



- (51) **International Patent Classification:**
H04N 7/26 (2006.01) *H04N 7/66* (2006.01)
- (21) **International Application Number:**
PCT/US2012/037055
- (22) **International Filing Date:**
9 May 2012 (09.05.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/103,494 9 May 2011 (09.05.2011) US
- (71) **Applicant (for all designated States except US):**
GOOGLE INC. [US/US]; 1600 Amphitheatre Parkway,
Mountain View, CA 94043 (US).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** **PANICONI, Marco**
[US/US]; 402 Union Ave. Unit C, Campbell, CA 95008
(US). **KOZICA, Ermin** [SE/SE]; Järla Gårdsäg 29, S-131
61 Nacka (SE).
- (74) **Agents:** **CARROLL, James** et al.; Lerner, David, Litten-
berg, Krumholz & Mentlik, LLP, 600 South Avenue West,
Westfield, NJ 07090 (US).
- (81) **Designated States (unless otherwise indicated, for every
kind of national protection available):** AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ,

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

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

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 (Rule 48.2(g))

(54) **Title:** PROVIDING ADAPTIVE MEDIA OPTIMIZATION

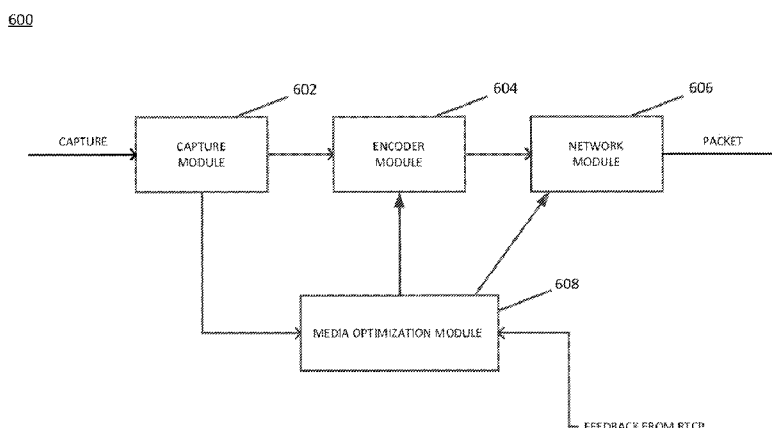


FIG. 6

(57) **Abstract:** A method and system for providing adaptive media optimization are described. Aspects of the invention modify video encoding and network transmission settings to optimize the user viewing experience. The system and method sample video content (102) to determine various content features (502) of the video. The system and method use the identified content features in conjunction with network statistics to modify (510) encoding settings and network transmission options to ensure a minimum of interruption in the transmitted video. Previously generated lookup tables (122) ensure efficient mapping of video content and network conditions to encoding and transmission settings.



PROVIDING ADAPTIVE MEDIA OPTIMIZATION

CROSS REFERENCE TO RELATED APPLICATION

[0001] The present application is a continuation of U.S. Patent Application No. 13/103,494, filed on May 9, 2011, the disclosure of which is hereby incorporated herein by reference.

BACKGROUND

[0002] Increased access to high speed computer networks has lead to an explosion in multimedia content available to modern users. In the course of a typical browsing session, the user may view images, listen to audio, and watch video. Each of these media types may be provided in various encoding formats to optimize the viewing experience for the user. Some content is provided in multiple formats, such that a user can select the most appropriate for their individual situation. For example, a video may be provided in both high definition (HD) and standard definition (SD) formats. A user with a slower connection may opt to view the video in SD format to reduce the delay while waiting for the video to load.

[0003] However, not all such decisions are straightforward. Different video formats and encoding methods may be optimal for some media, but not others, based on the content of the media. Network conditions may fluctuate, resulting in a particular format being optimal some times, but not others. A user may not be sophisticated enough to select an appropriate format for their system capabilities.

BRIEF SUMMARY

[0004] A method and system for providing adaptive media optimization are described. Aspects of the invention modify video encoding and network transmission settings to optimize the user viewing experience. The system and method sample video content to determine various content features of the video. The system and method use the identified content features in conjunction with network statistics to modify

encoding settings and network transmission options to ensure a minimum of interruption in the transmitted video. Previously generated lookup tables ensure efficient mapping of video content and network conditions to encoding and transmission settings. The use of lookup tables in this manner allows for the application of the system and method in real-time operations.

[0005] One aspect of the disclosure describes a computer-implemented method for providing adaptive media optimization. The method includes determining, using a processor, a content class of a video based on a set of content characteristics, determining a set of network conditions, configuring at least one of an encoding option or a transmission option based upon the content class and the network conditions, and encoding and transmitting the video in accordance with the configured encoding option or transmission option. The content characteristics comprise a motion level of the video. In one aspect, the encoding option is at least one of a source rate or an error resilience (ER) setting. The ER setting may be determined as a function of a packet loss value, an effective recovery rate, and the content class of the video. In one aspect, the ER setting is determined by at least one of a function $ER = p - \mu(ML)R$ or $ER = \mu_2(ML)RPL$, wherein ER is an error resilience value, p is a packet loss value, $\mu(ML)$ and $\mu_2(ML)$ are functions based on a motion level value, ML, R is an effective recovery rate and RPL is a residual packet loss value. The transmission option may be at least one of a forward error correction (FEC) factor or a negative-acknowledgement (NACK) setting. In some aspects, the method may further include selecting a set of parameters for a distortion function based upon the content class, and using the selected parameters to configure the at least one encoding option and transmission option. In yet further aspects, the method may further include sampling the video to identify the

one or more content characteristics. The content characteristics may include at least one of a frame rate, a spatial resolution, a motion level, a spatial texture level, and/or a color depth. The content class of the video may be determined by looking up the content characteristics in a video parameter lookup table. The transmission option may be configured by performing a table lookup on a FEC table indexed by the set of network statistics and the content class of the video. The encoding option may be configured by performing a table lookup on a Residual Packet Loss (RPL) table indexed by the set of network statistics, the content class of the video, and a FEC protection factor. In some aspects, the network statistics comprise a packet loss value, a round trip time value, and a network rate. The network statistics may be provided by Real-Time Transport Control Protocol (RTCP).

[0006] Additional aspects of the disclosure describe a method for generating a video parameter lookup table. The method may include extracting one or more source content characteristics from a plurality of source videos, mapping, using a processor, one or more of the plurality of source videos to a content class based on the extracted source characteristics, computing, for a given set of source videos mapped to a same content class, a distortion value over a range of simulated network rates, packet loss values, error resilience settings, and forward error correction (FEC) protection factors, determining nearest fit parameters to map a distortion value for each video in the given set of source videos to a common distortion function, and storing the nearest fit parameters in the video parameter lookup table, such that the video parameter table is indexed by content class. In some aspects, the error resilience settings are retrieved from a Residual Packet Loss (RPL) lookup table and a source motion level, and the source motion level is derived from the extracted content characteristics.

[0007] Yet further aspects of the disclosure describe a method for generating a Residual Packet Loss (RPL) lookup table. The method includes specifying a packet loss model, selecting a range for a packet loss value, a network rate, and a forward error correction (FEC) redundancy factor, selecting one or more stepping values for the packet loss value, the network rate, and the FEC redundancy factor, computing, using a processor, an effective recovery rate for one or more combinations of the range of the packet loss value, the network rate, and the FEC redundancy factor using the stepping values, and storing the computed effective recovery rate in a lookup table indexed by the packet loss value, the network rate, and the FEC redundancy factor.

[0008] Another aspect of the disclosure provides a method for generating a forward error correction (FEC) lookup table. The method includes generating a residual packet loss (RPL) lookup table, generating a video parameter lookup table, selecting a content class from the video parameter lookup table, calculating, using a processor, a FEC redundancy factor by minimizing a cost function for the selected content class and a simulated network state, and storing the FEC redundancy factor in a FEC table indexed by the content class and simulated network state.

