Techniques for delivering content to an end user device are disclosed. A policy server performs an ingest process in which multiple versions of a content object are created. The multiple versions of the content object are associated with a single network identifier which can be distributed to publishers of the content object. When the content object is requested using the single network identifier, the policy server determines a preferred version and delivery orchestration scheme from among the multiple versions for delivery to an end user device based on several criteria. The policy server may thereafter orchestrate delivery of the preferred version of the content object to the end user device. The single network identifier may be associated with an edge location in a content delivery network (CDN) and the policy server may orchestrate delivery in cooperation with servers at the CDN edge location.
— as to the applicant’s entitlement to claim the priority of the earlier application (Rule 4.17(Hi))

P₂, b₁, h₄, d:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))
SYSTEM AND METHOD FOR DELIVERY OF CONTENT OBJECTS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority to Australian Patent Application Serial No. 201 0201 379, filed April 7, 2010 (atty. docket no. 0401 52-007800 AU), which is incorporated herein by reference in its entirety.

BACKGROUND

[0002] The demand for network-based content, particularly audio/video content, has increased dramatically in recent years. Along with the increasing demand, different devices and technologies have emerged which can be used to access such content. Increasingly, these devices are not traditional desktop or laptop computers; they are handheld devices, mobile phones, internet-connected television sets, and other non-traditional devices, with widely varying capabilities. This confluence of events may present challenges to publishers who wish to distribute their content inexpensively, remain up-to-date with the latest technology, and ensure a satisfying experience for their customers.

BRIEF SUMMARY

[0003] Techniques for delivering content to an end user device are disclosed. A policy server performs an ingest process in which multiple versions of a content object are created. The multiple versions of the content object are associated with a single network identifier which can be distributed to publishers of the content object. When the content object is requested using the single network identifier, the policy server determines a preferred version from among the multiple versions for delivery to an end user device based on several criteria. The policy server may thereafter orchestrate delivery of the preferred version of the content object to the end user device. The single network identifier may be associated with an edge location in a content delivery network (CDN) and the policy server may orchestrate delivery in cooperation with servers at the CDN edge location.

[0004] According to one embodiment, a method of delivering content objects to an end user device comprises receiving notification at a policy server that a content object is available for processing. The method comprises associating a single network identifier with a plurality of
versions of the content object, where each version of the content object has a corresponding encoding profile in a set of encoding profiles. The method further comprises sending the network identifier to a publisher of the content object and receiving a request for the content object, where the request includes the network identifier and request information. The method further comprises determining a preferred version of the content object for distribution to the end user device based on the set of encoding profiles and the request information and determining a delivery orchestration scheme by which to send the preferred version of the content object to the end user device. The method comprises sending the preferred version of the content object to the end user device in accordance with the delivery orchestration scheme.

[0005] In another embodiment, a policy server comprises a publisher interface, a content request interface, a merge processor, and a policy engine. The publisher interface receives notification that a content object is available for processing. The merge processor associates a single network identifier with a plurality of versions of the content object, each version of the content object being created with a corresponding encoding profile in a set of encoding profiles. When the plurality of versions of the content object is available for distribution, the merge processor provides the network identifier to the publisher interface for delivery to a publisher of the content object. The content request interface receives an HTTP (hyper-text transport protocol) request for the content object using the network identifier and request information relating to capabilities of an end user device. The policy engine determines a preferred version of the content object for distribution to the end user device based on the encoding profiles and the request information. The policy engine also determines a delivery orchestration scheme including a delivery protocol and causes the preferred version to be sent to the end user device in accordance with the delivery orchestration scheme.

[0006] In an additional embodiment, a method of distributing content objects in a content delivery network (CDN) is disclosed. The method includes receiving notification at a policy server that a content object is available for processing and, in response to receiving the notification, creating a plurality of versions of the content object with a plurality of corresponding encoding profiles. The method also includes associating the plurality of versions of the content object with a single network identifier of the CDN and sending the network identifier from the policy server to a publisher of the content object. The method also includes receiving a request for the content object comprising the CDN network identifier and request information and determining device capabilities of the end user device based on the request.
information. The method includes sending the preferred version of the content object to the end user device from a CDN location corresponding to the CDN network identifier.

[0007] Further areas of applicability of the present disclosure will become apparent from the detailed description provided hereinafter. It should be understood that the detailed description and specific examples, while indicating various embodiments, are intended for purposes of illustration only and are not intended to necessarily limit the scope of the disclosure.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0008] FIG. 1 shows an embodiment of a content distribution system.

[0009] FIG. 2 shows an embodiment of a policy server.

[0010] FIG. 3 shows exemplary publisher information.

[0011] FIG. 4 shows exemplary encoding profiles.

[0012] FIG. 5 shows exemplary information relating to versions of a content object.

[0013] FIG. 6 shows exemplary device capabilities information.

[0014] FIG. 7 shows an interaction between elements of a content distribution system.

[0015] FIG. 8 shows an exemplary ingest process.

[0016] FIG. 9 shows an exemplary delivery process.

[0017] FIG. 10A shows aspects of HPD content delivery.

[0018] FIG. 10B shows aspects of RSD content delivery.

[0019] FIG. 10C shows aspects of RSDH content delivery.

[0020] FIG. 10D shows aspects of IHS content delivery.

[0021] In the figures, similar components and/or features may have the same reference label. In some cases, components of the same type are identified by following a first reference label with a dash and a second reference label that further distinguishes among the similar components. If only the first reference label is used, the description is applicable to any of the similar components designated by the first reference label.
DETAILED DESCRIPTION OF EMBODIMENTS

[0022] The ensuing description provides preferred exemplary embodiment(s) only, and such preferred exemplary embodiments are not intended to limit the scope or applicability of the present invention. Rather, the ensuing description will enable those who are skilled in the art to implement such preferred exemplary embodiment(s). Persons of skill in the art will recognize that various changes may be made in the function and arrangement of elements without departing from the spirit and scope of the invention as set forth in the appended claims.

