

(51) International Patent Classification:
G06F 19/00 (2011.01)(21) International Application Number:
PCT/US2011/040833(22) International Filing Date:
17 June 2011 (17.06.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/355,831 17 June 2010 (17.06.2010) US
61/368,029 27 July 2010 (27.07.2010) US(71) Applicant (for all designated States except US): **YALE UNIVERSITY** [US/US]; Two Whitney Avenue, New Haven, CT 06511 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **ALIMI, Richard, A.** [US/US]; 100 North Whisman Road, Apt.2014, Mountain View, CA 94043 (US). **TIAN, Chen** [CN/US]; 51 Prospect Street, New Haven, CT 06511 (US). **YANG, Yang, Richard** [CN/US]; 51 Prospect Street, New Haven, CT 06511 (US).(74) Agent: **PRITAKER, Randy, J.**; Wolf, Greenfield & Sacks, P.c., 600 Atlantic Avenue, Boston, MA 02210-2206 (US).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

(54) Title: TESTING LIVE STREAMING SYSTEMS

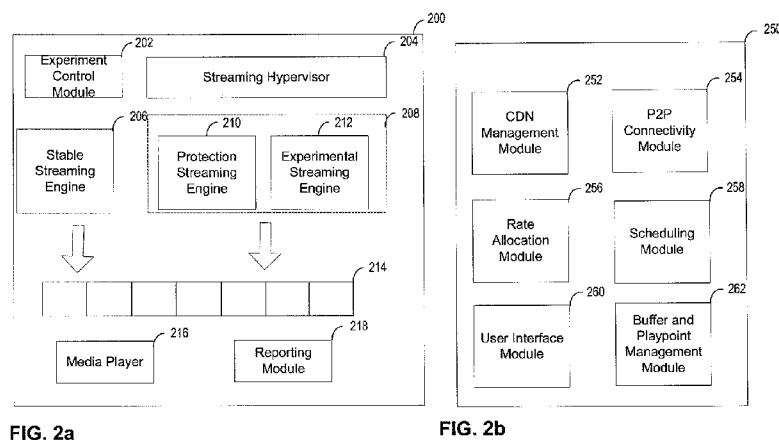


FIG. 2a

FIG. 2b

(57) **Abstract:** A live streaming system for streaming content to a plurality of client devices. The system comprises a controller configured to control testing of the live streaming system by determining a start time for an experiment for testing the live streaming system, sending a message to a subset of the plurality of client devices indicating that each client device in the subset can participate in the experiment, the message comprising experiment control parameters, and monitoring the experiment.

- 1 -

TESTING LIVE STREAMING SYSTEMS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims the benefit under 35 U.S.C. § 119(e) of U.S. Provisional Application Serial No. 61/355831, filed on June 17, 2010, titled "Robust, Internet Streaming Evaluation," and U.S. Provisional Application Serial No. 61/368029, filed on July 27, 2010, titled "Robust, Internet Streaming Evaluation," which are incorporated herein by reference in their entireties.

FEDERALLY SPONSORED RESEARCH

[0002] This invention was made with government support under 0831834 awarded by the National Science Foundation. The government has certain rights in this invention.

BACKGROUND

[0003] Streaming media is multimedia being constantly received by and presented to one or more end users over a network. A media publisher makes content streams available to end users and uses the network to distribute and deliver the content streams. Examples of streaming media include streaming audio and/or video content.

[0004] Live streaming is the process of streaming multimedia over the Internet. As such, live streaming is major Internet application and is widely used. For instance, live streaming has been used to broadcast live television channels, radio stations, music streams, daily events and major recent events ranging from sporting events (e.g., Winter Olympics and World Cup) to news (e.g., Obama inauguration address).

[0005] Live streaming systems comprise various components including client devices and software for receiving and processing streaming media, servers for publishing, distributing and broadcasting streaming media and heterogeneous networks connecting such client devices and servers. Today, live streaming systems are large and complex. Live streaming systems can include tens of thousands of client devices.

[0006] Live streaming systems need to be evaluated to understand and improve their performance. Conventional approaches to evaluating live streaming systems include theoretical modeling and laboratory/testbed testing.

- 2 -

SUMMARY

[0007] In some embodiments, a live streaming system for streaming content to a plurality of client devices is provided. The system includes a controller configured to control testing of the live streaming system. The controller includes a processor configured to execute a method comprising determining a start time for an experiment for testing the live streaming system, sending a message to a subset of the plurality of client devices indicating that each client device in the subset can participate in the experiment, the message comprising experiment control parameters, and monitoring the experiment, wherein each client device in the subset hosts a first streaming engine and a second streaming engine.

[0008] In some embodiments, a method is provided for testing a first streaming engine hosted on a client device hosting a second streaming engine. The method comprises using the first streaming engine to obtain a unit of streaming content, and determining if the unit of streaming content is obtained by the first streaming engine. The method further includes using the second streaming engine to obtain the unit of streaming content, if it is determined that the unit of streaming content is not obtained by the first streaming engine.

[0009] In some embodiments, a computer readable storage medium is provided. The computer readable storage medium may store a plurality of processor-executable components that when executed by a processor, comprise an experimental streaming engine configured to obtain streaming content, a protection streaming engine configured to obtain streaming content, and a streaming hypervisor configured to adaptively allocate tasks between the first streaming engine and the second streaming engine.

[0010] In some embodiments, a method for controlling testing of a live streaming system for streaming content to a plurality of client devices is provided. The method comprises determining a start time for an experiment for testing the live streaming system, sending a message to a subset of the plurality of client devices indicating that each client device in the subset can participate in the experiment, the message comprising experiment control parameters, and monitoring the experiment, wherein at least one client device in the subset hosts a first streaming engine and a second streaming engine.

- 3 -

[0011] In some embodiments a device hosting a first streaming engine and a second streaming engine is disclosed. The device comprises a processor configured to execute a method. The method comprises using the first streaming engine to obtain a unit of streaming content, determining if the unit of streaming content is obtained by the first streaming engine, and if it is determined that the unit of streaming content is not obtained by the first streaming engine, using the second streaming engine to obtain the unit of streaming content.

[0012] The foregoing is a non-limiting summary of the invention, which is defined by the attached claims.

BRIEF DESCRIPTION OF DRAWINGS

[0013] The accompanying drawings are not intended to be drawn to scale. In the drawings, each identical or nearly identical component that is illustrated in various figures is represented by a like numeral. For purposes of clarity, not every component may be labeled in every drawing. In the drawings:

[0014] FIG. 1 is a block diagram of an exemplary operating environment for a system, in accordance with some embodiments of the present disclosure.

[0015] FIGS. 2a and 2b are block diagrams illustrating a client and a streaming engine, in accordance with some embodiments of the present disclosure.

[0016] FIG. 3 is a block diagram illustrating a controller, in accordance with some embodiments of the present disclosure.

[0017] FIG. 4 is a flowchart of an illustrative process for controlling execution of an experiment, in accordance with some embodiments of the present disclosure.

[0018] FIG. 5 is a flowchart of an illustrative process for conducting an experiment, in accordance with some embodiments of the present disclosure.

[0019] FIGS. 6a and 6b are diagrams illustrating adaptive allocation of tasks between an experimental streaming engine and a protection streaming engine, in accordance with some embodiments of the present disclosure.

[0020] FIG. 7 is a diagram illustrating a compositional runtime architecture, in accordance with some embodiments of the present disclosure.

[0021] FIG. 8 is a block diagram generally illustrating an example of a computer system that may be used in implementing aspects of the present disclosure.

DETAILED DESCRIPTION

[0022] The inventors have recognized and appreciated that greater utility may be derived from live streaming systems that may be evaluated while they are in use. Deployed live streaming systems that provide performance evaluation as an intrinsic capability allow for both large-scale and realistic evaluation of their performance. Herein, “live testing” refers to the process of testing a deployed live streaming system while it is being used (e.g., by users to receive streaming content and/or by content providers to distribute, broadcast, or deliver streaming content).

[0023] Live streaming systems that support live testing allow for evaluating system performance in any of numerous scenarios. For instance, live testing may be used to evaluate performance of a live streaming system after one or more components in the live streaming system are altered. For instance, software on a client device for receiving and/or processing streaming media may be updated. As another example, system performance in the presence of a spike in demand for streaming content (e.g., after a major news event) may be evaluated. Many other scenarios in which live testing is desirable will be apparent to those skilled in the art.

[0024] The inventors have both appreciated the shortcomings of conventional approaches to evaluating live streaming systems, and recognized that, in various embodiments, some or all of these shortcomings may be overcome by building live streaming systems that support evaluation of their performance as a built-in capability.

[0025] The inventors have recognized that conventional approaches to evaluating live streaming systems are limited in the scale at which they may be applied. For instance, conventional simulations and testbeds cannot easily scale to evaluating the performance of live streaming systems comprising tens of thousands of client devices, even though, in practice, many live streaming events have reached and surpassed this scale.

[0026] The inventors have also recognized that because the Internet is complex, it is difficult to capture, in a modeling simulation or a testbed, all aspects of the Internet that may relate to performance of a live streaming system. For example, the Internet includes diverse Internet Service Provider network management practices, heterogeneous

- 5 -

networks supporting communications at different bandwidths (e.g., ADSL networks, Local Area Networks, wireless networks, cable networks, and etc), and network features such as shared bottlenecks at ISP peering or enterprise ingress/egress links. Furthermore, client hosts may differ in the type of their network connectivity, amounts of their system resources allocated to tasks associated with streaming, and network protocol stack implementations; routers may differ in their queue management policies, scheduling algorithms and buffer sizes; and background Internet traffic may change dynamically.

[0027] The inventors have also recognized that without the capability to conduct realistic large-scale evaluations, many live streaming systems are often poorly evaluated and operate sub-optimally and/or in a way that deviates from expectations.

[0028] The inventors have recognized and appreciated that providing live streaming systems that support live testing may overcome some of the above-mentioned drawbacks of conventional techniques for evaluating live streaming systems. However, not every embodiment addresses every one of these drawbacks, and some embodiments may not address any of them. As such, it should be appreciated that the invention is not limited to addressing all or any of the above-discussed drawbacks of these conventional techniques for evaluating live streaming systems.

[0029] Advantageously, a live streaming system that supports live testing may support live testing in a way that is transparent to an end user of the live streaming system. Accordingly, live testing may take place without affecting the quality of the end user's experience.

[0030] Live testing may comprise evaluating the performance of the live streaming system by evaluating performance of any component of the live streaming system. For example, live testing may comprise evaluating streaming performance at each client device participating in an evaluation experiment. In this case, live testing may comprise evaluating the performance of an experimental streaming engine. Streaming engines are described in greater detail below with reference to FIG. 3.

[0031] A live streaming system may support live testing by providing for a capability to orchestrate experiments using client devices already active in the live streaming system. For instance, client devices of users viewing streaming content may be used as part of an evaluation experiment (in contrast to conventional approaches in which

- 6 -

dedicated devices are used for testing). In some embodiments, a controller, comprising one or more computing devices, may orchestrate and control experiments as well as perform any other related functions. The controller may, for example, provide instructions to client devices using a live streaming system based at least in part on which the client devices may participate in an evaluation experiment.

[0032] FIG. 1 shows an exemplary live streaming system 100 that supports live testing. Live streaming system 100 comprises client devices 120, 122, 124, and 126 that may communicate with one another and other servers via network 110. A client device may be used by one or more users to view streaming content. In the illustrative example, user 121 may use client device 120 to view streaming content. Each client device may be any suitable physical computing device. In some embodiments, a client device may be a desktop computer, a laptop computer, or a mobile computing device. However, it should be appreciated that the form factor of the client device is not critical to the invention and the client device may be or include any other suitable device.

[0033] It should be appreciated that even though four client devices are shown in the illustrative example of FIG. 1, a live streaming system may comprise any suitable number of client devices, as indicated by ellipsis 125. For instance, a live streaming system may comprise thousands, tens of thousands, hundreds of thousands, or millions of client devices. In particular, a live streaming system may comprise at least 100, at least 1000, at least 10,000, at least 100,000, or at least 1,000,000 client devices. It should also be appreciated, that any suitable number of client devices may be used to test the live streaming system. For instance at least 1000, at least 10,000, at least 100,000, or at least 1,000,000 client devices may be used as part of an experiment to test the live streaming system.

[0034] Each client device may comprise software modules that, when executed by the client device, cause the client device to perform functions related to receiving and processing streaming content as well as testing of the live streaming system. The performance of the live streaming system 100 may depend on such software modules. Accordingly, live testing may be used to test the impact of altering one or more of such software modules on the performance of system 100. Such software modules are described in greater detail with reference to FIG. 2a and FIG. 2b.

- 7 -

[0035] Network 110 may be any suitable network and may comprise, for example, the Internet, a LAN, a WAN, and/or any other wired or wireless network, or combination thereof. As shown, client devices 120, 122, and 124, and 126 may be connected to network 110 via connections 132, 134, 136, and 138, respectively. These connections may be of any suitable type and of any suitable bandwidth. For instance, the connections may be wired or wireless connections. Any of the connections may be one of a LAN connection, an ADSL connection, a cable connection, a dial up connection, or any other type of connection as known in the art. Each of the above-mentioned types of connections may have a bandwidth associated with it. In some embodiments, connections 132, 134, 136, and 138 may have the same bandwidth, while in other embodiments the connections may have different bandwidths.