[0009] Another aspect of the disclosure describes a processing system for providing adaptive media optimization. The processing system includes at least one processor, a memory for storing a video parameter lookup table, a residual packet loss (RPL) lookup table, and a forward error correction (FEC) lookup table, and a media optimization module associated with the at least one processor. The memory is coupled to the at least one processor. The media optimization module is configured to identify one or more characteristics of a video, determine a content class of the video based on the one or more characteristics, retrieve one or more distortion model

parameters from the video parameter lookup table based upon the content class, determine a set of network conditions, retrieve a FEC redundancy factor from the FEC lookup table based on the set of network conditions and the content class, and retrieve an error resilience (ER) setting from the RPL table based on the FEC redundancy factor, the set of network conditions, and the content class of the video. The media optimization module may also determine a source encoding rate for the video based upon the maximum network rate, the FEC redundancy factor, and the ER setting. In some aspects, the processing system further includes an encoder module to receive the source encoding rate and the ER setting from the media optimization module, and to encode the video in accordance with the source encoding rate and the ER setting. The processing system may also include a network module to receive a FEC redundancy factor from the media optimization module and to transmit the video in accordance with the FEC redundancy factor. The set of network conditions may include a packet loss value, a network rate, and a round trip time. The one or more characteristics of the source video may include a resolution, a frame rate, a motion level, a spatial texture level, and/or a color depth.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Figure 1 is a system diagram depicting a server in communication with a video source and a client device in accordance with aspects of the invention.

[0011] Figure 2 is a method for generating a Residual Packet Loss table in accordance with aspects of the invention.

[0012] Figure 3 is a method for generating a video model parameter table in accordance with aspects of the invention.

[0013] Figure 4 is a method for generating a forward error correction (FEC) table in accordance with aspects of the invention.

[0014] Figure 5 is a method for providing adaptive media optimization in accordance with aspects of the invention.

[0015] Figure 6 is a block diagram depicting data flow throughout a system for providing adaptive media optimization in accordance with aspects of the invention.

DETAILED DESCRIPTION

[0016] Methods and systems for providing adaptive media optimization are described. Video encoding and network transmission settings may be modified to optimize the user viewing experience. The systems and methods sample video content to determine various content features of the video. The identified content features are used in conjunction with network statistics to modify encoding settings and network transmission options to ensure a minimum of interruption in the transmitted video. Previously generated lookup tables ensure efficient mapping of video content and network conditions to encoding and transmission settings. The use of lookup tables in this manner allows for the application of the system and method in real-time operations.

[0017] As shown in Figure 1, a system 100 in accordance with one aspect of the invention includes a video source 102, a media optimization server 104, a network 106, and a client device 108. The media optimization server 104 receives video data from the video source 102, and encodes and transmits to the video to the client device 108 via the network 106. The encoding and transmission processes are optimized based upon the content of the source video and various network conditions. Such optimization is described in further detail below. (see Figures 2-6).

[0018] The video source 102 may be any device capable of capturing or transmitting a video image. For example, the video source may be a digital camera, a digital camcorder, a computer server, a webcam, a mobile phone, a personal digital assistant, or any other device capable of capturing or

transmitting video. In some aspects, the media optimization server 104 may receive audio and/or video from multiple video sources 102, and combine the sources into a single stream.

[0019] The media optimization server 104 may include a processor 110, a memory 112 and other components typically present in general purpose computers. The memory 112 may store instructions and data that are accessible by the processor 110. The processor 110 may execute the instructions and access the data to control the operations of the media optimization server 104.

[0020] The memory 112 may be any type of memory operative to store information accessible by the processor 110, including a computer-readable medium, or other medium that stores data that may be read with the aid of an electronic device, such as a hard-drive, memory card, read-only memory ("ROM"), random access memory ("RAM"), digital versatile disc ("DVD") or other optical disks, as well as other write-capable and read-only memories. The system and method may include different combinations of the foregoing, whereby different portions of the instructions and data are stored on different types of media.

[0021] The instructions may be any set of instructions to be executed directly (such as machine code) or indirectly (such as scripts) by the processor 110. For example, the instructions may be stored as computer code on a tangible computer-readable medium. In that regard, the terms "instructions" and "programs" may be used interchangeably herein. The instructions may be stored in object code format for direct processing by the processor 110, or in any other computer language including scripts or collections of independent source code modules that are interpreted on demand or compiled in advance. Functions, methods, and routines of the instructions are explained in more detail below (see Figures 2-6).

[0022] Data may be retrieved, stored or modified by processor in accordance with the instructions. For instance, although the architecture is not limited by any particular data structure, the data may be stored in computer registers, in a relational database as a table having a plurality of different fields and records, Extensible Markup Language ("XML") documents or flat files. The data may also be formatted in any computer readable format such as, but not limited to, binary values or Unicode. By further way of example only, image data may be stored as bitmaps made up of grids of pixels that are stored in accordance with formats that are compressed or uncompressed, lossless (e.g., BMP) or lossy (e.g., JPEG), and bitmap or vector-based (e.g., SVG), as well as computer instructions for drawing graphics. The data may include any information sufficient to identify the relevant information, such as numbers, descriptive text, proprietary codes, references to data stored in other areas of the same memory or different memories (including other network locations) or information that is used by a function to calculate the relevant data.

[0023] The processor 110 may be any well-known processor, such as processors from Intel Corporation or AMD. Alternatively, the processor may be a dedicated controller such as an application-specific integrated circuit (ASIC). The processor may also be a programmable logic device (PLD) such as a field programmable gate array (FPGA).

[0024] Although Figure 1 functionally illustrates the processor and memory as each being within a single block, it should be understood that the processor 110 and memory 112 may actually include multiple processors and memories that may or may not be stored within the same physical housing. Accordingly, references to a processor, computer, or memory will be understood to include references to a collection of

processors, computers, or memories that may or may not operate in parallel.

[0025] The media optimization server 104 may be at one node of a network and be operative to directly and indirectly communicate with other nodes of the network. For example, the media optimization server 104 may include a web server that is operative to communicate with a client device via the network 106 such that the media optimization server 104 uses the network 106 to transmit and display information to a user on a display of the client device. While the concepts described herein are generally discussed with respect to a media optimization server 104, aspects of the invention may be applied to any computing node capable of managing media encoding operations.

[0026] In order to facilitate the media optimization operations of the media optimization server 104, the memory 112 may further include a capture module 114, an encoder module 116, a network module 118, a media optimization module 120, and a set of lookup tables 122.

[0027] The capture module 114 receives incoming data from the video source 102. For example, the capture module 114 may be a driver application interfacing with a webcam device, a server application receiving data from a client device transmitting a video stream, an application receiving an encoded video file from a remote source, and the like. The capture module 114 operates to accept the data from the video source and send a sample of the video data to the media optimization module 120.

[0028] The encoder module 116 manages the process by which the video received via the capture module 114 is processed into a format suitable for packetization and transmission by the network module 116. The encoder module 116 receives instructions from the media optimization module 120 to configure the encoding operations, such as the format, the

source encoding rate, the frame rate, the spatial resolution, and the Error Resilience (ER) settings associated with the video. ER is an encoder feature that forces intra-coding for some macro-blocks on P-frames (delta frames). The ER settings determine the amount of macro-block encoding present on the P-frames. By encoding extra data into the P-frames, ER allows for the ability to reduce error propagation in the event of errors (such as those caused by dropped or delayed packets) in one or more previous and/or subsequent frames.

[0029] The network module 118 manages the packetization and transmission of the video as encoded by the encoder module 116. The network module 118 receives instructions from the media optimization module 120 to configure the network parameters, such as the Forward Error Correction (FEC) protection/rate, and whether or not a negative acknowledgement character ("NACK") method is used to verify that a packet has been received by a client device. FEC methods generally operate to send extra/redundant packets to enable the receiver to recover lost packets. A traditional NACK method operates by sending a notification to a sender whenever the receiver has failed to receive a data packet, either due to a timeout or receiving a next packet out of order. When the receiver sends such a notification (a NACK), the server retransmits the packet to the receiver.

