

PATENT APPLICATION

**DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED
MEDIA: APPLICATION PROGRAMMING INTERFACE**

Inventors: Albert John McGowan, a citizen of The United States, residing at
 4201 E. Camelback Rd., #101
 Phoenix AZ, 85018

 Michael M. Gordon, a citizen of The United States, residing at
 6500 E. Cactus Wren Road
 Paradise Valley, AZ 85253

Assignee: Brightcove, Inc.
 290 Congress Street
 Boston, MA 02210

Entity: Small

**DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED LIVE
AND ON-DEMAND MEDIA: APPLICATION PROGRAMMING
INTERFACE**

[0001] This application is a continuation-in-part of co-pending, commonly assigned U.S.

Patent Application No. 14/918,368, filed October 20, 2015, entitled “DYNAMIC CHUNK
MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING
INTERFACE,” which is a continuation of commonly assigned U.S. Patent Application No.
14/086,801, filed November 21, 2013, entitled “DYNAMIC CHUNK MANIPULATION FOR
STREAMING MIXED LIVE AND ON-DEMAND MEDIA: APPLICATION

PROGRAMMING INTERFACE,” which issued as U.S. Patent No. 9,197,688, on November 24,
2015, which claims the benefit of commonly assigned U.S. Provisional Application No.
61/884,709, entitled “DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED
LIVE AND ON-DEMAND MEDIA,” filed September 30, 2013. The above-listed applications
are hereby incorporated by reference in their entirety for all purposes.

[0002] This application is also a continuation-in-part of co-pending, commonly assigned U.S.
Patent Application No. 13/339,680, filed December 29, 2011, entitled “MULTI-PLATFORM
MEDIA SYNDICATION CUSTOMIZATION,” which is a continuation-in-part of commonly
assigned U.S. Patent Application No. 13/245,372, filed September 26, 2011, entitled “SINGLE-
URL CONTENT DELIVERY.” The above-listed applications are hereby incorporated by
reference in their entirety for all purposes.

[0003] This application is also a continuation-in-part of co-pending, commonly assigned U.S.
Patent Application No. 13/791,789, filed March 8, 2013, entitled “DYNAMIC CHUNKING
FOR DELIVERY INSTANCES,” which is a continuation-in-part of commonly assigned U.S.
Patent Application No. 13/624,029, filed September 21, 2012, entitled “DYNAMIC
CHUNKING FOR DELIVERY INSTANCES,” which issued as U.S. Patent No. 8,645,504, on
February 4, 2014, which is a continuation of commonly assigned U.S. Patent Application No.
13/430,081, filed March 26, 2012, entitled "DYNAMIC CHUNKING FOR DELIVERY
INSTANCES", which issued as U.S. Patent No. 8,301,733, on October 30, 2012, which is a
continuation-in-part commonly assigned U.S. Patent Application No. 12/976,883, filed

December 22, 2010, entitled "DYNAMIC CHUNKING FOR MEDIA STREAMING," which issued as U.S. Patent No. 8,145,782, on March 27, 2012, which claims priority to Australian Patent Application Serial No. 2010202741, filed June 30, 2010, entitled "DYNAMIC CHUNKING FOR MEDIA STREAMING." The above-listed applications are hereby
5 incorporated by reference in their entirety for all purposes.

[0004] Additionally, Application No. 14/086,822, filed November 21, 2013, entitled "DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED LIVE AND ON-DEMAND MEDIA: DYNAMIC PERMUTATION LAYER" which issued as U.S. Patent No. 9,332,047, on May 3, 2016 is incorporated by reference into this application for all purposes.

10 BACKGROUND OF THE INVENTION

[0005] The delivery of media over data networks such as the Internet is in high demand. This is due, in large part, to much of the media being advertisement ("ad") supported. This even applies to streamed live content.

[0006] The insertion of ads and other on-demand content into a stream of live content can be
15 difficult. The unpredictable nature of live content can result in corresponding unpredictability of points in the live content at which ads may be shown. Furthermore, because live content can be generated for hours, weeks, or longer, the timestamps associated with live will not be synchronized with the timestamps of ads (or other on-demand content) inserted into the stream of live content. This can cause problems for some clients during playback.

20 BRIEF SUMMARY OF THE INVENTION

[0007] Techniques disclosed herein provide for dynamically rewriting the timestamps in streams of media that transition from one source to another (such as from live content to on-demand content, and vice versa). Embodiments can coordinate the creation of a client manifest with the dynamic creation of a requested segment of media (i.e., "chunk") to determine how to
25 rewrite timestamps of requested chunks are continuous through the transition.

[0008] An example method of providing media streaming via a data communications network, according to the disclosure, includes receiving a stream of data representing live media content, obtaining timing information of the stream of data, and receiving, via the data communications network, a request to stream the live media content. The method further includes creating, with a

processing unit, a manifest file where the manifest file includes information for streaming one or more segments of the live media content via the data communications network, the manifest file also includes information for streaming one or more segments of a media file, distinct from the live media content, and the manifest file further includes offset information for streaming the one or more segments of the media file, the offset information based on the timing information of the stream of data. The method further includes and sending, via the data communications network, the manifest file.

[0009] The method of providing media streaming via a data communications network can include one or more of the following features. The method can include providing an application programming interface (API), where request to stream the live media content is received via the API. Creating the manifest file further can include providing, in the manifest file, an indication of a discontinuity in the stream of the live media content. The method can also include determining a type of client associated with the request to stream the live media content, where providing the indication of the discontinuity is based on the determination of the type of client. Obtaining the timing information of the stream of data can include sending a request for the timing information, and receiving a response to the request, the response including the timing information. The method can also include receiving an indication of a period of time in the live media content designated for advertising, where sending the request for the timing information is based on the receipt of the indication of the period of time in the live media content designated for advertising. The timing information of the stream of data can include one or more timestamps associated with one or more segments of the stream of data.

[0010] An example server for providing media streaming via a data communications network, according to the disclosure, includes a communications interface, a memory, and a processing unit coupled to the communications interface and the memory. The processing unit is configured to cause the server to receive a stream of data representing live media content, obtain timing information of the stream of data, and receive, via the communications interface, a request to stream the live media content. The processing unit is further configured to create a manifest file where the manifest file includes information for streaming one or more segments of the live media content via the data communications network, the manifest file also includes information for streaming one or more segments of a media file, distinct from the live media content, and the manifest file further includes offset information for streaming the one or more segments of the

media file, the offset information based on the timing information of the stream of data. The processing unit is also configured to send, via the communications interface, the manifest file.

[0011] The server for providing media streaming via a data communications network can include one or more of the following features. The processing unit can be further configured to cause the server to provide an application programming interface (API), wherein request to stream the live media content is received via the API. The processing unit can be configured to cause the server to create the manifest file by providing, in the manifest file, an indication of a discontinuity in the stream of the live media content. The processing unit can be further configured to cause the server to determine a type of client associated with the request to stream the live media content where providing the indication of the discontinuity is based on the determination of the type of client. The processing unit can be configured to cause the server to obtain the timing information of the stream of data by sending a request for the timing information, and receiving a response to the request, the response including the timing information. The processing unit can be further configured to cause the server to receive an indication of a period of time in the live media content designated for advertising where sending the request for the timing information is based on the receipt of the indication of the period of time in the live media content designated for advertising. The timing information of the stream of data can include one or more timestamps associated with one or more segments of the stream of data.

[0012] An example non-transitory computer-readable medium, according to the disclosure, has instructions embedded thereon for providing media streaming via a data communications network. The instructions, when executed by a computer, cause the computer to perform functions including: receiving a stream of data representing live media content, obtaining timing information of the stream of data, and receiving a request to stream the live media content. The functions further include creating a manifest file, where the manifest file includes information for streaming one or more segments of the live media content via the data communications network, the manifest file also includes information for streaming one or more segments of a media file, distinct from the live media content, and the manifest file further includes offset information for streaming the one or more segments of the media file, the offset information based on the timing information of the stream of data. The functions also include sending the manifest file.

[0013] The computer-readable medium can include one or more of the following features. The computer-readable medium can include instructions for causing the computer to provide an application programming interface (API), wherein request to stream the live media content is received via the API. The instructions for creating the manifest file can further include

5 instructions for providing, in the manifest file, an indication of a discontinuity in the stream of the live media content. The computer-readable medium can include instructions for causing the computer to determine a type of client associated with the request to stream the live media content, where providing the indication of the discontinuity is based on the determination of the type of client. The instructions for obtaining the timing information of the stream of data can

10 further include instructions for causing the computer to send a request for the timing information, and receive a response to the request, the response including the timing information. The computer-readable medium can include instructions for causing the computer to receive an indication of a period of time in the live media content designated for advertising where sending the request for the timing information is based on the receipt of the indication of the period of

15 time in the live media content designated for advertising.

[0014] Items and/or techniques described herein may provide one or more of the following capabilities, as well as other capabilities not mentioned. Techniques allow for a client to transition from one source of streaming media to another without incurring errors due to inconsistencies in the timestamps of chunks between the two sources of media. Techniques can

20 be dynamically executed on the fly and customized for each client. These and other embodiments, along with many of its advantages and features, are described in more detail in conjunction with the text below and attached figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] The present disclosure is described in conjunction with the appended figures:

[0016] FIG. 1 is a block diagram illustrating a media servicing system, according to some embodiments of the present invention.

[0017] FIG. 2 is a functional block diagram illustrating a system configured to provide dynamic chunk manipulation according one embodiment.

[0018] FIG. 3 is a functional block diagram of a system, similar to the system of FIG. 2, incorporating ad server(s).

[0019] FIG. 4 is a simplified block diagram illustrating an example of how timestamps can be rewritten.

5 [0020] FIG. 5 is a swim-lane diagram illustrating example interactions between a client, API, and DPL.

[0021] FIG. 6 is a simplified flow chart illustrating an example method of providing a manifest file in accordance techniques described herein.

10 [0022] FIG. 7 is a simplified flow chart illustrating an example method of creating media chunks using offset information included in a request, in accordance with techniques described herein.

[0023] FIG. 8 illustrates an embodiment of a computer system.

15 [0024] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION OF THE INVENTION

20 [0025] The ensuing description provides preferred exemplary embodiment(s) only, and is not intended to limit the scope, applicability or configuration of the disclosure. Rather, the ensuing description of the preferred exemplary embodiment(s) will provide those skilled in the art with an enabling description for implementing various embodiments of the invention. It is understood that various changes may be made in the function and arrangement of elements without departing
25 from the spirit and scope as set forth in the appended claims.

[0026] The terms “ad” and “advertisement,” including alternate forms thereof, refer to marketing content distinct from user-requested media content. Although the techniques described herein discuss obtaining and providing advertising data, they also can be applied to

data for content other than advertising, such as other on-demand content. Furthermore, although techniques described herein are often provided in the context of video streaming, they can be applied to other forms of media content (e.g., audio streaming) as well. A person of ordinary skill in the art will recognize many alternate applications.

5 **[0027]** It can be noted that, although embodiments disclosed herein describe techniques as implemented by a cloud-hosted integrated multi-node pipelining system (CHIMPS), embodiments are not so limited. Other systems may be configured to implement the techniques disclosed.

10 **[0028]** The increased availability of media content over data communications networks such as the Internet has mirrored the increased bandwidth for these networks. Because media has recently taken a more prominent role in data communications, the distribution of media and the data associated with such distribution has become increasingly important, particularly to media content providers. Much of this media is ad-supported, allowing media content providers to receive advertising revenue from media content, while often allowing end users to consume the
15 media content free of charge.

20 **[0029]** Advertisements are often stored and maintained separate from the primary media content. The distribution of ad-supported media via the Internet can therefore involve a variety of entities. **FIG. 1** is a block diagram illustrating a media servicing system 100, according to some embodiments of the present invention. The system may deliver media content to a client
25 145, executed by an end user device 140 providing media playback to an end user. The client 145 can be, for example, a media player, browser, or other application adapted to request and/or play media files. The media content can be provided via a network such as the Internet 170 and/or other data communications networks, such as a distribution network for television content. The end user device 140 can be one of any number of devices configured to receive
media over the Internet 170, such as a mobile phone, tablet, personal computer, portable media device, set-top box, video game system, etc. Although only one client 145 and one end user device 140 are shown in FIG. 1, it will be understood that the media servicing system 100 can provide media to many (hundreds, thousands, millions, etc.) of clients 145 on many (hundreds, thousands, millions, etc.) of end user devices 140.

[0030] For on-demand content (e.g., requested media that is stored in its entirety), a media file provided by one or more media providers 130 can be processed and indexed by cloud-hosted integrated multi-node pipelining system (CHIMPS) 110. The media file may be stored on media file delivery service provider (MFDSP) 150, such as a content delivery network, media streaming service provider, cloud data services provider, or other third-party media file delivery service provider. Additionally or alternatively, the CHIMPS 110 may also be adapted to store the media file. On-demand content can be provided to the client 145 via progressive downloading and/or streaming.

[0031] For purposes of this disclosure, advertising content can be considered a form of on-demand content that is distributed much the same way. (As previously mentioned, techniques for interleaving live and on-demand content apply not only to inserting ads into live content, but can apply to inserting any form of on-demand content, including non-advertisement content as well.) Rather than media providers, however, ad content can come from ad networks 160. Furthermore, the ad networks 160 may determine what ads to provide at a given point in time for a given client 145 by implementing business rules.

[0032] For live content (e.g., requested content that is sent to one or more end user devices 140 as it is received from media provider(s) 130, that is, in real time or near-real time, depending on processing times and/or other buffer times), a similar process can take place. For example, media provider(s) 130 can provide a media stream (e.g., live video), which is processed and indexed by the CHIMPS 110. Encoded segments of the media stream can be stored as files (i.e., “chunks”), on the media file delivery service provider (MFDSP) 150 and/or the CHIMPS 110. Embodiments may have a delay from the time the media stream is received to the time the associated content is stored and available for streaming to the one or more end user devices 140. The delay can be due to transcoding and/or other types of processing for making the received media stream available for downloading. This delay can create a buffer period of time in which one or more of the techniques described herein can take place, such as processing cue tones and/or placing ads in the content. (These techniques are described in greater detail below.)

[0033] Both on-demand and live content can utilize any of a variety of forms of streaming media. One such method is chunk-based media streaming in which a media file or live stream is processed into smaller chunks and stored (e.g., in a server of the CHIMPS 110 or Media File

Delivery Service Provider 150) for serving to a client 145. The client 145 can make a URL request to the CHIMPS 110, which can provide a manifest file (also known as an index file) indicating the locations of each of the chunks of media using, for example, Uniform Resource Indicators (URIs) (e.g., Universal Resource Locators (URLs)) or other indicators. The client 145 can then use the information in the manifest file to stream the media content, following one location after the other to download each chunk of media. The client may sequentially request and receive multiple manifest files to stream requested media content. This is especially true for live media, where the live stream might still be received and processed by the CHIMPS 110 when the client makes the request for the manifest file. Additional detail regarding such chunking and indexing, as well as techniques for dynamically creating chunks and manifest files, can be found in U.S. Patent Nos. 8,327,013 entitled “Dynamic Index File Creation for Media Streaming” and 8,145,782, entitled “Dynamic Chunking For Media Streaming,” both of which are incorporated by reference herein in their entirety.

[0034] The CHIMPS 110 can further manage the processing and syndication of media (live or on-demand) received from the media provider(s) 130. For example, the CHIMPS 110 can provide transcoding and other services to enable media provided by the media provider(s) to be distributed in a variety of formats to a variety of different device types in a variety of locations. Furthermore, the CHIMPS 110 provide feedback to the media provider(s) 130 regarding the media’s syndication, including user behavior during media playback. For example, the CHIMPS 110 can provide a media provider 130 with information indicating that end users tend to stop watching a video at a certain point in playback, or that users tend to follow links associated with certain advertisements displayed during playback. With this data, media provider(s) 130 can adjust factors such as media content, advertisement placement and content, etc., to increase revenue associated with the media content and provide the end user device 140 with a more desirable playback experience. Additionally or alternatively, the CHIMPS 110 can dynamically provide a customized playback experience on the end user device 140 according to aspects of the context associated with the content at the time of the request, aspects of the content request itself, or both. It can be noted that although embodiments herein may utilize media files and live streams explicitly, other embodiments may utilized other forms of media assets, such as dynamic web pages, and may incorporate multiple media elements, including players, user interface components, user controls and control components, images, and other media content, objects, or

types. Additionally, it can be noted that various functions, operations, processes, or other aspects that are described in this example, and other examples, as being performed by or attributable to the CHIMPS 110 can be performed by another system operating in conjunction with the CHIMPS 110, loosely or tightly synchronized with the CHIMPS 110, or independently; for example, collecting data from other digital services to be combined and reported with data collected by the CHIMPS 110 can, in some implementations, be performed by a system other than the CHIMPS 110. Additional detail regarding the functionality of the CHIMPS 110 can be found in in U.S. Patent No. 8,301,733, entitled “Dynamic Chunking for Delivery Instances,” which is incorporated by reference herein in its entirety.

[0035] A content owner 120 can utilize one or more media provider(s) 130 to distribute media content owned by the content owner 120. For example, a content owner 120 could be a movie studio that licenses distribution of certain media through various media providers 130 such as television networks, Internet media streaming websites and other on-demand media providers, media conglomerates, and the like. In some configurations, the content owner 120 also can operate as a media provider 130.

[0036] The content owner 120 and/or media provider(s) 130 can enter into an agreement with one or more ad network(s) 160 to provide advertisements to numerous clients 145 on numerous end user devices 140. In this manner, the ad network(s) 160 allow companies to show advertisements to end users viewing the media content from the media provider(s) 130. Because ad network(s) 160 can maintain advertisements and/or advertisement data separate from media content, the advertisements can be updated and—as previously indicated—subject to business rules such that, two users viewing the same media content at different times and/or in different locations may see different advertisements.

[0037] Advertisements may be inserted directly into content consumed by the client, depending on the streaming techniques utilized. In such cases, a Live Advertising Processing Engine Service (APES) can be utilized to provide the client with advanced functionality for advertising during the consumption of live content by a client. Additional details regarding Live APES can be can be found in in U.S. Patent App. No. 14/069,961 entitled “Live Advertising Processing Engine Service,” which is incorporated by reference herein in its entirety.

[0038] The insertion of ads directly into content consumed by the client can cause playback issues in at least some types of clients. One such issue can arise from the mismatch between timestamps of the live media (e.g., presentation timestamp (PTS) values) and those of the ad. Some clients will incur playback errors when encountering such mismatches. Techniques disclosed herein can provide for dynamically re-writing timestamps for on-demand (e.g., ad) content into stream of live content, which can prevent these playback errors from occurring.

[0039] Although embodiments provided herein discuss dynamically re-writing timestamps for inserting on-demand media into live content, embodiments are not so limited. Principles provided herein can be extended to embodiments that insert live content into on-demand media, a first on-demand media into a second on-demand media, first live content into second live content, and so forth. A person of ordinary skill in the art will recognize many substitutions, additions, omissions, and other variations to the embodiments provided herein.

[0040] **FIG. 2** is a functional block diagram illustrating a system 200 configured to provide dynamic chunk manipulation, including timestamp rewriting techniques provided herein, according one embodiment. The system 200 can comprise a client 145, MFDSP 150, application programming interface (API) 260, dynamic permutation layer (DPL) 270 origin 240, transcode module 230, clock 250, ingest module 220, and encoder 210, and one or more ad servers 310. As shown, these components can be executed by and/or incorporated into larger systems, such as the end user device 140, CHIMPS 110, and/or media provider(s) 130 as shown in FIG. 1. In other embodiments, however, the illustrated components may be incorporated differently into a larger system and/or may be stand-alone components. Furthermore, other embodiments may combine, separate, add, omit, substitute, and/or modify components of the system 200, while providing the same overall functionality as described herein. A person of ordinary skill in the art will recognize many substitutions, additions, omissions, and other variations to the embodiments provided herein.

[0041] In this embodiment, a live stream is provided by the encoder 210 in chunks (i.e., segments), each of which can each have a PTS value and/or other timing information as dictated by the encoder. Because live media can vary widely in duration, PTS values for the live stream can vary correspondingly. To help keep track of PTS values of the live stream received by the ingest module 220, a local clock 250 can be used. For example, a “source manifest” can be

created by recording the times at which chunks (having unique PTS values) of the live stream are received from the live stream source (i.e., the encoder 210). This source manifest can then be used by other components, such as the API 260, to determine which chunk was received at a given local time.

5 **[0042]** Ingested chunks are then provided to the transcode module 230, which transcodes the chunks into various formats according to a variety of different media profiles. Media profiles can determine bit rate, resolution, frame rate, and other media characteristics that can accommodate the various types of networks, clients, and/or end user devices used to stream the live content. For example, chunks with lower bit rates are generally used to stream the live
10 content to an end user device 140 that has a poor network connection or low-resolution display, whereas chunks with higher bit rates are generally used to stream the live content to an end user device 140 that has a good network connection or high-resolution display.

[0043] The various chunks are stored in the origin 240. The origin 240 can be a data storage comprising one or more physical storage devices, which can be distributed geographically, and
15 may be local to and/or remote from the transcode module 230 and/or DPL, depending on desired functionality. The origin 240 can further store and/or manage an “origin manifest” that records information regarding the available media profiles of the stored chunks and their corresponding PTS values. Thus, together with the source manifest, the origin manifest can be used to track the time at which each chunk of the live stream is received and determine the available media
20 profiles of the corresponding chunks stored in the origin 240.

[0044] The client 145 can request to stream content of the live stream from the API 260 by, for example, following a URL received from the media provider(s) 130. (This URL, or other URI, may be provided to the media provider(s) 130 by the CHIMPS 110, indicating an Internet location at which the live stream may be accessed.) The API 260, upon receiving the client’s
25 request, can create a client manifest (also known as an index file), comprising a list of links (e.g., URLs) to chunks of the live content. Once it receives the client manifest from the API 260, the client 145 can request the chunks from the MFDSP 150. If a chunk is not already cached at the MFDSP 150, the chunk request can be relayed to the DPL, which can dynamically create the requested chunk (e.g., at runtime) by retrieving one or more corresponding chunks stored at the
30 origin 240, relaying it back to the MFDSP 150, which can then provide the requested chunk to

the client 145. Because the DPL can create requested chunks on the fly, the requested chunks can be customized according to each request. For example, although chunks stored at the origin 240 may comprise a five-second segment of the live content, a client may request a 20-second chunk, in which case the DPL 270 can retrieve a plurality of five-second chunks from the origin corresponding to the requested 20-second chunk and dynamically create the requested chunk and provide the requested chunk to the MFDSP 150. (Additional detail regarding such chunking can be found in U.S. Patent No. 8,145,782, which is incorporated herein above.) Because the API 260 creates the client manifest used by the client 145 to request chunks, it is the API 260 that determines which chunks are requested and when. The API 260 can therefore craft client manifests to dictate at what point in the live content clients will begin streaming and the length of the chunks each client requests.

[0045] This ability to dynamically create both client manifests and requested chunks can also enable the system 200 to rewrite timestamps on the requested chunks to help ensure a client 145 receives a stream of chunks with continuous time stamps when streaming live content containing interleaved on-demand content, such as advertisements.

[0046] **FIG. 3** illustrates how a system 300, similar to the system 200 of FIG. 2, can incorporate ad server(s) 310 to determine which ads to insert into a stream of live content. Here, the ad server(s) 310 can be maintained by one or more ad network(s) 160, which may have an agreement with the media provider(s) 130 regarding which ads to run during an ad break of a live stream. In some embodiments, the DPL 270 may communicate with the ad server(s) 310 to obtain the ad content for the ads to run.

[0047] A typical live stream can consist of continuous data including one or more “content” segments (e.g. a sporting event, a news report, etc.), followed by a first advertising “break” during which advertising is intended to be shown. After the first advertising break, there can be a second set of one or more content segments, a second advertising break, and so forth. Advertising breaks can be determined by the media provider(s) 130 and indicated by “cue tones,” which are analog or digital signals which indicate in a first form that a break is about to begin. Cue tones can also indicate in a second form that the break is about to end. These cue tones can occur at any point in the stream, and in many circumstances the timing of the breaks is not known before the live stream is initiated.

[0048] The API 260 can determine when advertisements are needed (e.g., by receiving a cue tone) and generate a manifest file accordingly. As previously indicated, the processing of a live stream (e.g., ingest and transcode) can take time, causing a buffer period between when a live stream is ingested to when the live content is provided to the client. This buffer period can provide the API 260 with enough time to determine that a cue tone has been detected at a certain point in the live stream and generate the client manifest which inserts advertisements at corresponding points in the live content provided to the client 145. Ad server(s) 310 can indicate to the API 260 which ad(s) to insert and, optionally, provide additional metadata regarding the ad(s). The ad content may also be provided by the ad server(s) 310 and previously stored as chunks at the origin 240 and/or other data storage, enabling the DPL 270 to dynamically manipulate requested ad chunks.

[0049] In some embodiments, when the API 260 determines an ad is to be inserted into live content provided to the client 145, it can request timing information from the DPL 270 for a corresponding chunk, such as a PTS value. The API 260 can therefore determine what PTS value corresponds to the point in the live stream at which an ad is to be inserted, and include offset information in the client manifest. When using the client manifest to request a chunk, the request provided by the client 145 to the MFDSP 150 (relayed to the DPL 270) can include metadata, including the offset information, signaling to the DPL 270 to rewrite the timestamps of the requested ad chunks so that the chunks provided to the client have continuous timestamps, despite changes between the live content and ad chunks.

[0050] **FIG. 4** is a simplified block diagram provided to help illustrate an example of how timestamps can be rewritten. In this illustration, the chunks of the live stream 410 have PTS (timestamp) values of 11111, 11112, etc. Cue tones associated with the live stream may be received, indicating to the API 260 a starting point 412 and ending point 413 for an advertisement. To replace the chunks of live content 415 between the starting point 412 and ending point 413 with ad chunks 420 (or other on-demand content), the API 260 can request information from the DPL 270 regarding the PTS value of the chunk 411 before the starting point 412, for example. (In other embodiments, the API 260 may receive timing information regarding other chunks.)

[0051] Using this timing information, the API 260 can create a client manifest with URLs that include offset information in parameter strings that reflects this timing information. For example, the offset information can include an indication of the PTS value and/or an indication of a number by which ad chunks 420 should be offset to match the PTS values of the live stream 410. When these URLs are used by the client 145 to request ad chunks from the DPL 270 (via the MFDSP 150), the DPL 270 can then determine, from the parameter strings, the offset information. The DPL 270 can then, each time an ad chunk is requested, retrieve the ad chunk from the origin 240, dynamically offset the ad chunks according to the offset information (as shown by chunk sequence 430) and provide the requested chunk to the client 145 (via the MFDSP 150). This ultimately results in the client 145 receiving a mixed stream of chunks 440 with continuous timestamps that includes both live and on-demand (e.g., ad) content.

[0052] It can be noted that the example provided in FIG. 4 is a simplified one. In practice, systems may employ variations on this basic technique to accommodate different scenarios. For example, ad chunks 420 may differ in length the chunks of the live stream 410. However, because the DPL 270 is able to dynamically create requested chunks from the ad chunks 420, it is also able to provide chunks of a requested length. The API 260 can ensure that the client manifest provided to the client 145 indicates the proper length of requested chunks so that they match the chunk length of the live stream 410.

[0053] Also, although an ending point 413 in the example of FIG. 4 is known at the time the API 260 creates the corresponding client manifest, this may not always be the case. Depending on ad break length, client manifest size, buffer times, and/or other factors, the client manifest may not include a transition back from ad content to live content. However, this information may simply be provided in a subsequent client manifest.

[0054] Additionally, although the example of FIG. 4 suggests matching PTS values of chunks of a live stream 410 as received from an encoder 210, this may not always be the case. PTS values may be modified (or replaced entirely) when transcoded by the transcode module 230. However, because the DPL 270 can provide timing information regarding the transcoded chunks as they are stored in the origin 240, this is not an issue. The chunks of the live stream 410 with modified PTS values are provided to the client 145, and the timestamp values of the ad chunks 420 are rewritten to match these modified PTS values.

[0055] In some embodiments, the API 260 may additionally include a discontinuity tag (or other indicator) in the client manifest to flag the transition from live to on-demand content (and/or vice versa). This can indicate to the client 145 to clear their video buffer, enabling the client 145 to make the transition without any playback errors. Because some clients may require the discontinuity tag while others may not, the API 260 can first determine the type of client 145 consuming the live content and provide a discontinuity tag if the determined client is of a type that would need such a tag.

[0056] FIG. 5 is a swim-lane diagram illustrating example interactions between a client, API, and DPL, such as the client 145, API 260, and DPL 270 illustrated in FIGS. 2 and 3. As with all other figures provided herein, FIG. 5 is provided as a non-limiting example. Embodiments of the invention may employ entities other than the client, API, and/or DPL, and/or may add, omit, substitute and/or otherwise alter the interactions illustrated in FIG. 5.

[0057] At block 505, the API receives an indication to insert an ad into a live content. The indication can be, for example, a cue tone in and/or related to a live stream of media and/or an indication that such a cue tone has been detected by another component (e.g., an ingest module). As described in previous embodiments, the API may maintain and/or have access to a database, list, or other data structure, enabling the API to determine a time in the live media an ad break in the live content begins.

[0058] At block 510, the API requests timing information from the DPL, and the request is received at block 515. Timing information can include information regarding the PTS value of a chunk and/or other information that can be used to determine an offset needed to ensure continuity in the timestamps of chunks when transitioning to and from live content to ad content. The timing information is provided by the DPL at block 520 and received at block 525.

[0059] At block 530 the client request a client manifest. Due to the nature of live content, which is being received and processed by a CHIMPS or other system at the same (or substantially the same) time it is being sent to the client, the client manifest typically contains only a portion of the live content available. Thus, in order to consume all or a larger portion of the live content, a client can request multiple client manifests in succession, requesting a new client manifest when the live content corresponding to a previous manifest is or soon will be completely consumed. Thus, the client manifest requested by the client at block 530 may be one

of a series of client manifests requested by the client over a period of time while consuming the live content.

[0060] At block 535, the request for the client manifest is received by the API. Because the DPL is capable of dynamically manipulating chunks to accommodate chunk requests, the API can provide URLs (or other URIs) in the client manifest with parameter strings and/or other information that can be relayed to the DPL to dictate how the DPL manipulates the chunks that are ultimately provided to the client. Thus, the API can utilize the timing information received at block 525 to provide offset information in the client manifest to cause the DPL to rewrite timestamps of ad chunks inserted into the live content to ensure the timestamps received by the client are continuous. At block 545, the API can then return client manifest based on the timing information, which is received by the client at block 550.

[0061] Optionally, the API may insert a discontinuity tag into the client manifest, as shown at block 540. The location of the discontinuity tag may vary, depending on factors such as the streaming protocol and/or format of the client manifest. In some embodiments, for example, the discontinuity tag may simply be placed in between URLs on a sequential list of URLs in the client manifest that correspond to chunks to be consumed by a client. Thus, for transitions from live content to ad content, the discontinuity tag can follow URLs for chunks of live content and precede URLs for chunks of ad content. Depending on the needs of the client a discontinuity tag may additionally or alternatively be used in the transition back from ad content to live content, in a similar manner.

[0062] At block 555, the client uses the client manifest at block 550 to send a request for live chunk(s) (i.e., chunks of live content) from the DPL, which is received by the DPL at block 560. The DPL then prepares the live chunk(s), as detailed herein above, and returns the live chunk(s) at block 565, which are received by the client at block 570. (It will be understood that, where multiple chunks are requested, the process shown in blocks 555-570 is repeated for each chunk. That is, the client sends a separate request for each chunk requested.)

[0063] At block 575 the client uses the client manifest to send a request for ad chunk(s) (i.e., chunks of ad content) from the DPL, which is received by the DPL at block 580. (Again, It will be understood that, where multiple chunks are requested, the process shown in blocks 575-595 is repeated for each requested ad chunk.) Here, each chunk request can include offset information

provided in the client manifest to indicate to the DPL how the timestamps should be rewritten to ensure the timestamps of the ad chunks provided to the client are continuous with the timestamps of the live chunks previously received by the client at block 570. At block 585, the DPL normalizes ad chunks accordingly, using the offset information to rewrite the timestamps of the ad chunks. At block 590 the DPL returns the normalized ad chunks, which are received by the client at block 595.

[0064] It can be noted that the manifest file provided by the API can accommodate various other scenarios. For example, a manifest file may cause the client to initially request ad chunk(s), rather than live chunk(s).

[0065] FIG. 6 is a simplified flow chart illustrating an example method 600 of providing a manifest file in accordance with the techniques described above. The method 600 can be implemented, for example, by an API 260 and/or other components shown in FIGS. 2 and 3 above. As with all other figures provided herein, FIG. 6 is provided as an example and is not limiting. Various blocks may be combined, separated, and/or otherwise modified, depending on desired functionality. Furthermore, different blocks may be executed by different components of a system and/or different systems. Such systems can include the computer system, described herein below with regard to FIG. 8.

[0066] At block 605, a stream of data representing live media content is received. The stream of data can include a live stream received by an encoder of a media provider, and may be processed according to previously-described techniques.

[0067] At block 615, timing information of the stream of data is obtained. The timing information can include a PTS value or other information sufficient to determine offset information for timestamps of chunks of on-demand content inserted into the live media content.

[0068] At block 625 a request to stream the live media content is received. A request can include, for example, a request for a manifest file (e.g., a “client manifest” as described above).

[0069] The manifest file is created at block 635. Here, the manifest file includes information for streaming one or more segments of live media content via a data communications network, such as the Internet. Such information can include, for example, a list of URLs that can be used to request chunks of the live media content. The manifest file also includes information for

streaming one or more segments of a media file, distinct from the live media content. The media file can include, for example, an advertisement or other on-demand content with timestamps that differ from the timestamps of the live media content. The manifest file further includes offset information for streaming the one or more segments of the media file. This offset information can be embedded in or otherwise associated with the URLs themselves, and can relay offset information to an entity, such as a DPL, when used by a client. Finally, the manifest file is sent at block 645.

[0070] FIG. 7 is a simplified flow chart illustrating an example method 700 of creating media chunks using offset information included in a request, in accordance with the techniques described above. The method 700 can be implemented, for example, by a DPL 270 and/or other components shown in FIGS. 2 and 3 above. Different blocks may be executed by different components of a system and/or different systems. Such systems can include the computer system, described herein below with regard to FIG. 8.

[0071] At block 705, a request for a segment of media is received. The request includes offset information for the requested segment of media. As indicated above, the offset information can be any information that enables the timestamp of a requested chunk (i.e., segment of media) to be rewritten, allowing for continuity in the timestamps of chunks of the media being streamed.

[0072] Block 715 includes determining one or more files to use for creating the segment of media, based on the request for the segment of media. As described above, chunks may be dynamically crafted to accommodate requests. Because the contents (e.g., start/end times, lengths, etc.) of requested chunks may vary from the content of files (e.g., stored chunks) used to create the requested chunks, more than one files may be used to create the requested chunk.

[0073] At block 725, the one or more files are obtained from a data storage. The files may be stored at a local system, or may be retrieved from a remote system, such as an advertisement server, for example.

[0074] Block 735 includes creating the segment of media from the one or more files by providing at least one timestamp for the segment of media based on the offset information. Rewriting timestamps in this manner allows for continuity in the timestamps of chunks of the

media being streamed when transitioning from one type of media (e.g., live) to another (e.g., on-demand). The segment of media is then sent at block 745.

[0075] Techniques described herein can be used in conjunction with those described in U.S. Patent Application No. 13/339,680 (incorporated by reference above) to provide multi-platform media syndication customization.

[0076] Techniques described herein can be used in conjunction with those described in U.S. Patent Application No. 13/791,789 (incorporated by reference above) to provide dynamic chunking for delivery instances.

[0077] **FIG. 8** illustrates an embodiment of a computer system 800, which may be configured to execute various components described herein using any combination of hardware and/or software. For example, one or more computer systems 800 can be configured to execute components of the CHIMPS 110 shown in systems 200 and 300 of FIGS. 2 and 3, including the API 260 and DPL 270, as well as the encoder 210, end user device 140, and/or other components of the systems described in relation to FIGS. 1-3. FIG. 8 provides a schematic illustration of one embodiment of a computer system 800 that can perform the methods provided by various other embodiments, such as the methods described in relation to FIGS. 5-7. It should be noted that FIG. 8 is meant only to provide a generalized illustration of various components, any or all of which may be utilized as appropriate. FIG. 8, therefore, broadly illustrates how individual system elements may be implemented in a relatively separated or relatively more integrated manner. In addition, it can be noted that components illustrated by FIG. 8 can be localized to a single device and/or distributed among various networked devices, which may be disposed at different physical locations.

[0078] The computer system 800 is shown comprising hardware elements that can be electrically coupled via a bus 805 (or may otherwise be in communication, as appropriate). The hardware elements may include processing unit(s) 810, which can include without limitation one or more general-purpose processors, one or more special-purpose processors (such as digital signal processors, graphics acceleration processors, and/or the like), and/or other processing structure, which can be configured to perform one or more of the methods described herein, including the methods described in relation to FIGS. 5-7, by, for example, executing commands stored in a memory. The computer system 800 also can include one or more input devices 815,

which can include without limitation a mouse, a keyboard, and/or the like; and one or more output devices 820, which can include without limitation a display device, a printer, and/or the like.

[0079] The computer system 800 may further include (and/or be in communication with) one or more non-transitory storage devices 825, which can comprise, without limitation, local and/or network accessible storage. This can include, without limitation, a disk drive, a drive array, an optical storage device, a solid-state storage device, such as a random access memory (“RAM”), and/or a read-only memory (“ROM”), which can be programmable, flash-updateable, and/or the like. Such storage devices may be configured to implement any appropriate data stores, including without limitation, various file systems, database structures, and/or the like.

[0080] The computer system 800 can also include a communications interface 830, which can include wireless and wired communication technologies. Accordingly, the communications interface can include a modem, a network card (wireless or wired), an infrared communication device, a wireless communication device, and/or a chipset (such as a Bluetooth™ device, an IEEE 802.11 device, an IEEE 802.15.4 device, a WiFi device, a WiMax device, cellular communication facilities, UWB interface, etc.), and/or the like. The communications interface 830 can therefore permit the computer system 800 to be exchanged with other devices and components of a network.

[0081] In many embodiments, the computer system 800 will further comprise a working memory 835, which can include a RAM or ROM device, as described above. Software elements, shown as being located within the working memory 835, can include an operating system 840, device drivers, executable libraries, and/or other code, such as one or more application programs 845, which may comprise computer programs provided by various embodiments, and/or may be designed to implement methods, and/or configure systems, provided by other embodiments, as described herein. Merely by way of example, one or more procedures described with respect to the method(s) discussed above, such as the methods described in relation to FIGS. 5-7, might be implemented as code and/or instructions executable by a computer (and/or a processing unit within a computer); in an aspect, then, such code and/or instructions can be used to configure and/or adapt a general purpose computer (or other device) to perform one or more operations in accordance with the described methods.

[0082] A set of these instructions and/or code might be stored on a non-transitory computer-readable storage medium, such as the storage device(s) 825 described above. In some cases, the storage medium might be incorporated within a computer system, such as computer system 800. In other embodiments, the storage medium might be separate from a computer system (e.g., a removable medium, such as an optical disc), and/or provided in an installation package, such that the storage medium can be used to program, configure, and/or adapt a general purpose computer with the instructions/code stored thereon. These instructions might take the form of executable code, which is executable by the computer system 800 and/or might take the form of source and/or installable code, which, upon compilation and/or installation on the computer system 800 (e.g., using any of a variety of generally available compilers, installation programs, compression/decompression utilities, etc.), then takes the form of executable code.

[0083] It will be apparent to those skilled in the art that substantial variations may be made in accordance with specific requirements. For example, customized hardware might also be used, and/or particular elements might be implemented in hardware, software (including portable software, such as applets, etc.), or both. Further, connection to other computing devices such as network input/output devices may be employed.

[0084] As mentioned above, in one aspect, some embodiments may employ a computer system (such as the computer system 800) to perform methods in accordance with various embodiments of the invention. According to a set of embodiments, some or all of the procedures of such methods are performed by the computer system 800 in response to processing unit(s) 810 executing one or more sequences of one or more instructions (which might be incorporated into the operating system 840 and/or other code, such as an application program 845) contained in the working memory 835. Such instructions may be read into the working memory 835 from another computer-readable medium, such as one or more of the storage device(s) 825. Merely by way of example, execution of the sequences of instructions contained in the working memory 835 might cause the processing unit(s) 810 to perform one or more procedures of the methods described herein. Additionally or alternatively, portions of the methods described herein may be executed through specialized hardware.

[0085] It should be noted that the methods, systems, and devices discussed above are intended merely to be examples. It must be stressed that various embodiments may omit, substitute, or

add various procedures or components as appropriate. For instance, it should be appreciated that, in alternative embodiments, the methods may be performed in an order different from that described, and that various steps may be added, omitted, or combined. Also, features described with respect to certain embodiments may be combined in various other embodiments. Different
5 aspects and elements of the embodiments may be combined in a similar manner. Also, it should be emphasized that technology evolves and, thus, many of the elements are examples and should not be interpreted to limit the scope of the invention.

[0086] Terms, “and” and “or” as used herein, may include a variety of meanings that also is expected to depend at least in part upon the context in which such terms are used. Typically,
10 “or” if used to associate a list, such as A, B, or C, is intended to mean A, B, and C, here used in the inclusive sense, as well as A, B, or C, here used in the exclusive sense. In addition, the term “one or more” as used herein may be used to describe any feature, structure, or characteristic in the singular or may be used to describe some combination of features, structures, or
15 characteristics. However, it should be noted that this is merely an illustrative example and claimed subject matter is not limited to this example. Furthermore, the term “at least one of” if used to associate a list, such as A, B, or C, can be interpreted to mean any combination of A, B, and/or C, such as A, AB, AA, AAB, AABBBCCC, etc.

[0087] Having described several embodiments, it will be recognized by those of skill in the art that various modifications, alternative constructions, and equivalents may be used without
20 departing from the spirit of the invention. For example, the above elements may merely be a component of a larger system, wherein other rules may take precedence over or otherwise modify the application of the invention. Also, a number of steps may be undertaken before, during, or after the above elements are considered. Accordingly, the above description should not be taken as limiting the scope of the invention.

25

WHAT IS CLAIMED IS:

1. A method of providing media streaming via a data communications network, the method comprising:
 - receiving a stream of data representing first media source;
 - obtaining timing information of the stream of data;
 - sending, via the data communications network, a manifest.

ABSTRACT OF THE DISCLOSURE

Timestamps for streams of media that transition from one media source to another (such as from live content to on-demand content, and vice versa) can be rewritten by a server to help ensure error-free streaming by the client. Embodiments can coordinate the creation of a client manifest with the dynamic creation of a requested segment of media (i.e., “chunk”) to determine how to rewrite timestamps of requested chunks such that they are continuous through the transition.

68856420V.1

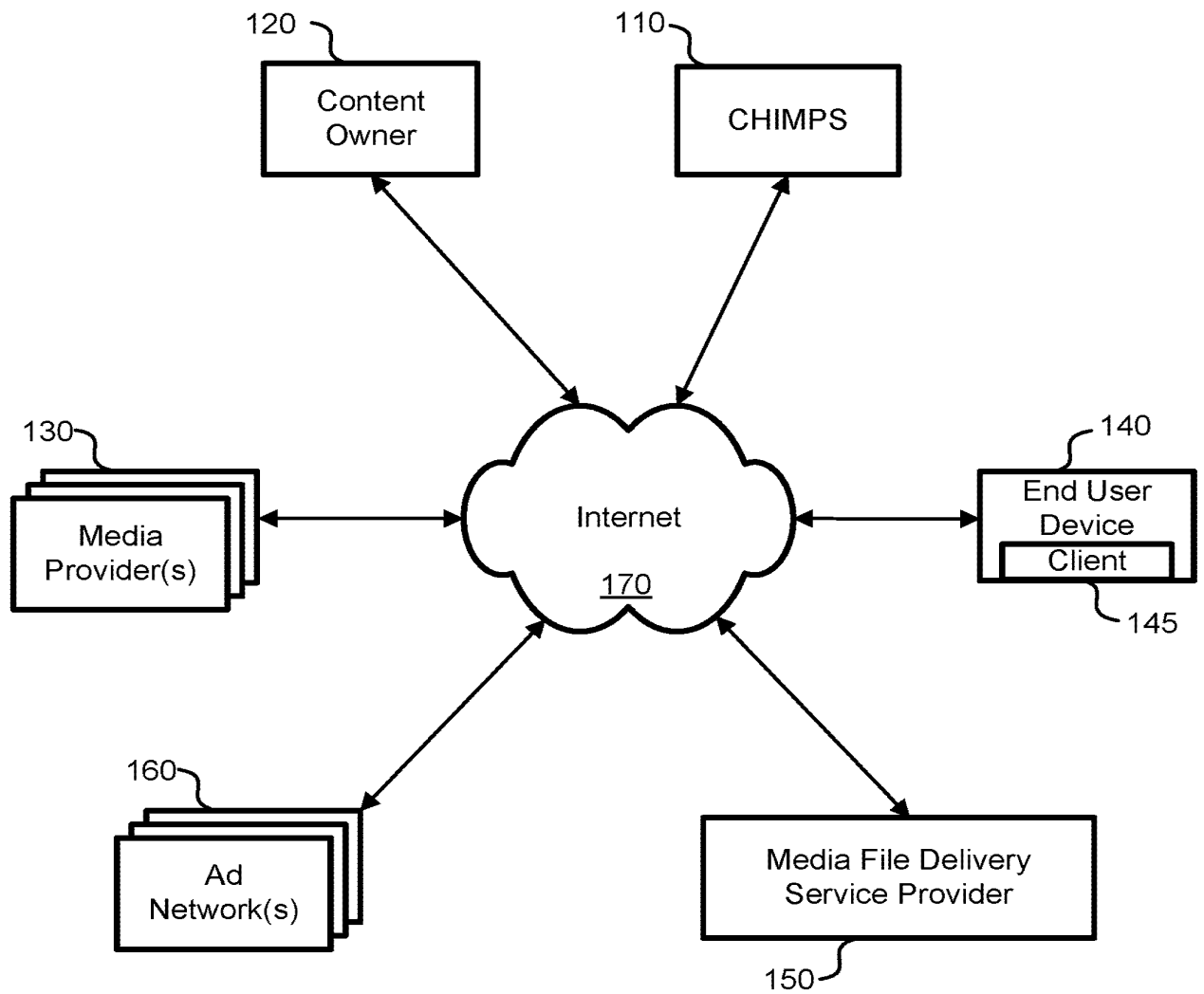


FIG. 1

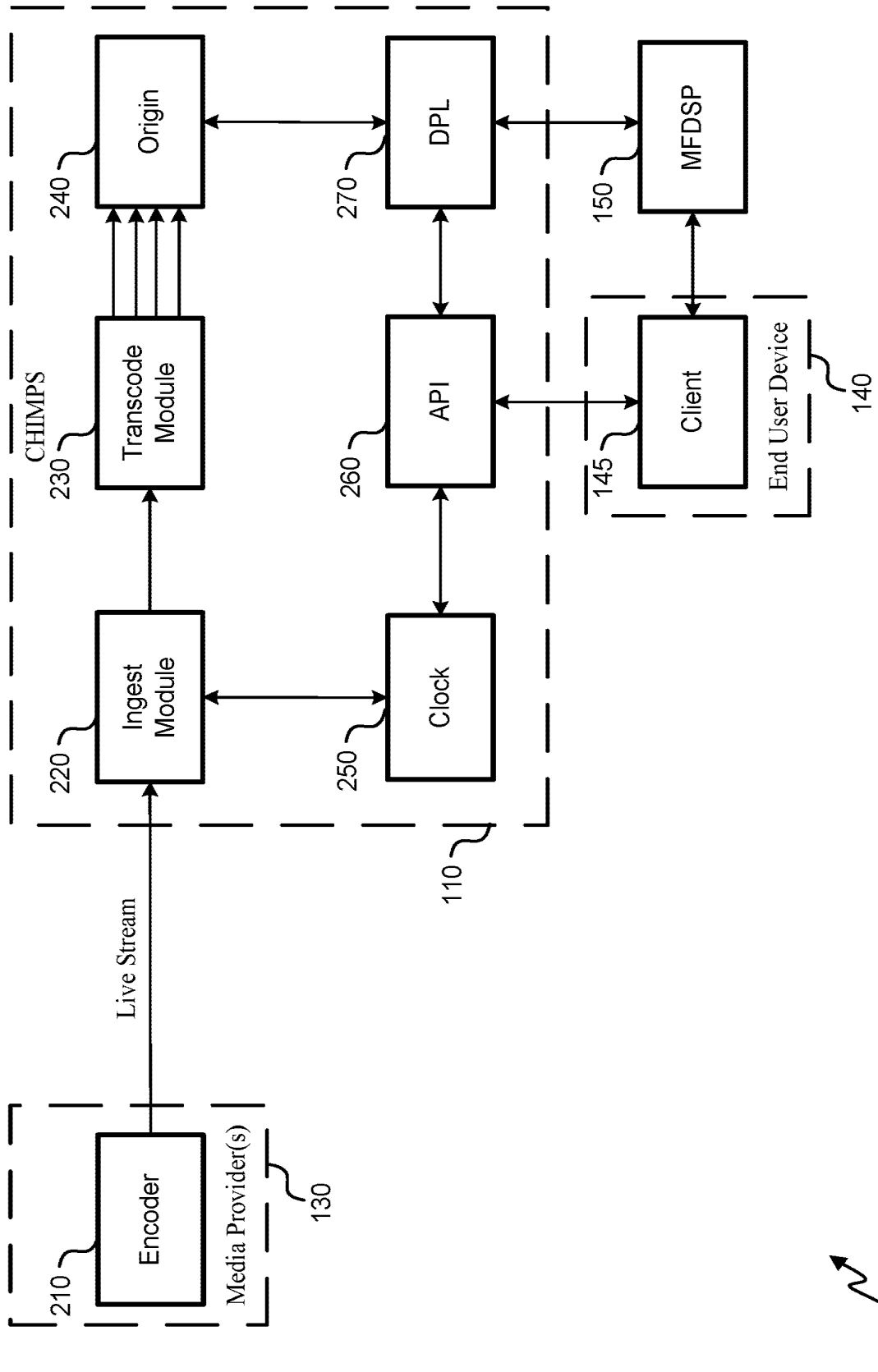


FIG. 2

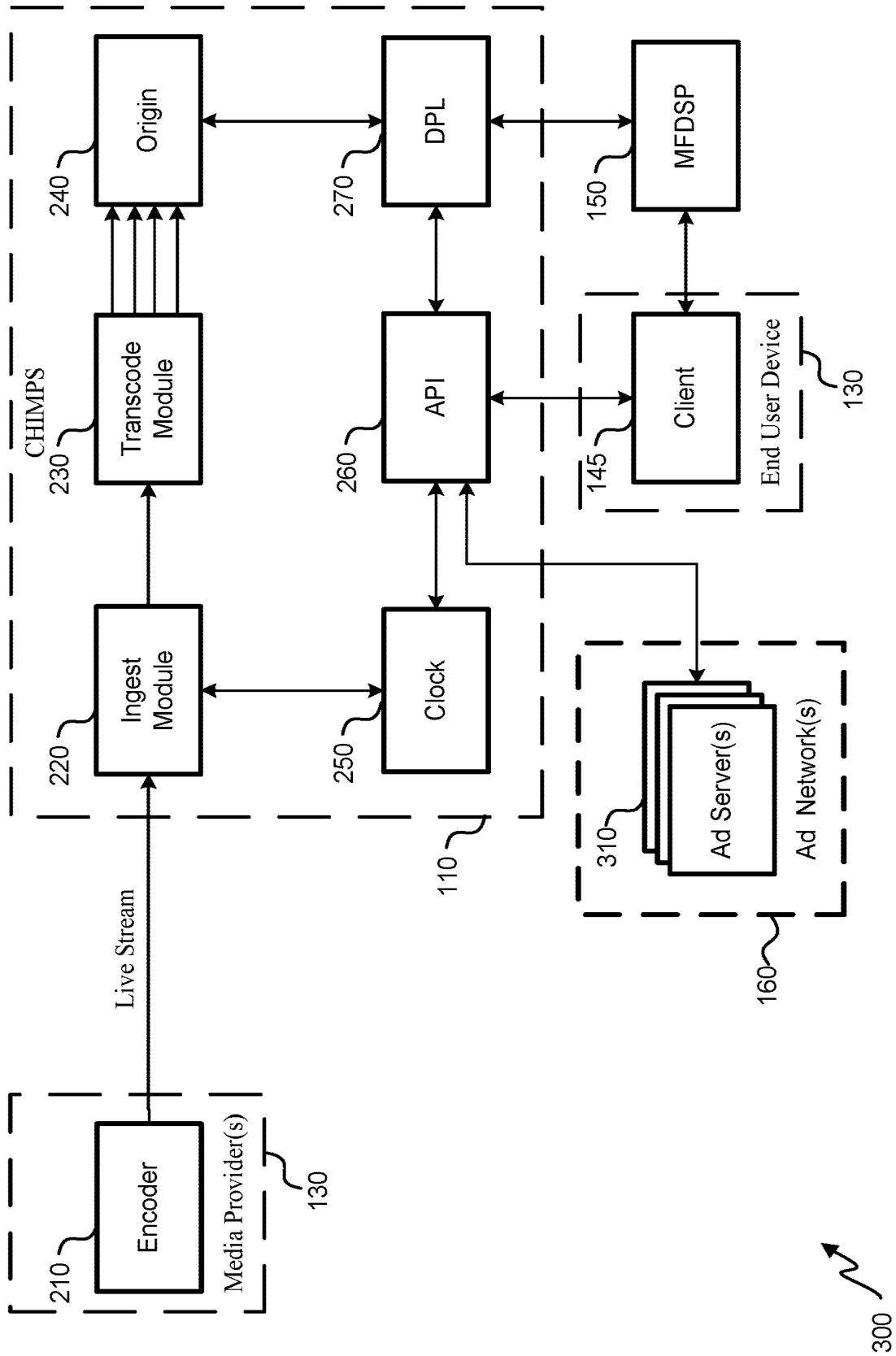


FIG. 3

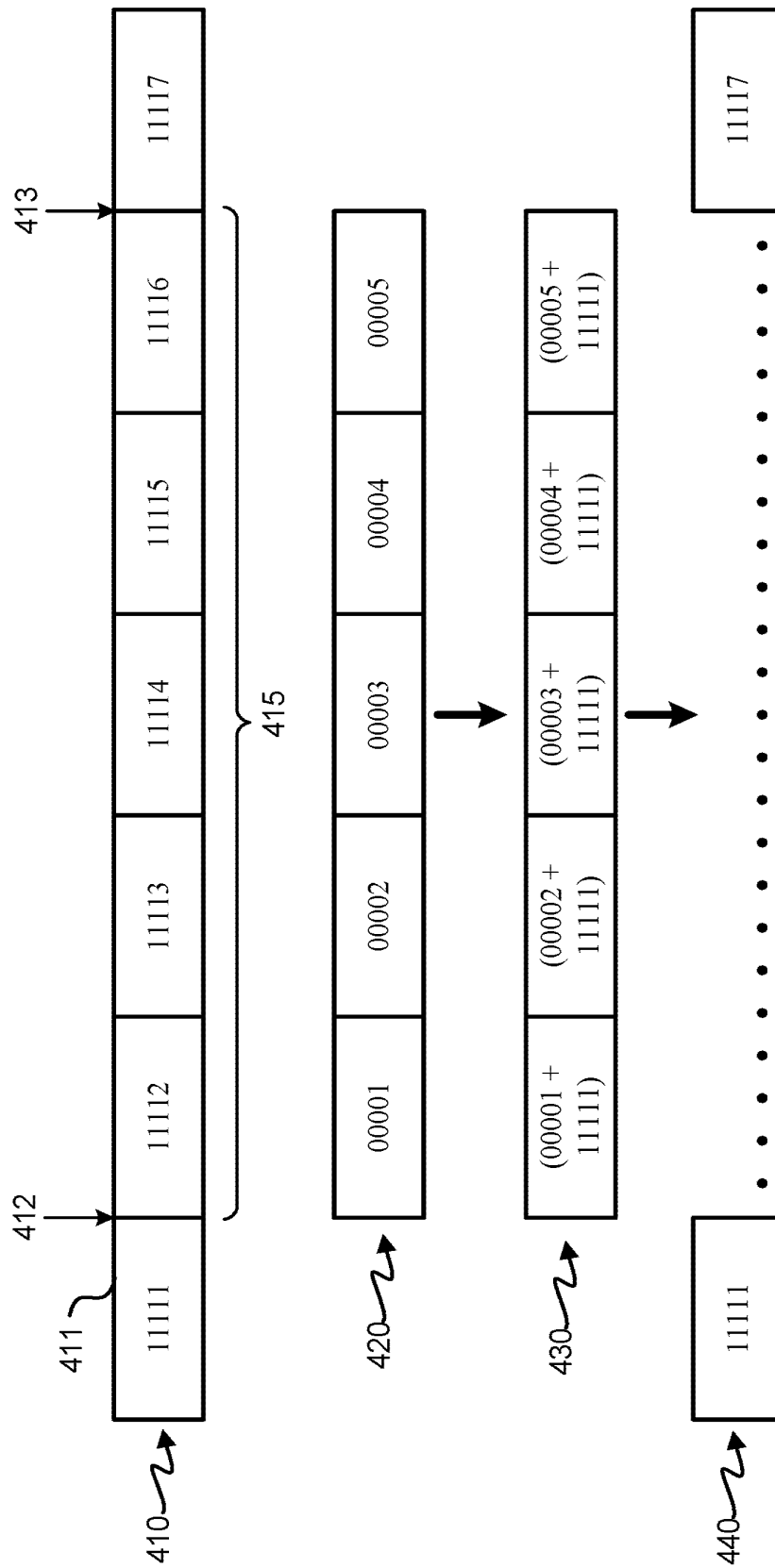


FIG. 4

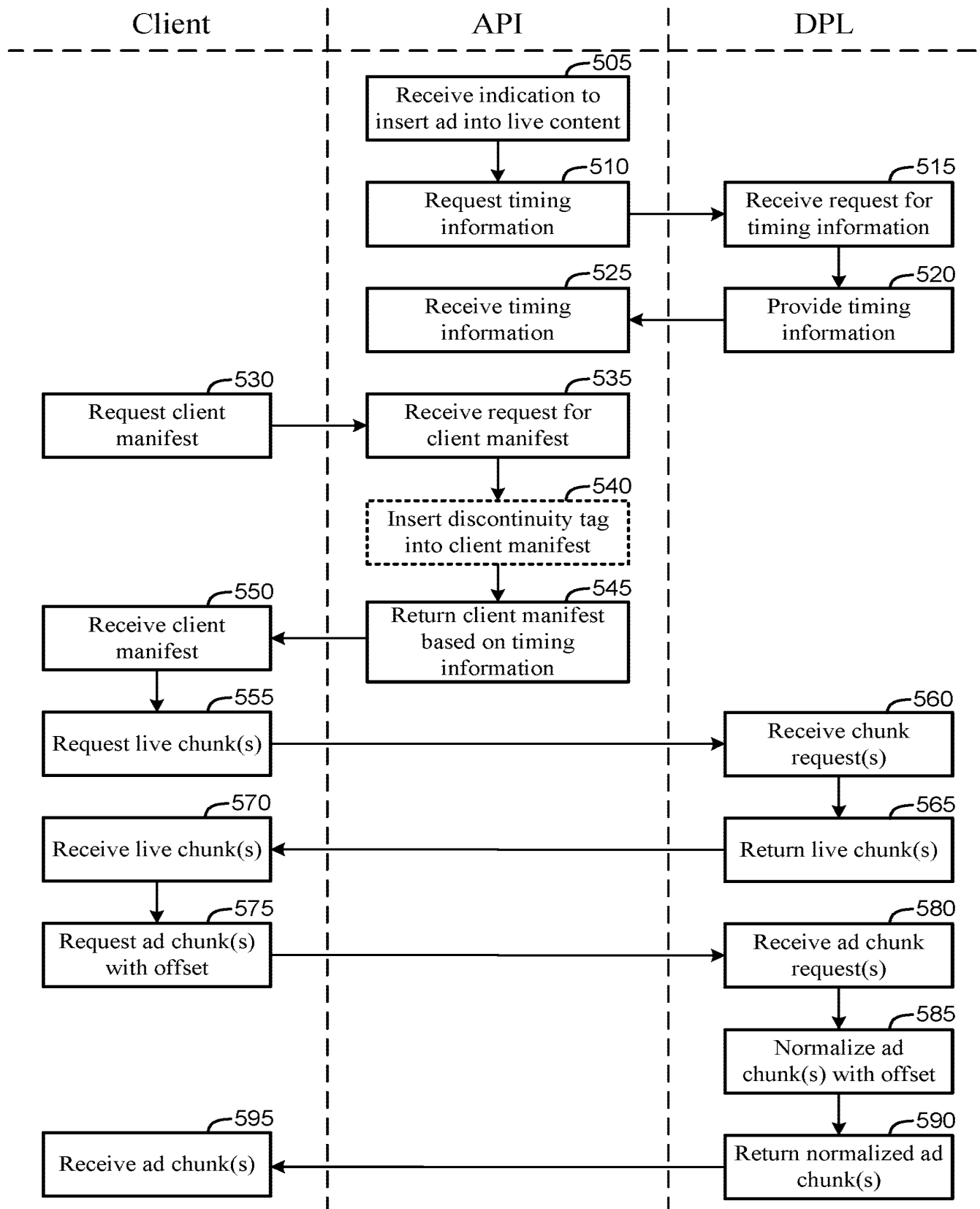


FIG. 5

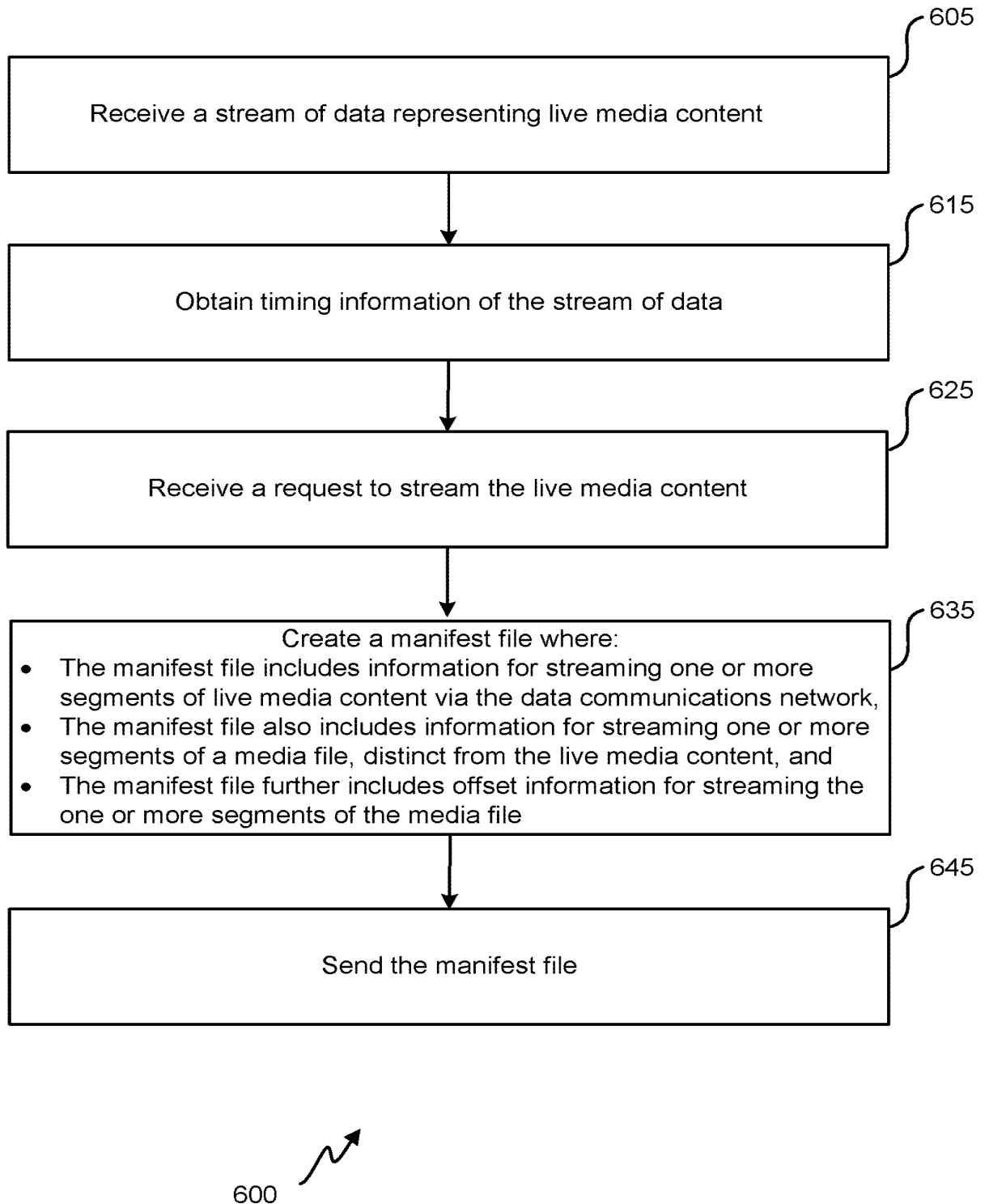


FIG. 6

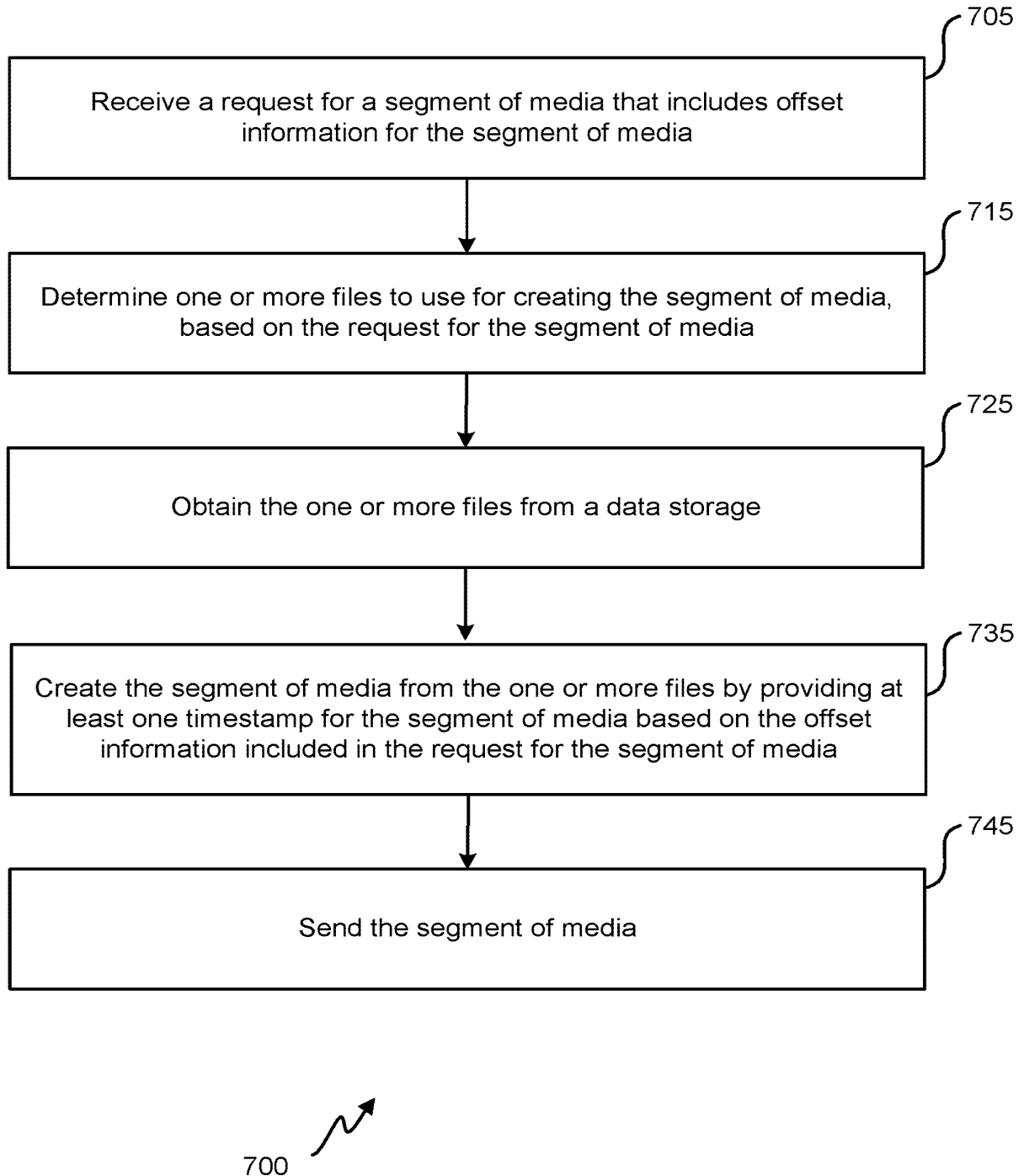


FIG. 7

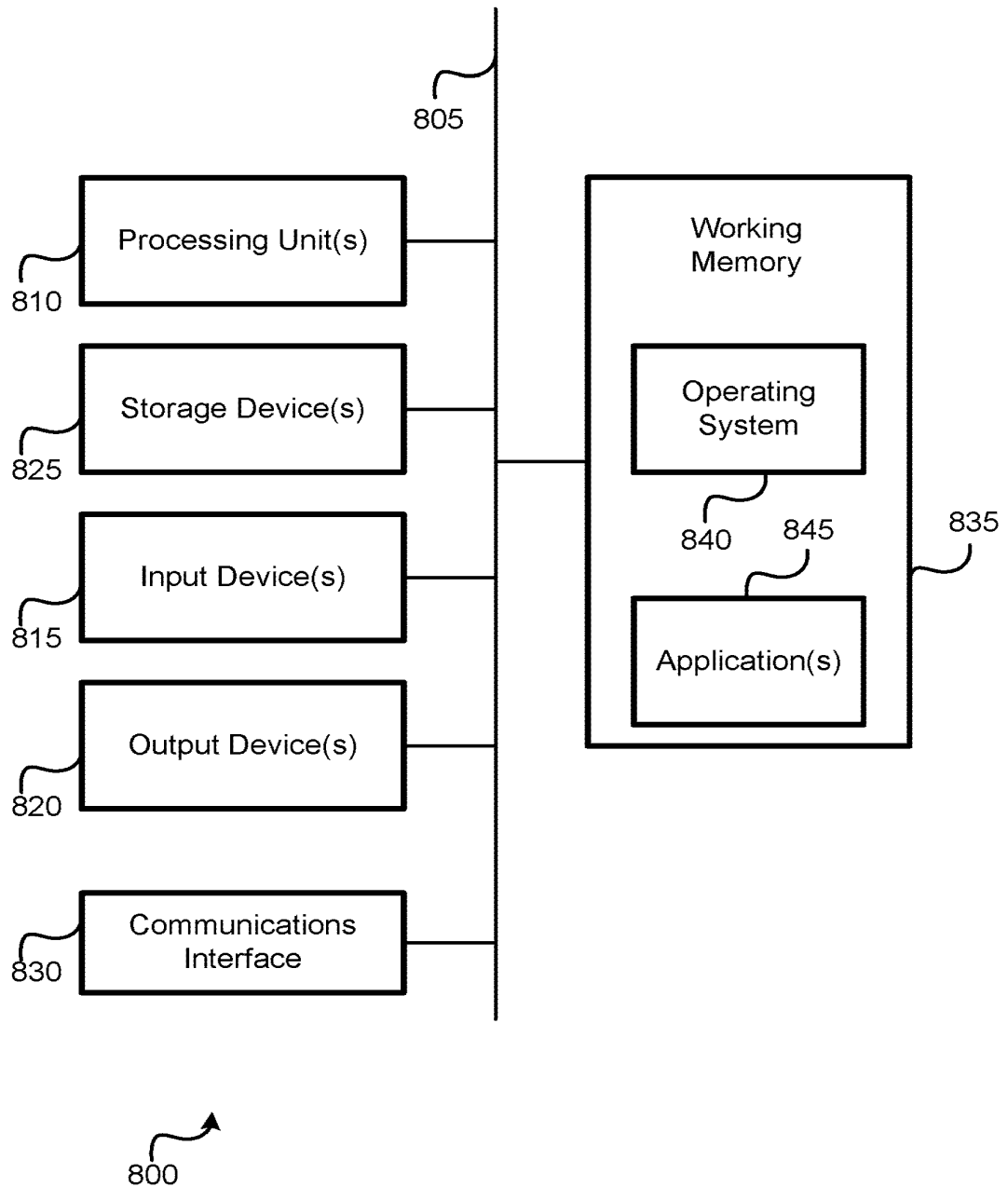


FIG. 8

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA/82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	To Be Assigned
Filing Date	October 28, 2016
First Named Inventor	Albert John McGowan
Title	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE
Art Unit	To Be Assigned
Examiner Name	To Be Assigned
Attorney Docket Number	092853-0961827 (004130US)

SIGNATURE of Applicant or Patent Practitioner

Signature	/Scott L. McMillan/	Date (Optional)	
Name	Scott L. McMillan	Registration Number	62,079
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)			

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

☐

*Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

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

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(c).

I hereby appoint:



Practitioners associated with Customer Number:

20350

OR



Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number

Name	Registration Number

As attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignments documents attached to this form in accordance with 37 CFR 3.73(c).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(c) to:



The address associated with Customer Number:

20350

OR

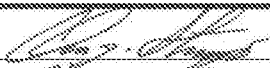
☐	Firm or Individual Name			
	Address			
	City	State	Zip	
	Country			
	Telephone	Email		

Assignee Name and Address: BRIGHTCOVE INC.
290 CONGRESS STREET
4TH FLOOR
BOSTON, MASSACHUSETTS 02210

A copy of this form, together with a statement under 37 CFR 3.73(c) (Form PTO/AIA/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(c) may be completed by one of the practitioners appointed in this form, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee

Signature		Date	2/3/2015
Name	Christopher Stynes	Telephone	
Title	VP, CAD		

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

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

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Acknowledgement Receipt

EFS ID:	27359773
Application Number:	15337865
International Application Number:	
Confirmation Number:	8780
Title of Invention:	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE
First Named Inventor/Applicant Name:	Albert John McGowan
Customer Number:	20350
Filer:	Scott Lee McMillan/Lelia Breeden
Filer Authorized By:	Scott Lee McMillan
Attorney Docket Number:	092853-0961827 (004130US)
Receipt Date:	28-OCT-2016
Filing Date:	
Time Stamp:	16:56:39
Application Type:	Utility under 35 USC 111(a)

Payment information:

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

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	2016_10_28_ADS_092853_102 8162_004130US.pdf	1510601 ded9bde154bf03dc6e7ec7a694e6df70662 c0622	no	7

Warnings:

Information:					
2		2016_10_28_Spec_092853_1028162_004130US.pdf	138518	yes	26
			ecce538a43999a2cc159610988b0234b27321208		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Transmittal of New Application		1	1	
	Specification		2	24	
	Claims		25	25	
	Abstract		26	26	
Warnings:					
Information:					
3	Drawings-only black and white line drawings	2016_10_28_Drawings_092853_1028162_004130US.pdf	107886	no	8
			3763410fe5d4a6cbe7736e12442ed12b5538ff30		
Warnings:					
Information:					
4	Power of Attorney	2016_10_28_POA_092853_0961827_004130US.pdf	1421633	no	3
			f38bac21f8a769e78b20b4439eb7e5485f5323df		
Warnings:					
Information:					
Total Files Size (in bytes):			3178638		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

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

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

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

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	092853-0961827 (004130US)
		Application Number	
Title of Invention	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE		
The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76. This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.			

Secrecy Order 37 CFR 5.2

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

Inventor Information:

Inventor	1				Remove
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Albert	John	McGowan		
Residence Information (Select One) • US Residency Non US Residency Active US Military Service					
City	Phoenix	State/Province	AZ	Country of Residence	US
Mailing Address of Inventor:					
Address 1	4201 E. Camelback Road, No. 101				
Address 2					
City	Phoenix	State/Province	AZ		
Postal Code	85018	Country	US		
Inventor	2				Remove
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Michael	M.	Gordon		
Residence Information (Select One) • US Residency Non US Residency Active US Military Service					
City	Paradise Valley	State/Province	AZ	Country of Residence	US
Mailing Address of Inventor:					
Address 1	6500 E. Cactus Wren Road				
Address 2					
City	Paradise Valley	State/Province	AZ		
Postal Code	85253	Country	US		
All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the Add button.					
Add					

Correspondence Information:

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

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

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	092853-0961827 (004130US)
		Application Number	
Title of Invention	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE		

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

Customer Number	20350		
Email Address		<input type="button" value="Add Email"/>	<input type="button" value="Remove Email"/>

Application Information:

Title of the Invention	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE		
Attorney Docket Number	092853-0961827 (004130US)	Small Entity Status Claimed	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	8	Suggested Figure for Publication (if any)	

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

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

Representative Information:

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

Please Select One:	☒ Customer Number	☐ US Patent Practitioner	☐ Limited Recognition (37 CFR 11.9)
Customer Number	20350		

Domestic Benefit/National Stage Information:

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

Prior Application Status	Pending	<input type="button" value="Remove"/>	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	14918368	2015-10-20
Prior Application Status	Patented	<input type="button" value="Remove"/>	

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	092853-0961827 (004130US)
		Application Number	
Title of Invention	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE		

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
14918368	Continuation of	14086801	2013-11-21	9197688	2015-11-21
Prior Application Status		Expired	Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
14086801	non provisional of	61884709	2013-09-30		
Prior Application Status		Pending	Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
	Continuation in part of	13339680	2011-12-29		
Prior Application Status		Pending	Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
13339680	Continuation in part of	13245372	2011-09-26		
Prior Application Status		Pending	Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
	Continuation in part of	13791789	2013-03-08		
Prior Application Status		Patented	Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13791789	Continuation in part of	13624029	2012-09-21	8645504	2014-02-04
Prior Application Status		Patented	Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13624029	Continuation of	13430081	2012-03-26	8301733	2012-10-30
Prior Application Status		Patented	Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13430081	Continuation in part of	12976883	2010-12-22	8145782	2012-03-27
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the Add button.					

Foreign Priority Information:

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

Remove			
Application Number	Country ⁱ	Filing Date (YYYY-MM-DD)	Access Code ⁱ (if applicable)
2010202741	AU	2010-06-30	

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	092853-0961827 (004130US)
		Application Number	
Title of Invention	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE		
Additional Foreign Priority Data may be generated within this form by selecting the Add button. <input type="button" value="Add"/>			

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

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

Authorization to Permit Access:

☐ Authorization to Permit Access to the Instant Application by the Participating Offices
<p>If checked, the undersigned hereby grants the USPTO authority to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application.</p> <p>In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application.</p> <p>In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date of filing this Authorization.</p>

Applicant Information:

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

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

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	092853-0961827 (004130US)
		Application Number	
Title of Invention	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE		

Applicant	1	Remove
<p>If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.</p>		
Clear		
☒ Assignee	Legal Representative under 35 U.S.C. 117	Joint Inventor
Person to whom the inventor is obligated to assign.		Person who shows sufficient proprietary interest
If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:		

<div style="border: 1px solid black; height: 20px; width: 100%;"></div>			
Name of the Deceased or Legally Incapacitated Inventor :			
If the Applicant is an Organization check here. ☒			
Organization Name	BRIGHTCOVE INC.		
Mailing Address Information For Applicant:			
Address 1	290 CONGRESS STREET		
Address 2			
City	Boston	State/Province	MA
Country	US	Postal Code	02210
Phone Number		Fax Number	
Email Address			
Additional Applicant Data may be generated within this form by selecting the Add button. Add			

Non-Applicant Assignee Information:

<p>Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.</p>			
Assignee	1	Remove	
<p>Complete this section only if non-applicant assignee information is desired to be included on the patent application publication in accordance with 37 CFR 1.215(b). Do not include in this section an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest), as the patent application publication will include the name of the applicant(s).</p>			
Remove			
If the Assignee is an Organization check here. ☐			

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

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	092853-0961827 (004130US)
		Application Number	
Title of Invention	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE		

Prefix	Given Name	Middle Name	Family Name	Suffix

Mailing Address Information For Non-Applicant Assignee:

Address 1				
Address 2				
City		State/Province		
Country		Postal Code		
Phone Number		Fax Number		
Email Address				

Additional Assignee Data may be generated within this form by selecting the Add button.

Add

Signature:

Remove

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

Signature	/Scott L. McMillan/		Date (YYYY-MM-DD)	2016-10-28
First Name	Scott	Last Name	McMillan	Registration Number
			62079	

Additional Signature may be generated within this form by selecting the Add button.

Add

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 15/337,865			
APPLICATION AS FILED - PART I									
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA		RATE(\$)	FEE(\$)	RATE(\$)	FEE(\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A	70	N/A			
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A		N/A	300	N/A			
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A	360	N/A			
TOTAL CLAIMS (37 CFR 1.16(i))	1 minus 20 =	*		x 40 =	0.00	OR			
INDEPENDENT CLAIMS (37 CFR 1.16(h))	1 minus 3 =	*		x 210 =	0.00				
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).				0.00				
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))					0.00				
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL	730	TOTAL			
APPLICATION AS AMENDED - PART II									
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)		
	Total (37 CFR 1.16(i))	*	Minus **	=	x =	OR	x =		
	Independent (37 CFR 1.16(h))	*	Minus ***	=	x =	OR	x =		
	Application Size Fee (37 CFR 1.16(s))					OR			
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					OR			
				TOTAL ADD'L FEE		TOTAL ADD'L FEE			
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)		
	Total (37 CFR 1.16(i))	*	Minus **	=	x =	OR	x =		
	Independent (37 CFR 1.16(h))	*	Minus ***	=	x =	OR	x =		
	Application Size Fee (37 CFR 1.16(s))					OR			
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					OR			
				TOTAL ADD'L FEE		TOTAL ADD'L FEE			
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.									



UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY DOCKET NO	TOT CLAIMS	IND CLAIMS
15/337,865	10/28/2016	2447	0.00	092853-1028162 (004130US)	1	1

CONFIRMATION NO. 8780

FILING RECEIPT



0000000087085039

20350
KILPATRICK TOWNSEND & STOCKTON LLP
Mailstop: IP Docketing - 22
1100 Peachtree Street
Suite 2800
Atlanta, GA 30309

Date Mailed: 11/09/2016

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

Inventor(s)

Albert John McGowan, Phoenix, AZ;
Michael M. Gordon, Paradise Valley, AZ;

Applicant(s)

BRIGHTCOVE INC., Boston, MA;

Power of Attorney: The patent practitioners associated with Customer Number 20350

Domestic Priority data as claimed by applicant

This application is a CIP of 14/918,368 10/20/2015 PAT 9485293
which is a CON of 14/086,801 11/21/2013 PAT 9197688
which claims benefit of 61/884,709 09/30/2013
This application 15/337,865
is a CIP of 13/339,680 12/29/2011
which is a CIP of 13/245,372 09/26/2011 ABN
This application 15/337,865
is a CIP of 13/791,789 03/08/2013
which is a CIP of 13/624,029 09/21/2012 PAT 8645504
which is a CON of 13/430,081 03/26/2012 PAT 8301733
which is a CIP of 12/976,883 12/22/2010 PAT 8145782

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.)

AUSTRALIA 2010202741 06/30/2010 No Access Code Provided

Permission to Access Application via Priority Document Exchange: No

Permission to Access Search Results: No

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

If Required, Foreign Filing License Granted: 11/08/2016

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 15/337,865**

Projected Publication Date: To Be Determined - pending completion of Missing Parts

Non-Publication Request: No

Early Publication Request: No

**** SMALL ENTITY ****

Title

DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION
PROGRAMMING INTERFACE

Preliminary Class

709

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

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

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

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

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

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

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

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

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

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

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

NOT GRANTED

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

SelectUSA

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

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



UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/337,865	10/28/2016	Albert John McGowan	092853-1028162 (004130US)

CONFIRMATION NO. 8780

FORMALITIES LETTER



0000000087065040

20350
KILPATRICK TOWNSEND & STOCKTON LLP
Mailstop: IP Docketing - 22
1100 Peachtree Street
Suite 2800
Atlanta, GA 30309

Date Mailed: 11/09/2016

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
- The application search fee must be submitted.
- The application examination fee must be submitted.
- Surcharge as set forth in 37 CFR 1.16(f) must be submitted.

The surcharge is due for any one of:

- late submission of the basic filing fee, search fee, or examination fee,
- late submission of inventor's oath or declaration,
- filing an application that does not contain at least one claim on filing, or
- submission of an application filed by reference to a previously filed application.

SUMMARY OF FEES DUE:

The fee(s) required within **TWO MONTHS** from the date of this Notice to avoid abandonment is/are itemized below. Small entity discount is in effect. If applicant is qualified for micro entity status, an acceptable Certification of Micro Entity Status must be submitted to establish micro entity status. (See 37 CFR 1.29 and forms PTO/SB/15A and 15B.)

- \$ **70** basic filing fee.
- \$ **70** surcharge.
- \$ **300** search fee.
- \$ **360** examination fee.
- \$(**0**) previous unapplied payment amount.
- \$ **800** TOTAL FEE BALANCE DUE.

Items Required To Avoid Processing Delays:

Applicant is notified that the above-identified application contains the deficiencies noted below. No period for reply is set forth in this notice for correction of these deficiencies. However, if a deficiency relates to the inventor's

oath or declaration, the applicant must file an oath or declaration in compliance with 37 CFR 1.63, or a substitute statement in compliance with 37 CFR 1.64, executed by or with respect to each actual inventor no later than the expiration of the time period set in the "Notice of Allowability" to avoid abandonment. See 37 CFR 1.53(f).

- A properly executed inventor's oath or declaration has not been received for the following inventor(s):
Albert John McGowan
Michael M. Gordon

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice".
<https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html>

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at <http://www.uspto.gov/ebc>.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at (571) 272-4000 or (571) 272-4200 or 1-888-786-0101.

/hachristian/

I hereby certify that this correspondence is being filed via
EFS-Web with the United States Patent and Trademark Office
on November 30, 2016.

PATENT
Attorney Docket No.: 092853-1028162 (004130US)

KILPATRICK TOWNSEND & STOCKTON LLP

By: /Lelia Breeden/
Lelia Breeden

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Albert John McGowan et al.

Application No.: 15/337,865

Filed: October 28, 2016

For: DYNAMIC CHUNK MANIPULATION
FOR STREAMING MIXED MEDIA:
APPLICATION PROGRAMMING
INTERFACE

Customer No.: 20350

Confirmation No.: 8780

Examiner: Not Yet Assigned

Technology Center/Art Unit: 2447

PRELIMINARY AMENDMENT

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

Commissioner:

Prior to examination of the above-referenced application, please enter the
following amendments and remarks:

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

Remarks/Arguments begin on page 3 of this paper.

Amendments to the Specification:

Please replace the title as filed on page 1 of the original specification with the following amended title:

DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED ~~LIVE AND ON-~~
~~DEMAND~~-MEDIA: APPLICATION PROGRAMMING INTERFACE

REMARKS/ARGUMENTS

Upon entry of the present amendment, claim 1 will be pending in this application. No claims have been amended, canceled, or added. No new matter is added.

CONCLUSION

In view of the foregoing, Applicant believes all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

Except for the issue fees payable under 37 C.F.R. § 1.18, the Director is authorized to charge any additional fees during pendency of this application, including any required extension of time fees, or credit any overpayment to Deposit Account Number 20-1430. This paragraph is intended to be a constructive petition for extension of time in accordance with 37 C.F.R. § 1.136(a)(3).

If the Examiner believes a telephone conference would expedite prosecution of this application, please contact the undersigned at (303) 571-4000 or SMcMillan@KilpatrickTownsend.com.

Respectfully submitted,

/Scott L. McMillan/

Scott L. McMillan
Registration No. 62,079

KILPATRICK TOWNSEND & STOCKTON LLP

Attachments

LB

68930708V.1

Electronic Acknowledgement Receipt

EFS ID:	27648749
Application Number:	15337865
International Application Number:	
Confirmation Number:	8780
Title of Invention:	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE
First Named Inventor/Applicant Name:	Albert John McGowan
Customer Number:	20350
Filer:	Scott Lee McMillan/Lelia Breeden
Filer Authorized By:	Scott Lee McMillan
Attorney Docket Number:	092853-1028162 (004130US)
Receipt Date:	30-NOV-2016
Filing Date:	28-OCT-2016
Time Stamp:	12:45:28
Application Type:	Utility under 35 USC 111(a)

Payment information:

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

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		2016_11_30_Preliminary_Ame ndment_092853_1028162_004 130US.pdf	76826 0756d01c9963518457c3e1367a4fca3d5f2a 71e5	yes	3

Multipart Description/PDF files in .zip description			
Document Description		Start	End
Preliminary Amendment		1	1
Specification		2	2
Applicant Arguments/Remarks Made in an Amendment		3	3

Warnings:

Information:

Total Files Size (in bytes):	76826
-------------------------------------	-------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 15/337,865			
APPLICATION AS FILED - PART I									
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA		RATE(\$)	FEE(\$)	RATE(\$)	FEE(\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A	70	N/A			
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A		N/A	300	N/A			
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A	360	N/A			
TOTAL CLAIMS (37 CFR 1.16(i))	1	minus 20 =	*	x 40 =	0.00	OR			
INDEPENDENT CLAIMS (37 CFR 1.16(h))	1	minus 3 =	*	x 210 =	0.00				
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).				0.00				
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))					0.00				
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL	730	TOTAL			
APPLICATION AS AMENDED - PART II									
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =	OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =	OR	x =	
	Application Size Fee (37 CFR 1.16(s))						OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =	OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =	OR	x =	
	Application Size Fee (37 CFR 1.16(s))						OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.									



UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/337,865	10/28/2016	Albert John McGowan	092853-1028162 (004130US)

CONFIRMATION NO. 8780

FORMALITIES LETTER



20350
KILPATRICK TOWNSEND & STOCKTON LLP
Mailstop: IP Docketing - 22
1100 Peachtree Street
Suite 2800
Atlanta, GA 30309

Date Mailed: 12/02/2016

NOTICE OF INCOMPLETE REPLY (NONPROVISIONAL)

Filing Date Granted

The U.S. Patent and Trademark Office has received your reply on to the Notice to File Missing Parts (Notice) mailed 11/09/2016 and it has been entered into the nonprovisional application. The reply, however, does not include the following items required in the Notice. A complete reply must be timely filed to prevent ABANDONMENT of the above-identified application. Replies should be mailed to: Mail Stop Missing Parts, Commissioner for Patents, P.O. Box 1450, Alexandria VA 22313-1450.

Applicant is given **TWO MONTHS** from the date of the Notice to File Missing Parts (Notice) mailed 11/09/2016 within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
- The application search fee was not received.
- The application examination fee was not received.
- Surcharge as set forth in 37 CFR 1.16(f) was not received.

The surcharge is due for any one of:

- late submission of the basic filing fee, search fee, or examination fee,
- late submission of inventor's oath or declaration,
- filing an application that does not contain at least one claim on filing, or
- submission of an application filed by reference to a previously filed application.

SUMMARY OF FEES DUE:

The fee(s) required within **TWO MONTHS** from the date of the Notice to avoid abandonment is/are itemized below. Small entity discount is in effect. If applicant is qualified for micro entity status, an acceptable Certification of Micro Entity Status must be submitted to establish micro entity status. (See 37 CFR 1.29 and forms PTO/SB/15A and 15B.)

- \$ **70** basic filing fee.
- \$ **70** surcharge.
- \$ **300** search fee.
- \$ **360** examination fee.
- \$(**0**) previous unapplied payment amount.
- \$ **800** TOTAL FEE BALANCE DUE.

Items Required To Avoid Processing Delays:

Applicant is notified that the above-identified application contains the deficiencies noted below. No period for reply is set forth in this notice for correction of these deficiencies. However, if a deficiency relates to the inventor's oath or declaration, the applicant must file an oath or declaration in compliance with 37 CFR 1.63, or a substitute statement in compliance with 37 CFR 1.64, executed by or with respect to each actual inventor no later than the expiration of the time period set in the "Notice of Allowability" to avoid abandonment. See 37 CFR 1.53(f).

- A properly executed inventor's oath or declaration has not been received for the following inventor(s):
Albert John McGowan
Michael M. Gordon

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice".
<https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html>

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at <http://www.uspto.gov/ebc>.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at (571) 272-4000 or (571) 272-4200 or 1-888-786-0101.

/ctuazon/

I hereby certify that this correspondence is being filed via
EFS-Web with the United States Patent and Trademark Office
on January 3, 2017.

PATENT
Attorney Docket No.: 092853-1028162 (004130US)

KILPATRICK TOWNSEND & STOCKTON LLP

By: /Lelia Breeden/
Lelia Breeden

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Albert John McGowan et al.

Application No.: 15/337,865

Filed: October 28, 2016

For: DYNAMIC CHUNK MANIPULATION
FOR STREAMING MIXED MEDIA:

APPLICATION PROGRAMMING

INTERFACE

Customer No.: 20350

Confirmation No.: 8780

Examiner: Not Yet Assigned

Technology Center/Art Unit: 2447

PRELIMINARY AMENDMENT

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

Commissioner:

Prior to examination of the above-referenced application, please enter the
following amendments and remarks:

Amendments to the Claims are reflected in the listing of claims which begins on page 1 of this
paper.

Amendments to the Specification begin on page 8 of this paper.

Remarks/Arguments begin on page 8 of this paper.

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (Canceled)
2. (New) A server for streaming media via a data communications network, the server comprising:
 - a network interface for communicating with the data communications network;
 - a memory; and
 - a processor communicatively coupled with the memory and the network interface, the processor further configured to cause the server to:
 - receive, via the network interface, a universal source locator (URL) in a first request from a first client, wherein the URL includes information indicative of a requested media source;
 - determine an identity of the first client;
 - determine a first device type based on the identity of the first client;
 - based at least in part on the determination of the first device type, generate a first manifest having information for streaming one or more segments of the requested media source in accordance with a first chunking protocol via the data communications network;
 - sending, via the network interface, the first manifest to the first client;
 - receive, via the network interface, the URL in a second request from a second client;
 - determine an identity of the second client, the identity of the second client different from the identity of the first client;
 - determine a second device type based on the identity of the second client, the second device type different from the first device type;

generate a second manifest having information for streaming the one or more segments of the requested media source in accordance with a second chunking protocol via the data communications network, wherein content of the second manifest is different from content of the first manifest; and
send, via the network interface, the second manifest to the second client.

3. (New) The server for streaming media via the data communications network as recited in claim 2, wherein the processor is configured to cause the server to determine a network type associated with the identity of the first client, wherein either or both of a format or content of the first manifest further is based, at least in part, on the network type.

4. (New) The server for streaming media via the data communications network as recited in claim 3, wherein the processor is configured to cause the server to include, in the first manifest, information indicative of a set of bitrates for streaming the requested media source, wherein the set of bitrates is based, at least in part, on either or both of the identity of the first client and the network type.

5. (New) The server for streaming media via the data communications network as recited in claim 4, wherein the processor is configured to cause the server to include, in the first manifest, information indicative of an initial bitrate of the set of bitrates for streaming the requested media source, wherein the initial bitrate is based, at least in part, on either or both of the identity of the first client and the network type.

6. (New) The server for streaming media via the data communications network as recited in claim 2, wherein the processor is configured to cause the server to:
receive, via the network interface, data regarding the requested media source, wherein the data includes information indicative of at least one point, during playback of the requested media source, at which at least one other media source is to be shown;
receive, via the network interface, information regarding the at least one other media source; and

configure either or both of the first manifest and the second manifest such that the at least one other media source is shown at the at least one point during the playback of the requested media source.

7. (New) The server for streaming media via the data communications network as recited in claim 6, wherein the processor is configured to cause the server to configure either or both of the first manifest and the second manifest to include at least one beaconing URL which, when used, is indicative of playback of the at least one other media source.

8. (New) The server for streaming media via the data communications network as recited in claim 2, wherein the processor is configured to cause the server to include in either or both of the first manifest and the second manifest, at least one beaconing URL which, when used, indicates a certain point has been reached in the playback of the requested media source.

9. (New) A method of streaming media via a data communications network, the method comprising:

- receiving a universal source locator (URL) in a first request from a first client, wherein the URL includes information indicative of a requested media source;
- determining an identity of the first client;
- determining a first device type based on the identity of the first client;
- based at least in part on the determination of the first device type, generating, with a processor, a first manifest having information for streaming one or more segments of the requested media source in accordance with a first chunking protocol via the data communications network;
- sending the first manifest to the first client;
- receiving the URL in a second request from a second client;
- determining an identity of the second client, the identity of the second client different from the identity of the first client;
- determining a second device type based on the identity of the second client, the second device type different from the first device type;
- based at least in part on the determination of the second device type, generating, with the processor, a second manifest having information for streaming the one

or more segments of the requested media source in accordance with a second chunking protocol via the data communications network, wherein content of the second manifest is different from content of the first manifest; and

sending the second manifest to the second client.

10. (New) The method of streaming media via the data communications network as recited in claim 9, further comprising determining a network type associated with the identity of the first client, wherein either or both of a format or content of the first manifest is further is based, at least in part, on the network type.

11. (New) The method of streaming media via the data communications network as recited in claim 10, wherein the network type includes mobile wireless network or wired network.

12. (New) The method of streaming media via the data communications network as recited in claim 10, further comprising including, in the first manifest, information indicative of a set of bitrates for streaming the requested media source, wherein the set of bitrates is based, at least in part, on either or both of the identity of the first client and the network type.

13. (New) The method of streaming media via the data communications network as recited in claim 12, further comprising including information, in the first manifest, indicative of an initial bitrate of the set of bitrates for streaming the requested media source, wherein the initial bitrate is based, at least in part, on either or both of the identity of the first client and the network type.

14. (New) The method of streaming media via the data communications network as recited in claim 9, further comprising:

receiving data regarding the requested media source, wherein the data includes information indicative of at least one point, during playback of the requested media source, at which at least one other media source is to be shown;

receiving information regarding the at least one other media source; and

including, in either or both of the first manifest and the second manifest, information for streaming one or more segments of the at least one other media source at the at least one point during the playback of the requested media source.

15. (New) The method of streaming media via the data communications network as recited in claim 14, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of playback of the at least one advertisement.

16. (New) The method of streaming media via the data communications network as recited in claim 9, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of a certain point has been reached in the playback of the requested media source.

17. (New) The method of streaming media via the data communications network as recited in claim 9, wherein either or both of the first manifest and the second manifest include at least one URL indicative of a request for an additional manifest.

18. (New) The method of streaming media via the data communications network as recited in claim 9, wherein either or both of the first manifest and the second manifest include at least one URL indicative of a request for content of the requested media source.

19. (New) A non-transitory computer-readable medium having instructions embedded thereon for streaming media via a data communications network, the instructions including computer code for:

receiving a universal source locator (URL) in a first request from a first client, wherein the URL includes information indicative of a requested media source;

determining an identity of the first client;

determining a first device type based on the identity of the first client;

based at least in part on the determination of the first device type, generating a first manifest having information for streaming one or more segments of the requested media source in accordance with a first chunking protocol via the data communications network;

sending the first manifest to the first client;

receiving the URL in a second request from a second client;
determining an identity of the second client, the identity of the second client different from the identity of the first client;
determining a second device type based on the identity of the second client, the second device type different from the first device type;
based at least in part on the determination of the second device type, generating a second manifest having information for streaming the one or more segments of the requested media source in accordance with a second chunking protocol via the data communications network, wherein content of the second manifest is different from content of the first manifest; and
sending the second manifest to the second client.

