(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 June 2008 (05.06.2008)

PCT

(10) International Publication Number
WO 2008/066593 A2

(51) International Patent Classification:
H04J 3/06 (2006.01)

(21) International Application Number:
PCT/US2007/017522

(22) International Filing Date: 7 August 2007 (07.08.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/603,849 22 November 2006 (22.11.2006) US

(71) Applicant *(for all designated States except US)*: **CISCO TECHNOLOGY, INC.** [US/US]; 170 West Tasman Drive, San Jose, CA 95134 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **FIRESTONE, Scott** [US/US]; 735 Tiana Lane, Mountain View, CA 94041 (US). **KADIYALA, Madhavi** [IN/US]; 1717 Marshall Court, #a, Los Altos, CA 94024 (US). **BAIRD, Randall, B.** [US/US]; 10100 Scull Creek Drive, Austin, TX 78730 (US). **BANGALORE, Manjunath, S.** [IN/US]; 2710 Nicasio Court, San Jose, CA 95127 (US). **SARKAR, Shantanu** [US/US]; 6758 Tunbridge Way, San Jose, CA 95120 (US).

(74) Agent: **BEREZNAK, Bradley, J.**; Burgess & Bereznak, 800 W. El Camino Real, Ste. 180, Mountain View, CA 94040 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

WO 2008/066593 A2

(54) Title: LIP SYNCRONIZATION FOR AUDIO/VIDEO TRANSMISSIONS OVER A NETWORK

(57) Abstract: In one embodiment, a system includes a video mixer coupled with an audio mixer for exchange of information that includes a first set of delay values respecting input audio streams received by the audio mixer from a plurality of source endpoints, and output audio streams sent from the audio mixer to a plurality of destination endpoints. The information further including a second set of delay values respecting the corresponding input video streams. The audio mixer calculates end-to-end video delays, and the video mixer calculates end-to-end audio delays. The audio mixer delays the output audio streams to equalize the end-to-end audio and video delays in the event that the end-to-eπd audio delays are less than the end-to-end video delays, and the video mixer delays the output video streams to equalize the end-to-eπd audio and video delays in the event that the end-to-end video delays are less than the end-to-end audio delays.

LIP SYNCHRONIZATION FOR AUDIO / VIDEO TRANSMISSIONS OVER A NETWORK

TECHNICAL FIELD

[0001] This disclosure relates generally to the field of transmission of audio / video data packets over a network.

BACKGROUND

[0002] Human beings can detect very small differences in the synchronization between video and its accompanying audio soundtrack. People are especially good at recognizing a lack of synchronizations between the lips of speakers in video media, such as in a video conference session, and the reproduced speech they hear. Lip-synchronization ("lipsync") between audio and video is therefore essential to achieving a high quality conferencing experience.

[0003] Past approaches for ensuring good lipsync include reliance upon the Real-Time Transport Control Protocol (RTCP) mapping between timestamps generated by the separate audio and video encoding devices and a Network Time Protocol (NTP) "wall clock" time. Although useful in achieving good lipsync for point-to-point audio/video sessions, this approach breaks down in situations such as where audio and video mixers are inserted for multi-party conferences. Other approaches to the problem of lipsync include co-locating the audio and video mixers on the same device, such as is typically done in most video multipoint control units (MCUs). However, when the audio mixer and video mixer/switch are distributed one or the other of the audio or video mixing devices typically

must delay its output stream to ensure that arrival time of the audio and video packets provides adequate lipsync, which can be difficult to achieve.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The present invention will be understood more fully from the detailed description that follows and from the accompanying drawings, which however, should not be taken to limit the invention to the specific embodiments shown, but are for explanation and understanding only.

[0005] **Figure 1** illustrates an example architecture for a video conferencing system.

[0006] **Figure 2** is an example of one embodiment of an audio mixer.

[0007] **Figure 3** is an example diagram of one embodiment of a video mixer.

[0008] **Figure 4** illustrates basic components of an example node or network device.

[0009] **Figure 5** illustrates an example method of operation for an audio mixer and a video mixer.

DESCRIPTION OF EXAMPLE EMBODIMENTS

[0010] In the following description specific details are set forth, such as device types, system configurations, communication methods, etc., in order to provide a thorough understanding of the present invention. However, persons having ordinary skill in the relevant arts will appreciate that these specific details may not be needed to practice the present invention.

[0011] In the context of the present application, a computer network is a geographically distributed collection of interconnected subnetworks for transporting data between nodes, such as intermediate nodes and end nodes (also referred to as endpoints). A local area network (LAN) is an example of such a subnetwork; a plurality of LANs may be further interconnected by an intermediate network node, such as a router, bridge, or switch, to extend the effective "size" of the computer network and increase the number of communicating nodes. Examples of the devices or nodes include servers, mixers, control units, and personal computers. The nodes typically communicate by exchanging discrete frames or packets of data according to predefined protocols.

[0012] In general, an endpoint represents an end user, client, or person who is capable of participating in an audio conference session via conferencing system. Endpoint devices that may be used to initiate or participate in a conference session include a personal digital assistant (PDA); a personal computer (PC), such as notebook, laptop, or desktop computer; an audio/video appliance; a streaming client; a television device with built-in camera and microphone; or any other device, component,

element, or object capable of initiating or participating in exchanges with a video conferencing system.

[0013] Figure 4 illustrates basic components of an example node or network device 40, which typically comprises a number of basic subsystems that includes a processor subsystem 41, a main memory 42 and an input/output (I/O) subsystem 45. Data is transferred between main memory ("system memory") 42 and processor subsystem 41 over a memory bus 43, and between the processor and I/O subsystems over a system bus 46. Examples of the system bus may include the conventional lightning data transport (or hyper transport) bus and the conventional peripheral component interconnect (PCI) bus. Device 40 may also comprise other hardware units/modules 44 coupled to system bus 46 for performing additional functions. Processor subsystem 41 may comprise one or more processors and a controller device that incorporates a set of functions including a system memory controller, support for one or more system buses and direct memory access (DMA) engines.

