



(51) International Patent Classification:

H04N 21/2381 (2011.01) *H04N 21/643* (2011.01)
H04N 21/242 (201.01) *H04N 21/8547* (201.01)
H04N 21/438 (201.01) *H04L 12/18* (2006.01)

(21) International Application Number:

PCT/US20 13/041525

(22) International Filing Date:

17 May 2013 (17.05.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/649,115 18 May 2012 (18.05.2012) US
13/896,358 17 May 2013 (17.05.2013) US

(71) Applicant: **MOTOROLA MOBILITY LLC** [US/US];
600 North US Highway 45, Libertyville, Illinois 60048 (US).

(72) Inventors: **GROSSMAN, Michael A.**; 5283 Mt. Burnham Drive, San Diego, California 92111 (US). **MOORTHY, Praveen N**; 17324 Eagle Canyon Way, San Diego, California 92127 (US).

(74) Agents: **BRETSCHER, John, T.** et al; 600 North US Highway 45, Libertyville, Illinois 60048 (US).

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

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available):

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

Published:

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

(54) Title: SYNCHRONIZING MULTIPLE TRANSCODING DEVICES UTILIZING SIMULTANEITY OF RECEIPT OF MULTICAST PACKETS

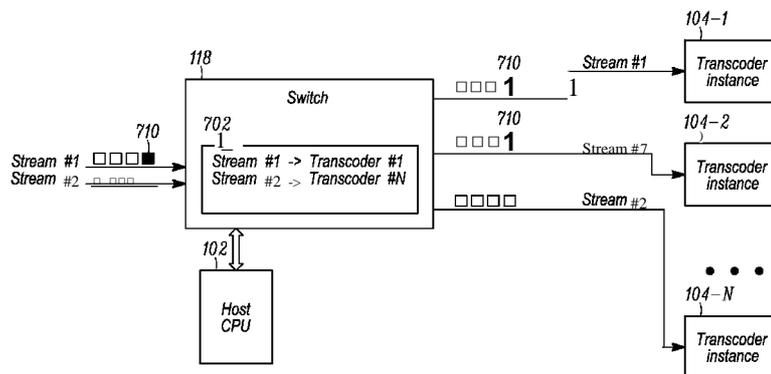


FIG. 7

(57) Abstract: Disclosed is a method for synchronizing video in a system including a host CPU and a plurality of transcoder instances, the method comprising: receiving, by two or more transcoder instances in the plurality of transcoder instances, a synchronization Internet Protocol ("IP") packet from the host CPU, wherein the synchronization IP packet is received as one packet in a multicast stream of IP packets; detecting, by the two or more transcoder instances in the plurality of transcoder instances, the synchronization IP packet at the same point in the multicast stream; and transcoding, by the two or more transcoder instances in the plurality of transcoder instances, the multicast stream of IP packets immediately after detecting the synchronization IP packet.

WO 2013/173683 A1

SYNCHRONIZING MULTIPLE TRANSCODING DEVICES UTILIZING SIMULTANEITY OF RECEIPT OF MULTICAST PACKETS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims priority to U.S. Provisional Patent Application 61/649,115, filed on May 18, 2012, which is incorporated by reference herein in its entirety.

TECHNICAL FIELD

[0002] The present disclosure is related generally to video encoding and, more particularly, to synchronized encoding.

BACKGROUND

[0003] Transcoder products have typically been limited to one or two channels. However, system-on-a-chip ("SOC") technologies allow a transcoder product to integrate a larger number of transcoder channels into a single chassis via multiple transcoder SOC instances. The transcoder SOC instances may perform transcoding in parallel. A host central processor unit ("CPU") communicates with each of the transcoder SOC instances to coordinate the transcoding. When using multiple transcoder instances, unless the transcoders are initialized or synchronized to begin transcoding at the same time, there may be a time delay or gap in information (e.g., misaligned segments) from one transcoder to another. This may cause problems for streaming video, such as when changing from one resolution or compression rate to another if the segments from one compression rate do not align with the segments from another compression rate. This not-so-seamless transition is undesirable for viewers and should be minimized or eliminated.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0004] While the appended claims set forth the features of the present techniques with particularity, these techniques, together with their objects and advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

- [0005] Figure 1 depicts a system for transcoding video according to some embodiments;
- [0006] Figure 2 depicts an example of routing IP packets according to some embodiments;
- [0007] Figure 3 depicts an example of processing IP packets within a transcoder instance according to some embodiments;
- [0008] Figure 4 depicts an example of processing IP packets that include control information according to some embodiments;
- [0009] Figure 5 shows an example of joining a multicast stream according to some embodiments;
- [0010] Figure 6 shows an example of a process for ingesting, transcoding, segmenting, and adaptive streaming in accordance with some embodiments;
- [0011] Figure 7 shows an example of routing IP packets using a multicast architecture according to some embodiments; and
- [0012] Figure 8 depicts a simplified flowchart of a method for processing IP packets in a transcoder instance according to some embodiments.

DETAILED DESCRIPTION

[0013] Turning to the drawings, wherein like reference numerals refer to like elements, techniques of the present disclosure are illustrated as being implemented in a suitable environment. The following description is based on embodiments of the claims and should not be taken as limiting the claims with regard to alternative embodiments that are not explicitly described herein.

[0014] In a first aspect, a method for synchronizing video in a system including a host CPU and a plurality of transcoder instances is disclosed, the method comprising:

receiving, by two or more transcoder instances in the plurality of transcoder instances, a synchronization Internet Protocol ("IP") packet from the host CPU, wherein the synchronization IP packet is received as one packet in a multicast stream of IP packets; detecting, by the two or more transcoder instances in the plurality of transcoder instances, the synchronization IP packet at the same point in the multicast stream; and transcoding, by the two or more transcoder instances in the plurality of transcoder instances, the multicast stream of IP packets immediately after detecting the synchronization IP packet.

[0015] In a second aspect, an apparatus configured to synchronize video in a system including a host CPU and a plurality of transcoder instances is disclosed, the apparatus comprising: one or more computer processors; and a non-transitory computer-readable storage medium containing instructions that, when executed, control the one or more computer processors to be configured for: receiving, by two or more transcoder instances in the plurality of transcoder instances, a synchronization IP packet from the host CPU, wherein the synchronization IP packet is received as one packet in a multicast stream of IP packets; detecting, by the two or more transcoder instances in the plurality of transcoder instances, the synchronization IP packet at the same point in the multicast stream; and transcoding, by the two or more transcoder instances in the plurality of transcoder instances, the multicast stream of IP packets immediately after detecting the synchronization IP packet.

[0016] Figure 1 depicts a system 100 for transcoding video according to some embodiments. In some embodiments, system 100 is integrated into a single transcoding device, such as a single chassis device (e.g., a 1RU rack unit). Additionally, system 100 may include multiple rack units that communicate together via an IP network backplane to transcode video.

[0017] A host CPU 102 is configured to communicate with multiple transcoder instances 104-1 through 104-N. Host CPU 102 coordinates transcoding operations to be performed on transcoder instances 104-1 through 104-N. Although one host CPU 102 is shown, system 100 may include multiple host CPUs.