20. (New) The non-transitory computer-readable medium as recited in claim 19, wherein the instructions further comprise computer code for including information in the first manifest indicative of an initial bitrate of a set of bitrates for streaming the requested media source, wherein the initial bitrate is based, at least in part, on either or both of the identity of the first client and a network type.

21. (New) The non-transitory computer-readable medium as recited in claim 19, wherein the instructions further comprise computer code for:

receiving data regarding the requested media source, wherein the data includes information indicative of at least one point, during playback of the requested media source, at which at least one other media source is to be shown;

receiving information regarding the at least one other media source; and

including, in either or both of the first manifest and the second manifest, information for streaming one or more segments of the at least one other media source at the at least one point during the playback of the requested media source.

Amendments to the Specification:

The specification is amended as indicated below.

Please replace the title with

**DYNAMIC MANIFEST GENERATION BASED ON CLIENT IDENTITY ~~CHUNK~~
~~MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION~~
~~PROGRAMMING INTERFACE~~**

REMARKS/ARGUMENTS

Upon entry of the present amendment, claims 2-21 will be pending in this application. Claims 2-21 have been added. Claim 1 has been canceled. No Claims have been amended. No new matter is added.

CONCLUSION

Applicant believes all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

Except for the issue fees payable under 37 C.F.R. § 1.18, the Director is authorized to charge any additional fees during pendency of this application, including any required extension of time fees, or credit any overpayment to Deposit Account Number 20-1430. This paragraph is intended to be a constructive petition for extension of time in accordance with 37 C.F.R. § 1.136(a)(3).

If the Examiner believes a telephone conference would expedite prosecution of this application, please contact the undersigned at (303) 571-4000 or SMcMillan@KilpatrickTownsend.com.

Respectfully submitted,

/Scott L. McMillan/

Scott L. McMillan
Registration No. 62,079

KILPATRICK TOWNSEND & STOCKTON LLP

Attachments

LB

Electronic Patent Application Fee Transmittal

Application Number:	15337865			
Filing Date:	28-Oct-2016			
Title of Invention:	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE			
First Named Inventor/Applicant Name:	Albert John McGowan			
Filer:	Scott Lee McMillan/Lelia Breeden			
Attorney Docket Number:	092853-1028162 (004130US)			
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
UTILITY FILING FEE (ELECTRONIC FILING)	4011	1	70	70
UTILITY SEARCH FEE	2111	1	300	300
UTILITY EXAMINATION FEE	2311	1	360	360
Pages:				
Claims:				
Miscellaneous-Filing:				
PUBL. FEE- EARLY, VOLUNTARY, OR NORMAL	1504	1	0	0
LATE FILING FEE FOR OATH OR DECLARATION	2051	1	70	70

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 1 month with \$0 paid	2251	1	100	100
Miscellaneous:				
Total in USD (\$)				900

Electronic Acknowledgement Receipt

EFS ID:	28264360
Application Number:	15337865
International Application Number:	
Confirmation Number:	8780
Title of Invention:	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE
First Named Inventor/Applicant Name:	Albert John McGowan
Customer Number:	20350
Filer:	Scott Lee McMillan/Lelia Breeden
Filer Authorized By:	Scott Lee McMillan
Attorney Docket Number:	092853-1028162 (004130US)
Receipt Date:	03-FEB-2017
Filing Date:	28-OCT-2016
Time Stamp:	18:39:26
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 900
RAM confirmation Number	020617INTEFSW18401700
Deposit Account	201430
Authorized User	IPPayments KilpatrickTownsend
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: 37 CFR 1.16 (National application filing, search, and examination fees) 37 CFR 1.17 (Patent application and reexamination processing fees)	

37 CFR 1.19 (Document supply fees)					
37 CFR 1.20 (Post Issuance fees)					
37 CFR 1.21 (Miscellaneous fees and charges)					
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Extension of Time	2017_02_03_092853_1mo_EO T_1028162_004130US.pdf	163711	no	2
			66f46c759254ae83c778c58a1ffd053a1507 acb1		
Warnings:					
Information:					
2	Applicant Response to Pre-Exam Formalities Notice	2017_02_03_092853_Response _to_NOMP_1028162_004130U S.pdf	88668	no	1
			5a9acc8575c1b79931f3056e1e29257aae21 2b79		
Warnings:					
Information:					
3		2017_02_03_092853_Prelimina ry_Amendment_1028162_004 130US.pdf	89752	yes	9
			b3e906a6a22b47b2776807f5664296f5faa7 15e5		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Preliminary Amendment		1	1	
	Claims		2	7	
	Specification		8	8	
	Applicant Arguments/Remarks Made in an Amendment		9	9	
Warnings:					
Information:					
4	Fee Worksheet (SB06)	fee-info.pdf	40671	no	2
			1d34954d08739dcf96c89624331309ba1ce 50d65		
Warnings:					
Information:					
Total Files Size (in bytes):			382802		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

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

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional) 092853-1028162 (004130US)
Application Number 15/337,865	Filed October 28, 2016	
For DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE		
Art Unit 2447	Examiner Not Yet Assigned	

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above-identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

	Fee	Small Entity Fee	Micro Entity Fee	
☒ One month (37 CFR 1.17(a)(1))	\$200	\$100	\$50	\$ <u>100.00</u>
☐ Two months (37 CFR 1.17(a)(2))	\$600	\$300	\$150	\$ _____
☐ Three months (37 CFR 1.17(a)(3))	\$1,400	\$700	\$350	\$ _____
☐ Four months (37 CFR 1.17(a)(4))	\$2,200	\$1,100	\$550	\$ _____
☐ Five months (37 CFR 1.17(a)(5))	\$3,000	\$1,500	\$750	\$ _____

☒ Applicant asserts small entity status. See 37 CFR 1.27.

☐ Applicant certifies micro entity status. See 37 CFR 1.29.
Form PTO/SB/15A or B or equivalent must either be enclosed or have been submitted previously.

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to
Deposit Account Number 20-1430.

☒ Payment made via EFS-Web.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

☐ applicant.

☒ attorney or agent of record. Registration number 62079.

☐ attorney or agent acting under 37 CFR 1.34. Registration number _____.

/Scott L. McMillan/

Signature

February 3, 2017

Date

Scott L. McMillan

Typed or printed name

303.405.1483

Telephone Number

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Submit multiple forms if more than one signature is required, see below*.

☐ * Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.136(a). The information is required to obtain or retain a benefit by the public, which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 6 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

I hereby certify that this correspondence is being filed via
EFS-Web with the United States Patent and Trademark Office
on February 3, 2017.

KILPATRICK TOWNSEND & STOCKTON LLP

PATENT
Attorney Docket No.: 092853-1028162 (004130US)

By: /Lelia Breeden/
Lelia Breeden

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Albert John McGowan et al.

Application No.: 15/337,865

Filed: Oct 28, 2016

For: DYNAMIC CHUNK MANIPULATION

FOR STREAMING MIXED MEDIA:

APPLICATION PROGRAMMING

INTERFACE

Customer No.: 20350

Confirmation No.: 8780

Examiner: Not Yet Assigned

Technology Center/Art Unit: 2447

RESPONSE TO NOTICE OF MISSING
PARTS

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

Commissioner:

Pursuant to the Notice to File Missing Parts of Nonprovisional Application, dated November 9, 2016, transmitted herewith are papers in the above-identified application.

- ☒ Payment in the amount of \$ **800.00** is submitted.
- ☒ 1 month Extension of Time
- ☒ The Commissioner is hereby authorized to charge any additional fees required under 37 CFR §1.16, or credit any overpayment, to Account No. 20-1430.

KILPATRICK TOWNSEND & STOCKTON LLP

By: /Scott L. McMillan/

Scott L. McMillan

Registration No. 62,079

Tel: (303) 571-4000

Email: SMcMillan@KilpatrickTownsend.com



UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
15/337,865	10/28/2016	2447	800	092853-1028162 (004130US)	20	3

CONFIRMATION NO. 8780

UPDATED FILING RECEIPT



0000000089054044

20350
KILPATRICK TOWNSEND & STOCKTON LLP
Mailstop: IP Docketing - 22
1100 Peachtree Street
Suite 2800
Atlanta, GA 30309

Date Mailed: 02/08/2017

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

Inventor(s)

Albert John McGowan, Phoenix, AZ;
Michael M. Gordon, Paradise Valley, AZ;

Applicant(s)

BRIGHTCOVE INC., Boston, MA;

Power of Attorney: The patent practitioners associated with Customer Number 20350

Domestic Priority data as claimed by applicant

This application is a CIP of 14/918,368 10/20/2015 PAT 9485293
which is a CON of 14/086,801 11/21/2013 PAT 9197688
which claims benefit of 61/884,709 09/30/2013
This application 15/337,865
is a CIP of 13/339,680 12/29/2011
which is a CIP of 13/245,372 09/26/2011 ABN
This application 15/337,865
is a CIP of 13/791,789 03/08/2013
which is a CIP of 13/624,029 09/21/2012 PAT 8645504
which is a CON of 13/430,081 03/26/2012 PAT 8301733
which is a CIP of 12/976,883 12/22/2010 PAT 8145782

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.)

AUSTRALIA 2010202741 06/30/2010 No Access Code Provided

Permission to Access Application via Priority Document Exchange: No

Permission to Access Search Results: No

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

If Required, Foreign Filing License Granted: 11/08/2016

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 15/337,865**

Projected Publication Date: 05/18/2017

Non-Publication Request: No

Early Publication Request: No

**** SMALL ENTITY ****

Title

DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION
PROGRAMMING INTERFACE

Preliminary Class

709

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

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

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

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

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

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

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

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

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

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

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

NOT GRANTED

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

SelectUSA

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

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

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 15/337,865			
APPLICATION AS FILED - PART I									
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA		RATE(\$)	FEE(\$)	RATE(\$)	FEE(\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A	70	N/A			
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A		N/A	300	N/A			
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A	360	N/A			
TOTAL CLAIMS (37 CFR 1.16(i))	20	minus 20 =	*	x 40 =	0.00	OR			
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3	minus 3 =	*	x 210 =	0.00				
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).				0.00				
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))					0.00				
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL	730	TOTAL			
APPLICATION AS AMENDED - PART II									
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =	OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =	OR	x =	
	Application Size Fee (37 CFR 1.16(s))						OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =	OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =	OR	x =	
	Application Size Fee (37 CFR 1.16(s))						OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.									



UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/337,865	10/28/2016	Albert John McGowan	092853-1028162 (004130US)

CONFIRMATION NO. 8780

20350
KILPATRICK TOWNSEND & STOCKTON LLP
Mailstop: IP Docketing - 22
1100 Peachtree Street
Suite 2800
Atlanta, GA 30309

INFORMAL NOTICE



0000000089054045

Date Mailed: 02/08/2017

INFORMATIONAL NOTICE TO APPLICANT

Applicant is notified that the above-identified application contains the deficiencies noted below. No period for reply is set forth in this notice for correction of these deficiencies. However, if a deficiency relates to the inventor's oath or declaration, the applicant must file an oath or declaration in compliance with 37 CFR 1.63, or a substitute statement in compliance with 37 CFR 1.64, executed by or with respect to each actual inventor no later than the expiration of the time period set in the "Notice of Allowability" to avoid abandonment. See 37 CFR 1.53(f).

The item(s) indicated below are also required and should be submitted with any reply to this notice to avoid further processing delays.

- A properly executed inventor's oath or declaration has not been received for the following inventor(s):
Albert John McGowan
Michael M. Gordon

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at (571) 272-4000 or (571) 272-4200 or 1-888-786-0101.

/dgela/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		15/337,865
	Filing Date		October 28, 2016
	First Named Inventor	Albert John McGowan	
	Art Unit	2454	
	Examiner Name	Not Yet Assigned	
	Attorney Docket Number	092853-1028162 (004130US)	

U.S. PATENTS						
Examiner Initial*	Cite No	Patent Number	Kind Code ₁	Issue Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	A1	5,612,742		03-18-1997	Krause, et al.	
	A2	5,887,243		03-23-1999	Harvey et al.	
	A3	6,370,199	B1	04-09-2002	Bock et al.	
	A4	6,505,169	B1	01-07-2003	Bhagavath, et al.	
	A5	6,678,332	B1	01-13-2004	Gardere, et al.	
	A6	6,792,047	B1	09-14-2004	Bixby, et al.	
	A7	6,792,618		09-14-2004	Bendinelli et al.	
	A8	6,912,315	B1	06-28-2005	Wong, et al.	
	A9	7,096,481	B1	08-22-2006	Forecast, et al.	
	A10	7,116,894	B1	10-03-2006	Chatterton, G.	
	A11	7,203,758		04-10-2007	Cook et al.	
	A12	7,788,581		08-31-2010	Bauermeister	
	A13	7,827,298		11-02-2010	Black et al.	
	A14	7,877,511		01-25-2011	Berger et al.	
	A15	7,925,549		04-12-2011	Looney et al.	
	A16	7,925,774		04-12-2011	Zhang et al.	
	A17	7,925,849		04-12-2011	Izumi	
	A18	7,958,007		06-07-2011	Urbanski et al.	
	A19	7,992,165		08-02-2011	Ludewig, et al.	
	A20	7,996,566		08-09-2011	Sylvain et al.	
	A21	8,027,787		09-27-2011	Odinak et al.	
	A22	8,027,876		09-27-2011	Taylor	
	A23	8,099,473		01-17-2012	Biderman et al.	
	A24	8,145,782		03-27-2012	McGowan et al.	
	A25	8,165,343		04-24-2012	McGowan	
	A26	8,218,651	B1	07-10-2012	Eshet et al.	
	A27	8,239,546		08-07-2012	McGowan	
	A28	8,250,617		08-21-2012	Hensgen et al.	
	A29	8,301,733		10-30-2012	McGowan	
	A30	8,327,013		12-04-2012	McGowan et al.	
	A31	8,386,630		02-26-2013	Atzmon	
	A32	8,862,754		10-14-2014	McGowan	
	A33	8,954,540		02-10-2015	McGowan et al.	
	A34	9,112,939		08-18-2015	Johnson, Matthew A.	
	A35	9,197,688		11-24-2015	McGowan et al.	
	A36	9,240,922		01-19-2016	McGowan	
	A37	9,485,293		11-01-2016	McGowan et al.	

U.S. PATENT APPLICATION PUBLICATIONS						
Examiner Initial*	Cite No	Publication Number	Kind Code ₁	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	A38	2001-0029525	A1	10-11-2001	Lahr	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		15/337,865
	Filing Date		October 28, 2016
	First Named Inventor	Albert John McGowan	
	Art Unit	2454	
	Examiner Name	Not Yet Assigned	
	Attorney Docket Number	092853-1028162 (004130US)	

	A39	2002-0029282	A1	03-07-2002	Buddhikot, et al.	
	A40	2002-0046404	A1	04-18-2002	Mizutani	
	A41	2002-0073084	A1	06-13-2002	Kauffman, et al.	
	A42	2002-0104096	A1	08-01-2002	Cramer, et al.	
	A43	2002-0122430	A1	09-05-2002	Haberman, et al.	
	A44	2002-0144262	A1	10-03-2002	Plotnick, et al.	
	A45	2002-0150239	A1	10-17-2002	Carney et al.	
	A46	2003-0004804	A1	01-02-2003	Landsman et al.	
	A47	2003-0018966	A1	01-23-2003	Cook et al.	
	A48	2003-0229900	A1	12-11-2003	Reisman	
	A49	2004-0022391	A1	02-05-2004	O'Brien	
	A50	2004-0268384	A1	12-30-2004	Stone	
	A51	2005-0060229	A1	03-17-2005	Riedl, et al.	
	A52	2005-0076368	A1	04-07-2005	Lee, Hye-soo	
	A53	2005-0151859	A1	07-14-2005	Kuriyama et al.	
	A54	2005-0163229	A1	07-28-2005	Okada, et al.	
	A55	2005-0193205	A1	09-01-2005	Jacobs et al.	
	A56	2005-0207569	A1	09-22-2005	Zhang, et al.	
	A57	2005-0209927	A1	09-22-2005	Aaltonen et al.	
	A58	2006-0015637	A1	01-19-2006	Chung	
	A59	2006-0075449	A1	04-06-2006	Jagadeesan et al.	
	A60	2006-0114985	A1	06-01-2006	Linzer	
	A61	2006-0122882	A1	06-08-2006	Brown et al.	
	A62	2006-0129907	A1	06-15-2006	Volk et al.	
	A63	2006-0184410	A1	08-17-2006	Ramamurthy et al.	
	A64	2006-0288112	A1	12-21-2006	Soelberg et al.	
	A65	2007-0038567	A1	02-15-2007	Allaire et al.	
	A66	2007-0053513		03-01-2007	Hoffberg, Steven M.	
	A67	2007-0078712	A1	04-05-2007	Ott, IV, et al.	
	A68	2007-0094082	A1	04-26-2007	Yruski et al.	
	A69	2007-0162571	A1	07-12-2007	Gupta, et al.	
	A70	2007-0168542	A1	07-19-2007	Gupta et al.	
	A71	2007-0198416	A1	08-23-2007	Ye	
	A72	2007-0204310	A1	08-30-2007	Hua et al.	
	A73	2007-0233891	A1	10-04-2007	Luby et al.	
	A74	2007-0255618	A1	11-01-2007	Meerbergen et al.	
	A75	2007-0294100	A1	12-20-2007	Chen et al.	
	A76	2007-0299870	A1	12-27-2007	Finch, Gregory D.	
	A77	2008-0005349	A1	01-03-2008	Li et al.	
	A78	2008-0059310	A1	03-06-2008	Lettow et al.	
	A79	2008-0091845	A1	04-17-2008	Wills et al.	
	A80	2008-0141027	A1	06-12-2008	Kim et al.	
	A81	2008-0195761	A1	08-14-2008	Jabri et al.	
	A82	2008-0207182	A1	08-28-2008	Maharajh, et al.	
	A83	2008-0215620	A1	09-04-2008	Folgnier, et al.	
	A84	2009-0003432	A1	01-01-2009	Liu, et al.	
	A85	2009-0022172	A1	01-22-2009	Haberman et al.	
	A86	2009-0031424	A1	01-29-2009	Ganesan et al.	
	A87	2009-0037211	A1	02-05-2009	McGill et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		15/337,865
	Filing Date		October 28, 2016
	First Named Inventor	Albert John McGowan	
	Art Unit	2454	
	Examiner Name	Not Yet Assigned	
	Attorney Docket Number	092853-1028162 (004130US)	

	A88	2009-0063280	A1	03-05-2009	Wurster, et al.	
	A89	2009-0089846	A1	04-02-2009	Wang et al.	
	A90	2009-0094634	A1	04-09-2009	Haberman et al.	
	A91	2009-0147840	A1	06-11-2009	Sahdra et al.	
	A92	2009-0150941	A1	06-11-2009	Riedl, et al.	
	A93	2009-0172197	A1	07-02-2009	Kalaboukis et al.	
	A94	2009-0182593	A1	07-16-2009	Whitmore, James J.	
	A95	2009-0216790	A1	08-27-2009	Dexter	
	A96	2009-0217316	A1	08-27-2009	Gupta	
	A97	2009-0254572		10-01-2009	Redlich et al.	
	A98	2009-0257435	A1	10-15-2009	Karlsson et al.	
	A99	2009-0259941		10-15-2009	Kennedy, Jr., Howard J.	
	A100	2009-0282077	A1	11-12-2009	Thomas, Edward Eric	
	A101	2009-0287841	A1	11-19-2009	Chapweske et al.	
	A102	2009-0296827	A1	12-03-2009	Karaoguz et al.	
	A103	2009-0300145	A1	12-03-2009	Musayev et al.	
	A104	2009-0320063	A1	12-24-2009	Barrett, Peter T.	
	A105	2009-0327896	A1	12-31-2009	Pall et al.	
	A106	2010-0057926	A1	03-04-2010	Cao et al.	
	A107	2010-0070608	A1	03-18-2010	Hosur, P.I.	
	A108	2010-0070996	A1	03-18-2010	Liao et al.	
	A109	2010-0095121	A1	04-15-2010	Shetty et al.	
	A110	2010-0100742	A1	04-22-2010	Courington et al.	
	A111	2010-0107200	A1	04-29-2010	Drang, et al.	
	A112	2010-0114943	A1	05-06-2010	Fu et al.	
	A113	2010-0118973	A1	05-13-2010	Rodriguez et al.	
	A114	2010-0122286	A1	05-13-2010	Begeja et al.	
	A115	2010-0129057	A1	05-27-2010	Kulkarni	
	A116	2010-0138892	A1	06-03-2010	Meuninck et al.	
	A117	2010-0161425	A1	06-24-2010	Sideman	
	A118	2010-0169303	A1	07-01-2010	Biderman et al.	
	A119	2010-0179987	A1	07-15-2010	Sebastian et al.	
	A120	2010-0189131	A1	07-29-2010	Branam, et al.	
	A121	2010-0205049	A1	08-12-2010	Long et al.	
	A122	2010-0235438	A1	09-16-2010	Narayanan et al.	
	A123	2010-0318672	A1	12-16-2010	Bouazizi	
	A124	2011-0029999	A1	02-03-2011	Foti	
	A125	2011-0058792	A1	03-10-2011	Towner et al.	
	A126	2011-0066703	A1	03-17-2011	Kaplan et al.	
	A127	2011-0071911	A1	03-24-2011	Tung et al.	
	A128	2011-0082924	A1	04-07-2011	Gopalakrishnan	
	A129	2011-0161181	A1	06-30-2011	Esteve Asensio et al.	
	A130	2011-0197219	A1	08-11-2011	Ross, Kevin	
	A131	2011-0231660		09-22-2011	Kanungo, Rajesh	
	A132	2011-0238507	A1	09-29-2011	Ben-Rubi	
	A133	2011-0246603	A1	10-06-2011	Lee	
	A134	2011-0246659	A1	10-06-2011	Bouazizi	
	A135	2011-0252082	A1	10-13-2011	Cobb et al.	
	A136	2011-0264506	A1	10-27-2011	Grant et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		15/337,865	
	Filing Date		October 28, 2016	
	First Named Inventor		Albert John McGowan	
	Art Unit		2454	
	Examiner Name		Not Yet Assigned	
	Attorney Docket Number		092853-1028162 (004130US)	

	A137	2011-0287748	A1	11-24-2011	Angel et al.	
	A138	2012-0005312	A1	01-05-2012	McGowan et al.	
	A139	2012-0005313	A1	01-05-2012	McGowan et al.	
	A140	2012-0023253	A1	01-26-2012	Rhyu Sung-Ryeul, et al.	
	A141	2012-0047542	A1	02-23-2012	Lewis et al.	
	A142	2012-0110117	A1	05-03-2012	Koreeda, Hiroyuki	
	A143	2012-0134355	A1	05-31-2012	Vendrow et al.	
	A144	2012-0137015	A1	05-31-2012	Sun, Wendell	
	A145	2012-0167132	A1	06-28-2012	Mathews et al.	
	A146	2012-0179788	A1	07-12-2012	McGowan, Albert John	
	A147	2012-0185530	A1	07-19-2012	Reza. Areef	
	A148	2012-0185608	A1	07-19-2012	McGowan et al.	
	A149	2012-0197419	A1	08-02-2012	Dhruv, et al.	
	A150	2012-0198492	A1	08-02-2012	Dhruv, et al.	
	A151	2012-0224592	A1	09-06-2012	Henry et al.	
	A152	2012-0233345		09-13-2012	Hannuksela, Miska Matias	
	A153	2012-0265892	A1	10-18-2012	Ma et al.	
	A154	2012-0272262	A1	10-25-2012	Alexander et al.	
	A155	2013-0007799	A1	01-03-2013	Sandoval, Francis R.	
	A156	2013-0080268	A1	03-28-2013	Gordon et al.	
	A157	2013-0080579	A1	03-28-2013	Gordon et al.	
	A158	2013-0083859	A1	04-04-2013	Nemiroff et al.	
	A159	2013-0110978	A1	05-02-2013	Gordon et al.	
	A160	2013-0219428	A1	08-22-2013	Jennings et al.	
	A161	2013-0254346	A1	09-26-2013	McGowan	
	A162	2013-0268961	A1	10-10-2013	Miles et al.	
	A163	2013-0317919	A1	11-28-2013	Raman et al.	
	A164	2014-0150019	A1	05-29-2014	Ma et al.	
	A165	2014-0189139	A1	07-03-2014	Cheng et al.	
	A166	2014-0316899	A1	10-23-2014	McGowan et al.	
	A167	2014-0351383	A1	11-27-2014	Wan et al.	
	A168	2014-0372624	A1	12-18-2014	Wang et al.	
	A169	2015-0095461	A1	04-02-2015	McGowan et al.	
	A170	2015-0095511	A1	04-02-2015	McGowan et al.	
	A171	2016-0119397	A1	04-28-2016	Johnson, Matthew A.	

FOREIGN PATENT DOCUMENTS										
Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ²	Kind Code ⁴	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	T ⁵		
	B1	2010202740	AU	B1	06-30-2010	Unicorn Media, Inc.				
	B2	2010/202741	AU	B1	12-23-2010	UNICORN MEDIA INC.				
	B3	2010202782	AU	B1	07-01-2010	Unicorn Media, Inc.				
	B4	2013240558	AU		04-07-2016	Brightcove Inc.				

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		15/337,865	
	Filing Date		October 28, 2016	
	First Named Inventor		Albert John McGowan	
	Art Unit		2454	
	Examiner Name		Not Yet Assigned	
	Attorney Docket Number		092853-1028162 (004130US)	

	B5	101282478	CN	A	10-08-2008	Shanghai Huaping Information				
	B6	2462732	GB	B	11-17-2010	NDS Limited				
	B7	2514027	GB	A	12-11-2014	Cacti Acquisition LLC				
	B8	2514519	GB	A	26-11-2014	Cacti Acquisition LLC				
	B9	2515683	GB		12-31-2014	Cacti Acquisition LLC				
	B10	2010/025686	WO	A1	03-11-2010	The Chinese University of Hong Kong (Lee, Yiyun)				
	B11	2011/009205	WO	A1	01-27-2011	JIGSEE INC.				
	B12	2011/039617	WO	A1	04-07-2011	NOKIA CORP				
	B13	2012/175145	WO	A1	12-27-2012	Telefonaktiebolaget L M Ericsson				
	B14	2013/101814	WO	A1	07-04-2013	UNICORN MEDIA INC.				
	B15	2013/101841	WO	A1	07-04-2013	UNICORN MEDIA INC.				
	B16	2013/148003	WO	A1	10-03-2013	UNICORN MEDIA INC.				
	B17	2015/047959	WO	A1	04-02-2015	CACTI ACQUISITION LLC				

NON-PATENT LITERATURE DOCUMENTS									
Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.						T ⁵	
	C1	Australian Patent Office, Examiner's First Report on Patent Application No. 2010202740 for Unicorn Media, Inc. (Atty. Ref. No. 028406-001100AU), dated August 10, 2010, 3 pages.							
	C2	Australian Patent Office, Examiner's First Report on Patent Application No. 2010202741 for Unicorn Media, Inc. (Atty. Ref. No. 028406-001000AU), dated August 9, 2010, 4 pages.							
	C3	International Search Report and Written Opinion of PCT/US2014/013724 (92853-897679 (003010PC)), mailed May 27, 2014, all pages.							
	C4	International Preliminary Report on Patentability for PCT/US2014/013724 (92853-897679 (003010PC)), issued August 18, 2015, all pages.							
	C5	International Search Report and Written Opinion of PCT/US2014/017945 (92853-901575 (002820PC)) mailed May 14, 2014, , 10 pages.							

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		15/337,865
	Filing Date		October 28, 2016
	First Named Inventor	Albert John McGowan	
	Art Unit	2454	
	Examiner Name	Not Yet Assigned	
	Attorney Docket Number	092853-1028162 (004130US)	

	C6	International Preliminary Report on Patentability for PCT/US2014/017945 (Atty. Docket No. 92853-901575 (002820PC)) mailed September 8, 2015, all pages.		
	C7	International Search Report and Written Opinion of PCT/US2013/023181 (92853-858085 (002810PC)), mailed on May 8, 2013, 91 pages.		
	C8	International Preliminary Report on Patentability for PCT/US2013/023181 (92853-858085 (002810PC)), mailed Oct. 9, 2014, 6 pages.		
	C9	International Search Report and Written Opinion of PCT/US2013/025180 (92853-858083 (001020PC)), mailed on June 5, 2013, 105 pages.		
	C10	International Preliminary Report on Patentability for PCT/US2013/025180 (Atty. Docket No. 092853-0858083 (001020PC)) mailed October 9, 2014, all pages.		
	C11	International Search Report and Written Opinion for PCT/US2014/056823 (Atty. Docket No. 092853-0916778 (004100PC)) mailed December 9, 2014, 10 pages.		
	C12	International Search Report and Written Opinion for PCT/US2012/071629 (92853-858104 (002700PC)) mailed May 4, 2013, 11 pages.		
	C13	International Preliminary Report on Patentability for PCT/US2012/071629 (Atty Docket No. 92853-858104 (002700PC)), mailed July 10, 2014, 8 pages.		
	C14	International Search Report and Written Opinion of PCT/US2012/071669 mailed on April 5, 2013, 74 pages.		
	C15	International Preliminary Report on Patentability for PCT/US2012/071669 (Atty Docket No. 92853-858102 (002600PC)), mailed July 10, 2014, 7 pages.		
	C16	BEN ATALLAH, S., et al., "Dynamic Configuration of Multimedia Applications," IFIP International Federation for Information Processing, 2003, 18 pages.		
	C17	DVB ORGANIZATION: "TM- IPI- 2552r12- CDS- Specification_clean.doc", DVB, Digital Video Broadcasting, c/o EBU- 17A Ancienne Route - CH-1218 Grand Saconnex, Geneva - Switzerland, 11 September 2008, XP017811009, sections 3.1-3.2, section 4; page 20, section 4.2.2.2, section 4.2.4, section 5.2.5		
	C18	GABIN, F. et al. "3GPP Mobile Multimedia Streaming Standards [Standards in a Nutshell]", IEEE SIGNAL PROCESSING MAGAZINE, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 27, no. 6, 1 Nov. 2010, pgs 134-138, XPO11317684, ISSN: 1053-5888, page 135 - page 137		
	C19	PRANGL, M. ET AL. "Towards QoS Improvements of TCP-Based Media Delivery," NETWORKING AND SERVICES, 2008, ICNS 2008: FOURTH INTERNATIONAL CONFERENCE ON IEEE, Piscataway, NJ, USA, March 16, 2008, 6 pages.		

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		15/337,865
	Filing Date		October 28, 2016
	First Named Inventor	Albert John McGowan	
	Art Unit	2454	
	Examiner Name	Not Yet Assigned	
	Attorney Docket Number	092853-1028162 (004130US)	

	C20	SDK Compatibility Guide, "Overview of SDK-Based Development", 15 November 2010, Apple Inc., 2 pages, https://developer.apple.com/library/mac/documentation/DeveloperTools/Conceptual/cross_development/Overview/overview.html		
	C21	U.S. Patent Application No. 14/086,801 (Atty. Docket No. 092853-0882531 (004100US)) filed November 21, 2013, Preinterview first office action mailed February 20, 2014, 15 pages.		
	C22	U.S. Patent Application No. 14/086,801 (Atty. Docket No. 092853-0882531 (004100US)) filed November 21, 2013, Non-Final Rejection mailed July 9, 2014, 15 pages.		
	C23	U.S. Patent Application No. 14/086,801 (Atty. Docket No. 092853-0882531 (004100US)) filed November 21, 2013, Final Rejection mailed October 24, 2014, 20 pages.		
	C24	U.S. Patent Application No. 14/086,801 (Atty. Docket No. 092853-0882531 (004100US)) filed November 21, 2013, Non-Final Rejection mailed April 8, 2015, 28 pages.		
	C25	U.S. Patent Application No. 14/086,801 (Atty. Docket No. 092853-0882531 (004100US)) filed November 21, 2013, Notice of Allowance mailed July 22, 2015, 31 pages.		
	C26	U.S. Patent Application No. 13/339,680 filed December 29, 2011 (Atty Docket No. 92853-827299 (002700US)), Final Office Action mailed on August 18, 2016, all pages.		
	C27	U.S. Patent Application No. 13/339,680 filed December 29, 2011 (Atty Docket No. 92853-827299 (002700US)), Non-Final Office Action mailed on January 14, 2016, all pages.		
	C28	U.S. Patent Application No. 13/339,680 filed December 29, 2011 (Atty Docket No. 92853-827299 (002700US)), Non-Final Office Action mailed on March 12, 2015, all pages.		
	C29	U.S. Patent Application No. 13/339,680 filed December 29, 2011 (Atty Docket No. 92853-827299 (002700US)), Non-Final Office Action mailed July 10, 2014, 22 pages.		
	C30	U.S. Patent Application No. 13/339,680 filed December 29, 2011 (Atty Docket No. 92853-827299 (002700US)), Final Office Action mailed on October 15, 2013, 13 pages.		
	C31	U.S. Patent Application No. 13/339,680 filed December 29, 2011 (Atty Docket No. 92853-827299 (002700US)), Final Office Action mailed on May 30, 2013, 12 pages.		
	C32	U.S. Patent Application No. 13/339,680 filed December 29, 2011 (Atty Docket No. 92853-827299 (002700US)), Non-Final Office Action mailed on July 25, 2012, 10 pages.		
	C33	U.S. Patent Application No. 13/339,680 filed December 29, 2011 (Atty Docket No. 92853-827299 (002700US)), Non-Final Office Action mailed on March 7, 2012, 14 pages.		

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		15/337,865
	Filing Date		October 28, 2016
	First Named Inventor	Albert John McGowan	
	Art Unit	2454	
	Examiner Name	Not Yet Assigned	
	Attorney Docket Number	092853-1028162 (004130US)	

	C34	U.S. Patent Application No. 13/430,081 filed March 26, 2012 (Atty. Docket No. 92853-827300 (002800US)), Notice of Allowance mailed June 22, 2012, 19 pages.		
	C35	U.S. Patent Application No. 13/624,029 filed September 21, 2012 (Atty. Docket No. 92853-853368 (002810US)), Non-Final Office Action mailed February 5, 2013, 10 pages.		
	C36	U.S. Patent Application No. 13/624,029 filed September 21, 2012 (Atty. Docket No. 92853-853368 (002810US)), Notice of Allowance mailed May 30, 2013, 13 pages.		
	C37	U.S. Patent Application No. 13/245,372 filed September 26, 2011 (Atty. Docket No. 92853-807568 (002200US)), Non-Final Office Action mailed January 26, 2012, all pages.		
	C38	U.S. Patent Application No. 13/245,372 filed September 26, 2011 (Atty. Docket No. 92853-807568 (002200US)), Non-Final Office Action mailed September 5, 2012, all pages.		
	C39	U.S. Patent Application No. 13/245,372 filed September 26, 2011 (Atty. Docket No. 92853-807568 (002200US)), Final Office Action mailed August 21, 2013, all pages.		
	C40	U.S. Patent Application No. 13/245,372 filed September 26, 2011 (Atty. Docket No. 92853-807568 (002200US)), Non-Final Office Action mailed June 4, 2015, all pages.		
	C41	U.S. Patent Application No. 13/245,372 filed September 26, 2011 (Atty. Docket No. 92853-807568 (002200US)), Non-Final Office Action mailed December 10, 2015, all pages.		
	C42	U.S. Patent Application No. 13/245,372 filed September 26, 2011 (Atty. Docket No. 92853-807568 (002200US)), Final Office Action mailed April 27, 2016, all pages.		
	C43	U.S. Patent Application No. 12/976,883 filed December 22, 2010 (Atty. Docket No. 92853-796968 (001000US)), Non-Final Office Action mailed July 7, 2011, all pages.		
	C44	U.S. Patent Application No. 12/976,883 filed December 22, 2010 (Atty. Docket No. 92853-796968 (001000US)), Final Rejection mailed December 20, 2011, all pages.		

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	15/337,865
	Filing Date	October 28, 2016
	First Named Inventor	Albert John McGowan
	Art Unit	2454
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	092853-1028162 (004130US)

EXAMINER SIGNATURE			
Examiner Signature		Date Considered	
<p>*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>			
<p>¹ See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. ² Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). ³ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document.</p> <p>⁴ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁵ Applicant is to place a check mark here if English language translation is attached.</p>			

I hereby certify that this correspondence is being filed via
EFS-Web with the United States Patent and Trademark Office
on February 13, 2017.

PATENT
Attorney Docket No.: 092853-1028162
(004130US)

KILPATRICK TOWNSEND & STOCKTON LLP

By: /Tina Bello/
Tina Bello

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Albert John McGowan, et al.

Application No.: 15/337,865

Filed: October 28, 2016

For: DYNAMIC MANIFEST

GENERATION BASED ON CLIENT

Customer No.: 20350

Confirmation No.: 8780

Examiner: Not Yet Assigned

Technology Center/Art Unit: 2454

INFORMATION DISCLOSURE

STATEMENT

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

Commissioner:

The references cited on attached form PTO/SB/08A are being called to the attention of the Examiner. In accordance with the provisions of 37 CFR §1.98(a)(2), copies of any cited U.S. Patent and U.S. Patent Application Publications are not provided. Copies of the remaining cited references are provided.

It is respectfully requested that the cited references be expressly considered during the prosecution of this application, and the references be made of record therein and appear among the “references cited” on any patent to issue therefrom.

As provided for by 37 CFR §1.97(g) and (h), no inference should be made that the information and references cited are prior art merely because they are in this statement and no representation is being made that a search has been conducted or that this statement encompasses all the possible relevant information.

Applicant believes that no fee is required for submission of this statement. However, if any additional fees are due for the submission of this Information Disclosure Statement, please deduct those fees from Deposit Account No. 20-1430.

Respectfully submitted,

/Scott L. McMillan/

Scott L. McMillan
Registration No. 62,079

KILPATRICK TOWNSEND & STOCKTON LLP

Attachment

Electronic Acknowledgement Receipt

EFS ID:	28333537
Application Number:	15337865
International Application Number:	
Confirmation Number:	8780
Title of Invention:	DYNAMIC CHUNK MANIPULATION FOR STREAMING MIXED MEDIA: APPLICATION PROGRAMMING INTERFACE
First Named Inventor/Applicant Name:	Albert John McGowan
Customer Number:	20350
Filer:	Scott Lee McMillan/Tina Bello
Filer Authorized By:	Scott Lee McMillan
Attorney Docket Number:	092853-1028162 (004130US)
Receipt Date:	13-FEB-2017
Filing Date:	28-OCT-2016
Time Stamp:	11:11:30
Application Type:	Utility under 35 USC 111(a)

Payment information:

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

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Foreign Reference	AU2010202740.pdf	4150306	no	41
			e374def4b83ef4bedb30d3e197b0d854a22a831d		

Warnings:

Information:					
2	Foreign Reference	FOR_AU2010202741.pdf	1106176	no	40
			807e0db510f8fcec4d4ac2ca75b0f8a2e9583379		
Warnings:					
Information:					
3	Foreign Reference	AU2010202782.pdf	7673450	no	54
			19129c536ace1fe6608f89315d1a828e9f6ee798		
Warnings:					
Information:					
4	Foreign Reference	AU_2013240558.pdf	14459259	no	44
			53bdb410fd24ee29d2d62839af04c1258ff78921		
Warnings:					
Information:					
5	Foreign Reference	GB2514027.pdf	465339	no	1
			d05c56ffc30ee830a29ae73534a0666d697494b8		
Warnings:					
Information:					
6	Foreign Reference	GB2514519.pdf	364731	no	1
			ed392d70baff47ea0607e6b2a66a5e4c25422a19		
Warnings:					
Information:					
7	Foreign Reference	GB2515683.pdf	354827	no	1
			9555195317666e57eb0b64e02fed31072af1f38b		
Warnings:					
Information:					
8	Foreign Reference	WO11009205.pdf	2815029	no	55
			ea28a22609efc210c8c6cafccf1431423ba9c86c		
Warnings:					
Information:					

9	Foreign Reference	FOR_WO11039617.pdf	2462866	no	50
			f92d53338c9d6624ee2728da4cfd8d42f994233c		
Warnings:					
Information:					
10	Foreign Reference	FOR_WO12175145.pdf	2380129	no	48
			b7a07f5f258b5907556ff28dd2482b0e0c83edcd		
Warnings:					
Information:					
11	Foreign Reference	WO13101814.pdf	12084077	no	69
			4fb6ecde178f884cb0590b8ab194f755bb726fc5		
Warnings:					
Information:					
12	Foreign Reference	WO13101841.pdf	12453881	no	70
			93cf5d01e3f00f8bda2c9bd3788a41f17854b386		
Warnings:					
Information:					
13	Foreign Reference	WO13148003.pdf	2053188	no	45
			638be150ec226c2d6f16687c70b438309e89dcd0		
Warnings:					
Information:					
14	Foreign Reference	FOR_WO2015047959.pdf	7480475	no	46
			1a0aecec5aec8c1342f76f5acb8224660d9a1b3d		
Warnings:					
Information:					
15	Non Patent Literature	NPL_2010202740_AU.pdf	119785	no	3
			996a7796e913858364662e0e06de220f017dccc7		
Warnings:					
Information:					

16	Non Patent Literature	NPL_2010202741_AU.pdf	405845	no	4
			13c92c6af1dbfaf56f621cb051cb22f7a1c7f5e8		
Warnings:					
Information:					
17	Non Patent Literature	NPL_BEN_ATALLAH.pdf	274800	no	18
			f2041bba85a1e029c8f6a30087df760d5a5507a0		
Warnings:					
Information:					
18	Non Patent Literature	NPL_DVB_ORGANIZATION.pdf	8872363	no	98
			f68b7c1dbf7b8f2e8a258ea8efeb40d270fd936a		
Warnings:					
Information:					
19	Non Patent Literature	GABIN_3gpp_mobile_multimedia_streaming_standards.pdf	97732	no	5
			e0af90c6041e8c91a0e29009f31786ea0e09caba		
Warnings:					
Information:					
20	Non Patent Literature	Prangl_Towards_QoS_Improvements_TCP.pdf	770909	no	6
			fe0e4c39a367d8623fa5f98b29825f185bd93958		
Warnings:					
Information:					
21	Non Patent Literature	NPL_Patentability_PCTUS2012_071669_07_10_2014.pdf	6304718	no	7
			c78eb5c317793af39f898157e35a1edc9446e04d		
Warnings:					
Information:					
22	Non Patent Literature	NPL_ISR_WO_PCTUS2012071669_04_05_2013.pdf	4578532	no	74
			b8f12c99f1821b14b677683f8d06de31a79ad674		
Warnings:					
Information:					

23	Non Patent Literature	NPL_Patentability_PCTUS2012_071629_July_10_2014.pdf	6925100	no	8
			8eec698a789cb6043317a3e22e08a5f227d2efb7		
Warnings:					
Information:					
24	Non Patent Literature	NPL_ISR_WO_PCTUS2012071629_04_05_2013.pdf	5674258	no	92
			0a549a0fcdce6ec9c08a9db84e8df1505cf63dd98		
Warnings:					
Information:					
25	Non Patent Literature	NPL_Patentability_Rpt_PCTUS2014013724.pdf	7041279	no	8
			a3a3a8f69dfde531f68466ffa4a4f952ebab8f39		
Warnings:					
Information:					
26	Non Patent Literature	NPL_ISR_WO_PCTUS2014013724.pdf	12317968	no	13
			fb12c02cc338a12d5f1f089fb847e8a958503814		
Warnings:					
Information:					
27	Non Patent Literature	NPL_ISR_WO_PCTUS2014_017945_05_14_2014.pdf	10838764	no	10
			d89e6b2193cbe01d7d00372f3433de90d1446e55		
Warnings:					
Information:					
28	Non Patent Literature	NPL_Patentability_Rpt_PCTUS2014017945.pdf	7378514	no	8
			3cc752c0bb71a4d5aa26fa54b6d47191cba9c342		
Warnings:					
Information:					
29	Non Patent Literature	NPL_Patentability_US2013023181.pdf	227285	no	6
			786d69389647ac23f55cc5f54720770e0e740fad		
Warnings:					
Information:					

30	Non Patent Literature	NPL_ISR_WO_PCTUS2013_023 181_05_08_2013.pdf	4671404	no	91
			3a73cf48b94e71e863ce4aafa4dfa086479260a0		
Warnings:					
Information:					
31	Non Patent Literature	NPL_ISR_WO_PCTUS2013_025 180_06_05_2013.pdf	5829006	no	105
			715c60557ac1dbc34c07d71e84c0c6eb7182477f		
Warnings:					
Information:					
32	Non Patent Literature	Patentability_Rpt_PCT_US2013_025180.pdf	7492373	no	9
			ca7dc44be013493518c6b559ff3fe36a8638100f		
Warnings:					
Information:					
33	Foreign Reference	FOR_WO10025686.pdf	1305177	no	30
			98a84a37c643ff39a1eff7cac7cf36148c38f5de		
Warnings:					
Information:					
34	Foreign Reference	FOR_2462732_GB.pdf	805070	no	25
			964eb84ca8626dc94bda5d9a4129829cca92cf85		
Warnings:					
Information:					
35	Foreign Reference	FOR_101282478_CN.pdf	943037	no	14
			062683156600c4eda49d5ccc3539033787e2836a		
Warnings:					
Information:					
36	Non Patent Literature	NPL_SDK_Compatibility_Guide.pdf	1455320	no	17
			b02f9c981077a3ba429e61a44860f3c10e56ae8c		
Warnings:					
Information:					

37	Non Patent Literature	NPL_ISR_WO_PCTUS2014056823.pdf	10164680 31d9b2e817a22924fd01f7fd006f1d65e3505647	no	10
Warnings:					
Information:					
38		2017_02_13_092853_1028162_IDS.pdf	195810 4ac40dbf60c59567b94846f9d206a98add2f135e	yes	11
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Information Disclosure Statement (IDS) Form (SB08)		3	11	
	Transmittal Letter		1	2	
Warnings:					
Information:					
Total Files Size (in bytes):			174993462		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

(19) AUSTRALIAN PATENT OFFICE

(54) Title
Dynamic indexing for ad insertion in media streaming

(51)⁶ International Patent Classification(s)
G06F 17/30 (2006.01) : main **G06F**
G06F 15/16 (2006.01) 15/16
G06F 17/30 20060101ALI2010070
20060101AFI2010070 : main

(21) Application No: 2010202740 (22) Application Date: 2010 06 30

(43) Publication Date : 2010 12 23

(43) Publication Journal Date : 2010 12 23

(71) Applicant(s)
Unicorn Media, Inc.

(72) Inventor(s)
McGowan, Albert J., Caris, Richard L.

(74) Agent/Attorney
Pizzey's Patent and Trade Mark Attorneys, Level 7 Woden Plaza Offices Woden Town Square,
Woden, ACT, 2606

(56) Related Art
US 2010/0129057 A1
GB 2462732 A
WO 2010/025686 A1
US 2009/0327696 A1

2010202740 30 Jun 2010

ABSTRACT OF THE DISCLOSURE

Systems and methods for generating a file index for streaming media over a network are disclosed. These systems and methods provide for receiving requests for an index file from and generating the requested index file dynamically. The index file may be generated in a manner to ensure multiple index file requests are received for streaming a single media file. These systems and methods can be used with systems that chunk media files dynamically to enable a high degree of flexibility in streaming chunked media files and preclude the need to chunk the media files and store the chunks prior to streaming.

2010202740 30 Jun 2010

Regulation 3.2

AUSTRALIA

Patents Act 1990

COMPLETE SPECIFICATION
STANDARD PATENT

APPLICANT: **UNICORN MEDIA, INC.**

Invention Title: **DYNAMIC INDEXING FOR AD INSERTION IN
MEDIA STREAMING**

The following statement is a full description of this invention, including the best method of performing it known to me:

DYNAMIC INDEXING FOR AD INSERTION IN MEDIA STREAMING

BACKGROUND OF THE INVENTION

[0001] This disclosure relates in general to cloud-based computer processing and, but not by way of limitation, to indexing and chunking media files for use in media streaming.

[0002] The delivery of media over networks such as the Internet can be accomplished in many ways, including progressive downloading or streaming. Streaming is often preferred to progressive downloading because it offers additional features such as content protection and the ability to seek to undownloaded parts of a media file. The process of preparing a media file for streaming typically involves "chunking" the file, or dividing files up into smaller segments for delivery. Information including where chunks may be accessed can be stored in an index file. This index file can be delivered to a client, such as a media player application, for use in streaming.

[0003] The processes of chunking and indexing files for streaming present challenges to a content delivery network or content provider desiring to host media files for streaming. For example, a significant amount of preprocessing is required to prepare media for streaming. Media content may be encoded into several different files to accommodate several different sub-streams. Each of these files typically are chunked, stored, and indexed before the media content is available for streaming. This preprocessing leaves little flexibility for inserting an advertisement during streaming and can require a large amount of storage space to store the preprocessed chunks of media.

[0004] The inventions embodied herein address these and other problems.

BRIEF SUMMARY OF THE INVENTION

[0005] Systems and methods for generating a file index for streaming media over a network are disclosed. These systems and methods may be part of a larger media servicing network that can be used to, among other things, process uploaded media content, provide it for streaming, and collect metric information regarding the streaming. The disclosed systems and methods provide for receiving requests for an index file from, for example, a client program, and generating the requested index file dynamically. The index file may be generated in a manner to

ensure multiple index file requests can be received for streaming a single media file. These systems and methods can be used with systems that chunk media files dynamically to enable a high degree of flexibility in streaming chunked media files and preclude the need to chunk the media files and store the chunks prior to streaming.

[0006] In other words, systems and methods disclosed herein generate a plurality of index files for use in streaming a media file, enabling dynamic advertisement and/or secondary media insertion by receiving a request for a first index file, then determining a first segment of media for streaming by determining whether to include media, other than the media file, in the first segment of media for streaming and determining a starting point and an ending point of the first segment of media for streaming. The requested first index file is generated and provided based, at least in part, on the determination of the first segment of media for streaming. A similar process is followed upon the receipt of a request for a second index file, and first and/or second segments of media for streaming can include media other than the media file, including an advertisement of a certain length. Further embodiments of systems and methods disclosed herein provide for generating index files and Universal Resource Indicators (URIs) separately, to enable the detection of which chunk of media is requested by a media streaming application.

[0007] Embodiments additionally include causing the server to create data associated with playback of the media file. The data can be based on when a request for the first and/or second index file is received, when a requested index file is provided, a length of time to play back a segment of media, the starting and/or ending point of a segment of media, whether the request for the second index file has been received within a certain amount of time since receipt of the request for the first index file; and whether a segment of media for streaming includes media other than the media file. This data may be transmitted over a network.

[0008] Embodiments further allow for the determination of a segment of media for streaming to be based, at least in part on one or more of information regarding an end user, preferences and/or selection of an end user, information based on the behavior of an end user as determined by machine-based logic, information regarding the content of the media file, the time of day, information regarding the availability of media other than the media file, information regarding the source of the media file, and more.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present disclosure is described in conjunction with the appended figures.

- [0010] FIG. 1 illustrates a block diagram of a media servicing system.
- [0011] FIG. 2A illustrates a block diagram of an embodiment of a kernel application center connected with application centers.
- [0012] FIG. 2B illustrates a block diagram of an alternative embodiment of a kernel application center.
- [0013] FIG. 3 illustrates a block diagram of an embodiment of an application center.
- [0014] FIG. 4 illustrates a block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion.
- [0015] FIG. 5A illustrates a simplified block diagram of an embodiment of a system configured to provide dynamic indexing and chunking for media streaming.
- [0016] FIG. 5B illustrates a simplified block diagram of another embodiment of a system configured to provide dynamic indexing and chunking for media streaming.
- [0017] FIG. 5C illustrates a simplified block diagram of another embodiment of a system configured to provide dynamic indexing and chunking for media streaming, utilizing a redirector.
- [0018] FIG. 6 illustrates a simplified flowchart of an embodiment of a method for implementing a dynamic index for media streaming.
- [0019] FIG. 7 illustrates a simplified flowchart of an embodiment of a method for dynamically chunking a media file for streaming.
- [0020] FIG. 8 illustrates a simplified swim lane flowchart describing the interaction of components in a system configured to provide dynamic indexing and chunking for media streaming, according to one embodiment.
- [0021] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION OF THE INVENTION

[0022] The ensuing description provides preferred exemplary embodiment(s) only, and is not intended to limit the scope, applicability or configuration of the disclosure. Rather, the ensuing description of the preferred exemplary embodiment(s) will provide those skilled in the art with an enabling description for implementing a preferred exemplary embodiment. It is understood that various changes may be made in the function and arrangement of elements without departing from the spirit and scope as set forth in the appended claims.

[0023] The increased availability of media content over data communications networks such as the Internet has mirrored the increased bandwidth for these networks. Because media has recently taken a more prominent role in data communications, the distribution of media and the data associated with such distribution has become increasingly important, particularly to media content providers. Media streaming has become a widely-used method of media distribution, but the preprocessing associated with streaming can be burdensome. Certain protocols, including forms of Hypertext Transfer Protocol (HTTP) streaming, require chunking and storing media assets, and generating a corresponding index files. These requirements can deprive a content provider of the ability to dynamically insert additional media such as advertisements into a media stream, and can consume a large amount of storage space to store chunks of media for a media asset, including chunks for any alternative sub-streams (e.g., streams with alternative bitrates, captions, alternative languages, etc.). Certain systems and methods can be utilized, however, to introduce the desired functionality back into the system.

[0024] A traditional approach to preprocessing media for streaming involves chunking and storing media assets, then creating corresponding index files to indicate where chunks may be located to download for streaming. Streaming protocols often provide for frequently updating an index file for instances where the corresponding media is frequently updated, such as during live streaming. Thus, an index file does not need to contain all chunks for a requested media asset. In addition, because media files are frequently stored in a format that requires little additional processing to chunk, the chunks can be created in real time, during the streaming of a media file. The systems and methods disclosed herein take advantage of these features to enable dynamic index file creation and dynamic media file chunking.

[0025] For instance, rather than preprocess media assets for streaming by chunking and indexing all files with relevant sub-streams prior to streaming the media, a server can

dynamically create and update an index file during streaming. The dynamically-created index file can contain information regarding a next chunk of media in the various available sub-streams. The next chunk of media may not be cached at a location specified in the index file, in which case a chunk may be dynamically created by pulling all or part of the media file of interest from a media file origin, chunking it, and making it available for download. The chunk also may be cached, thereby eliminating the need to create the chunk again if it is requested at some later time.

[0026] Because the chunks are created during streaming, a content provider and/or media distributor can have more information and control during the streaming process. Rather than generate single index file for a given media asset, an instance of the index file generator may be created at the beginning of the media streaming to provide individualized media content to a particular end user and unique information regarding the streaming session to a content provider. The file index generator can vary the length of each chunk by, for example, indicating starting and ending points in the index file. Thus, the file index generator may determine a uniform chunk length for a media asset, varying the length of the chunks for different media assets, or the file index generator may adjust the length of the chunks within a single media asset. The index file generator can further insert additional media, such as an advertisement, at any time during the streaming by specifying the location of the additional media in the index file. The determination to insert advertisements can be based on any information, including data collected during the streaming session.

[0027] As the index file generator receives requests for and generates index files, it can further gather data regarding the streaming session for reporting to a content provider. Content providers often rely on beaconing data collected from media player applications to determine when an end user stops, plays, pauses, skips, etc., the streaming media content. Such information can be vital in determining the value of the media.

[0028] Because not all media player applications provide this beaconing data, the data gathered by the index file generator can serve as a substitute for or complement to the beaconing data. For example, if a request is made for a chunk that does not immediately follow a previously-requested chunk, a skip was made. If the amount of time elapsed between a previous request and a subsequent request exceeds the time for playback of the previously-requested chunk, a pause

was made. If a request is not received within a certain time since a prior request, it can be determined that a stop was made.

[0029] As illustrated above, the state of a client may be determined from a variety of factors. This can include when the request for the index file is received, when the index file is provided, a length of time to play back the segment of media for streaming, and/or the starting and/or ending point of the segment of media for streaming. The determined state of a client may also be based on whether the request for the index file has been received within a certain amount of time since receipt of a previous request for an index file, whether the segment of media for streaming includes media other than the media file, and more. The state of a client and/or the data from which it was determined, may be used to create reporting data to serve as a substitute or complement to beaconing data from a client media player application. Because the index file generator can determine the length of the chunks, it therefore can determine the frequency of subsequent index file requests and the resolution of the reporting data based on the requests. The index file generator may log the reporting data and/or transmit the reporting data over a network during streaming.

[0030] The determined state of a client may be used by the index file generator and/or other services for various purposes. For example, it may be used in behavioral advertisement targeting and enforcement of session advertisement behavior, adjusting advertisement content and playback based on the behavior of a user as determined by the state of a client. The state of a client further may be used to support resume features on a per client basis, allowing a user to continue playback of a media asset from a point at which the user had previously stopped playback. The state of a client also may be used to support individual encryption keys in an encryption scheme and allow the index file generator to return secure URLs (e.g., time expiring or Internet Protocol (IP) allowed) for chunks to support functions such as payment services.

[0031] Additionally or alternatively, the tasks of generating the index file and providing a location a requested chunk can be split up, thereby enabling the system to determine which chunks are actually requested. For example, a system may be configured to dynamically create an index file having links to one or more redirectors on the system. These redirectors can be configured to issue the location of the chunk, which can be created dynamically. The redirectors can further determine which chunk is actually requested, thereby enabling, among other things, calculation of Quality of Service (QoS) metrics, an increase the accuracy of reporting data, a

decrease the frequency of index file generation if efficient to do so, and the ability to more easily handle keys of an encryption scheme.

[0032] While the above embodiments may be implemented in a variety of different systems, some particular embodiments may be implemented as part of a media service system. FIG. 1 is a block diagram illustrating a media servicing system 100, according to some embodiments of the present invention. The system may deliver media content to the end user device 140 through a network such as the Internet 120. The end user device 140 can be one of any number of devices configured to receive media over the Internet 120, such as a mobile phone, tablet computer, personal computer, portable media device, etc. A media asset provided by a content provider 130 can be processed and indexed by cloud-hosted integrated multi-node pipelining system (CHIMPS) 110, and further stored on content delivery network (CDN) 150. Additionally or alternatively, the CHIMPS 110 may also be adapted to store the media asset.

[0033] The media servicing system further enables a content provider 130 or other entity to gather information regarding user behavior during media playback. For example, a content provider 130 can be provided with data indicating that end users tend to stop watching a video at a certain point in playback, or that users tended to follow links associated with certain advertisements displayed during playback. With this data, a content provider 130 can adjust factors such as media content, advertisement placement and content, etc., to increase revenue associated with the media content and provide the end user device 140 with a more desirable playback experience.

[0034] End user device 140 can request a media asset to stream with a client program executed by the end user device 140. The client program can be, for example, a media player, browser, or other application adapted to request and/or play media assets. In response to a request for a media asset, the CHIMPS 110 can utilize any number of application centers 112 and/or kernel application center(s) 111 to provide the client program with a data object concerning the requested media asset. The data object can include information about the media asset, including where the media asset can be located, such as within the CDN 150 or within the CHIMPS 150 itself. Location information may be provided by Universal Resource Indicator (URI), a Universal Resource Locator (URL) or other indicator. During playback of the media asset, the CHIMPS 150 can collect data regarding the playback through beaconing provided by a client program executed by the end user device 140 and/or indexing service from within the CHIMPS

and/or CDN. The CHIMPS 150 can subsequently provide the data and/or any analytics information derived from the data to the content provider 130.

[0035] FIG. 2A is a block diagram illustrating an embodiment of a kernel application center 111-1 connected with application centers from within the CHIMPS 110-1. The kernel application center 111-1 and application centers 112 can be geographically distant and can be connected via the Internet 120, wide area network (WAN), and/or other data communication network. Because application centers can be geographically separated, DNS services (not shown) can be used to allow an end user device 140 to connect to the nearest available application center 112. The kernel application center 111-1 can connect with application centers 112 within the CHIMPS 110-1 through an internal interface 270, thereby enabling the application centers 112 access to the various components within the kernel application center 111-1.

[0036] Components within the kernel application center 111-1 can communicate through network 260 such as a local area network (LAN) and can include one or more origin servers 240 and a storage array 230 with which data objects and/or media assets may be stored and distributed. The storage array 230 may also be utilized by services running on processing server(s) 220 and/or transcoding server(s) 250 that may require temporary or long-term storage. Kernel server 210 can utilize processing server(s) 220, transcoding server(s) 250 to provide various functional capabilities to the CHIMPS 110.

[0037] For example, as described in more detail below, the CHIMPS 110-1 can provide transcoding service for media assets provided by a content provider 130 for syndication. Such a service can allow a content provider 130 to upload a media asset to an application center 112, after which the application center 112 would notify the kernel server 210 that the media asset has been uploaded. The kernel server can then notify services running on the processing server(s) 220 of the upload. These services can utilize transcoding server(s) to transcode the media asset, which can then be moved to a CDN and/or stored locally by storage array 230 and origin server(s) 240. Services running on the processing server(s) 220 can also update the associated data object stored by the storage array 230 and origin server(s) 240.

[0038] FIG. 2B is a block diagram illustrating an alternative embodiment of a kernel application center 111-2. In addition to the components of the embodiment of FIG. 2A, this embodiment incorporates an application center 112 within the kernel application center 111-2.

The application center 112 incorporated within kernel application center 111-2 may be located at or near the other components of the kernel application center 111-2, and can be communicatively connected to the other components via network 260. The incorporated application center 112 can therefore have faster access to kernel application center functionality because it does not need to communicate over long distances. In consideration of this advantage, it will be understood that the CHIMPS 110 can include multiple kernel centers with one or more application centers incorporated therein. Additionally or alternatively, components of the kernel application center may be incorporated into one or more application centers 112 in the CHIMPS 110 to provide quicker access to certain functionality.

[0039] FIG. 3 is a block diagram illustrating an embodiment of an application center 112. The application center 112 can include caching server(s) 330 and a storage array 310 for storing and distributing data objects of media assets requested by end user devices through end user interface 360. Caching server(s) 330 and storage array 310 can also be used to collect, process, and/or store metrics information from beaconing data, media chunk requests, and/or other data sources, including data collected through end user interface 360. The application center can further include ingest server(s) 320 for ingesting uploaded media assets from a content provider 330 through a content provider interface 370. The media assets may be stored on the storage array 310. As with the kernel application center 111, the components of the application center 112 can be communicatively linked through a network 340, such as a LAN. The application center can further include an internal interface 350, providing a communication link from the application center to the rest of the CHIMPS. It is through internal interface 350, for example, that media assets stored on storage array 310 can be made available to a kernel application center 111 for services such as transcoding.

[0040] FIG. 4 is a block diagram 400 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Although FIG. 4 further indicates the physical systems in which may execute or store these processes and objects, it will be understood that the processes and objects disclosed may be executed or stored on more than one system, including systems not disclosed in FIG. 4. In other words, the processes and objects shown in FIG. 4 allow for a variety of implementations through one or more of hardware, software, firmware, microcode, etc.

[0041] Media can be ingested into the CHIMP 110 when a content provider 130 uploads a media asset to ingest server(s) 410 in an application center 112 by utilizing a client 405. The client 405 can be a stand-alone application or browser based, for example, and can communicate with ingest server(s) 410 through an application programming interface (API) configured for the ingestion of media assets.

[0042] Ingest server(s) 410 can communicate with devices in the kernel application center 111 executing programs such as kernel server 425 and file replication service 430. The kernel server 425 can be configured to organize the workflow among services such as transcoding 440 file system manager 435, and other services 445 (e.g., analytics, dynamic API, etc.) Upon a particular event, for example, the kernel server can be configured to notify the relevant services of the event, causing the services to process tasks associated with the event.

[0043] The file replication service 430, under direction of the kernel server 425, can coordinate the movement of the media assets between services. For example, retrieving the uploaded media asset from the ingest server(s) 410 and storing it on the file archive 450, or retrieving transcoded media assets from transcoding server(s) 460 and storing them in the media asset origin.

[0044] The data object updater 420 keeps the data object origin 415 up to date in response to any changes in the system. When, for example, a file is uploaded, transcoded, and stored in media asset origin 455, the location and other metadata concerning the transcoded media assets need to be created or updated in the data object origin 415 to ensure an end user device that accesses the object in the data object origin 415 has the correct information regarding the related media asset. Because the data object updater 420 receives updates from the kernel server 425 (which is notified when a transcoded media asset is stored in the media asset origin 455), the system ensures the data objects in the data object origin are constantly up to date.

[0045] The upload of a media asset to the ingest server(s) 410, as described above, can provide an example of how the kernel server 425 may coordinate workflow. For instance, in response to the upload, the ingest server(s) 410 can notify the kernel server 425 that a media asset has been uploaded. The kernel server 425 informs the file replication service 430 of the uploaded media asset, and the file replication service 430 moves the uploaded media asset into the file archive 450 and notifies the kernel server 425 of the move. In response, the kernel server 425 notifies the file replication service 430, the file system manager 435, and the transcoding master 440 of the move. The file replication service 430 then will know it can delete the uploaded media asset

from the ingest server(s) 410, the file system manager 435 will update the file system accordingly, and the transcoding master 440 will notify transcoding service(s) 460 of different transcoding tasks to be performed. The transcoding service(s) 460 can then retrieve the uploaded media asset from the file archive 450 to create transcoded media assets. The transcoding service(s) 460 notify the kernel server 425 once transcoding is complete, and the kernel server 425 relays this information to the file replication service 430. The file replication service 425 then takes the transcoded media assets from the transcoding services 460 and moves them to the media asset origin 455. Once the file replication service 430 notifies the kernel server 425 of the move, the kernel server 425, in turn, notifies the file replication service 430 and the data object updater 420. The data object updater 420 which updates the data object origin 415 accordingly, and the file replication service 430 deletes the transcoded media assets from the transcoding services 460.

[0046] The modular nature of the system enables all tasks associated with an event to be completed quickly. As illustrated in the example above, workflow relating to a particular event, such as a media asset upload, can be spread among the various services simultaneously. Moreover, because the system's modularity enables it to be scaled to accommodate differing hardware capacities, and because the system can be configured to dynamically allocate hardware to different services according to the needs of the system, the speed of completing tasks relating to a particular event can further be increased. For example, a server of the CHIMPS 110 can be configured to dynamically switch its purpose based on external conditions such as load and overall system performance, providing functions such as transcode, upload, metrics collection, application web service, and more, on an as-needed basis.

[0047] Embodiments of such systems may include other systems that manage various requests from end users. For example, a system for dynamic index file generation on media file shunking. Referring to FIG. 5A, shows an embodiment of such a system 500-1. Media may be streamed to end user device 140 through a client 510. As mentioned above, the client 510 can be stand-alone media player, a plug-in, a browser, or other application, which can be executed on a personal computer or other electronic device.

[0048] An index file generator 530, as discussed previously, can be a program instantiated for media streaming to a particular client 510. The index file generator 530 can be executed on a server or other computing device within an application center 113 of the CHIMPS 110. Index

file generated by the index file generator can include a wide variety of information such as starting, ending, and/or run times for media chunks and locations for media chunks. This information can be embedded in a single string of data, such as a URI or a URL. If media includes various sub-streams (e.g., streams with alternative bitrates, captions, alternative languages, etc.) the index file can include data for chunks corresponding to each of the alternative sub-streams, as well as information regarding the bitrate and/or other unique information for each stream. Alternatively or in addition, index files indexing alternative sub-streams may be separate from index files indicating one or more media chunks for streaming.

[0049] It should be understood that the index file can further comprise a wide variety of formats, which can depend on the particular protocol. HTTP streaming may, for example, require index files to comprise one or more of M3U, M3U8, XMTL, and XML-based formats. Of course, other formats can be used in accordance with relevant streaming protocols.

[0050] Table 1 illustrates a simplified example of an generated index file in M3U9 format, indicating chunk of media for streaming. The index file in this example provides a URI for a chunk of media. The URI indicates the chunk is to be generated by dynamic segmentor 550, the chunk being 10 seconds long, starting at 9 seconds into the media file and ending 19 seconds into the media file.

