(n)}{\tau(n+1) - \tau(n)}. \qquad (21)$$

If the upper bound $t_b(n)$ aligns perfectly with the target schedule (i.e., $t_b(n) = t_T(n)$) and the arrival rate r_a is constant (i.e., the $w(n\text{-}1)$ term vanishes), from Eq. (18),

$$s(n) = \frac{t_b(n+1) - t_b(n)}{\tau(n+1) - \tau(n)} = \frac{r_c(n)}{r_a}. \qquad (22)$$

Thus initially, when the slope was low, i.e., less than $1/v$, r_a/r_c is greater than v and more than v seconds of content are received per second of client time, causing the client buffer (which is playing out only v seconds of content per second of client time) to grow. Over time, as the slope approaches $1/v$, r_a/r_c approaches v and the buffer remains relatively constant (except for changes due to variations in the instantaneous coding bit rate), since content is received and played back at the same speed v. Next two target schedule functions are presented that illustrate the general design idea.

1) Logarithmic Target Schedule

[0060] One way to choose the target schedule t_T is to have the client buffer duration grow logarithmically over time. Specifically, if t_d is the playback deadline, then for each t_d greater than some start time t_{d0},

$$t_T = t_d - \frac{b}{a}\ln(a(t_d - t_{d0}) + 1). \qquad (23)$$

Since by Eq. (1), $t_d = t_{d0} + (\tau_d - \tau_{d0})/v$ this yields

$$s = \frac{dt_T}{d\tau_d} = \frac{dt_T dt_d}{dt_d d\tau_d} = \frac{1}{v} - \frac{b}{a(\tau_d - \tau_{d0}) + v}, \qquad (24)$$

and hence the initial slope at frame 0 (when $t_d = t_{d0}$) is $s(0) = (1\text{-}b)/v$. Setting b=0.5 implies that initially r_c/r_a=0.5/v, causing the client buffer to grow initially at two times real time. Further setting a=0.15 implies that the client buffer duration will be 7.68 seconds after 1 minute, 15.04 seconds after 10 minutes, and 22.68 seconds after 100 minutes, regardless of v.

fuboTV Media Inc.
Exhibit 1002
Page 274

2) Two-piece Linear Target Schedule

[0061] Another way to choose the target schedule t_T is to have the client buffer duration grow linearly at rate b seconds of media time per second of client time until the buffer duration reaches a seconds of media time, after which it remains constant. Specifically, for each t_d greater than some start time t_{d0},

$$t_T = \begin{cases} t_d - b(t_d - t_{d0}) & t_d \le t_{d0} + a/b \\ t_d - a & t_d \ge t_{d0} + a/b \end{cases} \quad (25)$$

The initial slope is again $s(0)=(1-b)/v$. Setting b=0.5 implies that initially r_c/r_a = 05/v , causing the client buffer to grow initially at two times real time. Further setting a = 10 implies that the client buffer duration will reach 10 seconds of media time after 20 seconds of client time, regardless of v.

[0062] Figs. 7(a) and (b) show graphs of the above two target schedules. As one can see, if a client buffer duration of 10 seconds is considered to be a safe level against jitter, delay and network fluctuations, then the two-piece linear target schedule reaches the safe level in 20 seconds, much faster than the logarithmic target schedule. On the other hand, the slope of the two-piece linear target schedule remains lower for longer (hence the coding bit rate and quality are lower for longer) and furthermore experiences an abrupt change at 20 seconds when its slope changes from 0.5/v to 1/v. Consequently, the coding bit rate will not change as smoothly as with the logarithmic target schedule, although it will not be as abrupt as the schedule itself because of the smoothness objective in the controller design.

B. Optimal Controller Design

[0063] Recall from Eq. (18) the fundamental state transition equation, which describes the evolution of the buffer tube upper bound $t_b(n)$ in terms of the coding bit rate $r_c(n)$:

$$t_b(n+1) = t_b(n) + \frac{r_c(n+1)}{f\tilde{r}_a} + w(n). \quad (26)$$

Here it is now assumed that the frame rate f and the average arrival rate $\tilde{r}_a$ are relatively constant. Deviations from this assumption are captured by $w(n)$.

[0064] It is desired to control the upper bound by adjusting the coding bit rate. As each frame arrives at the client, a feedback loop can send a message to the server to adjust the coding bit rate. Note, however, that by the time frame n arrives completely at the client, frame $n+1$ has already started streaming from the server. Thus the coding bit rate $r_c(n+1)$ for frame $n+1$ must already be determined by time $t_a(n)$. Indeed, at time $t_a(n)$, frame n+2 is the earliest frame for which the controller can determine the coding bit rate. Hence at time $t_a(n)$, the controller's job must be to choose $r_c(n+2)$. This one-frame delay in the feedback loop must be explicitly accounted for.

[0065] For simplicity, the target schedule is linearized around the time that frame n arrives. The linearization is equivalent to using a line tangent to the original target schedule at a particular point as an approximate target schedule. Thus,

$$t_T(n+1) - 2t_T(n) + t_T(n-1) = 0. \quad (27)$$

Rather than directly control the evolution of the upper bound, which grows without bound, for the purposes of stability an error space formulation is used. Defining the error as

$$e(n) = t_b(n) - t_T(n), \quad (28)$$

fuboTV Media Inc.
Exhibit 1002
Page 275

results in

$$e(n+1)-e(n) = (t_b(n+1)-t_T(n+1))-(t_b(n)-t_T(n)) \quad (29)$$

$$= (t_b(n+1)-t_b(n))-(t_T(n+1)-t_T(n)) \quad (30)$$

$$= \frac{r_c(n+1)}{f\tilde{r}_a}-(t_T(n+1)-t_T(n))+w(n), \quad (31)$$

from which in turn

$$\begin{aligned}&(e(n+1)-e(n))-(e(n)-e(n-1))\\ &= [r_c(n+1)-r_c(n)]/f\tilde{r}_a\\ &\quad -(t_T(n+1)-2t_T(n)+t_T(n-1))\\ &\quad +(w(n)-w(n-1))\end{aligned} \quad (32)$$

$$= \frac{r_c(n+1)-r_c(n)}{f\tilde{r}_a}+(w(n)-w(n-1)) \quad (33)$$

The control input is then defined as

$$u(n) = \frac{r_c(n+2)-\hat{r}_c(n+1)}{\tilde{r}_a}, \quad (34)$$

where $\hat{r}_c(n+1)$ is a possibly quantized version of $r_c(n+1)$ (as defined in Section IV-D) and the disturbance is defined as

$$d(n) = \frac{\hat{r}_c(n)-r_c(n)}{f\tilde{r}_a}+w(n)-w(n-1). \quad (35)$$

Then Eq. (33) can be rewritten

$$e(n+1) = 2e(n)-e(n-1)+\frac{u(n-1)}{f}+d(n). \quad (36)$$

Therefore , defining the error vector

fuboTV Media Inc.
Exhibit 1002
Page 276

$$\mathbf{e}(n) = \begin{bmatrix} e(n) \\ e(n-1) \\ u(n-1) \end{bmatrix} = \begin{bmatrix} t_b(n) \\ t_b(n-1) \\ \dfrac{r_c(n+1)}{\tilde{r}_a} \end{bmatrix} - \begin{bmatrix} t_T(n) \\ t_T(n-1) \\ \dfrac{\hat{r}_c(n)}{\tilde{r}_a} \end{bmatrix} \qquad (37)$$

the error space representation of the system can be expressed as

$$\mathbf{e}(n+1) = \begin{bmatrix} 2 & -1 & \dfrac{1}{f} \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e}(n) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(n) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} d(n), \qquad (38)$$

or $e(n+1) = \Phi e(n) + \Gamma u(n) + \Gamma_d d(n)$ for appropriate matrices Φ, Γ and Γ_d.

[0066] Assuming the disturbance *d(n)* is a pure white noise, and assuming *perfect state measurement* (i.e., all components of *e*(*n*) can be measured without using an estimator), the disturbance *d*(*n*) does *not* affect the controller design. Thus a linear controller represented by

$$u(n) = -G\mathbf{e}(n) \qquad (39)$$

can be used where G is a *feedback gain.* By the time frame *n* is completely received, all elements of **e**(*n*) are available at the client and *u*(*n*) can thus be computed. The ideal coding bit rate for frame *n* + 2 can then be computed as

$$r_c(n+2) = \hat{r}_c(n+1) - G\mathbf{e}(n)\tilde{r}_a. \qquad (40)$$

Finding the optimal linear controller amounts to finding the feedback gain *G** that minimizes the quadratic cost function defined in Section II-D. Before continuing with the design, the system *controllability matrix C*,

$$C = \begin{bmatrix} \Gamma & \Phi\Gamma & \Phi^2\Gamma \end{bmatrix} = \begin{bmatrix} 0 & \dfrac{1}{f} & \dfrac{2}{f} \\ 0 & 0 & \dfrac{1}{f} \\ 1 & 0 & 0 \end{bmatrix} \qquad (41)$$

will be checked first which has full rank for any frame rate *f*. Thus, the system is *completely controllable* and the state e(*n*) can be regulated to any desirable value. Now recall that the cost function defined in Section II-D is

fuboTV Media Inc.
Exhibit 1002
Page 277

$$I = \sum_{n=0}^{N} \left\{ (t_b(n) - t_T(n))^2 + \sigma \left(\frac{r_c(n+1) - r_c(n)}{\tilde{r}_a} \right)^2 \right\} \qquad (42)$$

$$= \sum_{n=0}^{N} \{ e(n)^T Q e(n) + u(n-1)^T R u(n-1) \}, \qquad (43)$$

where $Q = C^T C$ (with C = [1 0 0]) and R = σ. Then, the original control problem of tracking the target schedule while smoothing the coding bit rate fluctuations (i.e., minimizing the cost function I) is converted to a standard regulator problem in the error space. Letting $N \to \infty$, the infinite horizon optimal control problem can be solved by applying the results in [11, Section 3.3] to obtain an optimal regulator in two steps: 1) solving, to get S, the *discrete algebraic Riccati equation* (DARE)

$$S = \Phi^T \{ S - S\Gamma [\Gamma^T S \Gamma + R]^{-1} \Gamma S \} \Phi + Q, \qquad (44)$$

and 2) computing the optimal feedback gain

$$G^* = [\Gamma^T S \Gamma + R]^{-1} \Gamma^T S \Phi. \qquad (45)$$

The existence and uniqueness of S (and in turn of G^*) is guaranteed when Q is nonnegative definite and R is positive definite, which is straightforward to verify in this case.

C. Frame Rate

[0067] In the previous section, it was assumed that the frame rate is constant. This assumption is reasonable when streaming a single medium, such as video without audio. Variable frame rate video is usually achieved by skipping frames, which can be accommodated by setting $b(n) = 0$. However, usually video and audio are streamed together, and their merged coding schedule may have no fixed frame rate. Even if there is a fixed frame rate f, it may be desirable to operate the controller at a rate lower than f, to reduce the feedback rate, for example.

[0068] To address these issues, in practice the notion of a *virtual frame rate* is used. A virtual frame rate f is chosen, for example f =1 frame per second (fps); media time is partitioned into intervals of size $1/f$; and all of the (audio and video) frames arriving within each interval are modeled as a *virtual frame* whose decoding and playback deadline is the end of the interval.

[0069] This approach has several advantages. First, it allows offline design of a universal feedback gain, which is independent of the actual frame rate of the stream or streams. Second, it allows the rate of feedback from the client to the server to be reduced. And finally, since the interval between virtual frames is typically safely larger than a round trip time (RTT), a one-frame delay in the error space model (as described in the previous section) is sufficient to model the feedback delay. Otherwise it would be necessary to model the feedback delay with approximately RTT/f additional state variables to represent the network delay using a shift register of length RTT/f.

[0070] In the remainder of this description a virtual frame rate f = 1 fps is employed, and is referred to simply as the frame rate. Likewise, a virtual frame is referred to simply as a frame.

[0071] It is noted that with the present controller design rationale, the deviation of the buffer tube upper bound from the control target can be reduced by decreasing the σ value. A smaller value of σ value implies a relative larger penalty on the deviation term in the cost function and thus forces the upper bound to track the target more closely. This, however, happens at the cost of sacrificing coding bit rate smoothness, since the corresponding term in the cost function will be weighted less. It has been found with σ = 500 that while the buffer tube upper bound deviates only slightly from the control target, the coding bit rate has undesirable oscillations. On the other hand, a large σ value will certainly yield smoother coding bit rates, but might also incur client buffer underflow since the buffer tube upper bound is allowed to

fuboTV Media Inc.
Exhibit 1002
Page 278

deviate significantly away from the control target. Therefore, a good choice of σ should take into account this trade-off. In tested embodiments of the present system and process, $\sigma = 4000$ was chosen when the coding bit rate switches up and $\sigma = 2000$ when it switches down. Note that a slightly more aggressive strategy is allowed in the latter case to further reduce the chance of client buffer underflow.

IV. Practical Issues with Streaming

A. Fast Startup

[0072] As discussed in previous sections, the startup delay is the length of the period from the time that content first begins to arrive at the client to the time that playback begins. During this period, content accumulates in the receiver buffer to counter packet jitter, retransmission delay, variations in network bandwidth, and variations in instantaneous coding bit rate. It is conceivable that a longer startup delay would increase the chances of being able to maintain continuous playback in a dynamic network environment. On the other hand, users expect the startup delay to be as small as possible. Thus, it is desirable to investigate techniques that can reduce the startup delay while retaining robustness. One possible approach is to transmit the content at a faster than normal rate at the beginning of streaming. This *bursting* technique will certainly build up the buffer duration in a small amount of time. It, however, puts extra pressure on the network by demanding a higher than normal initial bandwidth, which may not even be available.

[0073] The present system and process employs an alternative *fast startup* technique, which takes advantage of the properties of adaptive media. As discussed in previous sections, by choosing an initial coding bit rate r_c equal to half the arrival rate r_a (divided if necessary by the playback speed v), the client buffer duration can grow at two times real time during playback. Growing the client buffer during playback enables the startup delay to be low, because playback can begin while the buffer duration is still low. Beginning playback while the buffer duration is low is not particularly risky over the short term, because the probability of deep congestion occurring in any short interval is low. However, the probability of deep congestion occurring in a long interval is high, so it is important for the buffer duration to be high over the long term. Without the ability to grow the buffer duration during playback, startup would have to be delayed until the buffer duration was sufficiently high to guarantee continuous playback over the long term.

[0074] Moreover, if the transmission rate is twice the coding bit rate, the startup delay can be further reduced by taking advantage of properties of the leaky bucket model [10]. As detailed in Section II-B, the startup delay for a given bit stream is $D = F^d_{\min}(R)/R$ when the stream is transmitted at rate R. This is ordinarily equal to $F^d_{\min}(r_c)/r_c$ when transmitting the stream at its coding bit rate. However, when transmitting the stream at a rate $r_a > r_c$ ($r_c = 0.5r_a/v$), then the startup delay drops to $F^d_{\min}(r_a)/r_a$. Thus the startup delay D decreases both because the numerator decreases and because the denominator increases.

[0075] The graph in Fig. 8 illustrates the decrease in the initial decoder buffer fullness $F^d_{\min}(R)$ as R changes from r_c to r_a. In particular, it depicts the coding schedule for a given bit stream, as well as upper and lower bounds, denoted Tube I and Tube II, corresponding to two leaky buckets with leak rates r_c and r_a respectively, both containing the coding schedule. Tube II is smaller than Tube I, since the minimum size $B_{\min}(R)$ of a leaky bucket containing a given stream is decreasing in the leak rate R [10]. Likewise, the initial decoder buffer fullness $F_{\min}(R)$ is decreasing in R [10]. Hence the playback deadline for frame 0 can begin as early as client time $t_{0\_II} = F^d_{\min}(r_a)/r_a$, instead of $t_{0\_I} = F^d_{\min}(r_c)/r_a$. From there, the playback deadline advances at $1/v$ seconds of client time per second of media time.

B. Controller Initialization

[0076] As illustrated in Fig. 8, the target schedule starts at the same time as the playback deadline and grows according to a predefined function. The controller attempts to control the upper bound of Tube I to the target schedule. Initially the upper bound of Tube I is above the target schedule (and is indeed above the playback deadline, though it is known that this is safe). Hence, when the playback starts, the controller would try to close the gap by decreasing the coding bit rate. This, however, would not be desirable because the current coding bit rate is already lower than the arrival rate to allow the client buffer to grow. Further reduction of the coding bit rate would not be proper. To avoid this effect, the controller is initialized when the upper bound of Tube I exceeds the target schedule i.e., at point B in Fig. 8. Point B can be found analytically, but in practice there is no need to explicitly solve for it. The controller can be initialized as soon as the upper bound of Tube I exceeds the target.

fuboTV Media Inc.
Exhibit 1002
Page 279

C. Exponential Averaging of the Arrival Rate

[0077] Using the average arrival rate (instead of the instantaneous arrival rate) helps to reduce coding bit rate oscillations. This section details a new exponential averaging algorithm for the arrival rate.

[0078] Let $\tilde{r}_a(k)$ and $r(k)$ be the average arrival rate and the instantaneous arrival rate, respectively, when packet k is received. Note that unlike the controlling operation, the rate averaging operation may be performed after the arrival of every *packet,* rather than after the arrival of every *frame.* Hence the discrete packet index k is used rather than the frame index n . Instead of using the widely adopted exponentially weighted moving average (EWMA)

$$\tilde{r}_a(k) = \beta(k)\tilde{r}_a(k-1) + (1-\beta(k))r_a(k) \qquad (46)$$

with constant $\beta(k) = \beta$, the exponential averaging is performed more carefully. In the present algorithm, the factor $\beta(k)$ is not constant, but varies according to the packets' inter-arrival gaps. This new algorithm has several advantages over the EWMA algorithm with constant $\beta(k)$. First, the estimate of the average arrival rate $\tilde{r}_a(k)$ goes to zero naturally as the gap since the last packet goes to infinity, rather than being bounded below by $\beta\tilde{r}_a(k\text{-}1)$. Second, the estimate of the average arrival rate $\tilde{r}_a(k)$ does not go to infinity as the gap since the last packet goes to zero. This is especially important, since packets often arrive in bursts, causing extremely high instantaneous arrival rates. And finally, the estimate of the average arrival rate $\tilde{r}_a(k)$ does not over-weight the initial condition, as if it represented the infinite past. This is especially important in the early stages of estimation.

[0079] As in Eq. (11), the instantaneous arrival rate after packet k is defined as

$$r_a(k) = \frac{b(k)}{t_a(k) - t_a(k-1)}, \qquad (47)$$

where here $b(k)$ denotes the size of packet k and $t_a(k)$ denotes the arrival time of packet k. The discrete time function $r_a(k)$ is extended to the piecewise constant continuous time function $r_a(t)$ by

$$r_a(t) = r_a(k) \quad \text{for all } t \in (t_a(k-1), t_a(k)), \qquad (48)$$

as illustrated in the graph in Fig. 9. Then the function $r_a(t)$ is filtered by the exponential impulse response $\alpha e^{-\alpha t}$, $t \geq 0$, for some time constant $1/\alpha$:

$$\tilde{r}_a(k) = \frac{\int_{t(0)}^{t(k)} r_a(t')\alpha e^{-\alpha(t(k)-t')}dt'}{\int_{t(0)}^{t(k)} \alpha e^{-\alpha(t(k)-t')}dt'}. \qquad (49)$$

(Here and in the remainder of this sub-section the subscript is suppressed from the arrival time $t_a(k)$.) Note that $\int_t^{\infty} \alpha e^{-\alpha t'}dt' = e^{-\alpha t}$, the denominator integral can be expressed as $1 - e^{-\alpha(t(k)-t(0))}$. Now, the range of the numerator integral is split into ranges (t(0), t(k-1)] and (t(k-1),t(k)] to obtain a recursive expression for $\tilde{r}_a(k)$ in terms of $\tilde{r}_a(k\text{-}1)$ and $r_a(k)$,

fuboTV Media Inc.
Exhibit 1002
Page 280

$$\tilde{r}_a(k) = \frac{1-e^{-\alpha[t(k-1)-t(0)]}}{1-e^{-\alpha[t(k)-t(0)]}} e^{-\alpha[t(k)-t(k-1)]} \tilde{r}_a(k-1) + \frac{1-e^{-\alpha[t(k)-t(k-1)]}}{1-e^{-\alpha[t(k)-t(0)]}} r_a(k) \tag{50}$$

$$= \beta(k)\tilde{r}_a(k-1) + (1-\beta(k))r_a(k), \tag{51}$$

where

$$\beta(k) = \frac{e^{-\alpha[t(k)-t(k-1)]} - e^{-\alpha[t(k)-t(0)]}}{1-e^{-\alpha[t(k)-t(0)]}}. \tag{52}$$

Note that $\beta(k)$ is numerically stable as k goes to infinity. However, as the gap $\delta = t(k) - t(k\text{-}1)$ goes to zero, $1 - \beta(k)$ goes to zero while $r_a(k)$ goes to infinity. Their product, however, is well behaved. Indeed,

$$\tilde{r}_a(k) = \frac{1-e^{-\alpha[t(k-1)-t(0)]}}{1-e^{-\alpha[\delta+t(k-1)-t(0)]}} e^{-\alpha\delta} \tilde{r}_a(k-1) + \frac{1-e^{-\alpha\delta}}{1-e^{-\alpha[t(k)-t(0)]}} \frac{b(k)}{\delta} \tag{53}$$

$$\rightarrow \tilde{r}_a(k-1) + \frac{\alpha b(k)}{1-e^{-\alpha[t(k)-t(0)]}} \tag{54}$$

as $\delta \rightarrow 0$, using l'Hôpital's rule. Thus Eq. (54) is the update rule in the case when $t(k) = t(k\text{-}1)$.

D. Choosing a Stream Given a Coding bit rate

[0080] When the client requests a coding bit rate $r_c(n)$, the server complies by choosing a stream (or sub-stream of a scalable stream) having coding bit rate $\hat{r}_c(n)$ approximately equal to $r_c(n)$. There are several reasons that $\hat{r}_c(n)$ may differ from $r_c(n)$. The first reason is that there are only a finite number of streams (or sub-streams) in the media file, even if fine grain scalable coding is used. Thus there may be no stream in the media file with average coding bit rate exactly equal to $r_c(n)$. The second reason is that, even if there is a stream in the media file with average coding bit rate exactly equal to $r_c(n)$, the buffer tube for the stream may be too large to allow switching to the stream without risk of client buffer underflow. In fact, whenever the stream switches, there is generally a discontinuity in the upper bound, which may be either positive or negative. A positive shift in the upper bound is illustrated in the graph in Fig. 10, which, if large, could cause the client buffer to underflow either immediately or eventually.

[0081] Thus the server must choose a stream that causes the upper bound to shift up no more than some amount $\Delta^{max}g(n\text{-}1)$ supplied to it by the client. The client supplies $\Delta^{max}g(n\text{-}1)$ to the server in its feedback along with $r_c(n)$, shortly after client time $t_a(n\text{-}2)$ (after frame n-1 has already begun streaming). Upon receiving the feedback, the server selects a stream with coding bit rate $\hat{r}_c(n)$ as high as possible such that $\hat{r}_c(n) \le r_c(n)$ and, if $\hat{r}_c(n) > \hat{r}_c(n\text{-}1)$ (i.e., if it is a switch up in rate), then $g^{new}(n\text{-}1) - g^{old}(n\text{-}1) \le \Delta^{max}g(n\text{-}1)$, where $g^{new}(n\text{-}1)$ and $g^{old}(n\text{-}1)$ are illustrated in Fig. 10. The constraint given by $\Delta^{max}g(n\text{-}1)$ is not applied if it is a switch down in rate.

[0082] The client chooses $\Delta^{max}g(n - 1)$ to limit (its prediction of) what the upper bound would be at time $t_a(n - 1)$ if the new coding bit rate were in effect, namely,

fuboTV Media Inc.
Exhibit 1002
Page 281

$$t_b^{new}(n-1) \approx \frac{t_b(n-2)+\hat{r}_c(n-1)}{f\tilde{r}_a}+\frac{\Delta g(n-1)}{\tilde{r}_a} \qquad (55)$$

$$\leq t_T(n-1)+p[t_d(n-1)-t_T(n-1)]. \qquad (56)$$

That is, the client chooses $\Delta^{max}g(n\text{-}1)$ to limit $t_b^{new}(n-1)$ so that it would be no more than fraction p of the way from the target $t_T(n\text{-}1)$ to the playback deadline $t_d(n\text{-}1)$. In tested embodiments of the present system and process, p=1/3 was chosen.

[0083] It is noted that in an alternate embodiment, the information as to what coding bit rates the server supports can be provided to the client ahead of time and the client can be responsible for performing the foregoing analysis. In this case, the client would only request coding bit rates that are supported by the server, and there would be no need to provide the $\Delta^{max}g(n\text{-}1)$ to the server with the request.

E. Control Target Adjustment

[0084] When a frame with a new average coding bit rate $\hat{r}_c(n)$ arrives at the client at time $t_a(n)$, there is a shift in the upper bound. This shift can be on the order of seconds and hence, rather than being negligible, can be confusing to the controller. If the shift is upward, for example, the controller will immediately try to reduce the coding bit rate $r_c(n+2)$. If the shift is downward, on the other hand, the controller will immediately try to increase the coding bit rate $r_c(n+2)$. Either way is probably not good; the intention is that $\hat{r}_c(n)$ will be maintained unless there is a disturbance in the arrival rate. One solution is to introduce a simultaneous shift in the control target schedule equal to $\Delta g(n\text{ - }1)/\tilde{r}_a$; where $\Delta g(n\text{ - }1) = g^{new}(n\text{ - }1) - g^{old}(n\text{-}1)$ is the actual shift in the upper bound (in bits) at frame n-1 computed at the server, as illustrated in Fig. 10. The server can send this value to the client along with frame n. If there is no stream change, this value is simply zero.

[0085] If the control target schedule is adjusted whenever the coding bit rate changes, it will no longer follow the designed target schedule. The adjusted target schedule will be referred to as the *control target* schedule to distinguish it from the *designed target* schedule (or simply the *target schedule).*

[0086] The control target schedule, of course, should have a tendency to approach the designed target schedule. The basic idea is to decrease the slope of the control target schedule when it is above the designed target schedule and to increase the slope when it is below.

[0087] For the logarithmic target schedule $t_T = t_d - \frac{b}{a}\ln(at_d+1)$ (where $t_d = t_{d0} + (\tau_d - \tau_{d0})/v$), according to Eq. (24) the slope at media time τ_d is

$$s = \frac{dt_T}{d\tau_d} = \frac{1}{v} - \frac{b}{a(\tau_d-\tau_{d0})+v}. \qquad (57)$$

If d is defined as the distance between the playback deadline and the target schedule, namely

$$d = \frac{b}{a}\ln\left(a\left(\frac{\tau_d-\tau_{d0}}{v}\right)+1\right), \qquad (58)$$

then the slope may be expressed as a function of d

fuboTV Media Inc.
Exhibit 1002
Page 282

$$s = \frac{1}{\nu} - \frac{b}{\nu e^{(a/b)d}}. \qquad (59)$$

Hence whenever d is the distance between the playback deadline and the control target, the slope of the control target is set to s in Eq. (59). Specifically, if $t_T^A(n)$ is the control target at frame n after the shift, then $t_T^A(n\text{-}1)$ is reset to be $t_T^A(n)\text{-}s/f$. Then $t_T^A(n)$ and $t_T^A(n\text{-}1)$ is used in place of $t_T(n)$ and $t_T(n\text{-}1)$ to compute the error vector $e(n)$ in Eq. (37). The resulting error vector is then used to compute the ideal coding bit rate in Eq. (40).

[0088] For the two-piece linear target schedule, the slope is easy to compute by using a predefined time period over which the control target schedule is expected to return to the target schedule. The slope of the control target schedule can then be computed from the distance d and the period. In tested embodiments of the present system and process the period was set to 50 seconds.

V. Implementation Details

[0089] This section highlights implementation details on both the sender and the receiver side.

A. Generation of Virtual Streams

[0090] In tested embodiments of the present a system and process, a fine grained scalable (FGS) stream comprises a set of data units, each tagged by a Lagrange multiplier λ representing the per-bit decrease in distortion if the data unit is received by the client. If the λ for the data unit is above a threshold, then the data unit is included in a virtual stream corresponding to that threshold. Each threshold corresponds to an overall number of bits and hence an average coding bit rate for the virtual stream. In tested embodiments, $N = 50$ virtual streams are generated. A threshold is chosen for each stream such that the resulting streams have coding bit rates that are uniformly spaced in the log domain between lower and upper bounds.

[0091] During streaming, when the server reads a data unit from the media file, it includes the data unit in the virtual stream currently being transmitted if its Lagrange multiplier λ is above the threshold for the stream.

B. Leaky Bucket Computations at the Sender

[0092] For each virtual stream, leaky bucket parameters $(R, B_{\min}(R), F_{\min}^{d}(R))$ are pre-computed offline for $R = R_{avg}$ and $R = R_{\max}$, where $R_{avg} = r_c$ is the average coding bit rate of the stream, and $R_{max} = 2r_c$. These leaky bucket parameters are sent to the client in a preamble.

[0093] In addition, during streaming the server performs on-line leaky bucket simulations for each stream. Specifically, whenever the server reads a data unit from the media file, it determines the virtual streams to which the data unit belongs, using the Lagrange multiplier of the data unit and the list of thresholds for each stream. The sender then updates, for the determined streams, the states of those leaky buckets having leak rates equal to an average coding bit rate R_{avg}, using Eqs. (2) and (3). Once all the data units in a frame are read from the media file, the sender computes $g(n) = B_{\min}(R_{avg}) - B^e(n)$ for each of the virtual streams. On a stream switch (i.e., $\hat{r}_c(n) \neq \hat{r}_c(n-1)$), the gap $g^{new}(n)$ for the new stream is transmitted to the client along with $\Delta g(n\text{-}1)=g^{new}(n\text{-}1)\text{-}g^{old}(n\text{-}1)$ as described below. It is easy to see that the cost of updating the leaky bucket states is quite low. However, it is also possible to pre-compute these values and store them with each data unit in the media file.

C. Initial Coding bit rate Selection

[0094] At the beginning of a streaming session, the sender needs to have some knowledge of the available network bandwidth so that it can choose an initial coding bit rate (usually half of the bandwidth). The bandwidth estimate can be drawn from proactive measurements, using approaches such as packet pair [12], path chirp [3], etc., or reactive approximations based on history values. The exact form of the initial bandwidth estimation is beyond the scope of this work.

