IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

DISH TECHNOLOGIES L.L.C. and SLING TV L.L.C.,	)	
Plaintiffs,	)	C.A. No. _____
v.	)	
	)	**DEMAND FOR JURY TRIAL**
UNIVISION COMMUNICATIONS INC.,	)	
Defendant.	)	

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiffs DISH Technologies L.L.C. and Sling TV L.L.C. (collectively, "DISH") allege against Defendant Univision Communications Inc. ("Univision") as follows:

PARTIES

1. Plaintiff DISH Technologies L.L.C. is a limited liability company organized and existing under the laws of the State of Colorado, with its principal place of business at 9601 South Meridian Boulevard, Englewood, Colorado 80112. It provides innovation and technology services and products to, among others, the DISH Network® satellite pay TV service operated by DISH Network L.L.C. and the Sling TV® streaming pay TV service operated by Sling TV L.L.C.

2. Plaintiff Sling TV L.L.C. is a limited liability company organized and existing under the laws of the State of Colorado, with its principal place of business at 9601 South Meridian Boulevard, Englewood, Colorado 80112. It operates the Sling TV service.

3. On information and belief, Defendant Univision Communications Inc. is a corporation existing under the laws of the State of Delaware, with its principal place of business at 605 Third Avenue, New York, NY 10158. Univision has appointed The Corporation Trust

Company, Corporation Trust Center, 1209 Orange Street, Wilmington, Delaware 19801, as its agent for service of process.

JURISDICTION AND VENUE

4. DISH asserts a claim for patent infringement against Univision arising under the patent laws of the United States, Title 35 of the United States Code. Accordingly, this Court has subject matter jurisdiction over this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

5. This Court has personal jurisdiction over Univision for at least the following reasons: (1) Univision is incorporated in Delaware; (2) Univision has committed acts of patent infringement and contributed to and induced acts of patent infringement by others in this District; (3)Univision regularly does business or solicits business in this District; (4) Univision engages in other persistent courses of conduct and derives substantial revenue by offering and providing infringing products and services in this District; and (5) Univision has purposefully established substantial, systematic, and continuous contacts with this District and should reasonably expect to be haled into court here by its offering and providing of infringing products and services in this District.

6. Venue is proper in the District of Delaware under at least 28 §§ 1391(b), (c) and/or 1400(b). Univision is incorporated in Delaware. Additionally, the acts and transactions constituting the violations alleged herein occurred in part in this judicial district and Univision transacts business in this judicial district.

THE ABS PATENTS

7. On October 19, 2010, the United States Patent and Trademark Office ("PTO") duly and lawfully issued United States Patent No. 7,818,444 ("the '444 Patent"), entitled "Apparatus, system, and method for multi-bitrate content streaming." A true and correct

copy of the ’444 Patent is attached as Exhibit A. All rights, title, and interest in and to the ’444 Patent have been assigned to DISH Technologies L.L.C., which is the sole owner of the ’444 Patent.

8. On March 19, 2013, the PTO duly and lawfully issued United States Patent No. 8,402,156 (“the ’156 Patent”), entitled “Apparatus, system, and method for multi-bitrate content streaming.” A true and correct copy of the ’156 Patent is attached as Exhibit B. All rights, title, and interest in and to the ’156 Patent have been assigned to DISH Technologies L.L.C., which is the sole owner of the ’156 Patent.

9. On June 30, 2015, the PTO duly and lawfully issued United States Patent No. 9,071,668 (“the ’668 Patent”), entitled “Apparatus, system, and method for multi-bitrate content streaming.” A true and correct copy of the ’668 Patent is attached as Exhibit C. All rights, title, and interest in and to the ’668 Patent have been assigned to DISH Technologies L.L.C., which is the sole owner of the ’668 Patent.

10. On August 2, 2016, the PTO duly and lawfully issued United States Patent No. 9,407,564 (“the ’564 Patent”), entitled “Apparatus, system, and method for adaptive-rate shifting of streaming content.” A true and correct copy of the ’564 Patent is attached as Exhibit D. All rights, title, and interest in and to the ’564 Patent have been assigned to DISH Technologies L.L.C., which is the sole owner of the ’564 Patent.

11. DISH Technologies L.L.C. has entered into an exclusive license with Sling TV L.L.C. and assigned all substantial rights in the above identified patents to Sling TV L.L.C., including the right to sue thereon.

12. The claimed inventions in these patents are directed to various novel aspects and improvements to adaptive bitrate streaming (“ABS”) technology. The ’444, ’156,

’668, and ’564 Patents (collectively, “the ABS Patents”) are currently in full force and effect. The patent application underlying the ’564 Patent is a continuation of U.S. Patent Application No. 11/116,783. Each of the ’444, ’156, and ’668 Patents issued from patent applications that are continuations-in-part of U.S. Patent Application No. 11/116,783.

BACKGROUND OF THE DISPUTE

MOVE IS A PIONEER OF ADAPTIVE BITRATE TECHNOLOGY

13. MOVE Networks, Inc. (“MOVE”) was the original owner of the ABS Patents. Originally, Drew Major founded a company called XLon (renamed MOVE in 2006), which, in 2003, invented HTTP-based Adaptive Bitrate to improve the quality of streamed video content over the Internet. While at MOVE, inventors David Brueck, Mark Hurst, and Drew Major (collectively, “the ABS Inventors”) observed that the Internet was fast becoming a preferred method for distributing live and recorded video to individuals even though content delivery over the Internet at the time was notoriously unreliable, expensive and inferior in quality compared to cable and satellite delivered content. To access video content online, users were left with two mediocre choices: (1) waiting for their content to download (which did not support immediate viewing of live content and often required the user to select the quality desired: LOW, MEDIUM or HIGH, which in turn determined how long the user had to wait before viewing); or (2) streaming live or recorded content, which often was unreliable (pausing to “buffer”) or only worked at low-resolution.

14. The ABS Inventors knew that media streaming had not reached its full potential and that, through research and improvement, it was possible that streaming could rival the quality of cable and satellite delivered content. The current state-of-the-art was unacceptable prior to the inventions disclosed in the patents-in-suit. Often during playback, the streaming

technologies did a poor job selecting the video quality / resolution that the network bandwidth and reliability could support. Most commercial systems, from companies like RealNetworks, Adobe, Microsoft, or Apple, were proprietary implementations based on public Internet standards (RTP/RTSP). Common standards notwithstanding, the proprietary implementations were mutually incompatible. They were expensive to deploy by the Content Delivery Networks ("CDNs") and required many servers to scale to a large number of viewers. In addition, these technologies often required custom server architectures and routing IT configurations to penetrate Internet firewalls. The ABS Inventors recognized these shortcomings as an opportunity and developed a better solution.

15. The ABS Patents' specifications detail the need for improved data transport in content streaming. Users will generally choose streaming over downloading because "they tend to want to see or hear the media files instantaneously." *See, e.g.*, '668 Patent, Exhibit C, at col. 1, ll. 48–50. Unfortunately for protocols at the time, "[s]treaming offers the advantage of immediate access to the content but currently sacrifices quality compared with downloading a file of the same content." *See, e.g., id.* col. 1, lines 51–53. The ABS Inventors observed that "a need exists for an [invention] that alleviates the problems of reliability, efficiency, and latency" encountered in currently available content streaming systems. *See, e.g., id.* col. 2, ll. 39-41.

16. To address these needs, the ABS Inventors came up with a novel solution: HTTP-based Adaptive Bitrate Streaming. ABS segments the full content file into smaller units ("Streamlets") in multiple bitrates and delivers them over HTTP / TCP, the underlying protocols used for reliably transmitting data over the Internet. The ABS Inventors' approach enables content delivery to adapt to the bandwidth available at any particular time, ensuring delivery of

the highest possible quality content throughout the course of the stream. The playback client device continuously observes the quality of a user's network connection and adjusts the requested quality of the streamed content. The other RTP/RTSP-based technologies used a client / server architecture, where the server determined the bitrate to send to the client. The other technologies also did not segment the content, usually delivering it as a continuous stream of bits or as a single large file. Segmenting the content allows the playback device to easily change bitrates. The result is that today, MOVE's patented ABS technology allows Internet users to stream content from across the world in real time at the highest possible quality.

17. The ABS Patents specifications describe how the MOVE inventors significantly improved the user viewing experience of streaming content data over a network: "[A] need exists for an apparatus, system, and method that alleviate the problems of reliability, efficiency, and latency [during data transport streaming over a network]. 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." *See, e.g., id.* col. 2, ll. 37–43.

18. One unconventional but fundamental improvement described in the ABS patents is the creation of sets of streamlets from the original large content file, where a plurality of streamlets in each set are aligned by starting time and duration (typically a few seconds) but have different bitrates. Contiguous playback of the streamlets independently yields playback of the full content. The common alignment of the streamlets in each set allows a playback device to select one quality of streamlet from a particular set, and, as needed to adjust for changing bandwidth resources, to select a different quality of streamlet from the subsequent set. When the bandwidth of the user's network is constrained, the client can select a lower bitrate to maintain playback continuity instead of "buffering." This eliminates the need for users to download the

full content file before beginning playback. Segmenting the media into streamlets enables users to retrieve and enjoy content at the most appropriate bitrate possible as the media is streamed. It is also well suited for live stream playback.

19. Another non-routine and revolutionary improvement described in the ABS patents is that the client controls switching between different bitrates. The benefits of using an intelligent client to make the decisions and switch between different bitrate streamlets are two-fold. First, the client is in a better position to determine the appropriate streamlet by measuring the actual throughput of the network at its point of reception. Second, moving the decision-making to the client effectively eliminates the need for a customized video server. Instead a standard web server can be employed to host all the content's streamlets. Streamlets are requested by a client using the standard HTTP/TCP protocol—the web standard upon which the Internet is built. Custom IT configurations are unnecessary as the file requests operate on the same "port 80" as all web server requests. Access to the segmented content can be scaled exponentially through the use of standardized web caches. Together, these benefits represent a vast reduction in operating and publishing costs versus RTP/RTSP-based systems.

20. The ABS Inventors' improvements to streaming succeeded where others tried and failed. During the late 1990s, established streaming companies, including RealNetworks, Adobe, Microsoft, and Apple, separately attempted to develop a successful multiple bitrate streaming platform by using proprietary implementations of the RTP/RTSP standards. None of these systems succeeded at making bitrate switching consistently and actually work over the Internet.

ABS PATENTS SELL FOR $45 MILLION

21. DISH and its affiliated companies are a leading provider of satellite TV and Internet streaming services, employing approximately 16,000 people and serving nearly 13 million subscribers in the United States. It is a leading investor and innovator in infrastructure and technologies that will meet the personalized needs of its increasingly diverse pool of customers. Since its founding, DISH and its affiliated companies have invested millions in research and development and acquisition of novel technologies that will resolve long-felt problems and needs across its industry.

22. As the public continues to increasingly rely on the Internet for its informational and entertainment needs, one such problem into which DISH and its affiliated companies have dedicated great time and resources is improving the quality of streaming media. The specific entities that implement and own the technology covered by MOVE's patent portfolio have undergone significant evolution as these entities improve upon ABS and advance reliable delivery of high-resolution content over the Internet.

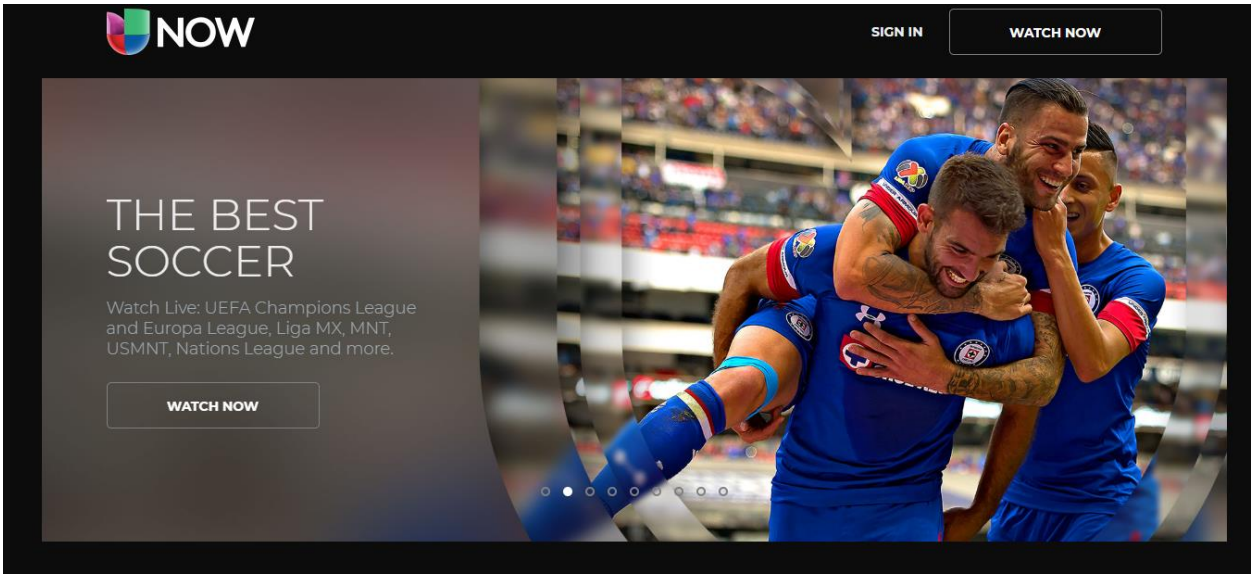
23. DISH's recent investments in ABS have already proven a success. ABS is one of the primary contributors to Sling TV's popularity. Sling TV L.L.C. is DISH and its affiliated companies' main Internet content provider, offering programming to numerous Internet streaming devices. Since the launch of Sling TV in the beginning of 2015, Sling TV has grown to over two million subscribers, who are now receiving a video playback experience comparable to cable or satellite.

UNIVISION'S PRODUCTS AND SERVICES INFRINGE THE ABS PATENTS

24. Univision has been and is now directly infringing and/or indirectly infringing the ABS Patents.

25. On information and belief, Univision is a distributor of content via the Internet and more specifically a distributor of Spanish language content. Exhibit E. Since 2018, Univision makes, uses, sells, and offers for sale in the United States its own branded streaming channel whose products and services infringe the ABS Patents. These infringing products include, without limitation, Univision NOW, the Univision app, and Univision Deportes services and related applications ("Univision NOW").

26. On information and belief, the Univision NOW service and related applications is a "digital video subscription service offering a live stream of Univision and UniMás and includes live sporting events, specials, series, and streams from local stations in select markets. Subscribers can instantly watch programs from the past 3 days (72 hours) of the live stream using the program guide, and can watch hundreds of primetime and library shows on demand." *Id.* The Univision and Univision Deportes services and related applications "offer viewers live streams / on demand content." *Id.* Here is an example of the Univision NOW service and related applications using a desktop platform:



CLAIMS FOR RELIEF
COUNT I: INFRINGEMENT OF U.S. PATENT NO. 7,818,444

DIRECT INFRINGEMENT

27. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-26 of the Complaint as if fully set forth herein.

28. On information and belief, Univision directly infringes, literally and/or under the doctrine of equivalents, at least claim 24 of the ’444 Patent, which recites:

> A method for multi-bitrate video and/or audio content streaming, the method comprising:
>
> capturing a single video;
>
> segmenting the single video and generating a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video;
>
> encoding each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the multi-bitrate content streaming, wherein each of the encoded streamlets in each of the sets is stored as a separate content file, wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the single video, wherein the separate content files within each of the sets are independently requestable by end user stations, and wherein shifts between the different bit rates are made at the time indexes during streaming of the single video;
>
> receiving requests from the end user stations over the Internet for different ones of the separate content files from different ones of the sets; and
>
> transmitting the requested ones of the separate content files to the requesting one of the end user stations, wherein each of the end user stations initiate each of the shifts between the different bit rates during streaming of the single video through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

Univision NOW receives segments of selected video program for playback of programming over a network connection. Univision NOW adapts requests for segments from a set of segments with the same content but varying quality based upon the quality of the network connection. Exhibit F to this Complaint is a claim chart that includes a more detailed infringement analysis of Univision NOW.[1]

29. For many years, DISH has been a satellite carrier and through Sling TV has been an Over the Top ("OTT") delivery company. OTT refers to content delivered via the Internet rather than delivery via satellite or cable. DISH carried Univision content for many years and as a content provider to DISH, Univision is well aware of DISH's operations, and in particular, is well aware of DISH's OTT service and technology including Sling TV. Without disclosing the specifics of any DISH / Univision content agreement, DISH carried Univision licensed content on both its satellite and Sling TV services and paid Univision based on that carriage. Thus, Univision is well aware that DISH has deployed an OTT system, since it charged DISH for the right to deliver Univision content on that system.

30. On information and belief, only recently in May of 2018 did Univision began providing a competing OTT service. In recent negotiations with DISH regarding future carriage of Univision content, Univision was put on notice of its infringement of the ABS Patents on July 25, 2018, and warned that further use required a license. Subsequent to that warning, Univision continued to provide OTT service in the U.S. using the ABS technology claimed in the ABS Patents and continued to offer and maintained the same service that DISH warned was

[1] DISH notes that Exhibits F and H–I, *see infra*, are based exclusively on publicly available information, and without the benefit of any Court claim construction. Accordingly, for each Count below, DISH reserves the right to supplement, amend or modify the analysis as warranted in light of additional facts, claim construction, or other developments. DISH further reserves the right to add additional claims as the case progresses.

infringing. Univision continues to provide Univision NOW with full knowledge and disregard of the ABS Patents, including the ’444 Patent.

31. Univision possesses knowledge of, and is aware of, the ’444 Patent, or became aware of this patent at the time of filing this lawsuit.

32. On information and belief, Univision intends to, and continues to intend to, directly infringe one or more claims of the ’444 Patent through the sale of Univision NOW.

33. On information and belief, Univision knew or should have known of the ’444 Patent and its infringement of the ’444 Patent, or at least learned of it by way of the Complaint, and has acted and continues to act, in an egregious and wanton manner by infringing the ’444 Patent.

34. On information and belief, despite knowing that its actions constituted infringement of the ’444 Patent and/or despite knowing that there was a high likelihood that its actions constituted infringement of the patent, Univision nevertheless continued its infringing actions, and continues to make, use, and sell Univision NOW.

35. Univision’s acts of infringement have injured and damaged DISH and will continue to injure and damage DISH.

36. Univision’s actions have caused DISH to suffer irreparable harm resulting from the loss of its lawful patent rights and the loss of its ability to exclude others from the market. Upon information and belief, Univision will continue these infringing acts unless enjoined by this Court.

INDIRECT INFRINGEMENT BY INDUCEMENT

37. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-36 of the Complaint as if fully set forth herein.

38. Univision is liable for inducing infringement of the ’444 Patent under 35 U.S.C. § 271(b) by having knowledge of the ’444 Patent and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, direct infringement of the ’444 Patent, with specific intent, by its customers.

39. Specifically, Univision actively induces infringement of the ’444 Patent by, *inter alia*, training its customers on the use of Univision NOW and/or promotion and/or sales of Univision NOW including Univision NOW, the Univision app, and Univision Deportes services and related applications to Univision’s customers including users and subscribers for implementing adaptive-rate content streaming as claimed in the ’444 Patent.

40. Univision’s customers directly infringe the ’444 Patent by using Univision NOW.

41. For example, Univision actively induces infringement of the ’444 Patent, because Univision has knowledge that Univision NOW customers including users and subscribers use Univision’s infringing Univision NOW service in the United States, and because Univision encourages such acts resulting in direct patent infringement, by, inter alia, training, promotion, and/or sales of Univision NOW to customers for their use of adaptive-rate content streaming as claimed in the ’444 Patent. *See* Exhibit E (answering questions such as, “How do I starting watching live TV?” and “What devices can I use to stream Univision NOW?”) (last visited Jan. 24, 2019).

42. On information and belief, Univision intends to, and continues to intend to, indirectly infringe the ’444 Patent through inducement of the use of Univision NOW.

43. The adaptive-rate content streaming technology market is a small and well-defined market with a few major players, including Apple, Microsoft, Adobe, and DISH,

since its acquisition and continuing development of MOVE's patent portfolio. On information and belief, as a provider of streamed content, Univision monitored developments of adaptive-rate content streaming technology, including DISH's ABS technology and knew, or at the very least, should have known, about the issuance of the '444 Patent.

44. On information and belief, Univision knew or should have known of the '444 Patent, as explained in Paragraphs 29–30, and has acted, and continues to act, in an egregious and wanton manner by infringing the '444 Patent.

45. On information and belief, despite knowing that its actions constituted inducement infringement of the '444 Patent and/or despite knowing that there was a high likelihood that its actions constituted inducement infringement of the patent, Univision nevertheless continued its infringing actions, and continues to make, use, and sell Univision NOW.

46. Univision continues to provide Univision NOW with full knowledge and disregard of the ABS Patents, including the '444 Patent.

47. Univision's acts of induced infringement have injured and damaged DISH and will continue to injure and damage DISH.

48. Univision's actions have caused DISH to suffer irreparable harm resulting from the loss of its lawful patent rights and the loss of its ability to exclude others from the market. Upon information and belief, Univision will continue these infringing acts unless enjoined by this Court.

INDIRECT INFRINGEMENT BY CONTRIBUTORY INFRINGEMENT

49. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-48 of the Complaint as if fully set forth herein.

50. Univision is liable for contributory infringement of the ’444 Patent under 35 U.S.C § 271(c) by having sold or offered to sell, and continuing to sell or offer for sale Univision NOW within the United States because Univision NOW constitutes a material part of the invention embodied in the ’444 Patent, which Univision knows to be especially made and/or especially adapted for use in infringement of the ’444 Patent, and which is not a staple article or commodity of commerce suitable for substantial non-infringing use.

51. Univision is liable for contributory infringement by having knowledge of the ’444 Patent and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, direct infringement of the ’444 Patent by its customers including users and subscribers who use Univision NOW.

52. Specifically, Univision contributes to infringement of the ’444 Patent by, inter alia, promotion, and/or sales of the infringing accused products and services to Univision’s customers including users and subscribers for their use of adaptive-rate content streaming as claimed in the ’444 Patent. Those customers directly infringe the ’444 Patent by using Univision NOW.

53. For example, Univision is liable for contributory infringement by having knowledge of the ’444 Patent, as explained in Paragraphs 29–30, and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, Univision NOW customers including users and subscribers to directly infringe the ’444 Patent by using Univision NOW in the United States.

54. The adaptive-rate content streaming technology market is a small and well-defined market with a few major players, including Apple, Microsoft, Adobe, and DISH, since its acquisition and continuing development of MOVE’s patent portfolio. On information

and belief, as a provider of streamed content, Univision would have monitored developments of adaptive-rate content streaming technology, including DISH's ABS technology and knew, or at the very least, should have known, about the issuance of the '444 Patent.

55. Univision continues to provide Univision NOW with full knowledge and disregard of the ABS Patents, including the '444 Patent.

56. Univision's past and ongoing infringement of the '444 Patent has and will continue to irreparably harm DISH.

57. Univision's past and ongoing infringement of the '444 Patent has and will continue to cause DISH damages.

58. Univision's past and ongoing infringement of the '444 Patent, upon information and belief, has been knowing and willful.

COUNT II: INFRINGEMENT OF U.S. PATENT NO. 8,402,156

DIRECT INFRINGEMENT

59. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-58 of the Complaint as if fully set forth herein.

60. On information and belief, Univision directly infringes, literally and/or under the doctrine of equivalents, at least claim 15 of the '156 Patent, which recites:

> A method for adaptive-rate content streaming videos for playback on a content player on an end user station, the method comprising:
>
> receiving a selected one of the videos for generating streamlets for adaptive-rate content streaming; and
>
> creating a plurality of different copies of the same selected video, wherein each of the different copies is encoded at a different bit rate and is divided into a plurality of streamlets that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the selected

> video, wherein the time indexes of the streamlets are the same for the different copies such that streamlets with the same time indexes form the different copies independently yield the same portions of the selected video, and wherein each of the streamlets of each of the pluralities is a separate content file that is independently playable by the end user station to thereby allow the end user station to initiate shifts in playback quality during streaming of the selected video through requests for separate content files storing different playback qualities of the encoded streamlets for subsequent ones of the time indexes.

Univision NOW receives segments of selected video program for playback of programming over a network connection. Univision NOW adapts requests for segments from a set of segments with the same content but varying quality based upon the quality of the network connection. Exhibit G to this Complaint is a claim chart with a more detailed infringement analysis of Univision NOW.

61. Univision possesses knowledge of, and is aware of, the ’156 Patent, or became aware of this patent at the time of filing this lawsuit.

62. On information and belief, Univision intends to, and continues to intend to, directly infringe one or more claims of the ’156 Patent through the sale of Univision NOW.

63. On information and belief, Univision knew or should have known of the ’156 Patent and its infringement of the ’156 Patent, or at least learned of it by way of the Complaint, and has acted and continues to act, in an egregious and wanton manner by infringing the ’156 Patent.

64. On information and belief, despite knowing that its actions constituted infringement of the ’156 Patent and/or despite knowing that there was a high likelihood that its actions constituted infringement of the patent, Univision nevertheless continued its infringing actions, and continues to make, use, and sell Univision NOW.

65. Univision's acts of infringement have injured and damaged DISH and will continue to injure and damage DISH.

66. Univision's actions have caused DISH to suffer irreparable harm resulting from the loss of its lawful patent rights and the loss of its ability to exclude others from the market. Upon information and belief, Univision will continue these infringing acts unless enjoined by this Court.

INDIRECT INFRINGEMENT BY INDUCEMENT

67. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-66 of the Complaint as if fully set forth herein.

68. Univision is liable for inducing infringement of the '156 Patent under 35 U.S.C. § 271(b) by having knowledge of the '156 Patent and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, direct infringement of the '156 Patent, with specific intent, by its customers.

69. Specifically, Univision actively induces infringement of the '156 Patent by, inter alia, training its customers on the use of Univision NOW and/or promotion and/or sales of Univision NOW including Univision NOW, the Univision app, and Univision Deportes services and related applications to Univision's customers including users and subscribers for implementing adaptive-rate content streaming as claimed in the '156 Patent.

70. Univision's customers directly infringe the '156 Patent by using Univision NOW.

71. For example, Univision actively induces infringement of the '156 Patent, because Univision has knowledge that Univision NOW customers including users and subscribers use Univision's infringing Univision NOW service in the United States, and because

Univision encourages such acts resulting in direct patent infringement, by, *inter alia*, training, promotion, and/or sales of Univision NOW to customers for their use of adaptive-rate content streaming as claimed in the ’156 Patent. *See* Exhibit E (answering questions such as, “How do I starting watching live TV?” and “What devices can I use to stream Univision NOW?”) (last visited Jan. 24, 2019).

72. On information and belief, Univision intends to, and continues to intend to, indirectly infringe the ’156 Patent through inducement of the use of Univision NOW.

73. The adaptive-rate content streaming technology market is a small and well-defined market with a few major players, including Apple, Microsoft, Adobe, and DISH, since its acquisition and continuing development of MOVE’s patent portfolio. On information and belief, as a provider of streamed content, Univision monitored developments of adaptive-rate content streaming technology, including DISH’s ABS technology and knew, or at the very least, should have known, about the issuance of the ’156 Patent.

74. On information and belief, Univision knew or should have known of the ’156 Patent, as explained in Paragraphs 29-30, and has acted, and continues to act, in an egregious and wanton manner by infringing the ’156 Patent.

75. On information and belief, despite knowing that its actions constituted inducement infringement of the ’156 Patent and/or despite knowing that there was a high likelihood that its actions constituted inducement infringement of the patent, Univision nevertheless continued its infringing actions, and continues to make, use, and sell Univision NOW.

76. Univision continues to provide Univision NOW with full knowledge and disregard of the ABS Patents, including the ’156 Patent.

77. Univision's acts of induced infringement have injured and damaged DISH and will continue to injure and damage DISH.

78. Univision's actions have caused DISH to suffer irreparable harm resulting from the loss of its lawful patent rights and the loss of its ability to exclude others from the market. Upon information and belief, Univision will continue these infringing acts unless enjoined by this Court.

INDIRECT INFRINGEMENT BY CONTRIBUTORY INFRINGEMENT

79. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-78 of the Complaint as if fully set forth herein.

80. Univision is liable for contributory infringement of the '156 Patent under 35 U.S.C § 271(c) by having sold or offered to sell, and continuing to sell or offer for sale Univision NOW within the United States because Univision NOW constitutes a material part of the invention embodied in the '156 Patent, which Univision knows to be especially made and/or especially adapted for use in infringement of the '156 Patent, and which is not a staple article or commodity of commerce suitable for substantial non-infringing use.

81. Univision is liable for contributory infringement by having knowledge of the '156 Patent and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, direct infringement of the '156 Patent by its customers including users and subscribers who use Univision NOW.

82. Specifically, Univision contributes to infringement of the '156 Patent by, inter alia, promotion, and/or sales of the infringing accused products and services to Univision's customers including users and subscribers for their use of adaptive-rate content streaming as

claimed in the ’156 Patent. Those customers directly infringe the ’156 Patent by using Univision NOW.

83. For example, Univision is liable for contributory infringement by having knowledge of the ’156 Patent, as explained in Paragraphs 29-30, and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, Univision NOW customers including users and subscribers to directly infringe the ’156 Patent by using Univision NOW in the United States.

84. The adaptive-rate content streaming technology market is a small and well-defined market with a few major players, including Apple, Microsoft, Adobe, and DISH, since its acquisition and continuing development of MOVE’s patent portfolio. On information and belief, as a provider of streamed content, Univision would have monitored developments of adaptive-rate content streaming technology, including DISH’s ABS technology and knew, or at the very least, should have known, about the issuance of the ’156 Patent.

85. Univision continues to provide Univision NOW with full knowledge and disregard of the ABS Patents, including the ’156 Patent.

86. Univision’s past and ongoing infringement of the ’156 Patent has and will continue to irreparably harm DISH.

87. Univision’s past and ongoing infringement of the ’156 Patent has and will continue to cause DISH damages.

88. Univision’s past and ongoing infringement of the ’156 Patent, upon information and belief, has been knowing and willful.

COUNT III: INFRINGEMENT OF U.S. PATENT NO. 9,071,668

DIRECT INFRINGEMENT

89. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-88 of the Complaint as if fully set forth herein.

90. On information and belief, Univision directly infringes, literally and/or under the doctrine of equivalents, at least claim 16 of the ’668 Patent, which recites:

> A method executable by a content player on an end user device to obtain a stream of a selected video program for playback by the content player, the method comprising:
>
> requesting the stream of the selected video program via a network connection to a video server, wherein the video server accesses a plurality of different copies of the same selected video each encoded at a different bit rate and each divided into a plurality of segments that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire playback duration of the selected video, wherein the time indexes of the segments are the same for the different copies such that the segments with the same time indexes from the different copies independently yield the same portions of the selected video, and wherein each of the segments of each of the pluralities is a separate content file that is independently playable by the end user device, wherein the requesting comprises the content player placing, for a set of sequential ones of the time indexes, segment requests over the network connection to the video server to thereby retrieve the separate segments from at least one of the different copies storing the portions of the single video according to the set of time indexes;
>
> receiving the separate segments from the video server at the content player via the network connection; and
>
> adapting subsequent segment requests placed by the content player to the video server based upon successive determinations by the content player to shift the playback quality to a higher or lower quality one of the different copies of the same selected video, wherein the shifts in playback quality occur at the time indexes.

Univision NOW receives segments of selected video program for playback of programming over a network connection. Univision NOW adapts requests for segments from a set of segments with the same content but varying quality based upon the quality of the network connection. Exhibit H to this Complaint is a claim chart with a more detailed infringement analysis of Univision NOW.

91. As explained in Paragraphs 29-30, Univision possesses knowledge of, and is aware of, the ’668 Patent, or became aware of this patent as a content provider to DISH and because Univision has received notice of infringement of the ’668 Patent.

92. On information and belief, Univision intends to, and continues to intend to, directly infringe one or more claims of the ’668 Patent through the sale of Univision NOW.

93. On information and belief, Univision knew or should have known of the ’668 Patent and its infringement of the ’668 Patent, or at least learned of it by way of the Complaint, and has acted and continues to act, in an egregious and wanton manner by infringing the ’668 Patent.

94. On information and belief, despite knowing that its actions constituted infringement of the ’668 Patent and/or despite knowing that there was a high likelihood that its actions constituted infringement of the patent, Univision nevertheless continued its infringing actions, and continues to make, use, and sell Univision NOW.

95. Univision’s acts of infringement have injured and damaged DISH and will continue to injure and damage DISH.

96. Univision’s actions have caused DISH to suffer irreparable harm resulting from the loss of its lawful patent rights and the loss of its ability to exclude others from the

market. Upon information and belief, Univision will continue these infringing acts unless enjoined by this Court.

INDIRECT INFRINGEMENT BY INDUCEMENT

97. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-96 of the Complaint as if fully set forth herein.

98. Univision is liable for inducing infringement of the ’668 Patent under 35 U.S.C. § 271(b) by having knowledge of the ’668 Patent and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, direct infringement of the ’668 Patent, with specific intent, by its customers.

99. Specifically, Univision actively induces infringement of the ’668 Patent by, inter alia, training its customers on the use of Univision NOW and/or promotion and/or sales of Univision NOW including Univision NOW, the Univision app, and Univision Deportes services and related applications to Univision’s customers including users and subscribers for implementing adaptive-rate content streaming as claimed in the ’668 Patent.

100. Univision’s customers directly infringe the ’668 Patent by using Univision NOW.

101. For example, Univision actively induces infringement of the ’668 Patent, because Univision has knowledge that Univision NOW customers including users and subscribers use Univision’s infringing Univision NOW service in the United States, and because Univision encourages such acts resulting in direct patent infringement, by, inter alia, training, promotion, and/or sales of Univision NOW to customers for their use of adaptive-rate content streaming as claimed in the ’668 Patent. *See* Exhibit E (answering questions such as, “How do I

starting watching live TV?" and "What devices can I use to stream Univision NOW?") (last visited Jan. 24, 2019).

102. On information and belief, Univision intends to, and continues to intend to, indirectly infringe the '668 Patent through inducement of the use of Univision NOW.

103. The adaptive-rate content streaming technology market is a small and well-defined market with a few major players, including Apple, Microsoft, Adobe, and DISH, since its acquisition and continuing development of MOVE's patent portfolio. On information and belief, as a provider of streamed content, Univision monitored developments of adaptive-rate content streaming technology, including DISH's ABS technology and knew, or at the very least, should have known, about the issuance of the '668 Patent.

104. On information and belief, Univision knew or should have known of the '668 Patent, as explained in Paragraphs 29-30, and has acted, and continues to act, in an egregious and wanton manner by infringing the '668 Patent.

105. On information and belief, despite knowing that its actions constituted inducement infringement of the '668 Patent and/or despite knowing that there was a high likelihood that its actions constituted inducement infringement of the patent, Univision nevertheless continued its infringing actions, and continues to make, use, and sell Univision NOW.

106. Univision continues to provide Univision NOW with full knowledge and disregard of the ABS Patents, including the '668 Patent.

107. Univision's acts of induced infringement have injured and damaged DISH and will continue to injure and damage DISH.

108. Univision's actions have caused DISH to suffer irreparable harm resulting from the loss of its lawful patent rights and the loss of its ability to exclude others from the market. Upon information and belief, Univision will continue these infringing acts unless enjoined by this Court.

INDIRECT INFRINGEMENT BY CONTRIBUTORY INFRINGEMENT

109. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-108 of the Complaint as if fully set forth herein.

110. Univision is liable for contributory infringement of the '668 Patent under 35 U.S.C § 271(c) by having sold or offered to sell, and continuing to sell or offer for sale Univision NOW within the United States because Univision NOW constitutes a material part of the invention embodied in the '668 Patent, which Univision knows to be especially made and/or especially adapted for use in infringement of the '668 Patent, and which is not a staple article or commodity of commerce suitable for substantial non-infringing use.

111. Univision is liable for contributory infringement by having knowledge of the '668 Patent and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, direct infringement of the '668 Patent by its customers including users and subscribers who use Univision NOW.

112. Specifically, Univision contributes to infringement of the '668 Patent by, inter alia, promotion, and/or sales of the infringing accused products and services to Univision's customers including users and subscribers for their use of adaptive-rate content streaming as claimed in the '668 Patent. Those customers directly infringe the '668 Patent by using Univision NOW.

113. For example, Univision is liable for contributory infringement by having knowledge of the ’668 Patent, as explained in Paragraphs 29-30, and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, Univision NOW customers including users and subscribers to directly infringe the ’668 Patent by using Univision NOW in the United States.

114. The adaptive-rate content streaming technology market is a small and well-defined market with a few major players, including Apple, Microsoft, Adobe, and DISH, since its acquisition and continuing development of MOVE’s patent portfolio. On information and belief, as a provider of streamed content, Univision would have monitored developments of adaptive-rate content streaming technology, including DISH’s ABS technology and knew, or at the very least, should have known, about the issuance of the ’668 Patent.

115. Univision continues to provide Univision NOW with full knowledge and disregard of the ABS Patents, including the ’668 Patent.

116. Univision’s past and ongoing infringement of the ’668 Patent has and will continue to irreparably harm DISH.

117. Univision’s past and ongoing infringement of the ’668 Patent has and will continue to cause DISH damages.

118. Univision’s past and ongoing infringement of the ’668 Patent, upon information and belief, has been knowing and willful.

COUNT IV: INFRINGEMENT OF U.S. PATENT NO. 9,407,564

DIRECT INFRINGEMENT

119. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-118 of the Complaint as if fully set forth herein.

120. On information and belief, Univision directly infringes, literally and/or under the doctrine of equivalents, at least claim 8 of the ’564 Patent, which recites:

> A method executable by an end user station to present rate-adaptive streams received via at least one transmission control protocol (TCP) connection with a server over a network, the method comprising;
>
> streaming, by a media player operating on the end user station, a video from the server via the at least one TCP connection over the network, wherein multiple different copies of the video encoded at different bit rates are stored as multiple sets of files on the server, wherein each of the files yields a different portion of the video on playback, wherein the files across the different copies yield the same portions of the video on playback, and wherein each of the files comprises a time index such that the files whose playback is the same portion of the video for each of the different copies have the same time index in relation to the beginning of the video, and wherein the streaming comprises:
>
> requesting by the media player a plurality of sequential files of one of the copies from the server based on the time indexes;
>
> automatically requesting by the media player from the server subsequent portions of the video by requesting for each such portion one of the files from one of the copies dependent upon successive determinations by the media player to shift the playback quality to a higher or lower quality one of the different copies, the automatically requesting including repeatedly generating a factor indicative of the current ability to sustain the streaming of the video using the files from different ones of the copies, wherein the factor relates to the performance of the network; and
>
> making the successive determinations to shift the playback quality based on the factor to achieve continuous playback of the video using the files of the highest quality one of the copies determined sustainable at that time, wherein the making the successive determinations to shift comprises upshifting to a higher quality one of the different copies when the at least one factor is greater than a first threshold and downshifting to a lower quality one of the different copies when the at least one factor is less than a second threshold; and
>
> presenting the video by playing back the requested media files with the media player on the end user station in order of ascending playback time.

Univision NOW receives segments of selected video program for playback of programming over a network connection. Univision NOW adapts requests for segments from a set of segments with the same content but varying quality based upon the quality of the network connection. Exhibit I to this Complaint is a claim chart with a more detailed infringement analysis of Univision NOW.

121. Univision possesses knowledge of, and is aware of, the ’564 Patent, or became aware of this patent at the time of filing this lawsuit.

122. On information and belief, Univision intends to, and continues to intend to, directly infringe one or more claims of the ’564 Patent through the sale of Univision NOW.

123. On information and belief, Univision knew or should have known of the ’564 Patent and its infringement of the ’564 Patent, or at least learned of it by way of the Complaint, and has acted and continues to act, in an egregious and wanton manner by infringing the ’564 Patent.

124. On information and belief, despite knowing that its actions constituted infringement of the ’564 Patent and/or despite knowing that there was a high likelihood that its actions constituted infringement of the patent, Univision nevertheless continued its infringing actions, and continues to make, use, and sell Univision NOW.

125. Univision’s acts of infringement have injured and damaged DISH and will continue to injure and damage DISH.

126. Univision’s actions have caused DISH to suffer irreparable harm resulting from the loss of its lawful patent rights and the loss of its ability to exclude others from the market. Upon information and belief, Univision will continue these infringing acts unless enjoined by this Court.

INDIRECT INFRINGEMENT BY INDUCEMENT

127. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-126 of the Complaint as if fully set forth herein.

128. Univision is liable for inducing infringement of the ’564 Patent under 35 U.S.C. § 271(b) by having knowledge of the ’564 Patent and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, direct infringement of the ’564 Patent, with specific intent, by its customers.

129. Specifically, Univision actively induces infringement of the ’564 Patent by, inter alia, training its customers on the use of Univision NOW and/or promotion and/or sales of Univision NOW including Univision NOW, the Univision app, and Univision Deportes services and related applications to Univision’s customers including users and subscribers for implementing adaptive-rate content streaming as claimed in the ’564 Patent.

130. Univision’s customers directly infringe the ’564 Patent by using Univision NOW.

131. For example, Univision actively induces infringement of the ’564 Patent, because Univision has knowledge that Univision NOW customers including users and subscribers use Univision’s infringing Univision NOW service in the United States, and because Univision encourages such acts resulting in direct patent infringement, by, inter alia, training, promotion, and/or sales of Univision NOW to customers for their use of adaptive-rate content streaming as claimed in the ’564 Patent. *See* Exhibit E (answering questions such as, “How do I starting watching live TV?” and “What devices can I use to stream Univision NOW?”) (last visited Jan. 24, 2019).

132. On information and belief, Univision intends to, and continues to intend to, indirectly infringe the ’564 Patent through inducement of the use of Univision NOW.

133. The adaptive-rate content streaming technology market is a small and well-defined market with a few major players, including Apple, Microsoft, Adobe, and DISH, since its acquisition and continuing development of MOVE’s patent portfolio. On information and belief, as a provider of streamed content, Univision monitored developments of adaptive-rate content streaming technology, including DISH’s ABS technology and knew, or at the very least, should have known, about the issuance of the ’564 Patent.

134. On information and belief, Univision knew or should have known of the ’564 Patent, as explained in Paragraphs 29-30, and has acted, and continues to act, in an egregious and wanton manner by infringing the ’564 Patent.

135. On information and belief, despite knowing that its actions constituted inducement infringement of the ’564 Patent and/or despite knowing that there was a high likelihood that its actions constituted inducement infringement of the patent, Univision nevertheless continued its infringing actions, and continues to make, use, and sell Univision NOW.

136. Univision continues to provide Univision NOW with full knowledge and disregard of the ABS Patents, including the ’564 Patent.

137. Univision’s acts of induced infringement have injured and damaged DISH and will continue to injure and damage DISH.

138. Univision’s actions have caused DISH to suffer irreparable harm resulting from the loss of its lawful patent rights and the loss of its ability to exclude others from the

market. Upon information and belief, Univision will continue these infringing acts unless enjoined by this Court.

INDIRECT INFRINGEMENT BY CONTRIBUTORY INFRINGEMENT

139. DISH re-alleges and incorporates herein by reference the allegations contained in Paragraphs 1-138 of the Complaint as if fully set forth herein.

140. Univision is liable for contributory infringement of the ’564 Patent under 35 U.S.C § 271(c) by having sold or offered to sell, and continuing to sell or offer for sale Univision NOW within the United States because Univision NOW constitutes a material part of the invention embodied in the ’564 Patent, which Univision knows to be especially made and/or especially adapted for use in infringement of the ’564 Patent, and which is not a staple article or commodity of commerce suitable for substantial non-infringing use.

141. Univision is liable for contributory infringement by having knowledge of the ’564 Patent and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, direct infringement of the ’564 Patent by its customers including users and subscribers who use Univision NOW.

142. Specifically, Univision contributes to infringement of the ’564 Patent by, inter alia, promotion, and/or sales of the infringing accused products and services to Univision’s customers including users and subscribers for their use of adaptive-rate content streaming as claimed in the ’564 Patent. Those customers directly infringe the ’564 Patent by using Univision NOW.

143. For example, Univision is liable for contributory infringement by having knowledge of the ’564 Patent, as explained in Paragraphs 29-30, and knowingly causing or intending to cause, and continuing to knowingly cause or intend to cause, Univision NOW

customers including users and subscribers to directly infringe the ’564 Patent by using Univision NOW in the United States.

144. The adaptive-rate content streaming technology market is a small and well-defined market with a few major players, including Apple, Microsoft, Adobe, and DISH, since its acquisition and continuing development of MOVE’s patent portfolio. On information and belief, as a provider of streamed content, Univision would have monitored developments of adaptive-rate content streaming technology, including DISH’s ABS technology and knew, or at the very least, should have known, about the issuance of the ’564 Patent.

145. Univision continues to provide Univision NOW with full knowledge and disregard of the ABS Patents, including the ’564 Patent.

146. Univision’s past and ongoing infringement of the ’564 Patent has and will continue to irreparably harm DISH.

147. Univision’s past and ongoing infringement of the ’564 Patent has and will continue to cause DISH damages.

148. Univision’s past and ongoing infringement of the ’564 Patent, upon information and belief, has been knowing and willful.

JURY DEMAND

Pursuant to Fed. R. Civ. P. 38(b), Plaintiffs hereby request a trial by jury of all issues so triable.

PRAYER FOR RELIEF

WHEREFORE, DISH respectfully requests that this Court enter:

A. A judgement in favor of DISH that Univision has infringed the ABS Patents, directly, jointly, and/or indirectly by way of inducing and/or contributing to the infringement of the ABS Patents;

B. An order of this Court permanently enjoining Univision and its officers, directors, agents, affiliates, employees, divisions, branches, subsidiaries, parents, and all others in active concert therewith from infringing, including inducing the infringement of, or contributing to the infringement of, the ABS Patents;

C. A judgment and order requiring Univision to pay DISH its damages, costs, expenses, and pre-judgment and post-judgment interest for Univision's infringement of the ABS Patents, as provided under 35 U.S.C. § 284;

D. A judgment and order requiring Univision to pay treble damages as provided under 35 U.S.C. § 284;

E. A judgment and order finding this case exceptional under 35 U.S.C. § 285 and awarding DISH its costs, disbursements, and attorneys' fees in connection with this action; and

F. Such other and further relief to which DISH may show itself to be entitled and/or as the Court may deem just and proper.

MORRIS, NICHOLS, ARSHT & TUNNELL LLP

/s/ Rodger D. Smith II

Rodger D. Smith II (#3778)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
(302) 658-9200
rsmith@mnat.com

Attorneys for Plaintiffs

OF COUNSEL:

G. Hopkins Guy, III
BAKER BOTTS L.L.P.
1001 Page Mill Road
Building One, Suite 200
Palo Alto, CA 94304
(650) 739-7500

Ali Dhanani
BAKER BOTTS L.L.P.
One Shell Plaza
San Francisco, CA 94111
(281) 250-2294

Jamie R. Lynn
BAKER BOTTS L.L.P.
The Warner
1299 Pennsylvania Avenue, NW
Washington, DC 20004
(202) 639-7700

January 25, 2019

EXHIBIT A

U 7701579

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

November 21, 2018

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF:

U.S. PATENT: ***7,818,444***
ISSUE DATE: ***October 19, 2010***

By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office

SYLVIA HOLLEY
Certifying Officer

US007818444B2

(12) **United States Patent**
Brueck et al.

(10) **Patent No.: US 7,818,444 B2**
(45) **Date of Patent: Oct. 19, 2010**

(54) **APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING**

(75) Inventors: **David F. Brueck**, Saratoga Springs, UT (US); **Mark B. Hurst**, Cedar Hills, UT (US)

(73) Assignee: **Move Networks, Inc.**, American Fork, UT (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 319 days.

(21) Appl. No.: **11/673,483**

(22) Filed: **Feb. 9, 2007**

(65) **Prior Publication Data**

US 2008/0195743 A1 Aug. 14, 2008

Related U.S. Application Data

(63) Continuation-in-part of application No. 11/116,783, filed on Apr. 28, 2005.

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

(51) **Int. Cl.**
G06F 15/16 (2006.01)

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

(58) **Field of Classification Search** 709/231, 709/232; 375/240.02
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,535,355	A	8/1985	Arn et al.	
5,953,506	A	9/1999	Kalra et al.	
6,091,775	A	7/2000	Hibi et al.	
6,091,777	A	7/2000	Guetz et al.	
6,195,680	B1	2/2001	Goldszmidt et al.	709/203
6,366,614	B1	4/2002	Pian et al.	375/240.02
6,486,803	B1 *	11/2002	Luby et al.	341/50
6,490,627	B1	12/2002	Kalra et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

CA	2466482	5/2003

(Continued)

OTHER PUBLICATIONS

Pathan, Al-Mukaddim, et al., "A Taxonomy and Survey of Content Delivery Networks", Australia, Feb. 2007. Available at http://www.gridbus.org/reports/CDN-Taxonomy.pdf.

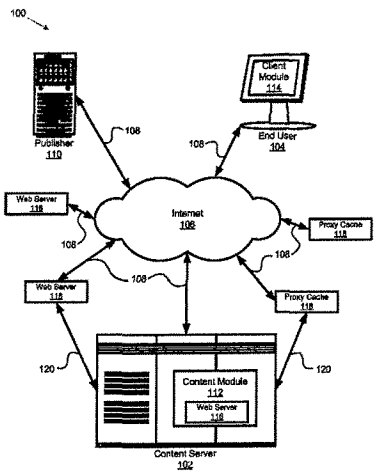
(Continued)

Primary Examiner—Nathan Flynn
Assistant Examiner—Chirag Patel
(74) *Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafman, LLP

(57) **ABSTRACT**

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.

55 Claims, 12 Drawing Sheets



U.S. PATENT DOCUMENTS

6,510,553	B1	1/2003	Hazra	
6,574,591	B1	6/2003	Kleiman et al.	
6,604,118	B2	8/2003	Kleiman et al.	
6,731,600	B1	5/2004	Patel et al.	
6,732,183	B1	5/2004	Graham	
6,968,387	B2	11/2005	Lanphear	
7,075,986	B2	7/2006	Girod et al.	
7,096,271	B1	8/2006	Omoigui et al.	
7,349,976	B1	3/2008	Glaser et al.	
7,408,984	B2 *	8/2008	Lu et al.	375/240.02
2002/0073167	A1	6/2002	Powell et al.	
2002/0144276	A1	10/2002	Radford et al.	
2002/0174434	A1	11/2002	Lee et al.	
2002/0188745	A1	12/2002	Hughes et al.	
2003/0005455	A1 *	1/2003	Bowers	725/90
2003/0067872	A1	4/2003	Harrell et al.	
2003/0081582	A1	5/2003	Jain et al.	
2003/0107994	A1	6/2003	Jacobs et al.	
2003/0135631	A1	7/2003	Li et al.	
2003/0151753	A1	8/2003	Li et al.	358/1.9
2003/0152036	A1	8/2003	Quigg Brown et al.	
2003/0204519	A1 *	10/2003	Sirivara et al.	707/101
2004/0003101	A1	1/2004	Roth et al.	
2004/0030797	A1	2/2004	Akinlar et al.	
2004/0031054	A1	2/2004	Dankworth et al.	
2004/0054551	A1 *	3/2004	Ausubel et al.	705/1
2004/0071209	A1	4/2004	Burg et al.	
2004/0093420	A1	5/2004	Gamble	
2004/0143672	A1 *	7/2004	Padmanabham et al.	709/231
2004/0170392	A1 *	9/2004	Lu et al.	386/96
2005/0108414	A1	5/2005	Taylor et al.	709/231
2005/0123058	A1	6/2005	Greenbaum et al.	
2007/0024705	A1	2/2007	Richter et al.	
2008/0028428	A1	1/2008	Jeong et al.	
2008/0056373	A1	3/2008	Newlin et al.	
2008/0162713	A1 *	7/2008	Bowra et al.	709/231
2008/0195744	A1	8/2008	Bowra et al.	
2008/0195745	A1	8/2008	Bowra et al.	