[0036] Client devices 120, 122, 124, and 126 may receive streaming content delivered over network 110. Streaming content may comprise any suitable streaming content and may, for example, comprise streaming audio content (e.g., music, radio broadcast, or any other audio content), and/or streaming video content (e.g., Internet television, videos, movies, or any other video content). Though, streaming content may comprise any other streaming multimedia content as known in the art.

[0037] Streaming content may be distributed, and/or delivered using any suitable approach or combination of approaches. In some cases, streaming content may be distributed and/or delivered by one or more content delivery networks (CDN). As known in the art, a content delivery network may be a system of computers containing copies of data placed at various nodes of a network. In the illustrative example of FIG. 1, streaming content may be distributed and/or delivered by using CDN server 104, which may be coupled to one or more content delivery networks.

[0038] In some cases, streaming content may be distributed and/or delivered by a peer-to-peer (P2P) network. In this case, streaming content may be distributed and/or delivered to client devices from other client devices (e.g., client 120 may download streaming content from client 124). Additionally, a P2P network may comprise a P2P connection manager such as P2P connection manager 106 shown in FIG. 1. A P2P connection manager may provide a device in a P2P network with connectivity information associated with another device. For instance, the P2P connection manager may provide a device (e.g., client device 120) with connectivity information associated

- 8 -

with another peer in the P2P network (e.g., client device 122). As another example, the P2P connection manager may provide a device (e.g., client device 120) with connectivity information associated with any other device on the network (e.g., controller 102). In some embodiments, the P2P connection manager may provide a client device (e.g., client 120) with other information downloaded from another device (e.g., controller 102). For instance, the P2P connection manager may provide a client device with information associated with a live testing experiment.

[0039] It should be recognized that a client device may receive streaming content distributed and/or delivered in any suitable way. For instance, a client device may obtain content via a P2P network, via a CDN, or via a combination of the two. Though, a client device may also obtain content using any other way as known in the art.

[0040] A live streaming system may comprise a controller to control live testing of the live streaming system. As described in greater detail below with reference to FIG. 4, a controller may control live testing by defining an experiment, determining a start time for the experiment, coordinating client device participation in the experiment, and monitoring the experiment. A controller may be implemented in any suitable way. For instance, in illustrative system 100, controller 102 may be a server. Though, it should be recognized that controller 102 may comprise one or more processors and/or one or more physical computing devices. Controller 102 may be connected to network 110 in any suitable way, as embodiments of the invention are not limited in this respect.

[0041] As previously mentioned, a client device may comprise one or more software modules for performing functions related to receiving and processing streaming content and testing a live streaming system. FIG. 2a illustrates software components that may execute within client device 200. In the embodiment illustrated, the software components may be stored as processor-executable instructions and configuration parameters and, for instance, may be stored in any storage device such as a memory or a disk (not shown) associated with client device 200. Client device 200 may be any suitable client device and may, for example, be one of the client devices 120, 122, 124, and 126 discussed with reference to FIG. 1.

[0042] Client device 200 may comprise one or more media players that that may be used to play streaming content to users. In the illustrated example, client device 200 comprises media player 216. Media player 216 be any suitable type of media player and

- 9 -

may play any suitable type of streaming content to users and may, for example, be used to play audio and/or video content to users.

[0043] Streaming content may comprise one or more units of streaming content. Units of streaming content may be of any suitable type. For instance, a unit of content may comprise a predetermined amount of streaming content. The amount of content may be any suitable amount and may, for example, correspond to a predetermined amount of data (e.g., one kilobyte of content) and/or to a predetermined duration (e.g., one second of content).

[0044] Units of streaming content may be ordered. For instance, they may be ordered temporally based on the order in which they should be played. Though, units of streaming content may be organized in any of other numerous ways such as being ordered in correspondence to the time of their being downloaded to client device 200.

[0045] A media player (e.g., media player 216) may play streaming content to a user by playing units of streaming content in sequence. The sequence may correspond to how units of streaming content may be ordered. Units of streaming content may be stored by client device 200 prior to being played by media player 216. In some embodiments, units of streaming content may be stored in a buffer, such as buffer 214 shown in FIG. 2a, prior to being played by media player 216. Though, it should be recognized that units of streaming content may be stored in any suitable way prior to being played by a media player.

[0046] Units of streaming content may be downloaded to client device 200 using one or more live streaming engines. A live streaming engine, herein also referred to as a streaming engine, comprises one or more software modules for performing various tasks associated with processing streaming content on client device 200. For example, a streaming engine may comprise one or more software modules for downloading/uploading units of streaming content from/to one or more devices over a network. Streaming engines are described in greater detail below with reference to FIG. 2b.

[0047] Client device 200 comprises stable streaming engine 206 and an experimental module 208 comprising protection streaming engine 210 and experimental streaming engine 212. Client device 200 may use any of these streaming engines to perform any of

- 10 -

numerous functions associated with processing streaming content. In some cases, when client device 200 is participating in live testing of a live streaming system (e.g., client device is part of an experiment for testing the live streaming system), client device 200 may use at least one of the streaming engines in experimental module 208 to perform any such functions. On the other hand, when client device 200 is not participating in live testing of a live streaming system, client 200 may use stable streaming engine 206 to perform any such functions. Though, in other embodiments, any streaming engine on client device 200 may be used regardless of whether client device 200 is participating in a live testing experiment.

[0048] Using one or more streaming engines in experimental module 208 when client device 200 is participating in one or more live testing experiment may provide for the capability to evaluate one or more experimental streaming engines during the experiment. In the illustrated example, client device 200 may evaluate experimental streaming engine 212 by participating in one or more live testing experiments.

[0049] Any of numerous aspects of an experimental streaming engine may be evaluated. For instance, experimental streaming engine 212 may differ from stable streaming engine 216 and evaluating the experiment engine may be used to evaluate the effect of the difference between the engines on streaming performance.

[0050] As one example of a difference, experimental streaming engine 212 may include a software module that stable streaming engine 216 does not include. It may be important to evaluate the performance of experimental streaming engine 212 to evaluate the effect of the software module on performance. The software module may, for example, be a new software module or a new version of a previously existing software module (e.g., bug fix, upgrade, or new component). The software module may be any of the software modules of a streaming engine described below with reference to FIG. 2b or may be any other software module of a streaming engine.

[0051] As another example of a difference, values of one or more parameters and/or settings of experimental streaming engine 212 may differ from those of stable streaming engine 216, and it may be important to evaluate the performance of experimental streaming engine 212 to evaluate the effect of these different values on the performance. The parameters and/or settings may be any suitable settings of a streaming engine as

- 11 -

known in the art and may, for example, may be parameters and/or settings associated with how a client device allocates bandwidth to download/upload content from/to other devices on a network.

[0052] The performance of an experimental streaming engine may not be known in advance. However, because live testing may involve testing the experimental streaming engine while a user is using the client device to receive and/or process streaming content (e.g., a user may be viewing streaming video content), it may be advantageous to shield the user of the client device from any degradation in performance that may result from using an experimental streaming engine instead of a stable streaming engine.

[0053] To ensure that live testing is transparent to a user of client device 200, experimental module 208 comprises a protection streaming engine 210. Protection streaming engine 210 is associated with experimental streaming engine 212 and may be used to shield a user from potentially poor performance of experimental streaming engine 212. Protection streaming engine 210 may, for example, perform any functions related to downloading/uploading and processing units of streaming content that experimental streaming engine 212 may not be able to perform. As a specific example, protection streaming engine 210 may download one or more units of streaming content if experimental streaming engine 212 is not able to download these units of streaming content. This may be beneficial because, for example, media player 216 may obtain the units of streaming content and play them to the user—even though it was the protection streaming engine and not the experimental streaming engine that downloaded these units of streaming content.

[0054] A protection streaming engine may be any suitable type of streaming engine as known in the art. In some cases, it may be a CDN streaming engine configured to obtain streaming content using one or more content delivery networks. In other cases, it may be a P2P streaming engine configured to obtain streaming content using peer-to-peer networks. Preferably, the protection streaming engine is a hybrid streaming engine configured to obtain streaming content using at least two different types of streaming engines. For instance, the protection streaming engine may be configured to use one or more content delivery networks and/or one or more P2P networks. For example, protection streaming engine 210 may comprise a CDN sub-engine and a P2P sub-engine

- 12 -

and may download one or more “missing” units of streaming content that experimental streaming engine 212 did not download. In this case, the CDN sub-engine may download missing units of streaming content up to a predetermined download capacity. The P2P sub-engine may download any missing units of streaming content that the CDN sub-engine was not able to download for any of a variety of reasons, such as constraints on CDN capacity due to enterprise bottlenecks.

[0055] Client device 200 further comprises a streaming hypervisor 204. Streaming hypervisor 204 may manage streaming engines in experimental module 208. Streaming hypervisor 204 may manage the allocation of tasks and computer resources (e.g., memory, CPU time, network resources, etc.) to protection streaming engine 210 and experimental streaming engine 212. For example, streaming hypervisor 204 may indicate to each streaming engine what units of streaming content that streaming engine is responsible for downloading, uploading, and/or processing. As a specific example, streaming hypervisor 204 may provide each streaming engine with a list of units of streaming content to download. As another specific example, streaming hypervisor 204 may allocate an amount of bandwidth to each streaming engine for downloading and/or uploading units of streaming content.

[0056] Streaming hypervisor 204 may allocate tasks to the protection and experimental streaming engines in any of numerous ways. In some embodiments, the streaming hypervisor may allocate tasks following a fixed proportional allocation (e.g., protection streaming engine is responsible for downloading, uploading, and/or processing a fixed percentage of units of streaming content). In other embodiments, as described in greater detail with reference to FIG. 5, streaming hypervisor 204 may allocate tasks adaptively.

[0057] Client device 200 may participate in one or more live testing experiments and comprises an experiment control module 202 that may be used control actions of client device 200 with respect to any such experiments. For example, experiment control module 202 may be used to decide when client device 200 may join or leave an experiment. The experiment control module may make such decisions based on experiment control parameters that may be obtained in any of numerous ways and, for

- 13 -

example, may be obtained over a network from a controller (e.g., controller 102 described with reference to FIG. 1).

[0058] Experiment control module 202 may indicate which streaming engine(s) may be used during a live testing experiment. For instance, experiment control module 202 may indicate that protection streaming engine 210 and experimental streaming engine 212 may be used during a live testing experiment. Experiment control module 202 may also indicate that stable streaming engine 206 may be used when an experiment is not in progress. Though, in other embodiments, experiment control module 202 may indicate that stable streaming engine 206 may be used during a live testing experiment.

[0059] Experiment control module 202 may monitor the performance of client device 200 during one or more live testing experiments and, for example, may monitor the performance of protection streaming engine 210 and/or experimental streaming engine 212 during the experiment. To this end, experiment module 202 may collect and process information related to performance of client device 200 and may report at least a portion of such information to another device (e.g., a controller) over a network by using reporting module 218.

[0060] Any suitable information related to performance of the client device may be obtained by experiment control module 202. Such information may be gathered during a live testing experiment or when no live testing experiment is ongoing. For example, such information may comprise data including the number of units of streaming content downloaded, uploaded, and/or processed by experimental streaming engine 212; the number of units of streaming content that experimental streaming engine 212 failed to download, upload, and/or process; the number of units of streaming content that protection streaming engine 210 downloaded, uploaded, and/or processed; data related to network behavior, data related to resource utilization on the client device (e.g., CPU, memory, network bandwidth, etc.). Further examples include, the number of units of streaming content that missed its playback deadline in the experimental streaming engine 212, the number of units of streaming content that missed its playback deadline in the protection streaming engine 210, the join time and/or departure time of client device 200 to the streaming system, the join time and/or departure time of each client device to a live streaming experiment, the number of other client devices directly connected to

- 14 -

client device 200. Though, it should be recognized that the above list is not limiting, and experiment control module 202 may collect, process, and/or send any other suitable information related to performance of the client device as known in the art.

[0061] It should be recognized that FIG. 2a is merely an illustrative example of a software architecture on a client device and that any of numerous variations are possible. For instance, performance of stable streaming engine 206 may be evaluated during a live testing experiment—even if no changes are made to stable streaming engine. This may be useful for evaluating performance of a live streaming system as a function of factors other than a streaming engine (e.g., network traffic characteristics) and for monitoring the performance of a live streaming system rather than evaluating its performance. As another example, client device 200 is illustrated as having one experimental streaming engine in FIG. 2a. However, client device 200 may comprise one or more experimental streaming engines, each associated with one or more protection streaming engines. Also, in the illustrated example, protection streaming engine 210 is a protection streaming engine for experimental streaming engine 212. However, in other embodiments, stable streaming engine 206 may be a protection streaming engine for experimental streaming engine 212. For instance, stable streaming engine 206 may be a P2P streaming engine and may serve as at least a part of a hybrid protection engine. Many other variations will be apparent to those skilled in the art.

