



(51) International Patent Classification:

H04N 21/24 (2011.01) H04N 21/233 (2011.01)  
H04N 21/2662 (2011.01) H04N 21/643 (2011.01)  
H04L 29/06 (2006.01)

(21) International Application Number:

PCT/SG2012/000038

(22) International Filing Date:

10 February 2012 (10.02.2012)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/490,125 26 May 2011 (26.05.2011) US

(71) Applicant (for all designated States except US):

**AGENCY FOR SCIENCE, TECHNOLOGY AND RESEARCH** [SG/SG]; 1 Fusionopolis Way #20-10, Connexis, Singapore 138632 (SG).

(72) Inventors; and

(75) Inventors/Applicants (for US only):

**BAO, Xiaoming** [SG/SG]; c/o PKM, Institute for Infocomm Research, 1 Fusionopolis Way #21-01, Connexis South Tower, Singapore

138632 (SG). **YU, Rongshan** [SG/SG]; c/o PKM, Institute for Infocomm Research, 1 Fusionopolis Way #21-01, Connexis South Tower, Singapore 138632 (SG). **RA-HARDJA, Susanto** [ID/SG]; c/o PKM, Institute for Infocomm Research, 1 Fusionopolis Way #21-01, Connexis South Tower, Singapore 138632 (SG).

(74) Agent: **VIERING, JENTSCHURA & PARTNER LLP;**

P. O. Box 1088, Rochor Post Office, Rochor Road, Singapore 911833 (SG).

(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.

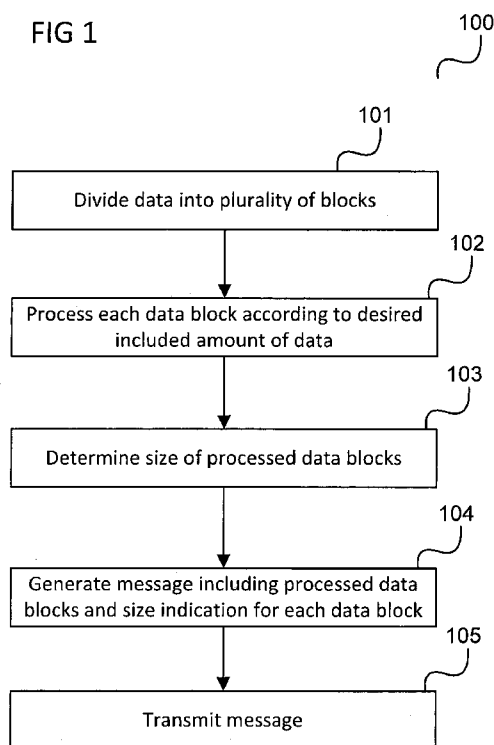
(84) Designated States (unless otherwise indicated, for every

kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ,

[Continued on next page]

(54) Title: METHODS FOR TRANSMITTING AND RECEIVING A DIGITAL SIGNAL, TRANSMITTER AND RECEIVER

FIG 1



(57) Abstract: According to one embodiment, a method for transmitting a digital signal is provided that includes dividing data representing the digital signal into a plurality of data blocks, processing each data block in accordance with a desired amount of data included in the data block, determining, for each processed data block, the size of the processed data block, generating a message including, in a message body of the message, the processed data blocks and, for each data block, a message field specifying the size of the processed data block and transmitting the message.

WO 2012/161652 A1

UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG). **Published:**  
— *with international search report (Art. 21(3))*

**Methods for transmitting and receiving a digital signal,  
transmitter and receiver**

Field of the invention

5

Embodiments of the invention generally relate to a method for transmitting a digital signal, a method for receiving a digital signal, a transmitter and a receiver.

10 Background of the invention

With the "HTTP live streaming protocol" and the latest efforts on standardizing "HTTP Streaming of MPEG Media" and HTML5, streaming multimedia over HTTP can be expected to be the trend in the future. There has been increasing demand from industries for efficient delivery of streaming multimedia over HTTP. The current "progressive download" technology used for HTTP streaming may be needed to be upgraded to address the new requirements such as dynamic adaptation of media content in domains of quality/fidelity during delivery based on network conditions and resource capabilities (i.e. adaptive streaming). Specifically, such an upgrade may be of high importance for enabling streaming to mobile devices due to more stringent resource constraints and bandwidth fluctuations of wireless networks.

The concept of adaptive streaming has already existed in a number of commercial streaming systems. In these systems, adaptive streaming is typically implemented with pre-encoding of the same media content into multiple files with different streaming qualities. During the streaming session, the file that best matches the current network conditions is selected as the streaming file. Such an approach not only takes up

additional storage space but also complicates the database management at the server side when hosting a large amount of media contents. Besides, this kind of "multiple sources" method usually only provides a few different stream qualities at what can be seen as "coarse granularity" such as low, medium and high to avoid maintaining too many source files for a piece of same media content. Furthermore, the choosing of the stream quality is typically done at the beginning of the transmission of the stream because there is usually no continuous bandwidth monitoring available at the server side.

#### Summary of the invention

In one embodiment, a method for transmitting a digital signal is provided that includes dividing data representing the digital signal into a plurality of data blocks, processing each data block in accordance with a desired amount of data included in the data block, determining, for each processed data block, the size of the processed data block, generating a message including, in a message body of the message, the processed data blocks and, for each data block, a message field specifying the size of the processed data block and transmitting the message.

According to various embodiments, a receiving method corresponding to the transmitting method described above, a corresponding transmitter and a corresponding receiver are provided.

Short description of the figures

Illustrative embodiments of the invention are explained below with reference to the drawings.

5

Figure 1 shows a flow diagram according to an embodiment.

Figure 2 shows a transmitter for transmitting a digital signal according to an embodiment.

10

Figure 3 shows a flow diagram according to an embodiment.

Figure 4 shows a receiver for receiving a digital signal according to an embodiment.

15

Figure 5 shows a communication arrangement according to an embodiment.

Figure 6 shows a processing flow according to an embodiment.

20

Figure 7 shows bandwidth-time diagram.

Figure 8 shows audio data according to an embodiment.

25

Figure 9 shows a first response message and a second response message.

Figure 10 shows a client station according to an embodiment.

30

Detailed description

According to various embodiments, a system is proposed that provides a "single source" based method for HTTP adaptive  
5 streaming of Fine Granular Scalable (FGS) audio such as MPEG-4 SLS over IP network. "Single source" can be understood as instead of requiring multiple files stored on a server for one media (e.g. audio) content (e.g. one piece of music) only one stored file is required. In addition, according to one  
10 embodiment, the server providing the media content is enabled to adjust the stream quality (e.g. audio stream quality) on the fly to avoid re-buffering that may typically be encountered in streaming applications such as online radio services and that may be annoying to the users.

15

According to one embodiment, a method for transmitting data is provided as illustrated in figure 1.

20

Figure 1 shows a flow diagram 100 according to an embodiment.

The flow diagram 100 illustrates a method for transmitting a digital signal.

25

In 101, data representing the digital signal is divided into a plurality of data blocks.

In 102, each data block is processed in accordance with a desired amount of data included in the data block.

30

In 103, for each processed data block, the size of the processed data block is determined.

In 104, a message is generated including, in a message body of the message, the processed data blocks and, for each data block, a message field specifying the size of the processed data block.

5

In 105, the message is transmitted.