FOREIGN PATENT DOCUMENTS

EP	0 711 077	A2	5/1996
EP	0 919 952	A1	6/1999
EP	1 641 271	A2	3/2006
EP	1 670 256	A2	6/2006
EP	1 777 969	A1	4/2007
WO	WO 00/67469		11/2000

OTHER PUBLICATIONS

On2 Technologies, Inc., "TrueMotion VP7 Video Codec", White Paper, Document Version 1.0, Jan. 10, 2005, (13 pages).
Supplemental European Search Report, Sep. 22, 2008, (3 pages).
Office Action mailed May 14, 2008 for U.S. Appl. No. 11/116,783, filed on Apr. 28, 2005.
Final Office Action mailed Feb. 20, 2009 for U.S. Appl. No. 11/116,783, filed on Apr. 28, 2005.
Intellegent Streaming, Bill Birney; May 2003, Microsoft, pp. 7 total.
Advisory Action for U.S. Appl. No. 11/116,783, Mailed Apr. 02, 2009, 3 pages.
Advisory Action for U.S. Appl. No. 11/116,783, Mailed May 12, 2009, 4 pages.
Wicker, Stephen B., "Error Control Systems for Digital Communication and Storage", Prentice-Hall, Inc., New Jersey, USA, 1995 (Book: see NPL's Parts 1-6).
PCT Notification of Transmittal of the International Search Report and Written Opinion of the International Searching Authority, for PCT/US05/15091, Oct. 29, 2007, 8 pages.
PCT Notification of Transmittal of International Preliminary Report on Patentability, for PCT/US05/15091, Oct. 29, 2007, 6 pages.
Office Action for U.S. Appl. No. 11/116,783, Aug. 20, 2009, 14 pages.
Albanese, Andres, et al. "Priority Encoding Transmission", TR-94-039, Aug. 1994, 36 pages, International Computer Science Institute, Berkeley, California.
Puri, Rohit, et al. "Multiple Description Source Coding Using Forward Error Correction Codes", Oct. 1999, 5 pages, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California.
Goyal, Vivek K., "Multiple Description Coding: Compression Meets the Network", Sep. 2001, pp. 74-93, IEEE Signal Processing Magazine.
Advisory Action for U.S. Appl. No. 11/116,783, Mailed Mar. 23, 2010, 3 pages.
Advisory Action for U.S. Appl. No. 11/116,783, Mailed May 17, 2010, 3 pages.
Final Office Action for U.S. Appl. No. 11/116,783, Mailed Feb. 22, 2010, 16 pages.

* cited by examiner

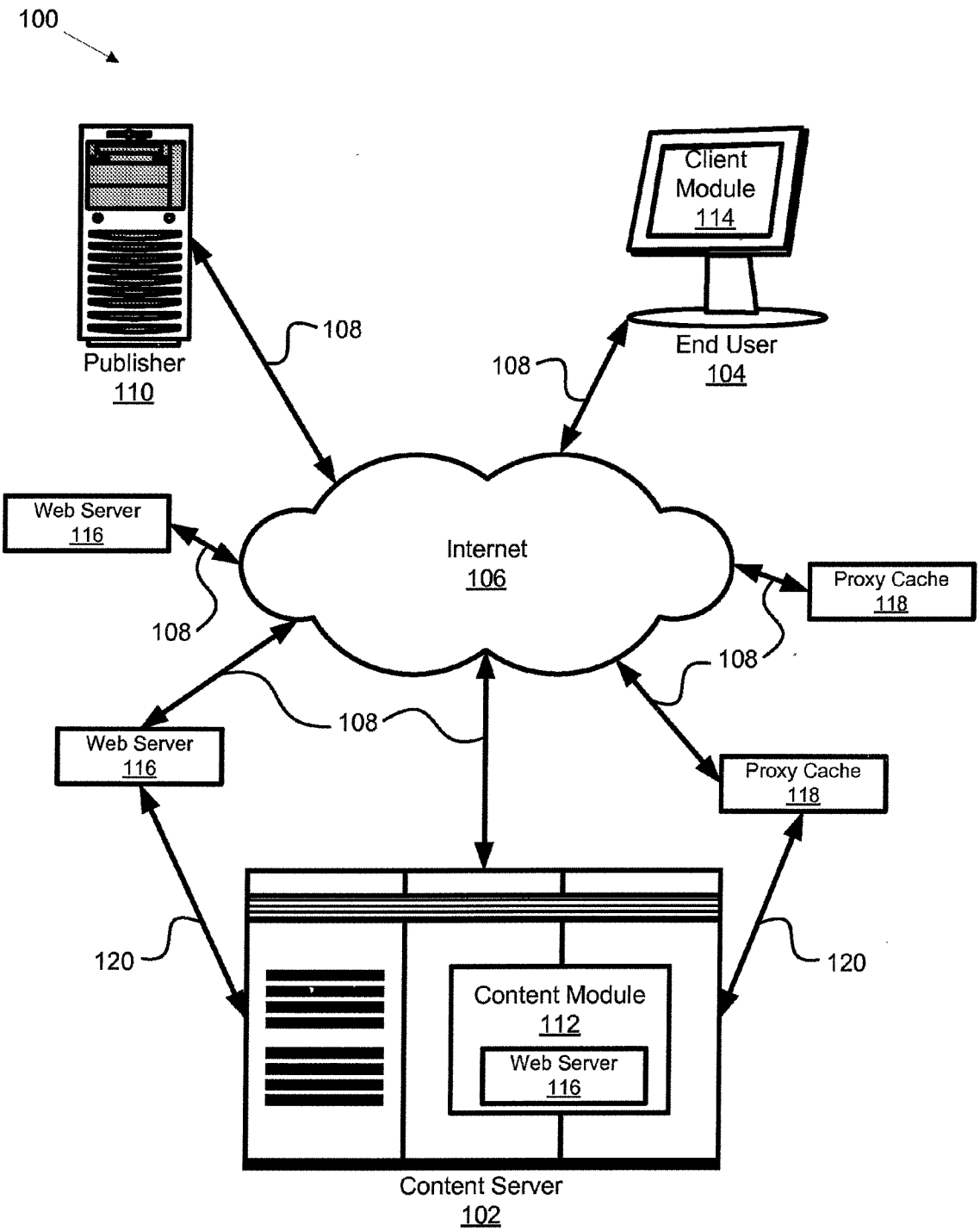


FIG. 1

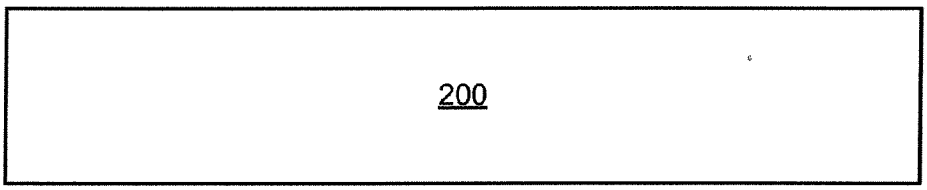


FIG. 2A

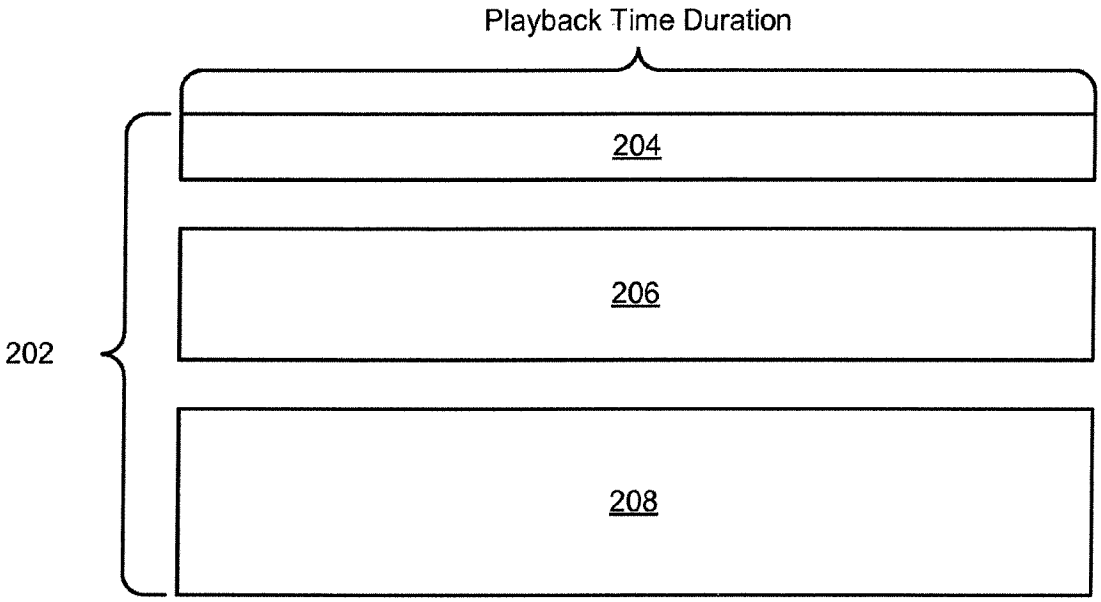


FIG. 2B

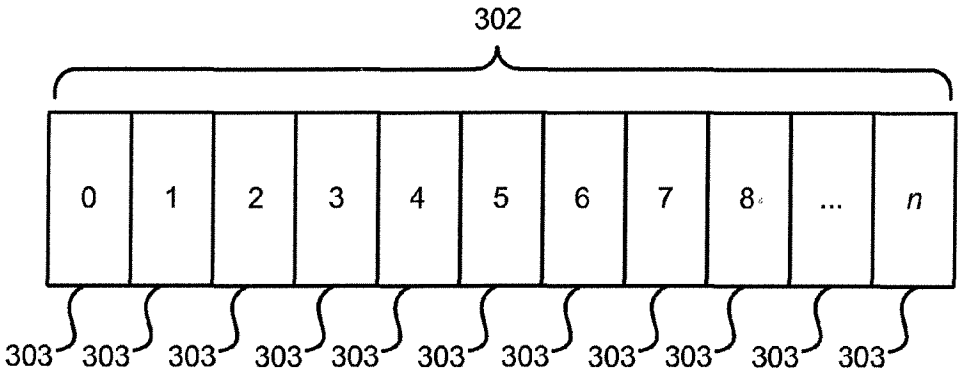


FIG. 3A

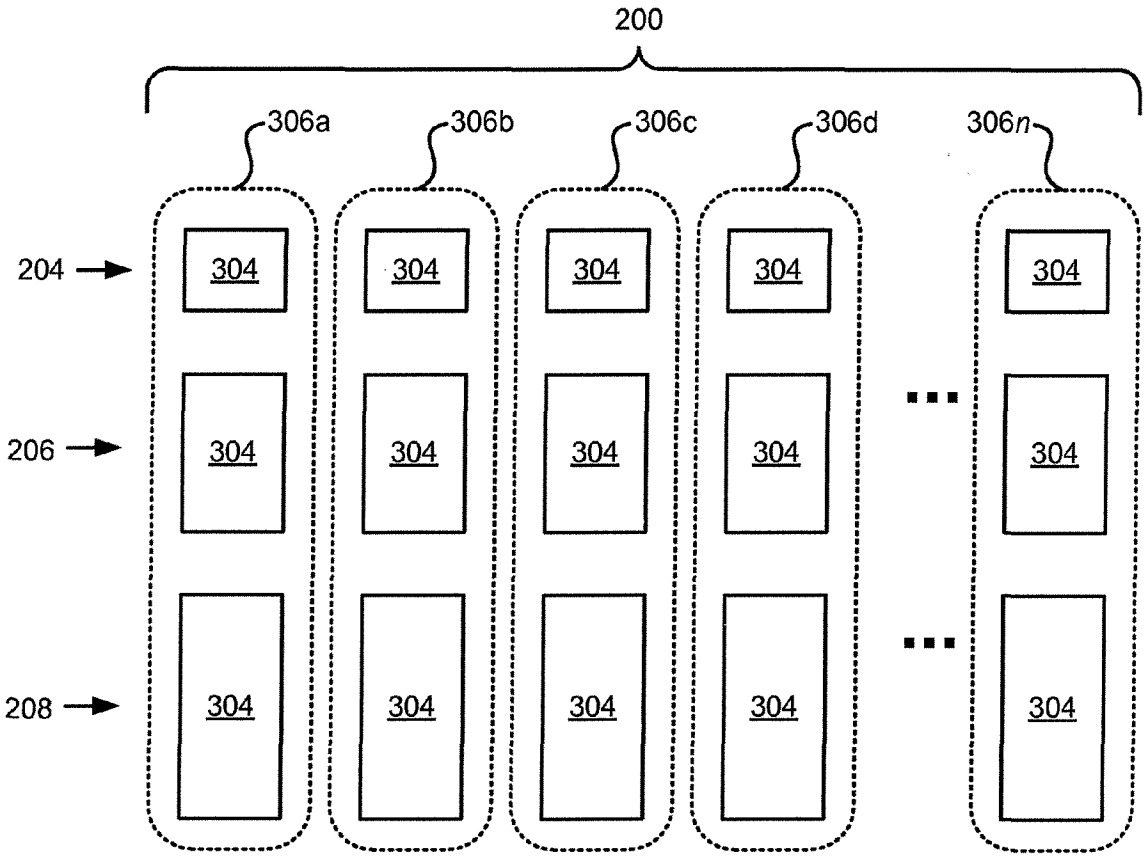


FIG. 3B

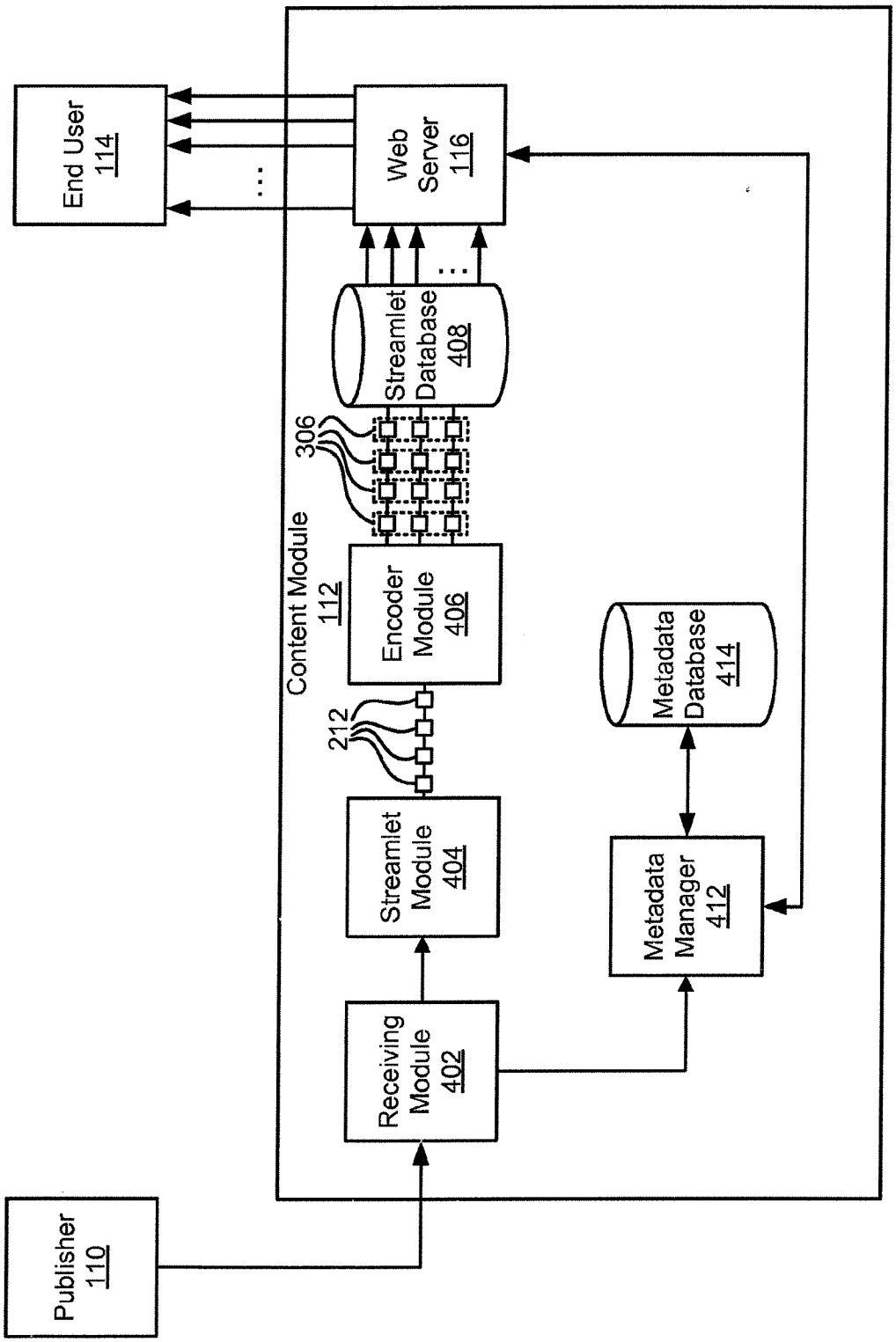


FIG. 4

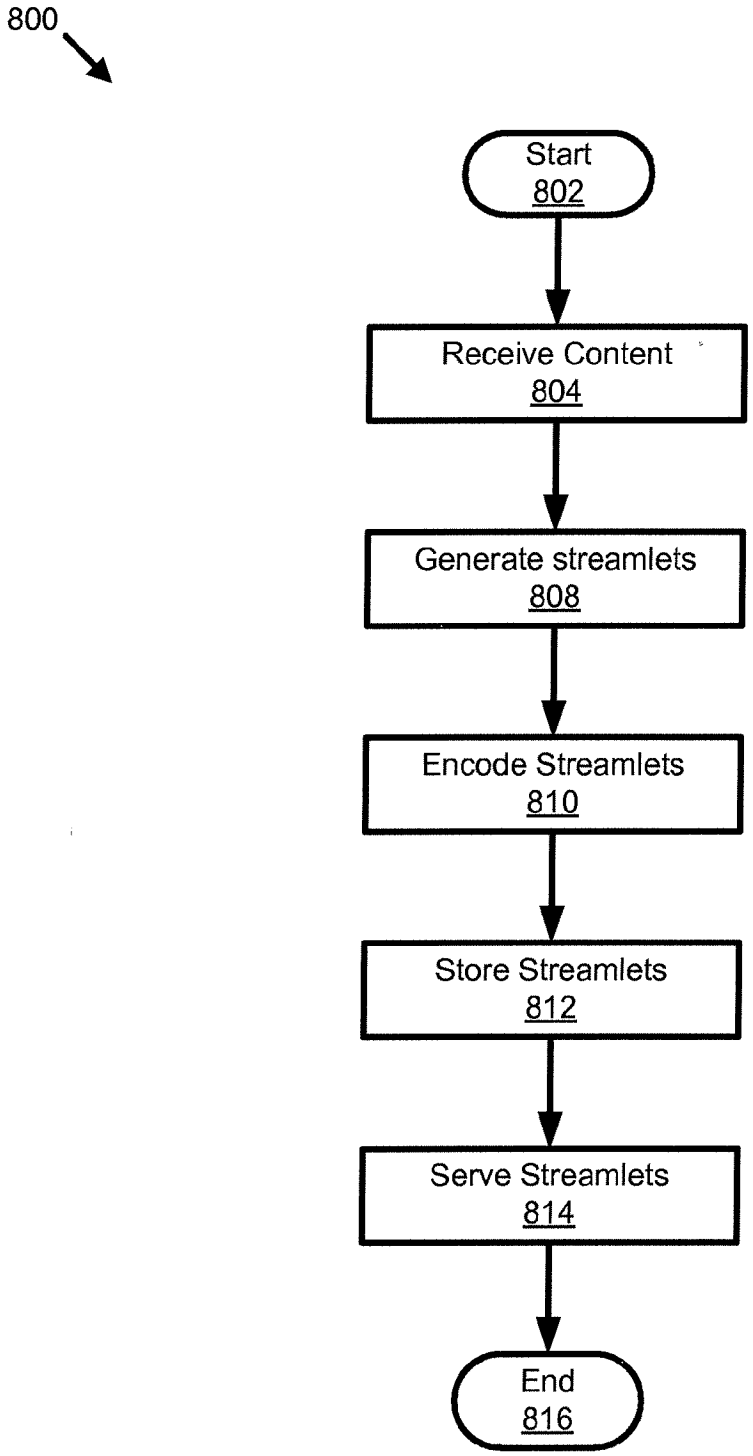


FIG. 5

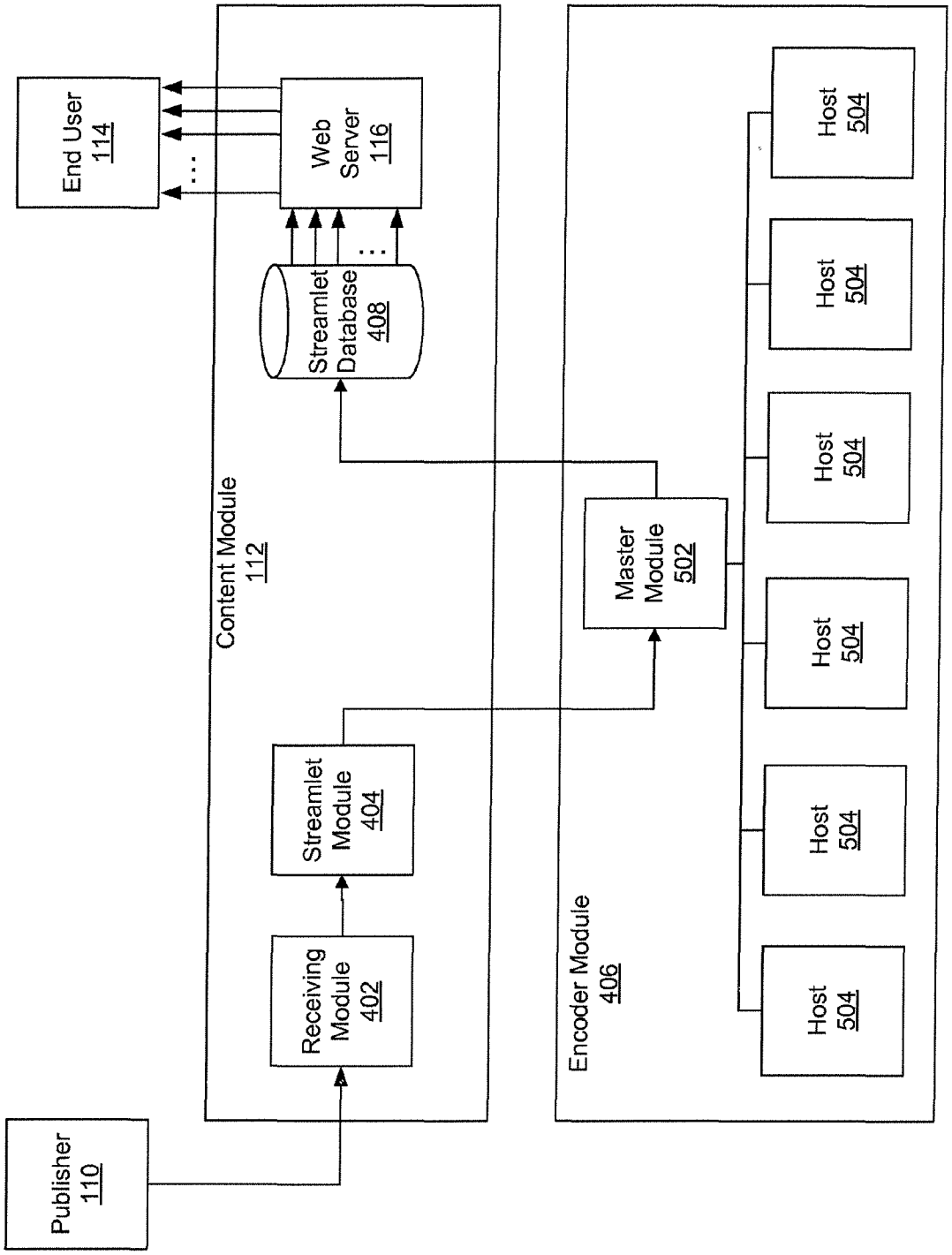


FIG. 5a

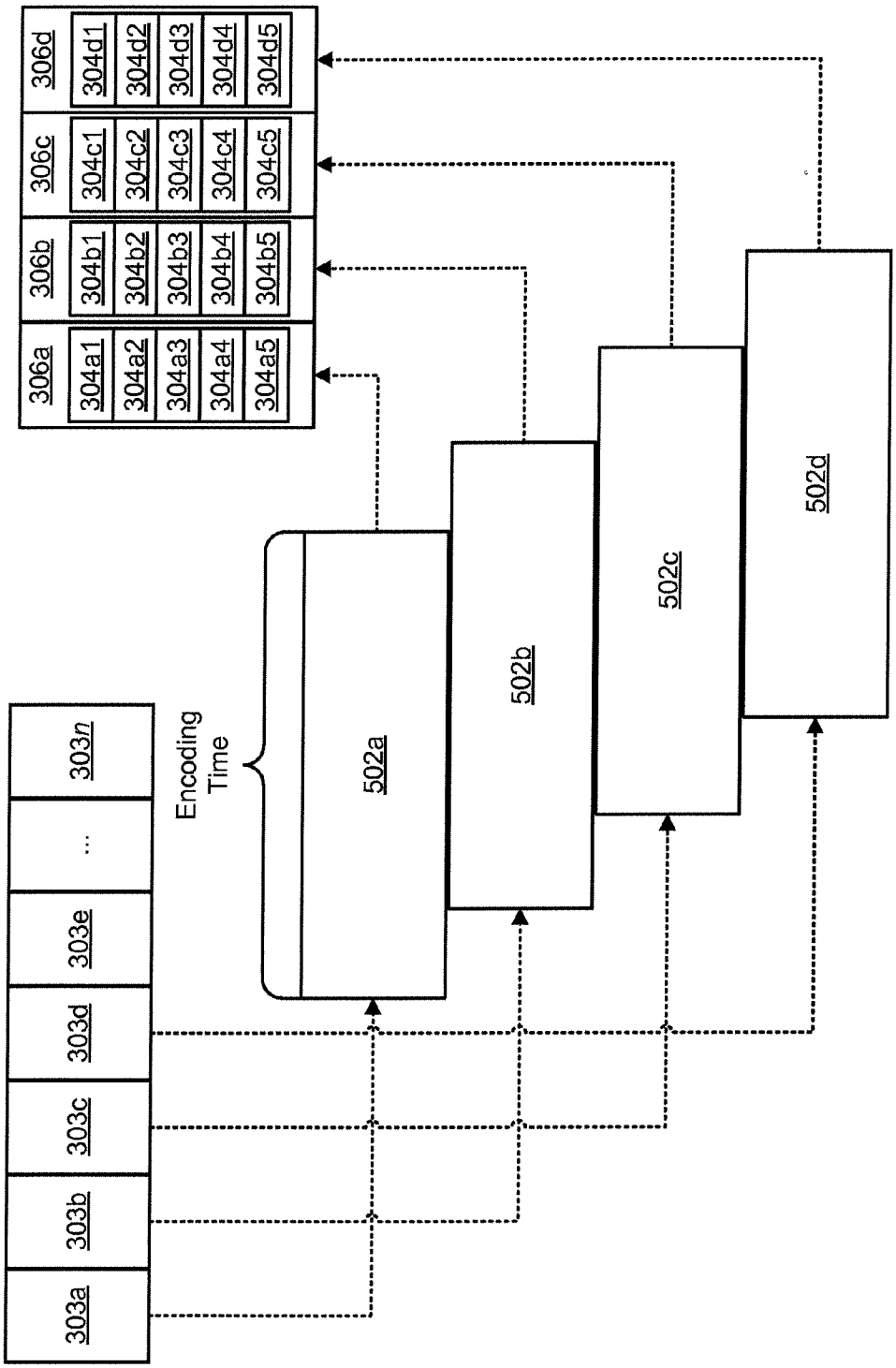


FIG. 5B

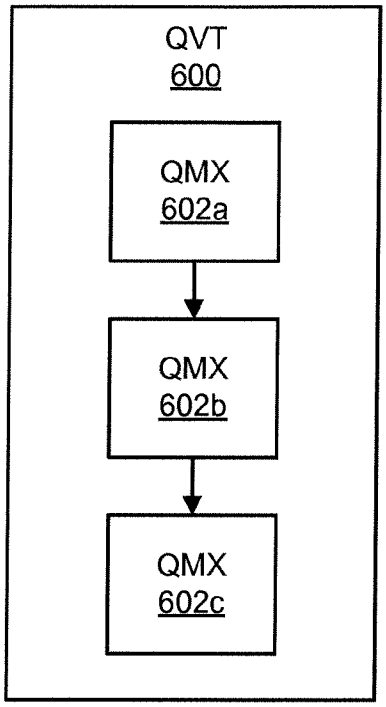


FIG. 6A

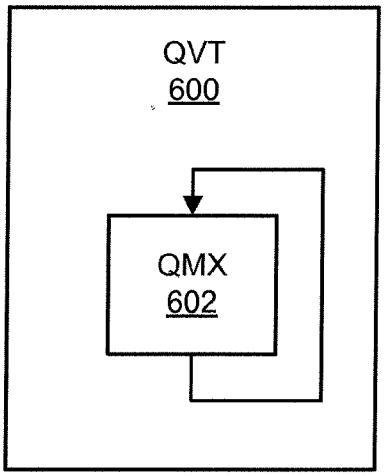


FIG. 6B

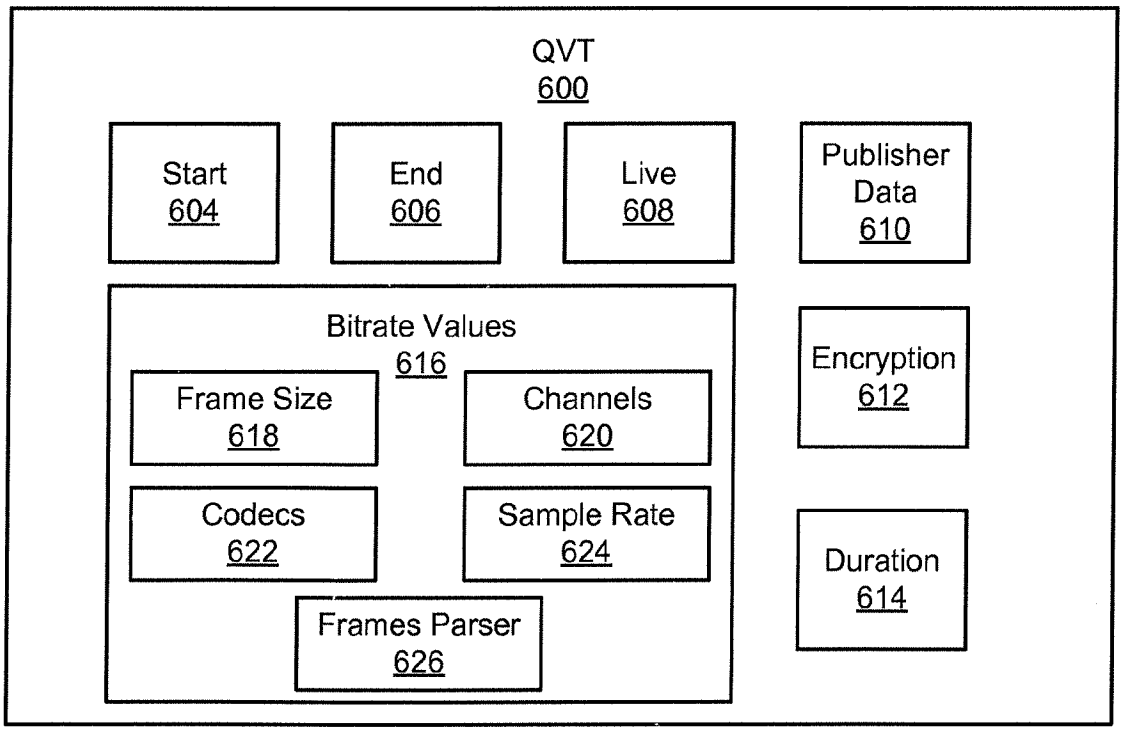


FIG. 6C

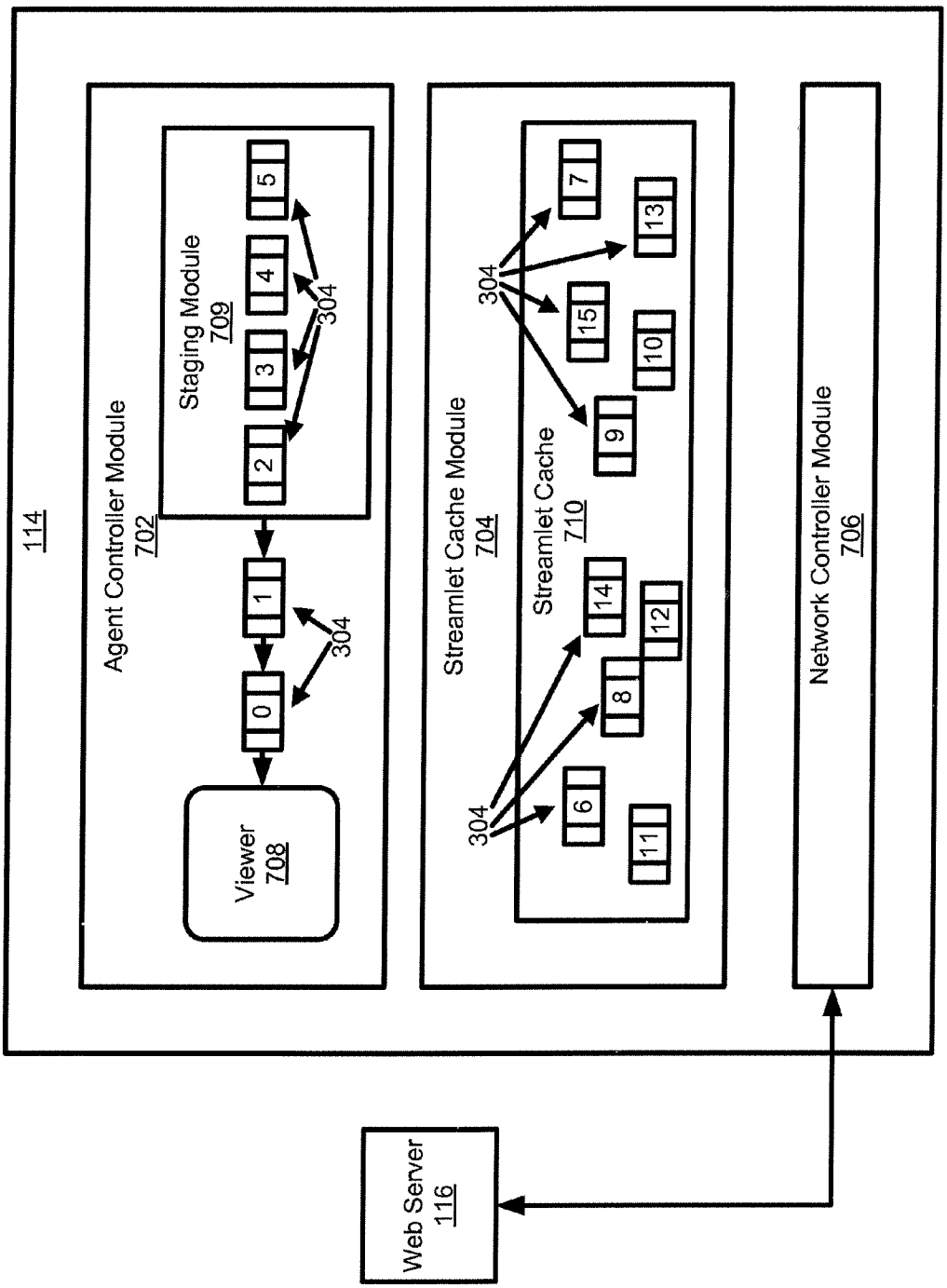


FIG. 7

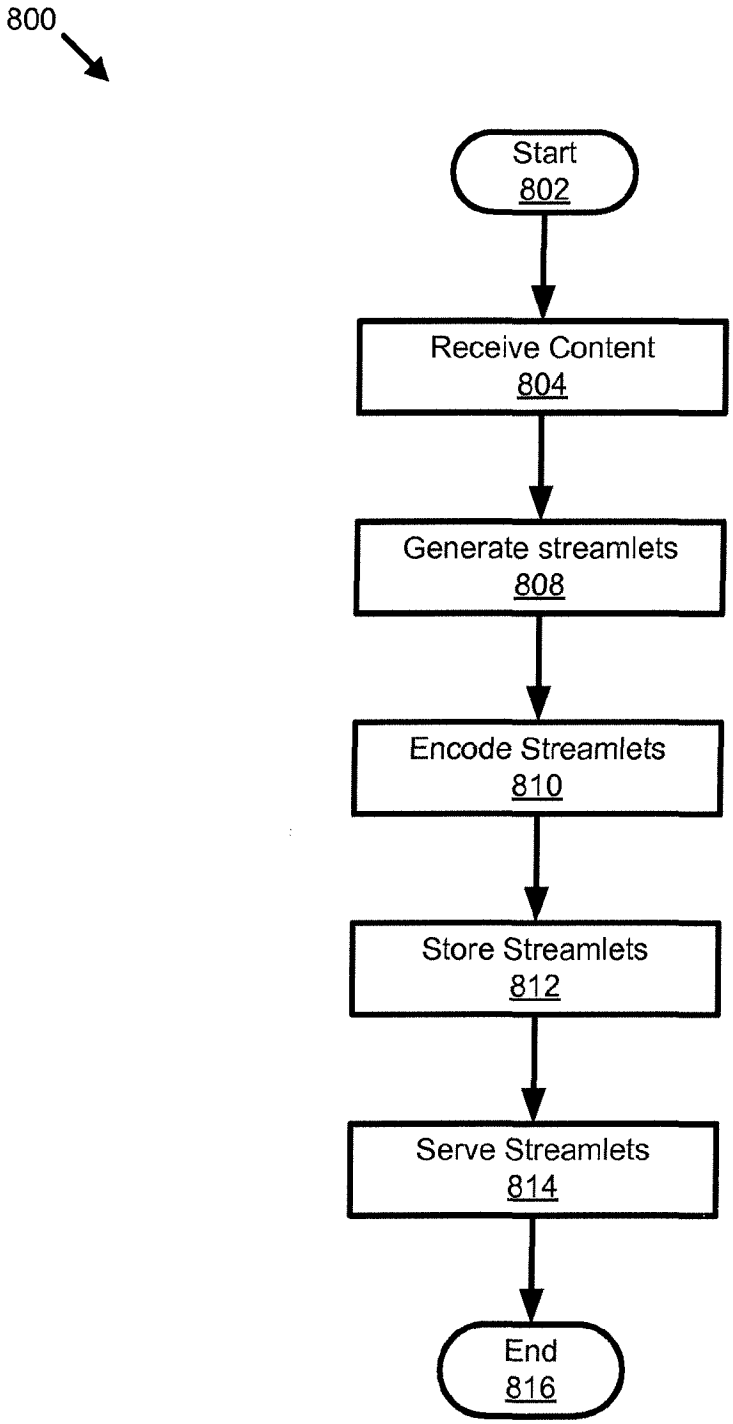


FIG. 8

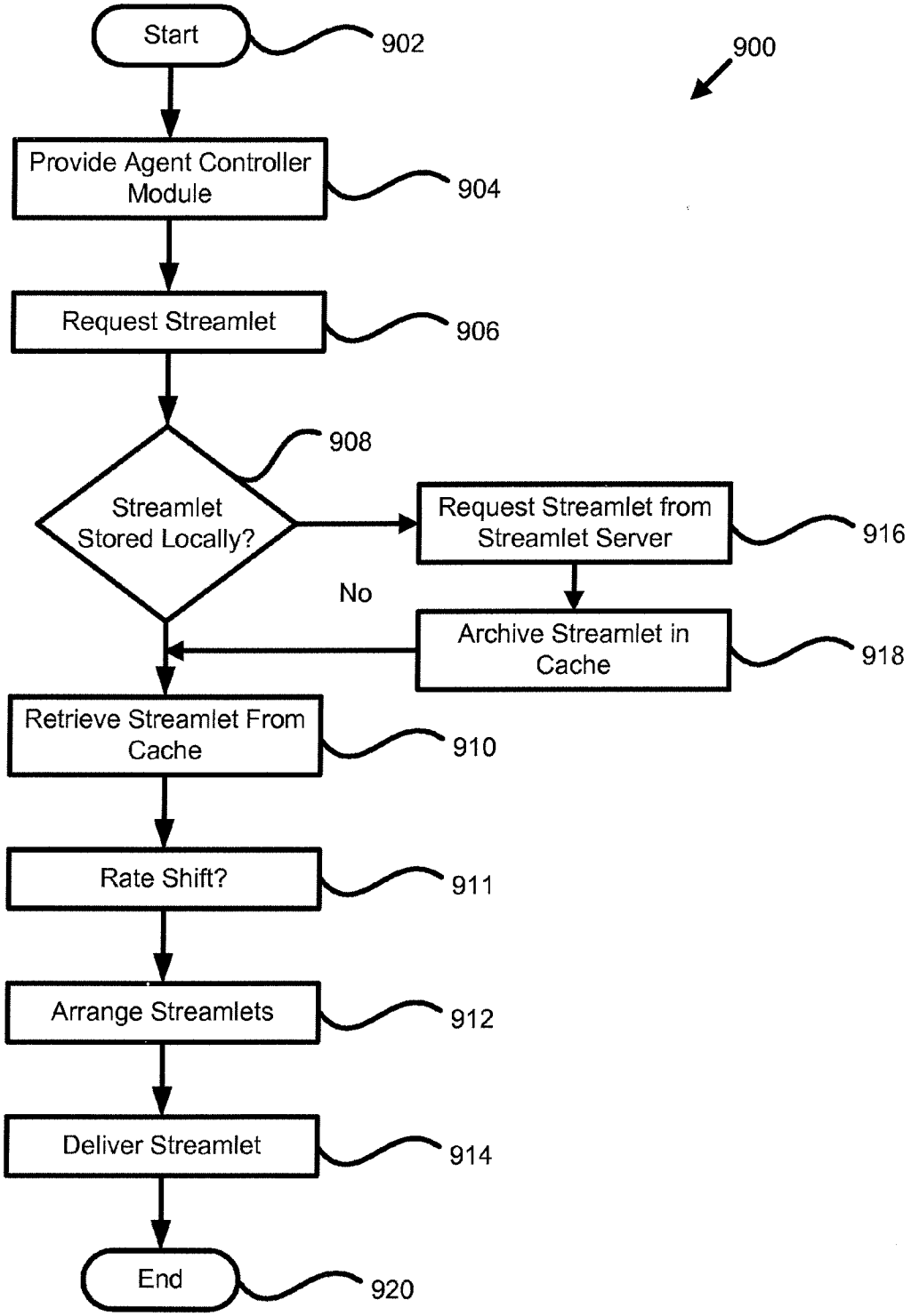


FIG. 9

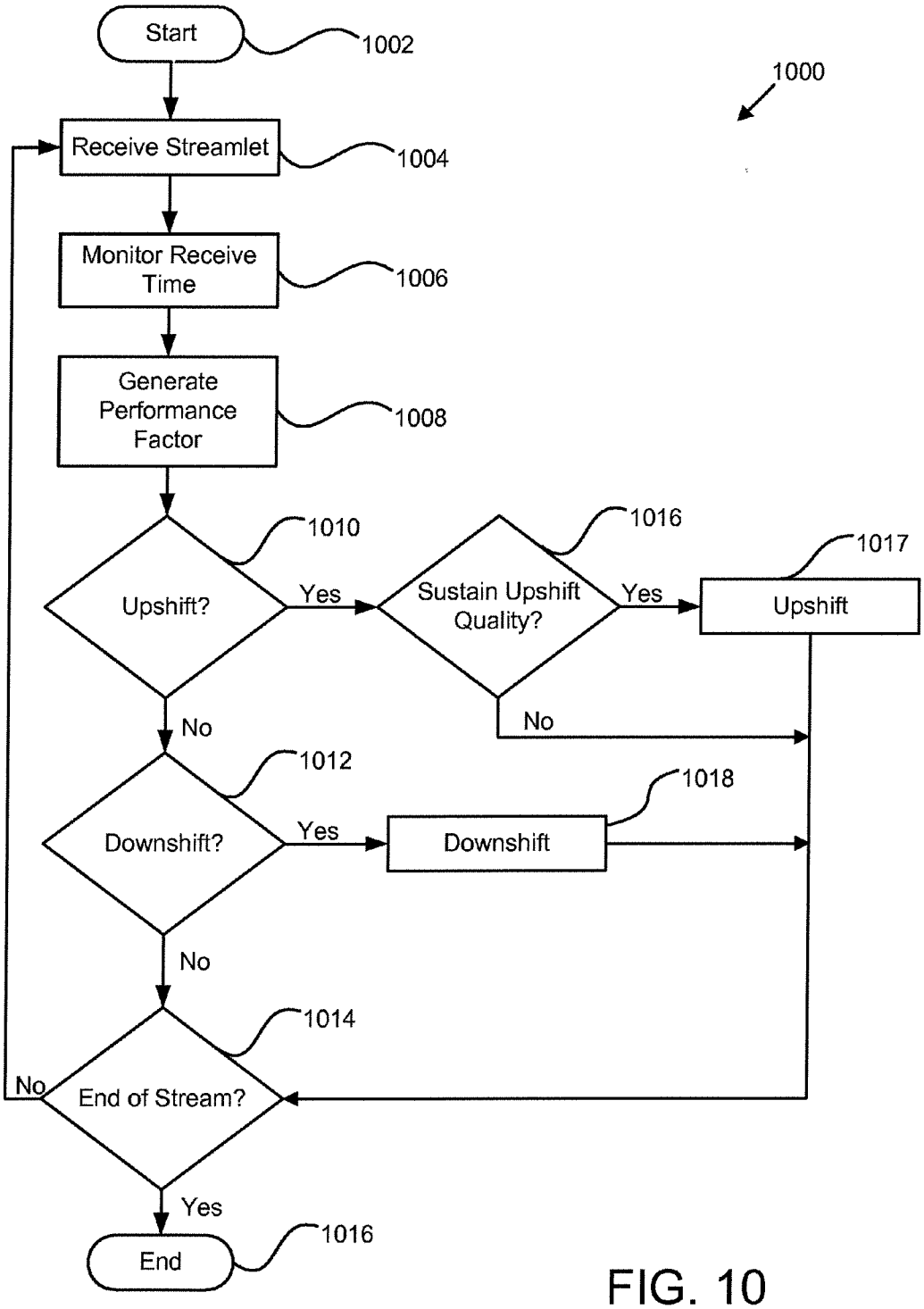


FIG. 10

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

CROSS-REFERENCES TO RELATED APPLICATIONS

This application is a continuation-in-part of application Ser. No. 11/116,783, filed on Apr. 28, 2005, which claims the benefit of U.S. Provisional Application No. 60/566,831, filed on Apr. 30, 2004, both of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

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.

2. Description of the Related Art

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.

"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 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.

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.

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.

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.

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.

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.

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.

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

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.

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.

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 0.1 and 5 seconds.

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.

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

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.

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.

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.

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.

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.

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.

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

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:

FIG. **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;

FIG. **2***a* is a schematic block diagram graphically illustrating one embodiment of a media content file;

FIG. **2***b* is a schematic block diagram illustrating one embodiment of a plurality of streams having varying degrees of quality and bandwidth;

FIG. **3***a* is a schematic block diagram illustrating one embodiment of a stream divided into a plurality of source streamlets;

FIG. **3***b* is a schematic block diagram illustrating one embodiment of sets of streamlets in accordance with the present invention;

FIG. **4** is a schematic block diagram illustrating in greater detail one embodiment of the content module in accordance with the present invention;

FIG. **5***a* is a schematic block diagram illustrating one embodiment of an encoder module in accordance with the present invention;

FIG. **5***b* is a schematic block diagram illustrating one embodiment of parallel encoding of streamlets in accordance with the present invention;

FIG. **6***a* is a schematic block diagram illustrating one embodiment of a virtual timeline in accordance with the present invention;

FIG. **6***b* is a schematic block diagram illustrating an alternative embodiment of a VT in accordance with the present invention;

FIG. **6***c* is a schematic block diagram illustrating one embodiment of a QMX in accordance with the present invention;

FIG. **7** is a schematic block diagram graphically illustrating one embodiment of a client module in accordance with the present invention;

FIG. **8** is a schematic flow chart diagram illustrating one embodiment of a method for processing content in accordance with the present invention;

FIG. **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

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

DETAILED DESCRIPTION OF THE INVENTION

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.

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.

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.

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.

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.

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.

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 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.

In the depicted embodiment, the system **100** also includes a publisher **110**, and a web server **116**. The publisher **110** may be a creator or distributor of content. For example, if the content to be streamed were a broadcast of a television program, the publisher **110** may be a television or cable network channel such as NBC®, or MTV®. Content may be transferred over the Internet **106** to the content server **102**, where the content is received by a content module **112**. The content module **112** may be configured to receive, process, and store content. In one embodiment, processed content is accessed by a client module **114** configured to play the content on the end user station **104**. In a further embodiment, the client module **114** is configured to receive different portions of a content stream from a plurality of locations simultaneously. For example, the client module **114** may request and receive content from any of the plurality of web servers **116**.

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**.

FIG. **2***a* 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.

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.

FIG. **3***a* 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.

FIG. **3***b* 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 **306***a* encompasses all streamlets having a time index of 00:00. The set **306***a* 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**.

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

FIG. **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.

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**.

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.

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, 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.

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 FIG. **3***b*, may comprise streamlets having an identical time index and duration, and a unique bitrate. As with FIG. **3***b*, 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.

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.

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.

Once the streamlets **304** have been received and processed, the client module **114** may request streamlets **304** using HTTP from the web server **116**. Using a standard protocol such as HTTP eliminates the need for network administrators to configure firewalls to recognize and pass through network traffic for a new, specialized protocol. Additionally, since the client module **114** initiates the request, the web server **116** is only required to retrieve and serve the requested streamlet **304**. In a further embodiment, the client module **114** may be configured to retrieve streamlets **304** from a plurality of web servers **116**.

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**

FIG. **5***a* 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.

The master module (hereinafter "master") **502** is configured to receive streamlets **303** from the streamlet module **404** and stage the streamlet **303** for processing. In one embodiment, the master **502** may decompress each source streamlet **303** to produce a raw streamlet. As used herein, the term "raw streamlet" refers to a streamlet **303** that is uncompressed or lightly compressed to substantially reduce size with no significant loss in quality. A lightly compressed raw streamlet can be transmitted more quickly and to more hosts. Each host **504** is coupled with the master **502** and configured to receive a raw streamlet from the master **502** for encoding. The hosts **504**, in one example, generate a plurality of streamlets **304** having identical time indices and durations, and varying bitrates. Essentially each host **504** may be configured to generate a set **306** from the raw streamlet **503** sent from the master **502**. Alternatively, each host **504** may be dedicated to producing a single bitrate in order to reduce the time required for encoding.

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.

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.

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.

The present invention, however, segments the content file **200** into source streamlets **303** and the two-pass encoding to a plurality of streams **202** may be performed on each corresponding raw streamlet without waiting for a TV show to end, for example. As such, the content module **112** is capable of streaming the streamlets over the Internet shortly after the content module **112** begins capture of the content file **200**. The delay between a live broadcast transmitted from the publisher **110** and the availability of the content depends on the computing power of the hosts **504**.

FIG. **5***b* 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 FIG. **4**) begins to capture the content file and the streamlet module **404** generates a first streamlet **303***a* and passes the streamlet to the encoding module **406**. The encoding module **406** may take 10 seconds, for example, to generate the first set **306***a* of streamlets **304***a* (**304**a_1, **304**a_2, **304**a_3, etc. represent streamlets **304** of different bitrates). FIG. **5***b* 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**.

During the 10 seconds required to encode the first streamlet **303***a*, the streamlet module **404** has generated five additional 2-second streamlets **303***b*, **303***c*, **303***d*, **303***e*, **303***f*, for encoding and the master **502** has prepared and staged the corresponding raw streamlets. Two seconds after the first set **306***a* is available the next set **306***b* 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.

A system as described above beneficially enables multi-pass encoding of live events. Multi-pass encoding systems of the prior art require that the entire content be captured (or be complete) because in order to perform multi-pass encoding the entire content must be scanned and processed more than once. This is impossible with prior art systems because content from a live event is not complete until the event is over. As such, with prior art systems, multi-pass encoding can only be performed once the event is over. Streamlets, however, may be encoded as many times as is deemed necessary. Because the streamlet is an encapsulated media object of 2 seconds (for example), multi-pass encoding may begin on a live event once the first streamlet is captured. Shortly after multi-pass encoding of the first streamlet **303***a* is finished, multi-pass encoding of the second streamlet **303***b* finishes, and as such multi-pass encoding is performed on a live event and appears live to a viewer.

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.

The parallel offset pipeline approach described with reference to FIG. **5***b* 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.

Returning now to FIG. **5***a*, as depicted, the master **502** and the hosts **504** may be located within a single local area network, or in other terms, the hosts **504** may be in close physical proximity to the master **502**. Alternatively, the hosts **504** may receive encoding jobs from the master **502** over the Internet or other communications network. For example, consider a live sports event in a remote location where it would be difficult to setup multiple hosts. In this example, a master performs no encoding or alternatively light encoding before publishing the streamlets online. The hosts **504** would then retrieve those streamlets and encode the streamlets into the multiple bitrate sets **306** as described above.

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.

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.

FIG. **6***a* 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 **602***a* followed by QMX **602***b* and QMX **602***c*. As such, the publisher may define a broadcast schedule in a manner similar to a television station.

FIG. **6***b* 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.

