US 20050262257A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2005/0262257 A1

(43) Pub. Date: Nov. 24, 2005

Major et al.

(54) APPARATUS, SYSTEM, AND METHOD FOR ADAPTIVE-RATE SHIFTING OF STREAMING CONTENT

(76) Inventors: **R. Drew Major**, Orem, UT (US); **Mark B. Hurst**, Cedar Hills, UT (US)

Correspondence Address:
KUNZLER & ASSOCIATES
8 EAST BROADWAY
SALT LAKE CITY, UT 84111 (US)

(21) Appl. No.: 11/116,783

(22) Filed: Apr. 28, 2005

Related U.S. Application Data

(60) Provisional application No. 60/566,831, filed on Apr. 30, 2004.

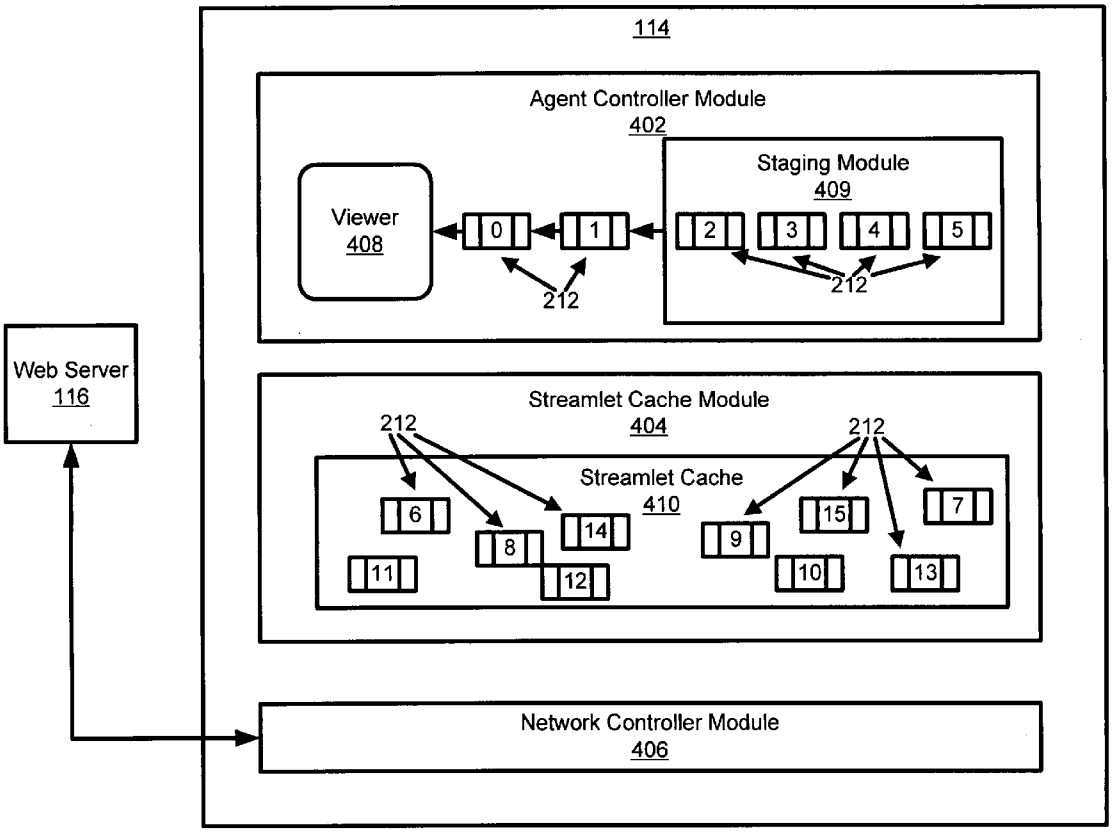
Publication Classification

(51) Int. Cl.[7] G06F 15/16

(52) U.S. Cl. 709/231

(57) **ABSTRACT**

An apparatus for adaptive-rate shifting of streaming content includes an agent controller module configured to simultaneously request at least portions of a plurality of streamlets. The agent controller module is further configured to continuously monitor streamlet requests and subsequent responses, and accordingly request higher or lower quality streamlets. A staging module is configured to stage the streamlets and arrange the streamlets for playback on a content player. A system includes a data communications network, a content server coupled to the data communications network and having a content module configured to process content and generate a plurality of high and low quality streams, and the apparatus. A method includes simultaneously requesting at least portions of a plurality of streamlets, continuously monitoring streamlet requests and subsequent responses, and accordingly requesting higher or lower quality streamlets, and staging the streamlets and arranging the streamlets for playback on a content player.



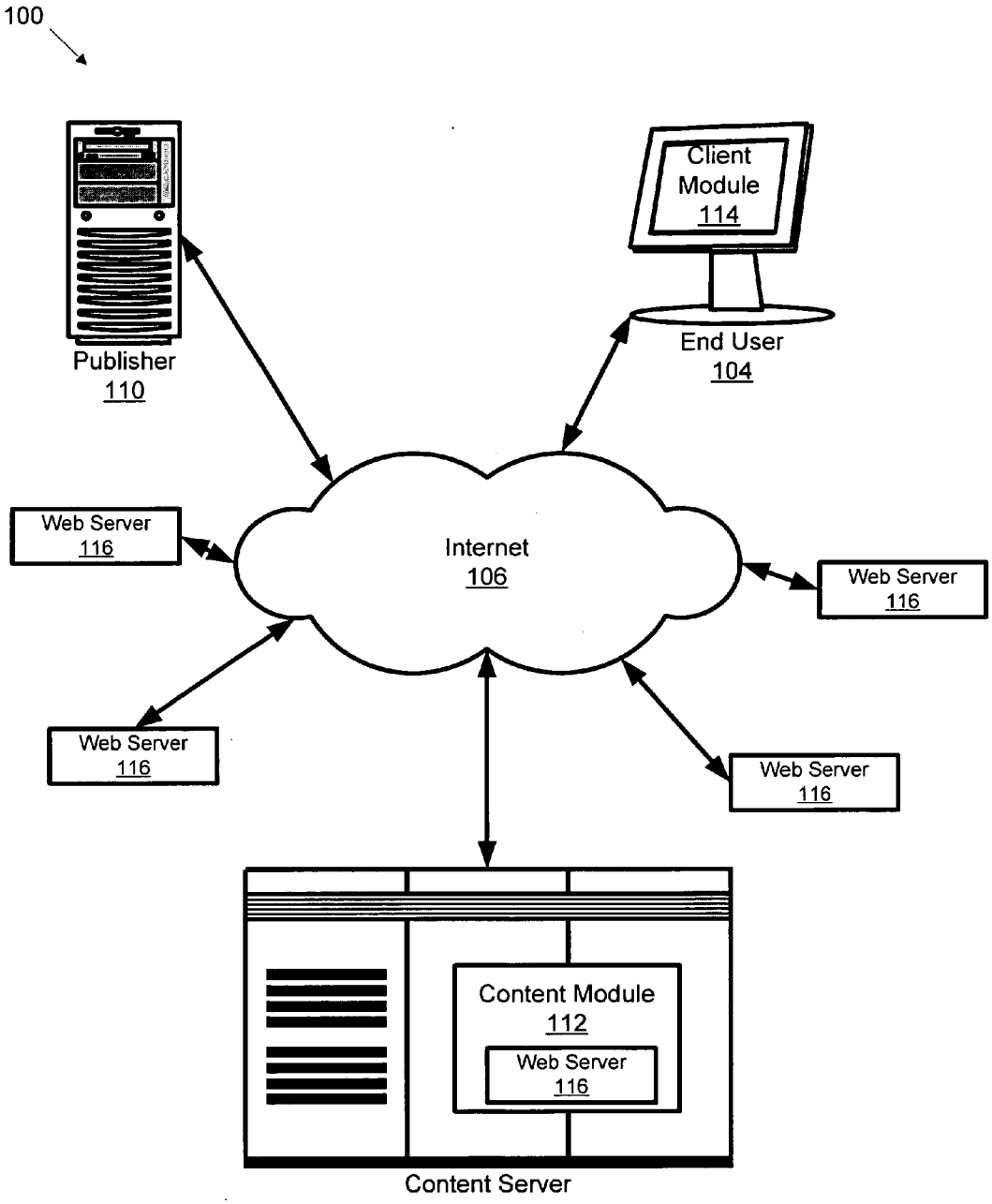

100
Publisher
110
Client
Module
114
End User
104
Web Server
116
Internet
106
Web Server
116
Web Server
116
Web Server
116
Content Module
112
Web Server
116
Content Server