[0062] An illustrative software architecture of a streaming engine 250 is shown in FIG. 2b. Streaming engine 250 may be any suitable streaming engine and may be one of the streaming engines described with reference to FIG. 2a. For example, streaming engine 250 may be stable streaming engine 206, protection streaming engine 210, and/or experimental streaming engine 212. In the embodiment illustrated, the software components may be stored as processor-executable instructions and configuration parameters and, for instance, may be stored in storage device such as a memory or a disk (not shown) associated with a client device hosting streaming engine 250 (e.g., client device 200 shown in FIG. 2a).

[0063] A streaming engine may perform any of numerous functions associated with processing streaming content. For instance, a streaming engine may download units of streaming content from one or more devices, upload units of streaming content to one or

- 15 -

more devices, store units of streaming content, buffer units of streaming content, provide units of streaming content to a media player, and process received units of streaming content in any suitable way. Though, it should be recognized that the above list is not limiting and that a streaming engine may perform any other suitable functions as known in the art.

[0064] Streaming engine 250 comprises software modules to download and/or upload units of streaming content. In the illustrated embodiment, streaming engine 250 comprises CDN management module 252, which may be configured to download and/or upload units of streaming content from/to a content delivery network. CDN management module may be active when the client device hosting the streaming engine has been allocated CDN resources.

[0065] Streaming engine 250 also comprises P2P connectivity module 254, which may be configured to download and/or upload units of streaming content from/to peers using peer-to-peer connections. P2P connectivity module 254 may be configured to determine P2P connectivity among client devices and perform any other tasks related to P2P networking. In some embodiments, P2P connectivity module 254 may connect to a P2P connectivity manager (e.g., P2P connectivity manager 106 described with reference to FIG. 1) and, for example, may download connectivity information and/or any other suitable information from the P2P connectivity manager. For example, P2P connectivity module may download experiment control parameters and/or any information related to a live testing experiment from the P2P connectivity manager.

[0066] Streaming engine 250 also comprises rate allocation module 256, which may be configured to determine how to allocate bandwidth to download/upload tasks. For instance, rate allocation module 256 may be configured to determine how a client device may allocate its bandwidth to upload data (e.g., units of streaming content) to other devices (e.g., neighbors in a P2P network).

[0067] Streaming engine 250 also comprises scheduling module 258. Scheduling module may be configured to determine which units of streaming content to download. It may also be used to determine the order in which to download the units of streaming content. In some embodiments, scheduling module 258 may work in concert with a streaming hypervisor to determine which units of streaming content to download.

- 16 -

Though, in other embodiments, scheduling module 258 may make this determination on its own.

[0068] Streaming engine 250 also comprises user interface module 260 that may be configured to present information to one or more users. Such information may include streaming content and/or advertising information. In some embodiments, a user interface module may comprise a media player. Though, in other embodiments (e.g., as shown in FIG. 2a), the media player may be external to a streaming engine.

[0069] Streaming engine 250 also comprises buffer and play-point management module 262, which is configured to manage a buffer of units of streaming content. This module may also be configured to maintain a play-point—an identifier (e.g., an index) of the unit of streaming content that is being played to a user or used by another module such as a media player or a streaming hypervisor. Module 262 may also comprise a download window indicating a list of units of streaming content that streaming engine 250 may download. Module 262 may also comprise an upload window indicating a list of units of streaming content streaming engine 250 has and can upload to other devices.

[0070] It should be recognized that the above-described software modules of streaming engine 250 are illustrative and a streaming engine may comprise any suitable additional and/or alternative software modules.

[0071] Live testing experiments in which one or more client devices may participate, may be controlled by a controller such as controller 102 previously described with reference to FIG. 1. FIG. 3 illustrates software components that may execute within an illustrative controller 300. In the embodiment illustrated, the software components may be stored as processor-executable instructions and configuration parameters and, for instance, may be stored in storage device such as a memory or a disk (not shown) associated with controller 300.

[0072] Controller 300 comprises experiment definition module 302 that may be used to specify a live testing experiment. The live testing experiment may be specified based on any suitable information such as user input to controller 300 or experiment parameters provided to controller 300.

- 17 -

[0073] The specification of a live-testing experiment may comprise any suitable information. For example, the specification may comprise a duration for the experiment and a client device behavior pattern for one or more client devices participating in the experiment. Though, it should be recognized that an experiment specification may comprise any other suitable information that may be needed to specify the experiment.

[0074] A client device behavior pattern may comprise information indicating how a client device may participate in a live streaming experiment. For instance, the client device behavior pattern may indicate a time at which a client device may begin to participate in the experiment. Such information may be provided either explicitly (e.g., the exact start time is provided) or implicitly. In the latter case, a behavior pattern may comprise a representation of an arrival rate function that may be used by a client device (or controller) to compute a time at which the client device may start participating in the experiment. The arrival rate function may encode a probability of starting to participate in an experiment at a particular time during the experiment. As such, a client device may begin to participate in the testing experiment at a particular time t with probability computed based at least in part on the arrival rate function.

[0075] Additionally, a behavior pattern may comprise information indicating when a client device may stop participating in the live streaming experiment. Such information may be provided either explicitly (e.g., the exact stop time is provided) or implicitly. In the latter case, a behavior pattern may comprise a representation of a life duration function that may be used by a client device (or controller) to compute a time at which the client device may stop participating in the experiment. The life duration function may encode a probability of stopping participating in an experiment at a particular time during the experiment. As such, a client device may stop participating in the experiment at a particular time t with probability calculated based at least in part on the life duration function.

[0076] The time at which a client device stops participating in an experiment may depend on any of numerous factors. For example, the time that at which the client device stops participating in an experiment may depend on the time that the client device starts participating in the experiment. As another example, the time that at which the client device stops participating in an experiment may depend on the performance of the client

- 18 -

device (e.g., poor performance may lead to a user turning off the streaming application on the client device). Accordingly, if a value indicative of client device performance is below a predetermined threshold (e.g., miss rate of units of streaming data is below 3%), the client device may stop participating in the experiment.

[0077] A behavior pattern may be assigned to more than one client device and may be assigned to a group of client devices. For instance, a behavior pattern may be assigned to a group of client devices having one or more properties in common with one another. Such properties include type of network connection (e.g., cable or DSL), estimated download/upload capacity, network location, type of software installed on the client (e.g., all have a particular version of an operating system, media player, streaming engine etc.). Though, it should be recognized, that any of other numerous ways of grouping client devices may be used.

[0078] Because a single behavior pattern may be assigned to a group of client devices, the behavior pattern may be used to effect a particular behavior for the group. For example, if the behavior pattern comprises a representation of an arrival rate function that has high values initially and then sharply decreases, many client devices in the group may join an experiment early in the experiment and the group may exhibit a behavior characteristic of a flash crowd.

[0079] Controller 300 also comprises a network monitor module 304 configured to monitor network conditions. Network monitor module may monitor the status of any client device in the live streaming network and may monitor network conditions to help controller 300 make a determination as to what an appropriate time for starting a live testing experiment may be. For instance, if a live testing experiment requires a certain number of client devices (perhaps of a particular type) be used in the experiment, network monitor module may monitor the network for the presence of that number of client devices and may be used to predict a time at which that number of client devices may be active in the live streaming network.

[0080] Network monitor module 304 may monitor network conditions in any suitable way. It may monitor network conditions either actively (by probing the network) or passively by receiving messages from client devices. For example, network monitor module 304 may receive user datagram protocol (UDP) messages from client devices.

- 19 -

[0081] Controller 300 comprises experiment control module 306. Experiment control module 306 may be used to perform any calculations and communications related to a live testing experiment. For example, experiment control module 306 may calculate a start time of an experiment, duration of an experiment, and/or a stop time for an experiment.

[0082] In some embodiments, experiment control module 306 may calculate one or more times for one or more client devices to start and/or stop participating in an experiment. In other embodiments, experiment control module 306 may calculate information that may be used by one or more client devices to determine whether to participate in the experiment and when to start/stop participating in the experiment.

[0083] Experiment control module 306 may provide client devices with any suitable information associated with a live testing experiment. In some instances, experiment control module 306 may communicate with one or more client devices directly to provide them with such information. In other instances, experiment control module may communicate with a P2P connectivity manager (e.g., P2P connectivity manager 106) to provide the P2P connectivity manager with such information. In turn, the P2P connectivity manager may provide one or more client devices to provide them with information associated with a live testing experiment.

[0084] Controller 300 also comprises reporting module 308 that may monitor the live streaming system during live testing. Reporting module 308 may obtain and process information associated with the performance of the live streaming system during a live test. Such information may be obtained from any of numerous sources and may be obtained in any suitable way. For instance, performance information may be obtained from one or more client devices participating in a live test and, for example, may be obtained from reporting modules running on any such client devices (e.g., reporting module 218 on client device 200). Reporting module 308 may process any obtained performance data in any suitable way. Additionally or alternatively, reporting module 308 may present performance data to the user (e.g., display such performance data to the user).

[0085] Controller 300 also comprises early departure queues 310. Early departure queues may store information comprising states of one or more client devices that may

- 20 -

have stopped participating in an experiment. Early departure queues 310 may comprise one queue or multiple queues. In the latter case, early departure queues 310 may comprise a queue for each group of client devices.

[0086] In some cases, an early departure queue may store information comprising states of one or more client devices that stopped participating in an experiment prematurely. A state of a client device may comprise the scheduled start/stop time of that client device's participation in the experiment. It may also comprise network information including a list of neighbors of the client device. The neighbors may comprise one or more peer-to-peer devices in a P2P network, which communicate with the client device. It may also comprise a list of units of streaming content available on the client device prior to the client device stopping participating in the experiment. Though, the above examples are not limiting and state information may comprise any other information related to the state of the client device as known in the art.

[0087] A controller, such as controller 300 described with reference to FIG. 3, may control execution of a live streaming experiment. FIG. 4 shows an illustrative process 400 for controlling execution of a live streaming experiment. Process 400 may execute on any controller and may, for example, execute on controller 102 described with reference to FIG. 1. At least some of the acts of process 400 may be performed by one or more controller software modules described with reference to FIG. 3.

[0088] Process 400 begins in act 404, where a live testing experiment is specified. In the illustrative controller 300 of FIG. 3, act 404 may be performed by experiment definition module 302. The experiment may be any suitable experiment to evaluate performance of a live streaming system (e.g., live streaming system 100 described with reference to FIG. 1). Specifying the experiment may comprise specifying parameters associated with the experiment. For instance, as previously described, the specification of an experiment may comprise duration of an experiment and one or more client device behavior patterns such that each behavior pattern is associated with one or more groups of client devices. Accordingly, specifying the experiment in act 404 of process 400 may comprise specifying such information. As a specific example, specifying an experiment may comprise specifying an arrival rate function for one or more groups of client devices and/or specifying a life duration function for one or more groups of client devices.

- 21 -

[0089] Next process 400 proceeds to act 406 and decision block 408 during which a start time for the experiment may be determined. In act 406, a network (e.g., network 110) associated with a live streaming system (e.g., system 100) may be monitored. In the illustrative controller 300 of FIG. 3, act 406 may be performed by monitor module 304. As previously described, the monitor module may monitor the network and may gather information about arrivals/departures of client devices to/from the live streaming system.

[0090] Next process 400 proceeds to decision block 408, where it may be decided that the experiment specified in act 404 may begin. Process 400 may proceed to decision block 408 at time t_0 and decision block 408 may decide whether to start the experiment at time t_0 or at a time that depends on t_0 . If it is determined that the experiment may start, process 400 proceeds to decision block 410, whereas if it is determined that the experiment may not start, process 400 loops back to act 406. In this case, act 406 and decision block 408 may repeat until it is determined, in decision block 408, that the experiment may start.

[0091] The decision to start the experiment may be made in any suitable way and may be made based at least in part on information about the state of the network obtained in act 406 and the client device behavior patterns specified in act 404. This information may be used to predict a number of client devices active in the live streaming system in a future time period such that these client devices are appropriate for the experiment. In turn, the decision to trigger the experiment may be made based at least in part on this number. Accordingly, the experiment may be started at time t if the predicted number of client devices appropriate for the experiment at time t (or at time $t + \text{an offset}$) exceeds a predetermined threshold.

[0092] Suppose, for example, that the experiment requires that a certain number (e.g., 10,000) client devices belonging to a particular group (e.g., client devices connected via a DSL connection) be used in the experiment and that the network monitor determines that currently there are 8000 such client devices active in the live streaming system. Then, if it is predicted that the number of such client devices at a future point in time is at least 10000, the experiment may start at that future time point.

[0093] A specific example of how to determine at a particular time, whether the experiment may start is described below. In this case, if the duration of an experiment is

- 22 -

given by t_{exp} , then for any time t between 0 and t_{exp} , an upper bound on the expected number of client devices active in the experiment from time t to time t_{exp} is denoted by $exp(t)$ defined by the following expression:

$$exp(t) = \Lambda(t) - \int_0^t \lambda(x) Pr\{L_x < t - x\} dx$$

where, $\Lambda(t) = \int_0^t \lambda(x) dx$, $\lambda(x)$ is an arrival rate function associated with a one or more groups of client devices, and L_x is a life-duration function that specifies the probability that a client device participates in an experiment for an amount of time less than t .