FIG. **6***c* is a schematic block diagram illustrating one embodiment of a QMX **602** in accordance with the present invention. In one embodiment, the QMX **602** contains a multitude of information generated by the content module **112** configured to describe the content file **200**. Examples of information include, but are not limited to, start index **604**, end index **606**, whether the content is live **608**, proprietary publisher data **610**, encryption level **612**, content duration **614** and bitrate values **616**. The bitrate values **616** may include frame size **618**, audio channel **620** information, codecs **622** used, sample rate **624**, and frames parser **626**.

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.

FIG. **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.

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.

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 FIG. **10**.

The agent controller module **702** may be configured to receive user commands from the viewer **708**. Such commands may include play, fast forward, rewind, pause, and stop. In one embodiment, the agent controller module **702** requests streamlets **304** from the streamlet cache module **704** and arranges the received streamlets **304** in a staging module **709**. The staging module **709** may be configured to arrange the streamlets **304** in order of ascending playback time. In the depicted embodiment, the streamlets **304** are numbered 0, 1, 2, 3, 4, etc. However, each streamlet **304** may be identified with a unique filename.

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 fur-

ther 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**.

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

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.

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**.

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.

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.

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.

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.

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.

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.

FIG. **8** is a schematic flow chart diagram illustrating one embodiment of a method **800** for processing content in accordance with the present invention. In one embodiment the method **800** starts **802**, and the content module **112** receives **804** content from the publisher **110**. Receiving content **804** may comprise receiving **804** a digital copy of the content file **200**, or digitizing a physical copy of the content file **200**. Alternatively, receiving **804** content may comprise capturing a radio, television, cable, or satellite broadcast. Once received **804**, the streamlet module **404** generates **808** a plurality of source streamlets **303** each having a fixed duration. Alternatively, the streamlets **303** may be generated with a fixed file size.

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

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**.

FIG. **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 control controller module **702** is provided **904** and associated with a viewer **708** and provided with a staging module **709**. The agent controller module **702** then requests **906** a streamlet **304** from the streamlet cache module **704**. Alternatively, the agent controller module **702** may simultaneously request **906** a plurality of streamlets **304** the streamlet cache module **704**. If the streamlet is stored **908** locally in the streamlet cache **710**, the streamlet cache module **704** retrieves **910** the streamlet **304** and sends the streamlet to the agent controller module **702**. Upon retrieving **910** or receiving a streamlet, the agent controller module **702** makes **911** a determination of whether or not to shift to a higher or lower quality stream **202**. This determination will be described below in greater detail with reference to FIG. **10**.

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.

Referring now to FIG. **10**, shown therein is a schematic flow chart diagram illustrating one embodiment of a method **1000** for requesting streamlets **304** within a 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 FIG. **9**. The method **1000** starts and the agent controller module **702** receives **1004** a streamlet **304** as described above with reference to FIG. **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 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.

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

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

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 per

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

The policy determination about whether or not to upshift **1010** 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 **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 $\phi_{higher} \geqq \Theta_{up}$ then use of the higher quality stream is considered sustainable. If the decision of whether or not the higher stream rate is sustainable **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**.

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 $\phi_{current} > \Theta_{down}$ then the stream quality may be adequate, and the agent controller module **702** does not downshift **1018** stream quality. However, if $\phi_{current} \leqq \Theta_{down}$, 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.

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 multi-bitrate video and/or audio content streaming, the apparatus comprising:

one or more computing devices including,

a receiving module configured to receive videos;

a streamlet module configured to segment the media content received videos to generate for each a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video;

an encoding module configured to encode each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the multi-bitrate content streaming, wherein each of the encoded streamlets in each of the sets is stored as a separate content file, wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the video, wherein the separate content files within each of the sets are independently requestable by end user stations, and wherein shifts between the different bit rates are made at the time indexes during streaming of each of the videos; and

a web server to be coupled over the Internet to the end user stations to receive requests from the end user stations for different ones of the separate content files from different ones of the sets and to transmit the requested ones of the separate content files to the requesting one of the end user stations, wherein each of the end user stations initiate each of the shifts between the different bit rates during streaming of each of the videos through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

2. The apparatus of claim **1**, wherein the encoding module is configured to encode multiple of the plurality of sequential raw streamlets in parallel.

3. The apparatus of claim **2**, wherein the encoding module is further configured to encode the multiple streamlets in a time-selective manner so that the plurality of sequential raw streamlets are processed successively at a regular time interval.

4. The apparatus of claim **3**, wherein the one or more computing devices further comprise a streamlet database, wherein the encoding module is further configured to transmit the separate content files each storing one of the encoded streamlets to the streamlet database at the regular time interval to be stored, and wherein the web server is coupled to the streamlet database to retrieve the requested ones of the separate content files from the streamlet database to transmit to the requesting one of the end user stations.

5. The apparatus of claim **1**, wherein the portions of each of the plurality of sequential raw streamlets has a predetermined length of time in the range of between about 0.1 and 5 seconds.

6. The apparatus of claim **1**, wherein the portions of each of the plurality of sequential raw streamlets has a predetermined length of time of the video in the range of between about 1 and 3 seconds.

7. The apparatus of claim **1**, wherein the portions of each of the plurality of sequential raw streamlets has a predetermined length of time in the range of between about 1.5 and 2.5 seconds.

8. The apparatus of claim **1**, wherein the receiving module is configured to receive the videos including encoded video and encoded audio and convert the encoded video to raw video and the encoded audio to raw audio.

9. The apparatus of claim **1**, wherein the encoding module comprises a master module configured to assign an encoding job to one of a plurality of hosts to generate at least one of the separate content files from at least one of the plurality of sequential raw streamlets in response to an encoding job completion bid received from the one host.

10. The apparatus of claim **9**, wherein each said host is configured to submit an encoding job completion bid based on a plurality of computing variables.

11. The apparatus of claim **10**, wherein the plurality of computing variables is selected from a group consisting of current encoding job completion percentage, average encoding job completion time, processor speed, and physical memory capacity.

12. The apparatus of claim **1**, wherein the encoding module is configured to encode the plurality of sequential raw streamlets with an offset parallel processing scheme such that the sets are published at regular time intervals to match the real-time submission rate at which each of the plurality of sequential raw streamlets are received at the encoding module, the time intervals being less than the time of encoding one of the sets and sufficient to play the video according to the real-time submission rate on at least one of the end user stations.

13. The apparatus of claim **1**, wherein the encoding module is configured to encode the plurality of sequential raw streamlets with an offset parallel processing scheme such that the sets are published at regular time intervals to match the real-time submission rate at which each of the plurality of sequential raw streamlets are received at the encoding module, the time intervals being greater than the time of encoding one of

sets and sufficient to play the video according to the real-time submission rate on at least one of the end user stations.

14. A system for multi-bitrate video and/or audio content streaming, the system comprising:

end user stations coupled to the Internet; and

one or more computing devices including,

a receiving module configured to receive videos;

a streamlet module configured to segment the received videos to generate for each a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video; content having a predetermined length of time;

an encoding module configured to encode each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the multi-bitrate content streaming, wherein each of the encoded streamlets in each of the sets is stored as a separate content file that individually stores data to playback only the same video portion as the raw streamlet from which it was generated;

wherein the separate content files within each of the sets are independently requestable by the end user stations; and

a web server to be coupled over the Internet to the end user stations to receive requests from the end user stations for different ones of the separate content files from different ones of the sets and to transmit the requested ones of the separate content files to the requesting one of the end user stations, wherein shifts between the different bit rates are made at the time indexes during streaming of each of the videos wherein each of the end user stations initiate each of the shifts between the different bit rates during streaming of each of the videos through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

15. The system of claim **14**, wherein the encoding module is configured to encode multiple of the plurality of sequential raw streamlets in parallel.

16. The system of claim **15**, wherein the encoding module is further configured to encode the multiple streamlets in a time-selective manner so that the plurality of sequential raw streamlets are processed successively at a regular time interval.

17. The system of claim **16**, wherein:

the encoding module comprises a master module configured to assign an encoding job to one of a plurality of host computing modules to generate at least one of the separate content files from at least one of the plurality of sequential raw streamlets in response to an encoding job completion bid received from the one host, and

the one or more computing devices further comprise a streamlet database, and wherein the encoding module is further configured to transmit the separate content files each storing one of the encoded streamlets to the streamlet database at the regular time interval to be stored, and wherein the web server is coupled to the streamlet database to retrieve the requested ones of the separate content files from the streamlet database to transmit to the requesting one of the end user stations.

18. The system of claim **14**, wherein the predetermined length of time is in the range of between about 0.1 and 5 seconds.

19. The system of claim **14**, wherein the predetermined length of time is in the range of between about 1 and 3 seconds.

20. The system of claim **14**, wherein the predetermined length of time is in the range of between about 1.5 and 2.5 seconds.

21. The system of claim **14**, wherein the receiving module is configured to receive the videos including encoded video and encoded audio and convert the encoded video to raw video and the encoded audio to raw audio.

22. The system of claim **17**, wherein each said host computing module is configured to submit an encoding job completion bid based on a plurality of computing variables.

23. The system of claim **22**, wherein the plurality of computing variables is selected from a group consisting of current encoding job completion percentage, average encoding job completion time, processor speed, and physical memory capacity.

24. A method for multi-bitrate video and/or audio content streaming, the method comprising:

capturing a single video;

segmenting the single video and generating a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video;

encoding each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the multi-bitrate content streaming, wherein each of the encoded streamlets in each of the sets is stored as a separate content file, wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the single video, wherein the separate content files within each of the sets are independently requestable by end user stations, and wherein shifts between the different bit rates are made at the time indexes during streaming of the single video;

receiving requests from the end user stations over the Internet for different ones of the separate content files from different ones of the sets; and

transmitting the requested ones of the separate content files to the requesting one of the end user stations, wherein each of the end user stations initiate each of the shifts between the different bit rates during streaming of the single video through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

25. The method of claim **24**, wherein said encoding comprises encoding multiple of the plurality of sequential raw streamlets in parallel.

26. The method of claim **25**, wherein said encoding the multiple streamlets comprises encoding the multiple streamlets in a time-selective manner so that the plurality of sequential raw streamlets are processed successively at a regular time interval.

27. The method of claim **25**, further comprising:

transmitting the separate content files each storing one of the encoded streamlets to a streamlet database at the regular time interval; and

storing the separate content files each storing one of the encoded streamlets in the streamlet database, wherein

said transmitting comprises retrieving the requested ones of the separate content files from the streamlet database.

28. The method of claim **24**, wherein said receiving comprises receiving the single video including encoded video and encoded audio, and the method further comprises converting the encoded video to raw video and the encoded audio to raw audio.

29. The method of claim **24**, wherein said encoding comprises assigning an encoding job to one of a plurality of host computing modules to generate at least one of the separate content files from at least one of the plurality of sequential raw streamlets in response to an encoding job completion bid received from the one host.

30. The method of claim **29**, wherein said encoding comprises receiving the encoding job completion bid based on a plurality of computing variables.

31. A non-transitory computer readable storage medium storing a computer readable program of computer instructions, wherein the computer readable program when executed on a computer causes the computer to carry out operations for multi-bitrate video and/or audio content streaming, the operations comprising:

capturing a single video;

segmenting the single video and generating a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video;

encoding each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the multi-bitrate content streaming, wherein each of the encoded streamlets in each of the sets is stored as a separate content file, wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the single video, wherein the separate content files within each of the sets are independently requestable by end user stations, and wherein shifts between the different bit rates are made at the time indexes during streaming of the single video;

receiving requests from the end user stations over the Internet for different ones of the separate content files from different ones of the sets; and

transmitting the requested ones of the separate content files to the requesting one of the end user stations, wherein each of the end user stations initiate each of the shifts between the different bit rates during streaming of the single video through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

32. The computer readable storage medium of claim **31**, wherein said encoding comprises encoding multiple of the plurality of sequential raw streamlets in parallel.

33. The computer readable storage medium of claim **32**, wherein said encoding the multiple streamlets comprises encoding the multiple streamlets in a time-selective manner so that the plurality of sequential raw streamlets are processed successively at a regular time interval.

34. The computer readable storage medium of claim **31**, wherein the operations further comprise:

transmitting the separate content files each storing one of the encoded streamlets to a streamlet database at the regular time interval; and

storing the separate content files each storing one of the encoded streamlets in the streamlet database, wherein said transmitting comprises retrieving the requested ones of the separate content files from the streamlet database.

35. An apparatus for multi-bitrate video and/or audio content streaming, the apparatus comprising

one or more computing devices including,

means for capturing videos;

means for segmenting the single video and generating a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video; and

means for encoding each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the multi-bitrate content streaming, wherein each of the encoded streamlets in each of the sets is stored as a separate content file, wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the single video, wherein the separate content files within each of the sets are independently requestable by end user stations, and wherein shifts between the different bit rates are made at the time indexes during streaming of the single video;

means for receiving requests from the end user stations over the Internet for different ones of the separate content files from different ones of the sets; and

means for transmitting the requested ones of the separate content files to the requesting one of the end user stations, wherein each of the end user stations initiate each of the shifts between the different bit rates during streaming of each of the videos through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

36. The apparatus of claim **1**, wherein the encoding module is further configured to start the generation of a second one of the sets from a second of the plurality of sequential raw streamlets prior to completing the generation of a first of the sets from a first of the plurality of sequential raw streamlets to generate the separate content files of the first and second sets in parallel, and wherein the second sequential raw streamlet sequentially follows the first sequential raw streamlet.

37. The apparatus of claim **1**, wherein the encoding module is further configured to multi-pass encode each of the plurality of sequential raw streamlets of at least one of the videos, and wherein the at least one of the videos is of a live event.

38. The apparatus of claim **1**, wherein least one of the videos is of a live event, wherein the encoding module is configured to receive the plurality of sequential raw streamlets of the at least one video according to a real-time submission rate, and wherein the encoding module is configured to encode the plurality of sequential raw streamlets at regular intervals using an offset parallel processing scheme, wherein the regular intervals match the intervals at which each of the plurality of sequential raw streamlets is received at the encoding module so the output timing of the encoding module does not fall behind the real-time submission rate.

39. The apparatus of claim **1**, wherein the one or more computing devices further comprising a streamlet database,

and wherein the encoding module is configured to transmit the separate content files each storing one of the encoded streamlets to the streamlet database to be stored, and wherein the web server is coupled to the streamlet database to retrieve the requested ones of the separate content files from the streamlet database to transmit to the requesting one of the end user stations.

40. The apparatus of claim **9**, wherein at least one of the plurality of hosts is configured to generate the separate content files within one of the sets.

41. The apparatus of claim **9**, wherein one of the plurality of host is configured to generate a first of the separate content files according to a first one of the supported bitrates for one of the plurality of sequential raw streamlets, and wherein another one of the plurality of hosts is configured to generate a second of the separate content files according to a second one of the supported bitrates for the same one of the plurality of sequential raw streamlets.

42. The system of claim **14**, wherein the encoding module is further configured to start the generation of a second one of the sets from a second of the plurality of sequential raw streamlets prior to completing the generation of a first of the sets from a first of the plurality of sequential raw streamlets to generate the separate content files of the first and second sets in parallel, and wherein the second sequential raw streamlet sequentially follows the first sequential raw streamlet.

43. The system of claim **14**, wherein the encoding module is further configured to multi-pass encode each of the plurality of sequential raw streamlets of at least one of the videos, and wherein the at least one of the videos is of a live event.

44. The system of claim **14**, wherein at least one of the videos is of a live event, wherein the encoding module is configured to receive the plurality of sequential raw streamlets of the at least one video according to a real-time submission rate, and wherein the encoding module is configured to encode the plurality of sequential raw streamlets at regular intervals using an offset parallel processing scheme, wherein the regular intervals match the intervals at which each of the plurality of sequential raw streamlets is received at the encoding module so the output timing of the encoding module does not fall behind the real-time submission rate.

45. The system of claim **14**, comprising wherein:

the encoding module comprises a master module configured to assign an encoding job to one of a plurality of host computing modules to generate at least one of the separate content files from at least one of the plurality of sequential raw streamlets in response to an encoding job completion bid received from the one host, and

the one or more computing devices further comprise a streamlet database, and wherein the encoding module is further configured to transmit the separate content files each storing one of the encoded streamlets to the streamlet database to be stored, and wherein the web server is coupled to the streamlet database to retrieve the requested ones of the separate content files from the streamlet database to transmit to the requesting one of the end user stations.

46. The system of claim **45**, wherein at least one of the plurality of hosts is configured to generate the separate content files within one of the sets.

47. The system of claim **45**, wherein one of the plurality of host is configured to generate a first of the separate content files according to a first one of the supported bitrates for one of the plurality of sequential raw streamlets, and wherein another one of the plurality of hosts is configured to generate a second of the separate content files according to a second one of the supported bitrates for the same one of the plurality of sequential raw streamlets.

48. The method of claim **24**, wherein said encoding further comprises starting the generation of a second one of the sets from a second of the plurality of sequential raw streamlets prior to completing the generation of a first of the sets from a first of the plurality of sequential raw streamlets to generate the separate content files of the first and second sets in parallel, and wherein the second sequential raw streamlet sequentially follows the first sequential raw streamlet.

49. The method of claim **24**, wherein said encoding further comprises multi-pass encoding each of the plurality of sequential raw streamlets of the video, and wherein the video is of a live event.

50. The method of claim **24**, wherein the video is of a live event, wherein said capturing comprises capturing the video of the live event according to a real-time submission rate, and wherein said encoding comprises encoding each of the plurality of sequential raw streamlets of the video at regular intervals using an offset parallel processing scheme, wherein the regular intervals match the intervals at which each of the plurality of sequential raw streamlets is generated so the output timing of said encoding does not fall behind the real-time submission rate.

51. The method of claim **24**, further comprising:

transmitting the separate content files each storing one of the encoded streamlets to a streamlet database; and

storing the separate content files each storing one the encoded streamlets in the streamlet database, wherein said transmitting comprises retrieving the requested ones of the separate content files from the streamlet database.

52. The method of claim **29**, wherein said assigning the encoding job comprises assigning at least one of the plurality of hosts to generate the separate content files within one of the sets.

53. The method of claim **29**, wherein said assigning the encoding job comprises assigning one of the plurality of host to generate a first of the separate content files according to a first one of the supported bitrates for one of the plurality of sequential raw streamlets, and wherein the method further comprises assigning a second encoding job to another one of the plurality of hosts to generate a second of the separate content files according to a second one of the supported bitrates for the same one of the plurality of sequential raw streamlets.

54. The apparatus of claim **35**, wherein at least one of the videos is of a live event, wherein said means for encoding receives the at least one video of the live event according to a real-time submission rate, and wherein said means for encoding encodes each of the plurality of sequential raw streamlets of the at least one video at regular intervals using an offset parallel processing scheme, wherein the regular intervals match the intervals at which each of the plurality of sequential raw streamlets is received at said encoding means so the output timing of said encoding means does not fall behind the real-time submission rate.

55. The apparatus of claim **35**, wherein the means for encoding further comprises means for multi-pass encoding each of the plurality of sequential raw streamlets of at least one of the videos, and wherein the at least one video is of a live event.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,818,444 B2
APPLICATION NO. : 11/673483
DATED : October 19, 2010
INVENTOR(S) : Brueck et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 22, line 55 claim 38, add the word --at-- after the word wherein and before the word least.

Signed and Sealed this
Twenty-sixth Day of April, 2011

David J. Kappos

David J. Kappos
Director of the United States Patent and Trademark Office

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,818,444 B2
APPLICATION NO. : 11/673483
DATED : October 19, 2010
INVENTOR(S) : Brueck et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

At Item (75), Inventors:
At Line number 3, after (US) add the name:
; R. Drew Major, Orem, UT (US)

Signed and Sealed this
Sixteenth Day of January, 2018

Joseph Matal

Joseph Matal
Performing the Functions and Duties of the
Under Secretary of Commerce for Intellectual Property and
Director of the United States Patent and Trademark Office

EXHIBIT B

U 7701579

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

November 21, 2018

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF:

U.S. PATENT: *8,402,156*
ISSUE DATE: *March 19, 2013*

By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office

SYLVIA HOLLEY
Certifying Officer

US008402156B2

(12) **United States Patent**
Brueck et al.

(10) **Patent No.: US 8,402,156 B2**
(45) **Date of Patent: *Mar. 19, 2013**

(54) **APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING**

(75) Inventors: **David F. Brueck**, Saratoga Springs, UT (US); **Mark B. Hurst**, Cedar Hills, UT (US)

(73) Assignee: **DISH Digital L.L.C.**, Englewood, CO (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 106 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/906,940**

(22) Filed: **Oct. 18, 2010**

(65) **Prior Publication Data**

US 2011/0035507 A1 Feb. 10, 2011

Related U.S. Application Data

(63) Continuation of application No. 11/673,483, filed on Feb. 9, 2007, now Pat. No. 7,818,444, which is a continuation-in-part of application No. 11/116,783, filed on Apr. 28, 2005.

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

(51) **Int. Cl.**
G06F 15/16 (2006.01)

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

(58) **Field of Classification Search** 709/231–233, 709/247–248, 208–211; 375/240.01–240.12; 718/102

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,535,355	A	8/1985	Arn et al.
5,168,356	A	12/1992	Acampora et al.
5,768,527	A	6/1998	Zhu et al.
5,953,506	A	9/1999	Kalra et al.
6,091,775	A	7/2000	Hibi et al.
6,091,777	A	7/2000	Guetz et al.
6,122,660	A	9/2000	Baransky et al.
6,185,736	B1	2/2001	Ueno
6,195,680	B1	2/2001	Goldszmidt et al.
6,366,614	B1	4/2002	Pian et al.
6,374,289	B2	4/2002	Delaney et al.
6,389,473	B1	5/2002	Carmel et al.

(Continued)

FOREIGN PATENT DOCUMENTS

CA	2466482	A1	5/2003
EP	0 711 077	A2	5/1996

(Continued)

OTHER PUBLICATIONS

Japanese Patent Office, "Final Office Action" mailed Feb. 28, 2012; Japanese Appln. No. 2007-511070.

(Continued)

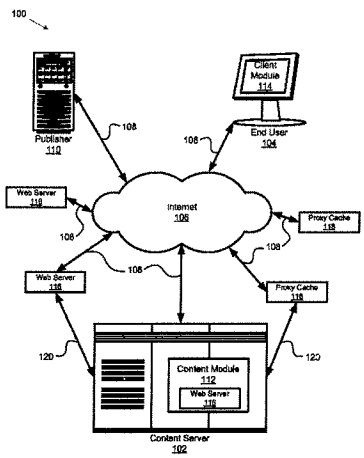
Primary Examiner — Chirag R Patel

(74) *Attorney, Agent, or Firm* — Ingrassia Fisher & Lorenz, P.C.

(57) **ABSTRACT**

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.

23 Claims, 11 Drawing Sheets



U.S. PATENT DOCUMENTS

6,449,719 B1 9/2002 Baker
6,486,803 B1 11/2002 Luby et al.
6,490,627 B1 12/2002 Kalra et al.
6,510,553 B1 1/2003 Hazra
6,574,591 B1 6/2003 Kleiman et al.
6,604,118 B2 8/2003 Kleiman et al.
6,618,752 B1 9/2003 Moore et al.
6,721,723 B1 4/2004 Gibson et al.
6,731,600 B1 5/2004 Patel et al.
6,732,183 B1 5/2004 Graham
6,760,772 B2 7/2004 Zou et al.
6,795,863 B1 9/2004 Doty, Jr.
6,845,107 B1 1/2005 Kitazawa et al.
6,850,965 B2 2/2005 Allen
6,859,839 B1 2/2005 Zahorjan et al.
6,874,015 B2 3/2005 Kaminsky et al.
6,968,387 B2 11/2005 Lanphear
6,976,090 B2 12/2005 Ben-Shaul et al.
7,054,365 B2 5/2006 Kim et al.
7,054,774 B2 5/2006 Batterberry et al.
7,054,911 B1 5/2006 Lango et al.
7,075,986 B2 7/2006 Girod et al.
7,093,001 B2 8/2006 Yang et al.
7,096,271 B1 8/2006 Omoigui et al.
7,099,954 B2 8/2006 Li et al.
7,116,894 B1 10/2006 Chatterton
7,174,385 B2 2/2007 Li
7,194,549 B1 3/2007 Lee et al.
7,240,100 B1 7/2007 Wein et al.
7,260,640 B1 8/2007 Kramer et al.
7,274,740 B2 9/2007 van Beek et al.
7,295,520 B2 11/2007 Lee et al.
7,310,678 B2 12/2007 Gunaseelan et al.
7,325,073 B2 1/2008 Shao et al.
7,334,044 B1 2/2008 Allen
7,349,358 B2 3/2008 Hennessey et al.
7,349,976 B1 3/2008 Glaser et al.
7,369,610 B2 * 5/2008 Xu et al. 375/240.08
7,376,747 B2 5/2008 Hartop
7,391,717 B2 6/2008 Klemets et al.
7,408,984 B2 * 8/2008 Lu et al. 375/240.02
7,412,531 B1 8/2008 Lango et al.
7,477,688 B1 1/2009 Zhang et al.
7,523,181 B2 4/2009 Swildens et al.
7,536,469 B2 5/2009 Chou et al.
7,546,355 B2 6/2009 Kalnitsky
7,577,750 B2 8/2009 Shen et al.
7,593,333 B2 9/2009 Li et al.
7,599,307 B2 10/2009 Seckni et al.
7,609,652 B2 10/2009 Kellerer et al.
7,719,985 B2 5/2010 Lee et al.
7,760,801 B2 7/2010 Ghanbari et al.
7,779,135 B2 8/2010 Hudson et al.
7,788,395 B2 8/2010 Bowra et al.
7,797,439 B2 9/2010 Cherkasova et al.
7,817,985 B2 10/2010 Moon
7,818,444 B2 10/2010 Brueck et al.
2001/0047423 A1 11/2001 Shao et al.
2002/0073167 A1 6/2002 Powell et al.
2002/0091840 A1 7/2002 Pulier et al.
2002/0097750 A1 7/2002 Gunaseelan et al.
2002/0131496 A1 * 9/2002 Vasudevan et al. 375/240.11
2002/0144276 A1 10/2002 Radford et al.
2002/0152317 A1 10/2002 Wang et al.
2002/0152318 A1 10/2002 Menon et al.
2002/0161898 A1 10/2002 Hartop et al.
2002/0161911 A1 10/2002 Pinckney, III et al.
2002/0169926 A1 11/2002 Pinckney, III et al.
2002/0174434 A1 11/2002 Lee et al.
2002/0176418 A1 11/2002 Hunt et al.
2002/0178330 A1 11/2002 Schlowsky-Fischer et al.
2002/0188745 A1 12/2002 Hughes et al.
2003/0005455 A1 1/2003 Bowers
2003/0014684 A1 1/2003 Kashyap
2003/0018966 A1 1/2003 Cook et al.
2003/0021166 A1 1/2003 Soloff
2003/0065803 A1 4/2003 Heuvelman
2003/0067872 A1 4/2003 Harrell et al.
2003/0081582 A1 5/2003 Jain et al.
2003/0107994 A1 6/2003 Jacobs et al.
2003/0135631 A1 7/2003 Li et al.
2003/0140159 A1 7/2003 Campbell et al.
2003/0151753 A1 8/2003 Li et al.
2003/0152036 A1 8/2003 Quigg Brown et al.
2003/0154239 A1 8/2003 Davis et al.
2003/0204519 A1 10/2003 Sirivara et al.
2004/0003101 A1 1/2004 Roth et al.
2004/0030547 A1 2/2004 Leaning et al.
2004/0030599 A1 2/2004 Sie et al.
2004/0030797 A1 2/2004 Akinlar et al.
2004/0031054 A1 2/2004 Dankworth et al.
2004/0049780 A1 3/2004 Gee
2004/0054551 A1 3/2004 Ausubel et al.
2004/0071209 A1 4/2004 Burg et al.
2004/0083283 A1 4/2004 Sundaram et al.
2004/0093420 A1 5/2004 Gamble
2004/0103444 A1 5/2004 Weinberg et al.
2004/0117427 A1 6/2004 Allen et al.
2004/0143672 A1 7/2004 Padmanabham et al.
2004/0168052 A1 8/2004 Clisham et al.
2004/0170392 A1 9/2004 Lu et al.
2004/0220926 A1 11/2004 Lamkin et al.
2004/0260701 A1 12/2004 Lehikoinen et al.
2005/0015509 A1 * 1/2005 Sitaraman 709/231
2005/0055425 A1 * 3/2005 Lango et al. 709/219
2005/0066063 A1 3/2005 Grigorovitch et al.
2005/0076136 A1 4/2005 Cho et al.
2005/0084166 A1 4/2005 Bonch et al.
2005/0108414 A1 5/2005 Taylor et al.
2005/0120107 A1 6/2005 Kagan et al.
2005/0123058 A1 6/2005 Greenbaum et al.
2005/0185578 A1 8/2005 Padmanabhan et al.
2005/0188051 A1 8/2005 Sneh
2005/0204046 A1 9/2005 Watanabe
2005/0262257 A1 11/2005 Major et al.
2006/0059223 A1 3/2006 Klemets et al.
2006/0080718 A1 4/2006 Gray et al.
2006/0130118 A1 6/2006 Damm
2006/0133809 A1 6/2006 Chow et al.
2006/0165166 A1 7/2006 Chou et al.
2006/0168290 A1 7/2006 Doron
2006/0168295 A1 7/2006 Batterberry et al.
2006/0184688 A1 8/2006 Ganguly et al.
2006/0206246 A1 9/2006 Walker
2006/0236219 A1 10/2006 Grigorovitch et al.
2007/0024705 A1 2/2007 Richter et al.
2007/0030833 A1 2/2007 Pirzada et al.
2007/0067480 A1 3/2007 Beek et al.
2007/0079325 A1 4/2007 de Heer
2007/0094405 A1 4/2007 Zhang
2007/0204310 A1 8/2007 Hua et al.
2007/0280255 A1 12/2007 Tsang et al.
2008/0028428 A1 1/2008 Jeong et al.
2008/0037527 A1 2/2008 Chan et al.
2008/0046939 A1 2/2008 Lu et al.
2008/0056373 A1 3/2008 Newlin et al.
2008/0133766 A1 6/2008 Luo
2008/0162713 A1 7/2008 Bowra et al.
2008/0195744 A1 8/2008 Bowra et al.
2008/0195745 A1 8/2008 Bowra et al.
2008/0205291 A1 8/2008 Li et al.
2008/0219151 A1 9/2008 Ma et al.
2008/0222235 A1 9/2008 Hurst et al.
2008/0263180 A1 10/2008 Hurst et al.
2008/0281803 A1 11/2008 Gentric
2009/0043906 A1 2/2009 Hurst et al.
2009/0055471 A1 2/2009 Kozat et al.
2009/0055547 A1 2/2009 Hudson et al.
2009/0210549 A1 8/2009 Hudson et al.
2010/0098103 A1 4/2010 Xiong et al.

FOREIGN PATENT DOCUMENTS

EP 0 919 952 A1 6/1999
EP 1641271 A2 3/2006
EP 1 670 256 A2 6/2006
EP 1 777 969 4/2007
JP 2000-201343 7/2000

JP	200192752		4/2001
JP	2004054930		2/2004
JP	2011004225	A	1/2011
WO	WO-0067469		11/2000
WO	0167264	A1	9/2001
WO	03003760	A2	1/2003
WO	03009581	A1	1/2003
WO	03027876	A1	4/2003
WO	2004025405	A2	3/2004
WO	2004036824	A1	4/2004
WO	2006010113	A2	1/2006

OTHER PUBLICATIONS

"The meaning of performance factor—English-Japanese Weblio Dictionary", [online], Feb. 24, 2012, [searched on Feb. 24, 2012], the Internet <URL:http://ejje.weblio.jp/content/performance+factor>.
Tsuru, et al., "Recent evolution of the Internet measurement and inference techniques", IEICE Technical Report, vol. 103, No. 123, pp. 37-42, Jun. 12, 2003.
PCT Notification of Transmittal of the International Search Report and Written Opinion of the International Searching Authority, for PCT/US05/15091, Oct. 29, 2007, 8 pages.
PCT Notification of Transmittal of the International Preliminary Report on Patentability, for PCT/US05/15091, Oct. 29, 2007, 8 pages.
Advisory Action for U.S. Appl. No. 11/116,783, Mailed Mar. 23, 2010, 4 pages.
Advisory Action for U.S. Appl. No. 11/116,783, Mailed May 17, 2010, 3 pages.
Final Office Action for U.S. Appl. No. 11/116,783, Mailed Feb. 22, 2010, 19 pages.
Office Action for U.S. Appl. No. 11/116,783, Mailed May 14, 2008, 19 pages.
Final Office Action for U.S. Appl. No. 11/116,783, Mailed Feb. 20, 2009, 16 pages.
Supplemental European Search Report, Sep. 30, 2008, (3 pages).
Office Action for U.S. Appl. No. 11/116,783, Aug. 20, 2009, 14 pages.
Advisory Action for U.S. Appl. No. 11/116,783, Mailed Apr. 2, 2009, 3 pages.
Advisory Action for U.S. Appl. No. 11/116,783, Mailed May 12, 2009, 4 pages.
Albanese, Andres , et al., "Priority Encoding Transmission", TR-94-039, Aug. 1994, 36 pages, International Computer Science Institute, Berkeley, California.
Birney, Bill , "Intelligent Streaming", May 2003, Microsoft.
Goyal, Vivek K., "Multiple Description coding: Compression Meets the Network", Sep. 2001, pp. 74-93, IEEE Signal Processing Magazine.
ON2 Technologies, Inc.,, "TrueMotion VP7 Video Codec", *White Paper*, Document Version 1.0, Jan. 10, 2005, (13 pages).
Pathan, Al-Mukaddim , et al., "A Taxonomy and Survey of Content Delivery Networks", Australia, Feb. 2007. Available at http://www.gridbus.org/reports/CDN-Taxonomy.pdf.
Puri, Rohit , et al., "Multiple Description Source Coding Using Forward Error Correction Codes", Oct. 1999, 5 pages, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California.
Wicker, Stephen B., "Error Control Systems for Digital Communication and Storage", Prentice-Hall, Inc., New Jersey, USA, 1995, Parts 1-6.
USPTO "International Search Report" mailed Dec. 12, 2008; International Appln. No. PCT/US2008/061035, filed Apr. 21, 2008.
Australian Government "Examiner's First Report" dated Oct. 17, 2011; Australian Patent Appln. No. 2011213730.
Korean Intellectual Property Office "Official Notice of Preliminary Rejection" issued Jul. 28, 2011; Korean Patent Appln. No. 10-2006-7025274.
Japan Patent Office "Notice of Rejection Ground" mailed Apr. 26, 2011; Japanese Patent Appln. No. 2007-511070.
Fujisawa, Hiroshi et al. "Implementation of Efficient Access Mechanism for Multiple Mirror-Servers" IPSJ SIG Technical Report, vol. 2004, No. 9 (2004-DPS-116), Jan. 30, 2004, Information Processing Society of Japan, pp. 37-42.
USPTO "Final Office Action" mailed Feb. 23, 2011; U.S. Appl. No. 12/075,475, filed Mar. 10, 2008.
USPTO "Final Office Action" mailed Mar. 17, 2011; U.S. Appl. No. 11/834,548, filed Mar. 17, 2011.
USPTO "Examiner's Answer" mailed Feb. 16, 2011; U.S. Appl. No. 11/116,783, filed Apr. 28, 2005.
Major, R. Drew et al. "Reply Brief" filed Apr. 18, 2011; U.S. Appl. No. 11/116,783, filed Apr. 28, 2005.
Liu, Jiangchuan et al. "Adaptive Video Multicast Over the Internet" IEEE Computer Society, 2003.
Rejaie, Reza et al. "Architectural Considerations for Playback of Quality Adaptive Video Over the Internet" University of Southern California, Information Sciences Institute, 1998.
Roy, Sumit et al. "A System Architecture for Managing Mobile Streaming Media Services" Streaming Media Systems Group, Hewlett-Packard Laboratories, 2003.
Xu, Dongyan et al. "On Peer-to-Peer Media Streaming" Department of Computer Sciences, Purdue University, 2002.
Kozamernik, Franc "Media Streaming Over the Internet—An Over of Delivery Technologies" EBU Technical Review, Oct. 2002.
Lienhart, Rainer et al. "Challenges in Distributed Video Management and Delivery" Intel Corporation, EECS Dept., UC Berkeley, 2000-2002.
Zhang, Xinyan et al. "CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming" IEEE 2005.
Guo, Yang "DirectStream: A Directory-Based Peer-To-Peer Video Streaming Service" LexisNexis, Elsevier B.V. 2007.
USPTO "Notice of Allowance" mailed Oct. 5, 2012 for U.S. Appl. No. 12/075,475, filed Mar. 10, 2008.
Japan Patent Office "Interrogation" dated Nov. 6, 2012 for Japanese Patent Appln. No. 2007-511070.
USPTO "Non-Final Office Action" mailed Dec. 17, 2012 for U.S. Appl. No. 13/617,114, filed Sep. 14, 2012.
USPTO "Non-Final Office Action" mailed Nov. 23, 2012 for U.S. Appl. No. 11/834,548, filed Aug. 6, 2007.