In one embodiment, in other words, data representing a digital signal (e.g. an encoded version of the digital signal, such as an encoded bit stream representing the digital signal) is separated into blocks (e.g. a sequence of blocks corresponding to sequential parts of the digital signal such as sequential frames), the size of each block is, if necessary, adjusted in accordance with a desired amount of data included in the data block (e.g. corresponding to a desired quality level of the digital signal when being reconstructed from the data) and the processed data blocks are included as parts in an overall message body, wherein each processed data block has its own size indication. Each data block may correspond to a certain time period of the digital signal such that each amount of data corresponds to a data rate of the reconstructed digital signal. For example, each data block corresponds to one or more frames such that each amount of data corresponds to a certain amount of data per frame and thus to a certain data rate of the digital signal reconstructed from the transmitted data.

In one embodiment, the size of each data block is adjusted (or set) in accordance of a data rate adaptation of the data representing the digital signal. According to one embodiment, the size indication (e.g. length information) of a data block (also for example referred to as a chunk) is set only after the rate adaptation of the data (e.g. FGS encoded audio data)

within this chunk has been completed. According to one embodiment, this is used to enable carrying FGS encoded audio through HTTP chunk encoded data transmission.

The digital signal is for example a digital audio signal or a digital video signal. Generally, the digital signal may be a digital signal to be transmitted in real-time, i.e. a digital signal that has an associated playback speed and that is to be transmitted such that it can be reconstructed and played at a receiver at the associated playback speed (for example without rebuffering).

In one embodiment, for each data block, processing of the data block includes reducing the amount of the data included in the data block such that the data block includes the desired amount of data if the amount of data included in the data block is higher than the desired amount of data included in the data block.

According to one embodiment, the digital signal is encoded in accordance with a scalable coding method (such as MPEG-4 SLS) to generate the data representing the digital signal. It should be noted that the data to be transmitted (e.g. streamed), i.e. the data representing the digital signal may be a stored scalably encoded digital signal, e.g. a pre-stored scalably encoded digital signal representing a whole piece of music or a whole video clip (generally e.g. a whole media data file). In other words, for example, the data representing the digital signal is for example not data generated by a real-time encoder with encoding rate adapting on the fly based on bandwidth information but is for example pre-generated data representing the digital signal, e.g. data pre-generated before the receipt of the transmission of the digital signal (e.g. a request by a communication terminal



for transmission of the digital signal) or data pre-generated (e.g. for a complete piece of music or a complete media data file) before the beginning of the transmission process (e.g. before the first part of the data is transmitted).

5

In one embodiment, for each data block, processing of the data block includes reducing the amount of the data included in the data block such that the data block includes the desired amount of data in accordance with the scalability provided by the scalable coding method if the data block includes more data than the desired amount of data included in the data block.

Each data block for example includes an encoded bit stream according to the scalable coding method as the data representing the digital signal and wherein, for each data block, processing of the data block includes truncating the encoded bit stream to the desired amount of data included in the data block if the amount of data included in the data block is higher than the desired amount of data included in the data block.

The message is for example generated according to an application layer protocol. For example, the message is generated according to HTTP (Hypertext Transfer Protocol).

According to one embodiment, the message is generated according to chunked transfer encoding, wherein each processed data block corresponds to a chunk.

30

The message for example includes a message header. The message fields specifying the sizes of the data blocks are for example not included in the (overall) message header of

the message but, for each data block, the specification of the size of the data blocks is included in a message field associated with the data block in the message body, for example in a message field preceding the data block in the message body.

Each data block for example includes data representing one or more frames of the digital signal.

10 According to one embodiment, dividing the data representing the digital signal into a plurality of data blocks comprises dividing the data representing the digital signal into a sequence of data blocks.

15 According to one embodiment, dividing the data representing the digital signal into a plurality of data blocks comprises dividing the data representing the digital signal into a plurality of data blocks representing sequential parts of the digital signal.

20

The method may further comprise, for each data block, determining an available data transmission rate for the transmission of the data block and determining the desired amount of data included in the data block based on the available data rate.

25

The message is for example transmitted by a transmitter and determining the available data transmission rate for example comprises determining the transmission bandwidth of a communication channel between the transmitter and a receiver of the data blocks.

30

According to one embodiment, the data included in each data block represents one or more frames of the digital signal, the digital signal has an associated frame rate and, for each data block, the desired amount of data included in the data block is determined such that the processed data block can be transmitted using the determined available data transmission rate such that the frames are transmitted at the associated frame rate.

10 The flow illustrated in the flow diagram 100 is for example carried out by a transmitter (e.g. a server computer) as illustrated in figure 2.

Figure 2 shows a transmitter 200 for transmitting a digital signal according to an embodiment.

The transmitter includes a divider 201 configured to divide data representing the digital signal into a plurality of data blocks and a processor 202 configured to process each data block in accordance with a desired amount of data included in the data block.

The transmitter further comprises a determiner 203 configured to determine, for each processed data block, the size of the processed data block and a generator 204 configured to generate a message including, in a message body of the message, the processed data blocks and, for each data block, a message field specifying the size of the processed data block.

30 Further, the transmitter comprises a sender 205 configured to transmit the message.

The message is for example received in accordance with a receiving method as illustrated in figure 3.

Figure 3 shows a flow diagram 300 according to an embodiment.

5

The flow diagram 300 illustrates a method for receiving a digital signal.

10 In 301, a message is received, including, in a message body of the message, a plurality of data blocks including data representing the digital signal and, for each data block, a message field specifying the size of the data block taking the sizes of the data blocks into account.

15 In 302, the digital signal is reconstructed from the data included in the plurality of data blocks.

20 The flow illustrated in figure 3 is for example carried out by a receiver (e.g. a client station) as illustrated in figure 4.

Figure 4 shows a receiver 400 for receiving a digital signal according to an embodiment.

25 The receiver 400 includes a receiving module 401 configured to receive a message, including, in a message body of the message, a plurality of data blocks including data representing the digital signal and, for each data block, a message field specifying the size of the data block taking  
30 the sizes of the data blocks into account.

The receiver 400 further includes a processor 402 configured to reconstruct the digital signal from the data included in the plurality of data blocks.

5 It should be noted that according to various embodiments, computer program elements which, when executed by a computer (including e.g. a smartphone), make the computer perform the method for transmitting a digital signal and the method for receiving a digital signal as described above with reference  
10 to figures 1 and 3 are provided.

In the following, embodiments are described in more detail.

Various embodiments provide a practical system solution to  
15 Network Adaptive Audio Streaming (NAAS). It may include a TCP-based bandwidth estimator, a dynamic NAAS linker to a HTTP web server which may be seen as a standard HTTP server, an FGS audio data block processor, and a customized streaming client. Such an architecture is illustrated in figure 5.

20

Figure 5 shows a communication arrangement 500 according to an embodiment.

The communication arrangement 500 comprises a server station  
25 501 (e.g. a server computer) and a client station 502 (e.g. a mobile phone such as a smartphone).

The server station 501 and the client station 502 are  
connected by a communication network 503, e.g. a wired or  
30 wireless IP (Internet Protocol) network.

The server station 501 comprises a bandwidth estimator 504, a media source 505, in this example a source of FGS (Fine

Granular Scalable) audio data, i.e. scalably encoded audio data, a linking component 506, in this example a dynamic NAAS linker, a media data processor 507, in this example an FGS audio data block processor, and an HTTP web server 508.

5

The server station 501 can be seen to implement an adaptive streaming system.

According to one embodiment, the adaptive streaming system works with a standard HTTP web server 508 without affecting any other function of the HTTP web server 508. In one embodiment, this is achieved by providing the bandwidth estimator 504 and the FGS audio data block processor 507, e.g. implemented by means of two software modules added to the HTTP web server software.

15

The bandwidth estimator 504 estimates in real-time the available streaming bandwidth between the server station 501 and the client station 502 and the FGS audio data block processor 507 truncates FGS audio data provided by the media source 505 according to the estimated available streaming bandwidth to ensure that the data rate of the audio data streamed from the server station 501 to the client station 502 is close (e.g. as close as possible, i.e. optimally close) to the available streaming bandwidth. For example, software modules with which the bandwidth estimator 504 and the FGS audio data block processor 507 are implemented are dynamically linked to the HTTP web server 508 via software hooking (i.e. a specific software interfacing technique).