[0030] The media optimization module 120 manages the encoding, packetization, and transmission operations as performed by the encoder module 116 and the network module 118. The media optimization module 120 receives a sample of video data from the capture module 114 and a set of network statistics, performs an analysis on the video sample and the network statistics using a set of lookup tables 122, and then instructs the encoder module 116 and the network module 118 based on the analysis. Exemplary methods by which this analysis may be performed are described below (see Figures 2-

4). The lookup tables 122 include a set of configuration parameters that are indexed by a set of video content features and network statistics. The lookup tables 122 are referenced by the media optimization module 120 to configure the settings of the encoder module 116 and the network module 118. In some aspects, the media optimization module 120 accesses three separate lookup tables 122 to configure the video encoding and transmission operations. These tables may include a Residual Packet Loss (RPL) table, a video parameter table, and a forward error correction table. Methods for generating these tables are described further below (see Figures 2-4).

[0031] The client device 108 is operable to store and/or display video content as received from the media optimization server 104. The client device 108 may be any device capable of managing data requests via the network 106. Examples of such client devices include a personal computer (PC), a mobile device, or a server. The client device 108 may also include a personal computer, a personal digital assistant ("PDA"): a tablet PC, a netbook, a smart phone, etc. Indeed, client devices in accordance with the systems and methods described herein may include any device operative to process instructions and transmit data to and from humans and other computers such as general purpose computers, network computers lacking local storage capability, etc.

[0032] The network 106, and the intervening nodes between the server 104 and the client device 108 may include various configurations and use various protocols such as the Internet, World Wide Web, intranets, virtual private networks, local Ethernet networks, private networks using communication protocols proprietary to one or more companies, cellular and wireless networks (e.g., Wi-Fi), instant messaging, hypertext transfer protocol ("HTTP") and simple mail transfer protocol ("SMTP"), and various combinations of the foregoing. While only one client device 108 and server 104 are shown in Fig. 1,

it should be appreciated that a large number of connected computers may be included on the system.

[0033] Although certain advantages are obtained when information is transmitted or received as noted above, other aspects of the system and method are not limited to any particular manner of transmission of information. For example, in some aspects, information may be sent via a medium such as an optical disk or portable drive. In other aspects, the information may be transmitted in a non-electronic format and manually entered into the system.

[0034] Figure 2 is a method 200 for generating an RPL table in accordance with aspects of the invention. The RPL table includes a set of data describing the effectiveness of FEC code for a given amount of network packet loss and a given network rate. The RPL represents the received packet loss minus the average recovery rate provided by the FEC code. RPL may be defined by the following function:

$$RPL = p - R(p, b, k, n) \quad (\text{Eq. 1})$$

where RPL is the residual packet loss, $R(p, b, k, n)$ is the effective recovery rate, p is the input/received packet loss, b is the burst length of the packet loss, k is the total number of source packets, and n is the total number of packets.

[0035] During the optimization process, the quantities (p , b) are input to Eq.1. The quantities may be received periodically as the media optimization module 120 receives updated network statistics. The process by which the media optimization module 120 performs the table lookup operation is described further below (see Figures 2-4).

[0036] The total number of packets, n , is related to the network bandwidth (total encoding rate) via the function:

$$n = \frac{R_t}{(FR \cdot MTU)} \quad (\text{Eq. 2})$$

where MTU is the maximum network packet size (~1400 bytes, for example), R_t is the network rate, and FR is the average frame rate of the incoming video. The amount of FEC packets is given as $n-k$. The FEC redundancy, (Δ , the protection ratio provided by FEC) is defined as:

$$\Delta = \frac{(n-k)}{n} \quad (\text{Eq. 3})$$

[0037] Using the above definitions, the residual packet loss may be expressed as:

$$RPL(p, b, \frac{R_t}{FR}, \Delta) = p - R(p, b, \frac{R_t}{FR}, \Delta) \quad (\text{Eq. 4})$$

[0038] The three parameters, (p, b, R_t) are input to the model to determine an effective recovery rate.

[0039] The effective recovery rate, $R(p, b, \frac{R_t}{FR}, \Delta)$, may be computed for different packet loss scenarios, including random packet loss, bursty packet loss, or constant packet loss. A loss model for the packet loss events is used to determine the effective recovery rate based upon the packet loss, the number of source packets, and the number of FEC packets. Different loss models, such as a random packet loss model, a bursty packet loss model, or a consistent packet loss model, may result in different effective recovery rates. For the purposes of notational simplicity, a random (i.e. non-bursty) loss model is used, removing the term b . In aspects involving a bursty loss model, the term b would be relevant to the calculation of the effective recovery rate. The effective recovery rate is defined as:

$$R(p, b, \frac{R_t}{FR}, \Delta) = R(p, b, k, n) = \frac{1}{n} \sum_{t=1}^n P(t, p, b, n) A(t, k, n) \quad (\text{Eq. 5})$$

where $P(t, p, b, n)$ is the probability of losing t packets (from n total packets), given the packet loss rate p and the burst length b , and $A(t, k, n)$ is the average number of packets that can be recovered when t packets (from n total

packets) are lost. As mentioned above, $n = \frac{R_t}{(FR \cdot MTU)}$ and

$\Delta = \frac{(n-k)}{n}$. The quantity $A(t, k, n)$ is the average number of packets that can be recovered from the FEC when t packets are lost, and can be obtained directly from counting possible loss configurations given the FEC code. The quantity $P(t, p, b, n)$ is the probability of losing t packets, which depends on the packet loss model. Example models for the packet loss events are random loss (Bernoulli process), or bursty model (two-state Markov process). Other more complicated packet loss models may be used. The feedback of the network state is obtained periodically and is typically an effective packet loss rate (p) and possibly a burst length (b).

[0040] The RPL table is computed by determining the effective recovery rate R over a range of the parameters it depends on, such as the packet loss (p), network rate ($\frac{R_t}{FR}$), and the FEC redundancy factor (Δ). In some aspects, the table is generated by performing a computing the RPL functions for a range of each of these factors. An example of such a table follows, where R is a selected packet loss model:

Packet Loss	Bit rate	FEC Redundancy	Effective Recovery Rate	Residual Packet Loss
P_1	$\frac{R_t}{FR}_1$	Δ_1	$R_1 = R(p_1, b_1, \frac{R_t}{FR}_1, \Delta_1)$	$RPL(p_1, b_1, \frac{R_t}{FR}_1, \Delta_1) = p_1 - R_1$
P_2	$\frac{R_t}{FR}_2$	Δ_2	$R_2 = R(p_2, b_2, \frac{R_t}{FR}_2, \Delta_2)$	$RPL(p_2, b_2, \frac{R_t}{FR}_2, \Delta_2) = p_2 - R_2$

Table 1.

[0041] The method 200 determines an effective recovery rate and residual packet loss for a particular packet loss model. The effective recovery rate is based upon possible input values for packet loss, packet loss burst length, number of source packets, and number of overall packets. The effective

recovery rates and RPL values may be stored in a lookup table for use in the video and FEC lookup table generation process (see Figures 3-4) and the media optimization process (see Figure 5).

[0042] At block 202, a packet loss model, is specified. The packet loss model is associated with a particular effective recovery rate, R . The packet loss model describes network behavior as a result of various factors, such as the recovery rate and resistance to packet loss of a set of data transmitted with given parameters. Although the method 200 is described with respect to a single packet loss model, the method 200 may also compare recovery rates over multiple models.

[0043] At block 204, a range for packet loss values, network rate values, and FEC redundancy factor values is selected. For example, the packet loss values p may be computed for a range from 0% packet loss to 50% in steps of 1%, the network rate values, $\frac{R_i}{FR}$, may be calculated from 10 kilobits to 500 kilobits, in steps of 10 kilobits, and the FEC redundancy factor value, Δ , may be computed from 0% redundancy to 50% redundancy, in steps of 1%. These exemplary ranges represent one possible aspect of the invention, and other ranges could be used as appropriate. For example the packet loss may be incremented in .5%, 10%, or 2% steps. Further, the bit rate may be incremented from 25 through 1000 kilobits in 5 kilobit increments. As an even further example, the FEC redundancy factor may be incremented in .5%, 10%, or 2% steps.