Table 1: Example Index File Contents

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-TARGETDURATION:10
#EXTINF:10,
http://video.example.com/seg9/19/seg1.ts
```

[0051] Referring again to FIG. 5A, the index file generator 530 can also include an indicator within an index file to indicate whether a chunk of media is to be dynamically created. If, for example, it is determined that a requested media asset has not been chunked and that the asset will be chunked dynamically, the index file generator can include the indicator in data corresponding to a chunk of media to be created. The indicator, which can be as simple as including the term “seg” in a URL, will indicate that a requested chunk of media needs to be generated.

[0052] The chunks of media can be generated during media streaming by a dynamic segmentor 550, which can be incorporated into an HTTP service 540. The HTTP service 540, as well as the

media asset origin 560 can be located within a kernel application center 111 of the CHIMPS 110 or, for example, a media asset origin server. The system 500-1 can be configured such that the kernel application center 111 provides dynamically-created chunks of media to a CDN 150 for delivery to client 510. The CDN 150 can store the chunks locally in, for example, a media asset cache 520, thereby forgoing the need to dynamically create a chunk again if the same chunk is requested in the future.

[0053] In sum, the system for dynamic index file generation and media asset chunking 500-1 can, after receiving a request for an index file from a client 510, dynamically generate an index file with an index file generator 530. The index file can, among other things, indicate where a next chunk of media may be located. A client can then request the chunk from the location indicated by the index file, which can comprise a media asset cache 520 in a CDN 150. If the chunk is not found in the media asset cache 520, the cache miss can redirect the request to a segmentor 550 of an HTTP service 540, which can dynamically generate the requested chunk of media by accessing the corresponding media asset in the media asset origin 560. The requested media chunk can then be provided to the CDN 150 for storage in the media asset cache 520 and delivery to the client 510. If the same chunk is requested at a later point in time, the CDN 150 can deliver the chunk from the media asset cache 520, thereby forgoing the need to redirect the request to the segmentor 550 to regenerate the chunk.

[0054] FIG. 5B illustrates an alternative embodiment 500-2 of a system for dynamic index file generation and media file chunking. Rather than utilize a CDN, this embodiment 500-2 includes a media caching server within an application center 112 of the CHIMPS 110. The media caching server can receive chunk requests from and provide the corresponding chunks to a client. It will be understood that such a media caching server(s) or similar device(s) can be located anywhere within the CHIMPS and/or in a system(s) communicatively linked to the CHIMPS.

[0055] FIG. 5C illustrates an embodiment 500-3 of a system for index file generation used in conjunction with a redirector 590. As discussed above, an index file generator 530 may create an index file having one or more URIs or other location information directing the client 510 to one or more redirectors 590. Redirector 590, can then provide the URI of the requested chunk with a redirect, such as a redirect under HTTP status code 302. The URI can be located on a CDN 520 or other location (such as a media caching server 570) and/or dynamically created by the dynamic segmentor 550. It will be understood that there can be any number of redirectors 590,

which can be at any location, including locations of the CHMPS 110 such as the application center 112 (as shown in FIG. 5C) or the kernel application center 111. It also will be understood that the URI or other location information provided by redirector 590 can be generated by or provided to the redirector 590 in any number of ways. The URI can be generated, for example, based on the request received by redirector 590 from client 510. Finally, it will be understood that the CHMPS 110 can be configured to dynamically implement any combination of embodiments 500-1, 500-2, and 500-3, further choosing whether to utilize one or more redirectors based on factors such as, for example, a detected type of client 510.

[0056] Embodiments utilizing one or more redirectors can have several advantages. For example, and not by way of limitation, if a certain client were implemented in such a way that it "reads ahead" to request chunks, it could result in incorrect reporting data. Thus, it would be advantageous to determine which chunk is actually requested by the client. Additionally or alternatively, where chunks are available in various sub-streams with different bitrates, determining the actual requested chunk can be useful in calculating Quality of Service (QoS) metrics. Furthermore, it there may be scenarios in which it is more efficient to create larger index files having many chunks comprising large segments of media, reducing the number of index files required to stream a media asset, and thereby reducing the processing requirements to create the index files. If encryption is used having, for example, a rotating key or a per client key encryption scheme in which a valid key might change during playback of a media asset, it also may be advantageous to incorporate redirector(s) for handling legacy keys for some period of time.

[0057] FIG. 6 illustrates a simplified flowchart of an embodiment of a method 600 for implementing a dynamic index for media streaming. The method 600, which can be executed by the index file generator 530, begins at block 610, where a request for an index file is received from a client 510. According to media streaming protocols contemplated by this embodiment, if data regarding a chunk of media is not provided in an initial index file, the client will continue to request or refresh an index file until it reaches an indicator in the index file that signals the end of a stream. Thus, the method can be assured to receive more than one request for an index file from a client provided that an initial index file does not include an indicator signaling the end of the stream.

[0058] At block 615, the method 600 additionally provides for receiving input from an advertising service. According to some embodiments, this input could be the availability of an advertisement, and can be provided by a service inside or outside the CHIMPS. In other embodiments, the input could come from a service that factors in any of a variety of factors to indicate that a specific advertisement or type of advertisement should be shown. Or that any advertisement should be shown.

[0059] At block 620, a determination is made whether to include an advertisement in the next chunk. According to some embodiments, this determination can be made with or without input from an advertisement service. It should be known that this determination can include the factors used by an advertisement service to provide the input of block 615. Whether the determination includes input from an advertisement service of block 615 or not, the determination can still include factors such as information about an end user collected before or during streaming of the media. This can include behavior of the end user during streaming of the media (as determined, for example, by machine-based logic through beaconing data and/or requested chunks of media provided by a client 510). Factors can also include information regarding the media asset used for streaming (such as type of content or preferred points within the media for an advertisement), preference(s) and/or selection(s) of an end user, when a previous advertisement was shown, time of day, and more. It can further include information regarding the source of a media asset, such as who created and/or provided the asset for viewing by an end user. It will be understood that other embodiments contemplate include secondary media, other than advertisements into the media stream in this manner. Moreover, the secondary media and/or advertisement can be of any length and also may be chunked. Thus, it may be determined that the next chunk includes all, or a select portion, of an advertisement of any specific length.

[0060] An index file is created based on the request as well as the determination of whether media, such as an advertisement, should be streamed, indicated by block 625. As discussed above, the index file can assume a variety of formats and include any amount of information regarding a next chunk of media for streaming. For example, HTTP streaming can utilize index files having the URLs of available chunks of media. Information can be embedded in these URLs to indicate a location to download the corresponding chunk of media, starting point and/or ending point of a chunk of media, an indicator to indicate whether the chunk is to be dynamically

created by a segmentor 550, a location of an advertisement to be streamed, and more. This information is included in the index file and sent to the client at block 630.

[0061] At block 635, reporting data can be created based on information included in the index file. As previously discussed, information included in an index file and/or index file request can indicate the behavior of an end user device 140 during streaming, such as a pause, stop, skip, play, etc. of the media. According to some embodiments, this information can be extracted from requests for an index file and/or providing the requested index file. The information can be gathered in addition to or as a substitute for beaconing data provided by a client 510. Moreover, if beaconing data is provided, the creation of reporting data may be omitted altogether.

[0062] Reporting data can include any amount of information regarding end user behavior, as indicated through index file requests and/or provided index files. This can include a particular action and when it was performed. Additionally or alternatively, the data may be kept in a more rudimentary form, depending on the application or embodiment, indicating the data included in a index file request and/or an index file. This reporting data may be stored in a log file for reporting after streaming and/or transmitted during streaming to a relevant service that collects such metrics.

[0063] As indicated by block 640, the reporting data may be sent to a metrics collector for analytics. A metrics collector, according to certain embodiments, may be an application executed by a server from within the application center 112 in which the index file generator 530 is executed, or it may be executed elsewhere, such as in a kernel application center 111 or in a system outside the CHIMPS 110. Depending on the form of the reporting data, the metrics collector can further process and store the information.

[0064] FIG. 7 illustrates a simplified flowchart of an embodiment of a method for dynamically chunking a media file for streaming 700. This method can be employed by a variety of systems and/or programs. For example, it may be executed by a dynamic segmentor 550 of an HTTP service 540 running on a server located in a kernel application center 111 of the CHIMPS 110.

[0065] The method 700 can begin at block 710, when a request for a chunk of media is received from a CDN 150. As discussed above, this request may be made in response to a cache miss at the CDN 150 and/or because an indicator was included in the request for the chunk of media that the chunk was to be created dynamically. As discussed herein, if the CDN 150 has the requested chunk cached from a prior request, the CDN 150 can provide the requested chunk

and preclude the need to send the request to a dynamic segmentor 150 to generate the chunk. It should be understood that the request may come from sources other than a CDN 150 according to alternative embodiments. One such source includes the media caching server 570 of embodiment 500-2, as shown in FIG. 5B.

[0066] The starting and ending points of a requested chunk of media are then determined at block 715. This information can be included directly in the request or derived from the request, a previous request, and/or other sources. At block 720, the information, as well as information identifying the requested chunk of media, can be used to retrieve all or part of the relevant media asset from a media asset origin 560. The retrieved portion will include at least the relevant media from the starting point to the ending point of the requested chunk of media.

[0067] At block 725, the requested media chunk is generated by converting the relevant portion of the media asset into a deliverable chunk. The media asset, as stored in the media asset origin, may not be chunked; it may be stored in its entirety as a media file (or group of alternative files corresponding to alternative sub-streams). Generating the chunk therefore can require determining the starting and ending points from the retrieved portion of the media asset and converting the resulting segment of media into a deliverable chunk.

[0068] Although the generation of the deliverable chunk may involve transcoding, it may not. The media asset can be stored in a format where transcoding may not be needed, thereby reducing the processing requirements for creating chunks of media during streaming. For example, media assets may be stored such as H.264 or MPEG-4 video format and/or AAC, HE-AAC, or MP3 audio format. According to some streaming protocols, such as some forms of HTTP streaming, chunks of media in these formats would not need transcoding before being wrapped in an MPEG-2 transport stream container format. Instead, such a conversion essentially would require the addition of metadata to create the streaming format from the format of the stored media asset. In other words, generating a deliverable chunk of media may only require identifying the stored media asset, extracting the relevant segment of the media from the media asset, and adding certain metadata in accordance with a container format. This process requires little processing power and can be easily performed on the fly during streaming. Once the deliverable chunk of media is generated, it is sent to the CDN 150 or other requesting entity, at block 730.

[0069] FIG. 8 illustrates a simplified swim lane flowchart describing the interaction of components in a system configured to provide dynamic indexing and chunking for media streaming, according to one embodiment. In this embodiment, a client can send a request for an index file 805, the request received by an index file generator 810. A particular request may be made to initiate the streaming of a media asset, or it may be during streaming. Depending on the streaming protocol, the request may be made while a client plays a chunk of media previously downloaded during streaming.

[0070] The index file generator generates an index file to indicate the next chunk of media 815. As described above, this chunk may include an advertisement, and the index file can include any amount of information about a chunk of media, including information regarding alternative sub-streams for streaming. The dynamic index file generator can include information regarding existing chunks of media, and, when used in conjunction with a dynamic segmenter may also include information regarding chunks of media that may need to be created. As detailed above, if a chunk of media is to be generated dynamically, the index file generator may indicate this by including an indicator in the generated index file, such as in a URL for one or more chunks described within the index file. Once the index file is generated, the index file generator sends the index file 820, which is received by the client 825.

[0071] Alternative embodiments may provide for the generation of index files containing more than a next chunk of media. For example, an index file generator may generate an index file containing information regarding several chunks of media, in which case the chunks of media can be dynamically generated by a dynamic segmenter when requested by the client. The determination of whether to include information regarding more than a next chunk of media can include factors such as whether the index generator is generating reporting data, the desired frequency of such reporting data, and more.

[0072] Using information contained in the index file, the client can then request the next chunk of media 830, and this request can be received by a CDN 835. The CDN then checks to see if the chunk is already stored in the cache 840. If so, the CDN can provide the requested chunk to the client, blocks 845 and 850. The requested chunk may be found in a CDN's cache if the chunk was created and stored in the CDN during preprocessing or if a the chunk was dynamically created and stored in the CDN from an earlier request.

[0073] If the chunk is not found on the CDN, the chunk can be requested of the dynamic segmentor, which receives the request 855 and retrieves the corresponding media asset from an asset origin server 860. As discussed above, the entirety of the relevant media asset does not need to be retrieved as long as at least the portion containing the relevant segment for the requested chunk is retrieved. It will be understood that alternative embodiments can provide for the media asset being stored in a variety of locations accessible, directly or indirectly, to the dynamic segmentor.

[0074] The dynamic segmentor can then generate the requested chunk by converting the retrieved media into a deliverable chunk. That is, the dynamic segmentor converts the retrieved media into an acceptable format for streaming, which can vary depending on the streaming protocol utilized. The dynamic segmentor can then return the requested chunk 870 to the CDN, which can cache the chunk and return it to the client 875. Once the chunk is received by the client 880, the client can play the chunk to an end user.

[0075] It should be noted that the methods, systems, and devices discussed above are intended merely to be examples. It must be stressed that various embodiments may omit, substitute, or add various procedures or components as appropriate. For instance, it should be appreciated that, in alternative embodiments, the methods may be performed in an order different from that described, and that various steps may be added, omitted, or combined. Also, features described with respect to certain embodiments may be combined in various other embodiments. Different aspects and elements of the embodiments may be combined in a similar manner. Also, it should be emphasized that technology evolves and, thus, many of the elements are examples and should not be interpreted to limit the scope of the invention.

[0076] Specific details are given in the description to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these specific details. For example, well-known circuits, processes, algorithms, structures, and techniques have been shown without unnecessary detail in order to avoid obscuring the embodiments. This description provides example embodiments only, and is not intended to limit the scope, applicability, or configuration of the invention. Rather, the preceding description of the embodiments will provide those skilled in the art with an enabling description for implementing embodiments of the invention. Various changes may be

made in the function and arrangement of elements without departing from the spirit and scope of the invention.

[0077] Also, it is noted that the embodiments may be described as a process which is depicted as a flow diagram or block diagram. Although each may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be rearranged. A process may have additional steps not included in the figure. Furthermore, embodiments of the methods may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware, or microcode, the program code or code segments to perform the necessary tasks may be stored in a computer-readable medium such as a storage medium. Processors may perform the necessary tasks.

[0078] Having described several embodiments, it will be recognized by those of skill in the art that various modifications, alternative constructions, and equivalents may be used without departing from the spirit of the invention. For example, the above elements may merely be a component of a larger system, wherein other rules may take precedence over or otherwise modify the application of the invention. Also, a number of steps may be undertaken before, during, or after the above elements are considered. Accordingly, the above description should not be taken as limiting the scope of the invention.

[0079] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0080] The reference to any prior art in this specification is not, and should not be taken as, an acknowledgement or any form of suggestion that the prior art forms part of the common general knowledge in Australia.

WHAT IS CLAIMED IS:

- 1 1. A server for generating a plurality of index files for use in streaming a
2 media file with a network, the server comprising:
3 a memory;
4 an interface for communicating with the network
5 a processor noncommunicatively coupled with the interface and the memory, the
6 processor further configured to execute a plurality of instructions for generating the plurality of
7 index files, the plurality of instructions causing the server to:
8 receive, with the interface, a request for a first index file for use in
9 streaming the media file;
10 determine a first segment of media for streaming, wherein:
11 the determination includes determining whether to
12 include media, other than the media file, in the first segment of
13 media for streaming;
14 the determination further includes determining a
15 starting point and an ending point of the first segment of media for
16 streaming;
17 generate the requested first index file based, at least in part, on the
18 determination of the first segment of media for streaming;
19 provide, with the interface, the requested first index file;
20 receive, with the interface, a request for a second index file for use
21 in streaming the media file;
22 determine a second segment of media for streaming, wherein:
23 the determination includes determining whether to
24 include media, other than the media file, in the second segment of
25 media for streaming;
26 the determination further includes determining a
27 starting point and an ending point of the second segment of media
28 for streaming;
29 generate the requested second index file based, at least in part, on
30 the determination of the second segment of media for streaming; and
31 provide, with the interface, the requested second index file.

2010202740 30 Jun 2010

32 2. The server recited in claim 1 wherein the memory is configured to store
33 information regarding the media file and the determination of the first segment of media for
34 streaming, the second segment of media for streaming, or both, is based, at least in part, on the
35 stored information regarding the media file.

1 3. The server recited in claim 1 wherein the plurality of instructions further
2 include instructions causing the server to create data associated with playback of the media file.

3 4. The server recited in claim 3, wherein the data is based on one or more of
4 when the request for the first index file, the request for the second index file, or
5 both, are received;

6 when the requested first index file, the requested second index file, or both are
7 provided;

8 a length of time to play back the first segment of media for streaming, the second
9 segment of media for streaming, or both;

10 the starting and/or ending point of the first segment of media for streaming, the
11 second segment of media for streaming, or both;

12 whether the request for the second index file has been received within a certain
13 amount of time since receipt of the request for the first index file; and

14 whether the first segment of media for streaming, the second segment of media
15 for streaming, or both, includes media other than the media file.

1 5. The server recited in claim 3 wherein the plurality of instructions further
2 include instructions causing the server to transmit the reporting data over the network.

3 6. The server recited in claim 1 wherein the first segment of media for
4 streaming, the second segment of media for streaming, or both, includes media other than the
5 media file.

6 7. The server recited in claim 6 wherein the media other than the media file
7 includes all or a portion of an advertisement of a certain length.

8 8. The server recited in claim 7 wherein the determination of the first
9 segment of media for streaming, the second segment of media for streaming, or both, is based, at
10 least in part, on one or more of:

4 information regarding an end user;
5 preferences and/or selection of an end user;
6 information based on the behavior of an end user as determined by machine-based
7 logic;
8 information regarding the content of the media file;
9 the time of day;
10 information regarding the availability of media other than the media file; and
11 information regarding the source of the media file.

1 9. A method for generating a plurality of index files for use in
2 communicating a media file with a network, the method comprising:
3 receiving, with the network, a request for a first index file for use
4 in streaming the media file;
5 determining a first segment of media for streaming, wherein:
6 the determination includes determining whether to
7 include media, other than the media file, in the first segment of
8 media for streaming;
9 the determination further includes determining a
10 starting point and an ending point of the first segment of media for
11 streaming;
12 generating the requested first index file based, at least in part, on
13 the determination of the first segment of media for streaming;
14 providing, with the network, the requested first index file;
15 receiving, with the network, a request for a second index file for
16 use in streaming the media file;
17 determining a second segment of media for streaming, wherein:
18 the determination includes determining whether to
19 include media, other than the media file, in the second segment of
20 media for streaming;
21 the determination further includes determining a
22 starting point and an ending point of the second segment of media
23 for streaming;

34 generating the requested second index file based, at least in part,
 35 on the determination of the second segment of media for streaming; and
 36 providing, with the network, the requested second index file.

1 10. The method recited in claim 9 wherein the generating the requested first
 2 index file, the requested second index file, or both, includes generating an indicator, wherein the
 3 indicator indicates that the first segment of media for streaming, the second segment of media for
 4 streaming, or both, are to be created dynamically.

1 11. The method recited in claim 9 further comprising creating reporting data.

1 12. The method recited in claim 11, wherein the reporting data is based on one
 2 or more of:

3 when the request for the first index file, the request for the second index file, or
 4 both, are received;

5 when the requested first index file, the requested second index file, or both are
 6 provided;

7 a length of time to play back the first segment of media for streaming, the second
 8 segment of media for streaming, or both;

9 the starting and/or ending point of the first segment of media for streaming, the
 10 second segment of media for streaming, or both;

11 whether the request for the second index file has been received within a certain
 12 amount of time since receipt of the request for the first index file; and

13 whether the first segment of media for streaming, the second segment of media
 14 for streaming, or both, includes media other than the media file.

1 13. The method recited in claim 11 further comprising transmitting the
 2 reporting data over the network.

1 14. The method recited in claim 11 wherein the determining of the first
 2 segment of media for streaming, the second segment of media for streaming, or both, is based, at
 3 least in part, on the reporting data.

1 15. The method for generating an index file for use in communicating a media
 2 file over a network recited in claim 9 wherein the determining of the first segment of media for

3 streaming, the second segment of media for streaming, or both, is based, at least in part, on one
 4 or more of:
 5 information regarding an end user;
 6 preferences and/or selection of an end user;
 7 information based on the behavior of an end user as determined by machine-based
 8 logic;
 9 information regarding the content of the media file;
 10 the time of day;
 11 information regarding the availability of media other than the media file; and
 12 information regarding the source of the media file.

1 16. A method for providing index files and Universal Resource Indicators
 2 (URIs) to indicate chunks of a media file for streaming, the method comprising:
 3 receiving a request for an index file, the request generated by a media streaming
 4 application;
 5 determining a chunk of media for streaming, wherein:
 6 the determining includes determining whether to include media
 7 other than the media file in the chunk of media for streaming;
 8 the determining further includes determining a starting point and
 9 an ending point of the chunk of media for streaming;
 10 generating the requested index file based, at least in part, on the determining of
 11 the chunk of media for streaming;
 12 providing the requested index file to the media streaming application;
 13 receiving a request, generated by a media streaming application, for the chunk of
 14 media for streaming, wherein the receipt of the request for the chunk of media for streaming is
 15 based, at least in part, on the index file;
 16 determining a URI for the chunk of media, wherein the URI is based, at least in
 17 part, on the request for the chunk of media for streaming; and
 18 providing the URI to the media streaming application.

1 17. The method recited in claim 16 wherein the URI includes an indicator
 2 indicating the chunk of media for streaming is to be generated dynamically.

1 18. The method recited in claim 16 further comprising generating data
2 associated with the media streaming application.

1 19. The method recited in claim 18, wherein the data is based on one or more
2 of:
3 when the request for the index file is received;
4 when the requested index file is provided;
5 a length of time to play back the chunk of media for streaming;
6 the starting and/or ending point of the chunk of media;
7 the requested chunk of media; and
8 whether the chunk of media for streaming includes media other than the media
9 file.

1 20. The method recited in claim 16 further comprising transmitting the data
2 over a network.

1 21. The method recited in claim 16 wherein the determination of the chunk of
2 media for streaming is based, at least in part, on one or more of:
3 information regarding an end user;
4 preferences and/or selection of an end user;
5 information based on the behavior of an end user as determined by machine-based
6 logic;
7 information regarding the content of the media file;
8 the time of day;
9 information regarding the availability of media other than the media file; and
10 information regarding the source of the media file.

1 22. A server for generating a plurality of index files for use in streaming a
2 media file with a network substantially as herein described.

1 23. A method for generating a plurality of index files for use in
2 communicating a media file with a network substantially as herein described.

2010202740 30 Jun 2010

- 1 24 A method for providing index files and Universal Resource Indicators
2 (URIs) to indicate chunks of a media file for streaming substantially as herein described.

2010202740 30 Jun 2010

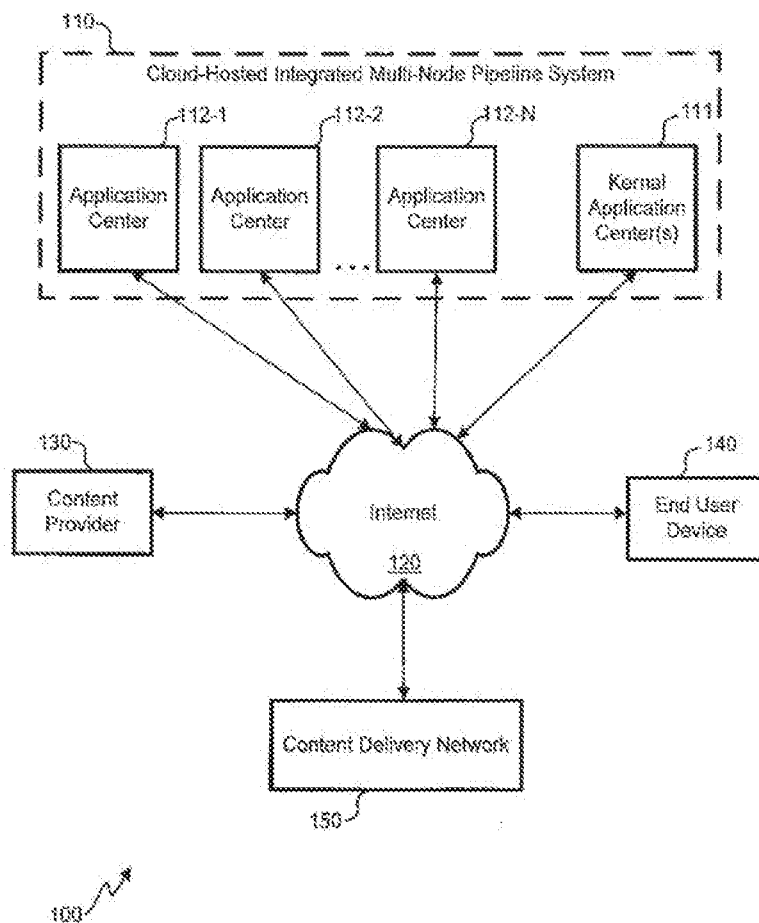


FIG. 1

2010202740 30 Jun 2010

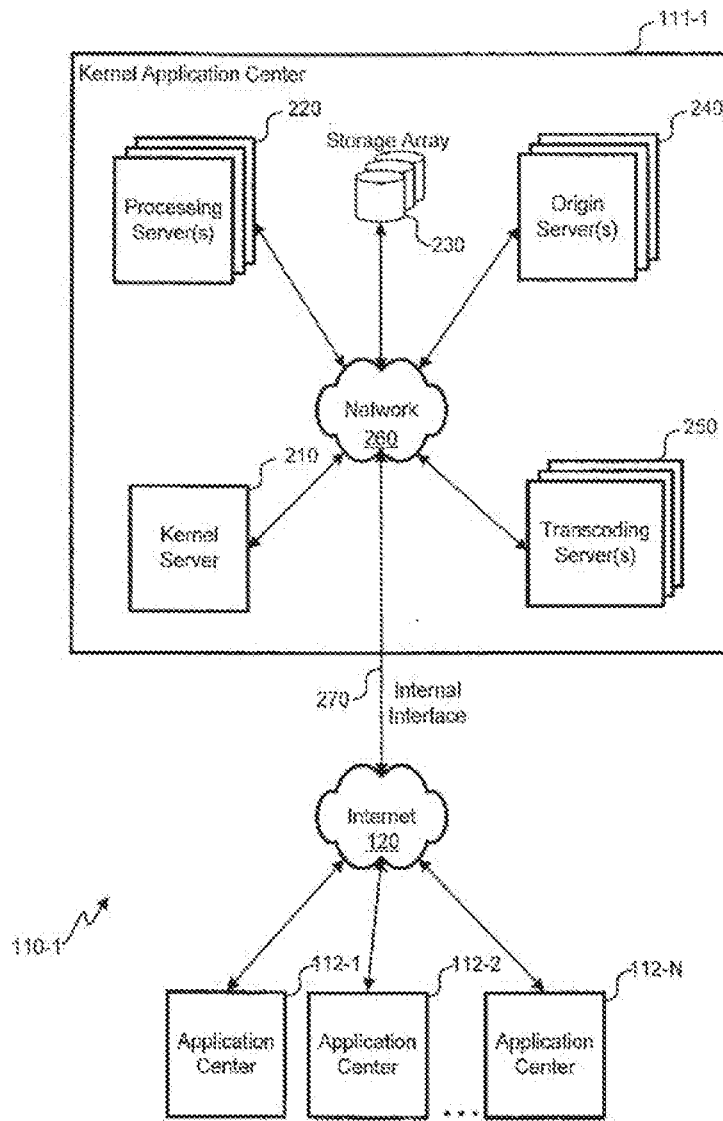


FIG. 2A

2010202740 30 Jun 2010

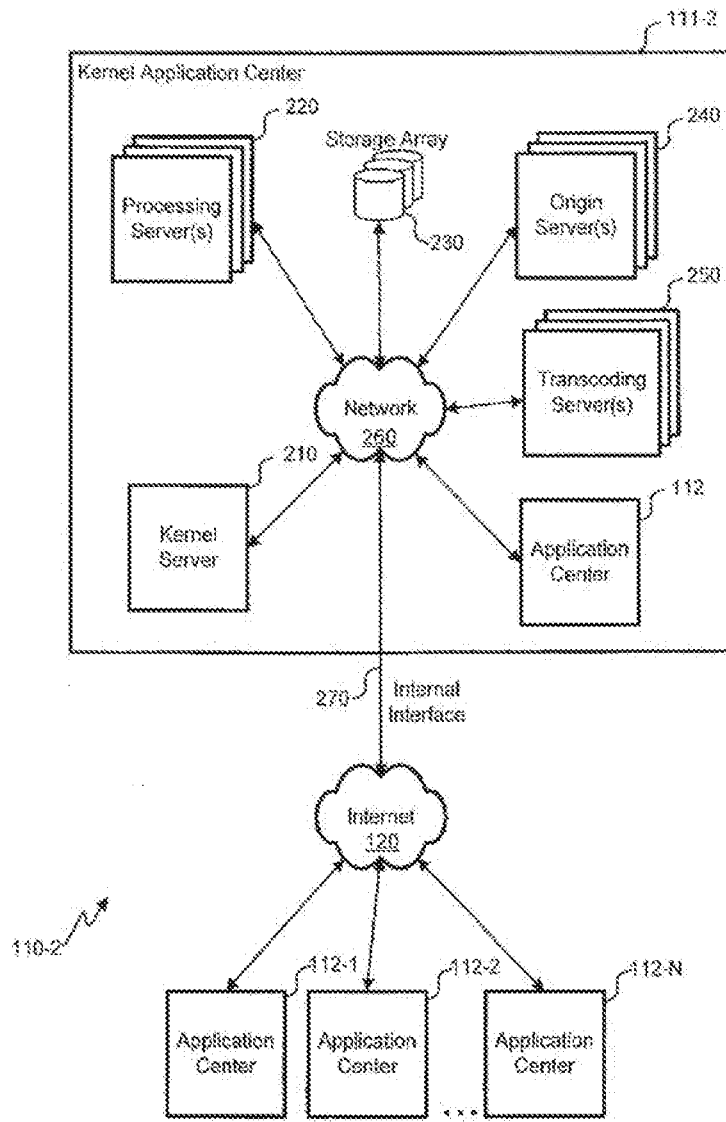


FIG. 2B

2010202740 30 Jun 2010

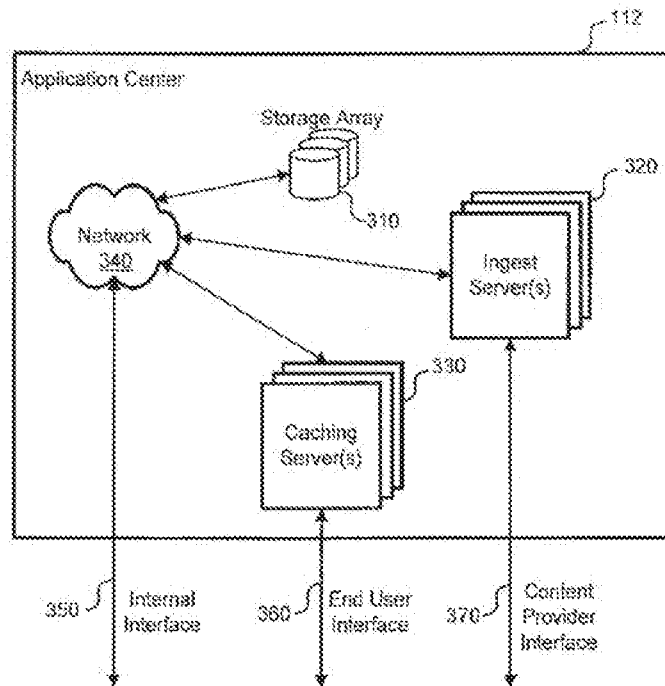


FIG. 3

2010202740 30 Jun 2010

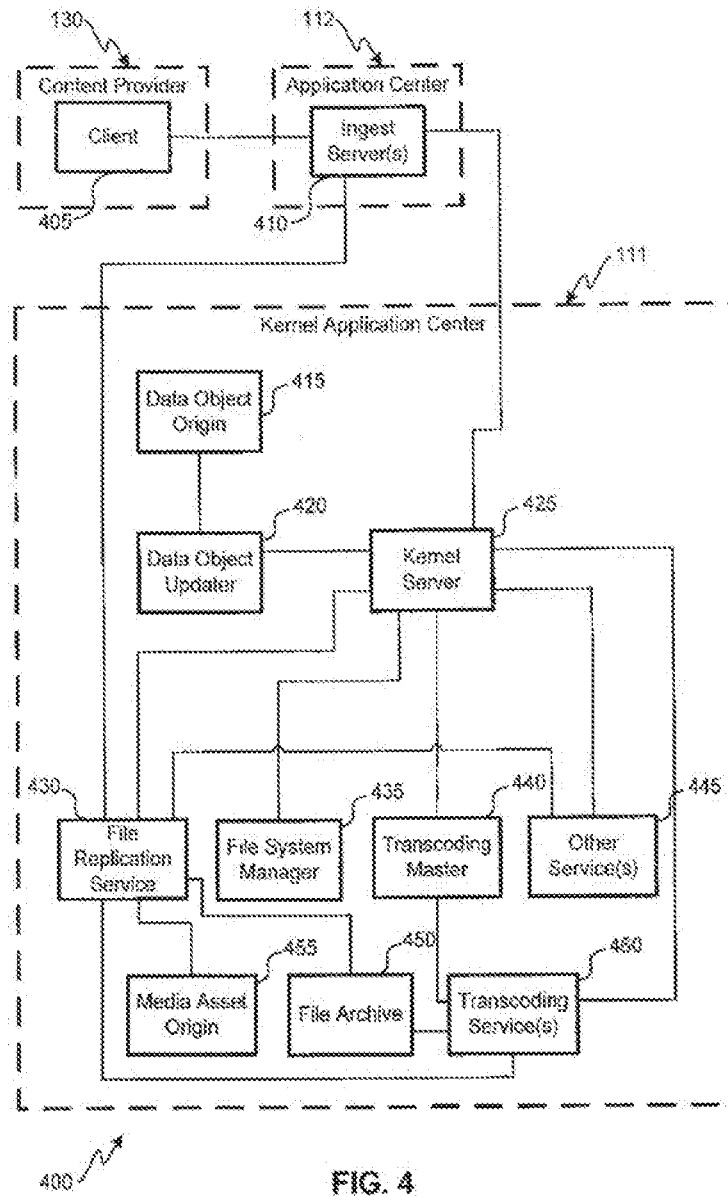


FIG. 4

2010202740 30 Jun 2010

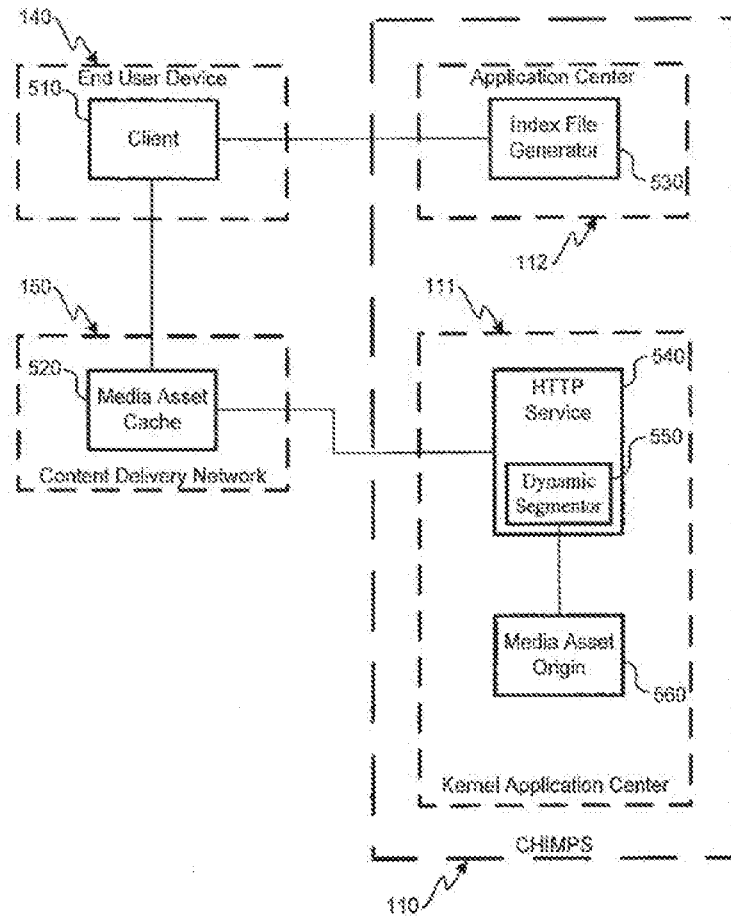


FIG. 5A

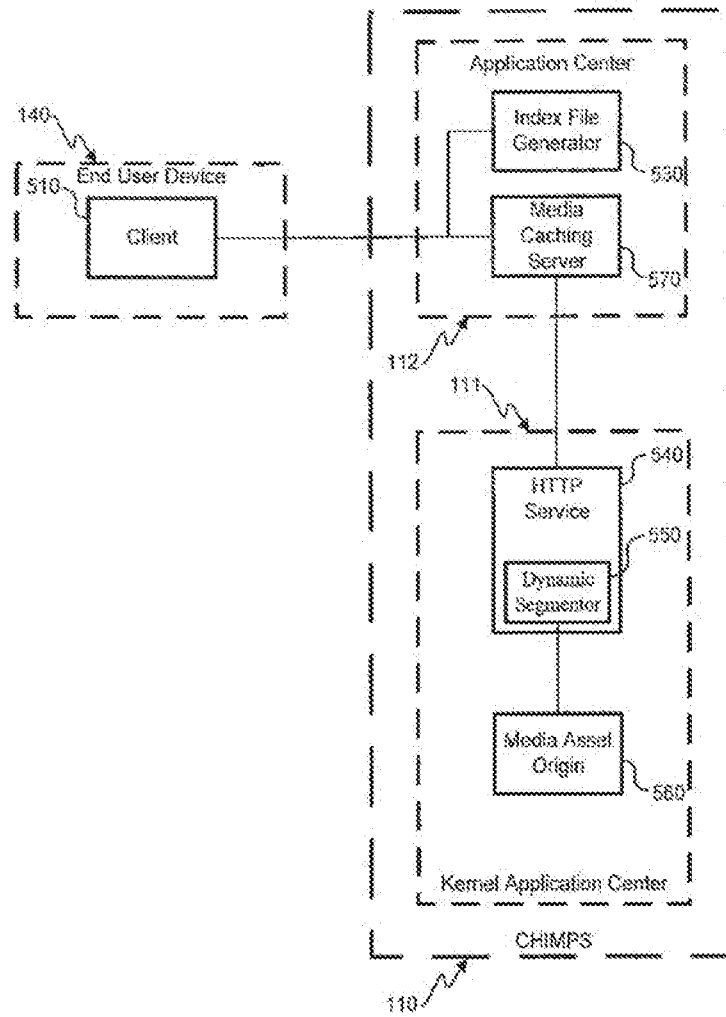
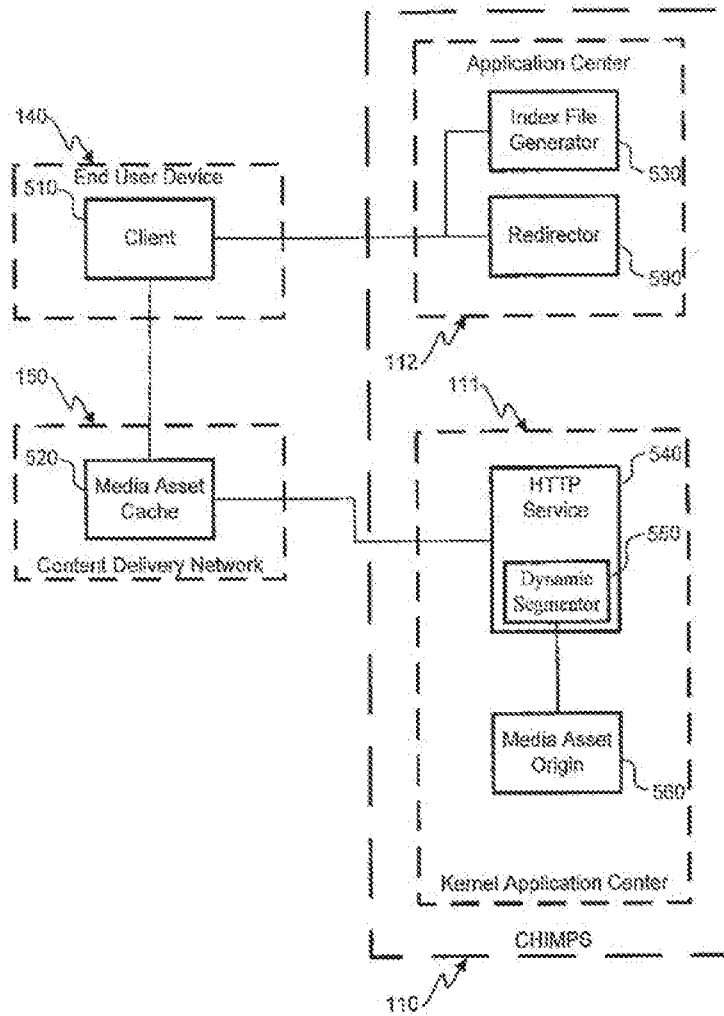


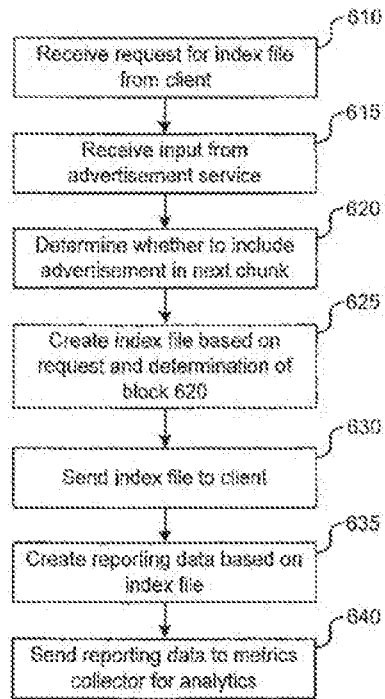
FIG. 5B



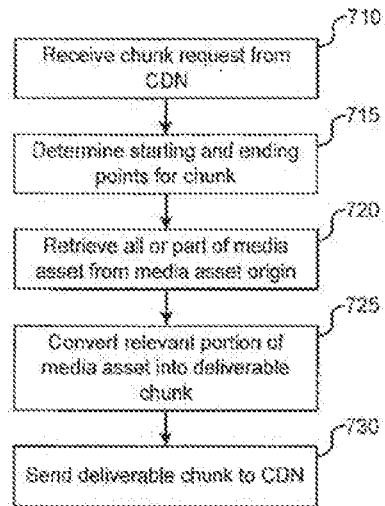
500-3

FIG. 5C

2010202740 30 Jun 2010



2010202740 30 Jun 2010



700

FIG. 7

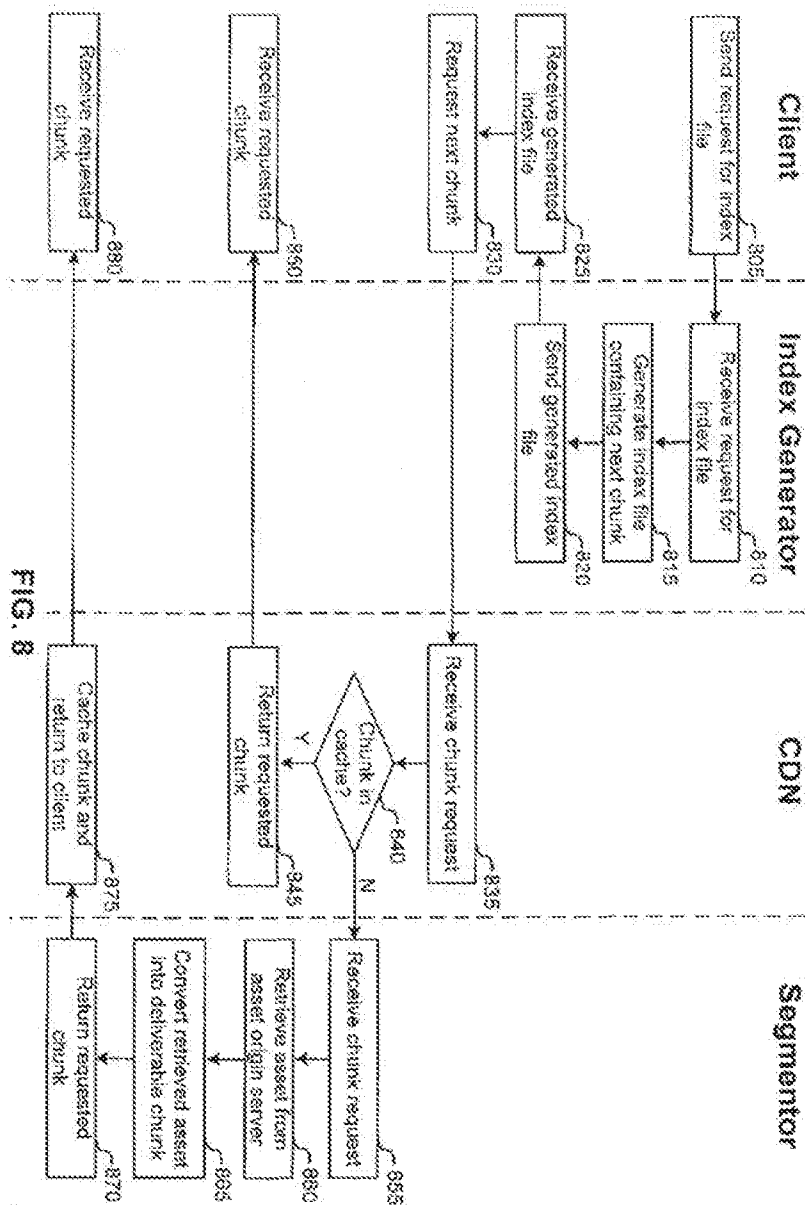


FIG. 8

(19) AUSTRALIAN PATENT OFFICE

(54) Title
Dynamic chunking for media streaming

(51)⁶ International Patent Classification(s)
G06F 17/30 (2006.01) ^{6BHAU} **G06F**
G06F 15/16 (2006.01) 15/16
G06F 17/30 20060101ALI2010070
20060101AFI2010070 ^{6BHAU}

(21) Application No: 2010202741 (22) Application Date: 2010 .06 .30

(43) Publication Date : 2010 .12 .23
(43) Publication Journal Date : 2010 .12 .23

(71) Applicant(s)
Unicorn Media, Inc.

(72) Inventor(s)
McGowan, Albert J.; Carls, Richard L.

(74) Agent/Attorney
Pizzeys Patent and Trade Mark Attorneys, Level 2 Woden Plaza Offices Woden Town Square,
Woden, ACT, 2606

(56) Related Art
US 2006/114985 A1
US 5612742 A
US 2010/0118973 A1
US 2009/0022172 A1
GB 2462732 A
US 2002/0122430 A1
CN 101282478 A

2010202741 30 Jun 2010

ABSTRACT OF THE DISCLOSURE

Systems and methods for chunking a media file for streaming over a network are disclosed. These systems and methods provide for receiving requests for chunks of a media file and responding to these requests by generating the requested chunks dynamically. The chunks can then be provided to the requesting entity. These systems and methods, which can be utilized with a dynamic file index generator, enable a high degree of flexibility in streaming chunked media files and preclude the need to chunk the media files and store the chunks prior to streaming.

2010202741 30 Jun 2010

Regulation 3.2

AUSTRALIA

Patents Act 1990

COMPLETE SPECIFICATION
STANDARD PATENT

APPLICANT: **UNICORN MEDIA, INC.**

Invention Title: **DYNAMIC CHUNKING FOR MEDIA
STREAMING**

The following statement is a full description of this invention, including the best method of performing it known to me:

DYNAMIC CHUNKING FOR MEDIA STREAMING

BACKGROUND OF THE INVENTION

- 5 [0001] This disclosure relates in general to cloud-based computer processing and, but not by way of limitation, to indexing and chunking media files for use in media streaming.
- 10 [0002] The delivery of media over networks such as the Internet can be accomplished in many ways, including progressive downloading or streaming. Streaming is often preferred to progressive downloading because it offers additional features such as content protection and the ability to seek to undownloaded parts of a media file. The process of preparing a media file for streaming typically involves "chunking" the file, or dividing files up into smaller segments for delivery. Information including where chunks may be accessed can be stored in an index file. This index file can be delivered to a client, such as a media player application, for use in streaming.
- 15 [0003] The processes of chunking and indexing files for streaming present challenges to a content delivery network or content provider desiring to host media files for streaming. For example, a significant amount of preprocessing is required to prepare media for streaming. Media content may be encoded into several different files to accommodate several different sub-streams. Each of these files typically are chunked, stored, and indexed before the media content is available for streaming. This preprocessing leaves little flexibility for inserting an advertisement during streaming and can require a large amount of storage space to store the preprocessed chunks of media.
- 20 [0004] The inventions embodied herein address these and other problems.

BRIEF SUMMARY OF THE INVENTION

- 25 [0005] Systems and methods for chunking a media file for streaming over a network are disclosed. These systems and methods may be part of a larger media servicing network that can be used to, among other things, process uploaded media content, provide it for streaming, and collect metric information regarding the streaming. The disclosed systems and methods provide for receiving requests for chunks of a media file from, for example, a content delivery network (CDN), and responding to these requests by generating the requested chunks dynamically and
- 30

providing the chunks to the requesting entity. These systems and methods, which can be utilized with a dynamic file index generator, enable a high degree of flexibility in streaming chunked media files and preclude the need to chunk the media files and store the chunks prior to streaming.

5 [0006] In other words, systems and methods disclosed herein provide for communicating a media file having a plurality of segments by receiving a request for a first segment of the media file and determining a starting point and an ending point of the requested first segment based, at least in part, on the request. A first portion of the media file having a first format, the first portion including at least the requested first segment of the media file, is retrieved. The
10 requested first segment is generated, having a second format and being generated, at least in part, from the first portion of the media file. The requested first segment of the media file is then provided. A similar process is followed upon receiving a request for a second segment of the media file.

15 [0007] Embodiments of the present invention can further allow for the first and second portions of the media file being stored in memory can comprising the same portion of the media file. Moreover, the requested first segment of the media file and the requested second segment of the media file can comprise contiguous segments of the media file. Embodiments further allow for the first format including one or more of an audio compression format, a video compression format, and/or an image compression format. The second format can include a
20 media container format.

[0008] Embodiments of the present invention additionally contemplate the process described above being executed during playback of the media file by a device communicatively linked to the network. The segments for a first media file may be of a different length than the segments of a second media file, and the requests for segments may be received from a Content Delivery
25 Network (CDN).

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present disclosure is described in conjunction with the appended figures:

[0010] FIG. 1 illustrates a block diagram of a media servicing system.

30 [0011] FIG. 2A illustrates a block diagram of an embodiment of a kernel application center connected with application centers.

[0012] FIG. 2B illustrates a block diagram of an alternative embodiment of a kernel application center.

[0013] FIG. 3 illustrates a block diagram of an embodiment of an application center.

5 [0014] FIG. 4 illustrates a block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion.

[0015] FIG. 5A illustrates a simplified block diagram of an embodiment of a system configured to provide dynamic indexing and chunking for media streaming.

[0016] FIG. 5B illustrates a simplified block diagram of another embodiment of a system configured to provide dynamic indexing and chunking for media streaming.

10 [0017] FIG. 5C illustrates a simplified block diagram of another embodiment of a system configured to provide dynamic indexing and chunking for media streaming, utilizing a redirector.

[0018] FIG. 6 illustrates a simplified flowchart of an embodiment of a method for implementing a dynamic index for media streaming.

15 [0019] FIG. 7 illustrates a simplified flowchart of an embodiment of a method for dynamically chunking a media file for streaming.

[0020] FIG. 8 illustrates a simplified swim lane flowchart describing the interaction of components in a system configured to provide dynamic indexing and chunking for media streaming, according to one embodiment.

20 [0021] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

25

DETAILED DESCRIPTION OF THE INVENTION

[0022] The ensuing description provides preferred exemplary embodiment(s) only, and is not intended to limit the scope, applicability or configuration of the disclosure. Rather, the ensuing description of the preferred exemplary embodiment(s) will provide those skilled in the art with

an enabling description for implementing a preferred exemplary embodiment. It is understood that various changes may be made in the function and arrangement of elements without departing from the spirit and scope as set forth in the appended claims.

5 [0023] The increased availability of media content over data communications networks such as the Internet has mirrored the increased bandwidth for these networks. Because media has recently taken a more prominent role in data communications, the distribution of media and the data associated with such distribution has become increasingly important, particularly to media content providers. Media streaming has become a widely-used method of media distribution, but the preprocessing associated with streaming can be burdensome. Certain protocols, including forms of Hypertext Transfer Protocol (HTTP) streaming, require chunking and storing media assets, and generating a corresponding index files. These requirements can deprive a content provider of the ability to dynamically insert additional media such as advertisements into a media stream, and can consume a large amount of storage space to store chunks of media for a media asset, including chunks for any alternative sub-streams (e.g., streams with alternative bitrates, captions, alternative languages, etc.). Certain systems and methods can be utilized, however, to introduce the desired functionality back into the system.

20 [0024] A traditional approach to preprocessing media for streaming involves chunking and storing media assets, then creating corresponding index files to indicate where chunks may be located to download for streaming. Streaming protocols often provide for frequently updating an index file for instances where the corresponding media is frequently updated, such as during live streaming. Thus, an index file does not need to contain all chunks for a requested media asset. In addition, because media files are frequently stored in a format that requires little additional processing to chunk, the chunks can be created in real time, during the streaming of a media file. The systems and methods disclosed herein take advantage of these features to enable dynamic index file creation and dynamic media file chunking.

25 [0025] For instance, rather than preprocess media assets for streaming by chunking and indexing all files with relevant sub-streams prior to streaming the media, a server can dynamically create and update an index file during streaming. The dynamically-created index file can contain information regarding a next chunk of media in the various available sub-streams. The next chunk of media may not be cached at a location specified in the index file, in which case a chunk may be dynamically created by pulling all or part of the media file of interest

from a media file origin, chunking it, and making it available for download. The chunk also may be cached, thereby eliminating the need to create the chunk again if it is requested at some later time.

[0026] Because the chunks are created during streaming, a content provider and/or media distributor can have more information and control during the streaming process. Rather than generate single index file for a given media asset, an instance of the index file generator may be created at the beginning of the media streaming to provide individualized media content to a particular end user and unique information regarding the streaming session to a content provider. The file index generator can vary the length of each chunk by, for example, indicating starting and ending points in the index file. Thus, the file index generator may determine a uniform chunk length for a media asset, varying the length of the chunks for different media assets, or the file index generator may adjust the length of the chunks within a single media asset. The index file generator can further insert additional media, such as an advertisement, at any time during the streaming by specifying the location of the additional media in the index file. The determination to insert advertisements can be based on any information, including data collected during the streaming session.

[0027] As the index file generator receives requests for and generates index files, it can further gather data regarding the streaming session for reporting to a content provider. Content providers often rely on beaconing data collected from media player applications to determine when an end user stops, plays, pauses, skips, etc., the streaming media content. Such information can be vital in determining the value of the media.

[0028] Because not all media player applications provide this beaconing data, the data gathered by the index file generator can serve as a substitute for or complement to the beaconing data. For example, if a request is made for a chunk that does not immediately follow a previously-requested chunk, a skip was made. If the amount of time elapsed between a previous request and a subsequent request exceeds the time for playback of the previously-requested chunk, a pause was made. If a request is not received within a certain time since a prior request, it can be determined that a stop was made.

[0029] As illustrated above, the state of a client may be determined from a variety of factors. This can include when the request for the index file is received, when the index file is provided, a length of time to play back the segment of media for streaming, and/or the starting and/or ending

point of the segment of media for streaming. The determined state of a client may also be based on whether the request for the index file has been received within a certain amount of time since receipt of a previous request for an index file, whether the segment of media for streaming includes media other than the media file, and more. The state of a client and/or the data from which it was determined, may be used to create reporting data to serve as a substitute or complement to beaconing data from a client media player application. Because the index file generator can determine the length of the chunks, it therefore can determine the frequency of subsequent index file requests and the resolution of the reporting data based on the requests. The index file generator may log the reporting data and/or transmit the reporting data over a network during streaming.

[0030] The determined state of a client may be used by the index file generator and/or other services for various purposes. For example, it may be used in behavioral advertisement targeting and enforcement of session advertisement behavior, adjusting advertisement content and playback based on the behavior of a user as determined by the state of a client. The state of a client further may be used to support resume features on a per client basis, allowing a user to continue playback of a media asset from a point at which the user had previously stopped playback. The state of a client also may be used to support individual encryption keys in an encryption scheme and allow the index file generator to return secure URLs (e.g., time expiring or Internet Protocol (IP) allowed) for chunks to support functions such as payment services.

[0031] Additionally or alternatively, the tasks of generating the index file and providing a location a requested chunk can be split up, thereby enabling the system to determine which chunks are actually requested. For example, a system may be configured to dynamically create an index file having links to one or more redirectors on the system. These redirectors can be configured to issue the location of the chunk, which can be created dynamically. The redirectors can further determine which chunk is actually requested, thereby enabling, among other things, calculation of Quality of Service (QOS) metrics, an increase the accuracy of reporting data, a decrease the frequency of index file generation if efficient to do so, and the ability to more easily handle keys of an encryption scheme.

[0032] While the above embodiments may be implemented in a variety of different systems, some particular embodiments may be implemented as part of a media service system. FIG. 1 is a block diagram illustrating a media servicing system 100, according to some embodiments of

the present invention. The system may deliver media content to the end user device 140 through a network such as the Internet 120. The end user device 140 can be one of any number of devices configured to receive media over the Internet 120, such as a mobile phone, tablet computer, personal computer, portable media device, etc. A media asset provided by a content provider 130 can be processed and indexed by cloud-hosted integrated multi-node pipelining system (CHIMPS) 110, and further stored on content delivery network (CDN) 150. Additionally or alternatively, the CHIMPS 110 may also be adapted to store the media asset.

[0033] The media servicing system further enables a content provider 130 or other entity to gather information regarding user behavior during media playback. For example, a content provider 130 can be provided with data indicating that end users tend to stop watching a video at a certain point in playback, or that users tended to follow links associated with certain advertisements displayed during playback. With this data, a content provider 130 can adjust factors such as media content, advertisement placement and content, etc., to increase revenue associated with the media content and provide the end user device 140 with a more desirable playback experience.

[0034] End user device 140 can request a media asset to stream with a client program executed by the end user device 140. The client program can be, for example, a media player, browser, or other application adapted to request and/or play media assets. In response to a request for a media asset, the CHIMPS 110 can utilize any number of application centers 112 and/or kernel application center(s) 111 to provide the client program with a data object concerning the requested media asset. The data object can include information about the media asset, including where the media asset can be located, such as within the CDN 150 or within the CHIMPS 150 itself. Location information may be provided by Universal Resource Indicator (URI), a Universal Resource Locator (URL) or other indicator. During playback of the media asset, the CHIMPS 150 can collect data regarding the playback through beaconing provided by a client program executed by the end user device 140 and/or indexing service from within the CHIMPS and/or CDN. The CHIMPS 150 can subsequently provide the data and/or any analytics information derived from the data to the content provider 130.

[0035] FIG. 2A is a block diagram illustrating an embodiment of a kernel application 111-1 center connected with application centers from within the CHIMPS 110-1. The kernel application center 111-1 and application centers 112 can be geographically distant and can be

connected via the Internet 120, wide area network (WAN), and/or other data communication network. Because application centers can be geographically separated, DNS services (not shown) can be used to allow an end user device 140 to connect to the nearest available application center 112. The kernel application center 111-1 can connect with application centers 112 within the CHIMPS 110-1 through an internal interface 270, thereby enabling the application centers 112 access to the various components within the kernel application center 111-1.

[0036] Components within the kernel application center 111-1 can communicate through network 260 such as a local area network (LAN) and can include one or more origin servers 240 and a storage array 230 with which data objects and/or media assets may be stored and distributed. The storage array 230 may also be utilized by services running on processing server(s) 220 and/or transcoding server(s) 250 that may require temporary or long-term storage. Kernel server 210 can utilize processing server(s) 220, transcoding server(s) 250 to provide various functional capabilities to the CHIMPS 110.

[0037] For example, as described in more detail below, the CHIMPS 110-1 can provide transcoding service for media assets provided by a content provider 130 for syndication. Such a service can allow a content provider 130 to upload a media asset to an application center 112, after which the application center 112 would notify the kernel server 210 that the media asset has been uploaded. The kernel server can then notify services running on the processing server(s) 220 of the upload. These services can utilize transcoding server(s) to transcode the media asset, which can then be moved to a CDN and/or stored locally by storage array 230 and origin server(s) 240. Services running on the processing server(s) 220 can also update the associated data object stored by the storage array 230 and origin server(s) 240.

[0038] FIG. 2B is a block diagram illustrating an alternative embodiment of a kernel application center 111-2. In addition to the components of the embodiment of FIG. 2A, this embodiment incorporates an application center 112 within the kernel application center 111-2. The application center 112 incorporated within kernel application center 111-2 may be located at or near the other components of the kernel application center 111-2, and can be communicatively connected to the other components via network 260. The incorporated application center 112 can therefore have faster access to kernel application center functionality because it does not need to communicate over long distances. In consideration of this advantage, it will be

understood that the CHIMPS 110 can include multiple kernel centers with one or more application centers incorporated therein. Additionally or alternatively, components of the kernel application center may be incorporated into one or more application centers 112 in the CHIMPS 110 to provide quicker access to certain functionality.

5 [0039] FIG. 3 is a block diagram illustrating an embodiment of an application center 112. The application center 112 can include caching server(s) 330 and a storage array 310 for storing and distributing data objects of media assets requested by end user devices through end user interface 360. Caching server(s) 330 and storage array 310 can also be used to collect, process, and/or store metrics information from beaconing data, media chunk requests, and/or other data sources, including data collected through end user interface 360. The application center can further include ingest server(s) 320 for ingesting uploaded media assets from a content provider 130 through a content provider interface 370. The media assets may be stored on the storage array 310. As with the kernel application center 111, the components of the application center 112 can be communicatively linked through a network 340, such as a LAN. The application center can further include an internal interface 350, providing a communication link from the application center to the rest of the CHIMPS. It is through internal interface 350, for example, that media assets stored on storage array 310 can be made available to a kernel application center 111 for services such as transcoding.

20 [0040] FIG. 4 is a block diagram 400 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Although FIG. 4 further indicates the physical systems in which my execute or store these processes and objects, it will be understood that the processes and objects disclosed may be executed or stored on more than one system, including systems not disclosed in FIG. 4. In other words, the processes and objects shown in FIG. 4 allow for a variety of implementations through one or more of hardware, software, firmware, microcode, etc.

25 [0041] Media can be ingested into the CHIMPS 110 when a content provider 130 uploads a media asset to ingestion server(s) 410 in an application center 112 by utilizing a client 405. The client 405 can be a stand-alone application or browser based, for example, and can communicate with ingest server(s) 410 through an application programming interface (API) configured for the ingestion of media assets.

[0042] Ingest server(s) 410 can communicate with devices in the kernel application center 111 executing programs such as kernel server 425 and file replication service 430. The kernel server 425 can be configured to organize the workflow among services such as transcoding 440, file system manager 435, and other services 445 (e.g., analytics, dynamic API, etc.) Upon a particular event, for example, the kernel server can be configured to notify the relevant services of the event, causing the services to process tasks associated with the event.

[0043] The file replication service 430, under direction of the kernel server 425, can coordinate the movement of the media assets between services. For example, retrieving the uploaded media asset from the ingest server(s) 410 and storing it on the file archive 450, or retrieving transcoded media assets from transcoding server(s) 460 and storing them in the media asset origin.

[0044] The data object updater 420 keeps the data object origin 415 up to date in response to any changes in the system. When, for example, a file is uploaded, transcoded, and stored in media asset origin 455, the location and other metadata concerning the transcoded media assets need to be created or updated in the data object origin 415 to ensure an end user device that accesses the object in the data object origin 415 has the correct information regarding the related media asset. Because the data object updater 420 receives updates from the kernel server 425 (which is notified when a transcoded media asset is stored in the media asset origin 455, the system ensures the data objects in the data object origin are constantly up to date.

[0045] The upload of a media asset to the ingest server(s) 410, as described above, can provide an example of how the kernel server 425 may coordinate workflow. For instance, in response to the upload, the ingest server(s) 410 can notify the kernel server 425 that a media asset has been uploaded. The kernel server 425 informs the file replication service 430 of the uploaded media asset, and the file replication service 430 moves the uploaded media asset into the file archive 450 and notifies the kernel server 425 of the move. In response, the kernel server 425 notifies the file replication service 430, the file system manager 435, and the transcoding master 440 of the move. The file replication service 430 then will know it can delete the uploaded media asset from the ingest server(s) 410, the file system manager 435 will update the file system accordingly, and the transcoding master 440 will notify transcoding service(s) 460 of different transcoding tasks to be performed. The transcoding service(s) 460 can then retrieve the uploaded media asset from the file archive 450 to create transcoded media assets. The transcoding service(s) 460 notify the kernel server 425 once transcoding is complete, and the kernel server

425 relays this information to the file replication service 430. The file replication service 425 then takes the transcoded media assets from the transcoding services 460 and moves them to the media asset origin 455. Once the file replication service 430 notifies the kernel server 425 of the move, the kernel server 425, in turn, notifies the file replication service 430 and the data object updat-
5 er 420. The data object updater 420 which updates the data object origin 415 accordingly, and the file replication service 430 deletes the transcoded media assets from the transcoding services 460.

[0046] The modular nature of the system enables all tasks associated with an event to be completed quickly. As illustrated in the example above, workflow relating to a particular event, such as a media asset upload, can be spread among the various services simultaneously.
10 Moreover, because the system's modularity enables it to be scaled to accommodate differing hardware capacities, and because the system can be configured to dynamically allocate hardware to different services according to the needs of the system, the speed of completing tasks relating to a particular event can further be increased. For example, a server of the CHIMPS 110 can be
15 configured to dynamically switch its purpose based on external conditions such as load and overall system performance, providing functions such as transcode, upload, metrics collection, application web service, and more, on an as-needed basis.

[0047] Embodiments of such systems may include other systems that manage various requests from end users. For example, a system for dynamic index file generation and media file
20 chunking. Referring to FIG. 5A, shows an embodiment of such a system 500-1. Media may be streamed to end user device 140 through a client 510. As mentioned above, the client 510 can be stand-alone media player, a plug-in, a browser, or other application, which can be executed on a personal computer or other electronic device.

[0048] An index file generator 530, as discussed previously, can be a program instantiated for
25 media streaming to a particular client 510. The index file generator 530 can be executed on a server or other computing device within an application center 112 of the CHIMPS 110. Index files generated by the index file generator can include a wide variety of information such as starting, ending, and/or run times for media chunks and locations for media chunks. This information can be embedded in a single string of data, such as a URI or a URL. If media
30 includes various sub-streams (e.g., streams with alternative bitrates, captions, alternative languages, etc.) the index file can include data for chunks corresponding to each of the

alternative sub-streams, as well as information regarding the bitrate and/or other unique information for each stream. Alternatively or in addition, index files indicating alternative sub-streams may be separate from index files indicating one or more media chunks for streaming.

5 [0049] It should be understood that the index file can further comprise a wide variety of formats, which can depend on the particular protocol. HTTP streaming may, for example, require index files to comprise one or more of M3U, M3U8, XML, and XML-based formats. Of course, other formats can be used in accordance with relevant streaming protocols.

10 [0050] Table 1 illustrates a simplified example of an generated index file in M3U9 format, indicating chunk of media for streaming. The index file in this example provides a URI for a chunk of media. The URI indicates the chunk is to be generated by dynamic segmentor 550, the chunk being 10 seconds long, starting at 9 seconds into the media file and ending 19 seconds into the media file.

Table 1: Example Index File Contents

<pre>#EXTM3U #EXT-X MEDIA-SEQUENCE:1 #EXT-X-TARGETDURATION:10 #EXTINF:10, http://video.example.com/seg/9/19/seg1.ts</pre>

15 [0051] Referring again to FIG. 5A, the index file generator 530 can also include an indicator within an index file to indicate whether a chunk of media is to be dynamically created. If, for example, it is determined that a requested media asset has not been chunked and that the asset will be chunked dynamically, the index file generator can include the indicator in data corresponding to a chunk of media to be created. The indicator, which can be as simple as including the term “seg” in a URL, will indicate that a requested chunk of media needs to be generated.

20 [0052] The chunks of media can be generated during media streaming by a dynamic segmentor 550, which can be incorporated into an HTTP service 540. The HTTP service 540, as well as the media asset origin 560 can be located within a kernel application center 111 of the CHIMPS 110 on, for example, a media asset origin server. The system 500-1 can be configured such that the kernel application center 111 provides dynamically-created chunks of media to a CDN 150 for

delivery to client 510. The CDN 150 can store the chunks locally in, for example, a media asset cache 520, thereby forgoing the need to dynamically create a chunk again if the same chunk is requested in the future.

[0053] In sum, the system for dynamic index file generation and media asset chunking 500-1 can, after receiving a request for an index file from a client 510, dynamically generate an index file with an index file generator 530. The index file can, among other things, indicate where a next chunk of media may be located. A client can then request the chunk from the location indicated by the index file, which can comprise a media asset cache 520 in a CDN 150. If the chunk is not found in the media asset cache 520, the cache miss can redirect the request to a segmentor 550 of an HTTP service 540, which can dynamically generate the requested chunk of media by accessing the corresponding media asset in the media asset origin 560. The requested media chunk can then be provided to the CDN 150 for storage in the media asset cache 520 and delivery to the client 510. If the same chunk is requested at a later point in time, the CDN 150 can deliver the chunk from the media asset cache 520, thereby forgoing the need to redirect the request to the segmentor 550 to regenerate the chunk.

[0054] FIG. 5B illustrates an alternative embodiment 500-2 of a system for dynamic index file generation and media file chunking. Rather than utilize a CDN, this embodiment 500-2 includes a media caching server within an application center 112 of the CHIMPS 110. The media caching server can receive chunk requests from and provide the corresponding chunks to a client. It will be understood that such a media caching server(s) or similar device(s) can be located anywhere within the CHIMPS and/or in a system(s) communicatively linked to the CHIMPS.

[0055] FIG. 5C illustrates an embodiment 500-3 of a system for index file generation used in conjunction with a redirector 590. As discussed above, an index file generator 530 may create an index file having one or more URIs or other location information directing the client 510 to one or more redirectors 590. Redirector 590, can then provide the URI of the requested chunk with a redirect, such as a redirect under HTTP status code 302. The URI can be located on a CDN 520 or other location (such as a media caching server 570) and/or dynamically created by the dynamic segmentor 550. It will be understood that there can be any number of redirectors 590, which can be at any location, including locations of the CHIMPS 110 such as the application center 112 (as shown in FIG. 5C) or the kernel application center 111. It also will be understood that the URI or other location information provided by redirector 590 can be generated by or

provided to the redirector 590 in any number of ways. The URI can be generated, for example, based on the request received by redirector 590 from client 510. Finally, it will be understood that the CHIMPS 110 can be configured to dynamically implement any combination of embodiments 500-1, 500-2, and 500-3, further choosing whether to utilize one or more
 5 redirectors based on factors such as, for example, a detected type of client 510.

[0056] Embodiments utilizing one or more redirectors can have several advantages. For example, and not by way of limitation, if a certain client were implemented in such a way that it “reads ahead” to request chunks, it could result in incorrect reporting data. Thus, it would be advantageous to determine which chunk is actually requested by the client. Additionally or
 10 alternatively, where chunks are available in various sub-streams with different bitrates, determining the actual requested chunk can be useful in calculating Quality of Service (QOS) metrics. Furthermore, it there may be scenarios in which it is more efficient to create larger index files having many chunks comprising large segments of media, reducing the number of index files required to stream a media asset, and thereby reducing the processing requirements to
 15 create the index files. If encryption is used having, for example, a rotating key or a per client key encryption scheme in which a valid key might change during playback of a media asset, it also may be advantageous to incorporate redirector(s) for handling legacy keys for some period of time.

[0057] FIG. 6 illustrates a simplified flowchart of an embodiment of a method 600 for
 20 implementing a dynamic index for media streaming. The method 600, which can be executed by the index file generator 530, begins at block 610, where a request for an index file is received from a client 510. According to media streaming protocols contemplated by this embodiment, if data regarding a chunk of media is not provided in an initial index file, the client will continue to request or refresh an index file until it reaches an indicator in the index file that signals the end of
 25 a stream. Thus, the method can be assured to receive more than one request for an index file from a client provided that an initial index file does not include an indicator signaling the end of the stream.

[0058] At block 615, the method 600 additionally provides for receiving input from an advertising service. According to some embodiments, this input could be the availability of an
 30 advertisement, and can be provided by a service inside or outside the CHIMPS. In other embodiments, the input could come from a service that factors in any of a variety of factors to

indicate that a specific advertisement or type of advertisement should be shown. Or that any advertisement should be shown.

5 [0059] At block 620, a determination is made whether to include an advertisement in the next chunk. According to some embodiments, this determination can be made with or without input from an advertisement service. It should be known that this determination can include the factors used by an advertisement service to provide the input of block 615. Whether the determination includes input from an advertisement service of block 615 or not, the determination can still include factors such as information about an end user collected before or during streaming of the media. This can include behavior of the end user during streaming of the media (as determined, for example, by machine-based logic through beaconing data and/or requested chunks of media provided by a client 510). Factors can also include information regarding the media asset used for streaming (such as type of content or preferred points within the media for an advertisement), preference(s) and/or selection(s) of an end user, when a previous advertisement was shown, time of day, and more. It can further include information regarding the source of a media asset, such as who created and/or provided the asset for viewing by an end user. It will be understood that other embodiments contemplate include secondary media, other than advertisements into the media stream in this manner. Moreover, the secondary media and/or advertisement can be of any length and also may be chunked. Thus, it may be determined that the next chunk includes all, or a select portion, of an advertisement of any specific length.

25 [0060] An index file is created based on the request as well as the determination of whether media, such as an advertisement, should be streamed, indicated by block 625. As discussed above, the index file can assume a variety of formats and include any amount of information regarding a next chunk of media for streaming. For example, HTTP streaming can utilize index files having the URLs of available chunks of media. Information can be embedded in these URLs to indicate a location to download the corresponding chunk of media, starting point and/or ending point of a chunk of media, an indicator to indicate whether the chunk is to be dynamically created by a segmentor 550, a location of an advertisement to be streamed, and more. This information is included in the index file and sent to the client at block 630.

30 [0061] At block 635, reporting data can be created based on information included in the index file. As previously discussed, information included in an index file and/or index file request can

indicate the behavior of an end user device 140 during streaming, such as a pause, stop, skip, play, etc. of the media. According to some embodiments, this information can be extracted from requests for an index file and/or providing the requested index file. The information can be gathered in addition to or as a substitute for beaconing data provided by a client 510. Moreover, if beaconing data is provided, the creation of reporting data may be omitted altogether.

[0062] Reporting data can include any amount of information regarding end user behavior, as indicated through index file requests and/or provided index files. This can include a particular action and when it was performed. Additionally or alternatively, the data may be kept in a more rudimentary form, depending on the application or embodiment, indicating the data included in a index file request and/or an index file. This reporting data may be stored in a log file for reporting after streaming and/or transmitted during streaming to a relevant service that collects such metrics.

[0063] As indicated by block 640, the reporting data may be sent to a metrics collector for analytics. A metrics collector, according to certain embodiments, may be an application executed by a server from within the application center 112 in which the index file generator 530 is executed, or it may be executed elsewhere, such as in a kernel application center 111 or in a system outside the CHIMPS 110. Depending on the form of the reporting data, the metrics collector can further process and store the information.

[0064] FIG. 7 illustrates a simplified flowchart of an embodiment of a method for dynamically chunking a media file for streaming 700. This method can be employed by a variety of systems and/or programs. For example, it may be executed by a dynamic segmentor 550 of an HTTP service 540 running on a server located in a kernel application center 111 of the CHIMPS 110.

[0065] The method 700 can begin at block 710, when a request for a chunk of media is received from a CDN 150. As discussed above, this request may be made in response to a cache miss at the CDN 150 and/or because an indicator was included in the request for the chunk of media that the chunk was to be created dynamically. As discussed herein, if the CDN 150 has the requested chunk cached from a prior request, the CDN 150 can provide the requested chunk and preclude the need to send the request to a dynamic segmentor 550 to generate the chunk. It should be understood that the request may come from sources other than a CDN 150 according to alternative embodiments. One such source includes the media caching server 570 of embodiment 500-2, as shown in FIG. 5B.

5 [0066] The starting and ending points of a requested chunk of media are then determined at block 715. This information can be included directly in the request or derived from the request, a previous request, and/or other sources. At block 720, the information, as well as information identifying the requested chunk of media, can be used to retrieve all or part of the relevant media asset from a media asset origin 560. The retrieved portion will include at least the relevant media from the starting point to the ending point of the requested chunk of media.

10 [0067] At block 725, the requested media chunk is generated by converting the relevant portion of the media asset into a deliverable chunk. The media asset, as stored in the media asset origin, may not be chunked; it may be stored in its entirety as a media file (or group of alternative files corresponding to alternative sub-streams). Generating the chunk therefore can require determining the starting and ending points from the retrieved portion of the media asset and converting the resulting segment of media into a deliverable chunk.

15 [0068] Although the generation of the deliverable chunk may involve transcoding, it may not. The media asset can be stored in a format where transcoding may not be needed, thereby reducing the processing requirements for creating chunks of media during streaming. For example, media assets may be stored such as H.264 or MPEG-4 video format and/or AAC, HE-AAC, or MP3 audio format. According to some streaming protocols, such as some forms of HTTP streaming, chunks of media in these formats would not need transcoding before being wrapped in an MPEG-2 transport stream container format. Instead, such a conversion essentially
20 would require the addition of metadata to create the streaming format from the format of the stored media asset. In other words, generating a deliverable chunk of media may only require identifying the stored media asset, extracting the relevant segment of the media from the media asset, and adding certain metadata in accordance with a container format. This process requires little processing power and can be easily performed on the fly during streaming. Once the
25 deliverable chunk of media is generated, it is sent to the CDN 150 or other requesting entity, at block 730.

30 [0069] FIG. 8 illustrates a simplified swim lane flowchart describing the interaction of components in a system configured to provide dynamic indexing and chunking for media streaming, according to one embodiment. In this embodiment, a client can send a request for an index file 805, the request received by an index file generator 810. A particular request may be made to initiate the streaming of a media asset, or it may be during streaming. Depending on the

streaming protocol, the request may be made while a client plays a chunk of media previously downloaded during streaming.

[0070] The index file generator generates an index file to indicate the next chunk of media 815. As described above, this chunk may include an advertisement, and the index file can include any amount of information about a chunk of media, including information regarding alternative sub-streams for streaming. The dynamic index file generator can include information regarding existing chunks of media, and, when used in conjunction with a dynamic segmentor may also include information regarding chunks of media that may need to be created. As detailed above, if a chunk of media is to be generated dynamically, the index file generator may indicate this by including an indicator in the generated index file, such as in a URL for one or more chunks described within the index file. Once the index file is generated, the index file generator sends the index file 820, which is received by the client 825.

[0071] Alternative embodiments may provide for the generation of index files containing more than a next chunk of media. For example, an index file generator may generate an index file containing information regarding several chunks of media, in which case the chunks of media can be dynamically generated by a dynamic segmentor when requested by the client. The determination of whether to include information regarding more than a next chunk of media can include factors such as whether the index generator is generating reporting data, the desired frequency of such reporting data, and more.

[0072] Using information contained in the index file, the client can then request the next chunk of media 830, and this request can be received by a CDN 835. The CDN then checks to see if the chunk is already stored in the cache 840. If so, the CDN can provide the requested chunk to the client, blocks 845 and 850. The requested chunk may be found in a CDN's cache if the chunk was created and stored in the CDN during preprocessing or if a the chunk was dynamically created and stored in the CDN from an earlier request.

[0073] If the chunk is not found on the CDN, the chunk can be requested of the dynamic segmentor, which receives the request 855 and retrieves the corresponding media asset from an asset origin server 860. As discussed above, the entirety of the relevant media asset does not need to be retrieved as long as at least the portion containing the relevant segment for the requested chunk is retrieved. It will be understood that alternative embodiments can provide for

the media asset being stored in a variety of locations accessible, directly or indirectly, to the dynamic segmentor.

[0074] The dynamic segmentor can then generate the requested chunk by converting the retrieved media into a deliverable chunk. That is, the dynamic segmentor converts the retrieved media into an acceptable format for streaming, which can vary depending on the streaming protocol utilized. The dynamic segmentor can then return the requested chunk 870 to the CDN, which can cache the chunk and return it to the client 875. Once the chunk is received by the client 880, the client can play the chunk to an end user.

[0075] It should be noted that the methods, systems, and devices discussed above are intended merely to be examples. It must be stressed that various embodiments may omit, substitute, or add various procedures or components as appropriate. For instance, it should be appreciated that, in alternative embodiments, the methods may be performed in an order different from that described, and that various steps may be added, omitted, or combined. Also, features described with respect to certain embodiments may be combined in various other embodiments. Different aspects and elements of the embodiments may be combined in a similar manner. Also, it should be emphasized that technology evolves and, thus, many of the elements are examples and should not be interpreted to limit the scope of the invention.

[0076] Specific details are given in the description to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these specific details. For example, well-known circuits, processes, algorithms, structures, and techniques have been shown without unnecessary detail in order to avoid obscuring the embodiments. This description provides example embodiments only, and is not intended to limit the scope, applicability, or configuration of the invention. Rather, the preceding description of the embodiments will provide those skilled in the art with an enabling description for implementing embodiments of the invention. Various changes may be made in the function and arrangement of elements without departing from the spirit and scope of the invention.

[0077] Also, it is noted that the embodiments may be described as a process which is depicted as a flow diagram or block diagram. Although each may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be rearranged. A process may have additional steps not included in

the figure. Furthermore, embodiments of the methods may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware, or microcode, the program code or code segments to perform the necessary tasks may be stored in a computer-readable medium such as a storage medium. Processors may perform the necessary tasks.

[0078] Having described several embodiments, it will be recognized by those of skill in the art that various modifications, alternative constructions, and equivalents may be used without departing from the spirit of the invention. For example, the above elements may merely be a component of a larger system, wherein other rules may take precedence over or otherwise modify the application of the invention. Also, a number of steps may be undertaken before, during, or after the above elements are considered. Accordingly, the above description should not be taken as limiting the scope of the invention.

[0079] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0080] The reference to any prior art in this specification is not, and should not be taken as, an acknowledgement or any form of suggestion that the prior art forms part of the common general knowledge in Australia.

WHAT IS CLAIMED IS:

1 1. A server for communicating, with a network, a media file having a
2 plurality of segments of a certain length, the server comprising:
3 an interface for communicating with the network;
4 a memory configured to store:
5 one or more of the plurality of media file segments; and
6 a plurality of instructions for communicating the media file over
7 the network;
8 a processor communicatively coupled with the memory and the interface, the
9 processor further configured to execute the plurality of instructions, the plurality of instructions
10 configured to cause the server to:
11 receive, with the interface, a request for a first segment of the
12 media file;
13 determine a starting point and an ending point of the requested first
14 segment of the media file, wherein the determination is based, at least in part, on
15 the request for the first segment of the media file;
16 retrieve a first portion of the media file from the memory, wherein:
17 the first portion of the media file includes at least the
18 requested first segment of the media file, and
19 the first portion of the media file includes a first
20 format;
21 generate the requested first segment of the media file, wherein:
22 the requested first segment of the media file has a
23 second format, and
24 the requested first segment of the media file is
25 generated, at least in part, from the first portion of the media file;
26 provide, with the interface, the requested first segment of the
27 media file;
28 receive, with the interface, a request for a second segment of the
29 media file;

30 determine a starting point and an ending point of the requested
 31 second segment of the media file, wherein the determination is based, at least in
 32 part, on the request for the second segment of the media file;
 33 retrieve a second portion of the media file from the memory;
 34 wherein:
 35 the second portion of the media file includes at least
 36 the requested second segment of the media file, and
 37 the second portion of the media file includes the first
 38 format;
 39 generate the requested second segment of the media file, wherein:
 40 the requested second segment of the media file has
 41 the second format, and
 42 the requested second segment of the media file is
 43 generated, at least in part, from the second portion of the media
 44 file; and
 45 provide, with the interface, the requested second segment of the
 46 media file.

1 2. The server of claim 1 further having instructions configured to cause the
 2 server to retrieve the first and second portions of the media file with the interface and store the
 3 first and second portions of the media file in the memory.

1 3. The server of claim 1 wherein the second portion of the media file is the
 2 same as the first portion of the media file.

1 4. The server of claim 1 wherein the requested first segment of the media file
 2 and the requested second segment of the media file comprise contiguous segments of the media
 3 file

1 5. The server of claim 1 wherein at least a portion of the instructions are
 2 executed during playback of the media file by a device communicatively linked to the network.

1 6. The server of claim 1 wherein the first format includes one or more of an
 2 audio compression format, a video compression format, and an image compression format.

- 1 7. The server of claim 1 wherein the second format includes a media
2 container format.
- 1 8. The server of claim 1 further configured to receive the requests for the
2 first and second segments of the media file from a Content Delivery Network (CDN).
- 1 9. The server of claim 1 wherein the plurality of segments of a first media
2 file have a different length than a plurality of segments of a second media file.
- 3 10. A method for providing, with a data network, a plurality of segments of a
4 media file, the segments having a certain size, the method comprising:
5 receiving a request for a first segment of the media file;
6 determining a starting point and an ending point of the requested first segment of
7 the media file, wherein the determination is based, at least in part, on the request for the first
8 segment of the media file;
9 retrieving a first portion of the media file, wherein:
10 the first portion of the media file includes at least the requested
11 first segment of the media file, and
12 the first portion of the media file includes a first format;
13 generating the requested first segment of the media file, wherein:
14 the requested first segment of the media file has a second format,
15 and
16 the requested first segment of the media file is generated, at least in
17 part, from the first portion of the media file;
18 providing the requested first segment of the media file;
19 receiving a request for a second segment of the media file;
20 determining a starting point and an ending point of the requested second segment
21 of the media file, wherein the determination is based, at least in part, on the request for the
22 second segment of the media file;
23 retrieving a second portion of the media file, wherein:
24 the second portion of the media file includes at least the requested
25 second segment of the media file, and
26 the second portion of the media file includes the first format;

27 generating the requested second segment of the media file, wherein:
 28 the requested second segment of the media file has the second
 29 format, and
 30 the requested second segment of the media file is generated, at
 31 least in part, from the second portion of the media file; and
 32 providing the requested second segment of the media file.

1 11. The method of claim 10 wherein the first portion of the media file is the
 2 same as the second portion of the media file.

1 12. The method of claim 10 wherein the first format includes one or more of
 2 an audio compression format, a video compression format, and an image compression format.

1 13. The method of claim 10 wherein the second format includes a media
 2 container format.

1 14. The method of claim 10 wherein the requests for the first and second
 2 segments of the media file are received from a CDN.

1 15. The method of claim 14 wherein the requested first and second segments
 2 of the media file are provided to the CDN.

1 16. The method of claim 10 wherein the receiving the requests for the first and
 2 second segments of the media file and the providing the requested first and second segments of
 3 the media file occur during playback of the media file by a device communicatively linked to the
 4 data network.

1 17. The method of claim 10 wherein the data network comprises a first
 2 network and the retrieving at least a portion of the media file from a data origin occurs over a
 3 second network.

1 18. A system for communicating, with a network, a media file having a
 2 plurality of segments of a certain size, the system comprising:
 3 a data storage configured to store the media file having a first format;
 4 a processing server communicatively coupled with the data storage, the
 5 processing server configured to:

6 receive, with a network, a request for a first segment of the media
7 file;
8 determine a starting point and an ending point of the requested first
9 segment of the media file, wherein the determination is based, at least in part, on
10 the request for the first segment of the media file;
11 retrieve a first portion of the media file from the data storage,
12 wherein:
13 the first portion of the media file includes at least the
14 requested first segment of the media file, and
15 the first portion of the media file includes the first
16 format;
17 generate the requested first segment of the media file, wherein:
18 the requested first segment of the media file has a
19 second format, and
20 the requested first segment of the media file is
21 generated, at least in part, from the first portion of the media file;
22 provide, with a network, the requested first segment of the media
23 file;
24 receive, with a network, a request for a second segment of the
25 media file;
26 determine a starting point and an ending point of the requested
27 second segment of the media file, wherein the determination is based, at least in
28 part, on the request for the second segment of the media file;
29 retrieve a second portion of the media file from the data storage,
30 wherein:
31 the second portion of the media file includes at least
32 the requested second segment of the media file, and
33 the second portion of the media file includes the first
34 format;
35 generate the requested second segment of the media file, wherein:
36 the requested second segment of the media file has
37 the second format, and

2010202741 30 Jun 2010

38 the requested second segment of the media file is
39 generated, at least in part, from the second portion of the media
40 file; and
41 provide, with the interface, the requested second segment of the media file.

1 19. The system of claim 18 further comprising a media caching server
2 configured to:
3 send the requests for the first and second segments of the media file with the
4 network to the processing server;
5 receive the requested first and second segments of the media file with the network
6 from the processing server; and
7 store the requested first and second segments of the media file in a memory.

1 20. The system of claim 19 wherein the media caching server is further
2 configured to provide the requested first and second segments of the media file to a device
3 communicatively coupled with the media caching server.

1 21. The system of claim 19 configured such that the media caching server will
2 not send the requests for the a segment of the media file with the network to the processing
3 server if the media caching server has the segment of the media file stored in the memory.

1 22. A server for communicating, with a network, a media file having a
2 plurality of segments of a certain length substantially as herein described.

1 23. A method for providing, with a data network, a plurality of segments of a
2 media file, the segments having a certain size substantially as herein described.

1 24. A system for communicating, with a network, a media file having a
2 plurality of segments of a certain size substantially as herein described.

2010202741 30 Jun 2010

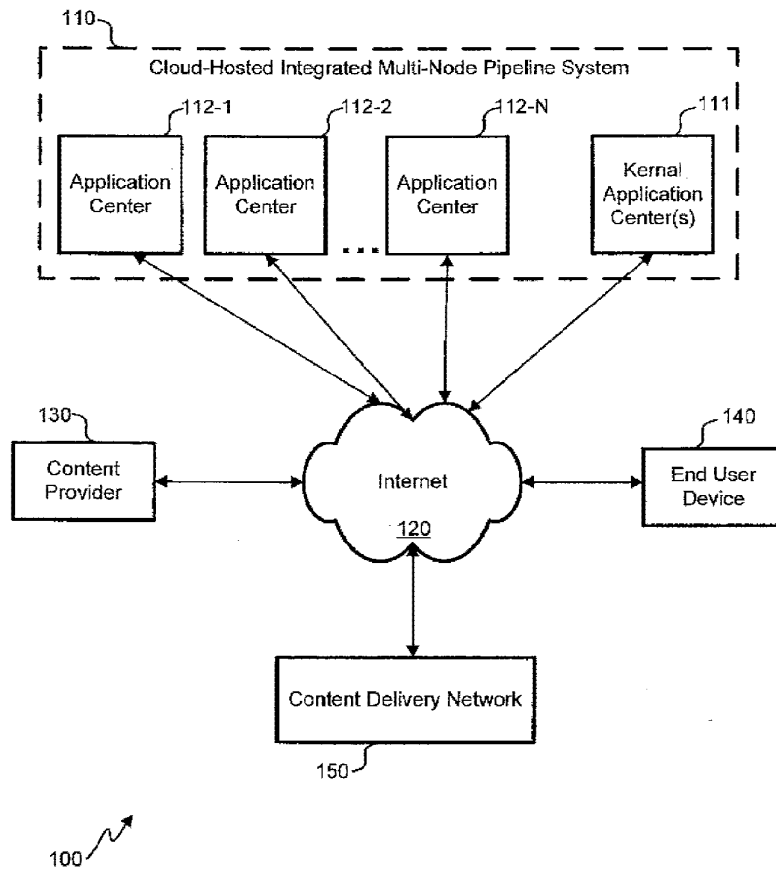


FIG. 1

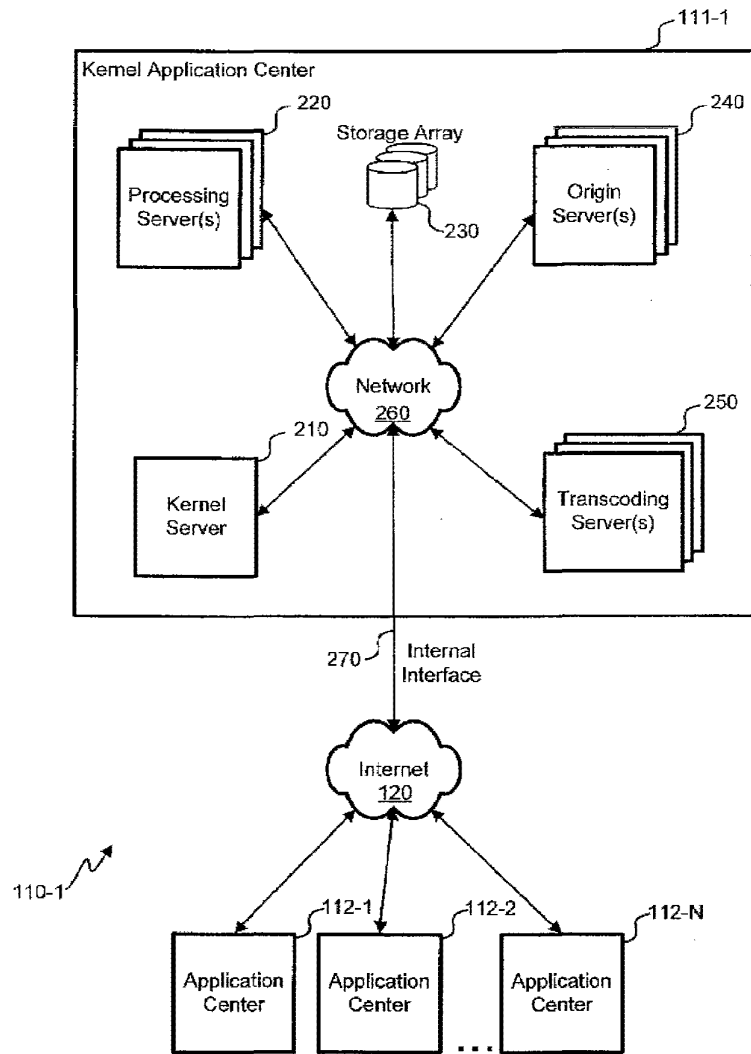


FIG. 2A

2010202741 30 Jun 2010

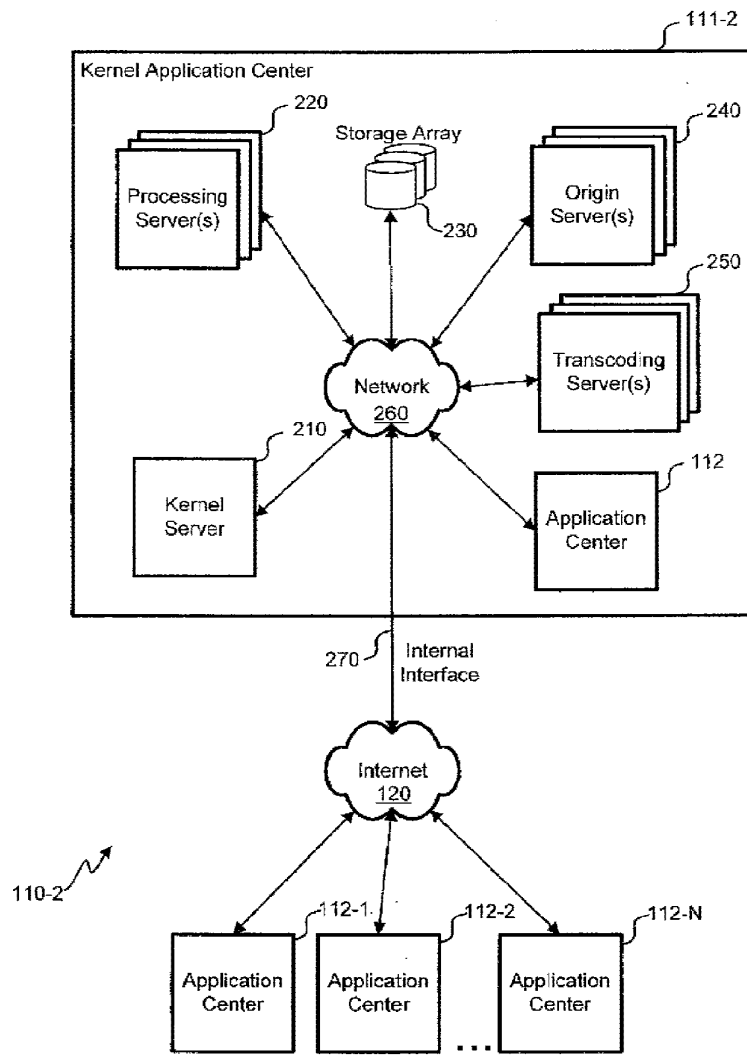
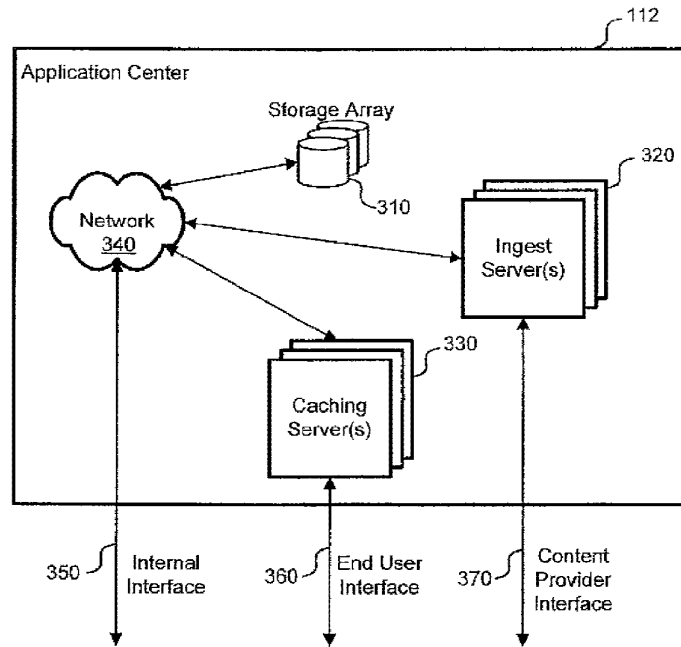


FIG. 2B

2010202741 30 Jun 2010

**FIG. 3**

2010202741 30 Jun 2010

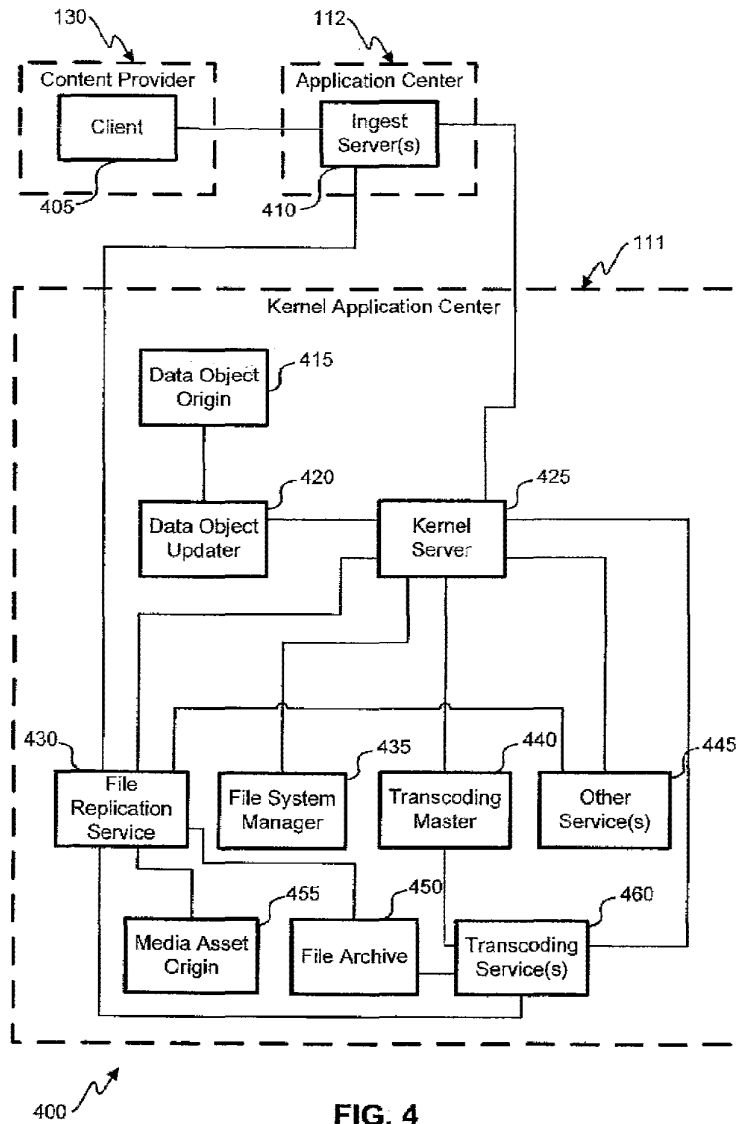
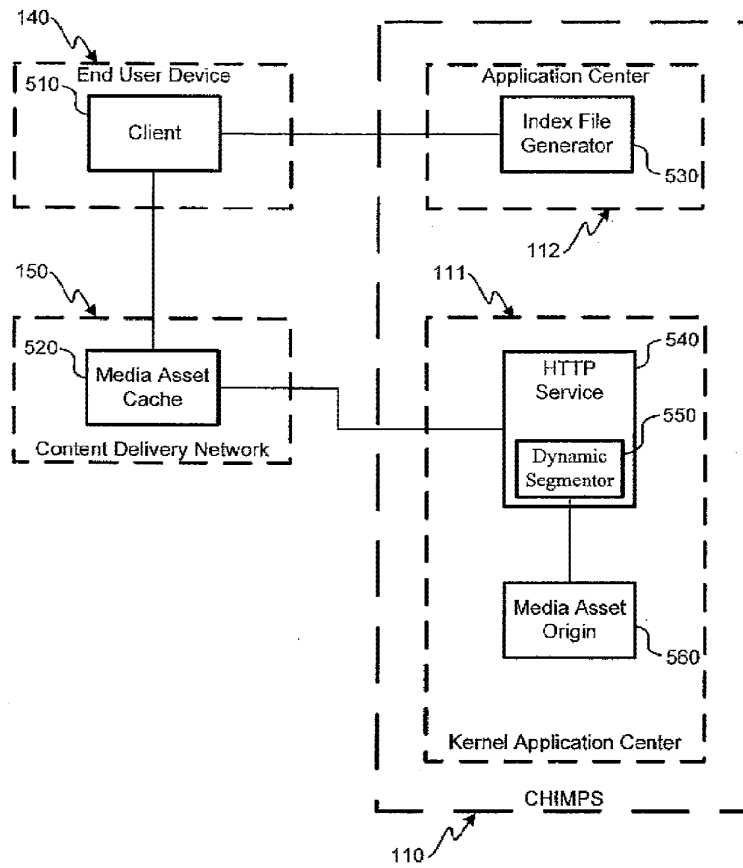


FIG. 4



500-1

FIG. 5A

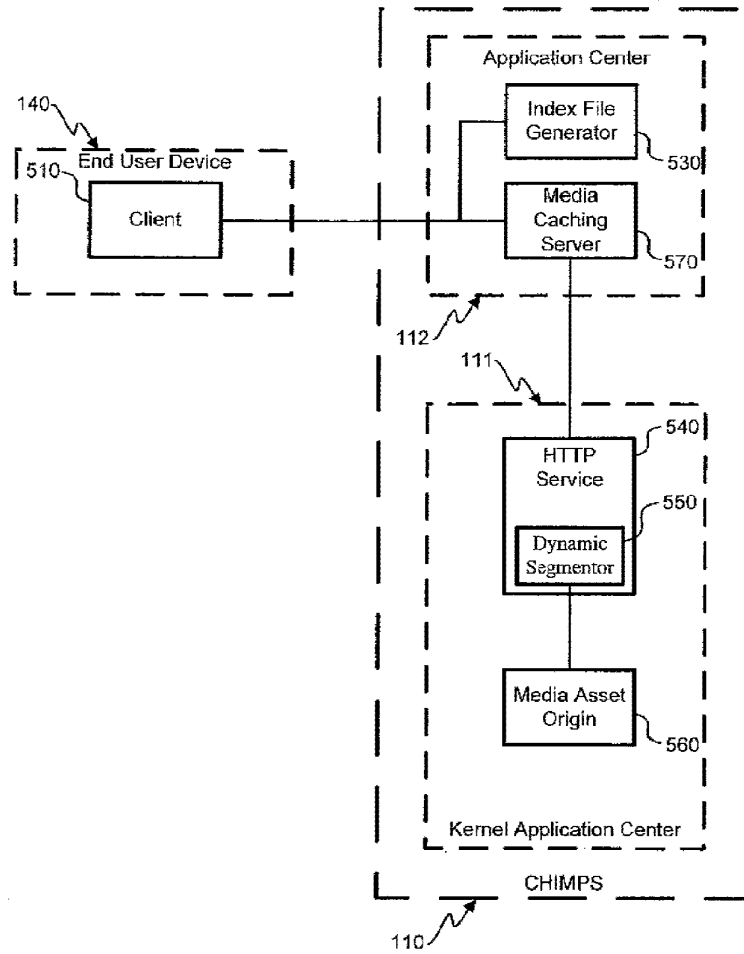
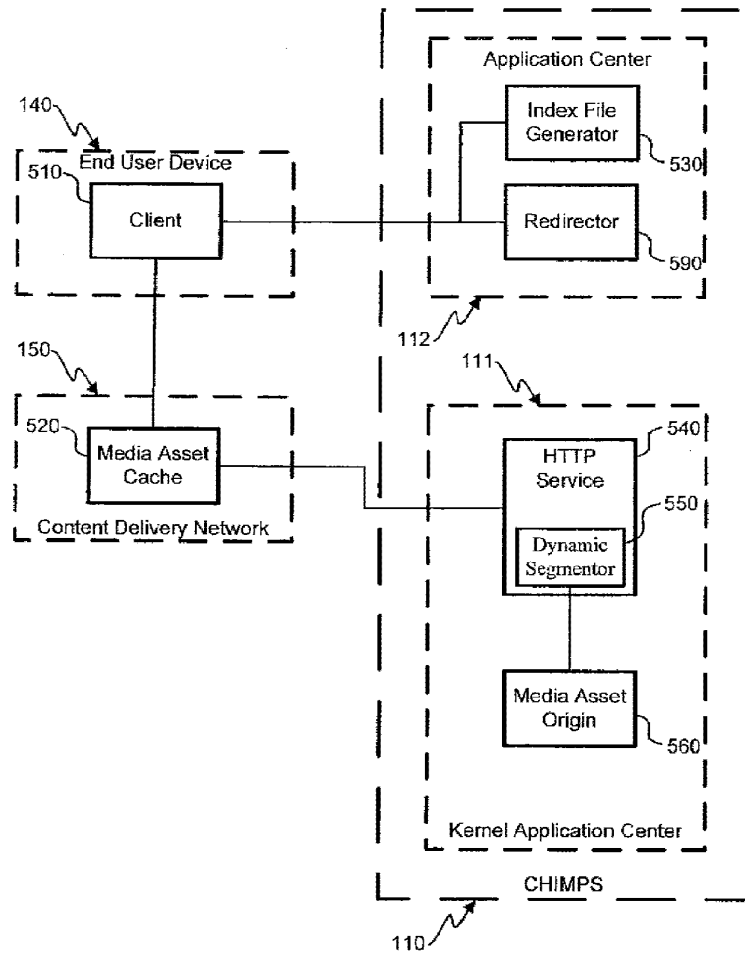
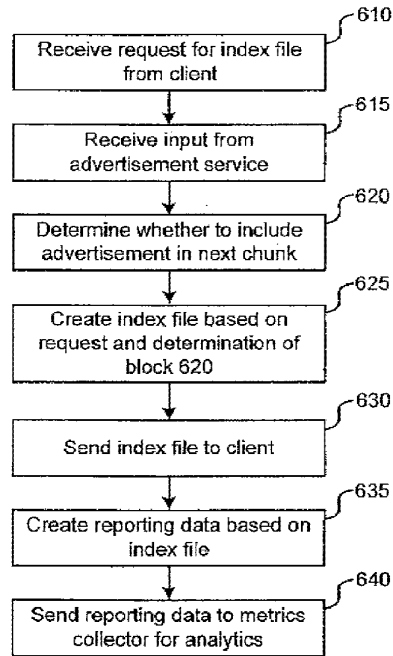


FIG. 5B



500-3

FIG. 5C



600 ↗

FIG. 6

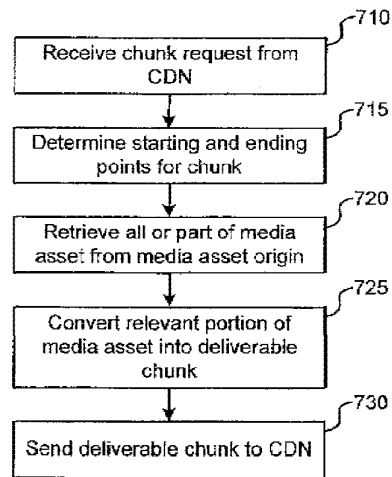
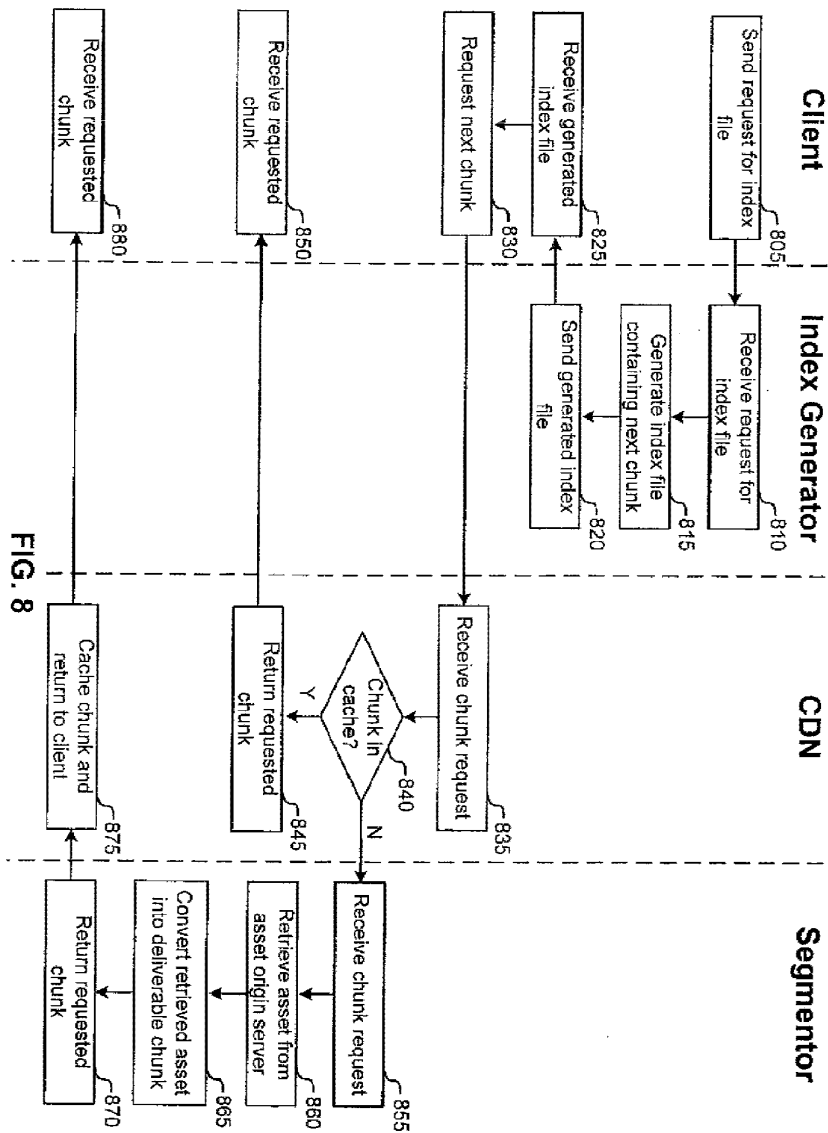


FIG. 7

700



2010202741 30 Jun 2010

(19) AUSTRALIAN PATENT OFFICE

(54) Title
Cloud data persistence engine

(51)⁶ International Patent Classification(s)
G06F 17/30 (2006.01)20060101AFI2010070
G06F 17/30 6BHAU

(21) Application No: 2010202782 (22) Application Date: 2010.07.01

(43) Publication Date : 2010.11.25

(43) Publication Journal Date : 2010.11.25

(71) Applicant(s)
Unicom Media, Inc.

(72) Inventor(s)
McGowan, Albert J.; Carls, Richard L.

(74) Agent/Attorney
Pizzey's, Level 2, Woden Plaza Offices Woden Town Square Woden, Canberra, ACT, 2606

(56) Related Art
US 2003/0061272

2010202782 01 Jul 2010

ABSTRACT OF THE DISCLOSURE

Various cloud data persistence systems and methods are described. In some embodiments, a client may request a data object from a server. This data object may contain a link to a media asset and other information linked to the video file, such as a title and duration. This media asset may be stored in a different location. If this first server does not have the data object stored locally, it may contact a second server to determine if it has the data object stored. If the second server does not have the data object stored, it may contact a data object origin server, which may have all existing data objects stored. This data object origin server may then send the data object to the second server, which may in turn transmit the data object to the first server. The first server may then transmit the data object to the client.

2010202782 01 Jul 2010

Regulation 3.2

AUSTRALIA

Patents Act 1990

COMPLETE SPECIFICATION STANDARD PATENT

APPLICANT: **UNICORN MEDIA, INC**

Invention Title: **CLOUD DATA PERSISTENCE ENGINE**

The following statement is a full description of this invention, including the best method of performing it known to me:

CLOUD DATA PERSISTENCE ENGINE

BACKGROUND OF THE INVENTION

5 [0001] This disclosure relates in general to cloud-based computer processing and, but not by way of limitation, to distribution, storage, and retrieval of data on a cloud networking arrangement.

[0002] In a typical data hosting architecture, multiple servers may be spread over a geographical area. The data stored on each of these servers may be the same (or substantially similar) data as that stored on each other server. A client attempting to access data hosted on 10 such a system may communicate with a particular server, often the server closest to the client's physical location.

[0003] While such an arrangement may allow the client to quickly access all data stored on the server system, it may not be efficient for all data to be stored on each server. Clients in different locations (e.g., different cities, states, countries, and/or continents) may request data that varies 15 significantly by location. For example, particular data may frequently be requested by clients in Europe, while the same data may be infrequently requested by clients in the Americas. Therefore, if all of the data is maintained at each server, such a system may result in significant quantities of data being stored at servers where the data is infrequently, if at all, accessed. Accordingly, such a system may result in a substantial amount of data being stored at servers that 20 wastes storage resources and is infrequently accessed.

BRIEF SUMMARY OF THE INVENTION

[0004] Various cloud data persistence systems and methods are described. In some embodiments, a client may request a data object from a server. This data object may contain a 25 link to a media asset and other information linked to the video file, such as a title and duration. This media asset may be stored in a different location. If this first server does not have the data object stored locally, it may contact a second server to determine if it has the data object stored. If the second server does not have the data object stored, it may contact a data object origin server, which may have all existing data objects stored. This data object origin server may then

send the data object to the second server, which may in turn transmit the data object to the first server. The first server may then transmit the data object to the client.