25  
30

It should be noted that in a conventional HTTP based streaming system, the media data being streamed may be typically transmitted in the form of HTTP messages for which

the length of the message body is signaled by means of a fixed, predetermined data element before the message body. According to one embodiment, as the length of FGS audio data is dynamic due to the truncation operation, a solution is provided that allows to effectively transmit FGS audio data with variable length information by means of HTTP messages. Specifically, according to one embodiment FGS audio data (e.g. an audio signal corresponding to a piece of music) are partitioned into a series (or sequence) of data blocks and each data block is transmitted in a separate chunk of a HTTP message according to HTTP chunked transfer encoding.

According to one embodiment, the length information of a chunk is set only after the rate adaption of the FGS audio data within this chunk message so that the chunk contains the correct length information.

The HTTP web server 508 may be a standard HTTP web server (e.g. implemented as an Apache Server) that does not provide adaptive streaming features (by itself). According to one embodiment, the functionality of adaptive streaming is added to the HTTP web server 508 by modifying and including new functions directly to program code of the HTTP server 508 (if available) and rebuilding the server software. According to another embodiment, a more practical way may be used: In an Apache server, for example, at certain stages of the process, customized software modules can be "hooked" to the server at run time in order to perform certain customized functions. These customized modules can be developed and built independently into a DLL (dynamic link library) like binary file and can be loaded by the server software at run time. In this way, the server capabilities can be extended by the functionality of adaptive streaming without touching any part

of the HTTP server 508 and thus system deployment can be significantly simplified.

The dynamic linking of a customized module for providing  
5 adaptive streaming (also referred to as the customized NAAS  
module in the following) is illustrated in figure 6.

Figure 6 shows a processing flow 600 according to an  
embodiment.

10

The processing flow 600 is carried out by a server 601, e.g.  
corresponding to the HTTP web server 508, a customized NAAS  
module 603 and linking points 602 between the server 601 and  
the (customized) NAAS module 603.

15

In 604, a worker thread is started.

In 605, a process connection is carried out, e.g. the server  
601 connects to the communication network 503 to be able to  
20 receive requests for media data (e.g. audio files).

In 606, a read request is received, e.g. a request from the  
client station 502 for media data.

25 In 607, the request is processed. This may involve, in 608, a  
processing of the URL (Uniform Resource Locator) specified in  
the request and a processing of one or more headers of the  
request in 609. Further, in 610, the type of the requested  
data is checked. For this, a list of linker functions  
30 registered as type checker 611 may be checked at linking  
points 602. For a specific data type (e.g. MIME, Multipurpose  
Internet Mail Extensions, type) that is supported (i.e. for  
which a linker function is registered as handler) the linker



function 612 to handle the specific data type is provided by the customized NAAS module 603.

5 In 613, after type checking, the handler for the requested data is invoked. For this, the linking points 602 provide a list of linker functions registered as handler 614. The customized NAAS module 603 provides the linker functions 615 that are registered as handler for the data.

10 In 616, the server 601 disconnects from the communication network 503 and the worker thread is stopped in 617.

For the dynamical linking to the server at the different stages in the process of handling a (HTTP) request for FGS audio data, firstly, a linker function is registered as type  
15 checker to add a specific MIME type "application/x-sls-audio" for MPEG-4 SLS bit-stream to the request record structure, i.e. is added to the list of linker functions registered as type checker 611. Secondly, another linker function is  
20 registered as handler to handle client requests for data with MIME type "application/x-sls-audio", i.e. is added to the list of linker functions registered as handler 614.

When the server receives a client request for FGS audio in  
25 606, it runs through the illustrated process steps described above, wherein the linker function registered as type checker to add a specific MIME type "application/x-sls-audio" is called when the server runs to the linking point where, in  
610, all the linker functions registered as type checker are  
30 examined. Finally, when the server runs to the handler linking point, in 613, the linker function registered as handler to handle client requests for data with MIME type "application/x-sls-audio" competes with the other registered

handlers to take over the tasks in handling the request for FGS audio.

5 After the customized NAAS module 603 has been dynamically linked to HTTP server 601, it extends the capability of the server 601 to make it an adaptive streaming server for FGS audio while all the other standard functions in the server can be left intact. For example, the server 601 can still provide web pages to the client station 502 using a web  
10 browser.

According to one embodiment, the bandwidth estimator 504 provides a TCP-based network bandwidth estimation to the media data processor. Network bandwidth estimation may for  
15 example be used for routing algorithms and congestion control mechanisms in traffic engineering. Techniques and tools for network bandwidth estimation typically use active probing to measure bandwidth related metrics. Further, for example, the idle rate of a wireless link may be calculated to estimate an  
20 available bandwidth. This, however, requires adding a module to the MAC (Medium Access Control) layer of each node in the network in order to get the idle rate.

According to one embodiment, instead of accessing the MAC  
25 layer, a more practical way is to estimate the available bandwidth at transport layer. A UDP (User Datagram Protocol) based video transport protocol (VTP) uses the timestamp information contained in the specially designed control packet to calculate the available bandwidth. In various  
30 embodiments, the acknowledgement mechanism in TCP is used to get the required information to estimate the available bandwidth instead of proposing a new transport protocol,

which is more practical in system implementation and deployment.

The sequence number in a TCP response is the number of  
 5 received bytes acknowledged by the receiver. Let  $s_i$  be the  
 sequence number acknowledged at time  $t_i$ ,  $s_{i-1}$  be the sequence  
 number acknowledged at time  $t_{i-1}$ , then the available bandwidth  
 $b_i$  at time  $t_i$  can be estimated by

$$10 \quad b_i = \frac{s_i - s_{i-1}}{t_i - t_{i-1}} \quad (1)$$

According to one embodiment, to reduce the noise in estimated  
 bandwidth and avoid rapid fluctuation in stream quality, a  
 low pass filter is applied by smoothing the estimated  
 15 bandwidth:

$$\overline{b}_i = \alpha_i \overline{b}_{i-1} + (1 - \alpha_i) b_i \quad (2)$$

where  $\alpha_i$  is the weighting coefficient between 0 and 1, which  
 20 is dependent on  $\Delta_i = t_i - t_{i-1}$ .

According to one embodiment, the bandwidth estimation  
 algorithm for NAAS is carried out in accordance with  
 equations (1) and (2). The behavior of the bandwidth  
 25 estimation algorithm is illustrated in figure 7.

Figure 7 shows bandwidth-time diagram 700.

Time is given (in seconds) along a time axis 701 and  
 30 bandwidth is given (in kbps) along a bandwidth axis 702.

The actual bandwidth is in this example given as a dashed line 704 and the bandwidth estimated by the bandwidth estimation algorithm is given as a solid line 703.

5 As can be seen, figure 7 illustrates the step response of the bandwidth estimation algorithm to the change of the actual bandwidth from 64kbps to 256kbps at 24 seconds and from 256kbps back to 64kbps at 64 seconds during a streaming process.

10

It should be noted that the above bandwidth estimation algorithm is described here for illustration purpose and other bandwidth estimation algorithms may be used according to various embodiments.

15

According to one embodiment, the FGS audio is encoded according to MPEG-4 scalable lossless (SLS) coding. MPEG-4 scalable lossless (SLS) coding was one of the latest additions to the MPEG-4 audio coding tool family from  
20 ISO/IEC. It allows the scaling up of a perceptually coded representation such as MPEG-4 AAC to a lossless representation with a wide range of intermediate bit-rate representations. It also has a non-core mode in which the MPEG-4 AAC core is not present, and the quality is scaled up  
25 virtually from 0 kbps.