D. Coding bit rate Switching

[0095] The rate control feedback from the client contains the frame number at which feedback is generated (s.g., n -

fuboTV Media Inc.
Exhibit 1002
Page 283

2 in the previous section) and the maximum allowable shift of the upper bound in bits (e.g., $\Delta^{max}g(n\text{-}1)$ in the previous section). If the sender finds a suitable coding bit rate and makes a switch at frame n, it will transmit three values to the client along with the frame: the new coding bit rate $\hat{r}_c^{new}(n)$, the current gap to the upper bound $g^{new}(n)$, and the shift $\Delta g(n\text{-}1) = g^{new}(n\text{-}1)\text{-}g^{old}(n\text{-}1)$. With this information, the client can properly adjust its control target schedule as well as its upper bound. Note that coding bit rate switching always happens at the beginning of a new frame, never inside a frame.

E. Optimal Rate Control at the Client

[0096] Whenever a new coding bit rate starts, the client receives the value $g(n)$ along with the new frame. The values of $g(n)$ for successive frames can be then inferred by the client itself based on the coding bit rate $\hat{r}_c(n)$ and the frame size $b(n)$. The client records the arrival frame time $t_a(n)$, calculates the buffer tube upper bound $t_b(n)$ and then computes the deviation $e(n)$. If there is a coding bit rate switch, it will also compute the buffer tube shift and adjust the control target schedule accordingly. Then, $e(n)$ is fed to the optimal rate controller, which outputs a desired new coding bit rate. The latest new coding bit rate is fed back to the sender whenever there is a feedback opportunity, which could be generated at regular intervals or on-demand.

F. Exemplary Process

[0097] An exemplary process for the implementation of the foregoing system will now be provided as applicable to one embodiment of the present invention.

[0098] Referring to Figs. 11A-C, the server first computes a set of parameters defining an initial state of the encoder buffer as it would exist if a data stream corresponding to a supported coding bit rate were streamed therefrom, for each of the supported coding bit rates (process action 1100). These parameters, which include the coding bit rate associated with the data stream, the size of the encoder buffer employed with the coding bit rate of the data stream, and a value indicative of the initial encoder buffer fullness exhibited at the coding bit rate of the data stream, are provided to the client in a preamble to the streaming media data (process action 1102). The server then streams the media data to the client at a prescribed initial coding bit rate, during a startup period (process action 1104). As the data is streamed, the server computes the upper bound gap for each frame generated for each coding bit rate supported by the server (process action 1106). Initially, the client employs the encoder buffer parameters to determine on a frame-by-frame basis a new, optimal coding bit rate for a future frame (process action 1108). In addition, the client computes the aforementioned maximum allowable difference value for the frame generated by the server just prior to the future frame, i.e., the maximum shift value (process action 1110). The client also begins playback of the frames it has received as soon as the client's decoder buffer has the minimum amount of data therein that is needed to ensure an underflow condition does not occur at the initial coding bit rate and continues to playback each frame received thereafter (process action 1112). In addition, once the aforementioned startup period is over, the client requests that the streaming media data be supplied at a new coding bit rate corresponding to the determined optimal coding rate, starting at the future frame, and provides the maximum shift value along with the request (process action 1114).

[0099] The server finds a supported coding bit rate that is equal to, or if none are equal, the closest smaller rate to, the optimum coding bit rate requested by the client (process action 1116). It is next determined by the server if the supported coding bit rate found is lower or higher than the coding bit rate associated with the frame generated immediately prior to the frame specified in the client request (process action 1118). If it is lower, then server computes the aforementioned shift value for the frame generated immediately prior to the frame specified in the client request (process action 1120), and generates all future frames beginning with the frame specified by the client at the supported rate found and streams them to the client, along with the computed upper bound gap and shift value, and an indication of the new coding bit rate (process action 1122). However, if the supported coding bit rate found is higher, then server computes the shift value for the frame generated immediately prior to the frame specified in the client request (process action 1124) and then determines if the shift value is less than or equal to the maximum shift value provided in the client request (process action 1126). If the shift value is less than or equal to the maximum shift value, then the server generates all future frames beginning with the frame specified by the client at the supported coding bit rate found and streams them to the client, along with the computed upper bound gap and shift value, and an indication of the new coding bit rate (process action 1122). But, if the shift value is greater than the maximum shift value, then the next lower supported coding bit rate is found (process action 1128) and the process actions 1118 through 1128 are repeated.

[0100] The client shifts the currently scheduled target arrival time associated with a just received first frame at the new coding bit rate by the shift value provided, and shifts the currently scheduled target arrival times for future frames such that they collectively approach, over a period of time, and eventually coincide with, the previous target arrival times for those frames (process action 1130). The client then employs the encoder buffer parameters and the upper bound gap value provided by the server to determine a new, optimal coding bit rate for a future frame and to compute the

fuboTV Media Inc.
Exhibit 1002
Page 284

maximum shift value for the frame generated by the server just prior to the future frame (process action 1132). The client requests that the streaming media data be supplied at a new coding bit rate corresponding to the determined optimal coding rate starting at the future frame specified by the client, and includes the computed maximum shift value (process action 1134). Process actions 1116 through 1134 are then repeated for the duration of the streaming media data transmission.

VI. Multiple Bit Rate Streaming

[0101] Multiple bit rate (MBR) streaming is a network adaptive technique that is widely used in commercial streaming media systems. In MBR streaming, in contrast to scalable streaming, the content is encoded into several (typically 5-7) independent streams at different coding bit rates. Often, each stream is optimized for a common type of network connection (e.g., dial-up, DSL, cable). During an MBR streaming session, the proper coding bit rate is dynamically selected based on the available network bandwidth, with the goal of achieving the maximum possible quality under the condition of uninterrupted playback. It is easy to see that MBR streaming is analogous to scalable streaming. Indeed MBR streaming can be viewed as a special case of scalable streaming with a limited number of coding bit rates available. Hence, the foregoing optimal control approach should be applicable to this case.

[0102] There are, however, several differences that complicate MBR streaming, which need to be carefully addressed. First, as just mentioned, in MBR streaming there are only a limited number of coding bit rates available. This coarse quantization of the desired coding bit rate introduces a significant nonlinearity into the closed loop system. In fact, the large gaps between the available coding bit rates introduce oscillations. For example, if two neighboring coding bit rates straddle a constant arrival rate, the controller will oscillate between the two coding bit rates in an attempt to keep the client buffer at a target level.

[0103] Second, in MBR streaming the coding bit rate cannot be switched at an arbitrary time. In fact, before the server can switch to a new stream, it may have to wait for the next clean point (e.g., I frame) in the new stream, which could be five or ten seconds away. Thus, the old coding bit rate may continue for quite a while before it changes to the new coding bit rate. From the controller's perspective, this long random extra delay tends to destabilize the closed-loop system.

[0104] Third and finally, in MBR streaming, server performance issues are critical. The commercial-grade streaming media systems that use MBR streaming do so because of the minimal computational load that it imposes on the server compared to scalable streaming. Thus, for MBR streaming it is important to keep almost all computation and state maintenance on the client side. In particular, the server will not be able to update the leaky bucket information for each stream. Instead, the client must use some mechanism for estimating and maintaining this information.

A. Conservative Up-Switching

[0105] In this subsection, a technique to help stabilize the control system and reduce steady state oscillations to a period of at least a minute is described. With this technique, rapid down-switching is permitted. In fact, the value of σ is reduced from 2000 to 500, changing the balance between responsiveness and smoothness of the coding bit rate in favor of rapid switching response. However, only conservative up-switching is permitted. Conservative up-switching ensures that spurious changes in coding bit rate do not occur, and that oscillations in the coding bit rate have a low frequency. In particular, conservative up-switching reduces the oscillations between two adjacent but widely spaced MBR coding bit rates, one above the arrival rate and one below the arrival rate.

[0106] In one embodiment, conservative up-switching establishes a conservative limit that limits the coding bit rate to no more than the arrival rate. However, in another embodiment, the method behind conservative up-switching is to establish a conservative limit on how high the coding bit rate can be raised above the arrival rate. If the current coding bit rate is below the arrival rate, and the client buffer duration begins to increase above its target level, then the coding bit rate can be switched up to a new coding bit rate above the arrival rate only if the new coding bit rate is below the conservative limit. When the client buffer duration begins at the target level, the conservative limit is equal to the arrival rate. However, as the client buffer duration increases, the conservative limit increases as well. Thus, if the current coding bit rate is below the arrival rate, and the next higher coding bit rate is above the arrival rate, then it will be possible to switch up to the next higher coding bit rate only after the client buffer duration has increased sufficiently so that the conservative limit rises above the higher coding bit rate. Once the coding bit rate is switched up to the higher coding bit rate, the client buffer begins to drain since the coding bit rate is then above the arrival rate. Eventually, when the buffer drains back below its target level, the controller will rapidly switch the coding bit rate back down to the coding bit rate below the arrival rate.

[0107] Given the current client buffer duration, the conservative limit is set to a value such that if the coding bit rate is switched up to a new coding bit rate at this value, the client buffer would take at least Δt seconds of client time to drain back to the target level. Thus, the mechanism ensures that the period of oscillation will be at least Δt seconds. In tested embodiments of the present system and process, Δt is set to 60 seconds.

fuboTV Media Inc.
Exhibit 1002
Page 285

[0108] Fig. 12 shows how the conservative limit is computed. Let $\Delta\tau_1$ be the client buffer duration (in media time) at the moment that the coding bit rate is switched up from r_c^{old} to r_c^{new}. Thus $\Delta\tau_1$ is the number of seconds of content that will be consumed at the old coding bit rate r_c^{old} before content at the new coding bit rate begins to be consumed. (For simplicity it is assumed that all of the content in the client buffer at the time of the switch is coded at rate r_c^{old}.) Let $\Delta\tau_2$ be the number of seconds of content that is consumed at the new coding bit rate r_c^{new} before the client buffer duration drops to some level $\Delta\tau_3$ seconds (in media time), greater than the target level $\Delta\tau_T$. The duration of this phase is determined such that the total time since the switch is exactly $\Delta t = (\Delta\tau_1 + \Delta\tau_2)/v$ seconds (in client time). Now, the number of bits that arrive in this time is $r_a\Delta t = r_c^{new}(\Delta\tau_2 + \Delta\tau_3) \geq r_c^{new}(\Delta\tau_2 + \Delta\tau_T) = r_c^{new}(v\Delta t - \Delta\tau_1 + \Delta\tau_T)$, or

$$r_c^{new} \leq \frac{r_a \Delta t}{v\Delta t - \Delta\tau_1 + v\Delta t_T}, \qquad (60)$$

where Δt_T is the target buffer duration in client time. The parameter Δt can be tuned to yield the desired behavior. A large Δt means that up-switching will be more conservative, while a smaller Δt means that up-switching will be more prompt. In tested embodiments, Δt is set to 60 seconds while the target Δt_T is typically about 10 seconds.

B. Buffer Tube Upper Bound Estimation

[0109] In Section V-D it was specified that the server sends three values to the client at the beginning of each change in coding bit rate: the new coding bit rate $\hat{r}_c^{new}$, the current gap to the upper bound $g^{new}(n)$, and the control target shift $\Delta g(n-1) = g^{new}(n-1) - g^{old}(n-1)$. The server computes the latter two values by running a leaky bucket simulator for each coding bit rate. The client continues to update $g(n)$ for the new coding bit rate by running its own leaky bucket simulator for the new coding bit rate. That is, beginning with the initial condition $F^e(n) = B - b(n) - g^{new}(n)$, for each successive frame the client computes

$$B^e(n) = F^e(n) + b(n) \qquad (61)$$

$$F^e(n+1) = \max\{0, B^e(n) - \hat{r}_c f(n)\}, \qquad (62)$$

where

$$f(n) = \frac{1}{\tau(n+1) - \tau(n)} \qquad (63)$$

is the instantaneous frame rate, as in Eqs. (2), (3) and (4). From this, the client can compute

$$g(n) = B - B^e(n) \qquad (64)$$

for each frame.

fuboTV Media Inc.
Exhibit 1002
Page 286

[0110] However, if the server is unable to simulate the leaky buckets and cannot send $g^{new}(n)$ to the client, then the client must estimate this information for itself. In this case it is recommended that the client estimates $g^{new}(n)$ as an upper bound such as $\hat{g}^{new}(n)=B-b(n)\geq g^{new}(n)$. Then, beginning with initial condition $\hat{F}^{e}(n)=B-b(n)-g^{new}(n)$ (which equals 0 in this case), for each successive frame the client computes

$$\hat{B}^{e}(n) = \hat{F}^{e}(n) + b(n) \tag{65}$$

$$\hat{F}^{e}(n+1) = \max\{0, \hat{B}^{e}(n) - \hat{r}_c/f(n)\}, \tag{66}$$

as well as

$$\hat{g}(n) = B - \hat{B}^{e}(n). \tag{67}$$

It is easy to see by induction that $\hat{F}^{e}(n) \leq F^{e}(n)$, $\hat{B}^{e}(n) \leq B^{e}(n)$, and $\hat{g}(n) \geq g(n)$. Moreover, these bounds each become tighter by $\delta(n) = \hat{r}_c/f(n) - B^{e}(n)$ whenever $\delta(n) > 0$, i.e., whenever $F^{e}(n+1)$ is clipped to 0 in Eq. (66). In fact, given enough time they may eventually become tight.

[0111] Note that whenever the bounds tighten by $\delta(n) > 0$, the control target must be shifted by $\Delta g(n)/\tilde{r}_a$, where $\Delta g(n) = -\delta(n)$. Furthermore, whenever n is the first frame of a new coding bit rate, the control target should be shifted by $\Delta g(n)/\tilde{r}_a$, where $\Delta g(n) = \hat{g}^{new}(n)-\hat{g}^{old}(n)$. Here, $\hat{g}^{old}(n)$ can be determined by running Eqs. (65), (66) and (67) for one extra step, namely if n is the first frame of the new coding bit rate,

$$\hat{F}^{e}(n) = \max\{0, \hat{B}^{e}(n-1) - \hat{r}_c^{old}/f(n-1)\} \tag{68}$$

$$\hat{B}^{e}(n) = \hat{F}^{e}(n) + b(n) \tag{69}$$

$$\hat{g}^{old}(n) = B - \hat{B}^{e}(n). \tag{70}$$

It is easy to see that if $\hat{g}^{new}(n) = B - b(n)$, then $\Delta g(n)=\hat{F}^{e}(n)$ as computed in Eq. (68).

VII. Sender-Driven Streaming

[0112] The preceding discussions have assumed that the control process for adapting the stream or its bit rate to the prevailing conditions is located within the client. This is called *receiver-driven* streaming because the client (i.e., the receiver) makes the decisions and informs the server (i.e., the sender). However, it should be clear that sender-driven streaming is also possible, in which the control process operates at the sending side. For example, using TCP for transmission of the streaming media data, the control process can operate at the server, measuring the transmission times of each frame and estimating the transmission rate, instead of the control process operating at the client, measuring the arrival times of each frame and estimating the arrival rate. Using TCP, the transmission time of a frame is substantially the same as its arrival time, and likewise the transmission rate is substantially the same as its arrival rate; hence these can be used interchangeably. Generally, the transmission time or the arrival time of a packet may be called a time stamp. In sender-driven streaming, the server can infer the state of the client buffer by knowing what it has already transmitted and by knowing the playout times for the transmitted frames. Sender-driven streaming reduces the overhead of communicating control information between the client and the server, but it also increases the computational burden on the server. In some situations, this may be a desirable trade-off. Those skilled in the art will appreciate that it is possible to appropriately modify the protocols described herein to cases where the control algorithm resides in the server or in

fuboTV Media Inc.
Exhibit 1002
Page 287

another location, rather than in the client.

VIII. References

[0113]

[1] G.J. Conklin, G.S. Greenbaum, K.O. Lillevold, A.F. Lippman, and Y.A. Reznik. Video coding for streaming media delivery on the Internet. *IEEE Trans. Circuits and Systems for Video Technology,* 11 (3):269-281, March 2001. special issue on Streaming Video.

[2] T. Wiegand and G. Sullivan. Joint video specification rec. h.264 & 14496-10 avc. Non-Final Draft of Final Draft International Standard (FDIS) JVT-G050, ITU-T & ISO/IEC, Pattaya, Thailand, March 2003.

[3] V. J. Ribeiro, R. H. Riedi, J. Navratil, L. Cottrell, and R. G. Baraniuk. pathchirp: efficient available bandwidth estimation for network paths. In *Proc. Passive and Active Measurement Workshop (PAM)*, La Jolla, CA, April 2003.

[4] R. Rejaie, M. Handley, and D. Estrin. Layered quality adap-tation for Internet streaming video. *IEEE J. Selected Areas in Communications,* 18(12):2530-2543, December 2000.

[5] Q. Zhang, Y.-Q. Zhang, and W. Zhu. Resource allocation for multimedia streaming over the Internet. *IEEE Trans. Multimedia*, 3(3):339-355, September 2001.

[6] B.-J. Kim, Z. Xiong, and W. A. Pearlman. Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3-D SPIHT). *IEEE Trans. Circuits and Systems for Video Technology,* 10(8):1374-1387, December 2000.

[7] F. Wu, S. Li, and Y.-Q. Zhang. A framework for efficient progressive fine granularity scalable video coding. *IEEE Trans. Circuits and Systems for Video Technology,* 11(3):301-317, March 2001.

[8] J. Li. Embedded audio coding (eac) with implicit psychoacoustic masking. In *Proc. Int'l Conf. Multimedia,* pages 592-601, Nice, France, December 2002. ACM.

[9] P. de Cuetos and K. W. Ross. Adaptive rate control for streaming stored fine-grained scalable video. In *Proc. Int'l Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDA V)*, Miami Beach, FL, May 2002.

[10] J. Ribas-Corbera, P. A. Chou, and S. Regunathan. A generalized hypothetical reference decoder for H.264/AVC. *IEEE Trans. Circuits and Systems for Video Technology,* 13(7), July 2003.

[11] B. D. O. Anderson and J. B. Moore. *Optimal Control: Linear Quadratic Methods.* Prentice Hall, 1990.

[12] S. Keshav. Packet-pair flow control. http://www.cs.cornell.edu/-skeshav/papers.html.

[13] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *Microsoft Research Technical Report MSR-TR-2001-35*, February 2001

Claims

1. A computer-implemented process for controlling a coding bit rate of streaming media data being transmitted to a client from a server over a computer network, comprising performing the following process actions:

   employing a linear quadratic control technique to establish, on an ongoing basis, a current coding bit rate for the streaming media data, which is estimated to provide a high quality playback of the streaming media data while still keeping a decoder buffer of the client used to receive streaming media data from the server filled to a desired duration level so as to reduce the chance of an underflow condition which would result in an interruption of the playback of the streaming media data by the client;
   using each coding bit rate established to control said rate for the streaming media data.

fuboTV Media Inc.
Exhibit 1002
Page 288

2. The process of Claim 1, wherein the process action of establishing the current coding bit rate comprises the actions of:

determining on a frame-by-frame basis the coding bit rate which is estimated to keep the client buffer level approximately at the desired target level; and
establishing a new coding bit rate only when it is determined the coding bit rate estimate will not change in relation to the last established rate by more than a desired degree.

3. The process of Claim 2, wherein the client buffer level is determined by a leaky bucket model of the streaming media data at the estimated coding bit rate.

4. The process of Claim 3, wherein the process action of determining on a frame-by-frame basis the coding bit rate which is estimated will keep the client buffer approximately at the desired level, comprises an action of identifying a coding bit rate that reduces a difference between an estimated latest anticipated arrival time of the frame under consideration and a prescribed target arrival time.

5. The process of Claim 4, wherein the process action of determining on a frame-by-frame basis the coding bit rate which is estimated will keep the client buffer approximately at the desired level, comprises an action of reducing the change in the coding bit rate to a prescribed degree, using a quadratic cost function.

6. The process of Claim 5, wherein the process action of estimating the latest anticipated arrival time of the frame under consideration, comprises the actions of:

re-estimating the latest anticipated arrival time associated with the frame immediately preceding the frame under consideration if it were to exhibit the current coding bit rate; and
adding said re-estimated time to the current coding bit rate divided by the product of a current instantaneous frame rate and an estimated arrival rate of the frames.

7. The process of Claim 6, wherein the process action of re-estimating the latest anticipated arrival time associated with the frame immediately preceding the frame under consideration comprises adding the actual arrival time of said immediately preceding frame as measured by the client, to an upper bound gap of an encoder buffer of the server for the immediately preceding frame divided by the estimated arrival rate associated with said immediately preceding frame, wherein the upper bound gap of a frame is defined as the number of data bits that the server's encoder buffer can contain over the bits currently contained therein after the frame is fully input into the encoder buffer.

8. The process of Claim 7, wherein the process action of establishing the current coding bit rate is accomplished by the client, and wherein the client estimates the latest anticipated arrival time of the frame under consideration based on information received from the server concerning the state of the encoder buffer.

9. The process of Claim 8, wherein the information received from the server comprises the current coding bit rate of the streaming media data being received by the client which is received as part of the first frame of the sequence of frames exhibiting that coding bit rate.

10. The process of Claim 9, further comprising, an action of estimating the upper bound gap for each frame received by the client, said estimating comprising:

computing what the encoder buffer fullness value would have been after the last received frame was placed therein as the sum of the last computed value of the encoder buffer fullness prior to insertion of the last received frame and the size of that last received frame,

wherein the encoder buffer fullness value prior to insertion of the last received frame associated with the first frame received by the client at the current coding bit rate of the streaming media is assumed to be zero regardless of its actual value,
and wherein the encoder buffer fullness value prior to insertion of the last received frame associated with each subsequent frame received by the client at the current coding bit rate is deemed to be either zero, or the difference between the last computed value for the encoder buffer fullness after the insertion of the last received frame and the coding bit rate divided by the instantaneous frame rate, whichever is larger, wherein the instantaneous frame rate is equal to the reciprocal of a scheduled playback time of the next frame to be received less a scheduled playback time of the last received frame; and

fuboTV Media Inc.
Exhibit 1002
Page 289

computing the upper bound gap associated with the last received frame as the encoder buffer size less the last computed value for the encoder buffer fullness after insertion of the last received frame.

11. The process of Claim 10, wherein the encoder buffer size varies depending on the coding bit rate and is received by the client from the server as a preamble to the streaming media data for all coding bit rates available from the server.

12. The process of Claim 10, further comprising, whenever a frame is received by the client which has a different coding bit rate than the immediately preceding frame, the actions of:

shifting the currently scheduled target arrival time associated with the just received frame by a shift value, wherein said shift value represents a difference between the upper bound gap of the frame and the upper bound gap that would be associated with the frame had it been coded at the immediately preceding coding bit rate;
shifting the currently scheduled target arrival times for future frames such that they, collectively approach over a period of time, and eventually coincide with, the previous target arrival times for those frames; and
shifting each shifted frame by the difference between the encoder buffer fullness value after insertion of the last received frame and the current coding bit rate, divided by the frame rate, and then divided by the estimated arrival rate associated with the last received frame.

13. The process of Claim 12, wherein the upper bound gap that would be associated with the frame had it been coded at the immediately preceding coding bit rate is computed as the encoder buffer size less the last computed value for the encoder buffer fullness value after insertion of the last received frame, wherein the encoder buffer fullness value after insertion of the last received frame is computed as the sum of the last computed value of the encoder buffer fullness value prior to insertion of the last received frame and the size of that last received frame, and wherein the encoder buffer fullness value prior to insertion of the last received frame is computed as either zero, or the difference between the value computed for the encoder buffer fullness after the insertion of the frame received prior to the last received frame and the immediately preceding coding bit rate divided by the instantaneous frame rate associated with the frame received prior to the last received frame, whichever is larger.

14. The process of Claim 8, wherein the information received from the server comprises:

the current coding bit rate of the streaming media data being received by the client which is received as part of the first frame of the sequence of frames exhibiting that coding bit rate; and
the current upper bound gap which is received as part of every frame.

15. The process of Claim 8, wherein the information received from the server comprises:

the current coding bit rate of the streaming media data being received by the client which is received as part of the first frame of the sequence of frames exhibiting that coding bit rate; and
the current upper bound gap associated with the first frame of the sequence of frames exhibiting the current coding bit rate, said current upper bound gap also being received as part of the first frame of said sequence.

16. The process of Claim 15, wherein the information received from the server further comprises a shift value which is received as part of the first frame of the sequence of frames exhibiting a coding bit rate that is different from the immediately preceding frame, said shift value representing a difference between the upper bound gap that would be associated with the immediately preceding frame received by the client had it been coded at the current coding bit rate and the upper bound gap actually associated with the immediately preceding frame coded at the previous coding bit rate.

17. The process of Claim 16, further comprising an action of whenever a frame is received by the client which has a different coding bit rate than the immediately preceding frame, shifting the currently scheduled target arrival time associated with the just received frame by the shift value and shifting the currently scheduled target arrival times for future frames such that they collectively approach over a period of time and eventually coincide with the previous target arrival times for those frames, said shifting of the target values being accomplished so as to prevent the client from attempting to establish new coding bit rates based on the shift in the upper bound gap between the old and new coding bit rates rather than a change in the arrival rate of the frames.

18. The process of Claim 15, further comprising, for each frame received by the client that is not the first frame of the sequence of frames exhibiting the current coding bit rate, an action of estimating the upper bound gap.

fuboTV Media Inc.
Exhibit 1002
Page 290

19. The process of Claim 6, wherein computing the estimated arrival rate of the frames of the streaming media data comprises:

computing, on a packet-by-packet basis, the product of the estimated arrival rate computed for the immediately preceding packet to the currently received packet and a first fractional weighting factor, added to the product of the instantaneous arrival rate of the currently received packet and a second fractional weighting factor, wherein at least one of the fractional weighting factors is not a constant but instead based on the time between the packets.

20. The process of Claim 5, wherein the process action of identifying the coding bit rate that reduces the difference between the estimated latest anticipated arrival time of the frame under consideration and the prescribed target arrival time, while at the same time reducing the change in the coding bit rate to a prescribed degree, comprises an action of choosing the prescribed target arrival time for the frame under consideration so as to be a prescribed amount of time earlier than a scheduled playback time for that frame.

21. The process of Claim 20, wherein the process action of choosing the prescribed target arrival time for the frame under consideration comprises an action of making said prescribed amount of time large enough after a startup period that network jitter, delays and throughput changes which may cause the actual arrival time of the frame to be after its target arrival time do not result in the frame arriving after its scheduled playback time.

22. The process of Claim 21, wherein the process action of choosing the prescribed target arrival time for the frame under consideration further comprises making the target arrival time closer to the playback time during the startup period than afterwards for those frames arriving during that period to assist in reducing a startup delay between when the streaming media data begins to be received by the client and when the first frame is played back.

23. The process of Claim 22, wherein the process action of choosing the prescribed target arrival time for the frame under consideration further comprises setting the target arrival time for each frame of the streaming media using a logarithmic target schedule.

24. The process of Claim 22, wherein the process action of choosing the prescribed target arrival time for the frame under consideration further comprises setting the target arrival time for each frame of the streaming media using a two-piece linear target schedule wherein the difference between the target arrival time and the playback time for a frame arriving during the startup period increases linearly to a prescribed amount of time that is large enough to account for said network jitter, delays and throughput changes after which the difference remains substantially constant.

25. The process of Claim 22, further comprising a process action of reducing the startup delay, said reducing action comprising beginning playback of the first frame of the streaming media data as early as when a decoder buffer of the client has a minimum amount of data therein that is needed to ensure an underflow condition does not occur at an initial coding bit rate.

26. The process of Claim 25, wherein the process action of beginning playback of the first frame of the streaming media data as early as when the decoder buffer of the client has a minimum amount of data therein, comprises the actions of:

obtaining an initial encoder buffer fullness value associated with said first frame, wherein the initial encoding buffer fullness value is defined as the size of the encoder buffer less the size of the first frame and less the upper bound gap associated with the first frame;
establishing said minimum amount of data as the encoder buffer size associated with the initial coding bit rate less an initial encoder buffer fullness.

27. The process of Claim 26, wherein the initial coding bit rate, initial encoder buffer fullness, and encoder buffer size are received from the server as a preamble to the streaming media data.

28. The process of Claim 26, wherein the process action of reducing the startup delay further comprises setting the initial coding bit rate to a level that is less than the anticipated arrival rate of the streaming media data at the client since said minimum amount of data decreases as the arrival rate increases relative to the coding bit rate and the startup delay is substantially said minimum amount of data divided by the arrival rate so if the coding bit rate is less than the arrival rate, said minimum amount of data required may be smaller and the startup delay may be less.

fuboTV Media Inc.
Exhibit 1002
Page 291

29. The process of Claim 5 wherein the process action of establishing the current coding bit rate comprises an action of initiating the establishment of coding bit rates only after a startup period, wherein said startup period is defined as the period of time prior to the first instance of an estimated latest arrival time of a frame under consideration computed for the initial coding bit rate being earlier than a target arrival time for that frame.

30. The process of Claim 5, wherein the prescribed degree to which the change in the coding bit rate is reduced varies depending on whether the coding bit rate increased, in which case the prescribed degree to which any future change in the coding bit rate is reduced is made greater in comparison to the prescribed degree in cases where the last change in the coding bit rate decreased the rate.

31. The process of Claim 3, wherein the process action of establishing a current coding bit rate for the streaming media data, comprises, in the case where the determined coding bit rate represents an increase from the immediately previous coding bit rate, establishing a new coding bit rate only if the new rate does not exceed the current estimated arrival rate of the frames.

32. The process of Claim 3, wherein the process action of establishing a current coding bit rate for the streaming media data, comprises, in the case where the determined coding bit rate represents an increase from the immediately previous coding bit rate which exceeds the current estimated arrival rate of the frames, establishing a new coding bit rate only if the current client buffer exceeds the desired level by an amount that it is estimated by the client will not be expended at the higher coding bit rate prior to the passing of a prescribed period of time.

33. The process of Claim 3, wherein the process action of establishing a current coding bit rate for the streaming media data, comprises, in the case where the determined coding bit rate represents a decrease from the immediately previous coding bit rate, establishing a new coding bit rate even if it represents a significant departure from the immediately previous coding bit rate.