FIG. 1

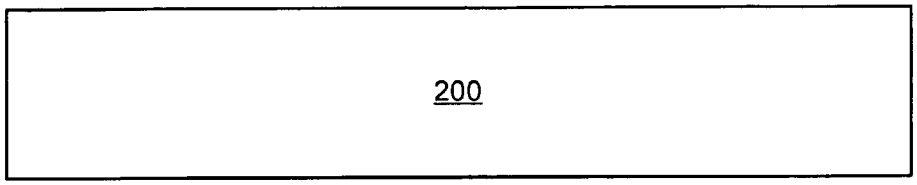


FIG. 2a

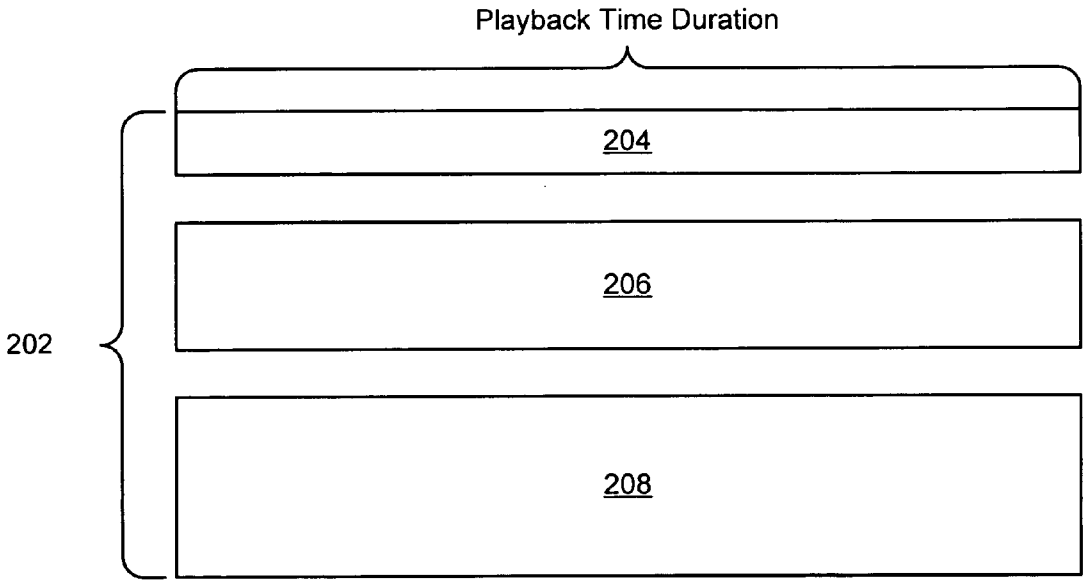


FIG. 2b

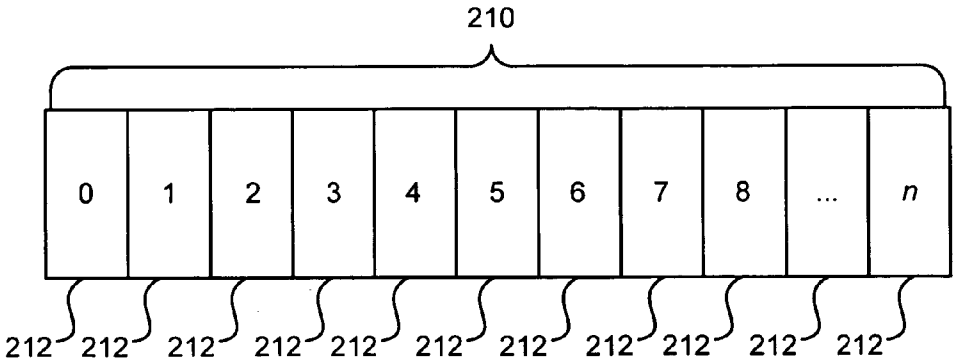


FIG. 2c

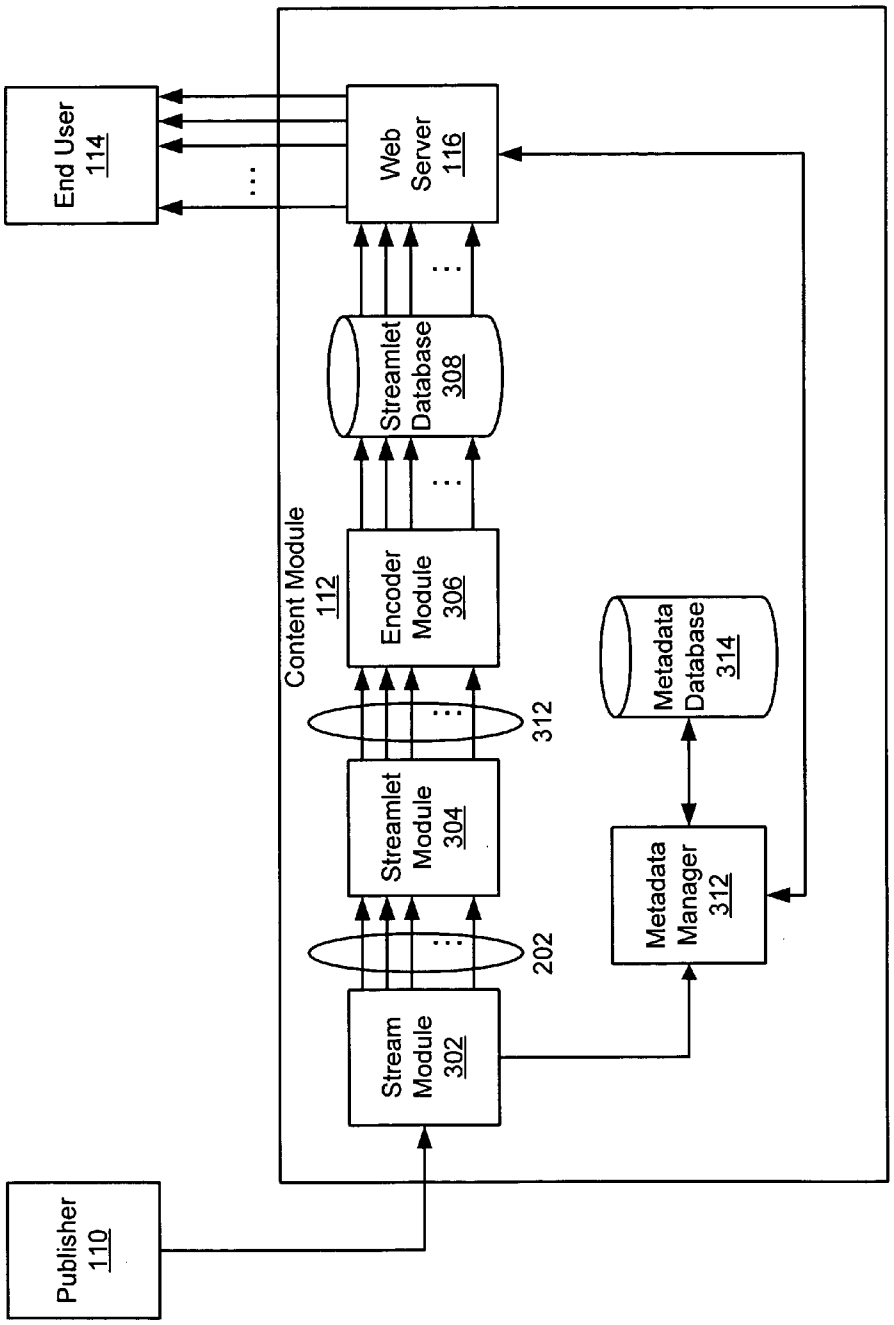

End User
114
Web
Server
116
Streamlet
Database
308
Content Module
112
Encoder
Module
306
312
Streamlet
Module
304
202
