TRICK MODES FOR NETWORK STREAMING OF CODED VIDEO DATA

TECHNICAL FIELD

[0001] This disclosure relates to storage and transport of encoded multimedia data.

BACKGROUND

[0002] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, digital cameras, digital recording devices, digital media players, video gaming devices, video game consoles, cellular or satellite radio telephones, video teleconferencing devices, and the like. Digital video devices implement video compression techniques, such as those described in the standards defined by MPEG-2, MPEG-4, ITU-T H.263 or ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), and extensions of such standards, to transmit and receive digital video information more efficiently.

[0003] Video compression techniques perform spatial prediction and/or temporal prediction to reduce or remove redundancy inherent in video sequences. For block-based video coding, a video frame or slice may be partitioned into macroblocks. Each macroblock can be further partitioned. Macroblocks in an intra-coded (I) frame or slice are encoded using spatial prediction with respect to neighboring macroblocks. Macroblocks in an inter-coded (P or B) frame or slice may use spatial prediction with respect to neighboring macroblocks in the same frame or slice or temporal prediction with respect to other reference frames.

[0004] After video data has been encoded, the video data may be packetized for transmission or storage. The video data may be assembled into a video file conforming to any of a variety of standards, such as the International Organization for Standardization (ISO) base media file format and extensions thereof, such as ITU-T H.264/AVC. Such packetized video data may be transported in a variety of ways, such as transmission over a computer network using network streaming.

SUMMARY

[0005] In general, this disclosure describes techniques for improving streaming of media data over a network. These techniques include support for trick modes, such as

fast forward, rewind, and seeking within media content streamed over a network. These techniques also include support for representation groups, such as signaling common characteristics for a group of representations, as well as individual characteristics of the representations. Moreover, the techniques include providing information for updating manifest files for streamed media content. The techniques also include providing media data for targeted advertisements as external periods for media content. These techniques further include providing and interpreting quality of experience reports from a client device to a service provider. Furthermore, these techniques include signaling of profile data to which a manifest file of media content conforms.

[0006] In one example, a method of retrieving video data includes analyzing at least a portion of a manifest file for multimedia content, wherein the portion of the manifest file includes information indicative of sets of representations of the multimedia content and information indicative of common characteristics for each of the sets of representations, selecting one of the sets of representations based on the common characteristics for the one of the sets of representations, selecting one of the representations of the selected one of the sets of representations based on one or more coding characteristics of the one of the representations of the one of the sets, and generating a request for data of the one of the representations based on the selection.

[0007] In another example, a device for receiving information for video data includes one or more processors configured to analyze at least a portion of a manifest file for multimedia content, wherein the portion of the manifest file includes information indicative of sets of representations of the multimedia content and information indicative of common characteristics for each of the sets of representations, select one of the sets of representations based on the common characteristics for the one of the sets of representations, select one of the representations of the selected one of the sets of representations based on one or more coding characteristics of the one of the representations of the one of the sets, and generate a request for data of the one of the representations based on the selection.

[0008] In another example, a device for receiving information for video data includes means for analyzing at least a portion of a manifest file for multimedia content, wherein the portion of the manifest file includes information indicative of sets of representations of the multimedia content and information indicative of common characteristics for each of the sets of representations, means for selecting one of the sets of representations based on the common characteristics for the one of the sets of representations, means for

selecting one of the representations of the selected one of the sets of representations based on one or more coding characteristics of the one of the representations of the one of the sets, and means for generating a request for data of the one of the representations based on the selection.

[0009] In another example, a computer program product includes a computer-readable storage medium comprises instructions that, when executed, cause a processor of a device for retrieving video data to analyze at least a portion of a manifest file for multimedia content, wherein the portion of the manifest file includes information indicative of sets of representations of the multimedia content and information indicative of common characteristics for each of the sets of representations, select one of the sets of representations based on the common characteristics for the one of the sets of representations, select one of the representations of the selected one of the sets of representations based on one or more coding characteristics of the one of the representations of the one of the sets, and generate a request for data of the one of the representations based on the selection.

[0010] In another example, a method of sending information for video data includes obtaining a set of representations of multimedia content having one or more common characteristics, wherein each of the representations in the set have one or more individual coding characteristics separate from the common characteristics, obtaining a manifest file for the multimedia content, wherein the manifest file includes information indicative of the representations in the set, information indicative of the common characteristics for the set of representations, and information indicative of the coding characteristics for each of the representations in the set, and sending at least a portion of the manifest file to a client device.

[0011] In another example, a device for sending information for video data, the device comprising one or more processors configured to obtain a set of representations of multimedia content having one or more common characteristics, wherein each of the representations in the set have one or more individual coding characteristics separate from the common characteristics, obtaining a manifest file for the multimedia content, wherein the manifest file includes information indicative of the representations in the set, information indicative of the common characteristics for the set of representations, and information indicative of the coding characteristics for each of the representations in the set, and send at least a portion of the manifest file to a client device.

[0012] In another example, a device for sending information for video data includes means for obtaining a set of representations of multimedia content having one or more common characteristics, wherein each of the representations in the set have one or more individual coding characteristics separate from the common characteristics, means for obtaining a manifest file for the multimedia content, wherein the manifest file includes information indicative of the representations in the set, information indicative of the common characteristics for the set of representations, and information indicative of the coding characteristics for each of the representations in the set, and means for sending at least a portion of the manifest file to a client device.

[0013] In another example, a computer program product includes a computer-readable storage medium comprises instructions that cause a processor of a device for providing video data to obtain a set of representations of multimedia content having one or more common characteristics, wherein each of the representations in the set have one or more individual coding characteristics separate from the common characteristics, obtain a manifest file for the multimedia content, wherein the manifest file includes information indicative of the representations in the set, information indicative of the common characteristics for the set of representations, and information indicative of the coding characteristics for each of the representations in the set, and send at least a portion of the manifest file to a client device.

[0014] In another example, a method of retrieving video data includes analyzing information of a manifest file for multimedia content, wherein the information of the manifest file indicates that at least one representation of the multimedia content includes a temporal sub-sequence, determining one or more locations of data for the temporal sub-sequence, and submitting one or more requests for the data for the temporal sub-sequence.

[0015] In another example, a device for retrieving video data includes one or more processors configured to analyze information of a manifest file for multimedia content, wherein the information of the manifest file indicates that at least one representation of the multimedia content includes a temporal sub-sequence, determine one or more locations of data for the temporal sub-sequence, and submit one or more requests for the data for the temporal sub-sequence.

[0016] In another example, a device for retrieving video data includes means for analyzing information of a manifest file for multimedia content, wherein the information of the manifest file indicates that at least one representation of the

multimedia content includes a temporal sub-sequence, means for determining one or more locations of data for the temporal sub-sequence, and means for submitting one or more requests for the data for the temporal sub-sequence.

[0017] In another example, a computer program product includes a computer-readable medium having stored thereon instructions that, when executed, cause a processor of a device for retrieving video data to analyze information of a manifest file for multimedia content, wherein the information of the manifest file indicates that at least one representation of the multimedia content includes a temporal sub-sequence, determine one or more locations of data for the temporal sub-sequence, and submit one or more requests for the data for the temporal sub-sequence.

[0018] In another example, a method of sending information for video data includes obtaining data for at least one representation of multimedia content that includes a temporal sub-sequence, obtaining data for a manifest file for the multimedia content, wherein information of the manifest file indicates that the at least one representation of the multimedia content includes the temporal sub-sequence, and sending at least a portion of the manifest file to a client device.

[0019] In another example, a device for sending information for video data includes one or more processors configured to obtain data for at least one representation of multimedia content that includes a temporal sub-sequence, obtain data for a manifest file for the multimedia content, wherein information of the manifest file indicates that the at least one representation of the multimedia content includes the temporal sub-sequence, and send at least a portion of the manifest file to a client device.

[0020] In another example, a device for sending information for video data includes means for obtaining data for at least one representation of multimedia content that includes a temporal sub-sequence, means for obtaining data for a manifest file for the multimedia content, wherein information of the manifest file indicates that the at least one representation of the multimedia content includes the temporal sub-sequence, and means for sending at least a portion of the manifest file to a client device.

[0021] In another example, a computer program product includes a computer-readable medium having stored thereon instructions that, when executed, cause a processor of a device for sending information for video data to obtain data for at least one representation of multimedia content that includes a temporal sub-sequence, obtain data for a manifest file for the multimedia content, wherein information of the manifest file

indicates that the at least one representation of the multimedia content includes the temporal sub-sequence, and send at least a portion of the manifest file to a client device.

[0022] In another example, a method of retrieving video data includes retrieving data of a first segment of a representation of multimedia content in accordance with data of a copy of a manifest file stored by a client device, retrieving a portion of a second segment of the representation in accordance with the manifest file, wherein the second segment occurs after the first segment in the representation, and wherein the portion of the second segment indicates that the manifest file is to be updated, updating the copy of the manifest file stored by the client device based on the indication that the manifest file is to be updated, and retrieving media data of the second segment in accordance with the updated manifest file.

[0023] In another example, a device for retrieving video data includes one or more processors configured to retrieve data of a first segment of a representation of multimedia content in accordance with data of a copy of a manifest file stored by the device, retrieve a portion of a second segment of the representation in accordance with the manifest file, wherein the second segment occurs after the first segment in the representation, and wherein the portion of the second segment indicates that the manifest file is to be updated, update the copy of the manifest file stored by the device based on the indication that the manifest file is to be updated, and retrieve media data of the second segment in accordance with the updated manifest file.

[0024] In another example, a device for retrieving video data includes means for retrieving data of a first segment of a representation of multimedia content in accordance with data of a copy of a manifest file stored by the device, means for retrieving a portion of a second segment of the representation in accordance with the manifest file, wherein the second segment occurs after the first segment in the representation, and wherein the portion of the second segment indicates that the manifest file is to be updated, means for updating the copy of the manifest file stored by the device based on the indication that the manifest file is to be updated, and means for retrieving media data of the second segment in accordance with the updated manifest file.

[0025] In another example, a computer program product includes a computer-readable medium having stored thereon instructions that, when executed, cause a processor of a device for retrieving video data to retrieve data of a first segment of a representation of multimedia content in accordance with data of a copy of a manifest file stored by the

device, retrieve a portion of a second segment of the representation in accordance with the manifest file, wherein the second segment occurs after the first segment in the representation, and wherein the portion of the second segment indicates that the manifest file is to be updated, update the copy of the manifest file stored by the device based on the indication that the manifest file is to be updated, and retrieve media data of the second segment in accordance with the updated manifest file.

[0026] In another example, a method of sending information for video data includes sending data of a manifest file of multimedia content to a client device, wherein the manifest file includes information indicative of a first segment of a representation of the multimedia content, sending at least a portion of the first segment of the representation to the client device in response to a request from the client device, wherein the portion of the first segment indicates that the manifest file is to be updated, wherein an updated version of the manifest file includes information indicative of a second, different segment of the representation, and sending, in response to a request received from the client device and formed according to the updated manifest file, data of the second segment to the client device.

[0027] In another example, a device for sending information for video data includes one or more processors configured to send data of a manifest file of multimedia content to a client device, wherein the manifest file includes information indicative of a first segment of a representation of the multimedia content, send at least a portion of the first segment of the representation to the client device in response to a request from the client device, wherein the portion of the first segment indicates that the manifest file is to be updated, wherein an updated version of the manifest file includes information indicative of a second, different segment of the representation, and send, in response to a request received from the client device and formed according to the updated manifest file, data of the second segment to the client device.

[0028] In another example, a device for sending information for video data includes means for sending data of a manifest file of multimedia content to a client device, wherein the manifest file includes information indicative of a first segment of a representation of the multimedia content, means for sending at least a portion of the first segment of the representation to the client device in response to a request from the client device, wherein the portion of the first segment indicates that the manifest file is to be updated, wherein an updated version of the manifest file includes information indicative of a second, different segment of the representation, and means for sending, in response

to a request received from the client device and formed according to the updated manifest file, data of the second segment to the client device.

[0029] In another example, a computer program product includes a computer-readable medium having stored thereon instructions that, when executed, cause a processor of a device for sending information for video data to send data of a manifest file of multimedia content to a client device, wherein the manifest file includes information indicative of a first segment of a representation of the multimedia content, send at least a portion of the first segment of the representation to the client device in response to a request from the client device, wherein the portion of the first segment indicates that the manifest file is to be updated, wherein an updated version of the manifest file includes information indicative of a second, different segment of the representation, and send, in response to a request received from the client device and formed according to the updated manifest file, data of the second segment to the client device.

BRIEF DESCRIPTION OF DRAWINGS

[0030] FIG. 1 is a block diagram illustrating an example system that implements techniques for streaming media data over a network.

[0031] FIG. 2 is a conceptual diagram illustrating elements of an example multimedia content.

[0032] FIG. 3 is a block diagram illustrating elements of an example video file, which may correspond to a segment of a representation of multimedia content.

[0033] FIG. 4 is a conceptual diagram illustrating an example multimedia content including a media presentation description (MPD) and various representation groups.

[0034] FIG. 5 is a conceptual diagram illustrating another example multimedia content, in which MPD data is separated into various portions for various representation groups.

[0035] FIG. 6 is a conceptual diagram illustrating another example multimedia content that may be used to support trick modes.

[0036] FIG. 7 is a conceptual diagram illustrating another example multimedia content in which segments may include MPD update boxes to indicate that an MPD of the multimedia content is to be updated.

[0037] FIG. 8 is a flowchart illustrating an example method for providing indications of representation groups by a server device, and for selecting representation groups by a

client device as well as an individual representation within the selected representation group.

[0038] FIG. 9 is a flowchart illustrating an example method for providing data representative of a trick mode by a server device, and for using the data by a client device to retrieve and play trick mode data of multimedia content.

[0039] FIG. 10 is a flowchart illustrating an example method for providing, by a server device, indications that a manifest file, such as an MPD, is to be updated, and for updating an MPD by a client device.

[0040] FIG. 11 is a flowchart illustrating an example method for constructing and using data of a quality of experience (QoE) report document.

DETAILED DESCRIPTION

[0041] In general, this disclosure describes techniques for streaming multimedia data, such as audio and video data, over a network. The techniques of this disclosure may be used in conjunction with dynamic adaptive streaming over HTTP (DASH). This disclosure describes various techniques that may be performed in conjunction with network streaming, any or all of which may be implemented alone or in any combination. As described in greater detail below, various devices performing network streaming may be configured to implement the techniques of this disclosure.