[0023] Referring first to FIG. 1, a block diagram of an embodiment of a content distribution system 100 is shown in which a content originator 110 offloads delivery of content objects to a content delivery network (CDN). The content originator 110 produces and/or distributes the content objects and may include one or more publishers 106 and content sites 108. The CDN delivers the content objects over the Internet 104 to end users 122 via corresponding end user devices 120.

[0024] As shown, the CDN may include an origin server 112, a policy server 116, and various points of presence (POPs) 118. POPs 118 can be deployed throughout content distribution system 100 and may serve content objects to end user devices 120 in a particular geographic area and/or in connection with a particular service provider. For example, a POP 118 may be designated to serve content objects over Internet 104 to end users 122 in a particular city, on a particular access network, etc. to promote efficient delivery and a good user experience. The various CDN elements may be connected by a private network such as WAN 114 and/or a public network such as Internet 104.

[0025] An end user 122 may browse for content objects at a content site 108 with its respective end user device 120. As used herein, a content object can be any computer-accessible content and preferably includes audio data, video data, and images in any number of computer-accessible formats. The terms content and content object may be used interchangeably wherever they appear. End user systems 120 can include personal computers, media players, handheld computers, Internet appliances, smart phones, personal digital assistants, streaming radios, or any other device that receives and processes content objects. The content site 108 could be a webpage from which content is accessible via a web browser.

[0026] Links to content at the content site 108 may point to locations in the content delivery network. When an end user requests delivery of a particular content object, the request may be
assigned to a POP 120 which, in turn, can deliver the requested content object to the end user device 120. If the content object is not available at the assigned POP location, the request may be propagated toward the core of the CDN and may ultimately be fulfilled from origin server 112. Content may be cached at various points between the core CDN and edge locations to improve efficiency.

[0027] Distribution of content objects often represents an important source of revenue for publishers 106. For example, content sites 108 may generate advertising revenue based on the number of times that a content object is viewed, clicked, or downloaded by end users 122. Thus, to maximize their revenue, publishers 106 may seek to reach as many end users 122 with their content as possible while providing a good overall user experience.

[0028] Unfortunately, end user systems 120 can vary widely in their respective capabilities and the manner in which they interact with content objects. Different end user systems 120 may support different collections of multimedia formats and different delivery schemes. For example, beginning with OS version 3.0, the iPhone™ from Apple, Inc. supports M3U8 playlists and MPEG-2 segmented video with iPhone™ HTTP Streaming (IMS) delivery, entirely over HTTP (Hypertext Transfer Protocol). On the other hand, the Blackberry Storm™ from Research in Motion, Ltd. supports playback of multimedia content in Third Generation Partnership Project (3GPP) format, over RTSP (Real-Time Streaming Protocol).

[0029] To further complicate matters, the manner in which delivery of a content object is initiated may vary from device to device. For example, some end user devices 120 may need help orchestrating a browser-to-player (B2P) handoff for certain types of content objects. Moreover, even when media formats and delivery methods are equally supported, the manner in which a content object is delivered may depend on the type of connection to Internet 104 available to the end user device 120 at a particular place and time. Thus, for example, the playback capabilities of the Blackberry Storm™ may differ depending upon whether it is connected to the Internet 104 via a WIFI connection in a cybercafe, or via a cellular network in a remote location.

[0030] In the present embodiment, policy server 116 is coupled to content site 108 via Internet 104 and receives a notification when new content objects are available from publishers 106. Alternatively, a publisher 106 may upload its content to an origin server 112 and policy server 116 may receive notifications via WAN 114 when a new content object becomes available.
Although shown separately, policy server 116 may be located within POPs 118, origin server 112, or other parts of the content delivery network. Also, it will be recognized that the various operations of policy server 116 may be carried out by multiple individual servers such as decisioning servers, merge servers, assembly servers, etc.

[0031] When a new content object is ready for processing, policy server 116 determines how it should be made available to end users. This may involve generating a number of different versions of the content object optimized for use with different end user devices 120, having different capabilities, and potentially used in different network environments. The different versions of the content object may correspond to different production or encoding profiles maintained at policy server 116. The production profiles, in turn, may be based upon a publisher's requirements for the distribution of its content objects. For example, a publisher may prefer to distribute its content in a specific media format or formats, to exploit device-specific capabilities (such as IHS streaming for iPhones), to optimize separately for high bitrate and low bitrate environments, to target specific operating systems and/or platforms such as Windows™ or Mac OS, etc.

[0032] Policy server 116 may associate the different versions of a content object with a single network identifier such as a uniform resource locator (URL). The single network identifier can then be returned to the publisher 106 which created the content. The publisher 106 can add the network identifier to one or more content sites 108 which are accessible to end users 122. When a request for the content object is received from an end user device, it can be sent to policy server 116 for analysis. Using all available information, policy server 116 can determine a preferred version of the content object for the end user device and can orchestrate its delivery to the requesting end user. The preferred version and delivery method can be customized for hardware and software capabilities of the end user device, bandwidth and connection quality, viewing habits, user preferences, or any combination of factors. The preferred version may also include a selection of advertisements which are matched to information about the end user device and/or the end user.

[0033] As described herein, policy server 116 provides publishers 106 with a one-to-many approach to optimized content delivery. Specifically, a single network identifier can point to multiple versions of a given content object from which policy server 116 selects a preferred version for use with a particular end user device. Policy server 116 thus relieves publishers 106
of the burden of staying up-to-date with technology. When a new platform emerges or device
capabilities change, appropriate versions of the content object can be made available to end users
122 through an existing network identifier without further effort from the publisher 106. Policy
server 116 determines the preferred version of a content object in a manner that is transparent to
the end user and thus avoids complicated configuration, specialized software, or manual
selection. The end user experience is further improved by selecting a delivery method and
sending the preferred version of the content object from a POP 118 location with a fast response
time for the user's location, network access, etc.