34. A computer-implemented process for controlling a coding bit rate of streaming media data being transmitted to a client from a server over a computer network, comprising using the server to perform the following process actions:

generating a streaming media data stream which exhibits a coding bit rate supported by the server, and which after an initial startup period exhibits a coding bit rate related to a rate requested by the client;
computing parameters indicative of the state of an encoder buffer of the server, said encoder buffer being used to stage portions of the streaming media data prior to transmission to the client; and
providing the encoder buffer state parameters and the streaming media data stream to the client, whereby said parameters are usable by the client to establish, on an ongoing basis, a current coding bit rate for the streaming media data, which is estimated will provide a high quality playback of the streaming media data while still keeping a decoder buffer of the client used to receive streaming media data from the server filled to a desired duration level so as to reduce the chance of an underflow condition which would result in an interruption of the playback of the streaming media data by the client.

35. The process of Claim 34, wherein the process action of computing parameters indicative of the state of an encoder buffer, comprises an action of, for each coding bit rate supported by the server, computing a set of parameters defining an initial state of the encoder buffer as it would exist if a data stream corresponding to the coding bit rate where streamed therefrom, wherein said encoder buffer parameters comprise the coding bit rate associated with the data stream, the size of the encoder buffer employed with the coding bit rate of the data stream, and a value indicative of the initial encoder buffer fullness exhibited at the coding bit rate of the data stream.

36. The process of Claim 35, wherein the process action of providing the encoder buffer state parameters and the streaming media data stream to the client, comprises an action of providing the encoder buffer parameters computed for each data stream exhibiting one of the supported coding bit rates to the client as part of a preamble sent to the client prior to sending the streaming media data.

37. The process of Claim 35, wherein the size of the encoder buffer employed with the coding bit rate of the data stream is chosen so as to be the minimum size buffer that will still contain the data stream at any point in the streaming process given the coding bit rate and the initial encoder buffer fullness.

38. The process of Claim 35, further comprising a process action of the server computing an upper bound gap for each frame of the streaming media data generated for each coding bit rate supported by the server, wherein the upper

fuboTV Media Inc.
Exhibit 1002
Page 292

bound gap is defined as the number of data bits that the server's encoder buffer can contain over the bits currently contained therein after the just-generated frame is fully input into the encoder buffer.

39. The process of Claim 38, further comprising a process action of providing the upper bound gap computed for the first frame of a sequence of frames exhibiting a coding bit rate that is different from that associated with the frame generated immediately before the first frame of said sequence of frames, to the client along with an indication of the coding bit rate associated with the sequence of frames, said gap value and coding bit rate being provided with said first frame of the sequence.

40. The process of Claim 39, wherein the client estimates the upper bound gap for frames provided by the server that do not include the associated gap value.

41. The process of Claim 40, wherein the client employs the encoder buffer parameters and the upper bound gap values to determine on a frame-by-frame basis the coding bit rate that will keep the client buffer approximately at the desired level, said determining comprising identifying a coding bit rate that reduces a difference between an estimated latest anticipated arrival time of the frame under consideration and a prescribed target arrival time.

42. The process of Claim 39, further comprising the process actions of:

the server computing a shift value for each frame representing the first frame of the sequence of frames exhibiting a coding bit rate that is different from the frame generated immediately before this first frame of the sequence, said shift value representing a difference between the upper bound gap that would be associated with the frame generated immediately before the first frame of the sequence had it been encoded at the current coding bit rate and the upper bound gap actually associated with the frame generated immediately before the first frame of the sequence encoded at the previous coding bit rate; and
providing the shift value computed for the first frame of each sequence of frames exhibiting a coding bit rate that is different from the frame generated immediately before the first frame of the sequence, to the client with the data comprising this first frame.

43. The process of Claim 42, wherein whenever a frame is provided to the client which has a different coding bit rate than the immediately preceding frame, the currently scheduled target arrival time associated with the just received frame is shifted by the shift value provided with the just-received frame and the currently scheduled target arrival times for future frames are shifted such that they collectively approach over a period of time and eventually coincide with the previous target arrival times for those frames.

44. The process of Claim 38, wherein a limited number of coding bit rates are supported by the server, and wherein the process action of generating a streaming media data stream which exhibits a coding bit rate supported by the server, and which after an initial startup period exhibits a coding bit rate related to a rate requested by the client, comprises the actions of:

(a) receiving a coding bit rate request from the client;
(b) finding a supported coding bit rate that is equal to, or if none are equal, the closest smaller rate to, the requested coding bit rate;
(c) whenever a supported coding bit rate is found that is a lower rate than the coding bit rate associated with the last generated frame of the streaming media data, generating all future frames at that supported rate; and
(d) whenever a supported coding bit rate is found that is a higher rate than the coding bit rate associated with the last generated frame of the streaming media data, determining if a difference, between the upper bound gap associated with the last generated frame had it been encoded at the supported coding bit rate found and the upper bound gap associated with that frame encoded at the current coding bit rate, is less than or equal to a maximum allowable difference value,

whenever the difference is less than the maximum allowable difference value, generating all future frames at the found supported rate, and
whenever the difference is not less than the maximum allowable difference value, finding the next lower supported coding bit rate and repeat actions (c)-(d).

45. The process of Claim 44, wherein the maximum allowable difference value is received from the client along with the coding bit rate request.

fuboTV Media Inc.
Exhibit 1002
Page 293

46. The process of Claim 45, wherein the client chooses the maximum allowable difference value such that a latest anticipated arrival time associated with a frame provided by the server immediately prior to a frame under consideration by the client had it been coded at the requested coding bit rate is no more than a prescribed fraction of the way from that frame's target arrival time to its playback deadline.

fuboTV Media Inc.
Exhibit 1002
Page 294

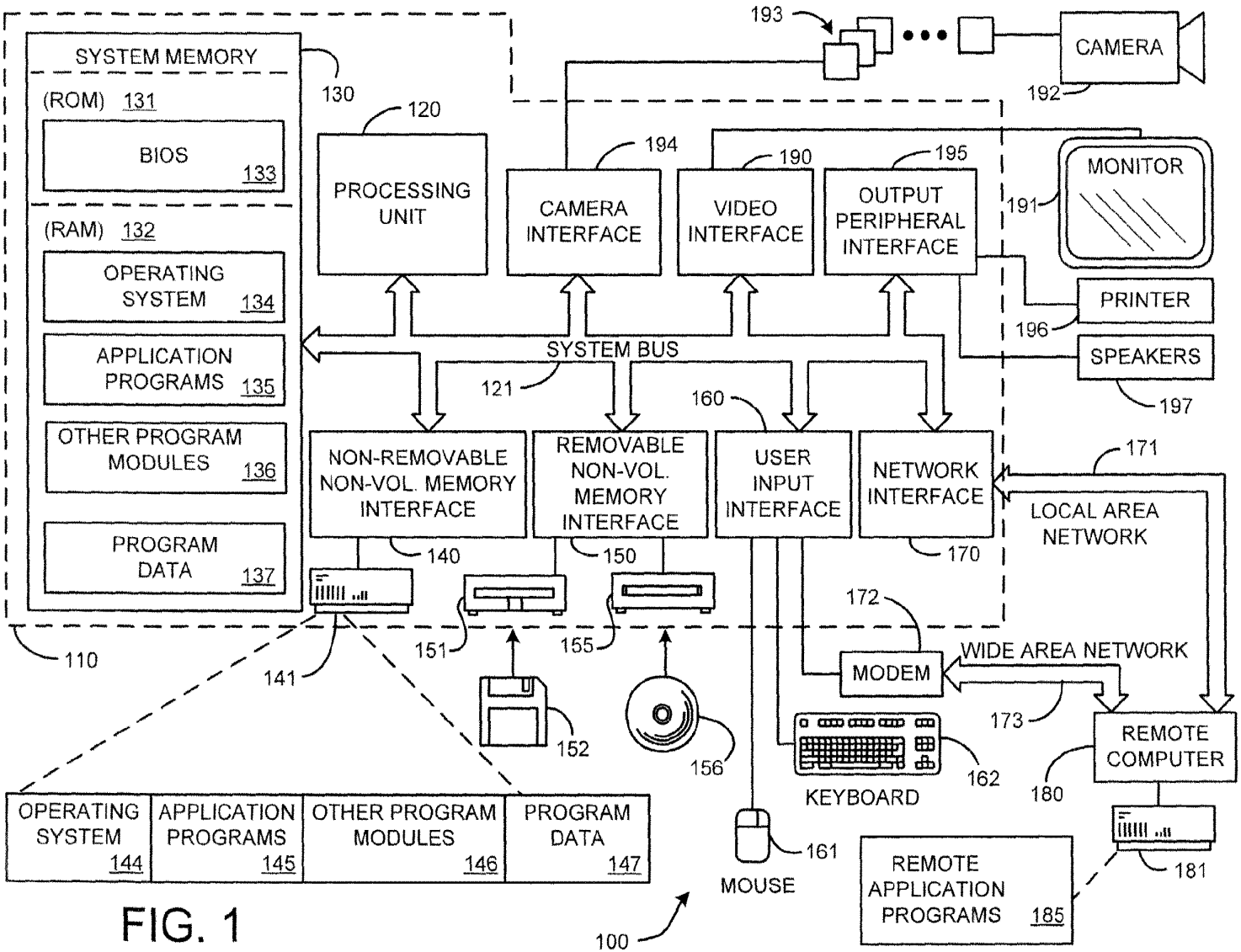


FIG. 1

fuboTV Media Inc.
Exhibit 1002
Page 295

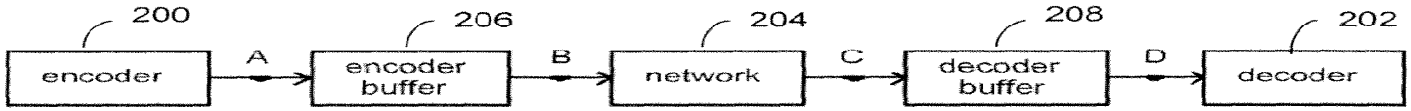


FIG. 2

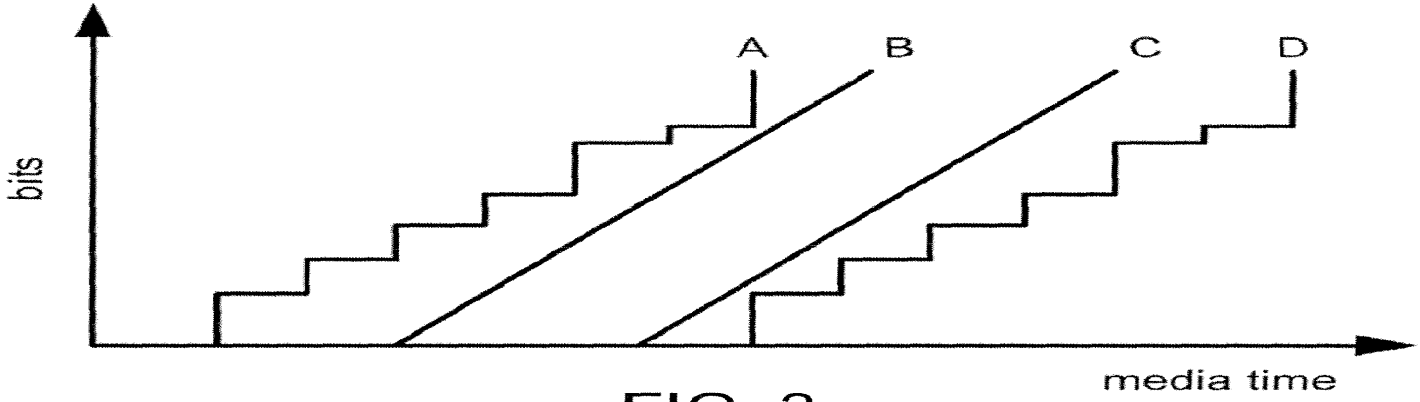


FIG. 3

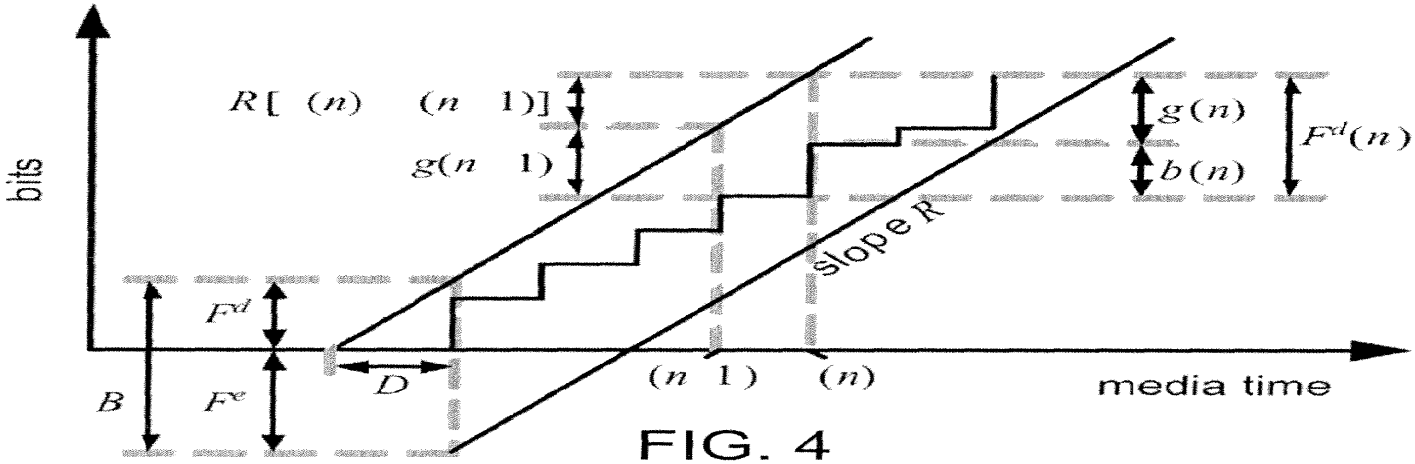


FIG. 4

fuboTV Media Inc.
Exhibit 1002
Page 296

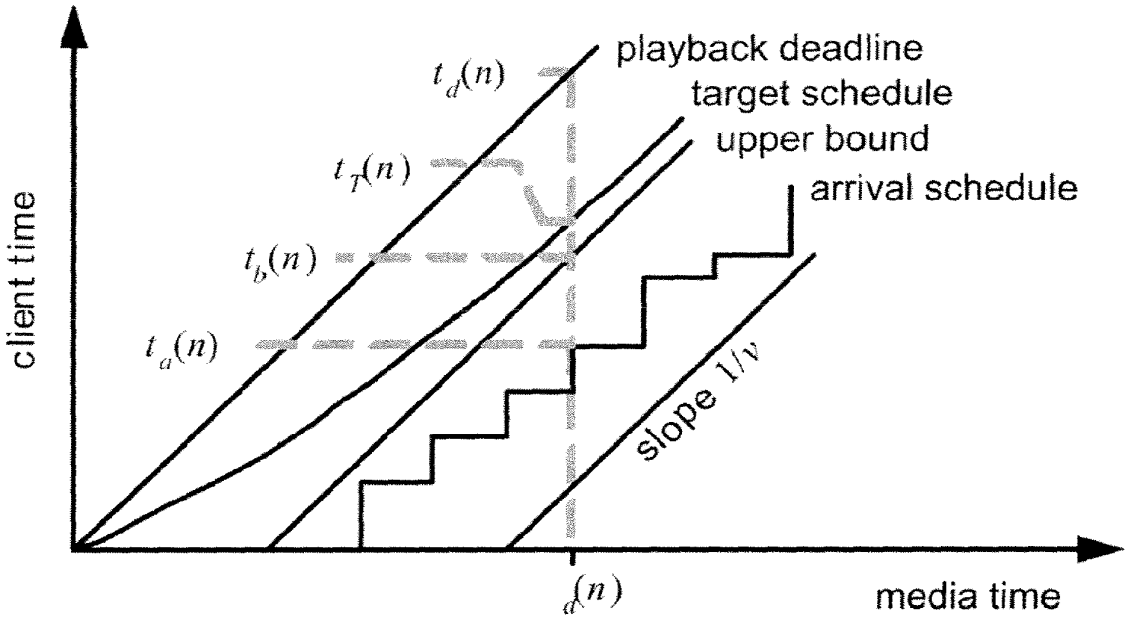


FIG. 5

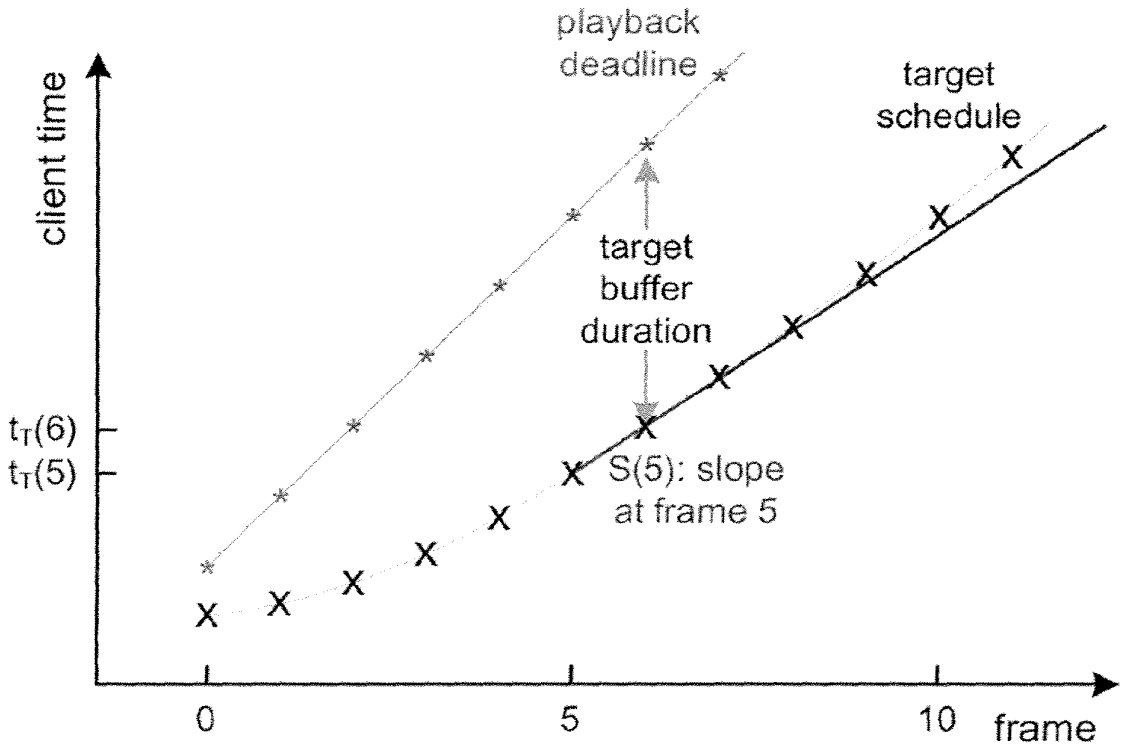


FIG. 6

fuboTV Media Inc.
Exhibit 1002
Page 297

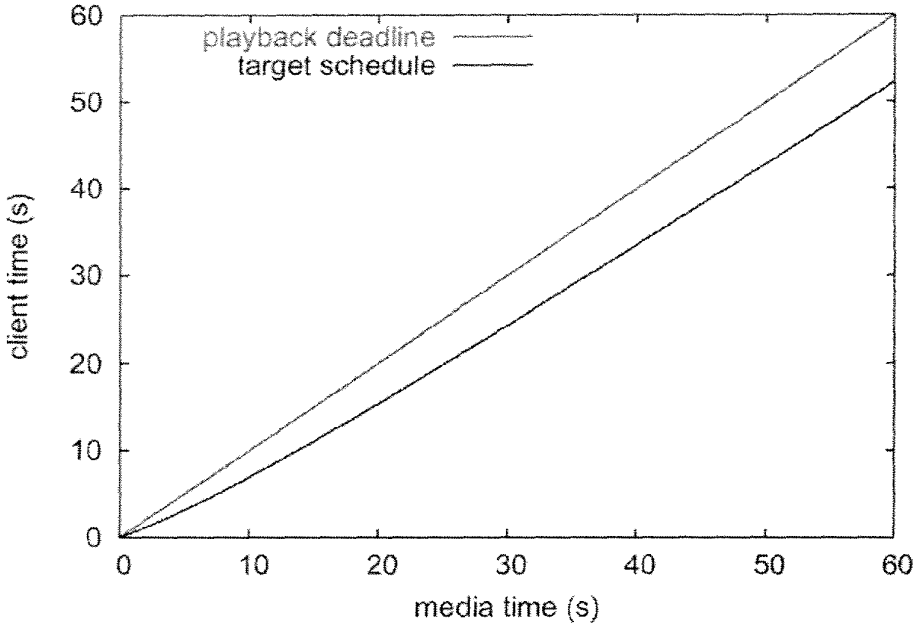


FIG. 7(a)

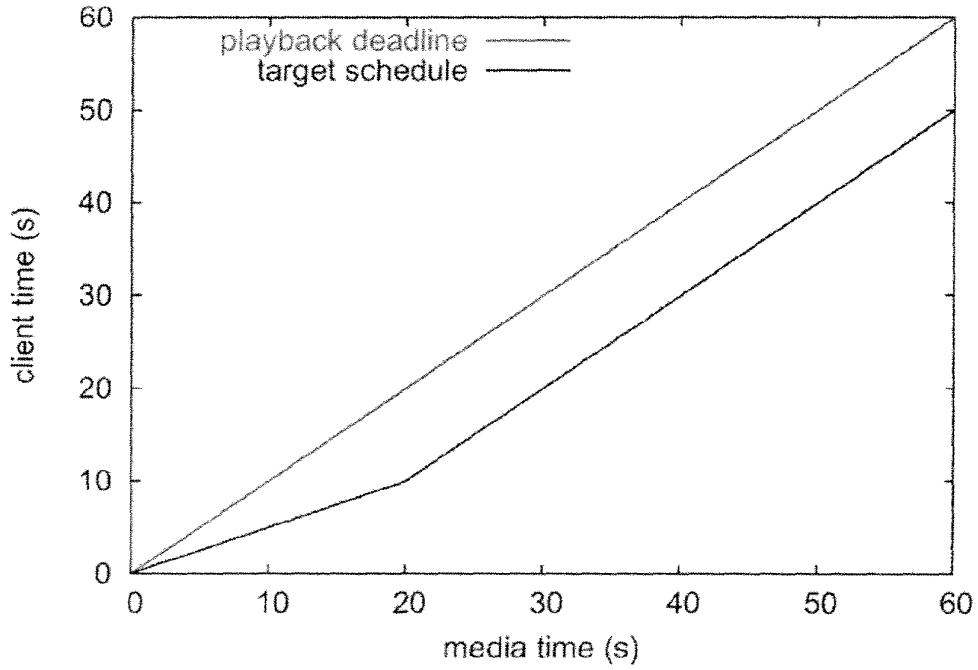


FIG. 7(b)

