

(43) International Publication Date  
27 November 2014 (27.11.2014)

## (51) International Patent Classification:

H04N 21/235 (2011.01) H04N 21/84 (2011.01)  
H04N 21/4722 (2011.01) H04N 21/8547 (2011.01)

## (21) International Application Number:

PCT/US2014/037732

## (22) International Filing Date:

12 May 2014 (12.05.2014)

## (25) Filing Language:

English

## (26) Publication Language:

English

## (30) Priority Data:

13/902,576 24 May 2013 (24.05.2013) US

(71) Applicant: CISCO TECHNOLOGY, INC. [US/US];  
SJC/10/2/1, 170 West Tasman Drive, San Jose, California  
95134-1706 (US).

(72) Inventors: FRIEDRICH, Eric Colin; 6 Gordon St. #2,  
Somerville, Massachusetts 02144 (US). CAULFIELD,  
Matthew Francis; 2 Nathan Dr., Clinton, Massachusetts  
01510 (US). ITURRALDE, Carol Etta; 19 Myrna Rd.,  
Framingham, Massachusetts 01701 (US). VIVE-  
GANANDHAN, Mahesh Vittal; 22826 Poplar Grove Sq.,  
Cupertino, California 95014 (US). LABROZZI, Scott C.;  
302 Lindemans Drive, Cary, North Carolina 27519 (US).

(74) Agent: FRAME, Thomas J.; Patent Capital Group, 2816  
Lago Vista Lane, Rockwall, Texas 75032 (US).

(81) Designated States (unless otherwise indicated, for every  
kind of national protection available): AE, AG, AL, AM,

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

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

## Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

## Published:

- with international search report (Art. 21(3))

(54) Title: ON-DEMAND ENCAPSULATING OF TIMED METADATA IN A NETWORK ENVIRONMENT

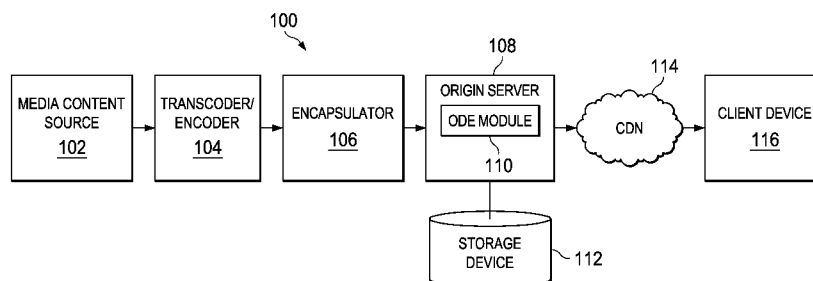


FIG. 1

(57) Abstract: A method is provided in one example and includes receiving media content including timed metadata, generating a common format asset including the timed metadata and a timed metadata index including at least one reference point corresponding to the common format asset. Each reference point includes state context information of the timed metadata at a particular instance of time. The method may further include receiving a request for the timed metadata from a particular client device, and retrieving a determined amount of the timed metadata from the common format asset using the timed metadata index. In a particular example, the determined amount of the timed metadata is an amount sufficient to display a current timed metadata context corresponding to the request.



## ON-DEMAND ENCAPSULATING OF TIMED METADATA IN A NETWORK ENVIRONMENT

TECHNICAL FIELD

[0001] This disclosure relates in general to the field of communications and, more particularly, to providing on-demand encapsulating of timed metadata in a network environment.

BACKGROUND

[0002] An MPEG-2 Transport Stream (MPEG2-TS) as developed by the Moving Picture Expert Group (MPEG) typically contains video, audio, and metadata tracks that are transmitted together in a multiplexed format. When the MPEG2-TS formatted data is converted from the MPEG2-TS format to an adaptive bitrate (ABR) streaming format, the metadata tracks are converted into a format supported by an ABR client. Adaptive bitrate streaming is a technique in which the quality of a media stream is adjusted when the media stream is delivered to a client in order to conform to a desired bitrate. The conversion of metadata tracks should occur for all types of timed metadata including, but not limited, to closed captions, subtitles, application specific metadata, and ad-insertion markers. Existing ABR pipelines convert the source asset into target specific formats and store the result on an origin server until requested by the client. This procedure produces multiple versions of audio, video and metadata tracks for each of the different formats required by each ABR client. This approach negatively impacts storage as each version of the audio, video and metadata tracks should be stored within one or more storage devices as well as negatively impacting bandwidth usage within the network.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] To provide a more complete understanding of the present disclosure and features and advantages thereof, reference is made to the following description, taken in conjunction with the accompanying figures, wherein like reference numerals represent like parts, in which:

[0004] FIGURE 1 is a simplified block diagram of a communication system for providing on-demand encapsulating of timed metadata in a network environment in accordance with one embodiment of the present disclosure;

[0005] FIGURE 2 illustrates a simplified block diagram of an embodiment of the encapsulator of FIGURE 1;

[0006] FIGURE 3 illustrates a simplified block diagram of an embodiment of the origin server and the storage device of FIGURE 1;

5 [0007] FIGURE 4 is a simplified diagram of an example embodiment of common format media;

[0008] FIGURE 5 is a simplified diagram of an embodiment of a media data index file as generated by the encapsulator;

10 [0009] FIGURE 6 is a simplified diagram of an embodiment of a timed metadata index file as generated by the encapsulator;

[0010] FIGURE 7 is a simplified flowchart illustrating one potential operation of the encapsulator of FIGURE 1;

[0011] FIGURE 8 is a simplified flowchart illustrating one potential operation of the origin server; and

15 [0012] FIGURE 9 is a simplified diagram of an embodiment of a common format asset as generated by the encapsulator of FIGURE 1.

## DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

### 20 OVERVIEW

[0013] A method is provided in one example and includes receiving common format media including timed metadata. The common format media can include any suitable information or data. The method further includes generating a common format asset including the timed metadata and a timed metadata index including at least one reference point  
25 corresponding to the common format media. Each reference point can include state context information of the timed metadata at a particular instance of time.

[0014] In more particular embodiments, the method further includes sending the common format asset to at least one server. In still more particular embodiments, the method further includes receiving a request for the timed metadata from a particular client device, and  
30 retrieving a determined amount of the timed metadata from the common format asset using the timed metadata index. In other particular embodiments, the determined amount of the

timed metadata is an amount sufficient to display a current timed metadata context corresponding to the request. In more particular embodiments, the method further includes converting the retrieved timed metadata to a target format suitable for the particular client device, and sending a response message including the retrieved timed metadata in the target  
5 format to the particular client device. In a particular embodiment, the common format media is an MPEG2-TS adaptive transport stream file. In another particular embodiment, the common format media is an ISO Base Media File Format (ISO-BMFF) file. In other particular embodiments, the timed metadata includes at least one of caption data, subtitle data, ad-insertion marker data and application-specific metadata.

10

#### EXAMPLE EMBODIMENTS

[0015] Referring now to FIGURE 1, FIGURE 1 is a simplified block diagram of a communication system 100 for providing on-demand encapsulating of timed metadata in a network environment in accordance with one embodiment of the present disclosure. FIGURE 1  
15 includes a media content source 102, a transcoder/encoder 104, an encapsulator 106, an origin server 108 having an On-Demand Encapsulation (ODE) module 110, a storage device 112, a content distribution network (CDN) 114, and a client device 116. In the particular illustrated embodiment, media content source 102 is in communication with transcoder/encoder 104, and transcoder/encoder 104 is in further communication with encapsulator 106. Encapsulator 106  
20 is in further communication with origin server 108, and origin server 108 is in further communication with storage device 112. Storage device 112 may include one or more of local storage, network storage, or any other suitable storage device. Origin server 108 is further in communication with client device 116 via CDN 114. Client device 116 may include one or more of a set-top box, a television, a computer, a mobile computing device, or any other suitable  
25 client device.

