

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
3 May 2007 (03.05.2007)

PCT

(10) International Publication Number  
WO 2007/051156 A2

(51) International Patent Classification: Not classified

(21) International Application Number:  
PCT/US2006/060289

(22) International Filing Date: 27 October 2006 (27.10.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/731,614 27 October 2005 (27.10.2005) US  
11/314,428 20 December 2005 (20.12.2005) US

(71) Applicant (for all designated States except US): QUALCOMM Incorporated [US/US]; 5775 Morehouse Drive, San Diego, California 92121 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): LEE, Yen-chi [CN/US]; 7675 Palmilla Drive, #6423, San Diego, California 92122 (US). EL-MALEH, Khaled, Helmi [CA/US]; 7689 Palmilla Drive, #1311, San Diego, California 92122 (US). TSAI, Ming-chang [US/US]; 10529 Whispering Hills Lane, San Diego, California 92130 (US).

(74) Agents: OGROD, Gregory, D. et al.; 5775 Morehouse Drive, San Diego, California 92121 (US).

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

AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, 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, LV, LY, MA, MD, 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, 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).

Declarations under Rule 4.17:

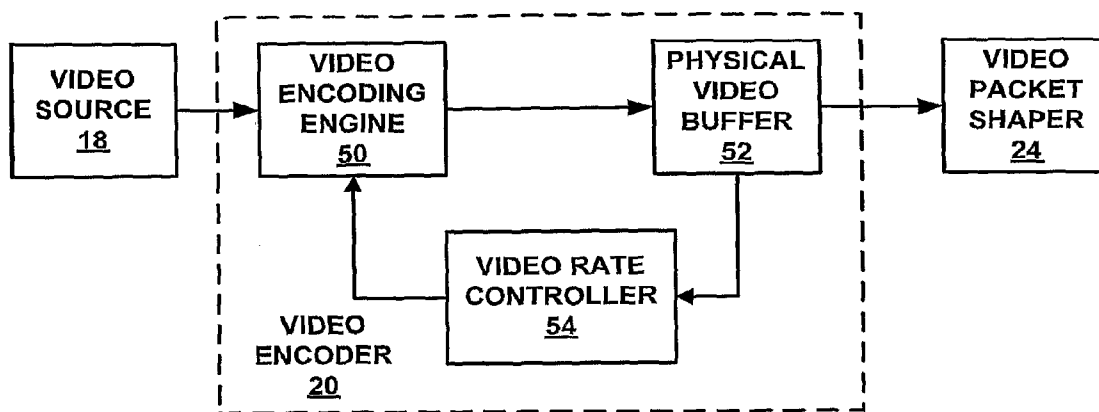
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

- without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: VIDEO SOURCE RATE CONTROL FOR VIDEO TELEPHONY



(57) Abstract: The disclosure relates to techniques for video source rate control for video telephony (VT) applications. The source video encoding rate may controlled using a dual-buffer based estimation of a frame budget that defines a number of encoding bits available for a frame of the video. The dual-buffer based estimation technique may track the fullness of a physical video buffer and the fullness of the virtual video buffer. The source video encoding rate is then controlled based on the resulting frame budget. The contents of the virtual buffer depend on constraints imposed by a target encoding rate, while the contents of the physical buffer depend on constraints imposed by varying channel conditions. Consideration of physical video buffer fullness permits the video source rate control technique to be channel-adaptive. Consideration of virtual video buffer fullness permits the video source rate control technique to avoid encoding excessive video that could overwhelm the channel.

WO 2007/051156 A2

## VIDEO SOURCE RATE CONTROL FOR VIDEO TELEPHONY

[0001] This application claims the benefit of U.S. provisional Application No. 60/731,614, filed October 27, 2005.

### TECHNICAL FIELD

[0002] The disclosure relates to video telephony (VT) and, more particularly, techniques for video source rate control in a VT system.

### BACKGROUND

[0003] Video telephony (VT) involves the real-time communication of packets carrying audio and video data. A VT device includes a video encoder that obtains video from a video capture device, such as a video camera or video archive, and generates video packets. Similarly, an audio encoder in a VT device obtains audio from an audio capture device, such as a microphone or speech synthesizer, and generates audio packets. The video packets and audio packets are placed in a radio link protocol (RLP) queue. A medium access control (MAC) layer module generates medium access control (MAC) layer packets from the contents of the RLP queue. The MAC layer packets are converted to physical (PHY) layer packets for transmission across a communication channel to another VT device.

[0004] In mobile VT applications, a VT device receives the physical layer packets via a wireless forward link (FL) (or “downlink”) from a base station to the VT device as a wireless terminal. A VT device transmits the PHY layer packets via a wireless reverse link (RL) (or “uplink”) to a base station. Each VT device includes PHY and MAC layers to convert the received PHY and MAC layer packets and reassemble the packet payloads into audio packets and video packets. A video decoder within the VT device decodes the video data for presentation to a user via a display device. An audio decoder within the VT device decodes the audio data for output via an audio speaker.

[0005] Mobile VT in a wireless environment can be challenging. The data rate over the wireless channel is limited and varies with time. For example, in a CDMA2000 1x EV-DO Release 0 network, the data rate may vary due to channel conditions within a wireless coverage area or traffic congestion among multiple VT users. In addition, when the data rate drops to zero, e.g., when there is no data to send, recovery to a

reasonable data rate may require time. As a result, mobile VT can be susceptible to undesirable video and audio delay, which undermines the ability to carry on smooth video conferencing in real-time.

### SUMMARY

**[0006]** In general, the disclosure is directed to techniques for video source rate control for VT applications. Channel conditions, excessive video content, or both can cause significant delays in transmission of video. When reverse link (RL) throughput is reduced, for example, video can overwhelm the RL and increase video transmission delay. The disclosure provides techniques for controlling the encoding rate of source video to reduce video delay over a range of channel conditions. In this sense, the video source rate control technique may be channel-adaptive. The techniques may be effective in reducing degradation of spatial and temporal quality when the video source encoding rate is reduced due to channel conditions or excessive video content or complexity.

**[0007]** In some embodiments, the source video encoding rate is controlled using a dual-buffer based estimation of a frame budget that defines a number of encoding bits available for a frame of the video. The dual-buffer based estimation technique may track the fullness of a physical video buffer and the fullness of the virtual video buffer relative to a threshold. The source video encoding rate is then controlled based on the resulting frame budget. The contents of the virtual video buffer depend on bit constraints imposed by a target encoding rate, while the contents of the physical video buffer depend on channel constraints imposed by varying channel conditions, e.g., RL throughput.

**[0008]** Consideration of physical video buffer fullness permits the video source rate control technique to be channel-adaptive. The amount of encoded video in the physical video buffer is, at least in part, a function of the rate at which the channel can accommodate additional video packets. Consideration of virtual video buffer fullness permits the video source rate control technique to avoid encoding excessive video that could overwhelm the channel.

**[0009]** The video source rate control technique may be used in conjunction with a channel-adaptive video packet shaping technique that controls the sizes of video packets extracted from the physical video buffer based on channel conditions. Channel

conditions influence the amount of video removed from the physical video buffer, which provides an indirect indication of channel conditions for use in formulating the frame budget. Once a frame budget is established, a video block budget also may be estimated in the rho domain to allocate a portion of the frame budget to individual video blocks within each frame.

[0010] In one embodiment, the disclosure provides a method comprising adding encoded video to a physical video buffer, removing a packet of the encoded video from the physical video buffer for transmission over a channel, maintaining a virtual video buffer representing an amount of the encoded video relative to a target encoding rate, and controlling an actual encoding rate of the video based on fullness of the physical video buffer and fullness of the virtual video buffer.

[0011] The disclosure also contemplates a computer-readable medium comprising instructions to cause one or more processors to perform such a method. For example, the disclosure also provides a computer-readable medium comprising instructions to cause one or more processors to add encoded video to a physical video buffer, remove a packet of the encoded video from the physical video buffer for transmission over a channel, maintain a virtual video buffer representing an amount of the encoded video relative to a target encoding rate, and control an actual encoding rate of the video based on fullness of the physical video buffer and fullness of the virtual video buffer.

[0012] In another embodiment, the disclosure provides a system comprising a video encoder, a physical video buffer that stores video encoded by the video encoder, a packet shaper that removes the encoded video from the physical video buffer for transmission over a channel, a virtual video buffer representing an amount of the encoded video relative to a target encoding rate, and a controller that controls an actual encoding rate of the video based on fullness of the physical video buffer and fullness of the virtual video buffer.

[0013] The details of one or more embodiments are set forth in the accompanying drawings and the description below. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

### **BRIEF DESCRIPTION OF DRAWINGS**

[0014] FIG. 1 is a block diagram illustrating an audio/video encoding and decoding system for VT applications.

[0015] FIG. 2 is a block diagram illustrating a video encoding system implementing video source rate control.

[0016] FIG. 3 is a block diagram illustrating an exemplary embodiment of the video encoding system of FIG. 2 with video source rate control using dual-buffer based frame budget estimation.

[0017] FIG. 4 is a block diagram illustrating integration of the video encoding system of FIG. 2 within an audio/video encoding system implementing channel-adaptive video packet shaping.

[0018] FIG. 5 is a flow diagram illustrating a technique for video source rate control.

[0019] FIG. 6 is a flow diagram illustrating a technique for video source rate control in greater detail.

## DETAILED DESCRIPTION

[0020] FIG. 1 is a block diagram illustrating a video encoding and decoding system 10. As shown in FIG. 1, system 10 includes an encoder system 12 and a decoder system 14 connected by a transmission channel 16. In the example of FIG. 1, encoder system 12 is associated with a first video communication device and includes an audio source 17, video source 18, video encoder 20, audio encoder 22, video packet shaper 24, real-time transport protocol (RTP)/user datagram protocol (UDP)/Internet protocol (IP)/point-to-point protocol (PPP) conversion module 26, radio link protocol (RLP) queue 28, MAC layer module 30 and physical (PHY) layer module 32. Decoder system 14 is associated with another video communication device and includes a PHY layer module 34, MAC layer module 36, RLP queue 38, RTP/UDP/IP/PPP conversion module 40, video decoder 42, audio decoder 44, audio output device 46 and video output device 48.

[0021] As will be described, video encoder 20 provides channel-adaptive video source rate control to reduce video delay under variable channel throughput conditions, such as those existing, e.g., in a CDMA2000 1x EV-DO Release 0 network. Reverse link (RL) throughput is a key factor in video delay within an EV-DO, Release 0 network. Video encoder 20 may control the video source encoding rate, at least in part, as a function of RL throughput. Video encoder 20 may be useful in an environment having low end-to-end delay requirements, e.g., less than 500 milliseconds. End-to-end delay refers to delay in transmission of video between a sender and a recipient, e.g., in a mobile wireless VT system. Channel-adaptive source rate control, as described in this

disclosure, may adaptively control end-to-end delay of video transmission while making a trade-off between spatial and temporal visual quality when operating with either constant bit rate (CBR) or variable bit rate (VBR) transmission channels, such as wireless channels.

**[0022]** Video encoder 20 may be effective in reducing degradation of spatial and temporal quality when the video source encoding rate is reduced due to channel conditions or excessive video content or complexity. Existing rate control schemes do not support desirable end-to-end delay with a reasonable trade-off between spatial and temporal video quality. The use of a small, fixed video buffer size, for example, can result in a large percentage of skipped frames, causing jerky motion and low spatial visual quality. Conversely, the use of a large video buffer size can produce better spatial quality, but tends to result in a larger, variable delay that can be disconcerting and annoying to an end user.

**[0023]** In general, the performance of video source encoding rate control can be evaluated by reference to end-to-end delay, which includes buffering and transmission delays, spatial visual quality, number of skipped video frames, encoder buffer underflow, which indicates bandwidth underutilization, encoder buffer overflow, which causes frame skipping, decoder buffer underflow, which indicates there is no data to decode and less display refresh, decoder buffer overflow, which indicates lost data, receiver display refresh rate, audio-video synchronization, encoder-side peak signal to noise ratio (PSNR), and initial buffer delay following a first intra (I) frame

**[0024]** Video encoder 20, in accordance with this disclosure, may employ a dual-buffer based scheme for estimation of a frame budget that defines a number of encoding bits available for a frame of the video. The dual-buffer based estimation technique may track the fullness of a physical video buffer and the fullness of a virtual video buffer relative to a threshold. The source video encoding rate is then controlled based on the resulting frame budget. The contents of the virtual video buffer depend on bit constraints imposed by a target encoding rate, while the contents of the physical video buffer depend on channel constraints imposed by varying channel conditions, e.g., reverse link (RL) throughput. In this manner, video encoder 20 may be capable of supporting a specified target encoding rate, and a specified end-to-end delay, with low frame skipping and good spatial quality.

[0025] In some embodiments, video encoder 20 may work in conjunction with a video packet shaper 24 that performs video packet shaping based on channel conditions. For example, video packet shaper 24 may packetize video with packet sizes based at least in part on channel conditions. In some embodiments, video packet shaper may prioritize audio packet transmission over video packet transmission in order to avoid excessive audio delays. Video packet shaper 24 may control the size of video packets and the rate at which video packets are removed from a physical video buffer for transmission over channel 16.

[0026] In turn, the size and rate of the video packets influences the fullness of the physical video buffer. For poor channel conditions, the sizes of video packets and the rate at which video packets are generated may be reduced by video packet shaper 24 to avoid overwhelming the RL. For better channel conditions, the sizes of video packets and the rate at which video packets are generated may be increased by video packet shaper 24. Video encoder 20 monitors the physical video buffer fullness, in conjunction with virtual video buffer fullness, to adaptively control encoding rate according to channel conditions and bit constraints.

[0027] System 10 may provide bi-directional video and audio transmission, e.g., for video telephony via transmission channel 16. Accordingly, generally reciprocal encoding, decoding, and conversion modules may be provided on opposite ends of channel 16. In some embodiments, encoder system 12 and decoder system 14 may be embodied within video communication devices such as wireless mobile terminals equipped for video streaming, video telephony, or both. The mobile terminals may support VT according to packet-switched standards such as RTP, UDP, IP, or PPP.

[0028] RTP/UDP/IP/PPP conversion module adds appropriate RTP/UDP/IP/PPP header data to audio and video data received from audio encoder 22 and video packet shaper 24, and places the data in RLP queue 28. RTP runs on top of UDP, while UDP runs on top of IP, and IP runs on top of PPP. MAC layer module 30 generates MAC RLP packets from the contents of RLP queue 28. PHY layer module 32 converts the MAC RLP packets into PHY layer packets for transmission over channel 16.