[0094] At any time t_0 , the controller may predict the number of client devices active in the experiment at any time point t in the interval $[t_0, t_0 + t_{exp}]$. Denote by $pred(t)$ the predicted value at time t . This prediction may be made in any suitable way. For instance, this prediction may be made based at least in part on the number of active client devices in the live streaming system. The prediction may be made by using any of numerous approaches known in the art. For example, a time series model such as an autoregressive integrated moving average (ARIMA) model may be used to make the prediction.

[0095] Returning to the specific example, a controller may check at time t_0 whether the following condition is satisfied:

$$exp(t) \leq predict(t_0 + \Delta + t') \forall t' \in [0, t_{exp}]$$

where Δ is a triggering delay and may be any number greater than or equal to 0. If this condition is satisfied, the controller may start the experiment at time $t_0 + \Delta$.

[0096] If it is determined, in decision block 408, that the experiment should start at time t_0 or at time t_0 plus an offset, the experiment start time may be set as t_{start} and process 400 proceeds to decision block 410, where it is decided whether the controller (e.g., controller 102) will directly control the times at which client devices start and stop participating in the experiment or, instead, allow client devices to make these decisions on their own—resulting in distributed control of client device participation in the experiment.

- 23 -

[0097] This determination may be made in any suitable way. For instance, if a large number of client devices is needed for the experiment (e.g., more than 1000), a distributed control approach may be selected to avoid the overhead of communication between the controller and the client devices that is inherent in the direct control scenario. On the other hand, if a small number of client devices are needed in the experiment, a direct control approach may be selected. Though, it should be recognized that the determination to use distributed or direct control may be made in another way and may result in using direct control even in the presence of a large number of client devices or using distributed control when there is a small number of client devices.

[0098] If it is determined, in decision block 410, that the controller should directly control the times at which client devices start and stop participating in the experiment, process 400 proceeds to act 414. During act 414, the controller may communicate with client devices to provide them with start and stop times at which they may start and stop participating in the experiment. Additionally, the controller may provide the client devices with commands to start and stop participating in the experiment.

[0099] On the other hand, if it is determined, in decision block 410, that distributed control may be employed, process 400 proceeds to act 412, where experiment control parameters are sent to client devices. Client devices may use the experiment control parameters to calculate start and/or stop times to start/stop participating in the experiment. Additionally or alternatively, experiment control parameters may be sent to a P2P connection manager (e.g., P2P connection manager 106 described with reference to FIG. 1) that may provide client devices with the experiment control parameters.

[00100] In some embodiments, the experiment control parameters sent to a client device may comprise a start time for the experiment, duration for the experiment, and a client device behavior pattern. In some cases, the experiment control parameters may also include a probability of participation in the experiment. The probability of participation may be obtained in any suitable way and, for example, may be calculated as a ratio of expected number of client devices in the experiment to the total number of available client devices. Providing each client device with a probability of participation in the experiment may allow the controller to reduce the variance on the number of client devices participating in the experiment. As described in greater detail with reference to

- 24 -

act 504 of FIG. 5, the client device may use the experiment control parameters to decide whether to join an experiment and calculate a time at which to join the experiment.

[00101] In the illustrative controller 300 shown in FIG. 3, acts 410, 412, and 414 of process 400 may be performed by the experiment control module 306.

[00102] Regardless of how client devices start to participate in an experiment and the manner in which they are controlled, process 400 proceeds to act 416, during which the experiment is monitored. In the illustrative controller 300 shown in FIG. 3, act 416 may be performed by the reporting module 308 of illustrative controller 300. Monitoring the experiment may comprise monitoring the state of the network. For instance, monitoring may comprise keeping track of which client devices joined the experiment and which client devices left the experiment.

[00103] Additionally or alternatively, monitoring may comprise receiving performance reports from one or more client devices. Performance reports may be received from the client devices and/or from a P2P connection manager that may have received the reports from the client devices. Each report may be associated with a client device and may comprise any information related to the performance of the client device. For instance, the report may comprise information related to the performance of an experimental streaming engine running on the client device during the experiment. Such information may include various quantities, described above, such as the number of units of streaming content downloaded, uploaded and/or processed by the experimental streaming engine; the number of units of streaming content that the experimental streaming engine failed to download, upload, and/or process. Additionally, such information may include quantities such as the number of units of streaming content that a protection streaming engine, running on the client device during the experiment, downloaded, uploaded, and/or processed and any data related to resource utilization (e.g., CPU, memory, network bandwidth, etc.) on the client device. Though, the above examples are not limiting and performance reports may comprise any other suitable performance information.

[00104] While monitoring the experiment, process 400 may detect, in decision block 418, if any client devices stopped participating in the experiment prematurely (e.g., before the end of the experiment or before the stop time associated with the client

- 25 -

device). If it is detected, in decision block 418, that a client device stopped participating in the experiment, the controller may obtain a state of the departed client, in act 420. As previously described, the state of the client may comprise the scheduled start and/or stop time of that client device's participation in the experiment, network information including a list of neighbors of the client device, and/or a list of units of streaming content available on the client device prior to the client device stopping to participate in the experiment.

[00105] The controller may store the state of the departed client device in any suitable way. For instance, the controller may store the state of the departed client device in a queue, such as one of the early departure queues 310 described with reference to FIG. 3. The controller may store the state in an early departure queue associated with client devices in the same group as the departed client device.

[00106] Next, in act 422, the controller may select another client device to replace the departed client device in the experiment. This may be done in any suitable way. For example, when another client device starts to participate in the experiment, the controller may bring the new client device into the same state as the departed client device was in just before departing. This may be done by sending the state information associated with the departed client device to the new client device. A departed client device may be replaced by another device in the same group. Additionally, process 400 may comprise biasing another client device to start to participate in the experiment for the purpose of replacing the prematurely departed client device.

[00107] On the other hand, if it is determined, in decision block 418, that no client devices stopped participating in the experiment prematurely, process 400 loops back to act 416, where monitoring continues until the experiment ends. Once the experiment ends, process 400 completes.

[00108] It should be recognized that process 400 is illustrative and that any of numerous variations are possible. For instance, though process 400 was described for controlling one live testing experiment, the process may be altered to control more than one live testing experiment. As another example, though specifying an experiment was part of process 400, in other embodiments experiments may be specified prior to the

- 26 -

beginning of process 400. For instance, one or more experiments may be specified by one or more users and provided to a controller.

[00109] Any of numerous processes may be performed by a client device during live testing of a live streaming system. One such process is illustrated in FIG. 5, which shows an illustrative process 500 for conducting actions associated with a live streaming experiment on a client device. Process 500 may be performed by any suitable client device and, for example, may be performed by any of client devices 120, 122, 124, and 126 described with reference to FIG. 1. Software modules of a client device described with reference to FIG. 2a may perform some of the acts of process 500.

[00110] Process 500 begins in act 502, where experiment control parameters are received. As previously described with reference to FIG. 4, values of experiment control parameters may be used to determine a start time at which the client device may start to participate in the experiment. The experiment control parameters may comprise a start time for the experiment, duration for the experiment, a client device behavior pattern, and/or a probability of participation in the experiment. The experiment control parameters may be received from a controller (e.g., controller 102) and/or a P2P connectivity manager (e.g., P2P connectivity manager 106).

[00111] Next, process 500 proceeds to decision block 503, where it is determined whether the client device may participate in the experiment. This determination may be made in any of numerous ways and, for example, may be made based at least in part on the experiment control parameters received in act 502. In some cases, this determination may be made based on the value of the probability of participation parameter received in act 502. As a specific example, the client device may draw a random number uniformly at random and compare this number with the value of the probability of participation parameter. If the random number is determined to be larger than the value, it may be determined that the client device will not participate in the experiment and process 500 completes. On the other hand, if it is determined that the random number is smaller than or equal to the value of the probability of participation parameter, it may be determined that the client device will participate in the experiment and process 500 proceeds to act 504. Though, it should be recognized that any of other numerous procedures may be used

- 27 -

to determine whether the client device may participate in the experiment including non-probabilistic and other probabilistic methods.

[00112] In act 504, a start time at which the client device may start to participate in the experiment is calculated. The start time may be calculated in any suitable way and, for example, may be calculated based at least in part on the experiment control parameters received in act 502. In some embodiments, the start time may be calculated based on the client device behavior pattern. As a specific example, the client device may draw a random number (which may be thought of as a waiting time) according to a distribution obtained based at least in part on arrival rate function (e.g., a suitably normalized arrival rate function) and may calculate the starting time as a sum of the start time of the experiment and the random number.

[00113] In the illustrative client device 200 shown in FIG. 2, acts 502, 504, and 414 of process 400 may be performed by the experiment control module 306.

[00114] After the start time is calculated in act 504, process 500 proceeds to act 506, where, at any time at or after the start time, the client device may start to participate in the experiment. Upon starting to participate in the experiment, the client device may take any suitable actions. For instance, the client device may start to use at least one different streaming engine and, for example, may start to use an experimental streaming engine. Additionally, the client device may start to use a protection streaming engine associated with the experimental streaming engine. In the illustrative example of FIG. 2a, client device 200 may start using protection streaming engine 210 and experimental streaming engine 212, instead of stable streaming engine 206. As another example, client device may start monitoring its own performance and may start monitoring the performance of a streaming engine the client device may be using.

[00115] Next, process 500 proceeds to act 508, where the experimental streaming engine and the protection streaming engine may be configured. The engines may be configured in any suitable way and, for example, may be configured by setting various parameters associated with the streaming engines. As previously described with reference to FIG. 2a, each streaming engine may comprise a download window (indicating a set of units of streaming content the streaming engine may download), an upload window (indicating a set of units of streaming content the streaming engine may

- 28 -

upload because it may already have them), and a play-point parameter (an index of a unit of streaming content being played to a user or being provided to another application). Accordingly, configuring a streaming engine may comprise setting parameters associated with its download/upload windows (e.g., left and right boundaries of the windows) and/or the play-point parameter to initial values. It should be recognized that values of parameters associated with download windows, upload windows, and the value of the play-point parameter may change over time.

[00116] In some embodiments, the value of the play-point parameter associated with the experimental streaming engine may be different than the value of the play-point parameter associated with the protection streaming engine. The value of the experimental streaming engine's play-point parameter may be greater than the value of the protection streaming engine's play-point parameter. Accordingly, there may be a delay between the play-points of the experimental and the protection streaming engines. Moreover, in some embodiments, the value of the play-point parameter associated with the experimental streaming engine may be greater than the value of the play-point parameter associated with the protection streaming engine at any time during a live testing experiment.

[00117] The time delay between the play-points of the protection and experimental engines may be any suitable time delay. The time delay may be fixed, but may also vary during a live testing experiment. The time delay may comprise an amount of time required by a protection streaming engine to perform an action (e.g., download a predetermined number of units of streaming content). In some embodiments, the time delay may be any suitable number of seconds. For instance, the time delay may be as at least 1 second, at least 5 seconds, at least 10 seconds, at least 15 seconds, at least 20 seconds, at least 25 seconds, or at least 30 seconds. Though time delays less than 1 second may also be used.

[00118] The play-point of the protection streaming engine may be the user-visible play-point. For instance, the media player may play a unit of streaming unit to the user if that unit is indicated by the protection streaming engine's play-point.

[00119] In some embodiments, the download window associated with the experimental streaming engine may indicate a set of units of streaming content disjoint from the set of units of streaming content indicated by the download window associated

- 29 -

with the protection streaming engine. For example, the download window associated with the experimental streaming engine may indicate a set of units of streaming content to be played after the units in the set of units of streaming content indicated by the download window associated with the protection streaming engine are played.

[00120] The delay between the play-points of the experimental and protection streaming engines may allow for the protection streaming engine to compensate for any errors that the experimental streaming engine may experience. The protection streaming engine may compensate for one or more experimental streaming engine errors because the protection streaming engine may have time to perform an action if it is determined that an experimental streaming engine error occurred. For instance, an experimental streaming engine may not be able to download a unit of streaming content. If the play-point of the experimental streaming engine were the user-visible point, then the user would experience degradation in streaming performance (because the user will not be presented with the missed unit of streaming content). However, when the value of the play-point parameter of the experimental streaming engine is greater than the value of the play-point parameter of the protection streaming engine, the protection streaming engine has time to attempt to download the missing unit of streaming content. Because the protection streaming engine play-point is the user-visible play-point, the above described scheme may allow the protection streaming engine to compensate for the error and download the missing unit of streaming content such that the user may not observe degradation in streaming performance. A specific example of this is described below with reference to FIG. 6a and FIG. 6b.

[00121] FIG. 6a shows an illustrative buffer of units of streaming content. The buffer may store units of streaming content downloaded by a protection streaming engine and/or an experimental streaming engine. In the illustrated example, a protection streaming engine downloaded streaming content units 610, 612, and 614 (as indicated by diagonal line shading) and experimental streaming engine downloaded streaming content units 616, 620, and 622 (as indicated by vertical line shading). Streaming content unit 618 needs to be downloaded, but has not yet been downloaded (as indicated by lack of any shading). In this example, the experimental streaming engine is responsible for downloading streaming content unit 618, but is unable to do so due to an error.

- 30 -

[00122] In this example, the user visible play-point 602 is also the play-point of the protection streaming engine, whereas the play-point of the experimental streaming engine 604 occurs later.