fuboTV Media Inc.
Exhibit 1002
Page 298

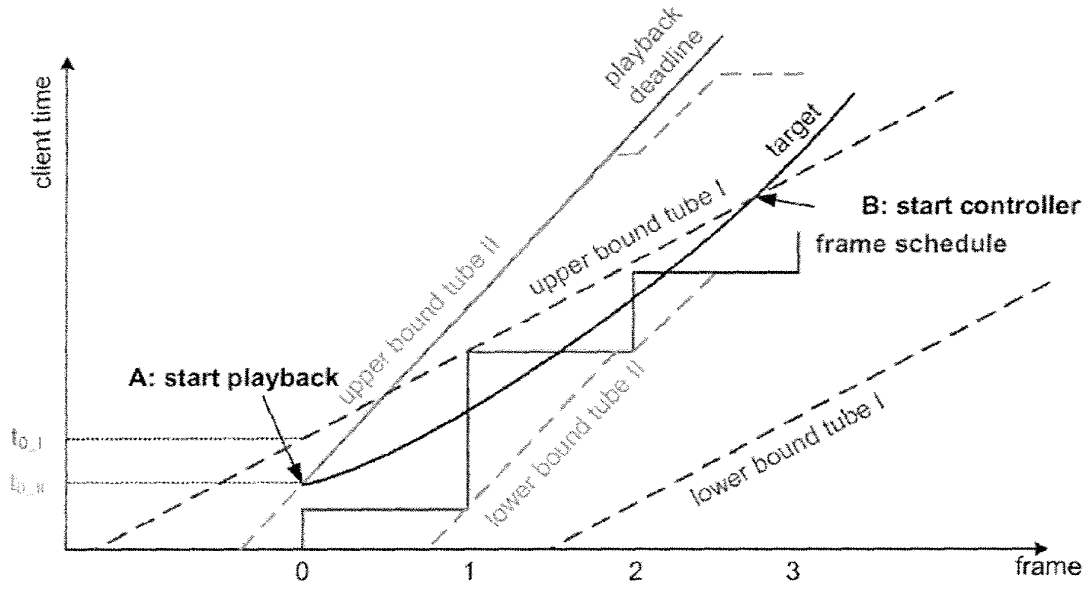


FIG. 8

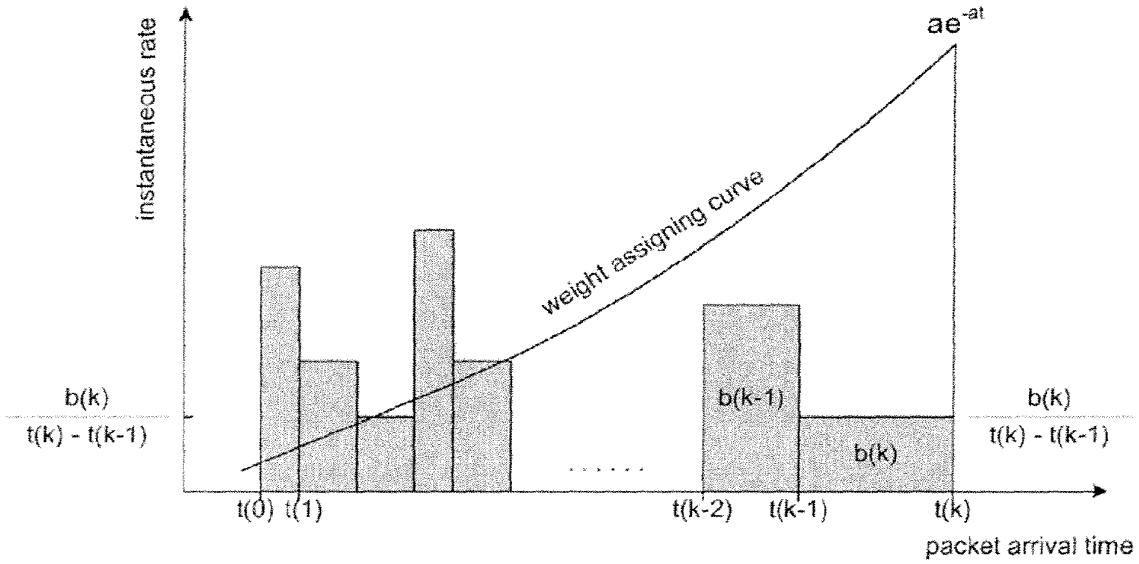


FIG. 9

fuboTV Media Inc.
Exhibit 1002
Page 299

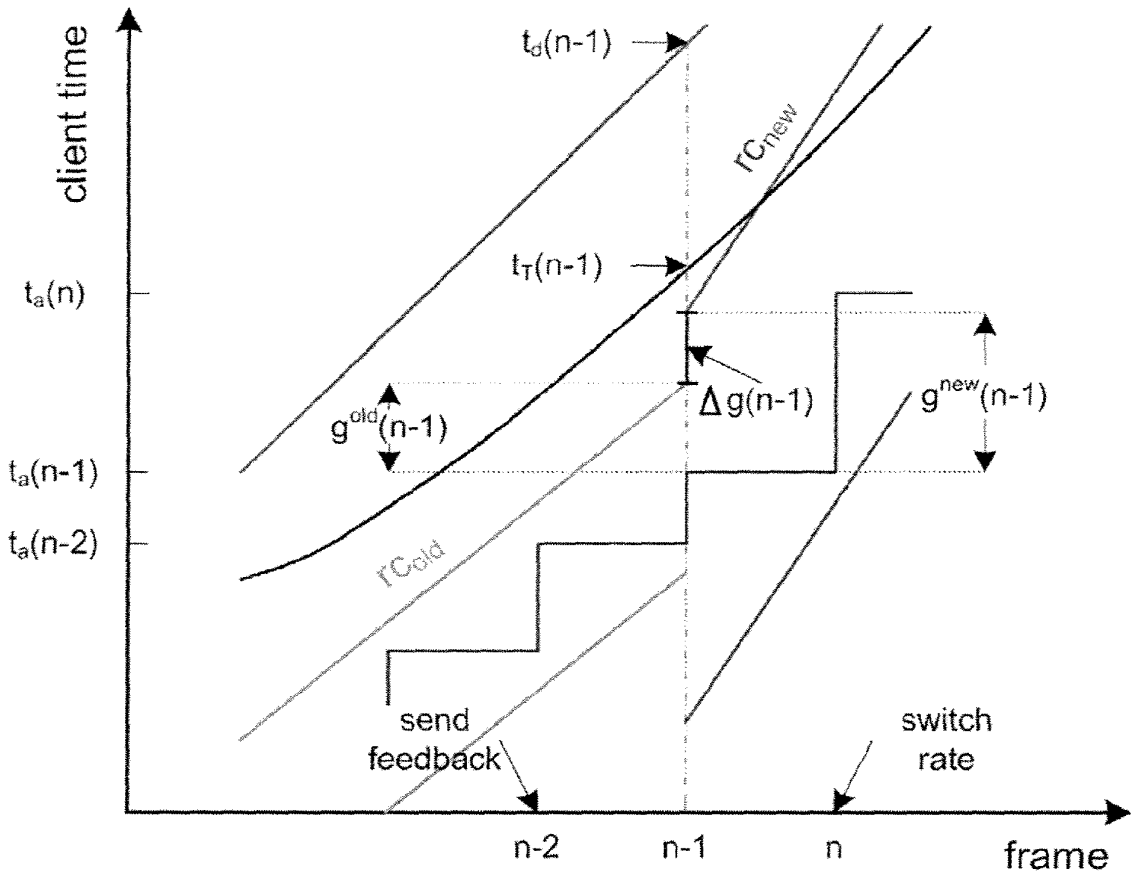


FIG. 10

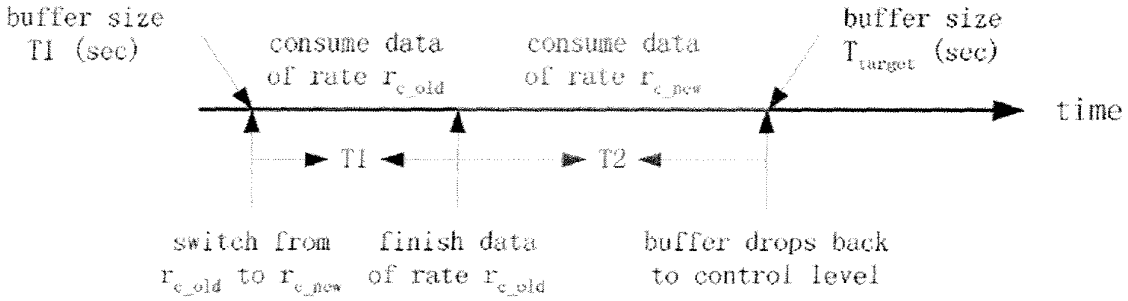


FIG. 12

fuboTV Media Inc.
Exhibit 1002
Page 300

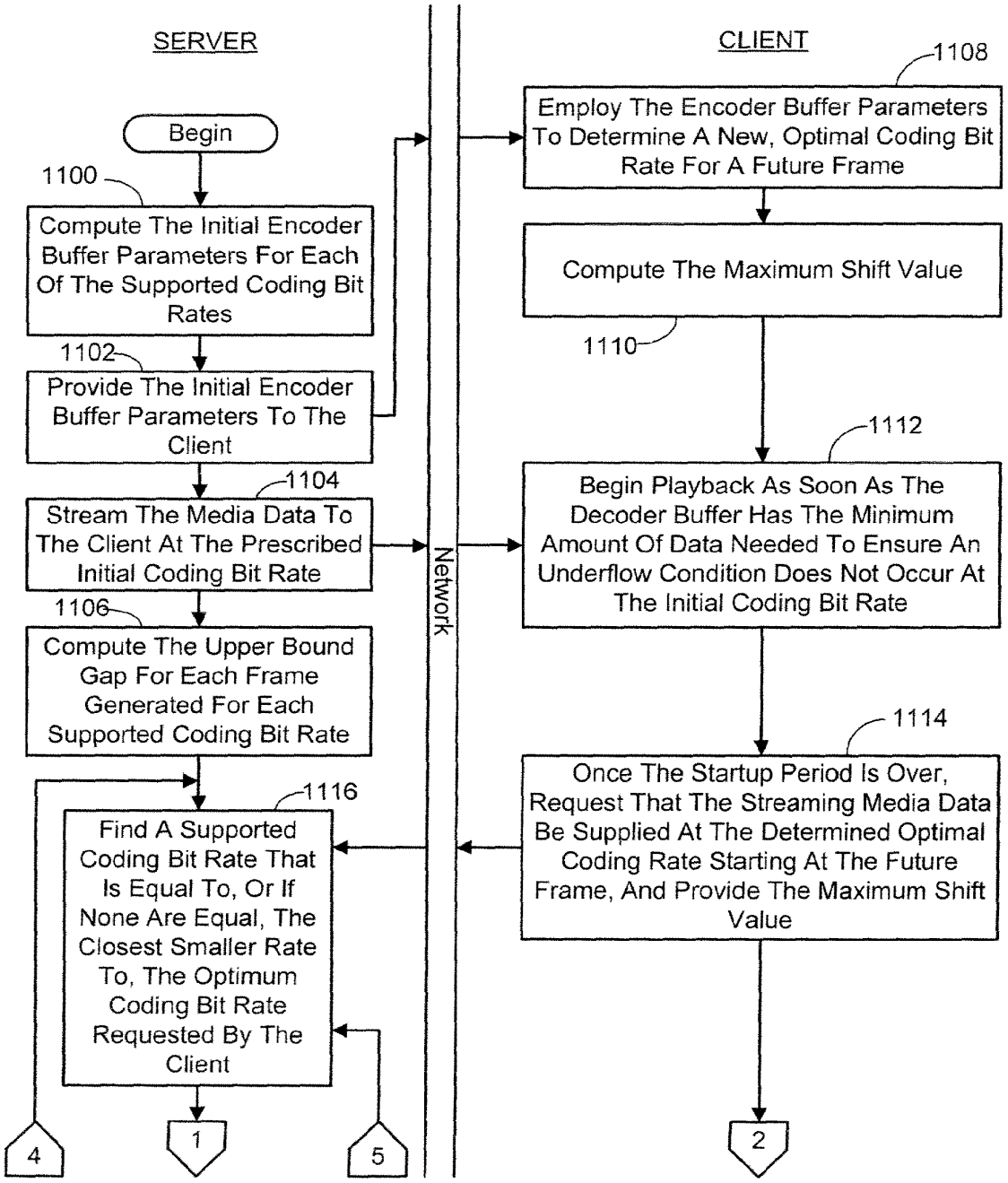


FIG. 11A

fuboTV Media Inc.
Exhibit 1002
Page 301

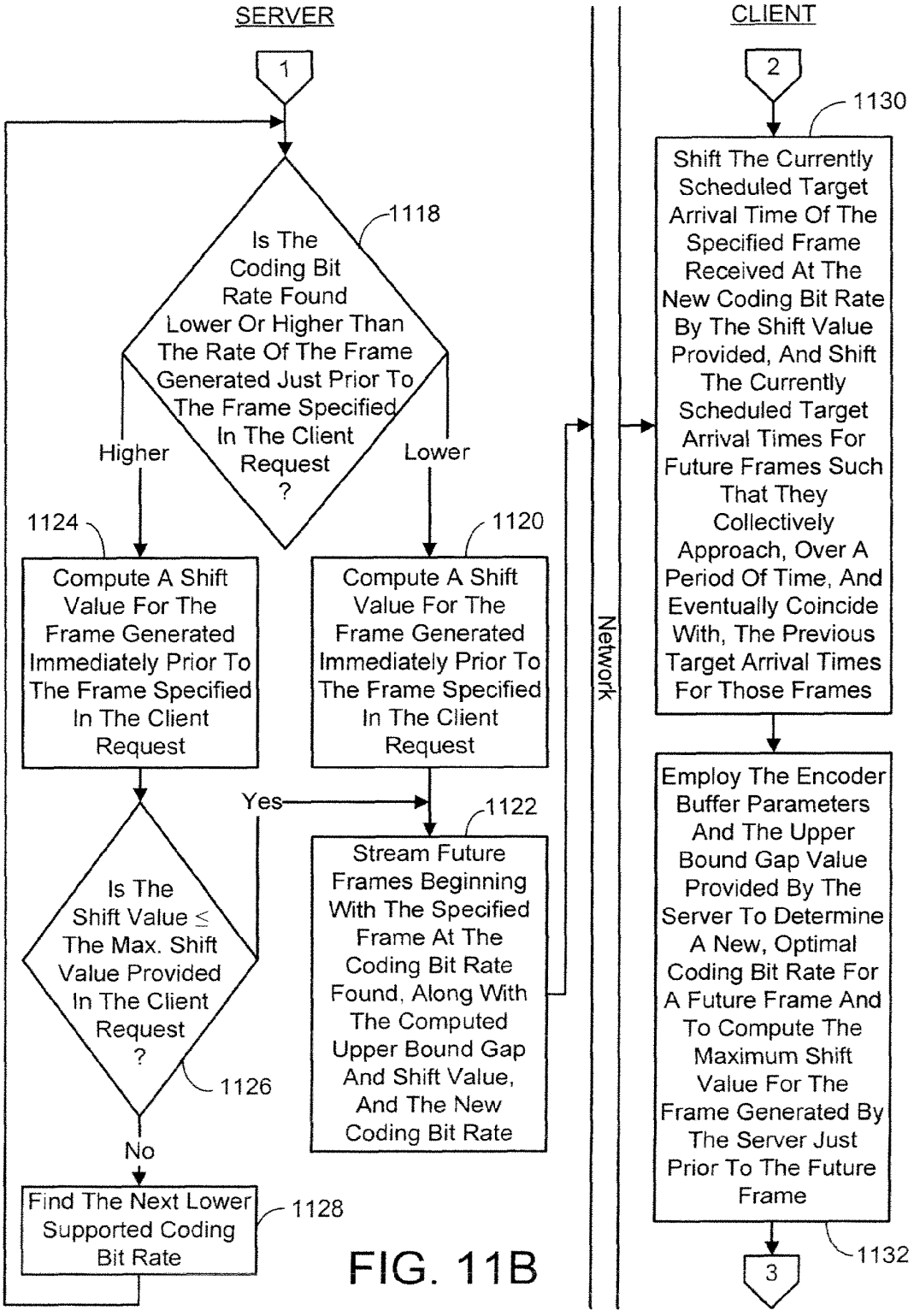


FIG. 11B

fuboTV Media Inc.
Exhibit 1002
Page 302

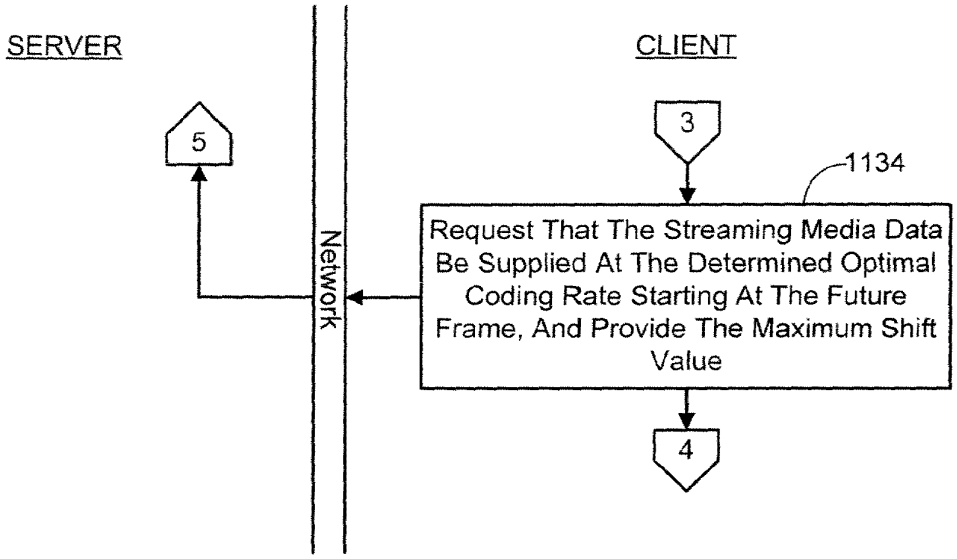


FIG. 11C

fuboTV Media Inc.
Exhibit 1002
Page 303

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 777 969 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
25.04.2007 Bulletin 2007/17

(51) Int Cl.:
H04N 7/46 *(2006.01)*

(21) Application number: **05256302.0**

(22) Date of filing: **10.10.2005**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IS IT LI LT LU LV MC NL PL PT RO SE SI SK TR
Designated Extension States:
AL BA HR MK YU

(71) Applicant: **BRITISH TELECOMMUNICATIONS public limited company**
London EC1A 7AJ (GB)

(72) Inventor: **The designation of the inventor has not yet been filed**

(74) Representative: **Lloyd, Barry George William et al**
BT Group Legal
Intellectual Property Department
PP C5A
BT Centre
81 Newgate Street
London EC1A 7AJ (GB)

(54) Adaptive video transmission with variable frame rate

(57) Video signals are sent via a network, such as an internet protocol network, the bandwidth of which varies with time. The amount of data to be sent is controlled by varying the frame rate. This is achieved by (a) measuring the transmission capacity to the receiver; (b) determining the rate of change of transmission capacity; and (c) adjusting the transmitted video frame rate as a function of the said rate of change.

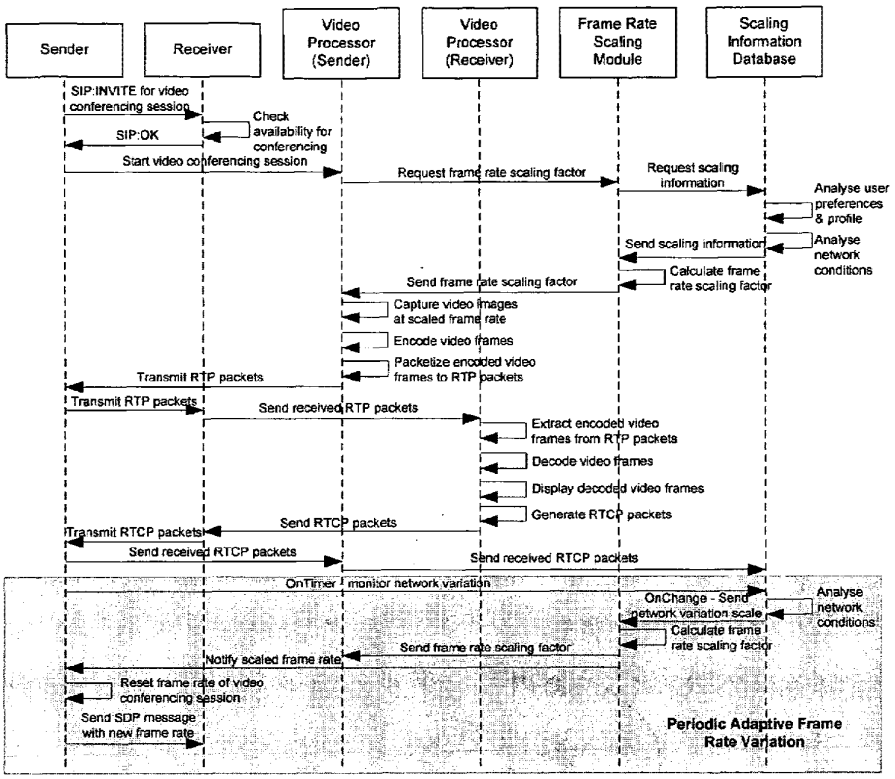


Figure 3 Sequence Diagram of Adaptive Video Frame Rate Application

Printed by Jouve, 75001 PARIS (FR)

EP 1 777 969 A1

fuboTV Media Inc.
Exhibit 1002
Page 304

Description

[0001] The present invention is concerned with the transmission of video signals via a network.

[0002] In the future, the next generation communication environment is envisaged as allowing users to roam freely and to communicate with each other at any time and any place [1]. Client devices with different capabilities, which range from low power mobile phones to high performance workstations, may be used for communications. Users will be interconnected by networks with various capacities, with packet loss rates that range from almost lossless transmission to quite error prone environments. The rapid development of the client device industry and the proliferation of network access technologies is expected to result in the extensive growth of bandwidth-intensive real-time multimedia internetworking applications (e.g. voice over IP, video streaming, video conferencing and so on).

[0003] Real-time multimedia transmission over a variable bandwidth network environment requires a strict guarantee of network bandwidth and of several other quality of service (QoS) parameters, such as end-to-end delay, delay variation and packet loss. Unfortunately, most of the current QoS mechanisms in existing networks (such as wireless and mobile networks) do not provide such guarantees. In dynamic and unpredictable network environments, due to diverse forms of mobility and interference at the air interface, it gives rise to varying bandwidth and dynamically changing error rates [2]. This further intensifies the challenge for consistent provision of QoS. Traditional QoS mechanisms (e.g. Integrated Services and Differentiated Services), which are typically designed for wired networks, assume a fixed network topology and a fixed amount of available resources. These are inadequate to satisfy users' QoS requirements or to utilize the system resources efficiently, especially in bandwidth-limited and error prone network environments.

[0004] In order to solve this, adaptive multimedia applications are proposed. By implementing adaptive applications, media quality can, in the case of bandwidth variation of network, be adapted appropriately. The bandwidth of the media stream can be scaled down to provide a degraded quality at the destination. In this invention, we are particularly interested in the implementation of video applications. Adaptive video applications use protocols such as Real-Time Streaming Protocol (RTSP) and Real-Time Protocol (RTP), to adapt to changing network conditions using a number of techniques. These techniques include hierarchical encoding, efficient compression, bandwidth smoothing, rate shaping, error control, and adaptive synchronization. Feedback from lower networking layers can be used by the application layer and this helps to match the transmission to the available bandwidth as well as to increase the robustness of the video transmission. However, the perceived quality of video applications is being neglected. Variations of frame rates are proposed in some of these adaptive mechanisms [3] [4]. However, abrupt changes of frame rates can annoy users and hence have a severe impact on the perceived quality of such video applications.

[0005] QoS provisioning is an important issue in real-time internetworking video applications. The dramatic fluctuation of network conditions further intensifies the problems of QoS provisioning. Under variable bandwidth network environments (such as users are moving from one network to another with different bandwidth capacity or network is being congested by heavy traffic), characteristics of video applications need to be adjusted in order to fit the requirements of the current operating network and also maintain the perceived QoS.

[0006] An IBM document on QoS policies [5] clearly states that "the priority for smooth video is higher than the priority for frame quality". It points out that the importance of smooth video is even more significant when users are watching high motion videos (such as sports programmes). Traditional adaptive frame rate variation mechanisms in video applications, which adjust the video requirements according to the rate supportable by the network, attempt to keep the quality of video constant by varying the frame rate. In these mechanisms, the perceived quality of video application is often degraded as frame rates are reduced to a much lower rate or being increased to a much higher rate in order to fit network conditions. The abrupt changes of frame rates (from higher rate to lower rate and vice versa), which cause jerkiness and unsteady video displays, annoy users who are expecting a smooth and continuous flow of video frames in real-time video applications. The users are able to notice the sudden change of frame rate in video transmission. For example, a video application may be sending video at 25 frames per second (fps) during normal network condition and frame rate is reduced to 8fps when network congestion is detected; the frame rate is then increased to 15fps again when network conditions improve. This drastic fluctuation of frame rates will affect the smoothness of the video display and hence have a severe impact on perceived quality.

[0007] References:

1. T. Takebayashi et al, Approaches to User-Centric Computing System for Ubiquitous Solutions, Fujitsu Sci. Tech. J., 39, 2, December 2003, p.261-269.

2. M. Naghshineh et al, End-to-End QoS Provisioning in Multimedia Wireless/Mobile Networks Using an Adaptive Framework, IEEE Communication Magazine November 1997.

3. J. Kim et al, TCP-Friendly Internet Video Streaming Employing Variable Frame-Rate Encoding and Interpolation, IEEE Transaction on Circuits and Systems for Video Technology, Vol.10, NO.7, October 2000.

fuboTV Media Inc.
Exhibit 1002
Page 305

[4] W. Guetz et al., U.S. Patent 6,091,777.

[5] IBM. Functions of mobile multimedia QoS control. http://www.trl.ibm.com/projects/mmqos/system_e.htm.

[0008] According to the present invention there is provided a method of transmitting video over a network to a receiver, comprising repeatedly

(a) measuring the transmission capacity to the receiver;
(b) determining the rate of change of transmission capacity;
(c) adjusting the transmitted video frame rate as a function of the said rate of change.

[0009] Other aspects of the invention are set out in the claims.

[0010] Some embodiments of the invention will now be described, with reference to the accompanying drawings.

[0011] The basic system operational view of our implementation is depicted in Figure 1. The figure shows one-way communication (sender-to-receiver), however, it can be implemented as a two-way communication application (client-to-client). Session Initiation Protocol (SIP) can be used to start a communication session between two or more clients. Real-Time Protocol (RTP) and

[0012] Real-Time Control Protocol (RTCP) can be used as underlying transport layer to send data packets and also control packets. In this implementation, Session Description Protocol (SDP) signalling can be used to reset the parameters of a video conferencing session and send them to the receiving end.

[0013] The video processor, which involves capture/display, encoder/decoder and transmitter/receiver, is mainly for media processing. The adaptation mechanism appears as a standalone module in the application. A scaling information database extracts and stores network statistical information (monitored periodically using a timer) and user profiles (set by users). The frame rate scaling module is the core mechanism which calculates a frame rate scaling factor based on information obtained from information database. A scaled frame rate will then be fed to the video processor for setting new video parameters.

[0014] Figure 2 illustrates how the different parameters influence the frame rate in the case of fluctuation in the network . To calculate the frame rate scaling factor, several parameters need to be taken into consideration. These parameters may include network statistics, RTCP feedback statistics and the user profile. The main contributor to the scaling factor is the network statistics, where the changes with time (Δbandwidth/Δt) are tracked.

[0015] The sequence diagram (Figure 3) shows the activity flows of our implementation. The periodic adaptive frame rate variation mechanism (shaded portion) is applied to avoid the situation that frame rate variation is only incurred when network congestion is detected. A timer is initiated for periodic network condition monitoring at the client end. The frame rate will be scaled based on the scaling factor extracted from scaling information as discussed previously. This will improve the perceived quality by giving a gradual change of frame rate and smooth playbacks of video throughout entire communication.

[0016] The control mechanism essentially comprises the following steps:

(a) measurement, at intervals, of the network bandwidth $B(t_i)$;
(b) determination of the gradient $\{B(t_i) - B(t_{i-1}))/(t_i - t_{i-1})$;
(c) determination of a scaling factor *M,* as a function of the gradient *m*;
(d) determination of a new frame rate, multiplying the existing rate by the scaling factor - subject however to the rate remaining within a range determined by preset maximum and minimum values..

[0017] In more detail:

(a) Bandwidth could be measured by any one of a number of conventional methods. If the communication session is peer to peer within a similar subnet, say, in an enterprise scenario, the backhaul link of the connecting equipment could expose its bandwidth availability through its SNMP agent residing in that particular equipment. This could be queried via SNMP calls. On the other hand, in an environment when there are several nodes in the path of its communication, an application level implementation of measuring its delay and thus bottleneck within that particular path could be used. This would suggest bandwidth availability.
Clearly it is important, in terms of reducing the risk of large changes in frame rate, to measure the bandwidth (and adjust the frame rate) at frequent intervals - that is to say, frequent in relation to the rate of change of network conditions. A typical rate might be every 10 seconds. It would be possible to measure the bandwidth once per frame, but this is not preferred, in that it involves additional overheads. In theory the bandwidth could be measured more often than once per frame, but naturally the frame rate adjustment cannot occur more often than once per frame.
(b) The gradient is determined by the formula given above. The use of the gradient rather than simply the absolute value of the bandwidth is significant because it introduces an element of prediction into the assessment of bandwidth availability.
(c) The determination of the scaling factor *M* as a function of the gradient *m* should be such that

fuboTV Media Inc.
Exhibit 1002
Page 306

If *m* is large & positive, multiplier is >1;
If *m* is small & positive, multiplier is > 1 (but smaller than when *m* is large);
If *m* is 0, multiplier is 1;
If *m* is small & negative, multiplier is <1;
If m is large & negative, multiplier is <1 (but smaller than when *m* is small and negative.
For example, a possible relationship for the situation where the bandwidth gradient is expected to vary over the range -50 to 50 kbit/s/s is: that M varies between 0.4 and 2.5. A simple, piecewise-linear implementation would be to specify that for m>0: M = 1 + 0.03 m; for m=0: M = 1; and for m<0: M = 1 + 0.012 m. However we prefer a nonlinear relationship in which frame rate changes are more extreme at the ends of the scale. Such a relationship would best be implemented by the use of a look-up table.
(d) A typical limit range for frame rate might be from 10 to 25 frames per second. Thus if the existing frame rate is f_i then the new frame rate will be

$$f_{i+1} = \mathrm{Max}\{\mathrm{Min}\{Mf_i, 25\}, 10\}.$$

[0018] The user preferences referred to earlier could mean constraints on the level of adaptation that the user is prepared to live with in a congested network situation, i.e reaction time, image resolution resizing, colour depth. This could influence the extent for the need to switch to a lower frame rate.

[0019] In order to achieve better perceived quality of real-time internetworking video application in variable bandwidth network environments, degradation of video quality needs to be occur gracefully. In the method we have described, an adaptive mechanism, which resides in the video application, allows smooth variation of frame rates under variable bandwidth network environments. It varies frame rates gradually and hence minimises the jerkiness and flickering of video displays. A gradual degradation or upgrade of video quality gives an illusion to users that there is only little change in video quality throughout the communication period. The variation incurred is less aggressive and significant as compared with traditional frame rate variation mechanisms.

[0020] The method improves the perceived QoS of real-time internetworking video applications by implementing smooth variation frame rate mechanism of adaptive video application. An adaptation mechanism resides in a real-time internetworking multimedia application to adapt video quality dynamically to changes in network conditions. Network conditions are absorbed and analysed from time to time and network statistical analysis is then fed back to the video processor of the application. The application is then triggered to set the frame rate of the video to a requested rate based on the analysis of network conditions. The frame rate changes dynamically when network conditions vary - but not only when network congestion is detected. The degree of variation in frame rate depends on the degree of variation in network conditions (such as network utilization) and the variations of frame rates occur throughout the communication session. Hence, it prevents sudden changes in frame rate of the video transmission and helps to minimise the jerkiness of video displays under variable bandwidth conditions. User preferences and profiles may also be taken into account when frame rate is being adjusted. This further improves user perceived QoS of the application.

Claims

1. A method oftransmitting video over a network to a receiver, comprising repeatedly

   (a) measuring the transmission capacity to the receiver;
   (b) determining the rate of change of transmission capacity;
   (c) adjusting the transmitted video frame rate as a function of the said rate of change.

2. A method according to claim 1 in which the frame rate is adjusted to equal the existing frame rate multiplied by a factor which is a monotonically increasing function of rate of change of transmission capacity.

3. A method according to claim 2 in which the factor is a nonlinear function of rate of change of transmission capacity.

4. A method according to claim 3 in which the factor is obtained by means of a look-up table.

5. A method according to any one of the preceding claims in which the rate of change of transmission capacity is evaluated to determine whether an adjustment of frame rate is necessary at regular intervals.

fuboTV Media Inc.
Exhibit 1002
Page 307

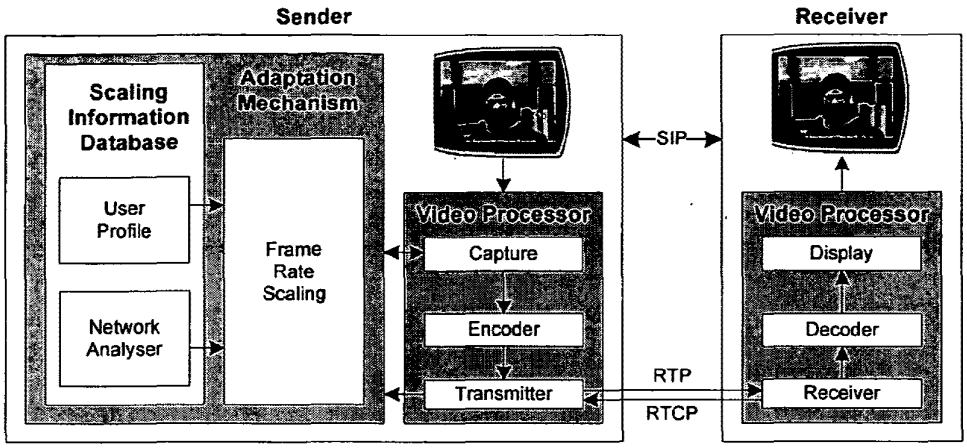


Figure 1 Basic System Operational View

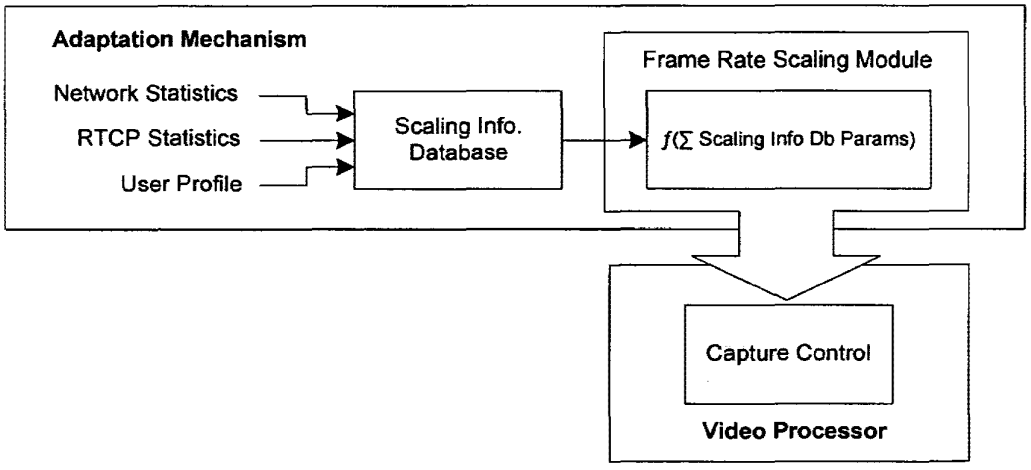


Figure 2

fuboTV Media Inc.
Exhibit 1002
Page 308

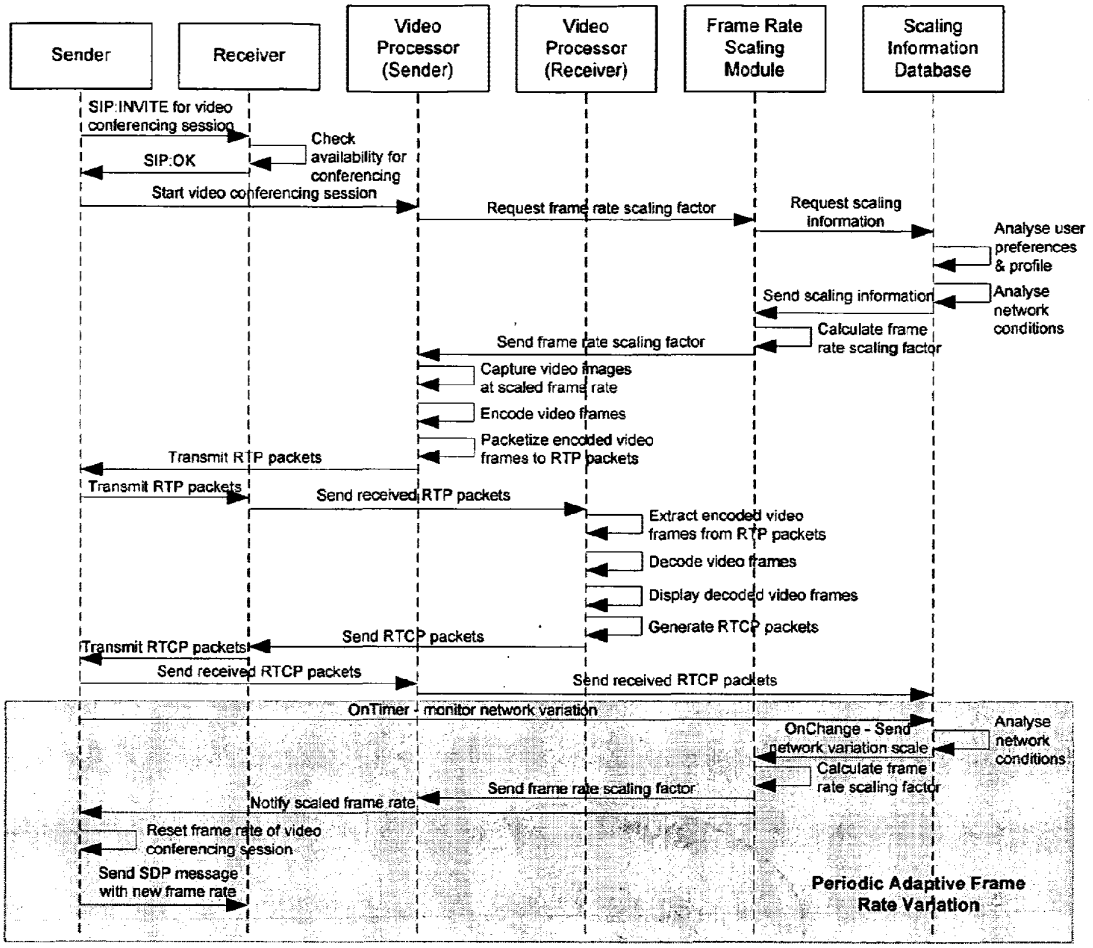


Figure 3 Sequence Diagram of Adaptive Video Frame Rate Application

fuboTV Media Inc.
Exhibit 1002
Page 309

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 05 25 6302

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X,D	KIM JONGWON ET AL: "TCP-Friendly Internet Video Streaming Employing Variable Frame-Rate Encoding and Interpolation" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 10, no. 7, 1 October 2000 (2000-10-01), pages 1164-1177, XP002369021 ISSN: 1051-8215 * page 1168, left-hand column, paragraph B *	1-5	H04N7/46
A	REED E C ET AL: "CONSTRAINED BIT-RATE CONTROL FOR VERY LOW BIT-RATE STREAMING-VIDEO APPLICATIONS" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 11, no. 7, July 2001 (2001-07), pages 882-889, XP001083363 ISSN: 1051-8215 * page 885, right-hand column - page 886, left-hand column *	1-5	TECHNICAL FIELDS SEARCHED (IPC) H04N
A	N. SEELAM, P. SETHI, W.FENG: "A hysteresis based approach for quality, frame rate, and buffer management for video streaming using TCP" SPRINGER VERLAG, INTERNATIONAL CONFERENCE ON MANAGEMENT OF MULTIMEDIA NETWORKS AND SERVICES, vol. 2216, 29 October 2001 (2001-10-29), pages 1-15, XP002374403 * paragraph [3.1.3] *	1-5	