[0029] PHY layer module 34 and MAC layer module 36 of decoding system 14 operate in a reciprocal manner. PHY layer module 34 converts PHY layer packets received from channel 16 to MAC RLP packets. MAC layer module 36 places the MAC RLP packets into RLP queue 38. RTP/UDP/IP/PPP conversion module 40 strips the header

information from the data in RLP queue 38, and reassembles the video and audio data for delivery to video decoder 42 and audio decoder 44, respectively.

**[0030]** System 10 may be designed to support one or more wireless communication technologies such as code division multiple access (CDMA), frequency division multiple access (FDMA), time division multiple access (TDMA), or orthogonal frequency divisional multiplexing (OFDM), or another suitable wireless technique. The above wireless communication technologies may be delivered according to any of a variety of radio access technologies. For example, CDMA may be delivered according to cdma2000 or wideband CDMA (WCDMA) standards. TDMA may be delivered according to the Global System for Mobile Communications (GSM) standard. The Universal Mobile Telecommunication System (UMTS) standard permits GSM or WCDMA operation. Typically, for VT applications, system 10 will be designed to support high data rate (HDR) technologies such as cdma2000 1x EV-DO, Release 0, or subsequent EV-DO releases.

**[0031]** Video encoder 20 generates encoded video data according to a video compression method, such as MPEG-4. Other video compression methods may be used, such as the International Telecommunication Union (ITU) H.263, ITU H.264, or MPEG-2 methods. Audio encoder 22 encodes audio data to accompany the video data. Video source 18 may be a video capture device, such as one or more video cameras, one or more video archives, or a combination of video cameras and video archives.

**[0032]** The audio data may be encoded according to an audio compression method, such as adaptive multi-rate narrow band (AMR-NB), or other techniques. The audio source 17 may be an audio capture device, such as a microphone, or a speech synthesizer device. For VT applications, the video will permit viewing of a party to a VT conference and the audio will permit the speaking voice of that party to be heard.

**[0033]** In operation, RTP/UDP/IP/PPP conversion module 26 obtains video and audio data packets from video encoder 20 and audio encoder 22. As mentioned previously, RTP/UDP/IP/PPP conversion module 26 adds appropriate header information to the audio packets and inserts the resulting data within RLP queue 28. Likewise, RTP/UDP/IP/PPP conversion module 26 adds appropriate header information to the video packets and inserts the resulting data within RLP queue 28. MAC layer module 30 retrieves data from RLP queue 28 and forms MAC layer packets. Each MAC layer



packet carries RTP/UDP/IP/PPP header information and audio or video packet data that is contained within RLP queue 28.

**[0034]** Audio packets are inserted into RLP queue 28 independently of video packets. However, packet shaper 24 controls the sizes of video packets added to RLP queue 28 so that each audio packet can be carried by the next available MAC layer packet. In some cases, a MAC layer packet generated from the contents of RLP queue 28 will carry only header information and video packet data. In other cases, the MAC layer packet will carry only header information and audio packet data.

**[0035]** In many cases, the MAC layer packet will carry header information, audio packet data and video packet data, depending on the contents of RLP queue 28. The MAC layer packets may be configured according to a radio link protocol (RLP), and may be referred to as MAC RLP packets. PHY layer module 32 converts the MAC RLP audio-video packets into PHY layer packets for transmission across channel 16.

**[0036]** Channel 16 carries the PHY layer packets to decoder system 14. Channel 16 may be any physical connection between encoder system 12 and decoder system 14. For example, channel 16 may be a wired connection, such as a local or wide-area wired network. Alternatively, as described herein, channel 16 may be a wireless connection such as a cellular, satellite or optical connection. Channel conditions may be a concern for wired and wireless channels, but is especially problematic for mobile VT applications performed over a wireless channel 16, in which channel conditions may suffer due to fading or congestion.

**[0037]** In accordance with this disclosure, video encoder 20 encodes video from video source 18, and adds the encoded video to a physical video buffer. Video packet shaper 24 packetizes video content within the physical video buffer, and removes packets of the encoded video from the physical video buffer for transmission over channel 16. The rate at which video packet shaper 24 removes video packets from the physical video buffer, as well as the size of the packets, depends in part on channel conditions associated with channel 16. For example, channel 16 may be characterized by a reverse link (RL) having a throughput that varies according to channel conditions.

**[0038]** Video encoder 20 maintains a virtual video buffer representing an amount of the encoded video relative to a target encoding rate. The target encoding rate may be a maximum encoding rate specified for video packets transmitted over channel 16. Video encoder 20 controls an actual encoding rate of the video from video source 18 based on

fullness of the physical video buffer and fullness of the virtual video buffer.

Consideration of physical video buffer fullness permits the video source rate control applied by video encoder 20 to be channel-adaptive. The amount of video in the physical video buffer is, in part, a function of the rate at which the channel can accommodate additional video packets. Consideration of virtual video buffer fullness permits the video source rate control technique applied by video encoder 20 to avoid encoding excessive video that could overwhelm the channel, given a maximum target encoding rate.

[0039] Video packet shaper 24 controls the size of each video packet provided to RTP/UDP/IP/PPP conversion module 26. Video packet shaper 24 may be any packetizer that packetizes video in the physical video buffer and removes the video packets for transmission over channel 16. In some embodiments, however, the video source rate control technique may be used in conjunction with a channel-adaptive video packet shaping technique that controls the sizes of video packets extracted from the physical video buffer based in part on channel conditions, and prioritizes transmission of audio packets from audio encoder 22.

[0040] For example, video packet shaper 24 may size video packets so that each audio packet can be accommodated by the next available MAC layer packet. Controlled sizing of video packets prevents audio delays caused by channel conditions, large video packets, or both. When an audio packet is available, it is placed in the RLP queue for inclusion in the next available MAC RLP packet generated by MAC layer module 30. The audio packet may be combined with a video packet that has been sized to permit space for placement of the audio packet within the MAC RLP packet. By prioritizing audio, the audio portion of a VT conference can be conveyed without substantial delay, even though the video portion may suffer from delay due to channel conditions.

Although video may be compromised by channel conditions, video packet shaper 24 prioritizes audio to ensure that the parties to the VT conference are still able to smoothly carry on a verbal conversation.

[0041] Video packet shaper 24 may be configured to be channel-adaptive in the sense that it is capable of adjusting video packet size based on channel conditions. In this manner, encoder system 12 can prioritize transmission of audio packets to avoid audio delays when channel conditions are poor. At the same time, video packet shaper 24 can ensure that audio prioritization does not result in video packets being under-packetized.

In other words, video packet shaper 24 sizes video packets sufficiently small to permit inclusion of one or more audio packets in the next available MAC RLP packet, but not so small that excessive space in the MAC RLP packet is wasted. Consequently, video packet shaper 24 may support both prioritization of audio packets and efficient transmission of video packets. When channel adaptive video packet shaping is applied, the higher priority given to audio data can come at the cost of increased video delay. Video delay increases when the reverse link (RL) throughput is reduced. To avoid excessive video delay, in accordance with this disclosure, the video source rate is controlled as a function of channel throughput.

**[0042]** PHY layer module 34 of decoder system 14 identifies the MAC layer packets from the PHY layer packets and reassembles the content into MAC RLP packets. MAC layer module 36 then reassembles the contents of the MAC RLP packets to provide video and audio packets for insertion within RLP queue 38. RTP/UDP/IP/PPP module 40 removes the accompanying header information and provides video packets to video decoder 42 and audio packets to audio decoder 44. Video decoder 42 decodes the video data frames to produce a stream of video data for use in driving a display device. Audio decoder 44 decodes the audio data to produce audio information for presentation to a user, e.g., via an audio speaker.

**[0043]** As discussed above, video packet shaper 24 controls the size of video packets submitted to RTP/UDP/IP/PPP conversion module 26. Specifically, video packet shaper 24 may be configured to control the size of video packets in order to prioritize transmission of audio packets in MAC RLP packets, and prevent video packets from overwhelming RLP queue 28. In this manner, the audio portion of a VT conference can be conveyed without substantial delay, even though the video portion may suffer from delay due to channel conditions. Although video may be compromised by channel conditions, video packet shaper 24 ensures that the parties to the VT conference are still able to smoothly carry on a verbal conversation.

**[0044]** The packet shaping technique applied by video packet shaper 24 may apply one or more rules to ensure prioritized transmission of audio packets. According to one rule, for example, an audio packet should be sent in the very next available MAC RLP packet generated from the contents of RLP queue 28. Audio frames are generated by audio encoder 22 at first periodic intervals. MAC RLP packets are generated by MAC layer module 30 at second periodic intervals. The audio frame generated at a given

interval should be placed in the next available MAC RLP packet generated by MAC layer module 30. In some embodiments, as an option, the total output queue size of RLP queue 28 along with the audio packet size should be able to be carried in one MAC RLP packet.

**[0045]** Various rules may be applied with respect to every packet of a VT sequence. Although some video packets may be inherently sized in a manner that ensures that audio and video can be carried in a single MAC RLP packet, others video packets may be larger and require size reduction in order to ensure that audio and video can be carried in a single MAC RLP packet, particularly when channel conditions degrade. By applying the techniques with respect to every packet of a VT sequence, satisfactory speech communication can be ensured even if the content of the video is expansive or channel bandwidth is substantially limited.

**[0046]** Video packet shaper 24 controls the size of each video packet submitted to RTP/UDP/IP/PPP conversion module 26 for insertion in RLP queue 28. Video packet shaper 24 may ensure that audio packets are not delayed due to consumption of successive MAC RLP packets by expansive video content. Instead, when audio is available, video from video encoder 20 is divided into packets with sizes selected to permit each MAC RLP packet to carry audio and video. Each audio frame may be used as the audio packet provided to RLP queue 28. Alternatively, in some embodiments, an audio packet may bundle multiple audio frames provided at successive intervals.

**[0047]** Video packet shaper 24 may determine the video packet size for each MAC layer packet, in some embodiments, based on an estimated channel throughput for the MAC layer packets generated between successive audio frames. The throughput may be estimated based on channel conditions, as represented by one or more of current wireless channel transmit rate, wireless base station activity, and transmit power limitations. For example, the channel conditions may be determined based on current MAC layer data rate, a reverse activity (RA) bit, and a power amplifier (PA) limit. In accordance with this disclosure, video encoder 20 applies adaptive source rate control based in part on physical video buffer fullness. In this case, the source video encoding rate may be reduced by video encoder 20 if the channel conditions do not support the video encoding rate.

**[0048]** FIG. 2 is a block diagram illustrating video encoding system 10 implementing video source rate control. As shown in FIG. 2, video encoder 20 includes a video

encoding engine 50, physical video buffer 52 and video rate controller 54. Video encoding engine 50 obtains video data from video source 18 and encodes the video data at a rate controlled by video rate controller 54. Video encoding engine 50 then places the encoded video in physical video buffer 52. Video packet shaper 24 removes video data from video buffer 52 and packetizes the video data for transmission over channel 16. The size of the video packets may be, in part, a function of channel conditions on channel 16, such as RL throughput. Video rate controller 54 monitors the fullness of physical video buffer 52 and controls the video encoding rate applied by video encoding engine 50, at least in part, based on the fullness.

[0049] The encoding rate may be updated on a frame-by-frame basis according to the current fullness of physical video buffer 52. In addition, as will be described, video rate controller 54 may maintain a virtual video buffer. In this manner, video rate controller 54 may control the source video encoding rate using a dual-buffer based estimation of a frame budget that defines a number of encoding bits available for a frame of the video. The dual-buffer based estimation technique tracks the fullness of physical video buffer 52 and the fullness of the virtual video buffer relative to a threshold. Video rate controller 54 then controls the source video encoding rate based on the resulting frame budget.

[0050] The contents of the virtual video buffer depend on bit constraints imposed by a target encoding rate, while the contents of physical video buffer 52 depend on channel constraints imposed by varying channel conditions, e.g., RL throughput. The size of the virtual video buffer may be scaled according to RL throughput. Consideration of the fullness of physical video buffer 52 permits the video source rate control technique to be channel-adaptive. The amount of video in the physical video buffer 52 is, in part, a function of the rate at which channel 16 can accommodate additional video packets.

Consideration of virtual video buffer fullness permits the video rate controller 54 to avoid encoding excessive video that could overwhelm channel 16. The fullness of the virtual video buffer may be a function of a target encoding rate, e.g., including a target bit rate and a target frame rate. In effect, the virtual video buffer may be used to control frame bit allocation and/or frame skipping.

[0051] Video encoder 20 may provide a number of desirable features. For example, video encoder 20 may provide a video source rate control scheme that is generally CODEC-independent. For example, video encoder 20 may be adapted for video

encoding according to MPEG4, ITU H.263 or ITU H.264. In addition, video encoder 20 may be susceptible to simple implementation within a DSP or embedded logic core. In some embodiments, video encoder 20 may apply model-based rate control, e.g., applying video block rate control in the rho domain. For example, once a frame bit budget is established for a particular video frame, the frame bit budget may be allocated among the video blocks, e.g., macroblocks (MBs), within the frame using rho domain rate control. The rho domain values for individual MBs can then be mapped to quantization parameter (QP) values.

**[0052]** FIG. 3 is a block diagram illustrating an exemplary embodiment of video encoding system 10 of FIG. 2, with video source rate control using dual-buffer based frame budget estimation. As shown in FIG. 3, video encoding engine 50 obtains video from video source 18 and generates an encoded video data bitstream, e.g., according to MPEG4 or other encoding techniques. The encoded video is placed in physical video buffer 52. In the example of FIG. 3, video rate controller 54 includes a rate control unit 56, buffer information unit 58 and virtual video buffer 60. Buffer information unit 58 monitors fullness of physical video buffer 52, which is a function of the size of video packets removed from the physical video buffer by video packet shaper 24. The size of the video packets is, in turn, a function of channel constraints such as RL throughput. In addition, buffer information unit 58 monitors the fullness of virtual video buffer 60, which is a function of the amount of video encoded by video encoder 20, and bit-rate constraints such as a target encoding rate, including target bit rate and target frame rate.

**[0053]** Based on the fullness of physical video buffer 52 and the fullness of virtual video buffer 60, buffer information unit 58 generates a buffer value for analysis by rate control unit 56. In some embodiments, buffer information unit 58 may generate a maximum of the physical video buffer fullness and virtual video buffer fullness as the buffer value provided to rate control unit 56. The physical video buffer fullness or virtual video buffer fullness may be weighted by a coefficient to prioritize physical video buffer 52 over virtual video buffer 60, or vice versa. Rate control unit 56 may first compare the buffer value received from buffer information unit 58 to a skipping threshold  $T_{skip}$ .