[0014] Figure 1 is an example architecture of a video conferencing system 100 which comprises endpoints devices 102-107 (labeled EP1-EP5) that each send/receive audio and video packets. In this example, endpoints 102-104 are sources of audio / video content – that is, they send audio from their microphones to an audio mixer (i.e., bridge) 120. Endpoints 102-104 also send video from their associated cameras to a video media switch (MS) 122 (also known as a video mixer). It is appreciated that the audio and video mixers shown in Figure 1 are separate, distinct devices or components that are physically located at different places on a

communications network. It should be further understood that audio mixer 120 represents any device or component that combines more than one audio input stream to produce a composite audio output stream. Similarly, video mixer 122 represents any device or component that combines more than one video input stream to produce a composite video output stream.

[0015] Endpoints 106-107 represent the destination or consumers of the media content. Each endpoint 102-104 and 106-107 is provided with separate audio and video network connections. Connections 108, 110, and 112 respectively provide input from the microphones of endpoints 102, 103, and 104 to audio mixer 120, with connections 114 and 116 providing the mixed audio output generated by mixer 120 (typically of the three or four loudest participants) to the speakers of endpoints 106 and 107, respectively. Similarly, connections 109, 111, and 113 respectively provide input from the cameras of endpoints 102, 103, and 104 to MS 122. Connections 115 and 117 provide the video output generated by MS 122 to the monitors of respective endpoints 106 and 107. Although one-way paths (e.g., source / sender endpoints on the left, and destination / receiver endpoints on the right) are explicitly depicted in Figure 1, it is appreciated that all of the endpoints shown in Figure 1 typically operate as both receivers and senders. That is, a video conference session usually involves transmission of audio and video data packets in both directions at the same time.

[0016] In the embodiment shown, mixer 120 and MS 122 may both include a digital signal processor (DSP) or firmware/software-based system that mixes and/or switches audio / video signals received at its input ports

under the control of a video conference server (not shown). The audio / video signals received at the conference server ports originate from each of the conference or meeting participants (e.g., individual conference participants using endpoint devices 102-107), and possibly from an interactive voice response (IVR) system. Conference server 20 may also incorporate or be associated with a natural language automatic speech recognition (ASR) module for interpreting and parsing speech of the participants, and standard speech-to-text (STT) and text-to-speech (TTS) converter modules.

[0017] Audio mixer 120, in addition to mixing the audio transmissions of the N loudest speakers, may also create output audio streams that have different combinations of speakers. For example, in the case where endpoint 106 is one of the loudest speakers in the video conference session, mixer 120 generates a mixed audio output to endpoint 106 via connection 114 that does not include the audio of endpoint 106. On the other hand, the audio mix output to endpoint 107 via connection 116 does include the audio generated by endpoint 106 since endpoint 106 is one of the loudest speakers. In this way, endpoint 106 does not receive an echo of its own audio output coming back from the audio mixer.

[0018] It is appreciated that in different specific implementations the media paths for the conference participants may include audio / video transmissions, e.g., Real-Time Transport Protocol (RTP) packets sent across a variety of different networks (e.g., Internet, intranet, PSTN, etc.), protocols (e.g., IP, Asynchronous Transfer Mode (ATM), Point-to-Point

Protocol (PPP)), with connections that span across multiple services, systems, and devices.

[0019] Figure 1 further illustrates the one-way network packet transmission delays from each endpoint 102-104 to audio mixer 120 and MS 122, as well as the transmission delays from audio mixer 120 and MS 122 to endpoints 106 & 107. For example, the delay on connection 108 from endpoint 102 to audio mixer 120 is labeled dEP1a. The delay on connection 109 from endpoint 102 to MS 122 is labeled dEP1v, and so on. Similarly, the mixed audio output stream sent on connection 114 from audio mixer 120 to endpoint 106 has an associated delay dEP4a, and the mixed video output stream sent from MS 122 to endpoint 106 via connection 115 has a transmission delay dEP4v.

[0020] Synchronization information is communicated between audio mixer 120 and MS 122 via connection 124. During normal operation, audio mixer 120 ensures that there is a constant offset between each of the audio streams. In other words, there is a relative synchronization in terms of the offsets between individual audio streams and the final output audio mix. This information is included in the synchronization information that audio mixer 120 sends to video mixer 122. When MS 122 receives this synchronization information it matches the relative video offsets in the summed video stream with the relative offsets that are established by audio mixer 120. (It is appreciated that this process may also work in reverse, with audio mixer 120 matching the relative audio offsets in the summed audio stream with the relative offsets that are established by MS 122.) After exchanging delay information that describes the offsets, the audio mixer

and video mixer coordinate with each other to achieve synchronization at the destination endpoints.

[0021] It should be understood that in the example of Figure 1, endpoints 102-104 and 106-107 comprise so-called "dumb" endpoints; that is, endpoint devices that do not send / receive RTCP data, nor do they rely upon timestamp information. Instead, endpoints 102-104 and 106-107 simply synchronize audio and video packets based on their arrival time at the network interface. In other words, endpoints 102-104 and 106-107 do not attempt any sort of synchronization. Senders send audio and video packets simultaneously, and receivers assume that audio and video arriving at the same time from the network are synchronous.

[0022] In order to ensure good lipsync at the destination endpoints, an out-of-band synchronization protocol is provided between mixers 120 & 122 that allows these audio and video mixers to send their respective media so that it arrives at dumb endpoints (e.g., endpoints 106 & 107) at approximately the same time, i.e., sufficient to ensure good lipsync. In one embodiment shown, mixer 120 and MS 122 execute a protocol to compute whether to delay their respective output streams, and if so, by how much, in order to insure that audio and video packets arrive synchronously at the destination endpoints. This protocol allows the separation of the audio mixer and video mixer components such that the audio from the audio mixer need not be relayed through the video media switch.

[0023] Practitioners in the art will appreciate that the delays shown in Figure 1 may be calculated by mixer 120 and MS 122 in a variety of ways. For instance, standard network ping techniques may be utilized to calculate

the network delays dEP1a-dEP5a, which information is then sent by audio mixer 120 to MS 122. Likewise, conventional delay measurement techniques may be utilized to compute the network delays dEP1v-dEP5v, which information is then sent by MS 122 to audio mixer 120. In the event that one or more of the endpoints is configured to send RTCP information, the audio mixer can send a RTCP sender report to the endpoint and receive back an RTCP receiver report in return, which may then be used to calculate a round-trip delay time.