[00123] FIG. 6b shows the state of the buffer, first shown in FIG. 6a, after two units (e.g., units 610 and 612) have been played. In this case, the user-visible play-point 606 is shifted two units to the right as is the experimental streaming engine play-point 608. Because the experimental streaming engine was not able to download streaming content unit 618 and the play-point of the experimental streaming engine has moved beyond unit 618, the user would experience an error if play-point 608 were the user-visible play-point. However, because the user-visible play-point is play-point 606, the protection streaming engine may take advantage of the delay between play-points 606 and 608, and download unit 618 (as indicated by diagonal line shading). Thus, a user will not observe any degradation in performance due to the error in the experimental streaming engine.

[00124] It should be recognized, that the above-described protection scheme is not limited to download errors by the experimental streaming engine. The protection streaming engine may compensate for any type of error made by an experimental streaming engine. As a result, the performance of an experimental streaming engine may be evaluated in a manner transparent to a user of the client device.

[00125] Accordingly, process 500 may proceed as described below. Regardless of how parameters of the protection and experimental streaming engines are set in act 508, process 500 proceeds to act 510, where the experimental streaming engine may perform an action related to processing of streaming content. For instance, the experimental streaming engine may obtain one or more units of streaming content. The experimental streaming engine may obtain one or more units of streaming content by downloading the units. As another example, the experimental streaming engine may upload a unit of streaming content to another client device.

[00126] Next, process 500 proceeds to decision block 512, where it is determined whether the experimental streaming engine successfully performed the action in act 510. This may be done in any suitable way and, for example, may be done by checking whether the action is completed. For instance, it may be determined in decision block

- 31 -

512 whether the experimental streaming engine downloaded one or more units of streaming content. For instance, this may be done by checking a buffer.

[00127] If it is determined that the experimental streaming engine successfully performed the action in act 510, process 500 proceeds to act 514. On the other hand, if it is determined that the experimental streaming engine did not successfully perform the action in act 510, process 500 proceeds to act 514 in which the protection streaming engine attempts to compensate for the experimental streaming engine.

[00128] During act 514, the protection streaming engine may compensate for the experimental streaming engine in any of numerous ways. For instance, the protection streaming engine may try to perform the action that the experimental streaming engine was not able to successfully perform. For example, the protection streaming engine may attempt to download one or more units of streaming content that the experimental streaming engine was not able to download. As another example, the protection streaming engine may attempt to upload one or more units of streaming content that the experimental streaming engine was not able to upload. It should be recognized that the above examples of actions are not limiting and that many other examples of actions taken by a streaming engine will be apparent to those of skilled in the art.

[00129] Regardless of which streaming engine was used to obtain a unit of streaming content, process 500 proceeds to decision block 516, either after act 512 or 514. In decision block 516, the client device may determine whether it may stop participating in the experiment. This determination may be made in any suitable way and, for example, may be made based at least in part on the experiment control parameters received in act 502. Because the experiment control parameters may comprise the duration of the experiment and the start time of the experiment, the client device may determine whether the current time exceeds the sum of the start time of the experiment and the duration of the experiment and may stop participating in the experiment if the current time exceeds this sum. As another example, the client device may leave the experiment based on an action taken by a user of the client device (e.g., user may turn off the client device, close the streaming application, view different content, etc.). As yet another example, the client device may leave the experiment if an error occurs in a hardware component of the client device or a software module running on the client device.

- 32 -

[00130] If it is determined, in decision block 516, that the client device may not leave the experiment, process 500 loops back to act 510, where another unit of streaming content may be downloaded by the experimental streaming engine. Acts 510-516 of process 500 keep repeating until it is determined in act 516 that the client device may leave the experiment.

[00131] If it is determined, in decision block 516, that the client device may leave the experiment, process 500 proceeds to act 518, where the client device may stop participating in the experiment and, afterward, may take any suitable actions. For instance, the client device may start to use at least one different streaming engine and, for example, begin to use a stable streaming engine instead of the experimental streaming engine. In the illustrative example of FIG. 2a, client device 200 may start using stable streaming engine 206 instead of protection streaming engine 210 and experimental streaming engine 212. As another example, the client device may stop monitoring its own performance and, in particular, may stop monitoring the performance of a streaming engine it may be using.

[00132] Next, process 500 may proceed to act 520, where any information related to performance of the client device during the experiment may be reported. In the illustrative example of FIG. 2a, the reporting may be performed by reporting module 218. The information related to performance may be any suitable information as previously described and may be reported to any suitable entity. For example, the performance information may be reported to a controller (e.g., controller 102). Alternatively, the performance information may be reported to a server (e.g., P2P connectivity manager 106) that may compile such performance information from one or more client devices in the experiment. After performance information is reported in act 520, process 500 completes.

[00133] It should be recognized that process 500 is illustrative and that any of numerous variations of process 500 are possible. For example, while in process 500 a protection streaming engine may compensate for errors due to an experimental streaming engine, in other embodiments a stable streaming engine (e.g., stable streaming engine 206 described with reference to FIG. 2a) may compensate for the errors. In addition, though not described as part of process 500, a client device may decide to prematurely

- 33 -

leave an experiment. This may occur at any point after the client device joins the experiment. As yet another example, process 500 comprises adaptively allocating tasks between a protection streaming engine and an experimental streaming engine. However, in other embodiments, the allocation between tasks may be a fixed allocation with each type of streaming engine receiving a fixed proportion of tasks to perform.

[00134] A live streaming system in accordance with embodiments of the present disclosure may be implemented in any suitable way. In some embodiments, the live streaming system may be implemented using a software architecture herein termed as “compositional runtime” architecture. Such a software architecture is advantageous for implementing large distributed peer-to-peer live streaming systems (e.g., system 100), and may comprise a set of algorithmic software modules including, for instance, software modules for connection management, upload scheduling, admission control, and enterprise coordination.

[00135] One aspect of the compositional runtime architecture is that software modules may be implemented as independent blocks, each of which may define a set of input and output ports over which messages may be received and/or transmitted. Additionally, a runtime scheduler may be responsible for delivering messages between blocks.

[00136] Each block may be downloaded and/or composed with one or more other blocks during runtime (this is why the architecture is termed compositional runtime architecture). Advantageously, this may enable software modules to be evaluated during live testing without requiring user input. For example, this may allow for a new software module to be downloaded and used as part of a live testing experiment without prompting and/or waiting for a user to restart their client devices to incorporate the software module. For instance, a client device may download an experimental streaming engine module (or any software module described with reference to FIG. 2a) and use it during live testing without requiring user input.

[00137] Different blocks in the compositional runtime architecture may share access to one or more data structures. For instance, multiple streaming engines may access a data structure storing a list of peer devices connected to the client device. As another example multiple streaming engines may access a data structure storing a buffer

- 34 -

containing units of streaming content (e.g., buffer 214 described with reference to FIG. 2a).

[00138] Blocks in the compositional runtime architecture may perform blocking operations. This may allow blocks to support operations such as calling “select” on a socket to support client-to-client communication or downloading one or more units of streaming content from a CDN.

[00139] FIG. 7 illustrates the compositional runtime architecture. In the illustrated example, a diagram of a portion 700 of software on a client device is shown. Portion 700 is responsible for managing connections to peer devices to the client device. In this case, block 702 is responsible for demultiplexing received messages and sending them to other connected blocks (blocks 704, 706, and 708) that are responsible for handling each type of message.

[00140] Next, assume that a designer wishes to add an admission control algorithm module. Such a module may be added for any of numerous reasons and, for example, may be added to avoid reduced performance for existing peers during flash crowds. The admission control module may be implemented as independent block 710 that may be configured to read handshake messages for newly connected peers. Block 710 may send either a handshake message, if a new connection is accepted, or a disconnect message to the peer, if a new connection is rejected. As shown by dotted lines 712 and 714, block 710 may be composed with the existing blocks (e.g., blocks 702 and 708) at runtime. This example illustrates how the compositional runtime framework may facilitate performing experiments in a variety of scenarios by composing different combinations of software modules.

[00141] Though described with respect to live streaming applications, embodiments of the present disclosure may be applied to a wide variety of problems in other domains. Techniques disclosed herein may be applied to any application that involves large-scale distributed systems. For instance, techniques described herein may be adapted for video-conferencing, network management, and voice over IP systems, to name only a few.

[00142] The above-described embodiments of the present invention can be implemented in any of numerous ways. For example, the embodiments may be implemented using hardware, software or a combination thereof. When implemented in

- 35 -

software, the software code may be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers.

[00143] It should be appreciated that a computer may be embodied in any of numerous forms, such as a rack-mounted computer, a desktop computer, a laptop computer, or a tablet computer. Additionally, a computer may be embodied in a device not generally regarded as a computer, but with suitable processing capabilities, including a Personal Digital Assistant (PDA), a smart phone or any other suitable portable or fixed electronic device.

[00144] Also, a computer may have one or more input and output devices. These devices may be used, among other things, to present a user interface. Examples of output devices that may be used to provide a user interface include printers or display screens for visual presentation of output, and speakers or other sound generating devices for audible presentation of output. Examples of input devices that may be used for a user interface include keyboards, microphones, and pointing devices, such as mice, touch pads, and digitizing tablets.

[00145] Such computers may be interconnected by one or more networks in any suitable form, including a local area network (LAN) or a wide area network (WAN), such as an enterprise network, an intelligent network (IN) or the Internet. Such networks may be based on any suitable technology and may operate according to any suitable protocol and may include wireless networks, wired networks, and/or fiber optic networks.

[00146] An illustrative implementation of a computer system 800 that may be used in connection with any of the embodiments of the invention described herein is shown in FIG. 8. The computer system 800 may include one or more processors 810 and one or more non-transitory computer-readable storage media (e.g., memory 820 and one or more non-volatile storage media 830). The processor 810 may control writing data to and reading data from the memory 820 and the non-volatile storage device 830 in any suitable manner, as the aspects of the invention described herein are not limited in this respect. To perform any of the functionality described herein, the processor 810 may execute one or more instructions stored in one or more computer-readable storage media

- 36 -

(e.g., the memory 820), which may serve as non-transitory computer-readable storage media storing instructions for execution by the processor 810.

[00147] The various methods or processes outlined herein may be coded as software that is executable on one or more processors that employ any one of a variety of operating systems or platforms. Additionally, such software may be written using any of numerous suitable programming languages and/or programming or scripting tools, and also may be compiled as executable machine language code or intermediate code that is executed on a virtual machine or a suitable framework.

[00148] In this respect, various inventive concepts may be embodied as at least one non-transitory computer readable storage medium (e.g., a computer memory, one or more floppy discs, compact discs, optical discs, magnetic tapes, flash memories, circuit configurations in Field Programmable Gate Arrays or other semiconductor devices, etc.) encoded with one or more programs that, when executed on one or more computers or other processors, implement the various embodiments of the present invention. The non-transitory computer-readable medium or media may be transportable, such that the program or programs stored thereon may be loaded onto any computer resource to implement various aspects of the present invention as discussed above.

[00149] The terms “program” or “software” are used herein in a generic sense to refer to any type of computer code or set of computer-executable instructions that can be employed to program a computer or other processor to implement various aspects of embodiments as discussed above. Additionally, it should be appreciated that according to one aspect, one or more computer programs that when executed perform methods of the present invention need not reside on a single computer or processor, but may be distributed in a modular fashion among different computers or processors to implement various aspects of the present invention.

[00150] Computer-executable instructions may be in many forms, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various embodiments.

- 37 -

[00151] Also, data structures may be stored in non-transitory computer-readable storage media in any suitable form. For simplicity of illustration, data structures may be shown to have fields that are related through location in the data structure. Such relationships may likewise be achieved by assigning storage for the fields with locations in a non-transitory computer-readable medium that convey relationship between the fields. However, any suitable mechanism may be used to establish relationships among information in fields of a data structure, including through the use of pointers, tags or other mechanisms that establish relationships among data elements.

[00152] Also, various inventive concepts may be embodied as one or more methods, of which an example has been provided. The acts performed as part of the method may be ordered in any suitable way. Accordingly, embodiments may be constructed in which acts are performed in an order different than illustrated, which may include performing some acts simultaneously, even though shown as sequential acts in illustrative embodiments.

[00153] All definitions, as defined and used herein, should be understood to control over dictionary definitions, definitions in documents incorporated by reference, and/or ordinary meanings of the defined terms.

[00154] The indefinite articles “a” and “an,” as used herein in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean “at least one.”

[00155] As used herein in the specification and in the claims, the phrase “at least one,” in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list of elements to which the phrase “at least one” refers, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, “at least one of A and B” (or, equivalently, “at least one of A or B,” or, equivalently “at least one of A and/or B”) can refer, in one embodiment, to at least one, optionally including more than one, A, with no B present (and optionally including elements other than B); in another embodiment, to at least one, optionally

- 38 -

including more than one, B, with no A present (and optionally including elements other than A); in yet another embodiment, to at least one, optionally including more than one, A, and at least one, optionally including more than one, B (and optionally including other elements); etc.