[0034] FIG. 2 shows one embodiment of a policy server 200. The present embodiment can be
as described in connection with policy server 116 and may be situated within a POP 118, an
origin server 112, or at some other location within the content distribution system 100 of FIG. 1.
Policy server 200 may include one or more hardware processors. In various embodiments,
policy server 200 also includes computer-readable media having encoded thereon data elements
and program instructions which, when executed by the one or more processors, cause the policy
server 200 to carry the operations described herein.

[0035] FIGS. 3, 4, 5, and 6 illustrate some exemplary data elements such as may be tangibly
recorded on computer readable media accessible to policy server 200. FIGS. 8-9 depict
exemplary processes such as may be carried out by the one or more processors of policy server
200 executing program instructions stored on such computer-readable media. Computer
readable media suitable for storing the data and program instructions may include magnetic
storage media including hard disk drives, optical storage media including optical discs, solid
state storage including flash-memory devices, and the like.

[0036] A publisher interface 210 of policy server 200 receives notification that a content object
is available for processing. The notification can be any type of network communication and may
be received directly from a publisher or from a CDN location. For example, a publisher 106 may
use an RSS (Really Simple Syndication) feed to signal that new content objects are available for
processing. Alternatively, the notification may be an event such as an AMQP (Advanced
Message Queuing Protocol) message received over WAN 114 when new content is added to a
file system of origin server 112, or other type of network communication.

[0037] When a notification is received, publisher interface 210 can provide information about
the publisher and the content object to a merge processor 220. Merge processor 220 retrieves the
content object over Internet 104 or wide area network 114 and is configured to create a plurality of versions of the content object. Creating versions of the content object can involve one or more transcoding, converting, splicing, truncating, padding, reformatting, resampling, repackaging, and related operations which may alter the media format and/or other attributes of the content object as originally created by the publisher 106.

[0038] Merge processor 220 can create different versions of the content object according to the requirements of each publisher 106. In one embodiment, merge processor 220 determines a set of production profiles with which to create the versions based on information received at publisher interface 210 or maintained at policy server 200. For instance, publishers 106 may specify various encoding parameters for use in creating the different versions of their content. The encoding parameters may be fashioned into profiles and stored at policy server 200.

[0039] FIG. 3 shows exemplary publisher data such as can be included in a publisher/profile database 230 of policy server 200. As shown, the publisher data maps various encoding profiles and advertising information to content sites 108. For instance, at the publisher level, Company A may require different encoding profiles and advertisements for use with its content objects than does Company B. In addition, at the site level, Company A may wish to provide the content from its news operations in different media formats and with different advertising material than content from its sports operations.

[0040] Using information from publisher interface 210, merge processor 220 may retrieve one or more publisher-specific and/or site-specific entries from database 230 with which to process content objects. In addition, merge processor 220 may make calls to an external entity to obtain advertising material ("creatives") for inclusion with the different versions of the content object based on advertising index values or other indicators. For example, merge processor 220 may be configured to make advertising calls over Internet 104 or WAN 114 to a hosted ad serving platform such as the DART for Publishers service from Google, Inc. or the Atlas AdManager service from Microsoft Corporation. The advertising calls may include additional publisher-specific indices or category information for matching the creatives to particular content.

[0041] FIG. 4 shows exemplary profile information such as can be included in database 230. As shown, each profile includes a collection of information with which to create versions of the content object. For example, among other things, each profile may include a codec, a container type, a frame size, a bit rate, a color depth, an audio format, etc. These and other parameters
may be optimized for use by different end user devices 120 under different operating conditions. Collections of encoding profiles can be matched to the capabilities of different end user devices 120 and/or different operating conditions. For example, profiles may be organized in database 230 into collections of iPhone profiles, Blackberry profiles, high-bandwidth profiles, low-bandwidth profiles, etc. Database 230 may also include one or more profiles which are representative of a lowest common denominator (LCD) of the end user devices 120 or of a particular category of end user devices, such as a generic mobile device profile.

[0042] When encoding profiles for a content object have been identified and the creatives are ready, merge processor 220 produces a plurality of versions of the content object. This can involve inserting creatives at different positions in the content object and generating versions of the content object having different combinations of creatives for each encoding profile. In some embodiments, a plurality of merge resources work in parallel to create versions of the content object for each profile-creative combination. For example, merge processor 220 may direct the operation of a server farm or cluster of computing resources in the large scale production of versions of the content object. As they become available, the content object versions may be placed into an object store 240 and readied for distribution to locations throughout the CDN.

[0043] Merge processor 220 associates each version of the content object with a single network identifier and, when processing of the content object is complete, may signal to publisher interface 210 that the network identifier is ready for distribution. Policy server 200 may then transmit the network identifier to publishers 106 for use at content sites 108. As described below, information about the network identifier is maintained in database 230 and used by a policy engine 260 to determine a preferred version of the content object for distribution to an end user system.

[0044] FIG. 5 shows exemplary network identifiers such as may be created by merge processor 220 and stored in database 230. Here, the network identifiers are universal-URLs (UURL). A universal-URL is a single URL that is associated with all versions of the content object available for distribution to end user devices. For example, all versions of the content object "homerun.mpg" (which may include versions created using encoding profiles E1, E2, and E3 with different combinations of creatives) are referenced by the single network identifier "http://company.x/sports/homerun.mpg." The various versions of content object
"interview.mov" are referenced by network identifier "http://company.x/news/interview.mov."
and so on.

[0045] When an end user 122 requests delivery of a content object using its UURL, the request
may be redirected to policy server 200 for selecting a preferred version of the requested content
object from the available versions and for orchestrating delivery of the preferred version to the
end user device. For example, the content request may be received at a POP 118 location in the
CDN and redirected to policy server 200 using HTTP redirects. GDN elements may also
forward information relating to the request ("request information") with the UURL for use by
policy server 200. Request information may include cookies, usage statistics, user profiles,
hardware identifiers, software identifiers, demographics information, viewing habits, content
preferences, group memberships, etc. The content request interface 250 receives the request
information and passes it along with the UURL to a policy engine 260.