[0042] In accordance with DASH and similar techniques for streaming data over a network, multimedia content (such as a movie or other audio/video content, which may also include text overlays or other data) may be encoded in a variety of ways and with a variety of characteristics. A content preparation device may form multiple representations of the same multimedia content. Each representation may correspond to a particular set of characteristics, such as coding and rendering characteristics, to provide data usable by a variety of different client devices with various coding and rendering capabilities. Moreover, representations having various bitrates may allow for bandwidth adaptation. That is, a client device may determine an amount of bandwidth that is currently available and select a representation based on the amount of available bandwidth, along with coding and rendering capabilities of the client device.

[0043] In some examples, a content preparation device may indicate that a set of representations has a set of common characteristics. The content preparation device may then indicate that the representations in the set form a representation group, in that

representations in the set can be used for bandwidth adaptation. That is, representations in the set may differ in bitrate, but otherwise share substantially the same characteristics. In this manner, a client device may determine various sets of common characteristics for representation groups of multimedia content, and select a representation group based on coding and rendering capabilities of the client device. Then, the client device may adaptively switch between representations in the selected representation group based on bandwidth availability.

[0044] The content preparation device may also provide separate network locations for different portions of a manifest file, such as a media presentation description (MPD) file in a format prescribed by 3GPP (Third Generation Partnership Project). That is, different portions of the manifest file may be independently addressable by, e.g., various uniform resource identifiers (URIs), such as uniform resource locators (URLs). An initial portion of the manifest file may include a URI, URL, or other location identifier of another portion of the manifest file. For example, a first portion of the manifest file may include descriptions of common characteristics of representation groups, as discussed above.

[0045] Each of the representation groups may be associated with a respective different portion of the manifest file, which may include data indicative of locations of media data of representations in the respective representation group. In this manner, a client device may receive the first portion of the manifest file, select an appropriate representation group, retrieve another portion of the manifest file for the selected representation group, select a representation of the selected group, and use the other portion of the manifest file to retrieve data of the selected representation. Moreover, the client device may adapt to changing network bandwidth using the other portion of the manifest file, that is, the portion specific to the selected representation group.

[0046] Additionally or alternatively, a portion of a manifest file may refer to another portion of the manifest file for other purposes. That is, a portion of the manifest file may direct a client device to another portion of the manifest file for inserting media data of a remote period into a movie during playback. The remote period may correspond to an advertisement, in some examples. These techniques may be used for targeted advertising, in some examples. A client device may provide user information, such as user identifier, user preferences for advertisements, and/or user demographic information, to a server device, which may select a portion of the manifest file based on the user information. Thus, when dereferenced, an external portion of the manifest file

may be incorporated into the original manifest file, e.g., by the client device. The server device may provide a location of the portion of the manifest file associated with targeted advertisement media content to the client device. The client device may then retrieve and present data of the targeted advertisement media content before retrieving data of a particular representation of a period of requested multimedia content. In this manner, a first portion of a manifest file for multimedia content may refer to a second portion of the manifest file.

[0047] In some instances, a user may wish to play video data in a manner other than from start to finish. For example, a user may wish to play video data in fast forward or rewind modes, or starting from a particular playback point. Such modes of video playback, which are modes other than playing from start to finish, may be referred to as "trick modes." In trick modes, because not all video data will ultimately be played, all of the video data need not be retrieved. This disclosure provides techniques for supporting trick modes as well. For example, a content preparation device may provide indications of byte range locations of frames in video data used for trick modes, such as instantaneous decoder refresh (IDR) pictures. In general, IDR pictures can be decoded without reference to data of any frames external to the IDR pictures themselves. Frames or slices of IDR pictures are generally coded in an intra-prediction mode, to avoid dependencies on other frames or slices. In this manner, the client device can retrieve information indicative of locations of IDR pictures to download only data for the IDR pictures for use in displaying video data in a trick mode, such as fast forward. Other data may be included in the temporal sub-sequence as well. Data may be arranged in coding order, such that data used for reference occurs earlier than (and in continuous byte sequence with) referencing data. For example, an I-frame may precede a P frame, which may proceed one or more B-frames, any or all of of which may precede other B-frames that may refer to the earlier B-frame in a hierarchical fashion.

[0048] In some examples, a manifest file, such as an MPD, may require occasional updates. This disclosure also provides techniques for signaling and receiving indications that an MPD requires an update. In particular, a content preparation device may include data in segments of representations indicating that a corresponding MPD requires an update. This data may correspond to an initial element of a segment, which may indicate the updates to apply to the MPD and/or locations from which a client device can retrieve updates to the MPD. The updates may comprise an entirely new MPD or incremental updates relative to a previous MPD for the multimedia content.

[0049] This disclosure further includes techniques for providing feedback from client devices to a server device and/or content preparation device. The feedback may correspond to, for example, information indicative of data that was retrieved for multimedia content. An administrator or other user of the content preparation device and/or server may use such information in a variety of ways. For example, the user may configure a content delivery network (CDN) to cache data of more frequently accessed representations in proxy devices of the CDN, such as routers or other devices. As another example, the user may determine representations that are accessed more frequently to determine whether certain representations should be added or removed to or from the current multimedia content, and/or how to encode representations of future multimedia content.

[0050] Video files, such as segments of representations of media content, may conform to video data encapsulated according to any of ISO base media file format, Scalable Video Coding (SVC) file format, Advanced Video Coding (AVC) file format, Third Generation Partnership Project (3GPP) file format, and/or Multiview Video Coding (MVC) file format, or other similar video file formats.

[0051] The ISO Base Media File Format is designed to contain timed media information for a presentation in a flexible, extensible format that facilitates interchange, management, editing, and presentation of the media. ISO Base Media File format (ISO/IEC 14496-12:2004) is specified in MPEG-4 Part-12, which defines a general structure for time-based media files. The ISO Base Media File format is used as the basis for other file formats in the family such as AVC file format (ISO/IEC 14496-15) defined support for H.264/MPEG-4 AVC video compression, 3GPP file format, SVC file format, and MVC file format. 3GPP file format and MVC file format are extensions of the AVC file format. ISO base media file format contains the timing, structure, and media information for timed sequences of media data, such as audio-visual presentations. The file structure may be object-oriented. A file can be decomposed into basic objects very simply and the structure of the objects is implied from their type.

[0052] Files conforming to the ISO base media file format (and extensions thereof) may be formed as a series of objects, called "boxes." Data in the ISO base media file format may be contained in boxes, such that no other data needs to be contained within the file and there need not be data outside of boxes within the file. This includes any initial signature required by the specific file format. A "box" may be an object-oriented

building block defined by a unique type identifier and length. Typically, a presentation is contained in one file, and the media presentation is self-contained. The movie container (movie box) may contain the metadata of the media and the video and audio frames may be contained in the media data container and could be in other files.

[0053] A representation (motion sequence) may be contained in several files, sometimes referred to as segments. Timing and framing (position and size) information is generally in the ISO base media file and the ancillary files may essentially use any format. This presentation may be 'local' to the system containing the presentation, or may be provided via a network or other stream delivery mechanism.

[0054] An optional metadata track can be used to tag each track with the "interesting characteristic" that it has, for which its value may differ from other members of the group (e.g., its bit rate, screen size, or language). Some samples within a track may have special characteristics or may be individually identified. One example of the characteristic is the synchronization point (often a video I-frame). These points may be identified by a special table in each track. More generally, the nature of dependencies between track samples can also be documented using metadata. The metadata can be structured as a sequence of file format samples, just like a video track. Such a track may be referred to as a metadata track. Each metadata sample may be structured as a metadata statement. There are various kinds of statements, corresponding to the various questions that might be asked about the corresponding file-format sample or its constituent samples.

[0055] When media is delivered over a streaming protocol, the media may need to be transformed from the way it is represented in the file. One example of this is when media is transmitted over the Real-time Transport Protocol (RTP). In the file, for example, each frame of video is stored contiguously as a file-format sample. In RTP, packetization rules specific to the codec used must be obeyed to place these frames in RTP packets. A streaming server may be configured to calculate such packetization at run-time. However, there is support for the assistance of the streaming servers.

[0056] The techniques of this disclosure may be applicable to network streaming protocols, such as HTTP streaming, e.g., in accordance with dynamic adaptive streaming over HTTP (DASH). In HTTP streaming, frequently used operations include GET and partial GET. The GET operation retrieves a whole file associated a given uniform resource locator (URL) or other identifier, e.g., URI. The partial GET operation receives a byte range as an input parameter and retrieves a continuous number

of bytes of a file corresponding to the received byte range. Thus, movie fragments may be provided for HTTP streaming, because a partial GET operation can get one or more individual movie fragments. Note that, in a movie fragment, there can be several track fragments of different tracks. In HTTP streaming, a media representation may be a structured collection of data that is accessible to the client. The client may request and download media data information to present a streaming service to a user.

[0057] In the example of streaming 3GPP data using HTTP streaming, there may be multiple representations for video and/or audio data of multimedia content. The manifest of such representations may be defined in a Media Presentation Description (MPD) data structure. A media representation may correspond to a structured collection of data that is accessible to an HTTP streaming client device. The HTTP streaming client device may request and download media data information to present a streaming service to a user of the client device. A media representation may be described in the MPD data structure, which may include updates of the MPD.

[0058] Multimedia content may contain a sequence of one or more periods. Periods may be defined by a *Period* element in the MPD. Each period may have an attribute *start* in the MPD. The MPD may include a *start* attribute and an *availableStartTime* attribute for each period. For live services, the sum of the *start* attribute of the period and the MPD attribute *availableStartTime* may specify the availability time of the period in UTC format, in particular the first Media Segment of each representation in the corresponding period. For on-demand services, the *start* attribute of the first period may be 0. For any other period, the *start* attribute may specify a time offset between the start time of the corresponding Period relative to the start time of the first Period. Each period may extend until the start of the next Period, or until the end of the media presentation in the case of the last period. Period start times may be precise. They may reflect the actual timing resulting from playing the media of all prior periods.

[0059] Each period may contain one or more representations for the same media content. A representation may be one of a number of alternative encoded versions of audio or video data. The representations may differ by various characteristics, such as encoding types, e.g., by bitrate, resolution, and/or codec for video data and bitrate, language, and/or codec for audio data. The term representation may be used to refer to a section of encoded audio or video data corresponding to a particular period of the multimedia content and encoded in a particular way.

[0060] Representations of a particular period may be assigned to a group, which may be indicated by a *group* attribute in the MPD. Representations in the same group are generally considered alternatives to each other. For example, each representation of video data for a particular period may be assigned to the same group, such that any of the representations may be selected for decoding to display video data of the multimedia content for the corresponding period. The media content within one period may be represented by either one representation from group 0, if present, or the combination of at most one representation from each non-zero group, in some examples. Timing data for each representation of a period may be expressed relative to the start time of the period.

[0061] A representation may include one or more segments. Each representation may include an initialization segment, or each segment of a representation may be self-initializing. When present, the initialization segment may contain initialization information for accessing the representation. In general, the initialization segment does not contain media data. A segment may be uniquely referenced by an identifier, such as a uniform resource locator (URL). The MPD may provide the identifiers for each segment. In some examples, the MPD may also provide byte ranges in the form of a *range* attribute, which may correspond to the data for a segment within a file accessible by the URL or URI.

[0062] Each representation may also include one or more media components, where each media component may correspond to an encoded version of one individual media type, such as audio, video, and/or timed text (e.g., for closed captioning). Media components may be time-continuous across boundaries of consecutive media segments within one representation.

[0063] FIG. 1 is a block diagram illustrating an example system 10 that implements techniques for streaming media data over a network. In this example, system 10 includes content preparation device 20, server device 60, and client device 40. Client device 40 and server device 60 are communicatively coupled by network 74, which may comprise the Internet. In some examples, content preparation device 20 and server device 60 may also be coupled by network 74 or another network, or may be directly communicatively coupled. In some examples, content preparation device 20 and server device 60 may comprise the same device.

[0064] Content preparation device 20, in the example of FIG. 1, comprises audio source 22 and video source 24. Audio source 22 may comprise, for example, a microphone

that produces electrical signals representative of captured audio data to be encoded by audio encoder 26. Alternatively, audio source 22 may comprise a storage medium storing previously recorded audio data, an audio data generator such as a computerized synthesizer, or any other source of audio data. Video source 24 may comprise a video camera that produces video data to be encoded by video encoder 28, a storage medium encoded with previously recorded video data, a video data generation unit such as a computer graphics source, or any other source of video data. Content preparation device 20 is not necessarily communicatively coupled to server device 60 in all examples, but may store multimedia content to a separate medium that is read by server device 60.

[0065] Raw audio and video data may comprise analog or digital data. Analog data may be digitized before being encoded by audio encoder 26 and/or video encoder 28. Audio source 22 may obtain audio data from a speaking participant while the speaking participant is speaking, and video source 24 may simultaneously obtain video data of the speaking participant. In other examples, audio source 22 may comprise a computer-readable storage medium comprising stored audio data, and video source 24 may comprise a computer-readable storage medium comprising stored video data. In this manner, the techniques described in this disclosure may be applied to live, streaming, real-time audio and video data or to archived, pre-recorded audio and video data.

[0066] Audio frames that correspond to video frames are generally audio frames containing audio data that was captured by audio source 22 contemporaneously with video data captured by video source 24 that is contained within the video frames. For example, while a speaking participant generally produces audio data by speaking, audio source 22 captures the audio data, and video source 24 captures video data of the speaking participant at the same time, that is, while audio source 22 is capturing the audio data. Hence, an audio frame may temporally correspond to one or more particular video frames. Accordingly, an audio frame corresponding to a video frame generally corresponds to a situation in which audio data and video data were captured at the same time and for which an audio frame and a video frame comprise, respectively, the audio data and the video data that was captured at the same time.

[0067] In some examples, audio encoder 26 may encode a timestamp in each encoded audio frame that represents a time at which the audio data for the encoded audio frame was recorded, and similarly, video encoder 28 may encode a timestamp in each encoded video frame that represents a time at which the video data for encoded video frame was

recorded. In such examples, an audio frame corresponding to a video frame may comprise an audio frame comprising a timestamp and a video frame comprising the same timestamp. Content preparation device 20 may include an internal clock from which audio encoder 26 and/or video encoder 28 may generate the timestamps, or that audio source 22 and video source 24 may use to associate audio and video data, respectively, with a timestamp.