[0044] At block 206, a stepping value for each of the packet loss, network rate, and FEC redundancy factor is selected. A smaller stepping value may result in more values for the RPL function present within the data table. In this regard, a more accurate representation of the RPL function for a given dataset may be provided. As described above, for example, the packet loss values may be stepped in 1%

increments, 2% increments, or 5% increments. For example, the network rate may be stepped in 1 kilobit increments, 10 kilobit increments, or 25 kilobit increments. For example, the FEC redundancy factor may be stepped in 1% increments, 2% increments, or 5% increments. In some aspects, a burst parameter, *b*, is further included in the table to model for different types of packet loss. For example, as shown in Table 1 below,

[0045] At block 208, an effective recovery rate for each of the stepping values for packet loss, network rate, and FEC redundancy factor is calculated. An effective recovery rate for each possible combination of packet loss, network rate, and FEC redundancy value is calculated to be stored in the data table.

[0046] At block 210, the effective recovery rates and associated residual packet loss values are stored in a lookup table, such as the RPL table. The lookup table is indexed by the packet loss, network rate, and FEC redundancy factors that determined each effective recovery rate and residual packet loss value. The RPL table is thus indexed by packet loss, bit rate, and FEC redundancy to provide a set of values for the effective recovery rate and RPL. In some aspects, the RPL table may be associated with multiple packet loss models, and each model may include a separate further table for looking up effective recovery rates.

[0047] Figure 3 is a method 300 for generating a video model parameter table in accordance with aspects of the invention. The method 300 determines a set of optimal parameters to determine a distortion function for a given content class of video. Videos are classified as a particular content class based upon various characteristics, such as the amount of motion in the video, the resolution of the video, a spatial texture level of the video, the color depth of the video, and the like. Once generated, the lookup table may be

used to determine an optimal set of parameters to best fit simulated distortion values of a given content class of videos to a distortion function. These parameters are stored in the lookup table for use in the FEC lookup table generation process (see Figure 4) and the media optimization process (see Figure 5).

[0048] The method 300 begins at block 302, with a plurality of video clips. The video clips are analyzed to extract a set of characteristics of the videos. For example, the videos may be given ratings for the spatial resolution of the video, the frame rate of the video, the amount of motion in the video, the spatial texture level of the video, or the color depth of the video. In some aspects, shorter segments of each video may be analyzed independent of the entire clip. In some aspects, the characteristics are extracted from the video using a preprocessor. For example, the video may be analyzed to determine a motion level. The motion level may be determined by averaging the magnitude of all non-zero motion vectors for each analyzed frame. This averaged value is denoted as the MotionMag of the video. The motion vectors for the input frame may be computed from a pre-processing of the input. A motion level may be defined by its relation to a threshold value. For example, the video may be given a motion level of 2 if MotionMag is ≥ 1.5 , a motion level of 1 if MotionMag is between 1.5 and .25, and a motion level of 0 if MotionMag is less than or equal to .25. A system and method for pre-processing of video to extract motion levels and other characteristics is described in related U.S. Patent Application No. 13/097,267, filed April 29, 2011, entitled, "System And Method For Providing Content Aware Video Adaptation," which is herein incorporated by reference in its entirety. The characteristics are associated with the particular video clip that they were extracted from, and the method 300 proceeds to block 304.

[0049] At block 304, the video clips are associated with a particular content class. A content class corresponds to one or more of the characteristics as extracted at block 302. For example, a given video may be mapped to a content class of all videos with a 640x480 resolution, all videos with a particular motion level, or all videos with a 1920x1080 resolution, a particular motion level, and a 16 bit color depth.

[0050] At block 306, a video codec is simulated and mean-square error (MSE) distortion values are computed for each video of a given content class over a range of simulated network rates, packet loss values, and FEC redundancy factors. This simulation may be performed using an error resilience setting associated with the particular video codec. The error resilience setting determines the amount/percentage of macro-block intra-coding on the intermediate frames of the video. The more intra-coding, the more resilient the video is to packet loss. The error resilience of a given video is determined based on the effective recovery rate associated with the simulated values for packet loss, network rate, and FEC value, and the motion level of the video. In one aspect, the ER setting is determined as $ER = p - \mu(ML)R$ where p is the network packet loss, R is the effective recovery rate defined above and $\mu(ML)$ modulates the setting based on the motion level in the scene. This function allows for control of the ER for sources with different motion content. For example, $\mu(ML)$ may equal 1 for $ML=0$, $\mu(ML)$ may equal 0.5 for $ML=1$, and $\mu(ML)$ may equal .25 for $ML=2$. In another aspect, the ER setting may be the RPL scaled with a motion level parameter such that $ER = \mu_2(ML)RPL$ where μ_2 is smaller for lower motion levels and larger for higher motion levels. The distortion values for each simulated set of variables are stored and associated with the video content analyzed as part of the content class.

[0051] At block 308, the best model parameters for each content class are extracted. These parameters are extracted by fitting the distortion values of the videos of the content class to a distortion function. The model parameters of the distortion function may describe the contributions to the distortion from various effects, such as (1) intrinsic coding distortion (coding artifacts in absence of packet loss), (2) distortion changes from ER settings (more intra-blocks generally trade-offs quality with robustness) and (3) distortion from actual packet loss. For example, the model distortion function may be

$$J = D = \alpha_i R_s^{\beta_i} + f(ML) \gamma_i RPL \quad (\text{Eq. 6})$$

where D is the distortion function, R_s is the source encoding rate, related to the network rate and FEC redundancy as $R_s = R_i(1-\Delta)$, $f(ML)$ is a weight value based on the motion level of the video, RPL is the residual packet loss associated with the simulated parameters, and α_i, β_i and γ_i are the model parameter values. The first two parameters may capture the intrinsic coding distortion and effects of ER, while the third parameter may describe additional distortion from the effects of packet loss, for a best fit of the distortion values over a range of simulated values for packet loss, network rate, and FEC redundancy factor. The weight value for motion level may be determined by associating a particular weight with each motion level, such as 1 for a motion level of 0, 1.5 for a motion level of 1, or 2.0 for a motion level of 2. In this manner, the distortion function is weighted to compensate for the fact that videos with more motion are more susceptible to distortion as a result of packet loss. Although the methods described herein are described with respect to a particular distortion function, it could also be applied to other distortion functions that model for different network interactions. For example, in addition to taking into account the motion level, the distortion function may factor in

effects of color distortion. An example could include a weighted distortion function, with different weights for each color component.

[0052] As a further example of an alternate distortion model, if NACK is enabled on the network, the total distortion or cost function might be modified to incorporate a delay term as follows:

$$J = \alpha R_s^\beta + g\left(\frac{RTT}{T_{rtt}}\right) f(ML) \gamma RPL + \frac{1}{w} (T_{nack} + T_{fec}) \quad (\text{Eq. } 7)$$

where w is the average wait time before a frame can be rendered on the receiver, T_{nack} is the delay associated with sending a NACK notification, and T_{fec} is the delay associated with the redundant transmission of FEC packets. The quantity RTT is the network roundtrip time, and $g\left(\frac{RTT}{T_{rtt}}\right)$ is a smooth function that varies from a small number ($\ll 1$) for RTT smaller than some threshold T_{rtt} , and increases close to 1 as RTT increases to T_{rtt} . This threshold parameter is application dependent, but in some examples may be $T_{rtt} \sim 100\text{ms}$, 200ms , 250ms , or the like. The meaning of the third term in (7) is that for small waiting time w , the third term becomes large and so the cost of the delay is important in the cost function (i.e., decoder/receiver does not want to wait). If the waiting time is large, then the cost of delay from NACK and FEC is less important is less important to the total cost function (i.e., decoder/receiver can afford to wait).

[0053] At block 310, the method 300 stores the model parameters α_i, β_i and γ_i in a lookup table indexed by the content class with which they are associated. In this manner the method 300 generates a lookup table for given model parameters as associated with a particular content class of video. A video parameter lookup table as generated by the method 300 might have a structure as such:

Resolution	Motion Level	α	β	γ
640x480	0	α_1	β_1	γ_1
640x480	1	α_2	β_2	γ_2
1920x1080	0	α_3	β_3	γ_3
1920x1080	1	α_4	β_4	γ_4

Table 2.