**[0054]** If the buffer value meets or exceeds the skipping threshold  $T_{skip}$ , rate control unit 56 instructs video encoding engine 50 to skip encoding of the next video frame. In this case, there is no need to determine a frame budget because the next frame will not

be encoded. If the buffer value is less than the skipping threshold  $T_{skip}$ , then rate control unit 56 determines a frame budget to be used by video encoding engine 50 to encode the next frame. The frame budget defines the number of encoding bits allocated for encoding the next video frame.

**[0055]** In addition to estimating a frame bit budget, i.e., the number of bits for an entire frame, rate control unit 56 also may estimate a video block budget. The video block budget defines the number of encoding bits to be allocated to each video block, e.g., macroblock (MB), within a frame. The frame budget may be evenly allocated over the video blocks, or allocated based on a video block budget scheme. For example, the video block budget may apply an MB-level parametric rho-domain mapping of a frame and an MB bit budget to quantization parameters (QP) for each MB. In some embodiments, each MB QP may be post-processed to maintain a uniform intra-frame spatial quality. QPs of possible skipped MBs may be considered in order to produce a compliant MPEG-4 bitstream.

**[0056]** In some embodiments, physical video buffer 52 may be configured to drop video data if it resides within the buffer for a period of time that exceeds a threshold. For example, after a particular frame of video data has been placed in video buffer 52 by video encoding engine 50, it may remain in the physical video buffer for an extended period of time due to poor channel conditions. In this case, although more encoded video may be available from video encoding engine 50, there may be no available space in physical video buffer 52. To accommodate the newly encoded data, physical video buffer 52 may be configured to drop the oldest video data, e.g., on a first-in-first-dropped basis. Dropping the video data may simply involve discarding the video data and overwriting the buffer space previously occupied by the video data with newly encoded video data. The dropped data may be a frame of data or some predetermined amount of data, e.g., measured in bytes.

**[0057]** The video data may be dropped when it has been in physical video buffer 52 for a period of time that exceeds the threshold. Alternatively, criteria for dropping the video data from physical video buffer 52 may include both a time that exceeds the threshold and an indication from encoding engine 50 that additional video data is being encoded, which demands space within the physical video buffer. In either case, at least a portion of the video data in physical video buffer 52 is dropped to make room for the newly encoded video data. In conjunction with dropping the video data, physical video

buffer 52 may provide feedback to encoding engine 50 indicating that video data will be dropped. In response to the feedback, encoding engine 50 may apply early error control techniques in the encoding process to mitigate the effects of the dropped video data on video quality.

**[0058]** FIG. 4 is a block diagram illustrating integration of the video encoding system 10 of FIG. 2 within an audio/video encoding system implementing channel-adaptive video packet shaping. As mentioned previously, channel adaptive video source rate control may be used in conjunction with any video packetization scheme. In some applications, however, it may be desirable to apply channel adaptive video source rate control with a channel adaptive video packet shaping scheme, such as a video packet shaping scheme that prioritizes audio over video for transmission over channel 16, as described previously.

**[0059]** As shown in FIG. 4, video packet shaper 24 removes video data from physical video buffer 52 and packetizes the video data for delivery to RTP/UDP/IP/PPP conversion module 26. Again, the size of each packet may be determined according to channel conditions, as evidenced by channel information and RLP queue size received by video packet shaper 24. Audio encoder 22 includes an audio encoding engine 61 and an audio buffer/bundler 62. Audio encoding engine 61 encodes audio received from an audio source. Audio buffer/bundler 62 receives audio data from audio encoding engine 61 and generates audio frames for delivery to RTP/UDP/IP/PPP conversion module 26. RTP/UDP/IP/PPP conversion module 26 places MAC RLP packets within RLP queue 28 for transmission over channel 16.

**[0060]** A reverse traffic channel (RTC) MAC unit 64 implements an RTC MAC protocol to provide the procedures followed by a communication device to transmit over the reverse link (RL) of channel 16. For convenience, MAC layer module 30 and PHY layer module 32 are not shown in FIG. 4. Video packet shaper 24 controls the size of each video packet based on one or more inputs. The inputs may relate to channel conditions, RLP queue characteristics, and audio packet size and status. Video packet shaper 24 generates video packets with a size estimated based on the inputs. The video packet size may be subject to a minimum size.

**[0061]** Video buffer 52 buffers video information received from video encoder 20, and stores the video information for packetization by video packet shaper 24. Audio buffer 62 buffers audio frame information received from audio encoder 22 and passes the



information to RTP/UDP/IP/PPP conversion module 26. Audio and video packets are inserted in RLP queue 28 independently of one another. The size of the video packets produced by video packet shaper 24 may be controlled to ensure that there will be sufficient space for an audio packet in the next available MAC RLP packet produced by MAC layer module 30 (not shown in FIG. 4). In particular, RLP queue 28 is not overwhelmed with video packets, ensuring that the audio packet in the RLP queue can be sent with the next MAC RLP packet.

**[0062]** In the example of FIG. 4, video packet shaper 24 may receive several inputs, including an audio packet timer, an audio priority value, RLP queue size, and channel information. The audio packet timer indicates whether audio information is presently available in audio buffer 62 and, if so, the timing at which each audio frame will be delivered. If audio frames are delivered at intervals of every 20 ms, for example, the audio packet timer will be set to 20 ms when audio frames are available. In some embodiments, audio buffer 62 may be configured to bundle successive audio frames for incorporation in a single packet. In this case, the audio packet timer may be a multiple corresponding to the number of frames bundled into the audio packet. In other words, the audio packet timer may have a value that is proportional or otherwise related to the number of bundled frames. If three audio frames are bundled, for example, the audio timer may be set to 60 ms. Hence, the audio packet timer also indicates the size of the audio packet generated by audio buffer 62 for insertion in RLP queue 28 via RTP/UDP/IP/PPP conversion module 26.

**[0063]** The audio priority value defines the relative priorities of audio and video, and hence influences the delays associated with audio and video. For example, the audio priority value may be established such that the smaller the priority value, the lower the audio delay. Accordingly, as the audio priority value increases, audio delay increases and video delay decreases. Conversely, as the audio priority value decreases, audio delay decreases and video delay increases. Hence, audio delay tracks the audio priority value. Video packet shaper 24 may use the audio priority value to control the size of each video packet, resulting in a prescribed audio packet delay, as will be described in greater detail below.

**[0064]** The RLP queue size received by video packet shaper 24 represents the size of the current data buffered in RLP queue 28. Video packet shaper 24 uses the RLP queue size to control the size of the video packets. If RLP queue 28 is relatively full, video

packet shaper 24 may adjust the size of the video packets downward to avoid overwhelming the RL and causing excessive audio delay. If RLP queue 28 is less full, video packet shaper 24 may increase the size of the video packets while still providing sufficient space for audio packets. With RLP queue size, video packet shaper 24 is able to dynamically adjust video packet size as a function of the fullness of RLP queue 28. RLP queue fullness may indicate excessive video content, degradation of channel conditions, or both. The use of RLP queue size is one of the ways in which video packet shaper 24 can react to overloading of video content or changes in channel conditions.

**[0065]** Video packet shaper 24 also may react more directly to changes in channel conditions by monitoring channel information provided by reverse traffic channel (RTC) MAC unit 64. RTC MAC unit 64 generates information relating to channel characteristics, such as current MAC RL rate, combined RA bit, and headroom limitation. The MAC RL rate indicates the current transmission rate available over the RL. The RA bit is the reverse activity bit, which indicates whether the pertinent wireless base station is busy. The headroom limitation may indicate the maximum rate that is allowed to be used in transmission, based on the current transmit power. The RA bit indicates when the RL is congested or unavailable due to base station inactivity. The PA limit represents transmit power headroom and indicates when channel conditions have degraded.

**[0066]** Based on the various inputs, video packet shaper 24 generates a payload size estimate. The payload size estimate is selected to permit an audio packet to be included in the next available MAC RLP packet, if the audio priority value specifies that audio is to be accorded high priority. Video packet shaper 24 receives video from video buffer 52 and packetizes the video based on the payload size estimation and a minimum video packet size. The minimum video packet size represents the minimum size of video packets to be produced by video packet shaper 24. In effect, minimum video packet size controls the granularity of video packet size and bandwidth efficiency. For smaller minimum video packet size values, video packet shaping is more effective in terms of accommodating audio and thereby avoiding audio delays, but less bandwidth efficient. For larger minimum video packet size values, video packet shaping is less effective in avoiding audio delays, but provides greater bandwidth efficiency.

[0067] As further shown in FIG. 4, video encoder 20 is configured to include video rate controller 54, which responds to a physical video buffer fullness value from physical video buffer 52. Video encoder 20 provides an adaptive source rate control feature based on video buffer fullness. When video buffer 52 is relatively full, video rate controller 54 responds by reducing the video encoding rate applied by video encoding engine 50. When video buffer 52 is less full, video rate controller 54 increases the source video encoding rate. In this manner, the video encoding rate is reduced if channel conditions cannot support the current video encoding rate. As described herein, this channel-adaptive source rate control feature may be implemented as a dual-buffer frame budget estimation technique. Additional details of the dual-buffer frame budget estimation technique are described below.

[0068] In general, video rate controller 54 controls the video source encoding rate based on the status of physical video buffer 52 and a virtual video buffer maintained by the video rate controller. Physical video buffer 52 receives frames of video encoded by video encoding engine 50. Virtual video buffer 60 (FIG. 3) is updated on a frame by frame basis to represent an amount of encoded video relative to a target encoding rate. The target encoding rate may be characterized by a target bit rate R and a target frame rate F. An example of a target bit rate R is 24 Kilobits per second (Kbps). An example of a target frame rate is 7.5 frames per second (fps). A virtual buffer fullness value can be represented by the following expression:

$$W_n = \max(W_{n-1} + B_{n-1} - R/F, 0) \quad (1)$$

where  $W_n$  is the virtual buffer fullness at encoded video frame n,  $W_{n-1}$  is the virtual buffer fullness at frame n-1,  $B_{n-1}$  is the actual number of coding bits used by video encoding engine 50 to encode frame n-1, R is the target bit rate and F is the target frame rate. Expression (1) generates the maximum of either zero or  $W_{n-1} + B_{n-1} - R/F$ . If the value  $W_{n-1} + B_{n-1} - R/F$  is negative, the amount of encoded video is relatively low.

[0069] Buffer information unit 58 (FIG. 3) generates a buffer value  $X_n$  based on the fullness of physical video buffer 52 and virtual video buffer 60. For example, the buffer value  $X_n$  may be represented by the following expression:

$$X_n = \max(w * V_n, W_n) \quad (2)$$

where the coefficient  $w$  is a positive integer and  $V_n$  represents the fullness of physical video buffer 52 at frame  $n$ . The coefficient  $w$  is a weighting value that determines the weight to be accorded to physical video buffer fullness versus virtual video buffer fullness. Expression (2) produces the greater of  $w*V_n$  or  $W_n$  as the buffer value  $X_n$ . [0070] Rate control unit 56 uses the buffer value  $X_n$  to determine whether to skip the next frame. In particular, if the buffer value  $X_n$  is greater than or equal to the skipping threshold  $T_{skip}$  (i.e.,  $X_n \geq T_{skip}$ ), then rate control unit 56 instructs video encoding engine 50 to skip encoding of the next frame  $n + 1$ . When  $X_n$  meets or exceeds the skipping threshold  $T_{skip}$ , either video encoder 20 is in danger of exceeding its bit constraints, as evidenced by the virtual video buffer fullness, or the physical video buffer 52 is too full, e.g., due to channel conditions. For example, if channel conditions are poor, video packet shaper 24 may reduce the size of the video packets removed from the physical video buffer 52 and/or the rate at which the packets are removed. In this case, it is necessary to reduce the encoding rate applied by video encoder 20. Channel-adaptive video packet shaping provides an interface that bridges channel condition and video source rate control. The buffer value  $X_n$  transforms the channel condition information into video buffer fullness, which provides an indirect indication of channel throughput.

[0071] If the buffer value  $X_n$  does not meet or exceed the skipping threshold  $T_{skip}$  (i.e.,  $X_n < T_{skip}$ ), rate control unit 56 (FIG. 3) determines a frame budget defining an number of bits allocated to encode the next frame. Rate control unit 56 may determine the frame budget  $B_n$  for frame  $n$  according to the following expression:

$$B_n = R/F - \Delta \quad (3)$$

where  $\Delta = X_n/F$  if  $X_n > Z*T_{skip}$ ,  $\Delta = X_n - Z*T_{skip}$  if  $X_n \leq Z*M$ , and  $Z$  is a coefficient. In general, the delta value  $\Delta$  is a function of buffer fullness ( $W_n$  or  $V_n$ ), frame rate ( $F$ ), and channel throughput ( $R$ ). The value  $R$ , the target bit rate, represents channel throughput. The value  $F$ , the target frame rate, represents the rate at which frames can be transmitted. The value  $R/F$ , i.e., channel throughput  $R$  divided by frame rate, represents the channel throughput per frame.

[0072] If the buffer value is greater than  $Z$  times the skipping threshold  $T_{\text{skip}}$ , the virtual or physical buffer, as applicable, is relatively full. In this case, the buffer value  $X_n$  is divided by the frame rate  $F$ , and the result is subtracted from  $R/F$ , which is the target bit rate divided by the target frame rate. In general, this value of  $\Delta$  ( $= X_n/F$ ) will result in a reduction in the encoding rate applied by video encoding engine 50. If the buffer value is less than or equal to  $Z$  times the skipping threshold  $T_{\text{skip}}$ , the applicable virtual or physical buffer has sufficient space to accommodate more encoded video. In this case, the buffer value the product of  $Z$  times the skipping threshold  $T_{\text{skip}}$  is subtracted from the buffer value  $X_n$ . In general, this value of  $\Delta$  ( $= X_n - Z * T_{\text{skip}}$ ) will result in an increase in the encoding rate applied by video encoding engine 50.

[0073] As an illustration, in one implementation, the following requirements may be applied to the video source rate control applied by video encoder 20. If channel bit rate  $R$  is 48 Kbps (6000 bytes/second) and frame rate is 15 fps, the corresponding frame budget is  $480000/15 = 3200$  bits = 400 bytes. Assuming a 0.4 to 0.6 second maximum end-to-end delay requirement (e.g., 400 to 600 ms, and more particularly 500 ms), the physical video buffer delay requirement is less than or equal to 200 milliseconds (ms). In this case, the size of virtual video buffer 60 can be set to 1200 bytes (200 ms \* 6000 bytes/second). To meet the above requirements, the video source rate control design should have a physical video buffer fullness that is less than or equal to 1200 bytes for every frame. In addition, the encoder buffering delay in the physical video buffer 52 should be less than or equal to 200 ms for every frame.