[0018] Transcoder instances 104 may be SOC entities. Transcoder instances 104 may include a transcoder CPU 106 in addition to other entities, such as a network interface 108, an audio/video ("A/V") processor 110, memory 112, and disk storage 114. Particular embodiments allow host CPU 102 to offload functions normally performed by host CPU 102 to transcoder instances 104 because each transcoder instance 104 includes a transcoder CPU 106 that is capable of performing at least some functions normally performed by host CPU 102. Transcoder CPUs 106 may perform the functions in parallel within transcoder instances 104. Each transcoder CPU 106 may be responsible for performing tasks within a respective transcoder instance 104 or for other transcoder instances.

[0019] In some embodiments, host CPU 102 uses a routing system, such as router 116 and switch 118, to communicate with transcoder instances 104. In some embodiments, the use of router 116 and switch 118 (or other comparable routing systems) allows host CPU 102 to communicate using IP packets. For example, each transcoder instance 104 may include a network interface 108 that can receive IP packets. This allows communication between host CPU 102 and transcoder instances 104 via a universal networking standard, such as Ethernet. In such cases, host CPU 102 does not need to send control signals via a PCI bus or communicate video via an MPEG transport stream. Rather, the control signals and video data may be encapsulated in IP packets and routed via router 116 and switch 118 to transcoder instances 104.

[0020] By sending video data in IP packets, FPGA logic is not needed to process or send video data in a video transport stream outside of transcoder instances 104. The video transport stream may use a standard for transcoding video, such as an MPEG. Host CPU 102 may offload the processing of the video transport stream to transcoder instances 104. For example, video data are encapsulated in the IP packets. Transcoder CPU 106 receives the IP packets through network interface 108. Because A/V processor 110 needs to perform transcoding operations on a video transport stream, transcoder instance 104 needs to convert the video in the payload of the IP packets to the video transport stream.

Network interface 108 can de-capsulate the received IP packets to retrieve the video data from the payload of the IP packets. Transcoder CPU 106 may then coordinate generation of a video transport stream. Once in this format, transcoder instance 104 can perform various transcoding services on the video transport stream. For example, A/V processor 110 may transcode (e.g., decode and encode) the video transport stream.

[0021] Host CPU 102 may also send control commands in IP packets. Transcoder CPU 106 may then de-capsulate the IP packets to determine the control signals, which are then processed by transcoder CPU 106. Not only are the video data and control signals sent via the same channel and network interface, expensive PCI silicon and FPGA logic are not needed outside of the transcoder instances 104.

[0022] Figure 2 depicts an example of routing IP packets according to some embodiments. Router 116 receives packets from the network infrastructure. For example, the packets include video that needs to be transcoded. Host CPU 102 coordinates the routing of packets to transcoder instances 104. For example, as is discussed in more detail below, a multicast configuration may be used to route packets to specific transcoder instances 104.

[0023] In some examples, router 116 is routing packets #1 through N. In some examples, router 116 uses a routing table 202 to determine where to route the packets. For example, the IP packets are routed depending on what stream the packets are associated with. As is described below, transcoder instances 104 subscribe to various streams. For simplicity, the routing is described as routing different packets to different transcoder instances 104. For example, router 116 routes an IP packet #1 to transcoder instance 104-1, routes IP packets #2 and #3 to transcoder instance 104-2, and routes IP packets #4 through #N to transcoder instance 104-N. Switch 118 sends the IP packets to transcoder instances 104.

[0024] In some embodiments, router 116 queries the network to determine which devices on a particular network port have joined the IP multicast. Router 116 does not

forward the multicast IP traffic to devices on the network that have not joined. Switch 118, connected to router 116, in turn may utilize Internet Group Management Protocol ("IGMP") snooping to determine which instances connected to switch ports have joined the IP multicast. Switch 118 then sets up Ethernet port filtering to inhibit the flow of IP packets to instances that have not joined. For instances that have joined, switch 118 forwards the identified packets from the correct queue.

[0025] Routing video within IP packets is different from sending a video transport stream to transcoder instances 104. That is, router 116 and switch 118 route IP packets using an IP protocol, such as Ethernet. When sending video in a video transport stream, the video data are sent serially. Serial data are sent 1 bit at a time over an interface. Thus, FPGA logic is needed to send the video. However, encapsulating the video in IP packets does not require FPGA logic to send a serial stream of data. Rather, multiple bits of data are sent in parallel.

[0026] Once receiving the IP packets, transcoder instances 104 need to process the IP packets to retrieve the information included within the payload of the IP packets. For example, the control information and video may be included in the payload of an IP packet. Figure 3 depicts an example of processing IP packets 302 within a transcoder instance 104 according to one embodiment. IP packets 302 are in the format governed by a universal networking standard, such as Ethernet. In general, each IP packet 302 includes a header 304 and a payload 306. Header 304 may include supplemental data that describe the data being transmitted and how the IP packet should be routed. For example, header 304 includes a destination address. Payload 306 includes the video being transmitted. As is known, the IP packet 302 encapsulates the video in payload 306. Transcoder instance 104 needs an interface to receive IP packet 302. Because IP is a universal standard, the interface that is used is a standard IP interface, such as an IP receiver interface 308.

[0027] To process the video within payload 306, IP receiver interface 308 needs to convert the video into a video transport stream, such as an MPEG transport stream, that is

in a format defined by the standard being used to encode or decode the video. IP receiver interface 308 may convert the data into the video transport stream as shown at 310. In some embodiments, this converts the video in payload 306 from a parallel format to a serial format. Internal to the transcoder SOC, the parallel MPEG transport stream payload may be processed by a hardware block that performs the serialization of the transport stream prior to video decoding. The video transport stream is the format in which transcoding can be performed.

[0028] A/V processor 110 receives the transport stream and performs a transcoding operation on the stream. For example, A/V processor 110 encodes or decodes the transport stream. Other operations may also be performed, such as packaging. The different types of services are described in more detail below.

[0029] After processing, A/V processor 110 outputs the processed video transport stream from transcoder instance 104 at 311. Before the stream can be transmitted from transcoder instance 104, transcoder instance 104 needs to encapsulate the video that is in the video transport stream format to IP packets again. This is because system 100 processes IP packets and not a video transport stream outside of transcoder instances 104. In such cases, an IP transmitter interface 312 encapsulates the processed video transport stream into one or more IP packets 314. This converts the data from a serial format to a parallel format. IP transmitter interface 312 then transmits IP packets 314 through switch 118 to router 116. Host CPU 102 then handles routing of IP packets 314. For example, host CPU 102 may route IP packets 314 to another transcoder instance 104 or to a destination, such as a decoder.

[0030] In addition to routing video through IP packets, system 100 may send control information through IP packets. The control information provides information to transcoder instances 104 that can be used when performing transcoding operations or services. For example, the control information may synchronize the transcoding being performed among multiple transcoder instances 104. Figure 4 depicts an example of processing IP packets that include control information according to some embodiments.

Router 116 receives an IP packet 402 with control information in a payload 404. IP packet 402 also includes a header 401 that identifies the payload as including control information. Router 116 routes IP packet 402 via switch 118 to a transcoder instance 104. The control information is thus sent through the same channel as the video data.