One of the major merits of MPEG-4 SLS can be seen in that the bit-stream generated from the encoder can be further truncated to lower data rates easily by dropping bits at the  
30 end of each frame. This is illustrated in figure 8.

Figure 8 shows audio data according to an embodiment.

In a first format 801, the audio data includes a losslessly encoded audio signal (or more generally the audio signal with highest quality) and is for example stored by the audio source 505. According to the first format 801, the audio data  
5 includes audio data for each frame 802 of a plurality of frames (N frames in this example). The audio data has the form of an MPEG4-SLS audio stream, such that the audio data for each frame 802 are arranged in a sequence in the stream.

10 The audio data for each frame 802 include a first header 803 for a first channel, first data 804 for a first channel, a second header 805 for a second channel and second data 806 for a second channel.

15 The audio data for each frame 802 also form a sequence, e.g. a bit stream, such that the whole audio data form an overall bit stream.

For the audio data for each frame 802, the first data 804 and  
20 the second data 806 also form a bit stream and may be truncated at the end such that the data for the frame 802 may be reduced. This truncation process is illustrated by an arrow 807 and is for example carried out by the data processor 507.

25 The result of the truncation is a second format 808 in which, as illustrated, the first data 804 and the second data 806 for the two channels for each frame 802 are reduced which leads to a quality of the encoded audio signal that is lower  
30 than the original quality.

In other words, an encoded audio signal with lower data rate can be generated from the original encoded audio signal (with

highest quality) by dropping bits at the end of each channel data bit stream.

Here, the term data rate is used as well as media rate to  
5 denote the number of bits or bytes per audio frame duration  
being, for example, provided by the server station 501,  
transmitted and eventually processed by the decoder of the  
client station 502. The higher the audio stream quality, the  
higher the media rate (data rate). According to one  
10 embodiment, HTTP adaptive streaming includes determining the  
media rate according to the estimated available network  
bandwidth so that media rate is always equal or less than the  
network bandwidth (available for the transmission of the  
encoded audio signal) in order to make sure the smooth  
15 playback of the audio stream at the client side.

For example, according to one embodiment, once the linker  
function registered as handler to handle client requests for  
data with MIME type "application/x-sls-audio" (also referred  
20 to as the NAAS handler) has captured the client request for  
FGS audio, it starts a separate thread to estimate the  
available bandwidth of the link using, e.g. using the  
bandwidth estimation method as described above with reference  
to equations (1) and (2). Meanwhile, it composes response  
25 message headers and the response message body. The response  
message body contains the requested FGS audio data that are  
read from the source audio data file provided by the media  
(in this example audio) source 505.

30 In non-adaptive (fixed bit rate) cases such as in a non-  
adaptive streaming applications, the length of the message  
body can be calculated in advance and included in the  
"Content-Length" response message header. Typically, a client

station needs to be signaled with this value before it starts to receive the following-up message body. Otherwise, either a premature termination of the HTTP connection or a time-out may occur where in both cases a client station will not be  
5 able to get the response message body correctly.

However, since the NAAS handler may truncate the FGS audio data of at least some of the frames in the audio data stream to be included in the response message due to an available  
10 network bandwidth that is insufficient for the audio data with highest encoding quality, it may not be possible to determine the amount of data in the response message in advance.

15 Accordingly, in one embodiment, the NAAS module 603, in order to overcome the requirement of signaling a fixed and pre-determined Content-Length message header according to HTTP protocol, uses the Chunked Transfer-Encoding mechanism according to HTTP/1.1 to support the adaptive streaming  
20 functionality of NAAS.

For this, according to one embodiment, the whole message body, which contains the compressed scalable encoded audio data (with highest quality, i.e. not yet truncated), is split  
25 into a number of smaller data blocks (chunks), each block containing the data of an integer number of FGS audio frames. The adjustment of media rate of the audio stream (i.e. the truncation) is performed for each data block before it is transmitted. After that, the size of the data block is re-  
30 calculated and inserted in the beginning of the data block. This inserted data block size is signaled with the HTTP/1.1 Transfer-Encoding: Chunked message header and hence does not

interfere the normal progressive downloading function of a HTTP/1.1 compliant client.

In this way, according to one embodiment, all the data blocks  
5 are sent to the client station 502 one by one independently and progressively without the need to inform the client about the size of the whole message in advance. This is illustrated in figure 9.

10 Figure 9 shows a first response message 901 and a second response message 902.

The first response message 901 can be seen to correspond to the case that the size of the whole message is known before  
15 start of the transmission, or in other words, to a fixed length of the audio stream. Accordingly, the size of the message (in this example 65536 byte) can be inserted into a header 903 of the first response message 901. A message body 904 of the first response message includes the audio data.

20

The second response message 902 can be seen to correspond to adaptive streaming. The original audio stream (i.e. the audio stream corresponding to highest quality) is split into data blocks, each data block is processed by truncating the audio  
25 data included in the data block depending on the network bandwidth currently available for the transmission of the data block and an indication of the amount of data 908 of the processed data block is included in a data block header 907 of the data block.

30

A header 905 of the second message 902 includes the indication that the message has been generated according to



HTTP chunked transfer encoding and a body 906 of the second message includes the data blocks.

The processed blocks are transmitted progressively, i.e. sequentially, to the client station 502 by means of TCP (Transport Control Protocol) PDUs (Packet Data Units). According to one embodiment, the media rate is adjusted by the FGS audio data block processor 507 according to the estimated available network bandwidth  $\overline{b}_i$  at time  $t_i$  so as to keep smooth playing of the audio stream at the client side.

The media rate can for example be adjusted based on the following calculations:

1) Determine the average frame size  $\overline{d}_i$  to be sent out from  $t_i$  to  $t_{i+1}$ .

$$\overline{d}_i = \overline{b}_i \times 1024 / f_0$$

where  $f_0$  is the sampling frequency (i.e. the number of frames per second of the encoded audio signal) and  $\overline{b}_i$  is the smoothed available network bandwidth.

2) Determine the truncation rate  $\lambda_{ij}$  for each frame between  $t_i$  and  $t_{i+1}$ .

Assume there are  $J$  frames between  $t_i$  and  $t_{i+1}$  and let  $d_{ij}$  be the frame size of the  $j^{\text{th}}$  frame ( $j \in \mathcal{J}$ ) then set

$$\lambda_{ij} = \eta \times \overline{d}_i / d_{ij}$$

where  $\eta$  is a constant coefficient between 0.9 and 1 to make sure the media rate being sent out does not exceed the available bandwidth so as to keep the playback on the client

side smooth. An upper bound and a lower bound may be applied to the  $\lambda_{ij}$  such that it is ensured that

$$\lambda_{min} \leq \lambda_{ij} \leq 1.$$

5

3) Adjust the media rate

For the  $j^{\text{th}}$  frame ( $j \in J$ ) between  $t_i$  and  $t_{i+1}$ ,

$$d_{ij} = h + d_{ij0} + d_{ij1}$$

10

where  $h$  is a constant representing the header size,  $d_{ij0}$  is the data size for channel 0, and  $d_{ij1}$  is the data size for channel 1. The new channel 0 data size and the new channel 1 data size of the frame (after truncation) may be calculated as

15

$$\overline{d_{ij0}} = \lambda_{ij} \times d_{ij0}, \quad \overline{d_{ij1}} = \lambda_{ij} \times d_{ij1}.$$

Finally the adjusted frame size based on the available bandwidth  $\overline{b_i}$  may be calculated as

20

$$\overline{d_{ij}} = h + \overline{d_{ij0}} + \overline{d_{ij1}}.$$

25

It should be noted that the above media rate adjustment method is included here for illustration purpose and other media rate adjustment algorithms may be used in the adaptive streaming system according to various embodiments.