[0016] A fundamental issue in content delivery is the need to serve a wide variety of types of end-client devices. In the context of adaptive bit rate (ABR) video, these various end-client types each typically require specific metadata and video and audio formats. Examples of prevalent ABR client types include Microsoft HTTP Smooth Streaming (HSS), Apple HTTP Live  
30 Streaming (HLS), Adobe HTTP Dynamic Streaming (HDS), and MPEG Dynamic Adaptive Streaming over HTTP (DASH). A server which handles requests from a heterogeneous pool of

ABR clients should typically store its media content including video, audio, and metadata in a form which can be easily translated to a target format that is suitable and recognizable by a particular client device. In a simple implementation, such a server could store a separate copy of a piece of media content for each end client device type. However, this approach negatively  
5 impacts storage and bandwidth usage. In a content distribution network (CDN), multiple formats of the same piece of content will be treated independently, further exacerbating the problem. A CDN is a network of intermediate nodes that function to cache content in a hierarchy of locations to decrease the load on the origin server 108 and to improve the quality of experience for the users using client device 116 to receive media content.

10 [0017] On-demand encapsulation (ODE) addresses the storage and bandwidth issues presented by the simple implementation. With ODE, a single representation of each piece of common format media is stored and cached by the server. Upon receiving a client request for the media content, the server re-encapsulates the common format media representation into an end-client-specific format. ODE provides a tradeoff between storage and computation  
15 requirements. While storing a common format media representation incurs lower storage overhead, re-encapsulating that representation on-demand is more usually expensive computationally than storing each end-client representation individually.

[0018] A common format asset should be chosen to meet the needs of all end-client ABR format types. The common format asset can include any suitable collection of items  
20 including the original common format media. The common format asset may also contain any suitable index files, both media indexes (i.e. audio/video), metadata index files, and a Media Presentation Description (MPD). The MPD is a manifest or file containing information about the media content such as one or more formats of segments of audio or video data. The common format asset and its associated metadata should be capable of being easily translated  
25 into an end-client format. An example of a common format asset that meets this requirement is Adaptive Transport Stream (ATS) with Dynamic Adaptive Streaming over HTTP (DASH) metadata. An Adaptive Transport Stream is an ABR conditioned annotated MPEG-2 Transport Stream (MPEG2-TS) stream with in-band metadata for signaling ABR fragment and segment boundaries. Dynamic Adaptive Streaming over HTTP (DASH) is a standard for describing ABR  
30 content. ISO Base Media File Format (ISO-BMFF) with DASH metadata (DASH/ISO-BMFF) is another example of a common format asset that may be used.

[0019] A typical ABR content workflow for on-demand encapsulation may be understood as a pipeline of functional blocks strung together for the purpose of delivering ABR content to end-clients. Raw/compressed media content arrives into the system and an encoding/transcoding stage converts the content into multiple ABR-conditioned compressed versions. This represents the common format media, which can include these elements, as well as any other information that may be propagating network, or that may be the byproduct of any suitable video processing. An encapsulation stage further processes the common format media to produce a common format asset, which contains the sourced common format media, various indexes of this media content and a media presentation description. A recording stage accepts the common format asset and writes it to storage. An origination stage reads the common format asset and performs re-encapsulation of the media into a target format when a request is received from a particular end client device. The origination stage serves media content in the target format based upon a request received from a client device. The target format of the media content may be based upon client type. In particular examples, a CDN may cache content in a hierarchy of locations to decrease the load on the origination stage and to improve the quality of experience for the users in the client stage. Finally, in a client stage, a client device receives the requested media content decodes and presents the content to the end-user.

[0020] A common format media stream such as an Adaptive Transport Stream typically contains multiplexed video, audio, and timed metadata. When the content is converted to an adaptive bitrate (ABR) streaming format, the metadata is converted into a target format supported by the particular ABR client such as client device 116. This conversion should occur for all types of timed metadata, which includes, but which is not limited to, captions, subtitles, application-specific metadata, ad-insertion markers, or any other suitable metadata. For example, a Microsoft Smooth™ client requires caption data formatted in SMPTE Timed Text Markup Language (TTML).

[0021] Existing non-on-demand ABR pipelines convert the source asset into target specific formats and store the result on an origin server until requested by the client. This typically includes multiple versions of audio, video and metadata tracks for the different ABR formats. However, with on-demand encapsulation technology, origin servers no longer need to store multiple versions of the same ABR asset. Instead, by storing the source asset data using a

common format asset, such as using the common intermediate file (CIF) format, ODE module 110 can create a specific ABR segment needed, in the correct target format, in response to a client's request.

[0022] Applying on-demand encapsulation to timed metadata tracks results in two types of outputs. First, the client manifest produced by on-demand encapsulation indicates the timeline for the timed metadata. Second, the on-demand encapsulation produces timed metadata segments in response to client requests. Just as with audio and video, ODE module 110 may use timed metadata indexing data that describes the location of timed metadata in the media content contained within the common format asset.

[0023] However, there are at least two ODE challenges that arise with respect to processing timed metadata in an on-demand encapsulation context. The first challenge relates to state based metadata, such as timed metadata, and the second to the efficiency of creating timed metadata tracks in the target format. In certain examples described herein, various forms of closed captioning are described. Closed captions are used as an example because they are widely deployed and present a wide range of typical challenges encountered with timed metadata. However, the principles described herein are not specific to caption data and may be applied to other forms of metadata, in particular timed or state based metadata as well.

[0024] ABR segments sent to client device 116 should be fully self-contained and independent of all other segments. In order to correctly render an ABR segment, no previous or future segment should be needed by client device 116. Compressed media such as video contains frame to frame state in which one frame depends on prior (and potentially future) frames, the exception being a closed GOP boundary via an instantaneous decoder refresh (IDR) frame. To be self-contained in the video domain each segment may start with an IDR frame. In on-demand encapsulation processing, for example, a particular video segment N can be generated independent of the processing of video segments prior to N. Given the self-containment nature of the video frames, on-demand encapsulation video segment creation can be non-linear, i.e. not in presentation time order.

[0025] However, some forms of metadata, such as caption data, contain a state information that does not necessarily have points within the timed metadata at which state is reset, so called timed metadata reference points. Caption data authored as paint-on and roll-up styles are continuous in nature and there is no IDR equivalent to delineate a segment. For TTML

HSS captions for example, if one processes data represented by the time domain of a segment, the discrete generation of only segment N would not be equivalent to the same segment N generated sequentially after segment N-1. As this form of metadata is not self-contained, the generation of such stateful metadata cannot be purely non-linear. If these metadata segments are cached, this non-equivalence could cause severe rendering issues. To avoid this, on-demand encapsulation needs to process some amount of continuous timed metadata prior to the particular segment N to ensure the equivalent reconstruction of state and the proper generation of segment N independent of the order in which it was requested.

[0026] Some authored styles of continuous metadata may have boundary points. The pop-on style of closed captions is such an example. However, authored styles may dynamically change in content, for example, transition from pop-on to paint-on and back. Instead of being implicitly derived from other media tracks, this may be taken advantage of in common format asset indexing. Timed metadata indexing may be extended to explicitly delineate metadata tracks, segmenting using known boundary points when available and flagging these while adopting other media tract segment delineations when boundaries are not known.

[0027] On-demand encapsulation efficiency is an issue for fetching small pieces of sparse timed metadata. Timed metadata tends to be of multiple magnitudes smaller than other media tracks such as video and audio and yet by default the metadata may be interleaved with these other tracks. Thus, within indexing of timed metadata the fetching of timed metadata may result in the fetching of other undesired media of much larger bandwidths than the timed metadata. Efficiency is also an issue when fetching data for the purposes of handling stateful continuous metadata. ODE module 110 should fetch only the minimum amount of data needed to reconstruct the correct state. Fetching more data than is needed results in state that is no more correct, but wasteful of bandwidth. Various embodiments described herein describe a procedure for indexing and creating timed metadata tracks by encapsulating, at play out time, only the timed metadata segments needed at that particular time.