Stream
Module
302
Metadata
Database
314
Metadata
Manager
312
Publisher
110

FIG. 3

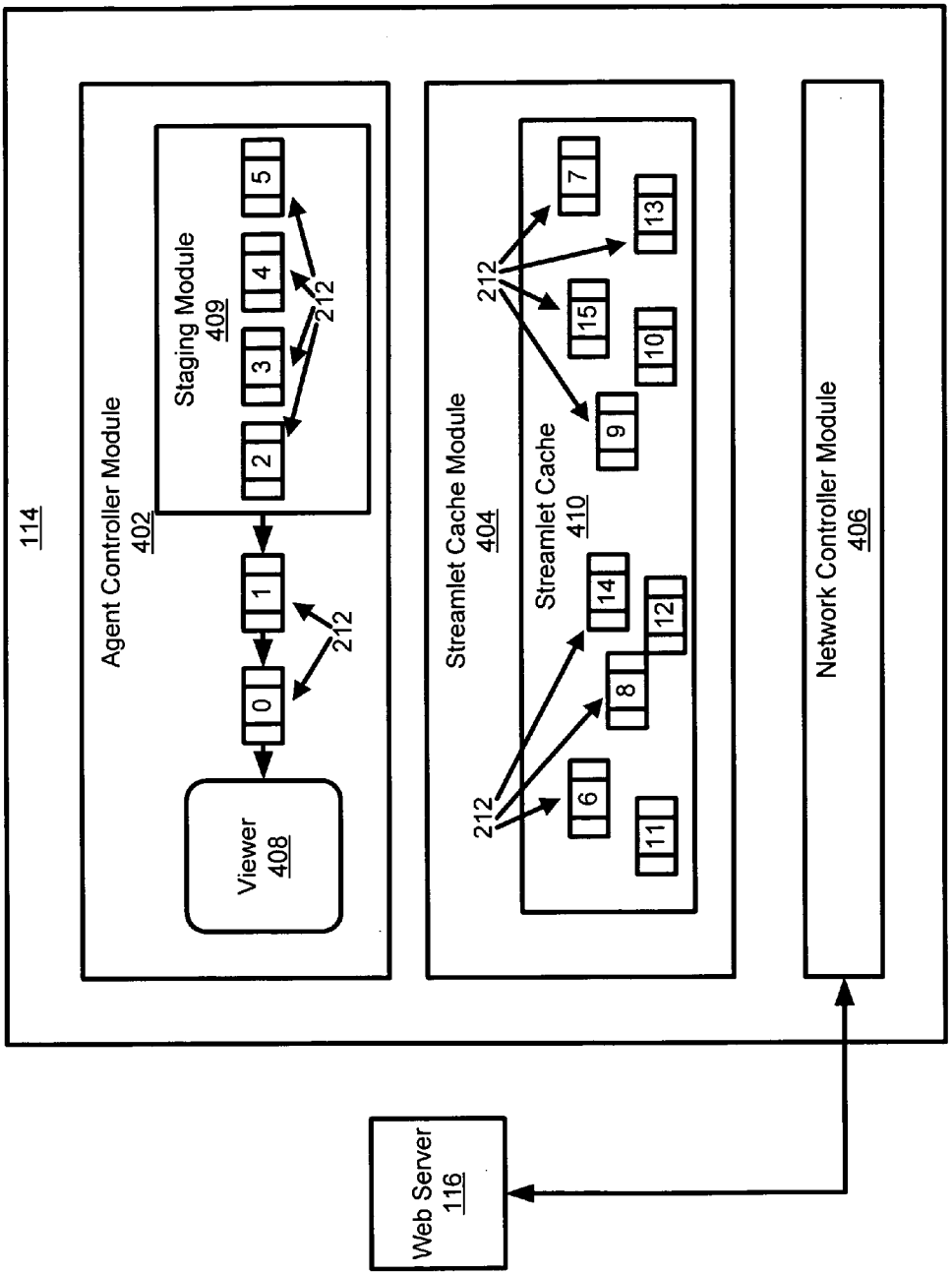

114
Agent Controller Module
402
Staging Module
409
0
1
2
3
4
5
212
Viewer
408
Streamlet Cache Module
404
Streamlet Cache
410
6
7
8
9
10
11
12
13
14
15
Network Controller Module
406
Web Server
116