[0054] In this example, the content class is defined by the resolution and motion level of the video. This table may be used to look up the parameters α_i , β_i , and γ_i associated with a video of a given resolution and motion level. The method 300 ends after constructing the table for all content classes.

[0055] Figure 4 is a method 400 for generating a forward error correction (FEC) table in accordance with aspects of the invention. The method 400 operates to create a lookup table including optimal FEC values for different content classes of videos and network state parameters. Although an increase in the FEC value results in an increased ability of the receiver/decoder to retrieve or compensate for lost packets, the extra data transmitted results in less actual frame data being transmitted to the receiver. In situations where the network is not losing packets, extra FEC results in a lower data bit rate due to the transmission of redundant data. As such, it is necessary to optimize the FEC factor based on the content of the video (i.e., how much can the video afford to lose a particular frame) and the network bit rate and packet loss (i.e. the cost overhead in transmitting redundant data). As such, the FEC factor is determined as a function of both video content and network factors.

[0056] At block 402, an RPL table is generated, such as by the method described with respect to Figure 2. The RPL table is a table of values that describe an RPL value associated with particular values of packet loss, network rate, forward

error correction, and the like. At block 404, a video parameter table is generated, such as by the method 300 of Figure 3. The video parameter table is a table indexed by video content classes to define a set of parameters for associating a particular video with a particular cost function. The parameters define the cost function.

[0057] At block 406, a content class is selected from the video parameter table. The content class defines a cost function, such as the cost function $D = \alpha_i R_s^{\beta_i} + f(ML)\gamma_i RPL$ described above, where the parameters α_i, β_i , and γ_i are numerical values associated with the particular content class. As above, the cost function could also be the alternate cost function associated with NACK, or any other cost function.

[0058] At block 408, an optimal FEC factor is calculated as a function of the content class and a set of simulated network statistics. The optimal FEC factor is calculated by determining a minimum of the cost function, such as the cost function D , for a given set of simulated network statistics. The content classes defined within the video parameter table have an optimal FEC factor calculated for each possible set of network statistics, such as network packet loss, network rate, and round trip time. Taking the minimum of the cost function in this manner ensures that the method compensates for the cost tradeoffs associated with increased FEC protection. Since the RPL is a function of the FEC protection factor and the network parameters, there is a dependency of the protection factor within the cost function. The cost function also depends on the FEC protection factor from the source encoding rate from the first term of Equation 6 or 7, $R_s = R_i(1 - \Delta)$.

[0059] In some aspects, the FEC redundancy factor is computed separately for key (intra) frames and P (delta/prediction) frames. Key frames include an entire image, while P frames include deltas from previous frames. As

such, key frames are more important to protect than P frames, so it is important to include extra FEC protection for key frames. For example, a scale parameter greater than 1 may be provided to the FEC factor for key frames, or key frames may be automatically associated with a higher level FEC factor.

[0060] At block 410, the FEC factor as calculated at block 408 is stored in a FEC table. The FEC table is a lookup table that is indexed by a set of network parameters and a set of content classes of videos. A FEC table as generated by the method 400 might have a structure as such, where the table is indexed on the first four columns to determine the FEC protection factor:

Network Rate	Packet Loss	Cost Function Parameters	Motion Level	FEC Protection Factor
--------------	-------------	-----------------------------	--------------	-----------------------------

Table 3.

[0061] Figure 5 is a method 500 for providing adaptive media optimization in accordance with aspects of the invention. In some aspects, the method 500 is performed by a media optimization module such as the media optimization module 120. The method 500 is operable to analyze content of a video and statistics from a network and optimize encoding and transmission operations of an encoder module and a network module, respectively based on the content and statistics. The method 500 provides for efficient analysis such that it is suitable for use in real-time systems.

[0062] At block 502, the media optimization module 120 determines a content class of a video. The content class is typically determined by analyzing frames of the video for various characteristics, such as the video resolution, frame rate, motion level, spatial texture level, and color depth. Each analyzed characteristics may be used to select a particular content class for the video. In some aspects, the content class is determined by a preprocessor that examines the video and sends content class information to the media

optimization module 120. For example, a given video may have a resolution of 640x480, a frame rate of 30 frames/second, and a motion level of 1. In some aspects, the preprocessing operations are performed by the capture module 114.

[0063] At block 504, the media optimization module 120 selects a set of model parameters based on the content class of the video determined during block 502. The content class is determined as discussed above, by analyzing frames of the video for various characteristics, such as the video spatial resolution, frame rate, motion level, spatial texture level, and color depth. To obtain the video parameters, for example, the video content described with respect to block 502 would be looked up in the table by finding the values associated with a 640x480 resolution, a 30 frames/second frame rate, and a motion level of 1. These values might direct the method to the set of model parameters α_i , β_i , and γ_i as associated with a distortion function for the content class.

[0064] At block 506, the media optimization module 120 determines a set of network characteristics. The network characteristics may include a packet loss value, a round-trip-time to a receiver, a network rate, and the like. In some aspects, these network characteristics are determined using Real-Time Transport Control Protocol (RTCP). RTCP is a network protocol that can be used to provide feedback on network quality, including the network characteristics described above. In some aspects, the media optimization module 120 is responsible for creating and receiving RTCP requests. In some aspects, the RTCP requests and the data received in response may be provided by another aspect of the computer system 104, such as the network module 114.

[0065] At block 508, robustness options for the encoding and transmission of the video are selected in response to the network conditions and the identified distortion function parameters. The robustness options are selected by performing

table lookups on an RPL table and a FEC table, such as the RPL table generated by the method 200 and the FEC table generated by the method 400. The FEC table provides for a FEC factor based on the network statistics and the model parameters. The FEC factor may then be used to perform a lookup operation on the RPL table, where error resilience settings are determined as a function of the network statistics the FEC factor, and the motion level of the scene. The FEC factor and the network rate determines the encoding source rate, because the more of the network rate that is consumed by FEC protection factor, the less can be devoted to transmission of the source video. As such, the method 500 also optimizes the source rate as a function of the determined FEC factor, network rate, and video content. The FEC factor and ER settings determine the maximum source rate, because the more of the network rate that is consumed by error correction and avoidance methods, the less can be devoted to transmission of the source video. As such, the method also optimizes the source rate as a function of the determined FEC factor and ER setting.

[0066] At block 510, the FEC factor, the ER setting, and the source rate as determined at block 508 are used to optimize the encoding and transmission of the video. The optimization process is described further below (see Figure 6).

[0067] The stages of the illustrated methods described above are not intended to be limiting. The functionality of the methods may exist in a fewer or greater number of stages than what is shown and, even with the depicted methods, the particular order of events may be different from what is shown in the figures.

[0068] Figure 6 is a block diagram depicting data flow throughout a system 600 for providing adaptive media optimization in accordance with aspects of the invention. The system 600 includes a capture module 602, an encoder module

604, a network module 606, and a media optimization module 608.

[0069] The capture module 602 receives a video capture from a video source, such as the video source 102. As described above, the source may be any type of device capable of supplying a video file or stream. The capture module 602 performs an analysis of the content of the video and sends the analysis to the media optimization module 608. The analysis includes various content characteristics of the video, such as the frame rate, the resolution, the motion level, the spatial texture level, the color depth, and the like. As described above, the capture module 602 may transmit the unencoded video directly to the media optimization module 608, and the media optimization module 608 may perform the content analysis. The media optimization module 608 uses the characteristic analysis to determine a content class of the video. The content class is then used to perform a lookup operation to determine a set of distortion model parameters for the video.

[0070] The media optimization 608 also receives a set of network statistics, such as from RTCP. The network statistics are used in conjunction with the distortion model parameters to map the video and network state to a FEC factor and ER setting. The FEC and ER setting are determined by a lookup operation on the FEC table, the RPL table, and the motion level of the scene. The FEC factor and network statistics are then used to determine an ER setting in an RPL lookup table. After determining the FEC factor and ER settings for the video, the media optimization module 608 determines the encoding source bit rate for the video by subtracting the rate needed to preserve the FEC factor from the maximum/allowed network transmission bit rate. In some aspects, the media optimization module 608 may also control whether NACK is used, based upon the distortion function associated with the video content class. In some aspects, whether NACK is used is a

setting as configured by the user. In such cases, different lookup tables may be provided for distortion models that account for NACK being enabled.