[0068] In some examples, audio source 22 may send data to audio encoder 26 corresponding to a time at which audio data was recorded, and video source 24 may send data to video encoder 28 corresponding to a time at which video data was recorded. In some examples, audio encoder 26 may encode a sequence identifier in encoded audio data to indicate a relative temporal ordering of encoded audio data but without necessarily indicating an absolute time at which the audio data was recorded, and similarly, video encoder 28 may also use sequence identifiers to indicate a relative temporal ordering of encoded video data. Similarly, in some examples, a sequence identifier may be mapped or otherwise correlated with a timestamp.

[0069] Audio encoder 26 generally produces a stream of encoded audio data, while video encoder 28 produces a stream of encoded video data. Each individual stream of data (whether audio or video) may be referred to as an elementary stream. An elementary stream is a single, digitally coded (possibly compressed) component of a representation. For example, the coded video or audio part of the representation can be an elementary stream. An elementary stream may be converted into a packetized elementary stream (PES) before being encapsulated within a video file. Within the same representation, a stream ID may be used to distinguish the PES-packets belonging to one elementary stream from the other. The basic unit of data of an elementary stream is a packetized elementary stream (PES) packet. Thus, coded video data generally corresponds to elementary video streams. Similarly, audio data corresponds to one or more respective elementary streams.

[0070] As with many video coding standards, H.264/AVC defines the syntax, semantics, and decoding process for error-free bitstreams, any of which conform to a certain profile or level. H.264/AVC does not specify the encoder, but the encoder is tasked with guaranteeing that the generated bitstreams are standard-compliant for a decoder. In the context of video coding standard, a "profile" corresponds to a subset of algorithms, features, or tools and constraints that apply to them. As defined by the H.264 standard, for example, a "profile" is a subset of the entire bitstream syntax that is

specified by the H.264 standard. A "level" corresponds to the limitations of the decoder resource consumption, such as, for example, decoder memory and computation, which are related to the resolution of the pictures, bit rate, and macroblock (MB) processing rate. A profile may be signaled with a profile_idc (profile indicator) value, while a level may be signaled with a level_idc (level indicator) value.

[0071] The H.264 standard, for example, recognizes that, within the bounds imposed by the syntax of a given profile, it is still possible to require a large variation in the performance of encoders and decoders depending upon the values taken by syntax elements in the bitstream such as the specified size of the decoded pictures. The H.264 standard further recognizes that, in many applications, it is neither practical nor economical to implement a decoder capable of dealing with all hypothetical uses of the syntax within a particular profile. Accordingly, the H.264 standard defines a "level" as a specified set of constraints imposed on values of the syntax elements in the bitstream. These constraints may be simple limits on values. Alternatively, these constraints may take the form of constraints on arithmetic combinations of values (e.g., picture width multiplied by picture height multiplied by number of pictures decoded per second). The H.264 standard further provides that individual implementations may support a different level for each supported profile.

[0072] A decoder conforming to a profile ordinarily supports all the features defined in the profile. For example, as a coding feature, B-picture coding is not supported in the baseline profile of H.264/AVC but is supported in other profiles of H.264/AVC. A decoder conforming to a level should be capable of decoding any bitstream that does not require resources beyond the limitations defined in the level. Definitions of profiles and levels may be helpful for interpretability. For example, during video transmission, a pair of profile and level definitions may be negotiated and agreed for a whole transmission session. More specifically, in H.264/AVC, a level may define, for example, limitations on the number of macroblocks that need to be processed, decoded picture buffer (DPB) size, coded picture buffer (CPB) size, vertical motion vector range, maximum number of motion vectors per two consecutive MBs, and whether a B-block can have sub-macroblock partitions less than 8x8 pixels. In this manner, a decoder may determine whether the decoder is capable of properly decoding the bitstream.

[0073] Video compression standards such as ITU-T H.261, H.262, H.263, MPEG-1, MPEG-2, H.264/MPEG-4 part 10, and the upcoming High Efficiency Video Coding (HEVC) standard, make use of motion compensated temporal prediction to reduce

temporal redundancy. The encoder, such as video encoder 28, may use a motion compensated prediction from some previously encoded pictures (also referred to herein as frames) to predict the current coded pictures according to motion vectors. There are three major picture types in typical video coding. They are Intra coded picture ("I-pictures" or "I-frames"), Predicted pictures ("P-pictures" or "P-frames") and Bi-directional predicted pictures ("B-pictures" or "B-frames"). P-pictures may use the reference picture before the current picture in temporal order. In a B-picture, each block of the B-picture may be predicted from one or two reference pictures. These reference pictures could be located before or after the current picture in temporal order.

[0074] Parameter sets generally contain sequence-layer header information in sequence parameter sets (SPS) and the infrequently changing picture-layer header information in picture parameter sets (PPS). With parameter sets, this infrequently changing information need not be repeated for each sequence or picture; hence, coding efficiency may be improved. Furthermore, the use of parameter sets may enable out-of-band transmission of header information, avoiding the need for redundant transmissions to achieve error resilience. In out-of-band transmission, parameter set NAL units are transmitted on a different channel than the other NAL units.

[0075] In the example of FIG. 1, encapsulation unit 30 of content preparation device 20 receives elementary streams comprising coded video data from video encoder 28 and elementary streams comprising coded audio data from audio encoder 26. In some examples, video encoder 28 and audio encoder 26 may each include packetizers for forming PES packets from encoded data. In other examples, video encoder 28 and audio encoder 26 may each interface with respective packetizers for forming PES packets from encoded data. In still other examples, encapsulation unit 30 may include packetizers for forming PES packets from encoded audio and video data.

[0076] Video encoder 28 may encode video data of multimedia content in a variety of ways, to produce different representations of the multimedia content at various bitrates and with various characteristics, such as pixel resolutions, frame rates, conformance to various coding standards, conformance to various profiles and/or levels of profiles for various coding standards, representations having one or multiple views (e.g., for two-dimensional or three-dimensional playback), or other such characteristics. A representation, as used in this disclosure, may comprise a combination of audio data and video data, e.g., one or more audio elementary stream and one or more video elementary streams. Each PES packet may include a stream_id that identifies the elementary

stream to which the PES packet belongs. Encapsulation unit 30 is responsible for assembling elementary streams into video files of various representations.

[0077] Encapsulation unit 30 receives PES packets for elementary streams of a representation from audio encoder 26 and video encoder 28 and forms corresponding network abstraction layer (NAL) units from the PES packets. In the example of H.264/AVC (Advanced Video Coding), coded video segments are organized into NAL units, which provide a "network-friendly" video representation addressing applications such as video telephony, storage, broadcast, or streaming. NAL units can be categorized to Video Coding Layer (VCL) NAL units and non-VCL NAL units. VCL units may contain the core compression engine and may include block, macroblock, and/or slice level data. Other NAL units may be non-VCL NAL units. In some examples, a coded picture in one time instance, normally presented as a primary coded picture, may be contained in an access unit, which may include one or more NAL units.

[0078] Non-VCL NAL units may include parameter set NAL units and SEI NAL units, among others. Parameter sets may contain sequence-level header information (in sequence parameter sets (SPS)) and the infrequently changing picture-level header information (in picture parameter sets (PPS)). With parameter sets (e.g., PPS and SPS), infrequently changing information need not to be repeated for each sequence or picture, hence coding efficiency may be improved. Furthermore, the use of parameter sets may enable out-of-band transmission of the important header information, avoiding the need for redundant transmissions for error resilience. In out-of-band transmission examples, parameter set NAL units may be transmitted on a different channel than other NAL units, such as SEI NAL units.

[0079] Supplemental Enhancement Information (SEI) may contain information that is not necessary for decoding the coded pictures samples from VCL NAL units, but may assist in processes related to decoding, display, error resilience, and other purposes. SEI messages may be contained in non-VCL NAL units. SEI messages are the normative part of some standard specifications, and thus are not always mandatory for standard compliant decoder implementation. SEI messages may be sequence level SEI messages or picture level SEI messages. Some sequence level information may be contained in SEI messages, such as scalability information SEI messages in the example of SVC and view scalability information SEI messages in MVC. These example SEI messages may convey information on, e.g., extraction of operation points and characteristics of the operation points. In addition, encapsulation unit 30 may form a manifest file, such as a

media presentation descriptor (MPD) that describes characteristics of the representations. Encapsulation unit 30 may format the MPD according to extensible markup language (XML).

[0080] Encapsulation unit 30 may provide data for one or more representations of multimedia content, along with the manifest file (e.g., the MPD) to output interface 32. Output interface 32 may comprise a network interface or an interface for writing to a storage medium, such as a universal serial bus (USB) interface, a CD or DVD writer or burner, an interface to magnetic or flash storage media, or other interfaces for storing or transmitting media data. Encapsulation unit 30 may provide data of each of the representations of multimedia content to output interface 32, which may send the data to server device 60 via network transmission or storage media. In the example of FIG. 1, server device 60 includes storage medium 62 that stores various multimedia contents 64, each including a respective manifest file 66 and one or more representations 68A–68N (representations 68). In accordance with the techniques of this disclosure, portions of manifest file 66 may be stored in separate locations, e.g., locations of storage medium 62 or another storage medium, potentially of another device of network 74 such as a proxy device.

[0081] In some examples, representations 68 may be separated into representation groups. That is, various subsets of representations 68 may include respective common sets of characteristics, such as codec, profile and level, resolution, number of views, file format for segments, text type information that may identify a language or other characteristics of text to be displayed with the representation and/or audio data to be decoded and presented, e.g., by speakers, camera angle information that may describe a camera angle or real-world camera perspective of a scene for representations in the representation group, rating information that describes content suitability for particular audiences, or the like.

[0082] Manifest file 66 may include data indicative of the subsets of representations 68 corresponding to particular representation groups, as well as common characteristics for the representation groups. Manifest file 66 may also include data representative of individual characteristics, such as bitrates, for individual representations of representation groups. In this manner, a representation group may provide for simplified network bandwidth adaptation. Representations in a representation group may be indicated using child elements of a representation group element of manifest file 66.

[0083] Manifest file 66 may also (that is, additionally or alternatively) signal trick mode information for one or more of representations 68. In some examples, one or more of representations 68 may include a respective temporal sub-sequence for trick mode support. A trick mode generally corresponds to a playback mode for a representation in which data of the representation is not played from start to finish, but instead, may begin at a specified temporal location (e.g., to allow for seeking to a particular temporal location), or skip one or more frames in either the forward or reverse temporal direction (e.g., fast forward or rewind).

[0084] To provide for trick modes, multimedia content 64 may include information representative of locations of data for temporal sub-sequences of corresponding representations 68. In some examples, manifest file 66 may include information representative of locations of data for the temporal sub-sequences. In other examples, representations 68 themselves may include the information representative of locations of data for the temporal sub-sequences. In still other examples, both representations 68 and manifest file 66 may include information representative of locations of data for the temporal sub-sequences.

[0085] In some examples, content preparation device 20 may prepare media content as the media content is being recorded, e.g., for live services. Encapsulation unit 30 may in some cases need to periodically update a manifest file for the media content. Encapsulation unit 30 may even update the manifest file within a particular period of the media content. In accordance with the techniques of this disclosure, encapsulation unit 30 may form segments of a representation that include data indicating that the manifest file is to be updated. Encapsulation unit 30 may provide the updates in the segments themselves or in a separate location from which client devices, such as client device 40 may retrieve the updates to the manifest file. In this manner, when manifest file 66 needs to be updated within a particular period of multimedia content 64, encapsulation unit 30 may form a segment of one or more of representations 68 indicating that manifest file 66 is to be updated.

[0086] In some examples, manifest file 66 may include data for inserting data of a remote period into multimedia content 64 during playback. For example, rather than encoding advertisements within multimedia content 64, content preparation device 20 may prepare one or more separate advertisement media contents to be incorporated into multimedia content 64 during playback. Client device 40 may in some examples provide user specific information such that advertisements can be targeted to a user of

client device 40, such that a user of client device 40 receives advertisements that are most preferable and informative to the user. In response to a set of user information, server device 60 may provide a targeted advertisement portion of the manifest file to client device 40, which may cause client device 40 to retrieve data of targeted advertisement multimedia content. In this manner, two or more viewers of the same multimedia content 64 may receive different targeted advertisements, such that the advertisements are most relevant and useful for the users.

[0087] Server device 60 includes request processing unit 70 and network interface 72. In some examples, server device 60 may include a plurality of network interfaces. Furthermore, any or all of the features of server device 60 may be implemented on other devices of a content delivery network, such as routers, bridges, proxy devices, switches, or other devices. In some examples, intermediate devices of a content delivery network may cache data of multimedia content 64, and include components that conform substantially to those of server device 60. In general, network interface 72 is configured to send and receive data via network 74.

[0088] Request processing unit 70 is configured to receive network requests from client devices, such as client device 40, for data of storage medium 72. For example, request processing unit 70 may implement hypertext transfer protocol (HTTP) version 1.1, as described in RFC 2616, "Hypertext Transfer Protocol – HTTP/1.1," by R. Fielding et al, Network Working Group, IETF, June 1999. That is, request processing unit 70 may be configured to receive HTTP GET or partial GET requests and provide data of multimedia content 64 in response to the requests. The requests may specify a segment of one of representations 68, e.g., using a URL of the segment. In some examples, the requests may also specify one or more byte ranges of the segment, thus comprising partial GET requests. Request processing unit 70 may further be configured to service HTTP HEAD requests to provide header data of a segment of one of representations 68. In any case, request processing unit 70 may be configured to process the requests to provide requested data to a requesting device, such as client device 40.

[0089] As illustrated in the example of FIG. 1, multimedia content 64 includes manifest file 66, which may correspond to a media presentation description (MPD). Manifest file 66 may contain descriptions of different alternative representations 68 (e.g., video services with different qualities) and the description may include, e.g., codec information, a profile value, a level value, a bitrate, and other descriptive characteristics

of representations 68. Client device 40 may retrieve the MPD of a media presentation to determine how to access segments of representations 68.

[0090] In particular, web application 52 may retrieve configuration data (not shown) of client device 40 to determine decoding capabilities of video decoder 48 and rendering capabilities of video output 44. The configuration data may also include any or all of a language preference selected by a user of client device 40, one or more camera perspectives corresponding to depth preferences set by the user of client device 40, and/or a rating preference selected by the user of client device 40. Web application 52 may comprise, for example, a web browser or a media client configured to submit HTTP GET and partial GET requests. Web application 52 may correspond to software instructions executed by one or more processors or processing units (not shown) of client device 40. In some examples, all or portions of the functionality described with respect to web application 52 may be implemented in hardware, or a combination of hardware, software, and/or firmware, where requisite hardware may be provided to execute instructions for software or firmware.