FIG. 4

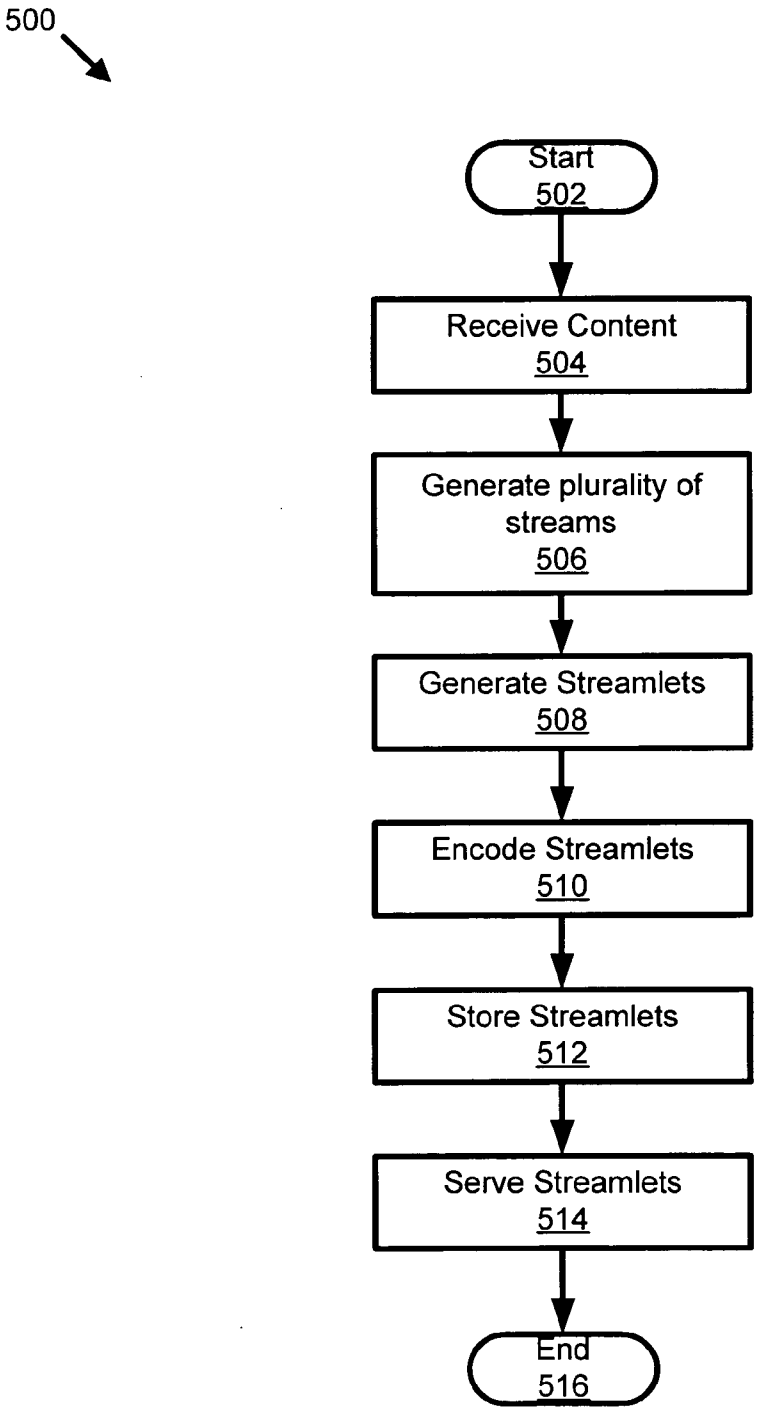

500
Start
502
Receive Content
504
Generate plurality of streams
506
Generate Streamlets
508
Encode Streamlets
510
Store Streamlets
512
Serve Streamlets
514
End
516

FIG. 5

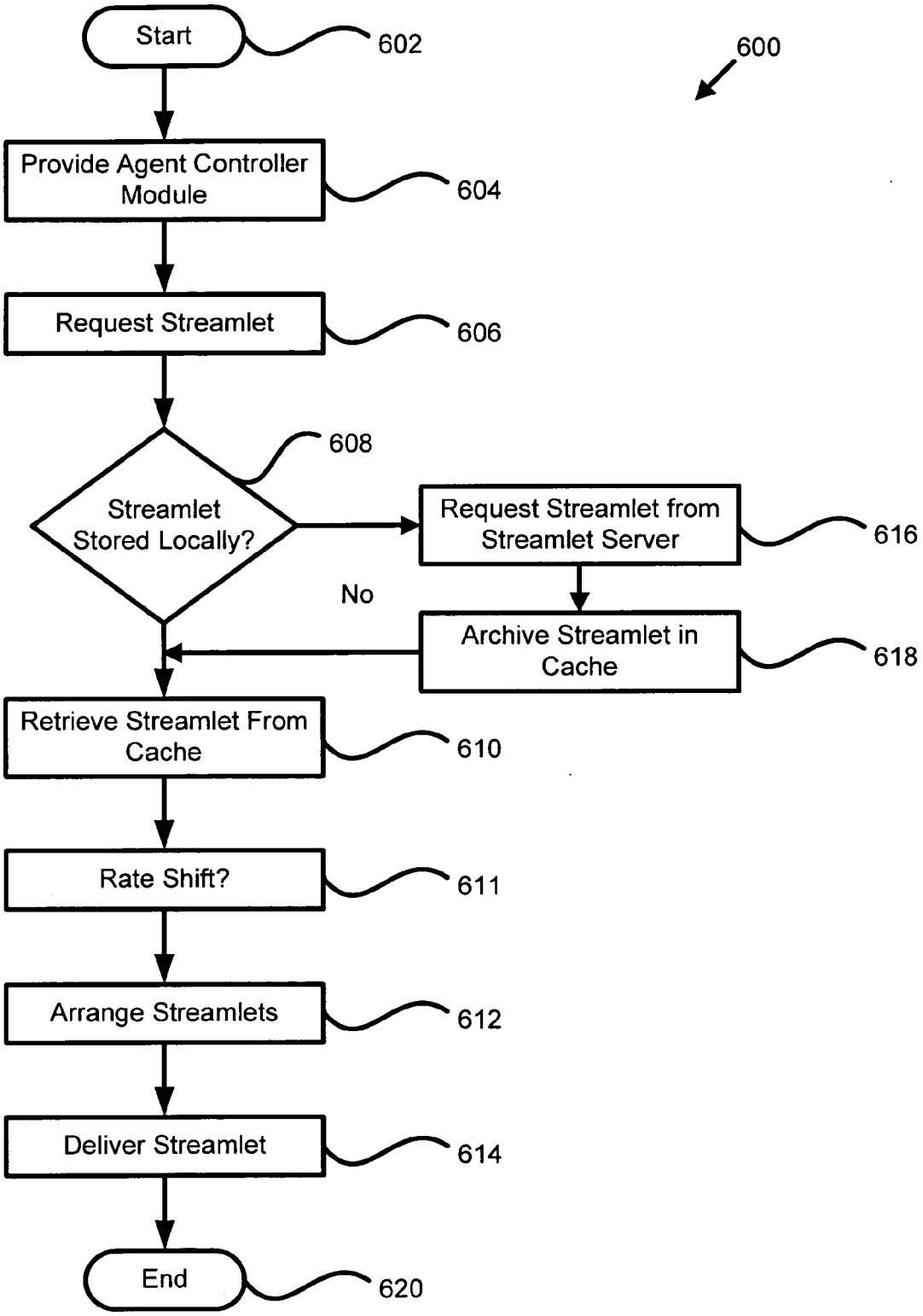

Start
602
600
Provide Agent Controller Module
604
Request Streamlet
606
608
Streamlet Stored Locally?
Request Streamlet from Streamlet Server
616
No
Archive Streamlet in Cache
618
Retrieve Streamlet From Cache
610
Rate Shift?
611
Arrange Streamlets
612
Deliver Streamlet
614
End
620