[00156] The phrase “and/or,” as used herein in the specification and in the claims, should be understood to mean “either or both” of the elements so conjoined, i.e., elements that are conjunctively present in some cases and disjunctively present in other cases. Multiple elements listed with “and/or” should be construed in the same fashion, i.e., “one or more” of the elements so conjoined. Other elements may optionally be present other than the elements specifically identified by the “and/or” clause, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, a reference to “A and/or B”, when used in conjunction with open-ended language such as “comprising” can refer, in one embodiment, to A only (optionally including elements other than B); in another embodiment, to B only (optionally including elements other than A); in yet another embodiment, to both A and B (optionally including other elements); etc.

[00157] As used herein in the specification and in the claims, “or” should be understood to have the same meaning as “and/or” as defined above. For example, when separating items in a list, “or” or “and/or” shall be interpreted as being inclusive, i.e., the inclusion of at least one, but also including more than one, of a number or list of elements, and, optionally, additional unlisted items. Only terms clearly indicated to the contrary, such as “only one of” or “exactly one of,” or, when used in the claims, “consisting of,” will refer to the inclusion of exactly one element of a number or list of elements. In general, the term “or” as used herein shall only be interpreted as indicating exclusive alternatives (i.e. “one or the other but not both”) when preceded by terms of exclusivity, such as “either,” “one of,” “only one of,” or “exactly one of.” “Consisting essentially of,” when used in the claims, shall have its ordinary meaning as used in the field of patent law.

[00158] Use of ordinal terms such as “first,” “second,” “third,” etc., in the claims to modify a claim element does not by itself connote any priority, precedence, or order of one claim element over another or the temporal order in which acts of a method are performed. Such terms are used merely as labels to distinguish one claim element

- 39 -

having a certain name from another element having a same name (but for use of the ordinal term).

[00159] The phraseology and terminology used herein is for the purpose of description and should not be regarded as limiting. The use of "including," "comprising," "having," "containing", "involving", and variations thereof, is meant to encompass the items listed thereafter and additional items.

[00160] Having described several embodiments of the invention in detail, various modifications and improvements will readily occur to those skilled in the art. Such modifications and improvements are intended to be within the spirit and scope of the invention. Accordingly, the foregoing description is by way of example only, and is not intended as limiting. The invention is limited only as defined by the following claims and the equivalents thereto.

[00161] What is claimed is:

- 40 -

CLAIMS

1. A live streaming system for streaming content to a plurality of client devices, the system comprising:

a controller configured to control testing of the live streaming system, the controller comprising a processor configured to execute a method comprising

determining a start time for an experiment for testing the live streaming system;

sending a message to a subset of the plurality of client devices indicating that each client device in the subset can participate in the experiment, the message comprising experiment control parameters; and

monitoring the experiment,

wherein at least one client device in the subset hosts a first streaming engine and a second streaming engine.

2. The live streaming system of claim 1, wherein the method further comprises:
specifying the experiment by specifying a client device behavior pattern for one or more groups of client devices.

3. The live streaming system of claim 2, wherein the client device behavior pattern comprises information indicating, either explicitly or implicitly, a start time at which the client device starts participating in the experiment.

4. The live streaming system of claim 2, wherein the client device behavior pattern comprises a representation of an arrival rate function.

5. The live streaming system of claim 1, wherein determining the start time for the experiment comprises determining the start time for the experiment based at least in part on a predicted number of client devices appropriate for the experiment at the start time.

6. The live streaming system of claim 1, wherein the experiment control

- 41 -

parameters comprise at least one parameter selected from the group consisting of a start time for the experiment, duration for the experiment, a client device behavior pattern, and a probability of participation in the experiment.

7. The live streaming system of claim 1, wherein the subset of client devices includes at least 10,000 client devices.

8. The live streaming system of claim 1, wherein the subset of client devices includes at least 100,000 client devices.

9. The live streaming system of claim 1, wherein the monitoring comprises:
detecting an early departure of a client device;
obtaining a state of the departed client device; and
sending the state of the departed client device to a replacement client device.

10. The live streaming system of claim 9, wherein obtaining the state of the departed client device comprises storing the state of the departed client device.

11. The live streaming system of claim 9, wherein the state comprises at least one of a start time of the departed client device, a departure time of the departed client device, a list of neighbors of the departed client device, and a list of units of streaming content downloaded by the departed client device.

12. The live streaming system of claim 1, wherein the monitoring comprises:
receiving a performance report from a client device in the subset of client devices.

13. The live streaming system of claim 12, wherein at least one of the performance reports comprises information indicative of the performance of the client device.

14. The live streaming system of claim 13, wherein the information comprises at least one of a number of units of streaming content downloaded, uploaded and/or processed by the first streaming engine; a number of units of streaming content that the

- 42 -

first streaming engine failed to download, upload, and/or process; data related to network behavior, and data related to utilization of a resource of the client device.

15. The live streaming system of claim 1, wherein the first streaming engine is an experimental streaming engine and the second streaming engine is a protection streaming engine associated with the experimental streaming engine.

16. A method for testing a first streaming engine hosted on a client device also hosting a second streaming engine, the method comprising:

using the first streaming engine to obtain a unit of streaming content;

determining if the unit of streaming content is obtained by the first streaming engine; and

if it is determined that the unit of streaming content is not obtained by the first streaming engine, using the second streaming engine to obtain the unit of streaming content.

17. The method of claim 16, further comprising determining a start time to start to use the first streaming engine based at least in part on experiment control parameters.

18. The method of claim 17, wherein the method comprises using the first streaming engine to obtain the unit of streaming content at or after the start time.

19. The method of claim 16, further comprising:

using the first streaming engine to obtain another unit of streaming content;

determining if the other unit of streaming content is obtained by the first streaming engine; and

if it is determined that the other unit of streaming content is not obtained by the first streaming engine, using the second streaming engine to obtain the other unit of streaming content.

20. The method of claim 17, further comprising:

receiving experiment control parameters associated with an experiment to test a

- 43 -

live streaming system.

21. The method of claim 17, wherein the experiment control parameters are received from a controller configured to control testing of the live-streaming system.

22. The method of claim 17, wherein the experiment control parameters comprise at least one parameter selected from the group consisting of a start time of the experiment, duration of the experiment, a client device behavior pattern, and a probability of participation in the experiment.

23. The method of claim 18, wherein the client device hosts a third streaming engine, the method further comprising:

using the third streaming engine to obtain streaming content prior to the start time.

24. The method of claim 17, further comprising:

stopping using the first streaming engine at a stop time determined based at least in part on the experiment control parameters; and

starting to use the third streaming engine at or after the stop time.

25. The method of claim 17, wherein determining the start time comprises selecting a start time based at least in part on a distribution function that depends on the experiment control parameters.

26. The method of claim 16, further comprising:

sending a performance report to a controller configured to control testing of the live streaming system.

27. A computer readable storage medium storing a plurality of processor-executable components that when executed by a processor, comprise:

an experimental streaming engine configured to obtain streaming content;

a protection streaming engine configured to obtain streaming content;

- 44 -

a streaming hypervisor configured to adaptively allocate a plurality of tasks between the first streaming engine and the second streaming engine.

28. The computer readable storage medium of claim 27, wherein the protection streaming engine is configured to obtain a unit of streaming content if the experimental streaming engine fails to obtain the unit of streaming content.

29. The computer readable storage medium of claim 27, wherein the experimental streaming engine comprises an experimental software module, and the protection streaming engine does not comprise the experimental software module.

30. The computer readable storage medium of claim 27, wherein the protection streaming engine is a content delivery network streaming engine, a peer-to-peer streaming engine, or a hybrid streaming engine.

31. The computer readable storage medium of claim 27, wherein the protection streaming engine comprises a content delivery network streaming engine and a peer-to-peer streaming engine.

32. The computer readable storage medium of claim 27, wherein the experimental streaming engine is a content delivery network streaming engine, a peer-to-peer streaming engine, or a hybrid streaming engine.

33. The computer readable storage medium of claim 27, wherein:
the experimental streaming engine comprises a first play-point parameter and a first download window indicating a set of units of streaming content the experimental streaming engine can download; and

the protection streaming engine comprises a second play-point parameter and a second download window indicating a set of units of streaming content the protection streaming engine can download.

34. The computer readable storage medium of claim 33, wherein:

- 45 -

a value of the first play-point parameter is greater than a value of the second play-point parameter.

35. The computer readable storage medium of claim 33, wherein:

the first download window indicates a set of units of streaming content disjoint from a set of units of streaming content indicated by the second download window.

36. The computer readable storage medium of claim 33, wherein:

the first download window indicates a set of units of streaming content to be played after units in the set of units of streaming content indicated by the second download window are played.

37. The computer readable storage medium of claim 33, wherein a task in the plurality of tasks comprises downloading a unit of streaming content.

38. The computer readable storage medium of claim 33, wherein a task in the plurality of tasks comprises uploading a unit of streaming content.

39. The computer readable storage medium of claim 26, further comprising a media player configured to play, to a user, units of streaming content obtained by the experimental streaming engine and/or the protection streaming engine.

40. A method for controlling testing of a live streaming system for streaming content to a plurality of client devices, the method comprising:

determining a start time for an experiment for testing the live streaming system;
sending a message to a subset of the plurality of client devices indicating that each client device in the subset can participate in the experiment, the message comprising experiment control parameters; and
monitoring the experiment,
wherein at least one client device in the subset hosts a first streaming engine and a second streaming engine.

- 46 -

41. A device hosting a first streaming engine and a second streaming engine, the device comprising:

a processor configured to execute a method comprising

using the first streaming engine to obtain a unit of streaming content,

determining if the unit of streaming content is obtained by the first

streaming engine, and

if it is determined that the unit of streaming content is not obtained by the first streaming engine, using the second streaming engine to obtain the unit of streaming content.