[0046] Policy engine 260 is responsible for understanding the content and determining how to
orchestrate a response to ensure its proper delivery to the requesting end user device. Using the
network identifier, policy engine 260 can retrieve information from database 230 about the
publisher of the requested object, the versions of the content object available for delivery through
the GDN, and their corresponding encoding profiles and creatives. Using the request
information, policy engine 260 can select a preferred version of the content object for the end
user device and can determine an appropriate delivery scheme.

[0047] The request information may include a user-agent associated with the end user device.
The user agent, for example, may include one or more hardware and/or software identifiers
corresponding to device capabilities. For purposes of illustration, two examples of user agent
information are provided below.

(1) BlackBerry9530/4.7.0.148 Profile/MIDP-2.0
    Configuration/CLDC-1.1 VendorID/105

(2) Mozilla/5.0 (iPhone; U; CPU iPhone OS 3.0.1 like Mac OS X; en-us) AppleWebKit/525.18.1 (KHTML, like Gecko)
    Mobile/5H11

[0048] The first example (1) shows user agent information for a Blackberry mobile phone.
The device is identified as a Model 9530 running version 4.7.0.148 of the browser software. The
user-agent information also indicates that the device supports Java application programming

interfaces (CLDC-1.1) and includes a Vendor string (VendorID/105) indicating that "Verizon US" is the wireless provider. The second example (2) shows user-agent information for an iPhone. The hardware is identified as an iPhone mobile device with firmware version 5H1.1. Software capabilities indicate the iPhone is running version 3.0.1 of the iPhone OS, with build 525.18.1 of AppleWebkit core classes and a Mozilla-based browser.

[0049] In addition to user-agent information, policy engine 260 may also receive metadata, cookies, profiles, or other device/user information from content request interface 250. Cookies and metadata, for example, may be forwarded to policy server 200 from the CDN server where the request was received and may contain indications of user preference for one media format or another, viewing history, etc. Based on the totality of available information, policy engine 260 determines a preferred version of the content object and an orchestration scheme for arranging its delivery to the requesting end-user device.

[0050] FIG. 6 shows exemplary device information such as may be included in device database 270 and used by policy engine 260 to determine a preferred version and delivery orchestration scheme for responding to a content request. Device database 270 can include extensive details of hardware and software capabilities corresponding to a number of end user devices. For example, as shown, the device capabilities information can include processor, display screen, operating system, codec, protocol, and B2P handover information for each device. Information for populating device database 270 may be obtained directly from manufacturers and deduced from publicly available sources and may be updated as new devices and technologies appear in the marketplace.

[0051] Using the publisher information (FIG. 3) in combination with network identifier information (FIG. 5), and device capabilities information (FIG. 6), policy engine 260 can determine a preferred version of a requested content object from among the available versions. For example, if a user requests UURL "http://company.x/sports/homerun.mpg," policy engine 260 may determine that the requested object is an asset of Company X and that three versions (El, E2, E3) of the content object are available for distribution through the CDN. Policy engine 260 can then cross-reference device capabilities with the encoding profiles to identify a subset of the available versions best suited for use with the requesting end-user device. The subset of versions can then be ranked in accordance with the request information and publisher requirements to identify the preferred version based on the best information available.
In the "homerun.mpg" example, if the user-agent information indicates that the end user device is an iPhone™, the preferred version may correspond to encoding profile E1 since that profile matches the 480x360 native resolution of the iPhone™ device and includes a (segmented) MPEG-2 video format. Since the requesting device has identified itself as an iPhone™, policy engine 260 could select an HPD delivery orchestration scheme. However, because it is known that the requesting iPhone™ is also running OS 3.x, the device capabilities information indicates that iPhone™ HTTP Streaming (1HS) is also available.

IHS delivery may be preferred by the publisher, it may match the end user’s preferences, or it may represent the default orchestration scheme for iPhones running OS 3.x. Continuing with the example, policy engine 260 may select as the preferred version of "homerun.mpg" one that is created with encoding profile E1 and that includes a selection of creatives based on the request information. Note that the version of the content object selected as the preferred version and the delivery orchestration scheme may change based on the hardware and software capabilities of the requesting device. For example, if an iPhone™ is detected as running version 2.x of the Apple operating system, policy engine 260 may instead choose HPD orchestration and a version of the content object having an MPEG-4 video format.

Of course, many variations are possible in determining the preferred version and delivery orchestration scheme. For example, the request information may include a type of connection for the end user device. The type of connection may be determined based on information received from the end user device, based on an IP address of the end user device, based on an entry point to the CDN, or any other means and may influence orchestration and the selection of the preferred version. In a simple case, the type of network connection may be used to choose between high bit rate and low bit rate versions of the content object. Cookies and metadata included with the request information may also influence selection of the preferred version of the content object. For example, cookies may provide an indication of user preference for media formats, bit rates, playback software, etc.

In some embodiments, policy engine 260 may determine a preferred version of the content object according to a hierarchy of information. For example, in descending order of priority, the hierarchy may include user preferences, special hardware capabilities, general hardware capabilities, special software capabilities, general software capabilities, and default settings. Policy engine 260 may perform a series of comparisons using available information.
and a predetermined order of precedence by which to identify the preferred version of the content object.

[0056] As one example, a user's preference for MOV media format with a QuickTime media player may take precedence over generalized software capabilities and default settings.

5 Similarly, versions of a content object which exploit special hardware capabilities, such as high resolution display screens, may be preferred to versions compatible with more general hardware capabilities; specialized software capabilities (such as 1HS streaming support) may take precedence over default HTTP delivery, and so on. Operation of policy engine 260 is not limited to a particular set of information, but may utilize any combination of user, publisher, network, encoding profile, and device capabilities information by which to determine a preferred content version and by which to orchestrate its delivery to the end user.