[0071] The media optimization module sends the determined source rate and ER setting to the encoder module 604 to encode the video with these settings. The media optimization module sends the determined FEC factor to the network module to ensure the video is transmitted with the determined FEC factor.

[0072] The encoder module 604 encodes the video with the settings as specified by the media optimization module 608. After encoding the video, the encoder module 604 sends the encoded video to the network module 606.

[0073] The network module 606 packetizes and transmits the encoded video in accordance with the FEC protection settings as specified by the media optimization module 608. In some aspects, the network module 606 also receives instructions from the media optimization module to enable NACK. The video is sent to the receiver from the network module 606 as one or more packets.

[0074] The systems and methods described herein advantageously provide optimized encoding and transmission of video. By analyzing the video for a content class, mapping the content class to a particular distortion function, and using network statistics to determine optimal transmission and encoding settings, aspects of the invention provide for an optimal mix of video robustness and quality. By accounting for the effect that various network settings have on different types of video, aspects of the invention ensure that the video is encoded and transmitted in the most efficient manner possible.

[0075] As these and other variations and combinations of the features discussed above can be utilized without departing from the invention as defined by the claims, the foregoing

description of the embodiments should be taken by way of illustration rather than by way of limitation of the invention as defined by the claims. It will also be understood that the provision of examples of the invention (as well as clauses phrased as "such as," "e.g.," "including" and the like) should not be interpreted as limiting the invention to the specific examples; rather, the examples are intended to illustrate only some of many possible embodiments.

CLAIMS

1. A computer-implemented method for providing adaptive media optimization, the method comprising:

determining, using a processor, a content class of a video based on one or more content characteristics of the video, wherein the content characteristics comprise a motion level of the video;

determining a set of network conditions;

configuring at least one of an encoding option or a transmission option based upon the content class and the network conditions; and

encoding and transmitting the video in accordance with the configured encoding option or transmission option.

2. The method of claim 1, wherein the encoding option is at least one of a source rate or an error resilience (ER) setting.

3. The method of any of claims 1-2, wherein the transmission option is at least one of a forward error correction (FEC) factor or a negative-acknowledgement (NACK) setting.

4. The method of any of claims 1-3, further comprising:

selecting a set of parameters for a distortion function based upon the content class; and

using the selected parameters to configure the at least one encoding option and transmission option.

5. The method of any of claims 1-4, further comprising sampling the video to identify the one or more content characteristics.

6. The method of claim 5, wherein the content characteristics further comprise at least one of a frame rate, a resolution, a spatial texture level, or a color depth.

7. The method of any of claims 5-6, wherein the content class of the video is determined by looking up the content characteristics in a video parameter lookup table.

8. The method of any of claims 1-7, wherein the transmission option is configured by performing a table lookup on a FEC table indexed by the set of network statistics and the content class of the video.

9. The method of any of claims 2-8, wherein the encoding option is an ER setting configured by performing a table lookup on the content class of the video and a Residual Packet Loss (RPL) table indexed by the set of network statistics, and a FEC protection factor.

10. The method of claim 9, wherein the ER setting is determined as a function of a packet loss value, an effective recovery rate, and the content class.

11. The method of claim 10, where the ER setting is determined by at least one of a function $ER = p - \mu(ML)R$ or $ER = \mu_2(ML)RPL$, wherein ER is an error resilience value, p is a packet loss value, $\mu(ML)$ and $\mu_2(ML)$ are functions based on a motion level value, ML, R is an effective recovery rate and RPL is a residual packet loss value.

12. The method of any of claims 1-11, wherein the network statistics comprise a packet loss value, a round trip time value, and a network rate.

13. The method of any of claims 1-12, wherein the network statistics are provided by Real-Time Transport Control Protocol (RTCP).

14. A method for generating a video parameter lookup table, the method comprising:

- extracting one or more source content characteristics from a plurality of source videos;

- mapping, using a processor, one or more of the plurality of source videos to a content class based on the extracted source characteristics;

- computing, for a given set of source videos mapped to a same content class, a distortion value over a range of simulated network rates, packet loss values, error resilience settings, and forward error correction (FEC) protection factors;

- determining nearest fit parameters to map a distortion value for each video in the given set of source videos to a common distortion function; and

- storing the nearest fit parameters in the video parameter lookup table, wherein the video parameter table is indexed by content class.

15. The method of claim 14, wherein the error resilience settings are derived from a Residual Packet Loss (RPL) lookup table and a source motion level, wherein the source motion level is derived from the extracted content characteristics.

16. A method for generating a Residual Packet Loss (RPL) lookup table, the method comprising:

- specifying a packet loss model;

- selecting a range for a packet loss value, a network rate, and a forward error correction (FEC) redundancy factor;

selecting one or more stepping values for the packet loss value, the network rate, and the FEC redundancy factor;

computing, using a processor, an effective recovery rate for one or more combinations of the range of the packet loss value, the network rate, and the FEC redundancy factor using the stepping values; and

storing the computed effective recovery rate in a lookup table indexed by the packet loss value, the network rate, and the FEC redundancy factor.

17. A method for generating a forward error correction (FEC) lookup table, the method comprising:

generating a residual packet loss (RPL) lookup table;

generating a video parameter lookup table;

selecting a content class from the video parameter lookup table;

calculating, using a processor, a FEC redundancy factor by minimizing a cost function for the selected content class and a simulated network state; and

storing the FEC redundancy factor in a FEC table indexed by the content class and simulated network state.

18. A processing system for providing adaptive media optimization comprising:

at least one processor;

memory for storing a video parameter lookup table, a residual packet loss (RPL) lookup table, and a forward error correction (FEC) lookup table, the memory coupled to the at least one processor; and

a media optimization module associated with the at least one processor to:

identify one or more characteristics of a video, determine a content class of the video based on the one or more characteristics,

retrieve one or more distortion model parameters from the video parameter lookup table based upon the content class,

determine a set of network conditions, retrieve a FEC redundancy factor from the FEC lookup table based on the set of network conditions and the content class, and

retrieve an error resilience (ER) setting from the content class of the video and the RPL table based on the FEC redundancy factor, and the set of network conditions.

19. The processing system of claim 18, wherein the media optimization module further determines a source encoding rate for the video based upon the maximum network rate, the FEC redundancy factor, and the ER setting.

20. The processing system of claim 19, further comprising an encoder module to receive the source encoding rate and the ER setting from the media optimization module, and to encode the video in accordance with the source encoding rate and the ER setting.

21. The processing system of any of claims 18-20, further comprising a network module to receive a FEC redundancy factor from the media optimization module and to transmit the video in accordance with the FEC redundancy factor.

22. The processing system of any of claims 18-21, wherein the set of network conditions comprise a packet loss value, a network rate, and a round trip time.

23. The processing system of any of claims 18-22, wherein the one or more characteristics comprise at least one of a resolution, a frame rate, a motion level, a spatial texture level, or a color depth.

24. A processing system for performing the method of any of claims 1-17.

25. A non-transitory computer readable storage medium storing instructions that, when executed by a processor, cause the processor to perform the method of any of claims 1-17.