The client station 502 may for example have a structure as illustrated in figure 10.

30

Figure 10 shows a client station 1000 according to an embodiment.

According to one embodiment, the client station 1000 acts as a media player for the adaptive streaming system described above with reference to figure 5 and corresponds to the client station 502. According to one embodiment, the client station 1000 may be seen as a typical HTTP based streaming client.

According to one embodiment, for operating as the client for the adaptive streaming system, the client station 1000 carries out three processes: a receiving process, a decoding process and an audio output process (referred to as threads 1 to 3 in figure 10).

For this, the client station 1000 includes an HTTP client 1001 which may for example start the streaming process by sending an HTTP request for FGS audio data.

Two FIFO (First In First Out) memories are allocated as stream buffer and audio buffer respectively.

As illustrated in figure 9, in the header 905 of the response message 902 it is signaled by the NAAS HTTP streaming server 901 that the incoming message body is Chunked Transfer Encoded.

The HTTP client 1001 retrieves the data block size inserted into the data block header 907 (at the beginning of each data block) and use it to read the data correctly from the data blocks block by block until an EOF (End of File) syntax is received.

A stream receiver 1002 retrieves the FGS audio frames contained in the data blocks and pushes them one by one into a stream buffer 1003. An FGS Audio Decoder 1004 process fetches the FGS audio frames from the stream buffer 1003,  
5 decodes the audio data of each frame and pushes the decoded audio data, in this case PCM (Pulse Code Modulation) audio samples, into an audio buffer 1005. An audio output 1006 plays the decoded FGS audio by reading the PCM audio samples from the audio buffer 1005 and for example passes them to a  
10 sound output device.

The adaptive audio (or generally media such as video) streaming system according to various embodiments as described above may be implemented using different  
15 communication network environments such as in a local area WiFi network with dedicated wireless router, in a shared office or building WiFi network, or in a 3G wireless network operated by local service provider. In an office WiFi network, for example, when bandwidth is high, the server  
20 station 501 responds to the available high TCP throughput by adjusting the bit stream to the highest quality ( $\lambda=1$ ). When the bandwidth is reduced to for example 64kbps (as illustrated in figure7), the server station 501 may reduce the sending bit rate by lowering the stream quality  
25 accordingly. After the bandwidth has recovered, the server station 501 can adjust the bit stream to highest quality again. As another example, in the case of a 3G wireless network, the bandwidth may for example keep fluctuating around 160kbps and the server station 501 keeps adjusting the  
30 bit-rate of the streamed MPEG-4 SLS bit-stream to fit into the available bandwidth. In both cases, the playback on the client station 502 is smooth and the user does not encounter re-buffering.

Claims

1. A method for transmitting a digital signal comprising:  
dividing data representing the digital signal into a  
5 plurality of data blocks;  
processing each data block in accordance with a desired  
amount of data included in the data block;  
determining, for each processed data block, the size of  
the processed data block;  
10 generating a message including, in a message body of the  
message, the processed data blocks and, for each data  
block, a message field specifying the size of the  
processed data block; and  
transmitting the message.
- 15
2. The method according to claim 1, wherein the digital  
signal is a digital audio signal or a digital video  
signal.
- 20
3. The method according to claim 1 or 2, wherein, for each  
data block, processing of the data block includes  
reducing the amount of the data included in the data  
block such that the data block includes the desired  
amount of data if the amount of data included in the  
25 data block is higher than the desired amount of data  
included in the data block.
- 30
4. The method according to any one of claims 1 to 3,  
wherein the digital signal is encoded in accordance with  
a scalable coding method to generate the data  
representing the digital signal.

5. The method according to claim 4, wherein, for each data block, processing of the data block includes reducing the amount of the data included in the data block such that the data block includes the desired amount of data in accordance with the scalability provided by the scalable coding method if the amount of data included in the data block is higher than the desired amount of data included in the data block.
6. The method according to claim 4, wherein each data block includes an encoded bit stream according to the scalable coding method as the data representing the digital signal and wherein, for each data block, processing of the data block includes truncating the encoded bit stream to the desired amount of data included in the data block if the amount of data included in the data block is higher than the desired amount of data included in the data block.
7. The method according to any one of claims 1 to 6, wherein the message is generated according to an application layer protocol.
8. The method according to any one of claims 1 to 7, wherein the message is generated according to HTTP.
9. The method according to any one of claims 1 to 8, wherein the message is generated according to chunked transfer encoding, wherein each processed data block corresponds to a chunk.
10. The method according to any one of claims 1 to 9, wherein the message includes a message header.

11. The method according to any one of claims 1 to 10,  
wherein each data block includes data representing one  
or more frames of the digital signal.
- 5
12. The method according to any one of claims 1 to 11,  
wherein dividing the data representing the digital  
signal into a plurality of data blocks comprises  
dividing the data representing the digital signal into a  
sequence of data blocks.
- 10
13. The method according to claims 1 to 12, wherein dividing  
the data representing the digital signal into a  
plurality of data blocks comprises dividing the data  
representing the digital signal into a plurality of data  
blocks representing sequential parts of the digital  
signal.
- 15
14. The method according to any one of claims 1 to 13,  
further comprising, for each data block, determining an  
available data transmission rate for the transmission of  
the data block and determining the desired amount of  
data included in the data block based on the available  
data rate.
- 20
- 25
15. The method according to claim 14, wherein the message is  
transmitted by a transmitter and wherein determining the  
available data transmission rate comprises determining  
the transmission bandwidth of a communication channel  
between the transmitter and a receiver of the data  
blocks.
- 30

16. The method according to claim 14 or 15, wherein the data included in each data block represents one or more frames of the digital signal, the digital signal has an associated frame rate and, for each data block, the  
5 desired amount of data included in the data block is determined such that the processed data block can be transmitted using the determined available data transmission rate such that the frames are transmitted at the associated frame rate.

10

17. A transmitter for transmitting a digital signal comprising:  
a divider configured to divide data representing the digital signal into a plurality of data blocks;  
15 a processor configured to process each data block in accordance with a desired amount of data included in the data block;  
a determiner configured to determine, for each processed data block, the size of the processed data block;  
20 a generator configured to generate a message including, in a message body of the message, the processed data blocks and, for each data block, a message field specifying the size of the processed data block; and  
a sender configured to transmit the message.

25

18. A method for receiving a digital signal comprising:  
receiving a message, including, in a message body of the message, a plurality of data blocks including data  
representing the digital signal and, for each data  
30 block, a message field specifying the size of the data block taking the sizes of the data blocks into account;  
and



reconstructing the digital signal from the data included in the plurality of data blocks.

19. A receiver for receiving a digital signal comprising:
- 5 a receiving module configured to receive a message, including, in a message body of the message, a plurality of data blocks including data representing the digital signal and, for each data block, a message field specifying the size of the data block taking the sizes
- 10 of the data blocks into account; and
- a processor configured to reconstruct the digital signal from the data included in the plurality of data blocks.