[0074] Video encoder 20 reduces the video encoding rate when RL throughput decreases, but gracefully adapts to RL throughput to prevent over- or under-reaction to the RL throughput. Ideally, there should be no buffer underflow, or minimal buffer underflow, as well as no buffer overflow or minimal buffer overflow, in physical video buffer 52. Buffer underflow indicates underutilization of bandwidth, while buffer overflow results in skipped frames, i.e., skipping the encoding of frames when there is insufficient space in physical video buffer 52 to accommodate the encoded video frame. With the dual buffer based rate control scheme described herein, end-to-end video delay can be reduced while enhancing spatial quality. In this manner, a good balance can be achieved to substantially maximize encoding quality at a given amount of bandwidth. For example, it is possible to meet a strict low end-to-end delay requirement, e.g., less than or equal to 500 ms, with low frame skipping and no or minimal buffer underflow.

[0075] The virtual video buffer 60 helps meet the delay requirement by using buffer-based bit allocation and frame skipping. The physical video buffer 52 permits adaptation of the video encoding rate to channel conditions. According to expressions (1)-(3) above, the virtual video buffer fullness value  $W_n$  is used for rate control when channel conditions are good. The physical video buffer fullness value  $V_n$  from the RTP layer is used to reduce the encoding rate when the channel condition is poor. In this manner, it is possible to meet an allocated frame budget under varying channel conditions due to time-varying channel throughput. MB-level rate control serves as another feature to improve the accuracy of rate control. With such features, overall end-to-end visual quality can be enhanced for a given bit rate, supporting more effective audio-video communication, e.g., in a VT application or in other applications.

[0076] FIG. 5 is a flow diagram illustrating a technique for video source rate control. In the example of FIG. 5, video encoding engine 50 encodes a video frame  $n-1$  (70) and adds the encoded video frame to physical video buffer 52 (72). Rate control unit 56 updates the virtual buffer 74 for the next frame  $n$ , e.g., according to expression (1), yielding a new virtual buffer fullness value  $W_n$  (74). In updating the virtual video buffer,  $B_{n-1}$  is the amount of bits used by video encoding engine 50 to encode the frame  $n-1$  added to physical video buffer 52. Buffer information unit 58 determines the physical video buffer fullness value  $V_n$  (76) and the virtual video buffer fullness value  $W_n$  (78). Rate control unit 56 controls the video encoding rate of video encoding engine 50 based on the physical video buffer fullness value  $V_n$  and the virtual video buffer fullness value  $W_n$  (80). Then, the process proceeds to the next frame to be encoded (82), and repeats.

[0077] FIG. 6 is a flow diagram illustrating a technique for video source rate control in greater detail. In the example of FIG. 6, video encoding engine 50 encodes a video frame  $n-1$  (84) and adds the encoded video frame to physical video buffer 52 (86). Rate control unit 56 updates the virtual buffer 74 for the next frame  $n$ , e.g., according to expression (1), yielding a new virtual buffer fullness value  $W_n$  (88). Buffer information unit 58 determines the physical video buffer fullness value  $V_n$  (90) and the virtual video buffer fullness value  $W_n$  (92). Rate control unit 56 receives a buffer value  $X_n$  from buffer information unit 58. The buffer value  $X_n = \max(w * V_n, W_n)$ , where  $w$  is a weighting coefficient. This corresponds to expression (2) above. If  $\max(w * V_n, W_n)$  is greater than or equal to the skipping threshold  $T_{skip}$  (94), rate control unit 56 instructs

video encoding engine 50 to skip the encoding of the next frame  $n$  (96). Upon skipping the next frame  $n$ , the process proceeds to the next frame after the skipped frame (101), and repeats.

**[0078]** If  $\max(w \cdot V_n, W_n)$  is not greater than or equal to the skipping threshold  $T_{\text{skip}}$  (94), rate control unit 56 estimates a frame budget for the next frame based on the target encoding rate, e.g., target bit rate  $R$  and target frame rate  $F$ , the physical video buffer fullness value  $V_n$ , the virtual video buffer fullness value  $W_n$ , and the skipping threshold  $T_{\text{skip}}$  (98). In particular, rate control unit 56 may apply expression (3) above to determine the frame budget  $B_n$ , which defines the number of bits available to encode the next frame. Upon determining the frame budget, rate control unit 56 may direct video encoding engine 50 to allocate the bits in the frame budget evenly over the video blocks, e.g., macroblocks (MBs), within the frame to be encoded. Alternatively, the frame budget bits may be allocated using a rho domain MB bit allocation (99). The rho domain MB bit allocation for each MB may then be mapped to a corresponding QP value (100). Once the encoding rate is determined for the current frame, the process proceeds to the next frame (101), and repeats.

**[0079]** The techniques described in this disclosure may be implemented within a general purpose microprocessor, digital signal processor (DSP), application specific integrated circuit (ASIC), field programmable gate array (FPGA), or other equivalent logic devices. For example, video encoder system 12, and its components and modules, may be implemented as parts of an encoding process, or coding/decoding (CODEC) process, running on a digital signal processor (DSP), microprocessor, embedded core, or other processing device. Accordingly, components described as modules may form hardware components or programmable features of an encoding process, or a separate process.

**[0080]** In some embodiments, encoding functions may be divided among different hardware components. For example, frame-level rate control may be performed in an embedded logic core, and MB-level rate control may be performed in a DSP. As an illustration, given a target bit rate ( $R$  Kbps) and a frame rate ( $F$  fps), frame-level rate control within the embedded logic core may involve updating rate control model parameters, e.g., rho domain model parameters, after encoding each frame, estimating the frame budget  $B$  for the next frame, and mapping the frame budget to a frame QP

(e.g., 1 to 31) using budget-to-rho and rho-to-QP mappings, e.g., via either a rho table or a rho parametric equation.

**[0081]** Upon post-processing of the QP values, including any additional constraints on frame QP, the embedded logic core sends the frame QP, rho budget and new model parameters to the DSP. The DSP then calculates the QP for each MB using the rho-to-QP mapping, and performs post-processing of the QP values. The DSP may preserve a rule that the MB delta QP value is within +2 and -2, as well as any additional constraints on MB QPs. Upon updating the rho domain model parameters after encoding a MB, the DSP repeats the process for the other MBs within the applicable video frame. After MB encoding is completed, the process returns to the embedded logic core to handle the next video frame to be encoded.

**[0082]** Video encoder system 12 may have a dedicated memory for storing instructions and data, as well as dedicated hardware, software, firmware, or combinations thereof. If implemented in software, the techniques may be embodied as instructions on a computer-readable medium such as random access memory (RAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, or the like. The instructions cause one or more processors to perform certain aspects of the functionality described in this disclosure.

**[0083]** Additional details of an exemplary video packet shaper 24 will now be described. The following description provides an example of a video packet shaping technique that may be useful with the video source rate control techniques described in this disclosure. However, techniques for video source rate control in accordance with this disclosure may be practiced with any of a variety of packet shaping techniques. Accordingly, the following details are provided for purposes of example and should not be considered limiting of the video source rate control techniques as broadly embodied and described in this disclosure.

**[0084]** In this example, video packet shaper 24 controls the size of each video packet provided to RTP/UDP/IP/PPP conversion module 26 (FIGS. 1 and 4) in order to prioritize transmission of audio. In particular, video packets are sized so that each audio packet can be accommodated by the next available MAC layer packet. Controlled sizing of video packets prevents audio delays caused by channel conditions, large video packets, or both. When an audio packet is available, it is placed in the RLP queue for



inclusion in the next available MAC RLP packet generated by MAC layer module 30. The audio packet may be combined with a video packet that has been sized to permit space for placement of the audio packet within the MAC RLP packet.

**[0085]** Video packet shaper 24 is configured to be channel-adaptive in the sense that it is capable of adjusting video packet size based on channel conditions. In this manner, encoder system 12 can prioritize transmission of audio packets to avoid audio delays when channel conditions are poor. At the same time, video packet shaper 24 can ensure that audio prioritization does not result in video packets being under-packetized. In other words, video packet shaper 24 sizes video packets sufficiently small to permit inclusion of one or more audio packets in the next available MAC RLP packet, but not so small that excessive space in the MAC RLP packet is wasted. Consequently, video packet shaper 24 may support both prioritization of audio packets and efficient transmission of video packets.

**[0086]** As discussed above, video packet shaper 24 is provided to control the size of video packets submitted to RTP/UDP/IP/PPP conversion module 26. Video packet shaper 24 controls the size of video packets to prioritize transmission of audio packets in MAC RLP packets, and prevent video packets from overwhelming RLP queue 28. In this manner, the audio portion of a VT conference can be conveyed without substantial delay, even though the video portion may suffer from delay due to channel conditions. Although video may be compromised by channel conditions, video packet shaper 24 ensures that the parties to the VT conference are still able to smoothly carry on a verbal conversation.

**[0087]** The packet shaping technique applied by video packet shaper 24 may apply one or more rules to ensure prioritized transmission of audio packets. According to one rule, for example, an audio packet should be sent in the very next available MAC RLP packet generated from the contents of RLP queue 28. Audio frames are generated by audio encoder 22 at first periodic intervals. MAC RLP packets are generated by MAC layer module 30 at second periodic intervals. The audio frame generated at a given interval should be placed in the next available MAC RLP packet generated by MAC layer module 30. In some embodiments, as an option, the total output queue size of RLP queue 28 along with the audio packet size should be able to be carried in one MAC RLP packet.

[0088] Various rules may be applied with respect to every packet of a VT sequence. Although some video packets may be inherently sized in a manner that ensures that audio and video can be carried in a single MAC RLP packet, other video packets may be larger and require size reduction in order to ensure that audio and video can be carried in a single MAC RLP packet, particularly when channel conditions degrade. By applying the techniques with respect to every packet of a VT sequence, satisfactory speech communication can be ensured even if the content of the video is expansive or channel bandwidth is substantially limited.

[0089] The size of each video packet submitted to RTP/UDP/IP/PPP conversion module 26 by packet shaper 24 for insertion in RLP queue 28 is controlled. The above rule ensures that audio packets are not delayed due to consumption of successive MAC RLP packets by expansive video content. Instead, when audio is available, video from video encoder 20 is divided into packets with sizes selected to permit each MAC RLP packet to carry audio and video. Each audio frame may be used as the audio packet provided to RLP queue 28. Alternatively, in some embodiments, an audio packet may bundle multiple audio frames provided at successive intervals.

[0090] Video packet shaper 24 may determine the video packet size for each MAC layer packet, in some embodiments, based on an estimated channel throughput for the MAC layer packets generated between successive audio frames. The throughput may be estimated based on channel conditions, as represented by one or more of current wireless channel transmit rate, wireless base station activity, and transmit power limitations. For example, the channel conditions may be determined based on current MAC layer data rate, a reverse activity (RA) bit, and a power amplifier (PA) limit. In addition, in some embodiments, video encoder 20 may further include adaptive source rate control based on video buffer occupancy. In this case, the source video encoding rate may be reduced by video encoder 20 if the channel conditions do not support the video encoding rate, given the need for prioritized audio packetization.

[0091] Audio encoder 22 generates audio frames, and video encoder 20 generates video frames. A series of successive MAC RLP packets are available to carry audio packets and video packets derived from the frames, which are buffered in RLP queue 28. Following generation of a first audio frame by audio encoder 22, the next available MAC RLP packet generated by MAC layer module 30 carries the first audio frame. A second packet also can be used to carry the first audio frame, if necessary. If the

contents of RLP queue 28 is overwhelmed by video packets, however, the first audio frame may not be delivered for a long period of time.

[0092] Each MAC RLP packet has an associated data rate derived from RL channel condition information. Under good RL conditions, each MAC RLP packet may carry a data rate of 76.8 Kilobits per second (Kbps). Under poor RL channel conditions, however, data rate fluctuates and is often low, e.g., 19.2 Kbps or 38.4 Kbps. Excessive video content, poor channel conditions, or both can cause significant delays in transmission of audio packets. Excessive video packet size can overwhelm the RL and increase audio delay, particularly when RL throughput is reduced due to low data rates.

[0093] A video packet, if left uncontrolled, may consume an excessive amount of the MAC RLP packet space, resulting in delays between successive transmissions of audio packets. In some cases, video may consume several consecutive MAC RLP packets, preventing audio from being transmitted promptly. Each MAC RLP packet provides roughly 26.67 ms of space for incorporation of audio and video packet information. Consider the case in which a large video frame is generated at substantially the same time as an audio frame. In this scenario, assume that successive video frames are generated 133 ms from one another in time. However, successive audio frames are generated only 60 ms from one another in time.

[0094] Even under good RL conditions, there may be insufficient space for incorporation of audio packets for the first audio frame, as well as for second and third audio frames. Instead, video packets associated with the large video frame may consume most of MAC RLP packets, resulting in significant audio delays. This problem is especially challenging when channel condition degrades, as in the case of Poor RL condition. To alleviate audio delays under a variety of channel conditions, system 10 of FIG. 1 may incorporate video packet shaper 24, which controls the size of video packets derived from video frames. By applying video limits with respect to every packet of a VT sequence, video packet shaper 24 can ensure that the audio associated with the VT sequence is not compromised.

[0095] Fixed length video packet shaping presents a partial solution to the problem of audio delay. However, fixed length video packet shaping does not consider channel conditions. Consequently, video can still overwhelm the channel when RL throughput is reduced. In addition, fixed length packet shaping does not consider the throughput

between two successive audio packets, resulting in over- or under-packetization of video data.

**[0096]** For fixed length video packet shaping, the size of video packets is controlled by fragmenting video frames into fixed size packets (e.g., 300-bytes) every 60 ms. Audio frames are also fragmented into fixed-size packets, e.g., every 60 ms. Video packets are transmitted immediately after audio data packets within MAC RLP packets. Under normal operating conditions, fixed length video packetization promotes the timely transmission of audio packets within MAC RLP packets.

**[0097]** The fixed length video packet shaping approach ensures that audio packets are transmitted without delay under good RL conditions. If RL conditions degrade, however, the fixed length video packets can overwhelm the RL, resulting in delays between successive audio packets. Due to the fixed length of video packets, there is no ability to react to changes in RL conditions. In some instances, under good RL conditions, the video data may be underpacketized, resulting in underutilization of the space provided by each MAC RLP packet, and general bandwidth inefficiency. Under poor RL conditions, the fixed size of the video packet may be too large for the RL to handle, resulting in audio delay. For this reason, channel-adaptive video packet shaping may be provided to produce adjustable length video packets, which have sizes that are adaptable in response to video content or bandwidth in order to maintain quality audio for an entire VT sequence.

**[0098]** For channel-adaptive video packet shaping, video packet size is adjusted based on channel conditions so that audio packets can be transmitted without substantial delay. Instead of a fixed video packet size, the size of each video packet is dynamically adjusted based on the size of audio packets and channel conditions. Under good RL conditions, video packet size may be increased, but not to the point that the video packets would overwhelm the RL and introduce audio delay. Under poor RL conditions, video packets are reduced in size to provide room for an audio frame to be packetized and placed in the next available MAC RLP packet.