[0031] Transcoder instance 104 may de-capsulate IP packet 402 to determine the control information from payload 404. For example, IP receiver interface 308 receives IP packet 402 and decapsulates the payload to output the control information shown at 405. Transcoder CPU 106 interprets the control information when performing services. For example, transcoder CPU 106 uses the control information to control A/V processor 110 when performing services such as a transcoding service 406-1, a synchronization (sync) service 406-2, a packaging service 406-3, or other services 406-N. Transcoding service 406-1 performs encoding or decoding of the video data. Synchronization service 406-2 synchronize the transcoding among multiple transcoding instances 104. Packaging service 406-3 cuts video and audio streams into files for delivery to Internet-connected devices, such as smartphones and tablets. Packaging may also need synchronization among different transcoder instances 104. Thus, sending control information via the same channel as the video allows the synchronization to be performed; whereas, when control information was sent through a different channel using a PCI bus, it was hard to synchronize the data that were received with the control information.

[0032] The routing of video to various transcoder instances 104 is simplified because IP packets can be routed to transcoder instances 104. For example, router 116 uses a multicast system to determine where to route IP packets. Figure 5 shows an example of joining a multicast stream according to one embodiment. When transcoder instances 104 want to process video through a stream, each transcoder instance 104 sends a "join stream" command to host CPU 102. For example, host CPU 102 receives join commands for different streams from different transcoder instances 104. In a specific example, transcoder instance 104-1 sends a join for stream #1, transcoder instance 104-2 sends a join for stream #2, and a transcoder instance 104-N sends a join for stream #N.

[0033] Based on receiving the join commands, host CPU 102 populates routing table 202 with information on how to route IP packets for various streams. For example, host CPU 102 adds an identifier to a stream in routing table 202 for transcoder instances that join certain streams. For stream #1, host CPU 102 adds the identifier for transcoder instance 104-1. For stream #2, host CPU 102 adds the identifier for transcoder instance 104-2. For stream #N, host CPU 102 adds the identifier for transcoder instance 104-N. Once routing table 202 is populated with which transcoder instances 104 joined which streams, router 116 leverages routing table 202 to route IP packets for the streams.

[0034] Recently, Hypertext Transfer Protocol ("HTTP") based live streaming protocols have been widely adopted to download or stream audio video ("AV") content to portable devices. For example, referring now to Figure 6, content prepared by or delivered from an HTTP server (not shown) is classified as HTTP adaptive streaming. Adaptive streaming operates by dynamically adjusting the play-out rate to stay within the actual network throughput to a given endpoint without the need for rebuffering. So, if the network throughput suddenly drops, the picture may degrade, but the end user still sees a picture.

[0035] As used herein, HTTP adaptive streaming is the generic term for various implementations: Apple HTTP Live Streaming, Microsoft IIS Smooth Streaming, Adobe HTTP Dynamic Streaming, and MPEG DASH.

[0036] Although each of the various implementations of HTTP adaptive streaming is different, they all share a set of common properties. For example, still referring to Figure 6, source content 610 is transcoded in parallel at multiple bit rates (e.g., multi-rate coding) in a transcoding process 620. Each bit rate is called a profile or a representation. As shown, the source content 610 may comprise media content such as live source content or file source content. For example, the file source content may include movies, TV programs, etc. The live source content includes live streaming format, such as a live broadcast of a sports program or game.

[0037] Encoded content is divided into fixed duration segments (e.g., chunks) in a segmentation process 630. The segments or chunks are typically between two and 10 seconds in duration, although they may be longer or shorter. In some embodiments, shorter segments reduce coding efficiency while larger segments impact speed to adapt to changes in network throughput.

[0038] A manifest file is created and updated as necessary to describe the encoding rates and URL pointers to segments in a manifest file creation process 640. As used herein, a manifest file or playlist describes how content is prepared in how many different encoding bitrates and, for each bitrate stream, how long a segment is and where to obtain each segment of each bitrate stream.

[0039] In some embodiments, the client uses HTTP to fetch segments from the network, buffer them, and then decode and play them. The client may utilize one or more algorithms designed to select the optimum profile so as to maximize quality without risking buffer underflow and stalling (e.g., rebuffering) of the play-out. For example, each time the client fetches a segment, it may choose the profile based on the measured time to download the previous segment.

[0040] In some embodiments, the HTTP server could publish a variant playlist file with multiple entries for the same AV content, each of these entries representing a stream with unique encoding parameters. The client initially downloads the variant playlist file and, depending upon the network conditions, downloads the media files with the most appropriate encoding parameters (e.g., using one or more algorithms described above).

[0041] Generating the multitude of streams each with unique encoding parameters commonly requires a multi-transcoder architecture. Synchronization across multiple transcoders becomes a challenge. Depending on network conditions, the client requests different bitrate streams from the initially downloaded variant playlist file. Subsequently, during bitrate switching, the client could request the next sequence number media file at a different bitrate with the expectation that the audio and video will play smoothly, without

any discontinuities. This requires that each of the segments will be perfectly aligned even though they might have been generated by a separate transcoder device.

[0042] Commonly, multicast is used as the IP protocol to alleviate the need to send separate streams to each transcoder. To avoid misalignment of the transcoded streams, a mechanism is needed to synchronize the transcoder devices so that they all start transcoding at the same point, ensuring that all segments are aligned.

[0043] Figure 7 shows an example of routing IP packets using a multicast architecture according to one embodiment. Switch 118 receives streams #1 and #2 (although there may be up to #N streams). When switch 118 receives the streams, switch 118 checks which transcoder instances 104 have joined each stream using switching table 702. In this case, switch 118 sends stream #1 to transcoder instance 104-1 and to transcoder instance 104-2 and sends stream #2 to transcoder instance 104-N. Unlike sending a video transport stream, the above switching scheme can be performed without using a large amount of host CPU resources. This is because host CPU 102 offloads the processing of the video transport stream to transcoder instances 104. This frees up host CPU 102, and host CPU 102 does not become a bottleneck in system 100.

[0044] In Figure 7, the multiple transcoder instances 104-1, 104-2, 104-N may be synchronized by using, e.g., a simultaneity of multicast protocol for both AV content and control information. In some embodiments, switch 118 comprises a managed switch that may be configured to perform IGMP snooping and to filter out multicast packets that have not joined the multicast. Generally IGMP snooping is the process of listening to IGMP network traffic. The feature allows switch 118 to listen in on the IGMP conversation between, e.g., hosts and routers. By listening to these conversations, the switch 118 maintains a map of which links need which IP multicast streams. Multicasts may be filtered from the links which do not need them, and thus the switch 118 controls which ports receive specific multicast traffic.

[0045] In some embodiments, transcoder instances 104-1, 104-2, 104-N are interconnected via switch 118. Switch 118 may be configured to send multicast traffic to each transcoder instance separately. For example, if a variant playlist file has 12 unique entries of encoded streams, and each transcoder instance is capable of creating four transcoded streams, a cluster of three transcoder instances will be required as a minimum.

[0046] In some embodiments, host CPU 102 is configured to allocate the resources and to configure the transcoder instances to join the multicast. Once all the transcoder instances have joined, the multicast stream may be simultaneously sent to each transcoder instance. In some embodiments, each transcoder instance will not start transcoding until host CPU 102 sends a synchronization multicast packet 710. Synchronization multicast packet 710 may be sent on the same multicast IP address inserted such that it may be detected at the same point in the stream at each transcoder instance (e.g., when there is no packet re-ordering). In some embodiments, the synchronization multicast packet is a MPEG null packet with a distinctive code word to identify it. A purpose of the synchronization packet is to simultaneously demarcate the transcoding start point to all transcoder instances.