[0005] In some embodiments, a cloud data persistence system for distribution of data is present. The system may include a first server which stores a first plurality of data objects and is communicatively coupled with a plurality of cache servers. The first server may be located at a first location. Each of the first plurality of data objects may comprise a link to one of a plurality of media assets. The plurality of media assets may be stored at a second location separate from the first location. The system may include a second server that stores a second plurality of data objects and is configured to request a data object from the first server if the data object is not present in the second plurality of data objects. The second server may be one of the plurality of cache servers. The second server may be communicatively coupled with a first plurality of application servers. The system may include a third server that stores a third plurality of data objects and is configured to request the data object from the first server if the data object is not present in the third plurality of data objects. The third server may be one of the plurality of cache servers. The third server may be communicatively coupled with a second plurality of application servers. The system may include a fourth server that stores a fourth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fourth plurality of data objects. The fourth server may be one of the first plurality of application server. The fourth server may be configured to receive a request from a first client for the data object. The system may also include a fifth server that stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects. The fifth server may be one of the first plurality of application servers. The fifth server may be configured to receive a request from a second client for the data object. The system may further include a sixth server that stores a sixth plurality of data objects and is configured to request the data object from the third server if the data object is not present in the sixth plurality of data objects. The sixth server may be one of the second plurality of application servers. The sixth server may be configured to receive a request from a third client for the data object. The system may include a seventh server that stores a seventh plurality of data objects and is configured to request the data object from the third server if the data object is not present in the seventh plurality of data objects. The seventh server may be one of the second plurality of application servers. The seventh server may be configured to receive a request from a fourth client for the data object.

[0006] In some embodiments, a method for retrieving data objects using a cloud data persistence system configured for distribution of data is presented. The method may include receiving a first request from a client for a data object. The application center may be one of a plurality of application centers that communicate with a plurality of clients. The application center may be located at a first location. The data object may comprise a link to a media asset. The media asset may be stored at a second location separate from the first location. The method may further include determining the data object is not stored among a first plurality of data objects stored at the application center. The method may include transmitting a second request for the data object to a first cache server. The first cache server may be communicatively coupled with the plurality of application centers. The first cache server may be one of a plurality of cache servers. The method may include receiving the second request for the data object. The method may also include determining the data object is not stored among a second plurality of data objects stored at the cache server. The method may further include transmitting a third request for the data object to an origin server. The origin server may be communicatively coupled with the plurality of cache servers. The method may also include receiving the third request for the data object. The method may include locating the data object among a third plurality of data objects. The method may also include transmitting the data object linked to the third request to the cache server. The method may include receiving the data object linked to the third request. The method may include transmitting the data object linked to the second request. Also, the method may include receiving the data object linked to the second request. Further, the method may include transmitting the data object linked to the first request to the client.

[0007] In some embodiments, a cloud data persistence system for distribution of data is present. The system includes a first server which stores a first plurality of data objects and is communicatively coupled with a second server and a third server. The first server may be located at a first location. Each of the first plurality of data objects may comprise a link to a media asset of a plurality of media assets. The plurality of media assets may be located at a second location. The second location may be distinct from the first location. The system may include the second server, which may store a second plurality of data objects and may be configured to request a data object from the first server if the data object is not present in the second plurality of data objects. The second server may be communicatively coupled with a fourth server and fifth server. The system may include the fourth server that stores a third plurality of data objects and may be configured to request the data object from the second server

if the data object is not present in the third plurality of data objects. The fourth server may be configured to receive a request from a first client for the data object.

[0008] In some embodiments, the first plurality of data objects comprises all data objects present in the second plurality of data objects and the third plurality of data objects. In some
 5 embodiments, the third server stores a fourth plurality of data objects and is configured to request the data object from the first server if the data object is not present in the fourth plurality of data objects. The third server and the second server may be data persistence cache servers. The third server may be communicatively coupled with a sixth and seventh server. In some embodiments,
 10 the fifth server stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects. The fifth server and the fourth servers may be application servers. The fifth server may be configured to receive requests from a second client for the data object. The fifth server may be geographically separated from the fourth server. In some embodiments, the request from the first
 15 client for the data object comprises a universal resource indicator. In some embodiments, each of the first plurality of data objects comprise a universal resource locator and a title linked to a media asset of the plurality of media assets. Also, in some embodiments, data objects in the second plurality of data objects and the third plurality of data objects are at least partially determined based on a frequency of requests for those data objects received from clients. In
 20 some embodiments, data objects in the second plurality of data objects and the third plurality of data objects are at least partially determined based on when requests for those data objects were last received from clients. Further, in some embodiments, the first plurality of data objects comprises all data objects in the second plurality of data objects, the third plurality of data objects, the fourth plurality of data objects, and the fifth plurality of data objects.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present disclosure is described in conjunction with the appended figures:

[0010] FIG. 1 illustrates a block diagram of a media servicing system.

[0011] FIG. 2A illustrates a block diagram of an embodiment of a kernel application center connected with application centers.

- [0012] FIG. 2B illustrates a block diagram of an alternative embodiment of a kernel application center.
- [0013] FIG. 3 illustrates a block diagram of an embodiment of an application center.
- [0014] FIG. 4A illustrates a block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion.
- [0015] FIG. 4B illustrates a block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion with a content delivery network.
- [0016] FIG. 4C illustrates another block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion with a content delivery network.
- [0017] FIG. 5A illustrates a simplified block diagram of an embodiment of a cloud data persistence system.
- [0018] FIG. 5B illustrates another simplified block diagram of an embodiment of a cloud data persistence system.
- [0019] FIG. 6 illustrates a simplified swim diagram of an embodiment of a method for responding to a client request to retrieve a data object from a cloud data persistence system.
- [0020] FIG. 7 illustrates a simplified swim diagram of an embodiment of a method for responding to a client request to modify a data object on a cloud data persistence system.
- [0021] FIG. 8 illustrates a simplified block diagram of an embodiment of an asset management and storage system.
- [0022] FIG. 9 illustrates another simplified block diagram of an embodiment of an asset management and storage system.
- [0023] FIG. 10 illustrates an embodiment of a method for retrieving data from an asset management and storage system.
- [0024] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is

applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION OF THE INVENTION

5 [0025] The ensuing description provides preferred exemplary embodiment(s) only, and is not intended to limit the scope, applicability or configuration of the disclosure. Rather, the ensuing description of the preferred exemplary embodiment(s) will provide those skilled in the art with an enabling description for implementing a preferred exemplary embodiment. It is understood that various changes may be made in the function and arrangement of elements without departing
10 from the spirit and scope as set forth in the appended claims.

[0026] As opposed to storing copies of a piece of data, such as a data object that contains information linked to a media object, at each server on a system of multiple servers, it may be more efficient to store only data objects that are frequently and/or recently accessed by clients at particular servers. Such an arrangement, generally referred to as a cloud data persistence engine
15 or system, may allow the data objects stored at each server to vary based on which data objects the server's clients are currently and/or have historically requested. Therefore, data objects stored at a first server in a first location may include some of the same data objects and some different data objects than data objects stored at a second server in a second location (possibly located in a different city, country or on a different continent than the first server). Storing only
20 data objects that are frequently and/or recently accessed by clients at each server may result in considerable efficiencies. For example, by storing only data objects that are frequently and/or recently accessed by clients, the servers may require significantly less storage space because they do not need to store a potentially large amount of data objects that are infrequently accessed or have not recently been accessed. A related benefit of this may be that by decreasing the amount
25 of storage necessary at each server, the speed of access to the storage at the servers may be increased. For example, instead of relying on hard drives, solid state drives (which may cost significantly more per megabyte, but may offer improved access times) may be used for the storage at the servers.

[0027] While a server may infrequently receive a request for a particular data object from a
30 client, the server may still need to be able to fulfill such a request. Therefore, if a server is contacted by a client for a data object that it does not have stored locally, the server may be able

to contact another server, which may have the requested data object stored. Such a server may be configured to satisfy multiple servers with requests for such data objects.

5 [0028] While this second cache server may contain the data object requested by the client, it may also only store data objects that are frequently and/or recently (albeit less frequently and/or recently than the data stored at the first server) accessed by clients. If the second server does not have the data object requested by the client stored, the second server may have the ability to contact yet another server. This server, referred to as the data object origin server, may maintain a copy of all data objects stored on the system. Therefore, if the data object is present on the server system, the data object origin server is able to fulfill the request for the data. The data
10 object origin server may then route the requested data to the client via the second server and the first server.

[0029] While the above arrangement may increase efficiencies when a client requests a data object, a client (perhaps the same or a different client) may request to modify (including, but not limited to changing, adding and/or deleting) a data object. In such instances, the modification to
15 the data object (assuming the client has the appropriate access rights) may be routed to the data object origin server by a server receiving the request from the client. The modification may be applied to the relevant data at the data object origin server. The data object origin server may then send a notification to other servers, or only the cache servers that have the previous version of the data stored, that the previous version of the data is no longer valid. These servers may in
20 turn send a notification to other associated servers, or only those servers that have the previous version of the data stored, that the previous version of the data object is no longer valid. This may prevent a client from receiving an out-of-date version of the data no matter which server is accessed by the client. The modified data residing at the origin may then propagate out to the other servers as requests are made by clients to those servers for the data object.

25 [0030] While the above embodiments may be implemented in a variety of different systems, some particular embodiments may be implemented as part of a media service system. FIG. 1 is a block diagram illustrating a media servicing system 100, according to some embodiments of the present invention. The system may deliver media content to the end user 140 (alternatively referred to as a user or a client) through a network such as the Internet 120. A media asset
30 provided by a content provider 130 can be processed and indexed by cloud-hosted integrated multi-node pipelining system (CIIMPS) 110, and further stored on content delivery network

(CDN) 130. Additionally or alternatively, the CHIMPS 110 may also be adapted to store the media asset.

[0031] The media servicing system further enables a content provider 130 or other entity to gather information regarding user behavior during media playback. For example, a content provider 130 can be provided with data indicating that end users tend to stop watching a video at a certain point in playback, or that users tended to follow links associated with certain advertisements displayed during playback. With this data, a content provider 130 can adjust factors such as media content, advertisement placement and content, etc., to increase revenue associated with the media content and provide the end user 140 with a more desirable playback experience.

[0032] End user 140 can request a media asset to hear and/or see with a client program running on a personal computer or other electronic device connected to Internet 120. The client program can be, for example, a media player, browser, or other application adapted to request and/or play media assets. In response to a request for a media asset, the CHIMPS 110 can utilize any number of application centers 112 and/or kernel application center(s) 111 to provide the client program with a data object concerning the requested media asset. The data object can include information about the media asset, including where the media asset can be located, such as within the CDN 150 or within the CHIMPS 150 itself. Location information may be provided by Universal Resource Indicator (URI), a Universal Resource Locator (URL) or other indicator. During playback of the media asset, the CHIMPS 150 can collect data regarding the playback through beaconing provided by a client program executed on a device operated by the end user 140 and/or indexing service from within the CHIMPS and/or CDN. The CHIMPS 150 can subsequently provide the data and/or any analytics information derived from the data to the content provider 130.

[0033] FIG. 2A is a block diagram illustrating an embodiment of a kernel application 111-1 center connected with application centers from within the CHIMPS 110-1. The kernel application center 111-1 and application centers 112 can be geographically distant and can be connected via the Internet 120, wide area network (WAN), and/or other data communication network. Because application centers can be geographically separated, DNS services (not shown) can be used to allow an end user 140 to connect to the nearest available application center 112. The kernel application center 111-1 can connect with application centers 112 within

the CHIMPS 110-1 through an internal interface 270, thereby enabling the application centers 112 access to the various components within the kernel application center 111-1.

[0034] Components within the kernel application center 111-1 can communicate through network 260 such as a local area network (LAN) and can include one or more data object origin servers 240 and a storage array 230 with which data objects and/or media assets may be stored and distributed. The storage array 230 may also be utilized by services running on processing server(s) 220 and/or transcoding server(s) 250 that may require temporary or long-term storage. Kernel server 210 can utilize processing server(s) 220, transcoding server(s) 250 to provide various functional capabilities to the CHIMPS 110.

[0035] For example, as described in more detail below, the CHIMPS 110-1 can provide transcoding service for media assets provided by a content provider 130 for syndication. Such a service can allow a content provider 130 to upload a media asset to an application center 112, after which the application center 112 would notify the kernel server 210 that the media asset has been uploaded. The kernel server can then notify services running on the processing server(s) 220 of the upload. These services can utilize transcoding server(s) to transcode the media asset, which can then be moved to a CDN and/or stored locally by storage array 230 and data object origin server(s) 240. Services running on the processing server(s) 220 can also update the associated data object stored by the storage array 230 and data object origin server(s) 240.

[0036] FIG. 2B is a block diagram illustrating an alternative embodiment of a kernel application center 111-2. In addition to the components of the embodiment of FIG. 2A, this embodiment incorporates an application center 112 within the kernel application center 111-2. The application center 112 incorporated within kernel application center 111-2 may be located at or near the other components of the kernel application center 111-2, and can be communicatively connected to the other components via network 260. The incorporated application center 112 can therefore have faster access to kernel application center functionality because it does not need to communicate over long distances. In consideration of this advantage, it will be understood that the CHIMPS 110 can include multiple kernel centers with one or more application centers incorporated therein. Additionally or alternatively, components of the kernel application center may be incorporated into one or more application centers 112 in the CHIMPS 110 to provide quicker access to certain functionality.

[0037] FIG. 3 is a block diagram illustrating an embodiment of an application center 112. The application center 112 can include caching server(s) 330 and a storage array 310 for storing and distributing data objects of media assets requested by end users through end user interface 360. Caching server(s) 330 and storage array 310 can also be used to collect, process, and/or store metrics information from beaconing data, media chunk requests, and/or other data sources, including data collected through end user interface 360. The application center can further include ingest server(s) 320 for ingesting uploaded media assets from a content provider 130 through a content provider interface 370. The media assets may be stored on the storage array 310. As with the kernel application center 111, the components of the application center 112 can be communicatively linked through a network 340, such as a LAN. The application center can further include an internal interface 350, providing a communication link from the application center to the rest of the CHIMPS. It is through internal interface 350, for example, that media assets stored on storage array 310 can be made available to a kernel application center 111 for services such as transcoding.

[0038] FIG. 4A is a block diagram 400-1 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Although FIG. 4A further indicates the physical systems in which may execute or store these processes and objects, it will be understood that the processes and objects disclosed may be executed or stored on more than one system, including systems not disclosed in FIG. 4A. In other words, the processes and objects shown in FIG. 4 allow for a variety of implementations through one or more of hardware, software, firmware, microcode, etc.

[0039] Media can be ingested into the CHIMPS 110 when a content provider 130 uploads a media asset to ingestion server(s) 410 in an application center 112 by utilizing a client 405. The client 405 can be a stand-alone application or browser based, for example, and can communicate with ingest server(s) 410 through an application programming interface (API) configured for the ingestion of media assets.

[0040] Ingest server(s) 410 can communicate with devices in the kernel application center 111 executing programs such as kernel server 425 and file replication service 430. The kernel server 425 can be configured to organize the workflow among services such as transcoding 440 file system manager 435, and other services 445 (e.g., analytics, dynamic API, etc.) Upon a

particular event, for example, the kernel server can be configured to notify the relevant services of the event, causing the services to process tasks associated with the event.

[0041] The file replication service 430, under direction of the kernel server 425, can coordinate the movement of the media assets between services. For example, retrieving the uploaded media asset from the ingest server(s) 410 and storing it on the file archive 450, or retrieving transcoded media assets from transcoding server(s) 460 and storing them in the media asset origin.

[0042] The data object updater 420 keeps the data object origin 415 up to date in response to any changes in the system. When, for example, a file is uploaded, transcoded, and stored in media asset origin 455, the location and other metadata concerning the transcoded media assets need to be created or updated in the data object origin 415 to ensure an end user who accesses the object in the data object origin 415 has the correct information regarding the related media asset. Because the data object updater 420 receives updates from the kernel server 425 (which is notified when a transcoded media asset is stored in the media asset origin 455, the system ensures the data objects in the data object origin are constantly up to date.

[0043] The upload of a media asset to the ingest server(s) 410, as described above, can provide an example of how the kernel server 425 may coordinate workflow. For instance, in response to the upload, the ingest server(s) 410 can notify the kernel server 425 that a media asset has been uploaded. The kernel server 425 informs the file replication service 430 of the uploaded media asset, and the file replication service 430 moves the uploaded media asset into the file archive 450 and notifies the kernel server 425 of the move. In response, the kernel server 425 notifies the file replication service 430, the file system manager 435, and the transcoding master 440 of the move. The file replication service 430 then will know it can delete the uploaded media asset from the ingest server(s) 410, the file system manager 435 will update the file system accordingly, and the transcoding master 440 will notify transcoding service(s) 460 of different transcoding tasks to be performed. The transcoding service(s) 460 can then retrieve the uploaded media asset from the file archive 450 to create transcoded media assets. The transcoding service(s) 460 notify the kernel server 425 once transcoding is complete, and the kernel server 425 relays this information to the file replication service 430. The file replication service 430 then takes the transcoded media assets from the transcoding services 460 and moves them to the media asset origin 455. Once the file replication service 430 notifies the kernel server 425 of the move, the kernel server 425, in turn, notifies the file replication service 430 and the data object

updater 420. The data object updater 420, which updates the data object origin 415 accordingly, and the file replication service 430 deletes the transcoded media assets from the transcoding services 460.

[0044] The modular nature of the system enables all tasks associated with an event to be completed quickly. As illustrated in the example above, workflow relating to a particular event, such as a media asset upload, can be spread among the various services simultaneously. Moreover, because the system's modularity enables it to be scaled to accommodate differing hardware capacities, and because the system can be configured to dynamically allocate hardware to different services according to the needs of the system, the speed of completing tasks relating to a particular event can further be increased.

[0045] FIG. 4B is a block diagram 400-2 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Block diagram 400-2 may be substantially similar to block diagram 400-1 except for the use of a content delivery network in place of media asset origin 455. Therefore, in block diagram 400-2, as opposed to the media assets being stored at media asset origin 455, they are stored at content data network 150, which may be operated and maintained by a third party.

[0046] FIG. 4C is a block diagram 400-3 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Block diagram 400-3 may be substantially similar to block diagram 400-1 except for the use of a content delivery network in conjunction with media asset origin 455. Here, in block diagram 400-3, media assets may reside at either media asset origin 455, content delivery network 150, or both.

[0047] Embodiments of such systems may include other systems that manage various requests from end users. One such system may be a cloud data persistence system. Referring to FIG. 5A, a simplified block diagram of an embodiment of a cloud data persistence system 500-1 is illustrated. Such a system may allow for information, such as data objects, to be efficiently distributed to multiple of clients.

[0048] Some embodiments of cloud data persistence systems are used to distribute data objects that are linked to media assets. A data object may contain information pertinent to the data object it is linked to. For example, a data object may include details about a media asset and permissions for the media asset. More specifically, a data object may include fields containing: a title, keywords linked to the data object, the data object's (and/or associated media asset's) date

of creation, the duration of the associated media asset, the file formats the associated media asset is available in, and what clients are permitted to access and/or edit the data object and/or the media asset. The data object may also contain a link to its associated media asset. Such a link may be in the form of a universal resource indicator, such as a universal resource locator. Other information may also be stored as a data object that is linked to a media asset. A media asset may represent many different forms of media, such as an audio file (in a variety of different formats), a video file, or a flash file, to name only a few examples.

[0049] Important to note, data objects may be stored separate from media assets. For example, a data object containing information linked to a particular media asset may not be stored in the same location as the media asset. For example, a media asset may reside on multiple servers that are part of a content delivery network, while the data object that contains information about the media asset is stored in some other location, possibly on some other network. Also, while the above discusses linked media assets, it should be understood that data objects may also be linked to other assets, files or data that do not constitute media. In such embodiments, the data objects may contain different fields of data.

[0050] Such a cloud data persistence system may be used to distribute data objects to one or several clients, but also may be used to distribute data objects to thousands of clients geographically disbursed worldwide. In the embodiment of cloud data persistence system 500-1, three levels of servers are present: a data object origin server 510, data persistence cache servers 520, and application centers 530. These application centers are in contact with one or more clients 540. It should be understood that each application center may include one or more servers. For example, an application center may be represented by application center 112 of FIG. 3. Alternatively, an application center may comprise a single server. While three levels of servers are present in the illustrated embodiment, it should be understood that greater or fewer levels of servers may be present in other embodiments. Also, various servers may be integrated into the same server. For example, one server may serve as both an application center 530 and a data persistence cache server 520. It should also be understood that the terms of "application center," "data persistence cache server," and "data object origin server" are used to simplify the nomenclature used to describe the various levels of the cloud data persistence system 500-1, and these terms should not be interpreted as limiting the functionality of their respective identified servers.

[0051] Referring to cloud data persistence system 500-1, clients 540 make requests for data objects from application centers 530 via one or more networks. In the illustrated embodiment, only two clients (540-1 and 540-2) are present to maintain simplicity; many more clients (possibly representing many end users) may be possible. These clients may communicate with the same application center (530-1) or may communicate with different application centers (530-1 and 530-2). Further, because a client at one time interacts with a particular application center, at a different time, the same client may interact with a different application center. For example, if a particular application center is experiencing a heavy client load, possibly caused by a large number of clients attempting to access it, a client may be routed to another application center.

[0052] The client may make a request for a particular data object to the application center. This request may be made via a network, such as the Internet. Other networks may also be possible, such as a private intranet. The request for the data object may involve the client sending a universal resource indicator (URI) to an application center. A URI may take a variety of different formats. In particular, the URI may be in the form of a universal resource locator (URL). However, it should be understood that other forms of URIs may also be possible.

[0053] If client 540-1 transmits a URI to application center 530-1, application center 530-1 may have the data object linked to the URI stored locally. If this is the case, application center 530-1 may respond by sending the data object identified by the URI back to client 540-1. However, application center 530-1 may have only a subset of the data objects linked to valid URIs stored locally. This may be due to space constraints at application center 530-1. For example, if clients infrequently request particular data objects, it may be inefficient to distribute those data objects to each of application centers 530.

[0054] If the data object requested by client 540-1 is not present at application center 530-1, application center 530-1 may contact another server to retrieve it. Application center 530-1 may be located in close proximity to data persistence cache server 520-1, or may be geographically separated from data persistence cache server 520-1. Application center 530-1 may communicate with data persistence cache server 520-1 directly (such as a direct cable connection) or may communicate using a network. This network may be a private network, such as a local area network or dedicated connection between the servers, or may use a public network, such as the Internet. Such a request from application center 530-1 to data persistence cache server 520-1 may include the URI received by application center 530-1 from client 540-1. In some

embodiments, data persistence cache servers supply multiple application centers with requested data objects. In cloud data persistence system 500-1, data persistence cache server 520-1 serves two application centers (530-1 and 530-2), while data persistence cache server 520-2 serves two different application centers (530-3 and 530-4). It should be understood that a data persistence
5 cache may serve a varying number of application centers. For example, a data persistence cache may serve 20 application centers.

[0055] It should also be recognized that the connections between application centers 530 and data persistence cache servers 520 are dynamic. If a particular data persistence cache is offline, busy, or otherwise unavailable, an application center may be able to route requests to a different
10 data persistence cache server. For example, referring to cloud data persistence system 500-1, if data persistence cache 520 became unavailable, application center 530-1 may be able to route a request for the data object to data persistence cache server 520-2.

[0056] While, if application centers 530 and data persistence caches 520 communicate via a network (e.g., the Internet), the servers may be located anywhere a network connection is
15 available worldwide. In some embodiments, a data persistence cache is located near a group of application centers. For example, if a region, such as New England, contains two application centers for each state, a single data persistence cache may be located in or near New England to serve these twelve scattered application centers. Having the data persistence cache located near a group of application centers may decrease communication latency when an application center
20 and the data persistence cache communicate.

[0057] Upon receiving a request for a data object from application center 530-1, data persistence cache server 520-1 may determine if it has the data object stored. If it does, data persistence cache server 520-1 may send the data object to application center 530-1, which may, in turn, send the data object to client 540-1. Upon the application center receiving the data
25 object, it may store the data object to be able to satisfy requests from the same or a different client for the same data object at a future time. Application center 530-1 may make a determination as to whether the data object should be stored or not. For example, the determination may be based on the frequency it is receiving requests for the data object from clients 540. If the data object is frequently being requested by clients 540, application center
30 530-1 may store the data object. This may only occur if the data object is being requested more frequently than some other data objects. For example, only the 10,000 most frequently requested

data objects may be stored by the application center. In some embodiments, application center 530-1 may store the most recently requested data objects. Therefore, this data object would be stored at application center 530-1, and possibly some other data object would no longer be stored at application center 530-1. In some embodiments, different application centers use different
5 mechanisms to determine what data objects to store. Also, in some embodiments, some data objects are always stored at application centers 530. This may occur if a particular data object has been labeled a "high priority" (or some equivalent thereof).

[0058] However, if data persistence cache server 520-1 determines that it does not have the data object requested by client 540-1 stored, data persistence cache server 520-1 may contact
10 another server. Data persistence cache server 520-1 may contact another server called a data object origin server 510. Data object origin server 510 may supply data persistence cache servers 520 with various data objects. In cloud data persistence system 500-1, data object origin server 510 is shown as communicating with two data persistence cache servers 520. However, it should be understood that data object origin server 510 may communicate with one, three, or
15 some other number of data persistence cache servers. For example, data object origin server 510 may communicate with more, such as 20 data persistence cache servers.

[0059] Data object origin server 510 may be physically located with one or more data persistence cache servers. Data object origin server 510 may communicate directly, such as via a direct connection, with data persistence cache servers. For example, in cloud data persistence
20 system 500-1, data object origin server 510 communicates directly with the data persistence cache server 520-2. Data object origin server 510 may also communicate via any of the previously mentioned networks with a data persistence cache server. Again, referring to cloud data persistence system 500-1, data object origin server 510 communicates with data persistence cache server 520-1 via a network, possibly the Internet.

25 [0060] While application centers 530 and data persistence cache servers 520 may store only a subset of the various data objects that clients 540 may request, data object origin server 510 may store all data objects that exist on system 500-1. In other words, if a data object requested by a client is not present on data object origin server 510, the data object may not exist on system 500-1.

30 [0061] Just as a data persistence cache server may be physically located near a group of application centers, data object origin server 510 may be located near the group of data

persistence cache servers 520. Returning to the previous regional example, if a data persistence cache server is located in New England, another may be located near New York City, another near Washington DC, and perhaps yet another near Chicago. If one data object origin server is servicing these data persistence cache servers, it may be located in a central location to serve them all. In some embodiments, the data object origin server may be co-located with one or more data persistence cache servers. Alternatively, a data object origin server may be located at a separate physical location from any other database servers.

[0062] A request from the data persistence cache server 520-1 to data object origin server 510 may comprise a URI. This may be the same URI that was sent from client 540-1 to application center 530-1, then from application center 530-1 to data persistence cache server 520-1. In some embodiments, the URI transmitted from data persistence cache 520-1 may be in a different format than the other URIs. Upon receiving the URI, data object origin server 510 may locate the data object the URI is linked to. The data object origin 510 may then transmit the data object to data persistence cache server 520-1. The data persistence cache server 520-1 may make determination as to whether to store the data objects as previously described in relation to application center 530-1. For example, the determination of whether to store the data object may be based on the frequency with which application centers 530 request the data object from data persistence cache server 520-1. In some embodiments, data persistence cache server 520-1 stores the most recently requested data objects. Data persistence cache server 520-1 may then transmit the data object to application center 530-1. Application center 530-1 may then transmit the data object to client 540-1.

[0063] If the data object was first found at application center 530-1, the latency between client 540-1's request and client 540-1 receiving the data object may be the shortest. If instead the data object is first found at data persistence cache server 520-1, the latency may be longer. Finally, if the data object is only found at data object origin 510, the latency may be the longest. Regardless of where the data object is found, client 540-1 may be unaware of whether application center 530-1, a data persistence cache server 520-1, or the data object origin server 510-1, had the data object stored.

[0064] While only one data object origin server 510 is present in cloud data persistence system 500-1, more than one data object origin server is possible. For example, a data object origin server may serve a particular region, such as a country. Other data object origin servers may be

present in other countries. These data object origin servers may each maintain identical or substantially similar data sets. Therefore, a data object found on one data object origin server would also be found on other data object origin servers. While not illustrated in FIG. 5, it should be understood that data object origin server 510 may be in communication with a kernel server and/or a data object updater server, such as illustrated in FIGs. 2A, 2B, and 4. Such an arrangement may allow multiple different data object origin servers to have the same data objects stored.

[0065] While the above deals with the communication flow in cloud data persistence system 500-1 when a client requests a data object, a client (or, of course, an end user acting through the client) may also modify a data object. Depending on the data object, a request for a particular data object may be much more frequent than a request to modify a data object. By way of example only, consider a video clip posted on an online video sharing website. If the video is popular, data object information, such as the video owner's name and the video's title may be sent to millions of different clients. However, the end user who created the video may only infrequently modify (via a client) data linked to the video.

[0066] In cloud data persistence system 500-1, the request to modify a data object, as opposed to a request to retrieve a data object, may be treated differently. If client 540-1 makes a request to application center 530-1 to modify a data object, application center 530-1 (assuming the end user and/or client has the proper permissions to modify the data object) may route the request to Kernel application center 111. Kernel application center 111 may then update data object origin server 510, and any other data object origin servers requiring updating. Application center 530-1 may not modify or delete the previous version of the data object if it is stored at application center 530. The data object may then be updated at data object origin server 510. While the data object may now be updated at data object origin server 510, other application centers and data persistence cache servers, such as application center 530-3 and data persistence cache 520-1, may still have out-of-date versions of the data object. Therefore, upon receiving the modifications to the data object, kernel application center 111 may notify data persistence cache servers 520 to no longer use their current version of the data object. This may involve data persistence cache servers 520 deleting their out-of-date version of the data object. In turn, data persistence cache servers 520 may notify application centers 530 to similarly no longer use their out-of-date versions of the data object. This may again involve the out-of-date data object being deleted from application centers 530. Application centers 530 and data persistence cache servers

520 may not receive a new version of the data object along with the notification. Rather, application centers 530 and data persistence cache servers 520 may only receive the new version of the data object if a client requests the data object and it is routed through the data persistence cache server and the application center.

5 [0067] If kernel application center 111 knows what particular data objects are stored at data persistence cache servers 520, a notification may only be sent to the data persistence cache server known to store the data object. Similarly, if data persistence cache servers 520 know what data objects are stored at application centers 530, only those application centers that have out-of-date versions of the data object may be sent the notification. In some embodiments, whenever a
10 modification of a data object is received at kernel application center 111, all data persistence cache servers 520 are notified, which in turn notify all application centers 530. In such embodiments, therefore, kernel application center 111 does not need to be aware of what particular data objects are stored at data persistence cache servers 520, and data persistence cache servers 520 do not need to be aware of what data objects are stored at application centers
15 530. In some embodiments, Kernel application center 111 notifies both data persistence cache servers 520 and application centers 530.

[0068] Referring to FIG. 5B, a simplified block diagram of an embodiment of a cloud data persistence system 500-2 is illustrated. Such a system may be substantially similar to system 500-1. However, system 500-2 illustrates how various components of system 500-2 may be
20 geographically separated. It should be understood that each component of system 500-2 may be located at a different physical location. In some embodiments, client 540-1 is at location 551 while client 540-2 is at location 552. Locations 551 and 552 may be close together, possibly only several miles if both client 540-1 and client 540-2 are using the same application center. Clients 540-1 and 540-2 may be a great distance apart if they are using different application
25 centers.

[0069] Application centers 530-1, 530-2, 530-3, and 530-4 are shown as different locations of 553, 554, 555, and 556, respectively. Application centers that use the same data persistence cache server may be closer together than application centers that do not use the same data persistence cache servers. Further, as those with skill in the art will recognize, while an
30 application server may not be physically closer to a data persistence cache server, due to the state

of the network, communication between two physically further apart servers may be quicker. In such a case, the further servers may be in communication as opposed to the closer servers.

[0070] Data persistence cache server 520-1 is at another location 557. Location 557 may be near or in between location 553 and 554. Data persistence cache server 520-2 is at the same location 555 as application center 530-3. In some embodiments, data persistence cache server 520-2 is at some other location from application center 530-3. Finally, data object origin server 510 may be present at location 558. This location may be at the kernel application center 111, or may be separate.

[0071] FIG. 6 illustrates a simplified swim diagram of an embodiment of a method 600 for responding to a client request to retrieve a data object. Such a method may use a cloud data persistence system, such as cloud data persistence system 500-1 of FIG. 5A. Alternatively, method 600 may rely on some other cloud data persistence system. The embodiment of FIG. 6 presupposes that the data object requested is not present on either the application center contacted or the cache server contacted. As described in relation to FIG. 5, if one of these servers has the data object stored, the method would differ.

[0072] At block 605, a request may be sent from a client to an application center. The client may be client 540-1 or 540-2 of FIG. 5, or may represent some other client. Similarly, the application center of FIG. 6 may represent any of application centers 530 of FIG. 5, or may represent some other application center. The request sent by the client at block 605 of FIG. 6 may contain a URI, possibly a URL, indicating the data object the client is requesting to retrieve. The request may be received either directly or via a network (such as the Internet), by the application center at block 610.

[0073] At block 615, the application center may determine if the data object requested by the client is present. This may involve the application center searching a storage device, such as a hard drive or solid state drive for the data object. In some embodiments, the application center may search an index of all the data objects it has stored. In this embodiment, the application center does not have the data object stored locally. Therefore, at block 620, a request is sent from the application center to a data persistence cache server. This request may contain the same URI received in the request from the client at block 610. The cache server contacted may be a data persistence cache server 520 of FIG. 5A or may represent some other data persistence cache

server. The request by the application center may be received by the data persistence cache server either directly or via a network (such as the Internet) at block 625.

[0074] At block 630, the data persistence cache server may determine if the data object requested by the application center is present. This may involve the data persistence cache server searching a storage device, such as a hard drive or solid state drive for the data object. In some embodiments, the data persistence cache server may search an index of all the data objects it has stored. In this embodiment, the data persistence cache server does not have the data object stored locally. Therefore, at block 635, a request is sent from the data persistence cache server to an data object origin server. This request may contain the same URI received in the request from the client at block 610 and the application center at block 625. The data object origin server contacted may be the data object origin server 510 of FIG. 5A or may represent some other data object origin server. The request by the data persistence cache server may be received by the object origin server either directly or via a network (such as the Internet) at block 640.

[0075] At block 645, the data object origin server may locate the data object requested. Assuming that the URI indicating the data object initially requested by the client is valid, the object origin server should have the data object stored. If the data object is not present at the data object origin server, the data object may be assumed to not exist or that the URI submitted by the client is incorrect. To retrieve the data object, the data object origin server may search one or more local (or remote) hard drives or solid state drives for the data object. In some embodiments, the data object origin server uses an index to locate and retrieve the data object. Once the data object has been located, at block 650 it may be transmitted to the data persistence cache server.

[0076] At block 655, the data persistence cache server may receive the data object. The data received from the data object origin server may also include the URI sent in the request at block 635. At block 660, the data persistence cache server may determine whether the data object should be stored locally. If the data persistence cache server stores the most recent data objects requested, the data persistence cache server may always store the data object when it is received from the data object origin server. If the data persistence cache server stores only the most frequently requested data objects, the data object may not be stored if other data objects are requested more frequently. As those with skill in the art will understand, there are other factors that may be used to determine whether the data object is stored locally at the data persistence

cache server. As another example, the decision as to whether to store the data object may be based partially on frequency of requests for the data object and partially on the data object being recently requested. Whether the data object is stored or not, it may be transmitted to the application center at block 665.

5 [0077] At block 670, the application center may receive the data object. This may be the same application center that received the request at block 610. The data received from the data persistence cache server may also include the URI sent in the request by the application center at block 620. At block 675, the application center may determine whether the data object should be stored locally. The factors used to consider whether to store the data object may be the same or
10 different from the factors used at the data persistence cache server. Again, if the application center stores the most recent data objects requested, the application center may always store data objects received from the data persistence cache server. If the application center stores only the most frequently requested data objects, the data object may not be stored if other data objects are requested by clients more frequently. As those with skill in the art will understand, there are
15 other factors that may be used to determine whether the data object is stored locally at the application center. As another example, the decision as to whether to store the data object may be based partially on the frequency of requests for the data object and partially on the data object being recently requested. Whether the data object is stored by the application center or not, it may be transmitted to the client at block 680. Further, to be clear, while it is possible that the
20 same data objects may be stored at both the application and the data persistence cache server, it is likely that at least some different data objects would be stored at the application center than the data persistence cache server. Also, the amount of storage available at the application center and the data persistence cache server may vary. For example, the data persistence cache server may have 50% more storage space than an application center; therefore, it may store roughly 50%
25 more data objects.

[0078] At block 685, the client may receive the data object it requested at block 605. The client may be unaware as to whether the data object was present at the application center, the data persistence cache server, or the data origin cache server. To the client, it may appear as if the client is only interacting with the application center.

30 [0079] While FIG. 6 illustrates an embodiment of a method of a client requesting a data object, assuming the associated end user and/or client has the proper permissions, the end user and/or

client may also request to modify a data object. FIG. 7 illustrates a swim diagram of a simplified embodiment of a method 700 of a client modifying a data object. Method 700 may be conducted using a cloud data persistence system, such as cloud data persistence system 700 of FIG. 7. Alternatively, some other cloud data persistence system may be used. A modification may

5 involve changing data within a data object, adding data to a data object, or deleting the data object, to name only a few examples. It should be understood that in some embodiments, modifications to data objects are much less frequent than requests to retrieve a data object. At block 705, the client may transmit the request to modify the data object to an application center. The client may be the same client as detailed in FIG. 6 or client 540-1 or 540-2 of FIG. 5.

10 Alternatively, the client of FIG. 7 may represent some other client. Similarly, the application center of FIG. 7 may represent the application center of FIG. 6 or any of application centers 530 of FIG. 5. The application center of FIG. 7 may also represent some other application center.

[0080] At block 710, the application center may receive the modification request from the client. The request may include a URI (such as a URL) indicating the data object the client is

15 attempting to modify. The request may also include the changes, additional information, or an indication to delete the data object. If the application center determines it has the data object stored, at block 712, the application center may delete or block access to the data object so that other clients do not receive an out-of-date version of the data object. At block 715, a modification request may be transmitted from the application center to the data object origin

20 server. This transmission may be routed from the application to the data object origin server (e.g., not through the data persistence cache server) or may be relayed by the data persistence cache server.

[0081] At block 720, the kernel application center may receive the modification request. The kernel application center may be the kernel application center 111 of FIG. 5A and FIG. 5B.

25 Kernel application center of FIG. 7 may also represent a data object origin server. After receiving the modification request at block 720, the kernel application center may modify the data object at one or more data object origin servers at block 725. Data object origin servers may always have a stored version of the data object because they maintain a stored version of all valid data objects. If the data object origin server does not have a version of the data object requested

30 to modify, it may be assumed that the data object does not exist or the URI sent by the client at block 705 was incorrect. The modification at block 725 may involve the data object being changed, having data added to it, or the data object being deleted. At block 730, the modified

data object may be stored at one or more data object origin servers. Obviously, if the data object is being deleted, this step may not occur.

5 [0082] At block 735, a notification may be sent by the kernel application center that indicates that the data object has been modified. Of note, this modification may not contain the modified data object, but rather may contain only information that indicates that previous versions of the data object are no longer valid. This notification may be transmitted to one or more data persistence cache servers at block 735. While the swim diagram illustrates the notification being sent to only one data persistence cache server, this is for simplicity only. The notification transmitted by the data object origin server may be sent to all data persistence cache servers (or those data persistence cache servers that have the previous version of the data object stored) in communication with the data object origin server. The notification may only be sent to those data persistence cache servers that have an out-of-date version of the data object if the kernel application center is aware of which data persistence cache servers have the data object stored. This may be possible if the data persistence cache server informs the kernel application center of its stored data objects. Alternatively, all data persistence caches may receive the notification whether they have the previous version of the data object stored or not. If a data persistence cache server does not have the previous version of the data object stored, the data persistence cache server may ignore the notification.

15 [0083] At block 740, the notification may be received by the data persistence cache server. The data persistence cache server may be the data persistence cache server of FIG. 6 and/or one of the data persistence cache servers 520 of FIG. 5. The data persistence cache server of FIG. 7 may represent some other data persistence cache server. At block 745, the previous version of the data object may be cleared (e.g., deleted or otherwise removed) from the data persistence cache server. At block 750, a notification that the previous version of the data object is out-of-date may be transmitted to one or more application centers at block 750.

20 [0084] While the swim diagram illustrates the notification being sent to only one application center, this is for simplicity only. The notification transmitted by the data persistence cache server may be sent to all application centers (or those application centers that have the previous version of the data object stored) in communication with the data persistence cache server. The notification may only be sent to those application centers that have the previous version of the data object if the data persistence cache server is aware of which application centers have the

previous version of the data object stored. This may be possible if the application center informs the data persistence cache server (or the kernel application center and/or data object origin server) of its stored data objects. Alternatively, all application centers may receive the notification whether they have the previous version of the data object stored or not. If an application center does not have the previous version of the data object stored, the application center may ignore a received notification. The notification sent by the data persistence cache server at block 740 may represent the same or a similar notification as sent by the kernel application center at block 735. For example, the notification may include a URI (possibly a URI) indicating the data object that has been modified. At block 760, the previous version of the data object may be cleared (e.g., deleted or otherwise demarked) from the application center.

[0085] While the previous description and figures relate to how a client may retrieve and modify a data object, the storage of data objects and other data and/or files may be more sophisticated than one server (e.g., an application center, a data persistence cache server, or a data object origin server) having one or more storage devices. For example, each application center of FIG. 5A may represent a cluster of servers, each connected to one or more storage devices, such as hard drives or solid state drives. FIG. 8 illustrates a simplified block diagram of an embodiment of an asset management and storage system 800 showing how various storage devices may be used to store data and accessed. While such a system may be used to store data objects, such as those data objects described previously, such a system may be used to store other forms of electronic data, such as media assets. A media asset may be a media file, such as a movie file, audio file, or image file.

[0086] In system 800, multiple different storage devices 810 may be present. These storage systems may be servers having storage devices such as hard drives and/or solid state drives. Storage devices 810-1 and 810-2 may be located in the same location or may be physically in different locations. Each of storage devices 810 may be linked to one or more storage servers 820. These storage servers 820 may not be physical servers, but rather logical abstractions of servers. Alternatively, they may be physically separate servers. Various storage drives on storage servers 820 may be mapped to different storage devices. For example, storage server 820-1 is linked to one storage drive. The storage drive is present on storage device 810-1. Storage server 820-1 and storage device 810-1 may be directly connected, or may be connected via a network, such as the Internet or a private corporate intranet. In system 800, storage server 820-2 has two storage drives. A first storage drive of storage server 820-2 maps to storage

device 810-1. Therefore, in the embodiment of the system 800, storage device 810-1 is mapped to two different storage servers. If, for example, storage device 810-1 has storage totaling 1 TB, this gross storage space may be divided among the two drives mapped to it. For example, the storage drive of storage server 820-1 mapped to storage device 810-1 may be 300 GB. And, the storage drive of storage device 820-2 mapped to storage device 810-1 may be 700 GB. Storage server 820-2 may also have another storage drive. The storage drive is mapped to storage device 810-2. In this embodiment, storage device 810-2 is only mapped to storage server 820-2.

Therefore, if storage device 810-2 has total storage of 2 TB, this entire space may be mapped to the linked storage drive of storage server 820-2. Or, as another example, only 800 GB of storage device 810-2 may be mapped to storage server 820-2, with the remaining space being free space.

[0087] Also, different storage devices may require communication, such as requests for media assets, to use varying protocols. For example, storage device 810-1 may be configured to accept file requests via FTP. However, storage device 810-2 may use MICROSOFT WINDOWS's file copy. It should also be understood that other protocols may also be used. Therefore, depending on what protocol a storage device accepts, the storage server links to that storage device may be configured to communicate in the proper protocol. This may be especially useful if the entity operating the storage servers 820 does not control one or more of storage devices 810. For example, storage device 810-2 may be maintained by some other entity, such as a client of the entity operating system 800.

[0088] Other logical abstractions that may be present are storage systems 830. Each storage system, such as storage system 830-1, may contain one or more than one storage servers 820. In system 800, storage system 830-1 includes two storage servers: 820-1 and 820-2. Therefore, a request for a particular piece of data, such as a media asset, received at storage system 830-1, may not indicate the storage server or the storage device where the media asset is located. However, upon receiving the request (possibly in the form of a URL) at storage system 830-1 for the media asset, it may be able to determine the appropriate storage server to contact. In turn, this storage server may be able to determine the appropriate storage drive (linked to a storage device) that contains the media asset. The logical abstractions of the storage system 830 and storage servers 820 may reside on the same physical servers as storage devices 810. In some embodiments, storage server 820 and storage system 830 are present on different servers than storage device 810. Also, storage systems 830 may be physically separate servers from storage servers 820.

[0089] Yet another logical abstraction may be present as storage cloud 840. Storage cloud 840 may include one or more storage systems. Storage cloud 840 may be viewed as the entire storage system. In system 800, storage cloud 840 includes storage systems: 830-1 and 830-2. Therefore, storage cloud 840 may contain all storage systems 830 in system 800. In turn, storage systems 830 may contain all storage servers 820 in system 800. Further, storage server 820 may be linked to all storage devices 810 in system 800.

[0090] FIG. 9 illustrates another simplified block diagram of an embodiment of a system 900 showing how various storage devices may be accessed. System 900 of FIG. 9 may represent the same system as system 800 of FIG. 8, or may represent a different system. In the embodiment illustrated, system 900 contains four storage devices: 910-1, 910-2, 910-3, and 910-4. These storage devices may represent any of storage devices 810 of FIG. 8. Also, these storage devices may represent different storage devices. As will be understood to those with skill in the art, fewer or more storage devices may also be possible. In the illustrated embodiment, storage device 910-1 and storage device 910-2 are illustrated as being operated by first entity 915. First entity 915 may be the entity that operates other portions of system 900, or is a client of the entity that operates other portions of system 900. These storage devices, 910-1 and 910-2, may be co-located with one or more storage servers 920, or may be geographically separated. Further, storage device 910-1 and storage device 910-2 may be present at the same location, or may be geographically separated. Storage device 910-3 may be operated by a second entity 916. In some embodiments, the second entity 916 may be a client of the first entity. In system 900, storage device 910-3 may communicate with other systems 917 of the second entity 916. Therefore, while storage servers 920 may be able to access storage device 910-3, client system 917 may also access and/or modify data stored on storage device 910-3. Storage device 910-4 is illustrated as part of a third entity 918. This entity may be an entity that sells or leases storage on their system. While FIG. 9 illustrates four storage devices operated by various entities, those skilled in the art will recognize that this arrangement is merely for exemplary purposes. Many other arrangements of storage devices, operated by one or more entities may be possible.

[0091] Whether operated by three different entities, as illustrated, or by one entity, storage devices 910 may communicate with multiple storage servers 920. System 900 includes four storage servers: 920-1, 920-2, 920-3, and 920-4. The storage servers 920, which may be logical abstractions, may communicate with the various storage devices 910 via a direct connection (such as 920-1 to storage device 910-1) or via a network (such as 920-2 to storage device 910-3).

The network may be a network such as the Internet or a private corporate intranet. One storage server may be mapped to one storage device, for example, in FIG. 9, storage server 920-1 only has one storage drive mapped to storage device 910-1. A storage device may also be mapped to multiple storage servers, for example, in FIG. 9, storage servers 920-2 and storage server 920-4 are both mapped to storage device 910-3. Also, one storage server may be mapped to multiple storage devices. For example, storage server 920-4 is mapped to storage device 910-3 and storage device and 910-4.

[0092] Also, storage servers 920 may communicate with storage devices 910 via a variety of different protocols. For example, the format of requests to some storage devices may be FTP, while others may be MICROSOFT WINDOWS file copy. Other protocols may also be possible. Therefore, any request for data or a media asset that is received by storage servers 920 does not need to indicate the appropriate protocol to retrieve the file or media asset from the appropriate storage device.

[0093] As in FIG. 8, one or more than one storage servers may be mapped to a storage system 930. Storage systems 930 may represent the same storage systems as storage systems 830 of FIG. 8. Alternatively, storage systems 930 may represent some other storage systems. In system 900 of FIG. 9, storage servers 920-1 and storage server 920-2 are mapped to storage system 930-1. Storage system 930-1 may communicate directly with the storage server (such as the connection between storage system 930-1 and storage server 920-1) or may use a network (such as the connection between storage system 930-1 and storage server 920-2). In system 900, storage system 930-2 may communicate with two storage servers: 920-3 and 920-4. While in system 900 storage systems 930 are shown as communicating with two storage servers each, those with skill in the art will recognize that a storage system may also communicate with one or more than two storage servers.

[0094] Storage systems 930 may make up storage cloud 940. Storage cloud 940 may represent the same storage cloud as storage cloud 840 of FIG. 8. Alternatively, storage cloud 940 may represent some other storage cloud. Therefore, storage cloud 940 may be viewed as including storage systems 930, storage servers 920, and storage devices 910.

[0095] Other systems may interact with various components of storage cloud 940. For example, a content data network 960 may communicate with storage system 930-1. If the client requests the media asset from content data network 960 (CDN) that the CDN does not have

stored (a cache miss), the content data network 960 may request the media asset from storage system 930-1. Storage system 930-1 may then contact the appropriate storage server 920, which may, in turn, retrieve the media asset via the correct protocol from the appropriate storage device. From the viewpoint of CDN 960, it may appear that the media asset is stored at storage system 930-1. While CDN 960 is shown only interacting with storage system 930-1, it should be understood that this is for simplicity only. Multiple CDN's may communicate with storage system 930-1. Further, CDN's may also communicate with storage system 930-2.

[0096] While CDN 960 may communicate with storage system 930-1, file replication service 950 may communicate directly with storage servers 920. File replication service 950 may be responsible for distributing various media assets to storage server 920 that need to be stored for possible future retrieval. While file replication service 950 is shown as connected to only storage server 920-1, it should be understood that file replication service 950 may be in communication with multiple storage servers 920. Besides file replication service 950, other services may interact directly with storage server 920.

[0097] FIG. 10 illustrates a simplified embodiment of a method for retrieving data (such as a media asset) from a storage cloud system. Such a method may be used in conjunction with a storage cloud system such as system 800 of FIG. 8, or system 900 of FIG. 9. The method 1000 may also be used in conjunction with some other storage cloud system. At block 1010, a request may be received for a piece of data, such as a media asset. Such a request may be received from an entity such as a content data network or some other entity accessing the storage cloud.

[0098] At block 1015, if the storage cloud contains more than one storage system, the appropriate storage system that contains the data requested may be identified. At block 1020, a request for the data may be sent to the appropriate storage system. At block 1025, the request may be received at the appropriate storage system. At block 1030, the storage system may determine the appropriate storage server that stores the data requested. Such a determination may involve an index being searched by the storage system to determine which storage server to contact. If the storage system only has one link to a storage server, such a determination may not be necessary. At block 1035, the request for the data may be sent to the appropriate storage server. This request may be received by the storage server at block 1040. At block 1045, a storage server may determine the appropriate storage device that stores the requested data. This step may not be necessary if only one storage device is linked to the storage server. At block

1050, the storage server may send a request for the data to the appropriate storage device. This request may be formatted using the appropriate protocol for the storage device receiving the request.

5 [0099] At block 1055, the request may be received at the appropriate storage device in the appropriate protocol. The storage device may then retrieve the appropriate data. At block 1060, the requested data may be transmitted from the storage device and received at the storage server at block 1065. The storage server may then transmit the requested data at block 1070, the requested data is then received by the storage system at block 1075. At block 1080, the requested data may be transmitted from the storage system to the requesting party, in this case
10 the CDN.

[0100] It should be noted that the methods, systems, and devices discussed above are intended merely to be examples. It must be stressed that various embodiments may omit, substitute, or add various procedures or components as appropriate. For instance, it should be appreciated that, in alternative embodiments, the methods may be performed in an order different from that
15 described, and that various steps may be added, omitted, or combined. Also, features described with respect to certain embodiments may be combined in various other embodiments. Different aspects and elements of the embodiments may be combined in a similar manner. Also, it should be emphasized that technology evolves and, thus, many of the elements are examples and should not be interpreted to limit the scope of the invention.

20 [0101] Specific details are given in the description to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these specific details. For example, well-known circuits, processes, algorithms, structures, and techniques have been shown without unnecessary detail in order to avoid obscuring the embodiments. This description provides example embodiments
25 only, and is not intended to limit the scope, applicability, or configuration of the invention. Rather, the preceding description of the embodiments will provide those skilled in the art with an enabling description for implementing embodiments of the invention. Various changes may be made in the function and arrangement of elements without departing from the spirit and scope of the invention.

30 [0102] Also, it is noted that the embodiments may be described as a process which is depicted as a flow diagram or block diagram. Although each may describe the operations as a sequential

process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be rearranged. A process may have additional steps not included in the figure. Furthermore, embodiments of the methods may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware, or microcode, the program code or code segments to perform the necessary tasks may be stored in a computer-readable medium such as a storage medium. Processors may perform the necessary tasks.

[0103] Having described several embodiments, it will be recognized by those of skill in the art that various modifications, alternative constructions, and equivalents may be used without departing from the spirit of the invention. For example, the above elements may merely be a component of a larger system, wherein other rules may take precedence over or otherwise modify the application of the invention. Also, a number of steps may be undertaken before, during, or after the above elements are considered. Accordingly, the above description should not be taken as limiting the scope of the invention.

[0104] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0105] The reference to any prior art in this specification is not, and should not be taken as, an acknowledgement or any form of suggestion that the prior art forms part of the common general knowledge in Australia.

WHAT IS CLAIMED IS:

- 1 I. A cloud data persistence system for distribution of data, the cloud data
- 2 persistence system comprising:
- 3 a first server which stores a first plurality of data objects and is communicatively
- 4 coupled with a plurality of cache servers, wherein:
- 5 the first server is located at a first location;
- 6 each of the first plurality of data objects comprise a link to one of a
- 7 plurality of media assets; and
- 8 the plurality of media assets are stored at a second location separate from
- 9 the first location;
- 10 a second server that stores a second plurality of data objects and is configured to
- 11 request a data object from the first server if the data object is not present in the second plurality
- 12 of data objects, wherein:
- 13 the second server is one of the plurality of cache servers; and
- 14 the second server is communicatively coupled with a first plurality of
- 15 application servers;
- 16 a third server that stores a third plurality of data objects and is configured to
- 17 request the data object from the first server if the data object is not present in the third plurality
- 18 of data objects, wherein:
- 19 the third server is one of the plurality of cache servers; and
- 20 the third server is communicatively coupled with a second plurality of
- 21 application servers; and
- 22 a fourth server that stores a fourth plurality of data objects and is configured to
- 23 request the data object from the second server if the data object is not present in the fourth
- 24 plurality of data objects, wherein:
- 25 the fourth server is one of the first plurality of application server; and
- 26 the fourth server is configured to receive a request from a first client for
- 27 the data object;
- 28 a fifth server that stores a fifth plurality of data objects and is configured to
- 29 request the data object from the second server if the data object is not present in the fifth plurality
- 30 of data objects, wherein:
- 31 the fifth server is one of the first plurality of application servers; and

32 the fifth server is configured to receive a request from a second client for
 33 the data object;
 34 a sixth server that stores a sixth plurality of data objects and is configured to
 35 request the data object from the third server if the data object is not present in the sixth plurality
 36 of data objects, wherein:
 37 the sixth server is one of the second plurality of application servers; and
 38 the sixth server is configured to receive a request from a third client for the
 39 data object; and
 40 a seventh server that stores a seventh plurality of data objects and is configured to
 41 request the data object from the third server if the data object is not present in the seventh
 42 plurality of data objects, wherein:
 43 the seventh server is one of the second plurality of application servers; and
 44 the seventh server is configured to receive a request from a fourth client
 45 for the data object.

1 2. The cloud data persistence system of claim 1, wherein the second location
 2 comprises a content delivery network.

1 3. The cloud data persistence system of claim 1, wherein:
 2 data objects in the second plurality of data objects are at least partially determined
 3 based on when requests for those data objects were last received at the second server;
 4 data objects in the third plurality of data objects are at least partially determined
 5 based on when requests for those data objects were last received at the third server;
 6 data objects in the fourth plurality of data objects are at least partially determined
 7 based on when requests for those data objects were last received at the fourth server;
 8 data objects in the fifth plurality of data objects are at least partially determined
 9 based on when requests for those data objects were last received at the fifth server;
 10 data objects in the sixth plurality of data objects are at least partially determined
 11 based on when requests for those data objects were last received at the sixth server; and
 12 data objects in the seventh plurality of data objects are at least partially
 13 determined based on when requests for those data objects were last received at the seventh
 14 server.

1 4. The cloud data persistence system of claim 1, wherein the first plurality of
2 data objects comprises all data objects in the second plurality of data objects, the third plurality
3 of data objects, the fourth plurality of data objects, the fifth plurality of data objects, the sixth
4 plurality of data objects, and the seventh plurality of data objects.

1 5. A cloud data persistence system for distribution of data, the cloud data
2 persistence system comprising:

3 a first server which stores a first plurality of data objects and is communicatively
4 coupled with a second server and a third server, wherein:

5 the first server is located at a first location;

6 each of the first plurality of data objects comprises a link to a media asset
7 of a plurality of media assets;

8 the plurality of media assets is located at a second location; and

9 the second location is distinct from the first location;

10 the second server stores a second plurality of data objects and is configured to
11 request a data object from the first server if the data object is not present in the second plurality
12 of data objects, wherein the second server is communicatively coupled with a fourth server and
13 fifth server; and

14 the fourth server stores a third plurality of data objects and is configured to
15 request the data object from the second server if the data object is not present in the third
16 plurality of data objects, wherein the fourth server is configured to receive a request from a first
17 client for the data object.

1 6. The cloud data persistence system of claim 5 wherein the first plurality of
2 data objects comprises all data objects present in the second plurality of data objects and the third
3 plurality of data objects.

1 7. The cloud data persistence system of claim 5, wherein the third server
2 stores a fourth plurality of data objects and is configured to request the data object from the first
3 server if the data object is not present in the fourth plurality of data objects, wherein:

4 the third server and the second server are data persistence cache servers;

5 and

6 the third server is communicatively coupled with a sixth and seventh
7 server.

1 8. The cloud data persistence system of claim 5, wherein the fifth server
2 stores a fifth plurality of data objects and is configured to request the data object from the second
3 server if the data object is not present in the fifth plurality of data objects, wherein:
4 the fifth server and the fourth servers are application servers; and
5 the fifth server is configured to receive requests from a second client for
6 the data object; and
7 the fifth server is geographically separated from the fourth server.

1 9. The cloud data persistence system of claim 5 wherein the request from the
2 first client for the data object comprises a universal resource indicator.

1 10. The cloud data persistence system of claim 5 wherein each of the first
2 plurality of data objects comprise a universal resource locator and a title linked to a media asset
3 of the plurality of media assets.

1 11. The cloud data persistence system of claim 5, wherein data objects in the
2 second plurality of data objects and the third plurality of data objects are at least partially
3 determined based on a frequency of requests for those data objects received from clients.

1 12. The cloud data persistence system of claim 5, wherein data objects in the
2 second plurality of data objects and the third plurality of data objects are at least partially
3 determined based on when requests for those data objects were last received from clients.

1 13. The cloud data persistence system of claim 5, wherein the first plurality of
2 data objects comprises all data objects in the second plurality of data objects, the third plurality
3 of data objects, the fourth plurality of data objects, and the fifth plurality of data objects.

1 14. A method for retrieving data objects using a cloud data persistence system
2 configured for distribution of data, the method comprising:
3 receiving, by an application center, a first request from a client for a data object,
4 wherein:

5 the application center is one of a plurality of application centers that
6 communicate with a plurality of clients;
7 the application center is located at a first location;
8 the data object comprises a link to a media asset; and
9 the media asset is stored at a second location separate from the first
10 location;
11 determining, by the application center, the data object is not stored among a first
12 plurality of data objects stored at the application center;
13 transmitting, by the application center, a second request for the data object to a
14 first cache server, wherein:
15 the first cache server is communicatively coupled with the plurality of
16 application centers; and
17 the first cache server is one of a plurality of cache servers;
18 receiving, by the cache server, the second request for the data object;
19 determining, by the cache server, the data object is not stored among a second
20 plurality of data objects stored at the cache server;
21 transmitting, by the cache server, a third request for the data object to an origin
22 server, wherein the origin server is communicatively coupled with the plurality of cache servers;
23 receiving, by the origin server, the third request for the data object;
24 locating, by the origin server, the data object among a third plurality of data
25 objects;
26 transmitting, by the origin server, the data object linked to the third request to the
27 cache server;
28 receiving, by the cache server, the data object linked to the third request;
29 transmitting, by the cache server, the data object linked to the second request;
30 receiving, by the application center, the data object linked to the second request;
31 and
32 transmitting, by the application center, the data object linked to the first request to
33 the client.

1 15. The method of claim 14 further comprising, after receiving the first data
2 object from the origin server, storing, by the cache server, the first data object.

1 16. The method of claim 14 further comprising, after receiving the first data
2 object from the cache server, storing, by the application center, the first data object.

1 17. The method of claim 14 wherein the cache server is located at a third
2 location.

1 18. The method of claim 14 wherein the first request for the data object
2 comprises a universal resource indicator.

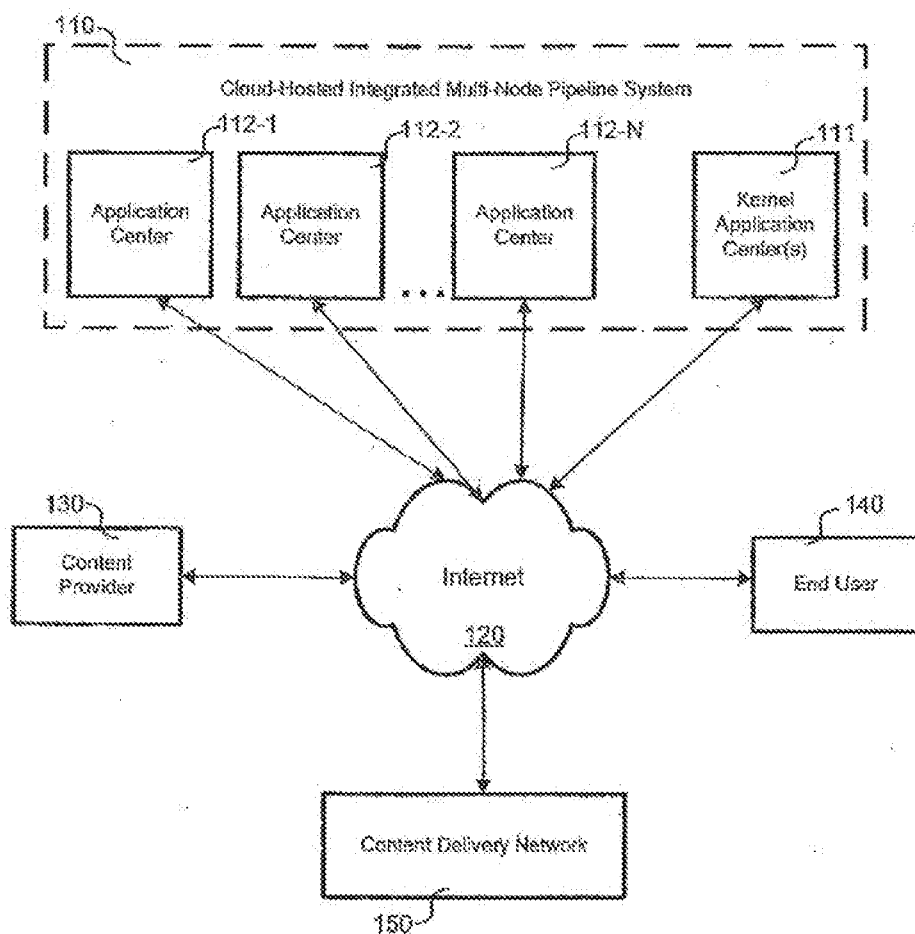
1 19. The method of claim 14, wherein data objects in the first plurality of data
2 objects and the second plurality of data objects are at least partially determined based on when
3 requests for those data objects were last received from clients.

1 20. The method of claim 14, wherein the third plurality of data objects
2 comprises all data objects in the first and the second pluralities of data objects.

1 21. A cloud data persistence system for distribution of data substantially as
2 herein described.

1 22. A method for retrieving data objects using a cloud data persistence system
2 configured for distribution of data substantially as herein described.

2010202782 01 Jul 2010



100

FIG. 1

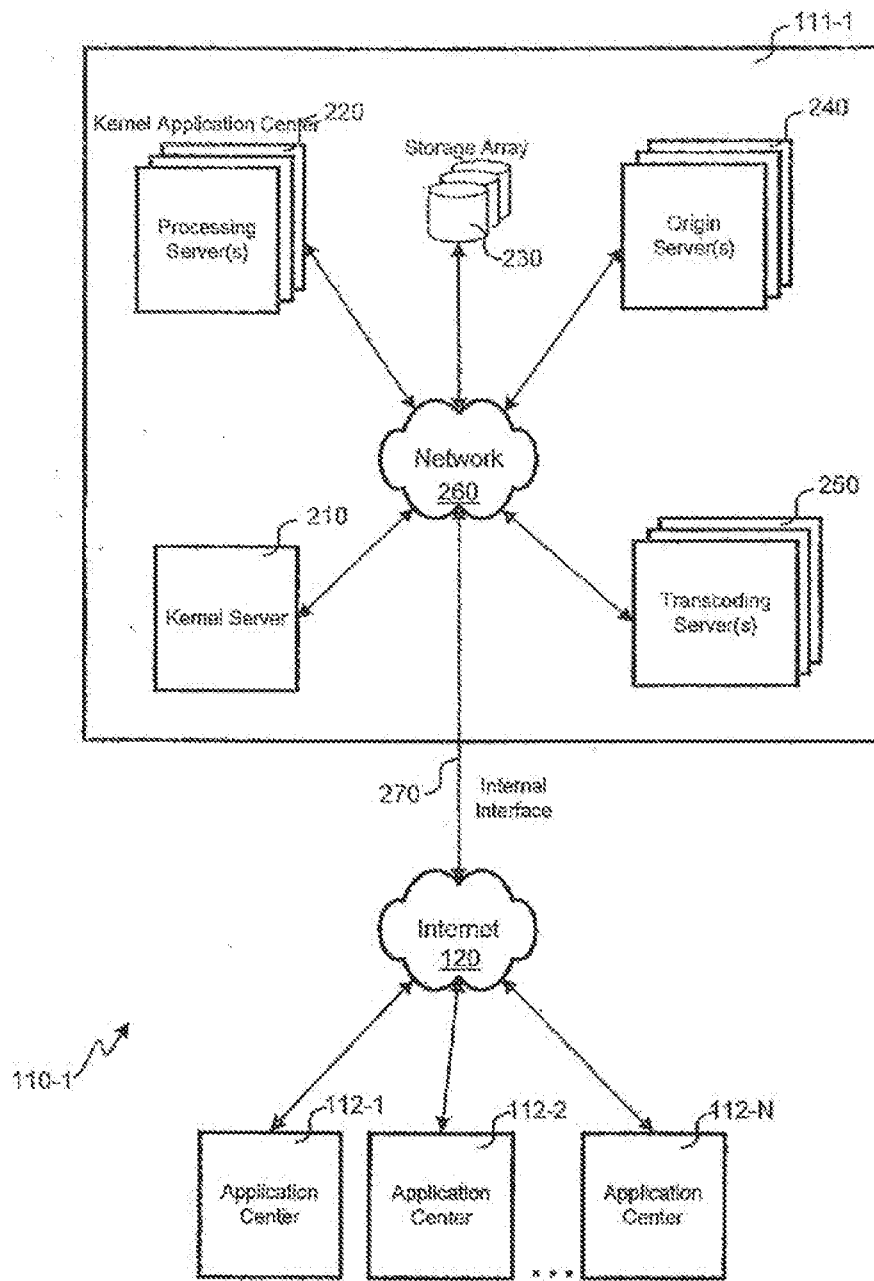


FIG. 2A

2010202782 01 Jul 2010

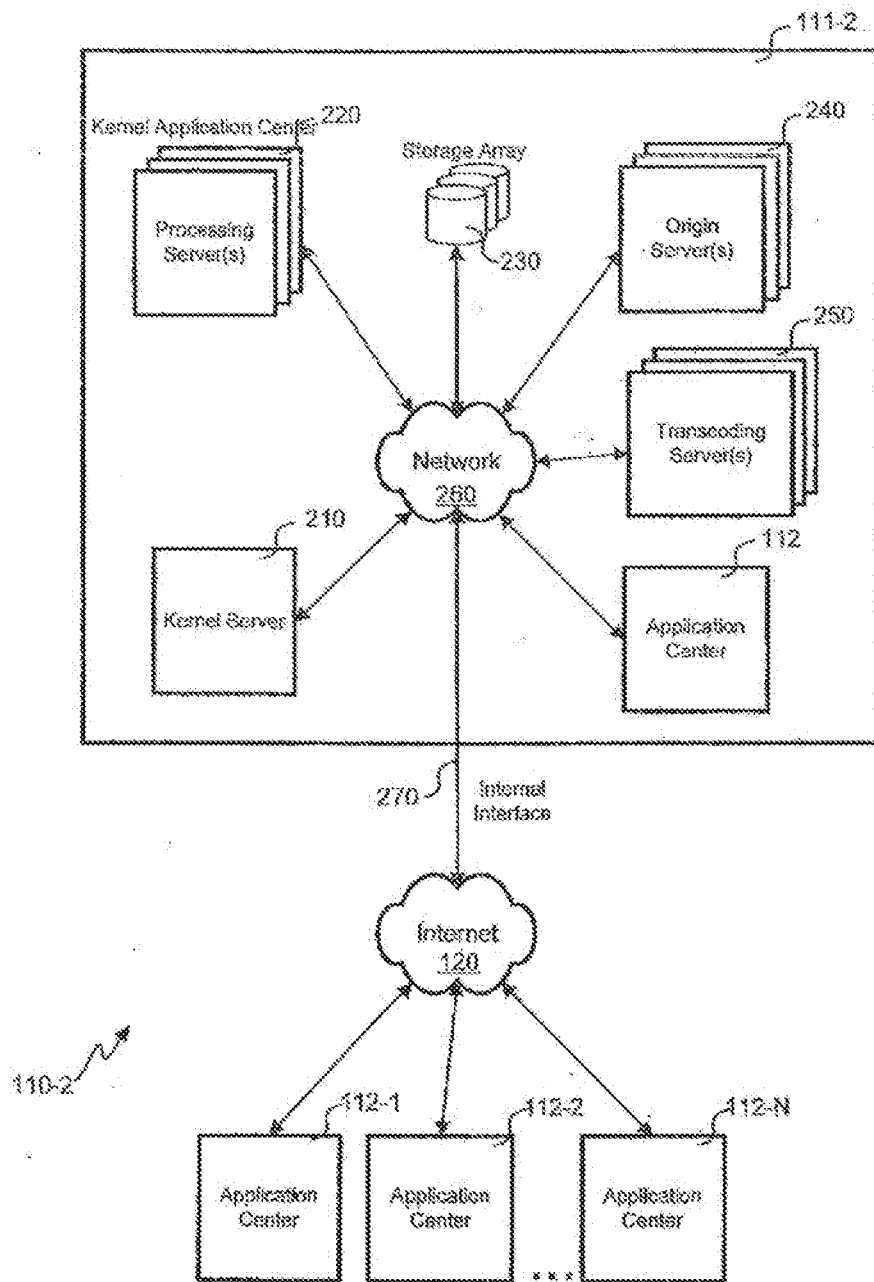


FIG. 2B

2010202782 01 Jul 2010

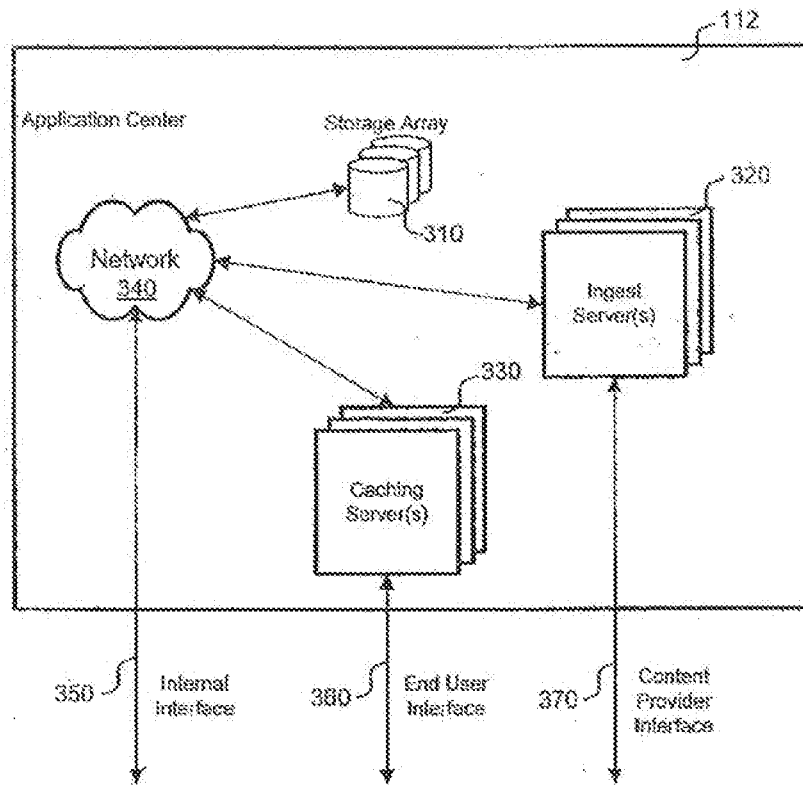


FIG. 3

2010202782 01 Jul 2010

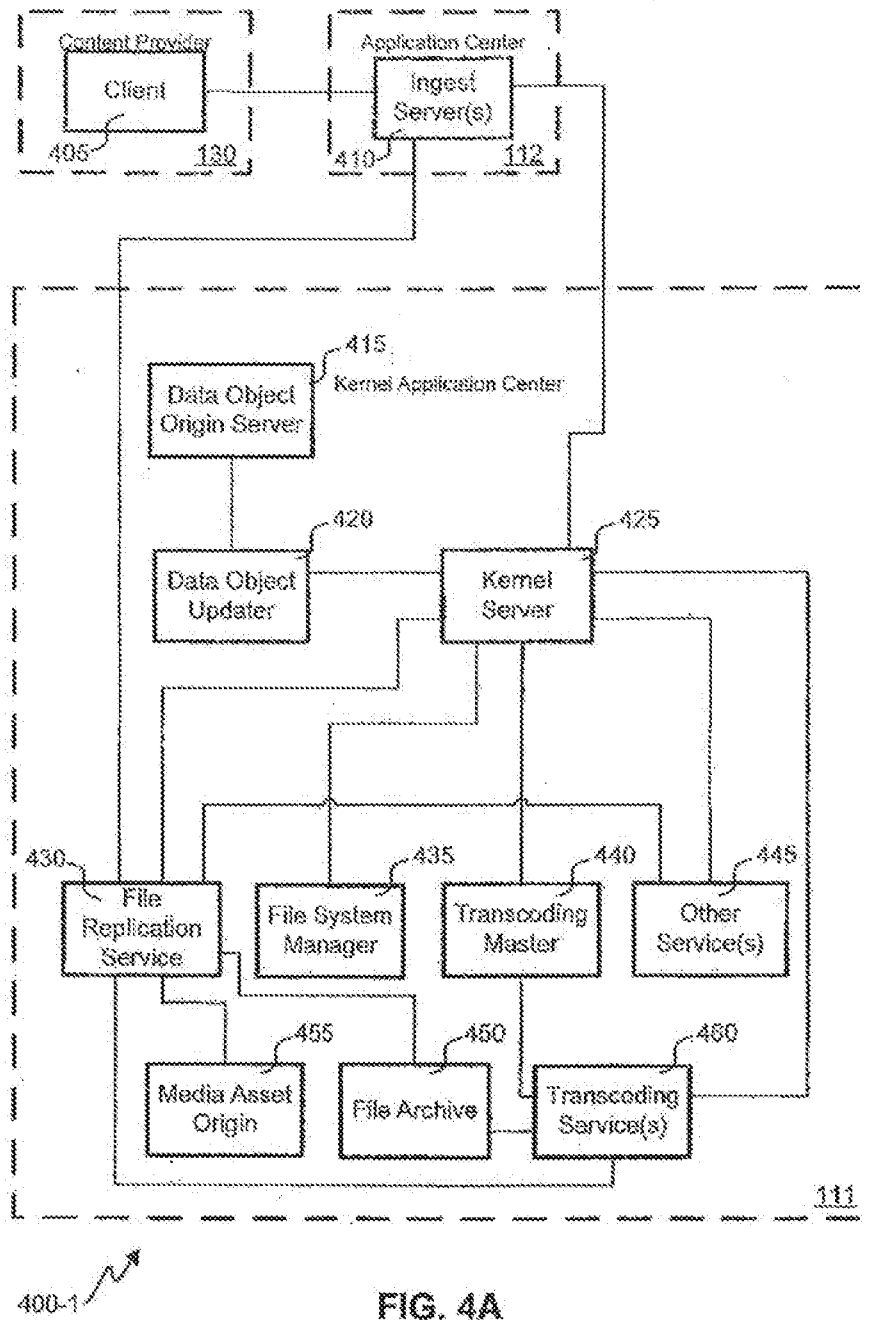


FIG. 4A

2010202782 01 Jul 2010

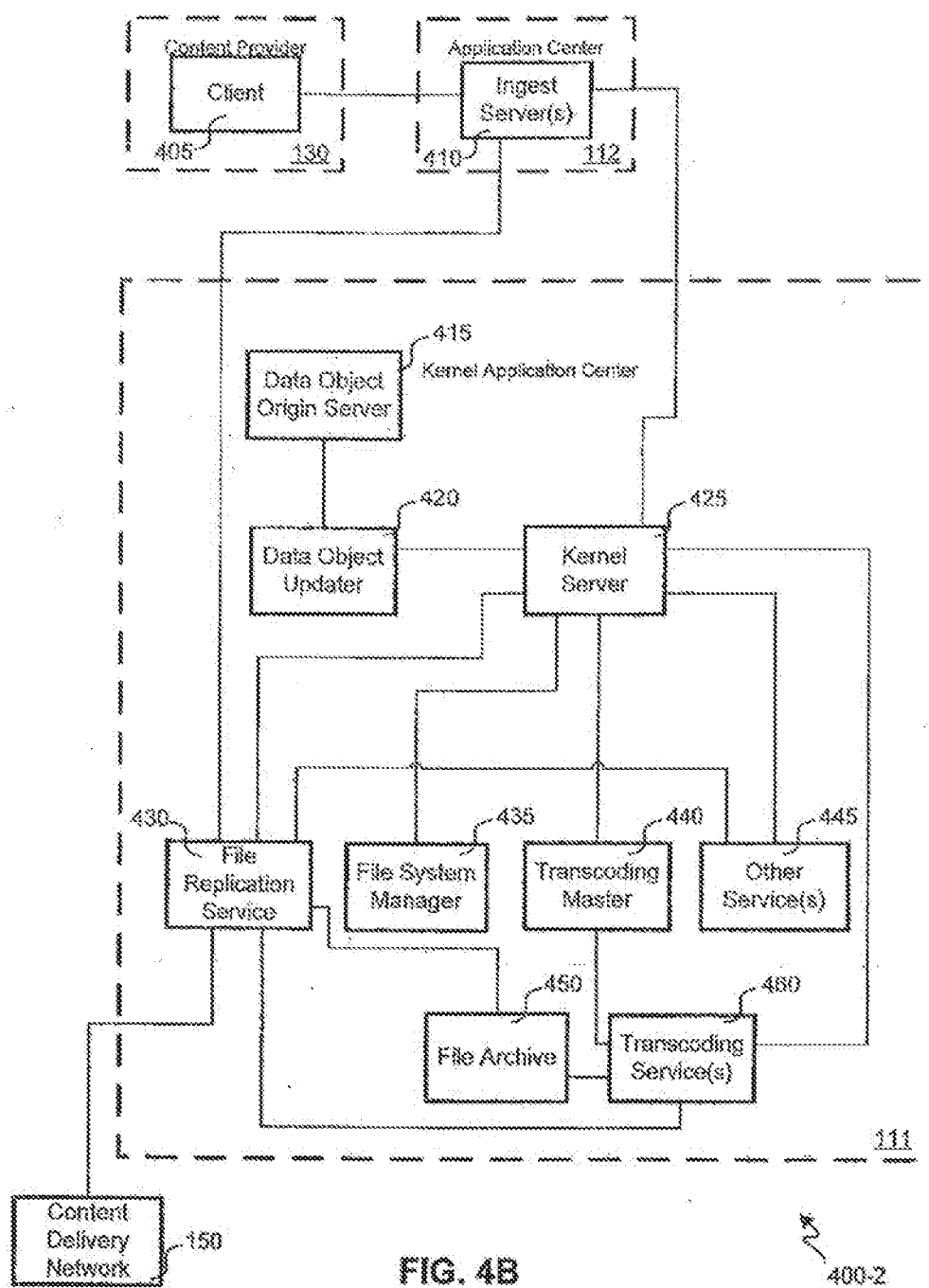


FIG. 4B

2010202782 01 Jul 2010

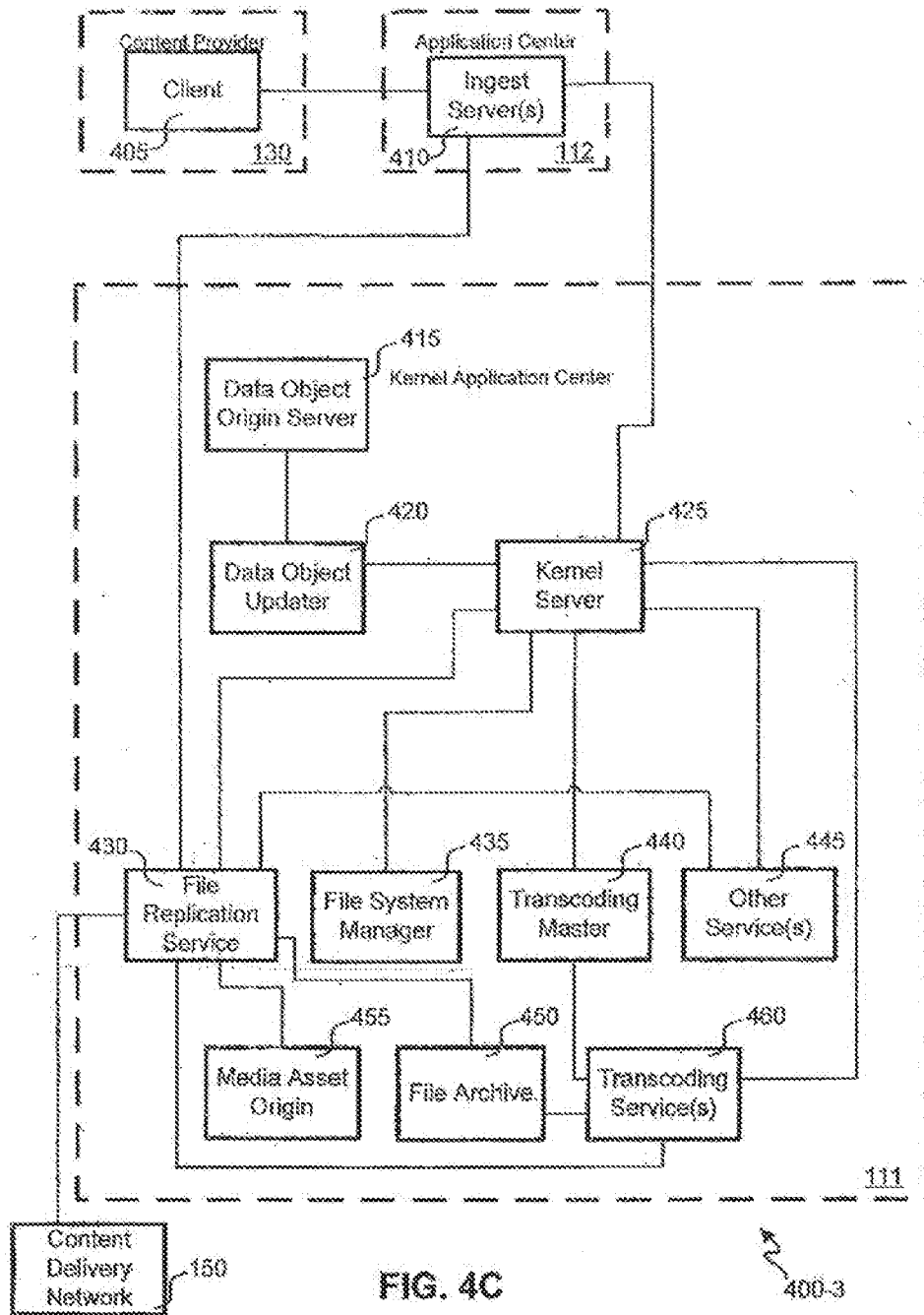


FIG. 4C

2010202782 01 Jul 2010

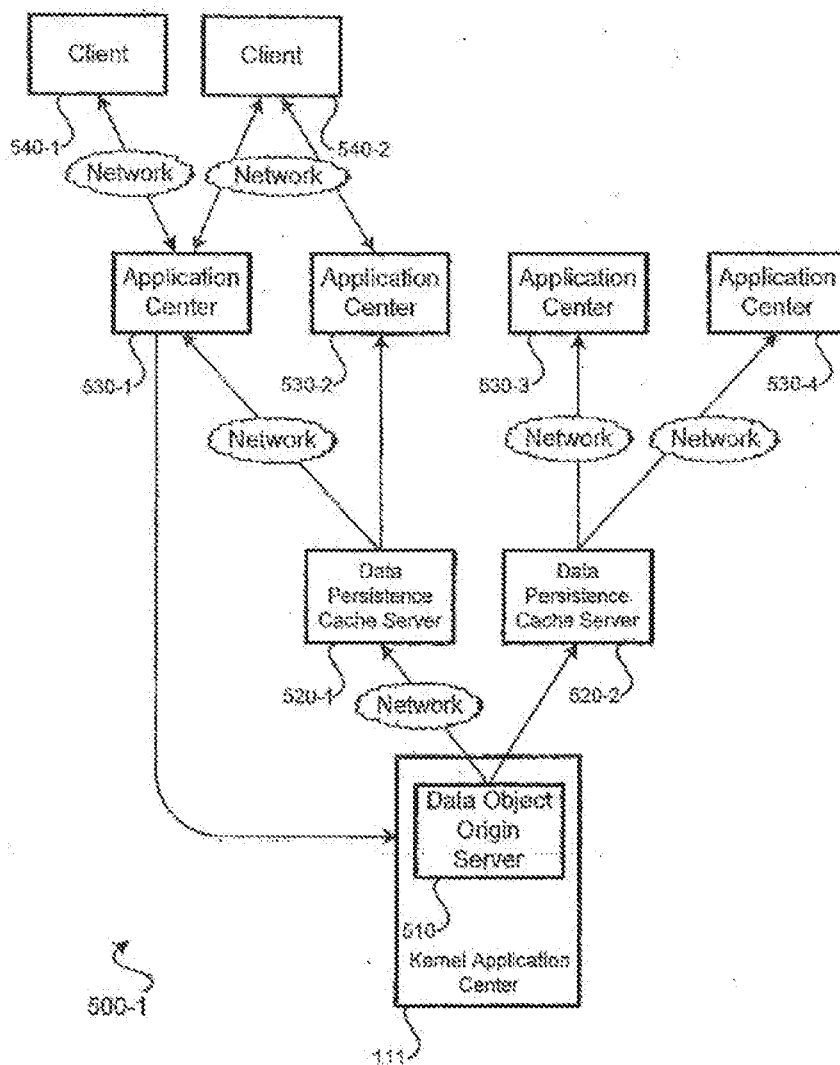


FIG. 5A

2010202782 01 Jul 2010

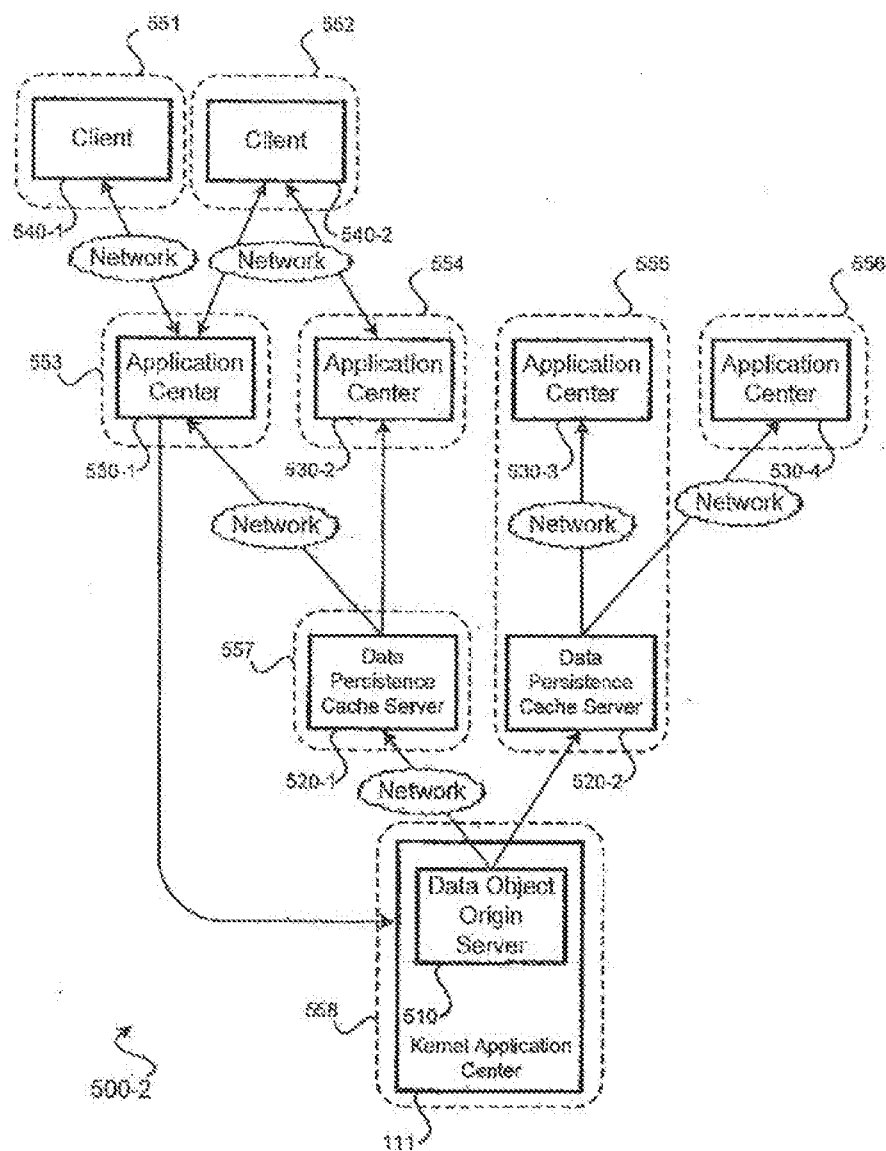


FIG. 5B

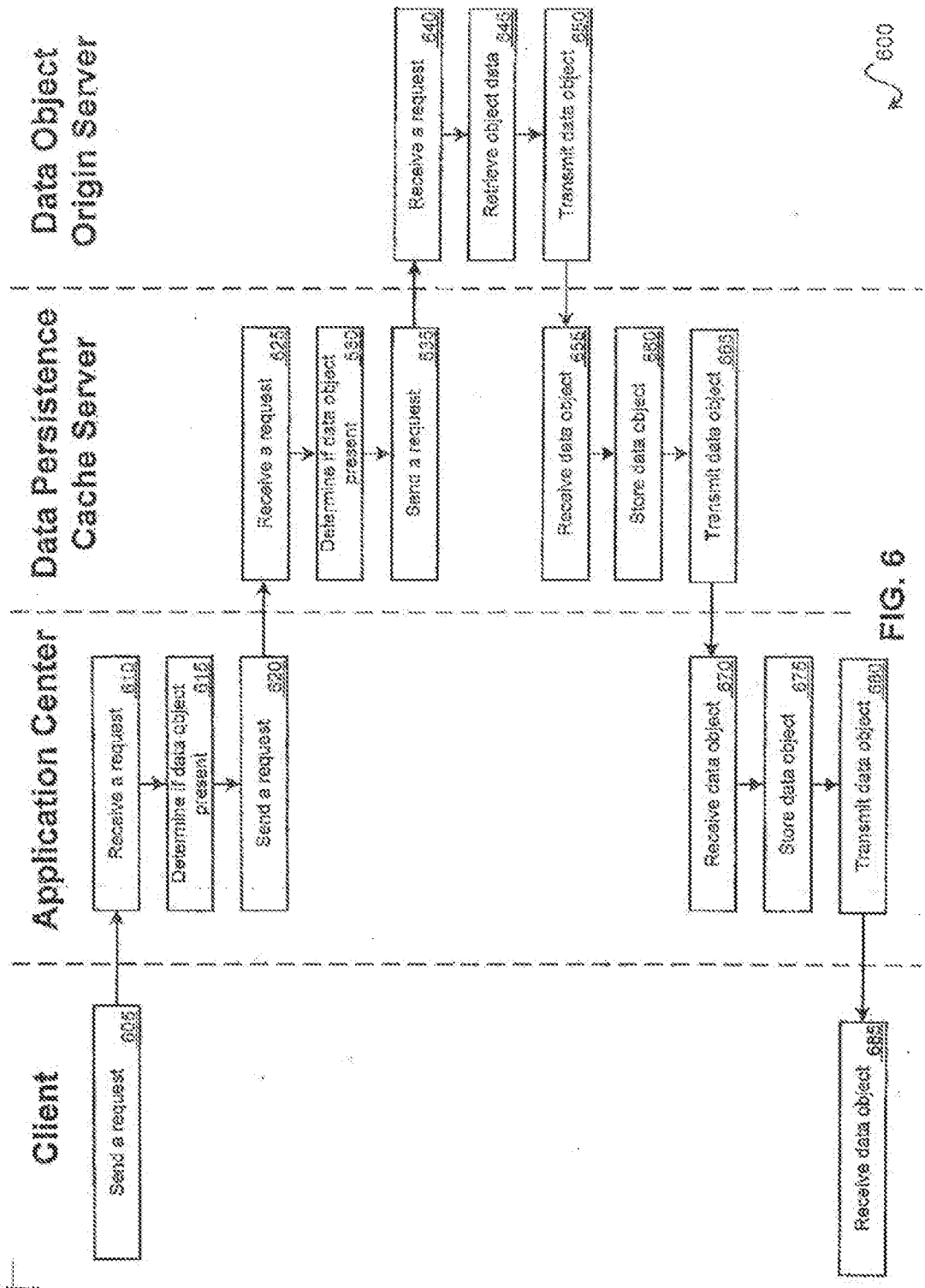


FIG. 6

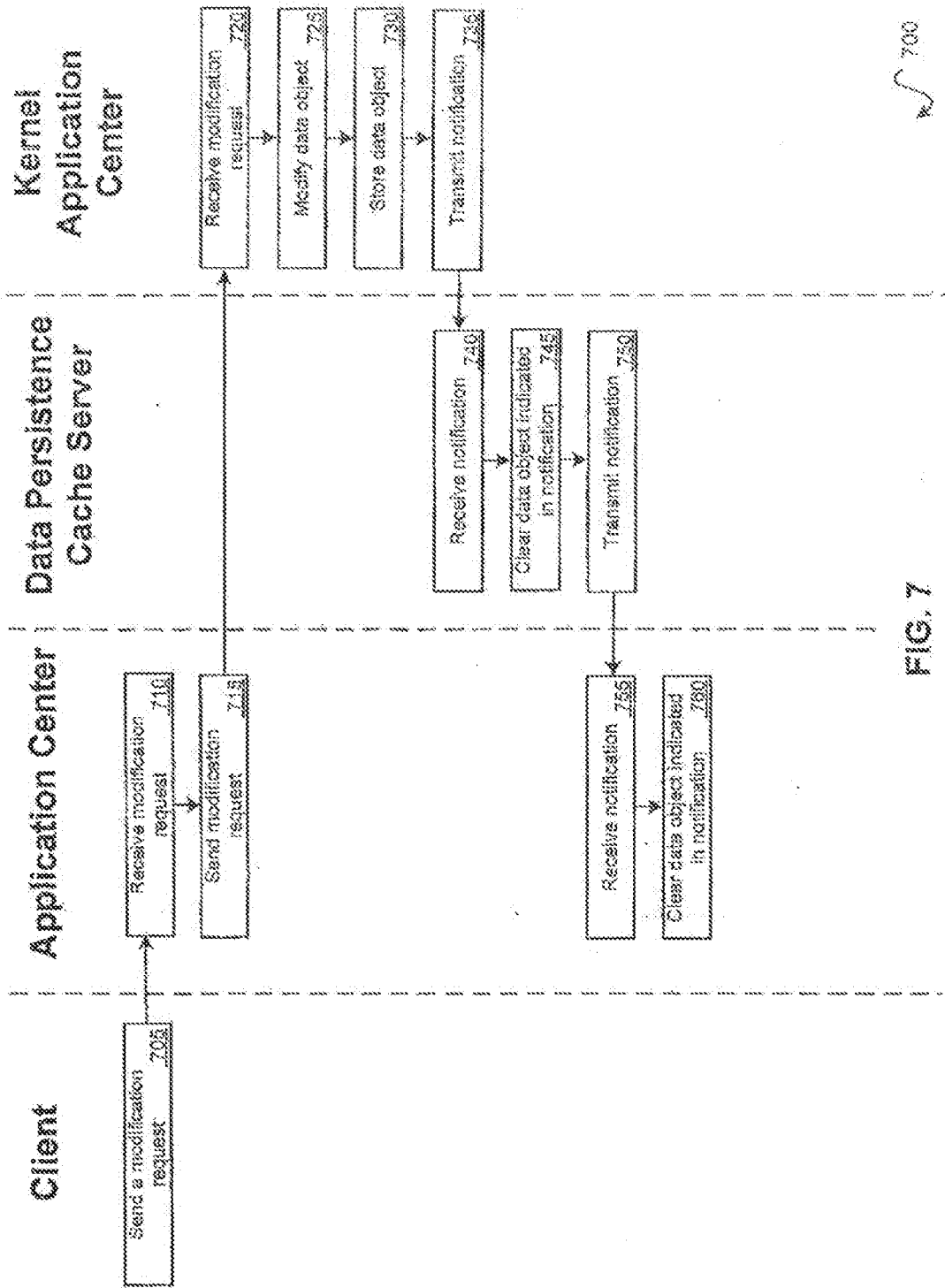


FIG. 7

700

2010202782 01 Jul 2010

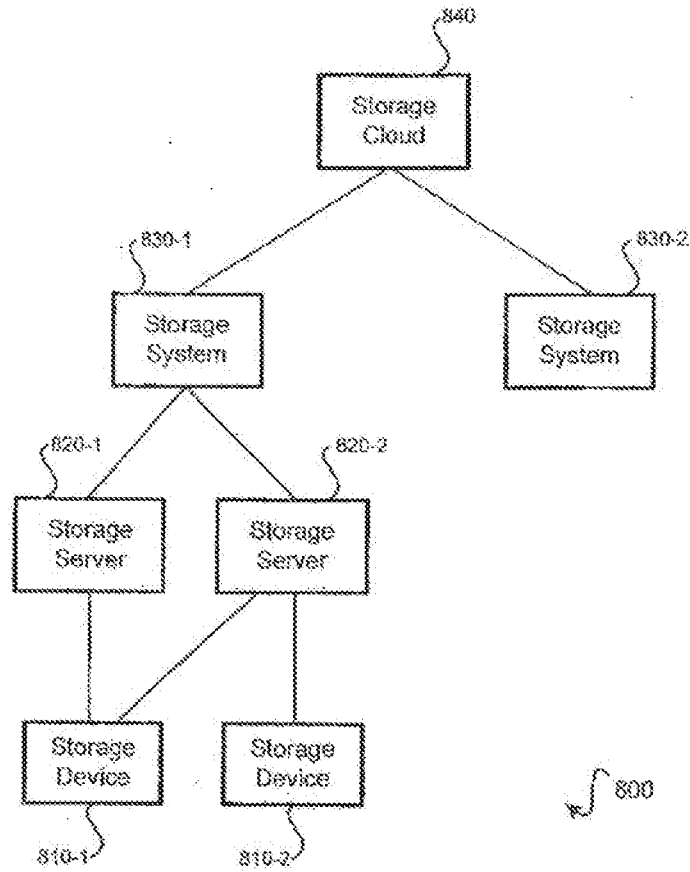


FIG. 8

2010202782 01 Jul 2010

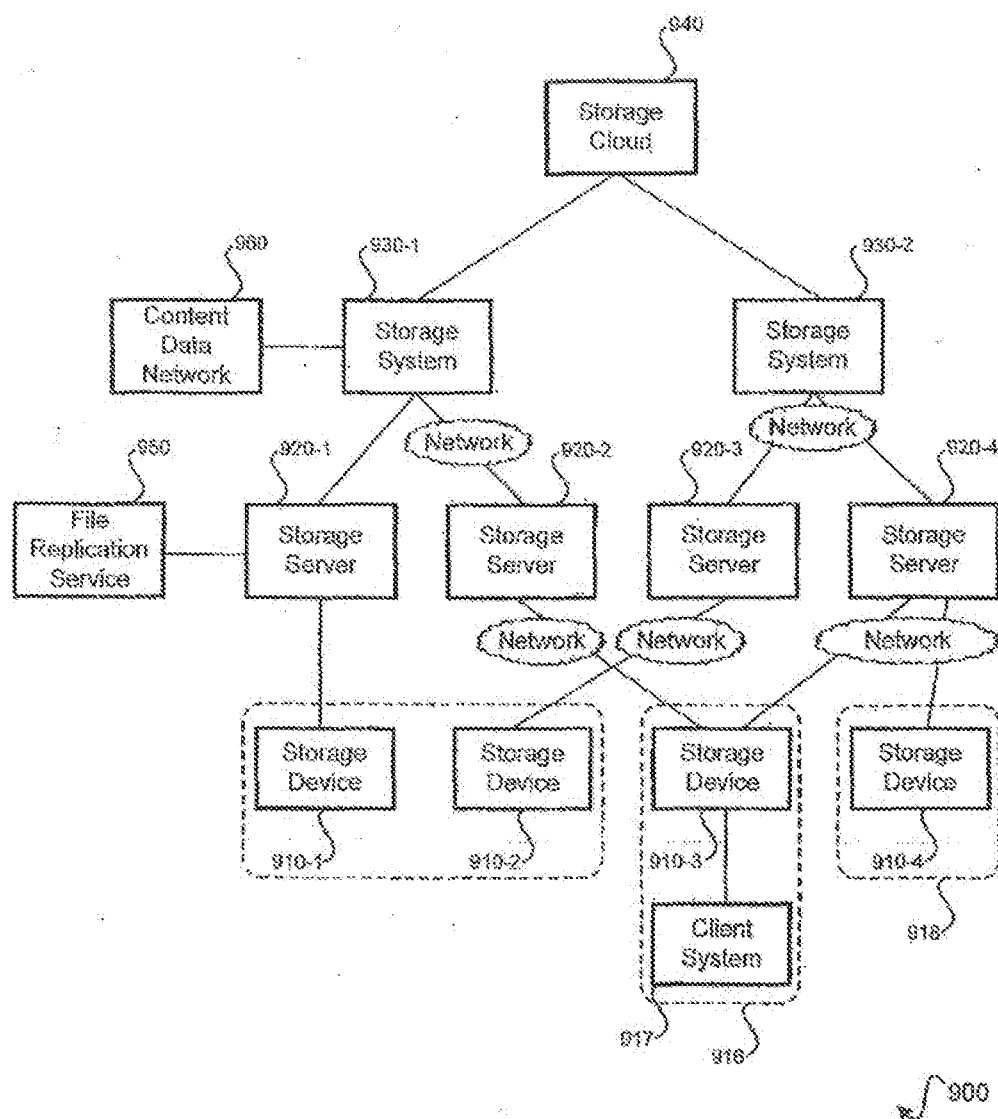


FIG. 9

2010202782 01 Jul 2010

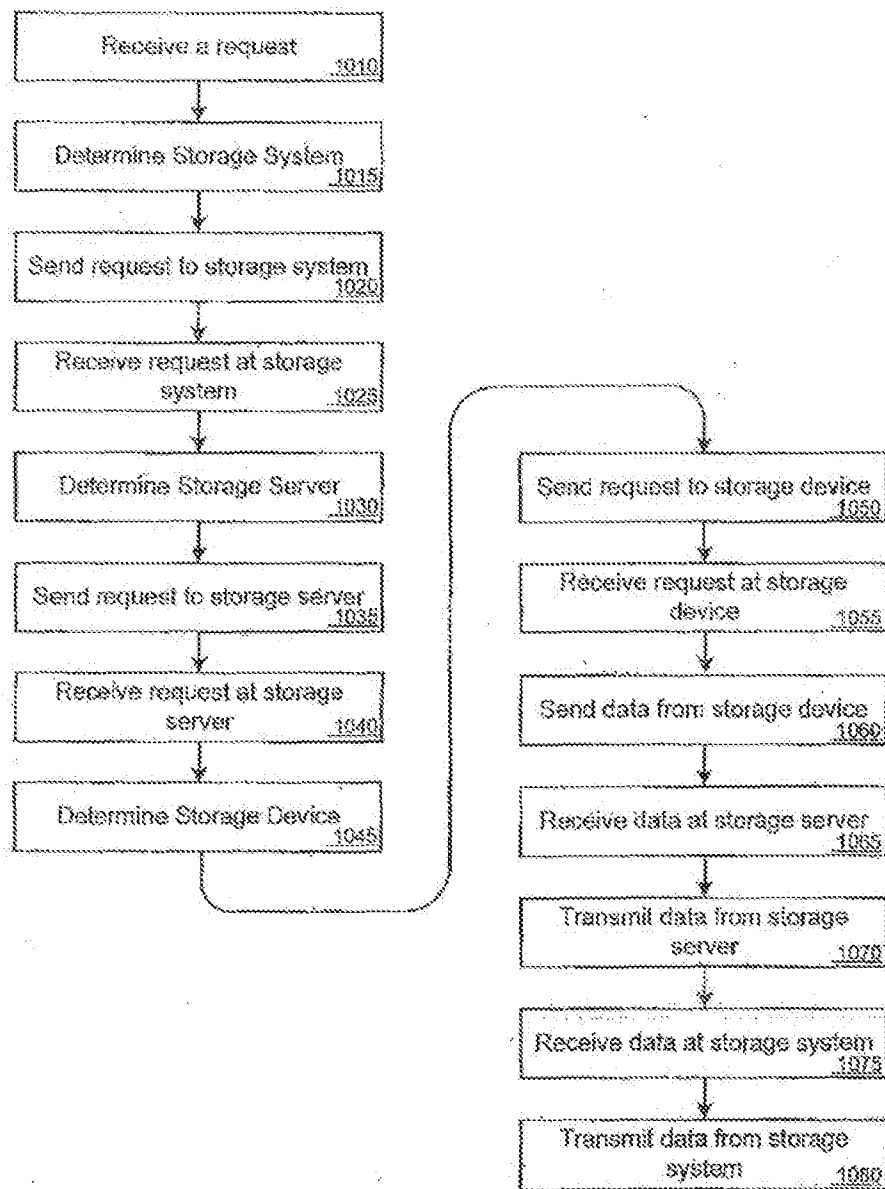


FIG. 10

1000



Australian Government

IP Australia

CERTIFICATE OF GRANT STANDARD PATENT

2013240558

The Commissioner of Patents has granted the above patent on 7 April 2016, and certifies that the following are the particulars of this patent appearing in the Register of Patents:

Name and Address of Patentee(s):

Brightcove Inc.

290 Congress Street 4th Floor Boston Massachusetts 02210 United States of America

Name of Actual Inventor(s):

McGowan, Albert John

Title of Invention:

Dynamic chunking for delivery instances

Term of Patent:

Twenty years from 25 January 2013

Priority Details

Number

13/624,029

13/430,081

Date

21 September 2012

26 March 2012

Filed with

US

US



Dated this 7th day of April 2016

Commissioner of Patents

PATENTS ACT 1990

The Australian Patents Register is the official record and should be referred to for the full details pertaining to this IP Right.



7 April 2016

Pizzeys Patent and Trade Mark Attorneys Pty Ltd
PO Box 291
WODEN ACT 2606
Australia

Notice of Grant

Patent Number: 2013240558

Patentee Name(s): Brightcove Inc.

Your Ref: 43769CAC/LOK

IP Australia is pleased to advise that a Patent has been granted in respect of this application. The patent was granted on 7 April 2016 and the Certificate of Grant is enclosed.

Please note the provisions of Section 67 of the Patents Act 1990, which states that the term of a standard patent is 20 years from the date of the patent.

To maintain this standard patent, Section 143(a) of the Patents Act 1990 requires you to pay the prescribed renewal fee annually, within the set time. If renewal fees are not paid the patent will cease.

Please Note:

1. Your first renewal fee payment, being for the 4th year anniversary is due on 25 January 2017.
2. The fee can be paid up to 6 months after the anniversary provided it is accompanied by the relevant late payment fee(s).

If you need any further information please contact 1300 651 010. Alternatively, please visit us at www.ipaustralia.gov.au.

Yours faithfully

Patent and Plant Breeder's Rights Administration



(12) STANDARD PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 2013240558 B2**

(54) Title
Dynamic chunking for delivery instances

(51) International Patent Classification(s)
H04L 29/06 (2006.01)

(21) Application No: **2013240558**

(22) Date of Filing: **2013.01.25**

(87) WIPO No: **WO13/147983**

(30) Priority Data

(31) Number	(32) Date	(33) Country
13/624,029	2012.09.21	US
13/430,081	2012.03.26	US

(43) Publication Date: **2013.10.03**

(44) Accepted Journal Date: **2015.12.24**

(71) Applicant(s)
Brightcove Inc.

(72) Inventor(s)
McGowan, Albert John

(74) Agent / Attorney
Pizzeys, PO Box 291, WODEN, ACT, 2606

(56) Related Art
US 7827298

(51) International Patent Classification:
H04L 29/06 (2006.01)

(31) International Application Number:

PCT/US2013/023181

(22) International Filing Date:

25 January 2013 (25.01.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

13/430,081 26 March 2012 (26.03.2012) US

13/624,029 21 September 2012 (21.09.2012) US

(71) Applicant: UNICORN MEDIA, INC. [US/US]; 34 West
5th Street, Tempe, Arizona 85281 (US).(72) Inventor: MCGOWAN, Albert, John; 4201 East Camel-
back Road, Unit 101, Phoenix, Arizona 85018 (US).(74) Agents: MCMILLAN, Scott et al.; Eighth Floor, Two
Embarcadero Center, San Francisco, CA 94111 (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, BN, 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, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU,
RW, 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, RW, SD, SL, SZ, TZ,
UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, 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, BI, CF, CG, CI, CM, GA, GN, GQ, GW,
ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a
patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the
earlier application (Rule 4.17(iii))

Published:

- with international search report (Art. 21(3))

(54) Title: DYNAMIC CHUNKING FOR DELIVERY INSTANCES

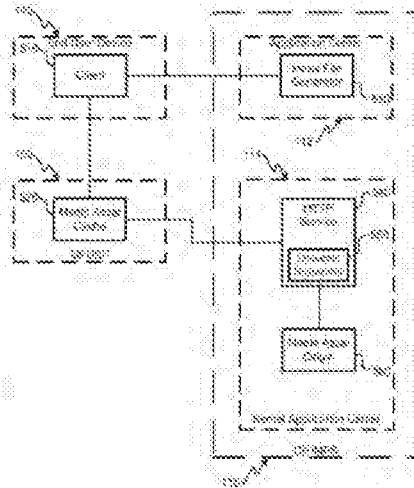


FIG. 5A

(57) Abstract: Systems and methods for dynamically chunking for delivery instances are provided that automatically implement chunking strategies based on one or more chunking considerations related to a request for a media file. These systems and methods may be part of a larger media servicing network that can be used to, among other things, process uploaded media content, provide it for streaming/downloading, and collect metric information regarding the streaming/downloading. The disclosed systems and methods provide for receiving a request having a Uniform Resource Locator (URL) and providing an index file to implement chunking strategies based on chunking considerations associated with the request.

DYNAMIC CHUNKING FOR DELIVERY INSTANCES

BACKGROUND OF THE INVENTION

[0001] The delivery of media over networks such as the Internet can be accomplished in many ways, including progressive downloading or streaming. Streaming a media file typically involves downloading "chunks," or small segments, of the media file. Information including where chunks may be accessed can be stored in an index file (also known as a "manifest" file). This index file can be delivered to a client, such as a media player application, for use in streaming. Additional information may also be provided, which can alter the appearance of the client.

BRIEF SUMMARY OF THE INVENTION

[0002] Systems and methods disclosed provide a technical solution improving quality of service of video streaming based on a determined state of a client media player application. The disclosed systems and methods provide for dynamically executing syndication services are provided that automatically implement business rules for syndication based on contextual data associated with a media file, a request for a media file, or both. These systems and methods may be part of a larger media servicing network that can be used to, among other things, process uploaded media content, provide it for streaming/downloading, and collect and distribute metric information regarding the streaming/downloading. The disclosed systems and methods provide for receiving a request having a Uniform Resource Locator (URL) or other content indicator and providing an index file in accordance with business rules based on contextual data associated with the request, the content, or both. Embodiments further enable media content owners to distribute a single URL or other content indicator corresponding to a particular media file among many media providers, enabling a single media delivery and analytics service to provide comprehensive metric information regarding content syndication and usage for all requests for the content across all media providers.

[0003] According to one embodiment, a server for customizing index file generation for streaming a media file via a data network can include a network interface for communicating with the data network, and an index file generator coupled to the network

interface. The index file generator is configured to obtain a request having a universal source locator (URL), where the URL includes information indicative of a requested media file, determine a state of a client media player application related to the request, and determine a chunking strategy based on the state of the client media player application. The media file generator is further configured to generate an index file having information for streaming the requested media file via the data network. Generating the index file is based, at least in part, on the chunking strategy. The index file generator is further configured to send the index file via the network interface.

[0004] According to another embodiment, a method for customizing index file generation for streaming a media file via a data network includes obtaining a request having a universal source locator (URL). The URL includes information indicative of a requested media file. The method further includes determining a state of a client media player application related to the request, determining a chunking strategy based on the state of the client media player application, and generating an index file having information for streaming the requested media file via the data network. Generating the index file is based, at least in part, on the chunking strategy. The method also includes sending the index file via the network interface.

[0005] According to yet another embodiment, a system for customizing index file generation for streaming a media file via a data network includes communication means for receiving a request having a universal source locator (URL) (where the URL includes information indicative of a requested media file) and sending an index file via the data network. The system further includes processing means for determining a state of a client media player application related to the request, determining a chunking strategy based on the state of the client media player application, and generating the index file having information for streaming the requested media file via the data network. Generating the index file is based, at least in part, on the chunking strategy.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The present disclosure is described in conjunction with the appended figures:

[0007] FIG. 1 illustrates a block diagram of a media servicing system.

[0008] FIG. 2A illustrates a block diagram of an embodiment of a kernel application center connected with application centers.

[0009] FIG. 2B illustrates a block diagram of an alternative embodiment of a kernel application center.

[0010] FIG. 3 illustrates a block diagram of an embodiment of an application center.