**[0099]** When audio frames are available, video frames are sized so that the respective audio frame can be placed in the next available MAC RLP packet. Each audio frame is packetized and then placed in RLP queue 28 for inclusion in the next available MAC RLP packet generated by MAC layer module 30, eliminating excessive delay between transmission of audio packets. Each MAC RLP packet may carry only audio, only

video, or both audio and video, depending on the contents of RLP queue 28. However, at each audio packet interval, a video packet is sized to permit incorporation of the audio packet in the next available MAC RLP packet.

**[00100]** Notably, as the available RL rate is reduced, e.g., due to channel conditions, the size of the audio packet increases relative to the size of the MAC RLP packet. In other words, each audio packet consumes a greater proportion of the MAC RLP packet as RL rate decreases because video packet size is reduced. Conversely, the size of each video packet is dynamically adjusted so that it consumes a smaller proportion of the MAC RLP packet as RL rate decreases. In this way, video packets are sized to permit placement of each audio packet within the next available MAC RLP packet. The result is that audio is given higher priority than video to reduce audio delay.

**[00101]** RTC MAC unit 64 (FIG. 4) may be configured to implement an RTC MAC protocol to provide the procedures followed by a communication device to transmit over the RL. In a particular embodiment, video packet shaper 24 may include a payload size estimator that controls the size of each video packet based on one or more inputs. The inputs may relate to channel conditions, RLP queue characteristics, and audio packet size and status. Video packet shaper 24 also may include, in a particular embodiment, a bandwidth efficient packetizer that generates video packets based on an estimated payload size specified by the payload size estimator, subject to a minimum video packet size.

**[00102]** Video buffer 52 buffers video information received from video encoder 20, and passes the video information to video packet shaper 24. Audio buffer 62 buffers audio frame information received from audio encoder 22 and passes the information to RTP/UDP/IP/PPP conversion module 26. Audio and video packets are inserted in RLP queue 28 independently of one another. The size of the video packets produced by video packet shaper 24 ensures that there will be sufficient space for the audio packet in the next available MAC RLP packet produced by MAC layer module 30. In particular, RLP queue 28 is not overwhelmed with video packets, ensuring that the audio packet in the RLP queue can be sent with the next MAC RLP packet.

**[00103]** In this example, the payload size estimator of video packet shaper 24 receives several inputs, including an audio packet timer, an audio priority value ACPredNumberPlus, RLP queue size, and channel information. The audio packet timer indicates whether audio information is presently available in audio buffer 62 and, if so,

the timing at which each audio frame will be delivered. If audio frames are delivered at intervals of every 20 ms, for example, the audio packet timer will be set to 20 ms when audio frames are available. In some embodiments, audio buffer 62 may be configured to bundle successive audio frames for incorporation in a single packet. In this case, the audio packet timer may be a multiple corresponding to the number of frames bundled into the audio packet. In other words, the audio packet timer may have a value that is proportional or otherwise related to the number of bundled frames. If three audio frames are bundled, for example, the audio timer may be set to 60 ms. Hence, the audio packet timer also indicates the size of the audio packet generated by audio buffer 62 for insertion in RLP queue 28 via RTP/UDP/IP/PPP module 26.

**[00104]** The audio priority value MACPredNumberPlus defines the relative priorities of audio and video, and hence influences the delays associated with audio and video. For example, MACPredNumberPlus is established such that the smaller the priority value, the lower the audio delay. Accordingly, as MACPredNumberPlus increases, audio delay increases and video delay decreases. Conversely, as the MACPredNumberPlus decreases, audio delay decreases and video delay increases. Hence, audio delay tracks the audio priority value MACPredNumberPlus. The payload size estimator of packet shaper 24 uses the MACPredNumberPlus value to control the size of each video packet, resulting in a prescribed audio packet delay, as will be described in greater detail below.

**[00105]** The RLP queue size received by the payload size estimator represents the size of the current data buffered in RLP queue 28. The payload size estimator uses the RLP queue size to control the size of the video packets. If RLP queue 28 is relatively full, the payload size estimator may adjust the size of the video packets downward to avoid overwhelming the RL and causing excessive audio delay. If RLP queue 28 is less full, the payload size estimator may increase the size of the video packets while still providing sufficient space for audio packets. With RLP queue size, the payload size estimator is able to dynamically adjust video packet size as a function of the fullness of RLP queue 28. Queue fullness may indicate excessive video content, degradation of channel conditions, or both. The use of RLP queue size is one of the ways in which the payload size estimator can react to overloading of video content or changes in channel conditions.

**[00106]** The payload size estimator of video packet shaper 24 also may react more directly to changes in channel conditions by monitoring channel information provided by RTC MAC unit 64. RTC MAC unit 64 generates information relating to channel characteristics, such as current MAC RL rate, combined RA bit, and headroom limitation. The MAC RL rate indicates the current transmission rate available over the RL. The RA bit is the reverse activity bit, which indicates whether the pertinent wireless base station is busy. The headroom limitation may indicate the maximum rate that is allowed to be used in transmission, based on the current transmit power. The RA bit indicates when the RL is congested or unavailable due to base station inactivity. The PA limit represents transmit power headroom and indicates when channel conditions have degraded.

**[00107]** Based on the various inputs, the payload size estimator of video packet shaper 24 generates a payload size estimate. The payload size estimate is selected to permit an audio packet to be included in the next available MAC RLP packet, if the MACPredNumPlus priority value specifies that audio is to be accorded high priority. The bandwidth efficient packetizer of video packet shaper 24 receives video from video buffer 52 and packetizes the video based on the payload size estimation specified by the payload size estimator and a minimum video packet size. The minimum video packet size represents the minimum size of video packets to be produced by video packet shaper 24. In effect, minimum video packet size controls the granularity of video packet size and bandwidth efficiency. For smaller minimum video packet size values, video packet shaping is more effective in terms of accommodating audio and thereby avoiding audio delays, but less bandwidth efficient. For larger minimum video packet size values, video packet shaping is less effective in avoiding audio delays, but provides greater bandwidth efficiency.

**[00108]** Additional implementation details will be described for purposes of illustration. Again, such details should be considered exemplary, and not limiting of the techniques broadly embodied and described in this disclosure. For a cdma2000 1x EV-DO Rel. 0 implementation, RL throughput can be estimated based on channel conditions. 3GPP2 Specification C.S0024-A (also referred to as TIA IS-856-A), at page 11-143, Table 11.9.6.1 specifies minimum and maximum payload sizes in bytes for a MAC RLP packet given different channel conditions expressed in terms of transmission rate in Kbps. Table 11.9.6.1 is reproduced below:

TABLE

Transmission Rate (Kbps)	Minimum Payload Size (bytes)	Maximum Payload Size (bytes)
9.6 Kbps	1	29
19.2 Kbps	30	61
38.4 Kbps	62	125
76.8 Kbps	126	253
153.6 Kbps	254	509

**[00109]** If each transmission level in the above table is expressed as an index value, then the maximum payload size of each MAC RLP packet, including both audio and video, is as follows:

$$\text{Maximum payload size} = 2^{\text{index}+4} - 3.$$

For the above expression, index values 1, 2, 3, 4, and 5 are assigned to transmission rate levels of 9.6, 19.2, 38.4, 76.8 and 153.6 Kbps, respectively.

**[00110]** Audio frames and video frames are packetized and included in MAC RLP packets. The MAC RLP packets each have an associated RL transmission rate, and are capable of carrying different maximum payload sizes corresponding to those transmission rates. For example, some MAC RLP packets may have RL transmission rates of 38.4 Kbps and are each capable of carrying a maximum payload size of 125 bytes. Other MAC RLP packets may have an RL transmission rate of 76.8 Kbps and be capable of carrying a maximum payload size of 253 bytes. Still other MAC RLP packets may have RL transmission rates of 19.2 Kbps and are each capable of carrying a maximum payload size of 61 bytes.

**[00111]** In an exemplary embodiment, the operation of the payload size estimator of video packet shaper 24 can be expressed as an algorithm in pseudo code. The algorithm relies on the following inputs: RA Bit, PA Limit, RL Rate, RLPQueueSize, AudioPacketSize, and MACPredNumberPlus. AudioPacketSize may be derived from an audio packet timer applied to video packet shaper 24. As mentioned previously, the combined RAbit is the reverse activity bit indicating the status of base station activity, PA Limit represents the transmit power headroom limitation imposed by power



requirements and indicates when channel conditions have degraded, RL rate is the transmission rate of the RL, RLPQueueSize indicates fullness of RLP queue 28, and AudioPacketSize indicates the size of the current audio packet, i.e., the audio packet to be added to the next available MAC RLP packet. MACPredNumberPlus indicates the relative priority to be accorded to audio packets versus video packets. The output of the algorithm is VideoPayloadSize.

**[00112]** For initialization of the algorithm, the value MACPredNumber is set as follows:

$$\text{MACPredNumber} = \text{floor}((\text{AudioFramesBundled} * \text{AudioFrameInterval}) / 26.67) + 1 + \text{MACPredNumberPlus}$$

MACPredNumber represents the number of MAC RLP packets necessary to carry a packet containing a single audio frame or set of bundled audio frames.

AudioFrameInterval represents the time interval between audio frames. The value 26.67 is the time allocated for each MAC RLP packet. Hence, if three audio frames are bundled and the audio frame interval is 20 ms, and MACPredNumberPlus is zero, indicating high audio priority, then MACPredNumber is 3. This means that the predicted number of MAC RLP packets for which video payload size will be estimated is 3.

**[00113]** For every bundled audio packet, after sending the bundled audio packets, the payload size estimator of video packet shaper 24 makes a MAC audio throughput determination. The MAC throughput determination may proceed as indicated by the following pseudo code:

```

MACThroughput = 0;
MACRateIncrease = 1 - RABit;
MACPredRate = CurRate;
for (i = 0; i < MACPredNumber; i++)
{
    MACPredRate = MIN(MIN(MACPredRate +
MACRateIncrease, 4), PALimit);
    MACThroughput += (2MACPredRate+4 - 3);
}

```

In the above MAC throughput determination, MACThroughput is the required throughput value for audio transmission, MACRateIncrease indicates whether the MAC

RL rate will be increased based on reverse activity, CurRate is the current MAC RL rate, MACPredRate is the amount of increase in the MAC RL rate, expressed as an index value. As indicated above, MACThroughput is the maximum payload size available for each of the three predicted MAC RLP packets.

**[00114]** Given the maximum payload size MACThroughput for each MAC RLP packet, the video payload size estimator estimates the maximum video payload size (VideoPayloadSize) as follows:

$$\text{VideoPayloadSize} = \text{MAX}(\text{MACThroughput} - \text{RLPQueueSize}, 0)$$

$$\text{VideoPayloadSize} = \text{MAX}(\text{VideoPayloadSize} - 2 * \text{AudioPacketSize} - 45, 0),$$

where RLPQueueSize indicates the fullness of RLP queue 28 and AudioPacketSize represents the size of the audio packet to be added to the next MAC RLP packet. The value 45 is a fixed number in bytes to account for RTP/UDP/IP/PPP overhead of header information introduced by RTP/UDP/IP/PPP conversion module 26. The value of this fixed overhead number could be different in other implementations.

**[00115]** For channel-adaptive video packet shaping over a range of channel conditions, the payload size estimator of video packet shaper 24 adapts to the changing channel conditions, as represented in part by RL transmission rate, to adjust the size of the video packet payload presented for incorporation in the MAC RLP packets generated by MAC layer module 30 from the contents of RLP queue 28. In one example, audio frames are generated at an interval of 60 ms. In this case, a decision is made every 60 ms concerning the available payload size in the next three MAC RLP packets.

**[00116]** At a first decision point, the current MAC RL rate is indexed at 3 to represent 38.4 Kbps, the RA bit is set at zero, the PA limit is equal to 4 and the RLP queue contains X1 bytes. In this case, according to the above formulas, the throughput for each of the next three MAC RLP packets is estimated to be 253 bytes. Accordingly, the overall throughput over the next three MAC RLP packets is 253 + 253 + 253 bytes minus the contents X1 already placed in RLP queue 28. Hence, the MACThroughput value at the first decision point 50 is 253 + 253 + 253 - X1 bytes.

**[00117]** At a second decision point, 60 ms later, the current RL rate is again indexed at 3 and the PA limit is 4, but the RA bit is set to 1 instead of 0. In this case, the RA bit indicates that the base station is busy and results in a prediction of a reduced

throughput over the next three MAC RLP packets. In particular, the estimated throughput  $MACThroughput$  is  $125 + 125 + 125 - X2$  bytes, where  $X2$  represents the contents of RLP queue 28 at the time of the second decision point 80.

**[00118]** At a third decision point, 60 ms after the second decision point, the RA bit is 0, but the RL rate has dropped to an index value of 2 (19.2 Kbps) and the PA limit has dropped to an index value of 2. Consequently, the overall throughput  $MACThroughput$  over the next three MAC RLP packets decreases to  $61 + 61 + 61 - X3$  bytes, where  $X3$  represents the contents of RLP queue 28 at the time of the third decision point 82..

**[00119]** When  $MACThroughput$  is reduced, the space available for video packets is also reduced as a result of prioritization of the audio packets. In this case, the payload size estimator of video packet shaper 24 reduces the estimated size of the video payload for packetization. When  $MACThroughput$  increases, however, the payload size estimator responds by increasing the estimated video payload size. In this manner, video packet shaper 24 not only prioritizes audio packets, but also supports bandwidth-efficient video packetization.

**[00120]** In this example, decisions are made for three MAC RLP packets at a time. In other embodiments, however, a more aggressive decision process may be applied. For example, a decision for estimation of  $MACThroughput$  may be made every 20 ms. In a first 20 ms interval, a decision may be made for three MAC RLP packets. Then, in a second 20 ms interval, a decision may be made for the remaining two MAC RLP packets in a set of three successive MAC RLP packets. Finally, a decision may be made for the last MAC RLP packet in the three-packet set during the next 20 ms interval. In this case, decisions are made over the course of a 60 ms interval and updated every 20 ms for any changes that may have occurred in channel condition or RLP queue fullness. After 60 ms, the process repeats for the next 60 ms and the next three MAC RLP packets, and continues iteratively.

**[00121]** Once  $MACThroughput$  is estimated, the video payload size estimator estimates the video payload size that can be accommodated given the  $MACThroughput$  value, as explained above. Then, the bandwidth efficient packetizer uses the estimated video payload size and a minimum video packet size value to generate the video packet for submission to RTP/UDP/IP/PPP conversion module 26. The operation of the

bandwidth efficient packetizer of video packet shaper 24 will now be described in greater detail.