[0024] It is appreciated that the total delay incurred on each stream from its originating endpoint up through the mixers is exchanged. This delay figure is half the aggregate round-trip time from the source endpoint (or endpoints), plus the mixer delay caused by the jitter buffers in the mixer, plus the actual mixer formation delay. This value is computed for every source pair of audio/video stream that are to be rendered at any instant in time. Information for the audio-video stream pairs that are not going to be immediately rendered by an endpoint may be omitted, as well as any audio stream that does not have a corresponding active video stream.

[0025] The first phase of the out-of-band protocol involves the audio mixer sending the video mixer (or vice versa) information regarding relative offsets between the individual components of the mix. The audio mixer sends the offset information to the video mixer so as to allow the video mixer to generate a mixed video stream that duplicates these relative offsets. It should be understood that the video mixer 122 includes its own input delay buffers, although those buffers are usually not used as jitter buffers, instead they are used to delay one or more streams so that when

those streams are summed together the relative offsets between those streams are the same as the relative offsets in the audio mix. Typically, one of those delays is set to zero and the remaining delays are increased accordingly in order to achieve the necessary relative offsets. It is appreciated that the audio and video mixers may resample their respective media as necessary in order to maintain relatively constant delays through their pipelines.

[0026] The second phase of the protocol involves the audio mixer and video mixer coordinating with each other so as to achieve synchronization (lipsync) at each of the various endpoint devices.

[0027] Delay and other information may be transported between the mixers in a number of different ways. One embodiment utilizes a dummy session created between audio mixer 120 and video mixer 122, with the relevant information being encapsulated in a RTCP APP message. Other embodiments may encapsulate the information in other types of messages, signals, or streams (e.g., a XML-encoded HTTP stream between the two devices). It is appreciated that in the case of voice-activated video streams, up-to-date information is generated and exchanged immediately following an active speaker change.

[0028] Figure 2 is an example of one embodiment of an audio mixer 200 that includes a network delay measurement and delay buffer control unit 220 that receives audio inputs from the various endpoint devices, e.g., audio inputs 1-3 on lines 201-203, respectively, and produces mixed audio outputs, e.g., audio outputs 1 & 2 on lines 204 & 205, respectively. In other words, audio mixer 200 sums together all the audio streams from multiple

source endpoints to create one or more mixed output audio streams delivered to destination endpoints. Audio mixer 200 includes input jitter buffers 206-208 are included to supply evenly timed audio input data to summation units 210 and 212, which add together selected ones of the buffered audio input streams. Note that different buffering delays (e.g., da1, da2, da3, etc.) corresponding to different buffer depths may be selected for each jitter buffer in order to absorb data starvation resulting from network jitter.

[0029] Summation units 210 & 212 each generate a customized output audio mix for a particular destination endpoint after the relative input stream delays (dEP1a, dEP2a, dEP3a, etc.) have been normalized. Additional summation units can be included for more destination endpoints. Output delay buffers 214 and 216 produce audio outputs 204 and 205 that take into account the network path delays to the respective destination endpoint for proper lipsync with the matching video stream. In other words, audio mixer 200 optimally delays the result of each summation unit in order to achieve lipsync with the video streams at the receiving endpoints. This operation, for example, is achieved in Figure 2 by delays aadj1, aadj2, etc.

[0030] A sync-info port 222 connects to the video mixer to provide audio delay / offset information and to coordinate video output delays that match the audio output deliveries at the corresponding destination endpoints. This information exchange allows the video mixer to generate a mixed video stream to match these relative delays / offsets.

[0031] Consistent with the example embodiment described above, for each audio input received from a source endpoint, audio mixer 200 sends the video mixer a value representing the sum of the one-way network audio delay from the source endpoint to the audio mixer summation (sum) unit plus the input jitter buffer delay. Additionally, for each output endpoint, the audio mixer sends a value representing the sum of the summation unit delay (i.e., adout1 & adout2 for summation units 210 & 212, respectively) plus the one-way network audio delay from the audio mixer sum unit to each corresponding destination endpoint (shown as dEP4a & dEP5a in Figure 1).

[0032] By way of further example, assume that dEP1a=40ms; dEP2a=12ms; dEP3a=24ms; da1=30ms; da2=50ms; da3=10ms; adout1=4ms; adout2=2ms; dEP4a=5ms; and dEP5a=7ms. Given these delays, and further assuming zero adjustment delays (aadj1 & aadj2) at this point and no other endpoints in the example of Figures 1 & 2, audio mixer 200 sends the values shown in Table 2 to the video mixer:

TABLE 2

Audio EP1-to-sum = dEP1a + da1 = 40ms + 30ms = 70ms
Audio EP2-to-sum = dEP2a + da2 = 12ms + 50ms = 62ms
Audio EP3-to-sum = dEP3a + da3 = 24ms + 10ms = 34ms
Audio sum-to-EP4 = adout1 + dEP4a + aadj1 = 4ms + 5ms + 0ms = 9ms
Audio sum-to-EP5 = adout2 + dEP5a + aadj2 = 2ms + 7ms + 0ms = 9ms

[0033] Figure 3 is an example diagram of one embodiment of a video mixer 300, which, similar to the example of Figure 2, includes a network

delay measurement and delay buffer control unit 320 that receives video inputs from each endpoint device, e.g., video inputs 1-3 on lines 301-303, respectively. Each video input line 301-303 is coupled to a respective input delay buffer 306-308. Mixed video outputs 1 & 2 are shown produced by multipoint switches 310 & 312 on lines 304 & 305 for destination receiver endpoints EP4 & EP5, respectively. Each independent video switch 310 & 312 is associated with a respective output delay buffer 314 & 316 for ensuring that the sum of the relative offsets in each mixed video stream is equal to the relative offsets in the corresponding mixed audio stream. In other words, output delay buffers 314 and 316 produce respective delayed, mixed video output streams 304 and 305 for proper lipsync with the corresponding audio streams.

[0034] Consider the audio endpoint-to-summation unit values in the example given above (shown in Table 2). When video mixer 300 receives those audio delay values on connection 222, it adjusts the input delays (dv1-dv3) of buffers 306-308 to ensure that the relative offsets of the video streams 301-303 in the video mix equal the relative offsets in the audio mix. For example, assume that endpoint-to-multi-point switch (sum) unit video delays (with zero delay for dv1, dv2 and dv3) are dEP1v=13ms, dEP2v=20ms and dEP3v=10ms.