[0028] Media content source 102 is in communication with transcoder/encoder 104 and is configured to provide media content to transcoder/encoder 104. In one or more embodiments, the source media may include video and/or audio and associated timed metadata. The timed metadata may include for example, closed caption data, subtitle data, advertisement insertion marker data or any other application specific metadata. In at least one embodiment,

the media content is provided to transcoder/encoder 104 in a raw format. In still other embodiments, the media content may first be encoded such that the raw format media content is converted into a compressed format before being provided to transcoder/encoder 104. In a particular embodiment, the media content is encoded in an MPEG2-TS format. In still other  
5   embodiments, the media content may be encoded in an H.264 format or any other suitable format.

[0029] Transcoder/encoder 104 is configured to transcode the source media into one or more transcoded versions of the media content having bitrate, quality or other parameters that differ from that of the original media content. For example, in particular embodiments,  
10   transcoder/encoder 104 encodes and/or transcodes the source media into one or more lower quality versions of the original media content in order for the media content to be more suitable for streaming. Transcoder/encoder 104 is further configured to pass the transcoded media content to encapsulator 106.

[0030] Encapsulator 106 is configured to generate one or more indexes that indicate  
15   the location of video and/or audio data and associated metadata within the media content. In one or more embodiments, the common format index files may be generated directly from the common format media or via a translation from another format of index file. In at least one embodiment, separate index files may be created for each of the video, audio, and timed metadata to separately store the associated indexing data. In still other embodiments, a single  
20   index file may be created to store indexing data for the video, audio, and metadata. In still other embodiments, video and audio data may be indexed together in a separate file and the timed metadata may be indexed in a separate file by itself. In accordance with various embodiments, encapsulator 106 creates additional state information in the timed metadata index to enable ODE module 110 to later retrieve the correct amount of timed metadata  
25   required for a current segment or chunk of media content currently being delivered to client device 116. In accordance with various embodiments, encapsulator 106 generates the common format asset that may be later converted into a specific target format appropriate for delivery to a particular ABR client device. Encapsulator 106 then sends the common format asset to origin server 108. In response, origin server 108 stores the common format asset  
30   containing one or more indexes including the timed metadata index within storage device 112.

Although the embodiment illustrated in FIGURE 1 shows a single storage device 112, it should be understood that in other embodiments one or more storage devices may be used.

[0031] At a later time, client device 116 requests the timed metadata from origin server 108 via CDN 114 and CDN 114 relays the request to origin server 108. In some embodiments, the request may also include a request for media such as audio or video in addition to a request for specific timed metadata within the media content. In still other embodiments, separate requests may be sent for the timed metadata, video data, and/or audio data. ODE module 110 then retrieves the timed metadata index from storage device 112 and determines the portions of the timed metadata needed to service the request. ODE module 110 then uses the timed metadata index to extract within the common format media only the portions of the timed metadata needed for the duration of time corresponding to the request. ODE module 110 then converts the timed metadata into a target format supported by client device 116, and encapsulates and sends the timed metadata in the target format to client device 116. In one or more embodiments, the timed metadata index file may contain the actual timed metadata, in which case ODE module 110 only needs to process the timed metadata and does not need to extract the timed metadata from the common format file.

[0032] As discussed, in one or more embodiments ODE module 110 is responsible for translating content from the common format into a client-specific target format. In a particular embodiment, the two main transformations that are performed include: 1) Transforming the Media Presentation Description (MPD) into a target format manifest (HLS Variant + Stream Playlists, Smooth Client Manifest); and 2) transforming the common format asset media into target format ABR segments. In a particular embodiment, a third transformation is defined which includes transforming from asset media into target format ABR timedmetadata segments. This transformation uses the common format asset indexing data to more efficiently locate the timed metadata in the common format asset.

[0033] In the creation of target format timed metadata segments, ODE module 110 creates a segment containing the correct timed metadata in an efficient manner. Creating the timed metadata track efficiently requires extracting the timed metadata from the source asset into a separate index file by encapsulator 106. This file may contain a single track of timed metadata along with information such that the presentation time of the timed metadata is preserved. Enabling ODE module 110 to use this separate file, instead of the high bandwidth

source asset, preserves network bandwidth and increases the efficiency of the timed metadata track creation. Instead of processing a full common format asset stream, which can be on the order of megabits per second, ODE module 110 now only needs to process the timed metadata track, which may be only tens or hundreds of bits per second.

5           [0034] To ensure the correctness of the timed metadata delivered to client device 116, the timed metadata segment N created by ODE module 110 (produced non-linearly) should be equivalent to a timed metadata segment N produced linearly. ODE module 110 uses the timed metadata index file to construct the timed metadata segment. The timed metadata index file is augmented to include reference points generated by encapsulator 106 that impose a structure  
10           onto the timed metadata. In a particular embodiment, a reference point for metadata may be envisioned as being similar to an IDR frame for video. To present timed metadata after the reference point, no metadata prior to the reference point is necessary. Accordingly, the reference point contains information to provide all necessary state information to reconstruct the correct timed metadata state from that point in time forward. For example, for a reference  
15           point at a time  $T_r$  and all times  $T > T_r$ , creating the timed metadata track at all times  $T$  uses only the timed metadata at times  $\geq T_r$ .

          [0035] The construction of timed metadata reference points may be specific to the particular timed metadata format. For example, closed caption reference points can be created based on the location of pop-on captions within the video data. Additionally, caption reference  
20           points may be created by rendering a series of paint-on or roll-up caption sequences. In general, in various embodiments timed metadata reference points may be created by simulated rendering of the timed metadata by encapsulator 106 and persisting the current state of the timed metadata into the timed metadata index file at the desired time. A particular example case for this is timed metadata describing a game clock in a sporting event. Assume  
25           that the clock is provided in timed metadata in one second increments from the previously provided update. Joining this timed metadata track in the middle of the sporting event without the timed metadata indexing as described herein does not provide enough information to reconstruct the current time. A naive approach is to process all metadata since the beginning of the event. A more intelligent approach provided by various embodiments includes  
30           generating a metadata index that contains periodic timed metadata reference points that reflect the absolute time in the sporting event. By using the index of timed metadata, joining

the timed metadata track at a particular time provides enough information to provide the current timed metadata information to client device 116 despite only receiving the timed metadata since the last reference point.

[0036] Accordingly, rather than parsing an entire asset to retrieve timed metadata, a  
5 separate timed metadata index file is used to reproduce the timed metadata. This timed metadata index file includes reference points in the timed metadata which bound the creation of state based timed metadata ABR segments.

[0037] Various embodiments may provide one or more of the following advantages:  
increasing efficiency, requiring the use of less storage space than that required for multiple on-  
10 disk stored representations of target format metadata tracks, the ability of on-demand translation to support changes in target formats without reprocessing the source asset, and the use of a separate index file for timed metadata avoids unnecessary transformations and fetching of adaptive transport stream (ATS) data.

[0038] FIGURE 2 illustrates a simplified block diagram of an embodiment of  
15 encapsulator 106 of FIGURE 1. Encapsulator 106 includes processor(s) 202, memory element 204, input/output (I/O) interface(s) 206, index module 208, an encapsulation module 210. Processor(s) 202 is configured to execute various tasks of encapsulator 106 as described herein and memory element 204 is configured to store data associated with encapsulator 106. I/O interface(s) 206 is configured to receive communications from and send communications to  
20 other devices or software modules such as transcoder/encoder 104 and origin server 108. Indexing module 208 is configured to receive source video and/or source audio including timed metadata and create an index of the source video and/or source audio and timed metadata as further described herein. Encapsulator module 210 is configured to convert the received source video and/or audio and timed metadata into a common media format encapsulation as  
25 further described herein.