**[00122]** In general, video packetization should conform to Network Working Group Request for Comment (RFC) 3016, dated November 2000, if MPEG4 video encoding is used, or to RFC 2190, dated September 1997, or RFC 2429, dated October 1998, if ITU H.263 video encoding is used. RFC3016 outlines the RTP payload format for MPEG4 streams. RFC2429 outlines the RTP payload format for the 1998 version of ITU H.263 streams, and RFC 2190 outlines the RTP format for the original version of ITU H.263 streams.

**[00123]** RFC 3016 specifies that a video packet (a) has to start with video object plane (VOP) header or video packet (VP) header, if any of them exists, (b) can contain more than one VP header, if previous rule is satisfied, (c) can contain only video data without any VOP and VP headers in it, and (d) cannot contain data across two video frames. RFC 2190 specifies that a video packet (a) has to start with picture start code (PSC) or group of blocks (GOB), (b) does not have to have GOB header or complete GOB, and (c) does not have to be GOB byte-aligned. RFC2429 specifies that a video packet (a) can start with byte-aligned PSC, GOB header, Slice header, and end of slice (EOS) marker, and (b) can be a Follow-on packet that does not start with any synchronization codes but allows synchronization codes in the middle of the video packet.

**[00124]** Given the above requirements, video encoder 20 may be configured to insert video data into video buffer 52 in the form of VOPs and VPs for MPEG4, or PSCs, GOBs, and SSCs for H.263. An MPEG4-compliant encoder generates data in the units of VOP or VPs. An H.263 encoder generates data in the units of PSCs, GOBs or SSCs, with GOBs byte-aligned. When RFC 2190 is used, Mode A is the default.

**[00125]** In an exemplary embodiment, the operation of the bandwidth efficient packetizer of video packet shaper 24 can be expressed as an algorithm that makes use of the following inputs: VideoDataInBuffer, EstimatedVideoPayloadSize, minVPSize. VideoDataInBuffer represents the size of the video in video buffer 52. EstimatedVideoPayloadSize represents the estimated video payload size determined by the payload size estimator of video packet shaper 24. The value minVPSize is the minimum video packet size to be produced by video packet shaper 24, and serves to control granularity and bandwidth efficiency. The output of the bandwidth efficient

packetization algorithm is one or more video packets for submission to RTP/UDP/IP/PPP conversion module 26. The operation of the bandwidth efficient packetizer, in an exemplary embodiment, is represented by the following pseudo code:

```

RemainingVideoPayloadSize = EstimatedVideoPayloadSize; VideoPayloadSize = 0;
VideoPayloadData[] : an array;
for (;;)
{
if (RemainingVideoPayloadSize < minVPSize/2)
    RemainingVideoPayloadSize = 0;
else if (RemainingVideoPayloadSize < minVPSize)
    RemainingVideoPayloadSize = minVPSize;
If ((RemainingVideoPayloadSize == 0) || (VideoDataInBuffer == NULL)) break;
If (VideoDataInBuffer->Size >= RemainingVideoPayloadSize + minVPSize)
{
if (RemainingVideoPayloadSize >= (minVPSize/2))
{
if (RFC3016 || RFC2429)
    memcpy(VideoPayloadData + VideoPayloadSize, VideoDataInBuffer-
>Data,
    RemainingVideoPayloadSize);
VideoPayloadSize += RemainingVideoPayloadSize;
memcpy(VideoDataInBuffer->Data,
    VideoDataInBuffer->Data + RemainingVideoPayloadSize,
    VideoDataInBuffer->Size - RemainingVideoPayloadSize,
    VideoDataInBuffer->Size - = RemainingVideoPayloadSize);
VideoDataInBuffer->Fragmented = 1;
else if (RFC2190)
    memcpy(VideoPayloadData + VideoPayloadSize, VideoDataInBuffer-
>Data,
    VideoDataInBuffer->Size);
VideoPayloadSize += VideoDataInBuffer->Size;
}
Make one VideoPacket from VideoPayloadData[] with payload size of
VideoPayloadSize;

```

37

```

    RemainingVideoPayloadSize = 0;
}
else
{
    memcpy(VideoPayloadData + VideoPayloadSize, VideoDataInBuffer->Data,
           VideoDataInBuffer->Size);
    VideoPayloadSize += VideoDataInBuffer->Size;
    RemainingVideoPayloadSize = MAX(RemainingVideoPayloadSize -
                                     VideoBuffer->Size - 45, 0);
    if (No more data in buffer || the current TS != the next TS ||
        RemainingVideoPayloadSize == 0 || VideoDataInBuffer->
        Fragmented == 1)
        Make one VideoPacket from VideoPayloadData[] with payload size of
        VideoPayloadSize;
    VideoPayloadSize = 0;
    VideoDataInBuffer = the next frame/GOB/slice unit in the video buffer, if any,
or
    NULL, if no more data
}
}

```

As represented by the above pseudo code, the bandwidth efficient packetizer of video packet shaper 24 produces a video packet based on the EstimatedVideoPayloadSize provided by the payload size estimator and minVPSsize. RemainingVideoPayloadSize represents the amount of payload still available at any point during the generation of a video packet. Initially, RemainingVideoPayloadSize is equal to the entire EstimatedVideoPayloadSize provided by the video payload size estimator.

VideoPayloadSize represents the actual size of the video payload in a packet, and is initially set to zero. VideoPayloadData[] identifies an array of video data segments in video buffer 52.

**[00126]** The bandwidth efficient packetizer first determines whether the RemainingVideoPayloadSize is less than minVPSsize/2. If so, the value RemainingVideoPayloadSize is set to zero. Alternatively, if RemainingVideoPayloadSize is less than minVPSsize, then the value of RemainingVideoPayloadSize is set to equal minVPSsize. Then, if

RemainingVideoPayloadSize is equal to zero or VideoDataInBuffer is null, the process resets as there is either no space remaining in the next available MAC RLP packet or no video remaining in video buffer 52.

**[00127]** If the size of VideoInBuffer is greater than or equal to the value RemainingVideoPayloadSize plus the minVPSize, the bandwidth efficient packetizer next determines whether the RemainingVideoPayloadSize is greater than or equal to minVPsize/2. If so, the bandwidth efficient packetizer determines whether RFC3016 or RFC2429 is applicable. If neither RFC3016 or RFC2429 applies, then the bandwidth efficient packetizer determines whether RFC2190 is applicable, i.e., whether the RTP payload format for the original version of ITU H.263 applies.

**[00128]** If RFC3016 or RFC2429 applies, then the bandwidth efficient packetizer copies (memcpy) video from video buffer 52, as determined by the starting address VideoPayloadData and the offset VideoPayloadSize, to the input buffer identified by VideoDataInBuffer. Initially, VideoPayloadSize is set to zero. The amount of the video copied from video buffer 52 is equal to RemainingVideoPayloadSize, which is initially set to EstimatedVideoPayloadSize. The bandwidth efficient packetizer then adjusts VideoPayloadSize to equal RemainingVideoPayloadSize. Next, the bandwidth efficient packetizer copies the video data from the input buffer to the address identified by offset RemainingVideoPayloadSize in an amount determined by RemainingVideoPayloadSize. The contents of VideoDataInBuffer is then fragmented for packetization.

**[00129]** If RFC2190 applies, then the bandwidth efficient packetizer copies (memcpy) video from video buffer 52, as determined by the starting address VideoPayloadData and the offset VideoPayloadSize, to the input buffer identified by VideoDataInBuffer. Again, VideoPayloadSize is initially set to zero. The amount of the video copied from video buffer 52 is equal to size of the VideoDataInBuffer. The VideoPayloadSize is then made equal to the size of VideoDataInBuffer.

**[00130]** Upon exiting either the RFC3016/RFC2429 operations or the RFC2190 operations, the bandwidth efficient packetizer next generates a VideoPacket from the VideoPayloadData with a payload size equal to the current value of VideoPayloadSize. The value RemainingVideoPayloadSize is then set to zero. At this point, a video packet has been generated by the bandwidth efficient packetizer for submission to RTP/UDP/IP/PPP conversion module 26. If RemainingVideoPayloadSize is not less

than  $\text{minVPSize}$ ,  $\text{RemainingVideoPayloadSize}$  is not equal to zero,  $\text{VideoDataInBuffer}$  is not null, and the size of  $\text{VideoDataInBuffer}$  is not greater than or equal to  $\text{RemainingVideoPayloadSize} + \text{minVPSize}$ , then the bandwidth efficient packetizer copies data from buffer 52 to  $\text{VideoDataInBuffer}$  using the address  $\text{VideoPayloadData}$  plus the offset of  $\text{VideoPayloadSize}$ . In this case, the amount of data copied is equal to  $\text{VideoPayloadSize}$ . Then, the bandwidth efficient packetizer sets  $\text{VideoPayloadSize}$  equal to the size of  $\text{VideoDataInBuffer}$ .

**[00131]** The bandwidth efficient packetizer of video packet shaper 24 next sets  $\text{RemainingVideoPayloadSize}$  equal to the maximum of  $\text{RemainingVideoPayloadSize}$  minus  $\text{VideoBufferSize}$  and zero.  $\text{VideoBufferSize}$  represents the size of video buffer 52. If there is no more data in video buffer 52, or a current timestamp (TS) is not equal to the next TS, or  $\text{RemainingVideoPayloadSize}$  is equal to zero, or  $\text{VideoDataInBuffer}$  is fragmented, the bandwidth efficient packetizer generates one  $\text{VideoPacket}$  from the  $\text{VideoPayloadData}$  with a payload size of  $\text{VideoPayloadSize}$ , and sets  $\text{VideoPayloadSize}$  to zero. Otherwise, the bandwidth efficient packetizer sets the  $\text{VideoDataInBuffer}$  to acquire the next frame, GOB, or slice unit in video buffer 52, if any, or null, if there is no more data in the video buffer.

**[00132]** As an example of channel-adaptive video packet shaping, assume that audio buffer 62 generates an audio packet. RTP/UDP/IP/PPP module 26 adds the audio packet to RLP queue 28. The payload size estimator of video packet shaper 24 determines RLP queue size, the audio-video priority value, and channel conditions. Based on those determinations, the payload size estimator of video packet shaper 24 estimates the payload size of the next video packet to be generated. The bandwidth efficient packetizer generates the video packet and sizes the video packet based on estimated payload size and a minimum video packet size. The bandwidth efficient packetizer adds the video packet to RLP queue 28. MAC layer module 30 generates a MAC RLP packet from the contents of RLP queue 28.

**[00133]** The techniques described in this disclosure may be implemented within a general purpose microprocessor, digital signal processor (DSP), application specific integrated circuit (ASIC), field programmable gate array (FPGA), or other equivalent logic devices. For example, video encoder system 12, video decoder system 14, and associated components and modules, may be implemented as parts of an encoding process, or coding/decoding (CODEC) process, running on a digital signal processor



(DSP) or other processing device. Accordingly, components described as modules may form programmable features of such a process, or a separate process.

**[00134]** Video encoder system 12 may have a dedicated memory for storing instructions and data, as well as dedicated hardware, software, firmware, or combinations thereof. If implemented in software, the techniques may be embodied as instructions executable by one or more processors. The instructions may be stored on a computer-readable medium such as random access memory (RAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage device, or the like. The instructions cause one or more processors to perform certain aspects of the functionality described in this disclosure.

**[00135]** Various embodiments have been described. These and other embodiments are within the scope of the following claims.

**CLAIMS:**

1. A method comprising:
  - adding encoded video to a physical video buffer;
  - removing a packet of the encoded video from the physical video buffer for transmission over a channel;
  - maintaining a virtual video buffer representing an amount of the encoded video relative to a target encoding rate; and
  - controlling an actual encoding rate of the video based on fullness of the physical video buffer and fullness of the virtual video buffer.
2. The method of claim 1, wherein the channel exhibits varying channel conditions, the method further removing varying amounts of the encoded video from the physical video buffer over time in response to the varying channel conditions.
3. The method of claim 1, wherein controlling an actual encoding rate includes skipping encoding of a video frame if a maximum of the fullness of the physical video buffer and the fullness of the virtual video buffer is greater than or equal to a skipping threshold.
4. The method of claim 3, further comprising estimating a frame budget defining a number of encoding bits available for a frame of the video if a maximum of the fullness of the physical video buffer and the fullness of the virtual video buffer is less than the skipping threshold, and controlling the actual encoding rate based on the frame budget.
5. The method of claim 4, wherein estimating a frame budget includes estimating the number of encoding bits based on a target bit rate, a target frame rate, the fullness of the physical video buffer, the fullness of the virtual video buffer, and the skipping threshold.

6. The method of claim 5, further comprising:  
determining the fullness of the virtual video buffer substantially based on the following expression:

$$W_n = \max (W_{n-1} + B_{n-1} - R/F, 0)$$

where  $W_n$  is the fullness of the virtual video buffer for video frame  $n$ ,  $W_{n-1}$  is the fullness of the virtual video buffer for video frame  $n-1$ ,  $B_{n-1}$  is the actual coding bits allocated to video frame  $n-1$ ,  $R$  is the target bit rate, and  $F$  is the target frame rate; and

determining the actual encoding rate substantially according to the following expressions:

$$X_n = \max (w \cdot V_n, W_n), \text{ and}$$

$$B_n = R/F - \Delta,$$

where  $\Delta = X_n/F$  if  $X_n > Z \cdot M$ ,  $\Delta = X_n - Z \cdot M$  if  $X_n \leq Z \cdot M$ ,  $V_n$  is the fullness of the physical video buffer for video frame  $n$ ,  $w$  is a positive integer,  $M$  is the skipping threshold, and  $Z$  is a coefficient.

7. The method of claim 1, further comprising estimating a frame budget defining a number of encoding bits available for a frame of the video if a maximum of the fullness of the physical video buffer and the fullness of the virtual video buffer is less than the skipping threshold, and controlling the actual encoding rate based on the frame budget.

8. The method of claim 7, wherein estimating a frame budget includes estimating the number of encoding bits based on a target bit rate, a target frame rate, the fullness of the physical video buffer, the fullness of the virtual video buffer, and the skipping threshold.