1/6

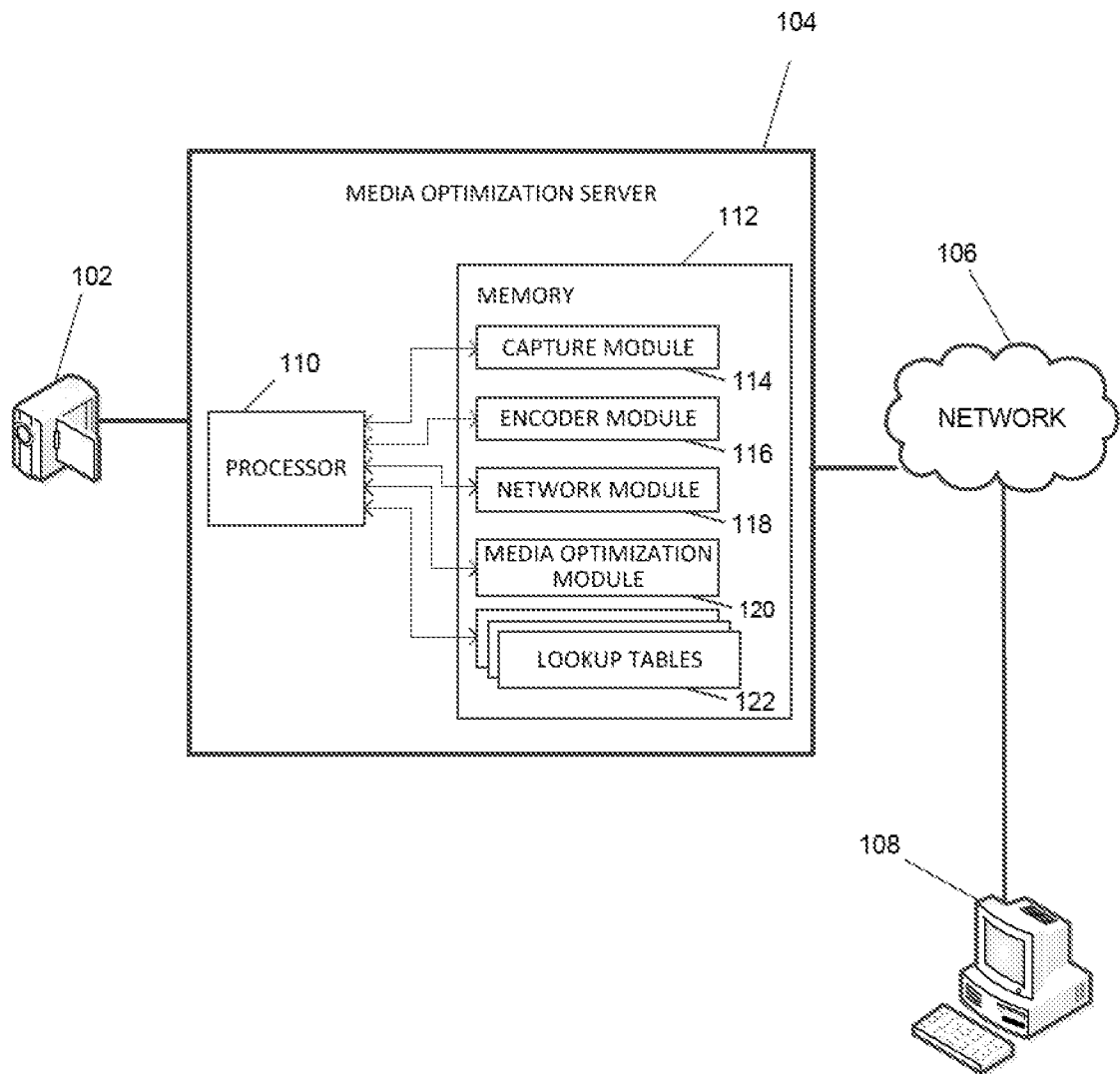
100

FIG. 1

2/6

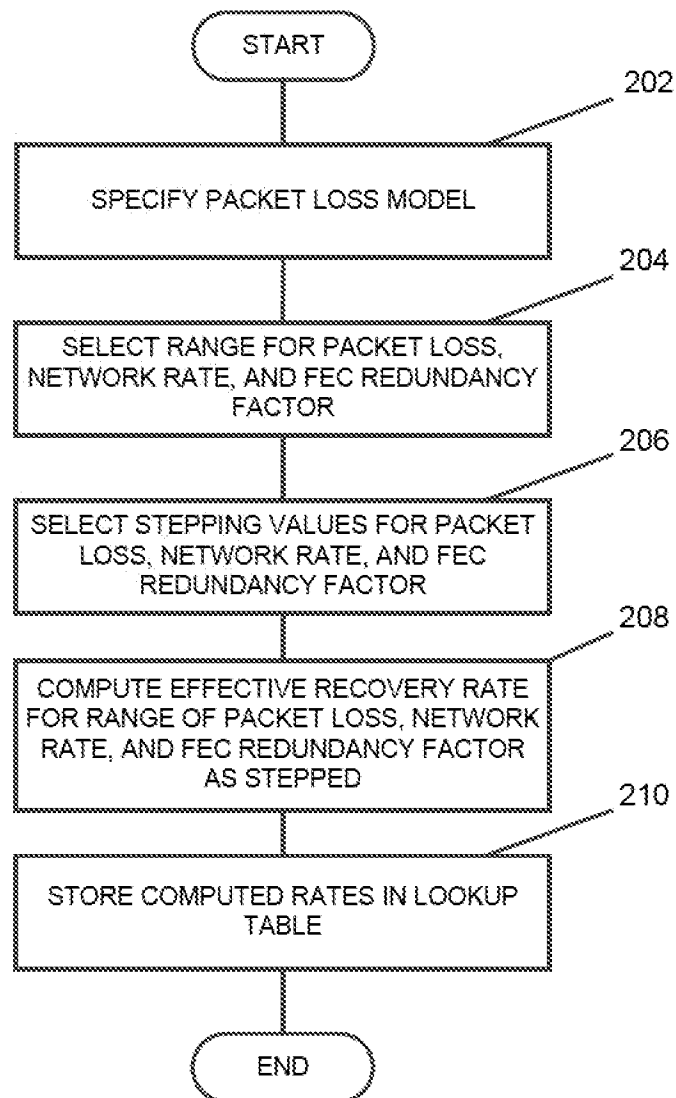
200

FIG. 2

3/6

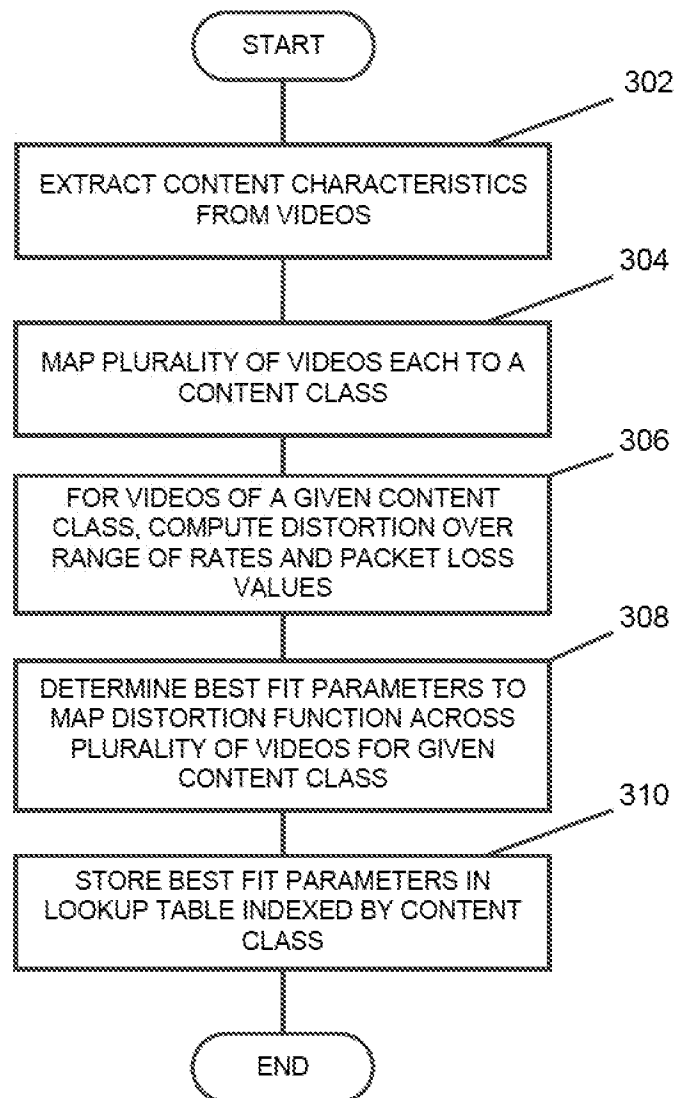
300

FIG. 3