[0039] In one implementation, encapsulator 106 is a network element that includes software to achieve (or to foster) the encapsulation and indexing operations as outlined herein in this Specification. Note that in one example, each of these elements can have an internal structure (e.g., a processor, a memory element, etc.) to facilitate some of the operations  
30 described herein. In other embodiments, these encapsulation and indexing operations may be executed externally to this element, or included in some other network element to achieve this

intended functionality. Alternatively, encapsulator 106 may include software (or reciprocating software) that can coordinate with other network elements in order to achieve the operations, as outlined herein. In still other embodiments, one or several devices may include any suitable algorithms, hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof.

[0040] FIGURE 3 illustrates a simplified block diagram of an embodiment of origin server 108 and storage device 112 of FIGURE 1. Origin server 108 includes processor(s) 302, memory element 304, I/O interface(s) 306, and ODE module 110. As illustrated in FIGURE 3, origin server 108 is further in communication with storage device 112. Processor(s) 302 is configured to execute various tasks of origin server 108 as described herein and memory element 304 is configured to store data associated with origin server 108. I/O interface(s) 306 is configured to receive communications from and send communications to other devices or software modules such as encapsulator 106 and client device 116. ODE module 110 is configured to perform the various on-demand encapsulation operations as further described herein.

[0041] In one implementation, origin server 108 is a network element that includes software to achieve (or to foster) the server and on-demand encapsulation operations as outlined herein in this Specification. Note that in one example, each of these elements can have an internal structure (e.g., a processor, a memory element, etc.) to facilitate some of the operations described herein. In other embodiments, these server and on-demand encapsulation operations may be executed externally to this element, or included in some other network element to achieve this intended functionality. Alternatively, origin server 108 may include software (or reciprocating software) that can coordinate with other network elements in order to achieve the operations, as outlined herein. In still other embodiments, one or several devices may include any suitable algorithms, hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof.

[0042] FIGURE 4 is a simplified diagram of an example embodiment of common format media 400. Common format media 400 includes a number of segments each containing one or more of video data, audio data, and timed metadata. In at least one embodiment, common format media 400 is an ATS media stream. In the particular embodiment illustrated in FIGURE 4, common format media 400 includes a first segment 402a, a second segment 402b, and a third segment 402c. First segment 402a contains four video frames 404a including video

content in which the first video frame is an instantaneous decoder refresh (IDR) frame. First segment 402a further includes an audio frame 404a including audio content, and a timed metadata frame 406a including timed metadata content such as a closed captioning, ad-insertion or other timed metadata content.

5           [0043] Similarly, second segment 402b includes contains four video frames 404b including further video content in which the first video frame is an instantaneous decoder refresh (IDR) frame, an audio frame 404b including audio content, and a timed metadata frame 406b including timed metadata content such as a closed captioning, ad-insertion or other timed metadata content. Third segment 402c includes contains four video frames 404c including  
10 further video content in which the first video frame is an instantaneous decoder refresh (IDR) frame, an audio frame 404c including audio content, and a timed metadata frame 406c including timed metadata content such as a closed captioning, ad-insertion or other timed metadata content.

          [0044] In at least one embodiment, encapsulator 106 provides common format media  
15 400 to origin server 108, which stores common format media 400 within storage device 112.

          [0045] FIGURE 5 is a simplified diagram of an embodiment of a media data index file 500 as generated by encapsulator 106. In a particular embodiment, media data index 500 is generated by encapsulator 106 from the video and audio frames of common format media 400. Media data index file 500 includes a video data index 502 and an audio data index 504. Video  
20 data index 502 is illustrated as including a first video index segment  $V_{SIDX1}$ , a second video index segment  $V_{SIDX2}$ , and a third video index segment  $V_{SIDX3}$ . Each of first video index segment  $V_{SIDX1}$ , second video index segment  $V_{SIDX2}$ , and third video index segment  $V_{SIDX3}$  corresponds to a portion of common format media 400 and includes information which may be used by ODE module 110 to locate a requested segment of video data within common format media 400, retrieve the requested segment of video data, and format the requested segment of video data  
25 into a format suitable for delivery to client device 116. Audio data index 504 is illustrated in FIGURE 5 as including a first audio index segment  $A_{SIDX1}$ , a second audio index segment  $A_{SIDX2}$ , and a third audio index segment  $A_{SIDX3}$ . Each of first audio index segment  $A_{SIDX1}$ , second audio index segment  $A_{SIDX2}$ , and third audio index segment  $A_{SIDX3}$  corresponds to a portion of common  
30 format media 400 and includes information which may be used by ODE module 110 to locate a requested segment of audio data within common format media 400, retrieve the requested

segment of audio data, and format the requested segment of audio data into a format suitable for delivery to client device 116.

[0046] FIGURE 6 is a simplified diagram of an embodiment of a timed metadata index file 600 as generated by encapsulator 106. In a particular embodiment, timed metadata index file is generated from the timed metadata frames of common format media 400 such as time metadata frames 406a-406c. Timed metadata index file 600 includes a first timed metadata reference point  $TM_{REF1}$ , a second timed metadata reference point  $TM_{REF1}$ , and a third timed metadata reference point  $TM_{REF1}$ . Each of first timed metadata reference point  $TM_{REF1}$ , second timed metadata reference point  $TM_{REF1}$ , and third timed metadata reference point  $TM_{REF1}$  includes all state information necessary state for ODE module 110 to reconstruct the correct timed metadata state from that point forward within the timed metadata in response to a request for the timed metadata received from client device 116. By using the timed metadata index file 600, joining the timed metadata track by client device 116 provides enough information to provide the current timed metadata information to client device 116 despite only receiving the timed metadata information since the last reference point.

[0047] FIGURE 7 is a simplified flowchart 700 illustrating one potential operation of encapsulator 106 of FIGURE 1. In 702, encapsulator 106 receives common format media that includes timed metadata. In at least one embodiment, the media includes data such as one or more of video data or audio data associated with a media presentation or program. In a particular embodiment, encapsulator 106 receives the common format media from transcoder/encoder 104, and transcoder/encoder 104 receives the media content from media content source 102. In 704, encapsulator 106 generates portions of the common format asset including the Media Presentation Description and media data index files corresponding to non-metadata within the common format media. In 706, encapsulator 106 further generates the timed metadata index file corresponding to the common format media. The timed metadata index file includes one or more timed metadata reference points corresponding to the common format media. In a particular embodiment, each reference point includes state context information of the timed metadata at a particular instance of time. The timed metadata file is included in the common format asset.

[0048] In 708, encapsulator 106 sends the common format asset to a server. In a particular embodiment, encapsulator 106 sends the common format asset to origin server 108

and origin server 108 stores the common format asset in one or more storage devices such as storage device 112. The flow then ends. As further discussed herein, in one or more embodiments ODE module 110 of origin server 108 may receive a request for timed metadata within the common format asset and ODE module 110 may use the timed metadata index to  
5 determine how far back within the common format asset that it should go to retrieve a sufficient determined amount of the timed metadata necessary to produce the current timed metadata context at the current presentation time. For example, in a case in which the timed metadata is closed captioning data, the ODE module 110 may use the timed metadata index file to retrieve an amount of the caption data from the common format asset that is necessary to  
10 completely produce the current on-screen text for that instance in time and sends the caption data to client device 116.

[0049] FIGURE 8 is a simplified flowchart illustrating one potential operation of origin server 108. In 802, origin server 108 receives a request for timed metadata from client device 116. In a particular embodiment, the requested timed metadata may include closed  
15 captioning, subtitles, ad-insertions or any other timed metadata associated with media content. In 804, origin server 108 passes the request to ODE module 110. In 806, ODE module 110 retrieves the timed metadata index corresponding to the timed metadata within the common format asset from storage device 112. In 808, ODE module 110 further retrieves the relevant portions of the common format media including the requested timed metadata from storage  
20 device 112. In particular embodiments, the common format media may further include media data such as video or audio data associated with the timed metadata.