[0035] Video mixer 300 establishes values for the video input delay buffers (dv1, dv2 and dv3) such that the relative endpoint-to-sum unit offsets for audio and video are the same. The video mixer performs this operation by first determining which of the audio streams is delayed the least, which in this case, happens to be the audio stream from EP3. As a

result, video mixer 300 sets the corresponding video of delay buffer value, dv3, to zero. Next, the video mixer sets the other video input delay values (dv1 & dv2) such that the relative offsets between the audio and video streams are the same, which, for the example given results in the delay values shown below in Table 3

TABLE 3

(Video EP2-to-sum - Video EP3-to-sum) = (Audio EP2-to-sum - Audio EP3-to-sum)
((dEP2v + dv2) - (dEP3v + 0ms)) = (Audio EP2-to-sum - Audio EP3-to-sum)
((20ms + dv2) - (10ms)) = (62ms – 34ms)
dv2 = 18ms
(Video EP1-to-sum - Video EP3-to-sum) = (Audio EP1-to-sum - Audio EP3-to-sum)
((dEP1v + dv1) - (dEP3v + 0ms)) = (Audio EP1-to-sum - Audio EP3-to-sum)
((13ms + dv1) - (10ms)) = (70ms – 34ms)
dv1 = 33ms

[0036] Now, the delays in the video path up to the video mixer summation units are given as:

dEP1v + dv1 = 13ms + 33ms = 46ms
dEP2v + dv2 = 20ms + 18ms = 38ms
dEP3v + dv3 = 10ms + 0ms = 10ms

[0037] Note that at this point the relative offsets for the video paths are the same as the relative offsets for the audio path; that is, EP2 is delayed 20ms more than EP3, and EP1 is delayed 8ms more than EP2, for both the audio and video streams.

[0038] In a second phase of operation, the network delay measurement and delay buffer control unit 320 determines the appropriate output buffer delays to be introduced in the video output streams in order to achieve synchronization with the corresponding audio streams at the destination endpoint devices. The network delay measurement and delay buffer control unit 320 first calculates the end-to-end delay for the audio streams using the values it has received from the audio mixer on connection 222. Note that unit 320 need only calculate values for a single input stream, going to each output stream. In this example, using EP1: Audio EP1-to-EP4 = Audio EP1-to-sum + Audio sum-to-EP4 = 70ms + 9ms = 79ms; and Audio EP1-to-EP5 = Audio EP1-to-sum + Audio sum-to-EP5 = 70ms + 9ms = 79ms. Assuming that the delays in the video switch (summation) units are vout1 = 2ms and vout2=2ms, and assuming further that unit 320 has calculated the one-way network delays from video mixer 122 to the endpoints as dEP4v=15ms and dEP5v=20ms, then the resulting delay values (assuming vadj1=vadj2=0) for the present example are given below in Table 4.

TABLE 4

Video EP1-to-EP4 = (dEP1v + dv1 + vout1 + vadj1 + dEP4v) = (13ms + 33ms + 2ms + 0ms + 15ms) = 63ms
Video EP1-to-EP5 = (dEP1v + dv1 + vout2 + vadj2 + dEP5v) = (13ms + 33ms + 2ms + 0ms + 20ms) = 68ms

[0039] And because the corresponding delay in the audio path for the same input stream, EP1, is 79ms, the video mixer sets the delay values as vdadj1 = 79ms - 63ms = 16ms; and vdadj2 = 79ms - 68ms = 11ms. Adding these adjustments to the video output streams via delay buffers 314 & 316 ensures that the end-to-end audio delays equal the corresponding end-to-end video delays.

[0040] It is appreciated that if the end-to-end audio delays are less than the end-to-end video delays, then the audio streams are delayed instead of the video streams. Consistent with the example method given above, in the case where the audio delays are less than the video delays, the audio mixer ends up adjusting its delay values to achieve the same end-to-end delay as the video streams.

[0041] In terms of the example shown in Figure 1, audio mixer 120 and video mixer 122 exchange messages periodically in order to keep the delay value information current. If either mixer device measures a change in delay values for any of its streams, that information is communicated to the other mixer device.

[0042] In the event that the audio mixer and video mixer cannot determine the relevant audio and video streams pairs, such information may be communicated by a third-party network device.

[0043] Figure 5 illustrates an example method of operation for an audio mixer and a video mixer for the purpose of achieving lipsync. The process begins at block 501, wherein the audio mixer and a video mixer exchange information regarding delays / offsets that have been measured / calculated by each device. Assuming that the end-to-end audio delays are

less than the end-to-end video delays, the media switch next calculates / sets the input buffer delay values such that the relative endpoint-to-summation unit offsets for each of the corresponding audio and video streams are identical (block 502).

[0044] Next, the video mixer calculates the end-to-end delay for audio streams originating at a single source endpoint destined for each of the various receiver endpoints (block 503). These calculations are performed using the relative delay / offset values previously provided by the audio mixer in the information exchange. The video mixer uses the results of the end-to-end audio delay calculations to calculate and then set the output delay buffer adjustment values in order to equalize the end-to-end audio and video delays (block 504). In the event that the delay / offset information changes for either the audio mixer or video mixer (block 505), the entire process is repeated.

[0045] In the case where the sending or source endpoints utilize RTCP, but the receiving or destination endpoints are dumb, the audio mixer and video mixer may achieve relative synchronization without measuring the network delays from the source endpoints to the audio mixer / video mixer. Instead, the devices may utilize information in the RTCP streams from the endpoints. In this case, the audio mixer and video mixer can generate audio and video streams that are synchronized using a master clock provided by the audio mixer. The timestamps are in units of NTP time. In such a scenario, the audio mixer and video mixer calculate the mapping between the NTP timestamps, and the arrival time at the destination endpoint. Then, each device determines the offset between the

arrival times of the audio and video streams. For the audio mixer, and this mapping is: NTP real = NTPts * M + k _audio, where NTPts is the NTP timestamp, M is the scale factor that accounts for skew between timestamps and a master clock, and k is an offset. Likewise, the mapping for the video mixer is: NTP real = NTPts * M + k_video. (The factor M is the same for both devices, and is close to 1.0000.) The audio mixer and video mixer may determine the difference in arrival times by exchanging k values.