[0057] FIG. 7 shows an exemplary interaction among elements of a content distribution system including a publisher, a policy server, a CDN edge server, and an end user device. The elements participating in exemplary interaction may be as described in connection with the content distribution system 100 of FIG. 1, including the policy server 200 of FIG. 2. For purposes of discussion, the interaction is divided into an ingest part (arrows 1-3) and a delivery part (items 4-6). The ingest part of the interaction begins at arrow (1) when the policy server receives notification of a new content object and is now described in connection with FIG. 8.

[0058] FIG. 8 shows an exemplary ingest process 800 such as may be performed by policy server 200 at this stage of the interaction. Some or all of ingest process 800 may be performed by one or more processors executing fixed or programmable instructions which, in turn, may be stored with related data elements on a computer-readable medium.

[0059] At block 810, the policy server retrieves publisher information and encoding profiles for the new content object. The publisher information may be similar to that described in connection with FIG. 3 and may specify a set of encoding profiles and advertising identifiers such as shown in FIG. 4 for processing the media assets of a particular content site.

[0060] At block 820, the policy server uses the advertising identifiers to communicate with ad servers. For instance, the policy server may transmit the advertising identifiers and other information specific to the content site to one or more ad servers using their respective application programming interfaces (API). Based on the responses from the ad servers, at block
830, creatives are obtained for inclusion with new versions of the content object. With audio and
video content, the response to the ad call may also include information to guide placement of the
creatives within the content object. For example, different types of creatives may be designated
for insertion at pre-roll, mid-roll, and post-roll positions in the content object received from the
publisher.

[0061] At block 840, the policy server generates, or causes to be generated, a plurality of
versions (1...N) of the content object. The versions may be created by transcoding, converting,
merging, splitting, truncating, padding, resampling, resizing, reformatting, repackaging, filtering,
and otherwise transforming and possibly modifying the original content object according to the
collection of encoding profiles associated with publisher and content site. During processing,
advertising material can be inserted or merged into each version of the content object at the
specified positions and, for each encoding profile, versions may be created using different
combinations of advertising material. Note that the N versions of the content object may be
created in a parallel fashion and that processing may be distributed across a collection of merge
computing resources to minimize the time to completion.

[0062] At block 850, the policy server associates the N versions of the content object with a
single network identifier. The network identifier can be, for example, a universal URL (UURL)
as described in connection with FIG. 5, or other identifier having a one-to-many relationship to
the versions of the content object. When the versions of the content object are ready, at block
860, the policy server may distribute them to locations in the CDN such as origin servers 112 or,
in some cases, to various POP locations as shown by arrow (2). The ingest part ends when the
policy server returns the network identifier to the publisher for inclusion on its content sites as
shown with arrow (3). However, additional versions of the content object may be generated as
new hardware and software capabilities emerge or when a publisher’s requirements change.
Because the additional versions are associated at the policy server with a previously published
network identifier, updates to the respective content sites are not required.

[0063] The delivery part of the interaction begins with arrow (4) when the end user device
requests a content object using the network identifier (UURL) and is now described in
connection with FIG. 9.

[0064] FIG. 9 shows an exemplary delivery process 900 such as may be performed by policy
server 200. Some or all of delivery process 900 may be performed by one or more processors.
executing fixed or programmable instructions which, in turn, may be stored with related data elements on computer-readable media.

[0065] At block 910, a request for a content object with a UURL is received. For example, with a UURL, an end user may click on a link at a publisher's web site and be redirected to a nearby CDN location. This can be accomplished through DNS resolution or in some other manner. As shown using arrow (5), the CDN location may forward the content request along with any related information such as the IP address of the end user device, hardware and software identifiers, usage statistics, cookies, metadata, profiles, etc. to the policy server for selection of a preferred version of the requested content.

[0066] At block 920, the policy server identifies versions of the content object associated with the UURL. As previously described, a multitude of versions may have been generated during the ingest process and each version may correspond to an encoding profile and a set of creatives. The encoding profiles, in turn, may be optimized for use with different hardware and software platforms under different network conditions, and the creatives may be selected according to the type of content, with different groups of creatives being designated for different target audiences of the content object.

[0067] At block 930, the policy server determines capabilities of the requesting end user device. The capabilities may be determined using device identifiers accompanying the content request or other sources of information available through the CDN. For instance, device information may be retrieved from a device capabilities database with information from a user-agent string, cookies stored on the end user's computer, records of previous interactions with the CDN, etc.

[0068] At block 940, a preferred version of the content object is selected based on the various sources of information. As shown by arrow (6), the policy server also determines a delivery orchestration scheme, block 950, for sending the preferred version of the content object to the requesting end user. Both the preferred version of the content object and the delivery orchestration scheme selected by the policy server may reflect a ranking of device capabilities, user preferences, publisher requirements, network conditions, and other factors relating to overall user experience of the requested content as these elements are variously described herein.
At block 960, the policy server orchestrates delivery of the preferred version of the content object according to the selected delivery option. As shown by a arrow (7), orchestration may involve multiple delivery protocols. For example, delivery orchestration schemes which may be utilized by the policy server include HTTP progressive download (HPD), RTSP streaming delivery (RSD), RTSP via HTML stub (RSDH), and iPhone HTTP streaming (1HS). As described below in connection with FIG. 10, orchestration may include, for example, coordinating with other elements of the CDN, redirecting an HTTP content request, transitioning from HTTP to RTSP, providing a stub or helper file, providing a master playlist, providing a bandwidth-specific playlist, or other combination of actions.

FIG. 10A shows aspects of an exemplary HPD delivery orchestration scheme involving a policy server, a CDN edge server, and an end user device. The exemplary HPD orchestration scheme may be used by a policy server in the content distribution system 100 of FIG. 1 in response to a request for content.