-/--

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Berlin	28 March 2006	Raeymaekers, P

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

fuboTV Media Inc.
Exhibit 1002
Page 310

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 05 25 6302

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	NEE P ET AL: "The performance of two-dimensional media scaling for Internet videoconferencing" NETWORK AND OPERATING SYSTEM SUPPORT FOR DIGITAL AUDIO AND VIDEO, 1997., PROCEEDINGS OF THE IEEE 7TH INTERNATIONAL WORKSHOP ON ST. LOUIS, MO, USA 19-21 MAY 1997, NEW YORK, NY, USA,IEEE, US, 19 May 1997 (1997-05-19), pages 223-234, XP010251701 ISBN: 0-7803-3799-9 * page 226, right-hand column * -----	1-5	

TECHNICAL FIELDS SEARCHED (IPC)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Berlin	28 March 2006	Raeymaekers, P

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

fuboTV Media Inc.
Exhibit 1002
Page 311

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 6091777 A, W. Guetz **[0007]**

Non-patent literature cited in the description

- **T. TAKEBAYASHI et al.** Approaches to User-Centric Computing System for Ubiquitous Solutions. *Fujitsu Sci. Tech. J.*, 02 December 2003, vol. 39, 261-269 **[0007]**
- **M. NAGHSHINEH et al.** End-to-End QoS Provisioning in Multimedia Wireless/Mobile Networks Using an Adaptive Framework. *IEEE Communication Magazine,* November 1997 **[0007]**
- **J. KIM et al.** TCP-Friendly Internet Video Streaming Employing Variable Frame-Rate Encoding and Interpolation. *IEEE Transaction on Circuits and Systems for Video Technology,* October 2000, vol. 10 (7 **[0007]**
- *IBM. Functions of mobile multimedia QoS control, http://www.trl.ibm.com/projects/mmqos/system_e.htm* **[0007]**

fuboTV Media Inc.
Exhibit 1002
Page 312

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2001 (13.09.2001)

PCT

(10) International Publication Number
WO 01/67264 A1

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

(21) International Application Number: PCT/US01/40264

(22) International Filing Date: 8 March 2001 (08.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/520,434 8 March 2000 (08.03.2000) US

(71) Applicant *(for all designated States except US)*: **STREAMCENTER, INC.** [US/US]; 1512 Wellington Place, Aberdeen, NJ 07747 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **MOGHE, Pratyush** [US/US]; 3002 Cricket Circle, Edison, NJ 08820 (US).

(74) Agents: **DIPPERT, William, H.** et al.; Cowan, Liebowitz & Latman, P.C., 1133 Avenue of the Americas, 35th Fl., New York, NY 10036 (US).

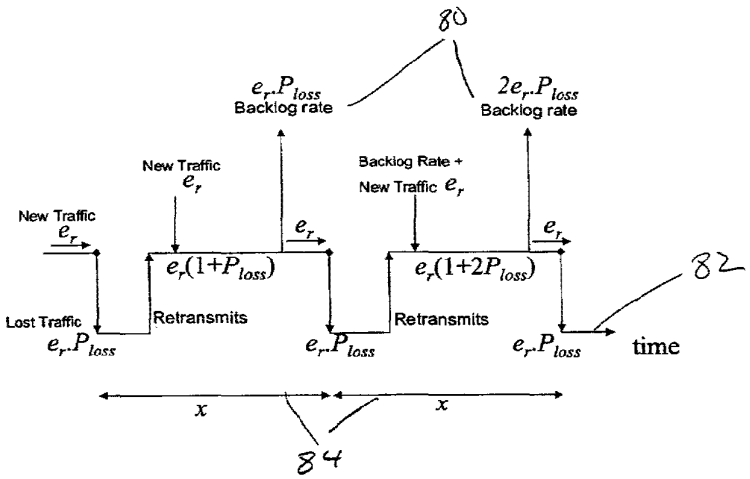
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE,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

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

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

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: APPARATUS AND METHOD FOR PREDICTABLE AND DIFFERENTIATED DELIVERY OF MULTIMEDIA STREAMING ON THE INTERNET



(57) Abstract: The inventive method detects network congestion which interferes with streaming of multimedia audio and video packets from and reacts to it and guarantees delivery of packets. The streaming of the multimedia audio and video packets is performed when a server process executing on a sending computing device sends the packets to a player process executing on a receiving computing device via a network, such as the Internet. The detection of the network congestion is performed by communicating configuration information including network parameters from a service manager via the network to the at least one sending computing device and to the at least one receiving computing device. The network congestion (80) is then confirming by transmitting and retracting network probes. At least one buffer management technique is executed to limit degradation in the streaming of the packets. The end of the network congestion is detected by transmitting and retracting said network probes and the rate of streaming of the packets is configured accordingly (82, 84).

WO 01/67264 A1

fuboTV Media Inc.
Exhibit 1002
Page 313

APPARATUS AND METHOD FOR PREDICTABLE AND DIFFERENTIATED DELIVERY OF MULTIMEDIA STREAMING ON THE INTERNET

TECHNICAL FIELD

This invention relates to the field of multimedia audio and video streaming on the Internet, and more particularly, to an apparatus and method for predictable and differentiated delivery of multimedia streaming on the Internet.

BACKGROUND OF THE INVENTION

Multimedia streaming involves broadcast of audio and video information from content providers called streaming servers or senders via the Internet to clients or receivers on user computing devices called streaming players. Real-time delivery of the multimedia streaming on the Internet is inherently uncontrollable and consequently unpredictable. When an Internet delivery path is congested, packets of multimedia streaming are typically dropped, thus causing degraded audio and video playback to be received by the streaming players. Typically, manifested forms of congestion can be classified into two types: instantaneous and sustained (longer-term). Instantaneous congestion is caused by random variations in the arrival patterns of packets of multimedia streaming at different Internet routers and switches, and due to the varying sizes of these packets.

These variations cause varying amounts of delay as the packets of multimedia streaming travel from the streaming server to the streaming receiver. If the delay accumulates beyond a certain point, the packets of multimedia streaming can get lost. This type of congestion typically causes sporadic delays and losses of over hundreds of milliseconds to a few seconds, and is hence referred to as "instantaneous congestion." In contrast, sustained congestion is longer-term and results during periods of excessive demand by the streaming players, e.g., demand for multimedia streaming from popular websites over the network paths on the Internet. During these periods, packet traffic surges can overwhelm the packet buffers in network routers, which are network traffic routing and forwarding computing devices, and packets of multimedia streaming can get

lost in bursts. This is similar to a rush-hour traffic scenario on highways at peak hours. Unlike instantaneous congestion, sustained congestion can last from tens of seconds up to a few minutes.

It is well known that instantaneous congestion is an easy problem to solve. The client or the streaming player can buffer incoming packets of multimedia streaming for a short duration before playback. This reduces the effect of short-term delay between packets of multimedia streaming. Additionally, by requesting that a lost packet be immediately retransmitted, the streaming player can counter the loss of packets of multimedia streaming due to instantaneous congestion. Several existing streaming systems use these techniques, i.e., short-term buffering along with retransmits of lost packets, to isolate the streaming player from the effects of instantaneous congestion. Examples of these systems include the Robust-User Datagram Protocol (UDP) feature in the Real system from Real Networks and the NetShow 3.0 MediaPlayer from Microsoft.

The problem of sustained congestion is harder to solve. This difficulty relates to the fact that the Internet delivery lacks a uniform real-time control. Several standardization and research efforts have proposed ways to introduce such control; however, none have succeeded in creating a global Internet standard. The referred to standardization and research efforts include the following:

1. L. Kleinrock, "On Flow Control in Computer Networks", International Conference on Communications (ICC'78), pages 27.2.1-27.2.5;
2. D.E. Comer, R. S. Yavatkar, "A Rate-based Congestion Avoidance and Control Scheme for Packet Switched Networks", The 10th International Conference on Distributed Computing Systems, pages 390-397, 1990;
3. K.K. Ramakrishnan, R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks", ACM Transactions on Computer Systems, Vol. 8, No. 2, May 1990, Pages 158-181; and
4. D. Mitra, "Asympotically Optimal Design of Congestion Control for High Speed Data Networks", IEEE Transactions on Communications, Vol. 40, No. 2, February 1992.

For example, the RSVP effort, described in Braden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S., "Resource Reservation Protocol (RSVP) - Version 1

fuboTV Media Inc.
Exhibit 1002
Page 315

Functional Specification", RFC 2205, Proposed Standard, September 1997, seeks to add "quality of service" control within each network router and switch within the Internet. Another effort that seeks to create fast paths on the Internet is the "differentiated service" effort within the Internet Engineering Task Force (IETF) described in the following publications:

1) Nichols, K., Jacobson, V. and Zhang, L., "A Two-bit Differentiated Services Architecture for the Internet", Internet Draft, December 1997;
2) Ferguson, P., "Simple Differential Services: IP TOS and Precedence, Delay Indication, and Drop Preference", Internet Draft, November 1997; and
3) Clark, D. and Wroclawski, J., "An Approach to Service Allocation in the Internet", Internet Draft, July 1997.

The problem with these approaches is that they require structural changes throughout the Internet. This requirement assumes an explicit cooperation between various managers of the equipment comprising the Internet, which is an impractical assumption. Consequently, these efforts have stalled or are expected to have limited success. An alternative approach is to offer "adaptive applications". This approach proposes that application content be adapted based on the level of congestion. Methods based on this approach degrade the "experience" of an application in a controlled manner, but do not attempt to guarantee predictability or isolation from congestion effects. This approach is discussed in I. Busse, B. Deffner, H. Schulzrinne, "Dynamic QoS Control of Multimedia Applications based on RTP", May 30, 1995 and Real Networks, "White Paper on SureStream Technology", Available at http://www.real.com/devzone/library/whitepapers/surestrm.html.

What is needed is a "delivery framework" that will isolate streaming players from the effect of sustained congestion by providing a "predictable" delivery. Furthermore, there is also a need for a "differentiated streaming" framework that will provide for multiple levels of delivery isolation to the streaming players, without requiring structural changes in the Internet, as in K. Nichols et al., P. Ferguson and D. Clark, discussed above.

fuboTV Media Inc.
Exhibit 1002
Page 316

SUMMARY OF THE INVENTION

The object of the present invention is to achieve a particular level of delivery isolation on particular streaming players. This level of delivery isolation may be based on factors like an importance "value" associated with particular streaming players, or the price the user of the streaming players are charged, etc. As an example, the streaming server may place specific streaming players in the high "value" category based on the demographic information on their owners, while other streaming players may be placed in the low "value" category.

Another object of the present invention is to ensure that the high value users of streaming players always receive consistent, predictable delivery of streaming content, even if there is an excessive user demand. If there is a situation of sustained congestion, wherein not all users can be satisfied, the content provider might want to ensure that the high value users are promised predictable delivery.

Accordingly, the present invention provides a method and an apparatus for predictable and differentiated streaming on the Internet. The system consists of software and random-access memory-based buffers installed at the streaming server and at the streaming players along with a service manager at a separate location. The method prescribes the manner in which the system is used to achieve predictable and differentiated delivery of streaming multimedia packets.

The invention is based on streaming multimedia packets between the software on the streaming server and the software on the streaming. The buffer sizes in the computing devices of the streaming player and of the streaming server are initialized based on the configuration information provided by the service manager software at the start of the execution of the inventive method. This configuration information includes:

a) the expected duration of sustained congestion periods,

b) the worst-case packet loss rate during the sustained congestion period,

c) the encoded bit-rate of streaming source information, and etc.

During the delivery of the streaming multimedia packets, the software on the streaming server relays the streaming packets to the software on the streaming player, which stores the packets in a playout buffer before playing them out. The playout buffer takes care of

fuboTV Media Inc.
Exhibit 1002
Page 317

delay variation that may arise due to instantaneous congestion on the network path. In case of short-term losses caused by instantaneous congestion, the software on streaming player immediately requests the server-end software to retransmit the lost information.

Predictable Delivery Method

To deal with sustained congestion ("congestion") the invention incorporates the following method. The software on the streaming player uses estimation techniques based on probes to detect and confirm the onset of sustained congestion. Once the congestion status is verified, the streaming player software informs the streaming server software about this event. The streaming server software then modifies the manner in which it streams packets to the streaming player software. In particular, the streaming server software ensures that the transmission rate of packets never exceeds the encoded bit-rate of the streaming source. Furthermore, it gives preemptive priority to the retransmitted packets over new packets. This creates a backlog of new packets at one of the streaming server buffers.

Meanwhile, the streaming player software continuously monitors the status of received packets and detects the end of congestion through estimation techniques based on probes. Once the end of congestion is confirmed, the streaming player software informs the streaming server software about this event. The streaming server software then begins to stream its backlog of new packets to the streaming player software. The transmission rate of this stream is increased based on the estimation information from the streaming player software along with feedback obtained from other streaming players in the network. This rate is carefully selected to ensure that the network is not flooded back into congestion. The buffer sizing at the streaming player and at the streaming server is based on the configuration information and ensures that the user experiences an uninterrupted playback during sustained congestion. This assumes that the configuration information is the "worst-case" scenario. If the actual service parameters turn out to be worse than the configuration information, the streaming player software contacts the service manager after the service. The configuration information at the service manager is now updated in a persistent manner with the actual worst-case parameters.

fuboTV Media Inc.
Exhibit 1002
Page 318

In one preferred embodiment, this update happens as follows. At the next service invocation, the buffers are sized based on the updated configuration information. The method thus automatically learns the worst-case behavior from the actual behavior of the network each time. Eventually, the method can guarantee a predictable delivery based on a non-interrupted playout in most cases. Various other embodiments are also possible, wherein the buffer size is fixed, but the encoding bit-rate of the stream is modified to choose the largest encoding bit-rate that leads to an uninterrupted playback.

Differentiated Streaming Method

The method for differentiated streaming involves identical operations as the predictable streaming method, except for the mechanism for streaming the backlog of packets from the streaming server software. For differentiated streaming, the streaming server software transmits the stream based on the "value" associated with the particular user, in addition to the information used for predictable streaming.

Accounting Delivery Service

Additionally, the methodology of the mentioned invention is utilized to offer the inventive method as an "accountable delivery service" to a large number of content providers, i.e., streaming servers, and users, i.e., streaming players. Such a service may deliver performance contract-based services to users of the streaming players on the Internet.

The service manager plays a central role in this service, by keeping and exchanging the state of several streaming players, delivery paths, and content providers/streaming servers in its database. The method of the present invention tracks the "actual" delivered viewing/listening experience of users of streaming servers by computing the delivery statistics including but not restricted to metrics that quantify "pauses or interruptions", "content throughput", and "content loss". In addition to the features of predictability and differentiation, the delivery service can also answer the question of "service feasibility". That is, at the beginning of a session, the service manager can estimate, based on the historical and instantaneous information, whether a

fuboTV Media Inc.
Exhibit 1002
Page 319

particular user can receive an "expected quality experience." While the present invention does not describe the details of this service, it should be obvious to those skilled in the art to build such a service around the framework of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects and advantages of the invention will be apparent upon consideration of the following detailed description, taken in conjunction with the accompanying drawings, in which the reference characters refer to like parts throughout and in which:

Figure 1 is a schematic diagram of the devices and programs comprising the invention.

Figure 2 is a schematic diagram of the components of the devices utilized by the invention.

Figure 3 is a schematic diagram of the operation of the invention during normal streaming.

Figure 4 is a schematic diagram of the operation of the invention during congestion.

Figure 5 is a schematic diagram of the recovery scheduling operation of the invention after congestion.

Figure 6 is a diagram of the modeling the backlog operation of the invention during congestion.

Figure 7 is a state transition diagram for streaming player module (normal streaming).

Figure 8 is a state transition diagram for streaming server module (normal streaming).

Figure 9 is diagram of a congestion processing protocol at the streaming player module (during and after congestion).

Figure 10 is a diagram of an congestion processing protocol at the streaming server module (during and after congestion).

fuboTV Media Inc.
Exhibit 1002
Page 320

DETAILED DESCRIPTION OF THE INVENTION

Architecture

The invention, shown in Figure 1, comprises at least one computing device 10 for executing a service manager process 12, at least one computing device 14 for executing streaming server process 16, and at least one computing device 18 for executing streaming player process 20. The service manager process 12 is shown as residing on the computing device 12 only for the clarity of presentation. It may be obvious to those skilled in the art that the service manager process 12 may reside on any computing devices 14 and 18. Furthermore, server processes 16 may be present on the computing devices 18 alongside player processes 20, and vice versa the player processes 20 may be present on the computing devices 14 alongside server processes 16. Each computing device 10, 14 and 18 are connected to a network 22 and establish data paths amongst each other. In the preferred embodiment, the network is the Internet.

The streaming server process 16 transmits a single multimedia stream to each streaming player process 20 and the streaming player process 20 receives that single multimedia stream. The streaming server process consists of three buffers:

a) Output buffer 24, used to transmit new packets from the server process 16 to the player process 20;

b) Backup buffer 26 , for maintaining temporary copies of the packets transmitted to the player process 20; and

c) Retransmit buffer 28, for maintaining retransmit requests received from the player process 20. The streaming player process 20 may comprise a player buffer 29.

Before a streaming session begins, the Service Manager process 12 downloads configuration information via the network 22 to both the computing device 14 for the use of the streaming server process 16 and to the computing device 18 for the use of the streaming player process 20. At the end of each streaming session, the configuration information on the computing device 10 is updated for the future use of the Service Manager process 12 via the network 22 by both the streaming server process 16 on the computing device 14 and the streaming player process 20 on the computing device 18.

fuboTV Media Inc.
Exhibit 1002
Page 321

The computing devices 10, 14, and 18 may take the configuration of any computer ranging from mainframes and personal computers (PCs) to digital telephones and hand held devices, e.g., palm pilots™. In one illustrative embodiment of this invention shown in Figure 2, such computing devices may comprise a bus 30, which is connected directly to each of the following:

1. a central processing unit (CPU) 32;
2. a memory 34;
3. a system clock 36;
4. a peripheral interface 38;
5. a video interface 40;
6. an input/output (I/O) interface 42;
7. a communications interface 44; and
8. a multimedia interface 46.

The common bus 30 is further connected

1. by the video interface 40 to a display 50;
2. by the I/O interface 42 to a storage device 52, which may illustratively take the form of memory gates, disks, diskettes, compact disks (CD), digital video disks (DVD), etc.;
3. by the multimedia interface 46 to any multimedia component 56;
4. by peripheral interface 38 to the peripherals 58, such as the keyboard, the mouse, navigational buttons, e.g., on a digital phone, a touch screen, and/or writing screen on full size and hand held devices, e.g., a palm pilot ™;
5. by the communications interface 44, e.g., a plurality of modems, to a network connection 60, e.g., an Internet Service Provider (ISP) and to other services, which is in turn connected to the network 22, whereby a data path is provided between the network 22 and the computing devices 10, 14, and 18 (Figure 1) and, in particular, the common bus 30 of these computing devices; and
6. furthermore, by the communications interface 44 to the wired and/or the wireless telephone system 54.

fuboTV Media Inc.
Exhibit 1002
Page 322

Normal Conditions

Turning now to Figure 3, the normal streaming operation is shown when there are either zero, i.e., no losses, or sporadic streaming losses, i.e., loss occurring due to instantaneous burst of streaming packets. The overall streaming sequence works as follows. Packets streamed from the streaming server process 16 are queued up in the streaming server output buffer 24 for transmission. The server process 16 is responsible for transmitting streaming packets 70 to the computing device 18 executing the player process 20. Periodically, this process 16 checks the output buffer 24 and the retransmit buffer 28 to determine if there are streaming packets 70 awaiting transmission or any requests form the player process. Any remaining streaming packets in the output buffer 24 are then transmitted. If retransmission requests are found, any streaming packets 70 requested for retransmission and found in the backup buffer 26 are transmitted. When streaming packets 70 are transmitted, a temporary backup copy of the transmitted streaming packets 70 is kept in the backup buffer.

The streaming player process 20 receives the streaming packets 70 and stores them at the tail-end of the player buffer 29. Periodically after a user-defined interval, the player process 20 examines the received streaming packets 70 to see if any streaming packets 70 are missing. If packets are missing, the player process 20 sends a request 72 to the server process 16 to retransmit the missing packets. After assuring that all the streaming packets 70 are received, the received streaming packets 70 are forwarded from the head-end of the player buffer to the streaming player process 20.

Congested Conditions

The invention detects congestion early on, and then may take steps so that the impact of congestion on the stream is minimized. A correctly configured framework can keep a stream completely transparent of sustained congestion, thus providing an effect equivalent to a "reservation." Two principles are at the basis of this framework, detection/reaction to congestion and congestion recovery.

1. Detecting and reacting to congestion

fuboTV Media Inc.
Exhibit 1002
Page 323

The invention detects congestion through probes launched by the player process 20. The probes are discussed in detail later. Once congestion is confirmed, the server process 16 reacts by using a technique called "constant bit-rate emulation." This name derives from the fact that the server process 16 tries to maintain a constant upper bound on the rate at which the streaming packets 70 are streamed to the player process 20. Typically, the upper bound is set equal to the encoding bit rate of the stream. In effect, the server process 16 emulates a constant bit-rate by substituting new server packets with packets that have been requested 72 for retransmission. Please recall that in congestion, the server has a large number of retransmit requests 72 issued from the player process 20. Figure 4 shows the flow of streaming packets 70 during this congestion situation. Two important reasons motivate the constant bit-rate emulation method:

a) The method demonstrates a "network-friendly" yet efficient behavior. The method is network-friendly, since it conserves bandwidth during congestion by lowering the upper bound of its streaming transmission rate. Additionally, the method is efficient for the following reasons. First, the upper bound on the transmission rate is set equal to the maximum encoding bit-rate of the source stream; this is not a restrictive setting. Unlike other schemes that voluntarily reduce the source rate during congestion, the present invention only reduces the transmission rate by the amount of incurred loss. This decision has an important implication in practice. Most practical congestion-processing algorithms will incur errors and delays in estimating and reacting to the start of congestion or end of congestion. Consequently, there will be temporary periods in which these algorithms "think" that the network is congested, when in practice, the network has cleared out of congestion. In situations like this, estimation errors do not degrade the streaming throughput. In fact, the invention delivers a throughput better or equal to the throughput of any other scheme, if there is no loss of streaming packets 70.

b) Assuming most connections sharing the network with streaming are using the Transmission Control Protocol (TCP), the constant bit-rate of the invention in congestion allows the TCP traffic on the same path to accurately sense the available bandwidth (most Internet or Web traffic uses TCP). An alternative design with

fuboTV Media Inc.
Exhibit 1002
Page 324

adapting bit-rate would confuse most other TCP connections and cause each to keep oscillating their TCP windows. If all TCP connections maintain a steady window-size, the performance of the network is much smoother in congestion. This behavior can drain out the congestion faster and reduce the estimation errors.

2. Detecting End-of-Congestion and Recovery from congestion

The invention detects the end of congestion through probes launched by the player process 20. Once the player process 20 confirms that congestion is over, the server process 16 enters a phase called "recovery scheduling". Figure 4 shows a recovery scheduling after congestion. To understand this phase, recall that in the previous state of congestion, the server transmits packets at a constant rate. By substituting new packets with older lost packets, the server is in effect creating a backlog of new packets in its output buffer. The idea in recovery scheduling is to exhaust the substantial backlog that may build up at the server output queue. Clearly, this backlog must not be cleared by flooding the network. Such action may cause significant losses in the intermediate switch buffers. In addition, if thousands of streams of packets 70 flood the backlog at the same time, this may cause certain network points to get congested again. Recovery scheduling is a procedure that configures the rate of streaming the backlog based on the network parameters. Additionally, recovery scheduling can also take into account the "premium" placed on the particular player process 20. The latter consideration allows the server process 16 to stream backlog of streaming packets at different priorities. "Higher value" users can receive backlog sooner than "lower value" users. This is the basic principle behind differentiated streaming.

Buffer Design

As described previously, the configuration information is downloaded to the computing devices 14 and 16 at the beginning of each streaming session. This configuration information includes the following parameters:

(1) Worst-case duration of congestion, $c*x$, where x is a sampling or retransmission interval, and c is a number of sampling intervals in a congestion period.

fuboTV Media Inc.
Exhibit 1002
Page 325

(2) Worst-case of loss of streaming packets 70 during congestion defined by P_{loss}

(3) Encoding bit-rate of the stream of streaming packets 70 defined by e_r

(4) Averaged and minimum service rate, SR_{avg} and SR_{min}. This is the rate at which streaming packets 70 are actually transferred from the server process 16 to the player process 20. SR_{min} is the minimum rate across the path via the Internet 22 from the server process 16 to the player process 20.

(5) Path bottleneck link bandwidth B_{min} and the hop link capacities, where B_{min} is the minimum physical bandwidth across all the hops, i.e., the routing devices between the server process 16 and the player process 20.

The parameters indicated in the list above may be obtained through historical measurements of the relevant values and through measurements made on streaming packets 70 at the computing device 14 of the server process 16. The parameters are in turn used to design the sizes of the buffers 24, 26, 28 and 29 of server process 16 and the player process 20. One preferred embodiment of achieving this is described immediately below. That embodiment ignores the upper bounds limits on the buffer sizes related to the available random-access memory 34 (Figure 2) in the computing devices 14 and 18.

Player Buffer-size (b_c)

The buffer on the computing device 18 used by the streaming player process 20 plays out the streaming packets 70 at the maximum rate of e_r bits per second. During sustained congestion, the packet loss is approximately P_{loss}, and the maximum transmission rate from the server, as explained above is e_r. It follows that the rate of the player buffer 29 fill rate equals to $e_r * (1 - P_{loss})$ bits per second. Thus, the net drain on the player buffer 29 proceeds at the rate of $e_r * P_{loss}$ bits per second. To prevent the player buffer 29 from draining out completely within the congestion period $c * x$, the buffer-size of the player buffer 29 should satisfy the constraint defined as:

fuboTV Media Inc.
Exhibit 1002
Page 326

$$\frac{b_c}{e_r * P_{loss}} \geq c * x \quad \text{or} \quad b_c \geq c * x * e_r * P_{loss} \qquad \text{(Equation 1)}$$

Streaming Server Output Buffer-size (b_o)

Recall that the streaming server output buffer holds the backlog of new packets created during the congestion period. Consequently, the size of this buffer is dictated by the worst-case amount of backlog during the congestion period. To calculate the worst-case backlog of the streaming packets 70, a simple model may be built up. Figure 6 shows the approximate amount of backlog as a function of time 82 during sustained congestion. As indicated, the backlog rate 80 after two retransmission intervals 84 is $2 * e_r * P_{loss}$. By extension, after c retransmission intervals 84, the backlog rate 80 will be $cx * e_r * P_{loss}$. This establishes the constraint on the size of the output buffer 14 (Figure 1) in the memory 34 (Figure 2) of the computing device 14 (Figure 1) executing streaming server process 16 (Figure 1).

$$b_o \geq c * x * e_r * P_{loss} \qquad \text{(Equation 2)}$$

Server Backup Buffer-size (b_{backup})

The backup buffer 26 (Figure 1) holds temporary copies of the server packets that have been transmitted to the player process 20 (Figure 1). The copies must be held until the server is sure that the packets have been successfully received in the player buffer 29 (Figure 1). To estimate the size of this backup buffer 26 (Figure 1) again, an operational model is built. First, the transmission rate in packets per seconds is defined as R_p. At the start of the first retransmit 84, the approximate rate of loss of the streaming packets 70 (Figure 3) is $P_{loss} * R_p$. Assuming the loss affects new as well as retransmitted streaming packets 70 (Figure 3) equally, the approximate rate of loss for older unsuccessful streaming packets 70 (Figure 3) at the second retransmit 84 is $P_{loss}^2 * R_p$. Supposing now that the number of retransmits that will reduce the loss rate for older

fuboTV Media Inc.
Exhibit 1002
Page 327

packets to a negligibly small number, e.g., 0.1 packets for every retransmission interval is defined by m. This implies that:

$$P_{loss}^{m} * R_p \leq \frac{0.1}{x}, \quad \text{or that} \quad m \geq \frac{\log\left(0.1/R_p * x\right)}{\log P_{loss}}$$

Alternatively, if s is the packet size, the above equation may be expressed as:

$$m \geq \frac{\log\left(0.1 * s/e_r * x\right)}{\log P_{loss}}$$

The minimum required buffer size may be defined as equal to $m * x$. Consequently, the size constraint of the backup buffer 26 (Figure 1) is

$$b_{backup} \geq m * x \quad \geq x * \frac{\log\left(0.1 * s/e_r * x\right)}{\log P_{loss}} \qquad \text{(Equation 3)}$$

The detailed operational method underlying the invention is captured through a protocol called "Streaming Congestion Protocol" (SCP). In the section below the SCP sequences during different phases of streaming are described. Figures 6 and 7 show the detailed operational sequence at the streaming server process 16 (Figure 3) and the streaming player process 20 (Figure 3) during normal streaming of the streaming packets 70 (Figure 3). The operational sequences are described with the help of state diagrams. A state diagram shows various states and transitions between states. Each state captures a specific operational state, and the transitions between states capture the events that cause the operational states to change. Figure 7 shows the state diagram of events at the streaming server process 16 (Figure 1). Figure 8 shows the corresponding state diagram at the streaming player process 20.