[0046] It should be understood that elements of the present invention may also be provided as a computer program product which may include a machine-readable medium having stored thereon instructions which may be used to program a computer (e.g., a processor or other electronic device) to perform a sequence of operations. Alternatively, the operations may be performed by a combination of hardware and software. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnet or optical cards, propagation media or other type of media/machine-readable medium suitable for storing electronic instructions. For example, elements of the present invention may be downloaded as a computer program product, wherein the program may be transferred from a remote computer or telephonic device to a requesting process by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection).

[0047] Additionally, although the present invention has been described in conjunction with specific embodiments, numerous

modifications and alterations are well within the scope of the present invention. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

We claim:

1. A method comprising:

receiving, by a video mixer, information from an audio mixer that includes delay values respecting input audio streams received by the audio mixer from a plurality of source endpoints, and output audio streams sent from the audio mixer to a plurality of destination endpoints, each input and output audio stream being respectively associated with a corresponding input and output video stream;

calculating, from the information received, an output delay value for each corresponding output video stream to ensure that source endpoint-to-destination endpoint audio and video delays are substantially equal; and

delaying, by the video mixer, each corresponding output video stream by the output delay value.

2. The method of claim 1 further comprising:

buffering, by the video mixer, each of the corresponding input and output video streams.

3. The method of claim 2 wherein the delaying comprises adjusting the buffering of each of the corresponding input and output video streams.

4. The method of claim 1 wherein the delay values include, for each of the input audio streams, a first delay from a source endpoint to a

summation unit of the audio mixer, and for each of the output audio streams, a second delay from the summation unit to a destination endpoint.

5. The method of claim 1 further comprising:

calculating, from the information received, a source endpoint-to-destination endpoint audio delay from one of the source endpoints to one of the destination endpoints.

6. The method of claim 1 further comprising:

receiving, by a video mixer, new information from the audio mixer that includes updated delay values.

7. A method comprising:

receiving, by an audio mixer, information from an video mixer that includes delay values respecting input video streams received by the video mixer from a plurality of source endpoints, and output video streams sent from the video mixer to a plurality of destination endpoints, each input and output video stream being respectively associated with a corresponding input and output audio stream;

calculating, from the information received, an output delay value for each corresponding output audio stream to ensure that source endpoint-to-destination endpoint audio and video delays are substantially equal; and

delaying, by the audio mixer, each corresponding output audio stream by the output delay value.

8. The method of claim 7 further comprising:

buffering, by the audio mixer, each of the corresponding input and output audio streams.

9. The method of claim 8 wherein the delaying comprises adjusting the buffering of each of the corresponding input and output audio streams.

10. The method of claim 7 wherein the delay values include, for each of the input video streams, a first delay from a source endpoint to a summation unit of the video mixer, and for each of the output video streams, a second delay from the summation unit to a destination endpoint.

11. The method of claim 7 further comprising:

calculating, from the information received, a source endpoint-to-destination endpoint video delay from one of the source endpoints to one of the destination endpoints.

12. The method of claim 7 further comprising:

receiving, by a audio mixer, new information from the video mixer that includes updated delay values.

13. A system comprising:

a video mixer;

an audio mixer coupled with the video mixer for exchange of information therebetween, the information including a first set of delay

values respecting input audio streams received by the audio mixer from a plurality of source endpoints, and output audio streams sent from the audio mixer to a plurality of destination endpoints, each input and output audio stream being respectively associated with a corresponding input and output video stream, the information further including a second set of delay values respecting the corresponding input video streams received by the video mixer from the plurality of source endpoints, and output video streams sent from the video mixer to the plurality of destination endpoints;

wherein the audio mixer is operable to calculate end-to-end video delays, and the video mixer is operable to calculate end-to-end audio delays, the audio mixer being further operable to delay the output audio streams to equalize the end-to-end audio and video delays in the event that the end-to-end audio delays are less than the end-to-end video delays, the video mixer being further operable to delay the output video streams to equalize the end-to-end audio and video delays in the event that the end-to-end video delays are less than the end-to-end audio delays.

14. The system of claim 13 wherein the audio mixer comprises:

input buffers to buffer each of the input audio streams;

output buffers to delay each of the output audio streams;

a plurality of summation units that mix the input audio streams to produce the output audio streams.

15. The system of claim 14 wherein the audio mixer further comprises:

a unit that controls the summation units and the input and output buffers.

16. The system of claim 15 wherein the unit is further operable to measure / calculate the first set of delay values.

17. The system of claim 13 wherein the video mixer comprises:

input buffers to buffer each of the input video streams;

output buffers to delay each of the output video streams;

a plurality of switches that mix the input video streams to produce the output video streams.

18. The system of claim 17 wherein the video mixer further comprises:

a unit that controls the switches and the input and output buffers.

19. The system of claim 18 wherein the unit is further operable to measure / calculate the second set of delay values.

20. Logic encoded in one or more media for execution and when executed is operable to:

during a video conference session, receive information from an audio mixer that includes delay values respecting input audio streams received by the audio mixer from a plurality of source endpoints, and output audio streams sent from the audio mixer to a plurality of destination endpoints,

each input and output audio stream being respectively associated with a corresponding input and output video stream;

calculate, from the information received, an output delay value for each corresponding output video stream to ensure that source endpoint-to-destination endpoint audio and video delays are substantially equal; and

adjust each corresponding output video stream by the output delay value.

21. The logic of claim 20 wherein the delay values include, for each of the input audio streams, a first delay from a source endpoint to a summation unit of the audio mixer, and for each of the output audio streams, a second delay from the summation unit to a destination endpoint.

22. Logic encoded in one or more media for execution and when executed is operable to:

during a video conference session, receive information from a video mixer that includes delay values respecting input video streams received by the video mixer from a plurality of source endpoints, and output video streams sent from the video mixer to a plurality of destination endpoints, each input and output video stream being respectively associated with a corresponding input and output audio stream;

calculate, from the information received, an output delay value for each corresponding output audio stream to ensure that source endpoint-to-destination endpoint audio and video delays are substantially equal; and

adjust each corresponding output audio stream by the output delay value.

23. The logic of claim 22 wherein the delay values include, for each of the input video streams, a first delay from a source endpoint to a switch of the video mixer, and for each of the output video streams, a second delay from the switch to a destination endpoint.