FIG 1

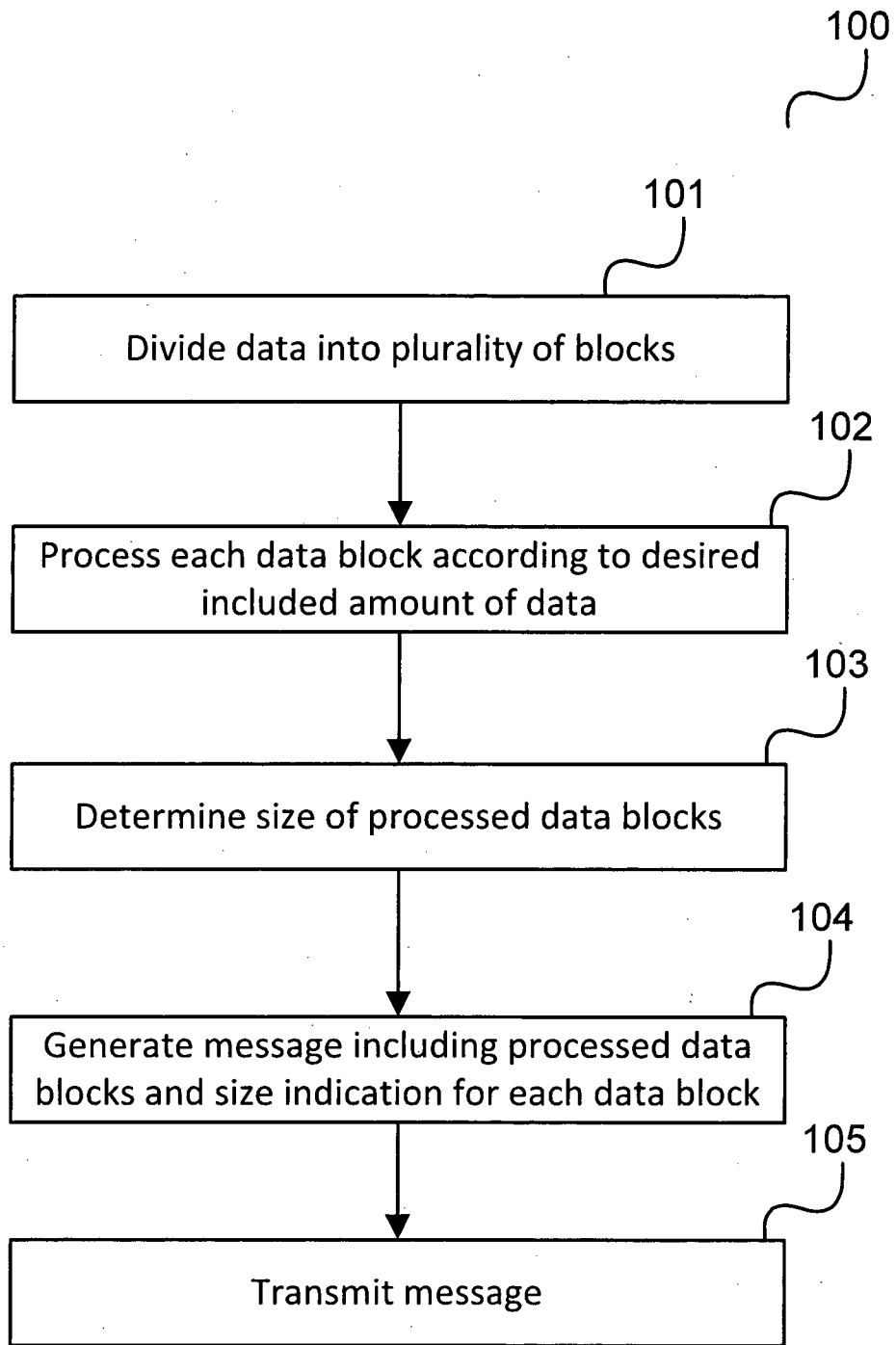


FIG 2

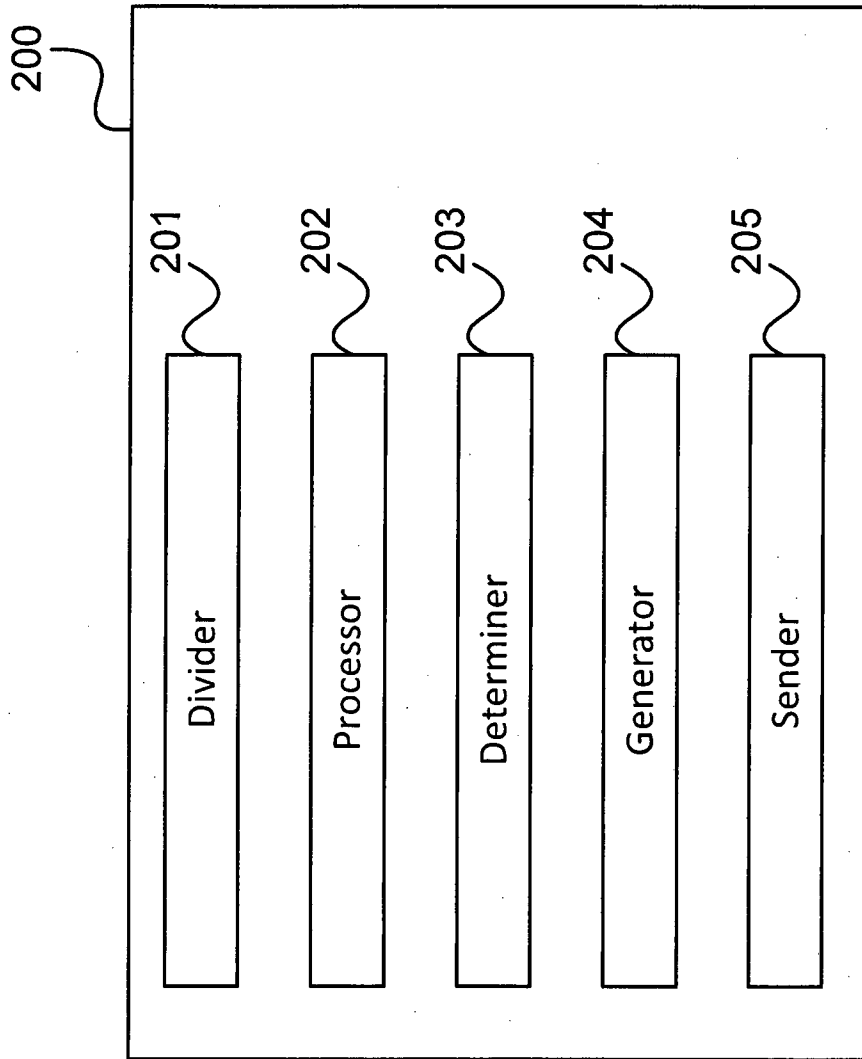


FIG 3

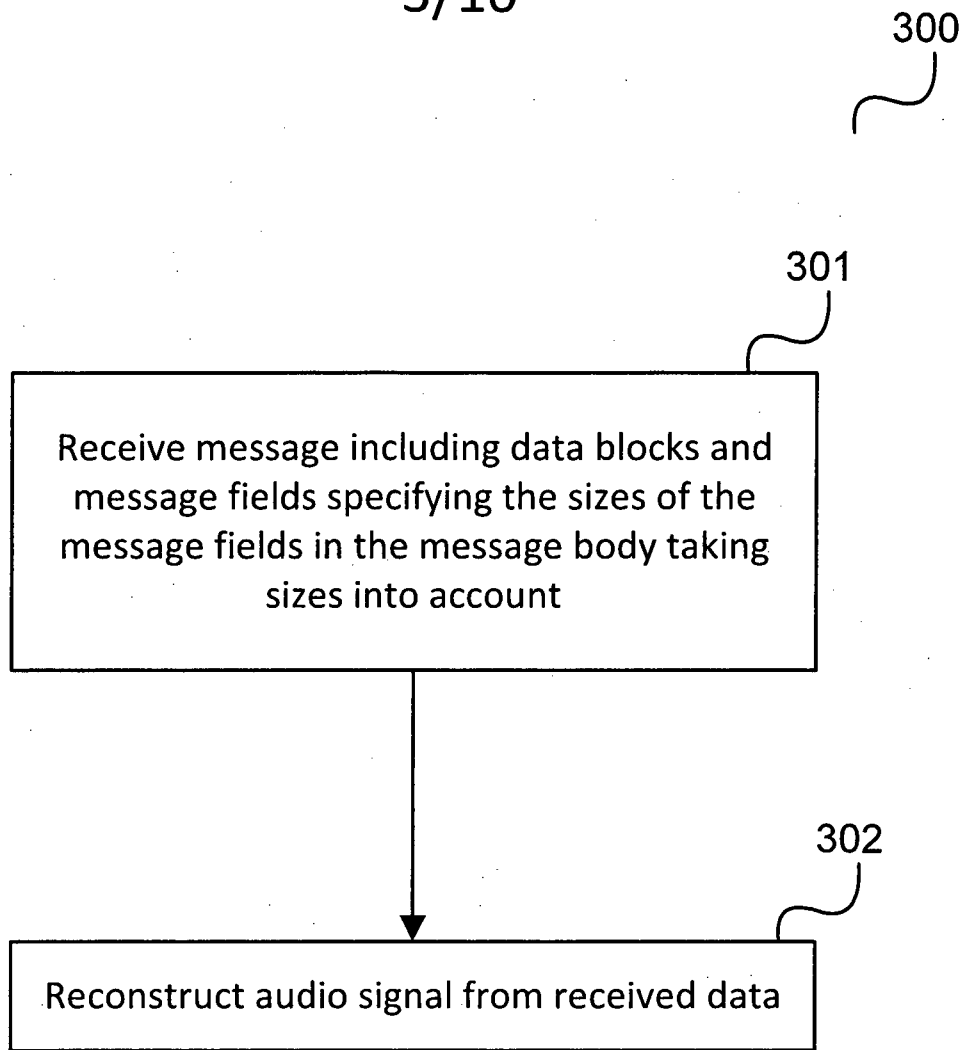