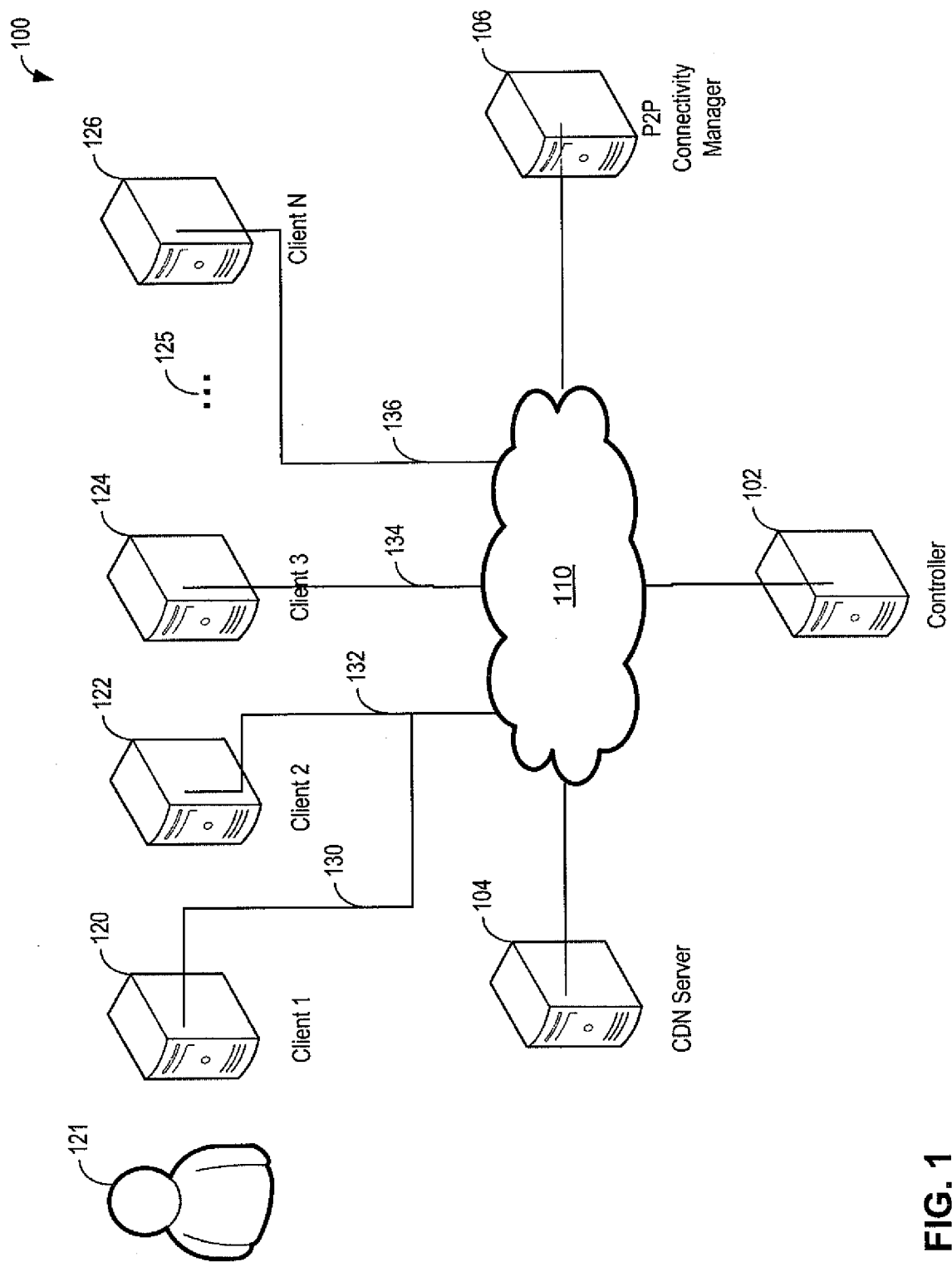


FIG. 1

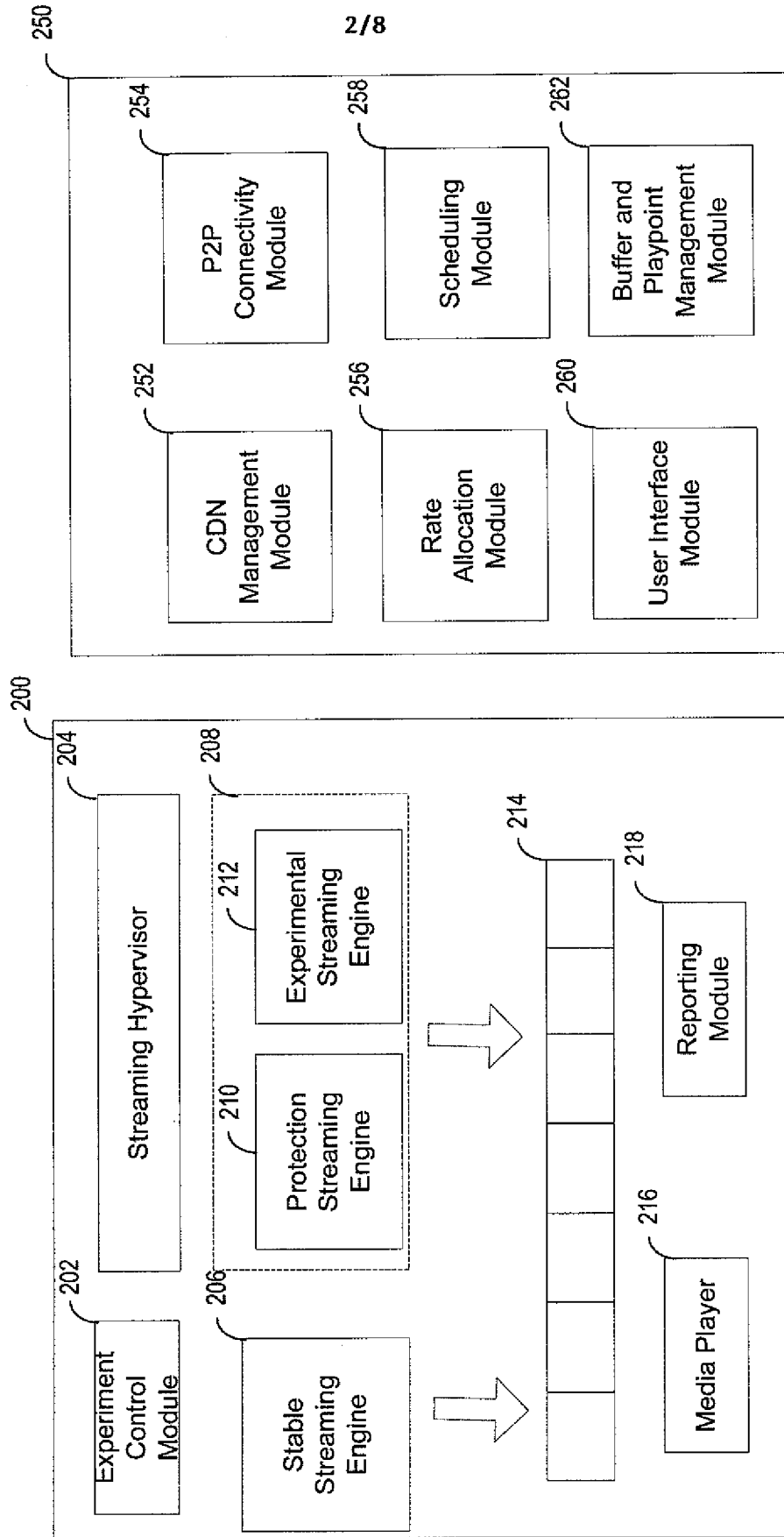
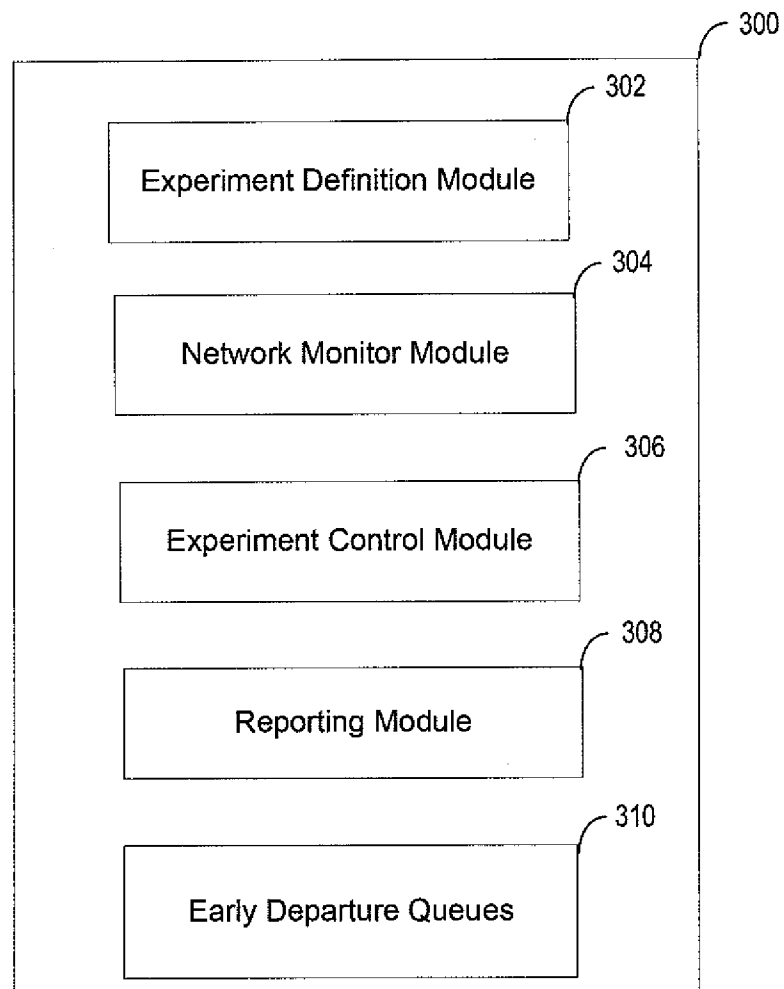


FIG. 2b

FIG. 2a

**FIG. 3**

4/8

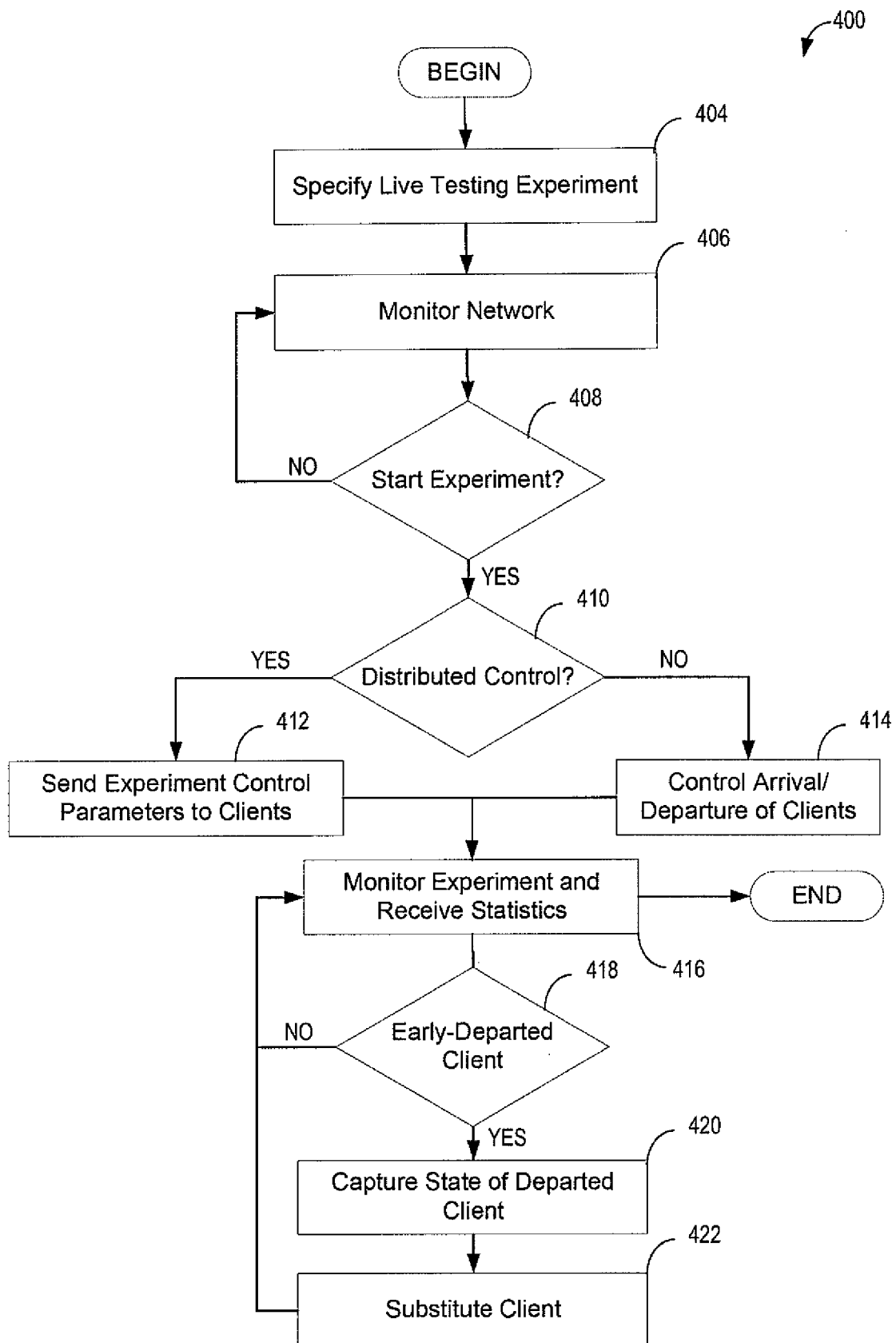
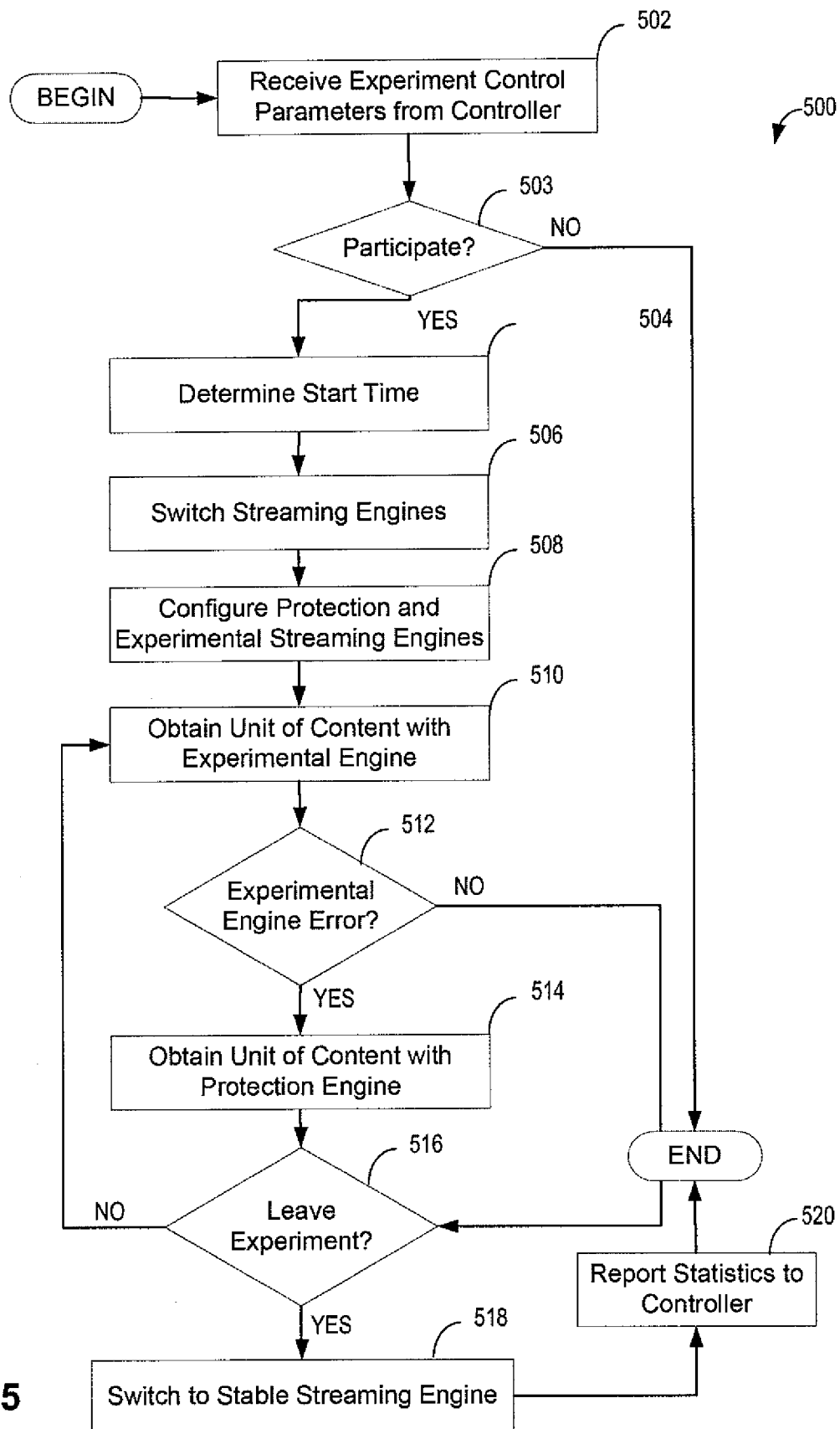