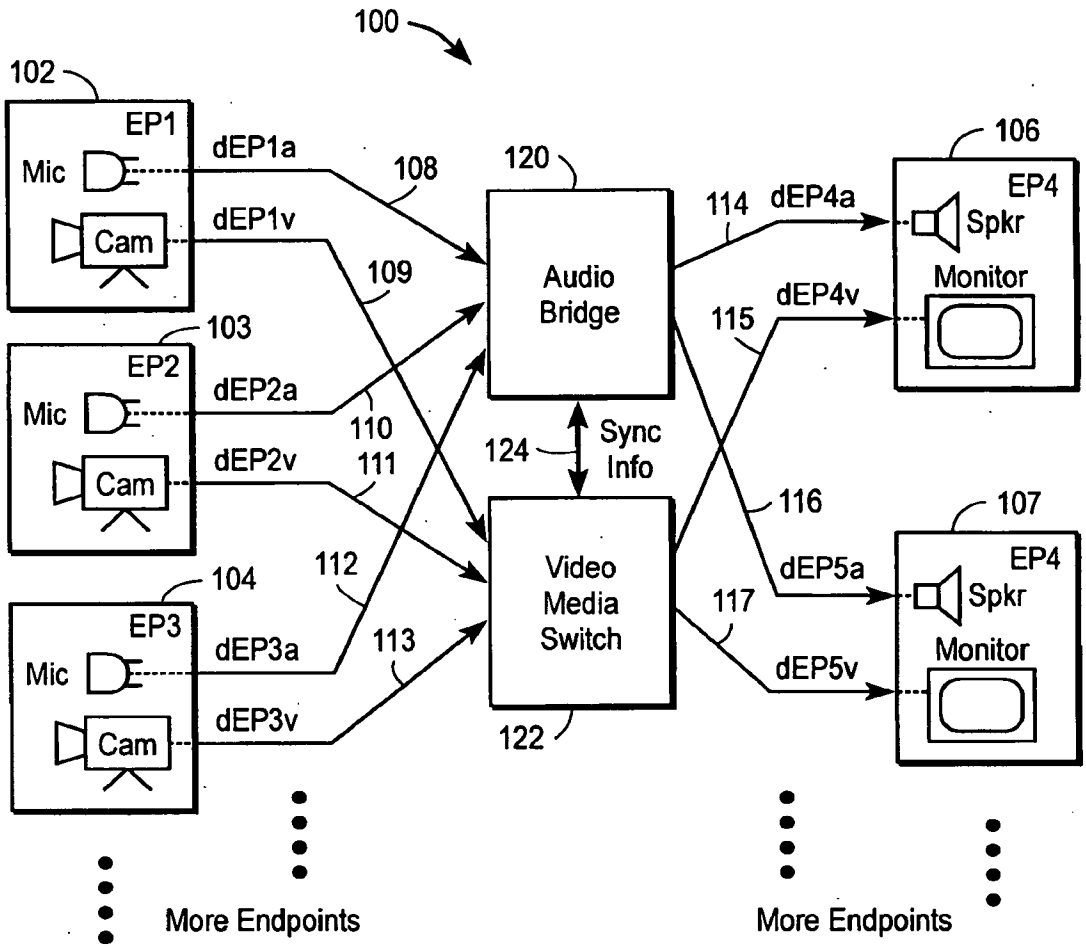
100
102
EP1
Mic
Cam
dEP1a
dEP1v
108
109
120
Audio Bridge
114
dEP4a
dEP4v
115
106
EP4
Spkr
Monitor
103
EP2
Mic
Cam
dEP2a
dEP2v
110
111
124
Sync Info
116
107
EP4
Spkr
Monitor
104
EP3
Mic
Cam
dEP3a
dEP3v
112
113
Video Media Switch
117
dEP5a
dEP5v
122
More Endpoints
More Endpoints

FIG. 1

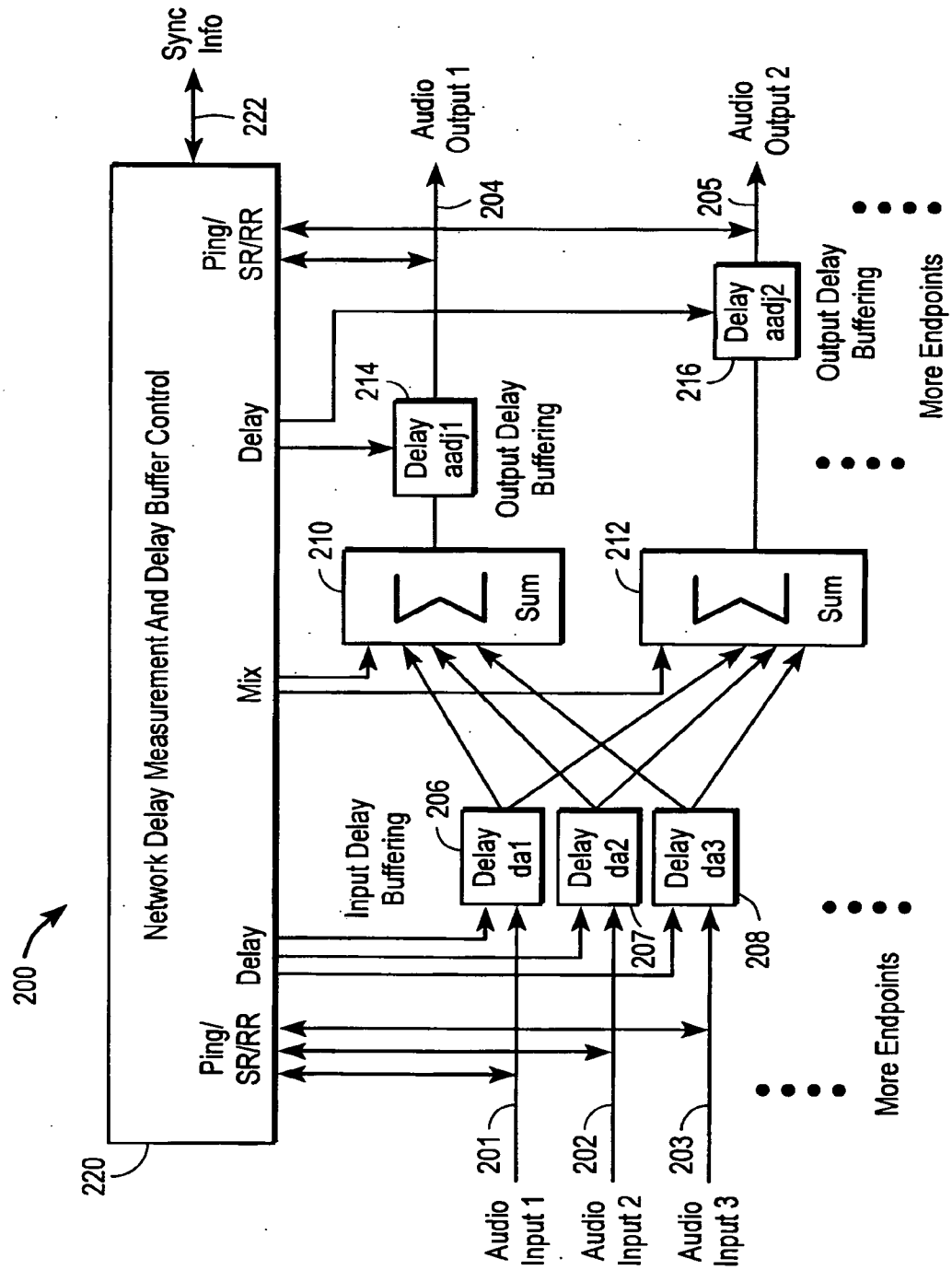
200
220
Network Delay Measurement And Delay Buffer Control
Ping/ SR/RR
Delay
Mix
Sync Info
222
Audio Output 1
204
Audio Output 2
205
Delay aadj1
214
Delay aadj2
216
Output Delay Buffering
More Endpoints
210
212
Sum
Input Delay Buffering
206
Delay da1
Delay da2
Delay da3
207
208
201
202
203
Audio Input 1
Audio Input 2
Audio Input 3