[0047] In some embodiments, switch 118 uses pointers to send out data. For example, switch 118 may use a pointer mechanism to directly copy the packet data from the input buffer to multiple switch output port destinations without making separate copies of the data, ensuring that the packets are inserted into the multicast stream at exactly the same point. Thus, switch 118 may be predicated on using pointers instead of moving actual data. This ensures that data are not actually copied or buffered, making it arbitrary where synchronization multicast packet 710 is inserted into the stream.

[0048] Additionally, in some embodiments, host CPU 102 may use a timer or clock to determine or dictate an initialization time for synchronization multicast packet 710. In some embodiments, synchronization multicast packet 710 is received by transcoder instances at the same time. This may ensure that each transcoder instance begins

transcoding at the same time (e.g., the same point in the AV content), thus creating aligned segments across the transcoder instances.

[0049] For example, still referring to Figure 7, host CPU 102 may be configured to insert one or more synchronization multicast packets 710 into each multicast stream. As shown, a single synchronization multicast packet 710 is inserted into Stream #1. Host CPU 102 may insert the synchronization multicast packet 710 into one or more multicast streams prior to the streams being received by switch 118. For example, host CPU 102 may transmit the synchronization multicast packet 710 on the same IP address and port number. Alternatively, the host CPU 102 may communicate with switch 118 to have switch 118 insert the synchronization multicast packets 710 into the correct streams (e.g., if switch 118 has an IP-connected CPU to generate the packet). After being sent by switch 118, transcoder instances 104-1 and 104-2 receive synchronization multicast packet 710 at the same point in time in the AV content. Because Stream #2 did not include a synchronization multicast packet 710, transcoder instance 104-N does not receive a synchronization multicast packet 710.

[0050] While the use of a synchronization multicast packet is an elegant solution to ensure that the transcoders begin transcoding at exactly the same time, so that all segments across transcoders are aligned, other tools and methods may also be implemented. For example, port filtering may be used to inhibit the flow of IP packets to the individual transcoders until all transcoders have joined the multicast. Thereafter, when the flow of IP packets is initiated, each transcoder instance would begin parsing at exactly the same point, ensuring transcoding commences at the next Group of Pictures boundary.

[0051] Figure 8 depicts a simplified flowchart 800 of a method for processing IP packets in a transcoder instance 104 according to one embodiment. At 802, two or more transcoder instances 104 receive IP packets, including a synchronization IP packet. At 804, the two or more transcoder instances detect the synchronization IP packet at the

same point in the IP packet stream. At 806, the two or more transcoder instances transcode the IP packets after detecting the synchronization IP packet.

[0052] System 100 may be configured in a modular architecture. For example, a modular architecture allows for a wide range of transcoder solutions to be packaged within the same chassis and backplane design without the need to re-design other parts of the system. By providing an IP-based interface, such as IP receiver interface 308 and IP transmitter interface 312, the interfaces are standardized for transcoder instances 104. That is, each transcoder instance 104 would use the same protocol to send and receive IP packets. However, if the conventional MPEG transport stream for video and PCI being used for control is used, different transcoder instances 104 may use proprietary non-IP based interfaces that would not allow common interfacing techniques to be used. That is, when a transcoder instance 104 needs to be inserted into system 100, the proprietary non-IP based interface would not allow for a modular architecture to be realized. Rather, programming to insert the new transcoder instance 104 would be needed to interface with the proprietary interface. However, because system 100 includes a common IP-based backplane design in which each transcoder instance 104 communicates via an IP protocol, such as Ethernet, to router 116 and switch 118 for both control and also communication of video, transcoder instances 104 may be removed and inserted in a modular fashion. That is, reconfiguration of the system is not needed to insert various transcoder instances 104.

[0053] Also, the use of an IP-based backplane design for communication enables system 100 to be highly scalable by cascading multiple Ethernet switches 118 and routers 116. That is, additional routers and switches may be added as additional transcoder instances 104 are added.

[0054] In view of the many possible embodiments to which the principles of the present discussion may be applied, it should be recognized that the embodiments described herein with respect to the drawing figures are meant to be illustrative only and should not be taken as limiting the scope of the claims. Therefore, the techniques as

described herein contemplate all such embodiments as may come within the scope of the following claims and equivalents thereof.

CLAIMS

We claim:

1. A method for synchronizing video in a system including a host central processing unit ("CPU") and a plurality of transcoder instances, the method comprising:
 - receiving, by two or more transcoder instances in the plurality of transcoder instances, a synchronization Internet Protocol ("IP") packet from the host CPU, wherein the synchronization IP packet is received as one packet in a multicast stream of IP packets;
 - detecting, by the two or more transcoder instances in the plurality of transcoder instances, the synchronization IP packet at the same point in the multicast stream; and
 - transcoding, by the two or more transcoder instances in the plurality of transcoder instances, the multicast stream of IP packets immediately after detecting the synchronization IP packet.
2. The method of claim 1 wherein the plurality of transcoder instances are in communication with a switch and wherein the switch is configured to receive the multicast stream of IP packets and send the multicast stream of IP packets to predetermined transcoder instances.
3. The method of claim 2 wherein the switch utilizes pointer technology to ensure that data are not copied or buffered by the plurality of transcoder instances.
4. The method of claim 2 wherein the host CPU is in communication with the switch and wherein the host CPU is configured to insert the synchronization IP packet in the multicast stream.

5. The method of claim 4 wherein the host CPU is configured to insert the synchronization IP packet into the multicast stream prior to the switch receiving the multicast stream.
6. The method of claim 4 wherein the host CPU utilizes a timer to ensure that the synchronization IP packet is received by the two or more transcoder instances in the plurality of transcoder instances at the same point in the multicast stream.
7. The method of claim 4 wherein the host CPU is configured to control the plurality of transcoder instances which join the multicast.
8. The method of claim 7 wherein the host CPU is configured to inhibit the flow of IP packets until all of the plurality of transcoder instances have joined the multicast.
9. The method of claim 1 wherein the synchronization IP packet comprises a null packet with a code word to identify the packet as a synchronization packet.
10. The method of claim 1 wherein transcoding the multicast stream of IP packets immediately after detecting the synchronization IP packet creates aligned segments across the two or more transcoder instances.

11. An apparatus configured to synchronize video in a system including a host central processing unit ("CPU") and a plurality of transcoder instances, the apparatus comprising:
 - one or more computer processors; and
 - a non-transitory computer-readable storage medium containing instructions that, when executed, control the one or more computer processors to be configured for:
 - receiving, by two or more transcoder instances in the plurality of transcoder instances, a synchronization Internet Protocol ("IP") packet from the host CPU, wherein the synchronization IP packet is received as one packet in a multicast stream of IP packets;
 - detecting, by the two or more transcoder instances in the plurality of transcoder instances, the synchronization IP packet at the same point in the multicast stream; and
 - transcoding, by the two or more transcoder instances in the plurality of transcoder instances, the multicast stream of IP packets immediately after detecting the synchronization IP packet.
12. The apparatus of claim 11 wherein the plurality of transcoder instances are in communication with a switch and wherein the switch is configured to receive the multicast stream of IP packets and send the multicast stream of IP packets to predetermined transcoder instances.
13. The apparatus of claim 12 wherein the switch utilizes pointer technology to ensure that data are not copied or buffered by the plurality of transcoder instances.
14. The apparatus of claim 12 wherein the host CPU is in communication with the switch and wherein the host CPU is configured to insert the synchronization IP packet in the multicast stream.