FIG. 4

5/8



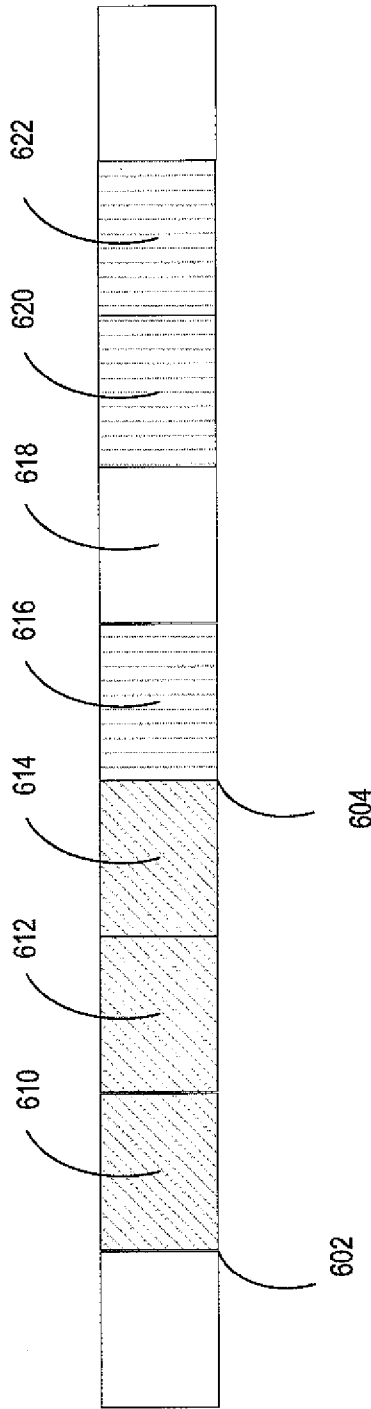


FIG. 6a

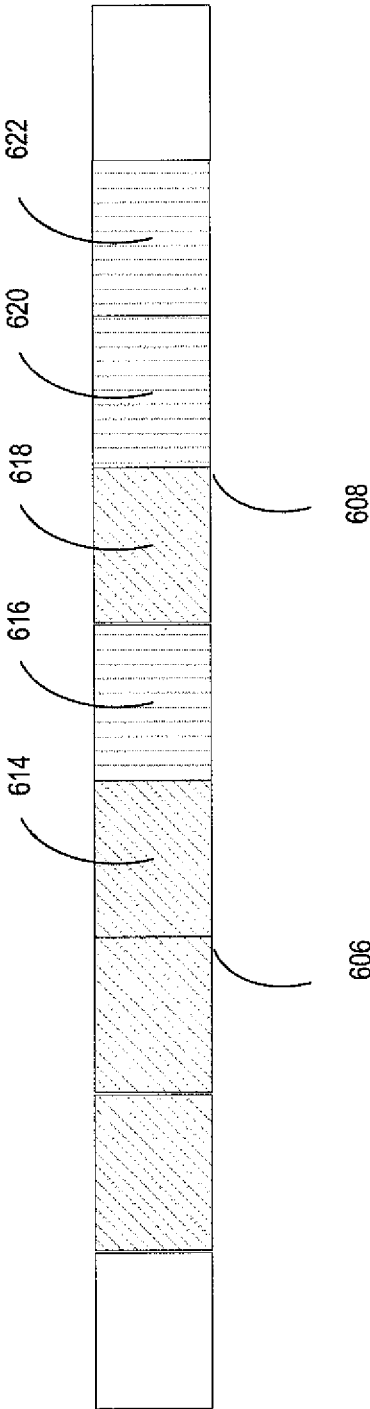


FIG. 6b

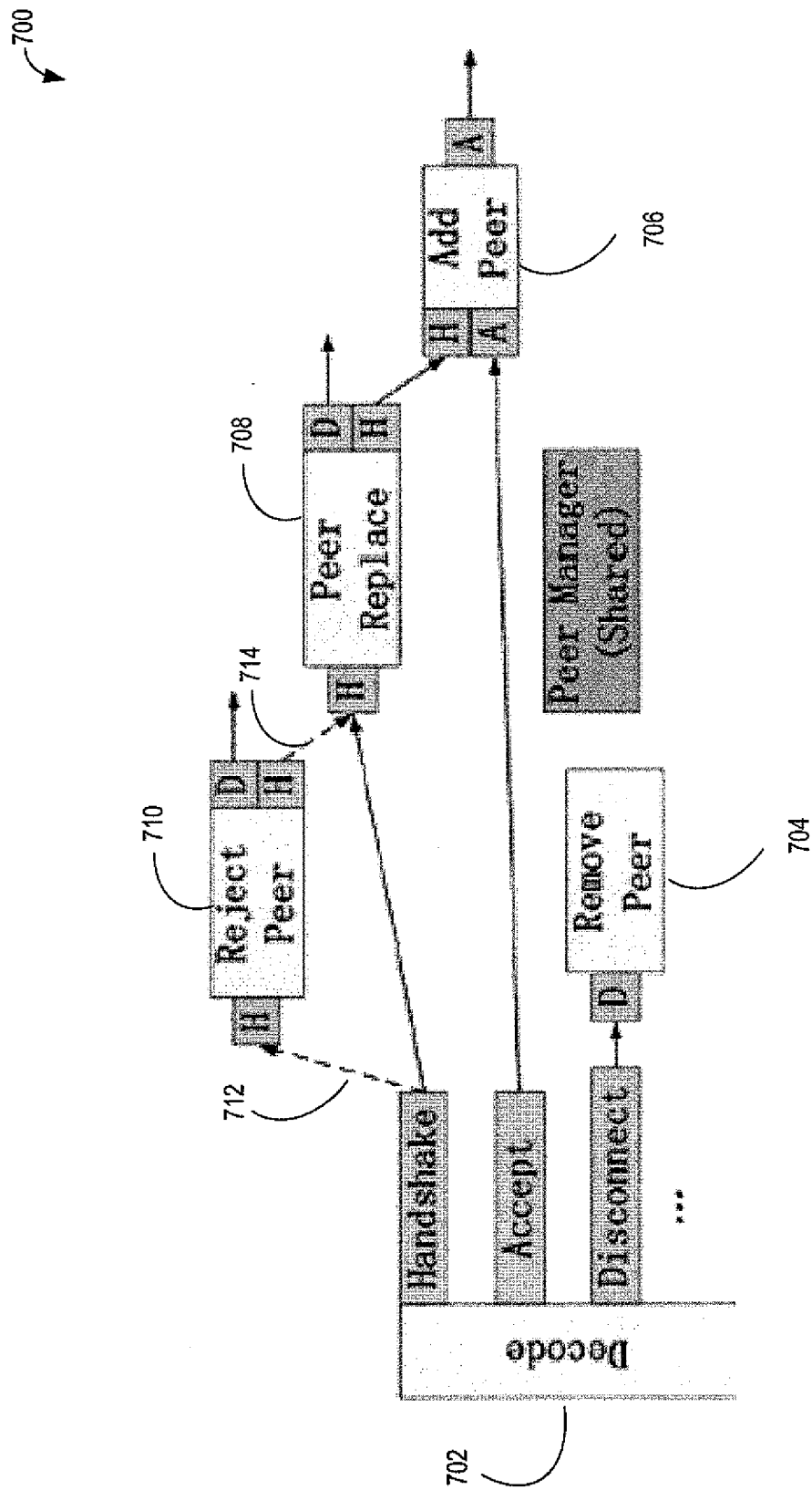


FIG. 7

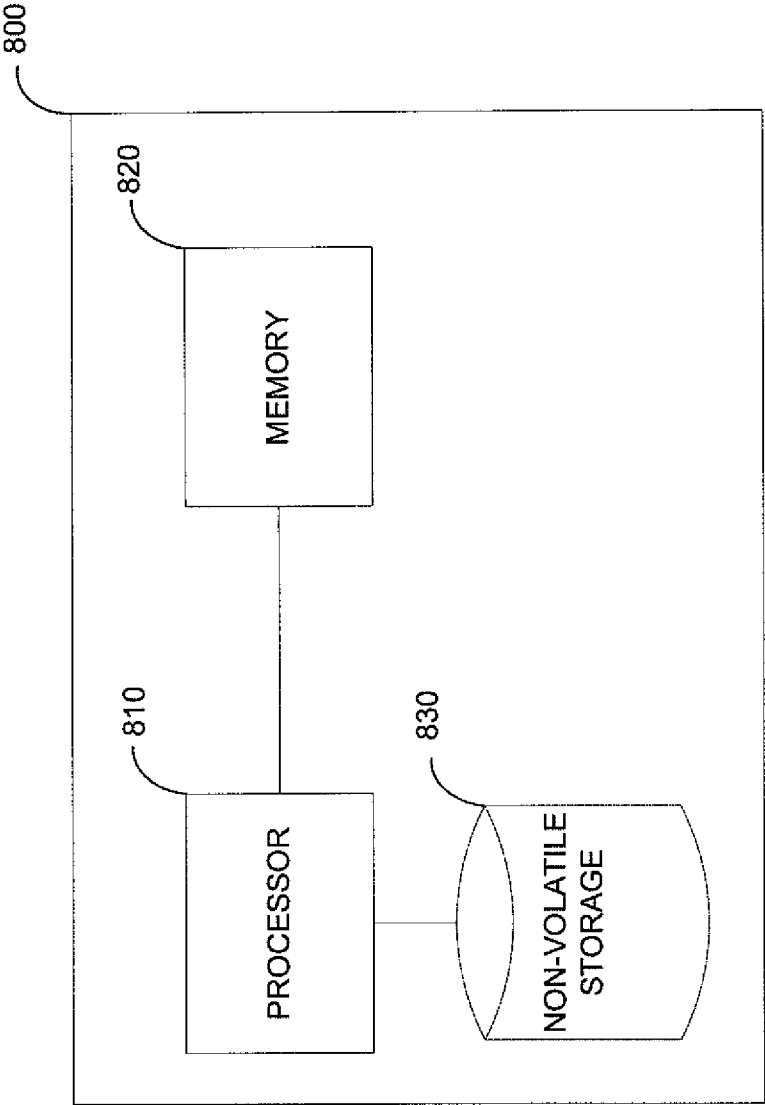


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 11/40833

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06F 19/00 (2011.01)

USPC - 702/122

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

USPC: 702/122

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 702/1, 108, 122, 127; 709/201, 203, 223, 224, 231 (text search)

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

PubWest (PGPB, USPT, EPAB, JPAB), Google,

Search terms used: stream, video, media, audio, content, multimedia, multi, media, test, experiment, monitor, analy, assess, rate, throughput, speed, quality, network, schedule, start, time, client, node, terminal, end, user, messag, report, engine, arriv, rate, duration, st

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X — Y	US 6,625,648 B1 (Schwaller et al.) 23 September 2003 (23.09.2003), col. 3, ln. 6-67, col. 4, ln. 8-48, col. 7, ln. 4-66, col. 8, ln. 7-66, col. 9, ln. 6-26, col. 10, ln. 45-46, col. 12, ln. 50, col. 14, ln. 39-49, col. 18, ln. 59-65	1-6, 9-41 ----- 7, 8
Y	US 2010/0030423 A1 (Nathanson) 04 February 2010 (04.02.2010), para [0356]	7, 8
A	US 2008/0146216 A1 (Newman et al.) 19 June 2008 (19.06.2008), entire document	1-41
A	US 2006/0184670 A1 (Beeson et al.) 17 August 2006 (17.08.2006), entire document	1-41
A	US 2009/0182848 A1 (Brown et al.) 16 July 2009 (16.07.2009), entire document	1-41
A	US 2008/0112332 A1 (Pepper) 15 May 2008 (15.05.2008), entire document	1-41
A	US 2007/0076605 A1 (Cidon et al.) 05 April 2007 (05.04.2007), entire document	1-41

☐ Further documents are listed in the continuation of Box C.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

03 October 2011 (03.10.2011)

Date of mailing of the international search report

13 OCT 2011

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450

Facsimile No. 571-273-3201

Authorized officer:

Lee W. Young

PCT Helpdesk: 571-272-4300

PCT OSP: 571-272-7774