* cited by examiner

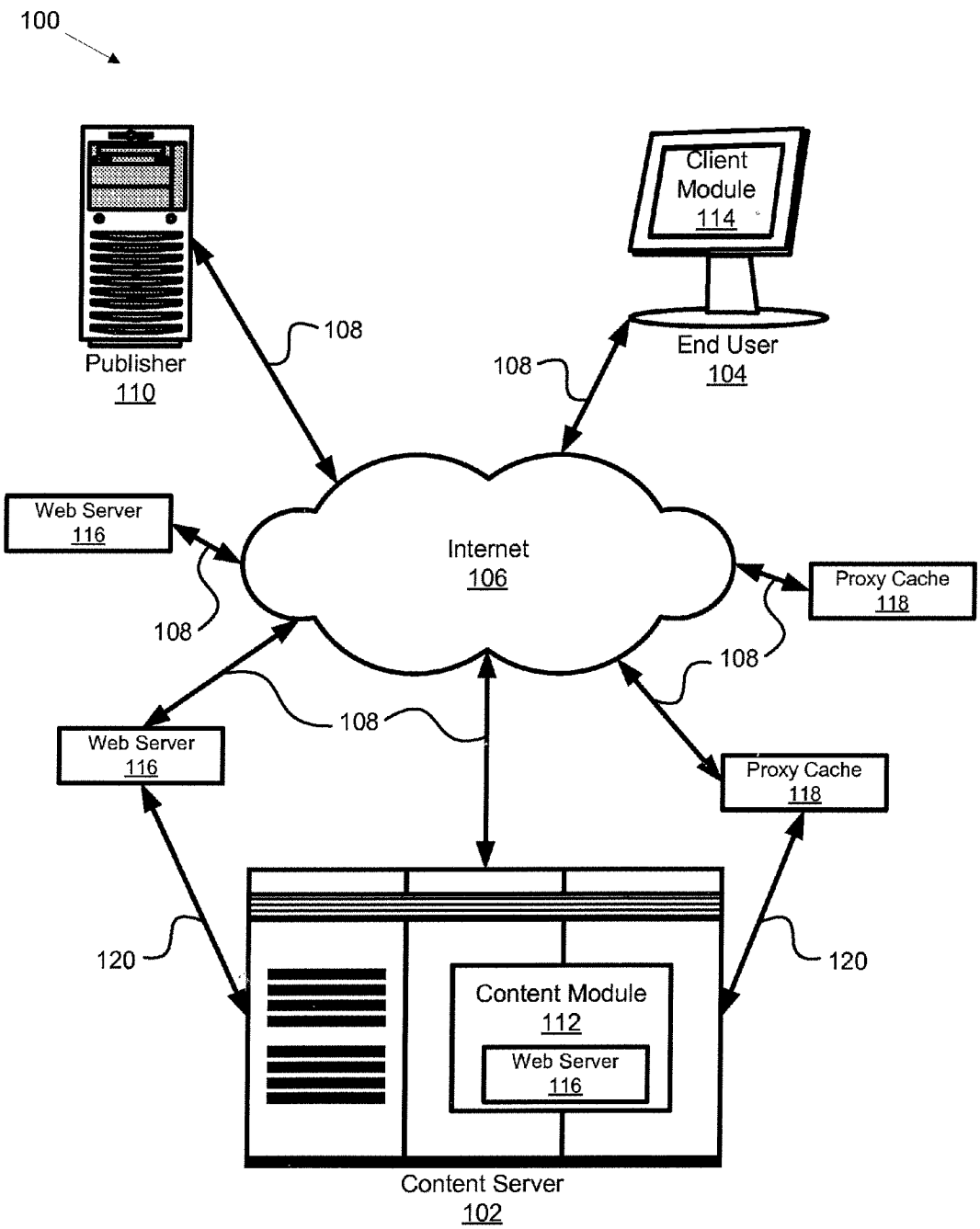


FIG. 1

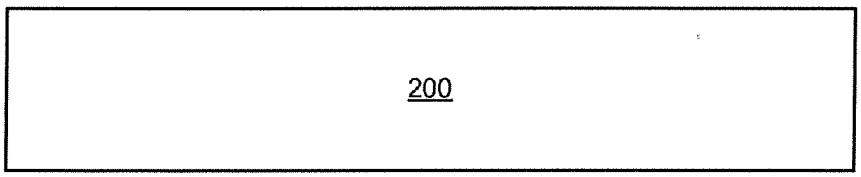


FIG. 2a

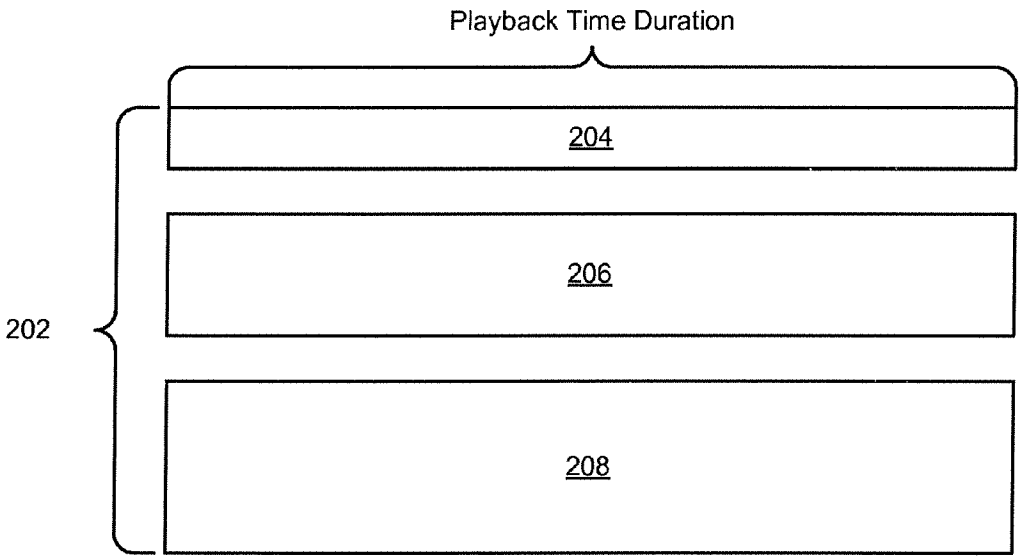


FIG. 2b

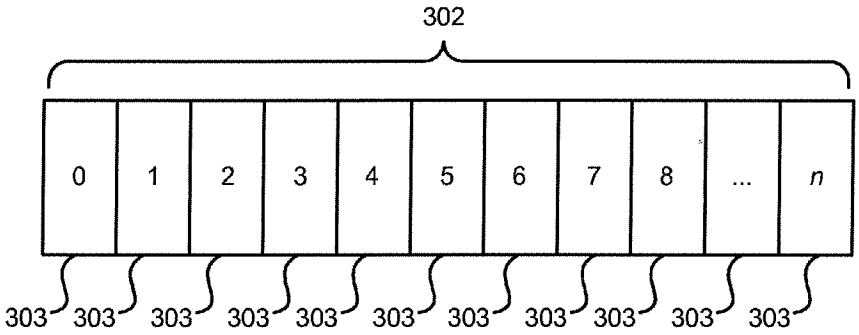


FIG. 3a

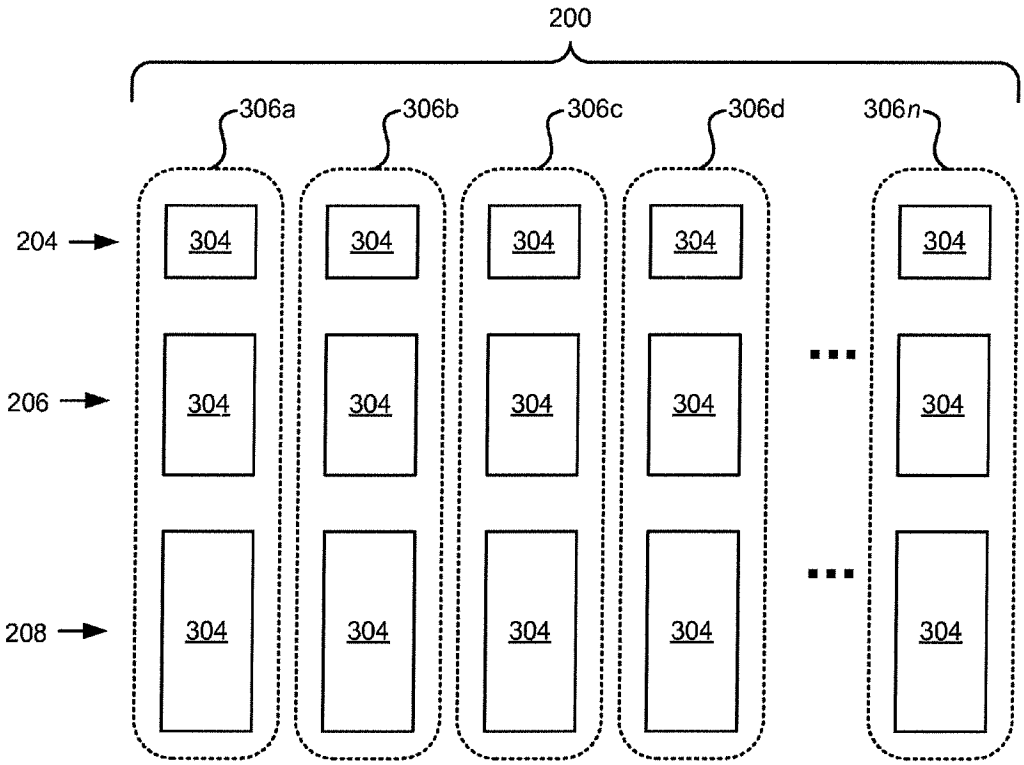


FIG. 3b

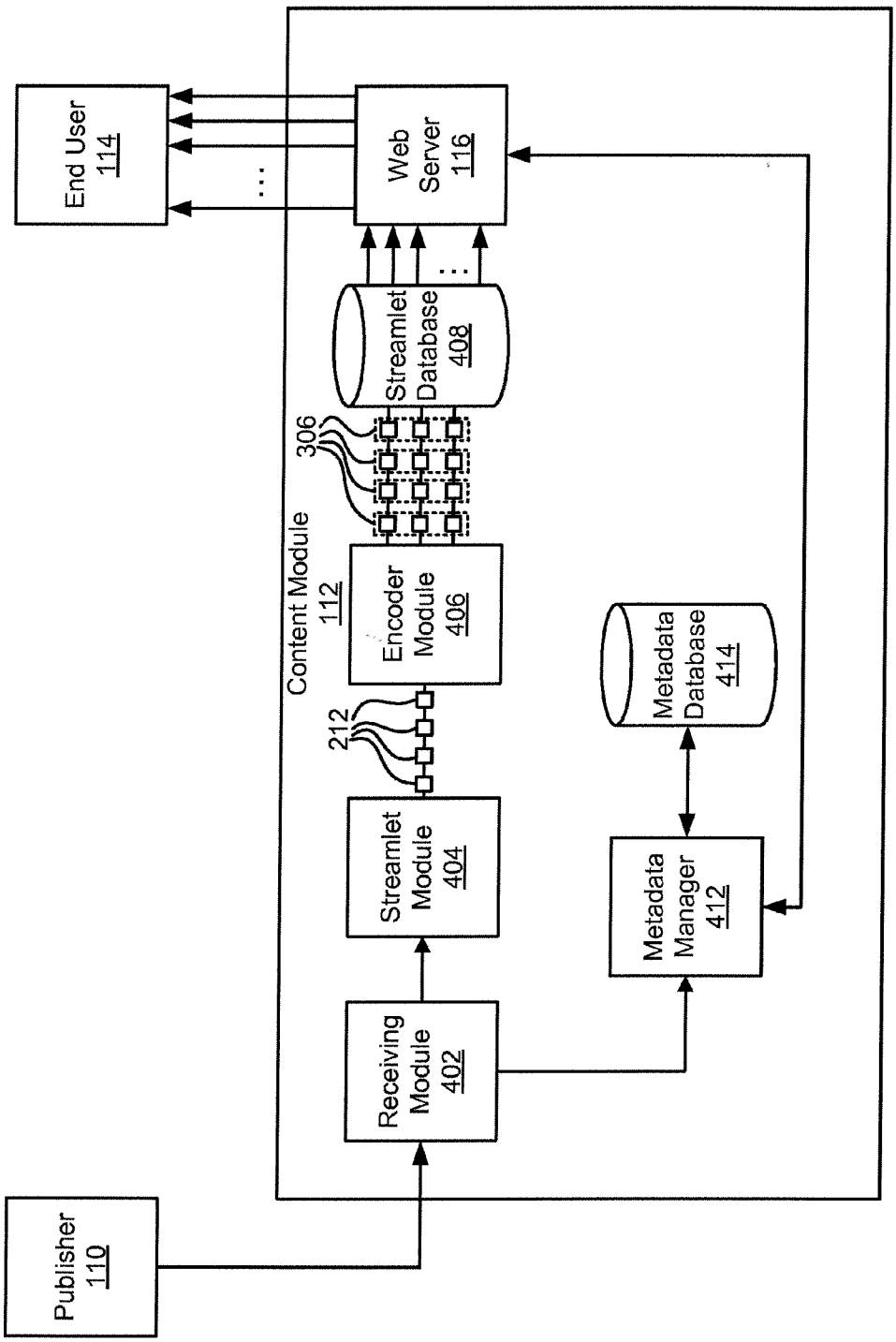


FIG. 4

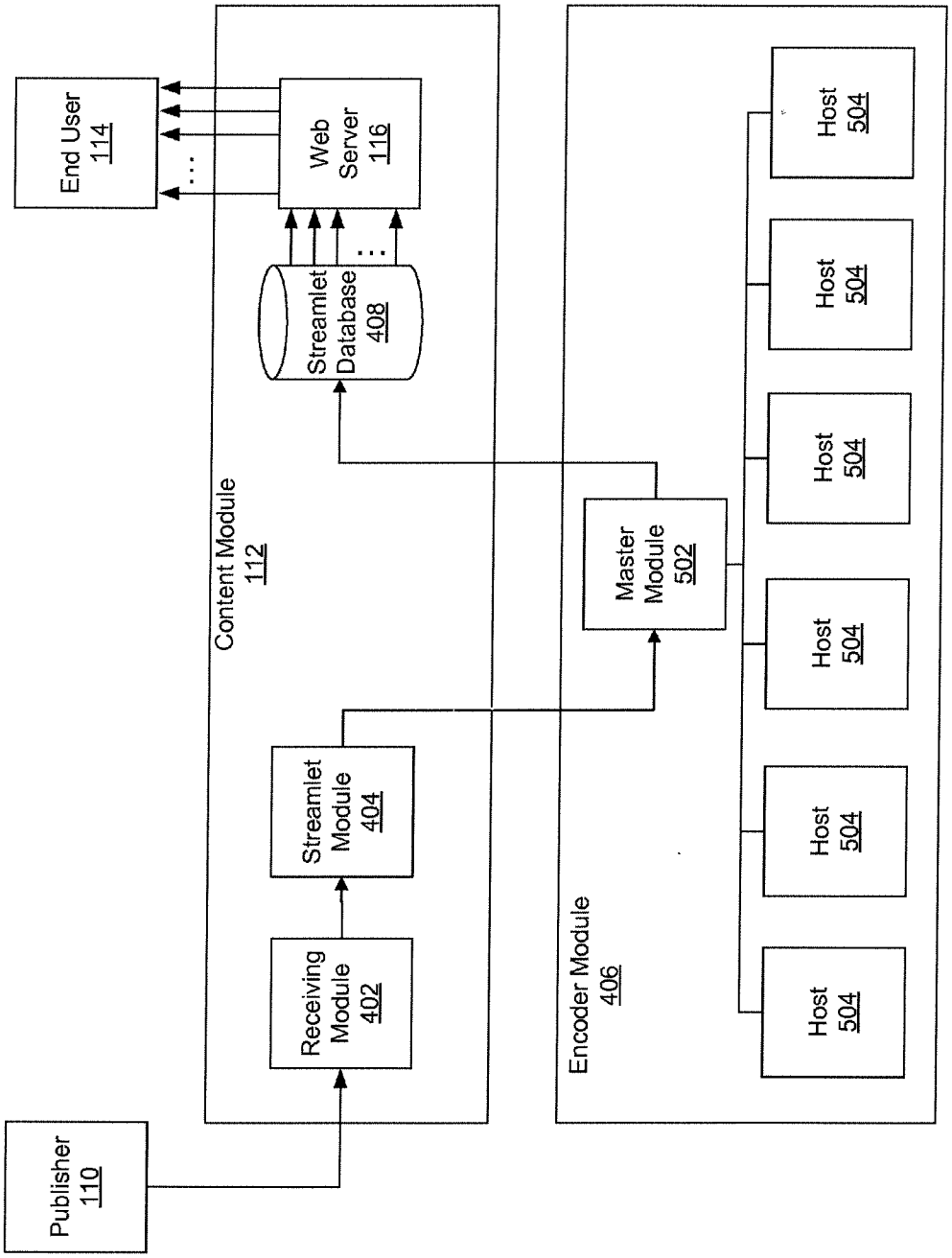


FIG. 5a

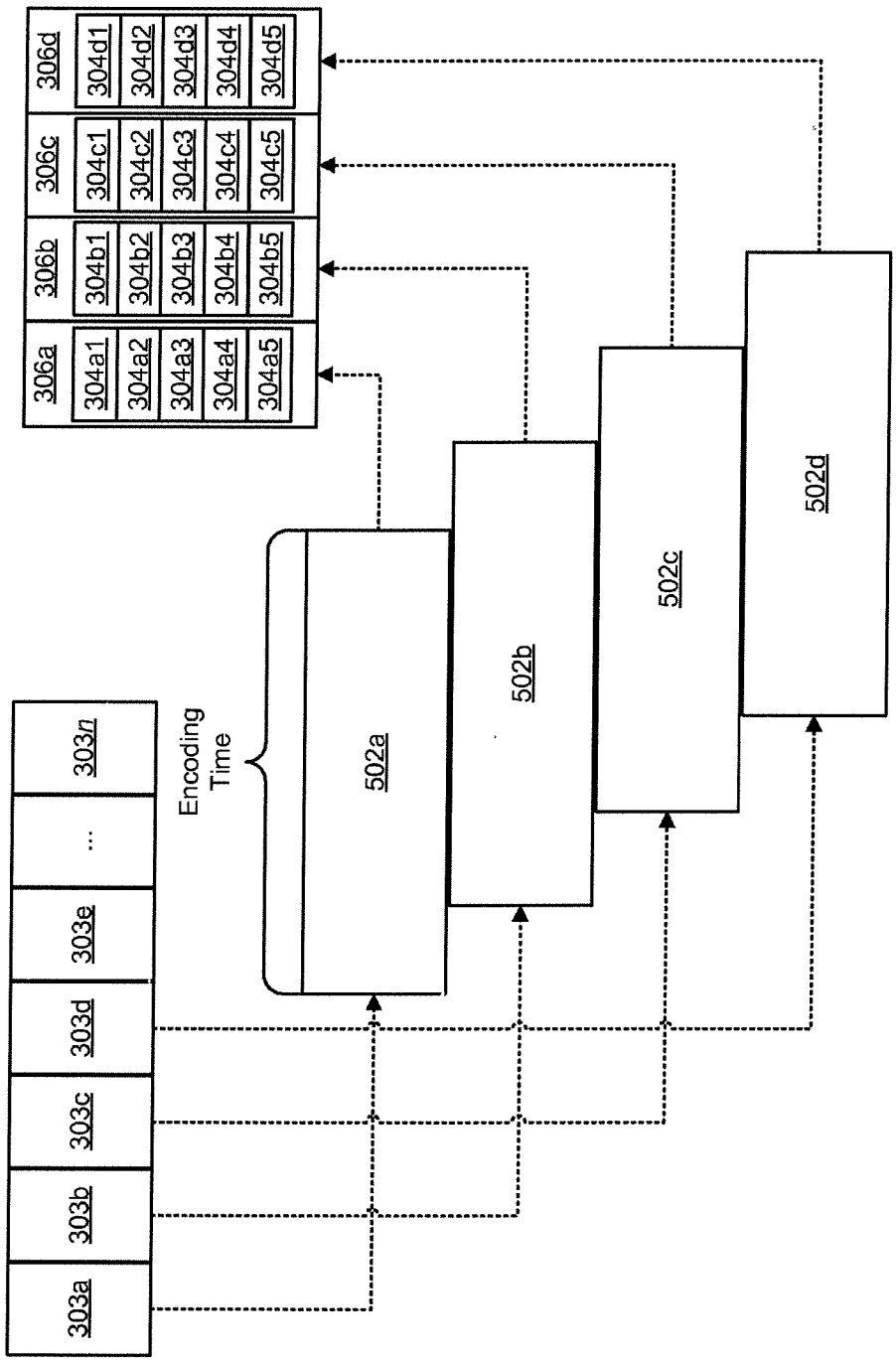


FIG. 5b

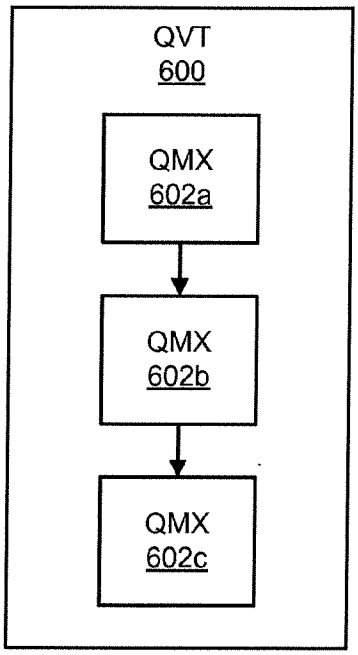


FIG. 6a

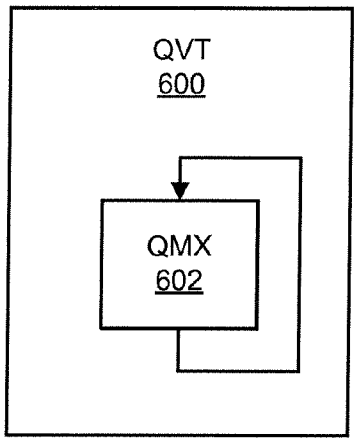


FIG. 6b

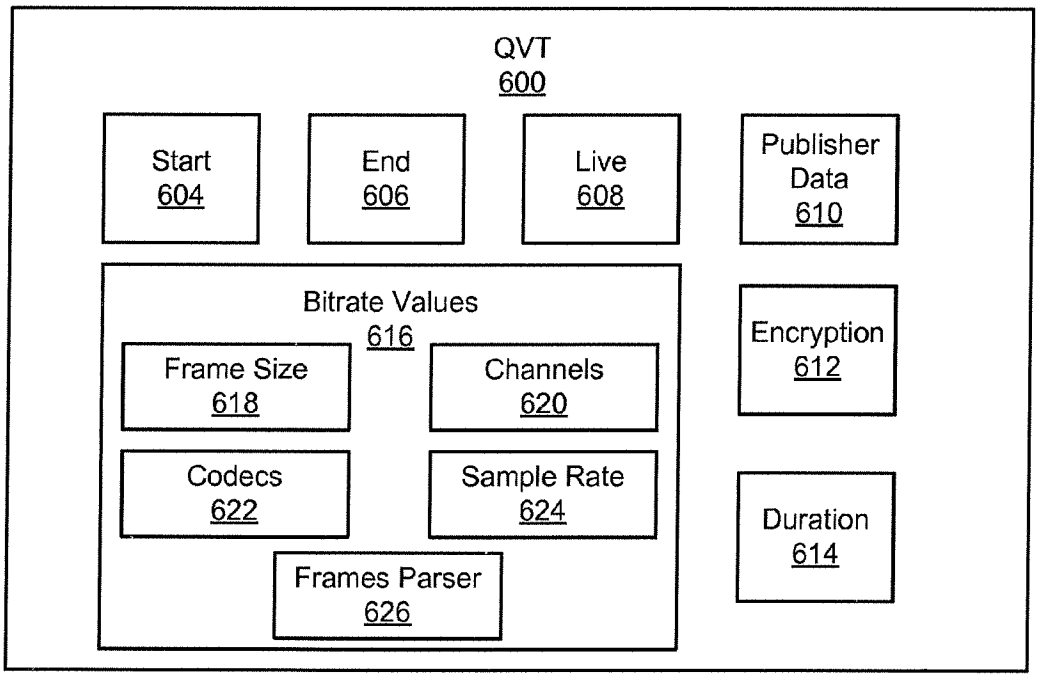


FIG. 6c

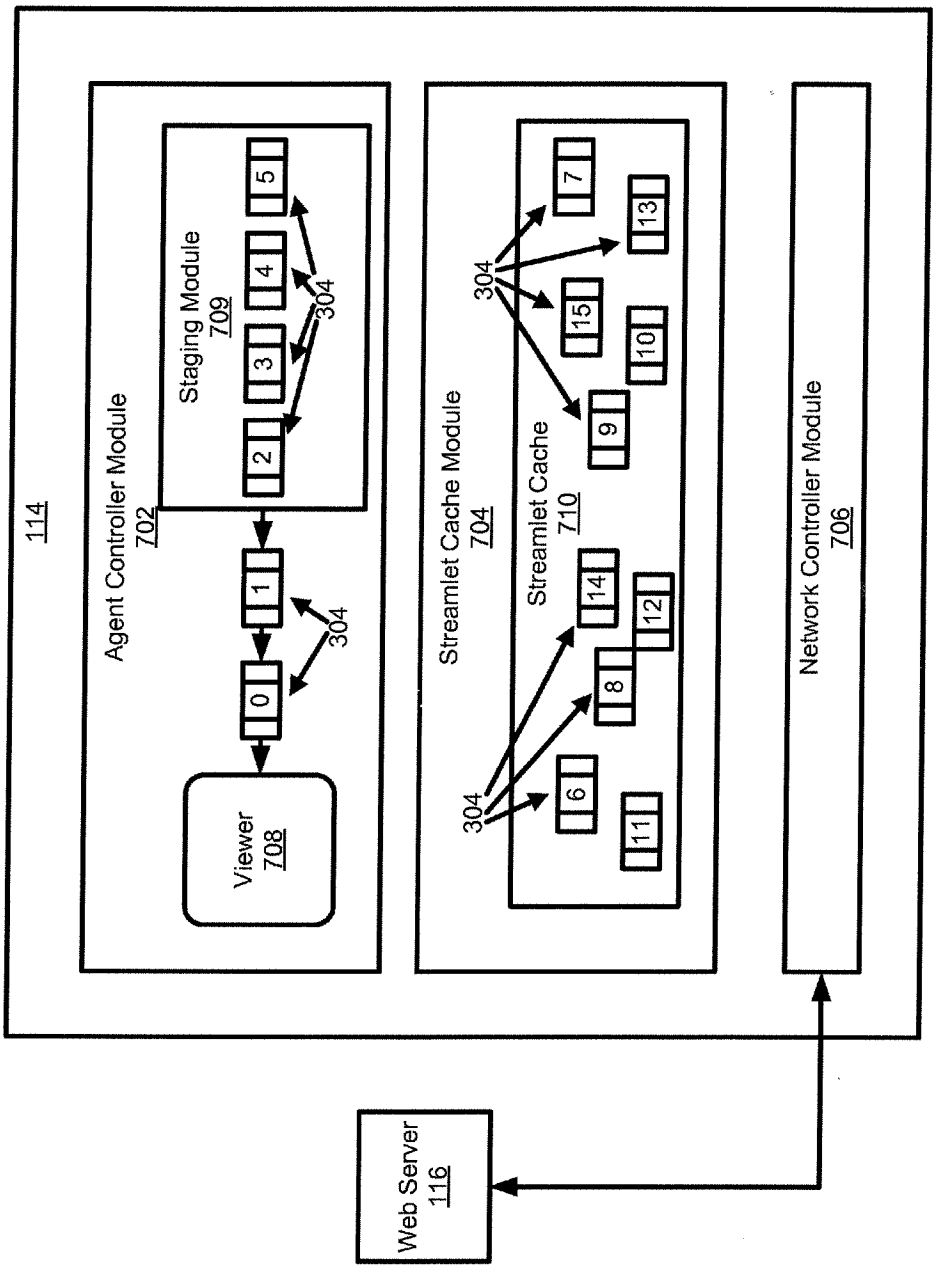


FIG. 7

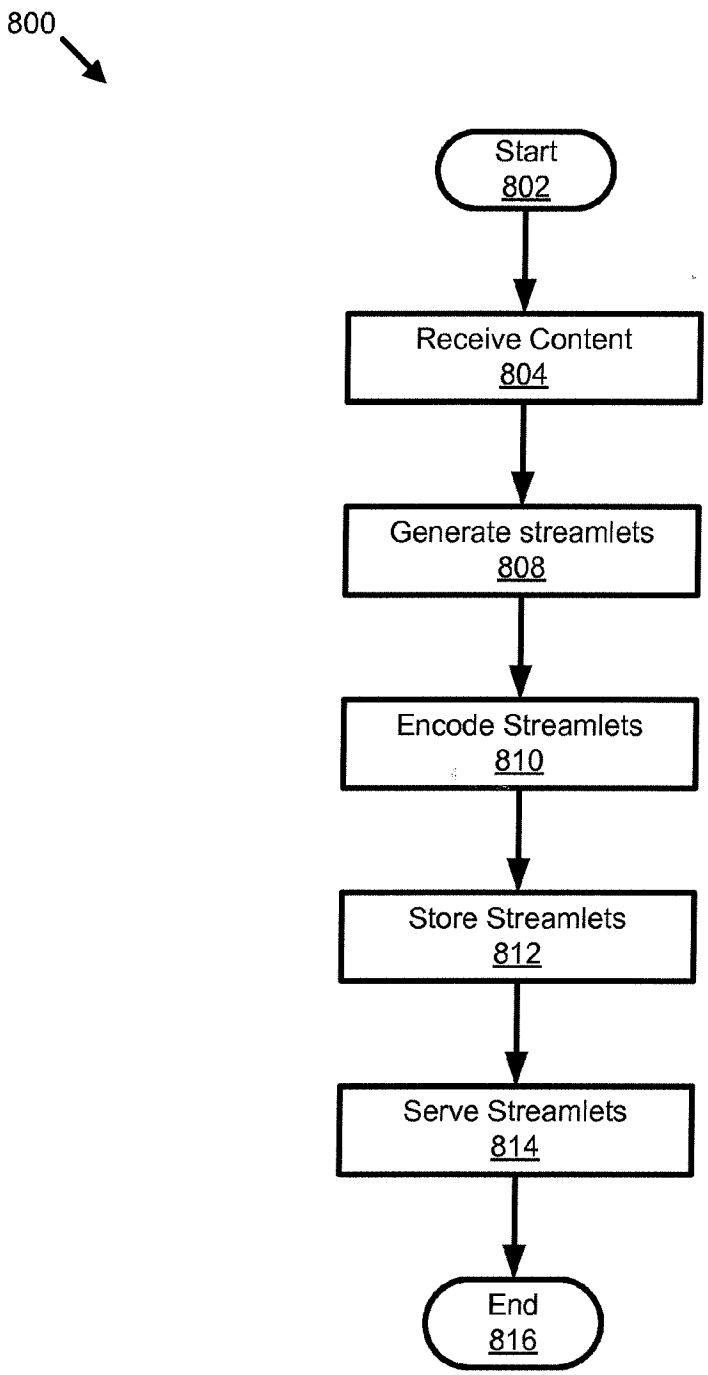


FIG. 8

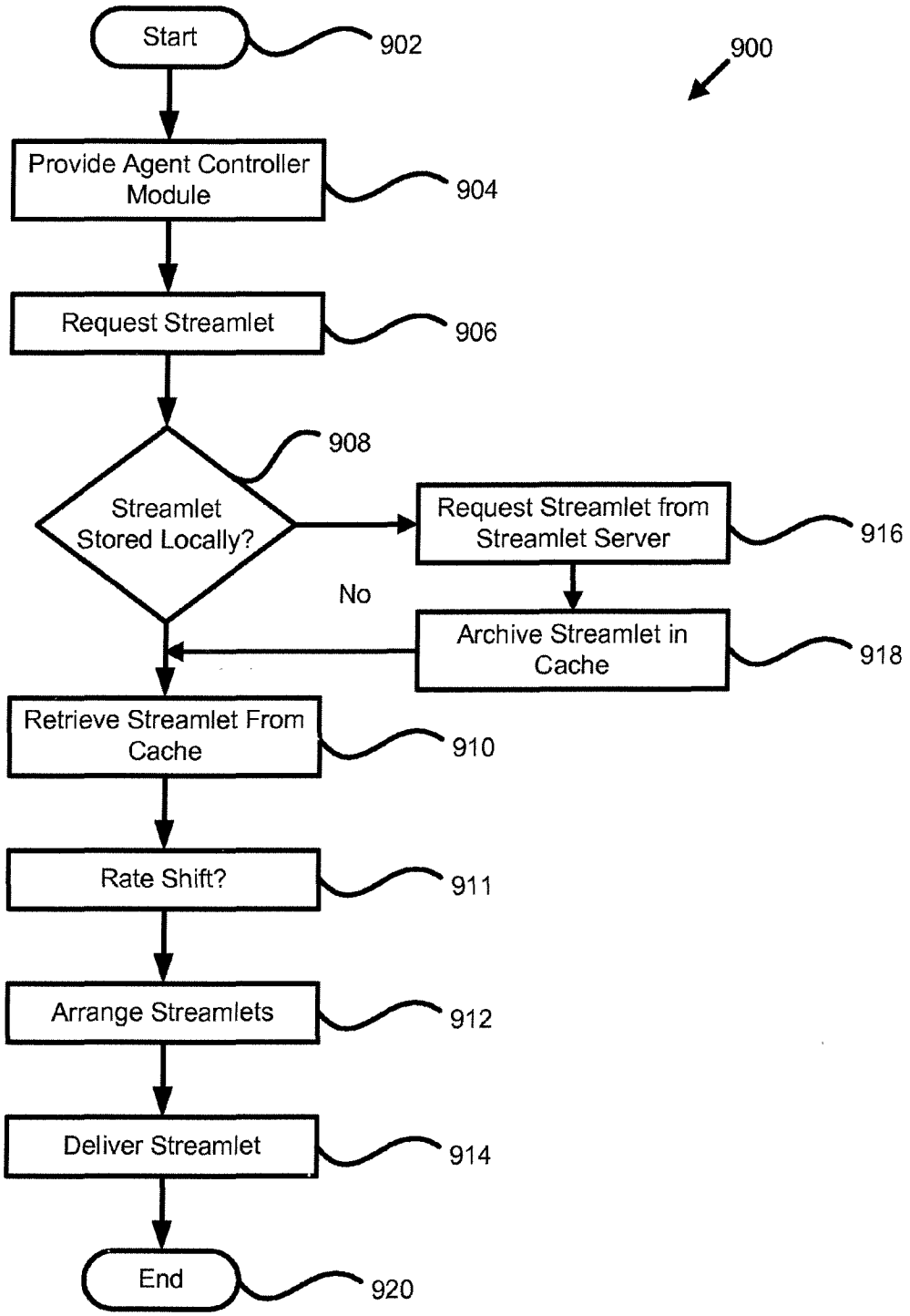


FIG. 9

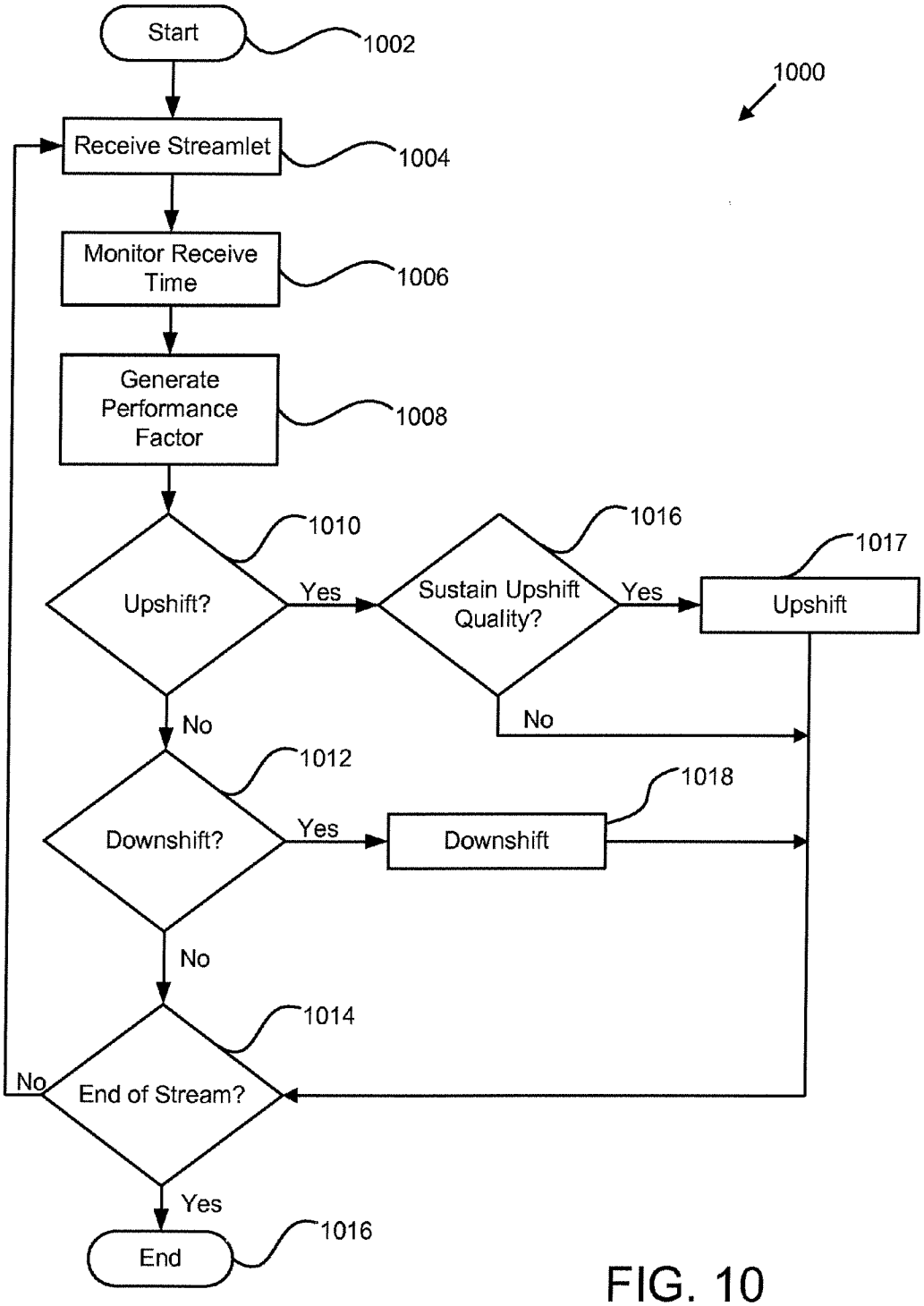


FIG. 10

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

CROSS-REFERENCES TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 11/673,483, filed on Feb. 9, 2007, which is a continuation-in-part of application Ser. No. 11/116,783, filed on Apr. 28, 2005, which claims the benefit of U.S. Provisional Application No. 60/566,831, filed on Apr. 30, 2004, all of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

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.

2. Description of the Related Art

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.

"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.

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.

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.

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.

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.

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.

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.

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

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.

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.

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 0.1 and 5 seconds.

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.

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

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.

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.

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.

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.

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.

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.

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

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:

FIG. **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;

FIG. **2***a* is a schematic block diagram graphically illustrating one embodiment of a media content file;

FIG. **2***b* is a schematic block diagram illustrating one embodiment of a plurality of streams having varying degrees of quality and bandwidth;

FIG. **3***a* is a schematic block diagram illustrating one embodiment of a stream divided into a plurality of source streamlets;

FIG. **3***b* is a schematic block diagram illustrating one embodiment of sets of streamlets in accordance with the present invention;

FIG. **4** is a schematic block diagram illustrating in greater detail one embodiment of the content module in accordance with the present invention;

FIG. **5***a* is a schematic block diagram illustrating one embodiment of an encoder module in accordance with the present invention;

FIG. **5***b* is a schematic block diagram illustrating one embodiment of parallel encoding of streamlets in accordance with the present invention;

FIG. **6***a* is a schematic block diagram illustrating one embodiment of a virtual timeline in accordance with the present invention;

FIG. **6***b* is a schematic block diagram illustrating an alternative embodiment of a VT in accordance with the present invention;

FIG. **6***c* is a schematic block diagram illustrating one embodiment of a QMX in accordance with the present invention;

FIG. **7** is a schematic block diagram graphically illustrating one embodiment of a client module in accordance with the present invention;

FIG. **8** is a schematic flow chart diagram illustrating one embodiment of a method for processing content in accordance with the present invention;

FIG. **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

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

DETAILED DESCRIPTION OF THE INVENTION

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.

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.

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.

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.

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

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.

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 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.

In the depicted embodiment, the system **100** also includes a publisher **110**, and a web server **116**. The publisher **110** may be a creator or distributor of content. For example, if the content to be streamed were a broadcast of a television program, the publisher **110** may be a television or cable network channel such as NBC®, or MTV®. Content may be transferred over the Internet **106** to the content server **102**, where the content is received by a content module **112**. The content module **112** may be configured to receive, process, and store content. In one embodiment, processed content is accessed by a client module **114** configured to play the content on the end user station **104**. In a further embodiment, the client module **114** is configured to receive different portions of a content stream from a plurality of locations simultaneously. For example, the client module **114** may request and receive content from any of the plurality of web servers **116**.

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**.

FIG. **2***a* 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.

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.

FIG. 3*a* 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.

FIG. 3*b* 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 **306***a* encompasses all streamlets having a time index of 00:00. The set **306***a* 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**.

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

FIG. **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.

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**.

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.

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, 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.

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 FIG. 3*b*, may comprise streamlets having an identical time index and duration, and a unique bitrate. As with FIG. 3*b*, 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.

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.

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.

Once the streamlets **304** have been received and processed, the client module **114** may request streamlets **304** using HTTP from the web server **116**. Using a standard protocol such as HTTP eliminates the need for network administrators to configure firewalls to recognize and pass through network traffic for a new, specialized protocol. Additionally, since the client module **114** initiates the request, the web server **116** is only required to retrieve and serve the requested streamlet **304**. In a further embodiment, the client module **114** may be configured to retrieve streamlets **304** from a plurality of web servers **116**.

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**

FIG. **5***a* 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.

The master module (hereinafter "master") **502** is configured to receive streamlets **303** from the streamlet module **404** and stage the streamlet **303** for processing. In one embodiment, the master **502** may decompress each source streamlet **303** to produce a raw streamlet. As used herein, the term "raw streamlet" refers to a streamlet **303** that is uncompressed or lightly compressed to substantially reduce size with no significant loss in quality. A lightly compressed raw streamlet can be transmitted more quickly and to more hosts. Each host **504** is coupled with the master **502** and configured to receive a raw streamlet from the master **502** for encoding. The hosts **504**, in one example, generate a plurality of streamlets **304** having identical time indices and durations, and varying bitrates. Essentially each host **504** may be configured to generate a set **306** from the raw streamlet **503** sent from the master **502**. Alternatively, each host **504** may be dedicated to producing a single bitrate in order to reduce the time required for encoding.

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.

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.

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.

The present invention, however, segments the content file **200** into source streamlets **303** and the two-pass encoding to a plurality of streams **202** may be performed on each corresponding raw streamlet without waiting for a TV show to end, for example. As such, the content module **112** is capable of streaming the streamlets over the Internet shortly after the content module **112** begins capture of the content file **200**. The delay between a live broadcast transmitted from the publisher **110** and the availability of the content depends on the computing power of the hosts **504**.

FIG. **5***b* 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 FIG. **4**) begins to capture the content file and the streamlet module **404** generates a first streamlet **303***a* and passes the streamlet to the encoding module **406**. The encoding module **406** may take 10 seconds, for example, to generate the first set **306***a* of streamlets **304***a* (**304***a***1**, **304***a***2**, **304***a***3**, etc. represent streamlets **304** of different bitrates). FIG. **5***b* 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**.

During the 10 seconds required to encode the first streamlet **303***a*, the streamlet module **404** has generated five additional 2-second streamlets **303***b*, **303***c*, **303***d*, **303***e*, **303***f*, for encoding and the master **502** has prepared and staged the corresponding raw streamlets. Two seconds after the first set **306***a* is available the next set **306***b* 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.

A system as described above beneficially enables multi-pass encoding of live events. Multi-pass encoding systems of the prior art require that the entire content be captured (or be complete) because in order to perform multi-pass encoding the entire content must be scanned and processed more than once. This is impossible with prior art systems because content from a live event is not complete until the event is over. As such, with prior art systems, multi-pass encoding can only be performed once the event is over. Streamlets, however, may be encoded as many times as is deemed necessary. Because the streamlet is an encapsulated media object of 2 seconds (for example), multi-pass encoding may begin on a live event once the first streamlet is captured. Shortly after multi-pass encoding of the first streamlet **303***a* is finished, multi-pass encoding of the second streamlet **303***b* finishes, and as such multi-pass encoding is performed on a live event and appears live to a viewer.

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.

The parallel offset pipeline approach described with reference to FIG. **5***b* 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.

Returning now to FIG. **5***a*, as depicted, the master **502** and the hosts **504** may be located within a single local area network, or in other terms, the hosts **504** may be in close physical proximity to the master **502**. Alternatively, the hosts **504** may receive encoding jobs from the master **502** over the Internet or other communications network. For example, consider a live sports event in a remote location where it would be difficult to setup multiple hosts. In this example, a master performs no encoding or alternatively light encoding before publishing the streamlets online. The hosts **504** would then retrieve those streamlets and encode the streamlets into the multiple bitrate sets **306** as described above.

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.

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.

FIG. **6***a* 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 **602***a* followed by QMX **602***b* and QMX **602***c*. As such, the publisher may define a broadcast schedule in a manner similar to a television station.

FIG. **6***b* 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.

FIG. **6***c* is a schematic block diagram illustrating one embodiment of a QMX **602** in accordance with the present invention. In one embodiment, the QMX **602** contains a multitude of information generated by the content module **112** configured to describe the content file **200**. Examples of information include, but are not limited to, start index **604**, end index **606**, whether the content is live **608**, proprietary publisher data **610**, encryption level **612**, content duration **614** and bitrate values **616**. The bitrate values **616** may include frame size **618**, audio channel **620** information, codecs **622** used, sample rate **624**, and frames parser **626**.

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.

FIG. **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.

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.

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 FIG. **10**.

The agent controller module **702** may be configured to receive user commands from the viewer **708**. Such commands may include play, fast forward, rewind, pause, and stop. In one embodiment, the agent controller module **702** requests streamlets **304** from the streamlet cache module **704** and arranges the received streamlets **304** in a staging module **709**. The staging module **709** may be configured to arrange the streamlets **304** in order of ascending playback time. In the depicted embodiment, the streamlets **304** are numbered 0, 1, 2, 3, 4, etc. However, each streamlet **304** may be identified with a unique filename.

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 fur-

ther 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**.

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

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.

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**.

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.

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.

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.

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.

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.

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.

FIG. **8** is a schematic flow chart diagram illustrating one embodiment of a method **800** for processing content in accordance with the present invention. In one embodiment the method **800** starts **802**, and the content module **112** receives **804** content from the publisher **110**. Receiving content **804** may comprise receiving **804** a digital copy of the content file **200**, or digitizing a physical copy of the content file **200**. Alternatively, receiving **804** content may comprise capturing a radio, television, cable, or satellite broadcast. Once received **804**, the streamlet module **404** generates **808** a plurality of source streamlets **303** each having a fixed duration. Alternatively, the streamlets **303** may be generated with a fixed file size.

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

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**.

FIG. **9** is a schematic flow chart diagram illustrating one embodiment of a method **900** for viewing a plurality of streamlets in accordance with the present invention. The method **900** starts and an agent controller module **702** is provided **904** and associated with a viewer **708** and provided with a staging module **709**. The agent controller module **702** then requests **906** a streamlet **304** from the streamlet cache module **704**. Alternatively, the agent controller module **702** may simultaneously request **906** a plurality of streamlets **304** the streamlet cache module **704**. If the streamlet is stored **908** locally in the streamlet cache **710**, the streamlet cache module **704** retrieves **910** the streamlet **304** and sends the streamlet to the agent controller module **702**. Upon retrieving **910** or receiving a streamlet, the agent controller module **702** makes **911** a determination of whether or not to shift to a higher or lower quality stream **202**. This determination will be described below in greater detail with reference to FIG. **10**.

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.

Referring now to FIG. **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 FIG. **9**. The method **1000** starts and the agent controller module **702** receives **1004** a streamlet **304** as described above with reference to FIG. **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.

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:

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

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:

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

The policy determination about whether or not to upshift **1010** 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 **1016**. In one embodiment, the trigger threshold ϕ_{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 ϕ_{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 $\phi_{higher} \geqq \Theta_{up}$ then use of the higher quality stream is considered sustainable. If the decision of whether or not the higher stream rate is sustainable **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**.

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 $\phi_{current} > \Theta_{down}$ then the stream quality may be adequate, and the agent controller module **702** does not downshift **1018** stream quality. However, if $\phi_{current} \leqq \Theta_{down}$, 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.

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. A system for adaptive-rate content streaming videos for playback on end user stations, the system comprising a set of one or more servers including,

a storage device to store a plurality of different copies of a same single video each encoded at a different bit rate and each divided into a plurality of streamlets that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire playback duration of the single video, wherein the 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, and wherein each of the streamlets of each of the pluralities is a separate content file that is independently playable by the end user stations, and

a web server, to be executed on the set of servers, wherein the web server is configured to:

receive, for a set of one or more initial and sequential ones of the time indexes, at least one streamlet request over a set of one or more network connections from a requesting one of the end user stations to retrieve the separate content files storing the portions of the single video starting at the set of time indexes from one of the different copies;

retrieve from the storage device the requested content files from the one copy;

send the retrieved content files from the one copy to the requesting one of the end user stations over the set of network connections;

receive, for each subsequent time index, at least one streamlet request over the set of network connections from the requesting one of the end user stations to retrieve the separate content files storing the portion of the single video starting at that time index from one of the different copies currently selected, wherein the requesting one of the end user stations selects the currently selected one of the different copies dependent upon ongoing determinations to shift the playback quality to a higher or lower quality one of the different copies, wherein the shifts in playback quality occur at the time indexes, and wherein the requesting end user station initiates the shifts in playback quality during streaming of the single video through requests for separate content files storing different playback qualities of the encoded streamlets for subsequent ones of the time indexes;

retrieve from the storage device the requested content files from the currently selected one of the different copies; and

send the retrieved content files from the currently selected one of the different copies to the requesting one of the end user stations over the set of network connections.

2. The system of claim **1**, wherein the set of servers does not require a specialized protocol for said adaptive-rate content streaming.

3. The system of claim **1**, wherein at least one of the set of one or more servers is a cache server.

4. The system of claim **1**, wherein the web server receives the streamlet requests using a standard protocol.

5. The system of claim **1**, wherein the web server receives the streamlet request as Hypertext Transport Protocol (HTTP) GET requests.

6. The system of claim **1**, wherein at least one of the set of one or more servers is a content server, and wherein the content server includes a content module including,

the web server;

a receiving module configured to receive the videos;

a streamlet module configured to segment the received videos to generate for each a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video; and

an encoding module configured to encode each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the adaptive-rate content streaming, wherein each of the encoded streamlets in each of the sets is stored as the separate content file, wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the corresponding video, and wherein the separate content files within each of the sets are independently requestable by end user stations.

7. The system of claim **6**, further comprising a streamlet database, wherein the encoding module is further configured to transmit the separate content files each storing one of the encoded streamlets to the streamlet database to be stored, and wherein the web server is coupled to the streamlet database to retrieve the requested ones of the separate content files from the streamlet database to transmit to the requesting one of the end user stations.

8. The system of claim **6**, wherein the content module further comprises:

a metadata module configured to receive metadata from a publisher, wherein the metadata comprises searchable content information; and

a metadata database to store the metadata.

9. The system of claim 8, wherein the metadata further comprises metadata associated with an end user navigational environment for the single video.

10. The system of claim 6, wherein the encoding module is configured to encode multiple of the plurality of sequential raw streamlets in parallel.

11. The system of claim 6, wherein the encoding module is configured to encode the plurality of sequential raw streamlets at regular intervals using an offset parallel processing scheme.

12. The system of claim 6, wherein the encoding module comprises a master module configured to assign an encoding job to one of a plurality of hosts to generate at least one of the separate content files from at least one of the plurality of sequential raw streamlets in response to an encoding job completion bid received from the one host.

13. The system of claim 6, wherein the encoding module is further configured to multi-pass encode each of the plurality of sequential raw streamlets of at least one of the videos.

14. The system of claim 13, wherein the at least one of the videos is of a live event.

15. A method for adaptive-rate content streaming videos for playback on a content player on an end user station, the method comprising:

receiving a selected one of the videos for generating streamlets for adaptive-rate content streaming; and

creating a plurality of different copies of the same selected video, wherein each of the different copies is encoded at a different bit rate and is divided into a plurality of streamlets that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the selected video, wherein the time indexes of the streamlets are the same for the different copies such that streamlets with the same time indexes form the different copies independently yield the same portions of the selected video, and wherein each of the streamlets of each of the pluralities is a separate content file that is independently playable by the end user station to thereby allow the end user station to initiate shifts in playback quality during streaming of the selected video through requests for separate content files storing different playback qualities of the encoded streamlets for subsequent ones of the time indexes.

16. The method of claim 15, wherein said creating comprises:

encoding the selected video into each of the plurality of different copies at each of the different bit rates supported by the adaptive-rate content streaming; and

segmenting each of the plurality of different copies into the plurality of streamlets, wherein each of the plurality of streamlets of each of the plurality of different copies is stored as the separate content file, and wherein the separate content files within each of the sets are independently requestable by end user stations.

17. The method of claim 15, wherein said creating comprises:

segmenting the selected video and generating a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only the portion that starts at the unique time index and whose duration is less than the entire duration of the corresponding video; and

encoding each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the adaptive-rate content streaming, wherein each of the encoded streamlets in each of the sets is stored as the separate content file, and wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the corresponding video, wherein the separate content files within each of the sets are independently requestable by end user stations.

18. The method of claim 17, wherein said encoding comprises encoding multiple of the plurality of sequential raw streamlets in parallel.

19. The method of claim 17, wherein said encoding further comprises multi-pass encoding each of the plurality of sequential raw streamlets of the corresponding video.

20. A method for streaming from a set of one or more servers videos for playback on a content player on an end user station, the method comprising:

streaming from the set of servers a selected one of the videos for playback on the content player, wherein the set of servers stores a plurality of different copies of the same selected video each encoded at a different bit rate and each divided into a plurality of streamlets that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire playback duration of the selected video, wherein the 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 selected video, and wherein each of the streamlets of each of the pluralities is a separate content file that is independently playable by the end user stations, and wherein said streaming comprises:

receiving, for a set of one or more initial and sequential ones of the time indexes, at least one streamlet request over a set of one or more network connections from a requesting one of the end user stations to retrieve the separate content files storing the portions of the single video starting at the set of time indexes from one of the different copies;

retrieving from the storage device the requested content files with the set of time indexes from the one copy;

sending the retrieved content files from the one copy to the requesting one of the end user stations over the set of network connections;

receiving, for each subsequent time index, at least one streamlet request over the set of network connections from the requesting one of the end user stations to retrieve the separate content files storing the portion of the single video starting at that time index from one of the different copies currently selected, wherein the requesting one of the end user stations selects the currently selected one of the different copies dependent upon successive determinations to shift the playback quality to a higher or lower quality one of the different copies, wherein the shifts in playback quality occur at the time indexes, and wherein the requesting end user station initiates the shifts in playback quality during streaming of the single video through requests for separate content files storing different playback qualities of the encoded streamlets for subsequent ones of the time indexes;

retrieving from the storage device the requested content files with the time index from the currently selected one of the different copies; and

sending the retrieved content files from the currently selected one of the different copies to the requesting one of the end user stations over the set of network connections.

21. The method of claim **20**, wherein the streamlet requests comprise the time index and the bit rate that identify which of the separate content files to be retrieved.

22. An apparatus for multi-bitrate video and/or audio content streaming, the apparatus comprising:

one or more computing devices including,

a streamlet module configured to segment videos to generate for each a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video;

an encoding module configured to encode each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the multi-bitrate content streaming, wherein each of the encoded streamlets in each of the sets is stored as a separate content file, wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the video, wherein the separate content files within each of the sets

are independently requestable by end user stations, and wherein shifts between the different bit rates are made at the time indexes during streaming of each of the videos; and

a web server to be coupled over the Internet to the end user stations to receive requests from the end user stations for different ones of the separate content files from different ones of the sets and to transmit the requested ones of the separate content files to the requesting one of the end user stations, wherein each of the end user stations initiate each of the shifts between the different bit rates during streaming of each of the videos through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

23. A method for multi-bitrate video and/or audio content streaming, the method comprising:

segmenting a single video and generating a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video;

encoding each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the multi-bitrate content streaming, wherein each of the encoded streamlets in each of the sets is stored as a separate content file, wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the single video, wherein the separate content files within each of the sets are independently requestable by end user stations, and wherein shifts between the different bit rates are made at the time indexes during streaming of the single video;

receiving requests from the end user stations over the Internet for different ones of the separate content files from different ones of the sets; and

transmitting the requested ones of the separate content files to the requesting one of the end user stations, wherein each of the end user stations initiate each of the shifts between the different bit rates during streaming of the single video through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 8,402,156 B2
APPLICATION NO. : 12/906940
DATED : March 19, 2013
INVENTOR(S) : Brueck et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

Item [75], insert --R. Drew Major--

Signed and Sealed this
Thirteenth Day of March, 2018

Andrei Iancu

Andrei Iancu
Director of the United States Patent and Trademark Office

EXHIBIT C

U 7701579

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

November 21, 2018

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF:

U.S. PATENT: *9,071,668*
ISSUE DATE: *June 30, 2015*

By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office

SYLVIA HOLLEY
Certifying Officer

UNITED STATES PATENT AND TRADEMARK OFFICE
DEPARTMENT OF COMMERCE

US009071668B2

(12) **United States Patent**
Brueck et al.

(10) **Patent No.: US 9,071,668 B2**
(45) **Date of Patent: *Jun. 30, 2015**

(54) **APPARATUS, SYSTEM, AND METHOD FOR MULTI-BITRATE CONTENT STREAMING**

(71) Applicant: **ECHOSTAR TECHNOLOGIES L.L.C.**, Englewood, CO (US)

(72) Inventors: **David F. Brueck**, Saratoga Springs, UT (US); **Mark B. Hurst**, Cedar Hills, UT (US)

(73) Assignee: **EchoStar Technologies L.L.C.**, Englewood, CO (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/106,051**

(22) Filed: **Dec. 13, 2013**

(65) **Prior Publication Data**

US 2014/0101329 A1 Apr. 10, 2014

Related U.S. Application Data

(63) Continuation of application No. 13/617,114, filed on Sep. 14, 2012, now Pat. No. 8,612,624, which is a continuation of application No. 12/906,940, filed on Oct. 18, 2010, now Pat. No. 8,402,156, which is a

(Continued)

(51) **Int. Cl.**
H04L 29/06 (2006.01)
H04N 7/24 (2011.01)
(Continued)

(52) **U.S. Cl.**
CPC ***H04L 65/608*** (2013.01); ***H04L 29/06027*** (2013.01); ***H04N 7/24*** (2013.01);

(Continued)

(58) **Field of Classification Search**
CPC H04L 65/80; H04L 65/4084
USPC 709/231–233
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,535,355 A 8/1985 Arn et al.
5,168,356 A 12/1992 Acampora et al.

(Continued)

FOREIGN PATENT DOCUMENTS

CA 2466482 A1 5/2003
EP 0919952 A1 6/1999

(Continued)

OTHER PUBLICATIONS

Canadian Intellectual Property Office, Office Action, dated Sep. 9, 2013 for Canadian Patent Application No. 2,564,861.

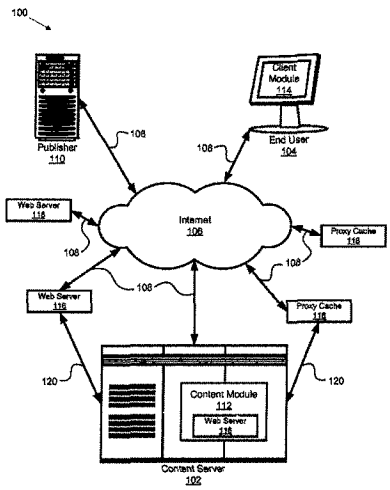
(Continued)

Primary Examiner — Chirag R Patel
(74) *Attorney, Agent, or Firm* — Ingrassia Fisher & Lorenz, P.C.

(57) **ABSTRACT**

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.

20 Claims, 11 Drawing Sheets



Related U.S. Application Data

continuation of application No. 11/673,483, filed on Feb. 9, 2007, now Pat. No. 7,818,444, which is a continuation-in-part of application No. 11/116,783, filed on Apr. 28, 2005, now Pat. No. 8,868,772.

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

(51) **Int. Cl.**

H04N 21/2343	(2011.01)
H04N 21/433	(2011.01)
H04N 21/84	(2011.01)
H04N 21/845	(2011.01)

(52) **U.S. Cl.**

CPC ***H04N21/23439*** (2013.01); ***H04N 21/4331*** (2013.01); ***H04N 21/84*** (2013.01); ***H04N 21/8456*** (2013.01); ***H04L 65/607*** (2013.01); ***H04L 65/80*** (2013.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,687,095	A	11/1997	Haskell et al.
5,768,527	A	6/1998	Zhu et al.
5,953,506	A	9/1999	Kalra et al.
6,091,775	A	7/2000	Hibi et al.
6,091,777	A	7/2000	Guetz et al.
6,122,660	A	9/2000	Baransky et al.
6,185,736	B1	2/2001	Ueno
6,195,680	B1	2/2001	Goldszmidt et al.
6,366,614	B1	4/2002	Pian et al.
6,374,289	B2	4/2002	Delaney et al.
6,389,473	B1	5/2002	Carmel et al.
6,449,719	B1	9/2002	Baker
6,486,803	B1	11/2002	Luby et al.
6,490,627	B1	12/2002	Kalra et al.
6,510,553	B1	1/2003	Hazra
6,574,591	B1	6/2003	Kleiman et al.
6,604,118	B2	8/2003	Kleiman et al.
6,618,752	B1	9/2003	Moore et al.
6,721,723	B1	4/2004	Gibson et al.
6,731,600	B1	5/2004	Patel et al.
6,732,183	B1	5/2004	Graham
6,760,772	B2	7/2004	Zou et al.
6,795,863	B1	9/2004	Doty, Jr.
6,845,107	B1	1/2005	Kitazawa et al.
6,850,965	B2	2/2005	Allen
6,859,839	B1	2/2005	Zahorjan et al.
6,874,015	B2	3/2005	Kaminsky et al.
6,968,387	B2	11/2005	Lanphear
6,976,090	B2	12/2005	Ben-Shaul et al.
7,054,365	B2	5/2006	Kim et al.
7,054,774	B2	5/2006	Batterberry et al.
7,054,911	B1	5/2006	Lango et al.
7,075,986	B2	7/2006	Girod et al.
7,093,001	B2	8/2006	Yang et al.
7,096,271	B1	8/2006	Omoigui et al.
7,099,954	B2	8/2006	Li et al.
7,116,894	B1	10/2006	Chatterton
7,174,385	B2	2/2007	Li
7,194,549	B1	3/2007	Lee et al.
7,240,100	B1	7/2007	Wein et al.
7,260,640	B1	8/2007	Kramer et al.
7,274,740	B2	9/2007	van Beek et al.
7,289,506	B1	10/2007	Hannuksela
7,295,520	B2	11/2007	Lee et al.
7,310,678	B2	12/2007	Gunaseelan et al.
7,325,073	B2	1/2008	Shao et al.
7,328,243	B2	2/2008	Yeager et al.
7,330,908	B2	2/2008	Jungek
7,334,044	B1	2/2008	Allen
7,349,358	B2	3/2008	Hennessey et al.
7,349,976	B1	3/2008	Glaser et al.
7,369,610	B2	5/2008	Xu et al.
7,376,747	B2	5/2008	Hartop
7,391,717	B2	6/2008	Klemets et al.
7,408,984	B2	8/2008	Lu et al.
7,412,531	B1	8/2008	Lango et al.
7,477,688	B1	1/2009	Zhang et al.
7,523,181	B2	4/2009	Swildens et al.
7,536,469	B2	5/2009	Chou et al.
7,546,355	B2	6/2009	Kalnitsky
7,577,750	B2	8/2009	Shen et al.
7,593,333	B2	9/2009	Li et al.
7,599,307	B2	10/2009	Seckni et al.
7,609,652	B2	10/2009	Kellerer et al.
7,640,352	B2	12/2009	Klemets et al.
7,653,735	B2	1/2010	Mandato et al.
7,660,906	B1	2/2010	Armour
7,719,985	B2	5/2010	Lee et al.
7,733,830	B2	6/2010	Curcio et al.
7,760,801	B2	7/2010	Ghanbari et al.
7,779,135	B2	8/2010	Hudson et al.
7,788,395	B2	8/2010	Bowra et al.
7,797,439	B2	9/2010	Cherkasova et al.
7,817,985	B2	10/2010	Moon
7,818,444	B2	10/2010	Brueck et al.
8,036,265	B1	10/2011	Reynolds et al.
8,135,852	B2	3/2012	Nilsson et al.
8,209,429	B2	6/2012	Jacobs et al.
8,370,514	B2	2/2013	Hurst et al.
8,402,156	B2	3/2013	Brueck et al.
8,521,836	B2	8/2013	Kewalramani et al.
8,612,624	B2	12/2013	Brueck et al.
8,868,772	B2	10/2014	Major et al.
8,880,721	B2	11/2014	Hurst et al.
2001/0013128	A1	8/2001	Hagai et al.
2001/0047423	A1	11/2001	Shao et al.
2002/0029274	A1	3/2002	Allen
2002/0073167	A1	6/2002	Powell et al.
2002/0091840	A1	7/2002	Pulier et al.
2002/0097750	A1	7/2002	Gunaseelan et al.
2002/0131496	A1	9/2002	Vasudevan et al.
2002/0144276	A1	10/2002	Radford et al.
2002/0152317	A1	10/2002	Wang et al.
2002/0152318	A1	10/2002	Menon et al.
2002/0161898	A1	10/2002	Hartop et al.
2002/0161911	A1	10/2002	Pinckney, III et al.
2002/0169926	A1	11/2002	Pinckney, III et al.
2002/0174434	A1	11/2002	Lee et al.
2002/0176418	A1	11/2002	Hunt et al.
2002/0178330	A1	11/2002	Schlowsky-Fischer et al.
2002/0188745	A1	12/2002	Hughes et al.
2003/0005455	A1	1/2003	Bowers
2003/0014684	A1	1/2003	Kashyap
2003/0018966	A1	1/2003	Cook et al.
2003/0021166	A1	1/2003	Soloff
2003/0065803	A1	4/2003	Heuvelman
2003/0067872	A1	4/2003	Harrell et al.
2003/0081582	A1	5/2003	Jain et al.
2003/0093790	A1	5/2003	Logan et al.
2003/0107994	A1	6/2003	Jacobs et al.
2003/0135631	A1	7/2003	Li et al.
2003/0140159	A1	7/2003	Campbell et al.
2003/0151753	A1	8/2003	Li et al.
2003/0152036	A1	8/2003	Quigg Brown et al.
2003/0154239	A1	8/2003	Davis et al.
2003/0204519	A1	10/2003	Sirivara et al.
2003/0204602	A1	10/2003	Hudson et al.
2004/0003101	A1	1/2004	Roth et al.
2004/0010613	A1	1/2004	Apostolopoulos et al.
2004/0030547	A1	2/2004	Leaning et al.
2004/0030599	A1	2/2004	Sie et al.
2004/0030797	A1	2/2004	Akinlar et al.
2004/0031054	A1	2/2004	Dankworth et al.
2004/0049780	A1	3/2004	Gee
2004/0054551	A1	3/2004	Ausubel et al.
2004/0071088	A1	4/2004	Curcio et al.
2004/0071209	A1	4/2004	Burg et al.
2004/0083283	A1	4/2004	Sundaram et al.
2004/0093420	A1	5/2004	Gamble
2004/0103444	A1	5/2004	Weinberg et al.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2004/0117427	A1	6/2004	Allen et al.
2004/0143672	A1	7/2004	Padmanabham et al.
2004/0153458	A1	8/2004	Noble et al.
2004/0168052	A1	8/2004	Clisham et al.
2004/0170392	A1	9/2004	Lu et al.
2004/0172478	A1	9/2004	Jacobs et al.
2004/0220926	A1	11/2004	Lamkin et al.
2004/0260701	A1	12/2004	Lehikoinen et al.
2005/0015509	A1	1/2005	Sitaraman
2005/0033855	A1	2/2005	Moradi et al.
2005/0055425	A1	3/2005	Lango et al.
2005/0066063	A1	3/2005	Grigorovitch et al.
2005/0076136	A1	4/2005	Cho et al.
2005/0084166	A1	4/2005	Boneh et al.
2005/0108414	A1	5/2005	Taylor et al.
2005/0120107	A1	6/2005	Kagan et al.
2005/0123058	A1	6/2005	Greenbaum et al.
2005/0185578	A1	8/2005	Padmanabhan et al.
2005/0188051	A1	8/2005	Sneh
2005/0204046	A1	9/2005	Watanabe
2005/0262257	A1	11/2005	Major et al.
2006/0059223	A1	3/2006	Klemets et al.
2006/0075446	A1	4/2006	Klemets et al.
2006/0080718	A1	4/2006	Gray et al.
2006/0130118	A1	6/2006	Damm
2006/0133809	A1	6/2006	Chow et al.
2006/0165166	A1	7/2006	Chou et al.
2006/0168290	A1	7/2006	Doron
2006/0168295	A1	7/2006	Batterberry et al.
2006/0184688	A1	8/2006	Ganguly et al.
2006/0206246	A1	9/2006	Walker
2006/0236219	A1	10/2006	Grigorovitch et al.
2006/0277564	A1	12/2006	Jarman
2007/0024705	A1	2/2007	Richter et al.
2007/0030833	A1	2/2007	Pirzada et al.
2007/0067480	A1	3/2007	Beek et al.
2007/0079325	A1	4/2007	de Heer
2007/0094405	A1	4/2007	Zhang
2007/0204310	A1	8/2007	Hua et al.
2007/0280255	A1	12/2007	Tsang et al.
2008/0028428	A1	1/2008	Jeong et al.
2008/0037527	A1	2/2008	Chan et al.
2008/0046939	A1	2/2008	Lu et al.
2008/0056373	A1	3/2008	Newlin et al.
2008/0104647	A1	5/2008	Hannuksela
2008/0133766	A1	6/2008	Luo
2008/0162713	A1	7/2008	Bowra et al.
2008/0195744	A1	8/2008	Bowra et al.
2008/0195745	A1	8/2008	Bowra et al.
2008/0205291	A1	8/2008	Li et al.
2008/0219151	A1	9/2008	Ma et al.
2008/0222235	A1	9/2008	Hurst et al.
2008/0263180	A1	10/2008	Hurst et al.
2008/0281803	A1	11/2008	Gentric
2009/0043906	A1	2/2009	Hurst et al.
2009/0055471	A1	2/2009	Kozat et al.
2009/0055547	A1	2/2009	Hudson et al.
2009/0210549	A1	8/2009	Hudson et al.
2010/0098103	A1	4/2010	Xiong et al.
2011/0035507	A1	2/2011	Brueck et al.
2013/0151626	A1	6/2013	Hurst et al.