15. The apparatus of claim 14 wherein the host CPU is configured to insert the synchronization IP packet into the multicast stream prior to the switch receiving the multicast stream.
16. The apparatus of claim 14 wherein the host CPU utilizes a timer to ensure that the synchronization IP packet is received by the two or more transcoder instances in the plurality of transcoder instances at the same point in the multicast stream.
17. The apparatus of claim 14 wherein the host CPU is configured to control the plurality of transcoder instances which join the multicast.
18. The apparatus of claim 17 wherein the host CPU is configured to inhibit the flow of IP packets until all of the plurality of transcoder instances have joined the multicast.
19. The apparatus of claim 11 wherein the synchronization IP packet comprises a null packet with a code word to identify the packet as a synchronization packet.
20. The apparatus of claim 11 wherein transcoding the multicast stream of IP packets immediately after detecting the synchronization IP packet creates aligned segments across the two or more transcoder instances.

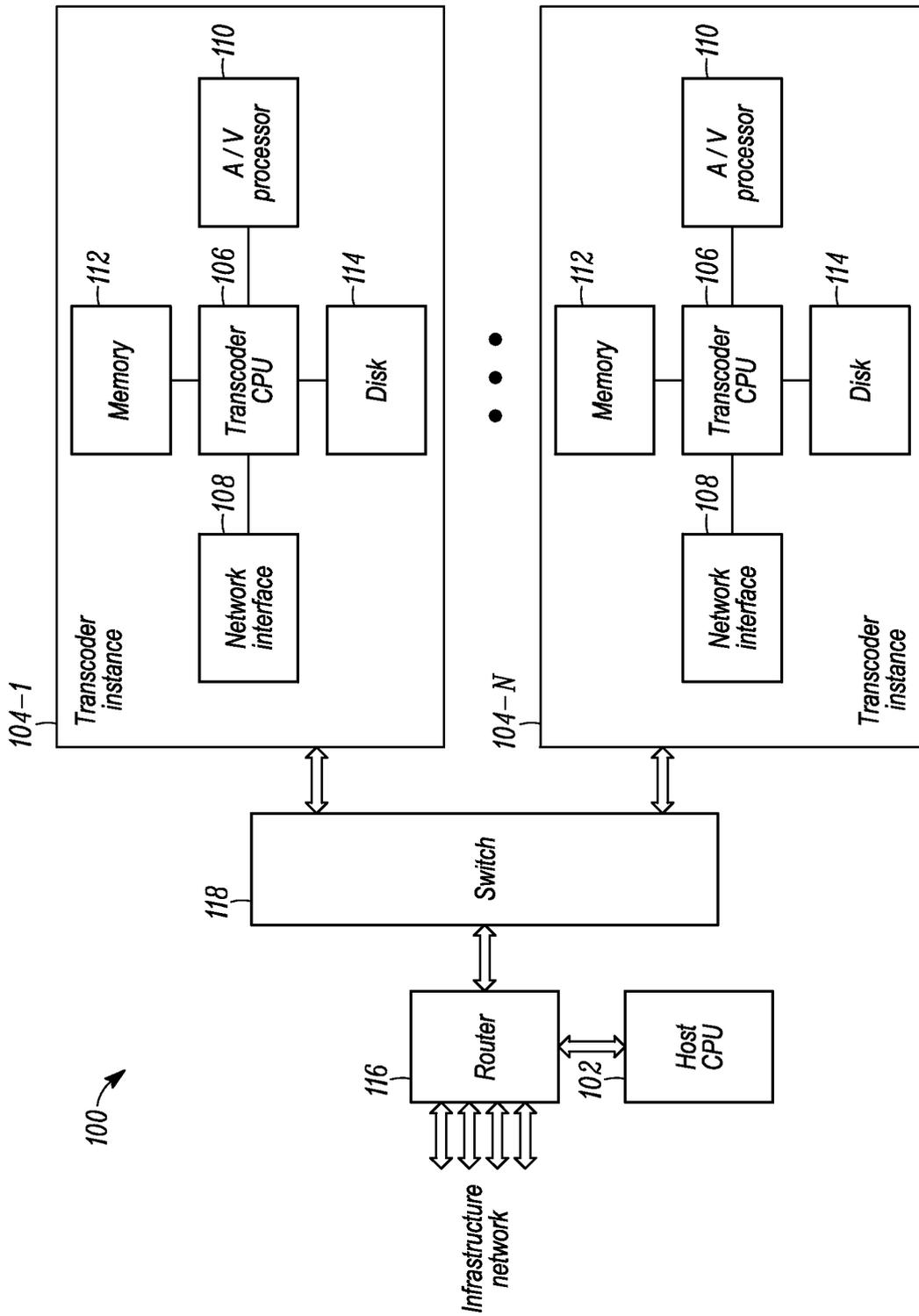


FIG. 1

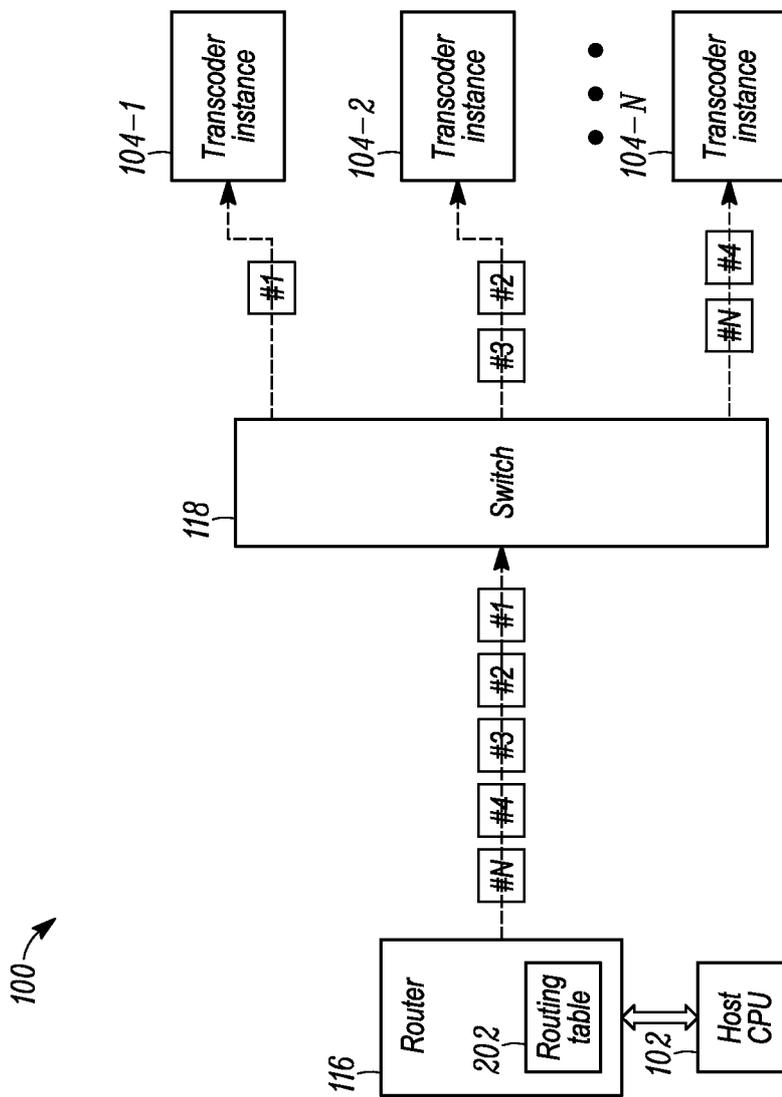


FIG. 2

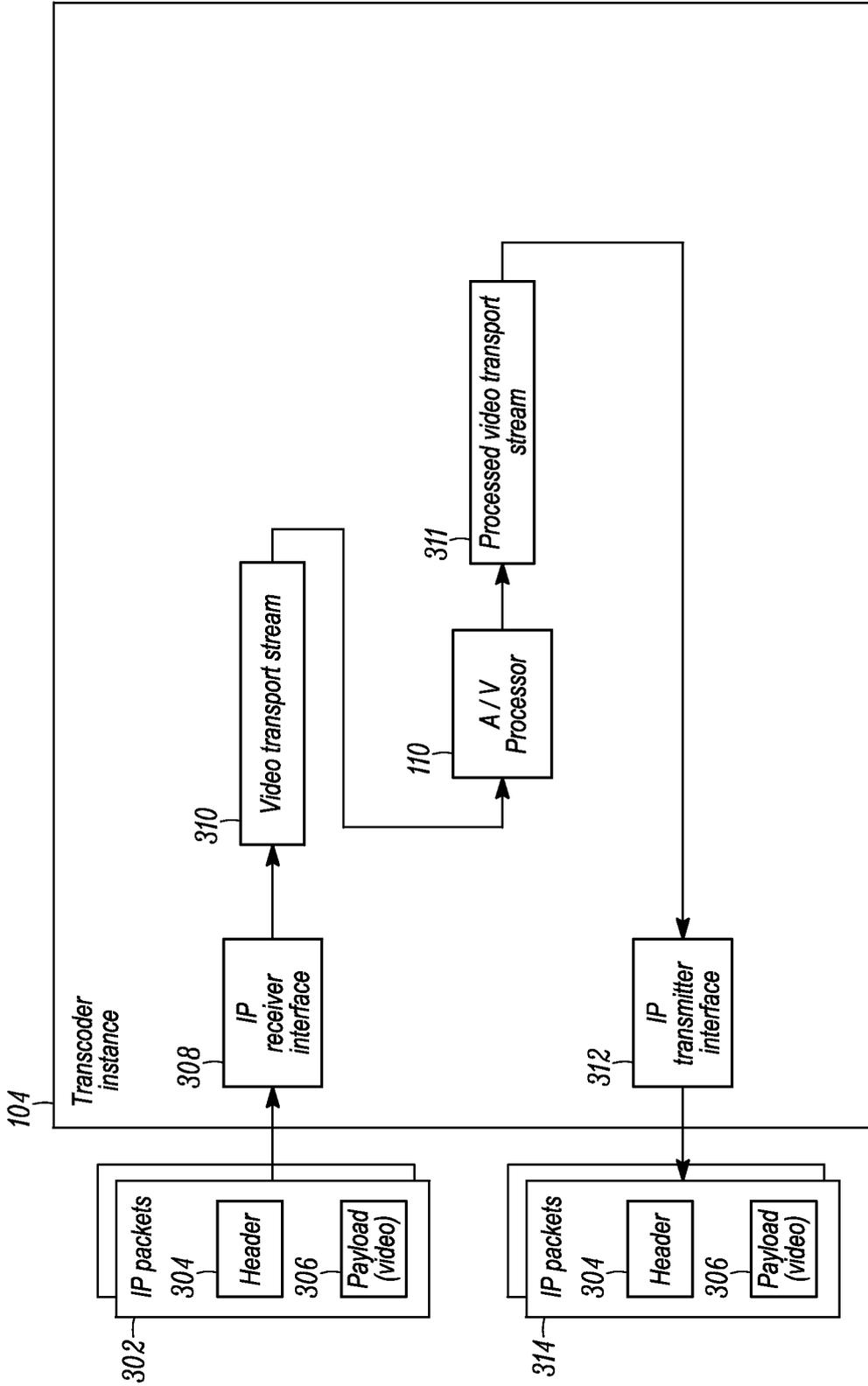


FIG. 3

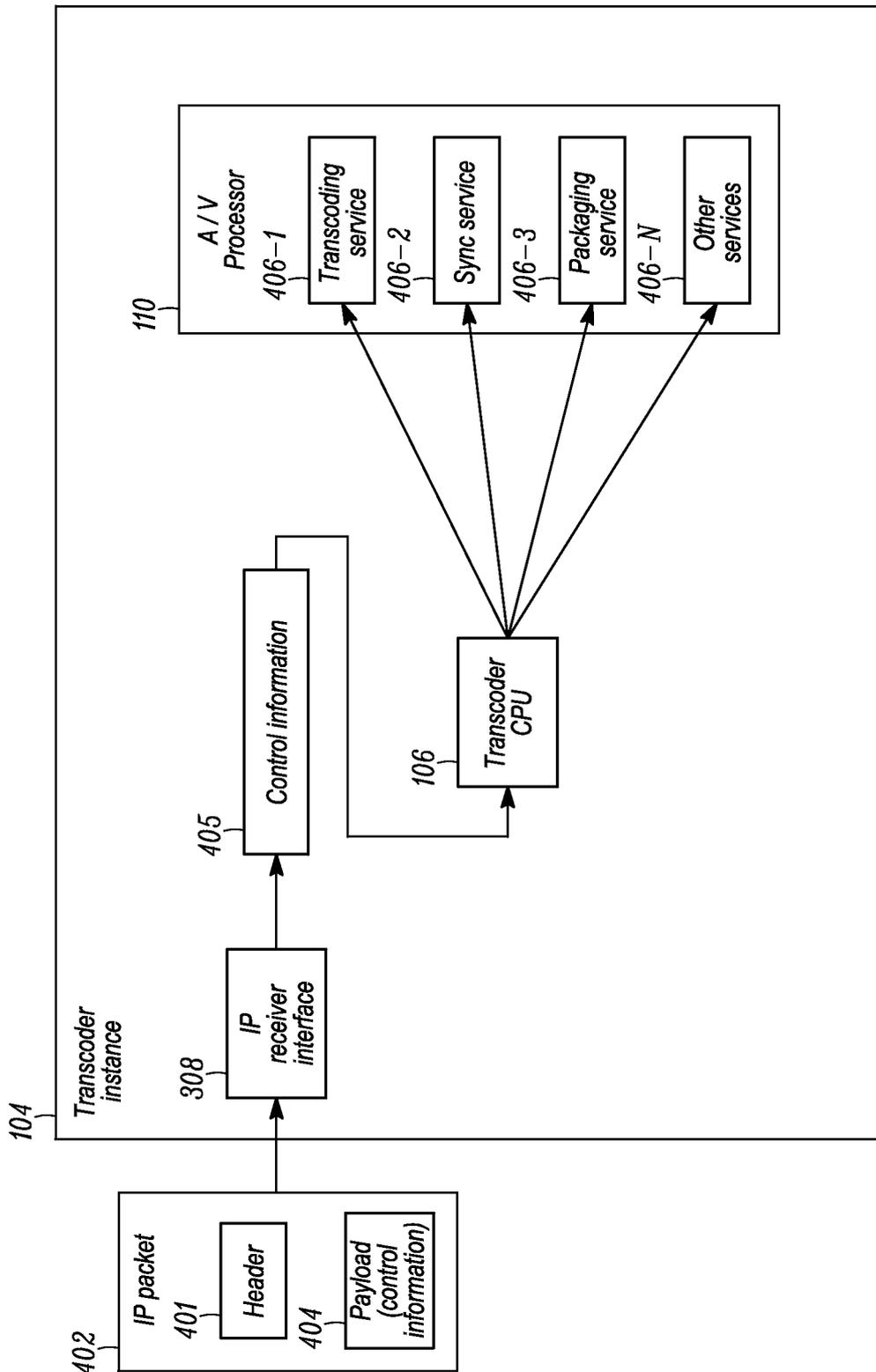


FIG. 4

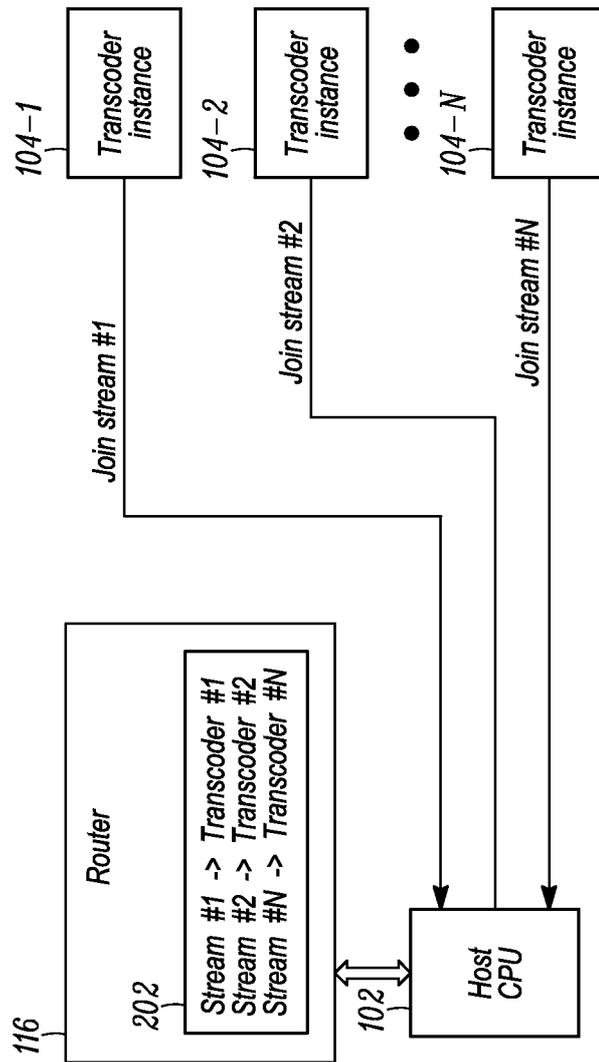


FIG. 5

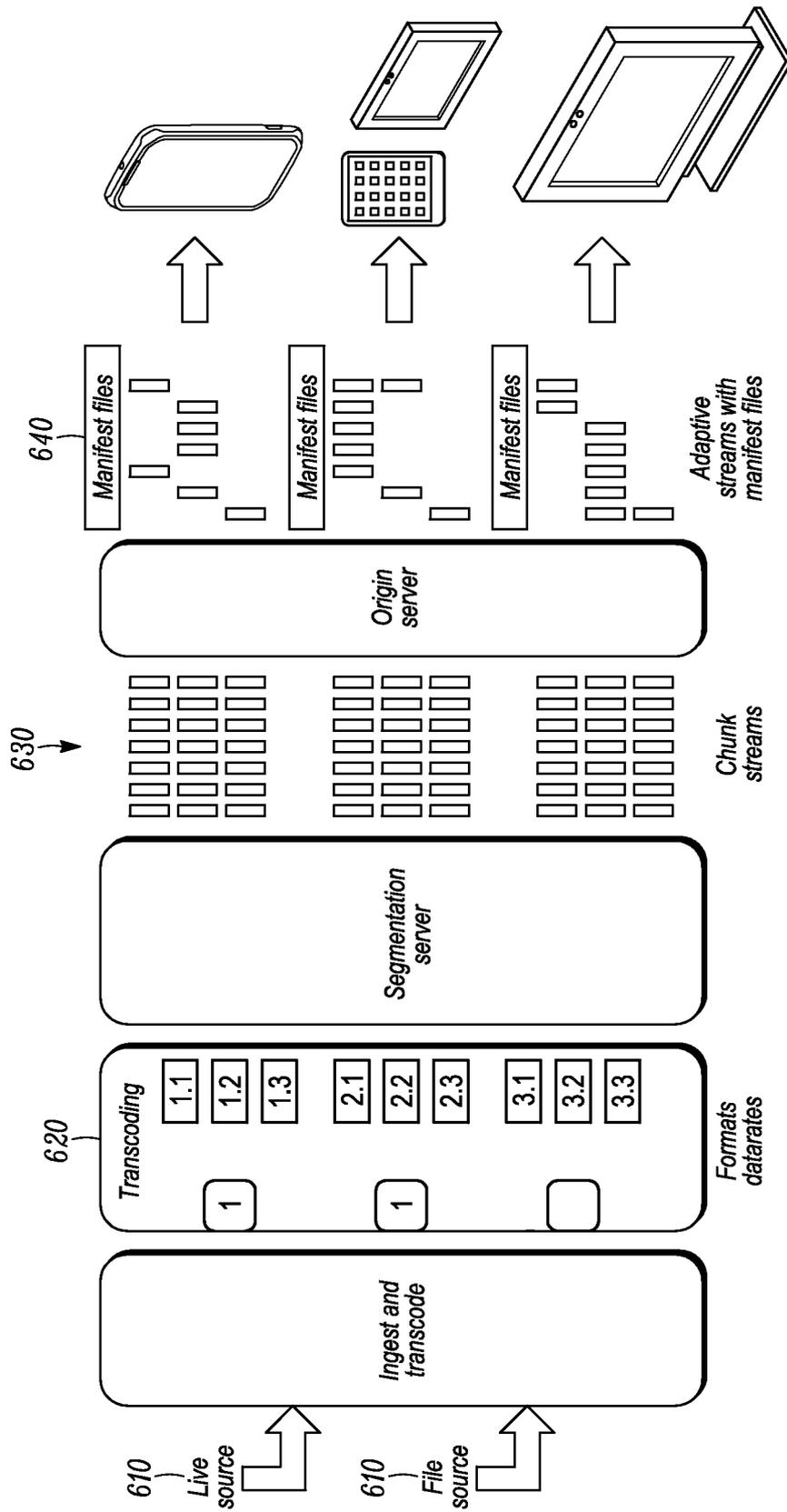


FIG. 6

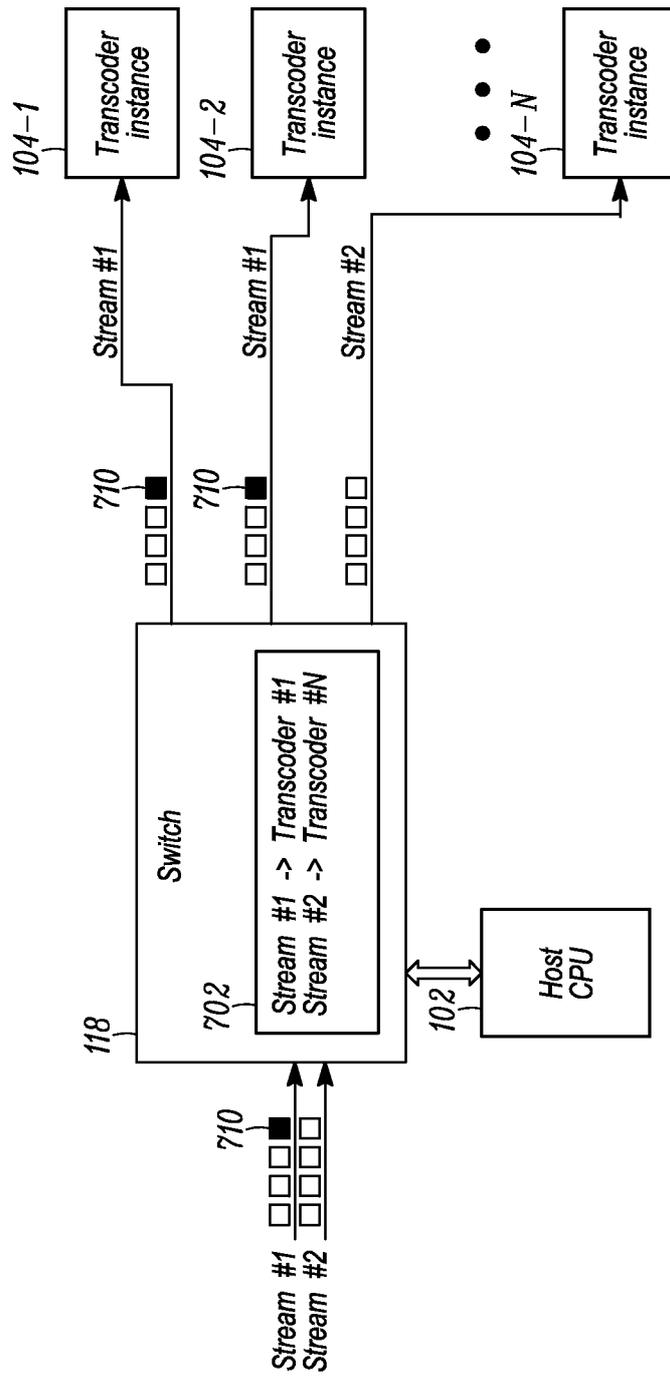


FIG. 7

800 →

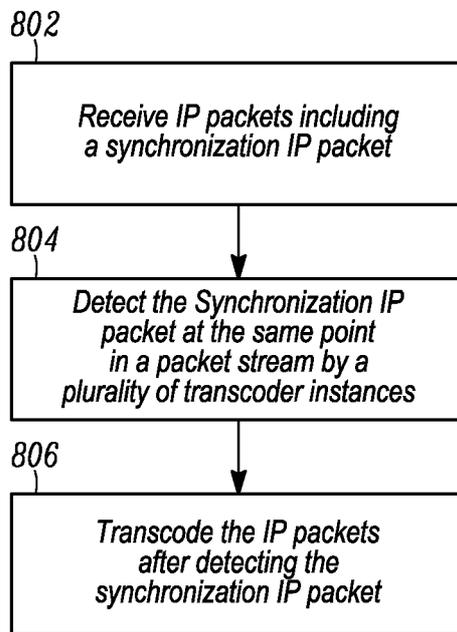


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No PCT/US2013/041525

A. CLASSIFICATION OF SUBJECT MATTER		
I NV . H94N2 1/2381 H04N2 1/242 HQ4N2 1/43 H04N2 1/438 H04N21/643 H04N2 1/8547 H04L12/ 18		
ADD . According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) H04N H04 L		
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 2010/ 106075 AI (KONINKL KPN NV [NL] ; TNO [NL] ; STOKKING HANS MAARTEN [NL] ; WALRAVEN FA) 23 September 2010 (2010-09-23)	1 , 3 , 7 - 9 , II . 13. 17 - 19