[0091] Web application 52 may compare the decoding and rendering capabilities of client device 40 to characteristics of representations 68 indicated by information of manifest file 66. Web application 52 may initially retrieve at least a portion of manifest file 66 to determine characteristics of representations 68. For example, web application 52 may request a portion of manifest file 66 that describes characteristics of one or more representation groups, in accordance with the techniques of this disclosure. Web application 52 may select a subset of representations 68 (e.g., a representation group) having characteristics that can be satisfied by the coding and rendering capabilities of client device 40. Web application 52 may then determine bitrates for representations in the representation group, determine a currently available amount of network bandwidth, and retrieve segments from one of the representations having a bitrate that can be satisfied by the network bandwidth.

[0092] In general, higher bitrate representations may yield higher quality video playback, while lower bitrate representations may provide sufficient quality video playback when available network bandwidth decreases. Accordingly, when available network bandwidth is relatively high, web application 52 may retrieve data from relatively high bitrate representations, whereas when available network bandwidth is low, web application 52 may retrieve data from relatively low bitrate representations. In

this manner, client device 40 may stream multimedia data over network 74 while also adapting to changing network bandwidth availability of network 74.

[0093] As noted above, in some examples, client device 40 may provide user information to, e.g., server device 60 or other devices of a content delivery network. Web application 52, for example, may collect a user identifier, user identifier, user preferences, and/or user demographic information, and provide such user information to server device 60. Web application 52 may then receive a manifest file associated with targeted advertisement media content, to use to insert data from the targeted advertisement media content into media data of requested media content during playback.

[0094] At times, a user of client device 40 may interact with web browser 52 using user interfaces of client device 40, such as a keyboard, mouse, stylus, touchscreen interface, buttons, or other interfaces, to request that a selected one of representations 68 be played in a trick mode. For example, the user may select a particular temporal location from which to begin playback, or to skip or seek to a particular temporal location. As another example, the user may elect to fast forward or rewind the representation.

[0095] In response to such requests from a user, web application 52 may determine whether one of representations 68 includes a temporal sub-sequence, to perform the requested trick mode. As an example, a user may elect to play video data in a fast forward mode. Rather than retrieving all data of segments of a representation, web application 52 may determine locations of data of the representation corresponding to a temporal sub-sequence of the representation. The data of the temporal sub-sequence may correspond, for example, to a set of instantaneous decoder refresh (IDR) pictures of the representation.

[0096] There may be an approximate temporal duration between IDR pictures of a representation, e.g., 2 seconds, 10 seconds, or other approximate temporal durations. Moreover, IDR pictures may be encoded in an intra-prediction mode, and thus, web application 52 need not retrieve data other than the IDR pictures. Web application 52 may cause the IDR pictures to be displayed at the same frame rate at which video data of the representation would otherwise be displayed. However, because many frames of data between IDR pictures may be skipped, the resulting video data may be played back at an increased frame rate, thus achieving the desired trick mode.

[0097] Web application 52 may determine locations of the data for the temporal sub-sequence using various techniques. In some examples, web application 52 may analyze

data of manifest file 66 to determine locations of the IDR pictures. The locations of the IDR pictures may be indicated using byte ranges within segments of a particular representation. In other examples, a particular box of segments of representations, such as a sub-fragment index box (also referred to as a sub-segment index box), may provide indications of the locations of the data for the temporal sub-sequence. For example, the sub-fragment index box may include data representative of byte ranges for IDR pictures within a corresponding segment. In still other examples, both manifest file 66 and representations 68 may include information used by web application 52 to retrieve data for a temporal sub-sequence. In any case, web application 52 may determine the byte ranges of the IDR pictures in the segments to construct partial GET requests for the IDR pictures, to avoid retrieving data that would not be used for decoding or display.

[0098] In some examples, encapsulation unit 30 may form segments such that IDR pictures are contiguous within the segments. That is, encapsulation unit 30 may ensure that bytes of segments corresponding to IDR pictures are contiguous, without intervening bytes for other types of pictures. In this manner, web application 52 need only specify a single byte range of segments of a representation to retrieve data for a temporal sub-sequence of the representation. In some examples, open decoder refresh (ODR) pictures may also be used for performing trick modes.

[0099] In some examples, web application 52 may determine that a portion of a received segment indicates that a manifest file is to be updated. Web application 52 may be configured to analyze a particular portion of each segment, such as a header portion or other initial portion of the segment, to determine whether the segment indicates that the manifest file is to be updated. When a segment indicates that the manifest file is to be updated, web application 52 may update a locally stored copy of the manifest file, either using data of the segment or by retrieving data to update the manifest file from a remote location, e.g., from server 60. After updating the manifest file, web application 52 may submit future requests for data of representations 68 based on data of the updated manifest file.

[0100] As an example, content preparation device 20 may encode live media data, such as a live sporting event, political event, or other newsworthy event that is typically broadcast live or nearly live, rather than pre-recorded. In such cases, segments corresponding to media data up to a particular time may be assigned identifiers, such as URLs, included in an initial manifest file. However, after a period of time has elapsed, segments following the particular time may be encoded and assigned identifiers such as

URLs. Encapsulation unit 30 of content preparation device 20 may provide the URLs for the segments following the particular time to an updated manifest file. Accordingly, to determine how to retrieve the segments following the particular time, client device 40 may receive information indicative of the updated manifest file, in order to construct requests to retrieve the segments following the particular time.

[0101] In some examples, a segment may indicate whether it is the last segment of a representation. When a segment is the last segment of a representation, a new manifest file may need to be retrieved to determine representations of a subsequent period of corresponding multimedia content. Accordingly, when web application 52 determines that a segment is the last segment of a representation in a period of the multimedia content, web application 52 may retrieve an updated manifest file for the multimedia content, e.g., an updated version of manifest file 66 of multimedia content 64.

[0102] In some examples, client device 40 may maintain a data structure indicative of particular representations 68 from which client device 40 requested data for multimedia content 64. Client device 40 may also maintain indications of exactly what was played out, and at what time. That is, the data structure may provide information representative of start and end times in both real (or "wall clock") time and presentation time. The data structure may further provide information representative of an initial startup time and the start of playback. After finishing playback of multimedia content 64, client device 40 may send the data structure to server device 60 and/or content preparation device 20. Server device 60 and/or content preparation device 20 may use information received from client device 40 to determine more optimal ways of improving quality of experience, such as to reduce pauses in playback.

[0103] Network interface 54 may receive and provide data of segments of a selected representation to web application 52, which may in turn provide the segments to decapsulation unit 50. Decapsulation unit 50 may decapsulate elements of a video file into constituent PES streams, depacketize the PES streams to retrieve encoded data, and send the encoded data to either audio decoder 46 or video decoder 48, depending on whether the encoded data is part of an audio or video stream, e.g., as indicated by PES packet headers of the stream. Audio decoder 46 decodes encoded audio data and sends the decoded audio data to audio output 42, while video decoder 48 decodes encoded video data and sends the decoded video data, which may include a plurality of views of a stream, to video output 44.

[0104] Video encoder 28, video decoder 48, audio encoder 26, audio decoder 46, encapsulation unit 30, web application 52, and decapsulation unit 50 each may be implemented as any of a variety of suitable processing circuitry, as applicable, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic circuitry, software, hardware, firmware or any combinations thereof. Each of video encoder 28 and video decoder 48 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined video encoder/decoder (CODEC). Likewise, each of audio encoder 26 and audio decoder 46 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined CODEC. An apparatus including video encoder 28, video decoder 48, audio encoder audio encoder 26, audio decoder 46, encapsulation unit 30, web application 52, and/or decapsulation unit 50 may comprise an integrated circuit, a microprocessor, and/or a wireless communication device, such as a cellular telephone.

[0105] FIG. 2 is a conceptual diagram illustrating elements of an example multimedia content 100. Multimedia content 100 may correspond to multimedia content 64 (FIG. 1), or another multimedia content stored in memory 62. In the example of FIG. 2, multimedia content 100 includes media presentation description (MPD) 102 and a plurality of representations 110–120. Representation 110 includes optional header data 112 and segments 114A–114N (segments 114), while representation 120 includes optional header data 122 and segments 124A–124N (segments 124). The letter N is used to designate the last movie fragment in each of representations 110, 120 as a matter of convenience. In some examples, there may be different numbers of movie fragments between representations 110, 120.

[0106] MPD 102 may comprise a data structure separate from representations 110–120. MPD 102 may correspond to manifest file 66 of FIG. 1. Likewise, representations 110–120 may correspond to representations 68 of FIG. 1. In general, MPD 102 may include data that generally describes characteristics of representations 110–120, such as coding and rendering characteristics, representation groups, a profile to which MPD 102 corresponds, text type information, camera angle information, rating information, trick mode information (e.g., information indicative of representations that include temporal sub-sequences), and/or information for retrieving remote periods (e.g., for targeted advertisement insertion into media content during playback). Remote periods may also be referred to as external periods. FIGS. 4–7, discussed in greater detail below,

illustrate various examples of multimedia content with various elements included in either or both of an MPD and/or representations (such as within segments of representations or header data of representations). Any or all of the MPDs of FIGS. 4–7 may correspond substantially to MPD 102 of FIG. 2.

[0107] Header data 112, when present, may describe characteristics of segments 114, e.g., temporal locations of random access points, which of segments 114 includes random access points, byte offsets to random access points within segments 114, uniform resource locators (URLs) of segments 114, or other aspects of segments 114. Header data 122, when present, may describe similar characteristics for segments 124. Additionally or alternatively, such characteristics may be fully included within MPD 102.

[0108] Segments 114 include one or more coded video samples, each of which may include frames or slices of video data. Each of the coded video samples of segments 114 may have similar characteristics, e.g., height, width, and bandwidth requirements. Such characteristics may be described by data of MPD 102, though such data is not illustrated in the example of FIG. 2. MPD 102 may include characteristics as described by the 3GPP Specification, with the addition of any or all of the signaled information described in this disclosure.

[0109] Each of segments 114, 124 may be associated with a unique uniform resource identifier (URI), e.g., a uniform resource locator (URL). Thus, each of segments 114, 124 may be independently retrievable using a streaming network protocol, such as DASH. In this manner, a destination device, such as client device 40, may use an HTTP Get request to retrieve segments 114 or 124. In some examples, client device 40 may use HTTP partial Get requests to retrieve specific byte ranges of segments 114 or 124.

[0110] As noted above, MPD 102 may conform to a particular MPD profile. MPD 102 may include information indicative of a Multipurpose Internet Mail Extension (MIME) type for MPD 102 and/or multimedia content 100. MIME types generally do not indicate what codec is needed to present multimedia content, however. In general, it is assumed that if a device can retrieve an MPD for multimedia content, such as MPD 102, that the device can playback data of the multimedia content corresponding to the MPD. However, this assumption may not always be safe. Therefore, in some examples, MPD 102 may include information indicative of a profile to which MPD 102 corresponds.

[0111] There may be a relatively small number of profiles to which MPDs may correspond. Profiles may be supported by levels to address capabilities, similar to the

manner in which H.264/AVC includes profiles and levels for video coding. MPD profiles may be onion-shelled, in that a higher profile may include all features of all lower profiles. There may be a registration process with a registration authority to register various profiles. In some examples, a client device, such as client device 40, may be configured to retrieve information indicative of the profile for the MPD, such as MPD 102, before retrieving other data of the MPD, such as characteristics of representations 110–120 signaled by MPD 102. In this manner, the profile for MPD 102 may be signaled before access to MPD 102 is provided.

[0112] A profile identifier may be provided in plain text (e.g., as a plain name), or a reversed domain name. Plain names may be reserved by a registration authority, such as 3GPP or another registration authority. A profile may be considered a claim and a permission, in that the profile may claim that a corresponding multimedia content conforms to the profile and gives permission to a reader (e.g., a client device) that implements that profile to read the MPD, interpret what it recognizes, and ignore material that it does not understand.

[0113] Profiles may describe characteristics such as, for example, features of MPD 102, usage of the network, media format(s), codec(s) used, protection formats, and/or quantitative measures such as bitrates, screen sizes, and the like. In this manner, the profile of MPD 102 may provide information indicative of what codecs need to be supported in order to retrieve data of MPD 102 and/or multimedia content 100. Profiles may also be described as "conformance points." Profiles with which an MPD complies may be indicated in a "Profiles" attribute of the MPD. Thus, a client device may be configured to retrieve a portion of MPD 102 including information relating to the "Profiles" attribute before retrieving additional data of MPD 102. Alternatively, profiles may be indicated as a parameter in the MIME type of the MPD. For example, profiles "X, Y, and Z" may be signaled in the following fashion:

video/vnd.mpeg.mpd;profiles="X,Y,Z."

[0114] In some examples, MPD 102 may refer to data of external periods (also referred to as remote periods). A period generally corresponds to a particular temporal section of multimedia content. Each period may include one or more representations, such as representations 110–120. An external period, however, may be inserted within or between periods of multimedia content 100. The external period may include multimedia data in addition to multimedia data of the multimedia content. For example, external periods may include advertisement data.

[0115] Periods may be defined by their duration, that is, the start time of the Period may depend on the duration of the previous Period. A client device may map external periods to an MPD structure. For live services, concatenation of MPDs may be achieved by a dynamic creation of the MPD on the server, such as server device 60, with appropriate update procedures. Other web technologies may also be used. URLs for externally defined periods may be processed in real-time to generate a new period containing advertisements targeted to a user of client device 40. Client device 40 may supply additional information with the request that may be used for advertisement targeting, for example, a user identifier, user preferences, user demographic information, or other information.

[0116] Table 1 below illustrates an example set of information that may be provided in MPD 102 to describe one or more Periods of multimedia content, and to indicate the presence of external periods:

TABLE 1---MPD Period Information

Period	E	1...N	M	Provides the information of a Period
PeriodAttributes	List		M	already existing period attributes
periodDuration	A		O	Provides the period duration, may be used as alternative to *start* attribute of next Period.
representationGroupListURI	A		O	URI that points to a document that contains a list of Representations.
RepresentationGroups	E	0..N		This element contains a description of a Representation Group
periodListURI	A		M	URI that points to a document that contains one or several Period elements.