FIG. 2

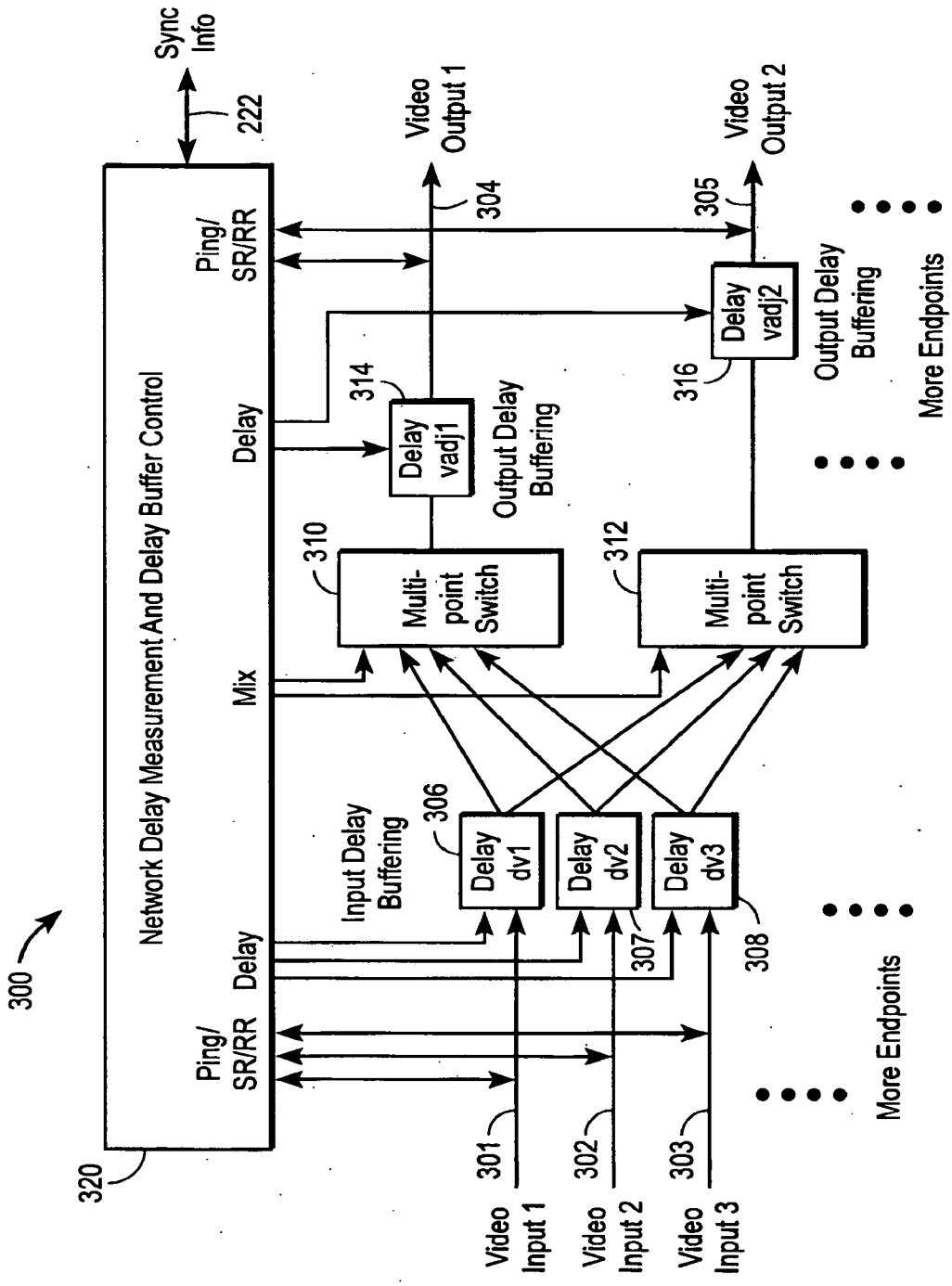
300
320
Network Delay Measurement And Delay Buffer Control
Ping/ SR/RR
Delay
Mix
Ping/ SR/RR
Delay
Sync Info
222
Video Input 1
Video Input 2
Video Input 3
301
302
303
Input Delay Buffering
306
Delay dv1
Delay dv2
Delay dv3
307
308
More Endpoints
Multi-point Switch
310
Multi-point Switch
312
Output Delay Buffering
314
Delay vadj1
Video Output 1
304
Video Output 2
305
Output Delay Buffering
316
Delay vadj2
More Endpoints