Processing during and after congestion

Please refer to Figure 9, by default, the streaming player process 20 is in a normal state 100. The player process 20 may keep a track of the packet loss of a received stream at a periodic interval that may be user determined. If the amount of loss exceeds a

fuboTV Media Inc.
Exhibit 1002
Page 328

threshold, e.g., Threshold1, across a time-window of some number of seconds which may also be user determined, e.g., TW1, the player process 20 moves into the Likely Congestion state 102. At this point, the streaming player process 16 (Figure 1) launches congestion probes to determine whether the congestion symptoms are verified. These probes form a part of a test called confirm congestion test. If the test indicates that congestion is confirmed, the process moves to a Congestion Confirmed state 104. Immediately, the streaming player process 20 contacts its server counterpart and informs it about the new state change along with some other relevant information.

Note that the transition from a Likely Congested state 102 to Congestion Confirmed State 104 can happen in two other ways. First, the player process 20 may actually receive a Remote Congestion Confirmed report in state 106 from its server process 16 (Figure 1) indicating that some other player process 20 on a related network path is in the Congestion Confirmed state. If this happens, the player immediately moves to the Congestion Confirmed State 104 itself. If the player process 20 has already started conducting congestion probes before this notification, the player process 20 may halt them at the instant it receives the notification. Alternatively, the player process 20 may receive a Local Congestion Confirmed report in state 102 reporting that some other local player process 20 on the same streaming player module has verified congestion. If the Remote or local Congestion Confirmed reports have verified congestion within a past time-window not exceeding the time defined as Tcc, the player process 20 moves to a confirmed congestion state 104.

Once the streaming player process 20 (Figure 1) enters the congestion confirmed state 104, it periodically monitors the loss of the received stream of streaming packets 70 (Figure 3). If the packet loss is less than a second user defined threshold, e.g., Threshold2, across a second user defined time-window, e.g., TW2 seconds, the player process 20 enters the likely end-of-congestion (EOC) state 106. At this point, the player process 20 launches an EOC probe test that either confirms or invalidates the EOC decision. If the EOC probe test confirms congestion, the streaming player process 20 enters the EOC state 108 and sends an EOC report 110 to its server process 16 (Figure 1) counterpart. With this transition, the process re-enters the normal state 100. Note that

fuboTV Media Inc.
Exhibit 1002
Page 329

during transitions from the EOC state 110 back to the Normal State 100, the streaming player process 20 also needs to continually process the arriving packets in step 120 (Figure 7), check and request for retransmission of missing packets in step 122 (Figure 7), and forward packets to the streaming player for playback in step 124 (Figure 7). These activities are indicated in Figure 7 and are not shown in Figure 9 for purposes of clarity.

Figure 10 shows the SCP state-diagram of the server process 16. After initialization, the server process 16 enters the normal state 130. If the server process 16 receives a report from its streaming player process 20 (Figure 1) indicating congestion, the server moves to the Confirmed Congestion or Confirmed Congestion State 132. At this point, the server process 16 shares the Confirmed Congestion status information with other server processes 16 running on the same computing device 14 (Figure 1) in the Propagate Confirmed Congestion state 134, and potentially also from other computing device 14 (Figure 1) through interaction with the service manager 12 (Figure 1) via the network 22 (Figure 1). The idea behind this is to have selected server processes report the congestion to their player process so that these player processes can avoid doing the Confirmed Congestion probing test. This cuts down on the congestion reaction time and significantly cuts down the probing traffic. In general, each server process that receives the Confirmed Congestion status in the Receive the Confirmed Congestion status state 140 from other server processes 16, first determines whether the Confirmed Congestion status is relevant to its target player process 20 (Figure 1). If this is the case, the server processes notify their player process 20 (Figure 1). Notification is done through a special signaling scheme involving reliable transport protocol (RTP) messages, as discussed in Schulzrinne, Casner, Frederick, Jacobson, "RTP: A Transport Protocol for Real-time Applications", RFC 1889, Internet Engineering Task Force.

The streaming server process 16 will then make transitions from the Confirmed Congestion State 132. Starting with the Confirmed Congestion state 132, the server process 16 transmits streaming packets 70 (Figure 3) at a constant bit-rate. The server process 16 moves to the Recovery Scheduling State 136 as soon as it receives an End-of-congestion or EOC report from the player process 20 (Figure 1) targeted for the reception of

fuboTV Media Inc.
Exhibit 1002
Page 330

its content. Thereafter the server process 16 changes its mode of transmission, using recovery scheduling to drain its backlogged output buffer 24 (Figure 1) in an intelligent way. As soon as the backlog is over, the server process 16 moves back to the normal state 130.

While in the Normal State 130 the server process 16 may receive and filter information from other server processes 16 by entering the Receive Confirmed Congestion State 140. It may also notify the player processes 20 (Figure 1) about the confirmed congestion status through the in-band RTP signaling, by entering the Confirmed Congestion Notification state 138 from where the process 16 will return to the Normal state 130.

The server process 16 also needs to continually process the departing packets in step 150 (Figure 8), check and process the requests for retransmission of missing packets in step 152 (Figure 8), and backup packets to the backup buffer 26 (Figure 1) for playback. These activities are indicated in Figure 8 and are not shown in Figure 10 for purposes of clarity.

Recovery scheduling algorithm

Supposing that the server process 16 (Figure 1) streams of streaming packets 70 (Figure 3) are classified into K number of priority classes. Class 1 being the highest priority and class "K" being the lowest priority. Each priority class "j" is associated with a weighting function α^j, where $0 < \alpha^K < \alpha^{K-1} < .. \alpha^j < .. < \alpha^1 = 1$. Considering the server process 16 (Figure 1) with priority equaling to "j", the rule for transmission rate e_t of this process is:

$$e_t = \min(\alpha^j \times SR_{min}, \alpha^j \times \frac{B - \lambda_{other}}{\sum_{i=1}^{i=K} \alpha^i n^i})$$

(Equation 4)

In equation 4, all the terms are measured at the service bottleneck hop between the computing device 14 executing the server process 16 and the computing device 18 executing the player process 16. B is the bottleneck link capacity at the bottleneck hop.

fuboTV Media Inc.
Exhibit 1002
Page 331

λ_{other} is the aggregate packet rate carried at the hop from traffic not utilizing the inventive method. This rate can be estimated by obtaining the aggregate packet rate of the streaming packets 70 (Figure 3) of the inventive method λ_{SME} from the service manager, and using the fact that $\lambda_{other} = B \times \rho - \lambda_{SME}$, where $\rho = 1 - \frac{SR_{\min}}{B}$ is the utilization of the bottleneck hop. The parameter n^i is defined as the number of priority "i" streams carried at the bottleneck hop. The hop is a section of a network between two network-computing devices such as routers. This parameter may be obtained from the service manager 12 (Figure 1). The logic behind this equation is as follows. The first term indicates that the new transmission rate should be set equal to the minimum service rate along the path, in order to recover from the backlog. The second term takes into account the result of increasing the transmission rate on the utilization at the bottleneck hop. In other words, since the maximum utilization can be 100%, each computing device 14 executing server process 16 must carefully weigh the effects of increasing the transmission rate. The best rate is the maximum rate that does not congest the bottleneck bandwidth during recovery. Hence the min operator. Finally, the priorities within the equation 4 provide for differentiated streaming.

Congestion Estimators

Earlier on, we described the congestion estimation tests as part of the SCP algorithm. In principle, congestion can be estimated in a variety of ways, and the invention is not restricted to any one particular method of congestion estimation. A particular embodiment of the invention for estimating congestion based on a mix of passive and active probes will now be described. This method comprises a set of rules configured around the results of the probes. A specific congestion test e.g., an end of congestion test or a confirmed congestion test mentioned in earlier sections, combines these rules in a particular way.

The following are some of a variety of methods that may be used to compute the bottleneck bandwidth on a path between the streaming player 20 (Figure 1) and the

fuboTV Media Inc.
Exhibit 1002
Page 332

streaming server 16 (Figure 1). These methods are shown here for exemplary purposes only and not to indicate that only these methods must be used with the present invention.

Capacity Probe

This method is described in detail in Van Jacobson, "pathchar - A tool to infer characteristics of Internet paths", MSRI, April 21, 1997; and R. L. Carter and M.E. Crovella, "Measuring Bottleneck Link Speed in Packet-switched Networks", TR-96-006, Boston University Computer Science Department, March 15, 1996. This method is initiated by the player process 20 (Figure 1) at various stages of a streaming session. The steps of this exemplary method are as follows:

1. Do a traceroute and get hop identities marked by IP addresses.
2.

a) For every hop from one onwards, send a series of ICMP timestamp requests (type 13) from the player to the hop. Packet sizes of queries are changed in steps of 1024, 512, 256, 128, and 64, where each queries are sent out with a spacing of 3 seconds between them.

b) For every hop i between the player and the server, the player must record the ICMP timestamp response (type 14) and compute the time difference between the hop i transmit time t_i and the player receive instant t_1.

c) Denote the time difference by $\tau_{i1} = t_i - t_1$.

d) Take four measurements for timestamp differences by varying packet size through five values. Discard the minimum and the maximum, and take average of the remaining two. Let s' and s'' denote the packet sizes of two selected queries to hop i. Let $\tau_{i1'}$ and $\tau_{i1''}$ denote the measurements for the respective packet sizes.

e) Calculate $X_{i1} = \frac{\tau_{i1'} - \tau_{i1''}}{s' - s''}$

fuboTV Media Inc.
Exhibit 1002
Page 333

$$B_{i,i-1} = \frac{1}{X_{i1} - \left(\sum_{k=1}^{i-2} \frac{1}{B_{i-k,i-k-1}} \right)}$$

It can be proved that link capacity can be computed recursively.

The path bottleneck bandwidth $B_{min} = \min_i B_{i,i-1}$. (Equation 5)

Service Rate Probe

The "Service Rate probe" computes the approximate service rate of transferring packets between the server and the player, as discussed in Van Jacobson, "pathchar - A tool to infer characteristics of Internet paths", MSRI, April 21, 1997; and R. L. Carter and M.E. Crovella, "Measuring Bottleneck Link Speed in Packet-switched Networks", TR-96-006, Boston University Computer Science Department, March 15, 1996. The minimum service rate is the least service rate on the path, while the average service rate is the "expected" service rate on the path. . The steps of this exemplary method are as follows:

1. Obtain the hop list through traceroute. Suppose there are n-1 hops between the player and the server.
2. Periodically, the player initiates a cycle. In each cycle, the player sends n-1 queries, each spaced by ¼ second. The player forms timestamp differences between query responses from consecutive hops. The results are sent back to the server.
3. The server uses results from several cycles along with the link capacities to calculate the service rate.
   a) First, the server calculates $\tau_{m,m-1}^{min} = min(\tau_{m,1} - \tau_{m-1,1})$ where the min operator works over all collected cycle results. We approximate $\tau_{m,m-1}^{min}$ to be the base-line delay between node m and node m-1.
   b) Denote the maximum queuing delay between node m and node m-1 by $q_{m,m-1}^{max}$, where node is a computing device, e.g., a computer or a router connected to the network.

fuboTV Media Inc.
Exhibit 1002
Page 334

c) Then, $q_{m,m-1}^{max} = max\left(\tau_{m,m-1} - \tau_{m,m-1}^{min}\right)$

d) Denote the average queuing delay between node m and node m-1 by $q_{m,m-1}^{avg}$.

e) Then, $q_{m,m-1}^{avg} = avg\left(\tau_{m,m-1} - \tau_{m,m-1}^{min}\right)$

Finally, the minimum service rate is:

$$SR_{m,m-1}^{min} = \frac{s}{\frac{s}{B_{m,m-1}} + q_{m,m-1}^{max}},$$

and the average service rate is:

$$SR_{m,m-1}^{avg} = \frac{s}{\frac{s}{B_{m,m-1}} + q_{m,m-1}^{avg}} \qquad \text{(Equation 6)}$$

Congestion Detection Rule (CC Test)

A robust but responsive congestion detection mechanism is an essential part of the inventive method. It is important to respond quickly to congestion; this can reduce the congestion period and hence make it easier for the recovery scheduling to do its job in a shorter period of time. Ultimately this can allow the method to work with a reduced buffer size and hence a reduced connect lag. It is equally important to robustly distinguish a sustained congestion condition. False detection can lead to reduced throughput and performance. Recall that the inventive method treats sustained congestion by reducing the throughput of the newer packets, so that the throughput of older retransmitted packets is maximized.

The present invention uses the following congestion detection rule:

1. Packet losses exceeding a threshold for a consecutive time-window of duration T_s. T_s is set to 10 seconds by default.
2. If condition 1 is triggered, three short service rate probes are sent, spaced 2 seconds apart, from the player to the SR_{min} hop. Responses are averaged to form

fuboTV Media Inc.
Exhibit 1002
Page 335

SR_{avg}.

The player process 20 (Figure 1) computes the expected utilization $\rho = 1 - \frac{SR_{avg}}{B}$. If condition 1 is satisfied and the utilization from condition 2 is more than a specified utilization threshold, e.g., 95%, the player process 20 (Figure 1) declares sustained congestion (Confirmed Congestion), and asks the server to enter the constant bit-rate emulation mode. An alternative method may consists of a simple rule where if packet losses exceed a threshold for N_1 consecutive sampling time-windows, the player process 20 (Figure 1) declares sustained congestion and asks the server to enter the constant bit-rate emulation mode. By default, N_1 is set to 1, and time-window is set equal to 1 second.

End of Congestion Rule (EOC Test)

The end of congestion rule is analogous to the congestion detection rule:

1. No packet losses for a consecutive time-window of duration T_e. T_e is set equal to 10 seconds by default.
2. If condition 1 is triggered, three short service rate probes are sent, spaced 2 seconds apart, from the player to the SR_{min} hop. Responses are averaged to form SR_{avg}.

   The player computes the expected utilization $\rho = 1 - \frac{SR_{avg}}{B}$.

If condition 1 is satisfied and the utilization from condition 2 is less than a specified threshold, e.g., 90%, the player declares sustained congestion to be over, and asks the server to transition to the state of recovery scheduling. An alternative method may consists of a simple rule where if in a state of sustained congestion, no packet losses are manifest for N_2 consecutive sampling time-windows, the player declares sustained congestion to be over, and asks the server to transition to the state of recovery scheduling. By default, N_2 may be set equal to five, and the sampling time-window is set equal to 1 second.

fuboTV Media Inc.
Exhibit 1002
Page 336

Thus, an apparatus and method for predictable and differentiated delivery of multimedia streaming on the Internet is provided. One skilled in the art will appreciate that the present invention can be carried out in other ways and practiced by other than the described embodiments, yet not departing from the spirit and essential characteristics depicted herein. The present embodiments therefore should be considered in all respects as illustrative, and the present invention is limited only by the claims that follow.

fuboTV Media Inc.
Exhibit 1002
Page 337

CLAIMS

Having thus described our invention, what we claim as new, and desire to secure by Letters Patent is:

1. A method of streaming of multimedia audio and video packets from at least one sending computing device to an at least one receiving computing device via a network, said method comprising the steps of:

a) at a start of said streaming communicating configuration information from a service manager via the network to both an at least one sending computing device for use by an at least one server process and an at least one receiving computing device for use by an at least one player process, said at least one sending computing device comprising an at least one retransmit buffer, an at least one output buffer, an at least one backup buffer and said at least one server process, said at least one receiving computing device, comprising an at least one player buffer and an at least one player process;

b) queuing same packets in the at least one output buffer and to the at least one backup buffer;

c) determining if there are packets in the at least one output buffer and transmitting the packets from the at least one output buffer to the at least one receiving computing device via the network;

d) receiving the packets on the at least one receiving device and storing them in the at least one player buffer;

e) determining if there are omissions in the packets stored in the at least one player buffer and forwarding retransmission requests to the at least one sending computing device and storing them in the at least one retransmit buffer; and

f) determining if there are said retransmit requests in the at least one retransmit buffer and transmitting the packets from the at least one backup buffer to the at least one receiving computing device via the network.

fuboTV Media Inc.
Exhibit 1002
Page 338

2. The method of claim 1 wherein, the network is Internet.

3. The method of claim 2 wherein, a size of the at least one player buffer is $\geq c*x*e_r*P_{loss}$, where c is a number of sampling intervals in a congestion period, x is a sampling or retransmission interval, $c*x$ is a worst-case duration of congestion, e_r is an encoding bit-rate of a stream of the packets and P_{loss} is a worst case of loss of the packets.

4. The method of claim 3 wherein, a size of the at least one output buffer is $\geq c*x*e_r*P_{loss}$.

5. The method of claim 4 wherein, a size of the at least one backup buffer is $\geq m*x \geq x*\frac{\log\left(0.1*s/e_r*x\right)}{\log P_{loss}}$, where m is a number of retransmits that will reduce the loss rate for older packets to a negligibly small number, and $m*x$ is a minimum required size of the at least one player buffer.

6. A method of detecting and reacting to a network congestion which interferes with streaming of multimedia audio and video packets from at least one sending computing device to an at least one receiving computing device via a network, said method comprising the steps of:

a) communicating configuration information including network parameters from a service manager via the network to the at least one sending computing device and to the at least one receiving computing device, the at least one sending computing device comprising at least one server process, the at least one receiving computing device comprising at least one player process;

b) confirming the network congestion by transmitting and retracting network probes;

fuboTV Media Inc.
Exhibit 1002
Page 339

c) executing at least one congestion control technique to limit degradation in the streaming of the packets;

d) detecting an end of the network congestion by transmitting and retracting said network probes;

e) configuring a rate of streaming of the packets according to said network parameters

7. The method of claim 6, wherein said at least one congestion control technique is the constant bit-rate emulation.

8. The method of claim 7, wherein said at least one congestion control technique only reduces a transmission rate of the packets by the amount of incurred loss of the packets.

9. The method of claim 7, further comprising the steps of:

detecting an end of congestion;

computing available bandwidth in the network; and

transmitting packets according to said bandwidth.

10. The method of claim 9, wherein some of the at least one player process are differentiated and protected from effects of said network congestion to a greater than others by receiving a stream backlog sooner.

11. The method of claim 10, wherein some of the at least one player process are differentiated and protected from effects of said network congestion to a greater than others by receiving a stream backlog sooner according to a rule for transmission rate of the packets

$$e_t = \min(\alpha^j \times SR_{min}, \alpha^j \times \frac{B - \lambda_{other}}{\sum_{i=1}^{i=K} \alpha^i n^i})$$

, where e_t indicates said transmission rate of packets, α^j is a weighting function, for class $\{1 \ldots K\}$.

fuboTV Media Inc.
Exhibit 1002
Page 340

12. The method of claim 11, wherein said detecting of said end of the network congestion is performed according to the expected utilization $\rho = 1 - \frac{SR_{avg}}{B}$, where SR_{avg} is the average rate at which the packets are transferred from the at least one server process to the at least one player process and B is a bottleneck link capacity at a bottleneck hop.

13. The method of claim 12, wherein said network probes compute an approximate service rate of transferring the packets from said at least one server process to said at least one player process according to the following equation:

$SR^{min}_{m,m-1} = \frac{s}{\frac{s}{B_{m,m-1}} + q^{max}_{m,m-1}}$ and $SR^{avg}_{m,m-1} = \frac{s}{\frac{s}{B_{m,m-1}} + q^{avg}_{m,m-1}}$, where SR_{avg} and SR_{min} are averaged and minimum service rate, $q^{max}_{m,m-1}$ is the maximum queuing delay between a node m and a node m-1.

14. A method of guaranteed delivery of streaming multimedia audio and video packets from at least one sending computing device to an at least one receiving computing device via a network, said method comprising the steps of:

a) continuously processing arriving packets;

b) keeping track of loss of the packets at the at least one receiving computing device, said at least one receiving computing device comprising one or more player processes;

c) determining if said loss exceeds a first threshold during a first interval;

d) verifying congestion symptoms;

e) communicating said verified congestion symptoms to the at least one sending computing device, said at least one sending computing device comprising one or more server processes;

fuboTV Media Inc.
Exhibit 1002
Page 341

f) monitoring said loss and determining if said loss exceeds a second threshold during a second interval to determine an end of congestion; and

g) verifying congestion symptoms.

15. The method of claim 14, wherein said step of keeping track is performed at user define intervals.

16. The method of claim 15, wherein said first threshold, said second threshold, said first interval, and said second interval are user defined.

17. The method of claim 16, wherein said congestion symptoms are verified by receiving a remote congestion confirmed report.

18. The method of claim 17, wherein said congestion symptoms are verified by performing a congestion probe.

fuboTV Media Inc.
Exhibit 1002
Page 342

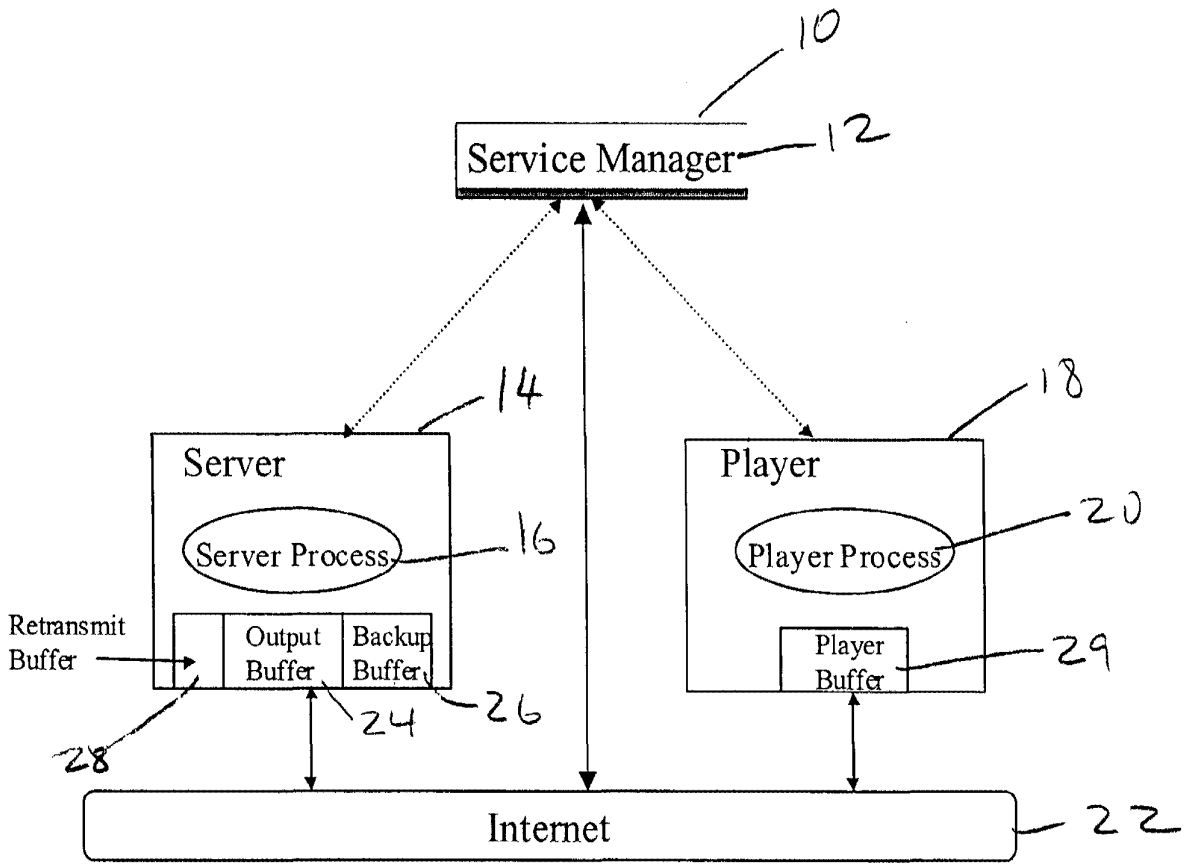


Figure 1

fuboTV Media Inc.
Exhibit 1002
Page 343

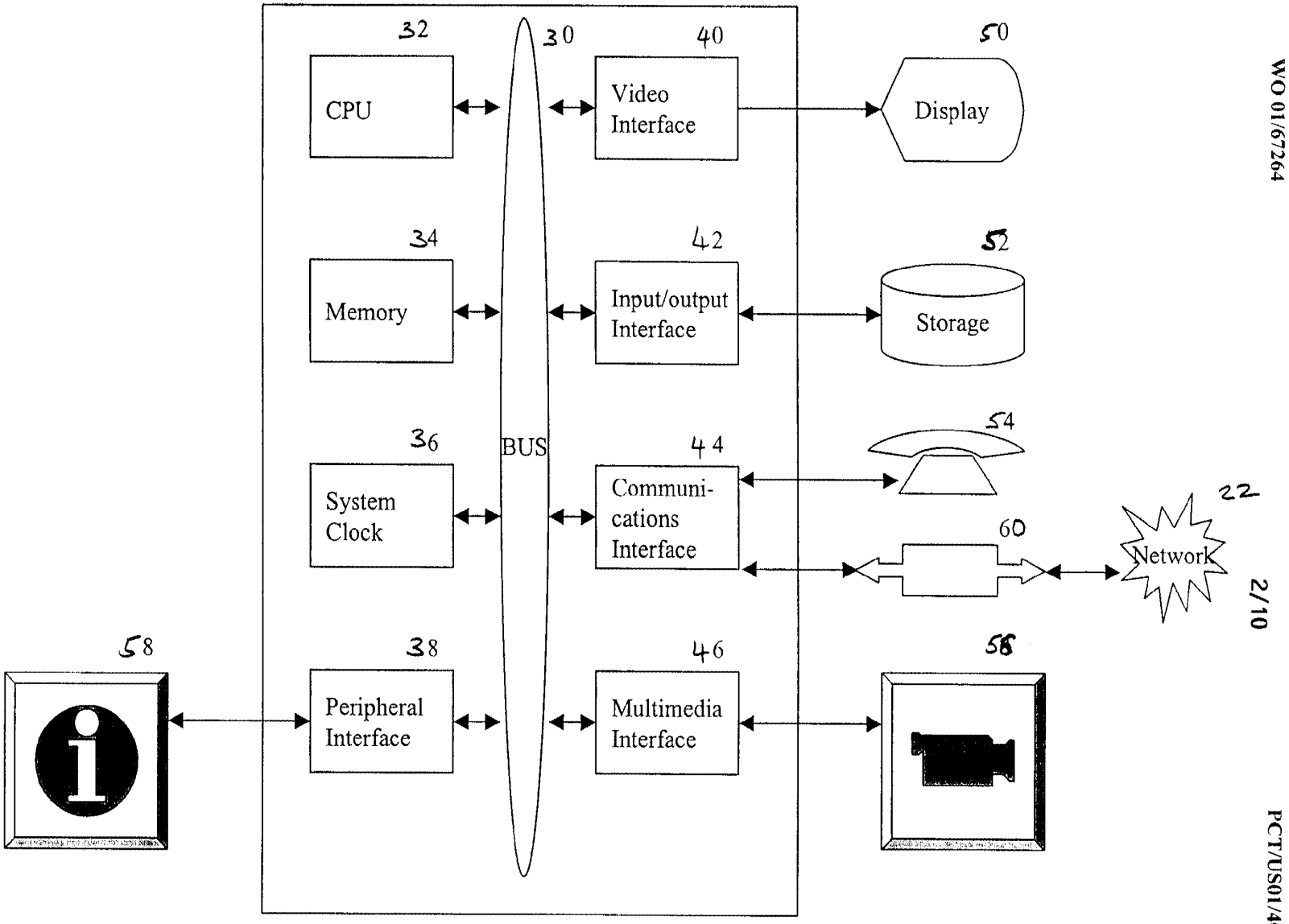


Figure 2

fuboTV Media Inc.
Exhibit 1002
Page 344

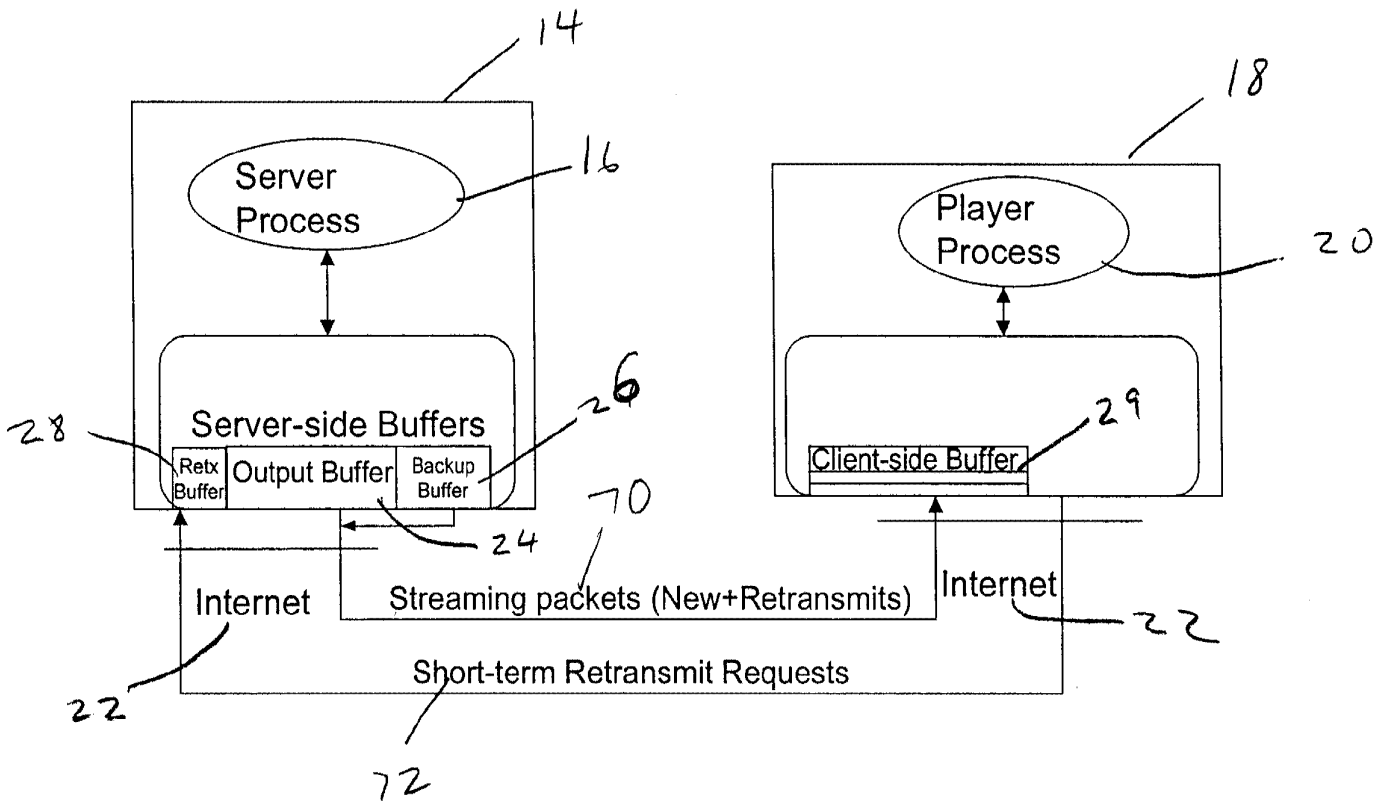

14
18
Server Process
16
Player Process
20
Server-side Buffers
26
28
Retx Buffer
Output Buffer
Backup Buffer
Client-side Buffer
29
24
70
Internet
Streaming packets (New+Retransmits)
Internet
22
22
Short-term Retransmit Requests
72