[0117] In this manner, the Period element of MPD 102 may refer to external (or remote) periods, e.g., using periodListURI. For On-Demand content, indications of period durations may be more useful for client devices, such as client device 40, than start times to support external periods. An MPD may include a sequence of Periods, where Periods can be internal or external. Using such remote Periods, along with user-specific information, may allow for targeted user advertising. Server device 60 and/or content preparation device 20 may be configured to dynamically generate separate MPDs for each user, or for each client device. Client device 40, or another device, may concatenate the playout of a targeted advertisement and a live service, e.g., using the dynamically-created MPD.

[0118] In this manner, the techniques of this disclosure may support situations in which a service provider offers On-Demand content through 3GPP AHS. The content may

include several scenes, and between each scene, an advertisement may be added. The advertisement may differ for each user. That is, targeted advertising may be added. In addition, each advertisement may be of a different duration. Likewise, a service provider may offer a particular live service (e.g., a free service). When accessing the live service, the service provider may add an advertisement that may or may not be targeted to a user. The duration of the advertisement may differ, depending on access time, the access location, the user, and the like. Server device 60 may be configured to only provide the URL of the live service after the advertisement has been completed, to ensure that the advertisement is seen.

[0119] FIG. 3 is a block diagram illustrating elements of an example video file 150, which may correspond to a segment of a representation, such as one of segments 114, 124 of FIG. 2. Each of segments 114, 124 may include data that conforms substantially to the arrangement of data illustrated in the example of FIG. 3. Similarly, segments of FIGS. 4–7 discussed below may also conform substantially to the structure of video file 150. As described above, video files in accordance with the ISO base media file format and extensions thereof store data in a series of objects, referred to as "boxes." In the example of FIG. 3, video file 150 includes file type (FTYP) box 152, movie (MOOV) box 154, movie fragment (MOOF) boxes 162, and movie fragment random access (MFRA) box 164.

[0120] File type (FTYP) box 152 generally describes a file type for video file 150. File type box 152 may include data that identifies a specification that describes a best use for video file 150. File type box 152 may be placed before MOOV box 154, movie fragment boxes 162, and MFRA box 164.

[0121] In some examples, a segment, such as video file 150, may include an MPD update box (not shown) before FTYP box 152. The MPD update box may include information indicating that an MPD corresponding to a representation including video file 150 is to be updated, along with information for updating the MPD. For example, the MPD update box may provide a URI or URL for a resource to be used to update the MPD. As another example, the MPD update box may include data for updating the MPD. In some examples, the MPD update box may immediately follow a segment type (STYP) box (not shown) of video file 150, where the STYP box may define a segment type for video file 150. FIG 7, discussed in greater detail below, provides additional information with respect to the MPD update box.

[0122] MOOV box 154, in the example of FIG. 3, includes movie header (MVHD) box 156, track (TRAK) box 158, and one or more movie extends (MVEX) boxes 160. In general, MVHD box 156 may describe general characteristics of video file 150. For example, MVHD box 156 may include data that describes when video file 150 was originally created, when video file 150 was last modified, a timescale for video file 150, a duration of playback for video file 150, or other data that generally describes video file 150.

[0123] TRAK box 158 may include data for a track of video file 150. TRAK box 158 may include a track header (TKHD) box that describes characteristics of the track corresponding to TRAK box 158. In some examples, TRAK box 158 may include coded video pictures, while in other examples, the coded video pictures of the track may be included in movie fragments 162, which may be referenced by data of TRAK box 158.

[0124] In some examples, video file 150 may include more than one track. Accordingly, MOOV box 154 may include a number of TRAK boxes equal to the number of tracks in video file 150. TRAK box 158 may describe characteristics of a corresponding track of video file 150. For example, TRAK box 158 may describe temporal and/or spatial information for the corresponding track. A TRAK box similar to TRAK box 158 of MOOV box 154 may describe characteristics of a parameter set track, when encapsulation unit 30 (FIG. 1) includes a parameter set track in a video file, such as video file 150. Encapsulation unit 30 may signal the presence of sequence level SEI messages in the parameter set track within the TRAK box describing the parameter set track.

[0125] MVEX boxes 160 may describe characteristics of corresponding movie fragments 162, e.g., to signal that video file 150 includes movie fragments 162, in addition to video data included within MOOV box 154, if any. In the context of streaming video data, coded video pictures may be included in movie fragments 162 rather than in MOOV box 154. Accordingly, all coded video samples may be included in movie fragments 162, rather than in MOOV box 154.

[0126] MOOV box 154 may include a number of MVEX boxes 160 equal to the number of movie fragments 162 in video file 150. Each of MVEX boxes 160 may describe characteristics of a corresponding one of movie fragments 162. For example, each MVEX box may include a movie extends header box (MEHD) box that describes a temporal duration for the corresponding one of movie fragments 162.

[0127] As noted above, encapsulation unit 30 may store a sequence data set in a video sample that does not include actual coded video data. A video sample may generally correspond to an access unit, which is a representation of a coded picture at a specific time instance. In the context of AVC, the coded picture include one or more VCL NAL units which contains the information to construct all the pixels of the access unit and other associated non-VCL NAL units, such as SEI messages. Accordingly, encapsulation unit 30 may include a sequence data set, which may include sequence level SEI messages, in one of movie fragments 162. Encapsulation unit 30 may further signal the presence of a sequence data set and/or sequence level SEI messages as being present in one of movie fragments 162 within the one of MVEX boxes 160 corresponding to the one of movie fragments 162.

[0128] Movie fragments 162 may include one or more coded video pictures. In some examples, movie fragments 162 may include one or more groups of pictures (GOPs), each of which may include a number of coded video pictures, e.g., frames or pictures. In addition, as described above, movie fragments 162 may include sequence data sets in some examples. Each of movie fragments 162 may include a movie fragment header box (MFHD, not shown in FIG. 3). The MFHD box may describe characteristics of the corresponding movie fragment, such as a sequence number for the movie fragment. Movie fragments 162 may be included in order of sequence number in video file 150.

[0129] MFRA box 164 may describe random access points within movie fragments 162 of video file 150. This may assist with performing trick modes, such as performing seeks to particular temporal locations within video file 150. MFRA box 164 is generally optional and need not be included in video files, in some examples. Likewise, a client device, such as client device 40, does not necessarily need to reference MFRA box 164 to correctly decode and display video data of video file 150. MFRA box 164 may include a number of track fragment random access (TFRA) boxes (not shown) equal to the number of tracks of video file 150, or in some examples, equal to the number of media tracks (e.g., non-hint tracks) of video file 150.

[0130] In some examples, movie fragments 162 may include one or more IDR and/or ODR pictures. Likewise, MFRA box 164 may provide indications of locations within video file 150 of the IDR and ODR pictures. Accordingly, a temporal sub-sequence of video file 150 may be formed from IDR and ODR pictures of video file 150. The temporal sub-sequence may also include other pictures, such as P-frames and/or B-frames that depend from IDR and/or ODR pictures. Frames and/or slices of the

temporal sub-sequence may be arranged within the segments such that frames/slices of the temporal sub-sequence that depend on other frames/slices of the sub-sequence can be properly decoded. For example, in the hierarchical arrangement of data, data used for prediction for other data may also be included in the temporal sub-sequence. Moreover, the data may be arranged in a continuous sub-sequence, such that a single byte range may be specified in a partial GET request to retrieve all data of a particular segment used for the temporal sub-sequence. A client device, such as client device 40, may extract a temporal sub-sequence of video file 150 by determining byte-ranges of movie fragments 162 (or portions of movie fragments 162) corresponding to IDR and/or ODR pictures. As discussed in greater detail below, video files such as video file 150 may include a sub-fragment index box and/or a sub-track fragment box, either or both of which may include data for extracting a temporal sub-sequence of video file 150.

[0131] FIG. 4 is a conceptual diagram illustrating an example multimedia content 200 including MPD 202 and representation groups 210–220. Multimedia content 200 may correspond to multimedia content 64 (FIG. 1), or another multimedia content stored in memory 62. In this example, representations of multimedia content 200 are arranged by representation group. That is, representations with a common set of characteristics may be formed into a representation group, which provides for simplified network bandwidth adaptation.

[0132] In this example, MPD 202 includes common representation characteristics 204A that includes information describing common characteristics of representation group 210 and common representation characteristics 204B describing common characteristics of representation group 220. The common characteristics may include coding and/or rendering characteristics of the representations, such as a codec, profile and level of the codec to which the representations in the representation conform, pixel resolution, frame rate, or other characteristics of representations.

[0133] In accordance with the techniques of this disclosure, the characteristics may include a text type value, a camera angle value, and/or a rating value in addition to the characteristics discussed above. The text type value may describe characteristics of text to be displayed with the video data (e.g., closed captioning text). The text type value may describe, for example, a language of the text, a location on the screen at which to display the text, font and/or size of the text, or other characteristics of the text.

[0134] The camera angle value may describe a real-world horizontal camera position for a camera used (either physically or conceptually) to generate coded video data of

corresponding representations. Using the camera angles, a client device may select data from two or more representations to be displayed substantially simultaneously, e.g., to produce a three-dimensional video playback effect. The horizontal real-world camera locations may enable the client device to select representations to increase or decrease the relative amount of depth in the three-dimensional playback of the video data.

[0135] The rating may describe content suitability for particular audiences. For example, in the United States, the Motion Picture Association of America defines ratings including G, PG, PG-13, R, and NC-17. As another example, in the United Kingdom, the British Board of Film Classification defines ratings including U, PG, 12A, 12, 15, 18, and R18. As yet another example, in the Republic of China (Taiwan), categories of motion pictures include a general audiences category, a protected category, a parental guidance category, and a restricted category.

[0136] By providing common characteristics 204 of respective representation groups, e.g., representation groups 210–220, a client device (e.g., client device 40) may select an appropriate one of representation groups 210–220, based at least in part on the corresponding common representation characteristics 204. In the example of FIG. 4, MPD 202 also includes individual representation characteristics 206A, 206B, 208A, and 208B, corresponding respectively to representations 212A, 212B, 222A, 222B. Individual representation characteristics 206A, 206B, 208A, and 208B may include information representative of characteristics of representations 212A, 212B, 222A, 222B not indicated by common representation characteristics 204. For example, individual representation characteristics 206A, 206B, 208A, and 208B may include information representative of bitrates for respective ones of representations 212A, 212B, 222A, 222B.

[0137] Representations of a representation group may be considered mutually exclusive, in that they may represent the same content (same video, same language audio, etc.) with different encoding or other parameters. MPD 202 may provide information for selecting one of representation groups 210–220, e.g., common representation characteristics 204. This information may include information indicative of whether a client can decode and render a given representation. In this manner, the client device may remove from consideration representations that the client device is incapable of decoding and/or rendering. Accordingly, client device 40 may select a suitable representation group that can be decoded and rendered, then select a representation from the group based on, e.g., network bandwidth availability.

[0138] Client device 40 may also be configured with user preferences for, e.g., rating, language, and/or depth. Accordingly, client device 40 may also select one or more representation groups such that the selected groups conform to the user preferences. Client device 40 may then select a subset of available representation groups that can be played concurrently. When client device 40 is only capable of displaying one view, client device 40 may elect to retrieve data only from one representation. On the other hand, when client device 40 is stereo view or multiview capable, client device 40 may retrieve data from two or more representations.

[0139] After selecting one or more representation groups, client device 40 may select representations from the representation groups based on, e.g., available network bandwidth. As available network bandwidth changes (e.g., increases or decreases), client device 40 may adjust selections of representations from the representation groups to adapt to the changing network bandwidth conditions. Of course, client device 40 may also change representation selections if user preferences or device capabilities (e.g., decoding and rendering capabilities) change.

[0140] Common representation characteristics 204 may correspond to RepresentationGroup XML elements of MPD 202, in some examples. Individual representation characteristics may correspond to sub-elements of corresponding RepresentationGroup elements of MPD 202, in some examples.

[0141] By grouping common characteristics of representations together, various optimizations may be achieved. For example, many representations may have the same values for various parameters. Thus, individually signaling characteristics in the MPD may result in substantial duplication in the MPD to signal characteristics individually. Many client devices are configured to discard the vast majority of the MPD that is received. There could therefore be optimization in the portion of the MPD that the client device receives. Moreover, if a Representation Group is discarded, the client device may have no need to access information currently present in the MPD (URLs, etc.) for the discarded representation or representation group. The client device may also avoid unnecessary updates of URLs, which tend to be updated frequently during, e.g., real-time network streaming of video data for live events. Even if redundancies in the MPD were eliminated, client device 40 would still need to parse the full MPD after receipt and reconstruction, which may waste a significant amount of computing time.

[0142] FIG. 5 is a conceptual diagram illustrating another example multimedia content 250, in which MPD data is separated into various portions for various representation

groups. Multimedia content 250 may correspond to multimedia content 64 (FIG. 1), or another multimedia content stored in memory 62. In particular, a manifest file for multimedia content 250 includes MPD portion 252, which generally includes data related to representation groups. In this example, MPD portion 252 includes representation group data 254A and 254B (representation group data 254), which corresponds to respective representation groups 270–280, as illustrated by arrows pointing from representation group data 254 to respective representation groups 270–280.

[0143] In this example, representation group data 254A includes representation group common characteristics 256A and location of MPD portion for the representation group 258A. That is, location of MPD portion for the representation group 258A indicates the location of MPD portion for representation group 260A. Location of MPD portion for the representation group 258A may correspond, for example, to a URI or URL of MPD portion for representation group 260A. Similarly, representation group data 254B includes representation group common characteristics 256B and location of MPD portion for the representation group 258B, corresponding to MPD portion for representation group 260B.

[0144] MPD portion for representation group 260A includes information representative of characteristics of specific representations 272A, 272B (representations 272) of representation group 270. Similarly, MPD portion for representation group 260B includes information representative of characteristics of specific representations 282A, 282B (representations 282) of representation group 280.

[0145] In this manner, a client device, such as client device 40, may determine an appropriate representation group from which to retrieve data, without receiving representation-specific signaling data for representations that client device 40 would not retrieve, decode, and display. Accordingly, client device 40 may avoid retrieving excess data that would otherwise just be discarded. In particular, after selecting one or more representation groups including representations that can be decoded and rendered by client device 40, client device 40 may retrieve only the MPD portions for the selected representation groups, without retrieving MPD portions for representation groups that could not be properly decoded and/or rendered by client device 40.

[0146] Data of multimedia content 250 may generally correspond substantially to respective elements of multimedia content 200. However, multimedia content 250 may simplify hierarchical download of MPD data for multimedia content 250 by client

devices. For example, rather than retrieving a full manifest file, which may include signaling data for all representations, a client device may simply determine one or more representation groups, then retrieve the MPD portions corresponding to those representation groups, without retrieving MPD portions corresponding to other representation groups that would not be retrieved by the client device (e.g., because the client device does not support decoding and/or rendering procedures to decode and display the representations). In this manner, the data of multimedia content 250 may mitigate inefficiencies of unnecessary downloading and parsing.