[0050] In 810, ODE module 110 generates timed metadata in a target format from the common format asset using the timed metadata index. In at least one embodiment, ODE module 100 uses the timed metadata index to retrieve a determined amount of the timed  
25 metadata from the common format asset sufficient to produce the current timed metadata context at the current presentation time. In at least one embodiment, the target format for the timed metadata is a format suitable for client device 116. In a particular embodiment, the target format may be, for example, an HLS format, an HSS format, an HDS format or a DASH format in accordance with the capabilities of client device 116.

30 [0051] In 812, origin server 108 sends a response message including the timed metadata in the target format to client device 116. The flow then ends. In one or more

embodiments, client device 116 may then present the timed metadata in association with media content such as video or audio content.

[0052] FIGURE 9 is a simplified diagram of an embodiment of a common format asset 900 as generated by the encapsulator of FIGURE 1. In the particular embodiment of FIGURE 9, common format asset 900 includes three common format media 400a-400c, three media data indexes 500a-500c, three timed metadata indexes 600a-600c, and a Media Presentation Description (MPD) 902. One or more of common format media 400a-400c may be formatted in the same or similar manner as common format media 400 as described with respect to FIGURE 4. One or more of media data indexes 500a-500c may be formatted in the same or similar manner as media data index 500 as described with respect to FIGURE 5. One or more of timed metadata indexes 600a-600c may be formatted in the same or similar manner as timed metadata index 600 as described with respect to FIGURE 6. MPD 902 is a manifest or file containing information about the common format media 400a-400c, such as one or more formats of segments of audio or video data that is used during presentation of the media content. Although the particular embodiment is illustrated as using three common format media 400a-400c, three media data indexes 500a-500c, three timed metadata indexes 600a-600c, and a single MPD 902, it should be understood that in other embodiments common format asset 900 may include any number of common format media, media data indexes, timed metadata indexes, and MPDs.

[0053] Communication network 100 represents a series of points or nodes of interconnected communication paths for receiving and transmitting packets of information that propagate through communication system 100. Communication network 100 offers a communicative interface between sources and/or hosts, and may be any local area network (LAN), wireless local area network (WLAN), metropolitan area network (MAN), Intranet, Extranet, WAN, virtual private network (VPN), or any other appropriate architecture or system that facilitates communications in a network environment. Communication network 100 may implement a UDP/IP connection and use a TCP/IP communication language protocol in particular embodiments of the present disclosure. However, communication network 100 may alternatively implement any other suitable communication protocol for transmitting and receiving data packets within communication system 100.

[0054] Transcoder/encoder 104, encapsulator 106, and origin server 108 are network elements that facilitate on-demand encapsulating of timed metadata in a given network (e.g., for networks such as that illustrated in FIGURES 1). As used herein in this Specification, the term 'network element' is meant to encompass routers, switches, gateways, bridges, loadbalancers, firewalls, inline service nodes, proxies, servers, processors, modules, or any other suitable device, component, element, proprietary appliance, or object operable to exchange information in a network environment. This network element may include any suitable hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information.

[0055] In one implementation, encapsulator 106 and origin server 108 include software to achieve (or to foster) the on-demand encapsulating of timed metadata operations, as outlined herein in this Specification. Note that in one example, each of these elements can have an internal structure (e.g., a processor, a memory element, etc.) to facilitate some of the operations described herein. In other embodiments, these on-demand encapsulating operations may be executed externally to these elements, or included in some other network element to achieve this intended functionality. Alternatively, encapsulator 106 and origin server 108 may include this software (or reciprocating software) that can coordinate with other network elements in order to achieve the operations, as outlined herein. In still other embodiments, one or several devices may include any suitable algorithms, hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof.

[0056] Note that in certain example implementations, the on-demand encapsulation functions outlined herein may be implemented by logic encoded in one or more non-transitory, tangible media (e.g., embedded logic provided in an application specific integrated circuit (ASIC), digital signal processor (DSP) instructions, software (potentially inclusive of object code and source code) to be executed by a processor, or other similar machine, etc.). In some of these instances, a memory element (as shown in FIGURE 2 and FIGURE 3) can store data used for the operations described herein. This includes the memory element being able to store software, logic, code, or processor instructions that are executed to carry out the activities described in this Specification. A processor can execute any type of instructions associated with the data to achieve the operations detailed herein in this Specification. In one example, the

processor (as shown in FIGURE 2 and/or FIGURE 3) could transform an element or an article (e.g., data) from one state or thing to another state or thing. In another example, the activities outlined herein may be implemented with fixed logic or programmable logic (e.g., software/computer instructions executed by a processor) and the elements identified herein could be some type of a programmable processor, programmable digital logic (e.g., a field programmable gate array [FPGA], an erasable programmable read only memory (EPROM), an electrically erasable programmable ROM (EEPROM)) or an ASIC that includes digital logic, software, code, electronic instructions, or any suitable combination thereof.

[0057] In one example implementation, encapsulator 106 and/or origin server 108 may include software in order to achieve the forwarding determination functions outlined herein. These activities can be facilitated by indexing module 208, encapsulation module 210, and/or ODE module 110 (where these modules can be suitably combined in any appropriate manner, which may be based on particular configuration and/or provisioning needs). Encapsulator 106 and origin server 108 can include memory elements for storing information to be used in achieving the on-demand encapsulation activities, as discussed herein. Additionally, encapsulator 106 and/or origin server 108 may include a processor that can execute software or an algorithm to perform the on-demand encapsulation operations, as disclosed in this Specification. These devices may further keep information in any suitable memory element (random access memory (RAM), ROM, EPROM, EEPROM, ASIC, etc.), software, hardware, or in any other suitable component, device, element, or object where appropriate and based on particular needs. Any of the memory items discussed herein (e.g., database, tables, trees, cache, etc.) should be construed as being encompassed within the broad term 'memory element.' Similarly, any of the potential processing elements, modules, and machines described in this Specification should be construed as being encompassed within the broad term 'processor.' Each of the network elements can also include suitable interfaces for receiving, transmitting, and/or otherwise communicating data or information in a network environment.

[0058] Note that with the example provided above, as well as numerous other examples provided herein, interaction may be described in terms of two, three, or four network elements. However, this has been done for purposes of clarity and example only. In certain cases, it may be easier to describe one or more of the functionalities of a given set of flows by

only referencing a limited number of network elements. It should be appreciated that communication system 100 (and its teachings) are readily scalable and can accommodate a large number of components, as well as more complicated/sophisticated arrangements and configurations. Accordingly, the examples provided should not limit the scope or inhibit the  
5 broad teachings of communication system 100 as potentially applied to a myriad of other architectures.

[0059] It is also important to note that the steps in the preceding flow diagrams illustrate only some of the possible signaling scenarios and patterns that may be executed by, or within, communication system 100. Some of these steps may be deleted or removed where  
10 appropriate, or these steps may be modified or changed considerably without departing from the scope of the present disclosure. In addition, a number of these operations have been described as being executed concurrently with, or in parallel to, one or more additional operations. However, the timing of these operations may be altered considerably. The preceding operational flows have been offered for purposes of example and discussion.  
15 Substantial flexibility is provided by communication system 100 in that any suitable arrangements, chronologies, configurations, and timing mechanisms may be provided without departing from the teachings of the present disclosure.

[0060] It should also be noted that many of the previous discussions may imply a single client-server relationship. In reality, there is a multitude of servers and clients in certain  
20 implementations of the present disclosure. Moreover, the present disclosure can readily be extended to apply to intervening servers further upstream in the architecture. Any such permutations, scaling, and configurations are clearly within the broad scope of the present disclosure.

[0061] Although the present disclosure has been described in detail with reference to  
25 particular arrangements and configurations, these example configurations and arrangements may be changed significantly without departing from the scope of the present disclosure. Additionally, although communication system 100 has been illustrated with reference to particular elements and operations that facilitate the communication process, these elements and operations may be replaced by any suitable architecture or process that achieves the  
30 intended functionality of communication system 100.