Figure 3

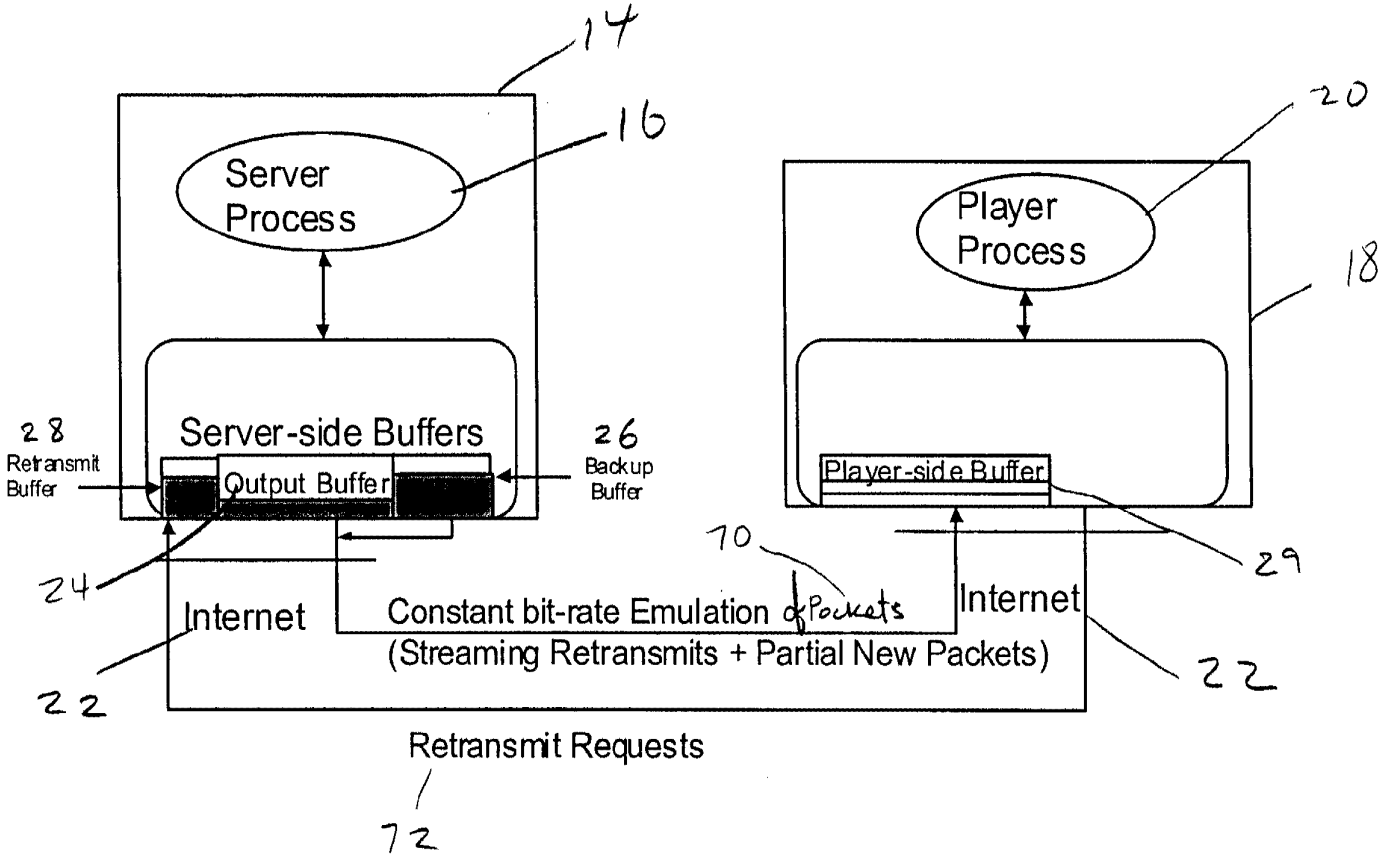

14
16
Server Process
20
Player Process
18
28
Retransmit Buffer
Server-side Buffers
Output Buffer
26
Backup Buffer
Player-side Buffer
29
24
70
Internet
Constant bit-rate Emulation of Packets
(Streaming Retransmits + Partial New Packets)
Internet
22
22
Retransmit Requests
72

Figure 4

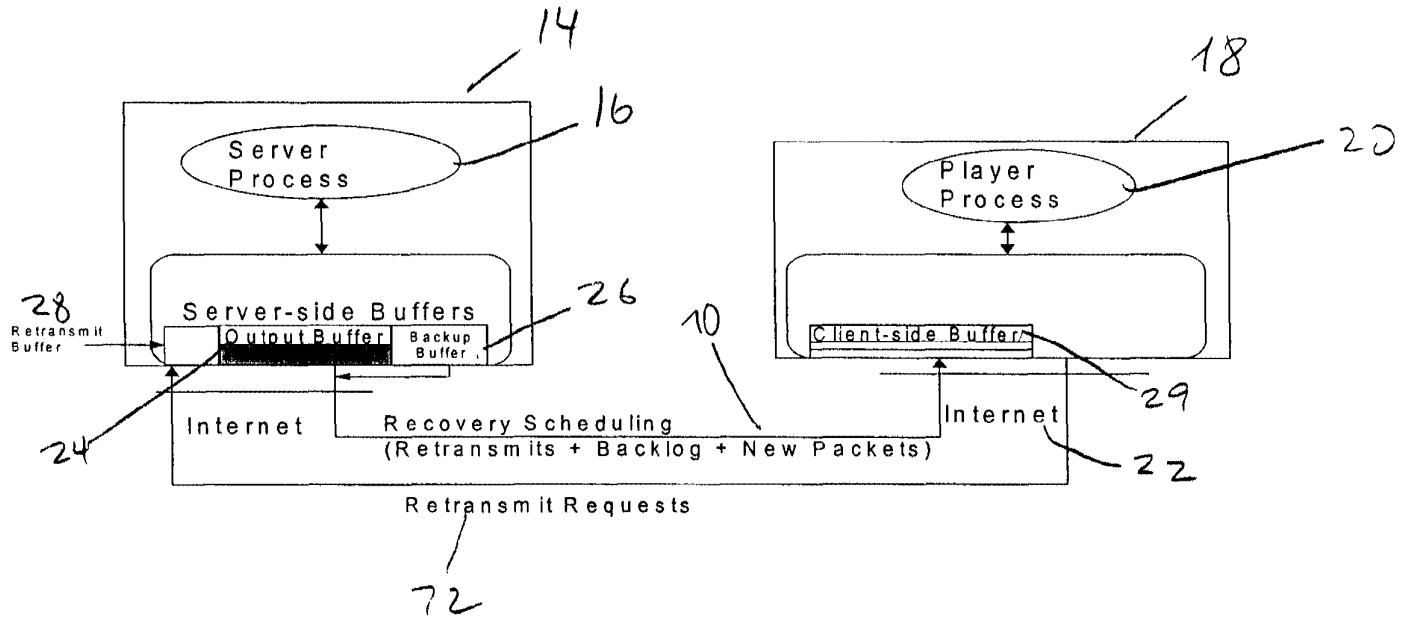


Figure 5

fuboTV Media Inc.
Exhibit 1002
Page 347

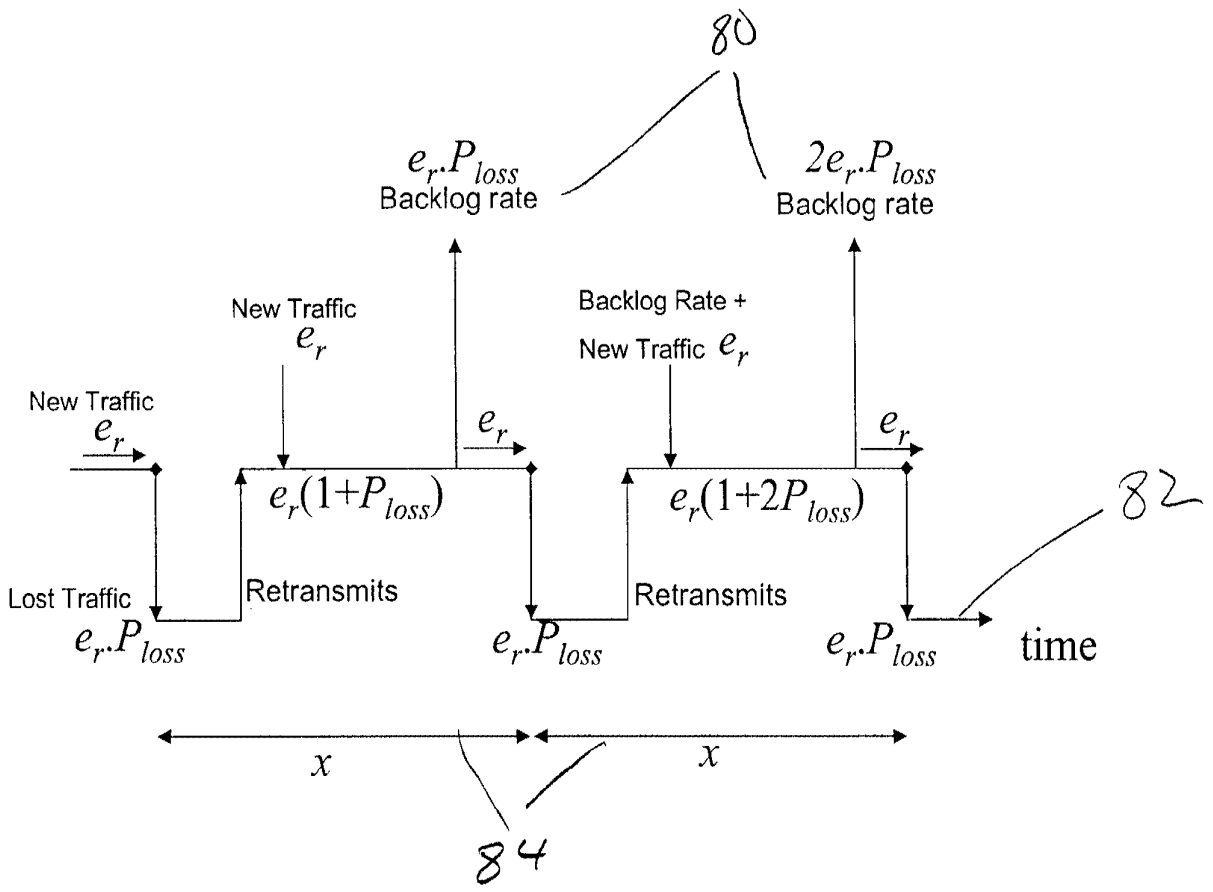


Figure 6

fuboTV Media Inc.
Exhibit 1002
Page 348

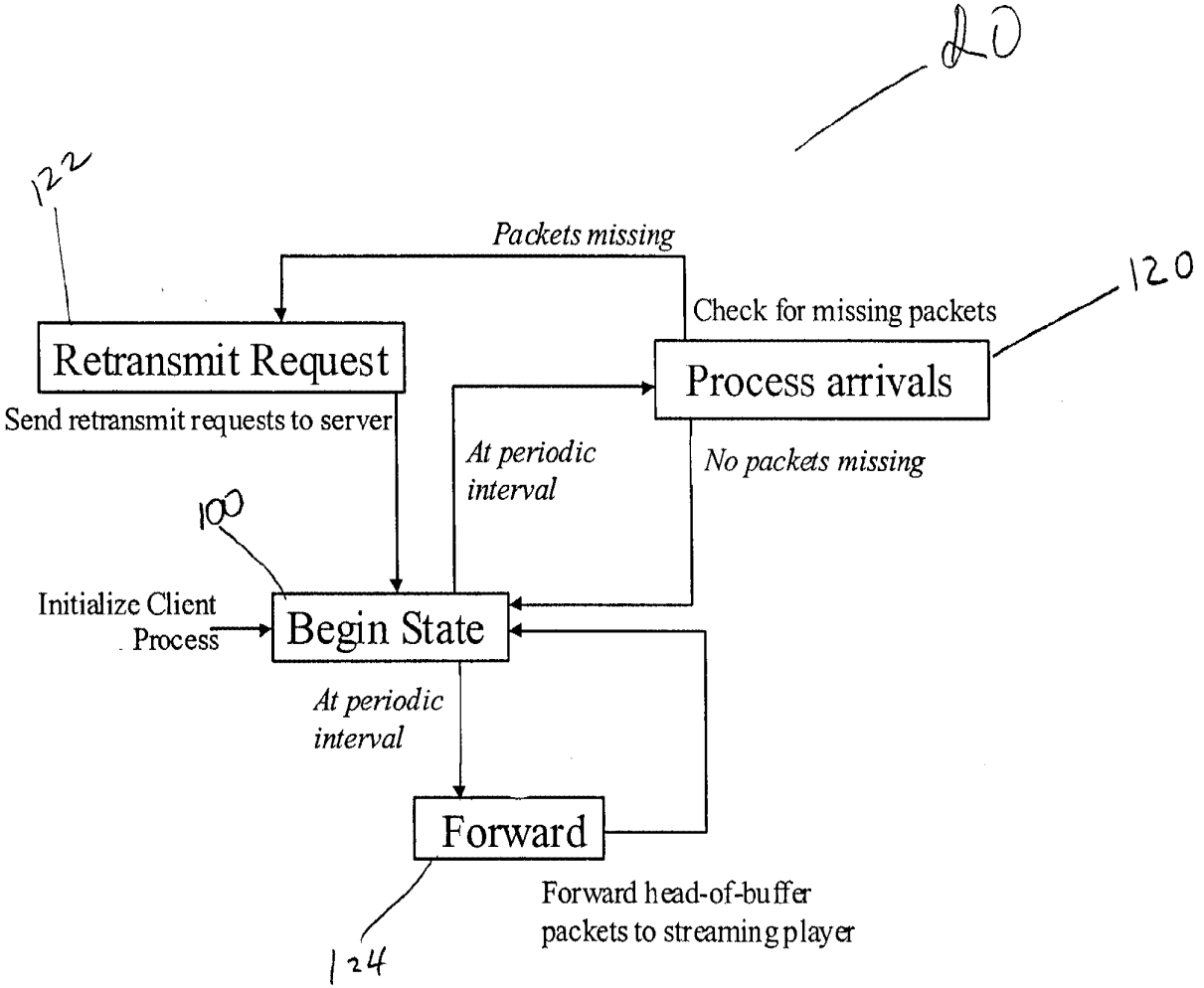

Packets missing
Check for missing packets
Retransmit Request
Process arrivals
Send retransmit requests to server
At periodic interval
No packets missing
Initialize Client Process
Begin State
At periodic interval
Forward
Forward head-of-buffer packets to streaming player

Figure 7

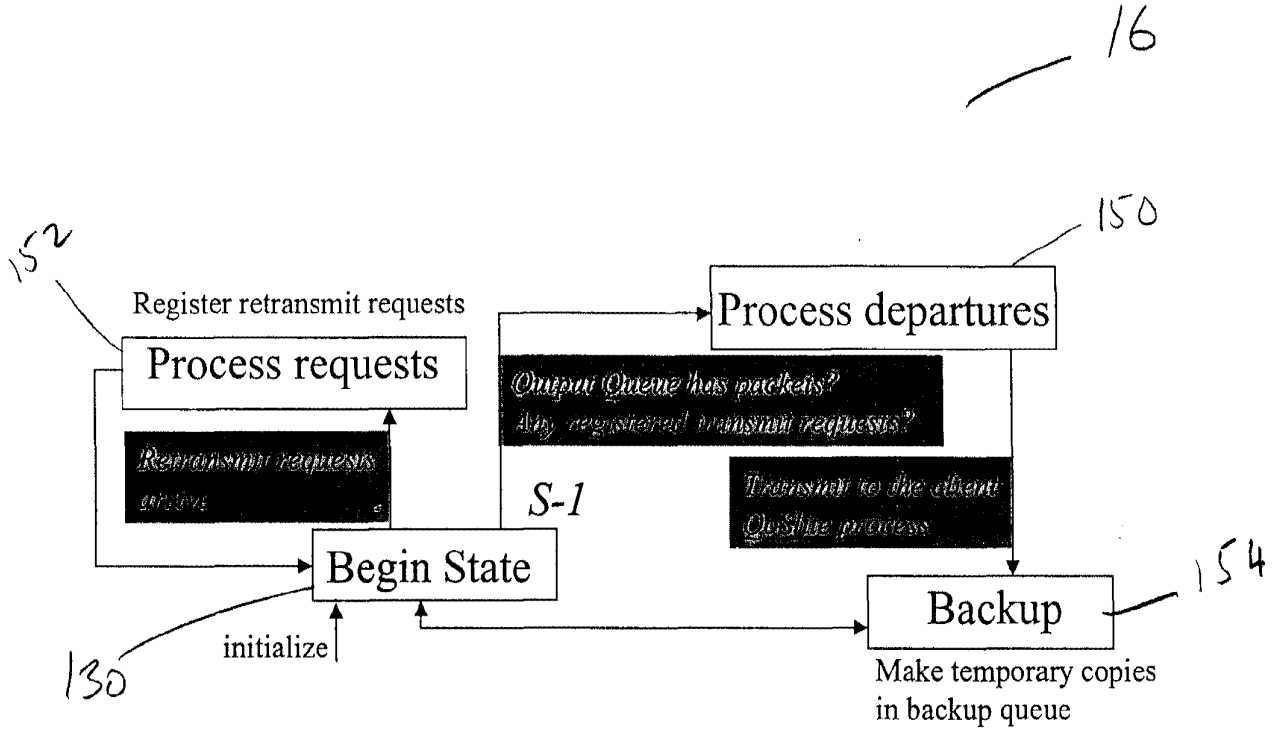


Figure 8

fuboTV Media Inc.
Exhibit 1002
Page 350

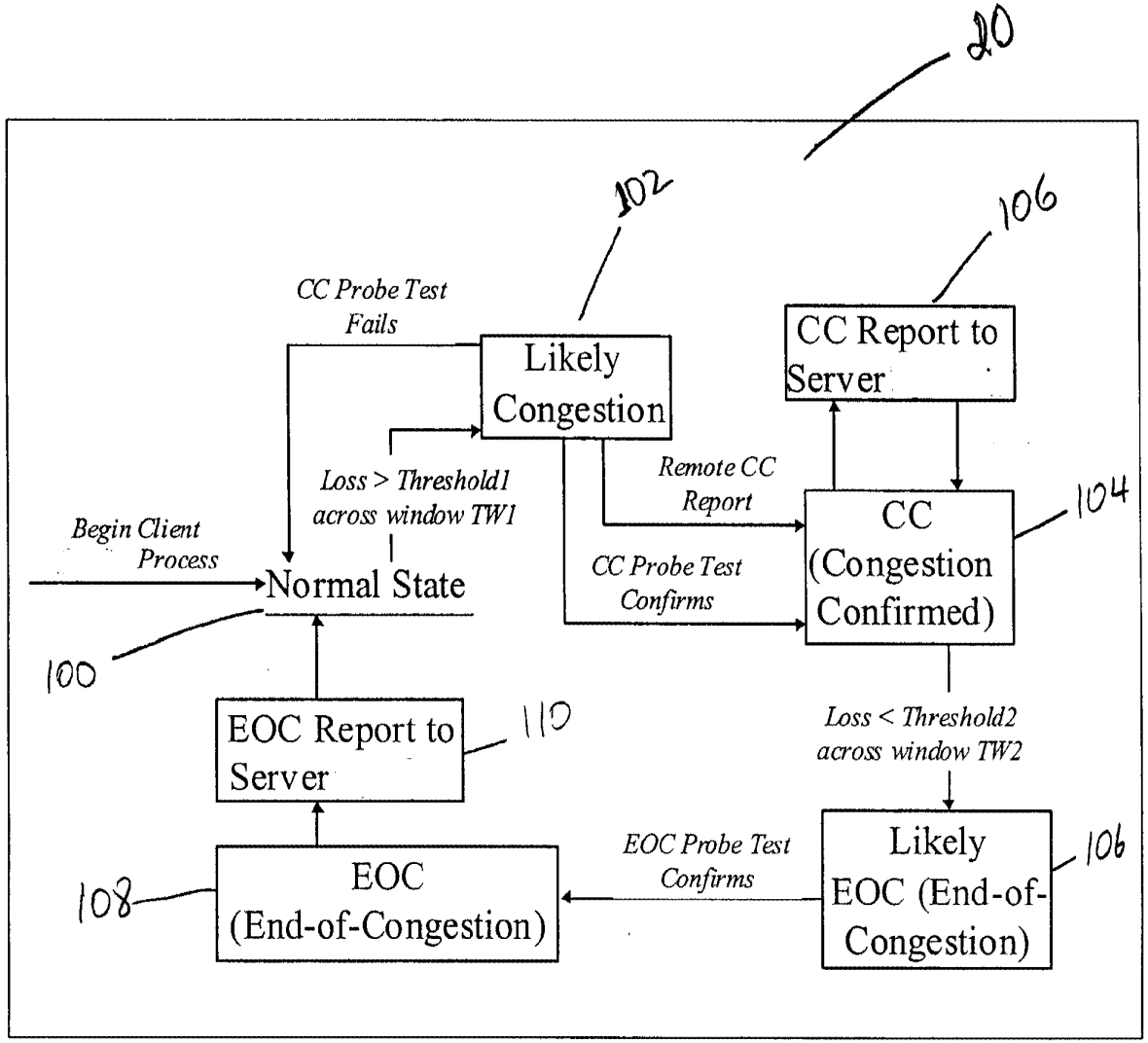


Figure 9

fuboTV Media Inc.
Exhibit 1002
Page 351

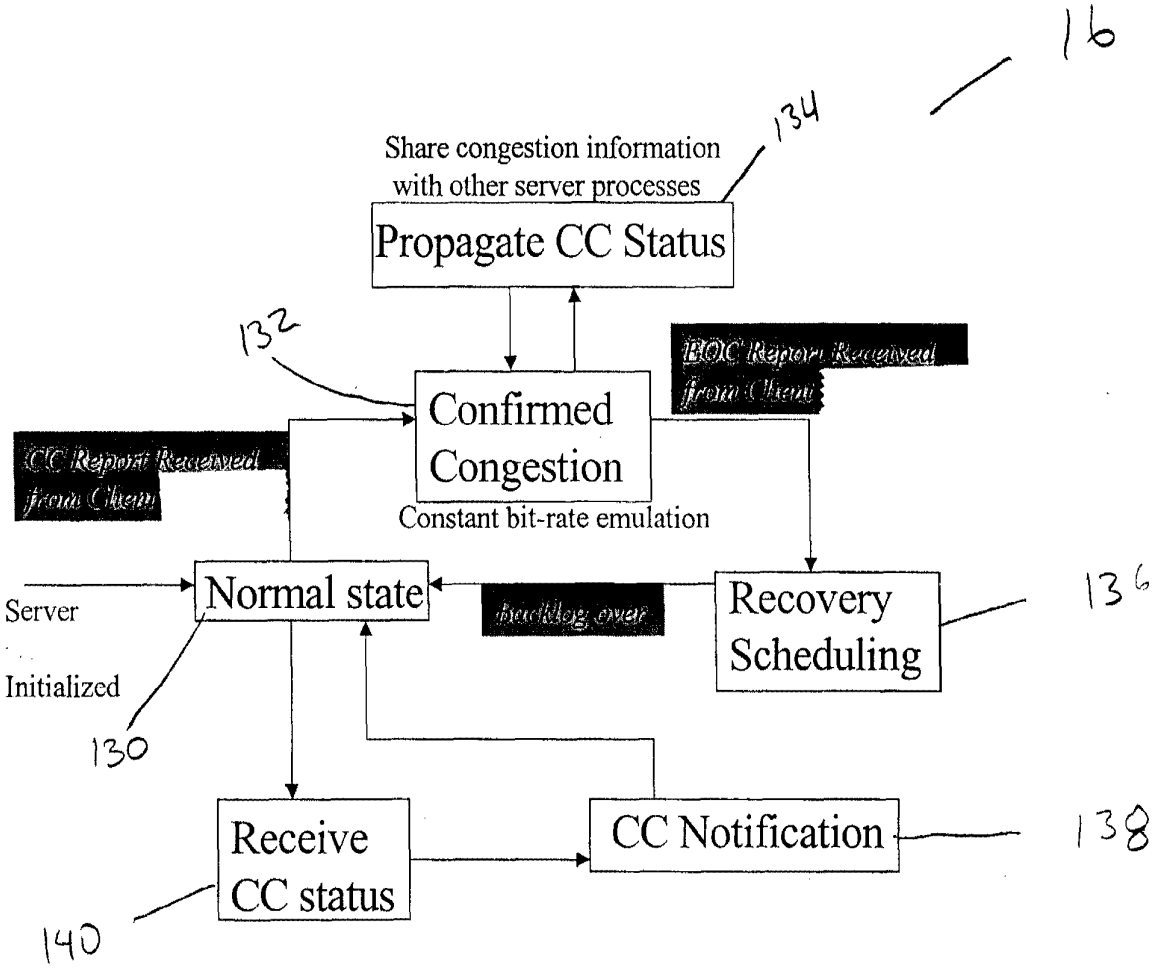


Figure 10

fuboTV Media Inc.
Exhibit 1002
Page 352

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/40264

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 15/16
US CL :709/231, 232, 235

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/231, 232, 235

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

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

EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 6,031,818 A (LO et al) 29 February 2000, col. 2, line 54-col. 5, line 28.	1, 2
X,P	US 6,018,515 A (SORBER) 25 January 2000, col. 4, line 18-col. 10, line 65.	6-10, 14-18
A, P	US 6,014,706 A (CANNON et al) 11 January 2000, see all.	1-18
A	US 5,339,392 A (RISBERG et al) 16 August 1994, see all.	1-18

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

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

Date of the actual completion of the international search 06 JUNE 2001	Date of mailing of the international search report 27 July 2001 (27.07.01)
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer Peggy Harwood TOD KUPSTAS Telephone No. 703-305-2655

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

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 202 487 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.05.2002 Bulletin 2002/18

(51) Int Cl.[7]: **H04L 1/00**

(21) Application number: **01309276.2**

(22) Date of filing: **31.10.2001**

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

(30) Priority: **31.10.2000 JP 2000333916**
29.06.2001 JP 2001200156

(71) Applicant: **Kabushiki Kaisha Toshiba**
Tokyo 105-8001 (JP)

(72) Inventors:
- **Nagai, Takeshi, c/o Kabushiki Kaisha Toshiba**
  Minato-ku, Tokyo 105-8001 (JP)
- **Kikuchi, Yoshihiro,**
  c/o Kabushiki Kaisha Toshiba
  Minato-ku, Tokyo 105-8001 (JP)
- **Masuda, Tadaaki, c/o Kabushiki Kaisha Toshiba**
  Minato-ku, Tokyo 105-8001 (JP)

(74) Representative: **Midgley, Jonathan Lee**
Marks & Clerk
57-60 Lincoln's Inn Fields
GB-London WC2A 3LS (GB)

(54) Data transmission apparatus and method

(57) A data transmission apparatus comprises a network condition determination device for determining a transmission condition of the transmission channel in the transmitter (103) based on at least Jitter information or a packet loss rate obtained from a receiver (104), and a video encoder (102) including means for changing at least one of a bit rate of transmission data and a error resilience level according to the transmission condition.