FOREIGN PATENT DOCUMENTS

EP	1202487	A2	5/2002
EP	1298931	A2	4/2003
EP	1395014	A1	3/2004
EP	1670256	A2	6/2006
EP	1777969		4/2007
GB	2367219	A	3/2002
JP	2000-201343		7/2000
JP	2000201343		7/2000
JP	200192752		4/2001
JP	2011004225	A	1/2011
WO	0167264	A1	9/2001
WO	2004025405	A1	3/2004
WO	2004025405	A2	3/2004
WO	2006010113	A2	1/2006

OTHER PUBLICATIONS

Yoshimura, Takeshi et al. "Mobile Streaming Media CDN Enabled by Dynamic SMIL", NTT DoCoMo, Multimedia Laboratories and Hewlett-Packard Laboratories,dated May 7-11, 2002, ACM 1-58113-449-5/02/0005; http://www2002.org/CDROM/refereed/515/.

Japan Patent Office "Interrogation" dated Nov. 6, 2012 for Japanese Patent Appln. No. 2007-511070.

USPTO, Non-Final Office Acton, dated Sep. 13, 2013 for U.S. Appl. No. 13/757,571.

USPTO "Notice of Allowance" mailed Nov. 14, 2012 for U.S. Appl. No. 12/906,940, filed Oct. 18, 2010.

USPTO "Non-Final Office Action" mailed Nov. 23, 2012 for U.S. Appl. No. 11/834,548, filed Aug. 6, 2007.

PCT Notification of Transmittal of the International Search Report and Written Opinion of the International Searching Authority, for PCT/US05/15091, Oct. 29, 2007, 8 pages.

PCT Notification of Transmittal of the International Preliminary Report on Patentability, for PCT/US05/15091, Oct. 29, 2007, 8 pages.

Advisory Action for U.S. Appl. No. 11/116,783, Mailed Mar. 23, 2010, 4 pages.

Advisory Action for U.S. Appl. No. 11/116,783, Mailed May 17, 2010, 3 pages.

Final Office Action for U.S. Appl. No. 11/116,783, Mailed Feb. 22, 2010, 19 pages.

Office Action for U.S. Appl. No. 11/116,783, Mailed May 14, 2008, 19 pages.

Final Office Action for U.S. Appl. No. 11/116,783, Mailed Feb. 20, 2009, 16 pages.

Office Action for U.S. Appl. No. 11/116,783, Aug. 20, 2009, 14 pages.

Advisor Action for U.S. Appl. No. 11/116,783, Mailed Apr. 2, 2009, 3 pages.

Advisory Action for U.S. Appl. No. 11/116,783, Mailed May 12, 2009, 4 pages.

Albanese, Andres , et al., "Priority Encoding Transmission", TR-94-039, Aug. 1994, 36 pages, International Computer Science Institute, Berkeley, California.

Birney, Bill , "Intelligent Streaming", May 2003, Microsoft.

Goyal, Vivek K., "Multiple Description coding: Compression Meets the Network", Sep. 2001, pp. 74-93, IEEE Signal Processing Magazine.

ON2 Technologies, Inc., "TrueMotion VP7 Video Codec", *White Paper*, Document Version 1.0, Jan. 10, 2005, (13 pages).

Pathan, Al-Mukaddim , et al., "A Taxonomy and Survey of Content Delivery Networks", Australia, Feb. 2007. Available at http://www.gridbus.org/reports/CDN-Taxonomy.pdf.

Puri, Rohit , et al., "Multiple Description Source Coding Using Forward Error Correction Codes", Oct. 1999, 5 pages, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California.

Wicker, Stephen B., "Error Control Systems for Digital Communication and Storage", Prentice-Hall, Inc., New Jersey, USA, 1995, Parts 1-6.

USPTO "International Search Report" mailed Dec. 12, 2008; International Appln. No. PCT/US2008/061035, filed Apr. 21, 2008.

Australian Government "Examiner's First Report" dated Oct. 17, 2011; Australian Patent Appln. No. 2011213730.

Korean Intellectual Property Office "Official Notice of Preliminary Rejection" issued Jul. 28, 2011; Korean Patent Appln. No. 10-2006-7025274.

Japan Patent Office "Notice of Rejection Ground" mailed Apr. 26, 2011; Japanese Patent Appln. No. 2007-511070.

(56) **References Cited**

OTHER PUBLICATIONS

Fujisawa, Hiroshi et al. "Implementation of Efficient Access Mechanism for Multiple Mirror-Servers" IPSJ SIG Technical Report, vol. 2004, No. 9 (2004-DPS-116), Jan. 30, 2004, Information Processing Society of Japan, pp. 37-42.
Japan Patent Office "Interrogation" dated Nov. 6, 2012 for Japanese Patent Appin. No. 2007-511070.
USPTO "Final Office Action" mailed Mar. 17, 2011; U.S. Appl. No. 11/834,548, filed Mar. 17, 2011.
USPTO "Examiner's Answer" mailed Feb. 16, 2011; U.S. Appl. No. 11/116,783, filed Apr. 28, 2005.
Major, R. Drew et al. "Reply Brief" filed Apr. 18, 2011; U.S. Appl. No. 11/116,783, filed Apr. 28, 2005.
Liu, Jiangchuan et al. "Adaptive Video Multicast Over the Internet" IEEE Computer Society, 2003.
Rejaie, Reza et al. "Architectural Considerations for Playback of Quality Adaptive Video Over the Internet" University of Southern California, Information Sciences Institute, 1998.
Roy, Sumit et al. "A System Architecture for Managing Mobile Streaming Media Services" Streaming Media Systems Group, Hewlett-Packard Laboratories, 2003.
Xu, Dongyan et al. "On Peer-to-Peer Media Streaming" Department of Computer Sciences, Purdue University, 2002.
Kozamernik, Franc "Media Streaming Over the Internet—An Over of Delivery Technologies" EBU Technical Review, Oct. 2002.
Lienhart, Rainer et al. "Challenges in Distributed Video Management and Delivery" Intel Corporation, EECS Dept., UC Berkeley, 2000-2002.
Zhang, Xinyan et al. "CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming" IEEE 2005.
Guo, Yang "DirectStream: A Directory-Based Peer-To-Peer Video Streaming Service" LexisNexis, Elsevier B.V. 2007.
Supplemental European Search Report dated Sep. 22, 2008.
USPTO International Searching Authority "International Preliminary Report on Patentability," mailed Feb. 18, 2010; International Appln. No. PCT/US2008/009281, filed Aug. 1, 2008.
USPTO International Searching Authority "International Search Report and Written Opinion," mailed Nov. 5, 2008; International Appln. No. PCT/US2008/009281, filed Aug. 1, 2008.
Japanese Patent Office, "Final Office Action" mailed Feb. 28, 2012; Japanese Appln. No. 2007-511070.
"The meaning of performance factor—English-Japanese Weblio Dictionary", [online], Feb. 24, 2012, [searched on Feb. 24, 2012], The Internet <URL:http://ejje.weblio.jp/content/performance+factor>.
Tsuru, et al., "Recent evolution of the Internet measurement and inference techniques", IEICE Technical Report, vol. 103, No. 123, pp. 37-42, Jun. 12, 2003.
USPTO "Final Office Action" mailed Mar. 5, 2010; U.S. Appl. No. 11/737,669, filed Apr. 19, 2007.
USPTO "Non-Final Office Action" mailed Jul. 24, 2009; U.S. Appl. No. 11/737,669, filed Apr. 19, 2007.
USPTO "Non-Final Office Action" mailed Jun. 28, 2010; U.S. Appl. No. 11/834,548, filed Aug. 6, 2007.
USPTO International Searching Authority "International Preliminary Report on Patentability," mailed Oct. 29, 2009; International Appln. No. PCT/US2008/061035, filed Apr. 21, 2008.
Clement, B. "Move Networks closes $11.3 Million on First Round VC Funding," Page One PR, Move Networks, Inc. Press Releases, Feb. 7, 2007, http://www.move.tv/press/press20070201.html.
Liu, Jiangchuan et al. "Opportunities and Challenged of Peer-to-Peer Internet Video Broadcast," School of Computing Science, Simon Fraser University, British Columbia, Canada.
Move Networks, Inc. "The Next Generation Video Publishing System," Apr. 11, 2007; http://www.movenetworks.com/wp-content/uploads/move-networks-publishing-system.pdf.
USPTO "Notice of Allowance" dated Jun. 24, 2013 for U.S. Appl. No. 11/834,548.
European Patent Office "International Search Report and Written Opinion" mailed Oct. 27, 2007 for International Appln. No. PCT/US2005/015091.
European Patent Office "International Preliminary Report on Patentability" mailed Oct. 29, 2007 for International Appln. No. PCT/US2005/015091.
USPTO "Final Office Action" mailed Feb. 22, 2010 for U.S. Appl. No. 11/116,783.
USPTO "Non-Final Office Action" mailed May 14, 2008 for U.S. Appl. No. 11/116,783.
USPTO "Final Office Action" mailed Feb. 20, 2009 for U.S. Appl. No. 11/116,783.
European Patent Office "Supplemental European Search Report" mailed Sep. 30, 2008.
USPTO "Non-Final Office Action" mailed Aug. 20, 2009 for U.S. Appl. No. 11/116,783.
Albanese, Andrew et al. "Priority Encoding Transmission", TR-94-039, Aug. 1994, 36 pgs, International Computer Science Institute, Berkeley, CA.
Birney, Bill "Intelligent Streaming", May 2003, Microsoft.
Goyal, Vivek K. "Multiple Description Coding: Compression Meets the Network," Sep. 2001, pp. 74-93, IEEE Signal Processing Magazine.
ON2 Technologies, Inc. "TrueMotion VP7 Video Codec" White Paper, Document Version 1.0, Jan. 10, 2005.
Pathan, Al-Mukaddim et al. "A Taxonomy and Survey of Content Delivery Networks" Australia, Feb. 2007, available at http://www.gridbus.org/reports/CDN-Taxonomy.pdf.
Puri, Rohit et al. "Multiple Description Source Coding Using Forward Error Correction Codes," Oct. 1999, 5 pgs., Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA.
Wicker, Stephen B. "Error Control Systems for Digital Communication and Storage," Prentice-Hall, Inc., New Jersey, USA, 1995, parts 1-6.
USPTO "Advisory Action" mailed May 12, 2009 for U.S. Appl. No. 11/116,783.
USPTO "Advisory Action" mailed Apr. 2, 2009 for U.S. Appl. No. 11/116,783.
USPTO "Advisory Action" mailed May 17, 2010 for U.S. Appl. No. 11/116,783.
USPTO "Advisory Action" mailed Mar. 23, 2010 for U.S. Appl. No. 11/116,783.
USPTO "Notice of Allowance" mailed Oct. 5, 2012 for U.S. Appl. No. 12/075,475, filed Mar. 10, 2008.
Japanese Patent Office "Final Office Action" mailed Feb. 28, 2012 for Japanese Appln. No. 2007-511070.
Tsuru, et al. "Recent evolution of the Internet measurement and inference techniques", IEICE Technical Report, vol. 103, No. 123, pp. 37-42, Jun. 12, 2003.
USPTO "International Search Report" mailed Dec. 12, 2008 for International Appln. No. PCT/US2008/061035, filed Apr. 21, 2008.
Australian Government "Examiner's First Report" dated Oct. 17, 2011 for Australian Patent Appln. No. 2011213730.
Korean Intellectual Property Office "Official Notice of Preliminary Rejection" issued Jul. 28, 2011 for Korean Patent Appln. No. 10-2006-7025274.
Japan Patent Office "Notice of Rejection Ground" mailed Apr. 26, 2011 for Japanese Patent Appln. No. 2007-511070.
Fujisawa, Hiroshi et al. "Implementaton of Efficient Access Mechanism for Multiple Mirror-Servers" IPSJ SIG Technical Report, vol. 2004, No. 9 (2004-DPS-116), Jan. 30, 2004, Information Processing Society of Japan, pp. 37-42.
USPTO "Final Office Action" mailed Feb. 23, 2011 for U.S. Appl. No. 12/075,475, filed Mar. 10, 2008.
USPTO "Final Office Action" mailed Mar. 17, 2011 for U.S. Appl. No. 11/834,548, filed Mar. 17, 2011.
USPTO "Examienr's Answer" mailed Feb. 16, 2011 for U.S. Appl. No. 11/116,783, filed Apr. 28, 2005.
Major, R. Drew et al. "Reply Brief" filed Apr. 18, 2011 for U.S. Appl. No. 11/116,783, filed Apr. 28, 2005.

(56) **References Cited**

OTHER PUBLICATIONS

Kozamerink, Franc "Media Streaming Over the Internet—An Over of Delivery Technologies" EBU Technical Review, Oct. 2002.
Lienhart, Rainer et al. "Challenges in Distributed Video Management and Delivery" Intel Corporation, EES Dept., UC Berkeley, 2000-2002.
U.S. Patent and Trademark Office, Notice of Allowance, mailed Jun. 24, 2014 for U.S. Appl. No. 13/757,571.
U.S. Patent and Trademark Office, Non-Final Office Action, mailed Sep. 13, 2013 for U.S. Appl. No. 13/757,571.
European Patent Office, Extended Search Report, dated Jul. 10, 2014 for European Application No. 12154559.4.
Nguyen, T. et al., Multiple Sender Distributed Video Streaming, IEEE Transactinos on Multimedia, IEEE Service Center, Piscataway, NJ, US, vol. 6, No. 2, Apr. 1, 2004, pp. 315-326, XP011109142, ISSN: 1520-9210, DOI: 10.1109/TMM,2003.822790.
USPTO, "Notice of Allowance and Fee(s) Due" dated Aug. 12, 2014 for U.S. Appl. No. 11/116,783.
Canadian Intellectual Property Office, Office Action, dated Sep. 10, 2014 for Canadian Patent Application No. 2,564,861.
U.S. Patent and Trademark Office, Non-Final Office Action, dated Oct. 24, 2014 for U.S. Appl. No. 14/222,245.

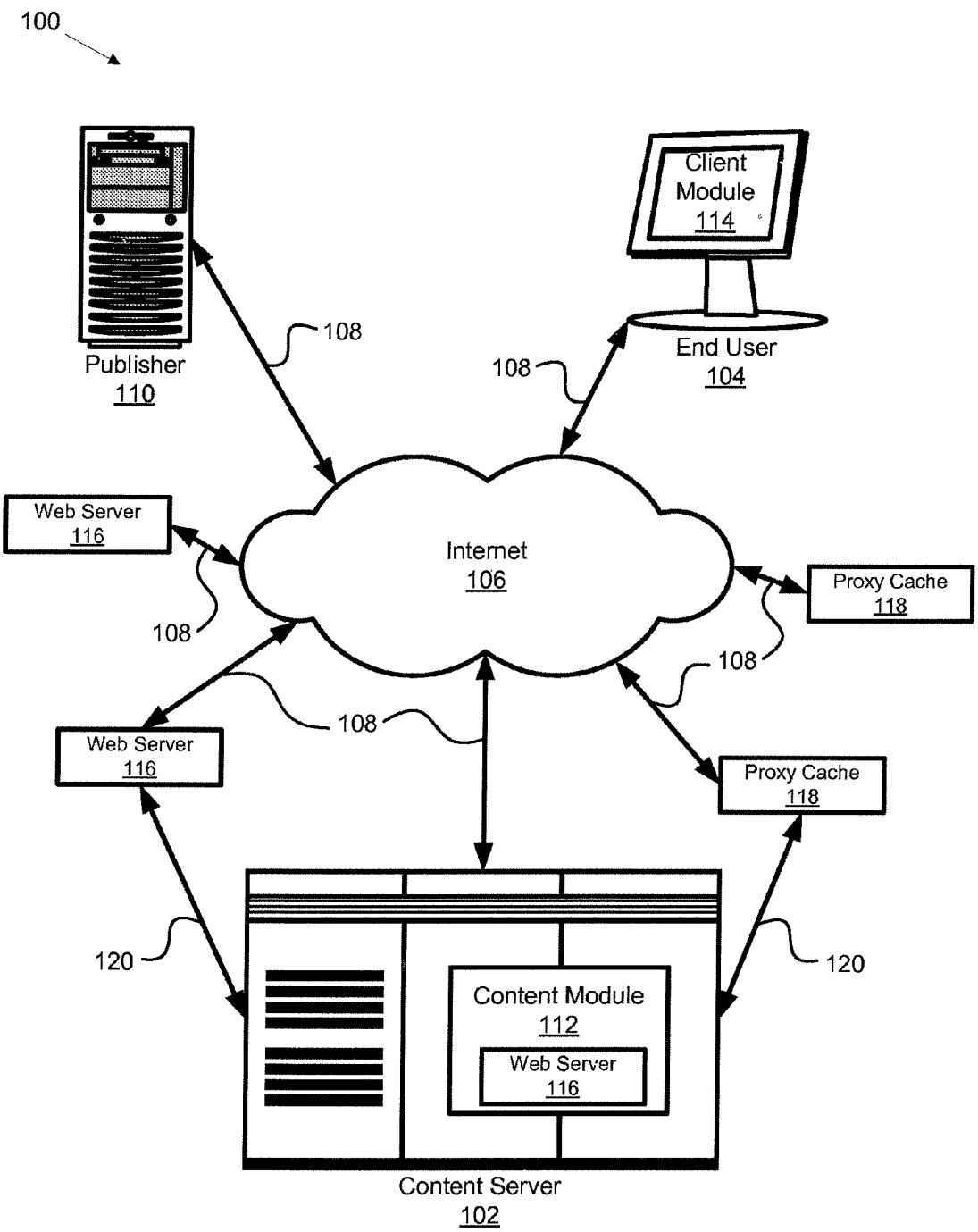


FIG. 1

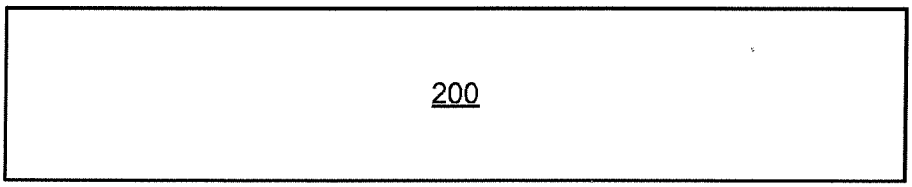


FIG. 2a

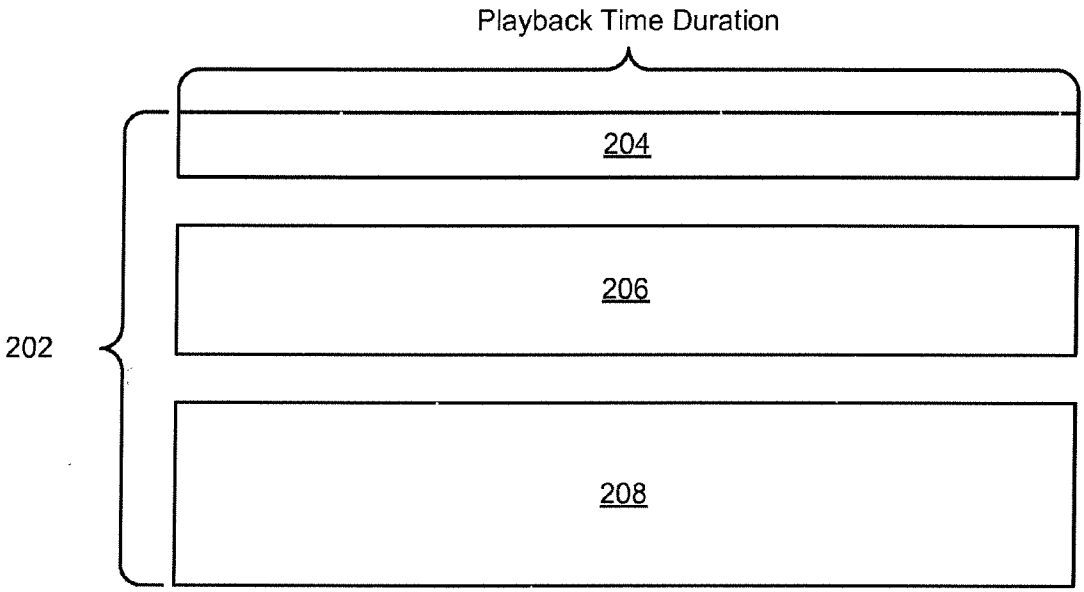


FIG. 2b

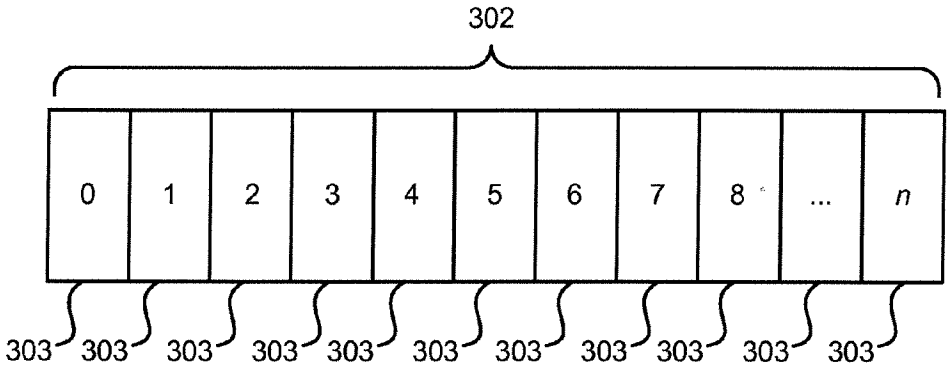


FIG. 3a

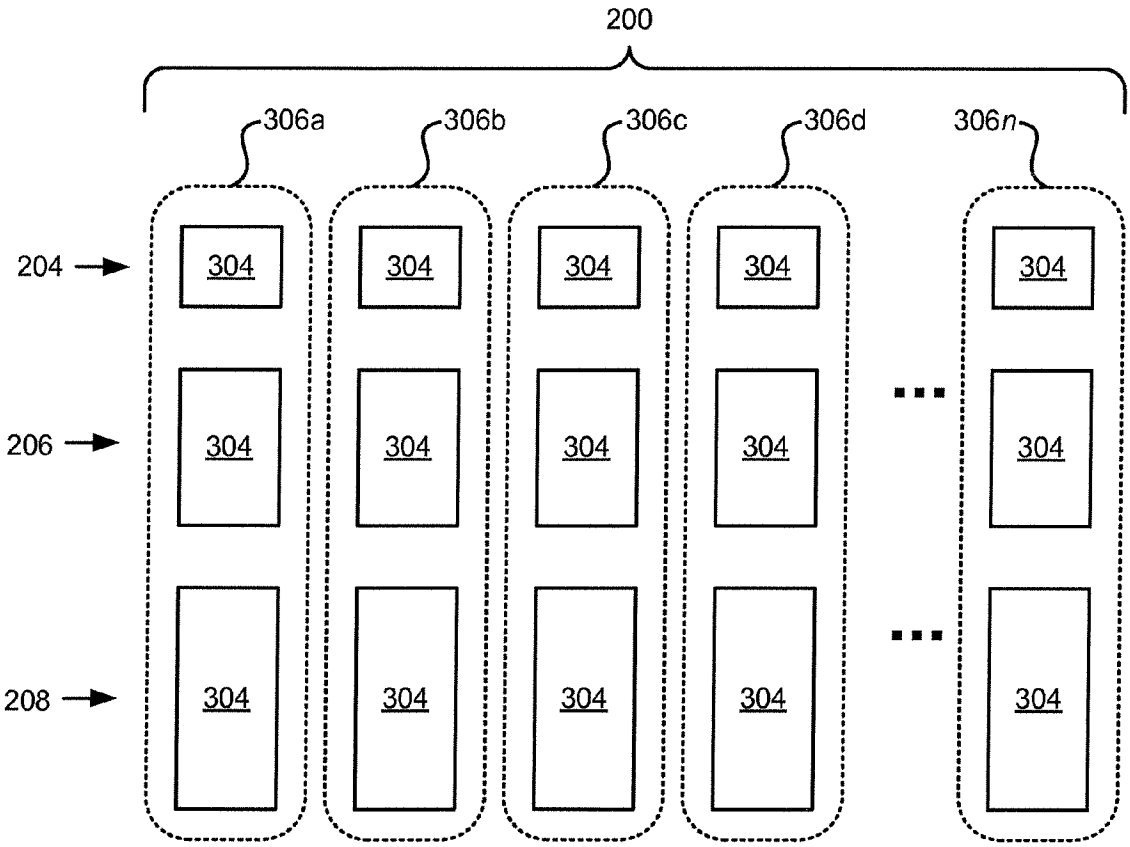


FIG. 3b

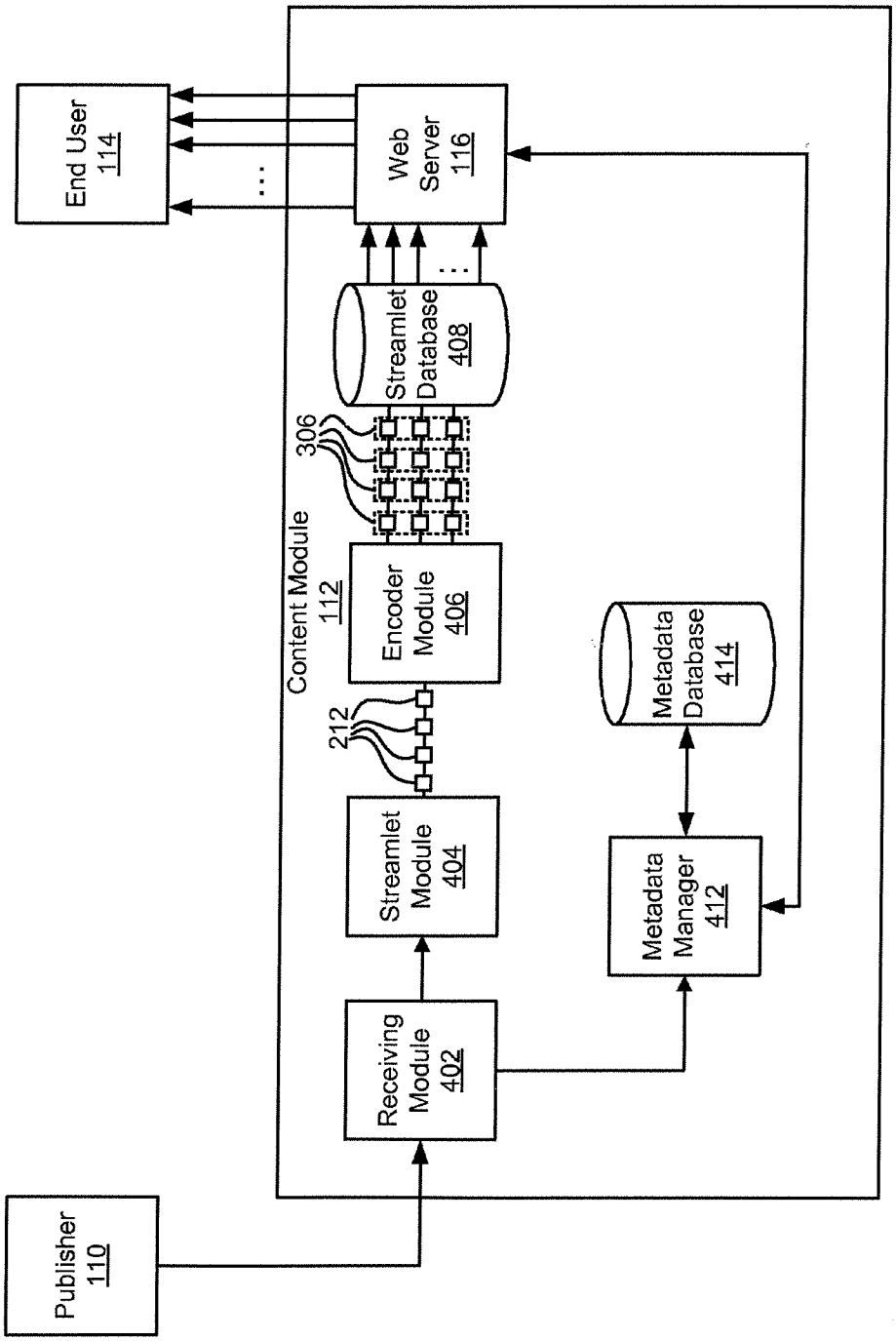


FIG. 4

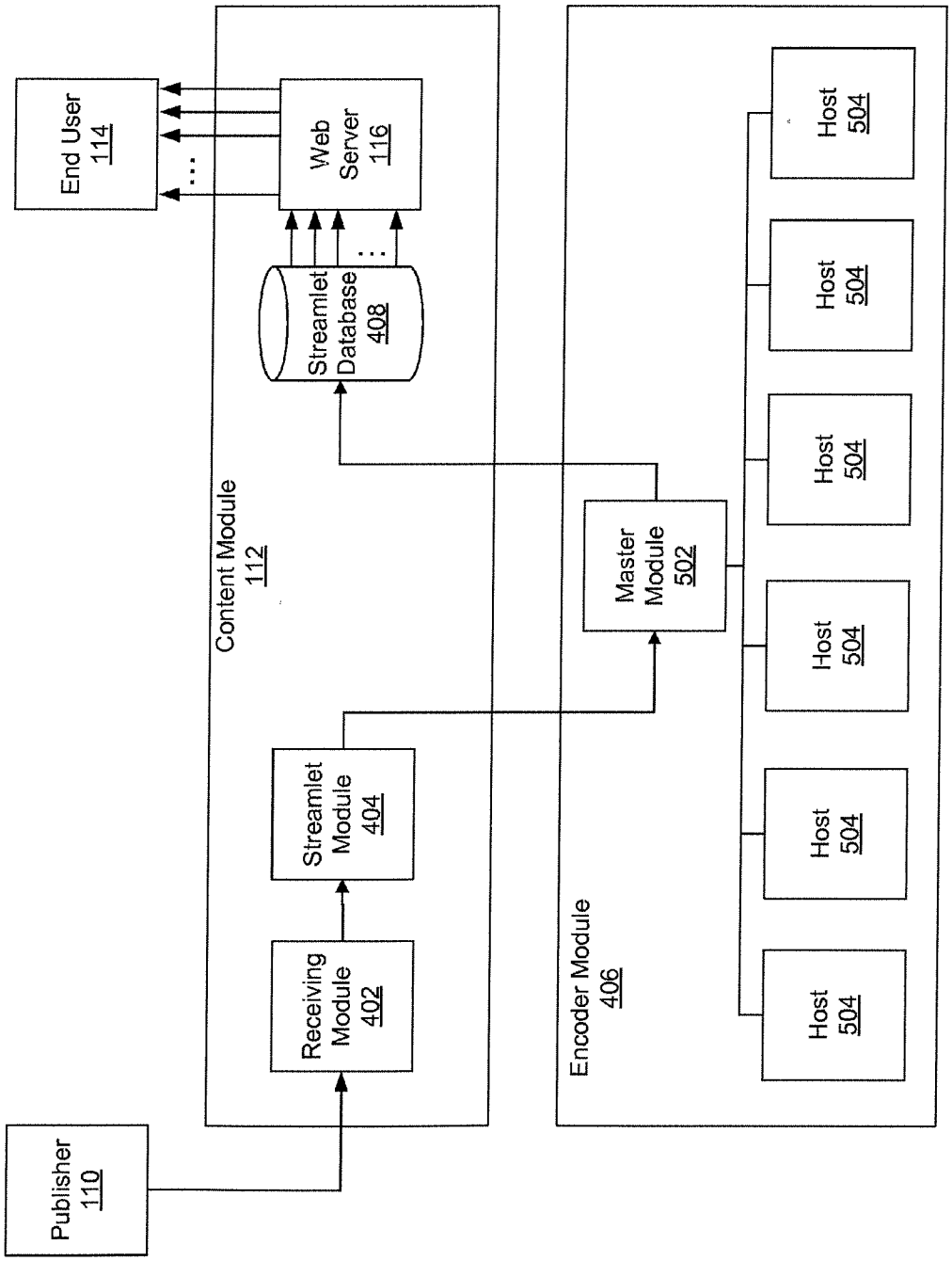


FIG. 5a

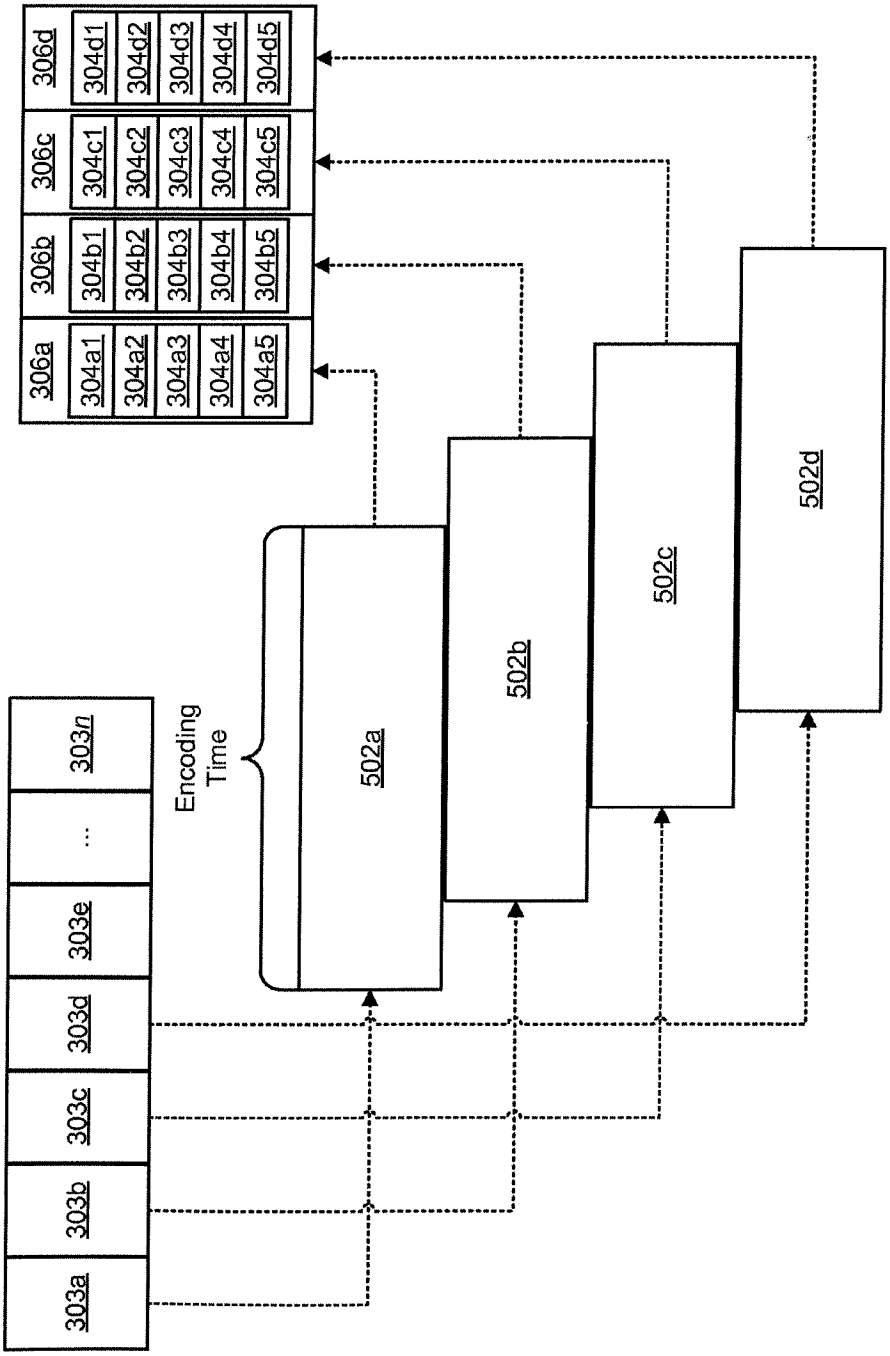


FIG. 5b

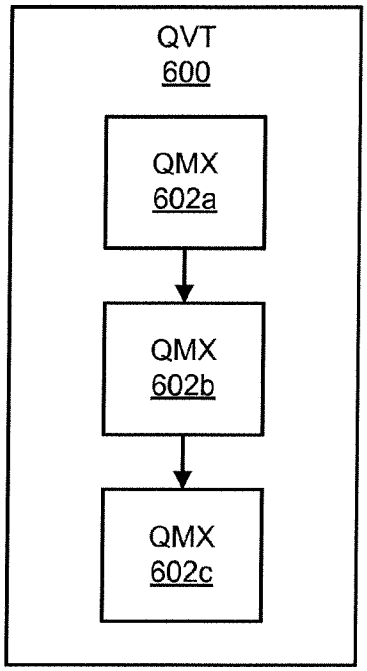


FIG. 6a

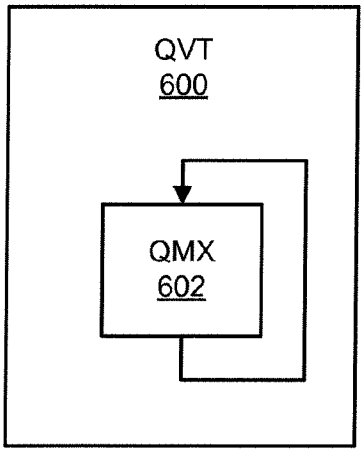


FIG. 6b

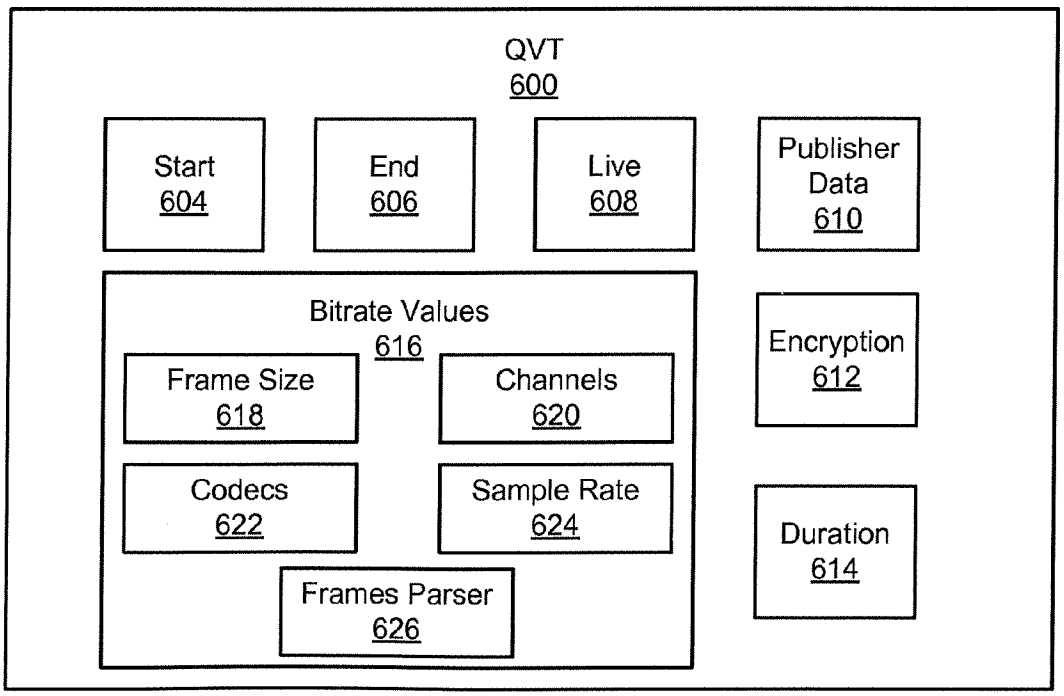


FIG. 6c

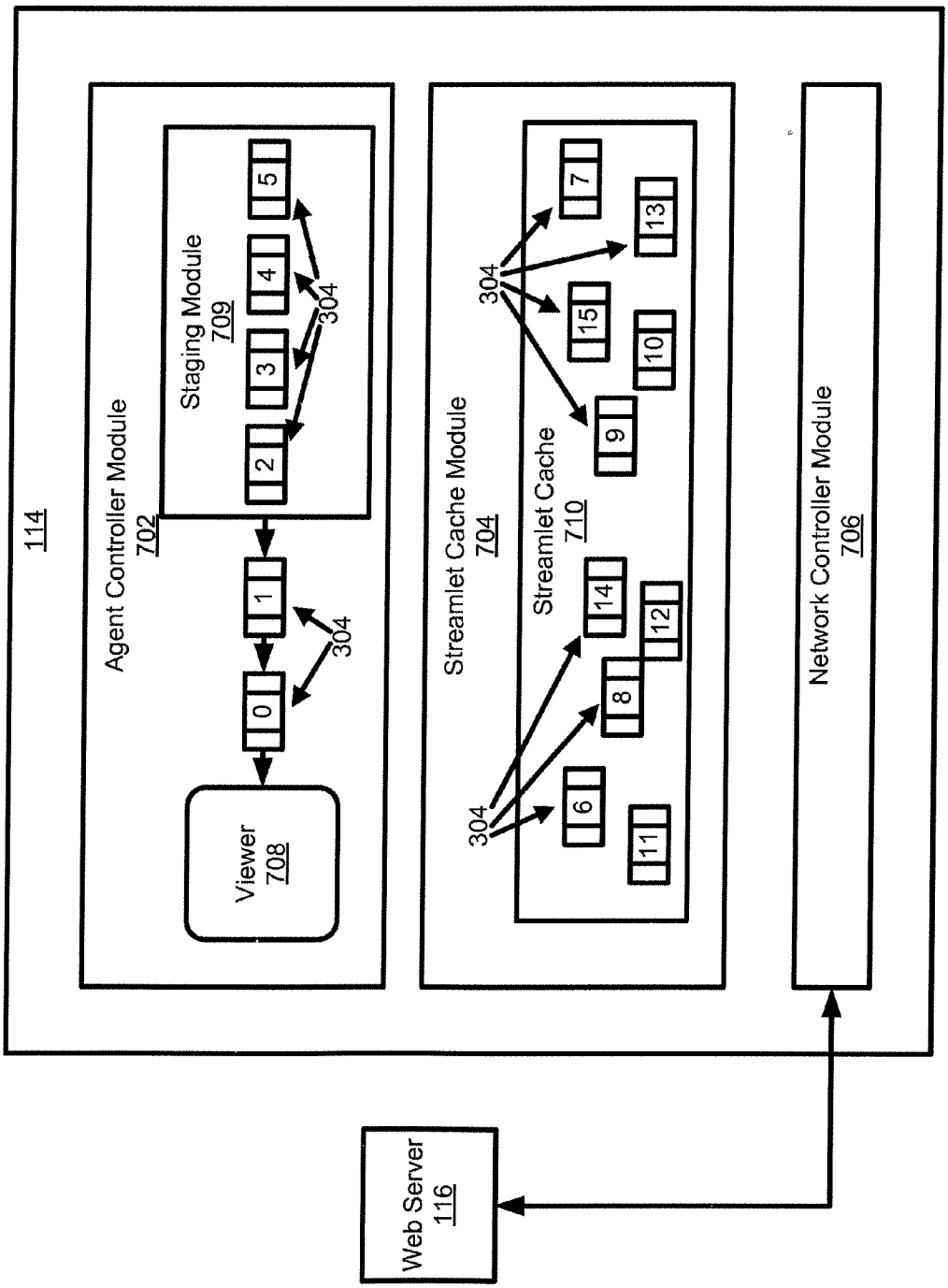


FIG. 7

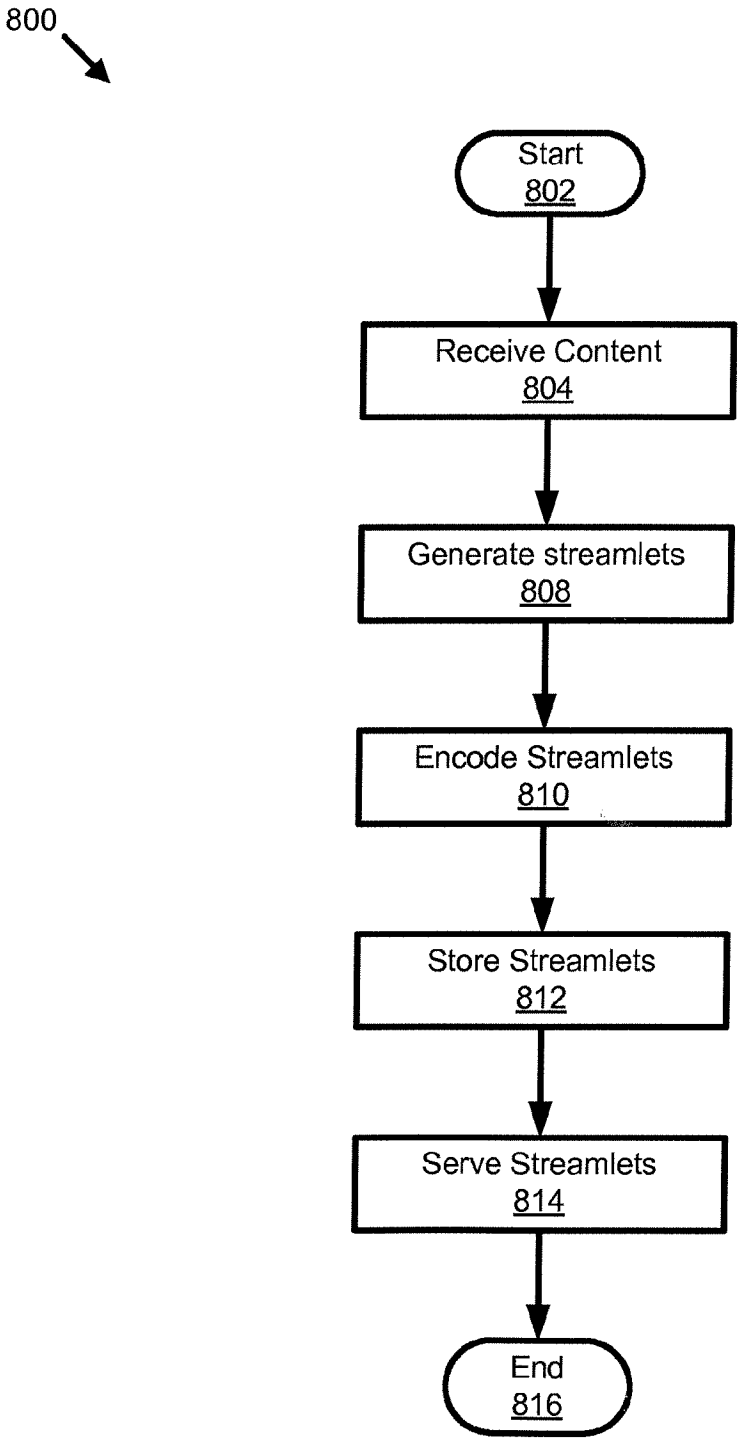


FIG. 8

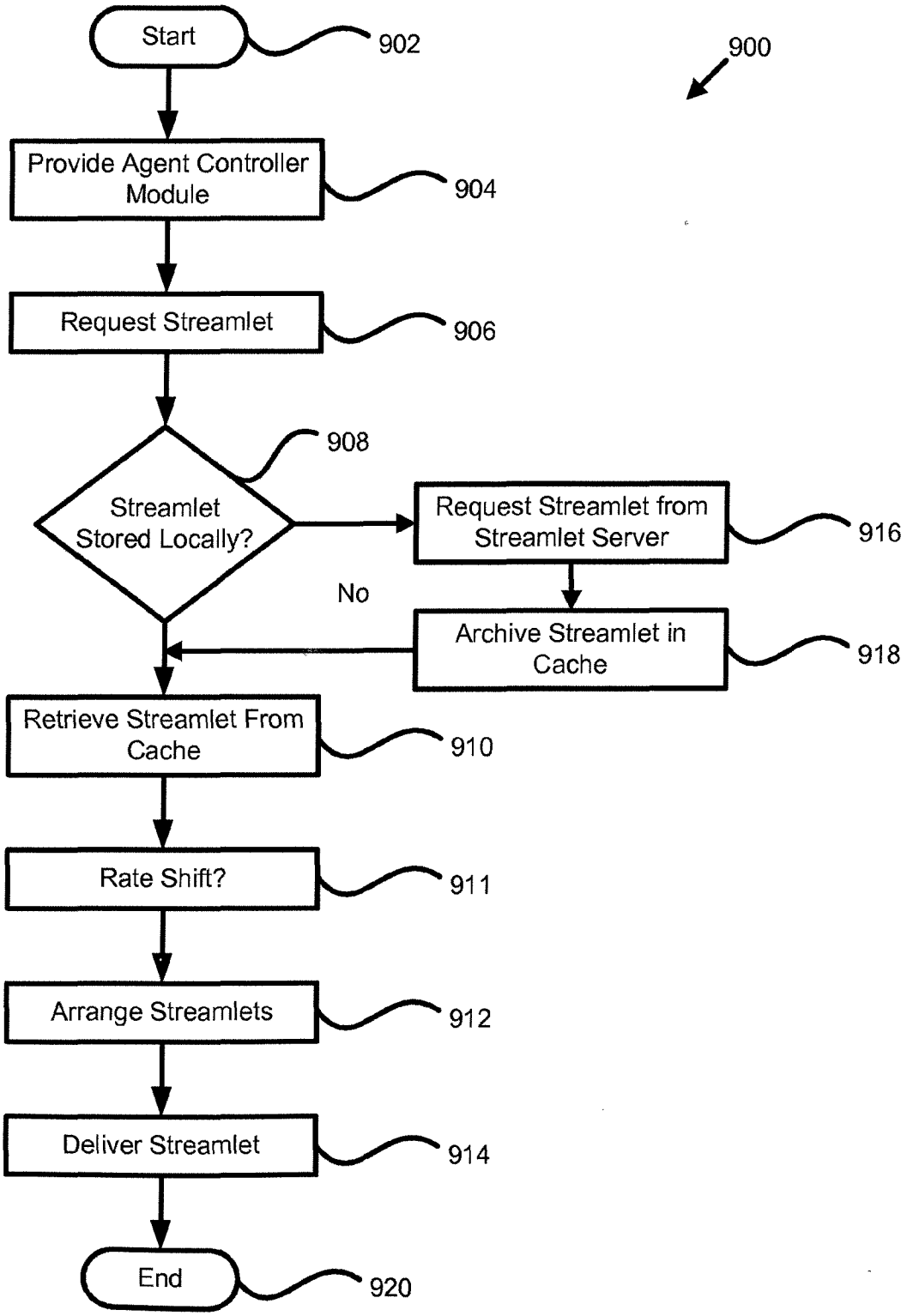


FIG. 9

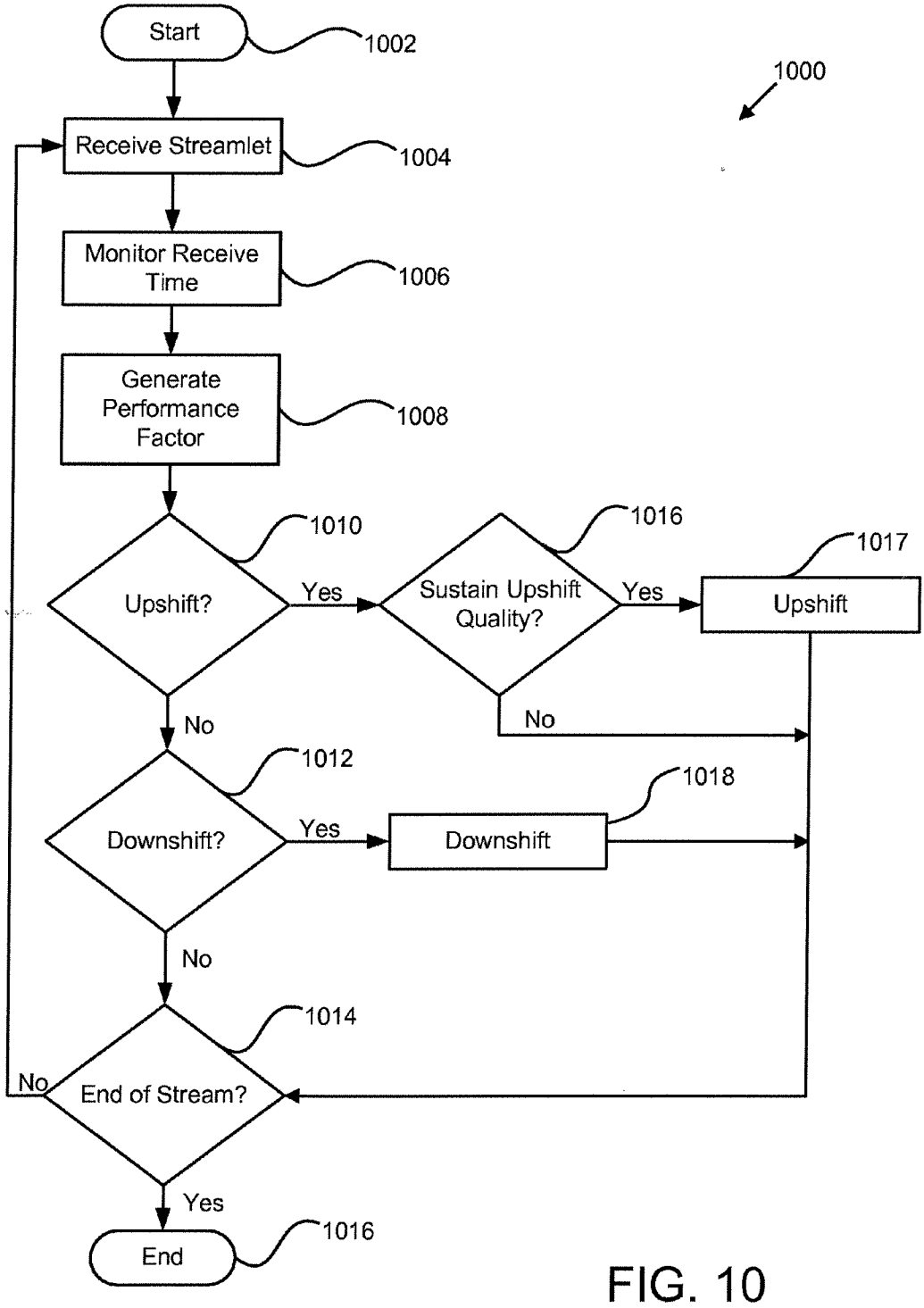


FIG. 10

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

CROSS-REFERENCES TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 11/673,483, filed on Feb. 9, 2007, which is a continuation-in-part of application Ser. No. 11/116,783, filed on Apr. 28, 2005, which claims the benefit of U.S. Provisional Application No. 60/566,831, filed on Apr. 30, 2004, all of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

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.