[0062] Numerous other changes, substitutions, variations, alterations, and modifications may be ascertained to one skilled in the art and it is intended that the present disclosure encompass all such changes, substitutions, variations, alterations, and modifications as falling within the scope of the appended claims. In order to assist the United States Patent and Trademark Office (USPTO) and, additionally, any readers of any patent issued on this application in interpreting the claims appended hereto, Applicant wishes to note that the Applicant: (a) does not intend any of the appended claims to invoke paragraph six (6) of 35 U.S.C. section 112 as it exists on the date of the filing hereof unless the words "means for" or "step for" are specifically used in the particular claims; and (b) does not intend, by any statement in the specification, to limit this disclosure in any way that is not otherwise reflected in the appended claims.

WHAT IS CLAIMED IS:

1. A method, comprising:  
receiving common format media including timed metadata; and  
generating a common format asset including the timed metadata and a timed metadata  
5 index including at least one reference point corresponding to the common format media, each  
reference point including state context information of the timed metadata at a particular  
instance of time.
2. The method of Claim 1, further comprising sending the common format asset to  
10 at least one server.
3. The method of Claim 1, further comprising:  
receiving a request for the timed metadata from a particular client device; and  
retrieving a determined amount of the timed metadata from the common format asset  
15 using the timed metadata index.
4. The method of Claim 3, wherein the determined amount of the timed metadata  
is an amount sufficient to display a current timed metadata context corresponding to the  
request.  
20
5. The method of Claim 3, further comprising:  
converting the retrieved timed metadata to a target format suitable for the particular  
client device; and  
sending a response message including the retrieved timed metadata in the target  
25 format to the particular client device.
6. The method of Claim 1, wherein the common format media is an MPEG2-TS  
adaptive transport stream file.
7. The method of Claim 1, wherein the common format media is an ISO Base Media  
File Format (ISO-BMFF) file.  
30

8. The method of Claim 1, wherein the timed metadata includes at least one of caption data, subtitle data, ad-insertion marker data and application-specific metadata.

9. One or more non-transitory tangible media that includes code for execution and  
5 when executed by a processor operable to perform operations comprising:  
receiving common format media including timed metadata;  
generating a common format asset including the timed metadata and a timed metadata  
index including at least one reference point corresponding to the common format media, each  
reference point including state context information of the timed metadata at a particular  
10 instance of time.

10. The media of Claim 9, wherein the operations further include sending the common format asset to at least one server.

11. The media of Claim 9, wherein the operations further include:  
15 receiving a request for the timed metadata from a particular client device; and  
retrieving a determined amount of the timed metadata from the common format asset  
using the timed metadata index.

12. The media of Claim 11, wherein the determined amount of the timed metadata  
20 is an amount sufficient to display a current timed metadata context corresponding to the request.

13. The media of Claim 11, wherein the operations further include:  
25 converting the retrieved timed metadata to a target format suitable for the particular client device; and  
sending a response message including the retrieved timed metadata in the target format to the particular client device.

14. The media of Claim 9, wherein the common format media is an MPEG2-TS  
30 adaptive transport stream file.

15. The media of Claim 9, wherein the common format media an ISO Base Media File Format (ISO-BMFF) file.

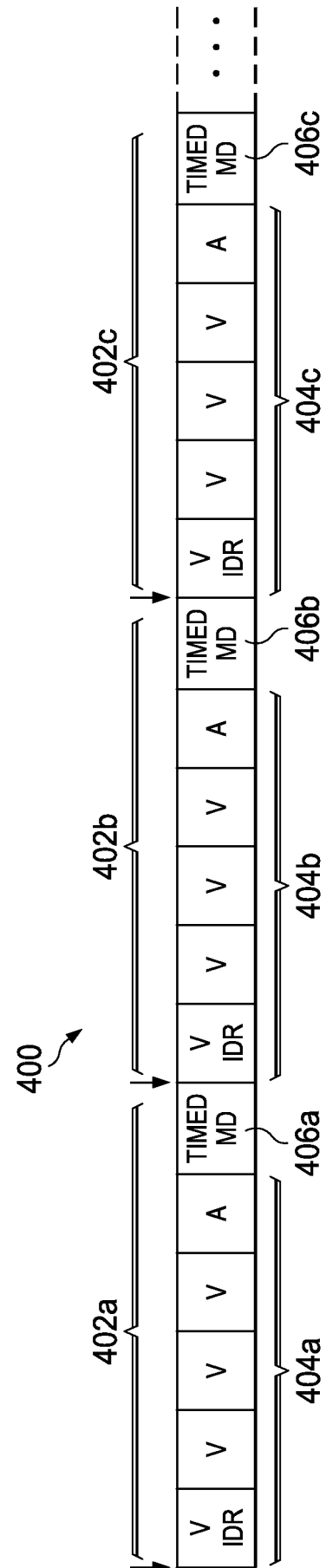
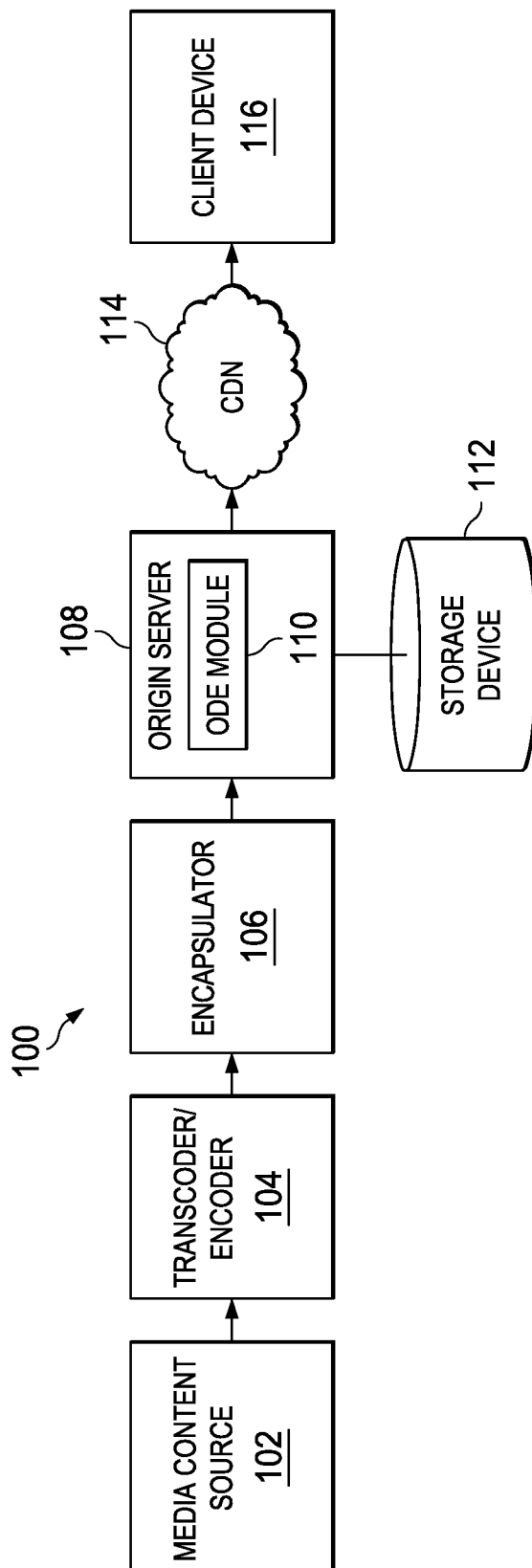
16. The media of Claim 9, wherein the timed metadata includes at least one of  
5 caption data, subtitle data, ad-insertion marker data and application-specific metadata.

17. An apparatus, comprising:  
a memory element configured to store data,  
a processor operable to execute instructions associated with the data, and  
10 at least one module being configured to:  
receive common format media including timed metadata; and  
generate a common format asset including the timed metadata and a  
timed metadata index including at least one reference point corresponding to the common  
format media, each reference point including state context information of the timed metadata  
15 at a particular instance of time.