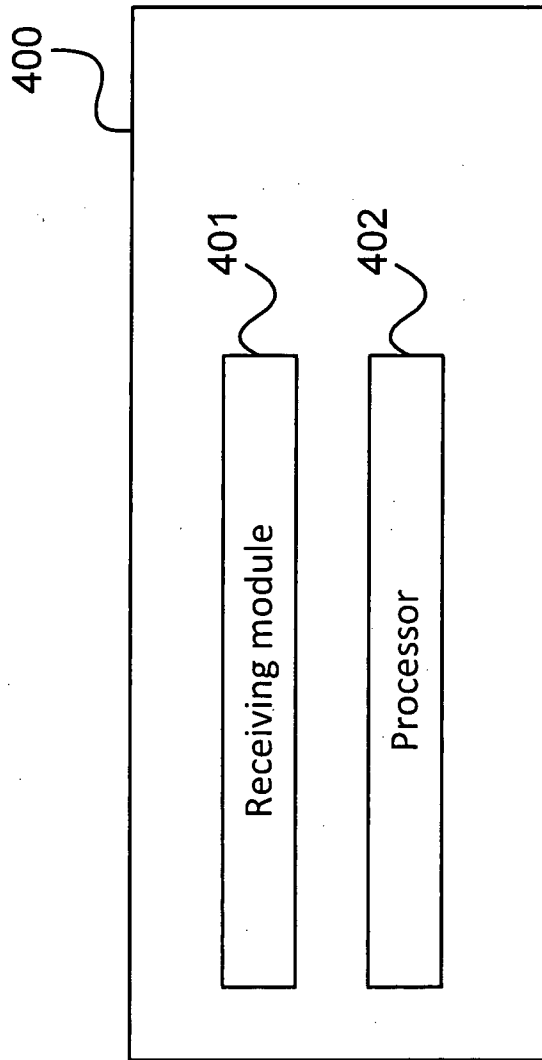


FIG 4



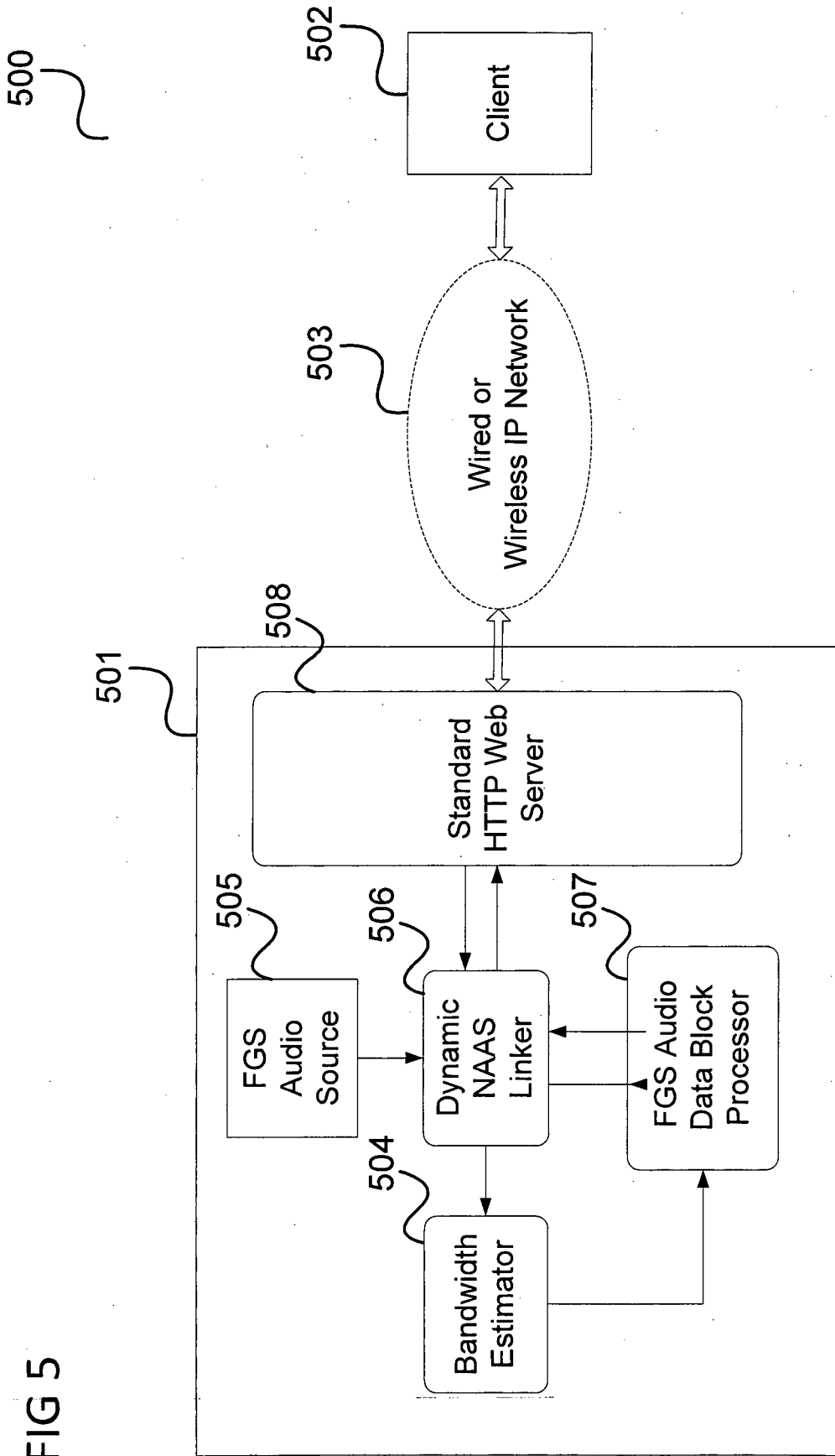
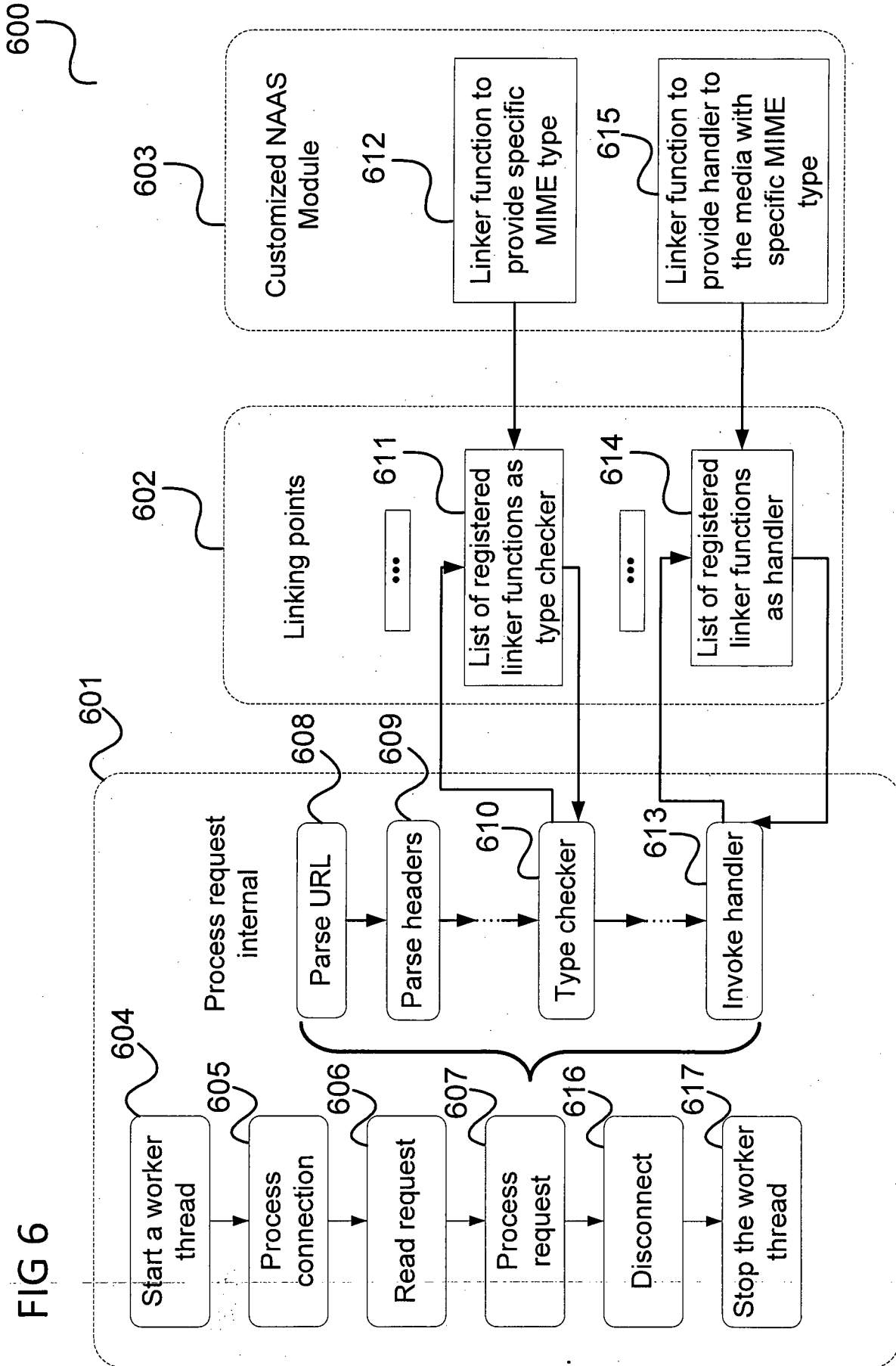