2. Description of the Related Art

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.

"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.

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.

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.

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.

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.

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.

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.

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

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.

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.

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 0.1 and 5 seconds.

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.

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

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.

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.

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.

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.

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.

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.

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

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:

FIG. **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;

FIG. **2***a* is a schematic block diagram graphically illustrating one embodiment of a media content file;

FIG. **2***b* is a schematic block diagram illustrating one embodiment of a plurality of streams having varying degrees of quality and bandwidth;

FIG. **3***a* is a schematic block diagram illustrating one embodiment of a stream divided into a plurality of source streamlets;

FIG. **3***b* is a schematic block diagram illustrating one embodiment of sets of streamlets in accordance with the present invention;

FIG. **4** is a schematic block diagram illustrating in greater detail one embodiment of the content module in accordance with the present invention;

FIG. **5***a* is a schematic block diagram illustrating one embodiment of an encoder module in accordance with the present invention;

FIG. **5***b* is a schematic block diagram illustrating one embodiment of parallel encoding of streamlets in accordance with the present invention;

FIG. **6***a* is a schematic block diagram illustrating one embodiment of a virtual timeline in accordance with the present invention;

FIG. **6***b* is a schematic block diagram illustrating an alternative embodiment of a VT in accordance with the present invention;

FIG. **6***c* is a schematic block diagram illustrating one embodiment of a QMX in accordance with the present invention;

FIG. **7** is a schematic block diagram graphically illustrating one embodiment of a client module in accordance with the present invention;

FIG. **8** is a schematic flow chart diagram illustrating one embodiment of a method for processing content in accordance with the present invention;

FIG. **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

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

DETAILED DESCRIPTION OF THE INVENTION

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.

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.

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.

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.

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

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.

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 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.

In the depicted embodiment, the system **100** also includes a publisher **110**, and a web server **116**. The publisher **110** may be a creator or distributor of content. For example, if the content to be streamed were a broadcast of a television program, the publisher **110** may be a television or cable network channel such as NBC®, or MTV®. Content may be transferred over the Internet **106** to the content server **102**, where the content is received by a content module **112**. The content module **112** may be configured to receive, process, and store content. In one embodiment, processed content is accessed by a client module **114** configured to play the content on the end user station **104**. In a further embodiment, the client module **114** is configured to receive different portions of a content stream from a plurality of locations simultaneously. For example, the client module **114** may request and receive content from any of the plurality of web servers **116**.

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**.

FIG. **2***a* 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.

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.

FIG. 3*a* 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.

FIG. 3*b* 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 **306***a* encompasses all streamlets having a time index of 00:00. The set **306***a* 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**.

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

FIG. **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.

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**.

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.

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, 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.

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 FIG. 3*b*, may comprise streamlets having an identical time index and duration, and a unique bitrate. As with FIG. 3*b*, 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.

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.

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.

Once the streamlets **304** have been received and processed, the client module **114** may request streamlets **304** using HTTP from the web server **116**. Using a standard protocol such as HTTP eliminates the need for network administrators to configure firewalls to recognize and pass through network traffic for a new, specialized protocol. Additionally, since the client module **114** initiates the request, the web server **116** is only required to retrieve and serve the requested streamlet **304**. In a further embodiment, the client module **114** may be configured to retrieve streamlets **304** from a plurality of web servers **116**.

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**

FIG. **5***a* 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.

The master module (hereinafter "master") **502** is configured to receive streamlets **303** from the streamlet module **404** and stage the streamlet **303** for processing. In one embodiment, the master **502** may decompress each source streamlet **303** to produce a raw streamlet. As used herein, the term "raw streamlet" refers to a streamlet **303** that is uncompressed or lightly compressed to substantially reduce size with no significant loss in quality. A lightly compressed raw streamlet can be transmitted more quickly and to more hosts. Each host **504** is coupled with the master **502** and configured to receive a raw streamlet from the master **502** for encoding. The hosts **504**, in one example, generate a plurality of streamlets **304** having identical time indices and durations, and varying bitrates. Essentially each host **504** may be configured to generate a set **306** from the raw streamlet **503** sent from the master **502**. Alternatively, each host **504** may be dedicated to producing a single bitrate in order to reduce the time required for encoding.

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.

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.

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.

The present invention, however, segments the content file **200** into source streamlets **303** and the two-pass encoding to a plurality of streams **202** may be performed on each corresponding raw streamlet without waiting for a TV show to end, for example. As such, the content module **112** is capable of streaming the streamlets over the Internet shortly after the content module **112** begins capture of the content file **200**. The delay between a live broadcast transmitted from the publisher **110** and the availability of the content depends on the computing power of the hosts **504**.

FIG. **5***b* 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 FIG. **4**) begins to capture the content file and the streamlet module **404** generates a first streamlet **303***a* and passes the streamlet to the encoding module **406**. The encoding module **406** may take 10 seconds, for example, to generate the first set **306***a* of streamlets **304***a* (**304***a***1**, **304***a***2**, **304***a***3**, etc. represent streamlets **304** of different bitrates). FIG. **5***b* 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**.

During the 10 seconds required to encode the first streamlet **303***a*, the streamlet module **404** has generated five additional 2-second streamlets **303***b*, **303***c*, **303***d*, **303***e*, **303***f*, for encoding and the master **502** has prepared and staged the corresponding raw streamlets. Two seconds after the first set **306***a* is available the next set **306***b* 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.

A system as described above beneficially enables multi-pass encoding of live events. Multi-pass encoding systems of the prior art require that the entire content be captured (or be complete) because in order to perform multi-pass encoding the entire content must be scanned and processed more than once. This is impossible with prior art systems because content from a live event is not complete until the event is over. As such, with prior art systems, multi-pass encoding can only be performed once the event is over. Streamlets, however, may be encoded as many times as is deemed necessary. Because the streamlet is an encapsulated media object of 2 seconds (for example), multi-pass encoding may begin on a live event once the first streamlet is captured. Shortly after multi-pass encoding of the first streamlet **303***a* is finished, multi-pass encoding of the second streamlet **303***b* finishes, and as such multi-pass encoding is performed on a live event and appears live to a viewer.

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.

The parallel offset pipeline approach described with reference to FIG. **5***b* 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.

Returning now to FIG. **5***a*, as depicted, the master **502** and the hosts **504** may be located within a single local area network, or in other terms, the hosts **504** may be in close physical proximity to the master **502**. Alternatively, the hosts **504** may receive encoding jobs from the master **502** over the Internet or other communications network. For example, consider a live sports event in a remote location where it would be difficult to setup multiple hosts. In this example, a master performs no encoding or alternatively light encoding before publishing the streamlets online. The hosts **504** would then retrieve those streamlets and encode the streamlets into the multiple bitrate sets **306** as described above.

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.

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.

FIG. **6***a* 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 **602***a* followed by QMX **602***b* and QMX **602***c*. As such, the publisher may define a broadcast schedule in a manner similar to a television station.

FIG. **6***b* 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.

FIG. **6***c* is a schematic block diagram illustrating one embodiment of a QMX **602** in accordance with the present invention. In one embodiment, the QMX **602** contains a multitude of information generated by the content module **112** configured to describe the content file **200**. Examples of information include, but are not limited to, start index **604**, end index **606**, whether the content is live **608**, proprietary publisher data **610**, encryption level **612**, content duration **614** and bitrate values **616**. The bitrate values **616** may include frame size **618**, audio channel **620** information, codecs **622** used, sample rate **624**, and frames parser **626**.

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.

FIG. **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.

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.

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 FIG. **10**.

The agent controller module **702** may be configured to receive user commands from the viewer **708**. Such commands may include play, fast forward, rewind, pause, and stop. In one embodiment, the agent controller module **702** requests streamlets **304** from the streamlet cache module **704** and arranges the received streamlets **304** in a staging module **709**. The staging module **709** may be configured to arrange the streamlets **304** in order of ascending playback time. In the depicted embodiment, the streamlets **304** are numbered 0, 1, 2, 3, 4, etc. However, each streamlet **304** may be identified with a unique filename.

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 fur-

ther 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**.

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

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.

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**.

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.

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.

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.

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.

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.

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.

FIG. **8** is a schematic flow chart diagram illustrating one embodiment of a method **800** for processing content in accordance with the present invention. In one embodiment the method **800** starts **802**, and the content module **112** receives **804** content from the publisher **110**. Receiving content **804** may comprise receiving **804** a digital copy of the content file **200**, or digitizing a physical copy of the content file **200**. Alternatively, receiving **804** content may comprise capturing a radio, television, cable, or satellite broadcast. Once received **804**, the streamlet module **404** generates **808** a plurality of source streamlets **303** each having a fixed duration. Alternatively, the streamlets **303** may be generated with a fixed file size.

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

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**.

FIG. **9** is a schematic flow chart diagram illustrating one embodiment of a method **900** for viewing a plurality of streamlets in accordance with the present invention. The method **900** starts and an agent controller module **702** is provided **904** and associated with a viewer **708** and provided with a staging module **709**. The agent controller module **702** then requests **906** a streamlet **304** from the streamlet cache module **704**. Alternatively, the agent controller module **702** may simultaneously request **906** a plurality of streamlets **304** the streamlet cache module **704**. If the streamlet is stored **908** locally in the streamlet cache **710**, the streamlet cache module **704** retrieves **910** the streamlet **304** and sends the streamlet to the agent controller module **702**. Upon retrieving **910** or receiving a streamlet, the agent controller module **702** makes **911** a determination of whether or not to shift to a higher or lower quality stream **202**. This determination will be described below in greater detail with reference to FIG. **10**.

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.

Referring now to FIG. **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 FIG. **9**. The method **1000** starts and the agent controller module **702** receives **1004** a streamlet **304** as described above with reference to FIG. **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.

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:

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

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 in, and obtain a performance factor ϕ:

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

The policy determination about whether or not to upshift **1010** playback quality begins by comparing $\phi_{current}$ with a trigger threshold Θ_{up}. If $\phi_{current} \geq \Theta_{up}$, 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 $\phi_{higher} \geq \Theta_{up}$ 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**.

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 $\phi_{current} > \Theta_{down}$ then the stream quality may be adequate, and the agent controller module **702** does not downshift **1018** stream quality. However, if $\phi_{current} \leq \Theta_{down}$, 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.

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. A system comprising a set of one or more servers for adaptive-rate content streaming of videos stored on a storage device for playback on remotely-located end user stations,

wherein a plurality of different copies of a same single video is stored on the storage device, each of the different copies encoded at a different bit rate and each divided into a plurality of streamlets that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire playback duration of the single video, wherein the 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, and wherein each of the streamlets of each of the pluralities is a separate content file that is independently playable by the end user stations, wherein the system comprises:

a web server, to be executed on the set of one or more servers, wherein the web server is configured to:

receive, for a set of one or more initial and sequential ones of the time indexes, at least one streamlet request over a network connection from a requesting one of the end user stations to retrieve the separate content files storing the portions of the single video starting at the set of time indexes from one of the different copies;

retrieve from the storage device the requested content files from the one copy;

send the retrieved content files from the one copy to the requesting one of the end user stations over the network connection;

receive, for each subsequent time index, at least one streamlet request over the network connection from the requesting one of the end user stations to retrieve the separate content files storing the portion of the single video starting at that time index from one of the different copies currently selected, wherein the requesting one of the end user stations selects the currently selected one of the different copies dependent upon ongoing determinations to shift the playback quality to a higher or lower quality one of the different copies, wherein the shifts in playback quality occur at the time indexes;

retrieve from the storage device the requested content files from the currently selected one of the different copies; and

send the retrieved content files from the currently selected one of the different copies to the requesting one of the end user stations over the network connection.

2. The system of claim **1**, wherein the set of servers does not require a specialized protocol for said adaptive-rate content streaming.

3. The system of claim **1**, wherein the storage device comprises a cache server.

4. The system of claim **1**, wherein the web server receives the streamlet requests using a standard protocol.

5. The system of claim **1**, wherein the web server receives the streamlet request as Hypertext Transport Protocol (HTTP) GET requests.

6. The system of claim **1**, wherein at least one of the set of one or more servers is a content server, and wherein the content server includes a content module that comprises:

a receiving module configured to receive the videos;

a streamlet module configured to segment the received videos to generate for each a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video; and

an encoding module configured to encode each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the adaptive-rate content streaming, wherein each of the encoded streamlets in each of the sets is stored as the separate content file, wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the corresponding video, and wherein the separate content files within each of the sets are independently requestable by end user stations.

7. The system of claim **6**, further comprising a streamlet database, wherein the encoding module is further configured to transmit the separate content files each storing one of the encoded streamlets to the streamlet database to be stored, and wherein the web server is coupled to the streamlet database to retrieve the requested ones of the separate content files from the streamlet database to transmit to the requesting one of the end user stations.

8. The system of claim **6**, wherein the content module further comprises:

a metadata module configured to receive metadata from a publisher, wherein the metadata comprises searchable content information; and

a metadata database to store the metadata.

9. The system of claim **8**, wherein the metadata further comprises metadata associated with an end user navigational environment for the single video.

10. The system of claim 6, wherein the encoding module is configured to encode multiple of the plurality of sequential raw streamlets in parallel.

11. The system of claim 6, wherein the encoding module is configured to encode the plurality of sequential raw streamlets at regular intervals using an offset parallel processing scheme.

12. The system of claim 6, wherein the encoding module comprises a master module configured to assign an encoding job to one of a plurality of hosts to generate at least one of the separate content files from at least one of the plurality of sequential raw streamlets in response to an encoding job completion bid received from the one host.

13. The system of claim 6, wherein the encoding module is further configured to multi-pass encode each of the plurality of sequential raw streamlets of at least one of the videos.

14. The system of claim 13, wherein the at least one of the videos is of a live event.

15. A method for streaming from a set of one or more servers videos for playback on a content player on an end user station, the method comprising:

streaming from the set of servers a selected one of the videos for playback on the content player, wherein a storage device stores a plurality of different copies of the same selected video each encoded at a different bit rate and each divided into a plurality of streamlets that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire playback duration of the selected video, wherein the 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 selected video, and wherein each of the streamlets of each of the pluralities is a separate content file that is independently playable by the end user stations, and wherein said streaming comprises:

receiving, for a set of one or more initial and sequential ones of the time indexes, at least one streamlet request over a network connection from a requesting one of the end user stations to retrieve the separate content files storing the portions of the single video starting at the set of time indexes from one of the different copies;

retrieving from the storage device the requested content files with the set of time indexes from the one copy;

sending the retrieved content files from the one copy to the requesting one of the end user stations over the network connection;

receiving, for each subsequent time index, at least one streamlet request over the network connection from the requesting one of the end user stations to retrieve the separate content tiles storing the portion of the single video starting at that time index from one of the different copies currently selected, wherein the requesting one of the end user stations selects the currently selected one of the different copies dependent upon successive determinations to shift the playback quality to a higher or lower quality one of the different copies, wherein the shifts in playback quality occur at the time indexes;

retrieving from the storage device the requested content files with the time index from the currently selected one of the different copies; and

sending the retrieved content files from the currently selected one of the different copies to the requesting one of the end user stations over the network connection.

16. A method executable by a content player on an end user device to obtain a stream of a selected video program for playback by the content player, the method comprising:

requesting the stream of the selected video program via a network connection to a video server, wherein the video server accesses a plurality of different copies of the same selected video each encoded at a different bit rate and each divided into a plurality of segments that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire playback duration of the selected video, wherein the time indexes of the segments are the same for the different copies such that the segments with the same time indexes from the different copies independently yield the same portions of the selected video, and wherein each of the segments of each of the pluralities is a separate content file that is independently playable by the end user device, wherein the requesting comprises the content player placing, for a set of sequential ones of the time indexes, segment requests over the network connection to the video server to thereby retrieve the separate segments from at least one of the different copies storing the portions of the single video according to the set of time indexes;

receiving the separate segments from the video server at the content player via the network connection; and

adapting subsequent segment requests placed by the content player to the video server based upon successive determinations by the content player to shift the playback quality to a higher or lower quality one of the different copies of the same selected video, wherein the shifts in playback quality occur at the time indexes.

17. The method of claim 16 further comprising playing back the requested stream by the content player to thereby render the selected video program to a viewer.

18. The method of claim 17, wherein the playing back comprises the content player staging the separate segments received by the content player according to the time index.

19. The method of claim 16 wherein the adapting comprises comparing a performance ratio to a threshold value, and wherein the content player requests segments from a copy of the same selected video that were encoded at a higher bitrate when the performance ratio exceeds the threshold value.

20. The method of claim 16, wherein the requesting comprises the content player placing the segment requests as Hypertext Transport Protocol (HTTP) GET requests.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 9,071,668 B2
APPLICATION NO. : 14/106051
DATED : June 30, 2015
INVENTOR(S) : Brueck et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

Item (72) Inventors should read:
--David F. Brueck, Saratog Springs, UT (US); Mark B. Hurst, Cedar Hills, UT (US);
R. Drew Major, Orem, UT (US)--

Signed and Sealed this
First Day of May, 2018

Andrei Iancu

Andrei Iancu
Director of the United States Patent and Trademark Office

EXHIBIT D

U 7701579

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

November 21, 2018

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF:

U.S. PATENT: ***9,407,564***
ISSUE DATE: ***August 02, 2016***

By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office

SYLVIA HOLLEY
Certifying Officer

US009407564B2

(12) **United States Patent**
Major et al.

(10) **Patent No.: US 9,407,564 B2**
(45) **Date of Patent: *Aug. 2, 2016**

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

(71) Applicant: **ECHOSTAR TECHNOLOGIES L.L.C.**, Englewood, CO (US)

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

(73) Assignee: **EchoStar Technologies L.L.C.**, Englewood, CO (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/516,303**

(22) Filed: **Oct. 16, 2014**

(65) **Prior Publication Data**

US 2015/0039782 A1 Feb. 5, 2015

Related U.S. Application Data

(63) Continuation of application No. 11/116,783, filed on Apr. 28, 2005, now Pat. No. 8,868,772.

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

(51) **Int. Cl.**
H04L 12/853 (2013.01)
H04L 12/825 (2013.01)
(Continued)

(52) **U.S. Cl.**
CPC ***H04L 47/25*** (2013.01); ***H04L 65/60*** (2013.01); ***H04N 21/25808*** (2013.01);
(Continued)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,535,355 A 8/1985 Arn et al.
5,168,356 A 12/1992 Acampora et al.
(Continued)

FOREIGN PATENT DOCUMENTS

CA 2466482 5/2003
EP 0 711 077 A2 5/1996
(Continued)

OTHER PUBLICATIONS

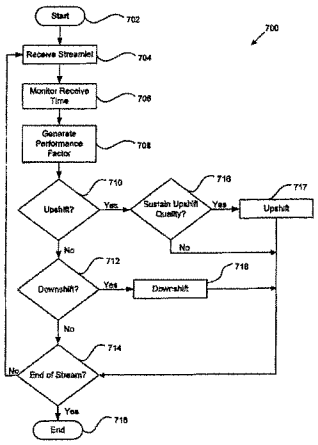
Bill Birney, Intellegent Streaming, May 2003, all pages.*
(Continued)

Primary Examiner — Ninos Donabed
(74) *Attorney, Agent, or Firm* — Ingrassia Fisher & Lorenz, P.C.

(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.

16 Claims, 7 Drawing Sheets



(51) **Int. Cl.**

H04N 21/258	(2011.01)
H04N 21/2662	(2011.01)
H04N 21/643	(2011.01)
H04N 21/647	(2011.01)
H04N 21/84	(2011.01)
H04N 21/845	(2011.01)
H04L 29/06	(2006.01)

(52) **U.S. Cl.**

CPC ***H04N 21/2662*** (2013.01); ***H04N 21/643*** (2013.01); ***H04N 21/64769*** (2013.01); ***H04N 21/64792*** (2013.01); ***H04N 21/84*** (2013.01); ***H04N 21/845*** (2013.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,267,334 A 11/1993 Normille et al.
5,404,446 A 4/1995 Bowater et al.
5,768,527 A 6/1998 Zhu et al.
5,841,432 A 11/1998 Carmel et al.
5,953,506 A 9/1999 Kalra et al.
6,091,775 A 7/2000 Hibi et al.
6,091,777 A 7/2000 Guetz et al.
6,122,660 A 9/2000 Baransky et al.
6,185,736 B1 2/2001 Ueno
6,195,680 B1 2/2001 Goldszmidt et al. 709/203
6,366,614 B1 4/2002 Pian et al. 375/240.02
6,374,289 B2 4/2002 Delaney et al.
6,389,473 B1 5/2002 Carmel et al.
6,449,719 B1 9/2002 Baker
6,486,803 B1 11/2002 Luby et al.
6,490,627 B1 12/2002 Kalra et al.
6,510,553 B1 1/2003 Hazra
6,552,227 B2 4/2003 Mendelovici et al.
6,574,591 B1 6/2003 Kleiman et al.
6,604,118 B2 8/2003 Kleiman et al.
6,618,752 B1 9/2003 Moore et al.
6,654,790 B2 11/2003 Ogle et al.
6,675,199 B1 1/2004 Mohammed et al.
6,697,072 B2 2/2004 Russell et al.
6,721,723 B1 4/2004 Gibson et al.
6,731,600 B1 5/2004 Patel et al.
6,732,183 B1 5/2004 Graham
6,760,772 B2 7/2004 Zou et al.
6,792,449 B2 9/2004 Colville et al.
6,795,863 B1 9/2004 Doty, Jr.
6,801,947 B1 10/2004 Li
6,845,107 B1 1/2005 Kitazawa et al.
6,850,965 B2 2/2005 Allen
6,859,839 B1 2/2005 Zahorjan et al.
6,874,015 B2 3/2005 Kaminsky et al.
6,885,471 B1 4/2005 Minowa et al.
6,968,387 B2 11/2005 Lanphear
6,976,090 B2 12/2005 Ben-Shaul et al.
7,031,700 B1 4/2006 Weaver et al.
7,046,805 B2 5/2006 Fitzhardinge et al.
7,054,365 B2 5/2006 Kim et al.
7,054,774 B2 5/2006 Batterberry et al.
7,054,911 B1 5/2006 Lango et al.
7,075,986 B2 7/2006 Girod et al.
7,093,001 B2 8/2006 Yang et al.
7,096,271 B1 8/2006 Omoigui et al.
7,099,954 B2 8/2006 Li et al.
7,111,044 B2 9/2006 Lee
7,116,894 B1 10/2006 Chatterton
7,124,164 B1 10/2006 Chemtob
7,174,385 B2 2/2007 Li
7,176,957 B2 2/2007 Ivashin et al.
7,177,642 B2 2/2007 Sanchez Herrero et al.
7,190,670 B2 3/2007 Varsa et al.
7,194,549 B1 3/2007 Lee et al.
7,240,100 B1 7/2007 Wein et al.
7,260,640 B1 8/2007 Kramer et al.
7,274,740 B2 9/2007 van Beek et al.
7,295,520 B2 11/2007 Lee et al.
7,310,678 B2 12/2007 Gunaseelan et al.
7,313,236 B2 12/2007 Amini et al.
7,325,073 B2 1/2008 Shao et al.
7,328,243 B2 2/2008 Yeager et al.
7,330,908 B2 2/2008 Jungck
7,334,044 B1 2/2008 Allen
7,349,358 B2 3/2008 Hennessey et al.
7,349,976 B1 3/2008 Glaser et al.
7,369,610 B2 5/2008 Xu et al.
7,376,747 B2 5/2008 Hartop
7,391,717 B2 6/2008 Klemets et al.
7,408,984 B2 8/2008 Lu et al.
7,412,531 B1 8/2008 Lango et al.
7,477,688 B1 1/2009 Zhang et al.
7,523,181 B2 4/2009 Swildens et al.
7,529,541 B2 5/2009 Cho et al.
7,536,469 B2 5/2009 Chou et al.
7,546,355 B2 6/2009 Kalnitsky
7,555,464 B2 6/2009 Candelore
7,558,472 B2 7/2009 Locket et al.
7,558,869 B2 * 7/2009 Leon H04L 29/06027 370/235
7,577,750 B2 8/2009 Shen et al.
7,593,333 B2 9/2009 Li et al.
7,599,307 B2 10/2009 Seckin et al.
7,609,652 B2 10/2009 Kellerer et al.
7,631,039 B2 12/2009 Eisenberg
7,653,735 B2 1/2010 Mandato et al.
7,657,644 B1 2/2010 Zheng
7,660,906 B1 2/2010 Armour
7,707,303 B2 * 4/2010 Albers H04L 1/1835 709/231
7,719,985 B2 5/2010 Lee et al.
7,733,830 B2 6/2010 Curcio et al.
7,760,801 B2 7/2010 Ghanbari et al.
7,761,609 B1 7/2010 Srinivasan et al.
7,779,135 B2 8/2010 Hudson et al.
7,788,395 B2 8/2010 Bowra et al.
7,797,439 B2 9/2010 Cherkasova et al.
7,817,985 B2 10/2010 Moon
7,818,444 B2 10/2010 Brueck et al.
7,873,040 B2 1/2011 Karlsgodt
8,036,265 B1 10/2011 Reynolds et al.
8,135,852 B2 3/2012 Nilsson et al.
8,209,429 B2 6/2012 Jacobs et al.
8,370,514 B2 2/2013 Hurst et al.
8,402,156 B2 3/2013 Brueck et al.
8,612,624 B2 12/2013 Frueck et al.
8,683,066 B2 3/2014 Hurst et al.
8,880,721 B2 11/2014 Hurst et al.
2001/0013128 A1 8/2001 Hagai et al.
2001/0047423 A1 11/2001 Shao et al.
2002/0073167 A1 6/2002 Powell et al.
2002/0087634 A1 7/2002 Ogle et al.
2002/0091840 A1 7/2002 Pulier et al.
2002/0097750 A1 7/2002 Gunaseelan et al.
2002/0118809 A1 8/2002 Eisenberg
2002/0122491 A1 9/2002 Karczewicz et al.
2002/0131496 A1 9/2002 Vasudevan et al.
2002/0133547 A1 9/2002 Lin
2002/0136406 A1 9/2002 Fitzhardinge et al.
2002/0138619 A1 9/2002 Ramaley et al.
2002/0144276 A1 10/2002 Radford et al.
2002/0146102 A1 10/2002 Lang
2002/0152317 A1 10/2002 Wang et al.
2002/0152318 A1 10/2002 Menon et al.
2002/0161898 A1 10/2002 Hartop et al.
2002/0161908 A1 * 10/2002 Benitez G06F 8/65 709/231
2002/0161911 A1 10/2002 Pinckney, III et al.
2002/0169926 A1 11/2002 Pinckney, III et al.
2002/0174434 A1 11/2002 Lee et al.
2002/0176418 A1 11/2002 Hunt et al.
2002/0178138 A1 11/2002 Ender et al.
2002/0178330 A1 11/2002 Schlowsky-Fischer et al.
2002/0184391 A1 12/2002 Phillips
2002/0188745 A1 * 12/2002 Hughes H04L 29/06027 709/231

(56) **References Cited**

U.S. PATENT DOCUMENTS

2002/0194608 A1 12/2002 Goldhor
2003/0005455 A1 1/2003 Bowers
2003/0007464 A1 1/2003 Balani
2003/0014684 A1 1/2003 Kashyap
2003/0018966 A1 1/2003 Cook et al.
2003/0021166 A1 1/2003 Soloff
2003/0021282 A1* 1/2003 Hospodor H04L 12/5695 370/401
2003/0037103 A1 2/2003 Salmi et al.
2003/0065803 A1 4/2003 Heuvelman
2003/0067872 A1 4/2003 Harrell et al.
2003/0078972 A1 4/2003 Tapissier et al.
2003/0081582 A1 5/2003 Jain et al.
2003/0093790 A1 5/2003 Logan et al.
2003/0107994 A1 6/2003 Jacobs et al.
2003/0135631 A1 7/2003 Li et al.
2003/0140159 A1 7/2003 Campbell et al.
2003/0151753 A1 8/2003 Li et al. 358/1.9
2003/0152036 A1 8/2003 Quigg Brown et al.
2003/0154239 A1 8/2003 Davis et al.
2003/0195977 A1* 10/2003 Liu H04L 29/06027 709/231
2003/0204519 A1 10/2003 Sirivara et al.
2003/0204602 A1 10/2003 Hudson et al.
2003/0220972 A1 11/2003 Montet et al.
2004/0003101 A1 1/2004 Roth et al.
2004/0010613 A1 1/2004 Apostolopoulos et al.
2004/0030547 A1 2/2004 Leaning et al.
2004/0030599 A1 2/2004 Sie et al.
2004/0030797 A1 2/2004 Akinlar et al.
2004/0031054 A1 2/2004 Dankworth et al.
2004/0049780 A1 3/2004 Gee
2004/0054551 A1 3/2004 Ausubel et al.
2004/0071209 A1 4/2004 Burg et al.
2004/0083283 A1 4/2004 Sundaram et al.
2004/0093420 A1 5/2004 Gamble
2004/0098748 A1 5/2004 Bo et al.
2004/0103444 A1 5/2004 Weinberg et al.
2004/0117427 A1 6/2004 Allen et al.
2004/0143672 A1 7/2004 Padmanabham et al.
2004/0153458 A1 8/2004 Noble et al.
2004/0168052 A1 8/2004 Clisham et al.
2004/0170392 A1 9/2004 Lu et al.
2004/0179032 A1* 9/2004 Huang H04N 19/61 715/723
2004/0199655 A1* 10/2004 Davies H04L 47/10 709/231
2004/0220926 A1 11/2004 Lamkin et al.
2004/0221088 A1* 11/2004 Lisitsa H04L 29/06027 710/316
2004/0260701 A1 12/2004 Lehikoinen et al.
2004/0267956 A1* 12/2004 Leon H04L 12/5602 709/231
2005/0009520 A1 1/2005 Herrero et al.
2005/0015509 A1 1/2005 Sitaraman
2005/0024487 A1 2/2005 Chen
2005/0033855 A1 2/2005 Moradi et al.
2005/0050152 A1 3/2005 Penner et al.
2005/0055425 A1 3/2005 Lango et al.
2005/0066063 A1 3/2005 Grigorovitch et al.
2005/0076136 A1 4/2005 Cho et al.
2005/0084166 A1 4/2005 Bonch et al.
2005/0108414 A1 5/2005 Taylor et al. 709/231
2005/0120107 A1 6/2005 Kagan et al.
2005/0123058 A1 6/2005 Greenbaum et al.
2005/0185578 A1 8/2005 Padmanabhan et al.
2005/0188051 A1 8/2005 Sneh
2005/0204046 A1 9/2005 Watanabe
2005/0204385 A1 9/2005 Sull et al.
2005/0223087 A1 10/2005 Van Der Stok
2005/0251832 A1 11/2005 Chiueh
2005/0254508 A1 11/2005 Aksu et al.
2005/0262257 A1 11/2005 Major et al.
2006/0010003 A1 1/2006 Kruse
2006/0047779 A1 3/2006 Deshpande
2006/0059223 A1 3/2006 Klemets et al.
2006/0080718 A1 4/2006 Gray et al.
2006/0130118 A1 6/2006 Damm
2006/0133809 A1 6/2006 Chow et al.
2006/0165166 A1 7/2006 Chou et al.
2006/0168290 A1 7/2006 Doron
2006/0168295 A1 7/2006 Batterberry et al.
2006/0184688 A1 8/2006 Ganguly et al.
2006/0206246 A1 9/2006 Walker
2006/0218264 A1 9/2006 Ogawa et al.
2006/0236219 A1 10/2006 Grigorovitch et al.
2006/0242315 A1 10/2006 Nichols
2006/0270404 A1 11/2006 Tuohino et al.
2006/0277564 A1 12/2006 Jarman
2006/0282540 A1 12/2006 Tanimoto
2006/0288099 A1 12/2006 Jefferson et al.
2007/0024705 A1 2/2007 Richter et al.
2007/0030833 A1 2/2007 Pirzada et al.
2007/0037599 A1 2/2007 Tillet et al.
2007/0067480 A1 3/2007 Beek et al.
2007/0078768 A1 4/2007 Dawson
2007/0079325 A1 4/2007 de Heer
2007/0094405 A1 4/2007 Zhang
2007/0204310 A1 8/2007 Hua et al.
2007/0280255 A1 12/2007 Tsang et al.
2008/0028428 A1 1/2008 Jeong et al.
2008/0037527 A1 2/2008 Chan et al.
2008/0046939 A1 2/2008 Lu et al.
2008/0056373 A1 3/2008 Newlin et al.
2008/0060029 A1 3/2008 Park et al.
2008/0091838 A1 4/2008 Miceli
2008/0133766 A1 6/2008 Luo
2008/0162713 A1 7/2008 Bowra et al.
2008/0195744 A1 8/2008 Bowra et al.
2008/0195745 A1 8/2008 Bowra et al.
2008/0205291 A1 8/2008 Li et al.
2008/0219151 A1 9/2008 Ma et al.
2008/0263180 A1 10/2008 Hurst et al.
2008/0281803 A1 11/2008 Gentric
2009/0006538 A1 1/2009 Risney, Jr. et al.
2009/0049186 A1 2/2009 Agnihotri et al.
2009/0055417 A1 2/2009 Hannuksela
2009/0055471 A1 2/2009 Kozat et al.
2009/0055547 A1 2/2009 Hudson et al.
2009/0132599 A1 5/2009 Soroushian et al.
2009/0132721 A1 5/2009 Soroushian et al.
2009/0210549 A1 8/2009 Hudson et al.
2010/0098103 A1 4/2010 Xiong et al.
2010/0158101 A1 6/2010 Wu et al.
2010/0262711 A1 10/2010 Bouazizi
2014/0207966 A1 7/2014 Hurst et al.
2015/0058496 A1 2/2015 Hurst et al.

FOREIGN PATENT DOCUMENTS

EP 0 919 952 A1 6/1999
EP 1202487 A2 10/2001
EP 1395014 A1 8/2002
EP 1298931 A2 2/2003
EP 1298931 A2 4/2003
EP 1 641 271 A2 3/2006
EP 1 670 256 A2 6/2006
EP 1 777 969 A1 4/2007
GB 2367219 A 9/2000
JP 2000201343 A 7/2000
JP 200192752 4/2001
JP 2004054930 2/2004
JP 2011004225 A 1/2011

(56) **References Cited**

FOREIGN PATENT DOCUMENTS

WO	WO 00/67469		11/2000
WO	0167264	A1	9/2001
WO	03003760	A2	1/2003
WO	03009581	A1	1/2003
WO	03027876	A1	4/2003
WO	2004025405	A2	3/2004
WO	2004036824	A1	4/2004
WO	2006010113	A2	1/2006

OTHER PUBLICATIONS

U.S. Patent and Trademark Office, Non-Final Office Action, dated Oct. 24, 2014 for U.S. Appl. No. 14/222,245.
USPTO, Notice of Allowance and Fee(s) Due for U.S. Appl. No. 14/106,051 mailed Feb. 24, 2015.
USPTO, Final Office Action for U.S. Appl. No. 14/222,245 mailed Mar. 18, 2015.
Canadian Intellectual Property Office, Office Action, dated Sep. 10, 2014 for Canadian Application No. 2564861.
USPTO "International Search Report" mailed Dec. 12, 2008; International Appln. No. PCT/US2008/061035, filed Apr. 21, 2008.
Australian Government "Examiner's First Report" dated Oct. 17, 2011; Australian Patent Appln. No. 2011213730.
Korean Intellectual Property Office "Official Notice of Preliminary Rejection" issued Jul. 28, 2011; Korean Patent Appln. No. 10-2006-7025274.
Japan Patent Office "Notice of Rejection Ground" mailed Apr. 26, 2011; Japanese Patent Appln. No. 2007-511070.
Fujisawa, Hiroshi et al. "Implementation of Efficient Access Mechanism for Multiple Mirror-Servers" IPSJ SIG Technical Report, vol. 2004, No. 9, Jan. 30, 2004, Information Processing Society of Japan, pp. 37-42.
Liu, Jiangchuan et al. "Opportunities and Challenged of Peer-to-Peer Internet Video Broadcast," School of Computing Science, Simon Fraser University, British Columbia, Canada.
USPTO International Searching Authority "International Search Report and Written Opinion," mailed Nov. 5, 2008; International Appln. No. PCT/US2008/009281, filed Aug. 1, 2008.
Zhang, Xinyan et al. "CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming" IEEE 2005.
Guo, Yang "DirectStream: A Directory-Based Peer-To-Peer Video Streaming Service" LexisNexis, Elsevier B.V. 2007.
Liu, Jiangchuan et al. "Adaptive Video Multicast Over the Internet" IEEE Computer Society, 2003.
Rejaie, Reza et al. "Architectural Considerations for Playback of Quality Adaptive Video Over the Internet" University of Southern California, Information Sciences Institute, 1998.)
Roy, Sumit et al. "A System Architecture for Managing Mobile Streaming Media Services" Streaming Media Systems Group, Hewlett-Packard Laboratories, 2003.
Xu, Dongyan et al. "On Peer-to-Peer Media Streaming" Department of Computer Sciences, Purdue University, 2002.
Kozamernik, Franc "Media Streaming Over the Internet—An Over of Delivery Technologies" EBU Technical Review, Oct. 2002.
Lienhart, Rainer et al. "Challenges in Distributed Video Management and Delivery" Intel Corporation, EECS Dept., UC Berkeley, 2000-2002.
Japan Patent Office "Final Office Action" mailed Feb. 28, 2012 in Patent Application No. 2007-511070 filed on Oct. 26, 2006.
Japan Patent Office "Interrogation" mailed Nov. 6, 2012 in Patent Application No. 2007-511070 filed on Oct. 26, 2006.
Canadian Intellectual Property Office "Office Action" mailed Sep. 9, 2013 in Patent Application No. 2,564,861 filed on Oct. 30, 2006.
USPTO "Office Action" mailed Sep. 13, 2013 in U.S. Appl. No. 13/757,571, filed Feb. 1, 2013.
USPTO "Notice of Allowance" mailed Jun. 24, 2014 in U.S. Appl. No. 13/757,571, filed Feb. 1, 2013.
European Patent Office "Extended Search Report" dated Jul. 10, 2014 in Patent Application No. 12154559.4 filed on Sep. 20, 2002.
Nguyen, Thinh, "Multiple Sender Distributed Video Streaming" in IEEE Transactions on Multimedia, vol. 6, No. 2, Published Apr. 2, 2004.
Weblio, The Meaning of Performance Factor—English-Japanese Weblio Dictionary, [online], Feb. 24, 2012; retrieved from the internet—URL:http://ejje, weblio.jp/content/performance+factor.
Masato Tsuru et al., Recent Evolution of the Internet Measurement and Inference Techniques, IEICE Technical Report, vol. 103, No. 123 (IN2003-16 to 23), IEICE, Jun. 12, 2003, pp. 37 to 42, ISSN: 0913-05685.
Takeshi Yoshimura et al., Mobile Streaming Media CDN Enabled by Dynamic SMIL, WWW2002, May 7-11, 2002; retrieved from the Internet at http://www2002.org/CDROM/refereed/515/.
Canadian Intellectual Property Office, Office Action, mailed Oct. 15, 2012 for Patent Application No. 2,564,861.
Clement, B., Move Networks Closes $11.3 Million on First Round VC Funding, Page One PR, Move Networks, Inc. Press Releases, Feb. 7, 2007, http://www.move.tv/press/press20070201.html.
Move Networks, Inc., The Next Generation Video Publishing System, Apr. 11, 2007; http://www.movenetworks.com/wp-content/uploads/move-networks-publishing-system.pdf.
U.S. Patent and Trademark Office, Non-Final Office Action, dated Aug. 7, 2014 for U.S. Appl. No. 14/106,051.
Final Office Action for U.S. Appl. No. 11/673,483, Feb. 4, 2010, 21 pages.
Advisory Action for U.S. Appl. No. 11/673,483, Apr. 9, 2010, 3 pages.
Advisory Action for U.S. Appl. No. 11/673,483, May 26, 2010, 3 pages.
Notice of Allowance for U.S. Appl. No. 11/673,483, Aug. 5, 2010, 7 pages.
Wicker, Stephen B., "Error Control Systems for Digital Communication and Storage", Prentice-Hall, Inc., New Jersey, USA, 1995 (Book: see NPL's Parts 1-6).
PCT Notification of Transmittal of the International Search Report and Written Opinion of the International Searching Authority, for PCT/US05/15091, Oct. 29, 2007, 8 pages.
PCT Notification of Transmittal of International Preliminary Report on Patentability, for PCT/US05/15091, Oct. 29, 2007, 6 pages.
Office Action for U.S. Appl. No. 11/673,483, Jul. 9, 2009, 14 pages.
Office Action for U.S. Appl. No. 11/673,483, Feb. 3, 2009, 9 pages.
Albanese, Andres, et al. "Priority Encoding Transmission", TR-94-039, Aug. 1994, 36 pages, International Computer Science Institute, Berkeley, California.
Puri, Rohit, et al. "Multiple Description Source Coding Using Forward Error Correction Codes", Oct. 1999, 5 pages, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California.
Goyal, Vivek K., "Multiple Description Coding: Compression Meets the Network", Sep. 2001, pp. 74-93, IEEE Signal Processing Magazine.
Supplemental European Search Report, Sep. 30, 2008, (3 pages).
Pathan, Al-Mukaddim, et al., "A Taxonomy and Survey of Content Delivery Networks", Australia, Feb. 2007. Available at http://gridbus.org/reports/CDN-Taxonomy.pdf.
On2 Technologies, Inc., "TrueMotion VP7 Video Codec", White Paper, Document Version 1.0, Jan. 10, 2005, (13 pages).
USPTO, Office Action for U.S. Appl. No. 14/531,804, mailed May 11, 2015.