18. The apparatus of Claim 17, wherein the at least one module is further configured  
to send the common format asset to at least one server.

19. The apparatus of Claim 17, wherein the at least one module is further configured  
20 to:  
receive a request for the timed metadata from a particular client device; and  
retrieve a determined amount of the timed metadata from the common format asset  
using the timed metadata index.

20. The apparatus of Claim 19, wherein the determined amount of the timed  
25 metadata is an amount sufficient to display a current timed metadata context corresponding to  
the request.



2/5

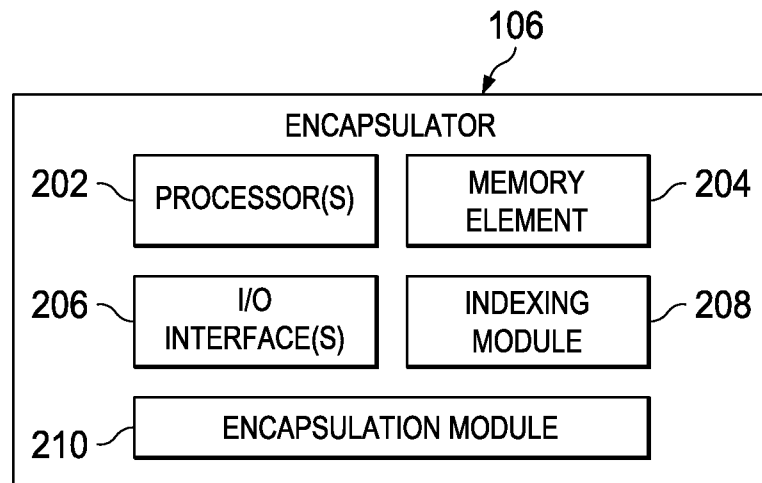


FIG. 2

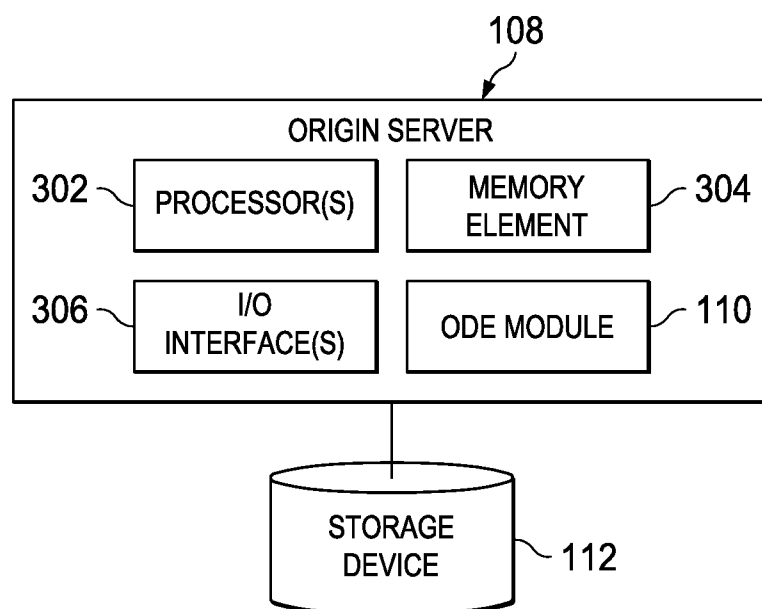


FIG. 3

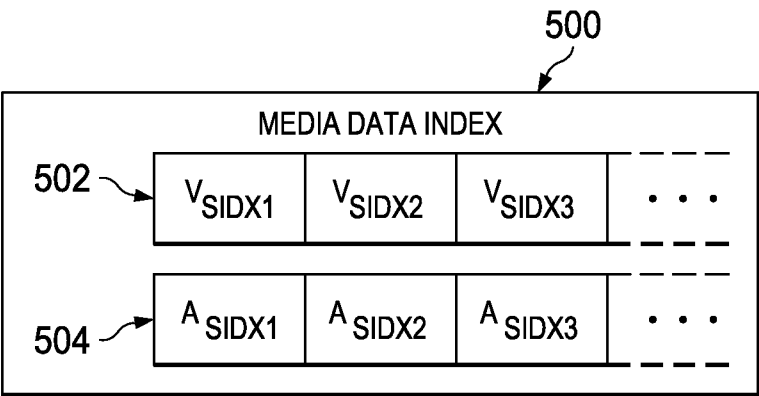


FIG. 5

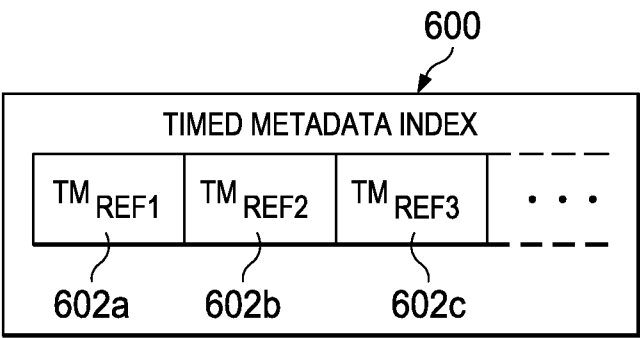


FIG. 6

4/5

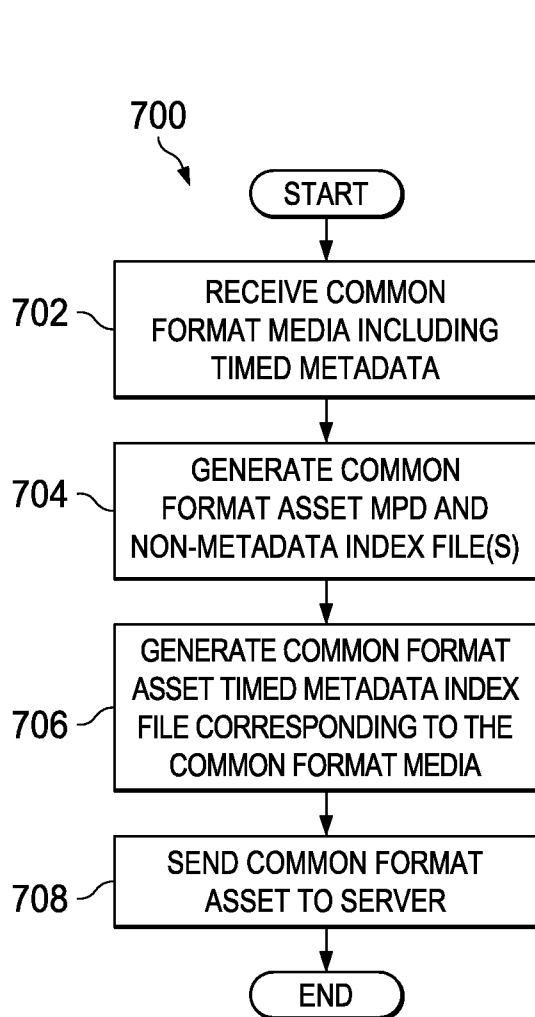


FIG. 7

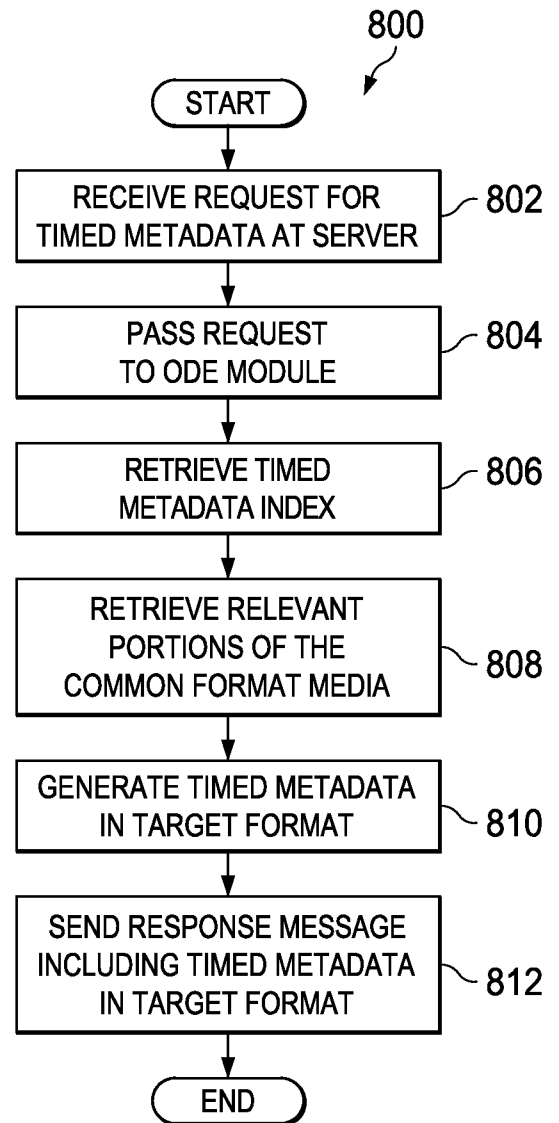


FIG. 8

5/5

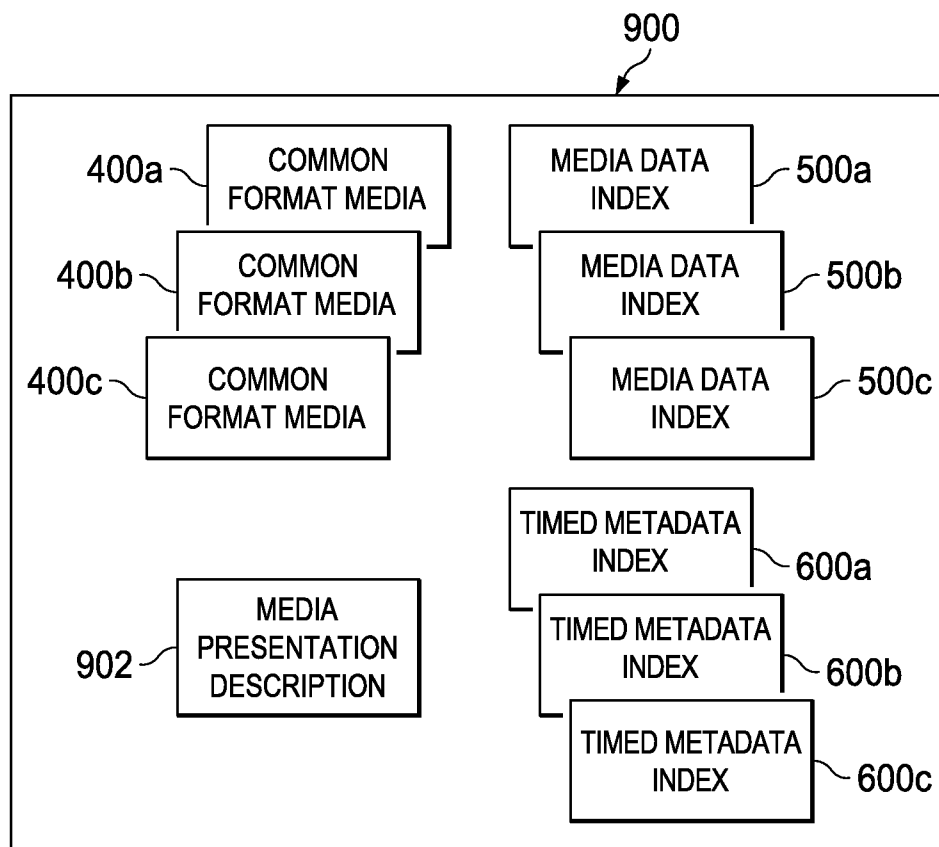


FIG. 9

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/US2014/037732

## A. CLASSIFICATION OF SUBJECT MATTER

INV. H04N21/235 H04N21/4722 H04N21/84 H04N21/8547  
ADD.

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

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

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

EPO-Internal, WPI Data

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                  | Relevant to claim No. |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| X         | WO 2009/024926 A1 (NOKIA CORP [FI]; NOKIA INC [US]; HANNUKSELA MISHA [FI])<br>26 February 2009 (2009-02-26)<br>abstract<br>paragraph [0006] - paragraph [0015]<br>claims<br>figures | 1-20                  |
| X         | US 7 913 277 B1 (RAHRER TIM [CA])<br>22 March 2011 (2011-03-22)<br>abstract<br>column 1, line 23 - page 2, paragraph 40<br>claims<br>figures                                        | 1-20                  |
|           | -----<br>-/--                                                                                                                                                                       |                       |



Further documents are listed in the continuation of Box C.



See patent family annex.

## \* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

22 July 2014

Date of mailing of the international search report

29/07/2014

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040,  
Fax: (+31-70) 340-3016

Authorized officer

Seibert, Joachim

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/US2014/037732

| C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT |                                                                                                                                                                                                                                                   |                       |
|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Category*                                            | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                                                | Relevant to claim No. |
| X                                                    | WO 2013/033565 A1 (QUALCOMM INC [US]; LUBY<br>MICHAEL GEORGE [US]; MINDER LORENZ<br>CHRISTOPH []) 7 March 2013 (2013-03-07)<br>paragraph [0121]<br>paragraph [0365] - paragraph [0366]<br>paragraph [0529] - paragraph [0533]<br>figures<br>----- | 1-20                  |