[0147] Table 2 below provides an example element that can be added to an MPD, such as MPD 202 of FIG. 4 and/or MPD portion 252 of FIG. 5, that describes characteristics of representation groups. Common representation characteristics 204 (FIG. 4) and/or representation group common characteristics 256 may be formatted according to the structure of Table 2.

TABLE 2

RepresentationGroup	E	1..N	M	This element contains a description of a Representation Group
RepresentationGroupAttri	List of Elements and Attributes			Describes the defaults for this group. may include profile information.
Representation	E	0..N	O	This element contains a description of a Representation.
RepresentationAttribut	List of Elements and Attributes	0,1	O	Describes the Representation attributes that are specific for this Representation
representationListURI	A	0..N	O	URI that points to a document that contains a list of Representations.

[0148] The XML below provides examples of Representation Group elements of an MPD data structure:

```
<RepresentationGroup annotation="ex0" annotation2="ex1">
    <representationListURI="http://www.example.com/representations1.xml/>
</RepresentationGroup>
<RepresentationGroup annotation="ex2" annotation2="ex3">
    <representationListURI="http://www.example.com/representations2.xml/>
</RepresentationGroup>
```

[0149] Table 3 below provides an example set of data that may be included for representations. This data may be provided for individual representations in some examples, while in other examples, all or a portion of the data may be provided for representation groups, according to, e.g., Table 2 above.

TABLE 3

Representation	E	1..N	M	This element contains a description of a Representation.
bandwidth	A		M	The minimum bandwidth of a hypothetical constant bitrate channel in bits per second (bps) over which the representation can be delivered such that a client, after buffering for exactly *minBufferTime* can be assured of having enough data for continuous playout.
...				
texttype	A		O	identifies the type of the text. Options are: sub-title closed-captioning <other to be identified>
cameraangle	A		O	provides the camera angle. Pure annotation, e.g. main, mid-field, players view
Rating	E	0 ... N		provides rating information
SchemeInformation	E	0,1	O	This element gives the information about the used rating scheme. The element can be extended to provide more scheme specific information.
schemeIdUri	A		O	Provides an absolute URL to identify the scheme. The definition of this element is specific to the scheme employed for rating.

[0150] In some examples, data for representation groups and data for individual representations within such groups may be presented within an MPD, such as MPD 202, with a hierarchical relationship. That is, individual representations may be signaled as child elements to a corresponding representation group element of, e.g., MPD 202. Likewise, for MPD portion 252 and MPD portions for representation groups 260,

individual representation characteristics 262, 264 may correspond to child elements of representation group common characteristics 256.

[0151] FIG. 6 is a conceptual diagram illustrating another example multimedia content 300 that may be used to support trick modes. Multimedia content 300 may correspond to multimedia content 64 (FIG. 1), or another multimedia content stored in memory 62. In this example, MPD 302 includes representation information 304, which may include temporal sub-sequence information 306. Representation information 304 includes characteristics of representation 310, in this example. Representation 310 includes segments 312A–312D (segments 312). In this example, each of segments 312 includes a respective sub-fragment index box 314 and random access point (RAP) data 316. In other examples, some segments may include no random access points, while some segments may include multiple random access points. Random access points may include IDR or ODR pictures.

[0152] Client device 40 may extract a temporal sub-sequence from representation 310. For example, client device 40 may extract each of RAPs 316 to form a temporal sub-sequence of representation 310. Alternatively, client device 40 may retrieve a subset of RAPs 316, such as RAPs 316A and 316C, or 316A and 316D. By retrieving and playing back only random access points 316 (or subsets thereof), client device 40 may play representation 310 in a trick mode, e.g., fast forward or rewind. Similarly, client device 40 may skip or seek to a particular one of random access points 316 to begin playback from a requested temporal position.

[0153] Multimedia content may include either or both of temporal sub-sequence information 306 and/or SFIX boxes 314 to indicate information for trick modes. Temporal sub-sequence information 306 may include a "Trick Mode" element of MPD 302, such as that defined by Table 4 below:

TABLE 4

TrickMode	E	0, 1	O	Provides the information for trick mode. It also indicates that the Representation may be used as a trick mode Representation.
alternatePlayoutRate	A		O	Specifies the maximum playout rate as a multiple of the regular playout rate, which this Representation supports with the same decoder profile and level requirements as the normal playout rate.
TemporalSubSequence	E	0 … N	O	Specifies that this Representation contains a

				temporal sub-sequence that can be easily accessed by byte ranges using the information of the Sub-Fragment Index ('sfix) Box.
frameRate	A		M	Specifies the frame rate of the temporal sub-sequence.
bandwidth	A		O	Specifies the minimum bandwidth of a hypothetical constant bitrate channel in bits per second (bps) over which the temporal sub-sequence can be delivered such that a client, after buffering for exactly *minBufferTime* can be assured of having enough data for continuous playout.
alternatePlayoutRate	A		O	Specifies the maximum playout rate as a multiple of the regular playout rate, which this temporal sub-sequence supports with the same decoder profile and level requirements as the normal playout rate.

[0154] In the example of Table 4, the Trick Mode element includes a Temporal SubSequence element that specifies that a corresponding representation contains a temporal sub-sequence that can be accessed by byte ranges using information of sub-fragment index boxes 314. RAPs 316 may correspond to portions of movie fragments, such as movie fragments 162 illustrated in FIG. 3.

[0155] Sub-fragment index boxes 314 may generally describe byte range locations of random access points 316 of corresponding segments 312. In general, sub-fragment index boxes 314 may appear after a segment index (SIDX) box (not shown in FIG. 6) of segments 312 and provide movie fragment prefix sizes for movie fragments referenced in the immediately preceding segment index box. Table 5 below provides properties of an example SFIX box.

TABLE 5----Sub-Fragment Index Box Properties

Box Type	SFIX
Container	None
Mandatory	No
Quantity	One per Segment Index Box

[0156] The pseudocode below provides an example syntax for Sub-Fragment Index Boxes 314:

```
aligned(8) class SubFragmentIndexBox
extends FullBox('strf', 0, 0) {
        unsigned int(32) fragment_count;
        unsigned int(8) sub_fragment_count;
        for( i=0; i < fragment_count; i++ )
          for( j=0; j < sub_fragment_count-1; j++ )
            unsigned int(32) prefix_size;
}
```

[0157] The description below provides an example set of semantics for the syntax described above:

fragment_count specifies the number of fragments for which sub-fragment information is specified in this box. This must be equal to the number of fragment references in the immediately preceding Segment Index Box.

sub_fragment_count specifies the number of sub-fragments per fragment

prefix_size specifies the size of the prefix of fragment i occupied by sub-fragment j.

[0158] In addition, or in the alternative, a sub-track fragment box may be included in segments 312. Whereas the sub-fragment index box may provide syntax information that can be retrieved by client device 40 along with a segment index box in advance of requesting media data, the sub-fragment index box may provide information for client device 40 to construct byte range requests which target subsets of fragment data, for example, temporal sub-layers.

[0159] The Sub-Track Fragment box may specify a re-ordering of the sample data of the track fragment, such that the samples of each sub-track fragment precede all samples that appear only in higher sub-track fragments. Samples of a sub-track fragment that do not appear in any lower sub-track fragment may be placed contiguously within the file (e.g., a corresponding one of segments 312) in the same order as they appear in the Track Run box. This may allow for samples to be stored in order of temporal scalability layer within the track fragment. When this box is present there may be only one Track Run box.

[0160] Table 6 describes properties of the sub-track fragment box:

TABLE 6—Sub-Track Fragment Box Properties

Box Type	STRF
Container	Track fragment box ("TRAF")
Mandatory	No
Quantity	Zero or one

[0161] The pseudocode below illustrates an example syntax for the sub-track fragment box:

```
aligned(8) class SubTrackFragBox
extends FullBox('strf', 0, 0) {
        unsigned int(8) sub_track_count;
        unsigned int(16) sample_count[sub_track_count-1];
        for( i=0; i < sub_track_count; i++ )
        {
                for (j=0; j < sample_count[i]; j++ )
                        bit(1) cur_sub_trak_flag;
        }
        reserved_trailing_bits;
}
```

[0162] The description below provides example semantics for the example syntax of the sub-track fragment box described above:

sub_track_count indicates the number of sub-track fragments; When this box is present, the sub_track_count can be equal or larger than 2.

sample_count[i] indicates the number of samples in sub-track fragment with an index of i+1. Samples of a sub-track fragment are considered to be members of all sub-track fragments with smaller index values,. The number of samples in sub-track fragment 0 is equivalent to the number of zeros of the first bitstring in the subsequent loop. The number of samples in sub-track fragment with an index of sub_track_count-1, which is sample_count[sub_track_count-1], is equal to the number of samples in the Track Fragment.

cur_sub_track_flag equal to 1 in the iteration i of the outer loop indicates that a sample belongs to sub-track fragment with an index of i+1. This value equal to 0

in the iteration of the outer loop indicates that a sample belongs to a sub-track fragment with an index smaller than i+1. Note: That is, the first iteration of the loop contains sample_count[0] flags indicating the positions of the samples in sub-track-fragment 1 that are not also in sub-track-fragment 0. The second iteration of the loop contains sample_count[1] flags indicating the positions of the samples in sub-track-fragment 2 and not also in sub-track-fragment 1 etc. sample_count[sub_track_count-1] is considered equal to the number of samples in the Track Fragment.

[0163] Trick modes may be applied to a variety of different scenarios. For example, trick modes may be used to pause a service temporarily, resume service after a pause, rewind for a period of time, and/or fast forward to proceed to a desired temporal position (e.g., after playback gets interrupted or to seek to a particular desired temporal position).

[0164] Supporting trick modes using temporal subsequences may provide a number of advantages. For example, temporal subsequences may support various frame rates relatively easily. Likewise, the representation including a temporal subsequence may be used for regular playout, as the representation is not limited to the temporal subsequence. Furthermore, encoding with temporal subsequences may be highly efficient. Temporal subsequences also need not require any new coding profiles or levels, can re-use the regular representations, avoid additional client complexity, enable simple content provisioning, provide bandwidth, cache, and storage efficiency, provide flexibility to the client implementation to optimize user experience, to be common among different trick mode operations, and can be applicable to a broad spectrum of client implementations, and can provide relatively good user experience in terms of start-up delay after seeking, as well as good frame rates, reactiveness, and other such metrics.

[0165] FIG. 7 is a conceptual diagram illustrating another example multimedia content 350 in which segments 362A–362D may include MPD update boxes 364 to indicate that MPD 352 is to be updated. Multimedia content 350 may correspond to multimedia content 64 (FIG. 1), or another multimedia content stored in memory 62. In general, MPD 352 includes representation information 354 for representation 360, such as characteristics of representation 360 and URIs or URLs of segments 362 of representation 360. In some cases, representation 360 may be formed from live content,

e.g., a sporting event, and therefore, URIs of segments 362 might not be determined in advance. Therefore, as segments of representation 360 are formed, one or more of the segments may include MPD update boxes to indicate that MPD 352 is to be updated.

[0166] For example, in FIG. 7, segment 362A includes MPD update box 364 and segment data 366A. Segment data 366A may generally be formed according to video file 150 (FIG. 3). However, segment 362A also includes MPD update box 364A. In this manner, client device 40 may update MPD 352 based on data of MPD update box 364A. MPD update box 364A may include updates to MPD 352, or may include a URI or URL of an update for MPD 352. It should be understood that data of MPD update boxes 364 is not necessarily included in explicit boxes. For example, data that conforms substantially to the data of MPD update boxes 364 may be included in other boxes of segments 362, or in a header portion of segments 362. In this manner, a "portion" of segments 362 that includes MPD update information may correspond to a header portion, an MPD update box similar to MPD update boxes 364, or data included in one or more other boxes of segments 362.

[0167] In this manner, after retrieving data of segment 362A, client device 40 may analyze MPD update box 364A to update MPD 352. Client device 40 may then use the updated version of MPD 352 to retrieve segments 362B and 362C. Segments 362B and 362C include segment data 366B, 366C, which again may be formatted according to video file 150 of FIG. 3. Client device 40 may also retrieve data of segment 362D. In this example, segment 362D includes MPD update box 364B, which client device 40 may use to perform another update to MPD 352, in a manner that conforms substantially to the first update. Accordingly, to receive segments beyond segment 362D of representation 360, client device 40 may use the newly updated version of MPD 352, based on updates performed with respect to data of MPD update box 364B.

[0168] An MPD update box, such as MPD update boxes 364A, 364B, may include properties according to Table 7 below:

TABLE 7—MPD Update Box Properties

Box Type	MUPE
Container	None
Mandatory	No
Quantity	Zero or One

[0169] The following syntax may be used to define an MPD update box, in some examples:

```
aligned(8) class MPDUpdateBox
    extends FullBox('mupe') {
    unsigned int(3)  mpd_information_flags;
    unsigned int(1)  new_location_flag;
    unsigned int(28) latest_mpd_update_time;

    /// The following are optional fields
    string mpd_location
}
```

[0170] An example set of semantics for the example MPD update box syntax is provided below:

mpd_information_flags contains the logical OR of zero or more of the following:

0x00 Media Presentation Description update now

0x01 Media Presentation Description update ahead

0x02 End-of-presentation

0x03-0x07 reserved

new_location_flag if set to 1, then the new Media Presentation Description is available at a new location specified in mpd_location.

latest_mpd_update_time specifies the time in ms by when the MPD update is necessary relative to the MPD issue time of the latest MPD. The client may choose to update the MPD any time between now.

mpd_location is present if and only if the new_location_flag is set and provides a Uniform Resource Locator for the new Media Presentation Description.

[0171] In this manner, in-band signaling at the segment level may be used to indicate updates to MPD 302. Updates may be provided at segment boundaries, in some examples. That is, MPD update boxes 364 may occur only at the beginnings or at the ends of respective segments, in various examples. In some examples, if bandwidth of MPD updates presents an issue, server device 60 (FIG. 1) may offer MPDs for certain device capabilities, such that only these parts are updated. Moreover, an MPD element of MPD 302 may provide a publishing time in wall-clock time of MPD 302. This may provide a unique MPD publish time, which may provide unique identifier for the MPD and when the MPD was issued. It may also provide an anchor for the update procedures. Furthermore, server device 60 and/or content preparation device 20 may

optimize MPD updates using hierarchical structures, e.g., to update only portions of MPD 302 that require updates, without changing other portions of MPD 302 that do not need updating.