9. The method of claim 8, further comprising:  
determining the fullness of the virtual video buffer substantially based on the following expression:

$$W_n = \max (W_{n-1} + B_{n-1} - R/F, 0)$$

where  $W_n$  is the fullness of the virtual video buffer for video frame  $n$ ,  $W_{n-1}$  is the fullness of the virtual video buffer for video frame  $n-1$ ,  $B_{n-1}$  is the actual coding bits allocated to video frame  $n-1$ ,  $R$  is the target bit rate, and  $F$  is the target frame rate; and determining the actual encoding rate substantially according to the following expressions:

$$X_n = \max (w \cdot V_n, W_n), \text{ and}$$

$$B_n = R/F - \Delta,$$

where  $\Delta = X_n/F$  if  $X_n > Z \cdot M$ ,  $\Delta = X_n - Z \cdot M$  if  $X_n \leq Z \cdot M$ ,  $V_n$  is the fullness of the physical video buffer for video frame  $n$ ,  $w$  is a positive integer,  $M$  is the skipping threshold, and  $Z$  is a coefficient.

10. The method of claim 1, wherein the channel is a wireless channel, the method further comprising:

estimating throughput of the wireless channel; and

generating a video packet from the physical video buffer with a video packet size determined based on the estimated throughput.

11. The method of claim 10, further comprising generating an audio packet, wherein the video packet size is determined based on the estimated throughput and a size of the audio packet.

12. The method of claim 11, further comprising generating a physical layer packet based on the audio packet and the video packet, and transmitting the physical layer packet to a remote device to support a mobile wireless video telephony conference.

13. The method of claim 12, wherein the physical layer packet conforms substantially to the cdma2000 1x EV-DO, Release 0, standard.

14. The method of claim 1, further comprising:

estimating a frame budget defining a number of encoding bits available for a frame of the video based on the actual encoding rate; and  
allocating rho domain values to macroblocks within the frame based on the frame budget.

15. The method of claim 14, further comprising mapping the rho domain values to corresponding quantization parameter (QP) values to allocate a number of encoding bits to each of the macroblocks.

16. The method of claim 1, further comprising dropping at least a portion of the encoded video from the physical video buffer when the portion of the encoded video has been in the physical video buffer for a period of time that exceeds a threshold.

17. A system comprising:  
a video encoder;  
a physical video buffer that stores video encoded by the video encoder;  
a packet shaper that removes the encoded video from the physical video buffer for transmission over a channel;  
a virtual video buffer representing an amount of the encoded video relative to a target encoding rate; and  
a controller that controls an actual encoding rate of the video based on fullness of the physical video buffer and fullness of the virtual video buffer.

18. The system of claim 17, wherein the channel exhibits varying channel conditions, and the packet shaper removes varying amounts of the encoded video from the physical video buffer over time in response to the varying channel conditions.

19. The system of claim 17, wherein the controller skips encoding of a video frame if a maximum of the fullness of the physical video buffer and the fullness of the virtual video buffer is greater than or equal to a skipping threshold.

20. The system of claim 19, wherein the controller estimates a frame budget defining a number of encoding bits available for a frame of the video if a maximum of

the fullness of the physical video buffer and the fullness of the virtual video buffer is less than the skipping threshold, and controls the actual encoding rate based on the frame budget.

21. The system of claim 20, wherein the controller estimates the frame budget based on a target bit rate, a target frame rate, the fullness of the physical video buffer, the fullness of the virtual video buffer, and the skipping threshold.

22. The system of claim 21, wherein the controller:  
determines the fullness of the virtual video buffer substantially based on the following expression:

$$W_n = \max (W_{n-1} + B_{n-1} - R/F, 0)$$

where  $W_n$  is the fullness of the virtual video buffer for video frame  $n$ ,  $W_{n-1}$  is the fullness of the virtual video buffer for video frame  $n-1$ ,  $B_{n-1}$  is the actual coding bits allocated to video frame  $n-1$ ,  $R$  is the target bit rate, and  $F$  is the target frame rate; and

determines the actual encoding rate substantially according to the following expressions:

$$X_n = \max (w * V_n, W_n), \text{ and}$$

$$B_n = R/F - \Delta,$$

where  $\Delta = X_n/F$  if  $X_n > Z * M$ ,  $\Delta = X_n - Z * M$  if  $X_n \leq Z * M$ ,  $V_n$  is the fullness of the physical video buffer for video frame  $n$ ,  $w$  is a positive integer,  $M$  is the skipping threshold, and  $Z$  is a coefficient.

23. The system of claim 17, wherein the controller estimates a frame budget defining a number of encoding bits available for a frame of the video if a maximum of the fullness of the physical video buffer and the fullness of the virtual video buffer is less than the skipping threshold, and controls the actual encoding rate based on the frame budget.

24. The system of claim 23, wherein the controller estimates the frame budget based on a target bit rate, a target frame rate, the fullness of the physical video buffer, the fullness of the virtual video buffer, and the skipping threshold.

25. The system of claim 24, wherein the controller:  
determines the fullness of the virtual video buffer substantially based on the following expression:

$$W_n = \max (W_{n-1} + B_{n-1} - R/F, 0)$$

where  $W_n$  is the fullness of the virtual video buffer for video frame  $n$ ,  $W_{n-1}$  is the fullness of the virtual video buffer for video frame  $n-1$ ,  $B_{n-1}$  is the actual coding bits allocated to video frame  $n-1$ ,  $R$  is the target bit rate, and  $F$  is the target frame rate; and determines the actual encoding rate according to the following expressions:

$$X_n = \max (w \cdot V_n, W_n), \text{ and}$$

$$B_n = R/F - \Delta,$$

where  $\Delta = X_n/F$  if  $X_n > Z \cdot M$ ,  $\Delta = X_n - Z \cdot M$  if  $X_n \leq Z \cdot M$ ,  $V_n$  is the fullness of the physical video buffer for video frame  $n$ ,  $w$  is a positive integer,  $M$  is the skipping threshold, and  $Z$  is a coefficient.

26. The system of claim 17, wherein the channel is a wireless channel, wherein the packet shaper estimates throughput of the wireless channel, and generates the video packet from the physical video buffer with a video packet size determined based on the estimated throughput.

27. The system of claim 26, further comprising an audio encoder that generates an audio packet, wherein the video packet size is determined based on the estimated throughput and a size of the audio packet.

28. The system of claim 27, further comprising a physical layer module that generates a physical layer packet based on the audio packet and the video packet; and transmits the physical layer packet to a remote device to support a mobile wireless video telephony conference.

29. The system of claim 28, wherein the physical layer packet conforms substantially to the cdma2000 1x EV-DO, R1 release 0, standard.

30. The system of claim 17, wherein the controller:  
estimates a frame budget defining a number of encoding bits available for a frame of the video based on the actual encoding rate; and  
allocates rho domain values to macroblocks within the frame based on the frame budget.
31. The system of claim 30, wherein the controller maps the rho domain values to corresponding quantization parameter (QP) values to allocate a number of encoding bits to each of the macroblocks.
32. The system of claim 17, wherein the physical video buffer drops at least a portion of the encoded video from the physical video buffer when the portion of the encoded video has been in the physical video buffer for a period of time that exceeds a threshold.
33. A system comprising:  
means for encoding video;  
physical video buffer means for storing the encoded video;  
packet shaping means for generating a video packet from the encoded video from the physical video buffer means for transmission over a channel;  
virtual buffer means for maintaining a virtual video buffer representing an amount of the encoded video relative to a target encoding rate; and  
control means for controlling an actual encoding rate of the video based on fullness of the physical video buffer and fullness of the virtual video buffer.
34. The system of claim 33, wherein the channel exhibits varying channel conditions, and the packet shaping means removes varying amounts of the encoded video from the physical video buffer over time in response to the varying channel conditions.
35. The system of claim 33, wherein the control means includes means for skipping encoding of a video frame if a maximum of the fullness of the physical video buffer and the fullness of the virtual video buffer is greater than or equal to a skipping threshold.



36. The system of claim 35, wherein the control means includes means for estimating a frame budget defining a number of encoding bits available for a frame of the video if a maximum of the fullness of the physical video buffer and the fullness of the virtual video buffer is less than the skipping threshold, and means for controlling the actual encoding rate based on the frame budget.

37. The system of claim 36, wherein the means for estimating a frame budget includes means for estimating the number of encoding bits based on a target bit rate, a target frame rate, the fullness of the physical video buffer, the fullness of the virtual video buffer, and the skipping threshold.

38. The system of claim 33, wherein the channel is a wireless channel, and the packet shaping means includes means for estimating throughput of the wireless channel, and means for generating a video packet from the physical video buffer with a video packet size determined based on the estimated throughput.

39. The system of claim 38, further comprising means for generating an audio packet, wherein the packet shaping means determines the video packet size based on the estimated throughput and a size of the audio packet.

40. The system of claim 38, further comprising means for generating a physical layer packet based on the audio packet and the video packet, and means for transmitting the physical layer packet to a remote device to support a mobile wireless video telephony conference.

41. The system of claim 40, wherein the physical layer packet conforms substantially to the cdma2000 1x EV-DO, Release 0, standard.

42. The system of claim 33, wherein the control means includes:  
means for estimating a frame budget defining a number of encoding bits available for a frame of the video based on the actual encoding rate;  
means for allocating rho domain values to macroblocks within the frame based on the frame budget; and

means for mapping the rho domain values to corresponding quantization parameter (QP) values to allocate a number of encoding bits to each of the macroblocks.

43. The method of claim 33, further comprising means for dropping at least a portion of the encoded video from the physical video buffer means when the portion of the encoded video has been in the physical video buffer means for a period of time that exceeds a threshold.

44. A computer-readable medium comprising instructions to cause one or more processors to:

add encoded video to a physical video buffer;

remove a packet of the encoded video from the physical video buffer for transmission over a channel;

maintain a virtual video buffer representing an amount of the encoded video relative to a target encoding rate; and

control an actual encoding rate of the video based on fullness of the physical video buffer and fullness of the virtual video buffer.