| X                                                    | CN 103 081 506 A (ELECTRONICA AND TELECOMM<br>RES INST) 1 May 2013 (2013-05-01)<br>abstract<br>& US 2013/159546 A1 (THANG TRUONG CONG<br>[KR] ET AL) 20 June 2013 (2013-06-20)<br>paragraph [0048] - paragraph [0056]<br>figures<br>-----         | 1-20                  |

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2014/037732

| Patent document<br>cited in search report | Publication<br>date | Patent family<br>member(s) | Publication<br>date         |
|-------------------------------------------|---------------------|----------------------------|-----------------------------|
| WO 2009024926                             | A1                  | 26-02-2009                 | CA 2695645 A1 26-02-2009    |
|                                           |                     |                            | CN 101802823 A 11-08-2010   |
|                                           |                     |                            | EP 2191402 A1 02-06-2010    |
|                                           |                     |                            | KR 20100043108 A 27-04-2010 |
|                                           |                     |                            | RU 2010111391 A 27-09-2011  |
|                                           |                     |                            | US 2009055417 A1 26-02-2009 |
|                                           |                     |                            | WO 2009024926 A1 26-02-2009 |
| -----                                     |                     |                            |                             |
| US 7913277                                | B1                  | 22-03-2011                 | NONE                        |
| -----                                     |                     |                            |                             |
| WO 2013033565                             | A1                  | 07-03-2013                 | CN 103858440 A 11-06-2014   |
|                                           |                     |                            | EP 2752025 A1 09-07-2014    |
|                                           |                     |                            | KR 20140054400 A 08-05-2014 |
|                                           |                     |                            | US 2013246643 A1 19-09-2013 |
|                                           |                     |                            | WO 2013033565 A1 07-03-2013 |
| -----                                     |                     |                            |                             |
| CN 103081506                              | A                   | 01-05-2013                 | CN 103081506 A 01-05-2013   |
|                                           |                     |                            | EP 2613543 A2 10-07-2013    |
|                                           |                     |                            | KR 20120035175 A 13-04-2012 |
|                                           |                     |                            | KR 20120076294 A 09-07-2012 |
|                                           |                     |                            | KR 20140038492 A 28-03-2014 |
|                                           |                     |                            | US 2013159546 A1 20-06-2013 |
| -----                                     |                     |                            |                             |