FIG. 6

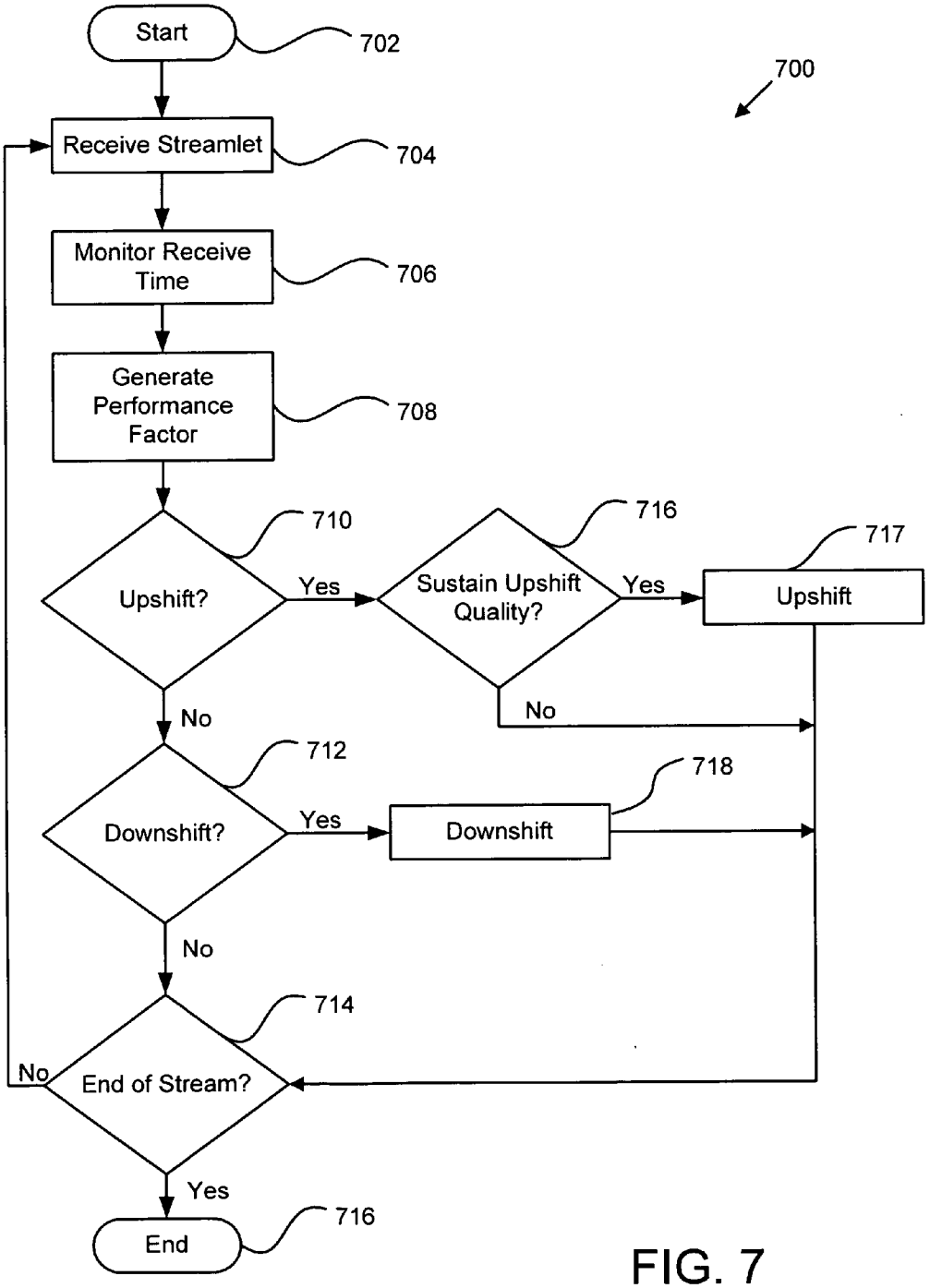

Start
702
700
Receive Streamlet
704
Monitor Receive Time
706
Generate Performance Factor
708
710
Upshift?
Yes
716
Sustain Upshift Quality?
Yes
717
Upshift
No
No
712
Downshift?
Yes
718
Downshift
No
714
No
End of Stream?
Yes
716
End

FIG. 7

APPARATUS, SYSTEM, AND METHOD FOR ADAPTIVE-RATE SHIFTING OF STREAMING CONTENT

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application claims benefit of U.S. Provisional Patent Application No. 60/566,831 entitled "APPARATUS, SYSTEM, AND METHOD FOR DYNAMIC RATE SHIFTING OF STREAMING CONTENT" and filed on Apr. 30, 2004 for R. Drew Major and Mark B. Hurst, which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

[0003] 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.

[0004] 2. Description of the Related Art

[0005] 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.

[0006] "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.

[0007] 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.

[0008] 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.

[0009] 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.

[0010] Generally, three basic challenges exist with regard to data transport streaming over a network such as the Internet that has a varying amount of data loss. The first challenge is reliability. Most streaming solutions use a TCP connection, or "virtual circuit," for transmitting data. A TCP connection provides a guaranteed delivery mechanism so that data sent from one endpoint will be delivered to the destination, even if portions are lost and retransmitted. A break in the continuity of a TCP connection can have serious consequences when the data must be delivered in real-time. When a network adapter detects delays or losses in a TCP connection, the adapter "backs off" from transmission attempts for a moment and then slowly resumes the original transmission pace. This behavior is an attempt to alleviate the perceived congestion. Such a slowdown is detrimental to the viewing or listening experience of the user and therefore is not acceptable.

[0011] 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.

[0012] 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.

[0013] From the foregoing discussion, it should be apparent that a need exists for an apparatus, system, and method that alleviate the problems of reliability, efficiency, and latency. Additionally, such an apparatus, system, and method would offer instantaneous viewing along with the ability to fast forward, rewind, direct seek, and browse multiple streams. Beneficially, such an apparatus, system, and method would utilize multiple connections between a source and destination, requesting varying bitrate streams depending upon network conditions.

SUMMARY OF THE INVENTION

[0014] 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 stream-

ing 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.

[0015] 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 an agent controller module configured to simultaneously request a plurality of streamlets, the agent controller module further configured to continuously monitor streamlet requests and subsequent responses, and accordingly request higher or lower quality streamlets, and a staging module configured to stage the streamlets and arrange the streamlets for playback on a content player.

[0016] The apparatus is further configured, in one embodiment, to establish multiple Transmission Control Protocol (TCP) connections with a content server, and request streamlets of varying bitrates. Each streamlet may further comprise a portion of a content file. Additionally, the agent controller module may be configured to generate a performance factor according to responses from streamlet requests.

[0017] In a further embodiment, the agent controller module is configured to upshift to a higher quality streamlet when the performance factor is greater than a threshold, and the agent controller module determines the higher quality playback can be sustained according to combination of factors. The factors may include an amount of contiguously available streamlets stored in the staging module, a minimum safety margin, and a current read ahead margin.

[0018] The agent controller module may be configured to downshift to a lower quality streamlet when the performance factor is less than a second threshold. Also, the agent controller module is further configured to anticipate streamlet requests and pre-request streamlets to enable fast-forward, skip randomly, and rewind functionality. In one embodiment, the agent controller module is configured to initially request low quality streamlets to enable instant playback of the content file, and subsequent upshifting according to the performance factor.

[0019] A system of the present invention is also presented to adaptive-rate content streaming. In particular, the system, in one embodiment, includes a data communications network, and a content server coupled to the data communications network and having a content module configured to process content and generate a plurality of high and low quality streams. In one embodiment, each of the high and low quality streams may include a plurality of streamlets.

[0020] In a further embodiment, the system also includes an agent controller module configured to simultaneously request a plurality of streamlets, the agent controller module further configured to continuously monitor streamlet requests and subsequent responses, and accordingly request higher or lower quality streamlets, and a staging module configured to stage the streamlets and arrange the streamlets for playback on a content player.

[0021] A method of the present invention is also presented for adaptive-rate content streaming. The method in the disclosed embodiments substantially includes the steps necessary to carry out the functions presented above with respect to the operation of the described apparatus and system. In one embodiment, the method includes simultaneously requesting a plurality of streamlets, continuously monitoring streamlet requests and subsequent responses, and accordingly requesting higher or lower quality streamlets, and staging the streamlets and arranging the streamlets for playback on a content player.

[0022] In a further embodiment, the method may include establishing multiple Transmission Control Protocol (TCP) connections with a content server, and requesting streamlets of varying bitrates. Also, the method may include generating a performance factor according to responses from streamlet requests, upshifting to a higher quality streamlet when the performance factor is greater than a threshold, and determining if the higher quality playback can be sustained. Furthermore, the method may include downshifting to a lower quality streamlet when the performance factor is less than a second threshold.