[0011] FIG. 4 illustrates a block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion.

[0012] FIG. 5A illustrates a simplified block diagram of an embodiment of a system configured to provide dynamic indexing and chunking for media streaming.

[0013] FIG. 5B illustrates a simplified block diagram of another embodiment of a system configured to provide dynamic indexing and chunking for media streaming.

[0014] FIG. 5C illustrates a simplified block diagram of another embodiment of a system configured to provide dynamic indexing and chunking for media streaming, utilizing a redirector.

[0015] FIG. 6 illustrates a simplified flowchart of an embodiment of a method for implementing a dynamic index for media streaming.

[0016] FIG. 7 illustrates a simplified flowchart of an embodiment of a method for dynamically chunking a media file for streaming.

[0017] FIG. 8 illustrates a simplified swim lane flowchart describing the interaction of components in a system configured to provide dynamic indexing and chunking for media streaming, according to one embodiment.

[0018] FIG. 9 illustrates a simplified flow chart providing a basic method for adapting chunking techniques for each delivery instance.

[0019] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION OF THE INVENTION

[0020] The ensuing description provides preferred exemplary embodiment(s) only, and is not intended to limit the scope, applicability or configuration of the disclosure. Rather, the ensuing description of the preferred exemplary embodiment(s) will provide those skilled in the art with an enabling description for implementing a preferred exemplary embodiment. It is understood that various changes may be made in the function and arrangement of elements without departing from the spirit and scope as set forth in the appended claims.

[0021] The increased availability of media content over data communications networks such as the Internet has mirrored the increased bandwidth for these networks. Because media has recently taken a more prominent role in data communications, the distribution of media and the data associated with such distribution has become increasingly important, particularly to media providers. Media streaming has become a widely-used method of media distribution, but the preprocessing associated with streaming can be burdensome. Certain protocols, including forms of Hypertext Transfer Protocol (HTTP) streaming, require chunking and storing media assets, and generating a corresponding index files. These requirements can deprive a media provider of the ability to dynamically insert additional media such as advertisements into a media stream, and can consume a large amount of storage space to store chunks of media for a media asset, including chunks for any alternative sub-streams (e.g., streams with alternative bitrates, captions, alternative languages, etc.). Certain systems and methods can be utilized, however, to introduce the desired functionality back into the system.

[0022] A traditional approach to preprocessing media for streaming involves chunking and storing media assets, then creating corresponding index files to indicate where chunks may be located to download for streaming. Streaming protocols often provide for frequently updating an index file for instances where the corresponding media is frequently updated, such as during live streaming. Thus, an index file does not need to contain all chunks for a requested media asset. In addition, because media files are frequently stored in a format that requires little additional processing to chunk, the chunks can be created in real time, during the streaming of a media file. The systems and methods disclosed herein take advantage of these features to enable dynamic index file creation and dynamic media file chunking.

[0023] For instance, rather than preprocess media assets for streaming by chunking and indexing all files with relevant sub-streams prior to streaming the media, a server can dynamically create and update an index file during streaming. The dynamically-created index file can contain information regarding a next chunk of media in the various available sub-streams. The next chunk of media may not be cached at a location specified in the index file, in which case a chunk may be dynamically created by pulling all or part of the media file of interest from a media file origin, chunking it, and making it available for download. The chunk also may be cached, thereby eliminating the need to create the chunk again if it is requested at some later time.

[0024] Because the chunks are created during streaming, a media provider and/or media distributor can have more information and control during the streaming process. Rather than generate single index file for a given media asset, an instance of the index file generator may be created at the beginning of the media streaming to provide individualized media content to a particular end user and unique information regarding the streaming session to a media provider. The file index generator can vary the length of each chunk by, for example, indicating starting and ending points in the index file. Thus, the file index generator may determine a uniform chunk length for a media asset, varying the length of the chunks for different media assets, or the file index generator may adjust the length of the chunks within a single media asset. The index file generator can further insert additional media, such as an advertisement, at any time during the streaming by specifying the location of the additional media in the index file. The determination to insert advertisements can be based on any information, including data collected during the streaming session.

[0025] As the index file generator receives requests for and generates index files, it can further gather data regarding the streaming session for reporting to a media provider. Media providers often rely on beaconing data collected from media player applications to determine when an end user stops, plays, pauses, skips, etc., the streaming media content. Such information can be vital in determining the value of the media.

[0026] Because not all media player applications provide this beaconing data, the data gathered by the index file generator can serve as a substitute for or complement to the beaconing data. For example, if a request is made for a chunk that does not immediately follow a previously-requested chunk, a skip was made. If the amount of time elapsed between a previous request and a subsequent request exceeds the time for playback of the

previously-requested chunk, a pause was made. If a request is not received within a certain time since a prior request, it can be determined that a stop was made.

[0027] As illustrated above, the state of a client may be determined from a variety of factors. This can include when the request for the index file is received, when the index file is provided, a length of time to play back the segment of media for streaming, and/or the starting and/or ending point of the segment of media for streaming. The determined state of a client may also be based on whether the request for the index file has been received within a certain amount of time since receipt of a previous request for an index file, whether the segment of media for streaming includes media other than the media file, and more. The state of a client and/or the data from which it was determined, may be used to create reporting data to serve as a substitute or complement to beaconing data from a client media player application. Because the index file generator can determine the length of the chunks, it therefore can determine the frequency of subsequent index file requests and the resolution of the reporting data based on the requests. The index file generator may log the reporting data and/or transmit the reporting data over a network during streaming.

[0028] The determined state of a client may be used by the index file generator and/or other services for various purposes. For example, it may be used in behavioral advertisement targeting and enforcement of session advertisement behavior, adjusting advertisement content and playback based on the behavior of a user as determined by the state of a client. The state of a client further may be used to support resume features on a per client basis, allowing a user to continue playback of a media asset from a point at which the user had previously stopped playback. The state of a client also may be used to support individual encryption keys in an encryption scheme and allow the index file generator to return secure URLs (e.g., time expiring or Internet Protocol (IP) allowed) for chunks to support functions such as payment services.

[0029] Additionally or alternatively, the tasks of generating the index file and providing a location a requested chunk can be split up, thereby enabling the system to determine which chunks are actually requested. For example, a system may be configured to dynamically create an index file having links to one or more redirectors on the system. These redirectors can be configured to issue the location of the chunk, which can be created dynamically. The redirectors can further determine which chunk is actually requested, thereby enabling, among other things, calculation of Quality of Service (QoS) metrics, an increase the

accuracy of reporting data, a decrease the frequency of index file generation if efficient to do so, and the ability to more easily handle keys of an encryption scheme.

[0030] While the above embodiments may be implemented in a variety of different systems, some particular embodiments may be implemented as part of a media service system. FIG. 1 is a block diagram illustrating a media servicing system 100, according to some embodiments of the present invention. The system may deliver media content to the end user device 140 through a network such as the Internet 120. The end user device 140 can be one of any number of devices configured to receive media over the Internet 120, such as a mobile phone, tablet computer, personal computer, portable media device, etc. A media asset provided by a media provider 130 can be processed and indexed by cloud-hosted integrated multi-node pipelining system (CHIMPS) 110, and further stored on media file delivery service provider (MFDSP) 150. Additionally or alternatively, the CHIMPS 110 may also be adapted to store the media asset.

[0031] The media servicing system further enables a media provider 130 or other entity to gather information regarding user behavior during media playback. For example, a media provider 130 can be provided with data indicating that end users tend to stop watching a video at a certain point in playback, or that users tended to follow links associated with certain advertisements displayed during playback. With this data, a media provider 130 can adjust factors such as media content, advertisement placement and content, etc., to increase revenue associated with the media content and provide the end user device 140 with a more desirable playback experience.

[0032] End user device 140 can request a media asset to stream with a client program executed by the end user device 140. The client program can be, for example, a media player, browser, or other application adapted to request and/or play media assets. In response to a request for a media asset, the CHIMPS 110 can utilize any number of application centers 112 and/or kernel application center(s) 111 to provide the client program with a data object concerning the requested media asset. The data object can include information about the media asset, including where the media asset can be located, such as within the MFDSP 150 or within the CHIMPS 150 itself. Location information may be provided by Universal Resource Indicator (URI), a Universal Resource Locator (URL) or other indicator. During playback of the media asset, the CHIMPS 150 can collect data regarding the playback through beaconing provided by a client program executed by the end

user device 140 and/or indexing service from within the CHIMPS and/or MFDSP. The CHIMPS 150 can subsequently provide the data and/or any analytics information derived from the data to the media provider 130.

[0033] FIG. 2A is a block diagram illustrating an embodiment of a kernel application 111-1 center connected with application centers from within the CHIMPS 110-1. The kernel application center 111-1 and application centers 112 can be geographically distant and can be connected via the Internet 120, wide area network (WAN), and/or other data communication network. Because application centers can be geographically separated, DNS services (not shown) can be used to allow an end user device 140 to connect to the nearest available application center 112. The kernel application center 111-1 can connect with application centers 112 within the CHIMPS 110-1 through an internal interface 270, thereby enabling the application centers 112 access to the various components within the kernel application center 111-1.

[0034] Components within the kernel application center 111-1 can communicate through network 260 such as a local area network (LAN) and can include one or more origin servers 240 and a storage array 230 with which data objects and/or media assets may be stored and distributed. The storage array 230 may also be utilized by services running on processing server(s) 220 and/or transcoding server(s) 250 that may require temporary or long-term storage. Kernel server 210 can utilize processing server(s) 220, transcoding server(s) 250 to provide various functional capabilities to the CHIMPS 110.

[0035] For example, as described in more detail below, the CHIMPS 110-1 can provide transcoding service for media assets provided by a media provider 130 for syndication. Such a service can allow a media provider 130 to upload a media asset to an application center 112, after which the application center 112 would notify the kernel server 210 that the media asset has been uploaded. The kernel server can then notify services running on the processing server(s) 220 of the upload. These services can utilize transcoding server(s) to transcode the media asset, which can then be moved to a MFDSP and/or stored locally by storage array 230 and origin server(s) 240. Services running on the processing server(s) 220 can also update the associated data object stored by the storage array 230 and origin server(s) 240.

[0036] FIG. 2B is a block diagram illustrating an alternative embodiment of a kernel application center 111-2. In addition to the components of the embodiment of FIG. 2A, this

embodiment incorporates an application center 112 within the kernel application center 111-2. The application center 112 incorporated within kernel application center 111-2 may be located at or near the other components of the kernel application center 111-2, and can be communicatively connected to the other components via network 260. The incorporated application center 112 can therefore have faster access to kernel application center functionality because it does not need to communicate over long distances. In consideration of this advantage, it will be understood that the CHIMPS 110 can include multiple kernel centers with one or more application centers incorporated therein. Additionally or alternatively, components of the kernel application center may be incorporated into one or more application centers 112 in the CHIMPS 110 to provide quicker access to certain functionality.

[0037] FIG. 3 is a block diagram illustrating an embodiment of an application center 112. The application center 112 can include caching server(s) 330 and a storage array 310 for storing and distributing data objects of media assets requested by end user devices through end user interface 360. Caching server(s) 330 and storage array 310 can also be used to collect, process, and/or store metrics information from beaconing data, media chunk requests, and/or other data sources, including data collected through end user interface 360. The application center can further include ingest server(s) 320 for ingesting uploaded media assets from a media provider 130 through a media provider interface 370. The media assets may be stored on the storage array 310. As with the kernel application center 111, the components of the application center 112 can be communicatively linked through a network 340, such as a LAN. The application center can further include an internal interface 350, providing a communication link from the application center to the rest of the CHIMPS. It is through internal interface 350, for example, that media assets stored on storage array 310 can be made available to a kernel application center 111 for services such as transcoding.

[0038] FIG. 4 is a block diagram 400 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Although FIG. 4 further indicates the physical systems in which may execute or store these processes and objects, it will be understood that the processes and objects disclosed may be executed or stored on more than one system, including systems not disclosed in FIG. 4. In other words, the processes and objects shown in FIG. 4 allow for a variety of implementations through one or more of hardware, software, firmware, microcode, etc.

[0039] Media can be ingested into the CHIMPS 110 when a media provider 130 uploads a media asset to ingestion server(s) 410 in an application center 112 by utilizing a client 405. The client 405 can be a stand-alone application or browser based, for example, and can communicate with ingest server(s) 410 through an application programming interface (API) configured for the ingestion of media assets.

[0040] Ingest server(s) 410 can communicate with devices in the kernel application center 111 executing programs such as kernel server 425 and file replication service 430. The kernel server 425 can be configured organize the workflow among services such as transcoding 440 file system manager 435, and other services 445 (e.g., analytics, dynamic API, etc.) Upon a particular event, for example, the kernel server can be configured to notify the relevant services of the event, causing the services to process tasks associated with the event.

[0041] The file replication service 430, under direction of the kernel server 425, can coordinate the movement of the media assets between services. For example, retrieving the uploaded media asset from the ingest server(s) 410 and storing it on the file archive 450, or retrieving transcoded media assets from transcoding server(s) 460 and storing them in the media asset origin.

[0042] The data object updater 420 keeps the data object origin 415 up to date in response to any changes in the system. When, for example, a file is uploaded, transcoded, and stored in media asset origin 455, the location and other metadata concerning the transcoded media assets need to be created or updated in the data object origin 415 to ensure an end user device that accesses the object in the data object origin 415 has the correct information regarding the related media asset. Because the data object updater 420 receives updates from the kernel server 425 (which is notified when a transcoded media asset is stored in the media asset origin 455, the system ensures the data objects in the data object origin are constantly up to date.

[0043] The upload of a media asset to the ingest server(s) 410, as described above, can provide an example of how the kernel server 425 may coordinate workflow. For instance, in response to the upload, the ingest server(s) 410 can notify the kernel server 425 that a media asset has been uploaded. The kernel server 425 informs the file replication service 430 of the uploaded media asset, and the file replication service 430 moves the uploaded media asset into the file archive 450 and notifies the kernel server 425 of the move. In

response, the kernel server 425 notifies the file replication service 430, the file system manager 435, and the transcoding master 440 of the move. The file replication service 430 then will know it can delete the uploaded media asset from the ingest server(s) 410, the file system manager 435 will update the file system accordingly, and the transcoding master 440 will notify transcoding service(s) 460 of different transcoding tasks to be performed. The transcoding service(s) 460 can then retrieve the uploaded media asset from the file archive 450 to create transcoded media assets. The transcoding service(s) 460 notify the kernel server 425 once transcoding is complete, and the kernel server 425 relays this information to the file replication service 430. The file replication service 425 then takes the transcoded media assets from the transcoding services 460 and moves them to the media asset origin 455. Once the file replication service 430 notifies the kernel server 425 of the move, the kernel server 425, in turn, notifies the file replication service 430 and the data object updater 420. The data object updater 420 which updates the data object origin 415 accordingly, and the file replication service 430 deletes the transcoded media assets from the transcoding services 460.

[0044] The modular nature of the system enables all tasks associated with an event to be completed quickly. As illustrated in the example above, workflow relating to a particular event, such as a media asset upload, can be spread among the various services simultaneously. Moreover, because the system's modularity enables it to be scaled to accommodate differing hardware capacities, and because the system can be configured to dynamically allocate hardware to different services according to the needs of the system, the speed of completing tasks relating to a particular event can further be increased. For example, a server of the CHIMPS 110 can be configured to dynamically switch its purpose based on external conditions such as load and overall system performance, providing functions such as transcode, upload, metrics collection, application web service, and more, on an as-needed basis.

[0045] Embodiments of such systems may include other systems that manage various requests from end users. For example, a system for dynamic index file generation and media file chunking. Referring to **FIG. 5A**, shows an embodiment of such a system 500-1. Media may be streamed to end user device 140 through a client 510. As mentioned above, the client 510 can be stand-alone media player, a plug-in, a browser, or other application, which can be executed on a personal computer or other electronic device.

[0046] An index file generator 530, as discussed previously, can be a program instantiated for media streaming to a particular client 510. The index file generator 530 can be executed on a server or other computing device within an application center 112 of the CHIMPS 110. Index files generated by the index file generator can include a wide variety of information such as starting, ending, and/or run times for media chunks and locations for media chunks. This information can be embedded in a single string of data, such as a URI or a URL. If media includes various sub-streams (e.g., streams with alternative bitrates, captions, alternative languages, etc.) the index file can include data for chunks corresponding to each of the alternative sub-streams, as well as information regarding the bitrate and/or other unique information for each stream. Alternatively or in addition, index files indicating alternative sub-streams may be separate from index files indicating one or more media chunks for streaming.

[0047] It should be understood that the index file can further comprise a wide variety of formats, which can depend on the particular protocol. HTTP streaming may, for example, require index files to comprise one or more of M3U, M3U8, XML, and XML-based formats. Of course, other formats can be used in accordance with relevant streaming protocols.

[0048] Table 1 illustrates a simplified example of a generated index file in M3U9 format, indicating chunk of media for streaming. The index file in this example provides a URI for a chunk of media. The URI indicates the chunk is to be generated by dynamic segmentor 550, the chunk being 10 seconds long, starting at 9 seconds into the media file and ending 19 seconds into the media file.

Table 1: Example Index File Contents

<pre>#EXTM3U #EXT-X-MEDIA-SEQUENCE:1 #EXT-X-TARGETDURATION:10 #EXTINF:10, http://video.example.com/seg/9/19/seg1.ts</pre>

[0049] Referring again to FIG. 5A, the index file generator 530 can also include an indicator within an index file to indicate whether a chunk of media is to be dynamically created. If, for example, it is determined that a requested media asset has not been chunked and that the asset will be chunked dynamically, the index file generator can include the indicator in data corresponding to a chunk of media to be created. The indicator, which can

be as simple as including the term “/seg/” in a URL, will indicate that a requested chunk of media needs to be generated.

[0050] The chunks of media can be generated during media streaming by a dynamic segmentor 550, which can be incorporated into an HTTP service 540. The HTTP service 540, as well as the media asset origin 560 can be located within a kernel application center 111 of the CHIMPS 110 or, for example, a media asset origin server. The system 500-1 can be configured such that the kernel application center 111 provides dynamically-created chunks of media to a MFDSP 150 for delivery to client 510. The MFDSP 150 can store the chunks locally in, for example, a media asset cache 520, thereby forgoing the need to dynamically create a chunk again if the same chunk is requested in the future.

[0051] In sum, the system for dynamic index file generation and media asset chunking 500-1 can, after receiving a request for an index file from a client 510, dynamically generate an index file with an index file generator 530. The index file can, among other things, indicate where a next chunk of media may be located. A client can then request the chunk from the location indicated by the index file, which can comprise a media asset cache 520 in a MFDSP 150. If the chunk is not found in the media asset cache 520, the cache miss can redirect the request to a segmentor 550 of an HTTP service 540, which can dynamically generate the requested chunk of media by accessing the corresponding media asset in the media asset origin 560. The requested media chunk can then be provided to the MFDSP 150 for storage in the media asset cache 520 and delivery to the client 510. If the same chunk is requested at a later point in time, the MFDSP 150 can deliver the chunk from the media asset cache 520, thereby forgoing the need to redirect the request to the segmentor 550 to regenerate the chunk.

[0052] FIG. 5B illustrates an alternative embodiment 500-2 of a system for dynamic index file generation and media file chunking. Rather than utilize a MFDSP, this embodiment 500-2 includes a media caching server within an application center 112 of the CHIMPS 110. The media caching server can receive chunk requests from and provide the corresponding chunks to a client. It will be understood that such a media caching server(s) or similar device(s) can be located anywhere within the CHIMPS and/or in a system(s) communicatively linked to the CHIMPS.

[0053] FIG. 5C illustrates an embodiment 500-3 of a system for index file generation used in conjunction with a redirector 590. As discussed above, an index file generator 530

may create an index file having one or more URIs or other location information directing the client 510 to one or more redirectors 590. Redirector 590, can then provide the URI of the requested chunk with a redirect, such as a redirect under HTTP status code 302. The URI can be located on a MFDSP 520 or other location (such as a media caching server 570) and/or dynamically created by the dynamic segmentor 550. It will be understood that there can be any number of redirectors 590, which can be at any location, including locations of the CHIMPS 110 such as the application center 112 (as shown in FIG. 5C) or the kernel application center 111. It also will be understood that the URI or other location information provided by redirector 590 can be generated by or provided to the redirector 590 in any number of ways. The URI can be generated, for example, based on the request received by redirector 590 from client 510. Finally, it will be understood that the CHIMPS 110 can be configured to dynamically implement any combination of embodiments 500-1, 500-2, and 500-3, further choosing whether to utilize one or more redirectors based on factors such as, for example, a detected type of client 510.

[0054] Embodiments utilizing one or more redirectors can have several advantages. For example, and not by way of limitation, if a certain client were implemented in such a way that it “reads ahead” to request chunks, it could result in incorrect reporting data. Thus, it would be advantageous to determine which chunk is actually requested by the client. Additionally or alternatively, where chunks are available in various sub-streams with different bitrates, determining the actual requested chunk can be useful in calculating Quality of Service (QOS) metrics. Furthermore, it there may be scenarios in which it is more efficient to create larger index files having many chunks comprising large segments of media, reducing the number of index files required to stream a media asset, and thereby reducing the processing requirements to create the index files. If encryption is used having, for example, a rotating key or a per client key encryption scheme in which a valid key might change during playback of a media asset, it also may be advantageous to incorporate redirector(s) for handling legacy keys for some period of time.

[0055] FIG. 6 illustrates a simplified flowchart of an embodiment of a method 600 for implementing a dynamic index for media streaming. The method 600, which can be executed by the index file generator 530, begins at block 610, where a request for an index file is received from a client 510. According to media streaming protocols contemplated by this embodiment, if data regarding a chunk of media is not provided in an initial index file, the client will continue to request or refresh an index file until it reaches an indicator in the

index file that signals the end of a stream. Thus, the method can be assured to receive more than one request for an index file from a client provided that an initial index file does not include an indicator signaling the end of the stream.

[0056] At block 615, the method 600 additionally provides for receiving input from an advertising service. According to some embodiments, this input could be the availability of an advertisement, and can be provided by a service inside or outside the CHIMPS. In other embodiments, the input could come from a service that factors in any of a variety of factors to indicate that a specific advertisement or type of advertisement should be shown. Or that any advertisement should be shown.

[0057] At block 620, a determination is made whether to include an advertisement in the next chunk. According to some embodiments, this determination can be made with or without input from an advertisement service. It should be known that this determination can include the factors used by an advertisement service to provide the input of block 615. Whether the determination includes input from an advertisement service of block 615 or not, the determination can still include factors such as information about an end user collected before or during streaming of the media. This can include behavior of the end user during streaming of the media (as determined, for example, by machine-based logic through beaconing data and/or requested chunks of media provided by a client 510). Factors can also include information regarding the media asset used for streaming (such as type of content or preferred points within the media for an advertisement), preference(s) and/or selection(s) of an end user, when a previous advertisement was shown, time of day, and more. It can further include information regarding the source of a media asset, such as who created and/or provided the asset for viewing by an end user. It will be understood that other embodiments contemplate include secondary media, other than advertisements into the media stream in this manner. Moreover, the secondary media and/or advertisement can be of any length and also may be chunked. Thus, it may be determined that the next chunk includes all, or a select portion, of an advertisement of any specific length.

[0058] An index file is created based on the request as well as the determination of whether media, such as an advertisement, should be streamed, indicated by block 625. As discussed above, the index file can assume a variety of formats and include any amount of information regarding a next chunk of media for streaming. For example, HTTP streaming can utilize index files having the URLs of available chunks of media. Information can be

embedded in these URLs to indicate a location to download the corresponding chunk of media, starting point and/or ending point of a chunk of media, an indicator to indicate whether the chunk is to be dynamically created by a segmentor 550, a location of an advertisement to be streamed, and more. This information is included in the index file and sent to the client at block 630.

[0059] At block 635, reporting data can be created based on information included in the index file. As previously discussed, information included in an index file and/or index file request can indicate the behavior of an end user device 140 during streaming, such as a pause, stop, skip, play, etc. of the media. According to some embodiments, this information can be extracted from requests for an index file and/or providing the requested index file. The information can be gathered in addition to or as a substitute for beaconing data provided by a client 510. Moreover, if beaconing data is provided, the creation of reporting data may be omitted altogether.

[0060] Reporting data can include any amount of information regarding end user behavior, as indicated through index file requests and/or provided index files. This can include a particular action and when it was performed. Additionally or alternatively, the data may be kept in a more rudimentary form, depending on the application or embodiment, indicating the data included in an index file request and/or an index file. This reporting data may be stored in a log file for reporting after streaming and/or transmitted during streaming to a relevant service that collects such metrics.

[0061] As indicated by block 640, the reporting data may be sent to a metrics collector for analytics. A metrics collector, according to certain embodiments, may be an application executed by a server from within the application center 112 in which the index file generator 530 is executed, or it may be executed elsewhere, such as in a kernel application center 111 or in a system outside the CHIMPS 110. Depending on the form of the reporting data, the metrics collector can further process and store the information.

[0062] FIG. 7 illustrates a simplified flowchart of an embodiment of a method for dynamically chunking a media file for streaming 700. This method can be employed by a variety of systems and/or programs. For example, it may be executed by a dynamic segmentor 550 of an HTTP service 540 running on a server located in a kernel application center 111 of the CHIMPS 110.

[0063] The method 700 can begin at block 710, when a request for a chunk of media is received from a MFDSP 150. As discussed above, this request may be made in response to a cache miss at the MFDSP 150 and/or because an indicator was included in the request for the chunk of media that the chunk was to be created dynamically. As discussed herein, if the MFDSP 150 has the requested chunk cached from a prior request, the MFDSP 150 can provide the requested chunk and preclude the need to send the request to a dynamic segmentor 550 to generate the chunk. It should be understood that the request may come from sources other than a MFDSP 150 according to alternative embodiments. One such source includes the media caching server 570 of embodiment 500-2, as shown in FIG. 5B.

[0064] The starting and ending points of a requested chunk of media are then determined at block 715. This information can be included directly in the request or derived from the request, a previous request, and/or other sources. At block 720, the information, as well as information identifying the requested chunk of media, can be used to retrieve all or part of the relevant media asset from a media asset origin 560. The retrieved portion will include at least the relevant media from the starting point to the ending point of the requested chunk of media.

[0065] At block 725, the requested media chunk is generated by converting the relevant portion of the media asset into a deliverable chunk. The media asset, as stored in the media asset origin, may not be chunked; it may be stored in its entirety as a media file (or group of alternative files corresponding to alternative sub-streams). Generating the chunk therefore can require determining the starting and ending points from the retrieved portion of the media asset and converting the resulting segment of media into a deliverable chunk.

[0066] Although the generation of the deliverable chunk may involve transcoding, it may not. The media asset can be stored in a format where transcoding may not be needed, thereby reducing the processing requirements for creating chunks of media during streaming. For example, media assets may be stored such as H.264 or MPEG-4 video format and/or AAC, HE-AAC, or MP3 audio format. According to some streaming protocols, such as some forms of HTTP streaming, chunks of media in these formats would not need transcoding before being wrapped in an MPEG-2 transport stream container format. Instead, such a conversion essentially would require the addition of metadata to create the streaming format from the format of the stored media asset. In other words, generating a deliverable chunk of media may only require identifying the stored media

asset, extracting the relevant segment of the media from the media asset, and adding certain metadata in accordance with a container format. This process requires little processing power and can be easily performed on the fly during streaming. Once the deliverable chunk of media is generated, it is sent to the MFDSP 150 or other requesting entity, at block 730.

[0067] **FIG. 8** illustrates a simplified swim lane flowchart describing the interaction of components in a system configured to provide dynamic indexing and chunking for media streaming, according to one embodiment. In this embodiment, a client can send a request for an index file 805, the request received by an index file generator 810. A particular request may be made to initiate the streaming of a media asset, or it may be during streaming. Depending on the streaming protocol, the request may be made while a client plays a chunk of media previously downloaded during streaming.

[0068] The index file generator generates an index file to indicate the next chunk of media 815. As described above, this chunk may include an advertisement, and the index file can include any amount of information about a chunk of media, including information regarding alternative sub-streams for streaming. The dynamic index file generator can include information regarding existing chunks of media, and, when used in conjunction with a dynamic segmentor may also include information regarding chunks of media that may need to be created. As detailed above, if a chunk of media is to be generated dynamically, the index file generator may indicate this by including an indicator in the generated index file, such as in a URL for one or more chunks described within the index file. Once the index file is generated, the index file generator sends the index file 820, which is received by the client 825.

[0069] Alternative embodiments may provide for the generation of index files containing more than a next chunk of media. For example, an index file generator may generate an index file containing information regarding several chunks of media, in which case the chunks of media can be dynamically generated by a dynamic segmentor when requested by the client. The determination of whether to include information regarding more than a next chunk of media can include factors such as whether the index generator is generating reporting data, the desired frequency of such reporting data, and more.

[0070] Using information contained in the index file, the client can then request the next chunk of media 830, and this request can be received by a MFDSP 835. The MFDSP then checks to see if the chunk is already stored in the cache 840. If so, the MFDSP can provide

the requested chunk to the client, blocks 845 and 850. The requested chunk may be found in a MFDSP's cache if the chunk was created and stored in the MFDSP during preprocessing or if the chunk was dynamically created and stored in the MFDSP from an earlier request.

[0071] If the chunk is not found on the MFDSP, the chunk can be requested of the dynamic segmentor, which receives the request 855 and retrieves the corresponding media asset from an asset origin server 860. As discussed above, the entirety of the relevant media asset does not need to be retrieved as long as at least the portion containing the relevant segment for the requested chunk is retrieved. It will be understood that alternative embodiments can provide for the media asset being stored in a variety of locations accessible, directly or indirectly, to the dynamic segmentor.

[0072] The dynamic segmentor can then generate the requested chunk by converting the retrieved media into a deliverable chunk. That is, the dynamic segmentor converts the retrieved media into an acceptable format for streaming, which can vary depending on the streaming protocol utilized. The dynamic segmentor can then return the requested chunk 870 to the MFDSP, which can cache the chunk and return it to the client 875. Once the chunk is received by the client 880, the client can play the chunk to an end user.

[0073] The techniques for creating index files and/or creating corresponding chunks of media can be adapted for each request. This allows the CHIMPS 110 (or similarly-enabled system) to provide optimal chunking for each delivery instance, which can differ even among similar devices, depending on any of a variety of factors. For example, a new version of a particular device type (e.g., a particular smart phone, set-top box, tablet, etc.) may have the same operating system and/or browser as an older version of the device type, but with a larger buffer than the older version. As such, optimized chunking for delivery instances involving the old and new versions of the device type may differ in that delivery to the new version can include using chunk sizes that would cause a buffer overflow in the older version. By dynamically determining and implementing a chunking strategy for each delivery instance in this manner, the CHIMPS 110 (or similarly-enabled system) can help provide the optimal user experience under any of a variety of circumstances.

[0074] FIG. 9 is a simplified flow chart illustrating a basic method 900 for adapting chunking techniques for each delivery instance. At block 910, a request for a media file is received. In other embodiments, the request may be for a live stream or other media asset

not stored as a media file. At block 920, one or more chunking considerations are determined. Because the streaming of a media file can involve multiple requests (and, correspondingly, multiple index files), the method 900 may be performed multiple times during the playback of a media file.

[0075] Chunking considerations are factors that can impact any aspect of the chunking of a media file and/or advertisements included in the delivery of the media file. Such considerations can include, but are not limited to, bandwidth, codec, processor type, buffer size, memory and/or processor utilization, battery, network type, geographic location, whether a device is moving, a location in the playback of the media file, and the like.

[0076] The bandwidth of a device, for example, can impact the size of the chunks delivered to the device. For a relatively low bandwidth, the chunk size may be reduced so that chunks are downloaded more quickly, which may reduce the chance that a media player would need to interrupt playback to while waiting for the next chunk to be delivered. Conversely, chunk size can be increased for higher bandwidths. Similarly, the chunks may be adapted to accommodate a particular network type (e.g., WiFi, mobile wireless network, land line (wired), etc.), which can determine, or be indicative of, an available bandwidth.

[0077] Different aspects of the hardware of a device can also be considered when chunking a media file. In addition to the buffer size, as indicated above, a processor type can also impact how a file is chunked—such as the size of the chunk and/or the codec used. Additionally or alternatively, QOS metrics can indicate memory and/or processor utilization, including during playback of the requested media file, which can impact the chunking of the media file.

[0078] Chunking considerations further can include a variety of factors related to mobile devices. For example, whether a device is situated in a particular geographic location, at a particular event (where large numbers of mobile devices may be located), and/or the device is moving can impact the available bandwidth to the device. The battery level of a mobile device may also impact the bandwidth in certain circumstances. The CHIMPS 110 (or other chunking system) can anticipate changes in bandwidth by determining such chunking considerations and adjusting the chunking accordingly. Moreover, QOS, bandwidth, and/or other information from multiple devices can be aggregated to help the CHIMPS 110 further adapt media chunking for related delivery instances. Thus, the chunking considerations for streaming to one device may include information regarding one or more other devices. For

example the CHIMPS 110 may anticipate a sudden reduced bandwidth for a particular delivery instance related to a cell phone where it is determined that other cell phones nearby on the same carrier have had sudden drops in bandwidth.

[0079] Chunking considerations further can include a type of codec used and/or a location in the playback of the media file. Certain codecs, for example, may require or prohibit the use of certain-sized chunks. Also, for example, if the playback of the media file has just started, smaller chunks may be used to help ensure that playback begins quickly, without the need to wait for larger chunks to be delivered. Also, chunk sizes can be altered to accommodate ad insertion or similar interleaving of media content at particular point(s) in the playback of the media file.

[0080] Chunking considerations may be determined from information included in the request, a database, and/or other sources, which may disclose data such as a URL, client ID, globally-unique identifier (GUID) or other identifier, a network type, a device type and/or capability, an operating system executed by the end user device, an application, application identifier, application state, or other application information, a location associated with the device, information associated with a user of the end user device 140, information regarding the requested media file (e.g., genre, length, rating(s), ownership, artist, etc.). Additionally or alternatively, the request may simply include information that enables one or more of these items to be determined. Additionally or alternatively, a repository may be stored, maintained or derived, and queried for authorization, authentication, validation, or selection; for example, a repository of application identifiers may be maintained and queried to determine whether an application is authorized to request the content and if so, to select further aspects of or for processing the content request. Additionally or alternatively, such stored or derived repository data may be used in conjunction with other data, either internally or externally identified, such as a secret key, shared key, public key, stored certificate, other stored data, or other data for authorization, authentication, validation, or selection, including data stored on another digital service, on another server, on the client device, in a device associated with the client device, in the operating system, in the application, in another application, in a network, or in another location from which it may be retrieved.

[0081] As indicated previously, the determination of chunking considerations can include utilizing information other than the information provided in the request. This may involve

accessing information stored in one or more a databases or other data structures internal or external to the CHIMPS 110. It may also involve communicating with other entities and/or systems, such as a content owner or media provider 130. Additionally or alternatively, chunking considerations can be gathered using data independent of information provided in the request, such as the time at which the request was received.

[0082] At block 930, a chunking strategy based on chunking considerations is determined. And at block 940, the corresponding index file and chunks are provided accordingly. As indicated above, the chunking strategy (i.e., the way in which the chunks are created and delivered) can be impacted by the chunking considerations, including the size of the chunks, the length of the playback of the chunks, and the choice of codec used. Furthermore, multiple considerations can be taken into account to determine the chunking strategy.

[0083] This chunking strategy can be implemented on the fly (i.e., immediately before or during playback of the media file) utilizing the dynamic indexing and chunking techniques discussed above. For example, an index file generator 530 can incorporate the chunking strategy into a dynamically-generated index file, and the dynamic segmentor 550 can produce the requested chunks accordingly. Because the contextual data and chunking strategies can vary for each delivery instance, the content of the index files corresponding to requests for the same media file can be different (e.g., indicate different chunk sizes, codecs to use, etc.), based on differing chunking strategies.

[0084] It should be noted that the methods, systems, and devices discussed above are intended merely to be examples. It must be stressed that various embodiments may omit, substitute, or add various procedures or components as appropriate. For instance, it should be appreciated that, in alternative embodiments, the methods may be performed in an order different from that described, and that various steps may be added, omitted, or combined. Also, features described with respect to certain embodiments may be combined in various other embodiments. Different aspects and elements of the embodiments may be combined in a similar manner. Also, it should be emphasized that technology evolves and, thus, many of the elements are examples and should not be interpreted to limit the scope of the invention.

[0085] Specific details are given in the description to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these specific details. For example, well-known

circuits, processes, algorithms, structures, and techniques have been shown without unnecessary detail in order to avoid obscuring the embodiments. This description provides example embodiments only, and is not intended to limit the scope, applicability, or configuration of the invention. Rather, the preceding description of the embodiments will provide those skilled in the art with an enabling description for implementing embodiments of the invention. Various changes may be made in the function and arrangement of elements without departing from the spirit and scope of the invention.

[0086] Also, it is noted that the embodiments may be described as a process which is depicted as a flow diagram or block diagram. Although each may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be rearranged. A process may have additional steps not included in the figure. Furthermore, embodiments of the methods may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware, or microcode, the program code or code segments to perform the necessary tasks may be stored in a non-volatile computer-readable medium such as a storage medium. Processors may perform the necessary tasks.

[0087] Having described several embodiments, it will be recognized by those of skill in the art that various modifications, alternative constructions, and equivalents may be used without departing from the spirit of the invention. For example, the above elements may merely be a component of a larger system, wherein other rules may take precedence over or otherwise modify the application of the invention. Also, a number of steps may be undertaken before, during, or after the above elements are considered. Accordingly, the above description should not be taken as limiting the scope of the invention.

WHAT IS CLAIMED IS:

1. A method for providing media with a data network, the method comprising:
 - receiving a first request having a universal source locator (URL), wherein the URL includes information indicative of a requested media file;
 - determining one or more factors related to the first request;
 - determining a first chunking strategy based on the one or more factors related to the first request;
 - generating a first index file having information for streaming the requested media file via the data network, wherein generating the first index file is based, at least in part, on the first chunking strategy;
 - providing the first index file;
 - receiving a second request having the URL;
 - determining one or more factors related to the second request, wherein the one or more factors related to the second request is different than the one or more factors related to the first request;
 - determining a second chunking strategy based on the one or more factors related to the second request;
 - generating a second index file having information for streaming the requested media file via the data network, wherein: the generating the second index file is based, at least in part, on the second chunking strategy; and
 - content of the second index file is different from content of the first index file;
 and
 - providing the second index file.
2. The method for providing media with the data network as recited in claim 1, wherein the one or more factors related to one or both of the first request or the second request comprises data corresponding to one or more of:
 - a bandwidth, a codec, a processor type, a buffer size, a memory utilization, a processor utilization, a battery, a network type, a geographic location, whether a device is moving, or a location in the playback of the requested media file.
3. The method for providing media with the data network as recited in claim 2, wherein the network type includes mobile wireless network or wired network.

4. The method for providing media with the data network as recited in claim 1, wherein one or both of the first chunking strategy or the second chunking strategy includes one or more of: a size of a chunk for streaming the requested media file, a length of playback of a chunk for streaming the requested media file, or a codec to use for streaming the requested media file.
5. The method for providing media with the data network as recited in claim 1, wherein generating the first index file occurs during playback of the requested media file.
6. The method for providing media with the data network as recited in claim 1, wherein: the first index file is provided to a first device; and the one or more factors related to the first request include information regarding a second device.
7. The method for providing media with the data network as recited in claim 1, wherein one or both of the first chunking strategy or the second chunking strategy includes providing smaller chunk sizes near the beginning of playback of the requested media file.
8. A server for providing a media file with a data network, the server comprising:
 - a network interface for communicating with the data network;
 - a memory; and
 - a processor communicatively coupled with the memory and the network interface, the processor further configured to cause the server to:
 - receive a first request having a universal source locator (URL), wherein the URL includes information indicative of a requested media file;
 - determine one or more factors related to the first request;
 - determine a first chunking strategy based on the one or more factors related to the first request;
 - generate a first index file having information for streaming the requested media file via the data network, wherein the generating the first index file is based, at least in part, on the first chunking strategy;
 - provide, via the network interface, the first index file;
 - receive a second request having the URL;

determine one or more factors related to the second request, wherein the one or more factors related to the second request is different than the one or more factors related to the first request;

determine a second chunking strategy based on the one or more factors related to the second request;

generate a second index file having information for streaming the requested media file via the data network, wherein:

generating the second index file is based, at least in part, on the second chunking strategy; and

content of the second index file is different from content of the first index file; and

provide, via the network interface, the second index file.

9. The server for providing the media file with the data network as recited in claim 8, wherein the processor is configured to cause the server to determine the one or more factors related to one or both of the first request or the second request from data corresponding to one or more of: a bandwidth, a codec, a processor type, a buffer size, a memory utilization, a processor utilization, a battery, a network type, a geographic location, whether a device is moving, or a location in the playback of the requested media file.

10. The server for providing the media file with the data network as recited in claim 9, wherein the network type includes mobile wireless network or wired network.

11. The server for providing the media file with the data network as recited in claim 8, wherein one or both of the first chunking strategy or the second chunking strategy includes one or more of: a size of a chunk for streaming the requested media file, a length of playback of a chunk for streaming the requested media file, or a codec to use for streaming the requested media file.

12. The server for providing the media file with the data network as recited in claim 8, wherein the processor is configured to cause the server to generate the first index file during playback of the requested media file.

13. The server for providing the media file with the data network as recited in claim 8, wherein the processor is configured to cause the server to determine the one or more factors related to the first request based on information regarding a plurality of devices.

14. The server for providing the media file with the data network as recited in claim 8, wherein one or both of the first chunking strategy or the second chunking strategy includes providing smaller chunk sizes near the beginning of playback of the requested media file.

15. A non-transitory computer-readable medium having instructions imbedded thereon for providing media with a data network, wherein the instructions, when executed by one or more computers, cause the one or more computers to:

- receive a first request having a universal source locator (URL), wherein the URL includes information indicative of a requested media file;

- determine one or more factors related to the first request;

- determine a first chunking strategy based on the one or more factors related to the first request;

- generate a first index file having information for streaming the requested media file via the data network, wherein the generating the first index file is based, at least in part, on the first chunking strategy;

- provide the first index file;

- receive a second request having the URL;

- determine one or more factors related to the second request, wherein the one or more factors related to the second request is different than the one or more factors related to the first request;

- determine a second chunking strategy based on the one or more factors related to the second request;

- generate a second index file having information for streaming the requested media file via the data network, wherein:

- generating the second index file is based, at least in part, on the second chunking strategy; and

content of the second index file is different from content of the first index file; and provide the second index file.

16. The non-transitory computer-readable medium having the instructions imbedded thereon for providing the media with the data network as recited in claim 15 further comprising instructions causing the one or more computers to determine the one or more factors related to one or both of the first request or the second request using data corresponding to one or more of: a bandwidth, a codec, a processor type, a buffer size, a memory utilization, a processor utilization, a battery, a network type, a geographic location, whether a device is moving, or a location in the playback of the requested media file.

17. The non-transitory computer-readable medium having the instructions imbedded thereon for providing the media with the data network as recited in claim 15 further comprising instructions causing the one or more computers to determine one or both of the first chunking strategy or the second chunking strategy that includes one or more of: a size of a chunk for streaming the requested media file, a length of playback of a chunk for streaming the requested media file, or a codec to use for streaming the requested media file.

18. The non-transitory computer-readable medium having the instructions imbedded thereon for providing the media with the data network as recited in claim 15 further comprising instructions causing the one or more computers to generate the first index file during playback of the requested media file.

19. The non-transitory computer-readable medium having the instructions imbedded thereon for providing the media with the data network as recited in claim 15 further comprising instructions causing the one or more computers to: provide the first index file to a first device; and determine the one or more factors related to the first request based, at least in part, on information regarding a second device.

20. The non-transitory computer-readable medium having the instructions imbedded thereon for providing the media with the data network as recited in claim 15 further comprising instructions causing the one or more computers to determine one or both of the first chunking strategy or the second chunking strategy to include providing smaller chunk sizes near the beginning of playback of the requested media file.

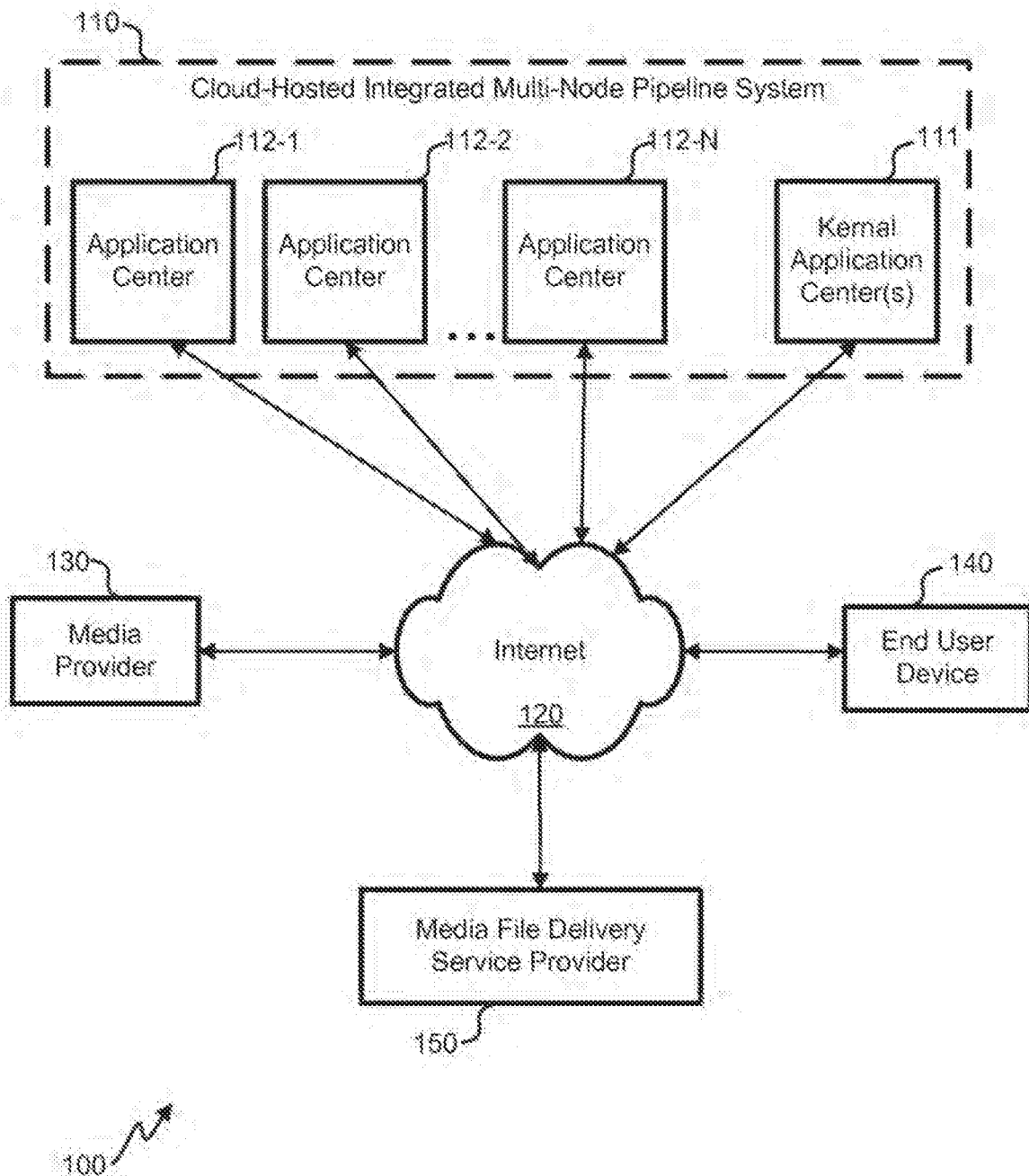
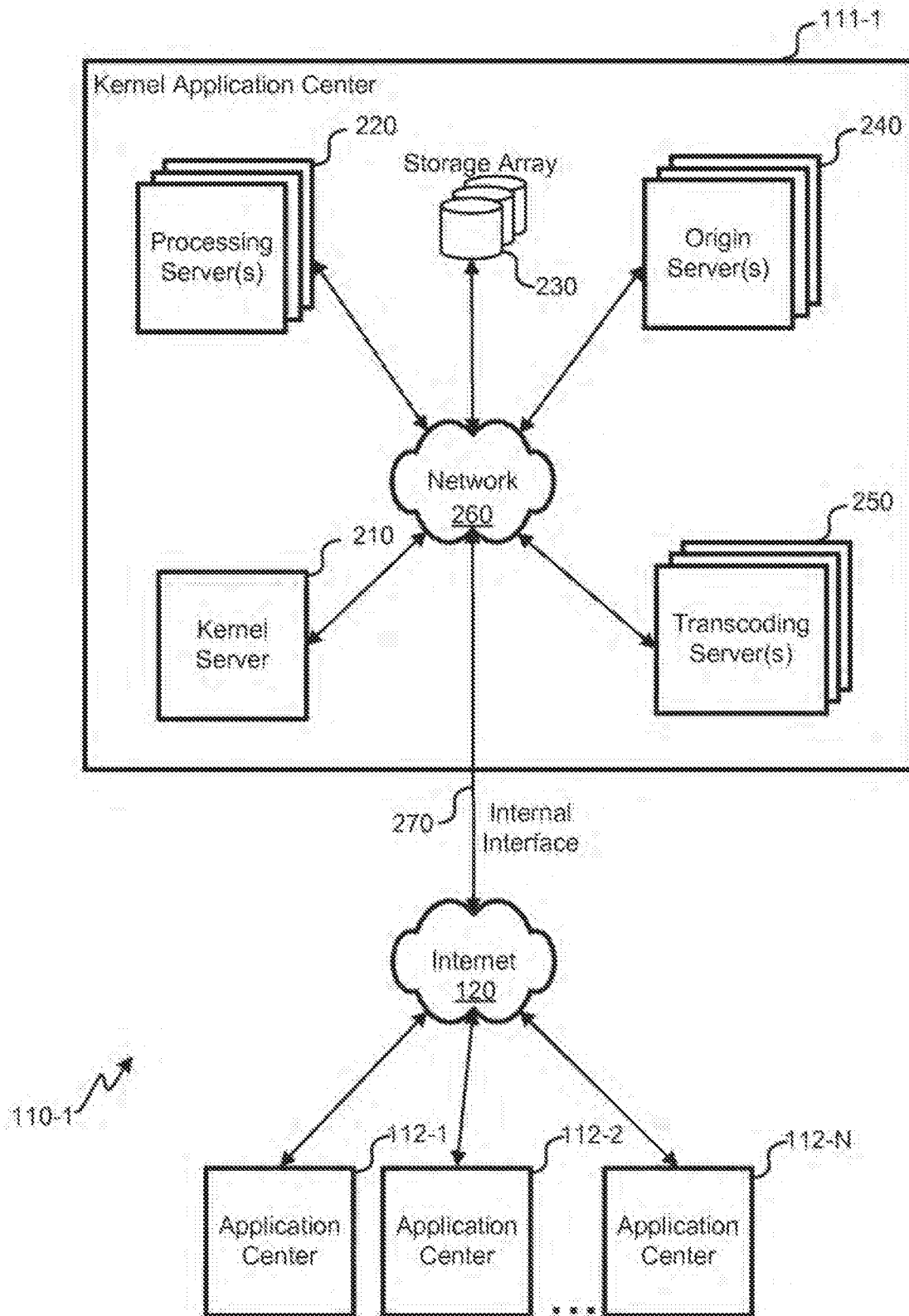


FIG. 1

2/12

**FIG. 2A**

3/12

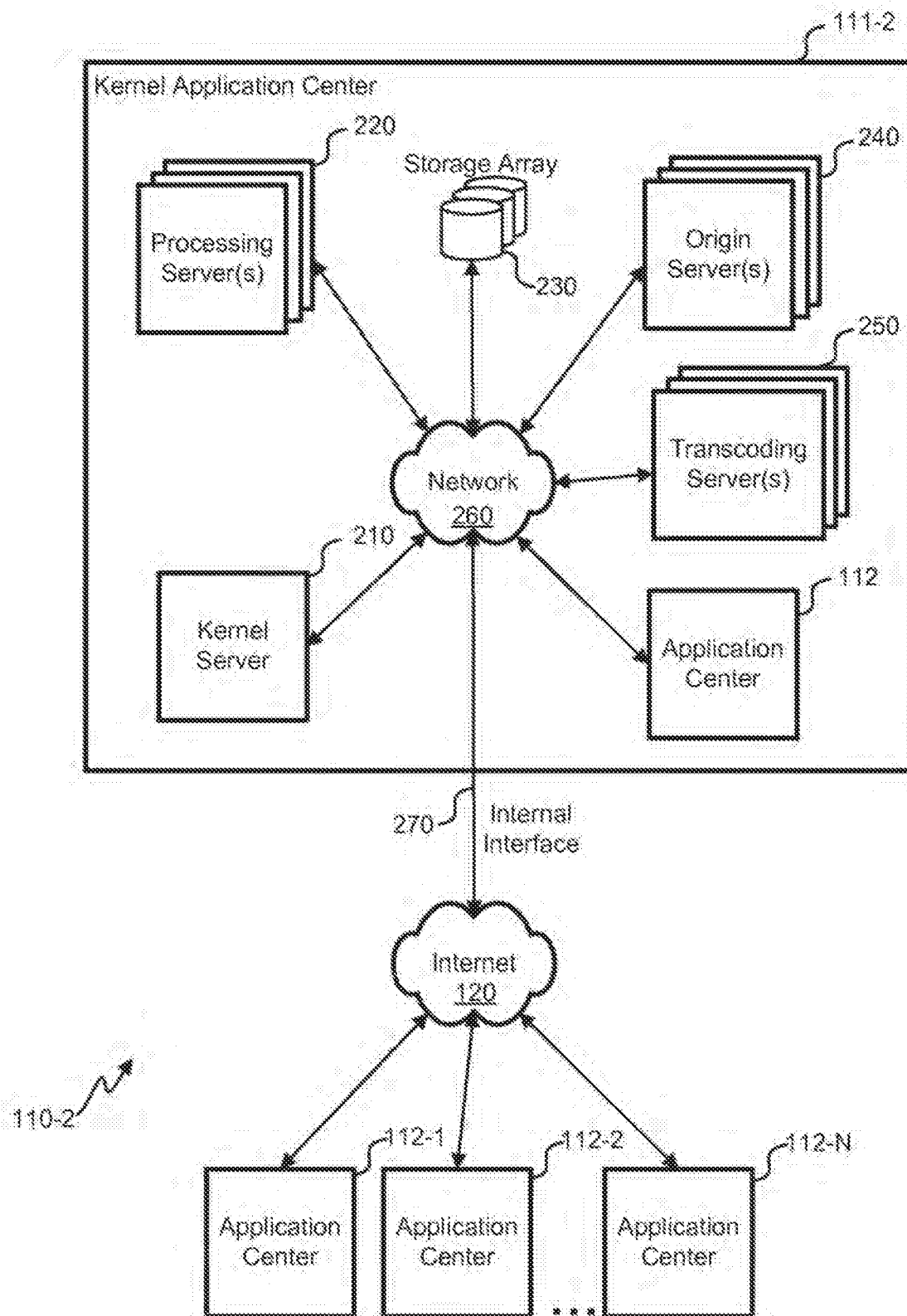


FIG. 2B

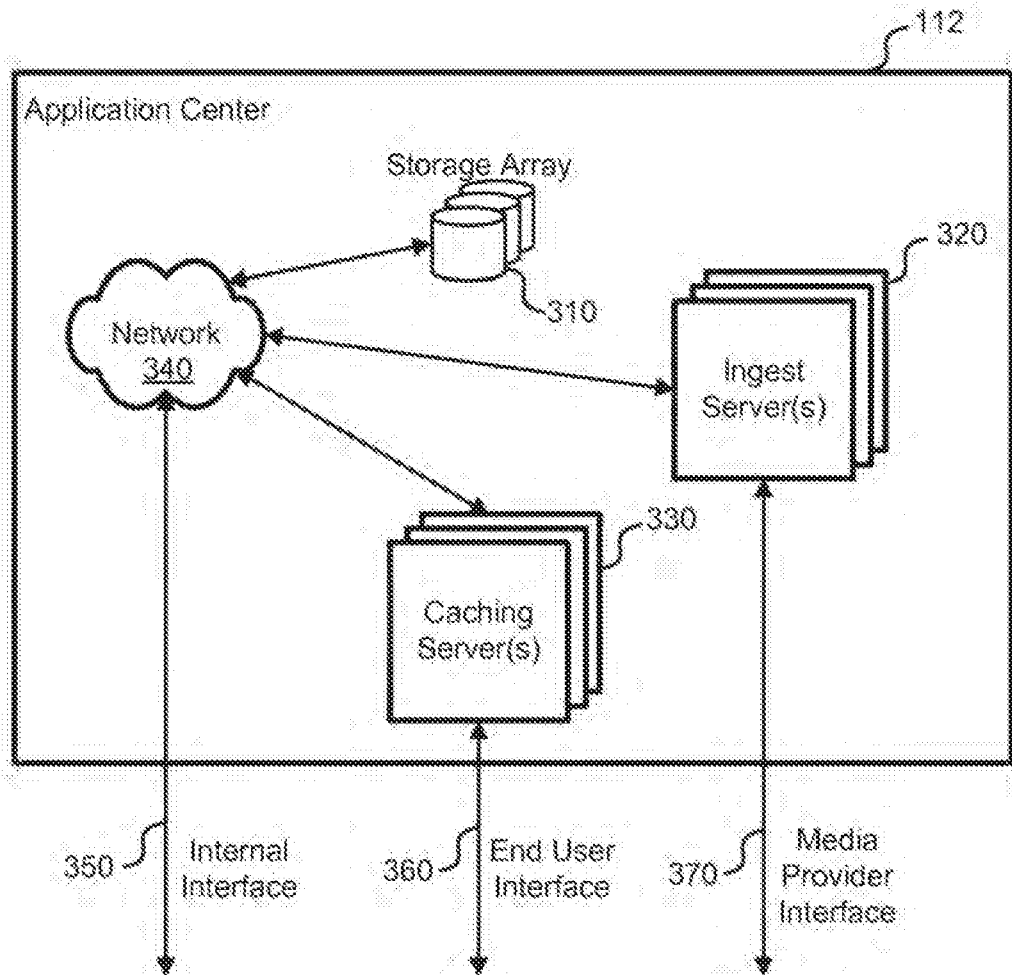


FIG. 3

5/12

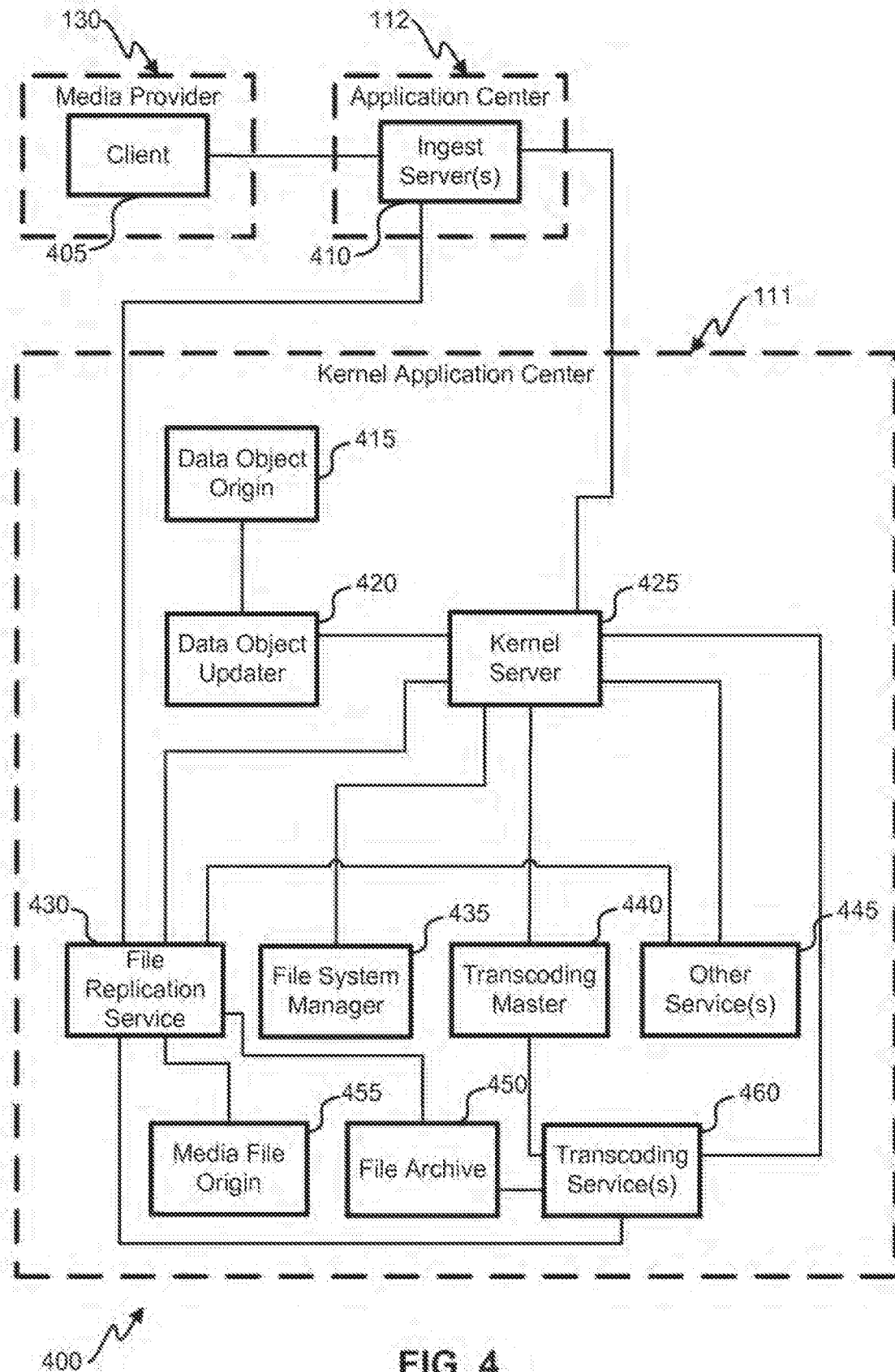
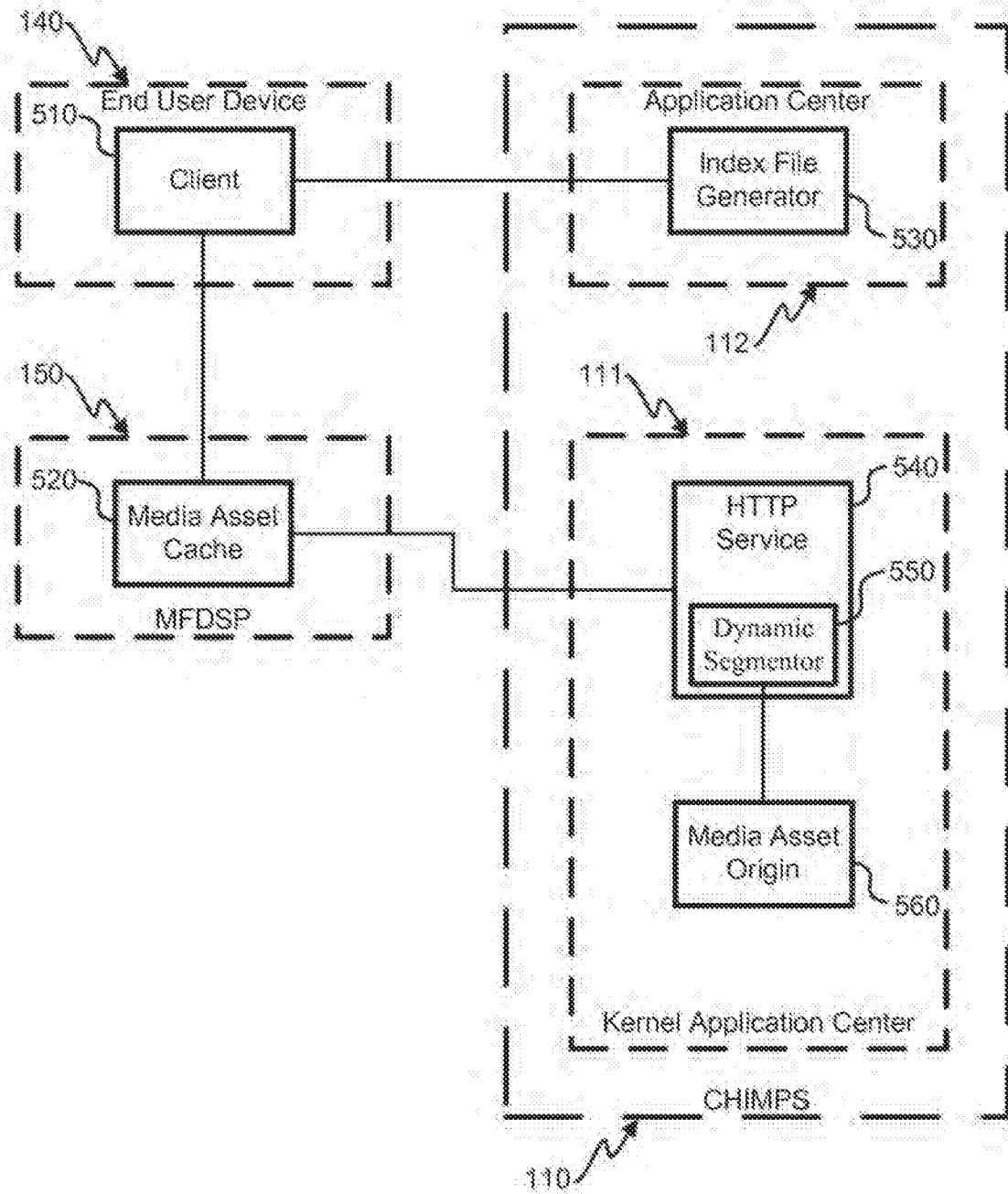


FIG. 4

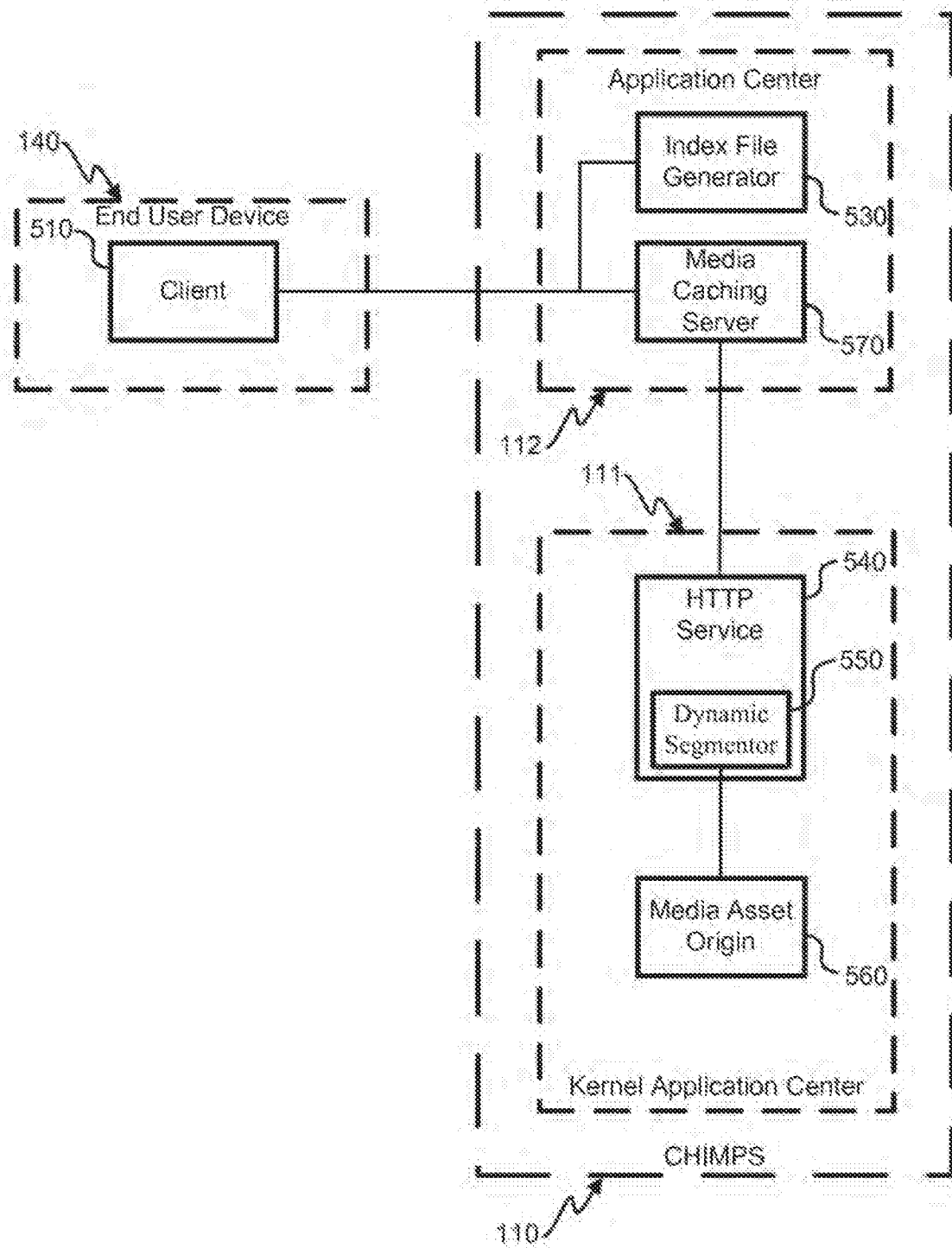
6/12



500-1

FIG. 5A

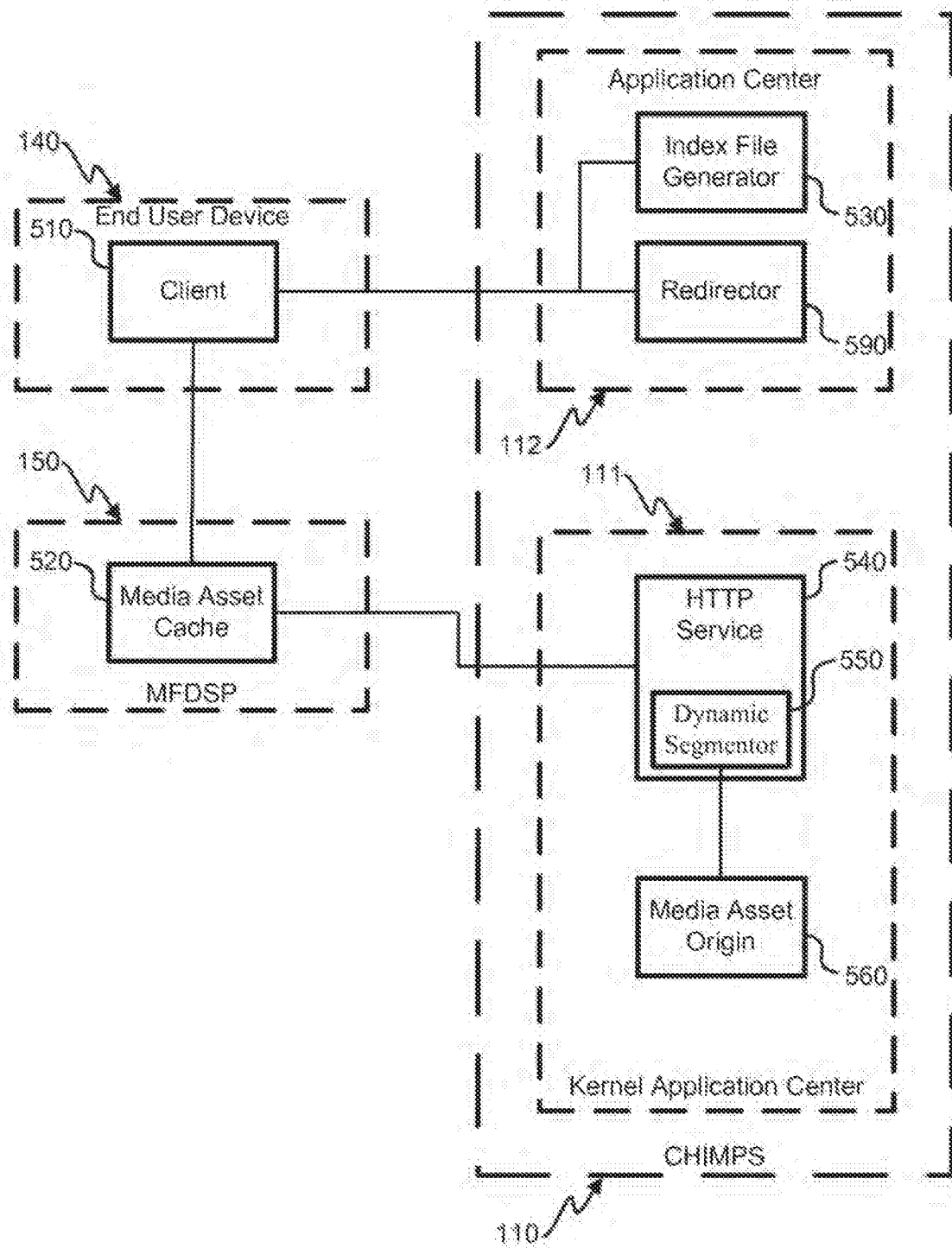
7/12



500-2

FIG. 5B

8/12



500-3

FIG. 5C

9/12

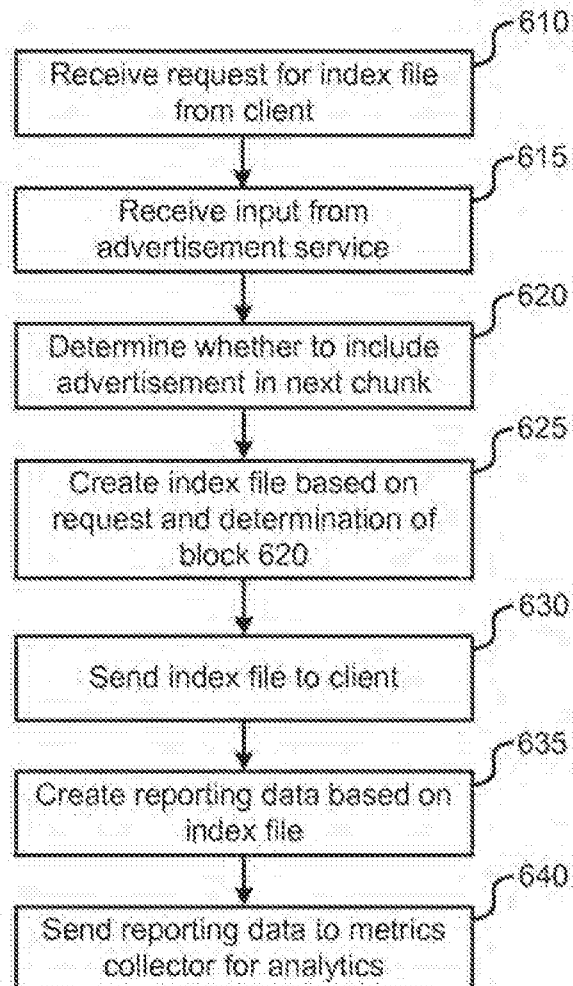
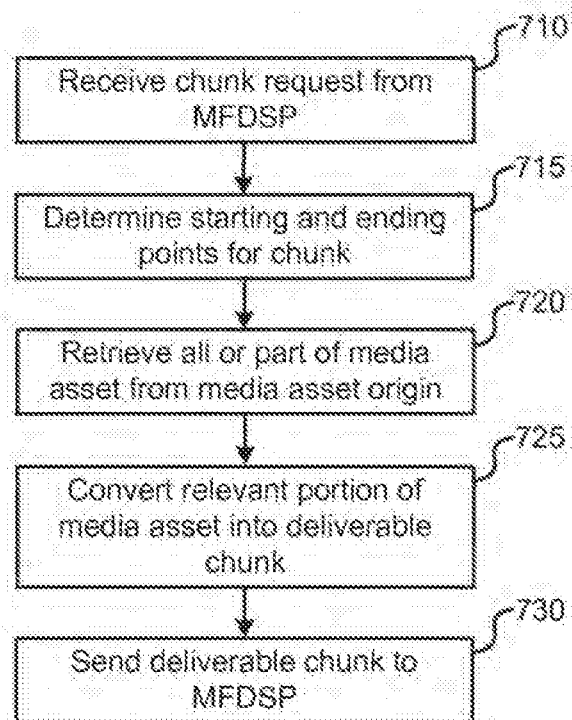


FIG. 6

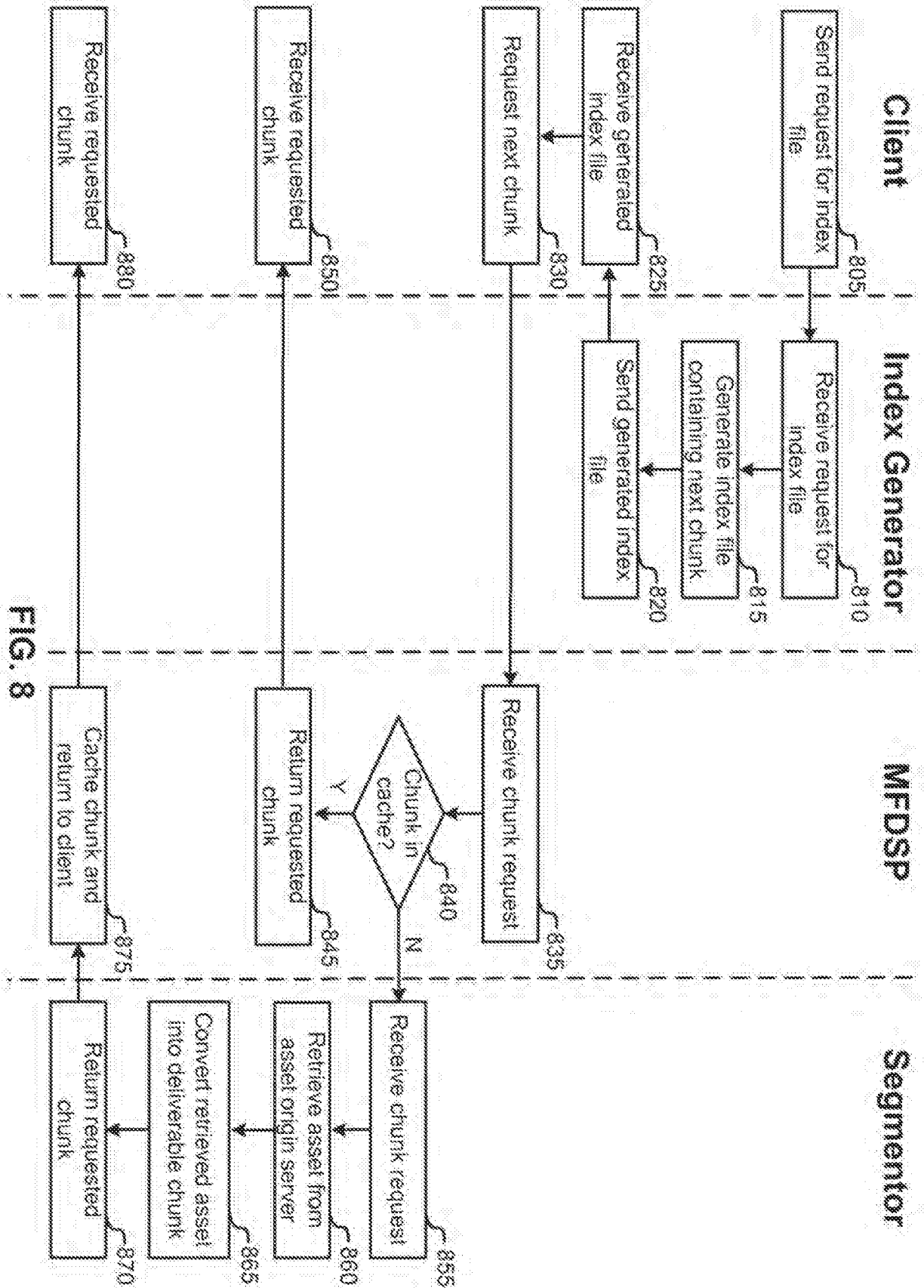
10/12



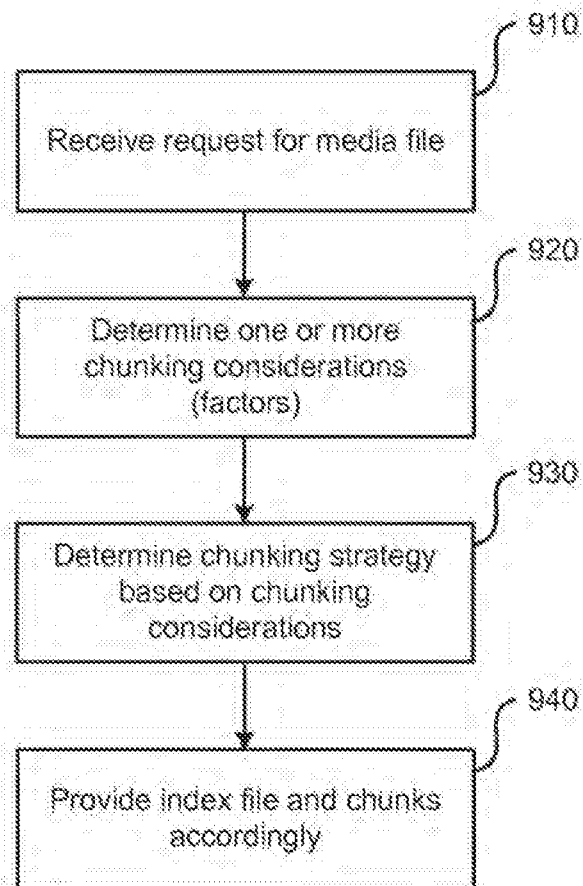
700

FIG. 7

11/12



12/12



900

FIG. 9

(21) Application No: **1412128.9**
 (22) Date of Filing: **26.12.2012**
 Date Lodged: **08.07.2014**
 (30) Priority Data:
 (31) **13339668** (32) **29.12.2011** (33) **US**
 (86) International Application Data:
PCT/US2012/071669 En 26.12.2012
 (87) International Publication Data:
WO2013/101841 En 04.07.2013

(71) Applicant(s):
Cacti Acquisition LLC
(Incorporated in USA - Delaware)
290 Congress Street, 4th Floor, Boston,
Massachusetts 02210, United States of America

(72) Inventor(s):
Michael M Gordon
Albert John McGowan

(74) Agent and/or Address for Service:
J A Kemp
14 South Square, Gray's Inn, London, WC1R 5JJ,
United Kingdom

(51) INT CL:
G06F 17/30 (2006.01) H04L 29/06 (2006.01)
H04N 21/24 (2011.01) H04N 21/266 (2011.01)

(56) Documents Cited:
US 20010029525 A1
AU2010202741
PRANGL M ET AL: "Towards QoS Improvements of
TCP-Based Media Delivery", NETWORKING AND
SERVICES, 2008. ICNS 2008. FOURTH
INTERNATIONAL CONFERENCE ON, IEEE,
PISCATAWAY, NJ, USA, 16 March 2008 (2008-03-16),
pages 188-193, XP031239297, ISBN:
978-0-7695-3094-9

(58) Field of Search:
 INT CL **G06F**
 Other: **EPO-Internal, WPI Data, INSPEC, COMPENDEX**

(54) Title of the Invention: **Dynamically-executed syndication services**
 Abstract Title: **Dynamically-executed syndication services**

(57) Systems and methods for dynamically executing syndication services are provided that automatically implement syndication information for syndication based on contextual data corresponding to a request for a media file. These systems and methods may be part of a larger media servicing network that can be used to, among other things, process uploaded media content, provide it for streaming/downloading, and collect metric information regarding the streaming/downloading. The disclosed systems and methods provide for receiving a request having a Uniform Resource Locator (URL) and providing an index file in accordance with syndication information based on contextual data associated with the request. Embodiments further enable media content owners to distribute a single URL corresponding to a particular media file among many media providers, allowing a single media delivery and analytics services to provide comprehensive metric information regarding syndication for the all the media providers.

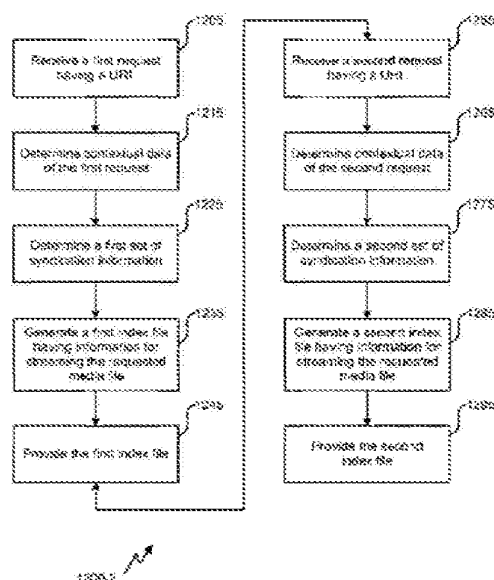


FIG. 12B

(12) UK Patent Application (19) GB (11) 2514519 (13) A

(43) Date of Reproduction by UK Office 26.11.2014

(21) Application No: 1416933.8
(22) Date of Filing: 07.02.2013
Date Lodged: 25.09.2014

(30) Priority Data:
(31) 13429656 (32) 26.03.2012 (33) US
(31) 13567681 (32) 06.08.2012 (33) US

(86) International Application Data:
PCT/US2013/025180 En 07.02.2013

(87) International Publication Data:
WO2013/148003 En 03.10.2013

(71) Applicant(s):
Cacti Acquisition LLC
(Incorporated in USA - Delaware)
290 Congress Street, 4th Floor, Boston,
Massachusetts 02210, United States of America

(72) Inventor(s):
Albert John McGowan

(74) Agent and/or Address for Service:
J A Kemp
14 South Square, Gray's Inn, London, WC1R 5JJ,
United Kingdom

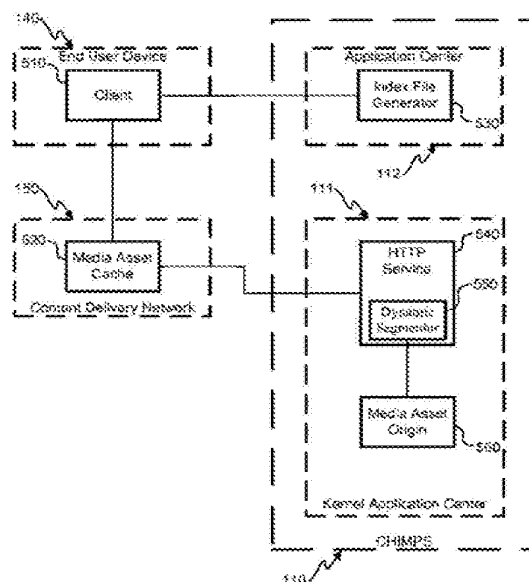
(51) INT CL:
H04L 29/06 (2006.01) H04N 21/6587 (2011.01)
H04N 21/845 (2011.01) H04N 21/854 (2011.01)

(56) Documents Cited:
WO 2011/039617 A1 US 20120023253 A1
US 20120005312 A1
GABIN F ET AL: "3GPP Mobile Multimedia Streaming Standards [Standards in a Nutshell]", IEEE SIGNAL PROCESSING MAGAZINE, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 27, no. 6, 1 November 2010 (2010-11-01), pages 134-138, XP011317684, ISSN: 1053-5886

(58) Field of Search:
INT CL H04L, H04N

(54) Title of the Invention: **Dynamic audio track selection for media streaming**
Abstract Title: **Dynamic audio track selection for media streaming**

(57) Systems and methods for providing index files for dynamic audio track selection for media streaming can include receiving requests for media file segments (e.g., "chunks") with a certain audio track and responding to the requests by dynamically generating the corresponding media file segment with the requested audio track. The generated media file segment can then be provided to the requesting entity.



500

FIG. 5A

GB 2514519 A

(51) INT CL:
H04L 29/06 (2006.01)

(56) Documents Cited:
WO 2011/009205 A1 US 20120047542 A1
US 20020029282 A1

(56) Field of Search:
INT CL H04L
Other: EPO-Internal, INSPEC

(86) International Application Data:
PCT/US2013/023181 En 25.01.2013

(71) Applicant(s):
Cacti Acquisition LLC
(Incorporated in USA - Delaware)
290 Congress Street, 4th Floor, Boston,
Massachusetts 02210, United States of America

(72) Inventor(s):
Albert John McGowan

(74) Agent and/or Address for Service:
J A Kemp
14 South Square, Gray's Inn, London, WC1R 5JJ,
United Kingdom

(54) Title of the Invention: **Dynamic chunking for delivery instances**
Abstract Title: **Dynamic chunking for delivery instances**

(57) Systems and methods for dynamically chunking for delivery instances are provided that automatically implement chunking strategies based on one or more chunking considerations related to a request for a media file. These systems and methods may be part of a larger media servicing network that can be used to, among other things, process uploaded media content, provide it for streaming/downloading, and collect metric information regarding the streaming/downloading. The disclosed systems and methods provide for receiving a request having a Uniform Resource Locator (URL) and providing an index file to implement chunking strategies based on chunking considerations associated with the request.

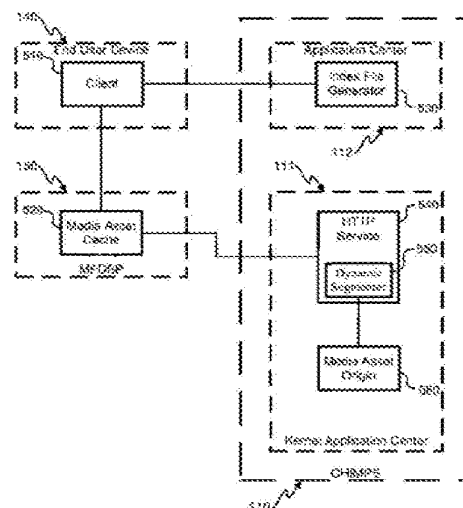


FIG. 5A

(43) International Publication Date
27 January 2011 (27.01.2011)(10) International Publication Number
WO 2011/009205 A1(51) International Patent Classification:
H04N 7/173 (2011.01)(21) International Application Number:
PCT/CA2010/001129(22) International Filing Date:
22 July 2010 (22.07.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/227,583 22 July 2009 (22.07.2009) US(71) Applicant (for all designated States except US): JIGSEE
INC. [CA/CA]; 1156 Bay Street, Suite 2105, Toronto,
Ontario M5S 2X8 (CA).

(72) Inventor; and

(75) Inventor/Applicant (for US only): REZA, Areef
[CA/CA]; 12 Ealing Street, Ottawa, Ontario K2W 1E4
(CA).(74) Agent: HILL & SCHUMACHER; 364 Avenue Road,
Toronto, Ontario M4V 2G7 (CA).(51) 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, TI, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.(54) 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, SE, SI, SK,
SM, TR), OAPI (BF, BI, CF, CG, CI, CM, GA, GN, GQ,
GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD OF STREAMING MEDIA TO HETEROGENEOUS CLIENT DEVICES

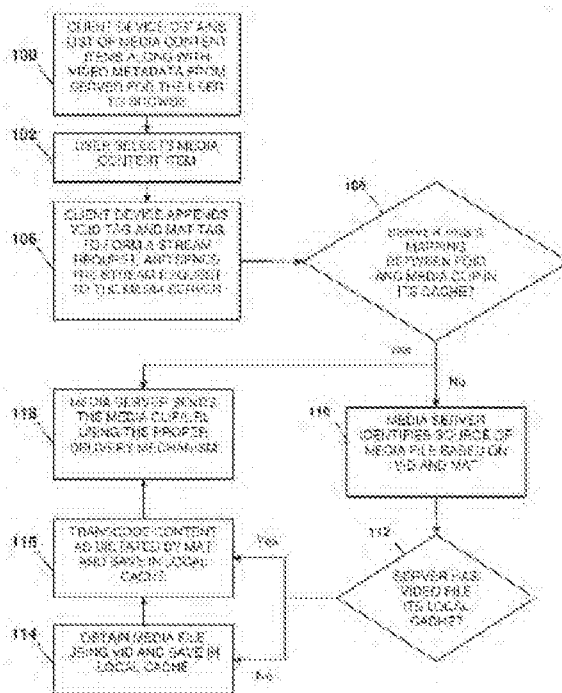


Figure 2

(57) Abstract: A method of dynamically preparing a media clip for delivery from a server to a client is provided. Media attributes comprising media playback formats compatible with the client are provided to the server. A request is made by the client to provide a media content item at a specified starting time, and the server identifies a source where a media content file related to the selected media content item is located. The server obtains the media file from the source, and obtains a transcoded media segment file having a playback format compatible with the client media attributes. The transcoded media segment file may further be prepared in a transport format that is compatible with transport format media attributes of the client device, and a location of the prepared transcoded media segment file may be communicated to the client for the client to access the media segment.



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))

METHOD OF STREAMING MEDIA TO HETEROGENEOUS CLIENT DEVICES

CROSS-REFERENCE TO RELATED APPLICATION

5 This application claims priority to U.S. Provisional Application No. 61/227683, titled "Method and Apparatus for Seamless Video Streaming to Mobile Devices" and filed on July 22nd, 2009, the entire contents of which are incorporated herein by reference.

10 FIELD OF THE INVENTION

 The present invention relates to digital media streaming, and more particularly, the present invention relates to a system and method for streaming digital media to heterogeneous client devices.

15 BACKGROUND OF THE INVENTION

 Video streaming from a server to a client entity presently faces many engineering challenges. For example, network connected devices come in different screen sizes and CPU power; media optimized for one device (say, a PC) may not be viewable on other devices (say, a cell phone). Not all device
20 supports the same streaming technology (e.g., RTSP streaming) or the same codec (e.g., some supports mp4, some may not).

 When watching a video content on the PC, it is typically not possible or not straight forward to find the same video to watch on a mobile phone. Furthermore, it is even more challenging to watch from the middle of a video
25 content on a phone where the first half was already viewed on a PC.

Traditionally when a video streaming starts from a video server, the server, content, delivery mechanism, etc. remain the same for the duration of the session. This imposes a restriction when multiple devices request access to the same stream within the same streaming session.

5 Accordingly, it would be advantageous to provide a system for video streaming from a server that addresses the above-mentioned challenges.

SUMMARY OF THE INVENTION

In a first aspect, there is provided a method of dynamically preparing a media clip for delivery from a server to a client, the client comprising a client device and a media application for providing media playback on the client device, the method comprising the steps of receiving media attributes from the client, the media attributes comprising one or more playback formats compatible with the client; receiving from the client a request to provide a media segment comprising a selected segment of a selected media content item, the request defining a starting offset of the media segment; identifying a source where a media content file relating to the selected media content item is stored; obtaining the media content file from the source; and obtaining a transcoded media segment file comprising a transcoded segment of the media content file, wherein the transcoded media segment file comprises a playback format selected from the media attributes. The step of obtaining a transcoded media segment file comprises generating a transcoded file comprising the selected segment of the media file. The method may be repeated for one or more additional clients.

At least one of the media attributes is preferably selected from the group consisting of frame rate, resolution, aspect ratio, encoding rate, audio codec, and video codec. The media attributes may further comprise a preferred playback format, and wherein the playback format selected from the media attributes is the preferred playback format.

The media attributes may further comprise a content format; wherein the step of identifying a source comprises identifying one or more sources (which may comprise a single source) where media files related to the selected media content item are stored; and wherein the step of obtaining the media content file from the source comprises selecting a media file from the media files, wherein the selected media file comprises a content format selected from the media attributes. The media attributes may further comprise a preferred content format, and wherein the content format selected from the media attributes is the preferred content format. At least one of the media attributes is preferably selected from the group consisting of content language, content edition, content subtitles, and content censorship.

The step of identifying a source where a media file relating to the media content item is stored preferably comprises the step of identifying the selected media content item from a list of available media content items; wherein the list further comprises at least one source of at least one media file related to the selected media content item.

The source may comprise a cached copy of the media file, and one or more of the media content file and the transcoded media segment file may be cached. The cached copy of the media file may comprises a segment substantially matching the selected segment, and wherein the media file

comprises a playback format selected from the media attributes, wherein the transcoded media segment file is obtained from the cached copy of the media file.

Prior to receiving the request to provide a segment of a selected media content item, media content availability information may be provided to the client, wherein the media content availability information relates to at least a portion of the list of available media content items.

The media attributes preferably further comprise one or more transport formats compatible with the client, the method further comprising the steps of selecting a transport format from the media attributes; preparing the transcoded media segment file for delivery according to the transport format; and communicating a location of the prepared transcoded media segment file to the client. The media attributes may further comprise a preferred transport format, and wherein the transport format selected from the media attributes is the preferred transport format. At least one of the media attributes is preferably selected from the group consisting of hypertext transfer protocol, real time streaming protocol, file transfer protocol, and adaptive hypertext transfer protocol streaming.

The request may be provided after the client is disconnected while streaming a previously generated prepared transcoded media segment, wherein the starting offset is an offset during the previously generated prepared transcoded media segment when the streaming of the previously generated prepared transcoded media segment was disconnected. The previously generated prepared transcoded media segment may have been prepared by the server. Alternatively, the previously generated prepared

transcoded media segment may have been prepared by a different server, and wherein the currently prepared transcoded media segment is generated using a different transport format than a transport format employed when streaming the previously generated prepared transcoded media segment, and wherein the different transport format is selected from the media attributes.

The request may be provided after a previous client is disconnected while streaming a previously generated prepared transcoded media segment, and wherein the starting offset is an offset during the previously generated prepared transcoded media segment when the streaming of the previously generated prepared transcoded media segment was disconnected.

The current client may comprise media attributes that are different from media attributes of the previous client.

The request may comprise metadata associated with semantic content of the selected media content item. The selected media content item is preferably related to any one of a video segment, an audio and video segment, and an image. Statistical information relating to the requests for the delivery of the media content item is preferably recorded.

In another aspect, there is provided a method of dynamically receiving a media clip on a client, the client comprising a client device and a media application for providing media playback on the client device, the method comprising the steps of communicating client media attributes from the client to a server, the media attributes comprising a playback format and a transport format compatible with the client; receiving from the server a list of available media content items; allowing a user to select a media content item from the list; allowing the user to identify a media segment of the media content item

by selecting a starting offset; communicating with the server to request dynamic preparation of the media segment; receiving from the server a location of a prepared transcoded media segment file comprising a transcoded media segment file prepared in a format for delivery to the client, wherein the transcoded media segment file comprises a transcoded segment of a media file related to the media content item, and wherein the transcoded media segment file comprises a playback format selected from the media attributes, and wherein the prepared transcoded media segment file comprises a transport format selected from the media attributes; and obtaining the transcoded media segment file from the location for playback on the client device. The method preferably comprises the step of playing the transcoded media segment file on the client device.

At least one of the media attributes may comprise a playback format attribute selected from the group of consisting of frame rate, resolution, aspect ratio, encoding rate, audio codec, and video codec. At least one of the media attributes is preferably a transfer protocol attribute selected from the group consisting of hypertext transfer protocol, real time streaming protocol, file transfer protocol, and adaptive hypertext transfer protocol streaming.

The media attributes may further comprise a preferred playback format, and wherein the playback format selected from the media attributes is the preferred playback format. Preferably, the media attributes further comprise a preferred transport format, and wherein the transport format selected from the media attributes is the preferred transport format.

The media attributes may further comprise a content format, and wherein the media file related to the media content item comprises a content

format selected from the media attributes. Preferably, the media attributes further comprise a preferred content format, and wherein the content format selected from the media attributes is the preferred content format.

At least one of the media attributes is preferably a content format attribute selected from the group consisting of content language, content edition, content subtitles, and content censorship.

The client device may be selected from the list comprising a cell phone, a smartphone, a tablet, a portable media player, and a computer.

The playing of the transcoded media segment file on the client device may be interrupted, in which case the method may further comprising the steps of recording an offset in the selected media segment file when the interruption occurred; re-establishing communication with the server; and obtaining the transcoded media segment file from the location for playback on the client device.

Alternatively, if the playing of the transcoded media segment file on the client device is interrupted, the method may further comprise the steps of recording an offset in the transcoded media segment file when the interruption occurred; and establishing communication with another server, wherein the media content item is available from the another server; communicating the client media attributes to the another server; identifying a new media segment of the media content item by selecting a starting offset corresponding to the offset in the transcoded media segment file when the interruption occurred; communicating with the another server to request dynamic preparation of the new media segment; receiving from the another server a location of a new prepared transcoded media segment file comprising a new transcoded media

segment file prepared in a format for delivery to the client, wherein the new transcoded media segment file comprises a transcoded segment of the media file related to the media content item, and wherein the new transcoded media segment file comprises a playback format selected from the media attributes, and wherein the new prepared transcoded media segment file comprises a transport format selected from the media attributes; and obtaining the new transcoded media segment file from the location for playback on the client device. The transport format of the new prepared transcoded media segment file may be different from a transport format of the prepared transcoded media segment file that was prepared prior to the interruption.

Prior to the step of receiving from the server the first list of available media content items, the following steps may be performed: communicating with the server to access a user account associated with the user; receiving a second list of media content items comprising media content items previously viewed by the user; wherein the step of allowing the user to select a media content item from the first list further comprises the step of allowing the user to select a media content item from the second list.

If the previously viewed media content item was stopped during playback by the user during a previous viewing session, and wherein an offset at which playback of the media content item was stopped is recorded in the user account; then the step of allowing the user to identify a media segment of the previously viewed media content item by selecting a starting offset may comprise the step of allowing the user to select the offset at which playback of the media content item was stopped. The client may be different than a client used during the previous viewing session.

Prior to the step of receiving from the server the first list of available media content items, the following steps may be performed: communicating with the server to access a user account associated with the user; receiving an additional list of media content items from another user; wherein the step of allowing the user to select a media content item from the first list further comprises the step of allowing the user to select a media content item from the additional list.

The method may further comprise the steps of communicating with the server to access a user account associated with the user; and allowing the user to share the media content item with an additional user.

The media content item is preferably related to any one of a video segment, an audio and video segment, and an image, and the client device is preferably selected from the list comprising a cell phone, a smartphone, a tablet, a portable media player, and a computer. The client may further comprise a user interface in communication with the server, the user interface displaying the list to the user.

A further understanding of the functional and advantageous aspects of the present invention can be realized by reference to the following detailed description and drawings.

5

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the invention will now be described, by way of example only, with reference to the drawings, in which:

Figure 1 is a diagram of a system for streaming video content from one or more media servers to one or more clients;

Figure 2 is a flow chart diagram illustrating an example method for obtaining a video server from a media server;

Figure 3 is a flow chart diagram illustrating a video transmission that is interrupted; and

5 **Figure 4** is a flow chart diagram illustrating a hand-off of a video transmission between more than one client.

DETAILED DESCRIPTION OF THE INVENTION

Without limitation, the majority of the systems described herein are
10 directed to systems and methods for transmitting digital media. As required, embodiments of the present invention are disclosed herein. However, the disclosed embodiments are merely exemplary, and it should be understood that the invention may be embodied in many various and alternative forms.

The Figures are not to scale and some features may be exaggerated or
15 minimized to show details of particular elements while related elements may have been eliminated to prevent obscuring novel aspects. Therefore, specific structural and functional details disclosed herein are not to be interpreted as limiting but merely as a basis for the claims and as a representative basis for teaching one skilled in the art to variously employ the present invention. For
20 purposes of teaching and not limitation, the illustrated embodiments are directed to systems and methods for transmitting digital video clips.

As used herein, the term "about", when used in conjunction with ranges of dimensions or other physical properties or characteristics is meant to cover slight variations that may exist in the upper and lower limits of the ranges of
25 dimensions as to not exclude embodiments where on average most of the

dimensions are satisfied but where statistically dimensions may exist outside this region. For example, in embodiments of the present invention configurations of video streaming systems are given but it will be understood that these are not meant to be limiting.

5 As used herein, the terms "comprises", "comprising", "includes" and "including" are to be construed as being inclusive and open ended, and not exclusive. Specifically, when used in this specification including claims, the terms "comprises", "comprising", "includes" and "including" and variations thereof mean the specified features, steps or components are included. These
10 terms are not to be interpreted to exclude the presence of other features, steps or components.

 As used herein, the coordinating conjunction "and/or" is meant to be a selection between a logical disjunction and a logical conjunction of the adjacent words, phrases, or clauses. Specifically, the phrase "X and/or Y" is
15 meant to be interpreted as "one or both of X and Y" wherein X and Y are any word, phrase, or clause.

 As used herein, the acronym "RTSP" refers to Real Time Streaming Protocol which is a transfer protocol designed for transmission of media; the acronym "HTTP" refers to "Hypertext Transfer Protocol" which is a transfer
20 protocol designed for the transmission of web pages; and the acronym "FTP" refers to File Transfer Protocol which is a transfer protocol designed for the transmission of digital files.

 As used herein, the term "transcoding" refers to the conversion of digital media from one digital encoding to another. If the second encoding is
25 lossy, signal content will be lost in the conversion operation. As an example,

one encoding of digital video may be H.264 and another encoding may be WMV.

Referring to Figure 1, an exemplary and non-limiting system 10 is shown for illustrating selected embodiments in which media content is provided to a client 5. The system includes a media server 16, which provides media content to a client comprising a client application 14 that is executed on client device 12 such as a phone, tablet, personal computer, portable media player, or other computing and/or media presentation device. The client device is operated by user 6, and the client application 14 includes a user interface for interacting with user 6. Client application 14 includes computer readable instructions stored on computer readable memory and may be referred to as a "media application."

Media server 16 accesses media to that is to be streamed and client application 14 is capable of decoding and playing media content (such as a media content stream) on the client device 12. Media server 16 may access media stored in a content management system including media store 20 and media database 22. Further, the media server 16 is connected to a network 24 (preferably the internet), which may connect to additional servers such as a web server 26, an AmazonTM media 28, and a RTSP server 30. The media server 16 has a media transcoder 18 which may be a separate machine from the media server 16, or it may be a program running on the media server 16. The media transcoder 18 may be a device connected over a network.

While the present system is directed to a method of streaming video, the system may be applied to any media content such as audio and images. In Figure 1, a cell phone 12 is shown as running the client application 14.

though other digital devices may serve as the client device 14 such as a smartphone, portable media player, desktop computer, or a tablet computer.

According to selected embodiments, a method of providing media to a client 5 employs a hierarchical system for categorizing or metadata tagging media. At the highest level for example, a media metadata tag may specify a movie title, at an intermediate level, a media tag may specify a viewing language, and at the lowest level, a media tag may specify a specific file representing digital media content. These examples of tag levels are meant to be exemplary and those skilled in the art would appreciate that a hierarchical tagging system may employ different media tags. For example, language may be the highest level in the hierarchy. As disclosed below, such categorization or metadata tagging of media content supports a method in which media content compatible with a client's media attributes can be dynamically provided.

By way of example, a media content item may be defined by a metadata such as a global video identifier (VID) that identifies a video by its semantic content, as opposed to by its digital encoding or formatting. There may be many copies of a video with the same VID, each with a different audio content, screen size, encoding parameters, subtitles, etc. For example, it is possible to have a video dubbed in a different language than the original, the two sharing the same VID. In other words VID is not a sufficient identifier to identify a video in its original source form. One example of a VID is a title of a movie, which is generic and can be applied to describe multiple media files of the same movie, but having different properties such as the presence or absence of subtitles, or the censorship status of the media item.

Considering now the client 5, the capability of the client receiving and playback of media playback and transport formats may be described and provided to the server 16. Such capability, or compatibility, of the client 5 may be referred to as media attributes (MAT). Non-limiting examples of media playback format attributes include the following: frames per second, spoken language, optimized screen size, encoding rate (audio, video), codec used, and many other attributes. Non-limiting examples of media transport format attributes include the following: RTSP, HTTP, adaptive HTTP streaming, FTP, and BitTorrent.

It is to be understood that media attributes may describe the capabilities or compatibility of a client, and that the media attributes may comprise more than one compatible media playback format attribute and/or media transport attribute. For example, a client 5 may be capable of playback of multiple encoding formats, or accepting multiple transport formats. In a preferred embodiment, the client media attributes comprise preferred media attributes, where the preferred media attributes may be selected by a media server for providing media content in a preferred playback or content format.

The media attributes of the client preferable further comprise preferred content format of media to be provided to, and played back, on the device.

Content format media attributes may include, for example, a preferred language, and a preference for the display of subtitles. By providing content format media attributes to a server, a specific media file compatible with the client content media attributes may be selected when a media content item is specified. For example, if a specific movie title is selected as a media content item, the server, armed with the knowledge of the client media content format

attributes, may select an appropriate version (i.e. specific media file) to generate and provide to the client.

It is also to be understood that a specific media file, which may relate to a media content item, may be described in terms of specific media attributes, such as the encoding format of the media file. Accordingly, in one embodiment, a specific media file may be identified the media content item and the media attributes of the file, for example, by a 2-tuple such as {VID, MAT}.

A segment of a specific media file may be identified by the starting time of the media segment, and optionally the ending time of the media segment. In one exemplary embodiment, a video clip identifier (VCID) may be used to identify a subset of a media file, where the VCID comprises a VID, a millisecond offset (Off), a millisecond length (Len), and a MAT. The offset denotes the starting point of the clip; the length indicates the running time of the clip from its start. Accordingly, in this exemplary embodiment, a 3-tuple {VID, Off, Len} may be termed as a Video Clip Identifier or a VCID. The 4-tuple {VID, Off, Len, MAT} may be termed as a Fully Qualified VCID, or FQID.

As described above, media content from a media server 16 may be prepared and delivered to a client in a dynamic fashion. A client may send a request to a server (e.g. media server 16) in which a media content item is selected for delivery. For example, a video clip may be selected by sending its VCID to the server. The media server 16 responds by providing a media file matching the request, where the media file is formatted and transported according to the media attributes of the client, which have been provided to the server.

The media file is preferably dynamically transcoded according the media playback attributes of the client, and transported according to the media transport attributes of the client. For example, if the current delivery mechanism chosen is RTSP, the media server 16 automatically skips to the Offset of the clip and automatically stops the session when it reaches the Length of the clip. If the current delivery mechanism is HTTP, FTP or BitTorrent for example, the media server 16 creates an appropriate video file dynamically with appropriate headers such that the video starts at Offset and ends at the Length. This is described in more detail below.

Referring to Figure 2, a flow chart is provided that illustrates a preferred method of providing a media file from a server 16 to client 5. In step 100, the client obtains, from the server, a list of media content items that are available for delivery. Such media content items may preferably be described by VCID tags that uniquely identifies a video clip, where the server maintains a list comprising the VCID tags and one or more sources where a media file relating to the VCID tag may be obtained. In selected embodiments, such VCID tags can be obtained from an online site, or via email, or similar sources.

In step 102, the user selects a media content item a client 5 to request for delivery from the server 16. The client may display user interpretable information such as the title of a movie, which may be cross-referenced with a VCID or other metadata record that specifically identifies the video content item in a format that can be understood by the server.

In step 106, the client 5 provides the request for the media content item to the media server 16. The address of such a media server can either be

preconfigured at the client, or can be found from a designated website, or from a discovery service. In one embodiment, the client contacts a proxy server (not shown in Figure 1) which functions as a load balancer. The load balancer can redirect this request to any number of media servers 16. In a preferred embodiment, the client employs a set of preconfigured DNS addresses for communicating with the proxy server. If the first preconfigured DNS address does not lead to a successful connection, the client 16 uses the next preconfigured DNS address.

The client 5 preferably issues a request to media server 16 by providing passing the VCID along with the client MAT. As described above the MAT identifies media attributes of the client, which may include playback format attributes, transport format attributes, and media content attributes, and may contain more than one alternative for each attribute.

XML listing 1, as shown below, provides an example of how such a request may be provided to media server 16 as an XML fragment. In the example below, 240x320 pixels is the preferred resolution; 64 is the preferred video bit-rate; 3gp and mp4 are the preferred video containers; h.263 and h.264 the two preferred video codecs; aac, amr-nb, amr-wb, mp3 are the preferred audio codecs; 32 is the preferred audio bit-rate; between 10 and 30 frames per second is the preferred frame-rate range; English is the preferred language; RTSP or HTTP is the preferred delivery protocol; and 60 seconds is the preferred clip size (explained later) when HTTP is the delivery protocol.

XML Listing 1: Example Request