Y	figures 1, 2, 5 page 8, line 1 - line 17 page 9, line 27 - page 10, line 8 page 13, line 17 - line 20 page 14, line 21 - line 27 page 16, line 21 - line 33 -----	2 , 4 - 6 , 12 , 14 - 16
Y	EP 1 553 774 AI (MATSUSHITA ELECTRIC IND CO LTD [JP] PANASONIC CORP [JP]) 13 July 2005 (2005 -07 - 13)	4 - 6 , 14 - 16
A	figure 1 paragraph s [0014] , [0040] ----- -/--	1 , 11
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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	Date of mailing of the international search report	
22 July 2013	22/08/2013	
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 Votini , Stefano	

INTERNATIONAL SEARCH REPORT

International application No PCT/US2013/041525

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2009/201988 A1 (GAZIER MICHAEL [CA] ET AL) 13 August 2009 (2009-08-13) figure 5 -----	2,12
A	US 2008/170531 A1 (PETRY BRIAN D [US] ET AL) 17 July 2008 (2008-07-17) paragraphs [0070] , [0080] -----	1-20
A	US 2004/218527 A1 (SCHWARTZ MAYER D [US]) 4 November 2004 (2004-11-04) figures 2A, 2B -----	1,11

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/041525

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
wo 2010106075	AI	23-09 -2010	
		CN 1023566 19 A	15-02 -20 12
		EP 2409432 AI	25-01-20 12
		JP 2012520648 A	06-09 -20 12
		KR 201 10 125674 A	21- 11-20 11
		US 2012036277 AI	09-02 -20 12
		WO 2010 106075 AI	23-09 -20 10

EP 1553774	AI	13-07 -2005	
		AU 2003281 136 AI	02-02 ~2004
		CN 1669320 A	14--09-2005
		EP 1553774 AI	13-07-2005
		JP 4042745 B2	06--02-2008
		JP 46661 10 B2	06--04-20 11
		JP 5 170166 B2	27-03 -20 13
		JP 2010226766 A	07- 10--20 10
		JP 2010246146 A	28--10--20 10
		KR 20050025607 A	14-03-2005
		KR 20060 1320 17 A	20- 12-2006
		US 2005237434 AI	27-10--2005
		US 201 1023078 AI	27-01-20 11
		WO 2004008760 AI	22-01-2004

US 2009201988	AI	13-08-2009	NONE

US 200817053 1	AI	17-07 -2008	
		CN 101730995 A	09-06-20 10
		EP 2 119 115 AI	18- 11-2009
		TW 200845637 A	16- 11-2008
		US 2008 17053 1 AI	17-07 -2008
		US 201 1116500 AI	19-05 -20 11
		WO 2008089030 AI	24-07 -2008

US 20042 18527	AI	04- 11-2004	
		EP 1618396 AI	25-01-2006
		US 20042 18527 AI	04- 11-2004
		WO 200409980 1 AI	18- 11-2004