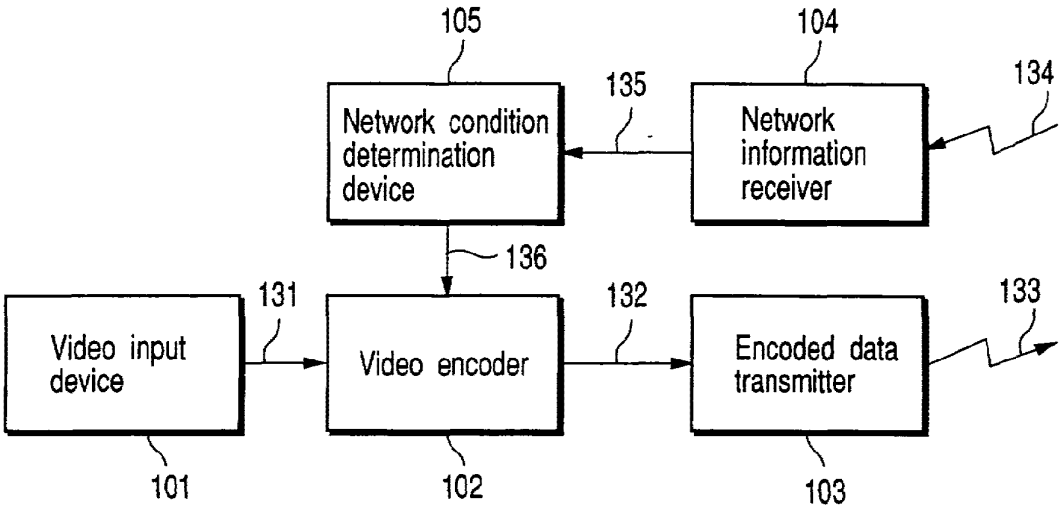


FIG. 1

EP 1 202 487 A2

Printed by Jouve, 75001 PARIS (FR)

fuboTV Media Inc.
Exhibit 1002
Page 354

Description

[0001] The present invention relates to a data transmission method for transmitting encoded moving/still picture by using a wired communication network such as ISDN (Integrated Services Digital Network) or a wireless communication network such as PHS (Personal handy-phone System) or satellite communication and a data transmission apparatus to which such method is applied.

[0002] In recent years, with progress of digital encoding technology of a variety of information such as image and a broadband network technology, the development of applications using these technologies becomes active, and a system of transmitting compression-encoded image by utilizing a communication network is developed.

[0003] For example, an input video signal is encoded at a video encoder, the obtained encoded data is provided to an encoded data transmitter, and transmission data is transmitted to a transmission channel. In this way, a video is encoded and transmitted. In recent years, with the spread of Internet/Intranet, there has been increased an application or system of transmitting/receiving a packet of data. Providing data as a packet is very effective means for efficiently sharing a bandwidth of a communication channel by a plurality of users.

[0004] In the meantime, as a protocol for transmitting/receiving packet data via Internet/Intranet, there exists TCP/IP (Transmission Control Protocol/Internet Protocol) or UDP/IP (User Datagram Protocol/Internet Protocol) and the like.

[0005] Among them, TCP/IP is an Internet standard protocol. This protocol is applied as the world's best known protocol to Intranet or LAN as well as Internet. TCP works in a transport layer that is a fourth layer of an OSI basic reference model, and IP works in a network layer that is a third layer.

[0006] In addition, UDP/IP is one of the TCP/IP suite transport layer protocols. TCP is a connection type protocol. UDP is a disconnection type protocol. As with TCP, a host application is identified by using a port number. This protocol is often used to carry short data for monitor control of a network or real time oriented data such as audio data or video data.

[0007] TCP/IP incorporates a frame such as retransmission. Thus, this protocol is strong against errors or the like. Even if a large amount of time is required, this protocol is effective to download type applications if an attempt is made to validly receive data. However, this protocol is not powerful for applications requiring real time properties.

[0008] In contrast, although UDP/IP does not have a retransmission frame, a delay associated with retransmission does not occur. Thus, this protocol is very effective to applications requiring real time processing.

[0009] Transmission of moving picture is exemplified as a typical example of application requiring real time processing. However, in the case of general video communication, video data has a very large amount of data, and overflows a network bandwidth in most cases. In this case, a technique for encoding a video data signal, and transmitting a reduced amount of data is used. Techniques such as motion compensation, discrete cosine transform (DCT), sub-band encoding, pyramid encoding, and variable-length encoding or a combination of these techniques are developed as compression-encoding of a video signal.

[0010] As an international standard system for moving picture encoding, there are ISO MPEG-1, MPEG-2, ITU-T H. 261, H. 262, and H. 263. In addition, as an international standard system for multiplying an encoded data stream having a speech/audio signal compressed or other data, there are an ISO MPEG system, ITU-T H. 221, and H. 223.

[0011] Internet or the like is encompassed via an infinite number of networks. Usually, no one knows which network is used or what state is. In addition, an amount of data flowing a network changes from time to time. Thus, a system of judging what amount of data can be communicated in a real time is required.

[0012] Therefore, with further advancement from real time applications utilizing UDP/IP, there have been increased in number applications using a packet format called RTP (Real-time Transport Protocol) for adding and transmitting time information or the like to a packet.

[0013] This RTP is a protocol for transferring audio data signal or video data signal ruled in RFC 1889 in real time. Usually, data is carried on UDP (User Datagram Protocol). This protocol is assumed to be applied to a multimedia system such as video conference, and data can be exchanged in a real time interactive scheme. However, the protocol has no sound quality or picture quality assurance function. Sequential number or time stamp data is assigned into an RTP header, thereby supporting a read time operation.

[0014] RFC is an abbreviation of Request for Comments, and denotes a document such as technical proposal or comments disclosed by TETF (Internet Engineering Task Force). Many de-fact standards such as a variety of protocols of TCP/IP suite are described in RFC.

[0015] By utilizing this RTP, time information and a packet number are added to a packet, so that a receiver can display a voice or image by using correct time information, can determine a packet or the like whose order is replaced in a network, and can detect that a packet is lost by referring to a packet number.

[0016] Moreover, the RTP provides with a system (RTCP) for notifying network state information such as jitter or packet loss rate from a transmitter to a receiver.

[0017] However, how this RTCP information is utilized depends on applications, and is not determined in standards.

[0018] In addition, in the case of an image, a bandwidth corresponding to video transmission cannot be al-

fuboTV Media Inc.
Exhibit 1002
Page 355

located in a network bandwidth, thus making it necessary to compress and deliver a video signal in an encoding system such as MPEG, as described previously. Although this encoding system is effective in a decreased amount of data, the system becomes very fragile to a packet loss or error entry by supplying data to unstable Internet. This is because a moving picture encoding system transmits only a difference between a current frame and a previous frame. Thus, partial data missing causes a very serious problem. In the case of using the UDP or RTP, data retransmission is not basically carried out, thus making it necessary to counteract this problem.

[0019] There generally exist two modes in picture encoding, that is, one is an inter-frame encoding mode for transmitting a difference between the current frame and the previous frame and an intra-frame encoding mode for carrying out encoding in one frame. In general, intra-frame encoded pictures are provided at a proper timing, and inter-frame encoded pictures are provided between them. An interval between the frames encoded in an intra-frame encoding mode is called a GOP (Group of Picture) interval.

[0020] When an encoded picture is missing during transmission to the decoder, the decoded picture corresponding to the missing encoded picture is destroyed. Thereafter, the intra-encoded pictures following the missing encoded picture are decoded based on the destroyed picture. Thus, all the subsequent decoding pictures are affected by the destroyed picture, and decoding cannot be carried out correctly. Therefore, an intra-frame encoded picture is inserted in the way of a stream of frames to cut off a propagation of missing error and recover adequate decoding.

[0021] In a conventional technology, it can determine that errors occur in a network. However, how to utilize the determination result has been unknown. Also, a system for notifying the network information to the transmitter in an unique format is conceived. However, the system is based on the specification of associated applications and lacks a general-purpose use. The system receives transmission data using an information receiver, and controls a video encoder within a range of the information received from the network via a transmission channel.

[0022] A conventional network condition determination is conceived on assumption of network congestion. However, since Internet is popularized in mobile environment, it is required to consider an error or the like in wireless environment. Current technique does not assume countermeasures against such error.

[0023] Further, in the case of dealing with the error, a case in which a network condition changes, and an error rate is not constant is not considered. If a short GOP interval is set in order to counteract a case in which errors frequently occur, the intra-frame encoding increases resulting in impairing the encoding efficiency. When errors occur frequently qualitatively, it may be required to establish such state in which such errors occur frequently. However, in the case of a network in which an error hardly occurs in a normal state, and an error occurs only at one moment, this setting is very wasteful. Conversely, when a long GOP interval is set, it is troublesome that a great influence occurs when an error occurs.

[0024] Thus, a real time transmission using RTP is noted. As described above, by utilizing the RTP that is a protocol for transferring audio data signal or video data signal in real time, time information and a packet number are added to a packet, whereby the receiver can display a voice or image by using correct time information, can determine a packet or the like whose order is replaced in a network, or can detect that a packet is lost by referring to a packet number.

[0025] Moreover, at the RTP, there is provided a system (RTCP) for notifying network state information such as jitter or packet loss rate from the transmitter or receiver.

[0026] It is an object of the present invention to provide a data transmission apparatus and method capable of adjusting a transmission rate by using the RTP characteristics in accordance with a transmission state of a transmission channel based on notification of network state information such as jitter or packet loss rate, obtained from the receiver in the transmitter or capable of carrying out controls such as changing error resilience, thereby making it possible to efficiently transmit data to the maximum, and fully utilize transmission that requires real time properties.

[0027] According to an aspect of the invention, there is provided a data transmission apparatus that transmits transmission data from a transmitter to a receiver via a transmission channel, comprising: an estimation device configured to estimate a transmission condition of the transmission channel in the transmitter based on at least Jitter information or a packet loss rate obtained from the receiver; and a controller configured to change at least one of a bit rate of transmission data and a error resilience level according to the estimated transmission condition.

[0028] According to another aspect of the invention, there is provided a data transmission method comprising: estimating a transmission condition of the transmission channel in the transmitter based on at least Jitter information or a packet loss rate obtained from the receiver; and changing at least one of a bit rate of transmission data and an error resilience level according to the estimated transmission condition.

[0029] According to another aspect of the invention, there is provided a data transmission program stored on a computer readable medium, comprising: instruction means for instructing a computer to estimate a transmission condition of the transmission channel in the transmitter based on at least Jitter information or a packet loss rate obtained from the receiver; and instruction means for instructing the computer to change at least

fuboTV Media Inc.
Exhibit 1002
Page 356

one of a bit rate of transmission data and a error resilience level according to the estimated transmission condition.

[0030] This summary of the invention does not necessarily describe all necessary features so that the invention may also be a sub-combination of these described features.

[0031] The invention can be more fully understood from the following detailed description when taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a view showing a basic configuration of a data transmission apparatus according to a first embodiment of the present invention;
FIG. 2 is a view showing another configuration in the present embodiment;
FIG. 3 is a view showing a configuration of a network condition determination device in the present embodiment;
FIG. 4 is a view showing a configuration of a video encoder in the present embodiment;
FIG. 5 is a view showing a configuration of an encoded parameter determining device in the present embodiment;
FIG. 6 is a view showing a basic configuration of a data transmission apparatus in a second embodiment of the present invention;
FIG. 7 is a view showing a configuration of an encoded data switch device in the present embodiment;
FIG. 8 is a view illustrating a timing of encoded data switching at an encoded data switch device in the present embodiment;
FIG. 9 is a view illustrating a timing of switching encoded data with its different frame interval at the encoded data switch device in the present embodiment;
FIG. 10 is a flow chart showing a basic example of the network condition determination method at the network condition determination device in the first embodiment of the present invention;
FIG. 11 is a flow chart showing a basic example when a delay is forced to occur by using the network condition determination method at the network condition determination device in the first embodiment of the present invention;
FIG. 12 is a flow chart showing a basic example considering a time direction change with the network condition determination method at the network condition determination device in the present embodiment;
FIG. 13 is a view showing an exemplary configuration according to a third embodiment of the present invention;
FIG. 14 is a flow chart showing an example of a GOP interval calculation method at a GOP interval calculator in the present embodiment;
FIG. 15 is a flow chart showing another example of a GOP interval calculation method at a GOP interval calculator in the present embodiment; and
FIG. 16 is a flow chart showing still another example of a GOP interval calculation method at a GOP interval calculator in the present embodiment.

[0032] Hereinafter, embodiments of the present invention will be described with reference to the accompanying drawings.

[0033] The present invention utilizes RTP or RTCP that is a protocol for transferring audio data signal or video data signal in real time. The transmitter performs controls such as adjusting a transmission rate or changing error resilience in accordance with a transmission state of a transmission channel based on notification of network state information acquired from the receiver such as jitter (Jitter information) or packet loss rate. In this manner, data transmission can be carried out efficiently to the maximum. Hereinafter, embodiments of the present invention will be described in detail.

(First Embodiment)

[0034] FIG. 1 shows a basic configuration of an image transmission apparatus according to a first embodiment of the present invention. In FIG. 1, a video signal 131 input from an image input device 101 is processed to be encoded. That is, an encoded data transmitter 103 processes encoded data 132 that has been encoded by a video decoder 102 in the form which conforms to a network, and transmits the encoded data to a receiver. A network information receiver 104 receives network information 134 delivered from the receiver, and outputs the received information to a network condition determination device 105. This network condition determination device 105 has a function that determines a network condition from network information 135 output from the network information receiver 104, and notifies the result as network condition information 136 to the video encoder 102.

[0035] The video encoder 102 encodes the video signal 131 using the network condition information 136. This encoding is carried out as follows. That is, in the case of RTP or RTCP, a packet loss rate or Jitter information is provided to the network condition determination device 105. The network condition determination device 105 determines that, if a packet loss rate is not "0" or if a delay time is longer than a predetermined value, any load is applied to a network, and a presumed amount of data cannot be supplied. Then, the network condition determination device 105 is configured so as to carry out processing so as to issue an instruction (network condition information 136) for lowering a bit rate to the video encoder 102, and to lowly set a target bit rate in encoding.

[0036] According to the thus configured apparatus, when the video signal 131 is inputted from the image input device 101, the inputted video signal 131 is encod-

fuboTV Media Inc.
Exhibit 1002
Page 357

ed by the video encoder 102. Encoded data 132 encoded by the video encoder 102 is inputted to the encoded data transmitter 103.

[0037] The encoded data transmitter 103 processes the encoded data in the form suitable to a network, and receives the processed data to a receiver. The network information receiver 104 receives the network information 134 delivered from the receiver, and outputs the received information to the network condition determination device 105.

[0038] The network condition determination device 105 determines a network condition from the network information 135 outputted from a network information receiver, and notifies the result as the network condition information 136 to the video encoder 102.

[0039] The video encoder 102 encodes the video signal 131 using the thus notified network condition information 136.

[0040] FIG. 10 shows a very simple example of a method for determining a network condition at the network condition determination device 105 by way of showing a flow chart.

[0041] According to this flow chart, at the step S1001, a packet loss rate of network information is checked to see if the rate is "0" or not. As the result of this check, if the packet loss is not "0", an effective bit rate is calculated from the packet loss rate in accordance with the step S1002. This bit rate can be calculated by formula (1) shown below.

$$b' = b \times (1 - r) \quad (1)$$

where "b" denotes a current bit rate, b' denotes a new bit rate, and "r" denotes a packet loss rate.

[0042] Next, at the step S1003, a parameter (encoding parameter) obtained at the step S1002 is notified to the video encoder 102. Here, the above parameter is used to determine a rate control or error resilience level for encode processing at the image encoder 102. This parameter is also used for adjustments such as adjusting and controlling the encode processing speed of the input image signal 131 to this parameter, thereby changing a bit rate, a frame interval, or an error resilience level.

[0043] On the other hand, at the step S1001, when the packet loss rate is "0", no parameter change occurs at the step S1004. This determination method is provided as a mere example. It is possible to employ any other method for obtaining a bit rate based on a case in which the criterion of the step S1001 is determined by thresholds without being set to "0" or others or based on a formula other than the above formula (1).

[0044] Therefore, according to the first embodiment, even when a transmission network bandwidth is unknown or when the bandwidth changes halfway, it is possible to reset a parameter or the like based on the receiver's network information, and then, encode and transmit an image in accordance with the parameter suitable to a current network. In this manner, even in the case of a network congestion, although a bit rate is automatically lowered, and an image quality is lowered, it is possible to communicate an image without an occurrence of a phenomenon that an image is destroyed due to such packet loss. This method is very effective in image transmission in real time.

[0045] The first embodiment can be carried out by modifying it as follows. For example, as shown in FIG. 11, a method other than that shown in FIG. 10 is employed as a method for determining a network condition at the network condition determination device 105. That is, in the flow chart shown in FIG. 10, a parameter is set at only a packet loss rate. In contrast, in the method shown in FIG. 11, even when no packet loss occurs at the step S1104, a delay quantity is checked. As a result of this check, if a delay occurs in excess of a predetermined value, it is determined that a network is congested. When it is determined that a network is congested, a bit rate is calculated, and a new parameter is set by using formula (2) below, for example, at the step S1105.

$$b' = b \times \{(TN-TS)+(d-dth)\}/(TN-TS) \quad (2)$$

where "b" denotes a delay time, b' denotes a new bit rate, TN denotes a current time, TS denotes a start time, "d" denotes a delay time, and "dth" denotes delay time thresholds.

[0046] FIG. 2 shows an example when the network condition determination device 105 is provided with a function for generating the network condition information 136 so as to provide the information to the image encoder 102 by extending the configuration shown in FIG. 1.

[0047] According to a configuration shown in FIG. 2, the network condition determination device 105 is provided with a function for acquiring a current parameter at the current image encoder 102 and a function for generating the network condition information 136 on what parameter should be set next from the thus acquired current parameter and network information 135, thereby providing the parameter to the image encoder 102.

[0048] In the case of this construction, the network condition determination device 105 can recognize a state of a video parameter, i.e., a state of the encode processing currently done at the video encoder 102, thus making it possible for the network condition determination device to calculate and grasp a bit rate or the like. This construction is advantageous in the case of determining a network condition by comparing the network information 135 with parameter information 137. Thus, a system is realized which is capable of changing a parameter so as to set an optimal bit rate in accordance with the network condition, and providing the parameter to the video encoder 102, thereby adjusting the encode processing.

[0049] In this way, according to the present invention,

fuboTV Media Inc.
Exhibit 1002
Page 358

transmission is adjusted in accordance with a transmission state of a transmission channel or error resilience is changed. The transmission state is adjusted by changing a bit rate or changing a frame interval. With respect to a change in quality of a transmission channel, the error resilience is changed according to the transmission state of the transmission channel (for example, the error resilience is changed by changing an interval of an MPEG4 synchronous signal). Therefore, data transmission can be carried out efficiently to the maximum, thereby enabling data transmission that makes it possible to fully utilize transmission requiring real time processing.

[0050] FIG. 3 shows a configuration when the configuration shown in FIG. 2 is further developed. This network condition determination device 105 comprises a network information storage device 301 that stores past network information and an encoded parameter information storage device 302 that stores encoded parameter information supplied to the current video encoder 102. A network condition determination device 303 recognizes the past network information and the current encoded parameter from these elements, decides an elapse of time in network condition from these elements to determine an optimal parameter assigned to the video encoder 102, and provides the parameter to the video encoder 102.

[0051] With this construction, it is possible to determine a network condition more accurately by referring to an elapse of time relevant to network information. FIG. 12 shows a flow chart of a network condition determination method in the case of using this system. Processing at the network condition determination device 303 will be described in accordance with this flow chart. First, the network condition determination device 303 determines whether or not packet loss is "0" at the step S1201. As a result of this determination, when the packet loss is present, it is determined by the step S1202 whether or not the past packet loss occurs. As a result, when the past packet loss occurs, the loss rate is compared with the current rate. If the current packet loss rate is higher than the past packet loss rate, it can be determined that the past correction has been invalid.

[0052] It is understood from this determination that the bandwidth overflow of transmission data such as network congestion is not troublesome, but a possibility that a noise or the like occurs over a channel resulting in destroying is high. Therefore, the network condition determination device 303 does not change a bit rate at the step S1204, and carries out parameter setting to ensure reliable error correction, and notifies the fact to the video encoder 102 at the step S1205.

[0053] If the current packet loss rate is lower than the past packet loss rate at the step S1203, the network condition determination device 303 determines that the past change works efficiently, and re-calculates a valid bit rate from the packet loss rate at the step S1206. When the past packet loss does not occur at the step S1202, the last loss is determined as a channel noise, and processing goes to the step S1204.

[0054] On the other hand, when no packet loss occurs at the step S1201, the network condition determination device 303 checks a delay quantity at the step S1207. As a result, when the delay quantity exceeds thresholds "Th", a valid bit rate is calculated from the delay quantity at the step S1208, and the calculation result is notified to the video encoder 102 at the step S1205.

[0055] Lastly, when the delay quantity does not exceeds the thresholds "Th", the network condition determination device 303 determines that successful communication is established, and does not change a parameter at the step S1209.

[0056] By causing the network condition determination device 303 to carry out such processing, the network condition determination device 105 can have a function capable of determining whether a packet loss occurs on a transmission channel due to network congestion or whether a noise is generated, and the loss occurs due to the impairment of a channel state such as radio equipment.

[0057] This function is provided as a mere example. For example, when the past packet loss does not occur at the step S1202, a control for lowering a bit rate may be made by determining network congestion instead of determination of noise. If the current packet loss rate is improved more remarkably than the past packet loss rate at the step S1203, similar parameter change may be made after checking how the past change has been made. In this way, in this system, a variety of determination systems can be set according to a network.

[0058] In the foregoing, although there has been shown an example in which the video encoder 102 carries out encode processing based on encoded parameters determined and assigned at the network condition determination device 105, a configuration for the video encoder 102 to determine a parameter can be provided without being limited to the network condition determination device 105. An example is shown in FIG. 4.

[0059] FIG. 4 is a block diagram depicting an inclusive configuration when the video encoder 102 determines an encoded parameter. In the case of this example, the video encoder 102 comprises an encoded parameter determination device 401 and a signal processor 402.

[0060] In the case of this configuration, the network condition determination device 105 is configured to output the network condition information 136 from the network condition determination device 303, and then, provide the information to the video encoder 102.

[0061] In such a configuration, the network condition determination device 105 provides the network condition information 136 delivered from the network condition determination device 303 to the video encoder 102. Then, the video encoder 102 first inputs the network condition information 136 to an encoded parameter determination device 401. Then, the encoded parameter determination device 401 recognizes a network condi-

fuboTV Media Inc.
Exhibit 1002
Page 359

tion from this network condition information 136, and generates an encoded parameter 431 in the form suitable to the network condition.

[0062] The thus generated encoded parameter 431 is inputted to a signal processor 402. Then, this signal processor 402 encodes the video signal 131 inputted from the video input device 101 by using this encoded parameter 431.

[0063] Encoding information 432 such as the number of encoded bits after encoding is inputted to the encoded parameter determination device 401, and is used for the next encoded parameter determination. Encoded data 132 encoded at the signal processor 402 is output to the encoded data transmitter 103.

[0064] FIG. 5 is a block diagram depicting a configuration including a intra-frame encoding determination device 501 that forcibly sets intra-frame encoding at the encoded parameter determination device 401 shown in FIG. 4. As shown in the figure, the encoded parameter determination device 401 comprises an intra-frame encoding/determination device 501 and an encoded parameter determination device 502.

[0065] In the case of this configuration, the network condition information 136 outputted from the network condition determination device 105 is input to the encoded parameter determination device 502 possessed by the encoded parameter determination device 401. Then, encoded parameter information 531 corresponding to the network condition information 136 is output from the encoded parameter determination device 502.

[0066] The encoded parameter information 531 is input to the intra-frame encoding determination device 501, and it is determined whether inter-frame encoding is carried out as is or intra-frame encoding is forcibly carried out. When it is determined as intra-frame encoding, the encoding parameter information 531 is updated so as to carry out intra-frame encoding, and is output as the encoded,parameter information 431 to the signal processor 402 of the video encoder 102.

[0067] By doing this, even if the past data is not transmitted correctly to a receiver, when the data can be received correctly after parameter change, a correct image can be reproduced.

[0068] It is possible to configure a video encoder capable of determining a network condition, which has functions of the video encoder 102 and network condition determination device 105, thereby making it possible to provide the previously described function in one block.

[0069] The encoded parameter determination device 401 or network condition determination device 105 makes determination concerning the next encoded parameter setting. However, if a picture quality is changed suddenly due to temporary instability of network, an unclear picture may be produced. In order to suppress such a phenomenon, it is possible to incorporate a mechanism in which a change in encoded parameter is suppressed within its predetermined range. In the above described circumstance, this mechanism functions efficiently.

[0070] Although a bit rate or an error resilience parameter may be set as its calculated value, it is possible to employ a method of predetermining some patterns, and then, selecting the closest pattern from among them. Further, instead of the closest pattern, it is possible to make a selection of the closest bit rate which is lower than the calculated bit rate. This makes it possible to prevent the generation of a video image with its unexpected image quality due to a combination of a variety of parameters, and makes it possible to prepare a combination of encoded parameters that have been tested to some extent.

[0071] In the foregoing example, when the packet loss rate is not "0" or when a delay exceeds the thresholds, there has been made a control such that parameters such as bit rate or error resilience are changed, and the changed is reset to that suitable to the current state.

[0072] However, after recovery of a network condition, when no packet loss occurs or when a small delay occurs, it is possible to make a control for increasing a bit rate or decreasing an error resilience parameter.

[0073] The foregoing example will be described below according to a second embodiment.

(Second Embodiment)

[0074] FIG. 6 is a view showing a basic configuration of an image transmission apparatus according to the second embodiment of the present invention. The image transmission apparatus shown in FIG. 6 comprises a plurality of media 601, an encoded data switch device 602, an encoded data selector 603, an encoded data transmitter 103, a network information receiver 104, and a network condition determination device 105.

[0075] The storage media 601 are used to store encoded data 631 on contents that have been encoded in advance. The storage media 601 each store encoded data on the same contents encoded by different encoded parameters, respectively.

[0076] The encoded data switch device 602 selects one of a plurality of storage media 601, and provides encoding data 631 stored in the thus selected storage medium 601 to the encoded data transmitter 103. A selection of the storage media is carried out in accordance with switch information 632 from the encoded data selector 603.

[0077] The network information receiver 104 has a function that receives network information 134 such as packet loss rate or Jitter information delivered from a receiver or a network, and provides the received information as network information 135 to the network condition determination device 105.

[0078] The network condition determination device 105 determines a network condition from the input network information 135, and outputs the result as network condition information 136 to the encoded data selector

fuboTV Media Inc.
Exhibit 1002
Page 360

603. The encoded data selector 603 has a function for estimating, from this network condition information 136, which of the encoded data on encoding parameters is optimal to the current network, and then, outputting to the encoded data switch device 602 the switch information 632 used for switching in order to select encoded data caused by the estimated optimal encoding parameter as an output.

[0079] The encoded data switch device 602 selects one item of encoded data from among the input encoded data 631, and outputs encoded data 132.

[0080] With such a configuration, the encoded data 631 encoded in advance and store in the storage medium 601 or the like is input to an encoding data switch device 602 when the data is reproduced from the storage medium 601. That is, respective encoded data encoded by the differential encoding parameters are maintained in a plurality of storage media 601. As a result of reproducing these items of data, the encoded data encoded by the different encoding parameters, respectively, is inputted. The encoded data switch device 602 selects one from among these items of data, and outputs it to the encoded data transmitter 103.

[0081] This selection is carried out in accordance with switch information 632 from the encoded data selector 603.

[0082] On the other hand, network information 134 such as packet loss rate or Jitter information from a receiver is received at the network information receiver 104, and then, the received information is delivered to the network condition determination device 105. The network condition determination device 105 determines a network condition from the inputted network information 135, and outputs the result as network condition information 136 to an encoded data selector 603.

[0083] The encoded data selector 603 estimates, from the network condition information 136, which of the encoded data of the encoding parameters is optimal to the current network, and outputs the switch information 632 to the encoded data switch device 602.

[0084] The encoded data switch device 602 selects one item of the encoded data from among the input encoded data 631, and outputs encoded data 132.

[0085] The present embodiment will be described by way of a specific example.

[0086] Now, assume that there exist five storage media 601a to 601e each having stored therein five items of encoded data encoded at 384 kbps, 128 kbps, 64 kbps, 32 kbps, and 16 kbps.

[0087] That is, for example, a storage medium having stored therein encoded data on contents encoded at 384 kbps is defined as a storage medium 601a; a storage medium having stored therein encoded data on contents encoded at 128 kbps is defined as a storage medium 601b; a storage medium having stored therein encoded data on contents encoded at 64 kbps is defined as a storage medium 601c; a storage medium having stored therein encoded data on contents encoded at 32 kbps is defined as a storage medium 601d; and a storage medium having stored therein encoded data on contents encoded at 16 kbps is defined as a storage medium 601e.

[0088] In the case of using a transmission channel having its quality capable of being transmitted at 384 kbps, first, reproduced encoded data is selected by the storage medium 601a that is a storage medium having stored therein encoded data on contents encoded at 384 kbps; transmission is started at 384 kbps, and reception of network information from the receiver starts at the same time.

[0089] Although network information is received at a predetermined interval, if a problem occurs with a network based on the received information, a valid bit rate is calculated at a stage at which a packet loss or delay occurs. The encoded data selector 603 selects encoded data on a bit rate close to the calculated bit rate, and the encoded data switch device 602 switches and transmits the selected data.

[0090] For example, the calculated bit rate is 140 kbps, a value close to this bit rate is 128 kbps. In this case, the above switch device selects and switches encoded data reproduced from the storage medium 601b having stored therein encoded data on contents encoded at 128 kbps. In addition, if the calculated bit rate is 100 kbps, a value close to a bit rate lower than the calculated bit rate is 64 kbps. In this case, the switch device selects and switches encoded data reproduced from the storage medium 601c having stored therein encoded data on contents encoded at 64 kbps.

[0091] In this way, in the present embodiment, contents are prepared after encoded in advance for type of transmission bit rate, and a current optimal transmission rate is obtained based on network information such as packet loss rate or Jitter information from a receiver that is a function possessed by RTP. A storage medium having stored therein encoded data obtained at the bit rate corresponding to this transmission rate is selected by the switch 602, and reproduced encoded data output of the storage medium is transmitted. In this manner, there is no need for a server to carry out encode processing in real time, making it possible to reduce a load on the server. This is very effective in distributing a pre-produced video such as broadcast in particular.

[0092] Now, a modified example of the second embodiment will be described here. FIG. 7 is a block diagram depicting a method for carrying out control for a switch timing of encoded data to be inputted at the encoded data switch device 602.

[0093] Encoded data 631 is inputted to the switching position detecting device 701 and a switch device 702. Switch information 632 is inputted to the switching position detecting device 701. If there occurs a need to execute switching from the current encoded data to the other encoded data according to the switch information 632, the switching position detecting device 701 analyzes encoded data, and detects a switching enable posi-

fuboTV Media Inc.
Exhibit 1002
Page 361