[0172] Advertisement insertion, such as targeted advertisement insertion, may also be performed using MPD update boxes similar to those of FIG. 7. That is, an MPD update box may be provided to direct client device 40 to retrieve data from advertisement multimedia content. This may occur during timeouts or other actions in sporting events that delay play of the game, and likewise, in timeouts or delays of exciting action for video playback. Because such events may happen somewhat randomly, times at which advertisements are to be inserted might not be known a priori.

[0173] Updating of MPD 302 may be feasible in an asynchronous manner to the delivery of segments. Server device 60 may provide guarantees to client device 40 that an MPD will not be updated for a particular amount of time. However, server device 60 need not explicitly signal when the MPD is updated before a minimum update period. Completely synchronous playout may be hardly achieved, as client devices may operate on different MPD update instances. Therefore, clients may experience drift. Time-shift viewing may be provided for by server device 60 and/or content preparation device 20.

[0174] FIG. 8 is a flowchart illustrating an example method for providing indications of representation groups by a server device, and for selecting representation groups by a client device as well as an individual representation within the selected representation group. Although the method of FIG. 8 is described with respect to server device 60 and client device 40, it should be understood that other devices may implement techniques similar to those of the method of FIG. 8. For example, content preparation device 20, or one or more network devices of a content delivery network, may perform some or all of the functions attributed to server device 60.

[0175] Server device 60 may initially obtain (e.g., create, or receive from content preparation device 20) data for a set of representations of multimedia content, where the representations in the set have one or more common characteristics, as well as a manifest file for the multimedia content. The set of representations may correspond to a representation group. Server device 60 may provide indications of representation groups to client device 40 (400). For example, server device 60 may provide MPD 202 (FIG. 4) or MPD portion 252 (FIG. 5) to client device 40. Other example MPDs of FIGS. 2, 6, and 7 may also include indications of representation groups, such as representation group XML elements. In any case, client device 40 may receive

information describing representation group characteristics (402), e.g., from the MPD file or portion of the MPD file received from server device 60.

[0176] Client device 40 may then analyze the representation group characteristics to eliminate representation groups that client device 40 cannot or would not elect to retrieve, decode, or render. For example, client device 40 may compare decoding and rendering capabilities to characteristics of the representation groups to determine inappropriate representation groups. As another example, client device 40 may compare user preferences for language, rating, and amount of depth (e.g., as provided by two or more views having particular camera angles), to eliminate undesirable representation groups. Client device 40 may then select an appropriate representation group based at least in part on the decoding and rendering capabilities of client device 40 (404). Of course, it should be understood that this selection may also (additionally or alternatively) be made based on user preferences, as discussed above. In this manner, client device 40 may select a set of representations based on common characteristics for the set of representations.

[0177] After selecting a representation group, client device 40 may request data for an MPD portion that specifically describes representations of the representation group. In response, server device 60 may provide indications of representation bitrates, among other individual representation characteristics, in the selected representation group to client device 40 (406). For example, server device 60 may send data for a specific one of MPD portions for representation groups 260 (FIG. 5) to client device 40. In other examples, client device 40 may have already received a full MPD for the multimedia content (e.g., MPD 202 of FIG. 4), but may particularly analyze portions of the MPD corresponding specifically to the selected representation group. In this manner, in some examples, step 406 of FIG. 8 may occur prior to step 402 and/or step 404.

[0178] In any case, after receiving characteristics specific to representations of the selected representation group including bitrates for the representations (408), client device 40 may determine a currently available amount of network bandwidth (410). Client device 40 may then select a representation from the selected representation group (412), such that the selected representation has a bitrate that can be accommodated by the determined currently available amount of network bandwidth. The bitrates of the representations represent examples of coding characteristics of the individual representations in the representation group. Client device 40 may then request data of the selected representation (414). For example, client device 40 may construct (e.g.,

generate and send) an HTTP GET request to request a segment of the selected representation. Alternatively, client device 40 may construct an HTTP partial GET that specifies a byte range of a segment of the selected representation. In any case, client device 40 may submit the request to server device 60.

[0179] Server device 60 may receive the request, and in response, send the requested data to client device 40 (416). For example, request processing unit 70 may determine a network address of client device 40 from data of the received request, e.g., a source Internet protocol (IP) address and source port of the received request. Request processing unit 70 may form network packets including the requested data and send the requested data to client device 40, e.g., destined for the determined IP address of client device 40.

[0180] After receiving the requested data, client device 40 may begin decoding and displaying the received data (418). While receiving requested data, client device 40 may continue analyzing currently available network bandwidth and submitting requests from representations having bitrates that can be accommodated by the currently available amount of network bandwidth (410–414). If the amount of network bandwidth changes, client device 40 may adaptively switch to a different representation in the selected representation group. For example, client device 40 may determine a segment in a new representation corresponding to the temporal location of the last segment requested from a previous representation in the representation group, then request the determined segment (or a portion thereof) in the new representation.

[0181] In some examples, server device 60 may provide an MPD corresponding to targeted advertisement insertion to client device 40 during the method of FIG. 8. The MPD may cause client device 40 to retrieve advertisement multimedia data targeted to a user of client device 40. In some examples, client device 40 may further provide user information to server device 60 to target the advertisement media data to the user of client device 40. The user information may include user preferences, user identifying information (such as a user ID), user demographic information, or other such information. The targeted add insertion may occur, for example, prior to step 400 of FIG. 8, or after step 418 and before selecting a subsequent representation, e.g., for a subsequent period of the multimedia content.

[0182] In this manner, the method of FIG. 8 represents an example of a method including analyzing at least a portion of a manifest file for multimedia content, wherein the portion of the manifest file includes information indicative of sets of representations

of the multimedia content and information indicative of common characteristics for each of the sets of representations, selecting one of the sets of representations based on the common characteristics for the one of the sets of representations, selecting one of the representations of the selected one of the sets of representations based on one or more coding characteristics of the one of the representations of the one of the sets, and generating a request for data of the one of the representations based on the selection.

[0183] Likewise, the method of FIG. 8 represents an example of a method including obtaining a set of representations of multimedia content having one or more common characteristics, wherein each of the representations in the set have individual coding characteristics separate from the common characteristics, obtaining a manifest file for the multimedia content, wherein the manifest file includes information indicative of the representations in the set, information indicative of the common characteristics for the set of representations, and information indicative of the coding characteristics for each of the representations in the set, and sending at least a portion of the manifest file to a client device.

[0184] FIG. 9 is a flowchart illustrating an example method for providing data representative of a trick mode by a server device, and for using the data by a client device to retrieve and play trick mode data of multimedia content. Although the method of FIG. 9 is described with respect to server device 60 and client device 40, it should be understood that other devices may implement techniques similar to those of the method of FIG. 9. For example, content preparation device 20, or one or more network devices of a content delivery network, may perform some or all of the functions attributed to server device 60. Moreover, the selection of a trick mode may be performed in conjunction with selection of a representation group and a representation from the representation group, as described with respect to FIG. 8 above.

[0185] Server device 60 may initially obtain (e.g., create, or receive from content preparation device 20) data for one or more representations of multimedia content, where at least one of the representations includes a temporal sub-sequence, as well as a manifest file for the multimedia content. The manifest file may indicate that the representation includes a temporal sub-sequence. Server device 60 may provide indications of representations of multimedia content, e.g., characteristics of the representations (430) to client device 40. In addition, server device 60 may provide indications of temporal sub-sequences of one or more of the representations (432). That is, server device 60 may provide information in an MPD file for the multimedia content

indicating that temporal sub-sequences are available for one or more representations of the multimedia content. For example, server device 60 may provide at least a portion of an MPD including a trick mode element having a temporal sub-sequence sub-element to client device 40, as described with respect to FIG. 6 above.

[0186] Client device 40 may select a representation based on characteristics of the representations of the multimedia content (434). Although client device 40 need not necessarily select a representation with a temporal sub-sequence, for purposes of discussion to illustrate these techniques, it is assumed for purposes of example that client device 40 selects a representation for which a temporal sub-sequence is available. Client device 40 may then receive a request to use a trick mode (436). For example, client device 40 may receive a selection of a particular temporal location from which to begin playback, e.g., from a user of client device 40. Alternatively, client device 40 may receive a request to fast forward or rewind the video data.

[0187] In response to the request to use the trick mode, client device 40 may determine whether a temporal sub-sequence for the representation is available, and if so, to request data for retrieving at least a portion of the temporal sub-sequence (438). Server device 60 may respond to the request by providing indications of locations of data for the temporal sub-sequence to client device 40 (440). In some examples, a portion of the MPD for the multimedia content may indicate locations of the data for the temporal sub-sequence. In other examples, client device 40 may request sub-fragment index boxes and/or sub-track fragment boxes from segments of the corresponding representation.

[0188] In any case, client device 40 may use the received data including information indicative of locations of data for the temporal sub-sequence to request data of the temporal sub-sequence from the specified locations (442). For example, client device 40 may determine locations (e.g., URLs of segments and, possibly, byte ranges of the segments) including IDR random access points and/or ODR random access points. Client device 40 may then construct HTTP GET or partial GET requests for data of the temporal sub-sequence, in order to play video data according to the trick mode.

[0189] After receiving the HTTP GET and/or partial GET requests from client device 40, server device 60 may provide the requested data to client device 40 (444). For example, server device 60 may send segments in response to HTTP GET requests or media fragments of segments (or portions of media fragments) in response to HTTP partial GET requests. After receiving the requested data, client device 40 may decode and display the received data (446). Likewise, client device 40 may continue requesting

data from the representation (or a different representation if the amount of available network bandwidth changes).

[0190] In this manner, the method of FIG. 9 represents an example of a method including analyzing information of a manifest file for multimedia content, wherein the information of the manifest file indicates that at least one representation of the multimedia content includes a temporal sub-sequence, determining one or more locations of data for the temporal sub-sequence, and submitting one or more requests for the data for the temporal sub-sequence.

[0191] Likewise, the method of FIG. 9 represents an example of a method including obtaining data for at least one representation of multimedia content that includes a temporal sub-sequence, obtaining data for a manifest file for the multimedia content, wherein information of the manifest file indicates that the at least one representation of the multimedia content includes the temporal sub-sequence, and sending at least a portion of the manifest file to a client device.

[0192] FIG. 10 is a flowchart illustrating an example method for providing, by a server device, indications that a manifest file, such as an MPD, is to be updated, and for updating an MPD by a client device. Although the method of FIG. 10 is described with respect to server device 60 and client device 40, it should be understood that other devices may implement techniques similar to those of the method of FIG. 10. For example, content preparation device 20, or one or more network devices of a content delivery network, may perform some or all of the functions attributed to server device 60. Moreover, updating of an MPD may be performed in conjunction with selection of a trick mode and/or selection of a representation group and a representation from the representation group, as described with respect to FIGS. 8 and 9 above.

[0193] In some examples, content preparation device 20 may encode and encapsulate encoded video data captured during a live event, such as a sporting event. In this manner, client device 40 may retrieve coded data of the event in nearly real time as the event is occurring. Initially, server device 60 may receive one or more representations of multimedia content corresponding to the live event and provide indications of characteristics for representations of the multimedia content in an MPD (460). The MPD may only describe characteristics and locations of segments up to a particular temporal location of the multimedia content, due to the multimedia content being formed as the event is being filmed live.

[0194] Client device 40 may use the information of the MPD to select a representation (462). Using the current MPD, client device 40 may request segments of the selected representation, e.g., up to the temporal location. In response, server device 60 may send the requested segments. However, in addition, server device 60 may send a segment including an MPD update box or other information indicating that the MPD is to be updated as of that segment (466).

[0195] In response, client device 40 may decode and display data of one or more of the received segments (468). Client device 40 may also receive the information indicating that the MPD is to be updated (470). For example, client device 40 may receive the last segment before the temporal location at which the MPD no longer applies. Client device 40 may determine that the last segment includes an MPD update box, similar to the MPD update boxes described with respect to FIG. 7.

[0196] Using the update box, client device 40 may request updates to the MPD (472). For example, client device 40 may determine a network location of updates for the MPD and request the updates from the determined location. Server device 60, or another device storing updates to the MPD (e.g., content preparation device 20) may send information indicative of the updates to the MPD (474), which client device 40 may use to update the MPD (476). Alternatively, in some examples, the MPD update box may include information indicative of the MPD updates itself, in which case client device 40 may update the MPD using the information of the MPD update box. In any case, client device 40 may then request segments following the temporal location at which the previous MPD no longer applied, using the updated version of the MPD (478). Client device 40 and server device 60 may continue performing similar steps until client device 40 has finished playback of the multimedia content.

[0197] In some examples, techniques similar to the method of FIG. 10 may be used to perform targeted advertisement insertion. For example, an updated MPD may include a portion that corresponds to advertisement media content. Client device 40 may be required to retrieve and play back data of the advertisement media content, based on the updated MPD, in order to receive data of one or more segments of the advertisement media content, which may include another updated MPD for retrieving subsequent media data of the desired media content.

[0198] In this manner, the method of FIG. 10 represents an example of a method including retrieving data of a first segment of a representation of multimedia content in accordance with data of a copy of a manifest file stored by a client device, retrieving a

portion of a second segment of the representation in accordance with the manifest file, wherein the second segment occurs after the first segment in the representation, and wherein the portion of the second segment indicates that the manifest file is to be updated, updating the copy of the manifest file stored by the client device based on the indication that the manifest file is to be updated, and retrieving media data of the second segment in accordance with the updated manifest file.

[0199] Likewise, the method of FIG. 10 represents an example of a method including sending data of a manifest file of multimedia content to a client device, wherein the manifest file includes information indicative of a first segment of a representation of the multimedia content, sending at least a portion of the first segment of the representation to the client device in response to a request from the client device, wherein the portion of the first segment indicates that the manifest file is to be updated, wherein an updated version of the manifest file includes information indicative of a second, different segment of the representation, and sending, in response to a request received from the client device and formed according to the updated manifest file, data of the second segment to the client device.

[0200] FIG. 11 is a flowchart illustrating an example method for constructing and using data of a quality of experience (QoE) report document. Although the method of FIG. 11 is described with respect to server device 60 and client device 40, it should be understood that other devices may implement techniques similar to those of the method of FIG. 11. For example, content preparation device 20, or one or more network devices of a content delivery network, may perform some or all of the functions attributed to server device 60. Moreover, providing a QoE report to server device 60 and/or content preparation device 20 may be performed in conjunction with any or all of updating of an MPD, selection of a trick mode, and/or selection of a representation group and a representation from the representation group, as described with respect to FIGS. 8, 9, and 10 above.