[0023] In one embodiment, the method includes anticipating streamlet requests and pre-requesting streamlets to enable fast-forward, skip randomly, and rewind functionality. The method may also comprise initially requesting low quality streamlets to enable instant playback of a content file, and subsequent upshifting according to the performance factor.

[0024] 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.

[0025] Furthermore, the described features, advantages, and characteristics of the invention may be combined in any suitable manner in one or more embodiments. One skilled in the relevant art will recognize that the invention may be practiced without one or more of the specific features or advantages of a particular embodiment. In other instances, additional features and advantages may be recognized in certain embodiments that may not be present in all embodiments of the invention.

[0026] These features and advantages of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] 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:

[0028] **FIG. 1** is a schematic block diagram illustrating one embodiment of a system for adaptive rate shifting of streaming content in accordance with the present invention;

[0029] **FIG. 2***a* is a schematic block diagram graphically illustrating one embodiment of a content file in accordance with the present invention;

[0030] **FIG. 2***b* is a schematic block diagram illustrating one embodiment of a plurality of streams having varying degrees of quality and bandwidth in accordance with the present invention;

[0031] **FIG. 2***c* is a schematic block diagram illustrating one embodiment of a stream divided into a plurality of streamlets in accordance with the present invention;

[0032] **FIG. 3** is a schematic block diagram illustrating one embodiment of a content module in accordance with the present invention;

[0033] **FIG. 4** is a schematic block diagram graphically illustrating one embodiment of a client module in accordance with the present invention;

[0034] **FIG. 5** is a schematic flow chart diagram illustrating one embodiment of a method for processing content in accordance with the present invention;

[0035] **FIG. 6** is a schematic flow chart diagram illustrating one embodiment of a method for playback of a plurality of streamlets in accordance with the present invention; and

[0036] **FIG. 7** is a schematic flow chart diagram illustrating one embodiment of a method for requesting streamlets within an adaptive-rate content streaming environment in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0037] 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.

[0038] 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.

[0039] 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.

[0040] 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.

[0041] Reference to a signal bearing medium may take any form capable of generating a signal, causing a signal to be generated, or causing execution of a program of machine-readable instructions on a digital processing apparatus. A signal bearing medium may be embodied by a transmission line, a compact disk, digital-video disk, a magnetic tape, a Bernoulli drive, a magnetic disk, a punch card, flash memory, integrated circuits, or other digital processing apparatus memory device.

[0042] 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.

[0043] **FIG. 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 **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.

[0044] In the depicted embodiment, the system **100** also includes a publisher **110**, and a web server **116**. The publisher **110** may be a creator or distributor of content. For example, if the content to be streamed were a broadcast of a television program, the publisher **110** may be a television or cable network channel such as NBC®, or MTV®. Content may be transferred over the Internet **106** to the content server **102**, where the content is received by a content module **112**. The content module **112** may be configured to receive, process, and store content. In one embodiment, processed content is accessed by a client module **114** con-

figured 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**.

[**0045**] **FIG. 2***a* is a schematic block diagram graphically illustrating one embodiment of a 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. Additionally, the content file **200** may be compressed. Examples of a compressed content file **200** include, but are not limited to, DivX®, Windows Media Video 9®, Quicktime 6.5 Sorenson 3®, or Quicktime 6.5/MPEG-4® encoded content.

[**0046**] **FIG. 2***b* is a schematic block diagram illustrating one embodiment of a plurality of streams **202** having varying degrees of quality and bandwidth. In one embodiment, the plurality of streams **202** comprises a low quality stream **204**, a medium quality stream **206**, and a high quality stream **208**. Each of the streams **204**, **206**, **208** is a copy of the content file **200** encoded and compressed to varying bit rates. For example, the low quality stream **204** may be encoded and compressed to a bit rate of 100 kilobits per second (kbps), the medium quality stream **206** may be encoded and compressed to a bit rate of 200 kbps, and the high quality stream **208** may be encoded and compressed to 600 kbps.

[**0047**] **FIG. 2***c* is a schematic block diagram illustrating one embodiment of a stream **210** divided into a plurality of streamlets **212**. As used herein, streamlet refers to any sized portion of the content file **200**. Each streamlet **212** may comprise a portion of the content contained in stream **210**, encapsulated as an independent media object. The content in a streamlet **212** may have a unique time index in relation to the beginning of the content contained in stream **210**. In one embodiment, the content contained in each streamlet **212** has 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 **212** may be any duration smaller than the entire playback duration of the content in stream **210**. In a further embodiment, the streamlets **212** may be divided according to file size instead of a time index.

[**0048**] **FIG. 3** 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 stream module **302**, a streamlet module **304**, an encoder module **306**, a streamlet database **308**, and the web server **116**. In one embodiment, the stream module **302** is configured to receive the content file **200** from the publisher **110** 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, 2 mbps. The content may be transferred from the publisher **110** to the content module **112** over the Internet **106**. Such transfers of data are well known in the art and do not require further discussion herein. Alternatively, the content may comprise a captured broadcast.

[**0049**] In the depicted embodiment, 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. The streamlet module **304** may be configured to receive the plurality of streams **202** from the stream module and generate a plurality of streams **312**, each stream comprising a plurality of streamlets **212**. As described with reference to **FIG. 2***c*, each streamlet **212** may comprise a pre-defined portion of the stream. The encoder module **306** is configured to encode each streamlet from the plurality of streams **312** and store the streamlets in the streamlet database **308**. The encoding module **306** may utilize encoding schemes such as DivX®, Windows Media Video 9®, Quicktime 6.5 Sorenson 3®, or Quicktime 6.5/MPEG-40®. Alternatively, a custom encoding scheme may be employed.

[**0050**] The content module **112** may also include a metadata module **312** and a metadata database **314**. 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 **312** is configured to receive the metadata from the publisher **110** and store the metadata in the metadata database **314**. In a further embodiment, the metadata module **312** 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.

[**0051**] Once the streamlets **212** have been received and processed, the client module **114** may request streamlets **212** using HTTP from the web server **116**. Such use of client side initiated requests requires no additional configuration of firewalls. Additionally, since the client module **114** initiates the request, the web server **116** is only required to retrieve and serve the requested streamlet. In a further embodiment, the client module **114** may be configured to retrieve streamlets **212** from a plurality of web servers **310**. Each web server **116** may be located in various locations across the Internet **106**. The streamlets **212** are essentially static files. As such, no specialized media server or server-side intelligence is required for a client module **114** to retrieve streamlets **212**. Streamlets **212** 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.

[0052] **FIG. 4** 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 **402**, a streamlet cache module **404**, and a network controller module **406**. In one embodiment, the agent controller module **402** is configured to interface with a viewer **408**, and transmit streamlets **212** to the viewer **408**. In a further embodiment, the client module **114** may comprise a plurality of agent controller modules **402**. Each agent controller module **402** may be configured to interface with one viewer **408**. Alternatively, the agent controller module **402** may be configured to interface with a plurality of viewers **408**. The viewer **408** may be a media player (not shown) operating on a PC or handheld electronic device.

[0053] The agent controller module **402** is configured to select a quality level of streamlets to transmit to the viewer **408**. The agent controller module **402** 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 **FIG. 7**.

[0054] The agent controller module **402** may be configured to receive user commands from the viewer **408**. Such commands may include play, fast forward, rewind, pause, and stop. In one embodiment, the agent controller module **402** requests streamlets **212** from the streamlet cache module **404** and arranges the received streamlets **212** in a staging module **409**. The staging module **409** may be configured to arrange the streamlets **212** in order of ascending playback time. In the depicted embodiment, the streamlets **212** are numbered 0, 1, 2, 3, 4, etc. However, each streamlet **212** may be identified with a unique filename.