The interaction begins with arrow (1) when the end user device requests a content object using its universal URL. As shown, the end user device may send an HTTP request for universally-identified content object C. The UURL may resolve to a CDN location that is best able to serve the end user device. At arrow (2), the CDN edge server forwards the HTTP request along with request information such as user profiles, device identifiers, cookies, metadata, network connection, etc. to the policy server.

The policy server processes the UURL request and the additional request information received from the CDN edge server and determines a preferred version (C.11) of the content object for delivery to the end user computer. At arrow (3), the policy server generates an HTTP redirect 302 in which information about version C.11 is added to the location header. At arrow (4), the CDN edge server performs a second HTTP redirect to pass the updated location header to the end user device. The end user device makes a second HTTP request at arrow (5) for content object C.11 using the location header information. The CDN edge server responds to the second HTTP request at arrow (6) and delivers the preferred C.11 version of the content object. In an alternative embodiment, the exchange shown by arrows (4) and (5) is omitted and the edge server responds directly to the end user device with version C.11 of the content object.

FIG. 10B shows aspects of an exemplary RSD delivery orchestration scheme involving a policy server, a CDN edge server, and an end user device. The exemplary RSD orchestration
scheme may be selected by a policy server in the content distribution system 100 of FIG. 1 in response to a request for content.

[0074]  The interaction begins with arrow (1) when the end user device requests a content object using its universal URL. As with HPD delivery, the device may send an HTTP request for content object C which may resolve to a CDN point of presence and be assigned to a CDN edge server. At arrow (2), the CDN edge server forwards the HTTP request along with related information such as user profiles, device identifiers, cookies, metadata, connecting network, etc. to the policy server.

[0075]  The policy server processes the UURL request and any additional information received from the CDN edge server to determine a preferred version (C.9) of the content object for delivery to the end user device. In this case, RTSP streaming delivery is selected. The selection of RTSP delivery may be based on user preferences, publisher preferences, software capabilities, or any number of factors.

[0076]  At arrow (3), the policy server generates an HTTP redirect (302) in which an RTSP URL corresponding to version C.9 is added to the location header. At arrow (4), the CDN edge server performs a further HTTP redirect by which the updated location header is passed to the end user device. At arrow (5), the end user device makes an RTSP request for content object C.9 using header information from the redirected HTTP request. A streaming CDN edge server responds to the RTSP request at arrow (6) and streams version C.9 of the requested content object to the end user device.

[0077]  FIG. 10C shows aspects of an exemplary RSDH delivery orchestration scheme involving a policy server, a CDN edge server, and an end user device. The exemplary RSDH orchestration scheme may be selected by a policy server in the content distribution system 100 of FIG. 1 in response to a request for content.

[0078]  The interaction begins with arrow (1) when the end user device requests a content object using its universal URL. The request for content may be an HTTP request which resolves to the address of a CDN edge server. At arrow (2), the CDN edge server forwards the HTTP request along with related information such as user profiles, device identifiers, cookies, metadata, connecting network, etc. to the policy server.
At arrow (3), the policy server generates an HTTP redirect (302) to an HTML stub wrapper. The HTML stub wrapper may include, within its HTML body content, an RTSP URL of the preferred version of the content object and certain directives to force the end user device to invoke a media player application. This type of redirection may be used, for example, to facilitate transition of control from a browsing application to a media player environment when the policy server determines that the end user device lacks the appropriate capabilities.

At arrow (4), the CDN edge server relays the HTTP request referencing the HTML stub wrapper to the end user device. At arrow (5), the end user device makes an HTTP request for the HTML stub wrapper. At arrow (6), the CDN edge server fulfills the request and transfers the HTML stub wrapper to the end user device. The HTML stub wrapper contains the RTSP URL for the C.2 version and causes the end user device to invoke its media player application. At arrow (7), responsive to the HTML stub wrapper, the end user device invokes its media player and requests an RTSP stream for C.2. At arrow (8), a streaming CDN edge server responds to the RTSP request and streams version C.2 of the content object to the end user device.

FIG. 10D shows an exemplary 1HS delivery orchestration scheme involving a policy server, a CDN edge server, and an end user device. The exemplary 1HS orchestration scheme may be selected by a policy server in the content distribution system 100 of FIG. 1 in response to a request for content.

The interaction begins at arrow (1) when the end user device requests a content object using its universal URL. As with the other delivery methods, the device may send an HTTP request for content object C which is assigned to a CDN edge server through a DNS resolution process. At arrow (2), the CDN edge server forwards the HTTP request along with related information such as user profiles, device identifiers, cookies, metadata, connecting network, etc. to the policy server.

Apple iPhones™ running iPhone OS 3.0 and higher include a built-in Quicktime player which supports streaming video over HTTP using a combination of M3U8 playlists and MPEG-2 segmented video. This approach enables the end user device to select between different versions of a preferred content object (for example, versions with different bitrates) and to switch between the different versions based on how much network bandwidth is available. Due to this flexibility, the policy server may prefer 1HS delivery for use with suitable iPhone™ end user devices.
When 1HS delivery of preferred content is selected, the policy server generates an HTTP redirect (302) to a master playlist. Here, the master playlist includes references to multiple versions (C.15 - C.20) of the preferred content which may, as previously indicated, correspond to multiple bandwidth choices. At arrow (3), the policy server sends the redirected request containing the master playlist to the CDN edge server; at arrow (4), it is returned to the end user device.

At arrow (5), the Quicktime player selects from the master playlist and requests a playlist of media segments corresponding to its selection. In this example, the iPhone™ selects version C.17 from the versions presented in the master playlist and issues a request at item (5) for the C.17 playlist. Version C.17, for example, may represent the highest bandwidth version for prevailing network conditions. At item (6), the CDN edge server responds by providing a playlist for C.17 to the iPhone. Thereafter, during playback of the content, the iPhone requests media segments for C.17 and the CDN edge server delivers the requested segments as is shown by items (7)-(8).