[0201] Initially, server device 60 may provide indications of characteristics of representations of multimedia content in an MPD to client device 40 (500). As discussed above, client device 40 may select a representation (502), e.g., based on decoding and/or rendering capabilities of client device 40, user preferences, available network bandwidth, and/or other characteristics of representations of the multimedia content. Client device 40 may then request one or more segments of the selected representation (504).

[0202] Server device 60 may send the requested segments to client device 40 (506). After receiving the requested segments, client device 40 may decode and display the received data (508). Client device 40 may then determine whether all video data has been received (510). If the last segment of the representation (or multimedia content generally) has not been received ("NO" branch of 510), client device 40 may again evaluate a currently available amount of network bandwidth and select a representation based on this analysis (502) and request segments of the representation (504).

[0203] In general, client device 40 may buffer the data and attempt to avoid buffer over- and under-flow by requesting data of the multimedia content from a representation that can be accommodated by currently available network bandwidth. At times, however, buffer overflow or underflow may occur, e.g., if actual coding characteristics of the multimedia content did not match the signaled coding characteristics or if there was insufficient data for client device 40 to make a proper selection. Other factors may also result in reduced quality of experience for a user of client device 40. Therefore, after the last segment of the representation (or multimedia content) has been received and properly decoded ("YES" branch of 510), client device 40 may provide a quality of experience (QoE) report to server device 60.

[0204] For example, client device 40 may construct the report to include indications of selected segments and representations (512). Client device 40 may also record occurrences of buffer overflow/underflow, which may result in pauses in media playback. Client device 40 may form a report including a sequence of PeriodReport elements, representing the Periods that were played out. A Period element may include a sequence of RepresentationReport elements, each representing a continuous playout of a portion of a Representation, and providing the start and end tomes in both real time and presentation time. The report can also include the initial startup time, this being the time from the user request for viewing the content and the start of playout. In this manner, the report document may comprise an extensible markup language (XML) formatted electronic document indicative of representations of the multimedia content from which the client device retrieved media data of the multimedia content.

[0205] Client device 40 may provide the report to server device 60 or another device of a content delivery network, such as content preparation device 20 or a dedicated report collection device. In this manner, server device 60 may receive the indications of the segments and representations received by client device 40 (514). Server device 60 may then provide the indications to, e.g., content preparation device 20 or another device

associated with a service provider or media assembler to improve content preparation (516). From the information provided by client device 40, a service provider may determine exactly what was played out, when there were pauses in playout, and when there were switches between representations. Alternatively or additionally, client device 40 may provide summary information in the form of the total playout duration and number of distinct, continuous playout periods for each representation, together with the number of pauses and the mean and variance of the pause durations.

[0206] Using this data, the service provider may analyze quality of experience information for a new piece of media content for streaming using Adaptive HTTP Streaming. The service provider may make a number of different representations at different bitrates available, and provide HTTP serving infrastructure to host the media files, then collect the feedback to determine quality of users' viewing experiences. The service provider may use this data to improve quality of service for this or future media content hosting. Quality of Experience metrics may refer to actual viewing as experienced by a user and may be independent of client algorithms used for scheduling of HTTP requests, representation selection decisions, and the like. In this manner, the service provider may obtain a relatively accurate portrayal of the quality of a user's viewing experience for a specific viewing session.

[0207] In this manner, the method of FIG. 11 represents an example of a method including constructing a document including information indicative of representations of multimedia content from which media data was retrieved, and sending the constructed document to a server from which the media data was retrieved. The method of FIG. 11 also represents an example of a method including receiving information indicative of data retrieved by a client device, comprising receiving an extensible markup language formatted electronic document including information indicative of representations of the multimedia content from which the client device retrieved media data of the multimedia content.

[0208] In one or more examples, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium and executed by a hardware-based processing unit. Computer-readable media may include computer-readable storage media, which corresponds to a tangible medium such as data storage media, or communication media including any medium that facilitates transfer of a computer program from one place to

another, e.g., according to a communication protocol. In this manner, computer-readable media generally may correspond to (1) tangible computer-readable storage media which is non-transitory or (2) a communication medium such as a signal or carrier wave. Data storage media may be any available media that can be accessed by one or more computers or one or more processors to retrieve instructions, code and/or data structures for implementation of the techniques described in this disclosure. A computer program product may include a computer-readable medium.

[0209] By way of example, and not limitation, such computer-readable storage media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage, or other magnetic storage devices, flash memory, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer. Also, any connection is properly termed a computer-readable medium. For example, if instructions are transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. It should be understood, however, that computer-readable storage media and data storage media do not include connections, carrier waves, signals, or other transitory media, but are instead directed to non-transitory, tangible storage media. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and blu-ray disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0210] Instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Accordingly, the term "processor," as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or software modules configured for encoding and decoding, or incorporated in a combined codec. Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0211] The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses, including a wireless handset, an integrated circuit (IC) or a set of ICs (e.g., a chip set). Various components, modules, or units are described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of interoperative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

WHAT IS CLAIMED IS:

1. A method of retrieving multimedia data, the method comprising:

analyzing information of a manifest file for multimedia content, wherein the information of the manifest file indicates that at least one representation of the multimedia content includes a temporal sub-sequence;

determining one or more locations of data for the temporal sub-sequence; and

submitting one or more requests for the data for the temporal sub-sequence.

2. The method of claim 1, further comprising presenting the data of the temporal sub-sequence in a trick mode for the representation.

3. The method of claim 1, wherein determining the locations of the data comprises determining the locations of the data for the temporal sub-sequence from the manifest file.

4. The method of claim 1, wherein determining the locations of the data comprises:

retrieving data for a portion of the representation, wherein the portion of the representation includes data indicative of one or more locations of data for the temporal sub-sequence; and

analyzing the data for the retrieved portion of the representation to determine the locations of the data for the temporal sub-sequence of the representation.

5. The method of claim 4, wherein the data for the portion of the representation corresponds to continuous byte sequences of respective segments of the representation.

6. The method of claim 4, wherein the portion of the representation comprises a sub-fragment index box of the representation.

7. The method of claim 4, wherein retrieving the data for the portion of the representation comprises:

determining a starting byte of the portion of the representation and an ending byte of the portion of the representation from data of the manifest file; and

sending a partial GET request that specifies the starting byte, the ending byte, and an identifier of the representation.

8. The method of claim 1, wherein the data for the temporal sub-sequence comprises one or more instantaneous decoder refresh (IDR) pictures of the representation.

9. A device for retrieving multimedia data, the device comprising one or more processors configured to analyze information of a manifest file for multimedia content, wherein the information of the manifest file indicates that at least one representation of the multimedia content includes a temporal sub-sequence, determine one or more locations of data for the temporal sub-sequence, and submit one or more requests for the data for the temporal sub-sequence.

10. The device of claim 9, wherein the one or more processors are configured to determine the locations of the data for the temporal sub-sequence from the manifest file.

11. The device of claim 9, wherein to determine the locations of the data, the one or more processors are configured to retrieve data for a portion of the representation, wherein the portion of the representation includes data indicative of one or more locations of data for the temporal sub-sequence, and analyze the data for the retrieved portion of the representation to determine the locations of the data for the temporal sub-sequence of the representation.

12. The device of claim 11, wherein the portion of the representation comprises a sub-fragment index box of the representation.

13. The device of claim 11, wherein to retrieve the data for the portion of the representation, the one or more processors are configured to determine a starting byte of the portion of the representation and an ending byte of the portion of the representation from data of the manifest file, and send a partial GET request that specifies the starting byte, the ending byte, and an identifier of the representation.

14. The device of claim 9, wherein the device comprises at least one of:
an integrated circuit;
a microprocessor; and
a wireless communication device that includes the one or more processors.

15. A device for retrieving multimedia data, the device comprising:

means for analyzing information of a manifest file for multimedia content, wherein the information of the manifest file indicates that at least one representation of the multimedia content includes a temporal sub-sequence;

means for determining one or more locations of data for the temporal sub-sequence; and

means for submitting one or more requests for the data for the temporal sub-sequence.

16. The device of claim 15, wherein the means for determining the locations of the data comprises means for determining the locations of the data for the temporal sub-sequence from the manifest file.

17. The device of claim 15, wherein determining the locations of the data comprises:

means for retrieving data for a portion of the representation, wherein the portion of the representation includes data indicative of one or more locations of data for the temporal sub-sequence; and

means for analyzing the data for the retrieved portion of the representation to determine the locations of the data for the temporal sub-sequence of the representation.

18. The device of claim 17, wherein the portion of the representation comprises a sub-fragment index box of the representation.

19. The device of claim 17, wherein the means for retrieving the data for the portion of the representation comprises:

means for determining a starting byte of the portion of the representation and an ending byte of the portion of the representation from data of the manifest file; and

means for sending a partial GET request that specifies the starting byte, the ending byte, and an identifier of the representation.

20. A computer program product comprising a computer-readable medium having stored thereon instructions that, when executed, cause a processor of a device for retrieving multimedia data to:

analyze information of a manifest file for multimedia content, wherein the information of the manifest file indicates that at least one representation of the multimedia content includes a temporal sub-sequence;

determine one or more locations of data for the temporal sub-sequence; and

submit one or more requests for the data for the temporal sub-sequence.

21. The computer program product of claim 20, wherein the instructions cause the processor to present the temporal sub-sequence of the representation in a trick mode for the representation.

22. The computer program product of claim 20, wherein the instructions that cause the processor to determine the locations of the data comprise instructions that cause the processor to determine the locations of the data for the temporal sub-sequence from the manifest file.

23. The computer program product of claim 20, wherein the instructions that cause the processor to determine the locations of the data comprise instructions that cause the processor to:

retrieve data for a portion of the representation, wherein the portion of the representation includes data indicative of one or more locations of data for the temporal sub-sequence; and

analyze the data for the retrieved portion of the representation to determine the locations of the data for the temporal sub-sequence of the representation.

24. The computer program product of claim 23, wherein the portion of the representation comprises a sub-fragment index box of the representation.

25. The computer program product of claim 23, wherein the instructions that cause the processor to retrieve the data for the portion of the representation comprise instructions that cause the processor to:

determine a starting byte of the portion of the representation and an ending byte of the portion of the representation from data of the manifest file; and

send a partial GET request that specifies the starting byte, the ending byte, and an identifier of the representation.

26. A method of sending information for multimedia data, the method comprising:

obtaining data for at least one representation of multimedia content that includes a temporal sub-sequence;

obtaining data for a manifest file for the multimedia content, wherein information of the manifest file indicates that the at least one representation of the multimedia content includes the temporal sub-sequence; and

sending at least a portion of the manifest file to a client device.

27. The method of claim 26, wherein the manifest file includes information indicative of locations of data for the temporal sub-sequence.

28. The method of claim 26, wherein a portion of the representation includes data indicative of one or more locations of data for the temporal sub-sequence, the method further comprising:

receiving a request for the portion of the representation from the client device; and

sending the portion of the representation to the client device in response to the request.

29. The method of claim 28, wherein the portion of the representation comprises a sub-fragment index box of the representation.

30. The method of claim 28, wherein the manifest file indicates a byte range for the portion of the representation, and wherein the request comprises a partial GET request that specifies the byte range for the portion of the representation.

31. A device for sending information for multimedia data, the device comprising one or more processors configured to obtain data for at least one representation of multimedia content that includes a temporal sub-sequence, obtain data for a manifest file for the multimedia content, wherein information of the manifest file indicates that the at least one representation of the multimedia content includes the temporal sub-sequence, and send at least a portion of the manifest file to a client device.

32. The device of claim 31, wherein the manifest file includes information indicative of locations of data for the temporal sub-sequence.

33. The device of claim 31, wherein a portion of the representation includes data indicative of one or more locations of data for the temporal sub-sequence, and wherein the one or more processors are configured to receive a request for the portion of the representation from the client device, and send the portion of the representation to the client device in response to the request.

34. The device of claim 33, wherein the portion of the representation comprises a sub-fragment index box of the representation.

35. The device of claim 33, wherein the manifest file indicates a byte range for the portion of the representation, and wherein the request comprises a partial GET request that specifies the byte range for the portion of the representation.

36. The device of claim 31, wherein the device comprises at least one of:

an integrated circuit;

a microprocessor; and

a wireless communication device that includes the one or more processors.

37. A device for sending information for multimedia data, the device comprising:

means for obtaining data for at least one representation of multimedia content that includes a temporal sub-sequence;

means for obtaining data for a manifest file for the multimedia content, wherein information of the manifest file indicates that the at least one representation of the multimedia content includes the temporal sub-sequence; and

means for sending at least a portion of the manifest file to a client device.

38. The device of claim 37, wherein the manifest file includes information indicative of locations of data for the temporal sub-sequence.

39. The device of claim 37, wherein a portion of the representation includes data indicative of one or more locations of data for the temporal sub-sequence, further comprising:

means for receiving a request for the portion of the representation from the client device; and

means for sending the portion of the representation to the client device in response to the request.

40. The device of claim 39, wherein the portion of the representation comprises a sub-fragment index box of the representation.

41. The device of claim 39, wherein the manifest file indicates a byte range for the portion of the representation, and wherein the request comprises a partial GET request that specifies the byte range for the portion of the representation.

42. A computer program product comprising a computer-readable medium having stored thereon instructions that, when executed, cause a processor of a device for sending information for multimedia data to:

obtain data for at least one representation of multimedia content that includes a temporal sub-sequence;

obtain data for a manifest file for the multimedia content, wherein information of the manifest file indicates that the at least one representation of the multimedia content includes the temporal sub-sequence; and

send at least a portion of the manifest file to a client device.

43. The computer program product of claim 42, wherein the manifest file includes information indicative of locations of data for the temporal sub-sequence.

44. The computer program product of claim 42, wherein a portion of the representation includes data indicative of one or more locations of data for the temporal sub-sequence, further comprising instructions that cause the processor to:

receive a request for the portion of the representation from the client device; and

send the portion of the representation to the client device in response to the request.

45. The computer program product of claim 44, wherein the portion of the representation comprises a sub-fragment index box of the representation.

46. The computer program product of claim 44, wherein the manifest file indicates a byte range for the portion of the representation, and wherein the request comprises a partial GET request that specifies the byte range for the portion of the representation.