[0055] Additionally, the agent controller module **402** may be configured to anticipate streamlet **212** requests and pre-request streamlets **212**. By pre-requesting streamlets **212**, 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 **402** may request the streamlets **212** that correspond to time index intervals of 30 seconds within the total play time of the content. Alternatively, the agent controller module **402** 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 **402** may be configured to pre-request streamlets **212** corresponding to specified index points within the content or within other content in anticipation of the end user **104** selecting new content to view.

[0056] In one embodiment, the streamlet cache module **404** is configured to receive streamlet **212** requests from the agent controller module **402**. Upon receiving a request, the streamlet cache module **404** first checks a streamlet cache **410** to verify if the streamlet **212** is present. In a further embodiment, the streamlet cache module **404** handles streamlet **212** requests from a plurality of agent controller modules **402**. Alternatively, a streamlet cache module **404** may be provided for each agent controller module **402**. If the requested streamlet **212** is not present in the streamlet cache **410**, the request is passed to the network controller module **406**. In order to enable fast forward and rewind capabilities, the streamlet cache module **404** is configured to store the plurality of streamlets **212** in the streamlet cache **410** for a specified time period after the streamlet **212** has been viewed. However, once the streamlets **212** have been deleted, they may be requested again from the web server **116**.

[0057] The network controller module **406** may be configured to receive streamlet requests from the streamlet cache module **404** and open a connection to the web server **116** or other remote streamlet **212** database (not shown). In one embodiment, the network controller module **406** opens a TCP/IP connection to the web server **116** and generates a standard HTTP GET request for the requested streamlet **212**. Upon receiving the requested streamlet **212**, the network controller module **406** passes the streamlet **212** to the streamlet cache module **404** where it is stored in the streamlet cache **410**. In a further embodiment, the network controller module **406** is configured to process and request a plurality of streamlets **212** simultaneously. The network controller module **406** may also be configured to request a plurality of streamlets, where each streamlet **212** is subsequently requested in multiple parts.

[0058] In a further embodiment, streamlet requests may comprise requesting pieces of any streamlet file. Splitting the streamlet **212** 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 **212**. Consequently, efficiency and network loss problems are overcome, and the streamlets arrive with more useful and predictable timing.

[0059] 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 **212** at a time in a manner referred to as "parallel retrieval," or more than one part of a streamlet **212** 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 **212** requests. The client module **114** may maintain additional open TCP connections as spares to be available should another connection fail. Streamlet **212** 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 **406** has requested a streamlet **212** in multiple parts, with each part requested on mutually independent TCP/IP connections, the network controller module **406** reassembles the parts to present a complete streamlet **212** for use by all other components of the client module **114**.

[0060] When a TCP connection fails completely, a new request may be sent on a different connection for the same streamlet **212**. 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 **212**. 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.

[0061] Several streamlet **212** 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 **212** 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.

[0062] 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 **212** request. Several streamlet **212** 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.

[0063] A single streamlet **212** request may be issued for an entire streamlet **212**, 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.

[0064] 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 **212** 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.

[0065] 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.

[0066] **FIG. 5** is a schematic flow chart diagram illustrating one embodiment of a method **500** for processing content in accordance with the present invention. In one embodiment the method **500** starts **502**, and the content module **112** receives **504** content from the publisher **110**. Receiving content **504** may comprise receiving **504** a digital copy of the content file **200**, or digitizing a physical copy of the content file **200**. Alternatively, receiving **504** content may comprise capturing a radio or television broadcast. Once received **504**, the stream module **302** generates **506** a plurality of streams **202**, each stream **202** having a different quality. The quality may be predefined, or automatically set according to end user bandwidth, or in response to predesignated publisher guidelines

[0067] The streamlet module **304** receives the streams **202** and generates **508** a plurality of streamlets **212**. In one embodiment, generating **508** streamlets comprises dividing the stream **202** into a plurality of two second streamlets **212**. Alternatively, the streamlets may have any length less than or equal to the length of the stream **202**. The encoder module **306** then encodes **510** the streamlets according to a compression algorithm. In a further embodiment, the algorithm comprises a proprietary codec such as WMV9®. The encoder module **306** then stores **512** the encoded streamlets in the streamlet database **308**. Once stored **512**, the web server **116** may then serve **514** the streamlets. In one embodiment, serving **514** the streamlets comprises receiving streamlet requests from the client module **114**, retrieving the requested streamlet from the streamlet database **308**, and subsequently transmitting the streamlet to the client module **114**. The method **500** then ends **516**.

[0068] **FIG. 6** is a schematic flow chart diagram illustrating one embodiment of a method **600** for viewing a plurality of streamlets in accordance with the present invention. The method **600** starts and an agent control module **402** is provided **604** and associated with a viewer **408** and provided with a staging module **409**. The agent controller module **402** then requests **606** a streamlet from the streamlet cache module **404**. Alternatively, the agent controller module **402** may simultaneously request **606** a plurality of streamlets from the streamlet cache module **404**. If the streamlet is stored **608** locally in the streamlet cache **410**, the streamlet cache module **404** retrieves **610** the streamlet and sends the streamlet to the agent controller module **402**. Upon retrieving **610** or receiving a streamlet, the agent controller module **402** makes **611** a determination of whether or not to shift to a higher or lower quality stream **202**. This determination will be described below in greater detail with reference to **FIG. 7**.

[0069] In one embodiment, the staging module **409** then arranges **612** the streamlets into the proper order, and the agent controller module **402** delivers **614** the streamlets to

the viewer **408**. In a further embodiment, delivering **614** streamlets to the end user comprises playing video and or audio streamlets on the viewer **408**. If the streamlets are not stored **608** locally, the streamlet request is passed to the network controller module **406**. The network controller module **406** then requests **616** the streamlet from the web server **116**. Once the streamlet is received, the network controller module **406** passes the streamlet to the streamlet cache module **404**. The streamlet cache module **404** archives **618** the streamlet. Alternatively, the streamlet cache module **404** then archives **618** the streamlet and passes the streamlet to the agent controller module **402**, and the method **600** then continues from operation **610** as described above.

[**0070**] Referring now to **FIG. 7**, shown therein is a schematic flow chart diagram illustrating one embodiment of a method **700** for requesting streamlets within a adaptive-rate shifting content streaming environment in accordance with the present invention. The method **700** may be used in one embodiment as the operation **611** of **FIG. 6**. The method **700** starts and the agent controller module **402** receives **704** a streamlet as described above with reference to **FIG. 6**. The agent controller module **402** then monitors **706** the receive time of the requested streamlet. In one embodiment, the agent controller module **402** monitors the time intervals A between successive receive times for each streamlet response. Ordering of the responses in relation to the order of their corresponding requests is not relevant.

[**0071**] 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 **402** calculates **708** 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

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

[**0072**] Due to multiple simultaneous streamlet processing, and in order to better judge the central tendency of the performance ratio r, the agent control module **402** may calculate a geometric mean, or alternatively an equivalent averaging algorithm, across a window of size m, and obtain a performance factor ϕ:

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

[**0073**] The policy determination about whether or not to upshift **710** playback quality begins by comparing $\phi_{current}$ with a trigger threshold Θ_{up}. If $\phi_{current} \geqq \Theta_{up}$, then an up shift to the next higher quality stream may be considered **716**. 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 **409** 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 **402** is able to sustain **716** upshift quality, then the agent controller module **402** will upshift **717** the quality and subsequently request higher quality streams. The determination of whether use of the higher quality stream is sustainable **716** is made by comparing an estimate of the higher quality stream's performance factor, ϕ_{higher}, with Θ_{up}. If $\phi_{higher} \geqq \Theta_{up}$ then use of the higher quality stream is considered sustainable. If the decision of whether or not the higher stream rate is sustainable **716** is "no," the agent control module **402** will not attempt to upshift **717** stream quality. If the end of the stream has been reached **714**, the method **618** ends **716**.