As will be understood by those skilled in the art, the present invention may be embodied in other specific forms. In one particular embodiment of the policy server, a new content object is assigned a single network identifier which, as previously described, can be associated with a plurality of versions of the content object. However, the versions of the content object need not be created prior to receiving a request from an end-user device. Instead, the different versions of the content object may be created dynamically at the direction of the policy server when a request is received. As described herein, the full set of available information (user-agent, preferences, network identifiers, cookies, metadata, usage information, etc) can be utilized to generate a preferred version of the content object for the requesting end-user device. As part of this on-demand process, the newly created preferred version of the content object is associated with the single network identifier previously assigned to the new content object and it may be placed into an object store and used with subsequent requests. Those skilled in the art will recognize, or be able to ascertain using no more than routine experimentation, many equivalents to the specific embodiments of the invention described herein. Such equivalents are intended to be encompassed by the following claims.
WHAT IS CLAIMED IS:

1. A method of delivering content objects to an end user device, comprising:
   - receiving notification at a policy server that a content object is available for processing;
   - associating a network identifier with a plurality of versions of the content object, each version of the content object being created with a corresponding encoding profile in a set of encoding profiles;
   - sending the network identifier to a publisher of the content object;
   - receiving a request for the content object comprising the network identifier and request information relating to capabilities of the end user device;
   - determining a preferred version of the content object based on the encoding profiles and the request information;
   - determining a delivery orchestration scheme for sending the preferred version of the content object to the end user device; and
   - sending the preferred version of the content object to the end user device in accordance with the delivery orchestration scheme.

2. The method of claim 1, wherein the plurality of versions of the content object comprise a plurality of audio/video media files.

3. The method of claim 1, further comprising causing the plurality of versions of the content object to be created in response to the notification.

4. The method of claim 1, wherein each encoding profile in the set of encoding profiles comprises a plurality of encoding parameters including a bitrate, a frame size, and a media format.

5. The method of claim 4, wherein the encoding parameters are related to end user device capabilities.

6. The method of claim 1, wherein the request information comprises at least one identifier of the end user device.
7. The method of claim 6, wherein the at least one Identifier comprises a hardware identifier, a software identifier, or a network connection of the end user device.

8. The method of claim 6, wherein the at least one identifier comprises a hardware identifier, the method further comprising:
   determining a display-capability of the end user device at the policy server based on the hardware identifier; and
   determining the preferred version of the content object based on the display capability.

9. The method of claim 6, wherein the device capabilities comprise a network connection of the end user device, the method further comprising:
   determining a bandwidth capability of the end user device based on the network connection; and
   determining the preferred version of the content object based on the bandwidth capability.

10. The method of claim 1, further comprising:
   detecting a playback application of the end user device to receive the preferred version of the content object; and
   directing a protocol transition based on the playback application.

11. The method of claim 10, wherein the protocol transition comprises a transition from HTTP to BTSP (real-time streaming protocol).

12. The method of claim 10, wherein the protocol transition comprises a transition from HTTP to IBS (iPhone HTTP streaming).

13. A policy server, comprising:
   a publisher interface configured to receive notification that a content object is available for processing;
   a merge processor configured to associate a network identifier with a plurality of versions of the content object, each version of the content object being created with a corresponding encoding profile in a set of encoding profiles,
the merge processing further configured to provide the network identifier to the
publisher interface for delivery to a publisher of the content object when the plurality of versions
of the content object is available for distribution;

a content request interface configured to receive an HTTP (hyper-text transport
protocol) request for the content object, the request comprising the network identifier and request
information relating to capabilities of an end user device; and

a policy engine configured to determine a preferred version of the content object
for distribution to the end user device based on the encoding profiles and the request information,
the policy engine further configured to determine a delivery orchestration scheme
and to cause the preferred version to be sent to the end user device in accordance with the
delivery orchestration scheme.

14. The policy server of claim 13, wherein the plurality of versions of the
content object comprise a plurality of audio/video media files.

15. The policy server of claim 13, wherein the merge processor comprises
merge resources configured to create the plurality of versions of the content object in accordance
with the set of encoding profiles responsive to receiving the notification at the publisher
interface.

16. The policy server of claim 13, wherein each encoding profile in the set of
encoding profiles comprises a plurality of encoding parameters including a bitrate, a frame size,
and a media format.

17. The policy server of claim 16, wherein the encoding parameters are
related to end user device capabilities.

18. The policy server of claim 13, wherein the request information comprises -
at least one identifier of the end user device.

19. The policy server of claim 18, wherein the at least one identifier comprises
a hardware identifier, a software identifier, a display resolution, or a connection type of the end
user device.

20. The policy server of claim 18, wherein the at least one identifier comprises
a hardware identifier and wherein the policy engine is configured to determine a display
capability of the end user device based on the hardware identifier and to determine the preferred
version of the content object based on the display capability.

21. The policy server of claim 18, wherein the at least one identifier comprises
a network connection of the end user device, and wherein the policy engine is configured to
determine a bandwidth capability of the end user device based on the network connection and to
determine the preferred version of the content object based on the bandwidth capability.

22. The policy server of claim 13, wherein the policy engine is configured to
detect a playback capability of the end user device and to direct a protocol transition based on the
playback capability.

23. The policy server of claim 22, wherein the protocol transition comprises a
transition from HTTP to RTSP (real-time streaming protocol), and wherein the end user device is redirected via HTTP to an RTSP URL (universal resource locator) in connection with the
protocol transition.

24. The policy server of claim 22, wherein the protocol transition comprises a
transition from HTTP to IMS (iPhone HTTP streaming).

23. The policy server of claim 22, wherein the protocol transition comprises a
transition from HTTP to IMS (iPhone HTTP streaming).