```
<request>
```

```

5      <vcid>
        <vid>http://www.company.com/video/
          34546KBK234.xml</vid>
        <off unit="sec">140</off>
        <len unit="sec">3600</len>
      </vcid>
      <mat>
        <screen>240x320</screen>
        <container>3gp,mp4</container>
10      <videocodec>h.263, h.264</videocodec>
        <videobitrate>64</videobitrate>
        <acodec>aac, amr-nb, amr-wb, mp3</acodec>
        <audiobitrate>32</audiobitrate>
        <framerate>10-30</framerate>
15      <lang>en</lang>
        <delivery>rtsp</delivery>
        <delivery splitsec=60>http</delivery>
      </mat>
    </request>

```

20

In step 108, the media server 16 receives the request sent in step 106, and identifies whether a media segment file having media playback attributes common with the client media attributes is resident in its local cache. Such a case may occur when the server 16 has already responded to the request in the past (for example, the FQID = {VCID, MAT} combination is in its local

25

cache). If such a copy exists, then the server responds directly in step 118, as described further below.

If the media file has not been previously cached, then in step 110, the media server 16 looks up the media content item (e.g. the VID) and identifies a source for a media file related to the media content item. In one embodiment, the source for a media file related to a media content item is specified by an XML description, such as XML Listing 2 as shown below. In this example, the media server 16 has located the XML video description file based on the VID. Here the exemplary VID is a URL, and the XML video description file is located at the URL. Alternatively, the VID may be a portion of a URL such as "34546KBK234.xml" in this example, where the media server 16 may obtain the XML description file by appending a base URL (e.g. "http://www.company.com/video/") with the VID. In another alternative embodiment, the VID may be an arbitrary string and the media server 16 has a list (e.g. a lookup table) describing the relationship between a media content item (e.g. a VID) and a source (e.g. an XML description file).

XML Listing 2: Example Video