* cited by examiner

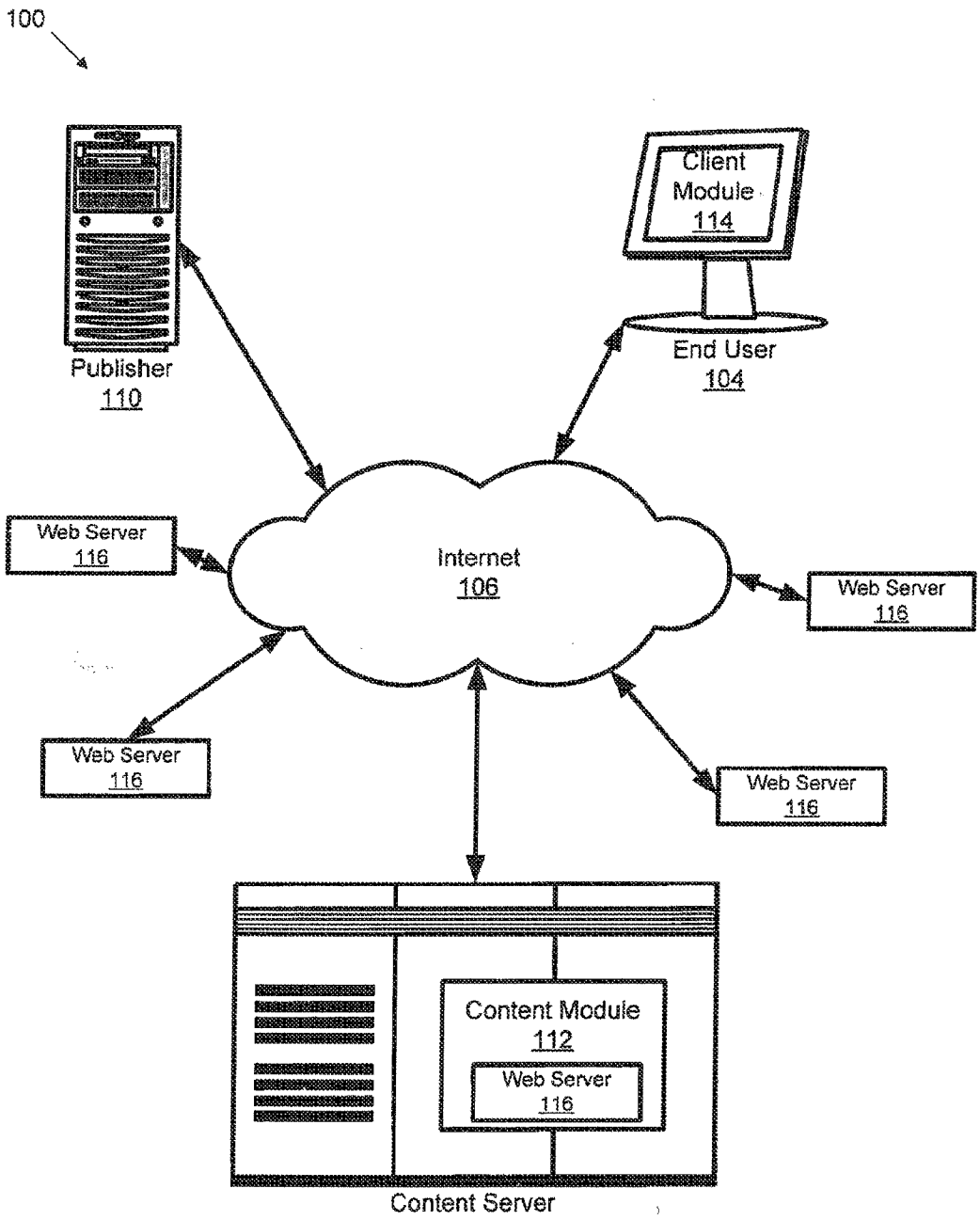


FIG. 1

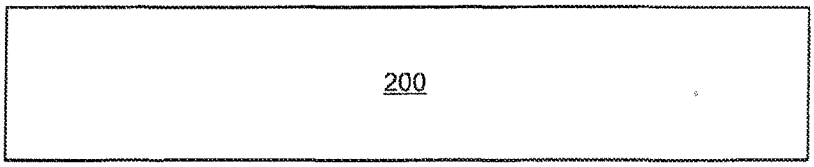


FIG. 2a

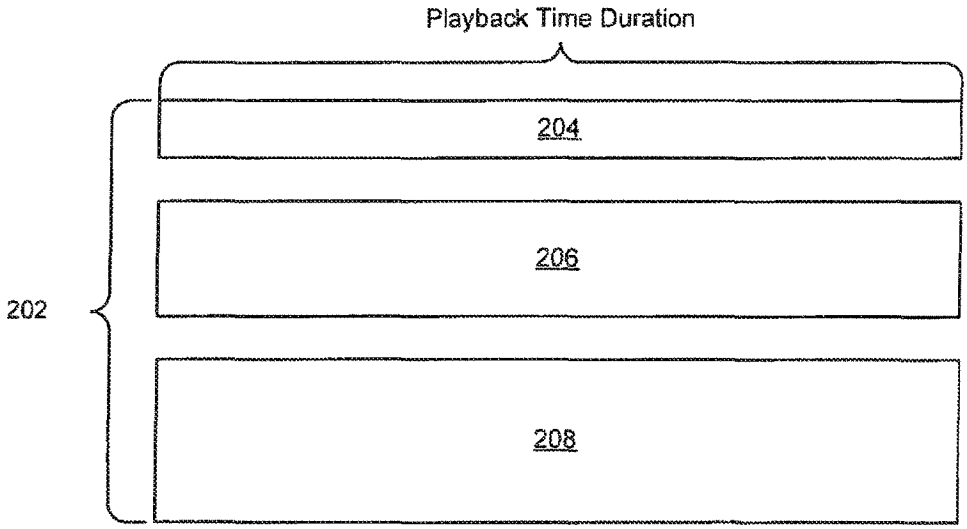


FIG. 2b

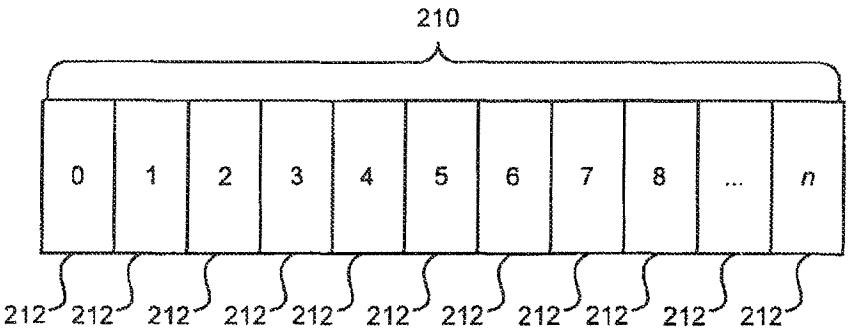


FIG. 2c

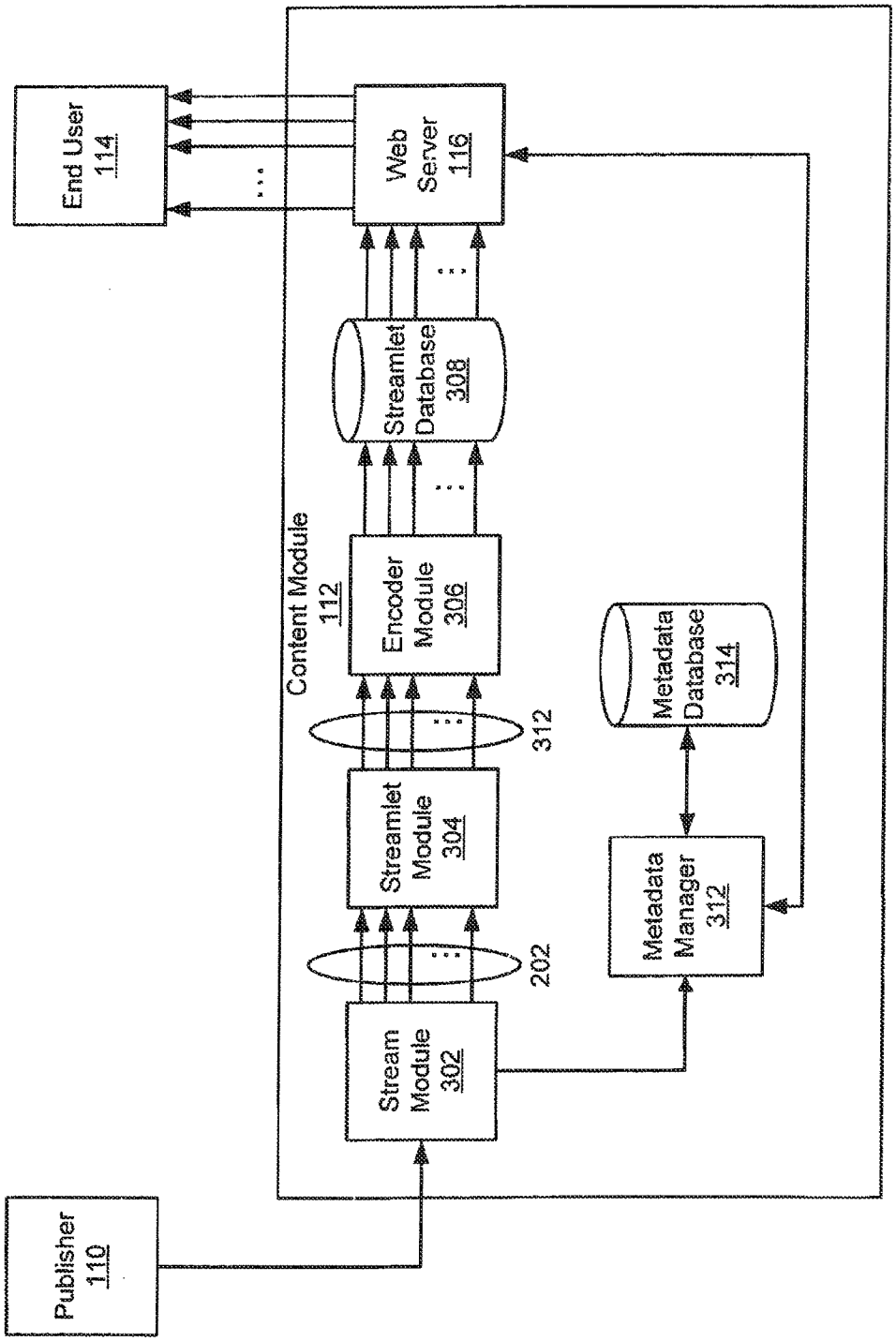


FIG. 3

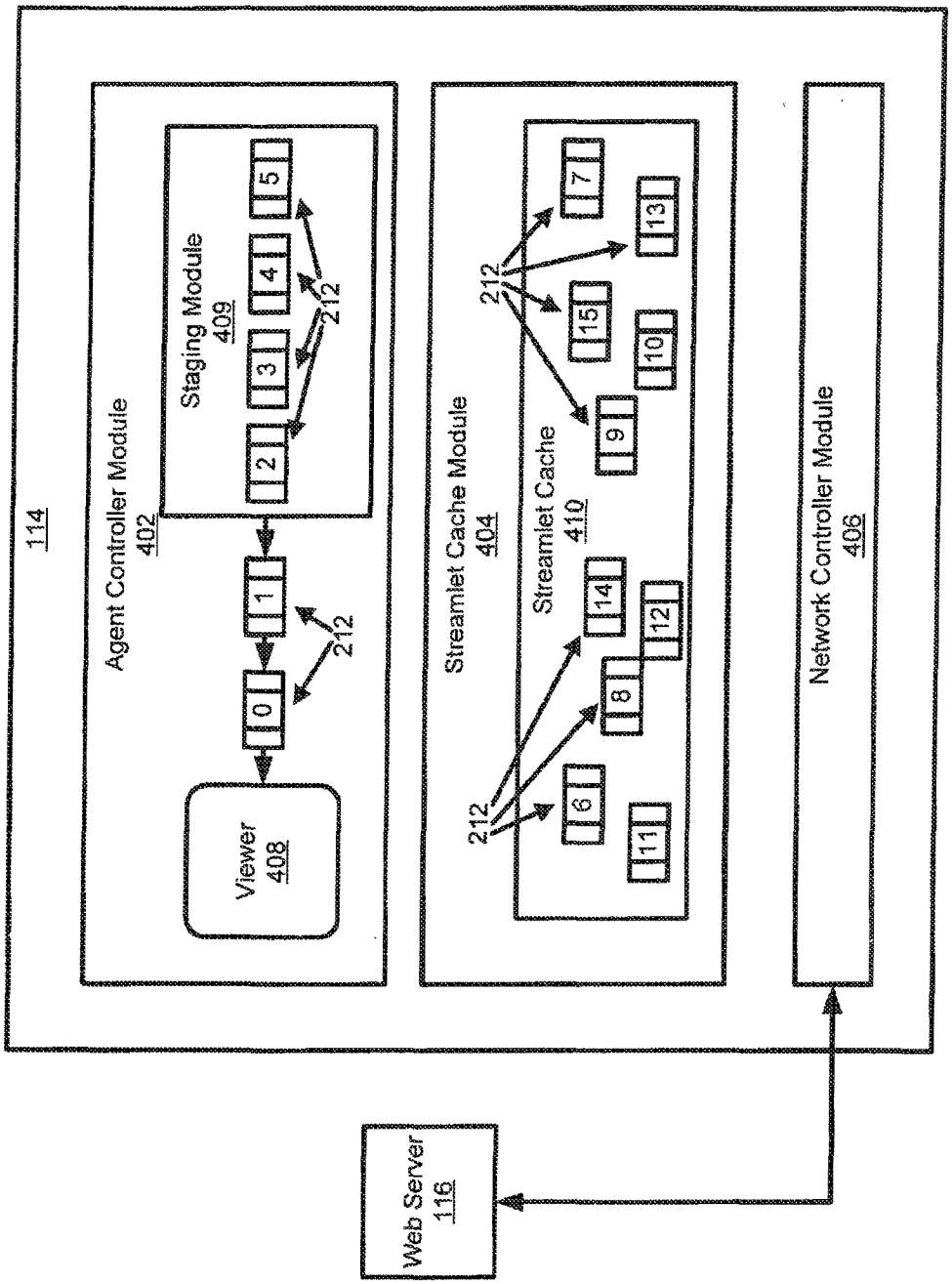


FIG. 4

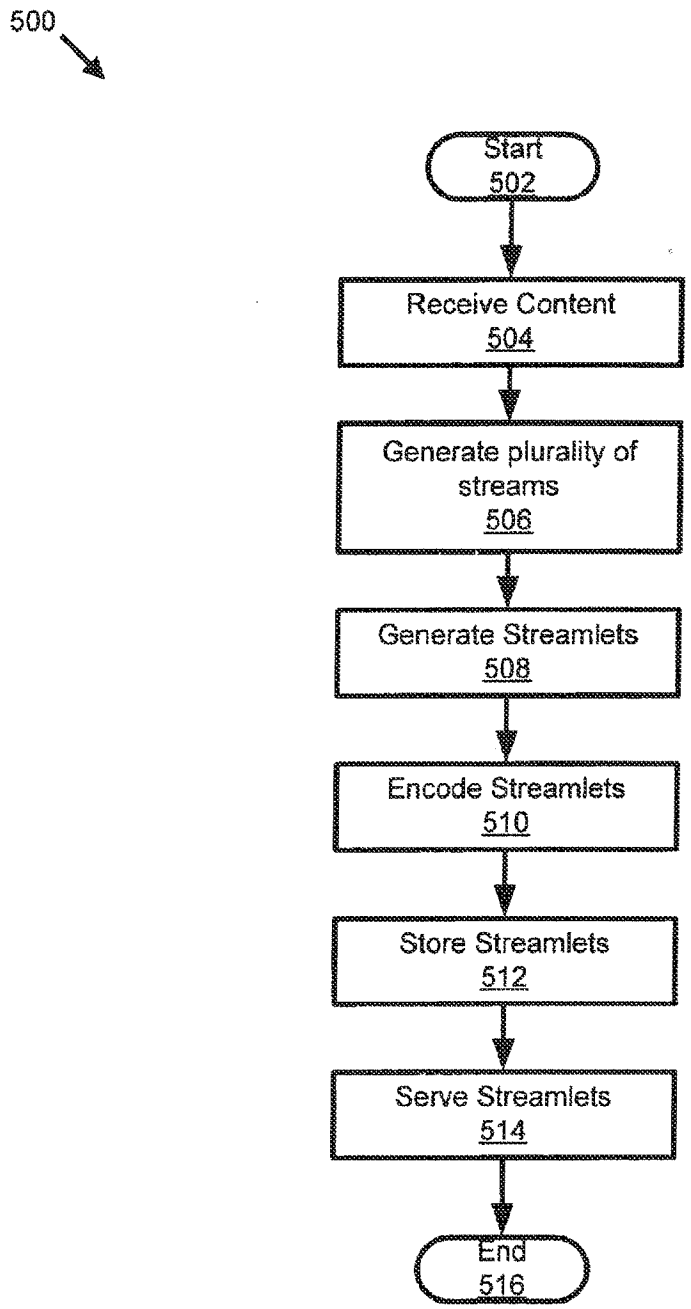


FIG. 5

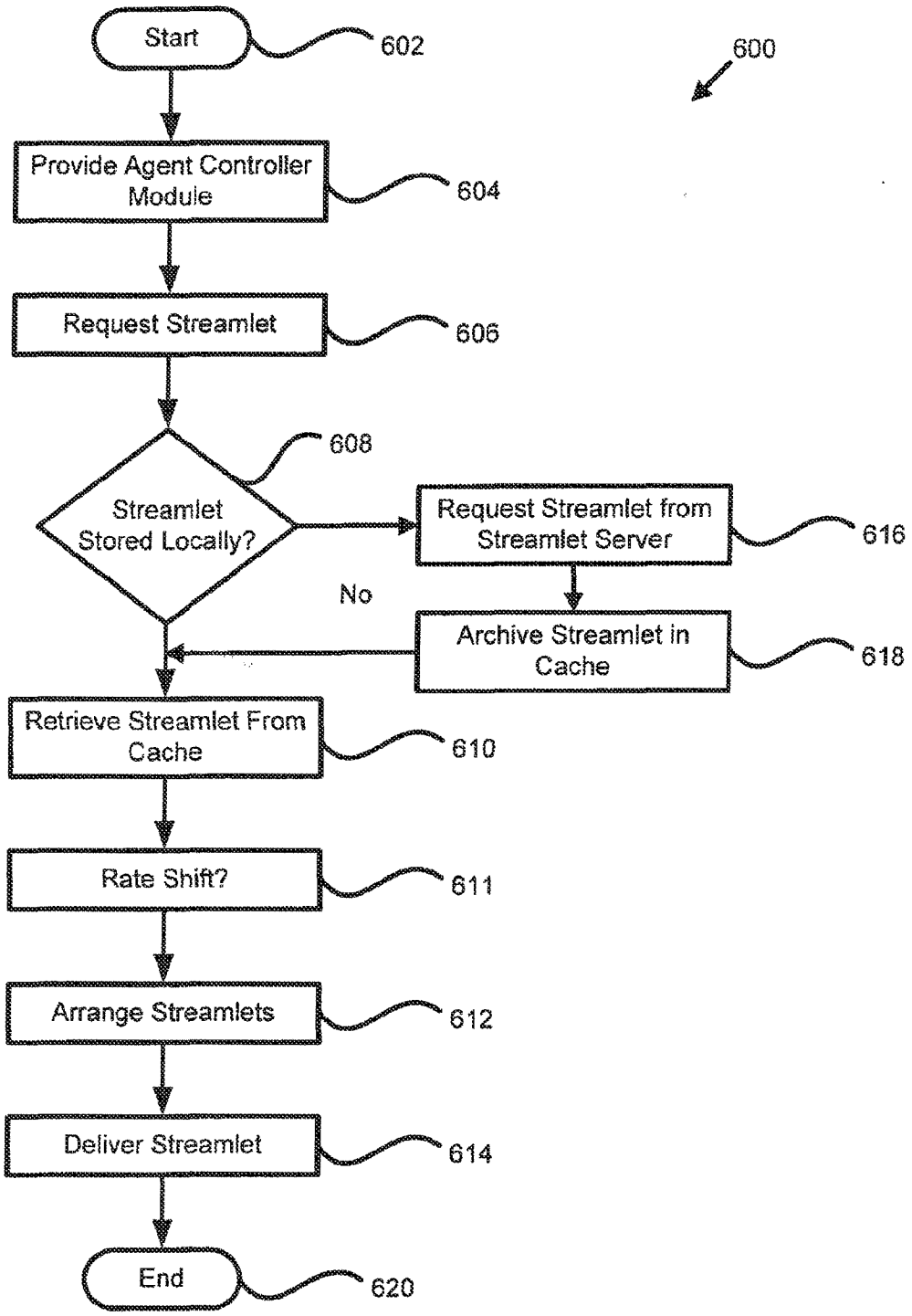


FIG. 6

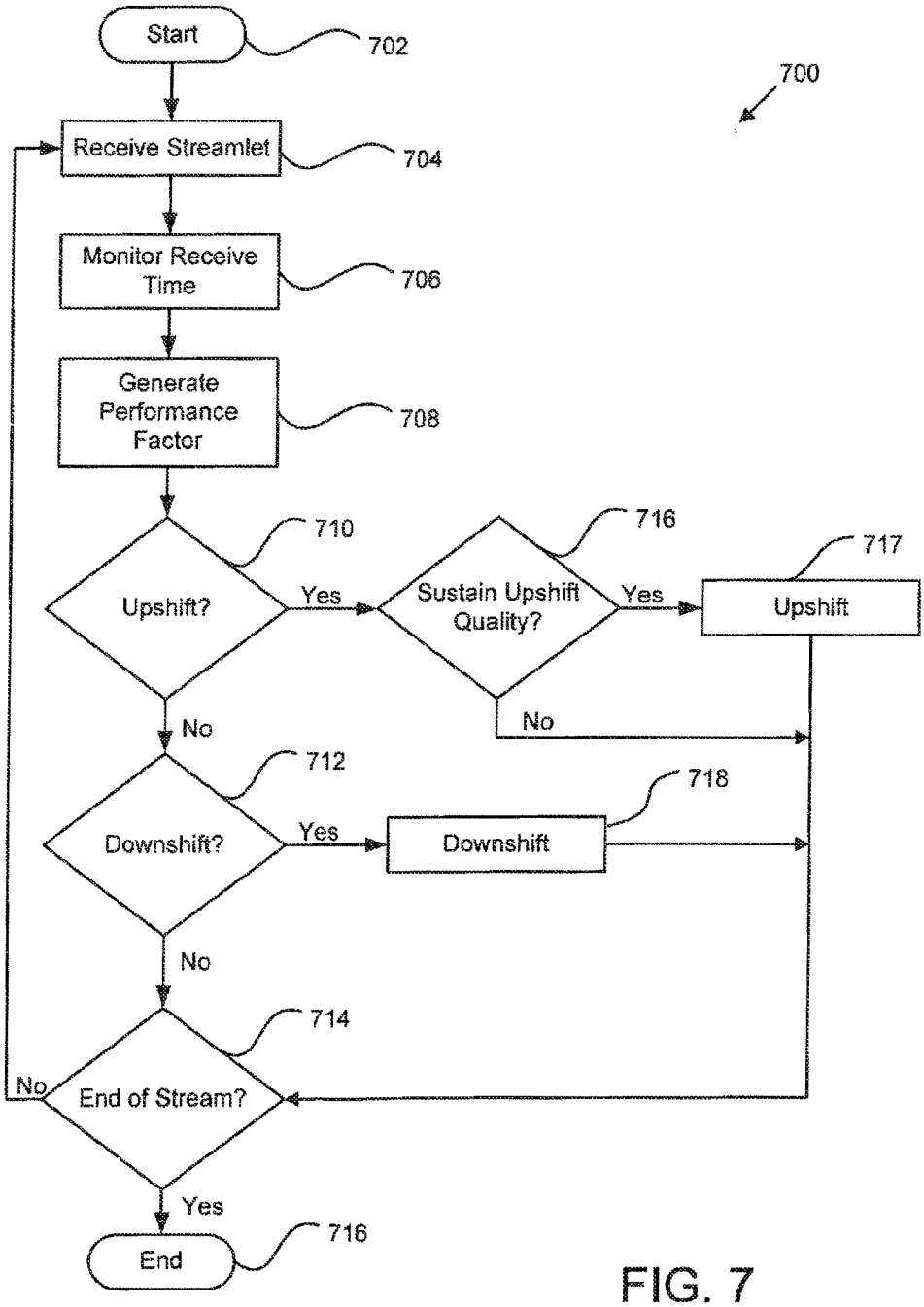


FIG. 7

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

CROSS-REFERENCES TO RELATED APPLICATIONS

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

1. Field of the Invention

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.

2. Description of the Related Art

The Internet is last 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.

"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 wading 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.

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.

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.

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.

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.

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.

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.

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

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.

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.

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.

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 a 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.

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.

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.

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.

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.

In a further embodiment, the method may include establishing multiple Transmission Control Protocol (TCP) connections with a content server, and requesting streamlets of varying nitrates. Also, the method may include generating & 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.

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.

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.

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.

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

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;

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;

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

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;

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;

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

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

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

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

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

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.

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.

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.

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.

Reference to a signal hearing 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.

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.

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.

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, it the content to be streamed were a broadcast of a television program, the publisher 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 fee 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 horn any of the plurality of web servers **116**.

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 98®, Quicklime 6.5 Sorenson 3®, or Quicklime 6.5/MPEG-4® encoded content.

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.

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.

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.

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®, Quicklime 6.5 Sorenson 3®, or Quicklime 6.5/MPEG-4®. Alternatively, a custom encoding scheme may be employed.

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, cause, 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.

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.

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.

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**.

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 fee configured to arrange the streamlets **212** in order of ascending playback time. In the depleted embodiment, the streamlets **212** are numbered 0, 1, 2, 3, 4, etc. However, each streamlet **212** may be identified with a unique filename.

Additionally, the agent controller module **402** may be configured to anticipate streamlet **212** requests and pre-request streamlets **212**. By pro-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.

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 m the streamlet cache **410**, the request is passed to the network controller module **406**. In order to enable last 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**.

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.

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.

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**.

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.

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.

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.

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.

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.

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 wading or monitoring period of unspecified duration between enumer-

ated 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.

FIG. **5** is a schematic flowchart 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 pre-designated publisher guidelines

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**.

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**.

In one embodiment, the staging module **409** then arranges **612** tire 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.

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 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.

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

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

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:

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

The policy determination about whether or not to upshift **710** playback quality begins by comparing $\phi_{current}$ with a trigger threshold Θ_{up}. If $\phi_{current} \geq \Theta_{up}$, 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 qualify 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 $\phi_{higher} \geq \Theta_{up}$ then use of the higher quality stream is considered sustainable. If the decision of whether or not the higher stream rate is sustainable **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**.

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} 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 $\phi_{current}\leq\Theta_{down}$, 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.

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 end user station for adaptive-rate content streaming of digital content from a video server over a network, the end user station comprising:

a media player operating on the end user station configured to stream a video from the video server via at least one transmission control protocol (TCP) connection over the network, wherein multiple different copies of the video encoded at different bit rates are stored on the video server as multiple sets of files, wherein each of the files yields a different portion of the video on playback, wherein the files across the different copies yield the same portions of the video on playback, and wherein each of the files comprises a time index such that the files whose playback is the same portion of the video for each of the different copies have the same time index in relation to the beginning of the video, and wherein the media player streams the video by:

requesting a plurality of sequential files of one of the copies from the video server based on the time indexes;

automatically requesting from the video server subsequent portions of the video by requesting for each such portion one of the files from one of the copies dependent upon successive determinations by the media player to shift the playback quality to a higher or lower quality one of the different copies, the automatically requesting including repeatedly generating a factor indicative of the current ability to sustain the streaming of the video using the files from different ones of the copies, wherein the set of one or more factors relate to the performance of the network;

making the successive determinations to shift the playback quality based on the factor to achieve continuous playback of the video using the files of the highest quality one of the copies determined sustainable at that time so that the media player upshifts to a higher quality one of the different copies when the factor is greater than a first threshold and downshifts to a lower quality one of the different copies when the factor is less than a second threshold; and

presenting the video by playing back the requested media files with the media player on the end user station in order of ascending playback time.

2. The end user station of claim **1**, wherein the at least one TCP connection comprises multiple Transmission Control protocol (TCP) connections with the content server.

3. The end user station of claim **1**, wherein the media player is configured to generate the factor according to the responses to segment requests.

4. The end user station of claim **1**, wherein the media player is configured to upshift to the higher quality copy when the factor is greater than the first threshold and the media player determines the higher quality playback can be sustained according to a combination of factors.

5. The end user station of claim **1** wherein the media player is configured to upshift to the higher quality copy when the performance factor is greater than the first threshold and the media player determines that the higher quality playback can be sustained according to an amount of contiguously available files stored by the media player.

6. The end user station of claim **1**, wherein the media player is further configured to anticipate file requests and to prerequest files to enable fast-forward, skip randomly, and rewind functionality.

7. The end user station of claim **1**, wherein the media player is configured to initially request low quality files to enable instant playback of the video, and to subsequently upshift to a better quality copy of the video according to the performance factor.

8. A method executable by an end user station to present rate-adaptive streams received via at least one transmission control protocol (TCP) connection with a server over a network, the method comprising;

streaming, by a media player operating on the end user station, a video from the server via the at least one TCP connection over the network, wherein multiple different copies of the video encoded at different bit rates are stored as multiple sets of files on the server, wherein each of the files yields a different portion of the video on playback, wherein the files across the different copies yield the same portions of the video on playback, and wherein each of the files comprises a time index such that the files whose playback is the same portion of the video for each of the different copies have the same time index in relation to the beginning of the video, and wherein the streaming comprises:

requesting by the media player a plurality of sequential files of one of the copies from the server based on the time indexes;

automatically requesting by the media player from the server subsequent portions of the video by requesting for each such portion one of the files from one of the copies dependent upon successive determinations by the media player to shift the playback quality to a higher or lower quality one of the different copies, the automatically requesting including repeatedly generating a factor indicative of the current ability to sustain the streaming of the video using the files from different ones of the copies, wherein the factor relates to the performance of the network; and

making the successive determinations to shift the playback quality based on the factor to achieve continuous playback of the video using the files of the highest quality one of the copies determined sustainable at that time, wherein the making the successive determinations to shift comprises upshifting to a higher quality one of the different copies when the at least one factor is greater than a first threshold and downshifting to a lower quality one of the different copies when the at least one factor is less than a second threshold; and

presenting the video by playing back the requested media files with the media player on the end user station in order of ascending playback time.

9. The method of claim **8**, wherein the at least one TCP connection comprises a plurality of different TCP connections, and wherein the requesting the plurality of sequential

files includes requesting sub-parts of the files over different ones of the plurality of different TCP connections, and wherein said presenting includes reassembling the files from the received sub-parts.

10. The method of claim **8**, wherein said making the successive determinations to shift comprises:

determining if the higher quality playback can be sustained.

11. The method of claim **8**, wherein the at least one TCP connection comprises a plurality of different TCP connections, and wherein the automatically requesting includes requesting sub-parts of the files over different ones of the plurality of TCP connections, and wherein said presenting includes reassembling the files from the received sub-parts, and wherein the factor is indicative of the available bandwidth of the plurality of TCP connections.

12. The method of claim **8**, wherein the factor is indicative of latency of the requested files, wherein the latency is a time measure between when one of the requests is issued and a time that response data of the request begins to arrive at the end user station.

13. The method of claim **8**, wherein the factor is indicative of time intervals between successive receive times for each response to the requested files.

14. The method of claim **8**, wherein the factor is indicative of delays or losses in one or more of the at least one TCP connection.

15. The method of claim **8**, wherein the server is a web server, and wherein the files are requested from the web server using Hyper Text Transfer Protocol (HTTP) messages sent via the at least one TCP connection.

16. The method of claim **8**, wherein the server comprises a cache server of a network infrastructure operator.

* * * * *

EXHIBIT E

SIGN IN WATCH NOW

ABOUT UNIVISION NOW

- **What is Univision NOW?**

  Univision NOW is a digital video subscription service offering a live stream of Univision and UniMás and includes live sporting events, specials, series, and streams from local stations in select markets. Subscribers can instantly watch programs from the past 3 days (72 hours) of the live stream using the program guide, and can watch hundreds of primetime and library shows on demand. Not all Univision and UniMás programming is available through the Univision NOW service.

- **Where is Univision NOW available?**

  Univision NOW is available only within the United States with the exception of Puerto Rico.

- **What is the difference between Univision NOW and the Univision or Univision Deportes apps?**

  Both Univision NOW, and the Univision and Univision Deportes apps offer viewers live streams / on demand content. All require at least one type of subscription.

  Univision NOW is a subscription service that you purchase directly from Univision instead of a traditional Cable TV provider. The content is accessed over the internet and does not require a subscription with a cable TV provider but does require an internet connection.

  The Univision and Univision Deportes apps are available, at no additional cost, to existing cable TV subscribers through traditional Cable TV providers.

- **I have a Cable TV subscription. Can I get Univision NOW for Free?**

  If you receive Univision with your Cable TV package, you may be able to get Univision NOW for Free. Click 'Sign In' and choose 'TV Provider Sign In.' Locate your provider and follow the on-screen instructions. Once you're logged in, you'll have access to live network streams and primetime shows on demand. If you don't see your TV service provider listed, check back with us soon. We are actively working to add more providers in the near future.

BILLING & ACCOUNT INFO

- **How do I subscribe to Univision NOW?**

  Via the Univision NOW app:
  1) Download the app to your device via the App Store, Google Play, or Roku store
  2) Open the app
  3) Click "Watch Now"
  4) Choose a plan
  5) Click "Continue"
  6) Create your account
  7) Follow the on-screen instructions to purchase an Univision NOW subscription

  Note: If you have a promotional code, you must visit www.UnivisionNOW.com to redeem.

  Via the web (desktop or mobile):
  1) Visit www.UnivisionNOW.com
  2) Click on the "Watch Now" button
  3) Choose a plan
  4) Create your account
  5) Enter your billing information
  6) Download the Univision NOW app via the App Store, Google Play, or Roku store and sign in

  Note: The email account and password you enter at registration will be your Univision NOW username and password

- **What forms of payment are accepted?**

  Visa, Mastercard, Discover, American Express, and PayPal.

- **How do I update my payment information?**

  If purchased via the App Store:

  https://support.apple.com/en-us/HT201266

  If purchased through the Google Play Store:

  https://support.google.com/googleplay/answer/2476088

  If purchased via the web (desktop or mobile), log onto your account, click on "My Account" and then scroll down to "Payment Information" and click on "Edit". Enter your new payment information and click "Update".

- **How do I cancel my Univision NOW account?**

  If purchased via the App Store:

  https://support.apple.com/en-us/HT202039

  If purchased via the Google Play Store:

  https://support.google.com/googleplay/answer/2476088?hl=en

  If purchased via the web (desktop or mobile):
  1) Click on "My Account"
  2) Scroll down to "Subscription"
  3) Click on "Cancel Subscription"

  You can cancel your Subscription to the Service at any time. Please note that you must cancel your Subscription before it renews for a subsequent term in order to avoid being charged for the next plan's Subscription Fees. We do not provide credit, refunds, or prorated billing for subscriptions that are cancelled mid-billing cycle. If you cancel your Subscription, the cancellation will become effective at the end of the then-current Subscription period.

- **Can I get a refund if I don't use my subscription or cancel my subscription before the end of the billing cycle?**

  No, refunds are not provided for any subscription. We do not provide credit, refunds, or prorated billing for subscriptions that are cancelled mid-billing cycle. If you cancel your subscription mid-billing cycle, you will continue to have access to your subscription until the end of the billing cycle.

  We reserve the right to offer refunds, discounts or other consideration in select circumstances at our sole discretion. Please note that each circumstance is unique and election to make such an offer in one instance does not create the obligation to do so in another.

- **What are these authorization charges on my bank statement?**

  Authorizations are charges we make to your financial institution to verify that the payment information you submitted to us is correct. These authorizations are temporary holds and are never collected.

  You may see these authorizations if you recently signed up, entered a free trial, edited your payment info or made changes to your subscription. You may also see an authorization if you were unable to sign up, this will happen when you enter your billing information but that information does not match the billing information your financial institution has on record.

  Authorizations are for certain dollar amounts depending on how you submit your payment information to us. These authorization amounts temporarily affect your available account balances because your financial institution is ensuring that there are sufficient funds available. But these funds are never collected, and the temporary hold will automatically be removed from your account after a few days. The number of days can vary depending on the region or card issuer. Check with your financial institution for details on how long it takes for them to remove the temporary hold.

- **How and when will I be billed for Univision NOW?**

CHAT

Your first charge occurs at the time you register. If you started your subscription as a free trial, we will charge you the following billing period on the date your free trial ends. As long as your subscription remains active, you will be charged on the same day of the month you signed up on, unless it is on a date that does not occur in a given month in which case you will billed on the prior day. For example, if you sign up for a monthly subscription plan on March 31st you will be charged on April 30th.

See our Terms of Use for more detail.

- **Is my credit card information secure?**

  We use administrative, technical, and physical safeguards to protect your payment information and adhere to the payment card industry data security standards to protect your credit card data.

- **How do I reset my password?**

  To reset your password, click on "My Account" and then click on "Change Password" in the "Profile" section. Enter your new password and then click "Update".

- **How do I login if I forgot my username or password?**

  For assistance retrieving your username, please contact customer support by clicking the "chat" on the web.

  To retrieve your password, click on the 'Forgot Password' link on the login page of the app or the www.UnivisionNOW.com website. A link will be emailed to you so you can reset your password.

PRODUCT & FEATURES

- **How do I start watching live TV?**

  Ensure that you're connected to the internet.

  Sign into your Univision NOW account via the Univision NOW app or the www.UnivisionNOW.com website. Then select the channel you want to stream to start viewing live TV.

  If you're using the app via your phone or tablet, make sure that "location services" is enabled, turned "on", so you receive a stream from your local market.

- **How often will I need to login into Univision NOW?**

  On mobile devices you will stay logged in until you log out.

  On PCs you will stay logged if you checked the "Remember Me" option until you logout or clear your browser setting/cache.

- **What devices can I use to stream Univision NOW?**

  Univision NOW is currently available on the following devices with more coming soon.
  - iPhone
  - iPad
  - Apple TV
  - Airplay
  - Roku
  - Chromecast
  - Android phones
  - Android tablets
  - Desktop web

- **Can I stream on multiple devices at one time? If so, how many?**

  Yes, you can stream Univision NOW on multiple devices. We allow 3 concurrent streams at any given time.

- **Can I access my local station?**

  At the moment, we offer live streams from our New York, Los Angeles, Houston, Dallas, Chicago, Miami, Atlanta, and San Francisco stations to users in those cities. Users in markets other than New York, Los Angeles, Houston, Dallas, Chicago, Miami, Atlanta, and San Francisco can still watch network programming live.

- **Can I access my local station if I travel to another market/city?**

  The Univision NOW app is geo-based. Local station feeds are only available in their respective markets. For example, you cannot watch the Los Angeles station feed if you are in New York.

- **When will my local station be available for streaming?**

  Currently, we only offer live streams from our New York, Los Angeles, Houston, Dallas, Chicago, Miami, Atlanta, and San Francisco stations.

- **What is 3-day replay?**

  72-hour playback viewing allows you to instantly watch / pause / rewind any program from the past 3 days (72 hours) of the live stream using the program guide. Not all Univision and UniMás programming is available through the Univision NOW Service.

- **What shows can I watch on demand viewing on Univision NOW?**

  Access to on demand content depends on your subscription plan.

  If you're subscribed to the "Total Access" plan you have access to the entire Univision NOW library of on demand content which include library and primetime shows as well as exclusive Univision NOW Originals.

  If you access Univision NOW via a participating cable TV provider, you have access to primetime shows on demand.

  Not all Univision and UniMás programming is available through the Univision NOW Service.

- **Why is some programming not available for live streaming, rewind or on demand on Univision NOW?**

  Due to licensing rights, certain shows that are offered on the Univision and UniMás broadcast networks are not available for Univision NOW live, via the rewind function or on demand streaming.

- **Is closed captioning available?**

  Yes, closed captioning is available for live TV, 72-hour live playback viewing and on demand shows.

- **Is Univision NOW available in HD?**

  Yes.

- **Where can I use this service?**

  You can use this service anywhere in the US with an internet connection with the exception of Puerto Rico.

- **Do I need to be connected to the internet to watch?**

  Yes, an internet connection via WiFi or your cellular data plan is required to watch Univision NOW.

CUSTOMER SERVICE

- **Who do I reach out to if I have a question about my account or Univision NOW?**

  Customer support is available via chat and email (support@univisionnow.com). We are happy to help answer your questions.

- **How can I submit general feedback about Univision NOW?**
  We welcome your feedback about your Univision NOW experience. Please submit your feedback via chat or email (support@univisionnow.com).

OTHER

- **Can I gift a subscription?**
  We currently do not offer gift subscriptions, however, we are working to make this option available in the future.

Need Any Help?

View frequently asked questions or email our friendly support team

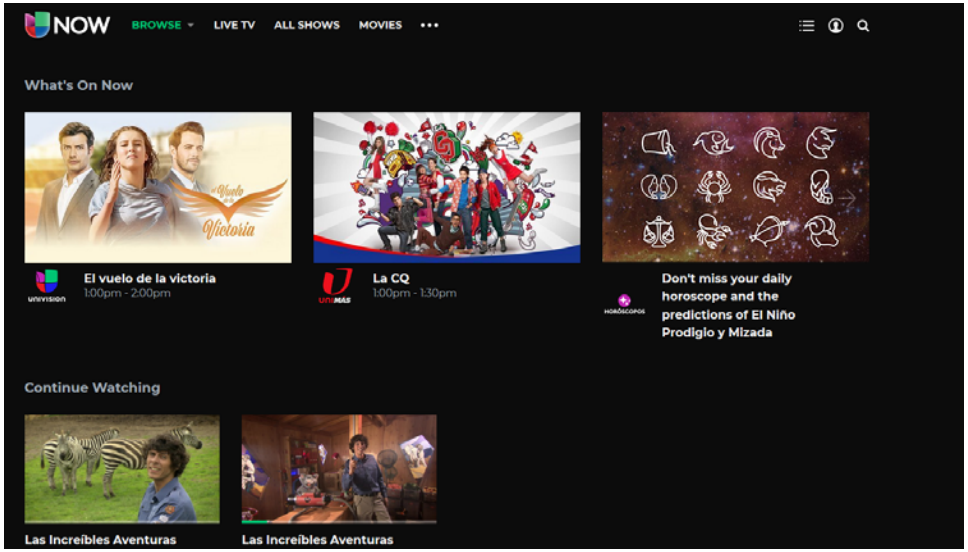
VIEW FAQ

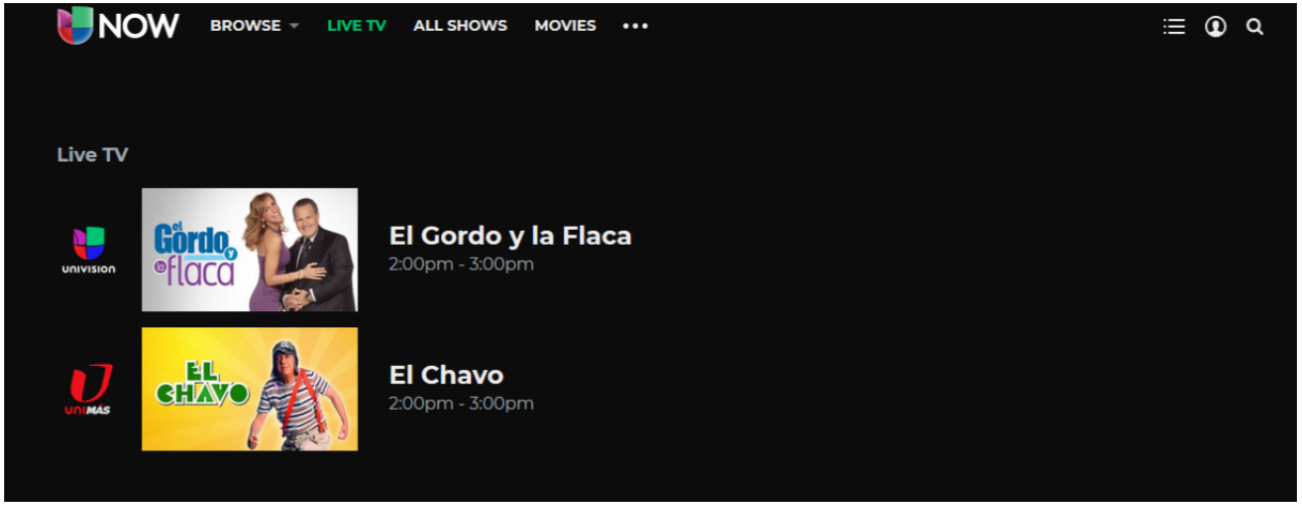
EMAIL US

EXHIBIT F

U.S. Patent No. 7,818,444 to Univision

The following claim chart shows exemplary aspects of the Univision Now Application ("Univision App"), Univision servers, and Univision services that infringe the claim below. The chart is exemplary and should not be read to limit DISH's claims against Univision to the products or services described below. The chart should also not be read to limit DISH's claims to the patent claim charted below. Nor should the chart below be read to limit how the Univision App, Univision servers, and Univision services infringe the claim below. DISH's claims against Univision may change based on the proceeding in this case and based on discovery provided by Univision.

Claim	Claim Limitation	Example Infringement Evidence
24	24. A method for multi-bitrate video and/or audio content streaming, the method comprising:	The Univision servers perform a method for multi-bitrate streaming of audio and video content to the Univision App. The Univision App is a content player that obtains streams of selected video program for playback from the Univision servers. The screenshots in this chart are from the Univision App running on a PC Firefox browser. The main menu of the Univision App after logging in:  As described in greater detail below, the Univision server interacts with the Univision App to execute a method for multi-bitrate video and/or audio content streaming.
	capturing a single video;	The Univision servers capture a single video by capturing live programming that is then transcoded and provided to users of the Univision App. The Univision App, based on user input, requests a stream of a selected video program via a network connection to a Univision server.

Claim	Claim Limitation	Example Infringement Evidence
		A user uses the on-screen guide to select a channel for streaming. First, the user selects "Live TV" from the Univision App main screen shown above. The Univision App offers both streaming of current Univision programming and VOD selections. The live streaming selection is shown by the "What's On Now" menu, or the "Live TV" link at the top of the page: 
	segmenting the single video and generating a plurality of sequential raw streamlets that collectively store data to playback the entire video and that individually store	The Univision servers segment the single video and generate a plurality of sequential raw streamlets. The streamlets collectively store data to playback the entire video (the airing program). The streamlets individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video. When the Univision App accesses a selected channel, it requests and receives a playlist file from the Univision server that shows the available versions of the channel at different resolutions. For example, a user navigates the menu above to select "Live TV" in the menu above and selects the stream of "**El Gordo y La Flaca**" as the "selected video program."

Claim	Claim Limitation	Example Infringement Evidence
	data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the corresponding video;	The Univision App makes an HTTPS GET request to **univisionadsvr01v2.nlst.neulion.com** for a master playlist named "**univision_east_hd_pc.m3u8**" that specifies the available streams and provides links to the playlists for those streams. The server returns the following playlist named "**univision_east_hd_pc.m3u8**": #EXTM3U #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=800000 univision_east_hd_800_pc.m3u8?x-b=800000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=400000 univision_east_hd_400_pc.m3u8?x-b=400000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1200000 univision_east_hd_1200_pc.m3u8?x-b=1200000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1600000 univision_east_hd_1600_pc.m3u8?x-b=1600000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d8

Claim	Claim Limitation	Example Infringement Evidence
		5fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=3000000 univision_east_hd_3000_pc.m3u8?x-b=3000000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=4500000 univision_east_hd_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 **Filename:** univision_east_hd_pc.m3u8 This is a master playlist file according to the HLS specification.[1] The master playlist shows six versions of the stream at the following bandwidths: • 800000 (Referred to herein as "**800 Bandwidth**") • 400000 (Referred to herein as "**400 Bandwidth**") • 1200000 (Referred to herein as "**1200 Bandwidth**") • 1600000 (Referred to herein as "**1600 Bandwidth**") • 3000000 (Referred to herein as "**3000 Bandwidth**") • 4500000 (Referred to herein as "**4500 Bandwidth**") For each of these versions, the master playlist provides a link to a playlist for the specified version of

[1] RFC 8216 (HLS Live Streaming), Section 4.3.4 (Master Playlist Tags).

Claim	Claim Limitation	Example Infringement Evidence
		the selected video program at a particular bandwidth and resolution. The Univision App initially selects the 800 Bandwidth version of the stream and makes a request to **univisionadsvr01v2.nlst.neulion.com** for the corresponding variant playlist file named "**univision_east_hd_800_pc.m3u8**." The server returns the file with the following contents: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=86c127a7cebe3eaf0e1a6a3eb06fcab6",IV=0x8C675584560D97EC2C6461F7C9BDB561 #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201140.ts

Claim	Claim Limitation	Example Infringement Evidence
		Filename: univision_east_hd_800_pc.m3u8 As noted above, the variant playlist file is an HLS playlist. Each line in the file "**univision_east_hd_800_pc.m3u8**" that begins with "#EXTINF" specifies the length of the segments in seconds. The line below the #EXTINF file is the location of the video file. The Univision App uses HTTPS GETs to request and retrieve the segments of the encoded live stream specified in the file above. The video files are at **neunlds134dal.akamaized.net**. The Univision App makes the request for https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts and plays back the file to begin streaming the selected program. The Univision App initially starts with the 800 Bandwidth low resolution version, but on a high-bandwidth connection quickly switches up to the 4500 Bandwidth version (see, for example, retrieval of univision_east_hd_4500_20180725201130.ts). Below is an exemplary .m3u8 file retrieved during the test for the highest resolution 4500 Bandwidth version of the video. #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201050.ts #EXTINF:10,

Claim	Claim Limitation	Example Infringement Evidence
		https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts **Filename**: univision_east_hd_4500_pc.m3u8 As long as the viewer stays on the channel and the bandwidth is adequate, the Univision App will continue to request and receive the chosen playlist for the current resolution. A subsequently retrieved playlist's contents are shown below: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:11:10Z #EXT-X-MEDIA-SEQUENCE:4245797 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10,

Claim	Claim Limitation	Example Infringement Evidence
		https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201140.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201150.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201200.ts **Filename:** univision_east_hd_4500_pc.m3u8 playlist The last video segments "**univision_east_hd_4500_20180725201140.ts"**, and "**univision_east_hd_4500_20180725201150.ts"**, and "**univision_east_hd_4500_20180725201200**" are new additions since the last playlist file was retrieved by the Univision App. The Univision App continues to request, receive, and playback successive video files to show the program. Before the Univision server encodes each of the .ts files, the contents of those files are raw streamlets. Collectively, the raw streamlet store data to playback the entire video (the selected video program).
	encoding each raw streamlet to generate, for each of said raw streamlets, a set including an encoded streamlet for each bitrate supported by the multi-bitrate content streaming,	The Univision servers encode each of the raw streamlets discussed above into a set of streamlets for each bitrate supported by the multi-bitrate content streaming. The Univision server encodes the streamlets of the selected video program, "El Gordo y La Flaca," such that the different bandwidth versions to generate a set of encoded streamlets for the supported bitrates. As described above with respect to the.m3u8 playlist file, the versions of the selected video program stream have different bandwidths and resolutions. Two different bandwidth variant playlist files are discussed below. Each of the segments is a portion of the selected video, where the selected video is

Claim	Claim Limitation	Example Infringement Evidence
		the entire program being sent on the selected channel. An exemplary .m3u8 playlist file retrieved during the test for the highest resolution 800 Bandwidth version of the video. #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=86c127a7cebe3eaf0e1a6a3eb06fcab6",IV=0x8C675584560D97EC2C6461F7C9BDB561 #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201140.ts **Filename:** univision_east_hd_800_pc.m3u8

Claim	Claim Limitation	Example Infringement Evidence
		An exemplary .m3u8 file retrieved during the test for the highest resolution 4500 Bandwidth version of the video. #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts **Filename**: univision_east_hd_4500_pc.m3u8 The Univision server provides files with similar file names (for example, here, “univision_east_hd_800_20180725201130.ts” and “univision_east_hd_4500_20180725201130.ts”)

Claim	Claim Limitation	Example Infringement Evidence
		in the same directory (here, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/) corresponding to separate resolution versions of the same segment. The Univision servers generate, for each of said raw streamlets, a set include an encoded streamlet for each bitrate supported by the multi-bitrate content streaming.
	wherein each of the encoded streamlets in each of the sets is stored as a separate content file,	Each of the encoded streamlets in each of the sets is stored as a separate content file. The Univision server stores files with similar file names (for example, here, "univision_east_hd_800_20180725201130.ts" and "univision_east_hd_4500_20180725201130.ts") in the same directory (here, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/) corresponding to separate resolution versions of the same segment.
	wherein the encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the single video,	The encoded streamlets within each of the sets have the same time index as their corresponding raw streamlet such that the encoded streamlets of the same set independently yield on playback the same portion of the single video. The encoded .ts files have the same time indexes as the corresponding raw streamlets from which they were encoded.
	wherein the separate content files within each of the sets are independently	The separate content files within each of the sets are independently requestable by the end user station. The Univision App makes requests for sequential segments of the stream (according to the variant

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td>requestable by end user stations, and</td><td>playlist file file) for the current channel over the network connection. The requests are sent to the Univision server and the Univision server responds with the requested segment.

In the sequence below, the Univision App requests and receives the univision_east_hd_800_20180725201050.ts and then the univision_east_hd_4500_20180725201100.ts segment. The filenames univision_east_hd_800_20180725201050.ts and univision_east_hd_4500_20180725201100.ts are separate content files. Each of the sets of files are independently requestable by the Univision App. As discussed above, the Univision server has six versions of each of these segments stored.</td></tr>
<tr><td></td><td>wherein shifts between the different bit rates are made at the time indexes during streaming of the single video;</td><td>The Univision App shifts between the different bit rates are made at the time indexes during streaming of the single video.

Different resolutions of the program were requested and displayed by the Univision App based on the bandwidth available to the computer. The Univision App plays segments at any of the six bandwidths and resolutions, and the segments can be played independently. That is, a 4500 Bandwidth segment may be played immediately after an 800 Bandwidth segment.

When the Univision App runs at full bandwidth, the Univision App transitions from the low-bitrate version of the video (the 800 Bandwidth version) to the high-bitrate version of the video (the 4500 Bandwidth version). Between these two segments, the Univision App requests and receives a 4500 Bandwidth variant playlist file (the “univision_east_hd_4500_pc.m3u8”) of the video:
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts</td></tr>
<tr><td>GET</td><td>univisionadsvr01v2.nlst.neulion.com</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&d</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>c=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts</td></tr>
</table>

As demonstrated above, the Univision App requests and receives the higher resolution encoded files after initially requesting a lower resolution encoded file. In this way, the Univision App shifts between the different bit rates during streaming of the single video.

These shift takes place at time indexes. For example, the Univision App requested the low bandwidth version of the univision_east_hd_800_20180725201120.ts segment file and the high bandwidth version of the univision_east_hd_4500_20180725201130.ts segment file.
</td></tr>
<tr><td></td><td>receiving requests from the end user stations over the Internet for different ones of the separate content files from different ones of the sets; and</td><td>
The Univision server receives requests from the end user stations (the Univision App) over the Internet for different ones of the separate content files from different ones of the sets.

The Univision App makes requests for sequential segments of the stream (according to the variant playlist file file) for the current channel over the network connection. The requests are sent to the Univision server and the Univision server responds with the requested segment.

In the sequence below the Univision App requests and receives the univision_east_hd_800_20180725201120.ts segment and then the univision_east_hd_4500_20180725201130.ts segment. The filenames univision_east_hd_800_20180725201120.ts and univision_east_hd_4500_20180725201130.ts are separate content files. As discussed above, the Univision server has six versions of each of these files for each time index. Between these two segments, the Univision App requests and receives a variant playlist file (“univision_east_hd_4500_pc.m3u8”).
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaize</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_h</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td>d.net</td><td>d_800_20180725201120.ts</td></tr>
<tr><td>GET</td><td>univisionadsvr01v2.nlst.neulion.com</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts</td></tr>
</table>
</td></tr>
<tr><td></td><td>transmitting the requested ones of the separate content files to the requesting one of the end user stations,</td><td>
The Univision server transmits the requested ones of the separate content files to the requesting one of the end user stations.

The Univision App makes requests for sequential segments of the stream (according to the variant playlist file file) for the current channel over the network connection. The requests are sent to the Univision server and the Univision server responds with the requested segment.

In the sequence below the Univision App requests and receives the univision_east_hd_800_20180725201120.ts segment and then the univision_east_hd_4500_20180725201130.ts segment. The filenames univision_east_hd_800_20180725201120.ts and univision_east_hd_4500_20180725201130.ts are separate content files. As discussed above, the Univision server has six versions of each of these files for each time index. Between these two segments, the Univision App requests and receives a variant playlist file (“univision_east_hd_4500_pc.m3u8”).
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>GET</td><td>univisionadsvr01v2.nlst.neulion.com</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts</td></tr>
</table>
</td></tr>
<tr><td></td><td>wherein each of the end user stations initiate each of the shifts between the different bit rates during streaming of the single video through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.</td><td>
Each of the end user stations (Univision App) initiate each of the shifts between the different bit rates during streaming of the single video through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

The Univision App initiates each of the shifts between the different bit rates during streaming of the single video through a request for the separate content file storing a different bit rate one of the encoded streamlets for a subsequent one of the time indexes.

Successive determinations by the content player shift the playback quality to a higher or lower quality one of the different copies of the same selected video by the Univision App. When the Univision App runs at full bandwidth, the Univision App transitions from the low-bitrate version of the video (the 800 Bandwidth version) to the high-bitrate version of the video (the 4500 Bandwidth version). Between these two segments, the Univision App requests and receives a 4500 Bandwidth variant playlist file (the “univision_east_hd_4500_pc.m3u8”) of the video:
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts</td></tr>
<tr><td>GET</td><td>univisionadsvr01v2.nlst.</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_h</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td>neulion.com</td><td>d_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts</td></tr>
</table>

As demonstrated above, the Univision App requests and receives the higher resolution encoded files after initially requesting a lower resolution encoded file. In this way, the Univision App adapts subsequent segment requests based on a determination to shift the playback to a higher or lower quality encoding.