26. A method of distributing content objects in a content delivery network
(CDN), comprising:

    receiving notification at a policy server that a content object is available for
    processing;
    creating a plurality of versions of the content object with a plurality of
corresponding encoding profiles responsive to the notification;
    associating the plurality of versions of the content object with a network identifier
    of the CDN;
sending the network identifier from the policy server to a publisher of the content object;
receiving a request for the content object, the request comprising the CON network identifier and request information;
determining device capabilities of the end user device based on the request information;
determining a preferred version of the content object for the end user device based on the device capabilities; and
sending the preferred version of the content object to the end user device from a CDN location corresponding to the CDN network identifier,

27. The method of claim 26, wherein the plurality of versions of the content object comprise a plurality of audio/video media files.

28. The method of claim 26, further comprising:
determining a delivery orchestration scheme based on the device capabilities; and
sending the preferred version of the content object to the end user device using the delivery orchestration scheme.

29. The method of claim 26, wherein the request information comprises at least one of device identifiers, usage statistics, cookies, or user preferences associated with the end user device.

30. The method of claim 26, wherein creating the plurality of versions of the content object comprises adding advertising material to the content object.

31. The method of claim 30, wherein the advertising material is obtained in response to a call to an internet-based advertising service.
### FIG. 3

<table>
<thead>
<tr>
<th>Publisher</th>
<th>Site</th>
<th>Encoding Profiles</th>
<th>Advertising Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company A</td>
<td>companyA.sports.com</td>
<td>E1, E2, E3</td>
<td>ID1, ID2, ID4</td>
</tr>
<tr>
<td>Company A</td>
<td>companyA.news.com</td>
<td>E5, E6, E9</td>
<td>ID2, ID5, ID17</td>
</tr>
<tr>
<td>Company B</td>
<td>companyB.com</td>
<td>E1-7</td>
<td>ID3, ID5, ID16</td>
</tr>
<tr>
<td>Company C</td>
<td>companyC.com</td>
<td>E2-5, E17, E23</td>
<td>ID1, ID2, ID15, ID27</td>
</tr>
</tbody>
</table>

### FIG. 4

<table>
<thead>
<tr>
<th>Profile</th>
<th>Codec</th>
<th>Container</th>
<th>Frame size</th>
<th>Bit rate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>QT</td>
<td>MPEG2</td>
<td>480x369</td>
<td>900 kbps</td>
<td>Mobile, high bitrate, streaming</td>
</tr>
<tr>
<td>E2</td>
<td>H.264</td>
<td>3GP</td>
<td>480x320</td>
<td>124 kbps</td>
<td>Streaming</td>
</tr>
<tr>
<td>E3</td>
<td>MP3</td>
<td>MP3</td>
<td>---</td>
<td>64 kbps</td>
<td>Podcast</td>
</tr>
<tr>
<td>LCD</td>
<td>MPEG4</td>
<td>3GP</td>
<td>176x144</td>
<td>100 kbps</td>
<td>Generic mobile device</td>
</tr>
<tr>
<td>Object</td>
<td>Processed</td>
<td>Profiles</td>
<td>UURL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
<td>-------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>homerun.mpg</td>
<td>3.16.09</td>
<td>E1, E2, E3</td>
<td><a href="http://company.x/sports/homerun.mpg">http://company.x/sports/homerun.mpg</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interview.mov</td>
<td>3.16.09</td>
<td>E4, E6</td>
<td><a href="http://company.x/news/interview.mov">http://company.x/news/interview.mov</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>popvideo.wmv</td>
<td>...</td>
<td>E1-5, E9-12</td>
<td>Pending...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIG. 5**
<table>
<thead>
<tr>
<th>Device ID</th>
<th>Processor</th>
<th>Screen Size</th>
<th>Color Depth</th>
<th>OS</th>
<th>Media/Codec</th>
<th>Media Delivery</th>
<th>Media Orch. Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown</td>
<td>1000 MHz</td>
<td>480x800</td>
<td>16M</td>
<td>Android 2.1</td>
<td>H.264, AAC, MPEG4</td>
<td>HTTP, RTSP</td>
<td>RSD</td>
</tr>
<tr>
<td>HTC Nexus One</td>
<td>600 MHz</td>
<td>480x320</td>
<td>16M</td>
<td>iPhone OS 3.x</td>
<td>H.264, AAC, MPEG4</td>
<td>HTTP, RTSP</td>
<td>RSD</td>
</tr>
<tr>
<td>iPhone/OS 2.x</td>
<td>600 MHz</td>
<td>480x320</td>
<td>16M</td>
<td>iPhone OS 3.x</td>
<td>H.264, AAC, MPEG4</td>
<td>HTTP, RTSP</td>
<td>RSD</td>
</tr>
<tr>
<td>BlackBerry Storm 9550</td>
<td>528 MHz</td>
<td>480x320</td>
<td>64k</td>
<td>BlackBerry OS</td>
<td>H.263, H.264, MP3, AAC</td>
<td>RTSP</td>
<td>RSDH</td>
</tr>
<tr>
<td>BlackBerry Bold 9000</td>
<td>624 MHz</td>
<td>480x320</td>
<td>64k</td>
<td>BlackBerry OS</td>
<td>H.263, H.264, MP3, AAC</td>
<td>RTSP</td>
<td>RSDH</td>
</tr>
</tbody>
</table>
FIG. 7
Start

Retrieve publisher information and encoding profiles for new content

Make call to external ad server

Obtain creatives for merge/insertion into content object

Generate version of content object for each profile/creative combination

Associate all content object versions with network identifier

Send universal network identifier to publisher for publication to end users

End

FIG. 8
Start

Receive request for content object with UURL

Determine publisher of content object and available versions

Determine device capabilities

Select preferred version of content object

Determine delivery orchestration scheme

Orchestrate delivery of content object

End
FIG. 10A
Diagram of network interaction:

1. End User Device requests URL(C).
2. CDN Edge Server requests info.
3. Redirect RTSP URL(C.9).
4. Relay RTSP URL(C.9).
5. RTSP Request(C.9).

FIG. 10B
FIG. 10C
FIG. 10D