```
<video>
  <content>
    <lang>en</lang>
    <title>The Golden Compass</title>
    <url>
      http://www.company.com/content/34546KBK234.flv
    </url>
```

```

    </content>
    <content>
        <lang>fr</lang>
        <title>
5           À la croisée des mondes - La boussole d'or
        </title>
        <url>
            http://www.company.com/content/34546KBK234_fr.flv</u
            rl>
10    </content>
</video>
```

This exemplary video description file indicates the different copies of the film in each language, video quality, or other attribute. Each copy of the video is contained within a <content> tag which indicates the URL of the raw
15 video file, and its metadata that distinguishes it from the remaining <content> tags. For example, an additional <content> tag (not in the above example) may be added which provides a URL to a high-definition version.

It will be appreciated that a XML video description file may point to
20 many versions of the same video file (as shown with many <content> tags). The media server 16 selects the most appropriate <content> for the client 14's request by matching as many MAT tags as possible. In one embodiment, where the client media attributes include a preferred media content format attribute, the media file is selected by matching the media content format of
25 the media file to the preferred client media content format attribute. Those

skilled in the art would appreciate that there are many ways of matching MAT tags to <content> tags and dealing with mismatches. If there is no direct match, the media server 16 may either select the <content> tag that matches as much of the MAT tag as possible, or it may alternatively reply to the user
5 with an error. In this example, the media server 16 selects the English version of the movie and extracts the URL tag (<url>).

In step 112, the media server 16 determines whether the media file identified in step 110 is already present in its own cache. If so, the media server 16 proceeds to step 116 (described later).

10 In step 114, the media server 16 obtains (e.g. downloads) the media file selected in step 110. For example, the media file may be downloaded by following the extracted URL from the <content> section.

In step 116, the media server 16 engages a transcoding entity to transcode the media file into a format that satisfies the client MAT, thus
15 generating a transcoded media segment file. In one embodiment, the media file is transcoded into a format that matches that client media attributes as much as possible (e.g., the exact bit rate asked by a client 14 may not be matched by transcoder). The transcoding entity may be a program running on the media server 16, or it may be a program running on a separate
20 computer. The transcoding entity itself may have a local cache that allows it to bypass the transcoding operation to return a previously transcoded file.

In one example, the transcoding entity is transcoding software running on the media server 16. The transcoding software decodes the original data into an intermediate uncompressed format. This intermediate uncompressed
25 format may be Pulse-code modulation (PCM) for audio data and YUV or RGB

format for digital data. Those skilled in the art will appreciate that there are many ways of storing intermediate uncompressed digital media. From this intermediate step, the uncompressed data may be re-encoded into the desired format.

5 In another example, the transcoding entity may be a digital signal processing (DSP) chip which performs the transcoding operation in hardware. The chip may either be a general purpose DSP chip programmed to perform the desired transcoding operation, or it may be a specialized digital processor which is hard-coded to perform the desired transcoding operation.

10 In a further example, a transcoding entity may additionally provide a number of pre-processing or post-processing filters which may include but are not limited to video de-interlacing, audio re-sampling, frame rate conversion, smoothing, and cutting. These filters improve the quality of the resulting digital media.

15 In step 118, the media server 16 selects a delivery method based on the client media transport format attributes, prepares the transcoded media segment file for delivery, and communicates a location where the prepared transcoded media segment file may be accessed by the client. As a non-limiting example, the delivery method may be HTTP, RTSP, FTP, BitTorrent,
20 or other communication protocols.

If the delivery method chosen is HTTP and the client transport format attributes indicated that the transcoded media segment file needs to be split into segments (as in the splitsec attribute in the <delivery> tag of the MAT), the media server 16 engages a splitter entity to split the transcoded file into
25 segments. Each segment represents a complete file that may be viewed. The

media server 16 then engages a merging entity to merge the segments into a file wrapper containing all of the segments.

If the delivery method chosen is HTTP, the media server 16 then uploads the prepared transcoded media segment file to a suitable media server 16 capable of serving the video file over HTTP. For example, a web media server 16 (such as Apache) or the Amazon™ S3 file service can be used as the video serving media server 16.

The media server 16 then responds to the client's request by sending a response XML file that indicates the location of the prepared transcoded media segment file (i.e. the transcoded media segment file prepared in a format that is suitable for transport). An example response can look like following (XML Listing 3):

XML Listing 3: Example Response

```
15      <response>
          <media>
              <content-url>
                  http://aws.s3.com/video/34546KBK234_01.jig
              </content-url>
              <off>20506</off>
              <len>10240</len>
              <index-url>
                  http://aws.s3.com/video/34546KBK234.xml
              </index-url>
25      </media>
```

</response>

In the above example, the content-url tag points to the transcoded and merged media segment file. The offset and length indicates the portion of the content-url file the client 14 needs to download for the requested media segment. Preferably, the media server 16 also sends an index-url tag which points to an index XML file that provides the index of the individual segments within this merged file. Using the index-url tag, a client 14 can download the index XML file and can determine the offsets and lengths of all segments within the content-url file. This allows the client 14 to skip to any position in the video file by simply reading the file from a certain offset. It should be noted that the entire index-url file can be embedded within the actual content-url file as well as the header, i.e. the index-url does not need to be an external file.

As an alternative example, where the media server 16 selects RTSP as the delivery mechanism, the media server 16 uploads the transcoded media segment file (the split/merge steps will not be required) to an RTSP media server 30. The media server's response in this case may look like the following XML response file (XML Listing 4):

XML Listing 4: Example Response

<response>

<media>

<content-url>

rtsp://somelocation.com/34546KBK234_01.3gp

</content-url>

</media>

</response>

It will be appreciated by those skilled in the art that other transport
5 protocols may be used. When HTTP/FTP/BitTorrent protocols are used, the
client 14 is treating the video file is exactly like any other file and progressively
downloading it. When RTSP protocol is concerned, then the client 14 is no
longer treating the video as a file to be downloaded, rather it uses the RTSP
protocol to download bits of video file in a real-time manner.

10 An adaptive HTTP streaming approach may be employed as the
protocol, wherein the media server 16 prepares several transcoded copies of
the video, split into small segments typically on the order of a few seconds
long (where the transcoded copies are compatible with the client media
attributes). Each copy may be at a different resolution allowing transmission
15 to happen at a number of different bandwidths. If the network transfer speed
is too slow, the client 14 may detect its current data rate and directly access
or download segments from the server without require that the server to re-
transcode the media file. In such a case, the index (e.g. the XML index)
provides information on other available data rate segments on the server, for
20 example by providing with references to other index XML files.

In one embodiment, the client may monitor the effective data rate that it
is receiving while downloading a media file from a server. If the client detects
significant changes in data rate (which could occur, for example, if a cell
phone switches from a 3G wireless coverage to a 2.5G wireless coverage or
25 vice-versa), then in its next request to the server, it can modify the

<audiobitrate> and <videobitrate> values in the MAT specification according to the current data rate. This will allow the server to provide a lower or a higher quality video clip in real-time to the client as per client's specified MAT. This will effectively allow the client to provide an automatic video quality
5 adjustment as per the current network data rate so that an ongoing media delivery session can proceed seamlessly.

In another embodiment of the invention, the server may record delivery and/or playback usage statistics related to a media content item, or related to a specific media file related to a media content item. For example, such
10 statistical data may be employed to determine the relative popularity of selected media content items and/or media files. Preferably, statistical data relating to the delivery and/or playback of media content items enables tracking of the media content files irrespective of their format or location.

Prior art systems such as YouTube™ cater for video clients that must
15 be able to play videos in the format YouTube has created its content. The method described herein allows a client 14 to stream videos from web sites like YouTube™ even if it is not capable of playing Flash™ content, or playing with the video formats the source file is encoded in. Many mobile clients are not capable of long form contents (specially entry level devices). The method
20 described herein allows playing long form content on such devices. Further, it will be appreciated that the resulting video file that the client 14 receives may not be present when the initial request was made. In this sense, system 10 allows for dynamic media transfer. In comparison, when a client 14 accesses a media site such as YouTube™ for video, it receives a static video file which
25 was prepared before the request is made.

In a preferred embodiment of the present invention, a method is provided for enabling the client 14 to resume media playback after a disconnection has occurred. Following the disconnection event, the client attempts to reconnect to media server 16 when disconnected. Preferably, the client 14 need not access the same media server 16 or the same transfer protocol when reconnecting. Accordingly, the source of the video file may be seamlessly switched from the point of the user without requiring user intervention.

It will be appreciated that client media requests such as the example in XML listing 1 may represent either an initial connection request or a reconnection request. In this way, a client 5 may send the same request when it initially connects and every time it gets disconnected, with the requested starting point of the media segment representing the point of disconnection during a previous playback session. The MAT included in each request may specify more than one allowable video format, so the response to each request may provide a different (but allowable) video format. This flexibility is hidden from the user and increases the chance of maintaining a continuous video stream.

Therefore, the system 10 includes a video streaming technique whereby a user's single video streaming session is actually composed of many intermediate sessions with potentially multiple media servers 16 using more than one delivery techniques and video formats. Such intermediate sessions are completely transparent to the end user who only sees one video session. Existing systems typically deal with single session between a client 14 and a media server 16 for a particular video file.

Preferably, the client 5 is only aware of the Internet address of the media server 16 and the media server 16 implements the protocol that the client is following for communicating VCID, MAT, and other information. This protocol is stateless as far as the media server 16 is concerned. The client 5 keeps track of all the possible state information (how long the clip has been played, etc.) so that it can initiate the next request.

This embodiment is illustrated by an exemplary method shown in Figure 3. In step 200, a client 14 detects that it has been disconnected from an ongoing video stream sent via HTTP. Those skilled in the art will appreciate many means of detecting disconnection, such as the media server 16 exceeding a timeout response time. In step 202, the client issues a new request, for example, as in XML Listing 1, wherein the <delivery> tags indicates that the client media transport format attributes include both RTSP and HTTP. In step 204, a media server different from the original media server 16 receives the request, routed as per a load balancing policy. In step 206, this media server chooses RTSP and sends a valid RTSP link to the client. It will be appreciated that RTSP may be chosen because the media server may only support RTSP. Alternatively, if the new media server supports HTTP, then the same transfer protocol may be used as before the client 14 was disconnected in step 200. In the present example, however, the media server selects RTSP streaming, and in step 208, the client 14 receives a response as in XML Listing 4 which provides a RTSP stream.

It should be appreciated that in preferred embodiments, the client request for a media file segment to be delivered by the media server. In its request to the media server, the client provides all the transport formats it

supports via the MAT. The media server may choose any of the transport formats as suggested in the MAT and preferably has no knowledge of previous delivery mechanisms used by the client.

5 Preferably, the media server 16 does not keep track of any session information of the client between successive requests. In other words, each client request is unique and can be handled by any media server and not necessarily an initial media server which serviced a previous request. In the above examples, the media server plays the role of a proxy media server whereby it transcodes or obtains a requested media file, uploads it to a
10 traditional media server (e.g., web server, AmazonTM S3 server, RTSP server, etc.) and then redirects the client 14 by sending an appropriate response.

It will be appreciated that the response does not need to contain URL pointers to media file. The media server may respond with the media file segment along with its response, thus playing the role of a web server and a
15 video streaming server. When a client is disconnected from its file media server from which it was receiving the transcoded media segment file, it communicates with the media server 16 passing in a VCID and MAT to resume the streaming operation. As noted above, since this request is completely an independent request, the media server may choose a delivery
20 mechanism that is different than the previous request.

Users of the system may watch portions of a given media segment using different clients. For example, a user may watch a portion of a media segment on a cell phone and the remainder of the video on a desktop computer. More specifically, a user may stop playback of a delivered media
25 segment during playback on a client device. When the client stops the media

segment, it saves the media content item and the time at which the playback was stopped (e.g. the VCID), and may upload this information to the media server or to a web account associated with the user. Once uploaded, the user may subsequently download the VCID from a different client and initiate the
5 playback of the remainder of the media segment on a different client.

Alternatively, the user may download the VCID to the same client to resume watching at a later time. This is applicable when the client 14 is a web browser and may not store the VCID on the user's hard drive. An example of hand-off between clients is illustrated in Figure 4.

10 Referring to Figure 4, in step 300, a user starts streaming of a full-length movie on a client 5 which comprises a mobile device in this example. In step 302, the user reaches his or her home and wishes to watch the remainder of the movie on his or her desktop computer. The user instructs the client to stop playing the video and to save the video location to his or her
15 web account. Each user may have a web account associated with the system that may be accessed by the media server and by the client. The client generates a VCID which indicates the [VID, Off, Len], i.e. video identifier, time offset, remaining time. The VCID is uploaded to the user's web account by the client 14 which may be accessed by the media server. It will be
20 appreciated that the VCID may be either sent to the media server or to another web server having access to the user's account. In step 306, the user may log onto his or her web account via a web browser on his or her desktop PC. In step 308, the user obtains the uploaded VCID from his or her web account in the form of a hyperlink. In step 310, the browser sends a
25 VCID to a media server 16 along with a MAT for the browser, which indicates

the browser playback capabilities. The browser has different capabilities from a cell phone and may be represented by a different MAT. In step 314, the media server 16 checks whether the transcoded content is available in the desired format as specified by the sent MAT. In step 314, if the transcoded content is not available, the server sends a request to the media transcoder 18 to transcode the content to FLV format with appropriate screen size for web viewing as specified by MAT. In step 316, if the transcoded content is available or was transcoded in step 314 to the desired format (which is FLV in this example), the server redirects the browser to a page with a Flash™ player that starts playing the video from the last position (i.e. offset from the FQID sent in step 304). It will be appreciated that the steps indicated in Figure 4 are merely exemplary and are stated for illustrative purposes.

As noted above, the media server preferably does not keep track of state information as far as streaming sessions are concerned; it is preferably only the client that keeps track of this information. When handing off between different clients, the client playing the video may upload the VCID to the user's account which may be optionally stored on media server or it may be stored on another server. When the user logs into his or her account, the client may obtain all pending sessions by downloading the VCIDs associated with the user account. The user may select a video, which causes the client to upload the associated VCID to the media server to initiate a new video streaming session.

The system includes provisions for broadcasting videos to multiple heterogeneous devices. Users may send other users VCID tags which allow the other users to watch the same video starting from the same offset. In this

sense, multiple users may watch the same video stream broadcast. The user may send VCID by SMS, email, instant messaging applications, web site posts, blog sites, or by other communication means. This enables video broadcasting to multiple recipients possibly using different types of devices (e.g. mobile phones, PC desktops, smart TVs, set-top boxes, etc.). The users receiving the VCID must have a copy of the client application which runs on a client device and allows communication with server and streaming by the methods described above. It will be appreciated that each of the users receiving the same video does not need to use the same protocol or view the video in the same language. Each user initiates a new session, which may use any protocol indicated allowable by the MAT sent by the user's client.

While exemplary embodiments provided above have focused primarily on the delivery of video media segments, it is to be understood that media content items, media clips, and media files as described herein may comprise any form of a digitized media file, including audio and images. In one embodiment, the media file may be a splash or background image for a media presentation device such as a mobile phone. Accordingly, the above embodiments enable a user of a client device to access a wide range of image for display on the client device, even through the images may not be initially formatted for the client device. By providing its media attributes to a server, a client device thus may obtain image files with device-specific custom formatting. This allows, for example, a publisher to publish a single version of an application that automatically brands itself with images in appropriate sizes based on a client device's display size and image capability.

The foregoing description of the preferred embodiments of the invention has been presented to illustrate the principles of the invention and not to limit the invention to the particular embodiment illustrated. It is intended that the scope of the invention be defined by all of the embodiments encompassed within the following claims and their equivalents.

5

THEREFORE WHAT IS CLAIMED IS:

1. A method of dynamically preparing a media clip for delivery from a server to a client, said client comprising a client device and a media application for providing media playback on said client device, said method comprising the steps of:

receiving media attributes from said client, said media attributes comprising one or more playback formats compatible with said client;

receiving from said client a request to provide a media segment comprising a selected segment of a selected media content item, said request defining a starting offset of said media segment;

identifying a source where a media content file relating to said selected media content item is stored;

obtaining said media content file from said source; and

obtaining a transcoded media segment file comprising a transcoded segment of said media content file, wherein said transcoded media segment file comprises a playback format selected from said media attributes.

2. The method according to claim 1 wherein at least one of said media attributes is selected from the group consisting of frame rate, resolution, aspect ratio, encoding rate, audio codec, and video codec.

3. The method according to any one of claims 1 and 2 wherein said media attributes further comprise a preferred playback format, and wherein

said playback format selected from said media attributes is said preferred playback format.

4. The method according to any one of claims 1 to 3 wherein said media attributes further comprise a content format;

wherein said step of identifying a source comprises identifying one or more sources where media files related to said selected media content item are stored; and

wherein said step of obtaining said media content file from said source comprises selecting a media file from said media files, wherein said selected media file comprises a content format selected from said media attributes.

5. The method according to claim 4 wherein said one or more sources comprises a single physical source.

6. The method according to any one of claims 4 and 5 wherein said media attributes further comprise a preferred content format, and wherein said content format selected from said media attributes is said preferred content format.

7. The method according to any one of claims 4 to 6 wherein at least one of said media attributes is selected from the group consisting of content language, content edition, content subtitles, and content censorship.

8. The method according to any one of claims 1 to 7 wherein said step of identifying a source where a media file relating to said media content item is stored comprises the step of:

identifying said selected media content item from a list of available media content items;

wherein said list further comprises at least one source of at least one media file related to said selected media content item.

9. The method according to any one of claims 1 to 8 wherein said source comprises a cached copy of said media file.

10. The method according to claim 9 wherein said cached copy of said media file comprises a segment substantially matching said selected segment, and wherein said media file comprises a playback format selected from said media attributes, wherein said transcoded media segment file is obtained from said cached copy of said media file.

11. The method according to any one of claims 1 to 10 wherein said step of obtaining a transcoded media segment file comprises generating a transcoded file comprising said selected segment of said media file.

12. The method according claim 8 wherein prior to receiving said request to provide a segment of a selected media content item, media content availability information is provided to said client, wherein said media content

availability information relates to at least a portion of said list of available media content items.

13. The method according to any one of claims 1 to 12 wherein one or more of said media content file and said transcoded media segment file are cached.

14. The method according to any one of claims 1 to 13 wherein said request comprises metadata associated with semantic content of said selected media content item.

15. The method according to any one of claims 1 to 14 wherein said media attributes further comprise one or more transport formats compatible with said client, said method further comprising the steps of:

selecting a transport format from said media attributes;

preparing said transcoded media segment file for delivery according to said transport format; and

communicating a location of said prepared transcoded media segment file to said client.

16. The method according to claim 15 wherein said media attributes further comprise a preferred transport format, and wherein said transport format selected from said media attributes is said preferred transport format.

17. The method according to any one of claims 15 and 16 wherein at least one of said media attributes is selected from the group consisting of hypertext transfer protocol, real time streaming protocol, file transfer protocol, and adaptive hypertext transfer protocol streaming.

18. The method according to any one of claims 15 to 17 wherein said prepared transcoded media segment file comprises a currently prepared transcoded media segment, and wherein said request is provided after said client is disconnected while streaming a previously generated prepared transcoded media segment, wherein said starting offset is an offset during said previously generated prepared transcoded media segment when said streaming of said previously generated prepared transcoded media segment was disconnected.

19. The method according to claim 18 wherein said previously generated prepared transcoded media segment was prepared by said server.

20. The method according to claim 19 wherein said previously generated prepared transcoded media segment was prepared by a different server, and wherein said currently prepared transcoded media segment is generated using a different transport format than a transport format employed when streaming said previously generated prepared transcoded media segment, and wherein said different transport format is selected from said media attributes.

21. The method according to any one of claims 1 to 17 wherein said transcoded media segment file comprises a currently prepared transcoded media segment and wherein said client comprises a current client, and wherein said request is provided after a previous client is disconnected while streaming a previously generated prepared transcoded media segment, wherein said starting offset is an offset during said previously generated prepared transcoded media segment when said streaming of said previously generated prepared transcoded media segment was disconnected.

22. The method according to claim 21 wherein said current client comprises media attributes that are different from media attributes of said previous client.

23. The method according to any one of claims 1 to 21 further comprising repeating said steps for one or more additional clients.

24. The method according to any one of claims 1 to 23 wherein said selected media content item is related to any one of a video segment, an audio and video segment, and an image.

25. The method according to any one of claims 1 to 24 wherein statistical information relating to the requests for the delivery of said media content item is recorded.

26. A method of dynamically receiving a media clip on a client, said client comprising a client device and a media application for providing media playback on said client device, said method comprising the steps of:
- communicating client media attributes from said client to a server, said media attributes comprising a playback format and a transport format compatible with said client;
 - receiving from said server a list of available media content items;
 - allowing a user to select a media content item from said list;
 - allowing said user to identify a media segment of said media content item by selecting a starting offset;
 - communicating with said server to request dynamic preparation of said media segment;
 - receiving from said server a location of a prepared transcoded media segment file comprising a transcoded media segment file prepared in a format for delivery to said client, wherein said transcoded media segment file comprises a transcoded segment of a media file related to said media content item, and wherein said transcoded media segment file comprises a playback format selected from said media attributes, and wherein said prepared transcoded media segment file comprises a transport format selected from said media attributes; and
 - obtaining said transcoded media segment file from said location for playback on said client device.
27. The method according to claim 26 wherein at least one of said media attributes is a playback format attribute selected from the group of consisting

of frame rate, resolution, aspect ratio, encoding rate, audio codec, and video codec.

28. The method according to any one of claims 26 and 27 wherein at least one of said media attributes is a transfer protocol attribute selected from the group consisting of hypertext transfer protocol, real time streaming protocol, file transfer protocol, and adaptive hypertext transfer protocol streaming.

29. The method according to any one of claims 26 to 28 wherein said media attributes further comprise a preferred playback format, and wherein said playback format selected from said media attributes is said preferred playback format.

30. The method according to any one of claims 26 to 29 wherein said media attributes further comprise a preferred transport format, and wherein said transport format selected from said media attributes is said preferred transport format.

31. The method according to any one of claims 26 to 30 wherein said media attributes further comprise a content format, and wherein said media file related to said media content item comprises a content format selected from said media attributes.

32. The method according to claim 31 wherein said media attributes further comprise a preferred content format, and wherein said content format selected from said media attributes is said preferred content format.

33. The method according to any one of claims 31 and 32 wherein at least one of said media attributes is a content format attribute selected from the group consisting of content language, content edition, content subtitles, and content censorship.

34. The method according to any one of claims 26 to 33 wherein said client device is selected from the list comprising a cell phone, a smartphone, a tablet, a portable media player, and a computer.

35. The method according to any one of claims 26 to 34 further comprising the step of playing said transcoded media segment file on said client device.

36. The method according to claim 35 wherein said playing of said transcoded media segment file on said client device is interrupted, said method further comprising the steps of:

- recording an offset in said selected media segment file when said interruption occurred;

- re-establishing communication with said server; and

- obtaining said transcoded media segment file from said location for playback on said client device.

37. The method according to claim 35 wherein said playing of said transcoded media segment file on said client device is interrupted, said method further comprising the steps of:

- recording an offset in said transcoded media segment file when said interruption occurred; and

- establishing communication with another server, wherein said media content item is available from said another server;

- communicating said client media attributes to said another server;

- identifying a new media segment of said media content item by selecting a starting offset corresponding to said offset in said transcoded media segment file when said interruption occurred;

- communicating with said another server to request dynamic preparation of said new media segment;

- receiving from said another server a location of a new prepared transcoded media segment file comprising a new transcoded media segment file prepared in a format for delivery to said client, wherein said new transcoded media segment file comprises a transcoded segment of said media file related to said media content item, and wherein said new transcoded media segment file comprises a playback format selected from said media attributes, and wherein said new prepared transcoded media segment file comprises a transport format selected from said media attributes;
- and

- obtaining said new transcoded media segment file from said location for playback on said client device.

38. The method according to claim 37 wherein said transport format of said new prepared transcoded media segment file is different from a transport format of said prepared transcoded media segment file that was prepared prior to said interruption.

39. The method according to any one of claims 26 to 35 wherein said list comprises a first list, and wherein prior to said step of receiving from said server said first list of available media content items, the following steps are performed:

communicating with said server to access a user account associated with said user;

receiving a second list of media content items comprising media content items previously viewed by said user;

wherein said step of allowing said user to select a media content item from said first list further comprises the step of allowing said user to select a media content item from said second list.

40. The method according to claim 39 wherein said previously viewed media content item was stopped during playback by said user during a previous viewing session, wherein an offset at which playback of said media content item was stopped is recorded in said user account; and

wherein said step of allowing said user to identify a media segment of said previously viewed media content item by selecting a starting offset comprises the step of:

allowing said user to select said offset at which playback of said media content item was stopped.

41. The method according to claim 40 wherein said client is different than a client used during said previous viewing session.

42. The method according to any one of claims 26 to 38 wherein said list comprises a first list, and wherein prior to said step of receiving from said server said first list of available media content items, the following steps are performed:

communicating with said server to access a user account associated with said user;

receiving an additional list of media content items from another user;

wherein said step of allowing said user to select a media content item from said first list further comprises the step of allowing said user to select a media content item from said additional list.

43. The method according to any one of claims 26 to 42 further comprising the steps of:

communicating with said server to access a user account associated with said user; and

allowing said user to share said media content item with an additional user.

44. The method according to any one of claims 26 to 43 wherein said media content item is related to any one of a video segment, an audio and video segment, and an image.

45. The method according to any one of claims 26 to 44 wherein said client device is selected from the list comprising a cell phone, a smartphone, a tablet, a portable media player, and a computer.

46. The method according to any one of claims 26 to 45 wherein said client further comprises a user interface in communication with said server, said user interface displaying said list to said user.

1/4

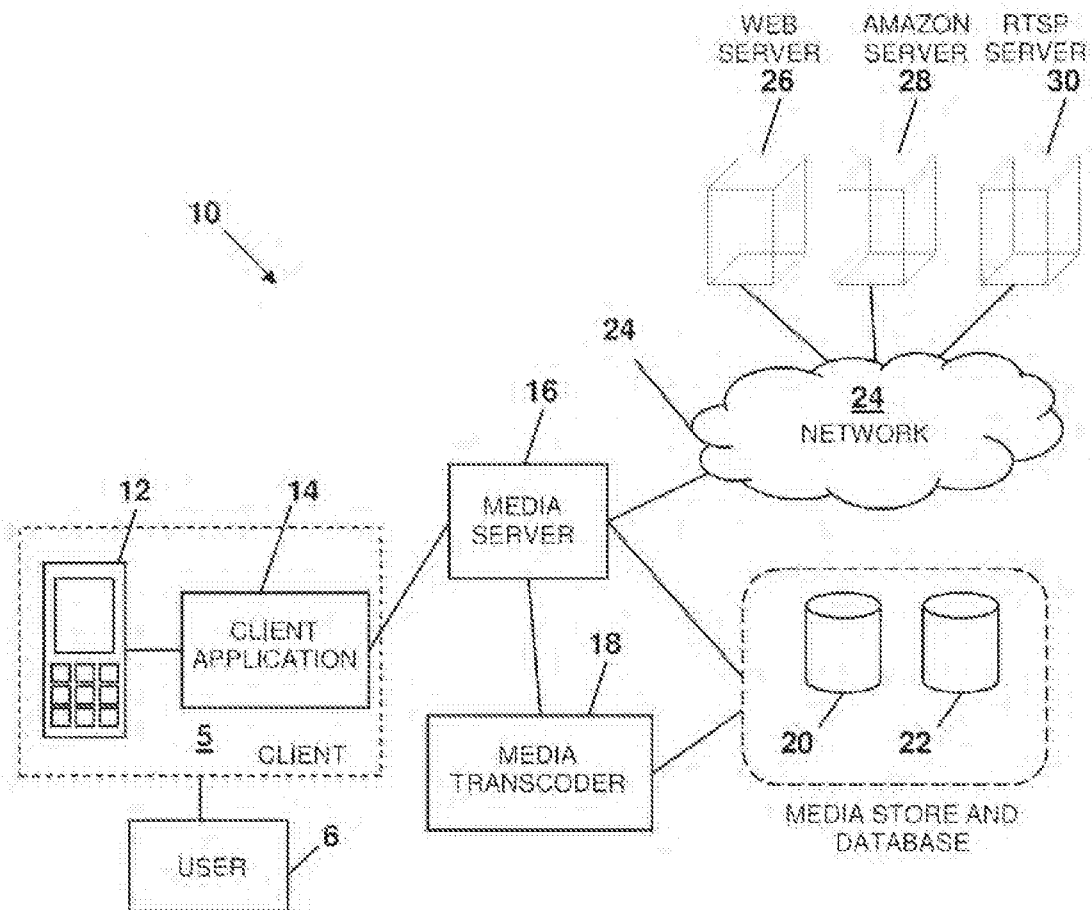


Figure 1

2/4

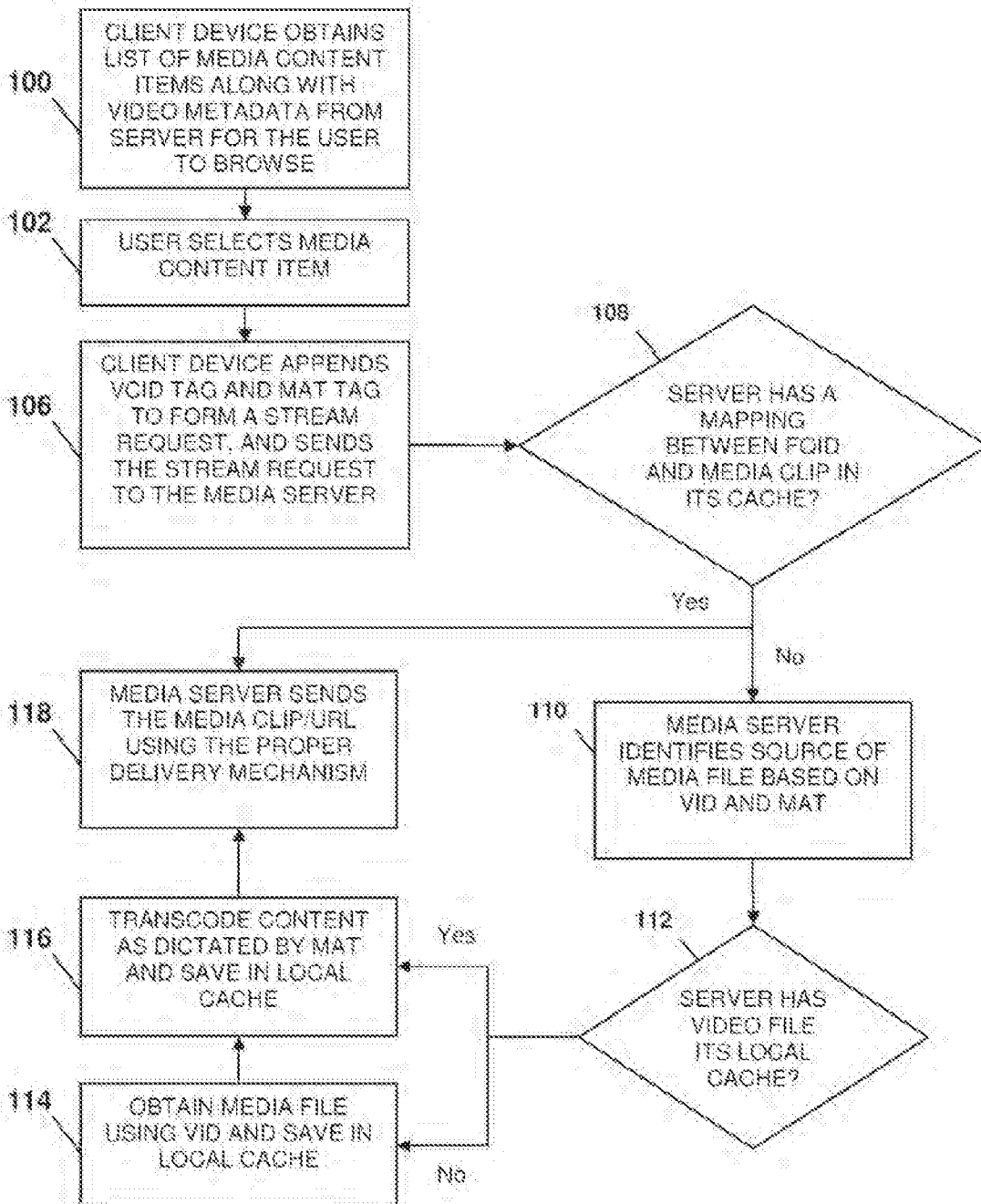


Figure 2

3/4

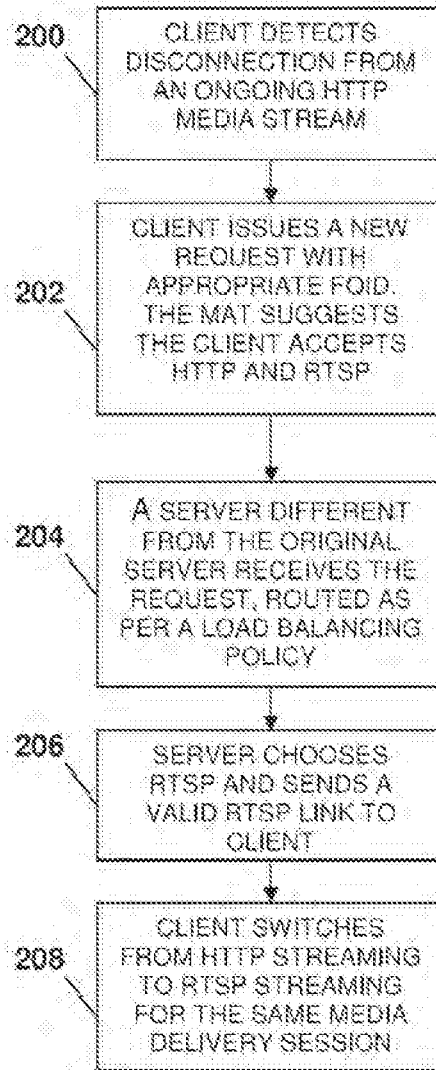


Figure 3

4/4

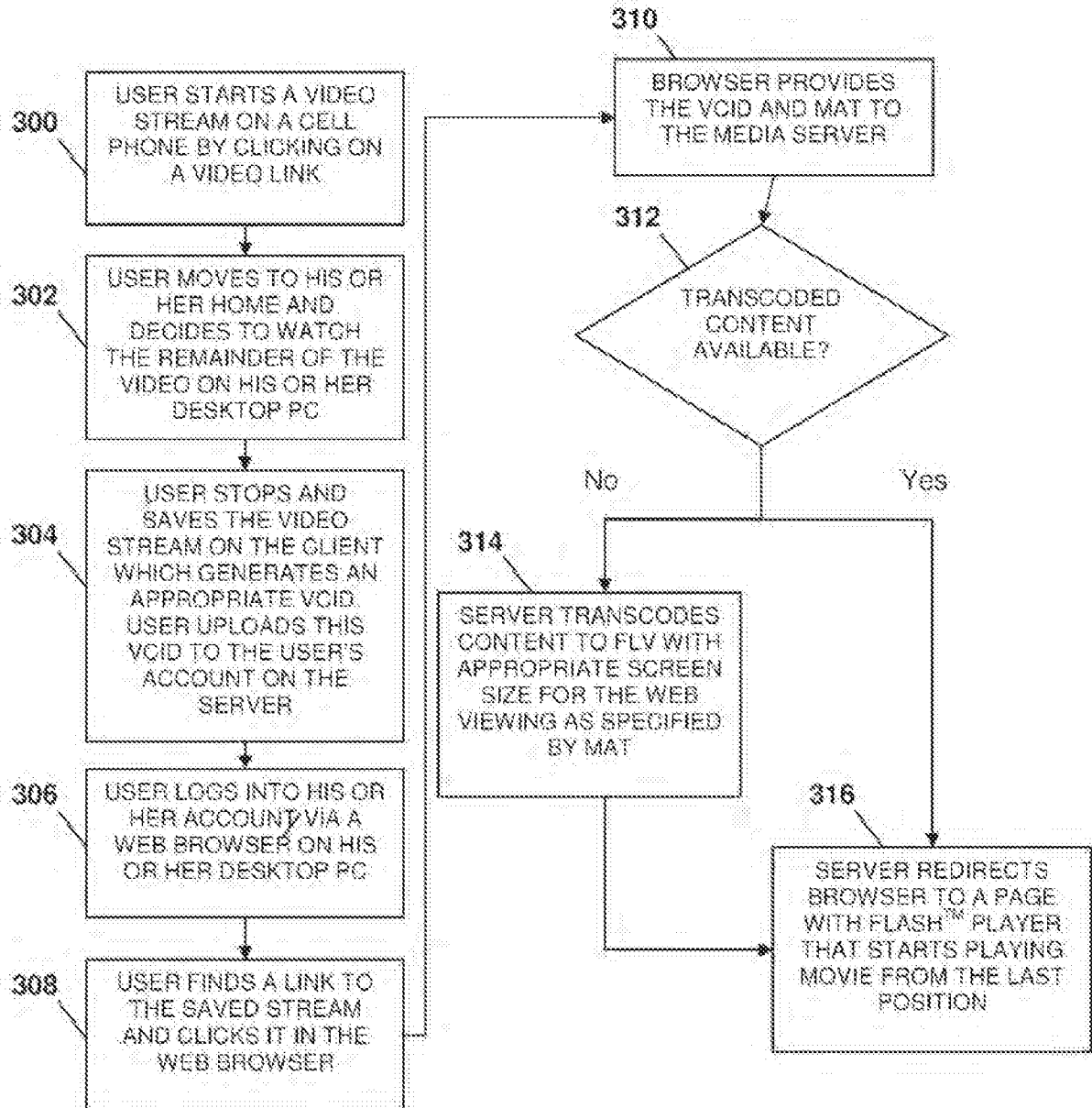


Figure 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2010/001129

A. CLASSIFICATION OF SUBJECT MATTER IPC: H04N 7/173 (2006.01) 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) IPC: H04N (2006.01)		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used) Canadian Patent Database/Intellect, EPOQUE, Total Patent, Keywords: (video, movie, server, client, heterogeneous, media, capabilities, attributes, parameters, descriptors, transcode, encode, decode, streaming, format, display size, language, subtitles, phone, cell, mobile, pda)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevance to claim No.
A	US 7504968 (Wee et al.), 17 March 2009 (17-03-2009) *See entire document	1 to 46
A	US 2004/010614 (Mukherjee et al.), 15 January 2004 (15-01-2004) *See entire document	1 to 46
A	US 7650621 (Thomas et al.), 19 January 2010 (19-01-2010) *See entire document	1 to 46
A	US 2005097615 (Moradil et al.), 05 May 2005 (05-05-2005) *See entire document	1 to 46
F, A	WO 2010/074830 (Kajos et al.), 01 July 2010 (01-07-2010) *See entire document	1 to 46
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: (X) document defining the general state of the art which is not considered to be of particular relevance (Y) earlier application or patent but published on or after the international filing date (Z) document which contains doubts on priority claims or in which is cited to establish the publication date of another citation or other special reason (to be specified) (G) document relating to an oral disclosure, test, exhibition or other means (P) document published prior to the international filing date but later than the priority date claimed	(1) 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 (2) document of particular relevance: the claimed invention cannot be considered novel or cannot be considered inventive on the prior art step when the document is taken alone (3) document of particular relevance: the claimed invention cannot be considered inventive on the prior art step when the document is considered with two or more other cited documents, such combination being obvious to a person skilled in the art (4) document mentioned in the same patent family	
Date of the actual completion of the international search 5 November 2010 (05-11-2010)		Date of mailing of the international search report 18 November 2010 (18-11-2010)
Name and mailing address of the ISA/CA Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PC1 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476		Authorized officer Lambros Pezoulas (819) 934-2677

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2010/001129

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US7504953B2	17 March 2009 (17-03-2009)	EP1714495A1 KR20060120267A KR100812909B1 US2005183118A1 WO2005081536A1	25 October 2006 (25-10-2006) 24 November 2006 (24-11-2006) 11 March 2008 (11-03-2008) 19 August 2005 (19-08-2005) 01 September 2005 (01-09-2005)
US2004010514A1	15 January 2004 (15-01-2004)	JP2004046679A US7133925B2 US2004139212A1 WO2005055610A1	12 February 2004 (12-02-2004) 07 November 2006 (07-11-2006) 15 July 2004 (15-07-2004) 15 June 2005 (15-06-2005)
US7850621B2	19 January 2010 (19-01-2010)	AR030869A1 AT406618T AU1155302A AU1157002A AU1307102A AU2002211553B2 BR0114571A BR0114805A CA2425472A1 CA2425479A1 CA2425482A1 CN1470129A CN100459808C CN1470130A CN100579203C CN1475081A CN1540990A CN100397399C CN1568104A CN101707693A CN101715109A DE60135567D1 DK1327209T3 EP1327209A2 EP1327209B1 EP1327257A2 EP1327358A2 EP1947858A1 EP1986436A1 ES2312475T3 HK1058079A1 JP2004511037T JP2004511974T JP2004511975T JP2005102240A JP2007181240A JP2007295563A JP2009213183A JP2010022054A JP2010187410A JP2010187411A KR20050128845A KR20070114325A KR20080050533A KR20090007637A KR20090031796A MXPA03003138A PT1327209E TW543327B TW545060B	03 September 2003 (03-09-2003) 15 September 2008 (15-09-2008) 22 April 2002 (22-04-2002) 22 April 2002 (22-04-2002) 22 April 2002 (22-04-2002) 19 July 2007 (19-07-2007) 06 January 2004 (06-01-2004) 01 July 2003 (01-07-2003) 18 April 2002 (18-04-2002) 18 April 2002 (18-04-2002) 18 April 2002 (18-04-2002) 21 January 2004 (21-01-2004) 04 February 2009 (04-02-2009) 21 January 2004 (21-01-2004) 05 January 2010 (05-01-2010) 11 February 2004 (11-02-2004) 27 October 2004 (27-10-2004) 25 June 2008 (25-06-2008) 14 September 2009 (14-09-2009) 12 May 2010 (12-05-2010) 26 May 2010 (26-05-2010) 09 October 2008 (09-10-2008) 08 December 2008 (08-12-2008) 16 July 2003 (16-07-2003) 27 August 2008 (27-08-2008) 16 July 2003 (16-07-2003) 16 July 2003 (16-07-2003) 16 July 2003 (16-07-2003) 23 July 2008 (23-07-2008) 29 October 2008 (29-10-2008) 01 March 2009 (01-03-2009) 24 April 2009 (24-04-2009) 15 April 2004 (15-04-2004) 15 April 2004 (15-04-2004) 15 April 2004 (15-04-2004) 14 April 2005 (14-04-2005) 12 July 2007 (12-07-2007) 08 November 2007 (08-11-2007) 17 September 2009 (17-09-2009) 28 January 2010 (28-01-2010) 26 August 2010 (26-08-2010) 26 August 2010 (26-08-2010) 08 December 2006 (08-12-2006) 30 November 2007 (30-11-2007) 05 June 2008 (05-06-2008) 19 January 2009 (19-01-2009) 27 March 2008 (27-03-2008) 14 July 2003 (14-07-2003) 28 November 2008 (28-11-2008) 21 July 2003 (21-07-2003) 01 August 2003 (01-08-2003)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2010/001129

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US2002088621A1		16 May 2002 (16-05-2002)	
		US2002042920A1	11 April 2002 (11-04-2002)
		US2002042921A1	11 April 2002 (11-04-2002)
		US2008209485A1	28 August 2008 (28-08-2008)
		US2009138922A1	29 May 2009 (29-05-2009)
		US2010166025A1	22 July 2010 (22-07-2010)
		WO20231701A2	18 April 2002 (18-04-2002)
		WO20231701A3	09 January 2003 (09-01-2003)
		WO20231701A3	06 May 2004 (06-05-2004)
		WO20232139A2	13 April 2002 (13-04-2002)
		WO20232139A3	30 January 2003 (30-01-2003)
		WO20232140A2	18 April 2002 (18-04-2002)
		WO20232140A3	24 October 2002 (24-10-2002)
US2005087615A1	05 May 2005 (05-05-2005)	CA2590874A1	22 June 2006 (22-06-2006)
		EP1825680A2	29 August 2007 (29-08-2007)
		US2005033855A1	10 February 2005 (10-02-2005)
		US2005165848A1	28 July 2005 (28-07-2005)
		US2008189752A1	07 August 2008 (07-08-2008)
		WO2006066077A2	22 June 2006 (22-06-2006)
		WO2006066077A9	20 July 2006 (20-07-2006)
		WO2006066077A3	15 March 2007 (15-03-2007)
WO2010074830A2	01 July 2010 (01-07-2010)	US2010181716A1	24 June 2010 (24-06-2010)
		WO2010074830A3	18 August 2010 (18-08-2010)

(43) International Publication Date
7 April 2011 (07.04.2011)(10) International Publication Number
WO 2011/039617 A1(51) International Patent Classification:
H04L 29/06 (2006.01)

(21) International Application Number:

PCT/IB2010/002475

(22) International Filing Date:

29 September 2010 (29.09.2010)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/246,846 29 September 2009 (29.09.2009) US

(71) Applicant (for all designated States except US): **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).

(72) Inventor; and

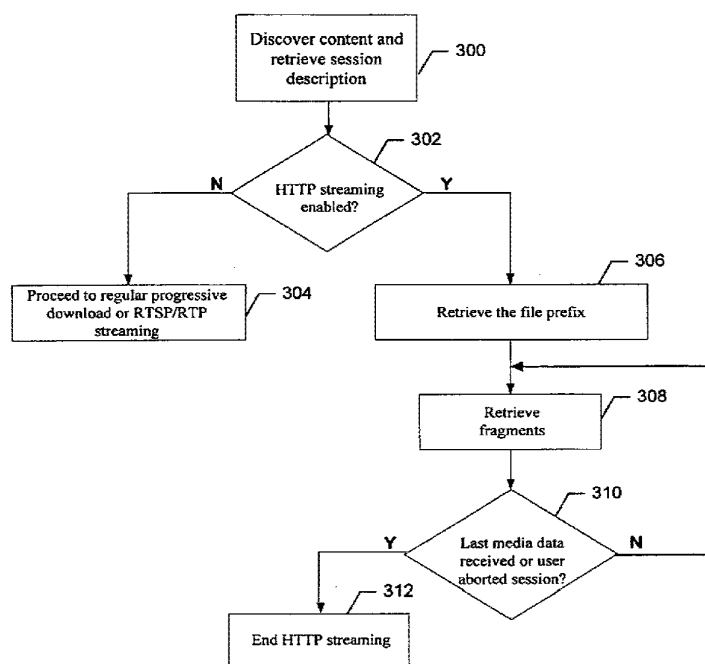
(75) Inventor/Applicant (for US only): **BOUAZIZI, Imed** [TN/FI]; Insinoorinkatu 22 A 37, FIN-33720 Tampere (FI).(74) Common Representative: **NOKIA CORPORATION**;
C/o Nokia Inc., Attn: IPR Docketing, 6021 Connection Drive, MS 2-5-520, Irving, Texas 75039 (US).

(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, SE, SI, SK,

[Continued on next page]

(54) Title: SYSTEM, METHOD AND APPARATUS FOR DYNAMIC MEDIA FILE STREAMING

**FIG. 3**

(57) Abstract: A method, apparatus, system and computer program product are provided to facilitate dynamic media streaming. In this regard, a server may receive a transport protocol for at least the portion of the media file indicating that at least a portion of the media file is to be streamed to a content consumption device requesting the media file. In response, a plurality of fragments to be transmitted to the content consumption device may be determined based upon the transfer protocol request and the plurality of fragments may then be transmitted to the content consumption device in response to the transfer protocol request.

WO 2011/039617 A1



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))*

SYSTEM, METHOD AND APPARATUS FOR DYNAMIC MEDIA FILE STREAMING

TECHNOLOGICAL FIELD

5 Embodiments of the present invention relate generally to communications technology and, more particularly, relate to a system, method and apparatus for media file streaming.

BACKGROUND

10 Networking technologies and the computing devices that make use of the networking technology have evolved in such a manner as to continue to facilitate the ease of information transfer and convenience to users. In this regard, the expansion of networks and the evolution of network computing devices have provided sufficient processing power, storage space and network bandwidth to enable the transfer and playback of increasingly complex digital media files. Accordingly, internet television, video sharing and the like are gaining in popularity.

15 In order to facilitate the transfer and playback of digital media files, digital media files may be streamed from a server to a content consumption device, such as a computing device. Media file streaming may be accommodated by fragmenting a media file into a plurality of fragments. The content consumption device may request a fragment of a media file and the server may then transmit the fragment to the content consumption device in response to the request.
20 Following the transmission and receipt of one fragment, the client consumption device may request another fragment from the server. This process may be repeated with the media file being transmitted from the server to the content consumption device one fragment at a time.

BRIEF SUMMARY OF SOME EXAMPLES OF THE INVENTION

25 A system, method, apparatus and computer program product are therefore provided according to one example embodiment for facilitating the streaming of media files using a transport protocol, such as hypertext transport protocol (HTTP). In one embodiment, a client consumption device may issue a transport protocol request that causes a server to transmit multiple fragments of a media file. By transmitting multiple fragments in response to a transport
30 protocol request, the media file may be streamed in a more efficient manner with a lower request overhead and a greater percentage of the data transmitted between the server and the content consumption device being dedicated to the content itself. In order to further increase the efficiency of the streaming process, a transport protocol request may be issued that identifies two or more different types of tracks such that a plurality of fragments of each of the two or more
35 different kinds of tracks may be provided in response to the transport protocol request. Additionally, the transmission of multiple fragments in response to a single transport protocol

request may permit the bitrate to be more smoothly adapted over the course of streaming a media file in correspondence with the quality of the network conditions, such that any changes in the bitrate are disconcerting to the viewer. Moreover, by permitting the transport protocol request to identify a portion of a media file to be downloaded without having to specify a particular
5 fragment to be downloaded, the server need not initially provide an index file of available fragments and, as such, may generate the fragments dynamically in response to the transport protocol request.

In one embodiment, a method is provided, which includes receiving a transport protocol request for at least the portion of the media file indicating that at least a portion of the media file
10 is to be streamed to a content consumption device requesting the media file. The method of this embodiment also determines a plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request and then provides for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

In another embodiment a computer program product is provided. The computer program
15 product includes at least one computer-readable storage medium having computer-readable program instructions stored therein. The computer-readable program instructions of this embodiment may include program instructions which when executed by a server are configured to receive a transport protocol request for at least the portion of the media file indicating that at least a portion of the media file is to be streamed to a content consumption device requesting the media
20 file. The computer-readable program instructions of this embodiment, when executed, are also configured to cause the server to determine a plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request and to then provide for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

In a further embodiment, an apparatus is provided. The apparatus of this embodiment
25 includes at least one processor and at least one memory storing computer program code. The at least one memory storing computer program code is configured to, working with the at least one processor, cause the apparatus to receive a transport protocol request for at least the portion of the media file indicating that at least a portion of the media file is to be streamed to a content
30 consumption device requesting the media file. The apparatus of this embodiment is also caused to determine a plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request and to then provide for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

In yet another embodiment, an apparatus is provided that includes means for receiving a
35 transport protocol request for at least the portion of the media file indicating that at least a portion of the media file is to be streamed to a content consumption device requesting the media file. The apparatus of this embodiment also includes means for determining a plurality of fragments to be

transmitted to the content consumption device based upon the transfer protocol request and means for providing for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

5 In another embodiment, a method is provided, which includes directing issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server. A method of this embodiment also comprises receiving a plurality of fragments from the server in response to the transfer protocol request.

10 In another embodiment a computer program product is provided. The computer program product includes at least one computer-readable storage medium having computer-readable program instructions stored therein. The computer-readable program instructions of this embodiment may include program instructions which, when executed by a processor, are configured to direct issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server. The computer-readable program instructions of this embodiment are also configured, when executed by a
15 processor, to cause a communication device to receive a plurality of fragments from the server in response to the transfer protocol request.

In a further embodiment, an apparatus is provided. The apparatus of this embodiment includes at least one processor and at least one memory storing computer program code. The at least one memory storing computer program code is configured to, working with the at least one
20 processor, cause the apparatus to direct issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server. The apparatus of this embodiment is also caused to receive a plurality of fragments from the server in response to the transfer protocol request.

25 In yet another embodiment, an apparatus is provided that includes means for directing issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server. The apparatus of this embodiment also includes means for receiving a plurality of fragments from the server in response to the transfer protocol request.

30 In a further embodiment, a system is provided that includes a server and a content consumption device configured to support media file streaming from the server to the content consumption device. In this embodiment, the content consumption device is configured to direct issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server and to receive a plurality of fragments from the server in response to the transfer protocol request. In addition, the server in this
35 embodiment is configured to receive a transport protocol for at least the portion of the media file indicating that at least a portion of the media file is to be streamed to a content consumption device requesting the media file. The server of this embodiment is also configured to determine a

plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request and to then provide for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

5 The above summary is provided merely for purposes of summarizing some example
embodiments of the invention so as to provide a basic understanding of some aspects of the
invention. Accordingly, it will be appreciated that the above described example embodiments are
merely examples and should not be construed to narrow the scope or spirit of the invention in any
way. It will be appreciated that the scope of the invention encompasses many potential
embodiments, some of which will be further described below, in addition to those here
10 summarized.

BRIEF DESCRIPTION OF THE DRAWING(S)

Having thus described embodiments of the invention in general terms, reference will now
be made to the accompanying drawings, which are not necessarily drawn to scale, and wherein:

15 FIG. 1 illustrates a system for facilitating streaming of media files using a transfer
protocol according to an exemplary embodiment of the present invention;

FIG. 2 is a schematic block diagram of a mobile terminal according to an exemplary
embodiment of the present invention;

20 FIG. 3 is a flowchart according to an exemplary method for facilitating streaming of
media files in accordance with one example embodiment of the invention;

FIG. 4 illustrates ISOFF-compliant structured media content that may be formatted in
accordance with one example embodiment of the invention; and

25 FIGs. 5-8 illustrate additional flowcharts according to exemplary methods for facilitating
streaming of media files according to an exemplary embodiment of the present invention.

DETAILED DESCRIPTION

Some embodiments of the present invention will now be described more fully hereinafter
with reference to the accompanying drawings, in which some, but not all embodiments of the
invention are shown. Indeed, it should be appreciated that many other potential embodiments of
30 the invention, in addition to those illustrated and described herein, may be embodied in many
different forms. Embodiments of the present invention should not be construed as limited to the
embodiments set forth herein; rather, the embodiments set forth herein are provided so that this
disclosure will satisfy applicable legal requirements. Like reference numerals refer to like
elements throughout.

35 As used herein, the term 'circuitry' refers to (a) hardware-only circuit implementations
(e.g., implementations in analog circuitry and/or digital circuitry); (b) combinations of circuits
and computer program product(s) comprising software and/or firmware instructions stored on one

or more computer readable memories that work together to cause an apparatus to perform one or more functions described herein; and (c) circuits, such as, for example, a microprocessor(s) or a portion of a microprocessor(s), that require software or firmware for operation even if the software or firmware is not physically present. This definition of 'circuitry' applies to all uses of this term herein, including in any claims. As a further example, as used herein, the term 'circuitry' also includes an implementation comprising one or more processors and/or portion(s) thereof and accompanying software and/or firmware. As another example, the term 'circuitry' as used herein also includes, for example, a baseband integrated circuit or applications processor integrated circuit for a mobile phone or a similar integrated circuit in a server, a cellular network device, other network device, and/or other computing device.

Some multimedia servers use real-time transport protocol (RTP) over user datagram protocol (UDP) for media streaming (together with the real-time streaming protocol (RTSP) as a control protocol). In this regard, UDP provides basic transport functionality such as application addressing and corruption detection. RTP complements UDP with media transport relevant functionality, such as loss detection, packet re-ordering, synchronization, statistical data collection, and session participant identification. However, RTP over UDP (RTP/UDP) does not provide built-in congestion control and/or error correction functionality. RTP/UDP may gather sufficient information for implementing congestion control and/or error correction functionality on a need basis at an application level. However, with the rising popularity of mobile and internet video, it is becoming even a greater priority to maintain good network behavior through appropriate rate control mechanisms. Additionally, in RTP/UDP-based streaming applications, the sender and/or receiver of the streaming media, if not appropriately configured, may fail to traverse network address translation (NAT) device(s) and/or a firewall(s) positioned in the streaming path between the sender and receiver.

Hypertext transfer protocol (HTTP) streaming media delivery is replacing RTSP/RTP-based streaming in some applications. Even though HTTP was not designed for the delivery of real-time media, HTTP provides benefits that make it attractive for streaming media content delivery applications. In this regard, HTTP is relatively easy to setup and deploy. Additionally, HTTP media delivery overcomes NAT and firewall traversal issues that hinder use of RTSP/RTP-based streaming. Further, HTTP is widely deployed with a very robust infrastructure (such as HTTP caches) that enables efficient data distribution.

HTTP is an application-level transfer protocol that was developed for the delivery of hyperlinked text documents. The usages of the protocol have evolved further to cover the delivery of files of different types of files and data. HTTP uses a request/response model to retrieve remote resources from a web server. The client establishes a transport protocol connection (e.g., a transmission control protocol (TCP) connection) with the server, typically on port 80, and exchanges HTTP messages with the server. An HTTP message may comprise a header portion

and an optional body portion. The header portion may comprise a sequence of HTTP header fields, each ending with a carriage return/line feed character. The HTTP message header may be separated from the HTTP message body by an empty line. The HTTP header fields are provided in a human-readable textual format. A Uniform Resource Locator (URL) or Uniform Resource Identifier (URI) is used to identify and locate the resource that is to be retrieved by the HTTP client. A URI comprises a protocol identifier, a fully qualified domain name, a path to the requested resource, and optionally a query string.

However, media streaming over HTTP is still not as efficient as desired due, for example, to the relatively high request overhead. In this regard, prior to streaming media, a content consumption device usually fetches an index file from the server. The index file includes a listing of one or more identifiers associated with all content fragments of a media file.

Since the content fragments are relatively short, such as two to four seconds, in comparison to the overall length of the media file, the index file may be sizeable and its transmission to the content consumption device may consume more bandwidth than is desired, particularly since the index file may generally be updated and re-transmitted on a fairly frequent basis, e.g., for live content. Moreover, the index file is generally provided in the form of an extensible markup language (XML) document that must first be parsed and processed by the content consumption device. The parsing of the index file may also consume more time and resources than is desired and may also undesirably increase the overhead associated with media file streaming. An additional issue associated with the advanced provision of an index file listing identifiers associated with all content fragments is that the provision of the index file presupposes that the media file has already been formatted into fragments. With respect to some types of media files, however, such as media files that capture live or ongoing events, such as a sporting event or a live musical performance, or on-demand media files, the media file may not be formatted into fragments in advance and, indeed, may not be in existence at the time at which the index file is compiled.

Following the transmission of the index file from the server to the content consumption device, the content consumption device may issue a request for a particular fragment of the media file. The server may then provide the requested fragment of the media file and only after receipt of the fragment by the content consumption device may the content consumption device request another fragment of the media file. This process by which the media file is requested and delivered on a fragment-by-fragment basis may then be repeated numerous times until the desired portion of the media file has been transmitted to the content consumption device. The repeated request for a fragment of the media file undesirously increases the overhead associated with the media file streaming. Additionally, since the content consumption device must await receipt of the immediately preceding fragment prior to issuing a request for the next fragment over the same TCP connection, a delay may be injected into the media file streaming process in instances in

which one of the fragments is misdirected or otherwise fails to be timely received by the content consumption device. The delay in receiving a data fragment may result in an interruption in the playback of the media content at the content consumption device.

5 The fragments of a media file may be available from a server at a plurality of different bitrates. Based upon the network conditions, a content consumption device may therefore tailor its request, for example, by requesting a fragment from a low bitrate track of the media content in order to adjust to a decrease in available network bandwidth. If an improvement in network bandwidth or network conditions is detected, the content consumption device may request a fragment from a high bitrate track of the media content. In this instance, a content consumption
10 device that detects relatively poor network conditions, such as relatively low bandwidth, may request a fragment having a lower bitrate and therefore a reduced quality, while a content consumption device that detects improved network conditions, such as greater bandwidth, may request a fragment having a greater bitrate and therefore improved quality. The determination of the network conditions is generally performed by the content consumption device on a request-
15 by-request basis. Since each request is for a single fragment, however, the content consumption device may not be provided with a sufficient duration in which to gauge the current network conditions. Moreover, since the bitrate and, therefore, the quality of the fragments may change with every fragment, such as every two to four seconds, the resulting changes in the quality of the display may be disconcerting for the viewer.

20 A media file may include a plurality of different tracks. As described above, a media file may include different tracks of the same type, such as video tracks, but at different bitrates, thereby facilitating bitrate adaptation in response to the current network conditions. By way of example of the different types of tracks, the media file may include video tracks, audio tracks and subtitle tracks for the same media file. Conventionally, a content consumption device would
25 separately have to issue a request for a fragment from each different type of track, such as one request for an audio track fragment, another request for a video track fragment, and yet another request for a subtitle track fragment. Since each request must be issued in a serial fashion, the necessity to issue separate requests for the different types of tracks further increases the request overhead and reduces the efficiency of the media file streaming process.

30 Accordingly, embodiments of the invention provide systems, methods, apparatuses, and computer program products that may address some of the deficiencies of conventional media streaming techniques. For example, in order to reduce request overhead and increase the efficiency of a media file streaming process as well as to provide other advantages, a system, apparatus, method and computer program product are provided according to embodiments of the
35 present invention that permit multiple fragments to be transmitted from a server to a content consumption device in response to a single transfer protocol request. In this regard, FIG. 1 illustrates a block diagram of a system 100 for facilitating streaming of media files according to

an exemplary embodiment of the present invention. As used herein, “exemplary” merely means an example and as such represents one example embodiment for the invention and should not be construed to narrow the scope or spirit of the invention in any way. It will be appreciated that the scope of the invention encompasses many potential embodiments in addition to those illustrated and described herein. As such, while FIG. 1 illustrates one example of a configuration of a system for facilitating streaming of media files, numerous other configurations may also be used to implement embodiments of the present invention. Further, it should be appreciated that hypertext transfer protocol (HTTP) is used as an example of an application layer transfer protocol that may be used for streaming of media files in accordance with some embodiments of the invention. Other embodiments of the invention are configured to stream media files using other application layer transfer protocols in addition to or in lieu of HTTP.

FIG. 1 illustrates a block diagram of a system 100 for streaming media files using an application layer transfer protocol, such as hypertext transfer protocol (HTTP), according to an example embodiment of the present invention. In an example embodiment, the system 100 comprises a content consumption device 102 and a server 104. The content consumption device 102 and the server 104 are configured to communicate over a network 108. The network 108, for example, comprises one or more wireline networks, one or more wireless networks, or some combination thereof. The network 108 may comprise a public land mobile network (PLMN) operated by a network operator. In this regard, the network 108, for example, comprises an operator network providing cellular network access, such as in accordance with 3GPP standards. The network 108 may additionally or alternatively comprise the internet.

The content consumption device 102 may comprise any device configured to access content from a server 104 over the network 108. For example, the content consumption device 102 comprises a server, a desktop computer, a laptop computer, a mobile terminal, a mobile computer, a mobile phone, a mobile communication device, a game device, a digital camera/camcorder, an audio/video player, a television device, a radio receiver, a digital video recorder, a positioning device, any combination thereof, and/or the like.

In an example embodiment, the content consumption device 102 is embodied as a mobile terminal, such as that illustrated in FIG. 2. In this regard, FIG. 2 illustrates a block diagram of a mobile terminal 10 representative of one embodiment of a content consumption device 102 in accordance with embodiments of the present invention. It should be understood, however, that the mobile terminal 10 illustrated and hereinafter described is merely illustrative of one type of content consumption device 102 that may implement and/or benefit from embodiments of the present invention and, therefore, should not be taken to limit the scope of the present invention. While several embodiments of the electronic device are illustrated and will be hereinafter described for purposes of example, other types of electronic devices, such as mobile telephones, mobile computers, portable digital assistants (PDAs), pagers, laptop computers, desktop

computers, gaming devices, televisions, and other types of electronic systems, may employ embodiments of the present invention.

As shown, the mobile terminal 10 may include an antenna 12 (or multiple antennas 12) in communication with a transmitter 14 and a receiver 16. The mobile terminal may also include a processor 20 that provides signals to and receives signals from the transmitter and receiver, respectively. These signals may include signaling information in accordance with an air interface standard of an applicable cellular system, and/or any number of different wireline or wireless networking techniques, comprising but not limited to Wireless-Fidelity (Wi-Fi), wireless local access network (WLAN) techniques such as Institute of Electrical and Electronics Engineers (IEEE) 802.11, and/or the like. In addition, these signals may include speech data, user generated data, user requested data, and/or the like. In this regard, the mobile terminal may be capable of operating with one or more air interface standards, communication protocols, modulation types, access types, and/or the like. More particularly, the mobile terminal may be capable of operating in accordance with various first generation (1G), second generation (2G), 2.5G, third-generation (3G) communication protocols, fourth-generation (4G) communication protocols, and/or the like. For example, the mobile terminal may be capable of operating in accordance with 2G wireless communication protocols IS-136 (Time Division Multiple Access (TDMA)), Global System for Mobile communications (GSM), IS-95 (Code Division Multiple Access (CDMA)), and/or the like. Also, for example, the mobile terminal may be capable of operating in accordance with 2.5G wireless communication protocols General Packet Radio Service (GPRS), Enhanced Data GSM Environment (EDGE), and/or the like. Further, for example, the mobile terminal may be capable of operating in accordance with 3G wireless communication protocols such as Universal Mobile Telecommunications System (UMTS), Code Division Multiple Access 2000 (CDMA2000), Wideband Code Division Multiple Access (WCDMA), Time Division-Synchronous Code Division Multiple Access (TD-SCDMA), and/or the like. The mobile terminal may be additionally capable of operating in accordance with 3.9G wireless communication protocols such as Long Term Evolution (LTE) or Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and/or the like. Additionally, for example, the mobile terminal may be capable of operating in accordance with fourth-generation (4G) wireless communication protocols and/or the like as well as similar wireless communication protocols that may be developed in the future.

Some Narrow-band Advanced Mobile Phone System (NAMPS), as well as Total Access Communication System (TACS), mobile terminals may also benefit from embodiments of this invention, as should dual or higher mode phones (e.g., digital/analog or TDMA/CDMA/analog phones). Additionally, the mobile terminal 10 may be capable of operating according to Wireless Fidelity (Wi-Fi) or Worldwide Interoperability for Microwave Access (WiMAX) protocols.

It is understood that the processor 20 may comprise circuitry for implementing audio/video and logic functions of the mobile terminal 10. For example, the processor 20 may,

for example, be embodied as various means including circuitry, one or more microprocessors with accompanying digital signal processor(s), one or more processor(s) without an accompanying digital signal processor, one or more coprocessors, one or more multi-core processors, one or more controllers, processing circuitry, one or more computers, various other processing elements including integrated circuits, such as, for example, an ASIC (application specific integrated circuit) or FPGA (field programmable gate array), or some combination thereof. The processor may additionally comprise an internal voice coder (VC) 20a, an internal data modem (DM) 20b, and/or the like. Further, the processor may comprise functionality to operate one or more software programs, which may be stored in memory. For example, the processor may be capable of operating a connectivity program, such as a web browser. The connectivity program may allow the mobile terminal 10 to transmit and receive web content, such as location-based content, according to a protocol, such as Wireless Application Protocol (WAP), hypertext transfer protocol (HTTP), and/or the like. The mobile terminal 10 may be capable of using a Transmission Control Protocol/Internet Protocol (TCP/IP) to transmit and receive web content across the internet or other networks.

The mobile terminal 10 may also comprise a user interface including, for example, an earphone or speaker 24, a ringer 22, a microphone 26, a display 28, a user input interface, and/or the like, which may be operationally coupled to the processor 20. Although not shown, the mobile terminal may comprise a battery for powering various circuits related to the mobile terminal, for example, a circuit to provide mechanical vibration as a detectable output. The user input interface may comprise devices allowing the mobile terminal to receive data, such as a keypad 30, a touch display (not shown), a joystick (not shown), and/or other input device. In embodiments including a keypad, the keypad may comprise numeric (0-9) and related keys (#, *), and/or other keys for operating the mobile terminal.

As shown in FIG. 2, the mobile terminal 10 may also include one or more means for sharing and/or obtaining data. For example, the mobile terminal may comprise a short-range radio frequency (RF) transceiver and/or interrogator 64 so data may be shared with and/or obtained from electronic devices in accordance with RF techniques. The mobile terminal may comprise other short-range transceivers, such as, for example, an infrared (IR) transceiver 66, a Bluetooth™ (BT) transceiver 68 operating using Bluetooth™ brand wireless technology developed by the Bluetooth™ Special Interest Group, a wireless universal serial bus (USB) transceiver 70 and/or the like. The Bluetooth™ transceiver 68 may be capable of operating according to ultra-low power Bluetooth™ technology (e.g., Wibree™) radio standards. In this regard, the mobile terminal 10 and, in particular, the short-range transceiver may be capable of transmitting data to and/or receiving data from electronic devices within a proximity of the mobile terminal, such as within 10 meters, for example. Although not shown, the mobile terminal may be capable of transmitting and/or receiving data from electronic devices according to various

wireless networking techniques, including Wireless Fidelity (Wi-Fi), WLAN techniques such as IEEE 802.11 techniques, and/or the like.

The mobile terminal 10 may comprise memory, such as a subscriber identity module (SIM) 38, a removable user identity module (R-UIM), and/or the like, which may store
5 information elements related to a mobile subscriber. In addition to the SIM, the mobile terminal may comprise other removable and/or fixed memory. The mobile terminal 10 may include volatile memory 40 and/or non-volatile memory 42. For example, volatile memory 40 may include Random Access Memory (RAM) including dynamic and/or static RAM, on-chip or off-chip cache memory, and/or the like. Non-volatile memory 42, which may be embedded and/or
10 removable, may include, for example, read-only memory, flash memory, magnetic storage devices (e.g., hard disks, floppy disk drives, magnetic tape, etc.), optical disc drives and/or media, non-volatile random access memory (NVRAM), and/or the like. Like volatile memory 40 non-volatile memory 42 may include a cache area for temporary storage of data. The memories may store one or more software programs, instructions, pieces of information, data, and/or the like
15 which may be used by the mobile terminal for performing functions of the mobile terminal. For example, the memories may comprise an identifier, such as an international mobile equipment identification (IMEI) code, capable of uniquely identifying the mobile terminal 10.

Referring again to FIG. 1, in an example embodiment, the client consumption device 102
comprises various means, such as a processor 110, a memory 112, a communication interface
20 114, a user interface 116, and a media playback circuitry 118, for performing the various functions herein described. The various means of the client device 102 as described herein comprise, for example, hardware elements, e.g., a suitably programmed processor, combinational logic circuit, and/or the like, a computer program product comprising computer-readable program instructions, e.g., software and/or firmware, stored on a computer-readable medium, e.g. memory
25 112. The program instructions are executable by a processing device, e.g., the processor 110.

The processor 110 may, for example, be embodied as various means including one or more microprocessors with accompanying digital signal processor(s), one or more processor(s) without an accompanying digital signal processor, one or more coprocessors, one or more controllers, processing circuitry, one or more computers, various other processing elements
30 including integrated circuits such as, for example, an application specific integrated circuit (ASIC) or a field programmable gate array (FPGA), or some combination thereof. Accordingly, although illustrated in FIG. 1 as a single processor, in some embodiments the processor 110 comprises a plurality of processors. The plurality of processors may be in operative communication with each other and may be collectively configured to perform one or more
35 functionalities of the content consumption device 102 as described herein. In embodiments wherein the content consumption device 102 is embodied as a mobile terminal 10, the processor 110 may be embodied as or otherwise comprise the processor 20. In an example embodiment, the

processor 110 is configured to execute instructions stored in the memory 112 or otherwise accessible to the processor 110. The instructions, when executed by the processor 110, cause the content consumption device 102 to perform one or more of the functionalities of the content consumption device 102 as described herein. As such, whether configured by hardware or software operations, or by a combination thereof, the processor 110 may represent an entity capable of performing operations according to embodiments of the present invention when configured accordingly. For example, when the processor 110 is embodied as an ASIC, FPGA or the like, the processor 110 may comprise specifically configured hardware for conducting one or more operations described herein. Alternatively, as another example, when the processor 110 is embodied as an executor of instructions, the instructions may specifically configure the processor 110, which may otherwise be a general purpose processing element if not for the specific configuration provided by the instructions, to perform one or more operations described herein.

The memory 112 may include, for example, volatile and/or non-volatile memory. Although illustrated in FIG. 1 as a single memory, the memory 112 may comprise a plurality of memories. The memory 112 may comprise volatile memory, non-volatile memory, or some combination thereof. In this regard, the memory 112 may comprise, for example, a hard disk, random access memory, cache memory, flash memory, a compact disc read only memory (CD-ROM), digital versatile disc read only memory (DVD-ROM), an optical disc, circuitry configured to store information, or some combination thereof. In embodiments in which the content consumption device 102 is embodied as a mobile terminal, the memory 112 may be embodied as or otherwise comprise the volatile memory 40 and/or non-volatile memory 42. The memory 112 may be configured to store information, data, applications, instructions, or the like for enabling the content consumption device 102 to carry out various functions in accordance with embodiments of the present invention. For example, in at least some embodiments, the memory 112 is configured to buffer input data for processing by the processor 110. Additionally or alternatively, in at least some embodiments, the memory 112 is configured to store program instructions for execution by the processor 110. The memory 112 may store information in the form of static and/or dynamic information. This stored information may be stored and/or used by the media playback unit 118 during the course of performing its functionalities.

The communication interface 114 may be embodied as any device or means embodied in hardware, a computer program product comprising computer readable program instructions stored on a computer readable medium (e.g., the memory 112) and executed by a processing device (e.g., the processor 110), or a combination thereof that is configured to receive and/or transmit data from/to a remote device over the network 108. In at least one embodiment, the communication interface 114 is at least partially embodied as or otherwise controlled by the processor 110. In this regard, the communication interface 114 may be in communication with the processor 110, such as via a bus. The communication interface 114 may include, for example, an antenna, a

transmitter, a receiver, a transceiver and/or supporting hardware or software for enabling communications with other entities of the system 100, e.g., antenna 12, transmitter 14 and/or receiver 16 of mobile terminal 10 of Fig. 2. The communication interface 114 may be configured to receive and/or transmit data using any protocol that may be used for communications between computing devices of the system 100. The communication interface 114 may additionally be in communication with the memory 112, user interface 116, and/or media playback circuitry 118, such as via a bus.

The user interface 116 may be in communication with the processor 110 to receive an indication of a user input and/or to provide an audible, visual, mechanical, or other output to a user. As such, the user interface 116 may include, for example, a keyboard, a mouse, a joystick, a display, a touch screen display, a microphone, a speaker, and/or other input/output mechanisms, e.g., earphone or speaker 24, microphone 26, display 28 and/or keypad 30 of mobile terminal 10 of Fig. 2. The user interface 116 may provide an interface allowing a user to select a media file and/or media tracks thereof to be streamed from the server 104 to the content consumption device 102 for playback on the content consumption device 102. In this regard, video from a media file may be displayed on a display of the user interface 116 and audio from a media file may be audibilized over a speaker of the user interface 116. The user interface 116 may be in communication with the memory 112, communication interface 114, and/or media playback circuitry 118, such as via a bus.

The media playback circuitry 118 may be embodied as various means, such as hardware, a computer program product comprising computer readable program instructions stored on a computer readable medium (e.g., the memory 112) and executed by a processing device (e.g., the processor 110), or some combination thereof and, in one embodiment, is embodied as or otherwise controlled by the processor 110. In embodiments where the media playback circuitry 118 is embodied separately from the processor 110, the media playback circuitry 118 may be in communication with the processor 110. The media playback circuitry 118 may further be in communication with the memory 112, communication interface 114, and/or user interface 116, such as via a bus.

The server 104 may comprise one or more computing devices configured to provide media files to a content consumption device 102. In at least one embodiment, the server 104 comprises one or more servers, such as an HTTP server, a dynamic streaming server, a content provider server, a web server, a network server or the like. While the server 104 may be the source of the media files, the server may also be an intermediary for receiving the media files from one or more content sources and for providing the media files to the content consumption device 102. In an exemplary embodiment, the server 104 includes various means, such as a processor 120, memory 122, communication interface 124, user interface 126, and media streaming circuitry 128 for performing the various functions herein described. These means of

the server 104 as described herein may be embodied as, for example, hardware elements (e.g., a suitably programmed processor, combinational logic circuit, and/or the like), a computer program product comprising computer-readable program instructions (e.g., software or firmware) stored on a computer-readable medium (e.g. memory 122) that is executable by a suitably configured
5 processing device (e.g., the processor 120), or some combination thereof.

The processor 120 may, for example, be embodied as various means including one or more microprocessors with accompanying digital signal processor(s), one or more processor(s) without an accompanying digital signal processor, one or more coprocessors, one or more controllers, processing circuitry, one or more computers, various other processing elements
10 including integrated circuits such as, for example, an ASIC (application specific integrated circuit) or FPGA (field programmable gate array), or some combination thereof. Accordingly, although illustrated in FIG. 1 as a single processor, in some embodiments the processor 120 comprises a plurality of processors. The plurality of processors may be embodied on a single computing device or distributed across a plurality of computing devices. The plurality of
15 processors may be in operative communication with each other and may be collectively configured to perform one or more functionalities of the server 104 as described herein. In an exemplary embodiment, the processor 120 is configured to execute instructions stored in the memory 122 or otherwise accessible to the processor 120. These instructions, when executed by the processor 120, may cause the server 104 to perform one or more of the functionalities of
20 server 104 as described herein. As such, whether configured by hardware or software methods, or by a combination thereof, the processor 120 may represent an entity capable of performing operations according to embodiments of the present invention when configured accordingly. Thus, for example, when the processor 120 is embodied as an ASIC, FPGA or the like, the processor 120 may comprise specifically configured hardware for conducting one or more
25 operations described herein. Alternatively, as another example, when the processor 120 is embodied as an executor of instructions, the instructions may specifically configure the processor 120, which may otherwise be a general purpose processing element if not for the specific configuration provided by the instructions, to perform one or more algorithms and operations described herein.

The memory 122 may include, for example, volatile and/or non-volatile memory. Although illustrated in FIG. 1 as a single memory, the memory 122 may comprise a plurality of memories, which may be embodied on a single computing device or distributed across a plurality of computing devices. The memory 122 may comprise volatile memory, non-volatile memory, or some combination thereof. In this regard, the memory 122 may comprise, for example, a hard
35 disk, random access memory, cache memory, flash memory, a compact disc read only memory (CD-ROM), digital versatile disc read only memory (DVD-ROM), an optical disc, circuitry configured to store information, or some combination thereof. The memory 122 may be

configured to store information, data, applications, instructions, or the like for enabling the media content source 104 to carry out various functions in accordance with embodiments of the present invention. For example, in at least some embodiments, the memory 122 is configured to buffer input data for processing by the processor 120. Additionally or alternatively, in at least some
5 embodiments, the memory 122 is configured to store program instructions for execution by the processor 120. The memory 122 may store information in the form of static and/or dynamic information. This stored information may be stored and/or used by the media streaming unit 128 during the course of performing its functionalities.

The communication interface 124 may be embodied as any device or means embodied in
10 hardware, a computer program product comprising computer readable program instructions stored on a computer readable medium, e.g., the memory 122, and executed by a processing device, e.g., the processor 120, or a combination thereof that is configured to receive and/or transmit data from/to a remote device over the network 108. In at least one embodiment, the communication interface 124 is at least partially embodied as or otherwise controlled by the processor 120. In
15 this regard, the communication interface 124 may be in communication with the processor 120, such as via a bus. The communication interface 124 may include, for example, an antenna, a transmitter, a receiver, a transceiver and/or supporting hardware or software for enabling communications with other entities of the system 100. The communication interface 124 may be configured to receive and/or transmit data using any protocol that may be used for
20 communications between computing devices of the system 100. The communication interface 124 may additionally be in communication with the memory 122, user interface 126, and/or media streaming circuitry 128, such as via a bus.

The user interface 126 may be in communication with the processor 120 to receive an indication of a user input and/or to provide an audible, visual, mechanical, or other output to the
25 user. As such, the user interface 126 may include, for example, a keyboard, a mouse, a joystick, a display, a touch screen display, a microphone, a speaker, and/or other input/output mechanisms. In embodiments wherein the server 104 is embodied as one or more servers, the user interface 126 may be limited, or even eliminated. The user interface 126 may be in communication with the memory 122, communication interface 124, and/or media streaming circuitry 128, such as via a
30 bus.

The media streaming circuitry 128 may be embodied as various means, such as hardware, a computer program product comprising computer readable program instructions stored on a computer readable medium, e.g., the memory 122, and executed by a processing device, e.g., the processor 120, or some combination thereof and, in one embodiment, is embodied as or
35 otherwise controlled by the processor 120. In embodiments wherein the media streaming circuitry 128 is embodied separately from the processor 120, the media streaming circuitry 128 may be in communication with the processor 120. The media streaming circuitry 128 may further

be in communication with the memory 122, communication interface 124, and/or user interface 126, such as via a bus.

In an example embodiment, the media playback circuitry 118 is configured to send a transfer protocol request for a media file to the server 104. In an example embodiment, the requested media file comprises a media file compliant with the ISO base media file format. Examples of an ISO base media file format comprise a 3GP media file and a moving picture experts group 4 (MPEG-4) Part 14 (MP4) file. The request, for example, is sent in response to a user input or request received via the user interface 116.

The transfer protocol request comprises an indication that the media file is to be streamed to the content consumption device 102. In an example embodiment, the transfer protocol request comprises an HTTP GET request. The HTTP GET request comprises a header field including a token indicating that the media file is to be streamed. For example, the header field may comprise the "Expect" header field and include a token, e.g. "http-streaming", defined to indicate that the server 104 is required to support HTTP streaming of media files, such as 3GPP based HTTP streaming of a 3GP media file. In another example, the header field comprises the "Pragma" header field and includes a token, e.g. "http-streaming", defined to indicate that the server 104 is being queried for support of HTTP streaming of the requested media file.

In an example embodiment, the media streaming circuitry 128 is configured to receive a transfer protocol request sent by the content consumption device 102. If the transfer protocol request includes an indication that the requested media file is to be streamed to the content consumption device 102 and the server 104 is not configured to stream a media file, the media streaming circuitry 128 is configured to send an error message to the content consumption device 102. If the server 104 is configured to stream a media file then the media streaming circuitry 128 is configured to include support in a reply message sent to the content consumption device 102. Such support may, for example, be indicated as part of the Pragma header field of a HTTP reply message.

In an example embodiment, the media streaming circuitry 128 is further configured to, in response to receipt of a transfer protocol request for a media file, access the requested media file from the memory 122 or other memory accessible to the server 104. The media streaming circuitry 128 is configured to extract at least a portion of information associated with media data in the media file. In an example embodiment, the extracted portion of information(s) comprises a portion(s) of the metadata associated with media data in the media file. For example, the extracted portion of metadata comprises general information about the content of the media file, e.g., the type(s) of media data and/or the different tracks in the media file. The extracted portion(s) of metadata comprises, for example, only information useful to the content consumption device to select at least one track from the media file.

The metadata associated with the media file, for example, may be structured in accordance with the International Organization for Standardization (ISO)-base media File Format (ISOFF) as outlined in the table below:

0	1	2	3	4	5	Description
typ						File type and compatibility
oov						Container for all metadata
	vhd					Movie header, overall declarations
	rak					Container for an individual trak or stream
		khd				Track header, overall information in a track
		ref				Track reference container
		dia				Container for media information in a track
			dhd			Media header, overall information about the media
			dlr			Handler, declares the media type
			inf			Media information container
				mhd		Video media header, overall information for video track only
				mhd		Sound media header, overall information for sound track only
				tbl		Sample table box, container for the time/space map
					tsd	Sample descriptions for the initialization of the media decoder
					tts	Decoding time-to-sample

					tts	Composition time-to-sample
					tsc	Sample-to-chunk
					tsz	Sample sizes
					tco	Chunk offset to beginning of the file
					tss	sync sample table for Random Access Points
oof						Movie fragment
	fhd					Movie fragment header
	raf					Track fragment
		fhd				Track fragment header
		run				Track fragment run
fra						Movie fragment random access
	fra					Track fragment random access
	fro					Movie fragment random access offset
dat						Media data container

The ISOFF is designed in an object-oriented manner. In this regard, an ISOFF compliant file is composed of a set of boxes that may be inherited and extended through the definition of new boxes. All information in an ISOFF-compliant file must be contained in a box. A box may itself contain other boxes. Each box is identified by a unique type, which is typically defined as a 4 byte type (4 characters). Each box also indicates the length of the box, including the header of the box. These two fields are defined by the "Box" box, which is inherited by all ISOFF boxes.

As illustrated above, ISOFF-compliant data comprises a hierarchy of a plurality of levels of metadata. Each level comprises one or more sublevels including more specific metadata related to the parent level. For example, a first level, "L0" comprises the metadata categories ftyp, moov, moof, mfra, and mdat. Ftyp and mdat may not include any sublevels. The second level, "L1" of moov may comprise, for example, mvhd and trak. The third level, "L2" of trak, for example, comprises tkhd, tref, and mdia. The fourth level, "L3" of mdia may, for example, comprise mdhd, hdlr, and minf. The fifth level, "L4" of minf may comprise vmhd, smhd, and stbl. The sixth level, "L5," of stbl may, for example, comprise stsd, stts, ctts, stsc, stsz, stco, and stss. Accordingly, the above table represents a nested hierarchy of blocks of metadata, wherein sublevels of a block of metadata are illustrated in rows below the row including the corresponding parent metadata block and in columns to the right of the column including the corresponding parent block of metadata. Thus, all sublevels of blocks of metadata of the moov block are shown in the rows of the table below the row including the moov block until reaching the row including the "moof" block, e.g., another parent block of metadata, which is on the same level as the moov block. Similarly, all sublevels of blocks of metadata of the stbl block are shown in the rows of the table below the row including the stbl block, until reaching the row including the moof block, which is the first block at a level the same as or higher than the stbl block.

Prior to streaming content from the server 104 to the content consumption device 102 as generally represented by FIG. 3 which depicts operations that may be performed at and/or by the content consumption device 102, such as by and/or under the control the media playback circuitry 118, the content consumption device 102 may request and be provided with session or content description information. See operation 300 of FIG. 3. In one embodiment, the server 104 is configured to provide content via several different streaming protocols, such as RTSP/RTP streaming, HTTP streaming or the like. Thus, the content consumption device 102 may be provided with information regarding the delivery methods supported by the server 104, e.g., the various streaming options, in session description information provided by the server 104 to the content consumption device 104 using session description protocol (SDP), via a web portal that is provided by the server 104 and accessed by the content consumption device 102 to discover the available content or the like.

In instances in which the server 104 provides information regarding the possible delivery options using SDP, the SDP may be extended with elements that indicate whether or not the content is available via HTTP streaming. The SDP may also contain a description of all of available media components of the content. The content consumption device 102 may then utilize the description of the available media components in selecting the appropriate content components to be streamed, such as via HTTP. By way of example, the content components that may be streamed over HTTP may include different types of tracks, such as video tracks, audio tracks, sub-title tracks and/or the like. For a particular type of track, such as a video or an audio

track, the available media content may also include content at different bit rates in order to support bit rate adaptation as described below based upon, for example, the network conditions at the time of streaming. By way of example, the SDP extensions that may be utilized by a server 104 and a corresponding content consumption device 102 to describe the media components available via HTTP streaming for an example media file may be as follows:

```

5      v=0
      o=- 950814089 950814089 IN IP4 144.132.134.67
      s=Example of content available via HTTP Streaming
      e=user@nokia.com
10     c=IN IP4 0.0.0.0
      b=AS:600
          b=TIAS:500000
      t=0 0
      a=maxprate:100
15     a=range:npt=0-7200
          a=httpstreaming: 1
      a=control: http://www.nokia.com/content/example.3gp

          m=audio 0 HTTP/TCP *
20     b=AS:80
          b=TIAS:64000
      a=maxprate:20
          a=control: trackID=1
      a=lang:en
25
          m=audio 0 HTTP/TCP *
      b=AS:80
          b=TIAS:64000
      a=maxprate:20
30     a=control: trackID=2
      a=lang:fr

          m=audio 0 HTTP/TCP *
      b=AS:80
35     b=TIAS:64000
      a=maxprate:20

```

a=control: trackID=3
a=lang:de

m=audio 0 HTTP/TCP *
5 b=AS:80
 b=TIAS:64000
a=maxprate:20
 a=control: trackID=4
 a=lang:ar
10
m=video 0 HTTP/TCP *
b=AS:520
 b=TIAS:450000
a=maxprate:80
15 ***a=control: trackID=5***

m=video 0 HTTP/TCP *
b=AS:400
 b=TIAS:350000
20 a=maxprate:80
 a=control: trackID=6

m=video 0 HTTP/TCP *
b=AS:300
25 b=TIAS:250000
a=maxprate:80
 a=control: trackID=7

30 As illustrated in the foregoing example, the available audio components include audio components in different languages, while the video components that are available include video components that are coded at different bit rates. In the foregoing example, the “a=control” attribute at the session level may include an identifier of the content, such as an HTTP URL to the content file. In the foregoing example, the HTTP URL is
HTTP://www.nokia.com/content/example.3gp. The content file may exist physically at the server
35 104 or may represent a virtual file. In contrast, the “a=control” attribute at the media level may include an identifier of the track in which the particular media component is stored, e.g., track ID

= 1, track ID=2, etc. The track ID may be utilized at the media level and the content streamed over HTTP may preserve the track ID mapping.

Based upon the session description information, the client consumption device 102 may determine if HTTP streaming is enabled as shown at operation 302 of FIG. 3 and, if not, may proceed to download the content in another manner, such as by progressive downloading and/or RTSP/RTP streaming as shown at operation 304 of FIG. 3. If HTTP streaming is enabled, however, the client consumption device 102 may issue a transfer protocol request for at least a portion of a media file. In one embodiment, the client consumption device 102 initially requests the file prefix and, in response to receipt of the file prefix, thereafter requests the content itself.

As shown in Figure 4, for example, the content file or container 402 that is streamed from the server 104 to the content consumption device 102 may include a file prefix and a plurality of fragments that comprise the requested portion of the media file. In this regard, FIG. 4 illustrates ISOFF-compliant structured media content that may be formatted by the media streaming circuitry 128 in accordance with one example embodiment of the invention. It will be appreciated, however, that ISOFF-compliant media files are used for purposes of example and not by way of limitation. Further, the structure of the media content illustrated in FIG. 4 and described in respect thereto is for purposes of example and the structure may be arranged differently in accordance with other embodiments of the invention. Additionally, the media content of FIG. 4 is described with respect to movie content. However, it will be appreciated that embodiments of the invention may be applied to other types of media content as well, such as audio only media content, video only media content, and audio/video media content.

In the illustrated embodiment, the prefix need only be transmitted once and is comprised of the ftyp and moov boxes 404 and 406 with the ftyp box 404 indicating the brand and the version number of the media content. The moov box 406 of the embodiment includes the mvhd box 408, one or more trak boxes 410 and the mvex box 412 including one or more trex boxes 414. Notably, the moov box 406 does not include any sample descriptions, e.g., no stts, ctts, stsc, stsz, stco or stss boxes, and thus is relatively small in size. Instead, the moov box 406 includes only information that describes the structure of the file, the existing tracks, the track types and the sample descriptions. In this regard, each trak box 410 of the moov box 406 includes information about a media track in an ISOFF-compliant file including, for example, information about the location and size of the track, descriptive information and timing information. Further, the mvex box 412 and its trex boxes 414 of the illustrated embodiment indicate that the portion of the media file that is to be provided will consist of movie fragments with each trex box 414 providing default information that need not be repeated with each subsequent movie fragment.

Following the file prefix, a movie fragment box moof 416, the mdat box 418 and optionally the mfra box 420 that comprise the description and media samples of a respective movie fragment may be repeatedly provided with one set of the moof and mdat boxes being

provided for each of the multiple fragments that comprise the requested portion of the media file. Following the provision of all fragments, that is, all moof and mdat boxes, an mfra box 418 may be optionally provided.

Regarding the request by the content consumption device 102 for the file prefix as shown in operation 306 of FIG. 3, the content consumption device 102 may issue a query to the server 104 for the file prefix including, for example, the ftyp and moov boxes 402 and 404. One example of a query to retrieve the file prefix that may be issued by the content consumption device 102 to the server 104 is GET

<http://www.nokia.com/httpstreaming.php?content=www.nokia.com/content/example.3gp&type=p>

refix HTTP/1.1 or in another example as follows: GET

<http://www.nokia.com/httpstreaming.php/example.3gp/prefix> HTTP/1.1

In the first of the foregoing queries, the query parameter “content” may be utilized to refer to the desired content and is typically already included in the original HTTP URL, such as the example provided above. In addition, the query parameter “type” is used to indicate the type of the request. In this context in which the prefix is being requested, the value of the type parameter is set to “prefix”. In one embodiment, the type parameter may utilize the augmented backus-naur format (ABNF) syntax as follows: Type = “type=” [“prefix” | “media” | “description”]. In this embodiment, the value “media” may be utilized to indicate that the media data that is to be requested is in the form of movie fragment, while the value “description” is utilized to indicate that a section description is requested, such as a SPD file. Other values for the type parameter may be alternatively utilized in other embodiments.

In another embodiment as in the URL example “GET <http://www.nokia.com/httpstreaming.php/example.3gp/prefix> HTTP/1.1”, the type parameter may follow the content parameter immediately, separated by a “/” character. In this instance, an identifier such as “type=” is not necessary as the URL will be constructed following a known pattern and the server 104 is able to extract the type indication from the request URL.

Following the request for the prefix and the delivery of the prefix by the server 104, the content consumption device 102 may issue the transfer protocol request for a portion of the media file. In response and as shown in operations 308, 310 and 312 of FIG. 3, the server 104 may then stream the fragments that comprise the requested media file until the entirety of the requested portion of the media file has been delivered or until the user has otherwise aborted the streaming.

In the transfer protocol request, the content consumption device 102 may indicate the streaming protocol to be utilized during download of the requested portion of the media file. For example, the transfer protocol request may request that the selected portion of the media file be streamed via HTTP. Although the transfer protocol request identifies the portion of the media file to be streamed from the server 104, the transfer protocol request does not identify and request a

single fragment. Instead, the transfer protocol request identifies a portion of a media file that may comprise, or may be formatted to comprise, a plurality of fragments.

The transfer protocol request issued by the content consumption device 102 may identify the portion of the media file to be streamed in various manners. In one embodiment, the transfer protocol request identify a start time and an ending time with the server 104 then streaming the portion of the media file, typically comprised of a plurality of fragments, that extends from the start time to the ending time. Alternatively, the transfer protocol request may identify a start time and a duration with the server 104 being configured to stream that portion of the media content that begins at the start time and that continues thereafter for the length of time identified by the duration. The content consumption device 102 may be configured to identify the portion of the media file to be streamed in other manners including, for example, a transfer protocol request that includes a number of frames and/or a number of scenes to be streamed with the frames and/or scenes generally being comprised of a plurality of fragments. Still further, the transfer protocol request issued by the content consumption device 102 of one embodiment may request the next piece or portion of the media file which, in turn, is interpreted by the server 104 as being that portion of the media file that is available at the time that the request is received. For example, the media file may be a recording of a live event such that a transfer protocol request issued by the client consumption device 102 for the next piece of the media file may trigger the streaming of that portion of the live broadcast that has been recorded up until the time at which the request is received by the server 104. Although various examples of the manner in which the content consumption device 102 may identify the portion of the media file to be streamed from the server 104 to the content consumption device 102, the content consumption device 102 may identify the portion of the media file to be streamed in other manners.

In one example, the content consumption device 102 may issue a GET request similar to the above-described GET request that is utilized to retrieve the file prefix, but which sets the type parameter to "media". In one embodiment, the content consumption device 102 may also identify a subset of the available tracks for streaming with the transfer protocol request soliciting the streaming of each of the requested tracks, such as a plurality of different types of tracks. Relative to the foregoing example, the content consumption device 102 may only be interested in English language audio and video data at 300 kbps, thereby resulting in a selection of a subset of the available tracks to include track 1 and track 7. As illustrated in the example provided below, a "tracks" parameter may be included in the GET request to identify the requested tracks. Although the GET request does not identify a particular fragment to be streamed, the content consumption device 102 of one embodiment may indicate the time frame of the portion of the media file that is to be streamed from the server 104 to the content consumption device 102. In one embodiment, for example, a start time and a duration are provided that define the portion of the media file to be streamed. As noted above, however, the portion of the media file that is requested to be streamed

may be identified in other manners. By way of illustration, but not of limitation, one example of an HTTP query issued by a content consumption device for media data is as follows: GET <http://www.nokia.com/httpstreaming.php/example.3gp/media/1+7/23.234/5/1/0> HTTP/1.1 or GET

5 <http://www.nokia.com/httpstreaming.php?content=www.nokia.com/content/example.3gp&type=media&tracks=1,7&ts=23.234&duration=5&fd=1&RAP=0> HTTP/1.1.

In addition to setting the type parameter to “media” and identifying tracks 1 and 7 to be streamed, the GET query includes a start time, e.g., “st”, of 23.234 which corresponds to the time stamp/playback time of the first fragment to be streamed by the server 104 to the content
10 consumption device 102. The GET query of this example also includes a duration of 5 that indicates the total duration in media time of the portion of the media file to be streamed from the server 104. By defining a duration that encompasses multiple fragments, the number of requests that must be issued by the content consumption device 102 and responded to by the server 104 may be reduced, thereby reducing the request overhead and increasing the efficiency of the
15 streaming process. In an alternative embodiment, the GET query may not include the duration parameter, but may, instead, include the end time “et” with the start and end times cooperating to define the window of time for which the requested portion of the media file is to be streamed. As also indicated in the foregoing example, the desired fragment duration “fd” may also be provided by the GET query. In one example, the fragment duration is in the range of one or more seconds,
20 e.g., one second in the foregoing example, since a fragment is independently playable and, as such, may have a reasonable duration to limit the initial playback delay and maintain the overhead at a reasonable level. Further, the GET query may include a random access point “RAP” parameter. If the random access point parameter is set to 1, the resulting query may indicate that the content consumption device 102 is requesting that the first fragment of the response begin
25 with a random access point in which case the actual start time may differ from the requested start time. Alternatively, if the random access point parameter is set to one as in the foregoing example, the first fragment of the response need to not start with a random access point and may, instead, begin at the requested start time, if the query has a requested start time as in the foregoing example.

30 Based upon the GET request, the server 104 may identify the fragments that comprise the requested portion of the media file and stream those fragments to the content consumption device 102. The server 104 may identify and retrieve the fragments from a stored file or set of files with prepared fragments, from a stored file or set of files without prepared fragments, e.g., a 3GP file or another file format, or from a live source. Since the content consumption device 102 is not
35 requesting the streaming of content on a fragment-by-fragment basis based upon an index file that was previously provided by the server 104, but is, instead, requesting the streaming of a portion of a media file that may include multiple fragments, the server 104 may, but need not be constrained

so as to have fragmented the media file prior to receipt of the request from the content consumption device 102. Instead, the server 104 may fragment the media file on the fly in response to the request from the content consumption device 102. By permitting the server 104 to fragment the requested portion of the media file in response to receipt of the request from the content consumption device 102, the server 104 may more readily stream portions of a media file that consist of a live recording or event since the live recording or event is incapable of having been fragmented in advance. Moreover, by permitting the server 104 to fragment the media file on the fly, the server 104 may reduce its storage requirements by only requiring storage of those fragments that are generated and not all possible track alternatives.

In regards to FIG. 3, for example, the server 104 may fragment the requested portion of the media file in response to receipt of the request from the content consumption device 102 and may then stream the content to the content consumption device 102, such as by repeatedly provide a sequence of moof and mdat boxes 416 and 418 and optionally an mfra box 420. In this regard, the mfra box 420 may delivered once at the end of each request, after each fragment or not at all. If delivered, the mfra box 420 includes information that indicates the time position of the samples in the fragments and, as such, may facilitate the playback of their requested portion of the media file.

As illustrated by the above example, a single request by a content consumption device 102 may include a request for different types of tracks, such as audio tracks, video tracks, subtitle tracks and the like. By permitting a single request to include a request different types of tracks, the number of requests that must be issued by the content consumption device 102 and separately responded to by the server 104 may be further reduced, thereby reducing the request overhead and increasing the efficiency of the streaming process.

Furthermore, the requested portion of the media file generally include multiple fragments and, therefore, extends over a longer period of time than conventional requests for a single fragment. As such, the content consumption device 102 may have a larger sampling window for determining the current network conditions, such as the bandwidth available to the content consumption device 102, and, as such, may be able to more accurately determine the corresponding bit rate for the requested portion of the media file. With respect to the foregoing example, a variety of different video tracks were available by the server 104 at different bit rates. Based upon the network conditions as determined by the content consumption device 102, the content consumption device 102 may request a video track having a particular bitrate that corresponds to the current network conditions. Since the requested portion of the media file encompasses multiple fragments, the content consumption device may not only be able to determine the network conditions with more reliability than prior determinations made on a fragment-by-fragment basis, but changes in the bit rate of the streamed video will change less frequently do, for example, to the increased lengths of the requested portion of the media file such

that the resulting display of the media file to a user will appear smoother with less disconcerting changes in the bit rate.

In an alternative embodiment, the content consumption device 102 and the server 104 may utilize a chunked mode in which the content consumption device 102 may request an open-ended range of the media file, such as by omitting the duration or end time parameters in the query. In this embodiment, the server 104 may employ a chunked mode to encapsulate each fragment in a separate chunk without having to prepare the entirety of the requested portion of the media file a priori. In other words, the content consumption device 102 of this embodiment does not have to indicate the size of the response body when commencing streaming of their response. The chunked mode may also be appropriate for requested portions of a media file having relatively long durations which would otherwise require more extensive processing by the server 104 and potentially delay the streaming of the requested portion of the media file.

A relatively long streaming response from a server 104 may be interrupted by a content consumption device 102 by pipelining HTTP requests. In this situation, a subsequent request may be utilized by the server 104 to terminate a prior response before starting the response to the new request. In this regard, the server 104 may need to retain session information, such as by utilizing HTTP cookies. By retaining session information, the server 104 may be able to identify that the pipeline requests are related.

FIG. 5 illustrates a flowchart according to an exemplary method for the dynamic streaming of media files according to an exemplary embodiment of the present invention. In this regard, FIG. 5 illustrates operations that may be performed at the server 104. The operations illustrated in and described with respect to FIG. 5 may, for example, be performed by or under the control of the media streaming circuitry 128. Operation 500 comprises receiving a transfer protocol request for at least a portion of a media file. Pursuant to operation 502, a plurality of fragments that are to be transmitted to the content consumption device 102 based upon the transfer protocol request are then determined. Operation 504 then provides for the transmission of the plurality of fragments to the content consumption device 102 in response to the transfer protocol request.

FIG. 6 illustrates a flowchart according to an exemplary method for the dynamic streaming of media files according to an exemplary embodiment of the present invention. In this regard, FIG. 6 illustrates operations that may be performed by a server 104. One or more of the processor 120, memory 122, communication interface 124, user interface 126, or media streaming circuitry 128 may, for example, cause the server 104 to perform respective operations illustrated in and described with respect to FIG. 6. Operation 600 comprises receiving a transfer protocol request for at least a portion of a media file. Pursuant to operation 602, a plurality of fragments that are to be transmitted to the content consumption device 102 based upon the transfer protocol

request are then determined. Operation 604 then transmits the plurality of fragments to the content consumption device 102 in response to the transfer protocol request.

FIG. 7 illustrates a flowchart according to an exemplary method for the dynamic streaming of media files according to an exemplary embodiment of the present invention. In this regard, FIG. 7 illustrates operations that may be performed at the content consumption device 102. The operations illustrated in and described with respect to FIG. 7 may, for example, be performed by or under the control of the media playback circuitry 118. Operation 700 comprises directing issuance of a transfer protocol request for at least a portion of a media file. Thereafter, at operation 702, a plurality of fragments are received from the server 104 in response to the transfer protocol request.

FIG. 8 illustrates a flowchart according to an exemplary method for the dynamic streaming of media files according to an exemplary embodiment of the present invention. In this regard, FIG. 8 illustrates operations that may be performed by a content consumption device 102. One or more of the processor 110, memory 112, communication interface 114, user interface 116, or media playback circuitry 118 may, for example, cause the content consumption device 102 to perform respective operations illustrated in and described with respect to FIG. 8. Operation 800 comprises transmitting a transfer protocol request for at least a portion of a media file. Thereafter, at operation 802, a plurality of fragments are received from the server 104 in response to the transfer protocol request.

FIGs. 3 and 5-8 are a flowchart of a system, method, and computer program product according to exemplary embodiments of the invention. It will be understood that each block of the flowchart, and combinations of blocks in the flowchart, may be implemented by various means, such as hardware and/or a computer program product comprising one or more computer-readable mediums having computer readable program instructions stored thereon. For example, one or more of the procedures described herein may be embodied by computer program instructions of a computer program product. In this regard, the computer program product(s) which embody the procedures described herein may be stored by one or more memory devices of a mobile terminal, server, or other computing device and executed by a processor (e.g., processor 110 or processor 120) in the computing device. In some embodiments, the computer program instructions comprising the computer program product(s) which embody the procedures described above may be stored by memory devices of a plurality of computing devices. As will be appreciated, any such computer program product may be loaded onto a computer or other programmable apparatus to produce a machine, such that the computer program product including the instructions which execute on the computer or other programmable apparatus creates means for implementing the functions specified in the flowchart block(s). Further, the computer program product may comprise one or more computer-readable memories on which the computer program instructions may be stored such that the one or more computer-readable memories can

direct a computer or other programmable apparatus to function in a particular manner, such that the computer program product comprises an article of manufacture which implements the function specified in the flowchart block(s). The computer program instructions of one or more computer program products may also be loaded onto a computer or other programmable apparatus (e.g., a content consumption device 102 and/or a server 104) to cause a series of operations to be performed on the computer or other programmable apparatus to produce a computer-implemented process such that the instructions which execute on the computer or other programmable apparatus implement the functions specified in the flowchart block(s).

Accordingly, blocks of the flowchart support combinations of means for performing the specified functions. It will also be understood that one or more blocks of the flowchart, and combinations of blocks in the flowchart, may be implemented by special purpose hardware-based computer systems which perform the specified functions, or combinations of special purpose hardware and computer program product(s).

The above described functions may be carried out in many ways. For example, any suitable means for carrying out each of the functions described above may be employed to carry out embodiments of the invention. In one embodiment, a suitably configured processor may provide all or a portion of the elements of the invention. In another embodiment, all or a portion of the elements of the invention may be configured by and operate under control of a computer program product. The computer program product for performing the methods of embodiments of the invention includes a computer-readable storage medium, such as the non-volatile storage medium, and computer-readable program code portions, such as a series of computer instructions, embodied in the computer-readable storage medium.

In a first example embodiment, a method is provided, which includes receiving a transport protocol request for at least the portion of the media file indicating that at least a portion of the media file is to be streamed to a content consumption device requesting the media file. The method of this embodiment also determines a plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request and then provides for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

The transport protocol request may identify the portion of the media file to be streamed in various manners, such as by identifying a time interval, identifying at least one scene or identifying a plurality of frames. In one embodiment, the transfer protocol request may identify different types of tracks such that the method of this embodiment provides for the transmission of a plurality of fragments which may, for example, contain multiplexed media data from each of the different types of tracks. The method of one embodiment also creates a plurality of fragments, such as by formatting at least the portion of the media file into a plurality of fragments, in response to the transfer protocol request, thereby supporting dynamic media file streaming and

avoiding requirements to have fragmented the media file in advance of the transport protocol request. Alternatively, the method of another embodiment identifies a plurality of fragments from among a plurality of predefined fragments in response to the transfer protocol request.

5 In a second example embodiment a computer program product is provided. The computer program product includes at least one computer-readable storage medium having computer-readable program instructions stored therein. The computer-readable program instructions of this embodiment may include program instructions which, when executed by a processor, are configured to receive a transport protocol request for at least the portion of the media file indicating that at least a portion of the media file is to be streamed to a content
10 consumption device requesting the media file. The computer-readable program instructions of this embodiment are also configured, when executed by a processor, to determine a plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request and to then provide for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

15 The transport protocol request may identify the portion of the media file to be streamed in various manners, such as by identifying a time interval, identifying at least one scene or identifying a plurality of frames. In one embodiment, the transfer protocol request may identify different types of tracks such that the method of this embodiment provides for the transmission of a plurality of fragments containing, for example, multiplexed media data of each of the different
20 types of tracks. The computer-readable program instructions of one embodiment are also configured, when executed by a processor, to create a plurality of fragments, such as by formatting at least the portion of the media file into a plurality of fragments, in response to the transfer protocol request, thereby supporting dynamic media file streaming and avoiding requirements to have fragmented the media file in advance of the transport protocol request.

25 Alternatively, the computer-readable program instructions of another embodiment are configured, when executed by a processor, to identify a plurality of fragments from among a plurality of predefined fragments in response to the transfer protocol request.

In a third example embodiment, an apparatus is provided. The apparatus of this embodiment includes at least one processor and at least one memory storing computer program
30 code. The at least one memory storing computer program code is configured to, working with the at least one processor, cause the apparatus to receive a transport protocol request for at least the portion of the media file indicating that at least a portion of the media file is to be streamed to a content consumption device requesting the media file. The apparatus of this embodiment is also caused to determine a plurality of fragments to be transmitted to the content consumption device
35 based upon the transfer protocol request and to then provide for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

The transport protocol request may identify the portion of the media file to be streamed in various manners, such as by identifying a time interval, identifying at least one scene or identifying a plurality of frames. In one embodiment, the transfer protocol request may identify different types of tracks such that the method of this embodiment provides for the transmission of a plurality of fragments containing, for example, multiplexed media data of each of the different types of tracks. The apparatus of one embodiment is also configured to create a plurality of fragments, such as by formatting at least the portion of the media file into a plurality of fragments, in response to the transfer protocol request, thereby supporting dynamic media file streaming and avoiding requirements to have fragmented the media file in advance of the transport protocol request. Alternatively, the apparatus of another embodiment is configured to identify a plurality of fragments from among a plurality of predefined fragments in response to the transfer protocol request.

The apparatus of this embodiment may be embodied as or may be embodied on a terminal. The apparatus of this embodiment may comprise a mobile phone further comprising user interface circuitry and user interface software stored on one or more of the at least one memory. The user interface circuitry and user interface software may be configured to facilitate user control of at least some functions of the mobile phone through use of a display. The user interface circuitry and user interface software may be further configured to cause at least a portion of a user interface of the mobile phone to be displayed on the display to facilitate user control of at least some functions of the mobile phone.

In a fourth example embodiment, an apparatus is provided that includes means for receiving a transport protocol request for at least the portion of the media file indicating that at least a portion of the media file is to be streamed to a content consumption device requesting the media file. The apparatus of this embodiment also includes means for determining a plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request and means for providing for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

The transport protocol request may identify the portion of the media file to be streamed in various manners, such as by identifying a time interval, identifying at least one scene or identifying a plurality of frames. In one embodiment, the transfer protocol request may identify different types of tracks such that the method of this embodiment provides for the transmission of a plurality of fragments containing, for example, multiplexed media data of each of the different types of tracks. The apparatus of one embodiment also includes means for creating a plurality of fragments, such as by formatting at least the portion of the media file into a plurality of fragments, in response to the transfer protocol request, thereby supporting dynamic media file streaming and avoiding requirements to have fragmented the media file in advance of the transport protocol request. Alternatively, the apparatus of another embodiment includes means for identifying a

plurality of fragments from among a plurality of predefined fragments in response to the transfer protocol request.

5 In a fifth example embodiment, a method is provided, which includes directing issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server. A method of this embodiment also receives a plurality of fragments from the server in response to the transfer protocol request.

10 The transport protocol request may identify the portion of the media file to be streamed in various manners, such as by identifying a time interval, identifying at least one scene or identifying a plurality of frames. In one embodiment, the issuance of the transfer protocol request includes the identification of different types of tracks to be streamed. As such, in this embodiment, the receipt of the plurality of fragments will include receiving the plurality of fragments of each of the different types of tracks.

15 In a sixth example embodiment a computer program product is provided. The computer program product includes at least one computer-readable storage medium having computer-readable program instructions stored therein. The computer-readable program instructions of this embodiment may include program instructions which, when executed by a processor, are configured to direct issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server. The computer-readable program instructions of this embodiment are also configured, when executed by a processor, to receive a plurality of fragments from the server in response to the transfer protocol request.

20 The transport protocol request may identify the portion of the media file to be streamed in various manners, such as by identifying a time interval, identifying at least one scene or identifying a plurality of frames. In one embodiment, the issuance of the transfer protocol request includes the identification of different types of tracks to be streamed. As such, in this embodiment, the receipt of the plurality of fragments will include receiving the plurality of fragments of each of the different types of tracks.

25 In a seventh example embodiment, an apparatus is provided. The apparatus of this embodiment includes at least one processor and at least one memory storing computer program code. The at least one memory storing computer program code is configured to, working with the at least one processor, cause the apparatus to direct issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server. The apparatus of this embodiment is also caused to receive a plurality of fragments from the server in response to the transfer protocol request.

30 The transport protocol request may identify the portion of the media file to be streamed in various manners, such as by identifying a time interval, identifying at least one scene or identifying a plurality of frames. In one embodiment, the issuance of the transfer protocol request

includes the identification of different types of tracks to be streamed. As such, in this embodiment, the receipt of the plurality of fragments will include receiving the plurality of fragments of each of the different types of tracks.

5 The apparatus of this embodiment may be embodied as or may be embodied on a terminal. The apparatus of this embodiment may comprise a mobile phone further comprising user interface circuitry and user interface software stored on one or more of the at least one memory. The user interface circuitry and user interface software may be configured to facilitate user control of at least some functions of the mobile phone through use of a display. The user interface circuitry and user interface software may be further configured to cause at least a portion
10 of a user interface of the mobile phone to be displayed on the display to facilitate user control of at least some functions of the mobile phone.

In an eighth example embodiment, an apparatus is provided that includes means for directing issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server. The apparatus of this
15 embodiment also includes means for receiving a plurality of fragments from the server in response to the transfer protocol request.

The transport protocol request may identify the portion of the media file to be streamed in various manners, such as by identifying a time interval, identifying at least one scene or identifying a plurality of frames. In one embodiment, the issuance of the transfer protocol request
20 includes the identification of different types of tracks to be streamed. As such, in this embodiment, the receipt of the plurality of fragments will include receiving the plurality of fragments of each of the different types of tracks.

In a further embodiment, a system is provided that includes a server and a content consumption device configured to support media file streaming from the server to the content
25 consumption device. In this embodiment, the content consumption device is configured to direct issuance of a transfer protocol request for at least a portion of the media file indicating that at least a portion of the media file is to be streamed from a server and to receive a plurality of fragments from the server in response to the transfer protocol request. In addition, the server in this embodiment is configured to receive a transport protocol for at least the portion of the media file
30 indicating that at least a portion of the media file is to be streamed to a content consumption device requesting the media file. The server of this embodiment is also caused to determine a plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request and to then provide for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

35 Many modifications and other embodiments of the inventions set forth herein will come to mind to one skilled in the art to which these inventions pertain having the benefit of the teachings presented in the foregoing descriptions and the associated drawings. Therefore, it is to

be understood that the embodiments of the invention are not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the invention. Moreover, although the foregoing descriptions and the associated drawings describe exemplary embodiments in the context of certain exemplary combinations of elements and/or functions, it should be appreciated that different combinations of elements and/or functions may be provided by alternative embodiments without departing from the scope of the invention. In this regard, for example, different combinations of elements and/or functions than those explicitly described above are also contemplated within the scope of the invention. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

WHAT IS CLAIMED IS:

1. A method comprising:
receiving a transport protocol request for at least a portion of a media file indicating that
at least the portion of the media file is to be streamed to a content consumption device requesting
the media file;
determining, via a processor, a plurality of fragments to be transmitted to the content
consumption device based upon the transfer protocol request; and
providing for transmission of the plurality of fragments to the content consumption device
in response to the transfer protocol request.
2. A method according to Claim 1 wherein receiving the transport protocol request
comprises receiving the transport protocol request that identifies the portion of the media file to
be streamed by at least one of identifying a time interval, identifying at least one scene or
identifying a plurality of frames.
3. A method according to Claim 1 wherein receiving the transport protocol request
comprises receiving the transfer protocol request that identifies different types of tracks, and
wherein providing for transmission of the plurality of fragments comprises providing for
transmission a plurality of fragments of each of the different types of tracks.
4. A method according to Claim 3 wherein providing for transmission of the
plurality of fragments comprises providing for transmission of a plurality of fragments that
contain multiplexed media data from each of the different types of tracks.
5. A method according to Claim 1 further comprising formatting at least the portion
of the media file into a plurality of fragments in response to the transfer protocol request.
6. A computer program product comprising at least one computer-readable storage
medium having computer-executable program code instructions stored therein that, upon
execution by the processor, cause performance of the method of Claim 1.

7. An apparatus comprising at least one processor and at least one memory storing computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to:

5 receive a transport protocol request for at least a portion of a media file indicating that at least the portion of the media file is to be streamed to a content consumption device requesting the media file;

determine a plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request; and

10 provide for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

8. An apparatus according to Claim 7 wherein the at least one memory and the computer program code are configured to, with the at least one processor, cause the apparatus to receive the transport protocol request by receiving the transport protocol request that identifies the
15 portion of the media file to be streamed by at least one of identifying a time interval, identifying at least one scene or identifying a plurality of frames.

9. An apparatus according to Claim 7 wherein the at least one memory and the computer program code are configured to, with the at least one processor, cause the apparatus to
20 receive the transport protocol request by receiving the transfer protocol request that identifies different types of tracks, and to provide for transmission of the plurality of fragments by providing for transmission a plurality of fragments of each of the different types of tracks.

10. An apparatus according to Claim 9 wherein the at least one memory and the
25 computer program code are configured to, with the at least one processor, cause the apparatus to provide for transmission of the plurality of fragments by providing for transmission of a plurality of fragments that contain multiplexed media data from each of the different types of tracks.

11. An apparatus according to Claim 7 wherein the at least one memory and the
30 computer program code are configured to, with the at least one processor, further cause the apparatus to format at least the portion of the media file into a plurality of fragments in response to the transfer protocol request.

12. A method comprising:
directing, via a processor, issuance of a transfer protocol request for at least a portion of a media file indicating that at least a portion of the media file is to be streamed from a server; and
receiving a plurality of fragments from the server in response to the transfer protocol request.
13. A method according to Claim 12 wherein directing issuance of the transfer protocol request comprises directing issuance the transport protocol request that identifies the portion of the media file to be streamed by at least one of identifying a time interval, identifying at least one scene or identifying a plurality of frames.
14. A method according to Claim 12 wherein directing issuance of the transfer protocol request comprises directing issuance of the transfer protocol request that includes an identification of different types of tracks to be streamed, and wherein receiving the plurality of fragments comprises receiving the plurality of fragments of each of the different types of tracks.
15. A method according to Claim 12 wherein directing issuance of the transfer protocol request comprises directing issuance of the transfer protocol request including an identification of a protocol via which the media file is to be streamed.
16. A computer program product comprising at least one computer-readable storage medium having computer-executable program code instructions stored therein that, upon execution by the processor, cause performance of the method of Claim 12.
17. An apparatus comprising at least one processor and at least one memory storing computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to:
direct issuance of a transfer protocol request for at least a portion of a media file indicating that at least a portion of the media file is to be streamed from a server; and
receive a plurality of fragments from the server in response to the transfer protocol request.
18. An apparatus according to Claim 17 wherein the at least one memory and the computer program code are configured to, with the at least one processor, cause the apparatus to direct issuance of the transfer protocol request by directing issuance the transport protocol request that identifies the portion of the media file to be streamed by at least one of identifying a time interval, identifying at least one scene or identifying a plurality of frames.

19. An apparatus according to Claim 17 wherein the at least one memory and the computer program code are configured to, with the at least one processor, cause the apparatus to direct issuance of the transfer protocol request by directing issuance of the transfer protocol request that includes an identification of different types of tracks to be streamed, and to receive the plurality of fragments by receiving the plurality of fragments of each of the different types of tracks.

20. An apparatus according to Claim 17 wherein the at least one memory and the computer program code are configured to, with the at least one processor, cause the apparatus to direct issuance of the transfer protocol request by directing issuance of the transfer protocol request including an identification of a protocol via which the media file is to be streamed.

21. An apparatus comprising:
means for receiving a transport protocol request for at least a portion of a media file indicating that at least the portion of the media file is to be streamed to a content consumption device requesting the media file;
means for determining a plurality of fragments to be transmitted to the content consumption device based upon the transfer protocol request; and
means for providing for transmission of the plurality of fragments to the content consumption device in response to the transfer protocol request.

22. An apparatus comprising:
means for directing issuance of a transfer protocol request for at least a portion of a media file indicating that at least a portion of the media file is to be streamed from a server; and
means for receiving a plurality of fragments from the server in response to the transfer protocol request.

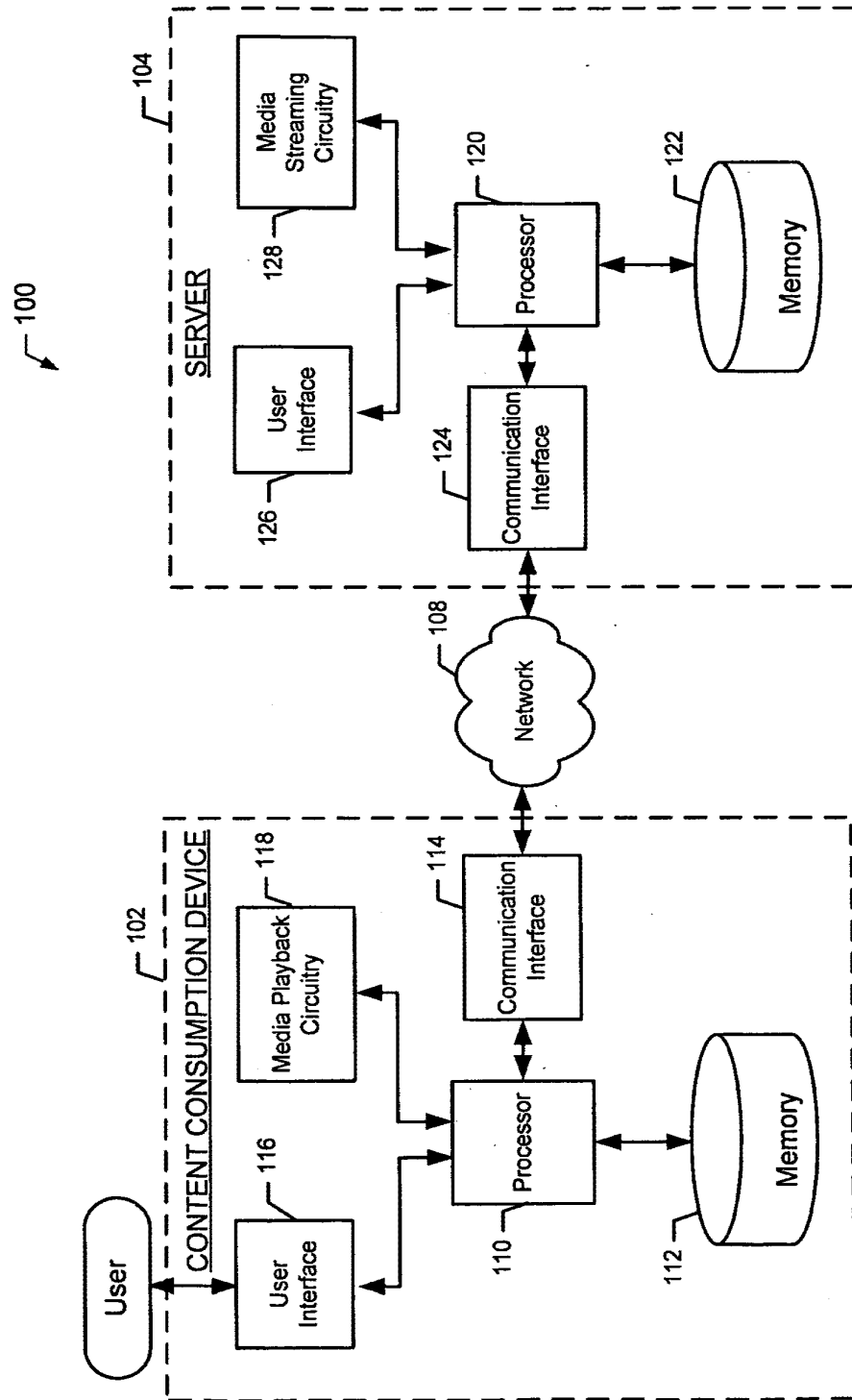


FIG. 1

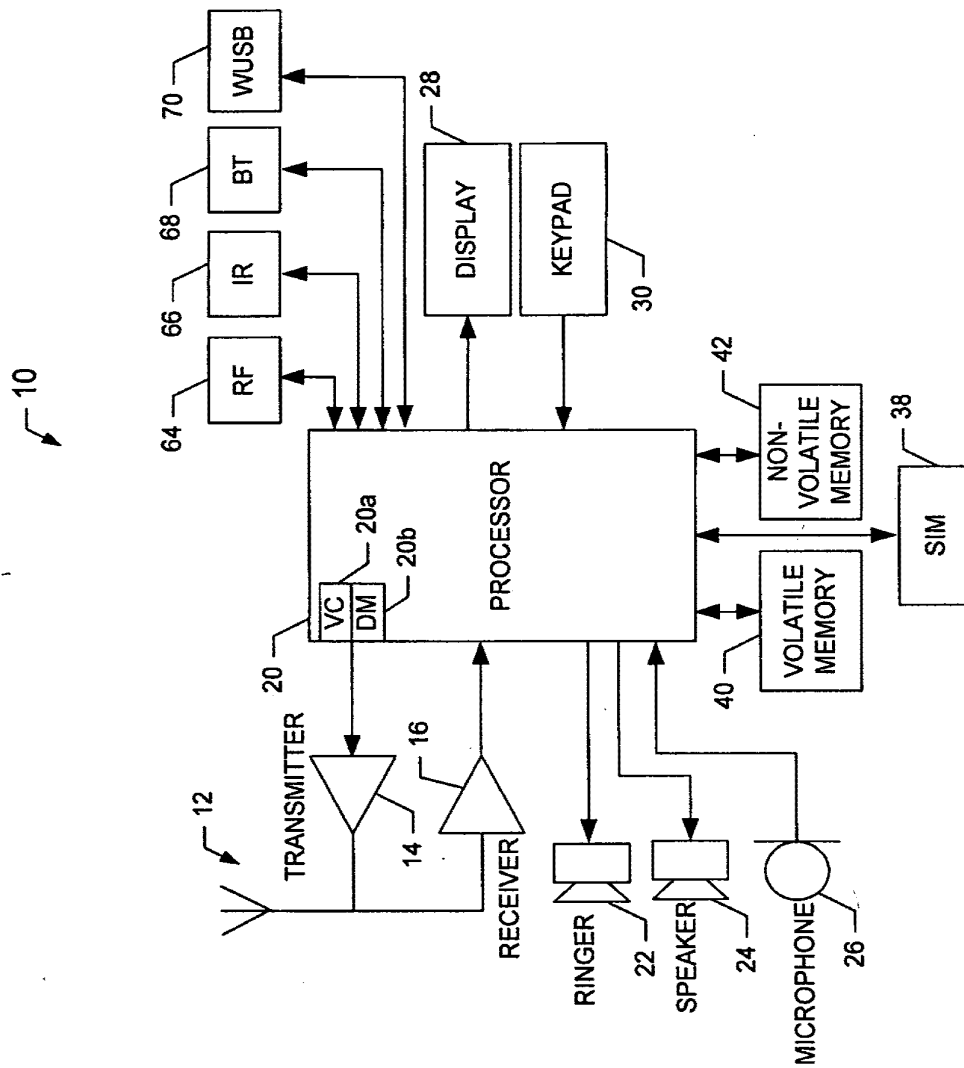
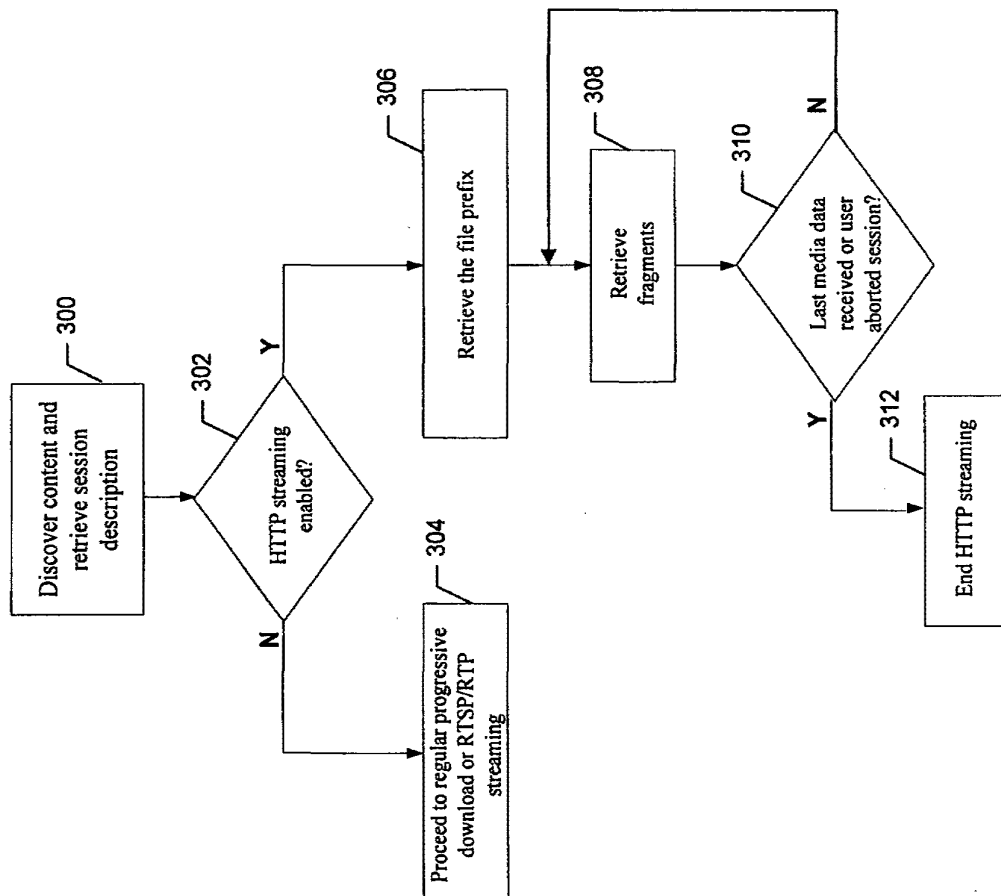


FIG. 2

FIG. 3

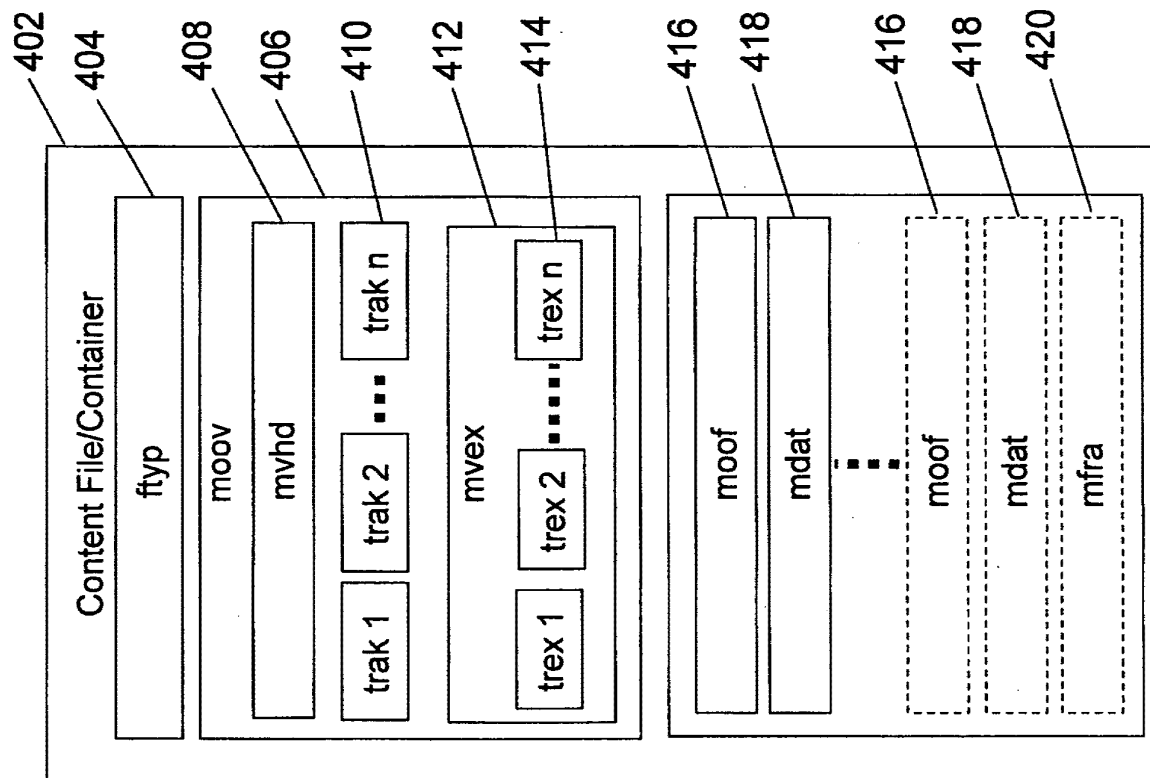
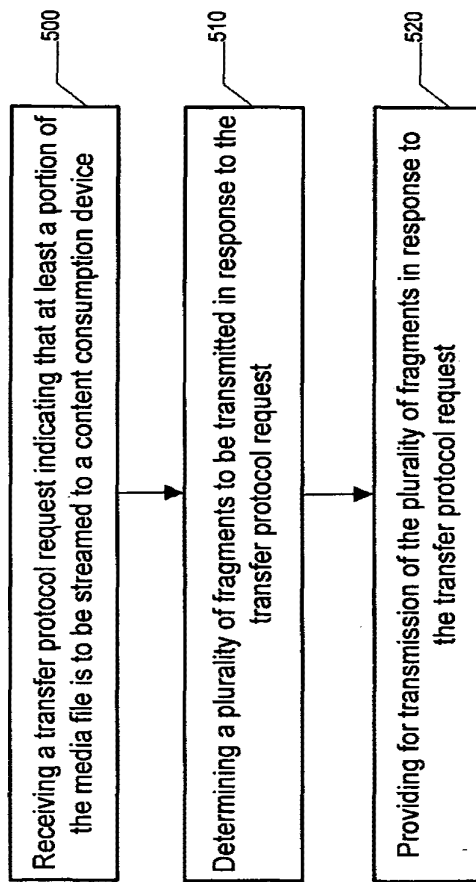
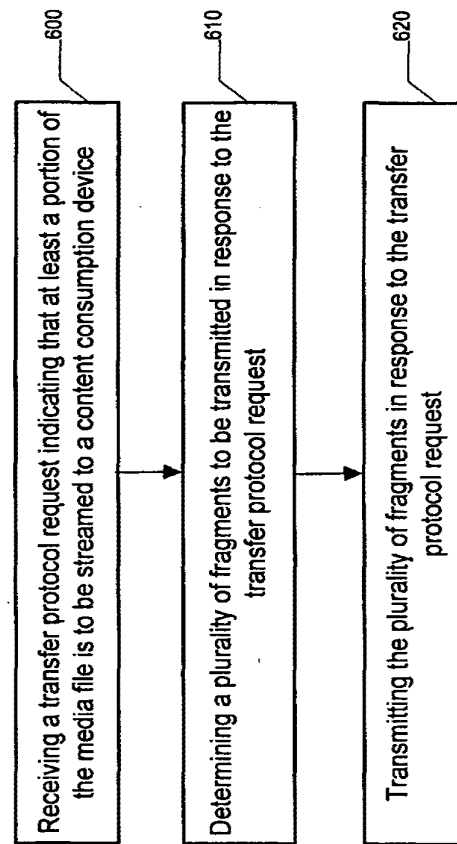
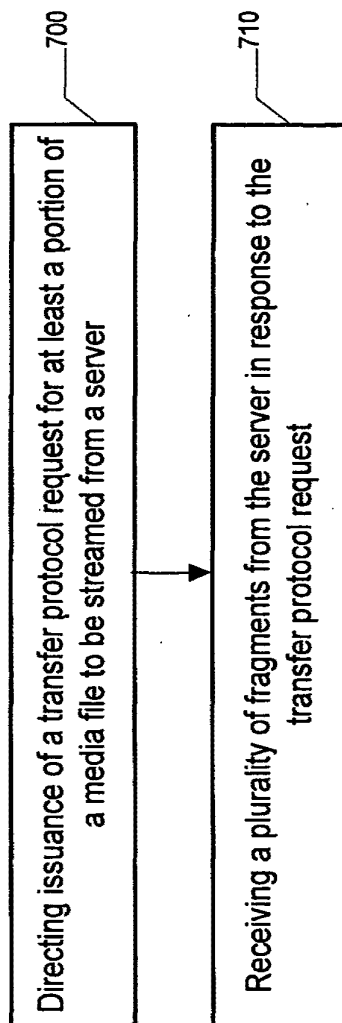
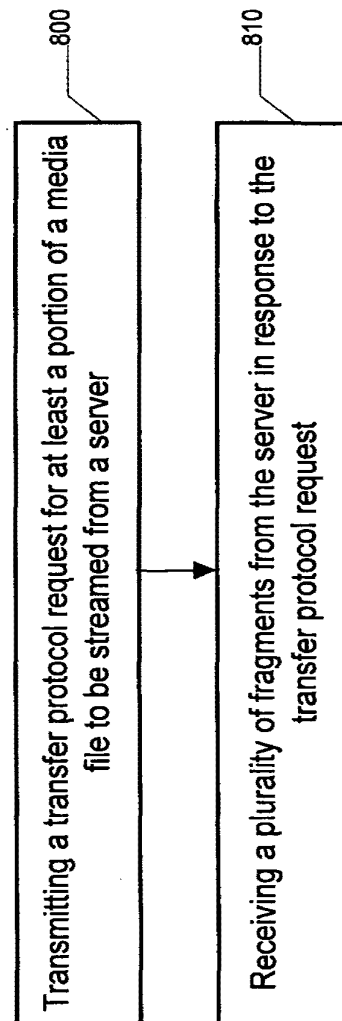


FIG. 4

FIG. 5FIG. 6

FIG. 7FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2010/002475

A. CLASSIFICATION OF SUBJECT MATTER

IPC: see extra sheet

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)

IPC: H04L, H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

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

EPO-INTERNAL, WPI DATA, PAJ, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 20080022005 A1 (WU, FENG ET AL), 24 January 2008 (24.01.2008), figures 1,6, claims 1,2,5,15, abstract, paragraphs [0001], [0002], [0011], [0021], [0025], [0046], [0047] --	1-2,5-8, 11-13,15-18, 20-22
A	WO 2009071978 A2 (NOKIA CORPORATION), 11 June 2009 (11.06.2009), page 4, line 5 - line 11; page 5, line 6 - line 33 --	1-22
A	US 20080010372 A1 (KHEDOURI, ROBERT ET AL), 10 January 2008 (10.01.2008), paragraphs [0098], [0108] --	1-22

☒ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* 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

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

"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

19 January 2011

Date of mailing of the international search report

20-01-2011

Name and mailing address of the ISA/

Swedish Patent Office

Box 5055, S-102 42 STOCKHOLM

Facsimile No. +46 8 666 02 86

Authorized officer

Jesper Bergstrand / Eö

Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2010/002475

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 20040078470 A1 (BAUMEISTER, SASCHA ET AL), 22 April 2004 (22.04.2004), claim 1 --	1-22
A	US 20080165906 A1 (HO, EDWIN ET AL), 10 July 2008 (10.07.2008), claims 1, 12 --	1-22
A	"Baseline Architecture and Definitions for HTTP Streaming", 3GPP Draft S4-090603_HTTP_Streaming_ Architecture, 20090812, [Retrieved on 2010-01-14], Retrieved from the Internet: <URL:http://www. 3gpp.org/ftp/Specs/html-info/TDocExMtg--S4-55-- 27369.htm>, whole document -- -----	1-22

Form PCT/ISA/210 (continuation of second sheet) (July 2009)

International patent classification (IPC)
H04L 29/06 (2006.01)

Download your patent documents at www.prv.se

The cited patent documents can be downloaded:

- From "Cited documents" found under our online services at www.prv.se (English version)
- From "Anförda dokument" found under "e-tjänster" at www.prv.se (Swedish version)

Use the application number as username. The password is **LJPQNGMLQ**.

Paper copies can be ordered at a cost of 50 SEK per copy from PRV InterPat (telephone number 08-782 28 85).

Cited literature, if any, will be enclosed in paper form.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IB2010/002475

US	20080022005	A1	24/01/2008	NONE		
WO	2009071978	A2	11/06/2009	EP	2215842 A	11/08/2010
US	20080010372	A1	10/01/2008	NONE		
US	20040078470	A1	22/04/2004	NONE		
US	20080165906	A1	10/07/2008	NONE		



- (51) **International Patent Classification:**
H04L 29/08 (2006.01) *H04W 4/18* (2009.01)
- (21) **International Application Number:**
PCT/EP2011/067834
- (22) **International Filing Date:**
12 October 2011 (12.10.2011)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
61/499,906 22 June 2011 (22.06.2011) US
- (71) **Applicant (for all designated States except US):** TELEFONAKTIEBOLAGET L M ERICSSON (PUBL) [SE/SE]; S-16483 Stockholm (SE).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** PERKUHN, Heiko [DE/DE]; Bergdriesch 40a, 52062 Aachen (DE). KAMP-MANN, Markus [DE/DE]; Bremenberg 36a, 52072 Aachen (DE). REMBARZ, Rene [DE/DE]; Anne-Frank-Str. 14, 52064 Aachen (DE).

- (74) **Agent:** MEIER, Florian; Thomas-Wimmer-Ring 15, 80539 München (DE).
- (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, QA, RO, RS, RU, RW, 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, RW, 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).

[Continued on next page]

(54) **Title:** METHODS AND DEVICES FOR CONTENT DELIVERY CONTROL

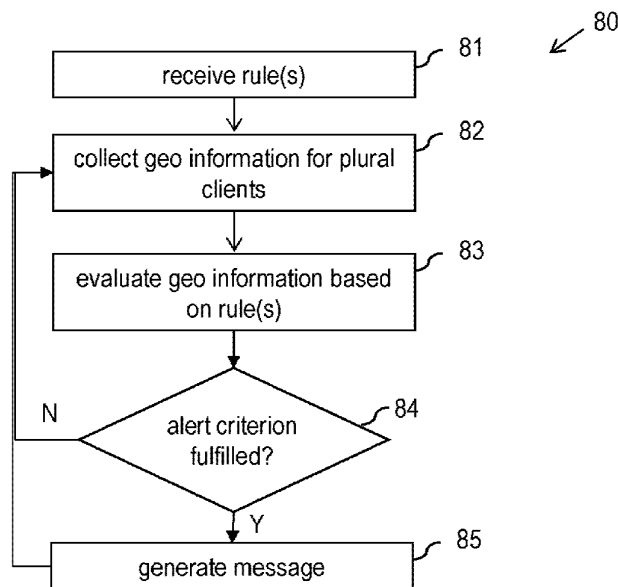


FIG. 8

(57) **Abstract:** To control content delivery to plural clients (6) of a mobile communication network (1), a network entity (5) of the mobile communication network (1) collects geo information of the plural clients (6) and evaluates the collected geo information of the plural clients (6) based on a rule to determine whether the content delivery is to be adapted. The network entity (5) identifies a region for which the content delivery is to be adapted. The network entity (5) selectively outputs a message (9) to a controlling entity (10) if it is determined that the content delivery is to be adapted. The controlling entity (10) controls the content delivery in response to receiving the message (9), to adapt the content delivery for clients located in the identified region.



Published:

— *with international search report (Art. 21(3))*

Methods and Devices for Content Delivery Control

5

Technical Field

The invention relates to methods of controlling content delivery to plural clients of a mobile communication network and to corresponding devices.

10

Background

The delivery of content to clients of a mobile communication network is of significant interest. Delivery of media, such as audio, pictures or video has widespread applicability. Technical implementations may be based on streaming technologies. The streaming may be live streaming or on demand. The delivery of content may also be made by download. For illustration, media delivery to vehicles is regarded as being interesting. A significant fraction of radio consumption happens in the car, and always being able to reach the favorite stations may be of interest to both users and service providers. At the same time, audio streaming has low driver distraction risk. Further examples for delivery of content include various car infotainment features, such as delivery of video to a backseat entertainment system, online gaming via a backseat entertainment system, but also traffic and navigation information. Movies may be delivered to a car over the mobile communication network for viewing at a backseat entertainment system. Additional examples for the delivery of content to clients of a mobile communication network include the delivery of video or other media to handheld devices, or online gaming for handheld devices. The access network of the mobile communication network may prone to experiencing an overload situation when a large number of people located in the same area, such as a school yard, wishes to view the same movie at the same time.

30

Different technical implementations of media streaming to vehicles or other clients are conceivable. For streaming over a wireless communication network, the streaming solution should provide adaptability to accommodate fluctuating link rates of the current wireless link.

35

Adaptive streaming may utilize the Real-Time Transport Protocol (RTP) as transport protocol, and the Real-Time Streaming Protocol (RTSP) as session initiation and control protocol. Another approach to implement adaptive streaming is based on adaptive Hypertext

Transfer Protocol (HTTP) streaming. In the latter case, media may be stored at a streaming server as a sequence of media files encoded with different rates. The client may be informed before the session or updated during the session about the different media files and the associated bit rates via the transfer of a corresponding manifest file. In response thereto, the client may execute rate adaptation by requesting a media file with the suited bit rate from the server. An adaptive streaming solution is described in 3GPP TS 26.233 V9.0.0 (2009-12), for example.

In such adaptive streaming solutions, rate adaptation is carried out on an individual session level. This limits flexibility in adapting content delivery. For illustration, as the number of clients which request content delivery increases, congestion situations in the radio access network may become more likely. The service provider which offers the content only has limited options to react to such situations when quality degradation is executed at an individual session and user level. Also, conventional adaptive streaming also makes it challenging to take into account constraints which may be due to, for example, agreements on data volumes which may be delivered by a given service provider over a given communication network.

Summary

It is the object to alleviate or obviate at least some of the above drawbacks, and to provide improved methods and devices for controlling content delivery.

According to an embodiment, a method of controlling content delivery to plural clients of a mobile communication network is provided. The content is delivered to the plural clients over a radio access network of the mobile communication network. A network entity associated with the mobile communication network collects geo information of the plural clients. The network entity evaluates the collected geo information of the plural clients based on a rule to determine whether the content delivery is to be adapted and to identify a region for which the content delivery is to be adapted. The network entity selectively outputs a message to a controlling entity if it is determined that the content delivery is to be adapted. The controlling entity controls the content delivery in response to receiving the message, to adapt the content delivery for clients located in the identified region.

According to another embodiment, a controlling entity configured to control content delivery to plural clients of a mobile communication network is provided. The controlling entity comprises an interface for communication with a network entity of the mobile communication

network and a controller. The controller is configured to output a rule for evaluating geo information of the plural clients to the network entity over the interface. The controller is configured to receive a message from the network entity over the interface, the message containing information on a location-dependent adaptation of the content delivery. The controller is configured to adapt the content delivery in a location-dependent manner for clients located in a region based on the received message, which region is determined based on the information on a location-dependent adaptation.

According to another embodiment, a method of controlling content delivery to plural clients of a mobile communication network is provided. The method comprises outputting a rule for evaluating geo information of the plural clients to a network entity of the mobile communication network. The method comprises receiving a message from the network entity, the message containing information on a location-dependent adaptation of the content delivery. The method comprises adapting the content delivery in a location-dependent manner for clients located in a region based on the received message, which region is determined based on the information on a location-dependent adaptation. The method may be performed by a controlling entity for controlling content delivery.

According to another embodiment, a network entity for a mobile communication network is provided. The network entity comprises a receiver configured to receive geo information of plural clients of the mobile communication network. The network entity comprises an interface configured to receive a rule from a controlling entity controlling content delivery to the plural clients. The network entity comprises a processing unit configured to evaluate the received geo information of the plural clients based on the received rule to determine whether the content delivery is to be adapted and identify a region for which the content delivery is to be adapted. The processing unit is configured to selectively output a message over the interface if it is determined that the content delivery is to be adapted.

According to another embodiment, a method is provided for determining whether a location-specific adaptation of content delivery to plural clients of a mobile communication network is required. A network entity of the mobile communication network collects geo information of the plural clients. The network entity evaluates the collected geo information of the plural clients based on a rule to determine whether the content delivery is to be adapted and to identify a region for which the content delivery is to be adapted. The network entity selectively outputs a message to a controlling entity if it is determined that the content delivery is to be adapted. The network entity may receive the rule from the controlling entity.

According to another embodiment, a system is provided which comprises a mobile communication network having a network entity of an embodiment and a controlling entity of an embodiment. The system may further comprise a content source for delivering content to clients of the mobile communication network via a radio access network of the mobile communication network.

In the methods and devices of embodiments, information about the locations of plural clients is taken into account for content delivery adaptation. This affords enhanced versatility in adapting content delivery.

According to further embodiments, other methods, devices, systems, or computer program products for implementing the methods may be provided.

Brief Description of the Drawings

Fig. 1 is a schematic illustration of a system in which concepts according to embodiments of the invention may be implemented.

Fig. 2 is a block diagram of a system operative in accordance with embodiments.

Figs. 3 and 4 schematically illustrate content delivery control based on geo information of plural clients in accordance with embodiments.

Fig. 5 is a block diagram of a controlling entity of an embodiment.

Fig. 6 is a flow diagram representing operation of the controlling entity in the method of an embodiment.

Fig. 7 is a block diagram of a network entity of an embodiment.

Fig. 8 is a flow diagram representing operation of the network entity in the method of an embodiment.

Fig. 9 is a block diagram of a system illustrating a signal flow in accordance with embodiments.

Fig. 10 is a block diagram of a system illustrating a signal flow in accordance with embodiments.

Fig. 11 is a block diagram of a system illustrating a signal flow in accordance with
5 embodiments.

Fig. 12 is a block diagram of a system illustrating a signal flow in accordance with embodiments.

10 Fig. 13 is a schematic view illustrating adaptation of content delivery based on an indicator for a number of clients in accordance with embodiments.

Detailed Description of Embodiments

15 In the following, the invention will be explained in more detail with reference to exemplary embodiments and to the accompanying drawings in which identical or corresponding reference numerals designate identical or corresponding entities. The illustrated embodiments relate to concepts of controlling content delivery. Examples for such content delivery include audio streaming, video streaming, or other media streaming. Additional
20 examples include the downloading of media or other content. The delivery may be live or on demand. Some embodiments and the entities used therein may be described in the context of mobile communication networks operative according to a given standard, such as the 3GPP (Third Generation Partnership Project) Technical Specifications (TSs). It is to be understood that the illustrated concepts may be applied in other types of communication
25 networks as well.

Fig. 1 illustrates a system in which concepts of embodiments may be implemented. Content is delivered to clients over a mobile communication network 1. A service provider 3 may offer media content via live streaming or may offer content on demand. The content may be stored
30 at one or plural content sources 7. The content sources 7 may provide the content, which is ultimately delivered to clients via the mobile communication network 1. The content may be delivered via a node of the mobile communication network 1, such as a GGSN (Gateway GPRS Support Node) 2. Other nodes may be used in other network technologies. For illustration, the node 2 may be a PDN (Packet Data Network) Gateway which provides
35 connectivity between a client and external packet data networks. The mobile communication network 1 may be configured for operation according to LTE (Long Term Evolution). The mobile communication network 1 may have a core network architecture which is the SAE

(System Architecture Evolution) of LTE. The content may be transmitted to the clients of the mobile communication network over a radio access network (RAN) 4.

According to embodiments, the content delivery may be adapted. Situations in which an
5 adaptation of content delivery is performed include situations in which there is a congestion
situation in the RAN, or in which a congestion situation is anticipated, in a certain region
covered by the mobile communication network. In this case, the content delivery may be
adapted to relieve the RAN in the affected region. A region in which a congestion in the RAN
already occurs, or in which the congestion is anticipated to occur in the future unless
10 counter-measures are taken, will be referred to as "affected region". The affected region is
identified by a network entity of the mobile communication network.

The adaptation of the content delivery may be implemented using information collected by, or
otherwise available in or retrieved by, the mobile communication network 1. The adaptation
15 of the content delivery may be based on geo information of plural clients. Thereby, a
location-dependent adaptation of content delivery may be implemented. Other external
information may be used, for example to anticipate local congestion situations in the RAN.
Such information may include any one, or any combination of, time of day, topological
information of the mobile communication network 1, road traffic conditions, information on
20 service types of service provider and information on subscription types of clients. It is not
required that the conditions in the RAN be actually measured to perform an adaptation of the
content delivery.

A network entity 5 is associated with the mobile communication network 1. The network
25 entity 5 may be located within the mobile communication network 1. The network entity 5
may be responsible for collecting geo information of clients, i.e. information on locations of
clients. The network entity 5 may communicate with other entities to collect the geo
information of clients. For illustration, the network entity 5 may retrieve the geo information
from a geo cast service, e.g. from a server of the geo cast service. The network entity 5 may
30 be configured to make this information directly available to a controlling entity 10 or to
another third party. For enhanced privacy and/or to reduce data traffic, the network entity 5
may be configured to evaluate the collected geo information in accordance with a rule. Based
on a result of the evaluation, the network entity 5 may selectively output a message 9 to the
controlling entity 10, or another third party, to inform the controlling entity 10, or the other
35 third party, of the result of the evaluation. The network entity 5 may have an interface to
receive a rule 8 for evaluating the collected geo information of the clients. The rule 8 may be
injected by the service provider 3, for example via the controlling entity 10.

The controlling entity 10 may be located outside the mobile communication network 1. I.e., the controlling entity 10 may be provided such that it is under the control of a party other than the operator of the mobile communication network 1. The controlling entity 10 may be operative to decide which measures are taken to relieve the RAN in regions which are affected by a present or anticipated congestion situation. The controlling entity 10 may be configured to take any one, or any combination, of plural different measures to relieve the RAN. The controlling entity 10 may also be configured to decide on an extent by which the content delivery is to be adapted. The different measures which may be taken under the control of the controlling entity 10 include an adaptation of a content delivery rate at which a content source 7 provides content to clients located in an identified region. Alternatively or additionally, the controlling entity 10 may perform a quality of service (QoS) bearer adaptation in the mobile communication network 1 to adjust a guaranteed bit rate for the QoS bearer over which the content is delivered. Alternatively or additionally, the controlling entity 10 may prioritize clients located in the identified region based on subscription information. The adaptation of the content delivery may be implemented at the application layer.

The controlling entity 10 may transmit one or several control messages to put the adaptation of the content delivery into effect. The execution of the adaptation may take place via execution points which may also be located outside the mobile communication network 1. The execution of the adaptation may alternatively or additionally take place in the clients of the mobile communication network.

The system of Fig. 1 may be configured to implement a closed control loop. Quality of Experience (QoE) reports may be received by the service provider 3 or the controlling entity 10. The QoE reports may be evaluated against the adaptation which was made, and/or against the rules 8 which were injected into the network entity 5. The QoE reports may be used as feedback to adapt the rules 8 injected into the network entity 5.

Fig. 2 illustrates in greater detail a system in which methods and devices of embodiments are used to control content delivery. Content is delivered to a client 6 of a mobile communication network 1. A controlling entity 10 interfaces with a network entity 5 of the mobile communication network. The controlling entity 10 uses information provided by the network entity 5 to control content delivery. The controlling entity 10 may take one or plural different measures to relieve the data transmission associated with the content delivery over the RAN, of the mobile communication network 1.

The network entity 5 may be a dedicated computing device operated by the operator of the mobile communication network 1, in order to evaluate geo information of clients based on rules received from the controlling entity 10. For illustration, the network entity 5 may be a computing device which interfaces with a geo messaging server to retrieve information on locations of clients from the geo messaging server. Alternatively, the network entity 5 which evaluates geo information of plural clients may have additional functionalities. For illustration, the network entity 5 may be a geo messaging server. In this case, the network entity 5 operating as geo messaging server which receives location updates from different clients as the clients move from one area to another area covered by the mobile communication network 1. For further illustration, the network entity 5 may be, or may interface with, any other location-aware node of the mobile communication network 1 which monitors geo information of plural clients. The geo information may, but does not need to include information on geographic coordinates. The geo information may have any one of a variety of formats. For illustration, the geo information may be indicative of a RAN cell in which the respective client is presently located. The geo information may also be more coarse-grained. For illustration, the geo information for each client may be indicative of an area, or several areas, in which the client is presently located. The areas may respectively cover plural RAN cells.

The network entity 5 has an interface to receive a rule 28 according to which the geo information is evaluated. The interface over which the rule 28 is received at the network entity 5 may be an application programming interface (API). The rule 28 may comprise one or plural thresholds which are used by the network entity 5 in a threshold comparison. More complex rules may be input into the network entity 5. Such rules may include monitoring the rate of change in numbers of clients located in the various areas, in addition to monitoring the number of clients that is located in the various areas at any given time. Using, for example, information on changes in the numbers of clients, congestion situations in certain areas of the RAN may be predicted before they actually occur. The rules may depend on other or additional parameters, such as time of day. Thereby, variations in RAN loads which occur in different cells in a regular manner (e.g. as a function of time of day) may be taken into account.

The network entity 5 collects geo information on plural clients to evaluate the rule based thereon. Based on the evaluation of the geo information using the rule 28, the network entity 5 may output a message 29 notifying the controlling entity 10 that a certain condition has occurred. As will be explained in more detail, the message 29 may indicate that a number or density of clients located in a certain region exceeds a threshold. The message 29 may also

indicate that the number or density of clients located in a certain region is likely to exceed a threshold in the future. The message 29 may also indicate that the number or density of clients falls below another threshold, or is likely to fall below the other threshold in the future. The message 29 may be location-specific in the sense that it may be indicative of the region
5 in which the rule 28 has been satisfied. The message 29 may include an identifier for this region and/or identifiers for clients located in the respective region. In some implementations, the controlling entity 10 may use this information to adapt the content delivery in a location-specific manner, such that clients located in the region are affected by the adaptation, while clients located in other regions may not be affected.

10 In response to receiving the message 29, the controlling entity 10 may generate one control message or several control messages 30. The control message(s) 30 may be transmitted to any one, or any combination, of different execution points at which the adaptation of the content delivery is implemented.

15 The controlling entity 10 may control an execution point 20 to implement an adaptation of content delivery. The execution point 20 may be, or may be coupled to, a HTTP streaming server, for example. A bit rate at which a content source outputs content for a client located in a specific region may be adapted. Alternatively or additionally, subscription types of
20 different clients may be taken into account. For illustration, the bit rate may be selectively adapted for content delivery to clients which have a basic subscription. Alternatively or additionally, agreements between service providers which offer the content delivery and the operator of the mobile communication network 1 may be taken into account. For illustration, the adaptation may be done selectively based on how much of an agreed data volume a
25 service provider has already used.

Adapting the bit rate at which a content source outputs content for certain clients may be attained in various ways. For illustration, the execution point 20 may be a server which redirects streaming requests or other requests for content to different files which correspond
30 to different bit rates. The execution point 20 may be provided with identifiers for the affected clients, e.g. with Internet Protocol (IP) addresses of these clients. The controlling entity 10 then implements a filtering at the IP level, which selectively causes certain clients to receive content from a content source associated with a given bit rate. For further illustration, the execution point 20 may be a content source 24 which emulates plural virtual content servers
35 to provide the content at different bit rates. While different bit rates may be on a per client level, the adaptation is performed in a coordinated manner with the mobile communication network 1. The conditions in the mobile communication network 1 may thus be taken into

account more in a more complete manner, as compared to cases where the content delivery to a given client is adapted based only on a measurement of the RAN condition performed by that very client.

5 Alternatively or additionally, the controlling entity 10 may control a second execution point 21 to implement an adaptation of content delivery. The second execution point 21 may be, or may be coupled to, a Real-Time Transport Protocol (RTP) server 25. Such an adaptation may be performed by instructing clients to use tags in their requests for content. If the RAN in a region is likely to become congested, all or some of the clients located in that region may
10 be instructed to include a tag in their requests. The second execution point 21 may redirect requests received from clients, based on the tag, to one of plural different content sources which provide the requested content at different bit rates. In such an implementation, the controlling entity 10 filters, or redirects, the requests from different clients at the application level.

15 Alternatively or additionally, the controlling entity 10 may control a third execution point 22 to implement an adaptation of content delivery. The third execution point 22 may be, or may be coupled to, a server which transmits a manifest file. Such a manifest file may be used in streaming. The manifest file includes address information of the content source from which
20 the content may be received. A manifest file which is modified such that it includes the address of another content source that provides the requested content, but at a different streaming bit rate, may be transmitted to selected clients. Such a modified manifest file may be sent to some, or all, of the clients located in a region in which the RAN is likely to become congested.

25 Alternatively or additionally, the controlling entity 10 may control a fourth execution point 23 to adapt the content delivery. In this case, a Quality of Service (QoS) bearer adaptation 27 may be performed. The fourth execution point 23 may interface with the mobile communication network 1 to cause the QoS bearer to be adapted. The QoS bearer may be
30 adapted selectively for some clients. For illustration, the QoS bearer may be adapted for some, or all, of the clients located in a region in which the RAN is likely to become congested.

For each one of the various ways to adapt the content delivery, clients for which the content
35 delivery is to be adapted may be identified based on the geo information for clients of the mobile communication network. Additional information may be taken into account when deciding how and to which extent the control delivery is to be adapted. For illustration, a

decision on whether the content delivery bit rate for a given client is to be adapted, and the new value to which it is to be set, may additionally be based on the subscription type(s) of clients located in the affected region and/or on data budgets agreed between the service provider for the respective content delivery and the operator of the mobile communication network 1.

The client 6 to which the content is delivered provides a QoE report 31. The QoE report 31 may be automatically generated and transmitted by the client 6. Based on the QoE report 31, the controlling entity 10 may modify the rule which is used to evaluate the geo information for plural clients. The modified rule may be provided to the network entity 5. For illustration, if the rule involves a threshold comparison at which a change in bit rate is initiated, the threshold may be shifted based on QoE reports 31 received from different clients.

Figs. 3 and 4 illustrate areas covered by a mobile communication network. The mobile communication network includes plural RAN cells 41-45. A geo-messaging area 40 may be defined as union of plural RAN cells 41-45. The geo-messaging area 40 may be a region used for geo messaging, for example. The geo-messaging area 40 may include plural RAN cells 41-44. When a client enters or leaves a geo-messaging area, such as geo-messaging area 40, a geo messaging server is notified thereof. Such information may be used as geo information when controlling content delivery. While geo-messaging areas defined for geo-messaging in the mobile communication network may be used to identify geo information of clients, any other geo information may also be used to identify a region where the RAN is to be relieved.

Fig. 3 illustrates a state in which three clients 46, 48, 51 are located within the geo-messaging area 40. Clients may enter or leave the geo-messaging area 40. Other clients 47, 49, 50 are initially located in other RAN cells. Fig. 4 illustrates a state in which the number of clients located in the geo-messaging area 40 has increased. Clients 47, 48 have moved into the geo-messaging area 40. The risk that the RAN in cells 41-44 becomes overloaded increases with an increasing number of clients located in the geo-messaging area 40. By monitoring the number, or density, of clients in the geo-messaging area 40 over time, the network entity 5 may determine whether the number, or density, exceeds a threshold. The threshold may be selected to correspond to a number, or density, of clients at which the RAN cells 41-44 are not yet overloaded, when using the current content delivery setting. A potential congestion situation may be anticipated before it occurs. This allows a graceful degradation of the content delivery to be implemented, before a congestion situation in the

RAN severely impacts the quality of experience. The RAN in cells 41-44 may be relieved by adapting content delivery, such that the congestion situation is less likely to occur.

When an adaptation in content delivery is made to relieve the RAN, the adaptation may be made selectively for certain clients. For illustration, in response to determining that the number or density of clients in the geo-messaging area 40 reaches or exceeds a threshold, the content delivery may be selectively adapted for some, or all, of the clients 46-49, 51 located in the geo-messaging area 40 at that time. Different measures may be taken for different clients. For illustration, it is possible to account of the fact that the clients 46-49, 51 located in the geo-messaging region 40 may have different subscription types for the content delivery and/or that different service providers may have different agreements with the operator of the mobile communication network 1. The content delivery may first be degraded for clients having a basic subscription, for example.

Fig. 5 is a block diagram representation of a controlling entity 10. The controlling entity 10 may be used in the systems of Figs. 1 and 2, for example, in order to control content delivery to clients of a mobile communication network. The controlling entity 10 may be a server. In operation, the controlling entity 10 may be located outside the mobile communication network. The controlling entity 10 may be located in the backend, e.g. in the backbone of a wide area network such as the internet.

The controlling entity 10 generally comprises an interface 11 and a controller 12 coupled thereto. The controller 12 may comprise one or several processors which execute computer-executable instruction code to perform the various procedures described herein. When the controlling entity 10 is used in a system to control content delivery, the interface 11 is used for communication with a network entity of the mobile communication network. A rule may be output to the network entity of the mobile communication network over the interface 11, thereby programming the network entity to evaluate geo information of plural clients in accordance with the rule. A message from the network entity may be received at the interface 11. While illustrated as a functional unit in Fig. 5, the interface 11 may include separate interface units for outputting the rule to the network entity and for receiving a message from the network entity. The controlling entity 10 comprises a further interface 18 for outputting a control message, to thereby effect adaptation of the content delivery. While the further interface 18 may be a separate functional unit different from the interface 11, the functions of the interface 11 and of the further interface 18 may be performed by one physical interface unit. The controller 12 is configured to determine, based on the message received at the interface 11, how the content delivery is to be adapted in a location-

dependent manner. The controller 12 may process the message received from the network entity. The message may contain identifier for a region in which the RAN is congested or may become congested unless counter-measures are taken. Additionally or alternatively, the message may include an identifier for clients located in a region in which the RAN is congested or may become congested unless counter-measures are taken. The controller 12 may generate the control message based on this information and may output the control message via the further interface 18.

As already described above, the controlling entity 10 may be configured to control adaptation of the content delivery in any one of a variety of ways.

For illustration, the controller 12 may generate a control message and may output the control message to a server which processes requests for content, such as a HTTP server. The controller 12 may generate the control message such that it includes identifiers for clients located in an affected region. It is not required that the identifiers for all clients located in the affected region be included in the control message. For illustration, if a selective adaptation of the content delivery is performed based on service levels of different clients, the controller 12 may generate the control message such that it include identifiers of a sub-set of the clients located in the affected region, the sub-set being selected based on the subscription levels. The control message may include information on maximum content delivery rates which may be used by the various clients. In response to receiving the control message, the server which processes requests for content delivery may redirect requests to one of plural different streaming servers, based on the identifier of the requesting client. The different streaming servers may be virtual servers. Each one of the different streaming servers may respectively store a sequence of files representing the content, which are sequentially delivered to the clients. The sequences of files may correspond to different streaming rates. The controlling entity thus causes a redirection of requests from different clients located in the affected region to one of the plural different streaming servers, thereby causing an adaptation in the content delivery. The client identifiers which are received at the interface 11 and which are used to redirect requests coming from these clients located in an affected region may include internet protocol (IP) addresses of these clients.

Alternatively or additionally, the controller 12 may generate another control message such that it includes one or plural tags, each tag being respectively transmitted in association with a corresponding maximum bit rate at which content may be delivered to a client having the respective tag. The tag is a data piece which is used to redirect requests. There may be one tag for maximum bit rate, and another tag for another maximum bit rate. The controlling entity

10 may output such another control message to a server, for example to a streaming server, which redirects requests for content that include a tag to one of plural different streaming servers, based on the tag. For illustration, the controlling entity 10 may output this other control message which includes tags and associated maximum bit rates to an RTP server.

5 The controlling entity 10 may transmit a control message to instruct clients located in the affected region to include the tag in forthcoming requests for content. The controlling entity 10 may cause a server in the mobile communication network to transmit client control messages including the tag to clients located in the affected region. The clients which receive the client control messages may add the tag to their requests for content, until they receive
10 another client control message which includes another tag or which indicates that the tag does no longer need to be used. It is not required that identifiers of clients located in the affected region be provided to the controlling entity 10 in such a solution. Privacy may thus be enhanced.

15 Alternatively or additionally, the controller 12 may generate a control message to adapt a Quality of Service (QoS) bearer in the mobile communication network. The controller 12 may generate the control message informing a bearer manager in the mobile communication network of the modifications of guaranteed bit rates of QoS bearers. The control message may indicate an adaptation of the guaranteed bit rate, e.g. by reducing the guaranteed bit
20 rate. The guaranteed bit rate may be adapted differently for different clients located in the affected region. Different clients may receive different treatment based on their subscription types to the respective content delivery service, for example.

Alternatively or additionally, the controller 12 may generate a control message which causes
25 a modified manifest file to be transmitted to clients located in the affected region. To this end, the controller 12 may generate and transmit the control message to a service gateway which transmits a manifest file including information on a content source, such as an address of the content source from which data may be retrieved. The control message may include information on maximum bit rates for clients located in the affected region, and identifiers for
30 the respective clients. The information on the maximum bit rate may include the address of one or more streaming servers which provide the same content at different bit rates, respectively associated with identifiers for clients located in the affected region. The service gateway is operative to transmit, in response to a request from a client, a manifest file which includes information on a content source. When the controlling entity 10 transmits the control
35 message to the service gateway, the service gateway will send the modified manifest file which includes the address of a streaming server which provides the content at a reduced bit rate. The manifest file may be a manifest file according to the Dynamic Adaptive Streaming

of HTTP (DASH), as defined for example in 3GPP TS 26.247 v. 10.0.0. The transmission of a modified manifest file may also be done in accordance with a proprietary streaming technology, such as Adobe Systems' HTTP Dynamic Streaming, Apple Inc.'s HTTP Live Streaming (HLS) and Microsoft's Smooth Streaming. It will be appreciated that, in accordance with embodiments, the modified manifest file is transmitted in coordination with the mobile communication network, rather than "over the top" of the mobile communication network.

The controlling entity 10 may implement combinations of the various procedures described above. For illustration, the controller 12 may transmit several control messages, so as to both cause a modified manifest file to be transmitted and requests for content to be selectively redirected based on the client's IP address. A client-based approach in which a modified manifest file is transmitted to clients located in the affected region may thus support the approaches which are implemented at the network side, such as redirecting requests to one of plural streaming servers based on filtering performed based on IP addresses.

In each one of the various procedures performed by the controlling entity 10 to adapt the content delivery, the controlling entity 10 may control the degree to which a bit rate is adapted. For illustration, the controlling entity 10 may receive a message which indicates that the actual number of clients located in a region has a value $n_{s,is}$. The controller 12 may calculate a maximum allowed bit rate, $b_{s,target}$, for clients located in that region as

$$b_{s,target} < C_A / n_{s,is}, \quad (1)$$

where C_A indicates the available capacity for the respective region.

Different target bit rates may be assigned to different clients located in the affected region. In this case, the controller 12 may set the target bit rates for the clients such that

$$\sum_j (b_{s,target})_j < C_A. \quad (2)$$

Here, $(b_{s,target})_j$ denotes the maximum allowed bit rate for a client designated by index j located in the affected region, and the sum is computed over all indices j for clients located in the respective affected region. The bit rates may be assigned based on, for example, client subscription types. For illustration, the maximum allowed bit rate for clients which have a basic subscription may be decreased until equation (2) is fulfilled, while the maximum allowed bit rate for clients which have subscriptions other than the basic subscription may remain unaltered.

It will be appreciated that, although the bit rates may be assigned individually at a per client level, the content delivery is adapted based on the locations of plural clients. The bit rate set

for one client will depend on how many other clients are located in the respective affected region, in which counter-measures are taken to prevent congestions of the RAN. While the bit rate may be assigned individually for each client, the location-based adaptation of the content delivery still takes into account the overall number or density of clients in the affected region.

In each one of the various procedures performed by the controlling entity 10 to adapt the content delivery, the controller 12 may take into account additional information when determining whether the content delivery is to be adapted for clients located in a region. For illustration, the controlling unit 10 may comprise a storage unit 14 coupled to the controller 12. The storage unit 14 may store any one or any combination of various kinds of information which may be utilized to determine whether the content delivery is to be adapted for clients located in a specific region.

For illustration, topological information 15 of the mobile communication network 15 may be stored in the storage unit 14. Such topological information may be used when evaluating whether the content delivery needs to be adapted, taking into account that clients may be moving to regions which have higher capacities for delivering content. Referring to the exemplary RAN cells of Figs. 3 and 4, the topological information 15 may include information on the number of RAN cells 41-44 which make up a region 40 in which a congestion situation is anticipated to occur in the RAN.

Alternatively or additionally, a repository 16 of historical data may be stored in the storage unit 14. Such historical data may be statistical information, which may be collected in ongoing use of the controlling entity 10. The historical data may include long-term statistical information on regular congestions in certain regions. The storage unit 14 may also store information on the service and the subscription type of the various clients. This information may be used when performing a prioritization such that the bit rate for some clients is degraded to a lesser extent than the bit rate for other clients located in the same affected region.

The controlling unit 10 may also include additional interfaces for receiving yet other information that may be used when determining whether, and to what extent, content delivery is to be adapted. For illustration, when the clients are vehicles respectively equipped with an interface for communication with the wireless communication network, traffic information may be used when determining whether and to which extent the content delivery is to be adapted. The controlling entity 10 may include a traffic message receiver 19 for receiving traffic

messages. The controller 12 may be coupled to the traffic message receiver 19 and may be configured to anticipate congestion situations in the RAN based on the received traffic information. For illustration, the controller 12 may combine the traffic messages received at the traffic message receiver 19 with information on regions in which the number or density of clients has already exceeded a threshold, in order to determine whether the number of clients in an adjacent region is likely to increase to a value at which the RAN in that region is prone to being congested. For further illustration, the controller 12 may anticipate that a congestion is likely to occur in a certain region if there is a traffic jam in that region, as indicated by the traffic message.

While the controlling entity 10 may take steps to relieve the RAN in certain regions based on geo information of clients, the controlling entity 10 does not need to have direct access to the geo information of all clients of the mobile communication network. The geo information of the clients may be monitored by a network entity. The controlling entity 10 may output one or several rules for evaluating the geo information of the clients via the interface 11 to the network entity. The rules may include a threshold to which the number of clients in a region is compared by the network entity. The rules may include plural thresholds to which the number of clients in a region is compared by the network entity. When either one of the thresholds is reached, the controlling entity 10 receives the message from the network entity at the interface 11. Thereby, the controlling entity is alerted of a situation in which the bit rate of the content delivery to clients in the respective region is to be adapted.

The controlling entity 10 may be configured to receive QoE reports from the clients. The controlling entity 10 may receive the QoE reports via the interface 11. Alternatively, the controlling entity may have a separate interface for receiving the QoE reports. The controller 12 may evaluate the QoE reports. Based on the QoE reports, the controller 12 may adapt the rules for evaluating geo information of clients. For illustration, if QoE reports show frequent disruptions of service, the rule for evaluating the geo information of the clients may be adapted. This may involve decreasing a threshold which triggers the degradation in content delivery. The new rule is then output to the network entity via the interface 11. This allows the controlling entity 10 to take steps to relieve the RAN even earlier. The evaluation of QoE reports and the associated adaptation of the rules may be done in a location-specific manner, based on the region from which QoE reports are received. By adapting the rules which are used for evaluating the geo information, a closed-loop control may be implemented in which the adaptation of the content delivery may be improved further.

Fig. 6 is a flow chart of a procedure 60 which may be performed by the controlling entity 10, in order to control content delivery to clients of a mobile communication network.

At 61, a rule or several rules for evaluating geo information are output by the controlling
5 entity 10. The rule(s) may be output to a network entity of the mobile communication network. The rule(s) may be output to the network entity in a message which is transmitted to configure the network entity to monitor the geo information of clients based on the rule(s) and to provide a message based on the evaluation. When the rule(s) are adapted in a closed-loop control, the rule(s) may be output to the network entity to re-configure the network entity
10 such that it applies the rule(s) which have been modified based on QoE reports.

At 62, the controlling entity 10 receives a message from the network entity. The message, which is generated in dependence on a result of the evaluation of geo information of clients using the rule(s), may include an identifier for a region in which one of the rules is satisfied.
15 The identifier for the region may be an identifier for a geo-messaging area, an identifier for a RAN cell, or similar. The message may alternatively or additionally include identifiers for clients located in that region. The identifiers for clients may include IP addresses of the clients, which are used for delivering the content to the clients.

At 63, in response to receiving the message, the controlling entity 10 adapts the content
20 delivery. The adaptation may be performed in a location-dependent manner. The adaptation may include adapting bit rates at which content is delivered to clients located in the affected region. The adaptation may include computing a bit rate for content delivery to the clients located in the affected region, with the bit rate for at least one client being computed based
25 on the number of other clients which are present in the same affected region. The controlling entity 10 may output a control message or plural control messages to selectively adjust the delivery of content to the plural clients located in the affected region.

The controlling entity 10 may perform any one of a variety of procedures to cause the bit rate
30 of the content delivery to be adapted. For illustration, any one of the procedures described above may be used. When the message received at 62 includes identifiers for plural clients located in the affected region, the identifiers may be used to selectively relay requests for contents to one of plural different streaming servers which offer this content at different bit rates. Alternatively or additionally, transmission of tags and/or of a modified manifest file to
35 the plural clients located in the affected region may be performed. Alternatively or additionally, the controlling entity 10 may output a control message to the mobile communication network to adjust guaranteed bit rates of QoS bearers. In some of the

procedures which may be employed to relieve the RAN cells in the affected region, the controller 12 of the controlling entity may generate plural different control messages. For illustration, the controller 12 may generate one control message to effect transmission of tags or modified manifest files to clients located in the affected region. The controller 12 may
5 generate another control message to configure a HTTP streaming server, a RTP server, a streaming gateway server or another server which receives request for content delivery such that requests for content are selectively relayed to one of plural different streaming servers.

The procedure 60 may include additional steps. For illustration, QoE reports may be
10 received. The rule(s) may be automatically adapted based on the QoE reports. Adapting the rules may include adjusting threshold(s) to which a number or density of clients are compared. The procedure may then return to 61. For further illustration, the procedure 60 may include a step of updating historical data stored in a storage unit, based on message(s) received from the network entity. Thereby, reliability of the historical data may be enhanced
15 and the amount of data upon which the historical data is based may be increased in ongoing operation of the controlling entity.

Fig. 7 is a block diagram representation of a network entity 5. The network entity 5 may be used in the systems of Figs. 1 and 2, for example, in order to control content delivery to
20 clients of a mobile communication network. The network entity 5 may be operated by the operator of the mobile communication network. The network entity 5 may also perform additional functions, such as geo-messaging functions. Alternatively, the network entity 5 may be coupled to a geo-messaging server or another node of the mobile communication network which collects geo information of clients.

25 The network entity 5 comprises an interface 71, a processing unit 72 and a receiver 73. The interface 71 may be a programming interface, e.g. an API. The receiver 73 may be an interface configured for communication with another node of the mobile communication network. The processing unit 72, which may include one or several processors or dedicated
30 circuits, is coupled to the interface 71 and to the receiver 73.

Geo information of plural clients of the mobile communication network is received at the receiver 73. The geo information may be actively reported by the plural clients, for example when the network entity 5 also servers as a geo messaging server. The network entity may
35 also query another node of the mobile communication network to receive the geo information of the plural clients at the receiver 73. The processing unit 72 evaluates the received geo information based on a rule or based on plural rules. The processing unit 72 may perform

additional processing on the received geo information. For illustration, the processing unit 72 may be configured to update a data base 75 stored in a storage unit 74 based on the received geo information. The data base 75 may define a mapping between clients and regions. For illustration, if the geo information indicates that a client has traversed a boundary between geo messaging areas, the processing unit 72 may update the data base 75 accordingly.

The geo information received at the receiver 73 may have any one of a variety of formats. For illustration, the geo information may include an identifier for a RAN cell in which the respective client is located. The geo information may alternatively or additionally include an identifier for a geo messaging area, which may cover more than one RAN cell. The geo information may alternatively or additionally include geo coordinates.

The rule(s) which are used by the processing unit 72 may be injected via the interface 71. In operation, the rule(s) may be injected from a controlling entity which is located outside the mobile communication network. For illustration, in some implementations, one threshold or at least two thresholds may be provided to the network entity 5 via the interface 71. The network entity 5 may respectively compare, for each one of plural different regions, the number of clients to the threshold(s). Different threshold(s) may be injected for different regions, e.g. for different geo messaging areas. The rule(s) may be more complex. For illustration, the rule(s) may include a criterion for outputting a message which is not only based on the present number of clients in a region, but which is also based on a rate of change of that number. The rule(s) may include an executable instruction code which is executed by the processing unit 71 to evaluate the geo information of the clients. The rule(s) may include parameters for an executable instruction code which is stored in the network entity 5. In the latter case, the controlling entity may configure the network entity such that it performs a pre-defined procedure to evaluate the geo information of the plural clients, while using the parameters injected by the controlling entity. The parameters may include one or plural threshold(s) to which the number of clients in a region is compared.

The processing unit 72 is configured to evaluate the geo information of the plural clients based on the rule(s) received at the interface 71. For illustration, the processing unit 72 may determine the number of clients respectively located in each one of plural different areas covered by the mobile communication network. The processing unit 72 may compare this number to an associated threshold, or to a pair of associated threshold. The processing unit 72 may be configured to predict a region which is affected by a congestion situation, now or in the future, by evaluating the geo information of the plural clients based on the rule(s).

Based on a result of the threshold comparison, the processing unit 72 may selectively generate a message and may output the message to the controlling entity. The message may be output over the interface 71. For illustration, a message may be generated and output when a number of clients of the mobile communication network located in an area exceeds a first threshold. Another message may be generated and output when a number of clients of the mobile communication network located in an area falls below a second threshold. The controlling entity is thereby informed that, based on the rules previously injected, a state is reached in which measures are to be taken to avoid a congestion in the RAN in an identified region. Alternatively, the controlling entity may be informed that such measures to adapt the content delivery in a location-specific manner which have previously been taken may be relieved, because the number of clients in that region has decreased sufficiently that there is a smaller risk of congestion in the RAN occurring even if content is delivered to the clients at higher bit rates. The message may be an alert, generated and transmitted to inform the controlling entity that content delivery should be adapted in a location-specific manner.

The evaluation of the geo information based on the rule(s) includes identifying the region for which the content delivery is to be adapted. This may be done in various ways. For illustration, the processing unit 72 may determine an identifier for the region in which the content delivery is to be adapted. The processing unit 72 may be configured to output such an identifier for the affected region in the message transmitted to the network entity. Alternatively, the processing unit 72 may determine identifiers for clients located in the region in which the content delivery is to be adapted. The processing unit 72 may access a storage unit 74 to determine clients located in the region, using a data base 75 which specifies the region in which a client is located for plural client identifiers. The client identifiers may, for example, be IP addresses of the clients which are used when content is delivered to the clients. The client identifiers may also be other client identifiers, and the processing unit 72 may be configured to map the client identifiers onto associated IP addresses of the clients which are used when content is delivered to the clients.

The network entity 5 is operative to provide information on anticipated congestion situations in the RAN to the controlling entity located outside of the mobile communication network. Identifiers for the respective clients may, but do not need to be output to the controlling entity. For illustration, the processing unit 72 of the network entity 5 may selectively generate a message based on the evaluation of the geo information using the rule(s), the message including an identifier for the region in which the content delivery is to be adapted. Such information may be sufficient for the controlling entity to adapt content delivery in a location-

selective manner. For illustration, when tags or modified manifest files are transmitted to clients located in the respective area, it may not be required for the controlling entity to be actually informed of identifiers for the clients. Enhanced privacy may thus be attained. In other implementations, the processing unit 72 of the network 5 entity may selectively
5 generate a message based on the evaluation of the geo information using the rule(s), the message including identifiers for clients located in the region in which the content delivery is to be adapted. The controlling entity may use client identifiers included in the message to relay requests for content to one of plural different streaming servers based on the IP address of the requesting client. Alternatively or additionally, the controlling entity may use
10 client identifiers included in the message to selectively configure a streaming gateway or streaming server such that a request for content is selectively directed to one of plural different streaming servers based on the IP address of the requesting client. The bit rate may thus be selectively adjusted for clients located in the affected region.

15 Fig. 8 is a flow chart of a procedure 80 which may be performed by the network entity 5, in order to inform a controlling entity that content delivery to clients of a mobile communication network is to be adapted

At 81, a rule or several rules for evaluating geo information are received by the network entity
20 5. The rule(s) may be received from a controlling entity located outside of the mobile communication network. The rule(s) may be received in a message, which configures the network entity to monitor the geo information of clients based on the rule(s) and to provide a message based on the evaluation.

At 82, the network entity 5 collects geo information for plural clients of the mobile communication network. The geo information for the plural clients may be collected even before the rule(s) are received. The geo information for the plural clients may be collected using information transmitted by the clients to a geo messaging server. The geo information for the plural clients may be collected by querying a node of the mobile communication
30 network which is aware of the location of the clients.

At 83, the geo information for the plural clients is evaluated based on the rule(s). As explained above, the rules may include threshold(s) which are used to evaluate the geo information. The evaluation of the geo information based on the rule may include determining
35 respectively an indicator for a number of clients for each one of plural different regions. The evaluation of the geo information based on the rule may include comparing the indicator for the number of clients located in a region to a threshold. The evaluation of the geo information

based on the rule may include comparing the indicator for the number of clients to both a first threshold and a second threshold. More complex evaluations may be performed. For illustration, not only the instant number of clients per region, but also the rate of change of the number of clients per region may respectively be evaluated for plural different regions.

5 The evaluation may include predicting a region in which a congestion situation may occur in the RAN, based on the number of clients located in that region and, optionally, adjacent regions.

10 At 84, it is determined whether an alert criterion is fulfilled. The alert criterion may be fulfilled when the number of clients in a region reaches or exceeds an associated threshold. The alert criterion may be fulfilled when the number of clients in a region either one of exceeds a first threshold or falls below a second threshold. If the alert criterion is not fulfilled, the procedure returns to 82. If the alert criterion is fulfilled, the procedure continues at 85.

15 At 85, the processing unit 72 of the network entity 5 generates a message. The message is output to the controlling entity. The message may include an identifier for the region for which a congestion situation is predicted based on the evaluating at 84. The message may include identifiers of clients which are located in the region for which a congestion situation is predicted. The procedure then returns to 82.

20

If new rule(s) are received by the network entity, the procedure may return to 81. The evaluation of the geo information is then continued using the new rules.

25 With reference to Figs. 9-12, signal flows in methods of controlling content delivery and corresponding systems according to embodiments will be explained. Devices or messages already explained in the context of Figs. 1-8 are designated with the same reference numerals.

30 Fig. 9 is a block diagram of a system illustrating a signal flow in accordance with embodiments. The controlling entity 10 controls an execution point 20 which relays or redirects requests for content to one of plural different streaming servers based on IP addresses of the requesting clients. The execution point 20 may be any server which receives requests for content, e.g. a HTTP server. The controlling entity 10 acts as a decision point operative to filter requests for content based on IP addresses, causing the
35 requests for content to be selectively redirected to one of plural different streaming servers.

The controlling entity 10 transmits one or several rule(s) 28 to the network entity 5. The network entity 5 collects geo information of plural clients to which content is delivered, and evaluates the collected geo information based on the rule(s) 28. Based on a result of the evaluation, a message 91 is transmitted to the controlling entity 10. The message 91 includes identifiers of clients which are located in an affected region. The message 91 may be transmitted before the RAN in the affected region becomes congested. The congestion situation may be anticipated using geo information of the clients.

Receipt of the message 91 triggers the controlling entity 10 to decide whether the content delivery to at least some of the clients located in the affected region is to be adapted, and for which clients located in the affected region the content delivery is to be adapted. For illustration, the controlling entity 10 may assign target bit rates to clients located in the affected region in accordance with one of Equations (1) and (2). The controlling entity 10 transmits a control message 92 to the execution point 20. The control message 92 may include identifiers for the clients located in the affected region and the associated target bit rates. It is not required that identifiers for all clients located in the affected region be included in the control message 92. If the controlling entity 10 decides that content delivery is to be degraded for only a sub-set of the clients located in the affected region, the control message 92 may be generated to include only the identifiers for these clients and the associated target bit rates. Using the control message 92, the controlling entity 10 configures the execution point 20 such that requests for content are selectively relayed or redirected to one of plural streaming servers 96-98, depending on the IP address of the requesting client. The execution point 20 may store the IP addresses of the clients and associated target bit rates for subsequent processing of requests.

The content source 95 may include the streaming servers 96-98. The streaming servers 96-98 do not need to be physically different entities. The streaming servers 96-98 may be implemented by one physical server which is configured to provide the same content at different streaming rates. If the content source 95 is configured to only provide content at a given, pre-defined bit rate, the controlling entity 10 or execution point 20 may send another control message to the content source 95, thereby requesting that the content be provided at plural different bit rates. Corresponding virtual streaming servers may be set up in the content source 95 in response to such a request.

When a client 6 transmits a request 93 for content, the execution point 20 determines the target bit rate for the client 6. The execution point 20 may select one of the plural streaming servers 96-98 based on the target bit rate for the client 6. For illustration, the execution point

20 may select one of the plural streaming servers 96-98 which offers the content with a bit rate which is less than or equal to the target bit rate for this client. If there are plural such streaming servers 96-98, the streaming server which offers the content with the greatest bit rate less than or equal to the target bit rate may be selected. The execution point 20 relays or
5 redirects the request to the corresponding streaming server 96-98.

Fig. 10 is a block diagram of a system illustrating a signal flow in accordance with embodiments. The controlling entity 10 causes a control message 103' to be transmitted to all clients located in an affected area. This control message 103' may include a tag which the
10 clients will include in their requests for content. The controlling entity 10 also controls an execution point 20 such that the execution point 20 redirects requests for content to one of plural streaming servers 96-98, based on the tag included in the respective request for content. The controlling entity 10 may act as a filter on the application level in this implementation.

15 The network entity 5 collects geo information of plural clients to which content is delivered, and evaluates the collected geo information based on the rule(s) 28. Based on a result of the evaluation, a message 101 is transmitted to the controlling entity 10. The message 101 may include an identifier for an affected region. The message 101 may further include information
20 on a number of clients located in that region. However, as the rule(s) 28 have previously been set by the controlling entity 10, it is not required that the number of clients located in the affected region is indicated in the message 101. The controlling entity 10 may use information on previously output rule(s) to determine the number of clients in the affected region which has triggered transmission of the message 101. The message 101 may, but
25 does not need to include identifiers of clients which are located in an affected region. The message 101 may be transmitted when a congestion situation in the RAN is predicted.

Receipt of the message 101 triggers the controlling entity 10 to decide whether the content delivery to clients located in the affected region is to be adapted. For illustration, the
30 controlling entity 10 may assign a target bit rate to clients located in the affected region according to one of Equations (1) and (2). The controlling entity 10 transmits a configuration message 102 to the execution point 20. The execution point 20 may be any server which receives requests for content, e.g. a streaming gateway or an RTP server. The configuration message 102 includes information on a tag and an associated target bit rate. The execution
35 point 20 is configured to redirect requests for content which include the tag to a streaming server 96-98 which offers the content at a bit rate which best matches the target bit rate. For illustration, the execution point 20 may redirect requests for content to a streaming server

which offers the content at a streaming rate which is less than the target bit rate, as indicated by the target bit rate assigned to the tag contained in the request.

5 Receipt of the message 101 also triggers controlling entity 10 to transmit a control message with the tag to the clients which are located in the affected region. The controlling entity 10 may transmit a control message 103 including the tag to the network entity 5. The network entity may relay this control message 103 as control message 103' to the clients located in the affected region. In other implementations, the controlling entity 10 may issue the control message 103 with the tag to a node of the mobile communication network 1 which is different
10 from the network entity 5. This other node may then transmit the tag to the clients located in the affected region. Geo-messaging techniques may be used to transmit the tag to the clients located in the affected region.

When a client 6 transmits a request 104 for content, the tag is included in the request 104.
15 The execution point 20 determines the target bit rate for the client 6 based on the tag, using the information included in the configuration message 102. The execution point 20 may select one of the plural streaming servers 96-98 based on the target bit rate for the client 6. For illustration, the execution point 20 may select one of the plural streaming servers 96-98 which offers the content with a bit rate which is less than or equal to the target bit rate for this
20 client. If there are plural such streaming servers 96-98, the streaming server which offers the content with the greatest bit rate which is still less than or equal to the target bit rate may be selected. The execution point 20 relays or redirects the request to the corresponding streaming server 96-98.

25 Fig. 11 is a block diagram of a system illustrating a signal flow in accordance with embodiments. The controlling entity 10 controls an execution point which is a bearer manager 115 of the mobile communication network 1. Thereby, the guaranteed bit rate may be adapted for QoS bearers which transport content to clients located in an affected region.

30 The controlling entity 10 transmits one or several rule(s) 28 to the network entity 5. The network entity 5 collects geo information of plural clients to which content is delivered, and evaluates the collected geo information based on the rule(s) 28. Based on a result of the evaluation, a message 111 is transmitted to the controlling entity. The message 111 includes identifiers of clients which are located in an affected region. The message 111 may be
35 transmitted before the RAN in the affected region becomes congested. The congestion situation may be anticipated using geo information of the clients.

Receipt of the message 111 triggers the controlling entity 10 to decide whether the content delivery to at least some of the clients located in the affected region is to be adapted, and for which clients located in the affected region the content delivery is to be adapted. For illustration, the controlling entity 10 may assign target bit rates to clients located in the affected region according to one of Equations (1) and (2). The controlling entity 10 transmits a control message 112 to the bearer manager 115. The control message 112 may include identifiers for the clients located in the affected region and the associated target bit rates. It is not required that identifiers for all clients located in the affected region be included in the control message 112. The controlling entity 10 configures the bearer manager 115 such that the guaranteed bit rates for the QoS bearers are adapted. When content is delivered to a client 6, the guaranteed bit rate for the QoS bearer 117 through which the content is delivered may be set based on the control message 112. This leads to an adaptation of the bit rate of data transmission 116 to the client 6.

Embodiments of the invention are not limited to the signal flows explained with reference to Figs. 9-11. For illustration, the controlling entity 10 may cause a manifest file to be modified and to be transmitted to clients located in an affected region of the mobile communication network 1. For further illustration, the various procedures may be combined with each other to control content delivery.

While only one mobile communication network 1 is illustrated in Figs. 1, 2 and 9-11, the controlling entity 10 may interface with plural different mobile communication networks to control content delivery.

Fig. 12 illustrates a signal flow in methods and systems of an embodiment. The system includes plural mobile communication networks, which may include a first mobile communication network 1 and a second mobile communication network 121. Content delivery to clients of the first mobile communication network 1 may be controlled as explained with reference to any one of Figs. 9-11, for example. Content delivery to clients of the second mobile communication network 121 may be similarly controlled using the procedures and signal flows explained with reference to any one of Figs. 9-11, for example.

The controlling entity 10 may inject rules 28 to the network entity 5 of the first mobile communication network 1, and may inject rules 128 to a network entity 125 of the second mobile communication network 121. The controlling entity 10 may select different rules for different mobile communication networks, i.e. for different operators. The rules may respectively be selected based on a service agreement between the service provider which

offers the content delivered to clients and the operators of the first mobile communication network 1 and the second mobile communication network 121, respectively. The rules may respectively be selected based on data volumes which have already been used for delivering content over the first mobile communication network 1 and the second mobile communication network 121, respectively, as compared to agreed data volumes that may be used in a pre-defined time interval. For illustration, the controlling entity 10 may set different thresholds to which the number of clients in a certain region is to be compared for the first mobile communication network 1 and for the second mobile communication network 121.

In response to receiving a message 91 from the first mobile communication network 1 or in response to receiving a message 129 from the second mobile communication network 121, the controlling entity 10 may decide on how the content delivery is to be adapted and/or by how much the bit rates are to be adjusted. The controlling entity 10 controls an execution point 20 to effect the adaptation in content delivery.

Fig. 13 is a schematic view illustrating adaptation of content delivery based on a number or density of clients in accordance with embodiments. A spatially selective adaptation of the content delivery, e.g. a reduction or increase in bit rate, may be performed by monitoring an indicator for the number of clients in a region, respectively for plural different regions. The clients may be vehicles equipped with an interface for communication with the mobile communication network. The indicator may be the number of clients, the density of clients, or another quantity which is a monotonous function of the number of clients. The indicator 130 is shown as a function of time. The indicator 130 may be compared to a first threshold 131 and to a second threshold 132. The first threshold 131 and the second threshold 132 may be set by the controlling entity 10.

When the indicator 130 exceeds the first threshold 131 at a first time, T1, 134, the network entity of the respective mobile communication network may output a message to the controlling entity. This causes the controlling entity to take measures to relieve the RAN in the respective region. When the indicator 130 falls below the first threshold at a second time, T2, 135, it is not required that a message be output to the controlling entity, and the controlling entity does not need to take any action. When the indicator 130 exceeds the second threshold 132 at a third time, T3, 136, the network entity of the respective mobile communication network may output a message to the controlling entity. This causes the controlling entity to adapt the bit rates for clients located in the respective region such that content delivery at higher bit rates is again allowed. In response to the message transmitted at the first time T1, the controlling entity implements a bit rate reduction for clients located in

the respective region at the first time T1. This degradation is continued for a period 137. Subsequent to the third time T3, the controlling entity takes measures which allow the content to be delivered again at higher bit rates. This reflects that the RAN is less likely to be congested in period 138.

5

Various effects may be attained using methods, devices and systems of embodiments. According to embodiments, the adaptation of the content delivery is coordinated with the mobile communication network. This allows a graceful adaptation of the content delivery when load peaks occur, e.g. due to a high vehicle density caused by a traffic jam. With the adaptation being performed in a location-specific manner for regions in which an overload situation is likely to occur, unnecessary degradation in other areas may be avoided. For clients located in the affected region, the adaptation of the content delivery may prevent disruptions of service or other severe negative impacts on the perceived quality.

10

15

For further illustration, when the adaptation of the content delivery is coordinated with the mobile communication network, contract frameworks between service providers and network operators may be taken into account. Such contract frameworks may foresee a limited amount of data to be transmitted, be it due to the data distribution agreement itself or to more sophisticated agreements such as selective prioritization of traffic. When there is a danger that the data volume could be exceeded, the service provider has the option to react to such a situation in a flexible way, e.g. by prioritization of clients based on their subscription. Enhanced versatility is attained as compared to approaches where rate adaptation is executed only at an individual session and user level. By considering the perspective of the content provider, an agreed capacity budget can be kept also in high load situation.

20

25

It is to be understood that the examples and embodiments as explained above are merely illustrative and susceptible to various modifications. For illustration, in each embodiment and aspect described herein, the clients may be vehicles. Methods, devices and systems of embodiments are not limited thereto and may also be used for other clients of mobile communication networks. For further illustration, in each embodiment and aspect described herein, the content delivery may be audio streaming. Methods, devices and systems of embodiments are not limited thereto and may also be used for video streaming, media streaming, multimedia streaming. The streaming may respectively be live streaming or on demand. Methods, devices and systems of embodiments may also be used when content is delivered by download. For further illustration, in each embodiment and aspect described herein, the controlling entity may be configured to perform plural different procedures to put the adaptation in content delivery into effect. The controlling entity may be configured to

30

35

determine, in response to receiving a message from the network entity, which one of the different procedures or which combination of these different procedures should be used. For further illustration, in each embodiment and aspect described herein, the controlling entity may be a server outside of the mobile communication network. The controlling entity may be
5 a dedicated server for controlling the functions described in the context of embodiments, but may also be integrated in other servers. For further illustration, in each embodiment and aspect described herein, the network entity may be configured to specifically monitor the geo information of clients to which a specific content is delivered, e.g. clients which receive a specific audio stream. However, in each embodiment and aspect described herein, the total
10 number of clients located in a region may be used when performing the evaluation of geo information in the network entity. The total number of clients interested in content delivery will generally scale with the total number of clients, so that the total number of clients may be used when determining whether an adaptation of the content delivery is to be made. For further illustration, in each embodiment and aspect described herein, the network entity may
15 be a server which is interfaced with a node of the mobile communication network. The network entity may be a geo cast server of a geo cast service, for example, or may be coupled to a geo cast server. The geo cast server may receive information on tiles of the geo cast service in which the clients are located. For further illustration, in each aspect and embodiment described herein, the controlling entity may be a dedicated server for controlling
20 the functions described in the context of embodiments, but may also be integrated in other servers. For illustration, the controlling entity may be integrated into a gateway or server of the content provider which offers the content delivered to the clients. Further, it is to be understood that the above concepts may be implemented by using correspondingly designed software to be executed by a processor of an existing device, or by using dedicated device
25 hardware.

Claims

1. A method of controlling content delivery to plural clients (6; 46-51; 6, 126) of a mobile communication network (1), the content being delivered to the plural clients (6; 46-51; 6, 126) over a radio access network of the mobile communication network (1), the method comprising:

collecting, by a network entity (5; 125) associated with the mobile communication network (1), geo information of the plural clients (6; 46-51; 6, 126);

evaluating, by the network entity (5; 125), the collected geo information of the plural clients (6; 46-51; 6, 126) based on a rule to determine whether the content delivery is to be adapted and to identify a region (40) for which the content delivery is to be adapted;

selectively outputting, by the network entity (5; 125), a message (9; 29; 91; 101; 111; 129) to a controlling entity (10) if it is determined that the content delivery is to be adapted;

controlling, by the controlling entity (10), the content delivery in response to receiving the message (9; 29; 91; 101; 111; 129), to adapt the content delivery for clients (46-49, 51) located in the identified region (40).

2. The method of claim 1,

wherein, in order to adapt the content delivery for the clients (46-49, 51) located in the identified region (40), the controlling entity (10) generates a control message (92; 103; 112) to at least one of:

- adapt a content delivery rate at which a content source (95) provides content to the clients (46-49, 51) located in the identified region (40); or
- perform a quality of service (QoS) bearer (117) adaptation in the mobile communication network (1) over which content is delivered; or
- prioritize clients (46-49, 51) located in the identified region (40) based on subscription information.

3. The method of claim 2,

wherein the controlling entity (10) selects a content delivery rate at which a content source (95) delivers content to the clients in the region (40) based on the received message (91; 101), and generates the control message (92; 103) based on the selected content delivery rate.

4. The method of claim 3,

wherein different content delivery rates are selected for at least two different cells (41, 45) of the radio access network.

5. The method of claim 3 or 4,
wherein the controlling entity (10) transmits the control message (92) to an execution point (20), and wherein, in response to receiving the control message (92), the execution point
5 (20) directs a request (93) for content coming from a client (6) located in the identified region (40) to a content server (96-98) of the content source (95) which is associated with the selected content delivery rate.

6. The method of claim 3 or 4,
10 wherein the controlling entity (10) sends the control message (103) to the clients (46-49, 51) located in the identified region (40), and wherein, in response to receiving the control message (103), the clients (46-49, 51) located in the identified region (40) request the content from a content server (96-98) of the content source (95) which is associated with the selected content delivery rate.

15 7. The method of any one of the preceding claims,
wherein the evaluating comprises: predicting a region (40) affected by a congestion situation in the radio access network based on the collected geo information of the plural clients (6; 46-51; 6, 126),
20 wherein the controlling entity (10) controls the content delivery to relieve the radio access network in the identified region (40).

8. The method of any one of the preceding claims, further comprising:
providing, by the controlling entity (10), the rule to the network entity (5; 125).

25 9. The method of claim 8, further comprising:
receiving, by the controlling entity (10), quality reports (31) from the plural clients (6; 46-51; 6, 126) to which the content is delivered,
wherein the controlling entity (10) modifies the rule based on the received quality reports
30 (31).

10. A controlling entity configured to control content delivery to plural clients (6; 46-51; 6, 126) of a mobile communication network (1), the controlling entity (10) comprising:
an interface (11) for communication with a network entity (5; 125) of the mobile
35 communication network (1); and
a controller (12) configured

- to output a rule for evaluating geo information of the plural clients (6; 46-51; 6, 126) to the network entity (5; 125) over the interface (11),
- to receive a message (9; 29; 91; 101; 111; 129) from the network entity (5; 125) over the interface (11), the message (9; 29; 91; 101; 111; 129) containing information on a location-dependent adaptation of the content delivery, and
- to adapt the content delivery in a location-dependent manner for clients (46-49, 51) located in a region (40) based on the received message (91; 101), which region (40) is determined based on the information on a location-dependent adaptation.

10 11. The controlling entity of claim 10,
wherein the controller (12) is configured to generate a control message (92; 103) based on the message (9; 29; 91; 101) to adapt a content delivery rate at which a content source (95) provides content to the clients (46-49, 51) located in the region (40).

15 12. The controlling entity of claim 11,
wherein the information comprises addresses of the clients (46-49, 51) located in the region (40), and the controller (12) is configured to generate the control message (92) based on the addresses.

20 13. The controlling entity of claim 12,
wherein the controller (12) is configured to send the control message (92) comprising at least a subset of the addresses to an execution point (20) which receives a client request (93) for content delivery.

25 14. The controlling entity of claim 11 or claim 12,
wherein the information comprises a region identifier for the region (40), and the controller (12) is configured to send the control message (103) to the clients (46-49, 51) located in the region (40).

30 15. The controlling entity of claim 14,
wherein the controller (12) is configured to generate the control message (103) such that it includes at least one of a tag and a command to transmit a manifest file, generated in dependence on a content delivery rate selected based on the received message (91; 101).

35 16. The controlling entity of any one of claims 10-15,

wherein the controller (12) is configured to generate a further control message (112) based on the message (9; 29; 111) and to output the further control message (112) to the mobile communication network (1) to perform a Quality of Service (QoS) bearer (117) adaptation.

5 17. The controlling entity of any one of claims 10-16, wherein the controller (12) is configured to adapt the content delivery based on subscription types of the clients (46-49, 51) located in the region (40).

10 18. The controlling entity of any one of claims 10-17, wherein the controller (12) is configured to evaluate quality reports (31) received from the plural clients (6; 46-51; 6, 126) to which the content is delivered, to adapt the rule based on the received quality reports (31), and to output the adapted rule to the network entity (5; 125) over the interface (11).

15 19. A network entity for a mobile communication network (1), the network entity (5; 125) comprising:
a receiver (73) configured to receive geo information of plural clients (6; 46-51; 6, 126) of a mobile communication network (1);
an interface (71) configured to receive a rule from a controlling entity (10) controlling content
20 delivery to the plural clients (6; 46-51; 6, 126);
a processing unit (72) configured to
- evaluate the received geo information of the plural clients (6; 46-51; 6, 126) based on the received rule to determine whether the content delivery is to be adapted and identify a region (40) for which the content delivery is to be adapted, and
25 - selectively output a message (9; 29; 91; 101; 111; 129) over the interface (71) if it is determined that the content delivery is to be adapted.

20. The network entity of claim 19, wherein the processing unit (72) is configured to predict regions affected by a congestion
30 situation in a radio access network of the mobile communication network (1) based on the collected geo information of the plural clients (6; 46-51; 6, 126) to identify the region (40), and to selectively output the message (9; 29; 91; 101; 111; 129) based on the predicted regions.

35 21. The network entity of claim 20, wherein the processing unit (72) is configured to respectively compare, for each one of a plurality of regions, an indicator (130) indicative of a number of clients (46-49, 51) located in

the region (40) to at least one threshold (131, 132) to predict the regions affected by the congestion situation.

22. The network entity of any one of claims 19-21,

5 wherein the processing unit (72) is configured to determine addresses of the clients (46-49, 51) located in the region (40) and to output the message (9; 29; 91; 129) including the determined addresses.

23. The network entity of any one of claims 19-22,

10 wherein the processing unit (72) is configured to output the message (9; 29; 101; 111) including an identifier for the region (40).

1/9

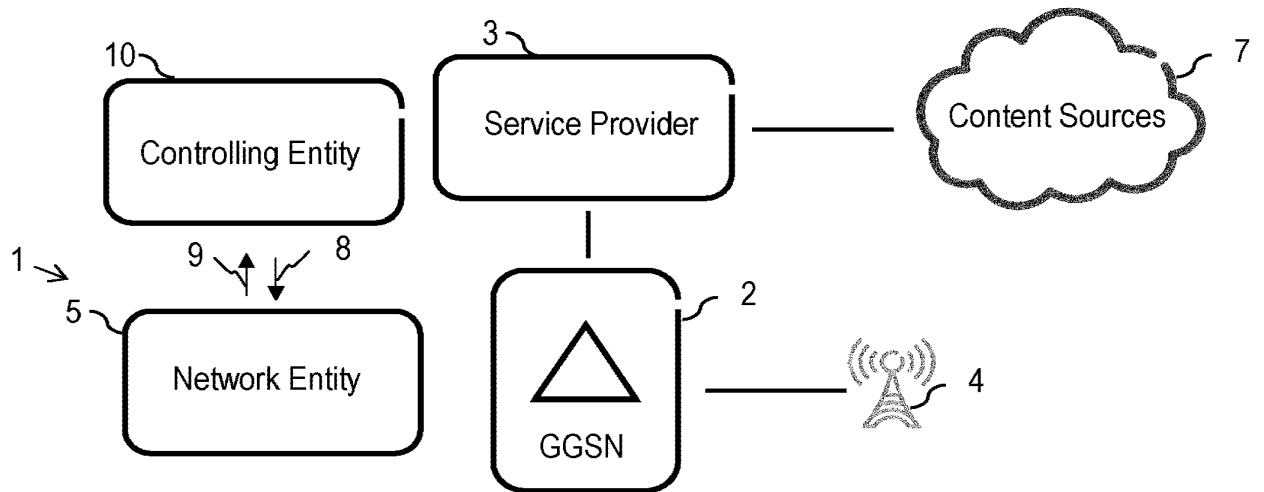


FIG. 1

2/9

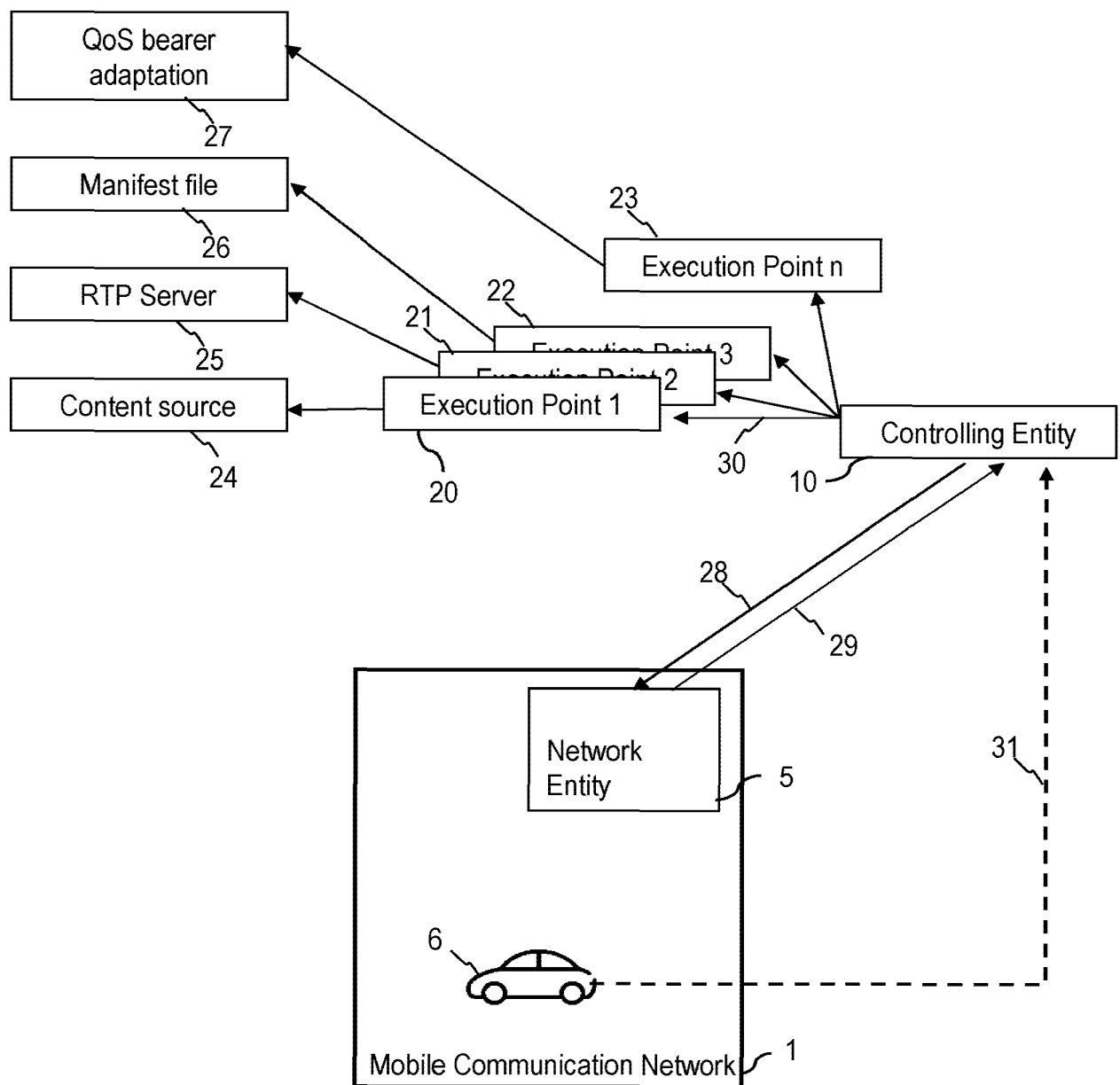
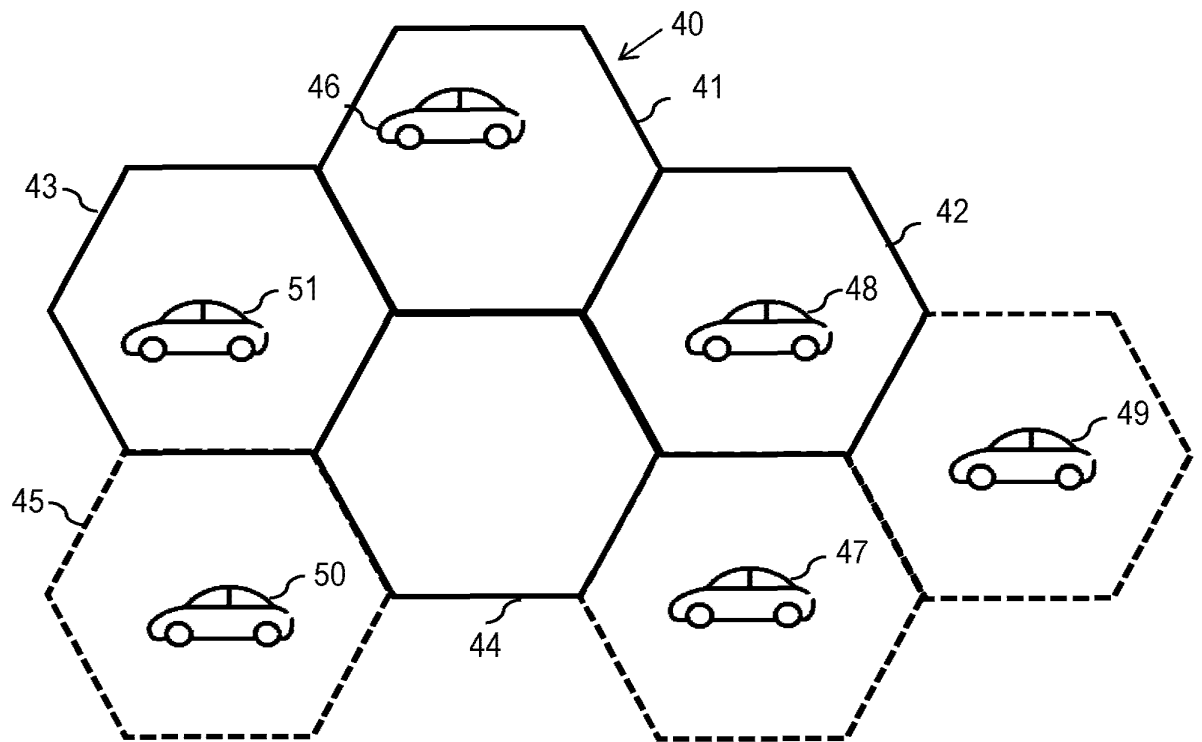
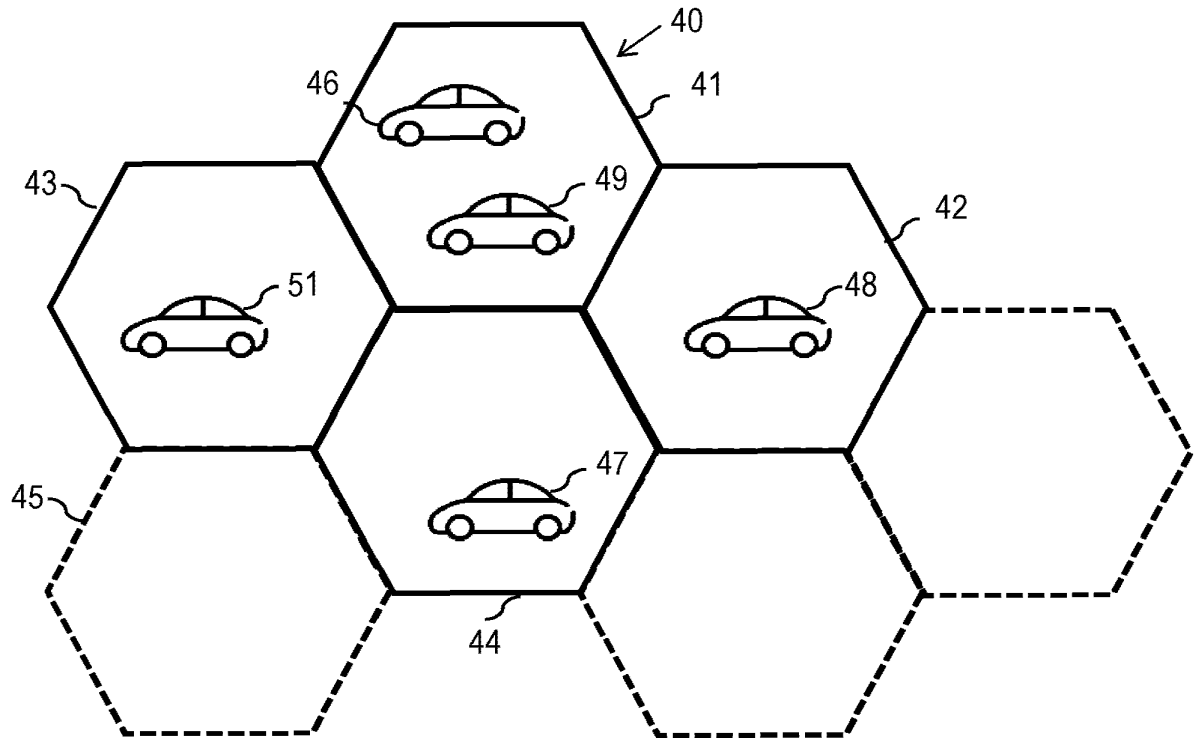


FIG. 2

3/9



4/9

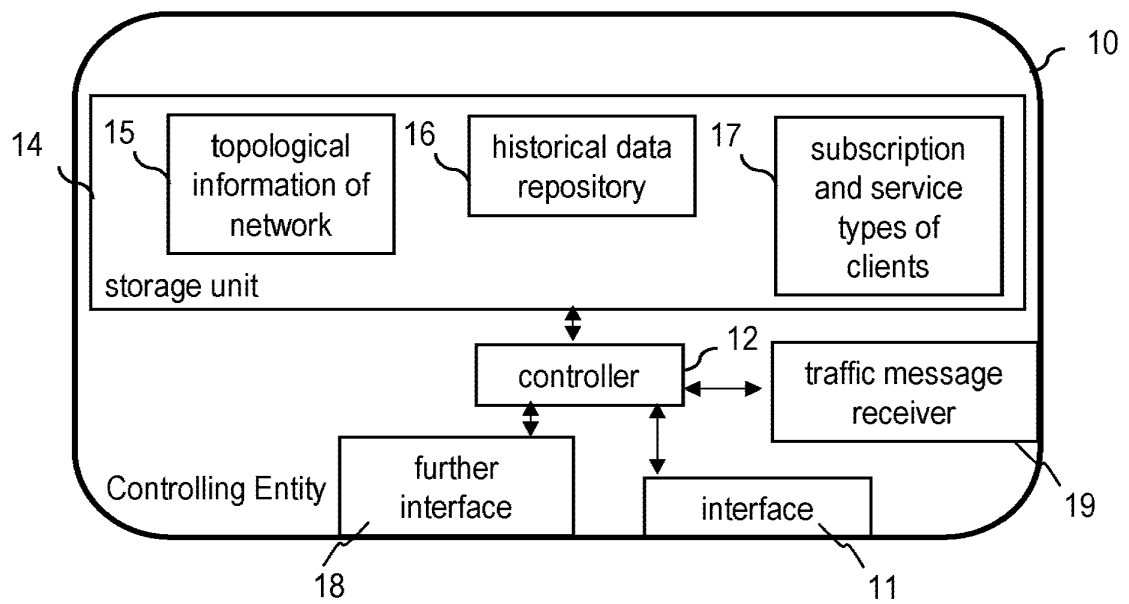


FIG. 5

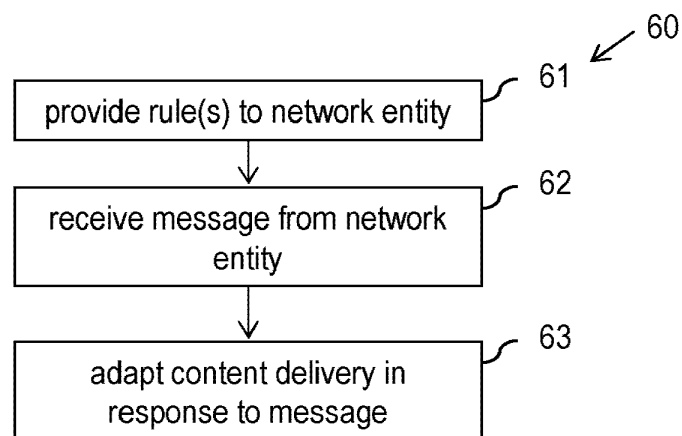


FIG. 6

5/9

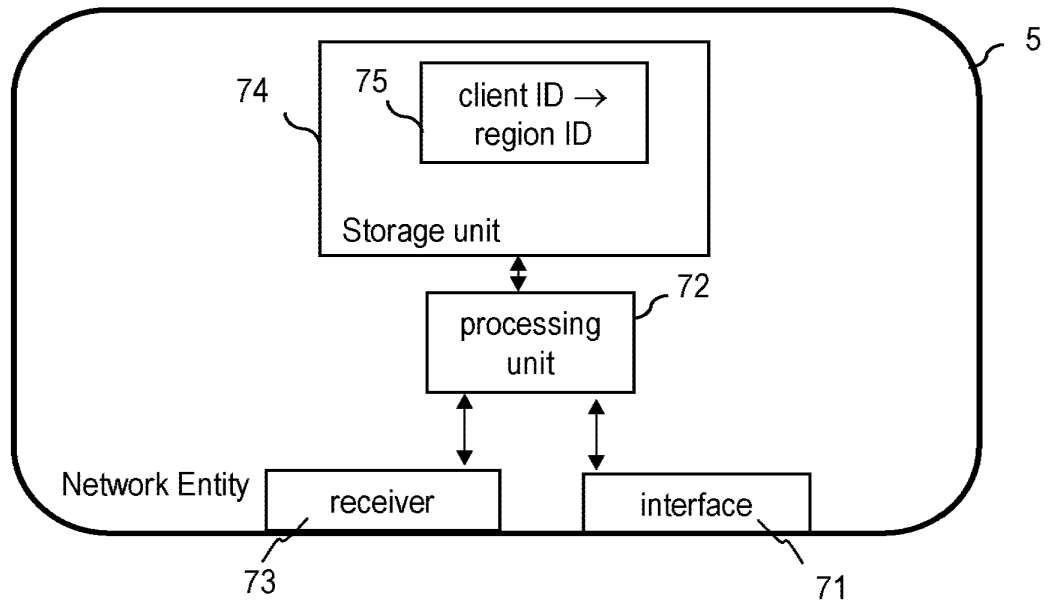


FIG. 7

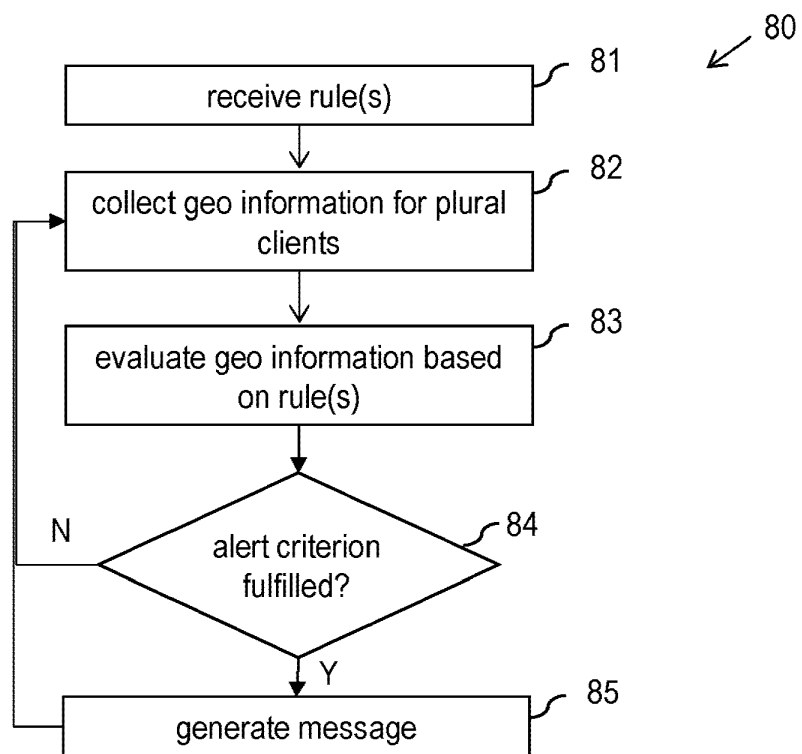


FIG. 8

6/9

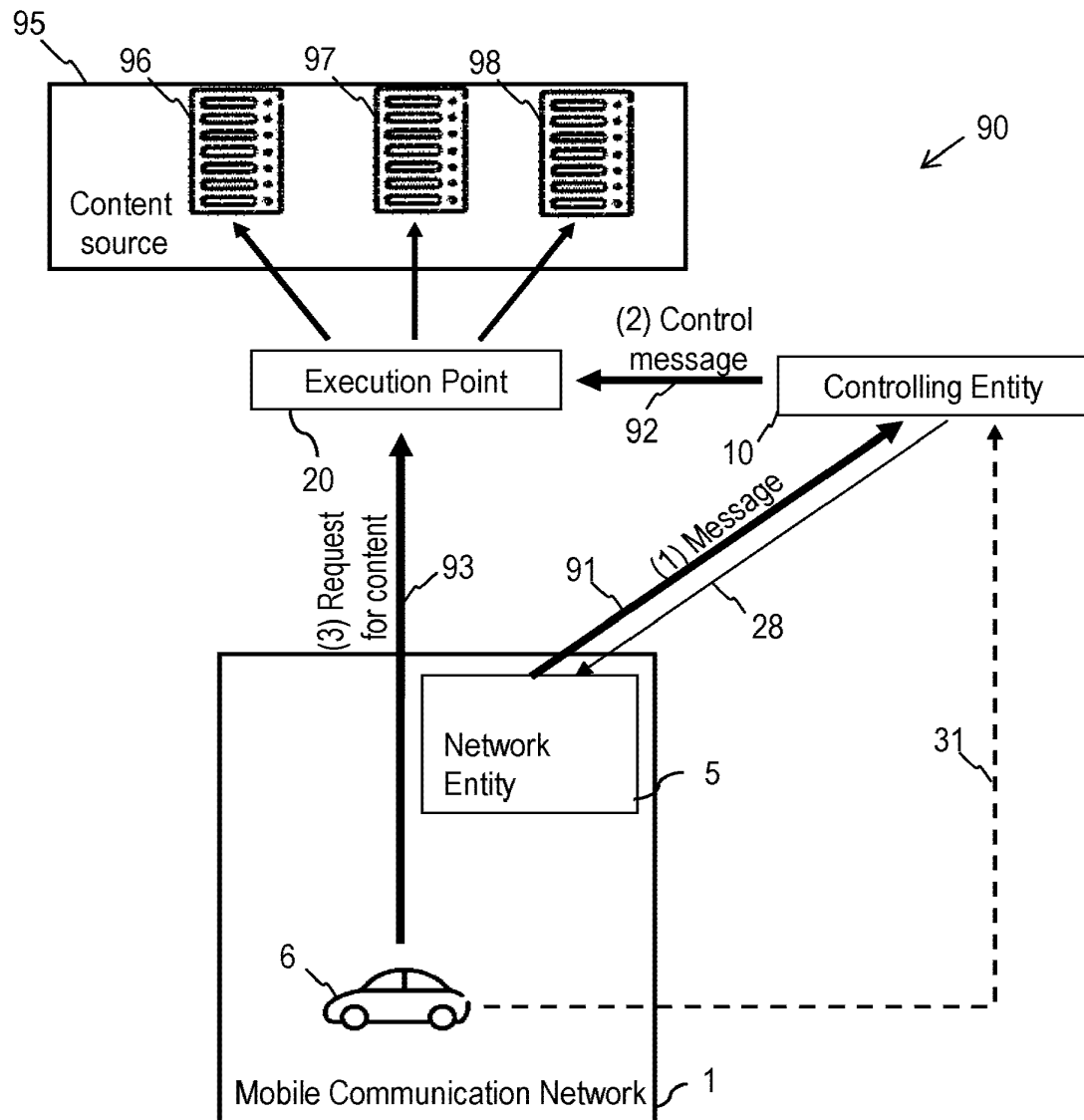


FIG. 9

7/9

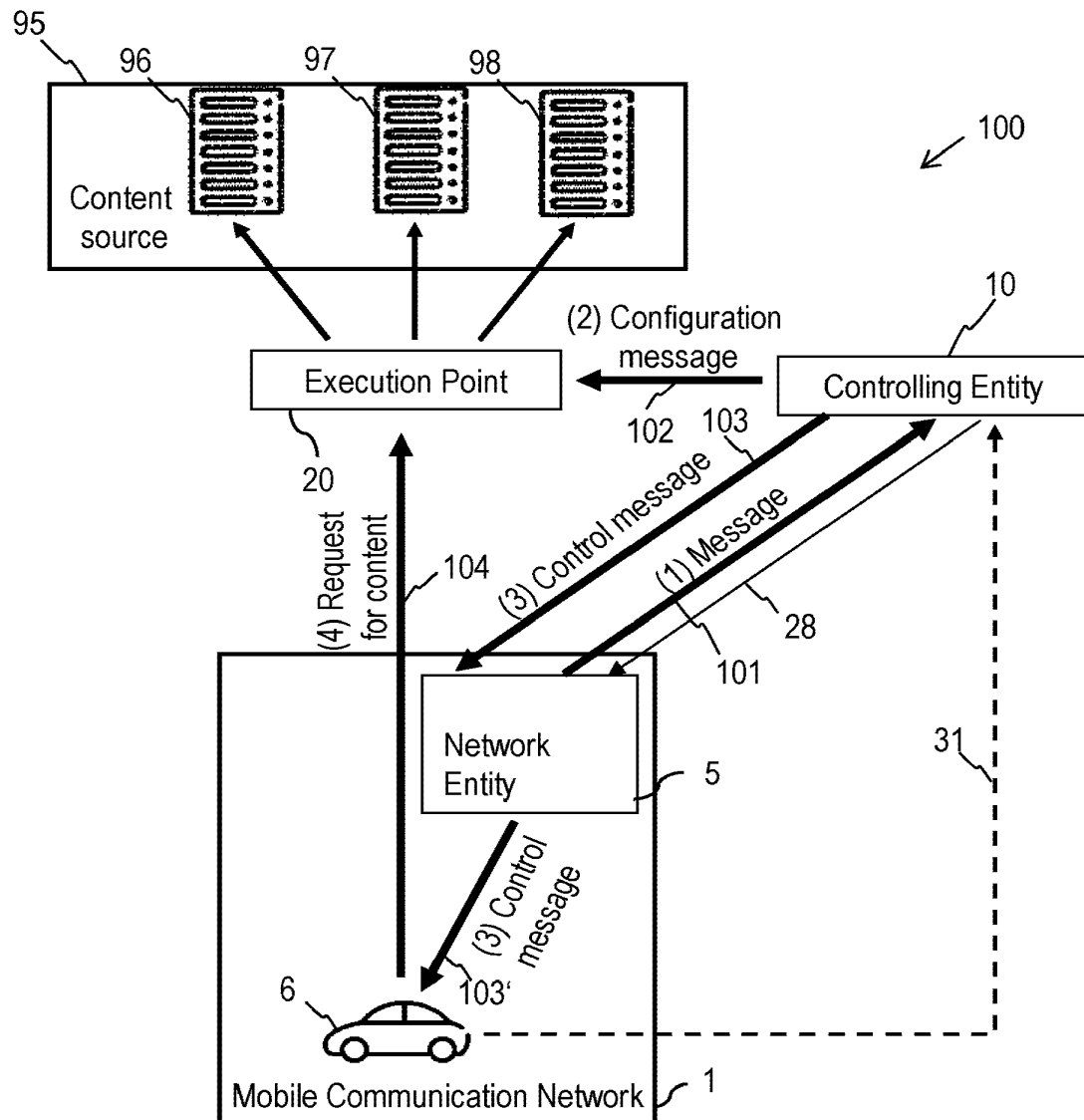


FIG. 10

8/9

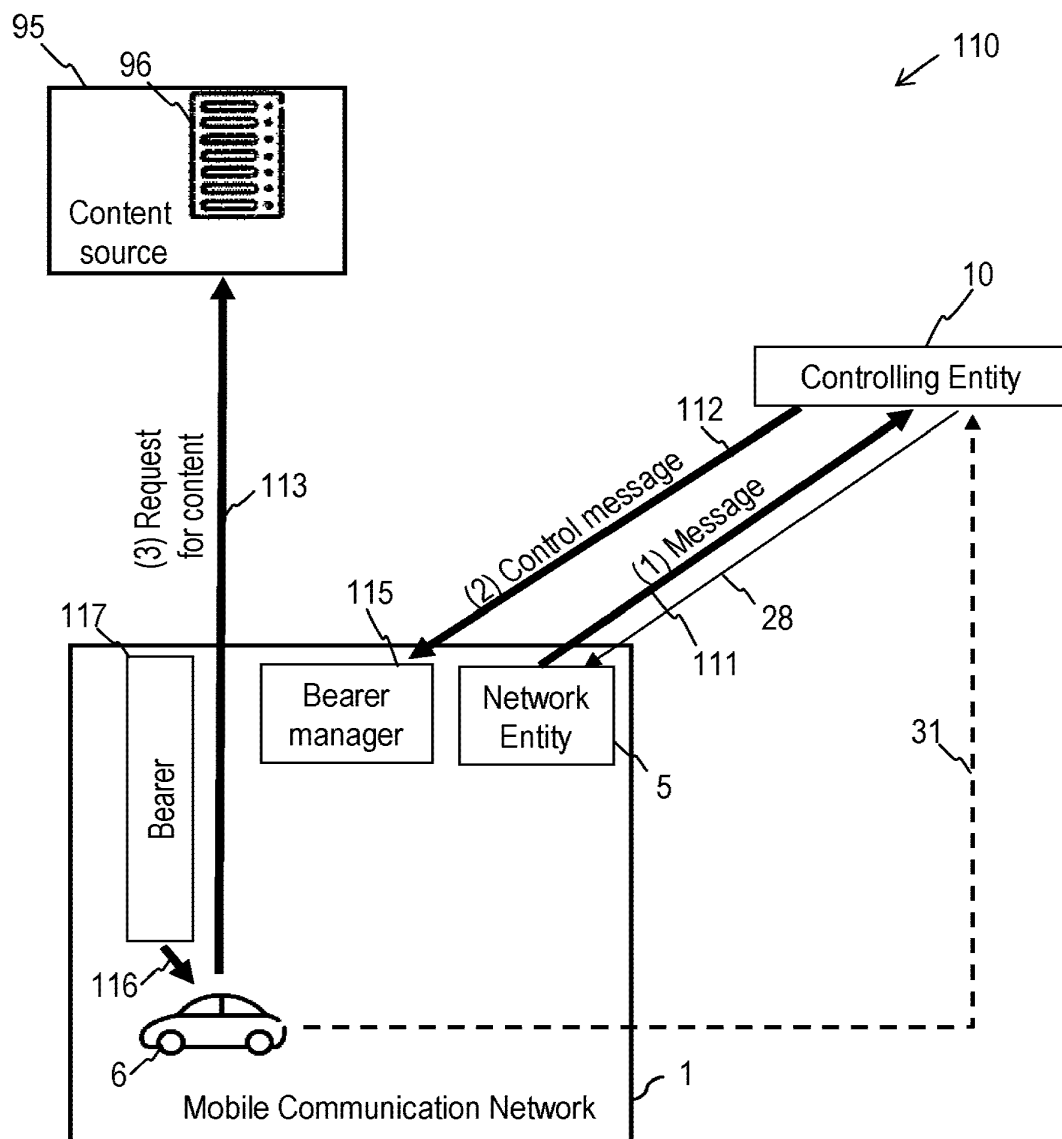


FIG. 11

9/9

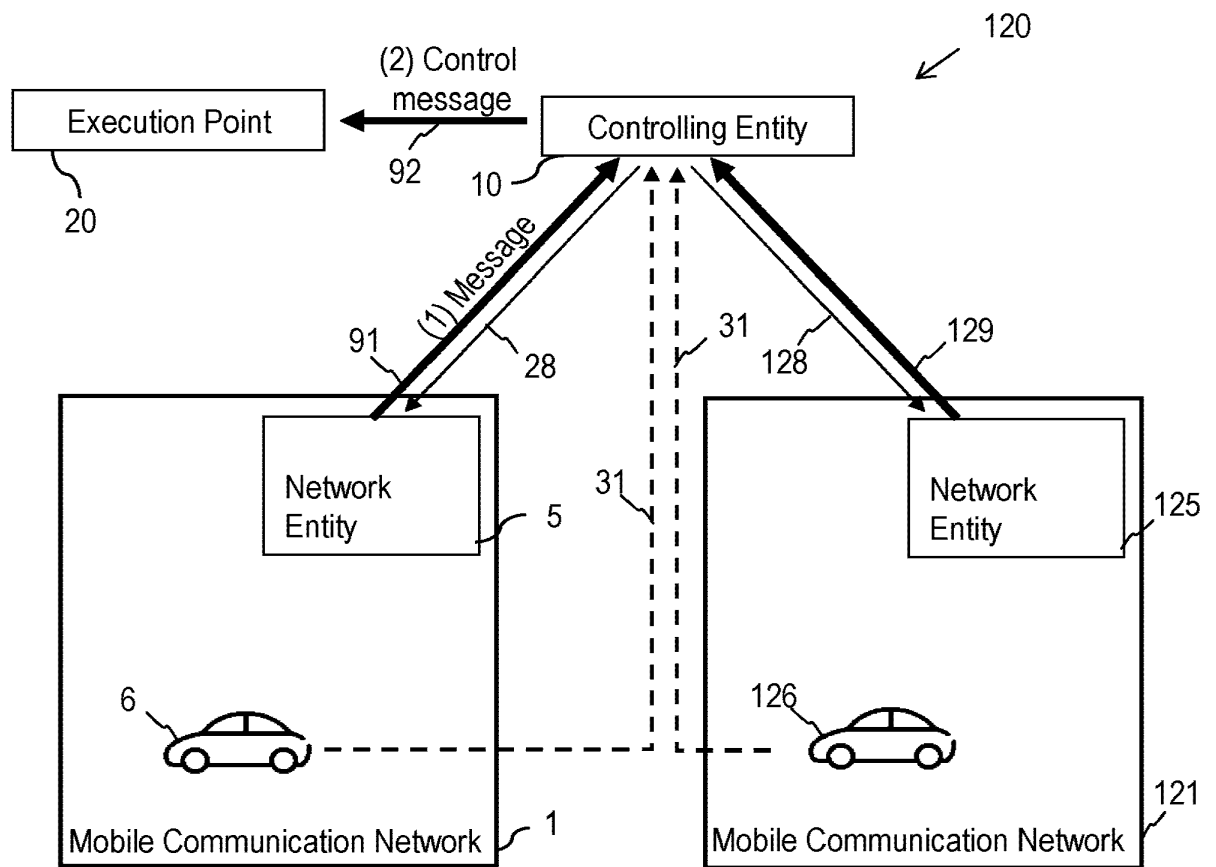


FIG. 12

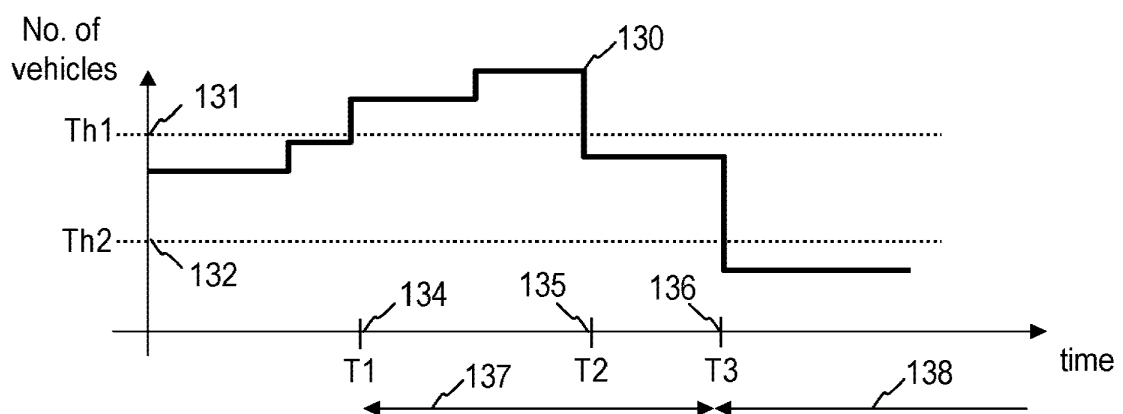


FIG. 13

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2011/067834

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L29/08 H04W4/18
ADD.

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)

H04L H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2011/044287 A1 (OPENWAVE SYS INC [US]; GOPALAKRISHNAN KUMAR [US]) 14 April 2011 (2011-04-14) abstract paragraphs [0001] - [0006], [0060], [0113], [0159], [0160]; figures 4,7,8 -----	1-23
X	WO 2008/016694 A2 (JITTR NETWORKS INC [US]; CRAYFORD AARON M [US] VUSION INC [US]; CRAYFO) 7 February 2008 (2008-02-07) abstract pages 5,32,86, -----	1-10, 12-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* 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 document but published on or after the international filing date

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

"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

29 February 2012

Date of mailing of the international search report

06/03/2012

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Hackl, Alexander

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2011/067834

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2011044287 A1	14-04-2011	US 2011082924 A1	07-04-2011
		US 2011082946 A1	07-04-2011
		WO 2011044287 A1	14-04-2011

WO 2008016694 A2	07-02-2008	EP 2113150 A2	04-11-2009
		US 2008072264 A1	20-03-2008
		WO 2008016694 A2	07-02-2008

(10) International Publication Number
WO 2013/101814 A1(43) International Publication Date
4 July 2013 (04.07.2013)

- (51) International Patent Classification:
G06F 17/30 (2006.01)
- (21) International Application Number:
PCT/US2012/071629
- (22) International Filing Date:
26 December 2012 (26.12.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
13/339,680 29 December 2011 (29.12.2011) US
- (71) Applicant: UNICORN MEDIA, INC. [US/US]; 24 West 5th Street, Tempe, Arizona 85281 (US).
- (72) Inventors: GORDON, Michael M.; 6500 East Cactus Wren Road, Paradise Valley, Arizona 85253 (US).
MCGOWAN, Albert John; 4201 East Camelback Road, Unit 101, Phoenix, Arizona 85018 (US).
- (74) Agents: MCMILLAN, Scott et al.; Eighth Floor, Two Embarcadero Center, San Francisco, CA 94111 (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, BN, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, 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).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

[Continued on next page]

(54) Title: MULTI-PLATFORM MEDIA SYNDICATION CUSTOMIZATION

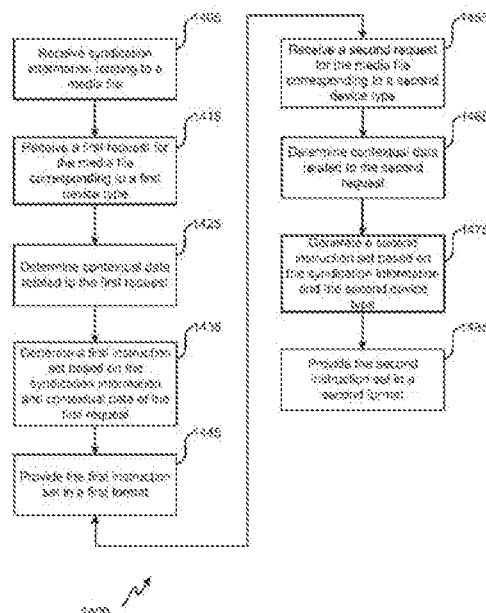


FIG. 14

(57) Abstract: Systems and methods for enabling syndication of a media file across multiple environments, or platforms, by interpreting and downscribing script language in which to implement syndication information. These systems and methods may be part of a larger media servicing network that can be used to, among other things, process uploaded media content, provide it for streaming, and collect metric information regarding the streaming. The disclosed systems and methods can utilize different software development kits (SDKs) to interpret script and provide platform-specific interpreted script such that syndication information provided in the script are dynamically implemented for various device types during runtime.

WO 2013/101814 A1



Published:

— with international search report (Art. 21(3))

MULTI-PLATFORM MEDIA SYNDICATION CUSTOMIZATION

BACKGROUND OF THE INVENTION

[0001] The delivery of media over networks such as the Internet can be accomplished in many ways, including progressive downloading or streaming. Streaming a media file typically involves downloading “chunks,” or small segments, of the media file. Information including where chunks may be accessed can be stored in an index file (also known as a “manifest” file). This index file can be delivered to a client, such as a media player application, for use in streaming. Additional information may also be provided, which can alter the appearance of the client. Due to the manner in which many media file delivery service providers deliver content, however, the degree to which media file delivery is customized, or media file syndication is controlled, is limited.

BRIEF SUMMARY OF THE INVENTION

[0002] Systems and methods are disclosed provide a technical solution for enabling customized media file delivery by utilizing syndication information based on contextual data extracted from a request for the media file. The disclosed systems and methods provide for receiving a request having a Uniform Resource Locator (URL) or other content indicator, such as an embed code, and providing an index file or other metadata suitably formatted and conveyed in accordance with syndication information based on contextual data associated with the content, the request for the content, both the content and the request for the content, the point in time corresponding to the content request, or other content or context attributes. Embodiments further enable media content owners to distribute, among many media providers, a single URL or other content indicator corresponding to a particular media file, such that the single URL or other content indicator is associated with different methods, characteristics, or aspects of identifying, presenting, monetizing, further distributing, sharing (by users or others), or otherwise handling, manipulating, or managing the content referenced by the single URL or other content indicator based on contextual data associated with the content, the request for the content, both the content and the request for the content, the point in time corresponding to the content request, or other content or context attributes, and enabling a single media delivery and analytics services to provide comprehensive metric information, including user, content, context, syndication, and other metrics, for the all the media providers and context with which the content is associated.

[0003] Systems and methods for enabling syndication of a media file across multiple environments, or platforms, by interpreting and downscribing script language in which to implement syndication information. These systems and methods may be part of a larger media servicing network that can be used to, among other things, process uploaded media content, provide it for streaming, and collect metric information regarding the streaming. The disclosed systems and methods can utilize different software development kits (SDKs) to interpret script and provide platform-specific interpreted script such that syndication information provided in the script are dynamically implemented for various device types during runtime.

[0004] An example method for providing a media file via a data network, according to the disclosure, includes extracting first contextual data from a first URL request for a media file. The first URL request is received via the data network and corresponds to a first device type. The method further includes processing the first contextual data using syndication information to generate a first instruction set in a first format, and sending the first instruction set via the data network. The method also includes extracting second contextual data from a second URL request for the media file. The second URL request is received via the data network and corresponds to a second device type. The method also includes processing the second contextual data using the syndication information to generate a second instruction set in a second format, and sending the second instruction set via the data network.

[0005] An example server enabling customized media file delivery with a data network, according to the disclosure, includes a network interface for communicating with the data network, a memory, and a processor communicatively coupled with the memory and the network interface. The processor is further configured to extract first contextual data from a first URL request for a media file. The first URL request is received via the network interface and corresponds to a first device type. The processor is also configured to process the first contextual data using syndication information to generate a first instruction set in a first format, send the first instruction set via the network interface, and extract second contextual data from a second URL request for the media file. The second URL request is received via the network interface and corresponds to a second device type. The processor is finally configured to process the second contextual data using the syndication information to generate a second instruction set in a second format, and send the second instruction set via the network interface.

[0006] An example system for enabling customized media file delivery via a data network, according to the disclosure, can include means for extracting first contextual data from a first URL request for a media file. The first URL request is received via the data network and corresponds to a first device type. The system can further include means for
5 processing the first contextual data using syndication information to generate a first instruction set in a first format, means for sending the first instruction set via the data network, and means for extracting second contextual data from a second URL request for the media file. The second URL request is received via the data network and corresponds to a second device type. The system can finally include means for processing the second
10 contextual data using the syndication information to generate a second instruction set in a second format, and means for sending the second instruction set via the data network.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The present disclosure is described in conjunction with the appended figures:

[0008] FIG. 1 is a block diagram of a media servicing system.

15 [0009] FIG. 2A is a block diagram of an embodiment of a kernel application center connected with application centers.

[0010] FIG. 2B is a block diagram of an alternative embodiment of a kernel application center.

[0011] FIG. 3 is a block diagram of an embodiment of an application center.

20 [0012] FIG. 4 is a block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion.

[0013] FIG. 5 is a system for providing an appropriate index file to any of a variety of clients utilizing a single URL.

25 [0014] FIG. 6A is a simplified flowchart of a method for providing a media file to any of a variety of clients 510 utilizing a single URL.

[0015] FIG. 6B is a simplified flowchart of another method for providing a media file to any of a variety of clients 510 utilizing a single URL.

[0016] FIG. 6C is a simplified flowchart of yet another method for providing a media file to any of a variety of clients 510 utilizing a single URL.

[0017] FIG. 7A is an embodiment of a system for delivering content, including media files, which can be chunked and/or encrypted.

[0018] FIG. 7B is another embodiment of a system for delivering content, including media files, which can be chunked and/or encrypted.

5 [0019] FIG. 8 is a simplified illustration of an embodiment of a system for dynamic encryption integrated into a traditional system that may not have dynamic chunking and/or dynamic indexing capabilities.

[0020] FIG. 9A is a simplified flowchart of a method for dynamically encrypting chunks of media for media streaming.

10 [0021] FIG. 9B is a simplified flowchart of another method for dynamically encrypting chunks of media for media streaming.

[0022] FIG. 9C is a simplified flowchart of yet another method for dynamically encrypting chunks of media for media streaming.

15 [0023] FIG. 10 is a simplified swim lane flowchart describing the interaction of components in a system configured to provide dynamically encrypt chunks of media for media streaming.

[0024] FIG. 11 is a simplified block diagram of an embodiment of a media servicing system, similar to the media servicing system shown in FIG. 1, but further including a content owner and, optionally, additional media provider(s).

20 [0025] FIG. 12A is a simplified flow chart illustrating an embodiment of a general method that can be executed by a system to provide these dynamically-executed syndication services.

[0026] FIG. 12B is a simplified flow chart illustrating an embodiment of a method that demonstrates the customization of index files for different requests utilizing the same URL,
25 based on different contextual data.

[0027] FIG. 13 is a simplified block diagram of an embodiment of a system that can utilize platform-specific software to interpret script provided by an entity such as a content owner.

[0028] FIG. 14 is a simplified flow chart illustrating how a system can utilize syndication information and contextual data to provide different instruction sets in response to different requests for the same media file by devices of different device types, according to one embodiment.

5 [0029] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first
10 reference label irrespective of the second reference label.

DETAILED DESCRIPTION OF THE INVENTION

[0030] The ensuing description provides preferred exemplary embodiment(s) only, and is not intended to limit the scope, applicability or configuration of the disclosure. Rather, the ensuing description of the preferred exemplary embodiment(s) will provide those skilled in
15 the art with an enabling description for implementing a preferred exemplary embodiment. It is understood that various changes may be made in the function and arrangement of elements without departing from the spirit and scope as set forth in the appended claims.

[0031] The increased availability of media content over data communications networks such as the Internet has mirrored the increased bandwidth for these networks. Because
20 media has recently taken a more prominent role in data communications, the distribution of media and the data associated with such distribution has become increasingly important, particularly to media content providers. Media streaming has become a widely-used method of media distribution, but the preprocessing associated with streaming can be burdensome. Certain protocols, including forms of Hypertext Transfer Protocol (HTTP)
25 streaming, require chunking and storing the chunked media files, and generating a corresponding index file(s) (also known as a "manifest" file). In a traditional approach, this can involve a large amount of preprocessing.

[0032] Preprocessing media for streaming traditionally involves chunking media files, storing the chunks, then creating corresponding index files to indicate where chunks may be
30 located to download for streaming. Streaming protocols often provide for frequently updating an index file for instances where the corresponding media is frequently updated, such as during live streaming. Thus, an index file does not need to contain all chunks for a

requested media file. In addition, because media files are frequently stored in a format that requires little additional processing to chunk, the chunks can be created in real time, during the streaming of a media file. The systems and methods disclosed in U.S. Pat. App. No. 12/976,883, entitled "DYNAMIC CHUNKING FOR MEDIA STREAMING," and U.S. Pat. App. No. 12/976,890, entitled "DYNAMIC INDEXING FOR AD INSERTION IN MEDIA STREAMING," which are incorporated herein by reference, take advantage of these features to enable dynamic index file creation and dynamic media file chunking.

[0033] In instances where a media provider desires secure streaming, preprocessing traditionally involves encrypting chunks of a media file as well. Such preprocessing can result in a large amount of stored encrypted chunks that can prove burdensome to manage. For example, if a media provider of a media file desires to update or change the encryption key(s) used to encrypt the stored encrypted chunks corresponding to the media file, each chunk would either need to be decrypted and re-encrypted or replaced altogether with a new chunk, encrypted with the new encryption key.

[0034] In contrast, the techniques provided herein enable dynamic encryption that can allow a system to encrypt chunks of a media file in real time, as the chunks are requested by a client streaming the media file. Such functionality can provide flexibility to a media provider to provide the encryption key(s) used to encrypt a media file at any time, including while media file is streaming to a client. In other embodiments, another entity, such as the content distributor, can provide the encryption key(s). This dynamic encryption can be utilized in a variety of systems, including those that preprocess and store chunks of a media file, as well as those that can dynamically create the chunks. Moreover, techniques described herein are not limited to encrypting chunks of a media file, but also can be utilized to encrypt whole files as well as non-media files and/or chunks of non-media files. Furthermore, the techniques described herein may also return a media file that has been dynamically stitched together from many chunks, which, to a client, can appear like a contiguous file for continuous streaming protocols (Real Time Messaging Protocol (RTMP), Real Time Streaming Protocol (RTSP), etc.) as well as for progressive download.

[0035] The index file(s) utilized to access chunks of a media file (or whole files, in some embodiments) can vary in content and format, depending on protocols utilized by a media player application configured to play the streamed media file. For example, different index files can include information corresponding to a different number of chunks and/or chunks

of media having differing playback parameters (e.g., bit rate, resolution, frame rate, etc.). Despite the differences in format and content, the techniques described herein can be utilized to enable any number of clients, having different index file requirements, to utilize a single Uniform Resource Locator URL (URL), or other indicator, to retrieve the index file
5 corresponding to a particular media file in a format suitable for that client. As a result, a media content provider can provide a single URL for each media file, regardless of the type of client and/or platform requesting the media file.

[0036] Furthermore, when a client uses the URL to request a media file, an index file generator receiving the request can determine whether advertisements are to be played
10 during the playback of the media file, and enable a media provider to select the advertisements to be played. Moreover, the media provider further can provide the index file generator with one or more beaconing URLs to insert into an index file, which can serve as beacons to indicate to the media provider that certain content, such as advertisements, is being and/or has been played. A media provider can find the beaconing information to be
15 vital in determining the value of the media.

[0037] The beaconing information may be used for various purposes. For example, it may be used to determine the state of a client (e.g., paused, skipping certain content, playing back certain content, etc.), which can be used in behavioral advertisement targeting and enforcement of session advertisement behavior, adjusting advertisement content and
20 playback based on the behavior of a user. The state of a client also may be used to support individual encryption keys in an encryption scheme and allow the index file generator to return secure URLs (e.g., time expiring or Internet Protocol (IP) allowed) for chunks to support functions such as payment services.

[0038] While the above embodiments may be implemented in a variety of different
25 systems, some particular embodiments may be implemented as part of a media service system. FIG. 1 is a block diagram illustrating a media servicing system 100, according to some embodiments of the present invention. The system may deliver media content to the end user device 140 through a network such as the Internet 120. The end user device 140 can be one of any number of devices configured to receive media over the Internet 120,
30 such as a mobile phone, tablet computer, personal computer, portable media device, etc. A media file provided by a media provider 130 can be processed and indexed by cloud-hosted integrated multi-node pipelining system (CHIMPS) 110, and further stored on media file

delivery service provider (MFDSP) 150, such as a content delivery network, media streaming service provider, cloud data services provider, or other third-party media file delivery service provider. Additionally or alternatively, the CHIMPS 110 may also be adapted to store the media file.

5 [0039] The media servicing system further enables a media provider 130 or other entity to gather information regarding user behavior during media playback. For example, a media provider 130 can be provided with data indicating that end users tend to stop watching a video at a certain point in playback, or that users tended to follow links associated with certain advertisements displayed during playback. With this data, a media provider 130 can
10 adjust factors such as media content, advertisement placement and content, etc., to increase revenue associated with the media content and provide the end user device 140 with a more desirable playback experience.

[0040] End user device 140 can request a media file to stream with a client program executed by the end user device 140. The client program can be, for example, a media
15 player, browser, or other application adapted to request and/or play media files. In response to a request for a media file, the CHIMPS 110 can utilize any number of application centers 112 and/or kernel application center(s) 111 to provide the client program with a data object concerning the requested media file. The data object can include information about the media file, including where the media file can be located, such as within the MFDSP 150 or
20 within the CHIMPS 110 itself. Location information may be provided by a URL or other indicator. During playback of the media file, the CHIMPS 110 can collect data regarding the playback through beaconing provided by a client program executed by the end user device 140 and/or indexing service from within the CHIMPS and/or MFDSP. The CHIMPS 110 can subsequently provide the data and/or any analytics information derived
25 from the data to the media provider 130. Additionally, as discussed below, the media provider 130 can provide additional beaconing URLs to an index file generator with which the content provider can determine whether particular content has been viewed.

[0041] FIG. 2A is a block diagram illustrating an embodiment of a kernel application center 111-1 connected with application centers from within the CHIMPS 110-1. The
30 kernel application center 111-1 and application centers 112 can be geographically distant and can be connected via the Internet 120, wide area network (WAN), and/or other data communication network. Because application centers can be geographically separated,

DNS services (not shown) can be used to allow an end user device 140 to connect to the nearest available application center 112. The kernel application center 111-1 can connect with application centers 112 within the CHIMPS 110-1 through an internal interface 270, thereby enabling the application centers 112 access to the various components within the kernel application center 111-1.

[0042] Components within the kernel application center 111-1 can communicate through network 260 such as a local area network (LAN) and can include one or more origin servers 240 and a storage array 230 with which data objects and/or media files may be stored and distributed. The storage array 230 may also be utilized by services running on processing server(s) 220 and/or transcoding server(s) 250 that may require temporary or long-term storage. Kernel server 210 can utilize processing server(s) 220, transcoding server(s) 250 to provide various functional capabilities to the CHIMPS 110.

[0043] For example, as described in more detail below, the CHIMPS 110-1 can provide transcoding service for media files provided by a media provider 130 for syndication. Such a service can allow a media provider 130 to upload a media file to an application center 112, after which the application center 112 would notify the kernel server 210 that the media file has been uploaded. The kernel server can then notify services running on the processing server(s) 220 of the upload. These services can utilize transcoding server(s) to transcode the media file, which can then be moved to a MFDSP and/or stored locally by storage array 230 and origin server(s) 240. Services running on the processing server(s) 220 can also update the associated data object stored by the storage array 230 and origin server(s) 240.

[0044] FIG. 2B is a block diagram illustrating an alternative embodiment of a kernel application center 111-2. In addition to the components of the embodiment of FIG. 2A, this embodiment incorporates an application center 112 within the kernel application center 111-2. The application center 112 incorporated within kernel application center 111-2 may be located at or near the other components of the kernel application center 111-2, and can be communicatively connected to the other components via network 260. The incorporated application center 112 can therefore have faster access to kernel application center functionality because it does not need to communicate over long distances. In consideration of this advantage, it will be understood that the CHIMPS 110 can include multiple kernel centers with one or more application centers incorporated therein. Additionally or alternatively, components of the kernel application center may be incorporated into one or

more application centers 112 in the CHIMPS 110 to provide quicker access to certain functionality.

[0045] FIG. 3 is a block diagram illustrating an embodiment of an application center 112. The application center 112 can include caching server(s) 330 and a storage array 310 for storing and distributing data objects of media files requested by end user devices through end user interface 360. Caching server(s) 330 and storage array 310 can also be used to collect, process, and/or store metrics information from beaconing data, media chunk requests, and/or other data sources, including data collected through end user interface 360. The application center can further include ingest server(s) 320 for ingesting uploaded media files from a media provider 130 through a content provider interface 370. The media files may be stored on the storage array 310. As with the kernel application center 111, the components of the application center 112 can be communicatively linked through a network 340, such as a LAN. The application center can further include an internal interface 350, providing a communication link from the application center to the rest of the CHIMPS. It is through internal interface 350, for example, that media files stored on storage array 310 can be made available to a kernel application center 111 for services such as transcoding.

[0046] FIG. 4 is a block diagram 400 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Although FIG. 4 further indicates the physical systems in which may execute or store these processes and objects, it will be understood that the processes and objects disclosed may be executed or stored on more than one system, including systems not disclosed in FIG. 4. In other words, the processes and objects shown in FIG. 4 allow for a variety of implementations through one or more of hardware, software, firmware, microcode, etc.

[0047] Media can be ingested into the CHIMPS 110 when a media provider 130 uploads a media file to ingestion server(s) 410 in an application center 112 by utilizing a client 405. The client 405 can be a stand-alone application or browser based, for example, and can communicate with ingest server(s) 410 through an application programming interface (API) configured for the ingestion of media files.

[0048] Ingest server(s) 410 can communicate with devices in the kernel application center 111 executing programs such as kernel server 425 and file replication service 430. The kernel server 425 can be configured organize the workflow among services such as transcoding 440 file system manager 435, and other services 445 (e.g., analytics, dynamic

API, etc.) Upon a particular event, for example, the kernel server can be configured to notify the relevant services of the event, causing the services to process tasks associated with the event.

5 [0049] The file replication service 430, under direction of the kernel server 425, can coordinate the movement of the media files between services. For example, retrieving the uploaded media file from the ingest server(s) 410 and storing it on the file archive 450, or retrieving transcoded media files from transcoding server(s) 460 and storing them in the media file origin.

10 [0050] The data object updater 420 keeps the data object origin 415 up to date in response to any changes in the system. When, for example, a file is uploaded, transcoded, and stored in media file origin 455, the location and other metadata concerning the transcoded media files need to be created or updated in the data object origin 415 to ensure an end user device that accesses the object in the data object origin 415 has the correct information regarding the related media file. Because the data object updater 420 receives updates from the kernel
15 server 425 (which is notified when a transcoded media file is stored in the media file origin 455, the system ensures the data objects in the data object origin are constantly up to date.

[0051] The upload of a media file to the ingest server(s) 410, as described above, can provide an example of how the kernel server 425 may coordinate workflow. For instance, in response to the upload, the ingest server(s) 410 can notify the kernel server 425 that a
20 media file has been uploaded. The kernel server 425 informs the file replication service 430 of the uploaded media file, and the file replication service 430 moves the uploaded media file into the file archive 450 and notifies the kernel server 425 of the move. In response, the kernel server 425 notifies the file replication service 430, the file system manager 435, and the transcoding master 440 of the move. The file replication service 430 then will know it
25 can delete the uploaded media file from the ingest server(s) 410, the file system manager 435 will update the file system accordingly, and the transcoding master 440 will notify transcoding service(s) 460 of different transcoding tasks to be performed. The transcoding service(s) 460 can then retrieve the uploaded media file from the file archive 450 to create transcoded media files. The transcoding service(s) 460 notify the kernel server 425 once
30 transcoding is complete, and the kernel server 425 relays this information to the file replication service 430. The file replication service 430 then takes the transcoded media files from the transcoding services 460 and moves them to the media file origin 455. Once

the file replication service 430 notifies the kernel server 425 of the move, the kernel server 425, in turn, notifies the file replication service 430 and the data object updater 420. The data object updater 420 which updates the data object origin 415 accordingly, and the file replication service 430 deletes the transcoded media files from the transcoding services 460.

5 [0052] The modular nature of the system enables all tasks associated with an event to be completed quickly. As illustrated in the example above, workflow relating to a particular event, such as a media file upload, can be spread among the various services simultaneously. Moreover, because the system's modularity enables it to be scaled to accommodate differing hardware capacities, and because the system can be configured to
10 dynamically allocate hardware to different services according to the needs of the system, the speed of completing tasks relating to a particular event can further be increased. For example, a server of the CHIMPS 110 can be configured to dynamically switch its purpose based on external conditions such as load and overall system performance, providing functions such as transcode, upload, metrics collection, application web service, and more,
15 on an as-needed basis.

[0053] Embodiments of such systems may include other systems that manage various requests from end users. For example, a system for providing an appropriate index file to any of a variety of clients utilizing a single URL. Referring to FIG. 5, an embodiment of such a system 500 is shown. Media may be streamed to end user device 140 through a client
20 510. As mentioned above, the client 510 can be stand-alone media player, a plug-in, a browser, or other application, which can be executed on a personal computer or other electronic device.

[0054] An index file (i.e. manifest file) generator 530 can be a program instantiated for media streaming to a particular client 510. The index file generator 530 can be executed on
25 a server or other computing device within an application center 112 of the CHIMPS 110. Index files generated by the index file generator can include a wide variety of information such as starting, ending, and/or run times for media chunks and locations for media chunks. This information can be embedded in a single string of data, such as a URL or other type of URI. If media includes various sub-streams (e.g., streams with alternative bitrates, captions,
30 alternative languages, etc.) the index file can include data for chunks corresponding to each of the alternative sub-streams, as well as information regarding the bitrate and/or other unique information for each stream. Alternatively or in addition, index files indicating

alternative sub-streams may be separate from index files indicating one or more media chunks for streaming.

[0055] It should be understood that the index file can further comprise a wide variety of formats, which can depend on a particular streaming protocol utilized by the client 510.

5 HTTP streaming may, for example, require index files to comprise one or more of M3U, M3U8, Extensible Markup Language (XML), and XML-based formats. Of course, other formats can be used in accordance with relevant streaming protocols.

[0056] The index file generator 530 can determine the relevant streaming protocol from information included in a request sent from the client 510 to stream a media file. For
10 example, a client 510 can utilize a URL, obtained from a media provider 130 or other entity to stream a particular media file, to request the media file from the index file generator 530. In addition to the URL, the request can include information regarding the identification of the client 510 (or "client ID"; e.g., a user agent identification in a request header) that the index file generator 530 can use to determine the proper format and content of an index file
15 for the client 510.

[0057] A proper format and content of an index file can be determined in numerous ways. For example, the index file generator 530 itself may recognize the type of client from the client ID and determine a proper index file type accordingly. The index file generator 530, therefore, may include information regarding common client IDs and/or special use cases
20 for which particular index file types are used. This information can be updated periodically, and/or as index file types are determined for different client IDs.

[0058] Alternatively, the index file generator 530 can access a client information database(s) 540 to determine the proper index file type. Such a database(s) can be located within the CHIMPS 110 (shown) and/or external to the CHIMPS 110 (not shown),
25 depending on desired functionality. One example of an external client information database(s) 540 is the Wireless Universal Resource File (WURFL), a device description repository maintained by ScientiaMobile, Inc. The proper index file type can be determined by identifying an index file type known to work for a particular client ID or matching a client ID to an index file type based on a profile of capabilities associated with the client ID.

30 [0059] If a proper index file type cannot be determined, the index file generator 530 can provide an index file of a default index file type. The default index file type can include information for streaming the requested media file using a basic media stream compatible

with virtually any media client. For example, parameters associated with a basic video stream could include a resolution of 160 x 120, a 3GP (Third Generation Partnership Project file format) multimedia container format, and/or a streaming bit rate of 100 kilobits per second (kbps).

5 [0060] The index file generator 530 further can utilize a file information database 550 in the creation of an index file. The file information database 550 can provide information regarding the requested media file (e.g., length, genre, author, etc.) as well as information regarding whether any advertisements are to be shown during the playback of the requested media file. If advertisements are to be shown during the playback of the requested media
10 file, the database further can provide points at which advertisements are to be played during playback of the media file by the client.

[0061] An advertisement server 520, which can be maintained by a media provider 130, can provide the index file generator 530 with additional information regarding advertisements to be shown during the playback of the requested media file. For example,
15 the index file generator 530 can determine, using information from the file information database 550, that three video advertisements are to be shown at certain points during the playback of a particular video file. The index file generator 530 then can request information from the advertisement server 520 regarding which advertisements to show. (This can be in the form of three different requests, or a single request, depending on the
20 desired functionality of the system.) Moreover, the index file generator 530 can use the forwarded IP address and forwarded user agent of the client to identify the client, thereby allowing the media provider 130 to provide customized advertisements for the client. The advertisement server 520 can specify the advertisements to show (if an advertisements have been previously uploaded to the CHIMPS, and the index file generator can receive metadata
25 regarding the advertisements from the file information database 550. Alternatively, the advertisements may be uploaded to and chunked by the CHIMPS 110 after the index file generator 530 requests the information from the advertisement server 520 regarding which advertisements to show. In this latter case, metadata regarding the advertisements would also be extracted and used in the creating of the index file. Regardless of when the
30 advertisements are uploaded to the CHIMPS 110, the advertisement server 520 enables a media provider 130 to traffic new advertisements into the playback of a media file shortly before the index file is generated, which can occur shortly before or even during the playback of a media file.

[0062] In addition to providing information regarding advertisement content, the advertisement server 520 can designate certain URLs in the index file for beaconing. These beaconing URLs can be similar to normal URLs of an index file, but with additional information attached, designating it as a providing a "beacon" to report back to the media provider 130. The content provider can use these beacons to determine, among other things, if a particular advertisement is played. For example, a beaconing URL can be a redirect URL included in a request for a first chunk of an advertisement. The request, which initially is directed to an API server of the CHIMPS 110, is interpreted as a beacon by the API server and added to a list of items for which the API server requests of the advertisement server 520 (or other system of the media provider 130). The beacon itself can be, for example, a getRequestURL() or similar request that the advertisement server 520 can use to determine that a particular URL was made. The API server can use the forwarded IP address and forwarded user agent of the client to help ensure that the media provider 130 can correctly determine that a beacon corresponds with a request from a particular client 510. The API server also can redirect the client to a particular media file delivery service provider (MFDSP) (or other system hosting the requested chunk) to receive the requested chunk. In alternative embodiments, the media provider 130 can provide additional beaconing URLs that can be used to provide beaconing information regarding the playback of the media file itself. Through the use of such beaconing URLs, the media provider 130 is able to provide its own beaconing data in addition or as an alternative to any beaconing data gathered by the CHIMPS 110.

[0063] The index file generator 530 then uses the information regarding the client ID, the requested media file, the advertisement(s) to be shown during playback of the media file, and the points of the media file at which the advertisement(s) are to be played, to create an index file of the right index file type to return to the client 510. As indicated above, the index file can include, among other things, a number of URLs indicating the location of each chunk of the media file to be played by the client 510, as well as chunks of the advertisement(s). The chunks of the advertisement(s) are included in a manner such that they are shown at a point(s) during the playback of the media file corresponding to the points designated by the file information database 550. Additionally, the index file can include one or more beaconing URLs which, when used, can be indicative of the playback of an advertisement as discussed above.

[0064] The URLs provided by an index file additionally can direct a client 510 to additional index files. For example, under certain adaptive bit rate streaming protocols, a first index file typically can include a number of URLs to additional index files, where each additional index file corresponds to a particular bit rate for streaming. The client 510 then can choose a bit rate based on one or more factors such as connection speed, device type, etc. Other streaming rates (bytes, etc.) may be used additionally or alternatively.

[0065] To this end, the index file generator 530 can be configured to create an index file that provides the client 510 with a particular set of bit rates adapted to the client's circumstances. The client's circumstances not only can include the type of end user device 140 (also referred to herein as "device type") on which the client is running, but also the type of network to which the device is connected, among other things. These circumstances may be determined from a request header provided by the client along with a URL, and/or they may be determined utilizing other data obtained from and/or regarding the client 510. (The index file generator 530, for example, can determine that the Autonomous System (AS) number of a particular client's IP address is associated with a provider of a mobile wireless network.) Because the set of bit rates provided in the index file provides a customized selection for the client 510, the resulting playback can be optimized to provide the best user experience.

	Device/Network Type			
	Smart Phone/ Mobile Wireless Network	Smart Phone/ Wired Network	Tablet/ Mobile Wireless Network	Tablet/ Wireless Network
Streaming Bit Rate (kbps)	1200		X	X
	800		(X)	X
	600	X	(X)	(X)
	400	(X)	X	X
	200	X	X	X
	100	X	X	X
	64	X	X	X

Table 1: Example Bit Rates for Certain Device/Network Types

[0066] **Table 1**, provided merely as an example, illustrates the different sets of streaming bit rates an index file can make available to a client, based on the device type and the network type. Not only can the index file include a selection of available bitrates, indicated by "X", but the index file further can designate an initial bit rate, indicated by "(X)", for the

client 510. The client can then choose to stream the media file using the initial bit rate designated by the index file, or it may choose to stream the media file using one of the other bit rates provided in the index file. Alternatively, if the client 510 does not utilize an adaptive bit rate protocol, the index file generator can provide an index file of a single bit rate, where the single bit rate can be determined based on device type, network type, etc.

[0067] For example, an index file for a smart phone connected to a mobile wireless network (e.g., a wireless carrier for mobile phones and other wireless devices) can provide URLs for streaming a requested media file at 600, 400, 200, 100, and 64 kbps, where 400 kbps is designated as the initial rate at which the client can begin streaming the media file.

However, if a smart phone is connected with a wired network (e.g., a cable or DSL internet connection), including a wireless local area network (LAN) connected to the internet through a wired network, the index file may provide the same set of URLs for streaming the requested media file, but designate a higher initial bit rate at which the client can begin streaming the media file. On the other hand, because a tablet computer may have a monitor capable of displaying higher-quality resolutions associated with a higher bit rate, the index file can provide a tablet computer with different sets of bit rates and different starting bit rate designations, including higher bit rates, that are not included in an index file provided to a client 510 running on a smart phone.

[0068] FIG. 6A illustrates a simplified flowchart of a method 600-1 for providing a media file to any of a variety of clients 510 utilizing a single URL. This method can be executed, for example, by the index file generator 530 of FIG. 5. The method 600-1 begins at block 610 where a request for a media file is received. Among other things, the request can contain a URL corresponding to the media file.

[0069] At block 612, the device type and/or network type is determined. As discussed above, the request can include a header with client ID information. The client ID information can be indicative of a particular device type, including the type of physical device, as well as the type of operating system and/or client the physical device is running. Determining the device type can include using one or more databases and/or resorting to a default device type if a particular device type is not identified. As discussed above, the network type can be determined, for example, from the AS number of the client's IP address, which can be associated with a particular network provider (e.g., wireless mobile network provider, wired network provider, etc.).

[0070] At block 615, metadata regarding the requested media file is retrieved. The metadata, which can be stored on one or more databases in the CHIMPS 110, for example, can include information regarding the media file such as length, title, author, etc. Additionally, at block 620, advertisement support can be determined. Information
5 regarding advertisement support, which also can be stored on one or more databases, can include whether advertisements can be included in the playback of a media file, and if so, at what points during the playback of the media file.

[0071] At block 625, if the media file includes advertisement support, the advertisement(s) to include in the playback of the media file are determined. As discussed
10 previously, determining the advertisement(s) to include can comprise communicating with a media provider 130 (or other entity), who can indicate the advertisement(s) to include. The advertisement(s) (which can be files with a video and/or audio) may be uploaded beforehand to a MFDSP 150, server, or other delivery system, or they may be uploaded by the media provider 130 (or other entity) after the request for the media file is received. The
15 advertisement(s) further may be chunked beforehand, dynamically chunked once requested, comprise complete file(s), or may be already included as part of a permutation of a media file.

[0072] At block 630 metadata regarding the advertisement(s) is retrieved. Similar to the metadata for the media file, the metadata for the advertisement(s) can include length, title,
20 etc., which can be used in creating the index file. At block 635, the index file is created using the metadata of the media file and advertisement(s) as well as information regarding the device type, which can impact the format and/or content of the index file. Finally, at block 640, the index file is returned.

[0073] The method 600-1 can be executed with every time a media file is requested.
25 Even though a single URL can correspond with a single media file, the content of the index file returned at block 640 may be different. Depending on the type of client (e.g., client ID) and/or type of network and the advertisements to be included in the playback, among other things, the index file can vary to conform to different formats, include different available streaming bit rates, include information regarding different advertisements, and more.
30 Thus, despite the fact that a media provider 130 can provide a single URL to correspond to a particular media file, the streaming experience can be tailored to a particular client 510.

[0074] FIG. 6B illustrates a simplified flowchart of a method 600-2 for providing a media file to any of a variety of clients 510 utilizing a single URL, similar to the method 600-1 of FIG. 6A. Here, however, the illustrated method 600-2 demonstrates how there can be a reduced number of blocks if it is determined, in block 620, that there is no advertisement support for the requested media file. In this case, the index file can be built at block 635 without any additional determination of advertisement(s) to include in the playback of the media file. That said, there may be one or more advertisements already integrated into the media file.

[0075] FIG. 6C illustrates a simplified flowchart of a method 600-3 for enabling a system to provide a media file to any of a variety of clients 510 utilizing a single URL, similar to the methods 600-1, 600-2 of FIGs. 6A-6B. In this method 600-3, however, an index file is not returned. Instead, a URL (or other indicator) is determined, at block 637, and returned, at block 642, to the client 510. This method 600-3 illustrates how the systems and methods described herein can be used in applications where the client does not utilize an index file, but rather requests an entire media file at once. The URL returned to the client at block 642 can indicate the location of a particular permutation of the requested media file with advertisements included as determined at block 625. Depending on the capabilities of the system providing the media file, the particular permutation of the media file can be dynamically generated upon request by the client if not otherwise stored on the system.

[0076] Dynamic generation of chunks and/or entire media files may or may not involve transcoding. The media file can be stored in a format where transcoding may not be needed, thereby reducing the processing requirements for creating chunks of media during streaming. For example, media files may be stored such as H.264 or MPEG-4 video format and/or AAC, HE-AAC, or MP3 audio format. According to some streaming protocols, such as some forms of HTTP streaming, chunks of media in these formats would not need transcoding before being wrapped in an MPEG-2 transport stream container format. Instead, such a conversion essentially would require the addition of metadata to create the streaming format from the format of the stored media file. In other words, generating a deliverable chunk of media may only require identifying the stored media file, extracting the relevant segment of the media from the media file, and adding certain metadata in accordance with a container format. This process requires little processing power and can be easily performed on the fly during streaming. More details regarding this process can be found in U.S. Pat. App. No. 13/092,299, entitled "TRANSCODELESS ON-THE-FLY AD

INSERTION,” which is incorporated herein in its entirety. Once the deliverable chunk of media is generated, it can be encrypted according to the techniques described herein.

5 [0077] Where an index file is used, the client can stream the requested media file by using the URLs designated in the index file to download the chunks from a content delivery service. FIG. 7A, shows an embodiment of a system 700-1 for delivering content, including media files, which can be chunked and/or encrypted. The client 510 and index file generator 530 are also illustrated for reference.

10 [0078] In this system 700-1, chunks of media can be generated during media streaming by a dynamic segmentor, which of a dynamic permutation layer (DPL) 740 providing an HTTP service. The DPL 740, as well as the media file origin 455 can be located within a kernel application center 111 of the CHIMPS 110 on, for example, a media file origin server. The system 700-1 can be configured such that the kernel application center 111 provides dynamically-created chunks of media to a MFDSP 150 for delivery to client 510. The MFDSP 150 can store the chunks locally in, for example, a media file cache 720, thereby
15 forgoing the need to dynamically create a chunk again if the same chunk is requested in the future.

[0079] After a chunk is dynamically created, if encryption is desired, the DPL 740 also can encrypt the chunk using an encryption key. The encryption key can be, for example, a private key of an asymmetric encryption scheme. Because the overhead of encrypting a
20 chunk of a media file is relatively small, the DPL 740 can encrypt the chunks in real time, as the client 510 is streaming the media file (i.e., as the chunks are being requested). Such a scheme can be implemented in numerous ways.

[0080] In one embodiment, the DPL 740 can request a private key through an Application Programming Interface (API) server 730 of the media provider 130. The API server 730
25 can return the requested encryption key to the DPL 740 via a secure communication link 785, which can be encrypted and/or otherwise secured to help ensure the security of the encryption key is not compromised. The DPL 740 can then use the encryption key to encrypt one or more chunks of a media file, returning the encrypted chunk(s) to the MFDSP 150 for delivery to the client 510. The client can obtain the corresponding decryption key
30 (e.g., public key) from the media provider 130, the CHIMPS 110, or other source.

[0081] The functionality provided by this system 700-1 enables the media provider 130 to control encryption of chunks of media. Depending on the desired encryption scheme, the

DPL 740 can request a new encryption key—which is provided by the API server 730—for each chunk of a media file. Additionally or alternatively, the DPL 740 can request a new encryption key less frequently, such as with each media file and/or group of media files. Moreover, changing an encryption key may be time based, such that the DPL 740 requests a
5 new encryption key every minute, hour, day, etc. In addition, or as an alternative, the API server 730 may provide a new encryption key to the DPL 740 without a request from the DPL 740.

[0082] In another embodiment, the DPL 740 can generate an encryption key. In this embodiment, the DPL 740 can utilize an algorithm provided by the API server 730 via the
10 secure communication link 785. The API server 730 and DPL 740 can run the algorithm in synchronization to generate identical encryption/decryption keys, such that the encryption key does not need to be communicated between the API server 730 and the DPL 740. Moreover, the API server 730 can provide an algorithm in each response to the DPL's requests, thereby allowing the DPL 740 to generate the encryption key without the need to
15 store an algorithm or otherwise have access to the algorithm beforehand. Alternatively, the DPL 740 can store a variety of algorithms for encryption key generation, and the API server 730 could indicate an algorithm to use in response to an algorithm request from the DPL 740. Such functionality can give the allow media provider 130 control of encryption keys and/or encryption key generation.

[0083] Alternatively, the DPL 740 can simply generate the encryption key (which can be generated for each chunk, media file, etc., or may be time based, as discussed above). In this case, the DPL 740 can provide the encryption key to the API server 730, or retain the encryption key without sharing it, depending on the desired functionality.

[0084] In sum, the system 700-1 for indexing and encrypting dynamically-created chunks
25 of a media file can, after receiving a request for an index file from a client 510, dynamically generate an index file with an index file generator 530. The index file can, among other things, indicate where a next chunk of a media file may be located. A client can then request the chunk from the location indicated by the index file, which can comprise a media file cache 720 in a MFDSP 150. If the chunk is not found in the media file cache 720, or if
30 the chunk is encrypted with an outdated encryption key, the cache miss can redirect the request to a DPL 740, which can dynamically generate the requested chunk by accessing the corresponding media file in the media file origin 455. The DPL 740 determines an