[**0074**] If the decision on whether or not to attempt upshift **710** is "no", a decision about whether or not to downshift **712** is made. In one embodiment, a trigger threshold Θ_{down} down is defined in a manner analogous to Θ_{up}. If $\phi_{current} > \Theta_{down}$ then the stream quality may be adequate, and the agent controller module **402** does not downshift **718** stream quality. However, if $\phi_{current} \leqq \Theta_{down}$, the agent controller module **402** does downshift **718** the stream quality. If the end of the stream has not been reached **714**, the agent controller module **402** begins to request and receive **704** lower quality streamlets and the method **618** starts again. Of course, the above described equations and algorithms are illustrative only, and may be replaced by alternative streamlet monitoring solutions.

[**0075**] 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.

What is claimed is:

1. An apparatus for adaptive-rate content streaming, the apparatus comprising:

an agent controller module configured to simultaneously request at least portions of a plurality of streamlets;

the agent controller module further configured to continuously monitor the progress of subsequent responses to streamlet requests, and in accordance with the progress of the responses, request higher or lower quality streamlets; and

a staging module configured to stage the streamlets and arrange the streamlets for playback on a content player.

2. The apparatus of claim 1, wherein the agent controller module is configured to establish multiple Transmission Control Protocol (TCP) connections with a content server, and request streamlets of varying bitrates.

3. The apparatus of claim 1, wherein each streamlet further comprises a portion of a content file.

4. The apparatus of claim 1, wherein the streamlet comprises a Hyper Text Transfer Protocol file.

5. The apparatus of claim 3, wherein the agent controller module is configured to generate a performance factor according to the responses to streamlet requests.

6. The apparatus of claim 4, wherein the agent controller module is configured to upshift to a higher quality streamlet when the performance factor is greater than a selected threshold and the agent controller module determines the higher quality playback can be sustained according to a combination of factors.

7. The apparatus of claim 5, wherein the combination of factors is selected from the group consisting of an amount of contiguously available streamlets stored in the staging module, a minimum safety margin, and a current read ahead margin.

8. The apparatus of claim 4, wherein the agent controller module is configured to downshift to a lower quality streamlet when the performance factor is less than a second selected threshold.

9. The apparatus of claim 1, wherein the agent controller module is further configured to anticipate streamlet requests and pre-request streamlets to enable fast-forward, skip randomly, and rewind functionality.

10. The apparatus of claim 1, wherein the agent controller module is configured to initially request low quality streamlets to enable instant playback of the content file, and subsequent upshifting according to the performance factor.

11. A system for adaptive-rate content streaming, the system comprising:

a data communications network;

a content server coupled to the data communications network and having a content module configured to process content and generate a plurality of high and low quality streams;

each of the high and low quality streams comprising a plurality of streamlets;

an agent controller module configured to simultaneously request at least portions of a plurality of streamlets;

the agent controller module further configured to continuously monitor the progress of subsequent responses to streamlet requests, and in accordance with the progress of the responses, request higher or lower quality streamlets; and

a staging module configured to stage the streamlets and arrange the streamlets for playback on a content player.

12. The system of claim 11, wherein the agent controller module is configured to establish multiple Transmission Control Protocol (TCP) connections with the content server, and request streamlets of varying bitrates.

13. The system of claim 11, wherein each streamlet further comprises a portion of a content file.

14. The apparatus of claim 13, wherein the agent controller module is configured to generate a performance factor according to the responses to streamlet requests.

15. The system of claim 14, wherein the agent controller module is configured to upshift to a higher quality streamlet when the performance factor is greater than a selected threshold and the agent controller module determines the higher quality playback can be sustained according to a combination of factors.

16. The system of claim 15, wherein the combination of factors is selected from the group consisting of an amount of contiguously available streamlets stored in the staging module, a minimum safety margin, and a current read ahead margin.

17. The system of claim 14, wherein the agent controller module is configured to downshift to a lower quality streamlet when the performance factor is less than a second selected threshold.

18. The system of claim 11, wherein the agent controller module is further configured to anticipate streamlet requests and pre-request streamlets to enable fast-forward, skip randomly, and rewind functionality.

19. The system of claim 11, wherein the agent controller module is configured to initially request low quality streamlets to enable instant playback of the content file, and subsequent upshifting according to the performance factor.

20. A method for presenting rate-adaptive streams, the method comprising:

simultaneously requesting at least portions of a plurality of streamlets;

continuously monitoring the progress of subsequent responses to streamlet requests, and in accordance with the progress of the responses, requesting higher or lower quality streamlets; and

staging the streamlets and arranging the streamlets for playback on a content player.

21. The method of claim 20, further comprising establishing multiple Transmission Control Protocol (TCP) connections with a content server, and requesting streamlets of varying bitrates.

22. The method of claim 20, further comprising generating a performance factor according to responses from streamlet requests.

23. The method of claim 20, further comprising upshifting to a higher quality streamlet when the performance factor is greater than a selected threshold, and determining if the higher quality playback can be sustained.

24. The method of claim 20, further comprising downshifting to a lower quality streamlet when the performance factor is less than a second selected threshold.

25. The method of claim 20, further comprising anticipating streamlet requests and pre-requesting streamlets to enable fast-forward, skip randomly, and rewind functionality.

26. The method of claim 20, further comprising initially requesting low quality streamlets to enable instant playback of a content file, and subsequent upshifting according to the performance factor.

27. A computer readable storage medium comprising computer readable code configured to carry out operations for presenting rate-adaptive content streams, the operations comprising:

simultaneously requesting at least portions of a plurality of streamlets;

continuously monitoring the progress of subsequent responses to streamlet requests, and in accordance with the progress of the responses, requesting higher or lower quality streamlets; and

staging the streamlets and arranging the streamlets for playback on a content player.

28. The computer readable storage medium of claim 27, wherein the operations further comprise establishing mul-

tiple Transmission Control Protocol (TCP) connections with a content server, and requesting streamlets of varying bitrates.

29. The computer readable storage medium of claim 27, wherein the operations further comprise generating a performance factor according to responses from streamlet requests.

30. The computer readable storage medium of claim 27, wherein the operations further comprise upshifting to a higher quality streamlet when the performance factor is greater than a threshold, and determining if the higher quality playback can be sustained.

31. The computer readable storage medium of claim 27, wherein the operations further comprise downshifting to a lower quality streamlet when the performance factor is less than a second threshold.

32. The computer readable storage medium of claim 27, wherein the operations further comprise anticipating streamlet requests and pre-requesting streamlets to enable fast-forward, skip randomly, and rewind functionality.

33. The computer readable storage medium of claim 27, wherein the operations further comprise initially requesting low quality streamlets to enable instant playback of a content file, and subsequent upshifting according to the performance factor.

34. An apparatus to present rate-adaptive content streams, the apparatus comprising:

means for simultaneously requesting at least portions of a plurality of streamlets;

means for continuously monitoring the progress of subsequent responses to streamlet requests, and in accordance with the progress of the responses, requesting higher or lower quality streamlets; and

means for staging the streamlets and arranging the streamlets for playback on a content player.

* * * * *