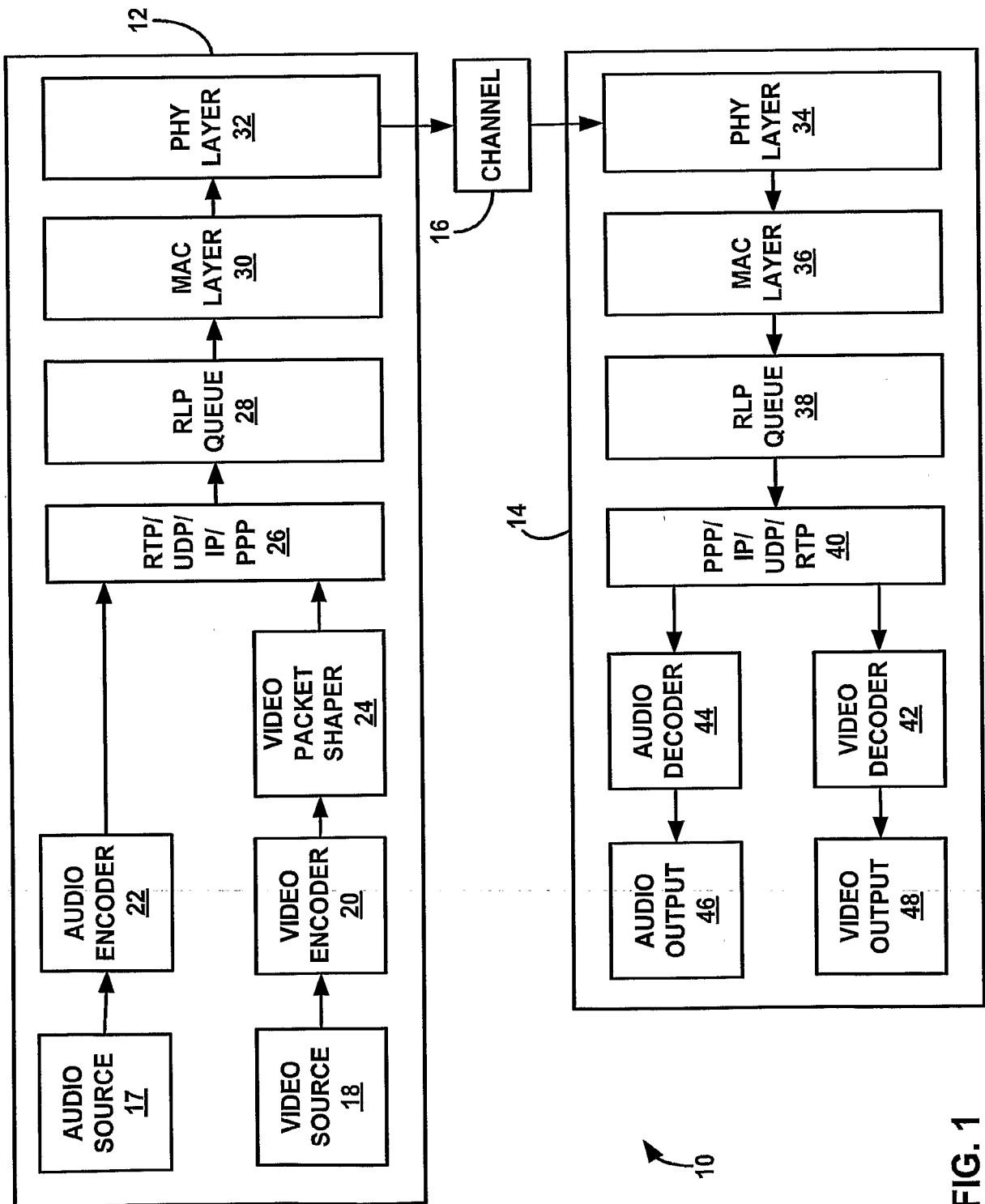


FIG. 1

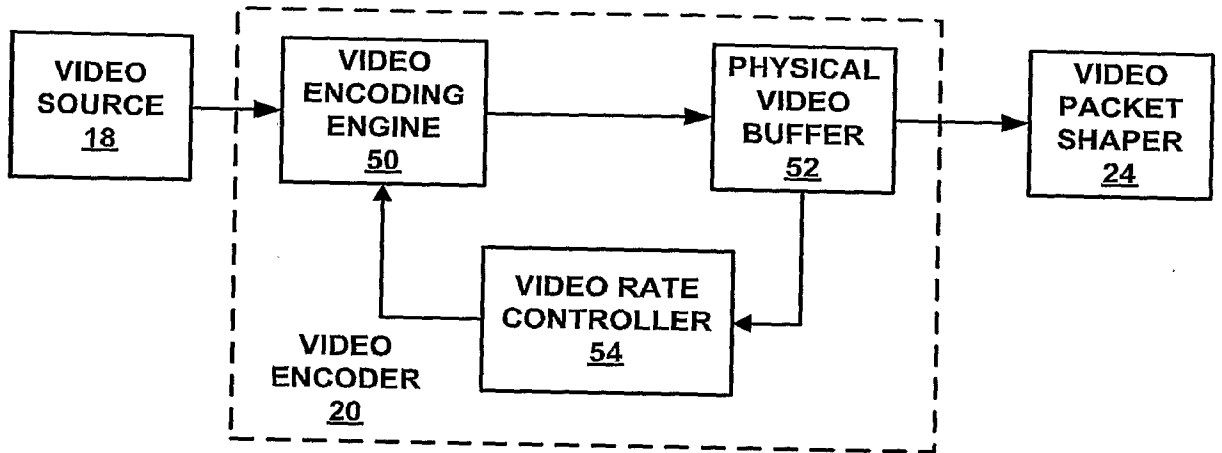


FIG. 2

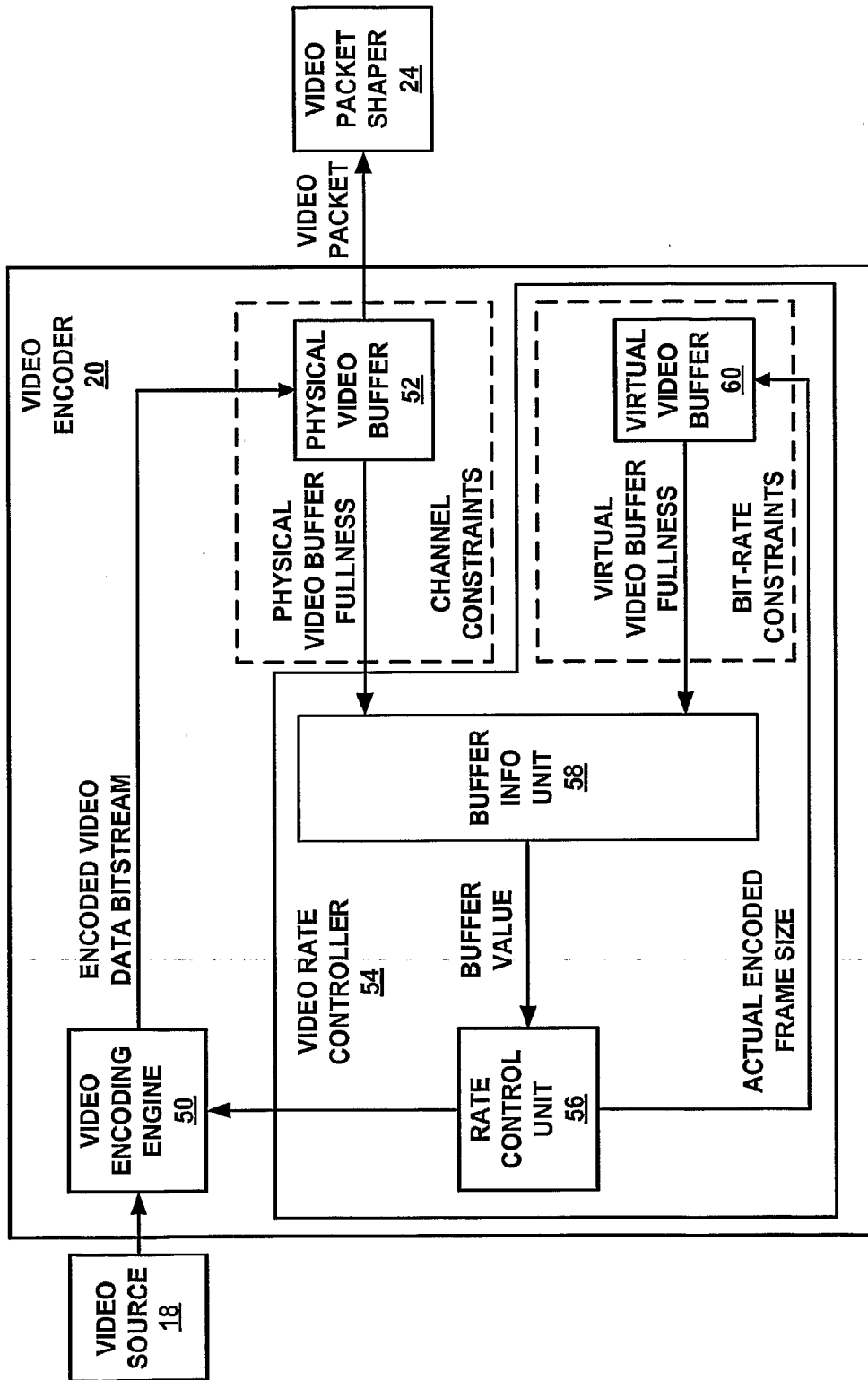


FIG. 3

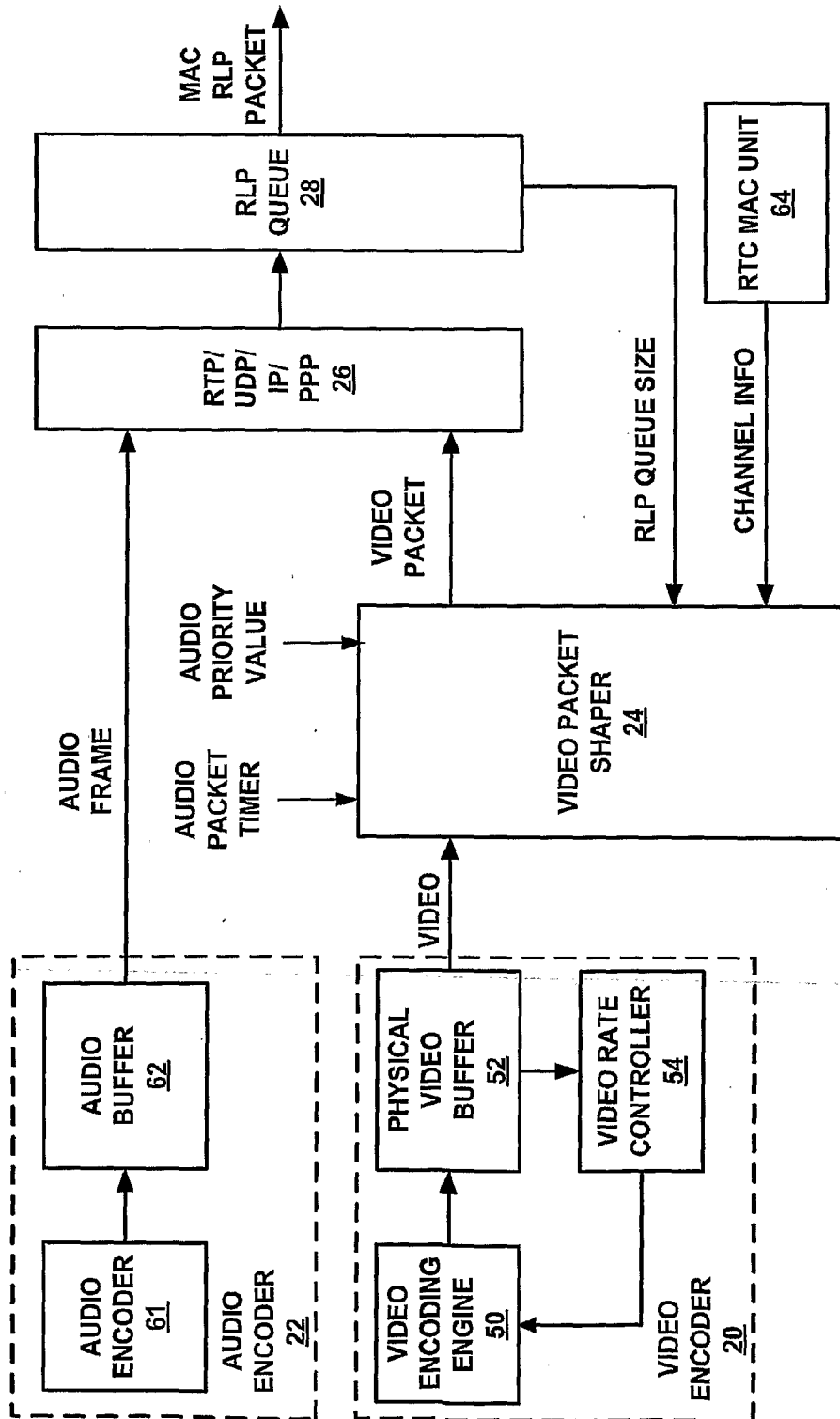


FIG. 4

5/6

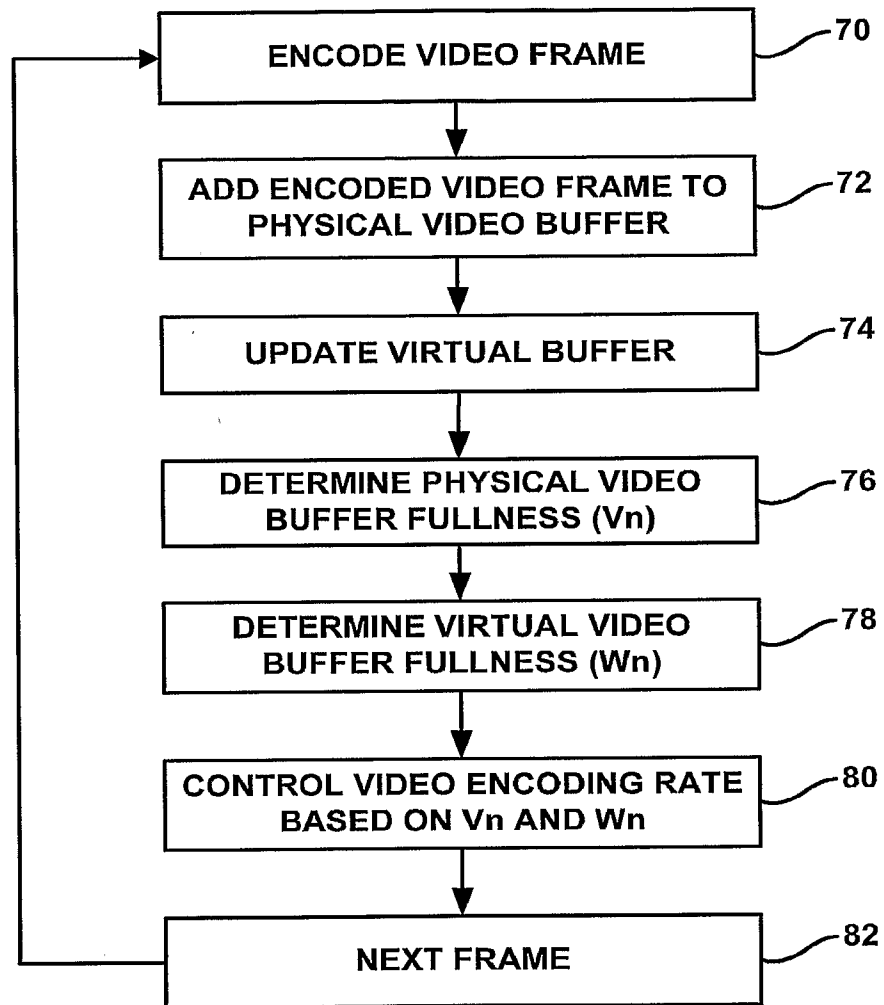


FIG. 5

6/6

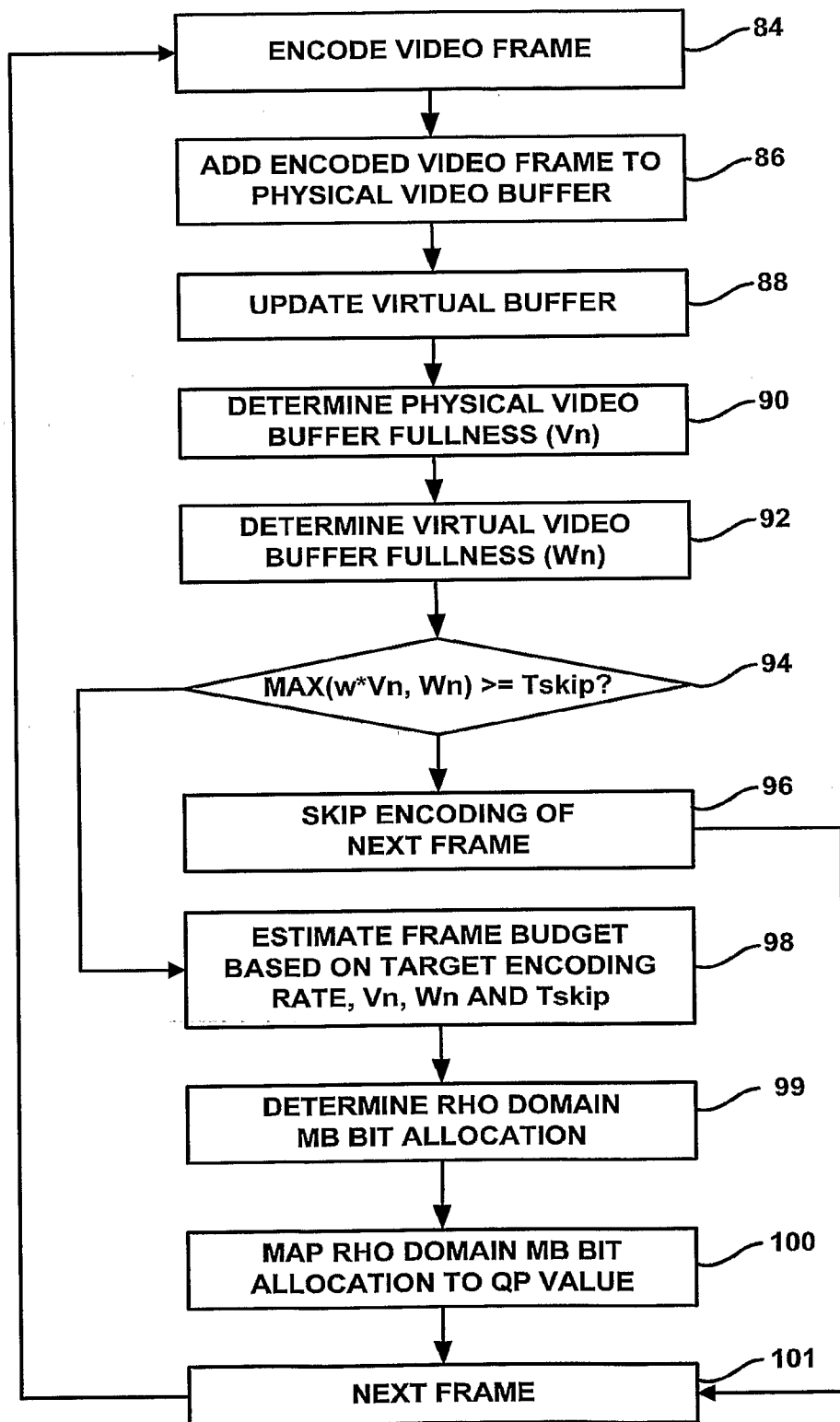


FIG. 6