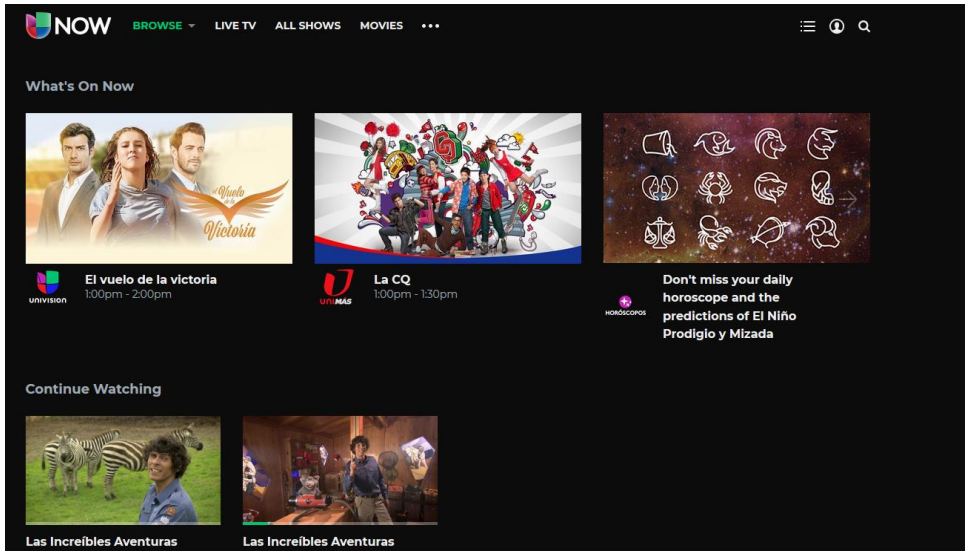
These shift takes place at time indexes. For example, the Univision App requested the low bandwidth version of the univision_east_hd_800_20180725201120.ts segment file and the high bandwidth version of the univision_east_hd_4500_20180725201130.ts segment file.
</td></tr>
</table>

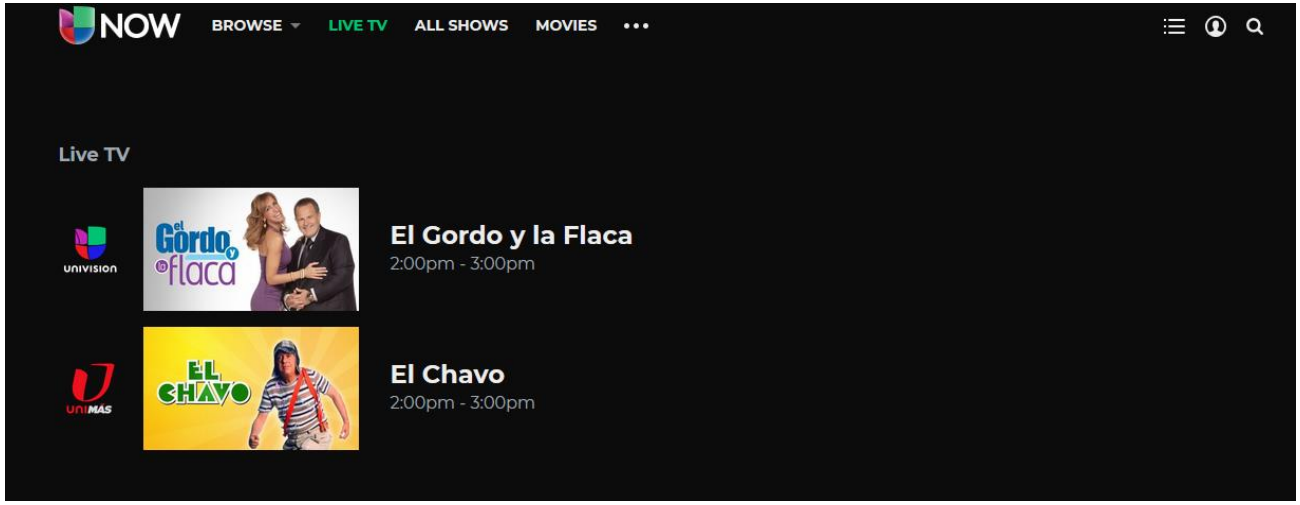
EXHIBIT G

U.S. Patent No. 8,402,156 to Univision

The following claim chart shows exemplary aspects of the Univision Now Application ("Univision App"), Univision servers, and Univision services that infringe the claim below. The chart is exemplary and should not be read to limit DISH's claims against Univision to the products or services described below. The chart should also not be read to limit DISH's claims to the patent claim charted below. Nor should the chart below be read to limit how the Univision App, Univision servers, and Univision services infringe the claim below. DISH's claims against Univision may change based on the proceeding in this case and based on discovery provided by Univision.

Certain portions of the evidence below have been redacted.

Claim	Claim Limitation	Example Infringement Evidence
15	15. A method for adaptive-rate content streaming videos for playback on a content player on an end user station, the method comprising:	The Univision servers perform a method adaptive-rate content streaming videos for playback on the Univision App. The Univision App is a content player that obtains streams of selected video program for playback from the Univision servers. The screenshots in this chart are from the Univision App running on a PC Firefox browser. The main menu of the Univision App after logging in:  As described in greater detail below, the Univision App executes a method to play back live streaming and VOD programming over a network connection.
	receiving a selected one of the videos for generating streamlets for adaptive-rate content streaming; and	The Univision servers receive a selected one of the videos for generating streamlets for adaptive-rate content streaming. The Univision App, based on user input, requests a stream of a selected video program via a network connection to a Univision server.

Claim	Claim Limitation	Example Infringement Evidence
		A user uses the on-screen guide to select a channel for streaming. First, the user selects "Live TV" from the Univision App main screen shown above. The Univision App offers both streaming of current Univision programming and VOD selections. The live streaming selection is shown by the "What's On Now" menu, or the "Live TV" link at the top of the page: 
	creating a plurality of different copies of the same selected video, wherein each of the different copies is encoded at a different bit rate and is divided into a plurality of	The Univision servers create a plurality of different copies of the same selected video. Each of the different copies created by the Univision server is encoded at a different bit rate and is divided into a plurality of streamlets that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the selected video. When the Univision App accesses a selected channel, it requests and receives a playlist file from the Univision server that shows the available versions of the channel at different resolutions. For example, a user navigates the menu above to select "Live TV" in the menu above and selects the

Claim	Claim Limitation	Example Infringement Evidence
	streamlets that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire duration of the selected video,	stream of "**El Gordo y La Flaca**" as the "selected video program." The Univision App makes an HTTPS GET request to **univisionadsvr01v2.nlst.neulion.com** for a master playlist named "**univision_east_hd_pc.m3u8**" that specifies the available streams and provides links to the playlists for those streams. The Univision server returns the following playlist named "**univision_east_hd_pc.m3u8**": #EXTM3U #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=800000 univision_east_hd_800_pc.m3u8?x-b=800000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=400000 univision_east_hd_400_pc.m3u8?x-b=400000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1200000 univision_east_hd_1200_pc.m3u8?x-b=1200000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1600000 univision_east_hd_1600_pc.m3u8?x-b=1600000&x-i=d118292fb7379d90ed41b5783b409b21&x-

Claim	Claim Limitation	Example Infringement Evidence
		c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=3000000 univision_east_hd_3000_pc.m3u8?x-b=3000000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=4500000 univision_east_hd_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 **Filename:** univision_east_hd_pc.m3u8 This is a master playlist file according to the HLS specification.[1] The master playlist shows six versions of the stream at the following bandwidths: • 800000 (Referred to herein as "**800 Bandwidth**") • 400000 (Referred to herein as "**400 Bandwidth**") • 1200000 (Referred to herein as "**1200 Bandwidth**") • 1600000 (Referred to herein as "**1600 Bandwidth**") • 3000000 (Referred to herein as "**3000 Bandwidth**") • 4500000 (Referred to herein as "**4500 Bandwidth**")

[1] RFC 8216 (HLS Live Streaming), Section 4.3.4 (Master Playlist Tags).

Claim	Claim Limitation	Example Infringement Evidence
		For each of these versions, the master playlist provides a link to a playlist for the specified version of the selected video program at a particular bandwidth and resolution. The Univision App initially selects the 800 Bandwidth version of the stream and makes a request to **univisionadsvr01v2.nlst.neulion.com** for the corresponding variant playlist file named "**univision_east_hd_800_pc.m3u8**." The server returns the file with the following contents: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=86c127a7cebe3eaf0e1a6a3eb06fcab6",IV=0x8C675584560D97EC2C6461F7C9BDB561 #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201140.ts

Claim	Claim Limitation	Example Infringement Evidence
		Filename: univision_east_hd_800_pc.m3u8 As noted above, is an HLS playlist. Each line in the file "**univision_east_hd_800_pc.m3u8**" that begins with "#EXTINF" specifies the length of the segments in seconds. The line below the #EXTINF file is the location of the video file. The Univision App uses HTTPS GETs to request and retrieve the segments of the encoded live stream specified in the file above. The video files are at **neunlds134dal.akamaized.net**. The Univision App makes the request for https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts and plays back the file to begin streaming the selected program. The Univision App initially starts with the 800 Bandwidth low resolution version, but on a high-bandwidth connection quickly switches up to the 4500 Bandwidth version (see, for example, retrieval of univision_east_hd_4500_20180725201130.ts). Below is an exemplary .m3u8 file retrieved during the test for the highest resolution 4500 Bandwidth version of the video. #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201050.ts

Claim	Claim Limitation	Example Infringement Evidence
		#EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts **Filename**: univision_east_hd_4500_pc.m3u8 As long as the viewer stays on the channel and the bandwidth is adequate, the Univision App will continue to request and receive the chosen playlist for the current resolution. A subsequently retrieved playlist's contents are shown below: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:11:10Z #EXT-X-MEDIA-SEQUENCE:4245797 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts

Claim	Claim Limitation	Example Infringement Evidence
		#EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201140.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201150.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201200.ts **Filename:** univision_east_hd_4500_pc.m3u8 playlist The last video segments "**univision_east_hd_4500_20180725201150.ts"**, and "**univision_east_hd_4500_20180725201140.ts"**, and "**univision_east_hd_4500_20180725201200**" are new additions since the last playlist file was retrieved by the Univision App. The Univision App continues to request, receive, and playback successive video files to show the program. Before the Univision server encodes each of the .ts files, the contents of those files are raw streamlets. Collectively, the raw streamlet store data to playback the entire video (the selected video program).
	wherein the time indexes of the streamlets are the same for the different copies such that streamlets with the same time indexes form the different copies independently	The time indexes of the streamlets are the same for the different copies such that streamlets with the same time indexes form the different copies independently yield the same portions of the selected video. The Univision server encodes the streamlets of the selected video program, "El Gordo y La Flaca," such that the different bandwidth versions to generate a set of encoded streamlets for the supported bitrates. As described above with respect to the playlist.m3u8 manifest file, the versions of the selected video

Claim	Claim Limitation	Example Infringement Evidence
	yield the same portions of the selected video, and	program stream have different bandwidths and resolutions. Two different bandwidth chunklists are discussed below. Each of the segments is a portion of the selected video, where the selected video is the entire program being sent on the selected channel. An exemplary .m3u8 file retrieved during the test for the highest resolution 800 Bandwidth version of the video. #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=86c127a7cebe3eaf0e1a6a3eb06fcab6",IV=0x8C675584560D97EC2C6461F7C9BDB561 #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201140.ts

Claim	Claim Limitation	Example Infringement Evidence
		Filename: univision_east_hd_800_pc.m3u8 An exemplary .m3u8 file retrieved during the test for the highest resolution 4500 Bandwidth version of the video. #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts **Filename**: univision_east_hd_4500_pc.m3u8

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>The Univision server provides files with similar file names (for example, here, “univision_east_hd_800_20180725201130.ts” and “univision_east_hd_4500_20180725201130.ts”) in the same directory (here, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/) corresponding to separate resolution versions of the same segment.

The Univision servers generate, for each of said raw streamlets, a set include an encoded streamlet for each bitrate supported by the multi-bitrate content streaming.</td></tr>
<tr><td></td><td>wherein each of the streamlets of each of the pluralities is a separate content file that is independently playable by the end user station to thereby allow the end user station to initiate shifts in playback quality during streaming of the selected video through requests for separate content files storing different playback qualities of the encoded streamlets for subsequent ones of the time indexes.</td><td>Each of the streamlets of each of the pluralities is a separate content file that is independently playable by the Univision App. The Univision App initiates shifts in playback quality during streaming of the selected video through requests for separate content files storing different playback qualities of the encoded streamlets for subsequent ones of the time indexes.

Different resolutions of the program were requested and displayed by the Univision App when the bandwidth to the computer was limited to force adaptation to a lower bit rate version of the program. The Univision App plays segments at any of the six bandwidths and resolutions, and the segments can be played independently. That is, a 4500 Bandwidth segment may be played immediately after an 800 Bandwidth segment.

When the Univision App runs at full bandwidth, the Univision App transitions from the low-bitrate version of the video (the 800 Bandwidth version) to the high-bitrate version of the video (the 4500 Bandwidth version). Between these two segments, the Univision App requests and receives a 4500 Bandwidth chunklist (the “univision_east_hd_4500_pc.m3u8”) of the video:
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts</td></tr>
<tr><td>GET</td><td>univisionadsvr01v2.nlst.neulion.com</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts</td></tr>
</table>

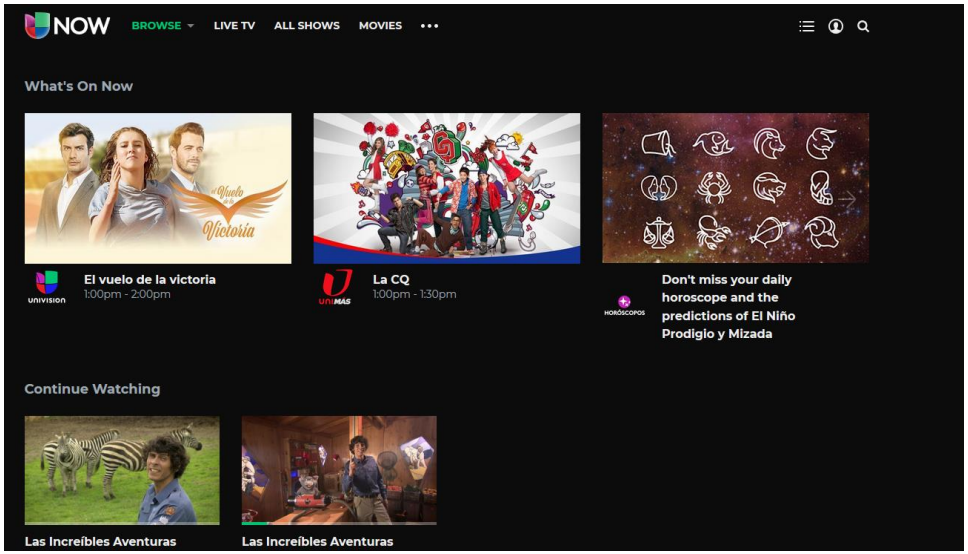
As demonstrated above, the Univision App requests and receives the higher resolution encoded files after initially requesting a lower resolution encoded file. In this way, the Univision App shifts between the different bit rates during streaming of the single video.

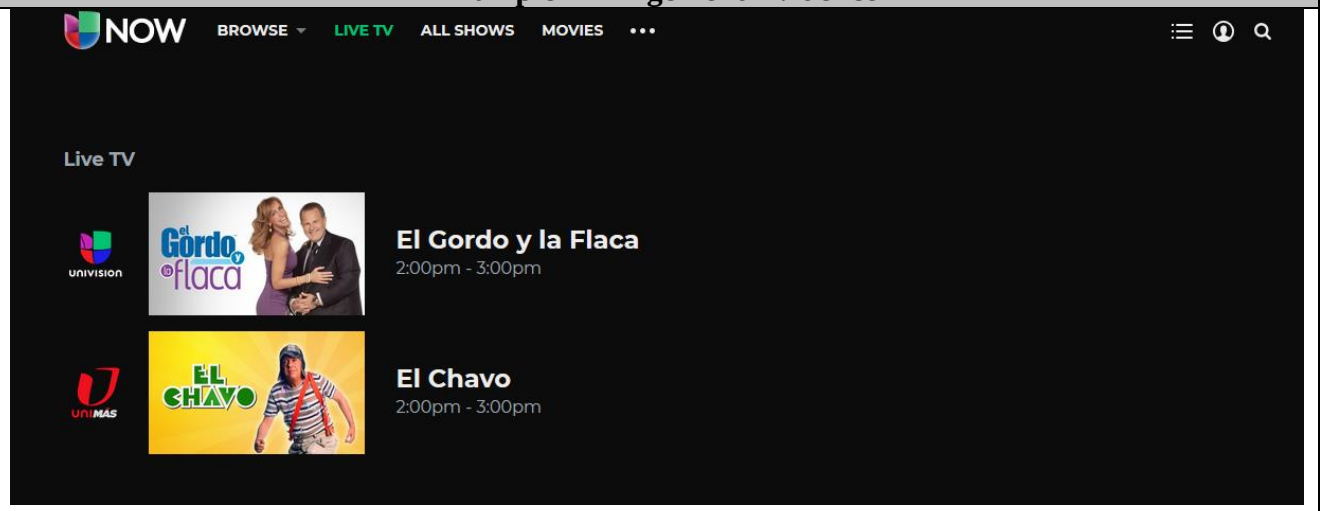
These shift takes place at time indexes. For example, the Univision App requested the low bandwidth version of the univision_east_hd_800_20180725201120.ts segment file and the high bandwidth version of the univision_east_hd_4500_20180725201130.ts segment file.
</td></tr>
</table>

EXHIBIT H

U.S. Patent No. 9,071,668 to Univision

The following claim chart shows exemplary aspects of the Univision Now Application ("Univision App"), Univision servers, and Univision services that infringe the claim below. The chart is exemplary and should not be read to limit DISH's claims against Univision to the products or services described below. The chart should also not be read to limit DISH's claims to the patent claim charted below. Nor should the chart below be read to limit how the Univision App, Univision servers, and Univision services infringe the claim below. DISH's claims against Univision may change based on the proceeding in this case and based on discovery provided by Univision.

Claim	Claim Limitation	Example Infringement Evidence
16	16. A method executable by a content player on an end user device to obtain a stream of a selected video program for playback by the content player, the method comprising:	The Univision App is a content player that obtains streams of selected video program for playback. The screenshots in this chart are from the Univision App running on a PC Firefox browser. The main menu of the Univision App after logging in:  The Univision App offers both streaming of current Univision programming and VOD selections. The live streaming selection is shown by the “What’s On Now” menu, or the “Live TV” link at the top of the page:

Claim	Claim Limitation	Example Infringement Evidence
		 As described in greater detail below, the Univision App executes a method to play back live streaming and VOD programming over a network connection.
	requesting the stream of the selected video program via a network connection to a video server,	The Univision App requests a stream of a "selected video program via a network connection to a video server. " A user navigates the menu above to select "Live TV" in the menu above. The stream of "**El Gordo y La Flaca**" is the "selected video program." The Univision App makes an HTTPS GET request to **univisionadsvr01v2.nlst.neulion.com** for a master playlist named "**univision_east_hd_pc.m3u8**" that specifies the available streams and provides links to the playlists for those streams. The server returns the following playlist named "**univision_east_hd_pc.m3u8**": #EXTM3U #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=800000

Claim	Claim Limitation	Example Infringement Evidence		
			univision_east_hd_800_pc.m3u8?x-b=800000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=400000 univision_east_hd_400_pc.m3u8?x-b=400000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1200000 univision_east_hd_1200_pc.m3u8?x-b=1200000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1600000 univision_east_hd_1600_pc.m3u8?x-b=1600000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=3000000 univision_east_hd_3000_pc.m3u8?x-b=3000000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5	

Claim	Claim Limitation	Example Infringement Evidence
		#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=4500000 univision_east_hd_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 **Filename:** univision_east_hd_pc.m3u8 This is a master playlist file according to the HLS specification.[1] The master playlist shows that there are six versions of the stream at the following bandwidths: • 800000 (Referred to herein as "**800 Bandwidth**") • 400000 (Referred to herein as "**400 Bandwidth**") • 1200000 (Referred to herein as "**1200 Bandwidth**") • 1600000 (Referred to herein as "**1600 Bandwidth**") • 3000000 (Referred to herein as "**3000 Bandwidth**") • 4500000 (Referred to herein as "**4500 Bandwidth**") For each of these versions, the master playlist provides a link to a playlist for the specified version of the selected video program at a particular bandwidth and resolution, which is called a "variant" in HLS. For this test, the Univision App initially selects the 800 Bandwidth version of the stream and makes a request to **univisionadsvr01v2.nlst.neulion.com** for the corresponding variant playlist file named "**univision_east_hd_800_pc.m3u8**." The server returns the variant playlist with the following contents: #EXTM3U

[1] RFC 8216 (HLS Live Streaming), Section 4.3.4 (Master Playlist Tags).

Claim	Claim Limitation	Example Infringement Evidence
		#EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=86c127a7cebe3eaf0e1a6a3eb06fcab6",IV=0x8C675584560D97EC2C6461F7C9BDB561 #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201140.ts **Filename:** univision_east_hd_800_pc.m3u8 As noted above, the file "**univision_east_hd_800_pc.m3u8**" is an HLS variant playlist. Each line that begins with "#EXTINF" specifies the length of the segments in seconds. The line below the #EXTINF file is the location of the video file. The Univision App uses HTTPS GETs to request and retrieve the segments of the encoded live stream specified in the file above. The video files are at **neunlds134dal.akamaized.net**.

Claim	Claim Limitation	Example Infringement Evidence
		The Univision App makes the request for https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts and plays back the file to begin streaming the selected program. The Univision App initially starts with the 800 Bandwidth low resolution version, but on a high-bandwidth connection quickly switches up to the 4500 Bandwidth version (see, for example, retrieval of univision_east_hd_4500_20180725201130.ts). Below is an exemplary variant playlist file retrieved during the test for the highest resolution 4500 Bandwidth version of the video. #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10,

Claim	Claim Limitation	Example Infringement Evidence
		https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts **Filename**: univision_east_hd_4500_pc.m3u8 As long as the viewer stays on the channel and the bandwidth is adequate, the Univision App will continue to request and receive the chosen playlist for the current resolution. The Univision server updates the variant file to indicate what the Univision App should retrieve and playback next. A subsequently retrieved variant playlist's contents are shown below: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:11:10Z #EXT-X-MEDIA-SEQUENCE:4245797 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201140.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_e

Claim	Claim Limitation	Example Infringement Evidence
		ast_hd_4500_20180725201150.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201200.ts **Filename:** univision_east_hd_4500_pc.m3u8 playlist The last video segments "**univision_east_hd_4500_20180725201150.ts"**, and "**univision_east_hd_4500_20180725201140.ts"**, and "**univision_east_hd_4500_20180725201200**" are new additions since the last playlist file was retrieved by the Univision App. The Univision App continues to request, receive, and playback successive video files to show the program.
	wherein the video server accesses a plurality of different copies of the same selected video each encoded at a different bit rate and each divided into a plurality of segments that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire playback duration of the selected video,	The Univision servers access a plurality of versions of the same selected video program encoded at different bit rates, based on requests from the Univision App. Each of the versions of the selected video program are encoded at a different bit rate and each divided into a plurality of segments that collectively store data to playback the entire video but that individually store data to playback only a portion that starts at a unique time index and whose duration is less than the entire playback duration of the selected video. The playlists from the Univision servers provide the contents of a "stream." The playlist shows which segments will be retrieved and loaded over time. When the bandwidth changed, the playlists show that the Univision video server has video segment files with similar names that are referenced in playlists for different resolutions. Compare a received playlist for two different bit rates with the following contents: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionn

Claim	Claim Limitation	Example Infringement Evidence
		ow2/univision_east/as/live/univision_east_hd_800/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=86c127a7cebe3eaf0e1a6a3eb06fcab6",IV=0x8C675584560D97EC2C6461F7C9BDB561 #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201140.ts **Filename:** univision_east_hd_800_pc.m3u8 The segments for the 800 Bandwidth version of the stream begin with the filename "**univision_east_hd_800_**" (referred to herein as the "**800 Identifier**"). Compare the 800 Bandwidth version of the stream with the files in the 4500 Bandwidth version of the stream. #EXTM3U #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-

Claim	Claim Limitation	Example Infringement Evidence
		128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts **Filename:** univision_east_hd_4500_pc.m3u8 The segments for the 4500 Bandwidth version begin with the filename "**univision_east_hd_4500_**" (referred to herein as the "4500 Identifier.") These two playlists refer to segments with identical filenames and identical lengths (in seconds) with two different prefixes. In particular, files with names containing "**20180725201050.ts**," "**20180725201100.ts**," "**20180725201110.ts**," and "**20180725201120.ts**" (after the 800 Identifier and 4500 Identifier), are all 10 seconds in length and are available for both bandwidths (and corresponding resolutions and bit rates). The Univision servers access the various versions of the segments with different bandwidths and different resolutions based on requests from the Univision App. The 800 Bandwidth version of the segments of the program are encoded and stored with a prefix of "**univision_east_hd_800_**" and the

Claim	Claim Limitation	Example Infringement Evidence
		4500 Bandwidth version of the segments of the program are stored with a prefix of "**univision_east_hd_4500_**." These versions have different bit rates. The identical segments of the filenames after the 800 Identifier and 4500 Identifier demonstrate that these segments are copies of the same video. The 10 second segments' lengths demonstrate that they are shorter than the length of the selected program (the show airing on Univision East).
	wherein the time indexes of the segments are the same for the different copies such that the segments with the same time indexes from the different copies independently yield the same portions of the selected video,	The time indexes of the segments are the same for the different copies such that the segments with the same time indexes from the different copies independently yield the same portions of the selected video. Each of the copies, when played back by the Univision App, yields the same portions of the selected video. A received playlist for the 1600 Bandwidth version has the following contents: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:31:33Z #EXT-X-MEDIA-SEQUENCE:4245929 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=140e8dc7ef6c4bcd603e3ec1378a0266",IV=0xE97C12A3AE6BB88448F90DE80FFCE7E8 #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725203130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725203140.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725203150.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_e

Claim	Claim Limitation	Example Infringement Evidence
		ast_hd_1600_20180725203200.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725203210.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725203220.tshttps://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725203140.tsFilename: univision_east_hd_1600_pc.m3u8 **Filename**: univision_east_hd_1600_pc.m3u8 The segments for the 1600 Bandwidth version of the stream begin with the filename "**univision_east_hd_1600_**." A close-in-time playlist for the 4500 Bandwidth Version has the following contents: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:31:53Z #EXT-X-MEDIA-SEQUENCE:4245931 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725203150.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725203200.ts #EXTINF:10,

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725203210.ts
#EXTINF:10,
https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725203220.ts
#EXTINF:10,
https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725203230.ts
#EXTINF:10,
https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725203240.ts</td></tr>
</table>

Filename: univision_east_hd_4500_pc.m3u8

The segments for the 4500 Bandwidth version of the stream begin the filename “univision_east_hd_4500_.”

These two playlists refer to segments with similar filenames and identical lengths (in seconds) in the same directory, with each set of segments being identified by the 800 Identifier or the 4500 Identifier. In particular, filenames prepended with the 800 Identifier and 4500 Identifier, 20180725203200.ts, 20180725203210.ts, and 20180725203220.ts, are all 10 seconds in length and are available for both resolutions.
</td></tr>
<tr><td></td><td>and wherein each of the segments of each of the pluralities is a separate content file that is independently playable by the end user device,</td><td>
Each of the versions of the encoded segment is a separate content file that is playable by the Univision App. During testing, different resolutions of the program were requested and displayed by the Univision App when the bandwidth to the computer was limited to force adaptation to a lower bit rate version of the program. The Univision App plays segments at any of the six bandwidths and resolutions, and the segments can be played independently. For example, a 4500 Bandwidth segment may be played immediately after a 800 Bandwidth segment.
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>GET</td><td>univisionadsvr01v2.nlst.neulion.com</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts</td></tr>
</table>
The Univision App runs on an end user device (for example, a laptop, tablet, phone, or STB).
</td></tr>
<tr><td></td><td>wherein the requesting comprises the content player placing, for a set of sequential ones of the time indexes, segment requests over the network connection to the video server to thereby retrieve the separate segments from at least one of the different copies storing the portions of the single video according to the set of time indexes;</td><td>
The Univision App makes requests for sequential segments of the stream (according to the variant file) for the current channel over the network connection. The requests are sent to the Univision server and the Univision server responds with the requested segment.

Below is an excerpt of the Firefox “Sequence” listing showing the Univision App requesting and receiving three sequential segments of the program: **20180725201140.ts**, **20180725201150.ts,** and **20180725201200.ts**.
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201140.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201150.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201200.ts</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td>receiving the separate segments from the video server at the content player via the network connection; and</td><td>The Univision App receives the separate requested segments from the Univision video server via the network connection.

Below is an excerpt of the Firefox sequence listing showing the Univision App requesting and receiving three sequential segments of the program: “univision_east_hd_4500_20180725201140.ts,” “univision_east_hd_4500_20180725201150.ts,” and “univision_east_hd_4500_20180725201200.ts.”
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201140.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201150.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201200.ts</td></tr>
</table></td></tr>
<tr><td></td><td>adapting subsequent segment requests placed by the content player to the video server based upon successive determinations by the content player to shift the playback quality to a higher or lower quality one of the different copies of the same selected video, wherein the shifts in</td><td>As part of the testing, the computer running the Univision App connected to the Internet though the Firefox browser. In order to demonstrate the bandwidth adaptation, the throttling feature of the Firefox browser was used to limit the Univision App’s bandwidth to approximate DSL speeds, and then the throttling setting was removed.

When the bandwidth for the Univision App is reduced, the Univision App engages in adaptation to request a lower bit rate version of the content at the next time index. Recall from the discussion about the playlist.m3u8 file that the variant playlists and segments are stored with different prefixes as part of the filename, based on the resolution and bandwidth.
<table>
<tr><th>Bandwidth</th><th>Playlist Filename</th></tr>
<tr><td>800000</td><td>univision_east_hd_800_pc.m3u8</td></tr>
<tr><td>400000</td><td>univision_east_hd_400_pc.m3u8</td></tr>
<tr><td>1200000</td><td>univision_east_hd_1200_pc.m3u8</td></tr>
</table></td></tr>
</table>

Claim	Claim Limitation	Example Infringement Evidence
	playback quality occur at the time indexes.	

1600000	univision_east_hd_1600_pc.m3u8
3000000	univision_east_hd_3000_pc.m3u8
4500000	univision_east_hd_4500_pc.m3u8

The chosen resolution of the Univision App can be determined based on which playlist and associated segments of media are retrieved. A portion of the Firefox sequence listing shows requested and retrieved files just prior to and just after bandwidth was constrained is shown below.

Method	Host	Path
GET	neunlds134dal.akamaized.net	/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725210830.ts
GET	univisionadsvr01v2.nlst.neulion.com	/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_pc.m3u8?x-b=1600000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5
GET	univisionadsvr01v2.nlst.neulion.com	/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_pc.m3u8?x-b=800000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5
GET	neunlds134dal.akamaized.net	/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725210840.ts
GET	neunlds134dal.akamaize	/nlds/univisionnow2/univision_east/as/live/univision_east_h

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td>d.net</td><td>d_1600_20180725210850.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725210900.ts</td></tr>
</table>

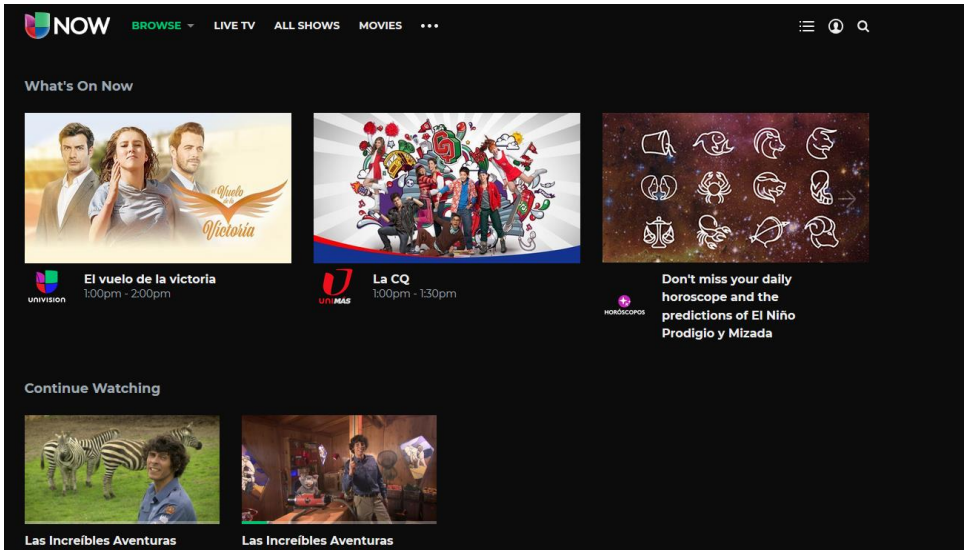
As demonstrated above, the Univision App requests and receives the lower resolution encoded files while the bandwidth is constrained. In this way, the Univision App adapts subsequent segment requests based on a determination to shift the playback to a higher or lower quality encoding.

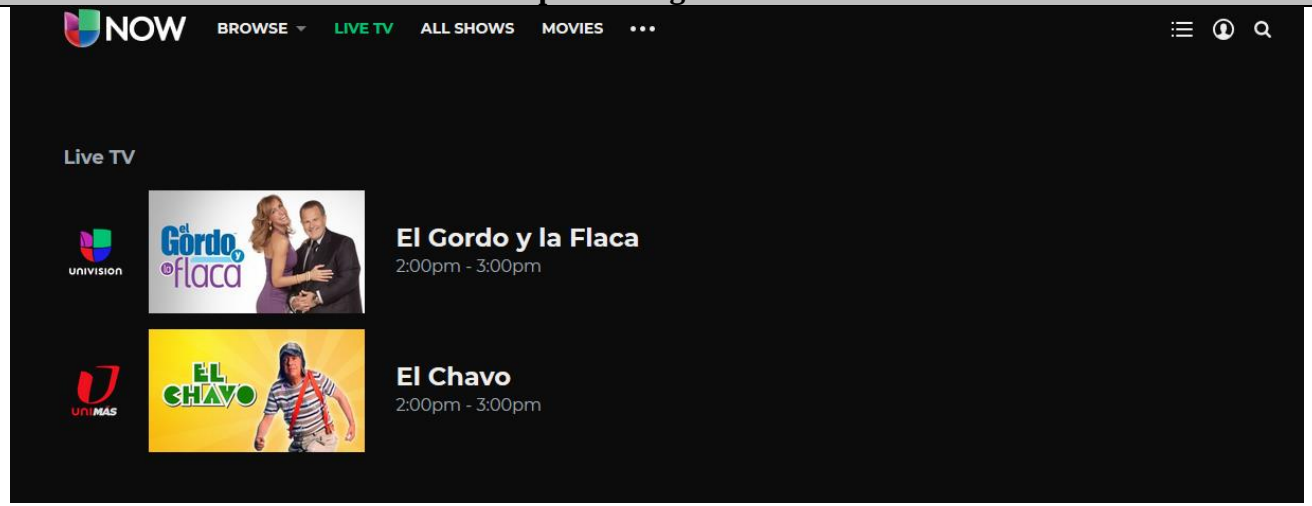
These shifts take place at time indices. For example, the Univision App requested and played back the 4500 Bandwidth of **20180725210830.ts**, then the 800 Bandwidth version of **20180725210840.ts**, and then the 1600 Bandwidth version of the **20180725210850.ts** and **20180725210900.ts** segments.
</td></tr>
</table>

EXHIBIT I

U.S. Patent No. 9,407,564 to Univision

The following claim chart shows exemplary aspects of the Univision Now Application ("Univision App"), Univision servers, and Univision services that infringe the claim below. The chart is exemplary and should not be read to limit DISH's claims against Univision to the products or services described below. The chart should also not be read to limit DISH's claims to the patent claim charted below. Nor should the chart below be read to limit how the Univision App, Univision servers, and Univision services infringe the claim below. DISH's claims against Univision may change based on the proceeding in this case and based on discovery provided by Univision.

Claim	Claim Limitation	Example Infringement Evidence
8	8. A method executable by an end user station to present rate-adaptive streams received via at least one transmission control protocol (TCP) connection with a server over a network, the method comprising;	The Univision App is an executable application that may be executed on an end user station that obtains streams of selected video program for playback. The screenshots in this chart are from the Univision App running on a PC Firefox browser. The main menu of the Univision App after logging in:  The Univision App offers both streaming of current Univision programming and VOD selections. The live streaming selection is shown by the “What’s On Now” menu, or the “Live TV” link at the top of the page:

Claim	Claim Limitation	Example Infringement Evidence
		 As demonstrated in greater detail below, the Univision App presents rate-adaptive streams over a TCP connection with a server over a network.
	streaming, by a media player operating on the end user station, a video from the server via the at least one TCP connection over the network,	The Univision App streams video from the server via the at least one TCP connection over the network. A user navigates the menu above to select "Live TV" in the menu above. The stream of "**El Gordo y La Flaca**" is the "selected video program." The Univision App makes an HTTPS GET request to **univisionadsvr01v2.nlst.neulion.com** for a master playlist named "**univision_east_hd_pc.m3u8**" that specifies the available streams and provides links to the playlists for those streams. The server returns the following playlist named "**univision_east_hd_pc.m3u8**": #EXTM3U #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=800000 univision_east_hd_800_pc.m3u8?x-b=800000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-

Claim	Claim Limitation	Example Infringement Evidence		
			play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=400000 univision_east_hd_400_pc.m3u8?x-b=400000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1200000 univision_east_hd_1200_pc.m3u8?x-b=1200000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1600000 univision_east_hd_1600_pc.m3u8?x-b=1600000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=3000000 univision_east_hd_3000_pc.m3u8?x-b=3000000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=4500000 univision_east_hd_4500_pc.m3u8?x-b=4500000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-	

Claim	Claim Limitation	Example Infringement Evidence
		play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5 **Filename:** univision_east_hd_pc.m3u8 This is a master playlist file according to the HLS specification.[1] The master playlist shows that there are six versions of the stream at the following bandwidths: • 800000 (Referred to herein as "**800 Bandwidth**") • 400000 (Referred to herein as "**400 Bandwidth**") • 1200000 (Referred to herein as "**1200 Bandwidth**") • 1600000 (Referred to herein as "**1600 Bandwidth**") • 3000000 (Referred to herein as "**3000 Bandwidth**") • 4500000 (Referred to herein as "**4500 Bandwidth**") For each of these versions, the master playlist provides a link to a playlist for the specified version of the selected video program at a particular bandwidth and resolution, which is called a "variant" in HLS. For this test, the Univision App initially selects the 800 Bandwidth version of the stream and makes a request to **univisionadsvr01v2.nlst.neulion.com** for the corresponding variant playlist file named "**univision_east_hd_800_pc.m3u8**." The server returns the file with the following contents: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800/m3u8.key&x-

[1] RFC 8216 (HLS Live Streaming), Section 4.3.4 (Master Playlist Tags).

Claim	Claim Limitation	Example Infringement Evidence
		play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=86c127a7cebe3eaf0e1a6a3eb06fcab6",IV=0x8C675584560D97EC2C6461F7C9BDB561 #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201140.ts **Filename:** univision_east_hd_800_pc.m3u8 As noted above, the file "**univision_east_hd_800_pc.m3u8**" is an HLS playlist. Each line that begins with "#EXTINF" specifies the length of the segments in seconds. The line below the #EXTINF file is the location of the video file. The Univision App uses HTTPS GETs to request and retrieve the segments of the encoded live stream specified in the file above. The Univision App makes the request for https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts and plays back the file to begin streaming the selected program. The Univision App initially starts with the 800 Bandwidth low resolution version, but on a high-bandwidth connection quickly switches up to the 4500 Bandwidth version (see, for example, retrieval of

Claim	Claim Limitation	Example Infringement Evidence
		univision_east_hd_4500_20180725201130.ts). Below is an exemplary .m3u8 file retrieved during the test for the highest resolution 4500 Bandwidth version of the video. #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts **Filename**: univision_east_hd_4500_pc.m3u8 As long as the viewer stays on the channel and the bandwidth is adequate, the Univision App will

Claim	Claim Limitation	Example Infringement Evidence
		continue to request and receive the chosen playlist for the current resolution. The Univision server updates the variant file to indicate what the Univision App should retrieve and playback next. A subsequently retrieved playlist's contents are shown below: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:11:10Z #EXT-X-MEDIA-SEQUENCE:4245797 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201140.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201150.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201200.ts **Filename:** univision_east_hd_4500_pc.m3u8 playlist

Claim	Claim Limitation	Example Infringement Evidence
		The last video segments "**univision_east_hd_4500_20180725201150.ts"**, and "**univision_east_hd_4500_20180725201140.ts"**, and "**univision_east_hd_4500_20180725201200**" are new additions since the last playlist file was retrieved by the Univision App. The Univision App continues to request, receive, and playback successive video files to show the program.
	wherein multiple different copies of the video encoded at different bit rates are stored as multiple sets of files on the server,	The Univision servers store and access different copies of the video encoded at different bit rates, based on requests from the Univision App. The playlists from the Univision servers provide the contents of a "stream." The playlist shows which segments will be retrieved and loaded over time. When the bandwidth changed, the playlists show that the Univision video server has video segment files with identical names that are referenced in playlists for different resolutions. Compare a received playlist for two different bit rates with the following contents: #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=86c127a7cebe3eaf0e1a6a3eb06fcab6",IV=0x8C675584560D97EC2C6461F7C9BDB561 #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201110.ts

Claim	Claim Limitation	Example Infringement Evidence
		#EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201130.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725201140.ts **Filename:** univision_east_hd_800_pc.m3u8 The segments for the 800 Bandwidth version of the stream begin with the filename "**univision_east_hd_800_**" (referred to herein as the "**800 Identifier**"). Compare the 800 Bandwidth version of the stream with the files in the 4500 Bandwidth version of the stream. #EXTM3U #EXTM3U #EXT-X-VERSION:4 #EXT-X-TARGETDURATION:10 #EXT-X-PROGRAM-DATE-TIME:2018-07-25T20:10:50Z #EXT-X-MEDIA-SEQUENCE:4245795 #EXT-X-KEY:METHOD=AES-128,URI="/key/nlsk3/nlsk4/hls/securekey?app=dal&id=134&ndks=1&url=/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500/m3u8.key&x-play=ec3f4368146f4d45bda7970ea32ddef2&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&hkst=9fd49b58d0d24a390b231d7a1b43d576",IV=0x015DF1436856D5A2D2AC104FBA13EB5B #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201050.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201100.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_

Claim	Claim Limitation	Example Infringement Evidence
		hd_4500_20180725201110.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201120.ts #EXTINF:10, https://neunlds134dal.akamaized.net/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201130.ts **Filename:** univision_east_hd_4500_pc.m3u8 The segments for the 4500 Bandwidth version begin with the filename "**univision_east_hd_4500_**" (referred to herein as the "4500 Identifier.") These two playlists refer to segments with identical filenames and identical lengths (in seconds) with two different prefixes. In particular, files with names containing "**20180725201050.ts**," "**20180725201100.ts**," "**20180725201110.ts**," and "**20180725201120.ts**" (after the 800 Identifier and 4500 Identifier), are all 10 seconds in length and are available for both bandwidths (and corresponding resolutions and bit rates). The Univision servers store and access the various versions of the segments with different bandwidths and different resolutions based on requests from the Univision App. The 800 Bandwidth version of the segments of the program are encoded and stored with a prefix of "**univision_east_hd_800_**" and the 4500 Bandwidth version of the segments of the program are stored with a prefix of "**univision_east_hd_4500_**." These versions have different bit rates. The identical segments of the filenames after the 800 Identifier and 4500 Identifier demonstrate that these segments are copies of the same video. The 10 second segments' lengths demonstrate that they are shorter than the length of the selected program (the show airing on Univision East). The Univision servers therefore meet the claimed "multiple different copies of the video encoded at different bit rates are stored as multiple sets of files on the server" because: • the Univision servers (located at **neunlds134dal.akamaized.net**) store and access up to six different copies of the video (the six encoded bandwidths and resolutions available by HTTPS GET);

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>the .ts files described above collectively store data to play back the entire video (the entire program being streamed) when each of the 10 second segments are played end-to-end;</td></tr>
<tr><td></td><td>wherein each of the files yields a different portion of the video on playback,</td><td>Each of the files yields a different portion of the video on playback. Below is an excerpt of the Firefox sequence listing showing the Univision App requesting and receiving three sequential segments of the program: “univision_east_hd_4500_20180725201140.ts,” “univision_east_hd_4500_20180725201150.ts,” and “univision_east_hd_4500_20180725201200.ts.” The portion of the filenames after the bandwidth identifier (e.g., “20180725201140.ts,” “20180725201150.ts,” and “20180725201200.ts”) yield different portions of the video on playback. As discussed above, the Univision server has six versions of each of these files for each time index.
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201140.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201150.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201200.ts</td></tr>
</table></td></tr>
<tr><td></td><td>wherein the files across the different copies yield the same portions of the video on playback, and</td><td>The files across the different copies yield the same portions of the video on playback by the Univision App. As described above, each of the playlists includes links to files with the same video content at different resolutions.</td></tr>
<tr><td></td><td>wherein each of the files comprises a time index such that the files whose playback is the same portion of the video for each of the different copies</td><td>Each of the files comprises a time index such that the files whose playback is the same portion of the video for each of the different copies have the same time index in relation to the beginning of the video. As described above, each of the playlists includes links to files with the same video content at different resolutions. Files corresponding to the same time index have the same names after the bandwidth identifier portion of the filename.</td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td>have the same time index in relation to the beginning of the video, and wherein the streaming comprises:</td><td></td></tr>
<tr><td></td><td>requesting by the media player a plurality of sequential files of one of the copies from the server based on the time indexes;</td><td>The Univision App makes requests for sequential segments of the stream (according to the playlist files) for the current video over the network connection. The requests are sent to the Univision server and the Univision server responds with the requested segment.

Below is an excerpt of the Firefox sequence listing showing the Univision App requesting and receiving three sequential segments of the program: “univision_east_hd_4500_20180725201140.ts,” “univision_east_hd_4500_20180725201150.ts,” and “univision_east_hd_4500_20180725201200.ts.” The portion of the filenames after the bandwidth identifier (e.g., “20180725201140.ts,” “20180725201150.ts,” and “20180725201200.ts”) are separate segments for different time indexes.
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201140.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201150.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201200.ts</td></tr>
</table></td></tr>
<tr><td></td><td>automatically requesting by the media player from the server subsequent portions of the video by requesting for each such portion one</td><td>The Univision App automatically requests subsequent portions of the video by requesting for each such portion one of the files from one of the copies dependent upon successive determinations by the media player to shift the playback quality to a higher or lower quality one of the different copies.

Below is an excerpt of the Firefox sequence listing showing the Univision App requesting and receiving three sequential segments of the program: “univision_east_hd_4500_20180725201140.ts,” “univision_east_hd_4500_20180725201150.ts,” and “univision_east_hd_4500_20180725201200.ts.”</td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td>of the files from one of the copies dependent upon successive determinations by the media player to shift the playback quality to a higher or lower quality one of the different copies,</td><td>The portion of the filenames after the bandwidth identifier (e.g., “20180725201140.ts,” “20180725201150.ts,” and “20180725201200.ts”) are separate segments for different time indexes. As discussed above, the Univision server has six versions of each of these files for each time index. Which version of the next segment to request is dependent upon a determination by the media player to shift the playback quality to a higher or lower quality.
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201140.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201150.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725201200.ts</td></tr>
</table></td></tr>
<tr><td></td><td>the automatically requesting including repeatedly generating a factor indicative of the current ability to sustain the streaming of the video using the files from different ones of the copies, wherein the factor relates to the performance of the network; and</td><td>The Univision App repeatedly generates a factor indicative of the current ability to sustain the streaming of the video using the files from different ones of the copies, wherein the factor relates to the performance of the network.

The generation factor indicative of the current ability to sustain streaming of the video using the files from different ones of the copies is demonstrated by testing where the bandwidth available to the Univision App is throttled.

As part of the testing, the computer running the Univision App connected to the Internet though the Firefox browser. In order to demonstrate the bandwidth adaptation, the throttling feature of the Firefox browser was used to limit the Univision App’s bandwidth to approximate DSL speeds, and then the throttling setting was removed.

When the bandwidth for the Univision App is reduced, the Univision App engages in adaptation to request a lower bit rate version of the content at the next time index. Recall from the discussion about the playlist.m3u8 file that the variant playlists and segments are stored with different prefixes as part of the filename, based on the resolution and bandwidth.</td></tr>
</table>

Claim	Claim Limitation	Example Infringement Evidence

Bandwidth	Playlist Filename
800000	univision_east_hd_800_pc.m3u8
400000	univision_east_hd_400_pc.m3u8
1200000	univision_east_hd_1200_pc.m3u8
1600000	univision_east_hd_1600_pc.m3u8
3000000	univision_east_hd_3000_pc.m3u8
4500000	univision_east_hd_4500_pc.m3u8

The chosen resolution of the Univision App can be determined based on which playlist and associated segments of media are retrieved. A portion of the Firefox sequence listing shows requested and retrieved files just prior to and just after bandwidth was constrained is shown below.

Method	Host	Path
GET	neunlds134dal.akamaized.net	/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725210830.ts
GET	univisionadsvr01v2.nlst.neulion.com	/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_pc.m3u8?x-b=1600000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5
GET	univisionadsvr01v2.nlst.neulion.com	/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_pc.m3u8?x-b=800000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725210840.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725210850.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725210900.ts</td></tr>
</table>

As demonstrated above, the Univision App requests and receives the lower resolution encoded files while the bandwidth is constrained. In this way, the Univision App adapts subsequent segment requests based on a determination to shift the playback to a higher or lower quality encoding.

These shifts take place at time indices. For example, the Univision App requested and played back the 4500 Bandwidth of 20180725210830.ts, then the 800 Bandwidth version of 20180725210840.ts, and then the 1600 Bandwidth version of the 20180725210850.ts and 20180725210900.ts segments.
</td></tr>
<tr><td></td><td>making the successive determinations to shift the playback quality based on the factor to achieve continuous playback of the video using the files of the highest quality one of the copies determined sustainable at that time,</td><td>The Univision App makes successive determinations to shift the playback quality based on the factor to achieve continuous playback of the video using the files of the highest quality one of the copies determined sustainable at that time. The shift from the higher to lower version of the selected video demonstrates the determination to shift playback quality on the factor to achieve continuous playback of the video.

As noted above, the Univision App shifts between playback quality based on a factor that includes, for example, bandwidth limitations to enable continuous playback of the video content. Further, the Univision App requests a lower (but not the lowest) quality content. Accordingly, the Univision App continues to request the highest quality content that is sustainable when doing so.</td></tr>
<tr><td></td><td>wherein the making</td><td>The Univision App makes the successive determinations to shift comprises upshifting to a higher</td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td>the successive determinations to shift comprises upshifting to a higher quality one of the different copies when the at least one factor is greater than a first threshold and downshifting to a lower quality one of the different copies when the at least one factor is less than a second threshold; and</td><td>quality one of the different copies when the at least one factor is greater than a first threshold and downshifting to a lower quality one of the different copies when the at least one factor is less than a second threshold. This is demonstrated by the Univision App shifting from the highest to mid quality version based on bandwidth constraints.

For example, when the Univision App is allowed to run at the full bandwidth, the first version retrieved is the highest resolution the Univision App requested and played back the 4500 Bandwidth of **20180725210830.ts**. Once the bandwidth is throttled, the Univision App requested and played back the the 800 Bandwidth version of **20180725210840.ts**.
<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_4500_20180725210830.ts</td></tr>
<tr><td>GET</td><td>univisionadsvr01v2.nlst.neulion.com</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_pc.m3u8?x-b=1600000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5</td></tr>
<tr><td>GET</td><td>univisionadsvr01v2.nlst.neulion.com</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_pc.m3u8?x-b=800000&x-i=d118292fb7379d90ed41b5783b409b21&x-c=UNIVISIONNOW&ppid=267143a8793d41b89fd021eef9c45198&x-play=ec3f4368146f4d45bda7970ea32ddef2&nldsid=134&dc=.dal&hst=62edee2360a9e0d85fc23d1b3a223134.-844328504&csid=d.univisionnow_subscriber&caid=univision_simulcast&nw=112214&prof=univision_live_html5</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaize</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_h</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Claim</th><th>Claim Limitation</th><th>Example Infringement Evidence</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td>d.net</td><td>d_800_20180725210840.ts</td></tr>
</table>

Accordingly, the Univision App determines the performance ratio was less than a threshold value prior to requesting lower quality segments.

As the example testing continued, the Univision App requested the 1600 Bandwidth version of the 20180725210850.ts and 20180725210900.ts segments

<table>
<tr><th>Method</th><th>Host</th><th>Path</th></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_800_20180725210840.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725210850.ts</td></tr>
<tr><td>GET</td><td>neunlds134dal.akamaized.net</td><td>/nlds/univisionnow2/univision_east/as/live/univision_east_hd_1600_20180725210900.ts</td></tr>
</table>

Accordingly, the Univision App determines the performance ratio exceeds a second threshold value prior to requesting higher quality segments.
</td></tr>
<tr><td></td><td>presenting the video by playing back the requested media files with the media player on the end user station in order of ascending playback time.</td><td>The Univision App presents the video by playing back the requested media files with the media player on the end user station in order of ascending playback time. The Univision App plays back the requested .ts files in order of ascending playback time after they are retrieved.

The Univision App receives the playlist file that lists the .ts files in order of ascending playback time and the Univision App requests those same .ts files in order of ascending playback time. Accordingly, the requested media files are played back by the Univision App in order of ascending playback time.</td></tr>
</table>