FIG. 3

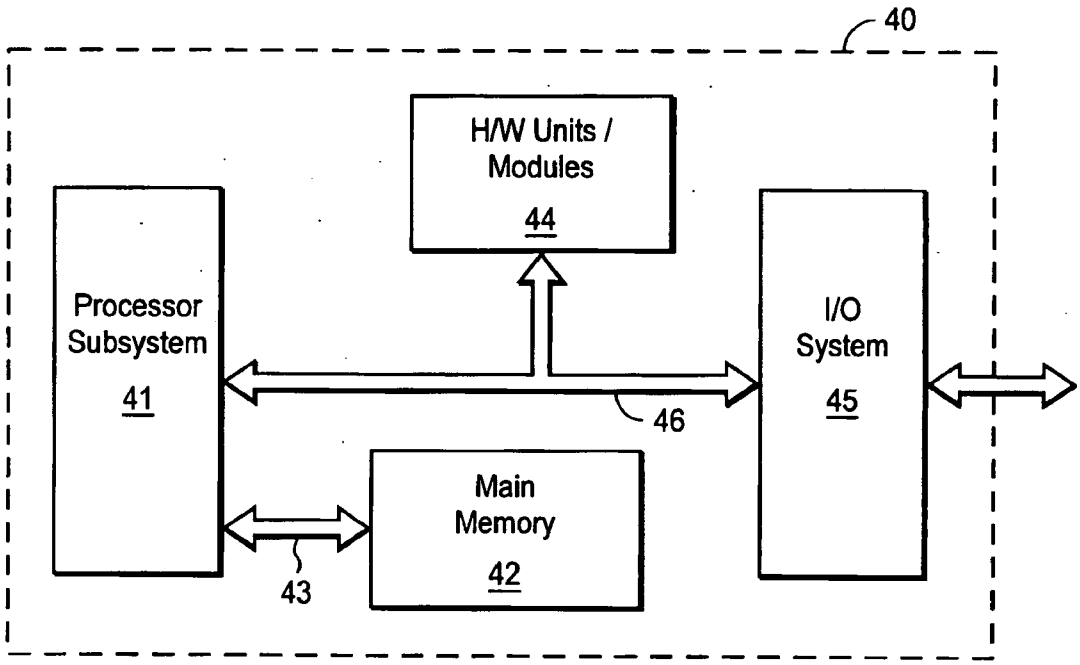
40
H/W Units / Modules
44
Processor Subsystem
41
I/O System
45
46
Main Memory
42
43

FIG. 4

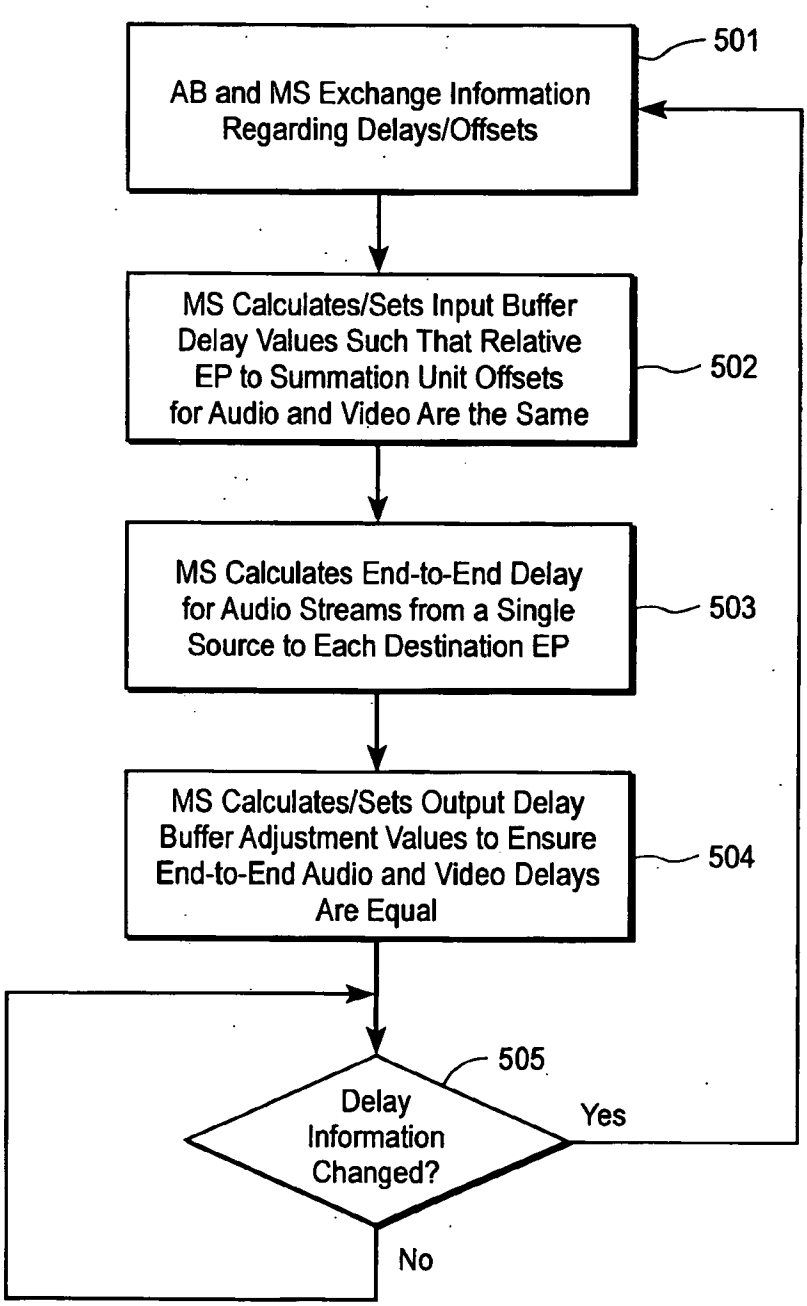
AB and MS Exchange Information Regarding Delays/Offsets
501
MS Calculates/Sets Input Buffer Delay Values Such That Relative EP to Summation Unit Offsets for Audio and Video Are the Same
502
MS Calculates End-to-End Delay for Audio Streams from a Single Source to Each Destination EP
503
MS Calculates/Sets Output Delay Buffer Adjustment Values to Ensure End-to-End Audio and Video Delays Are Equal
504
505
Delay Information Changed?
Yes
No

FIG. 5