4/6

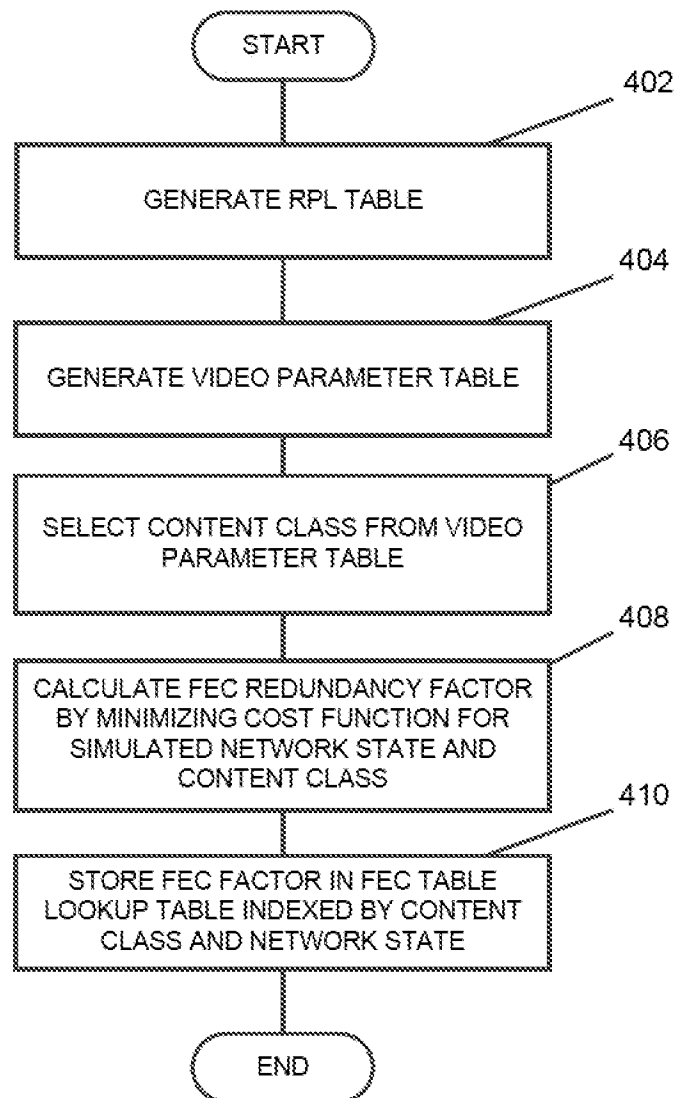
400

FIG. 4

5/6

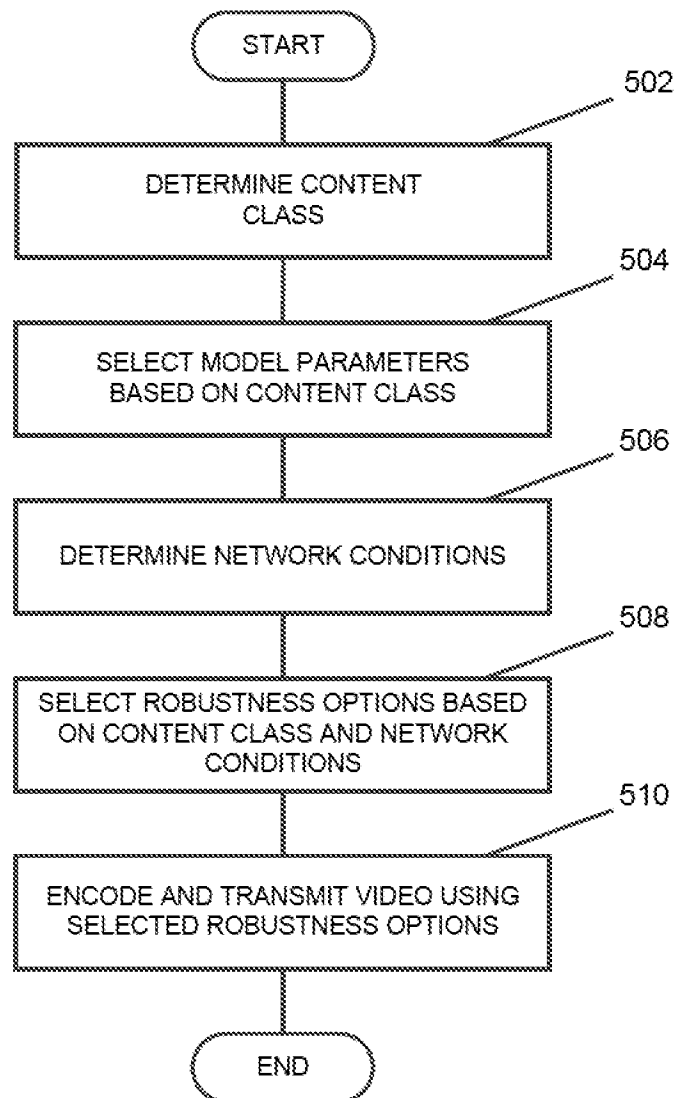
500

FIG. 5

600

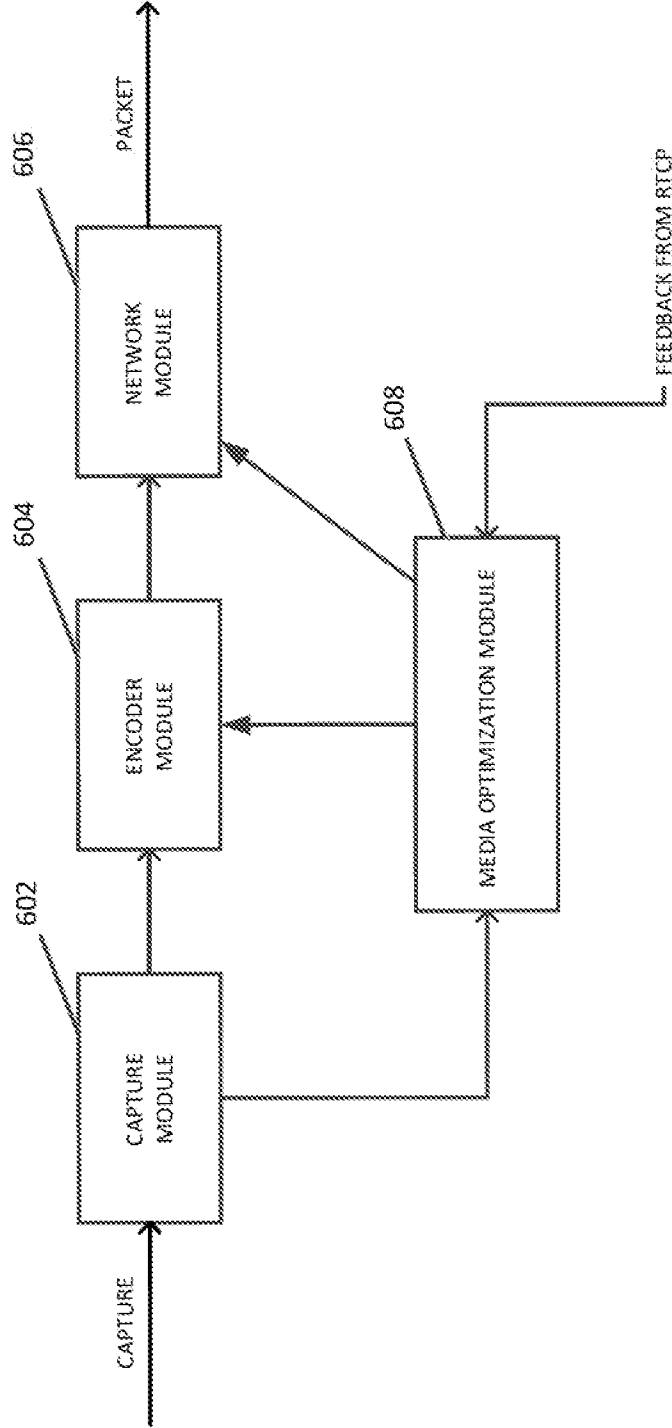


FIG. 6