FIG 5



700

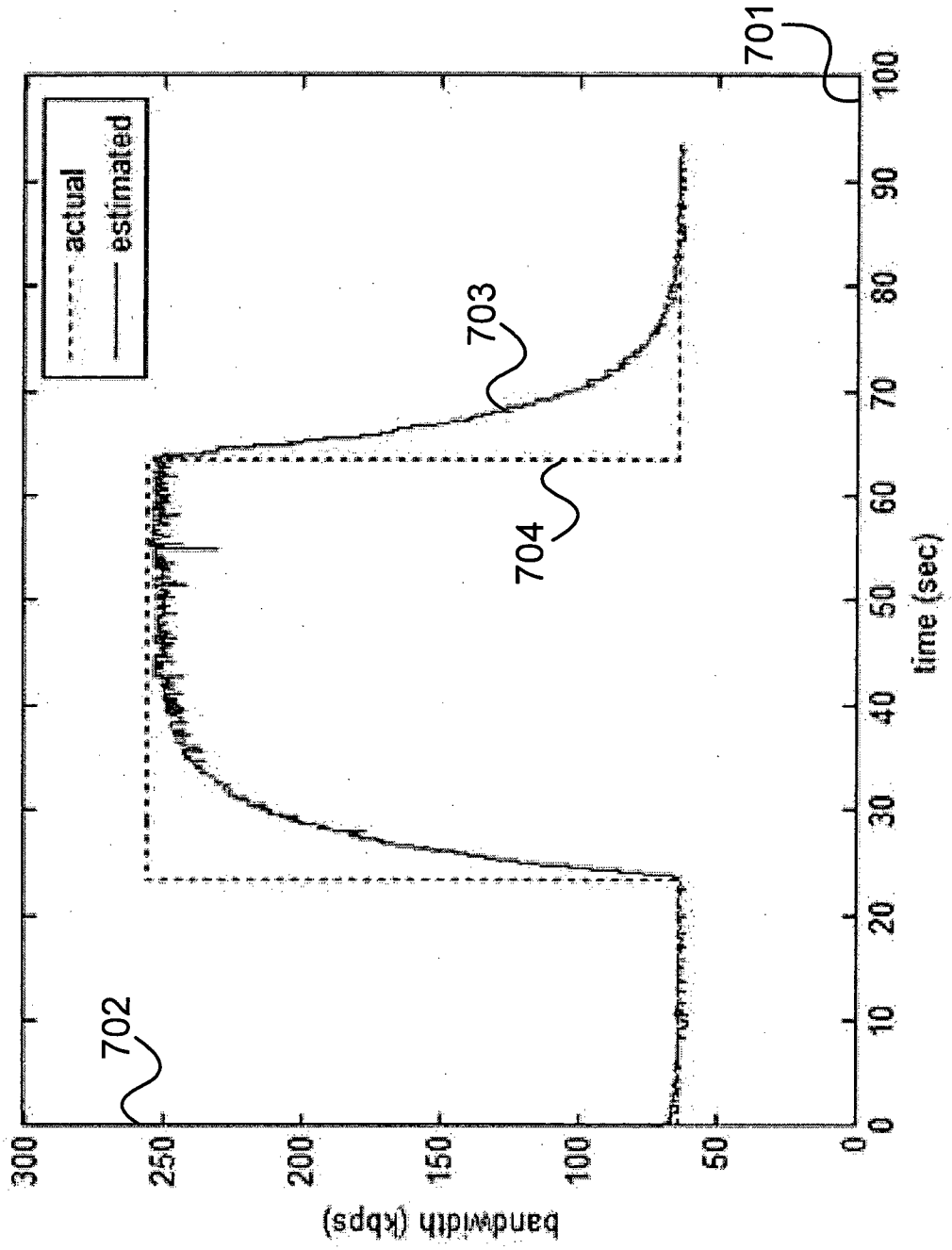


FIG 7



FIG 8

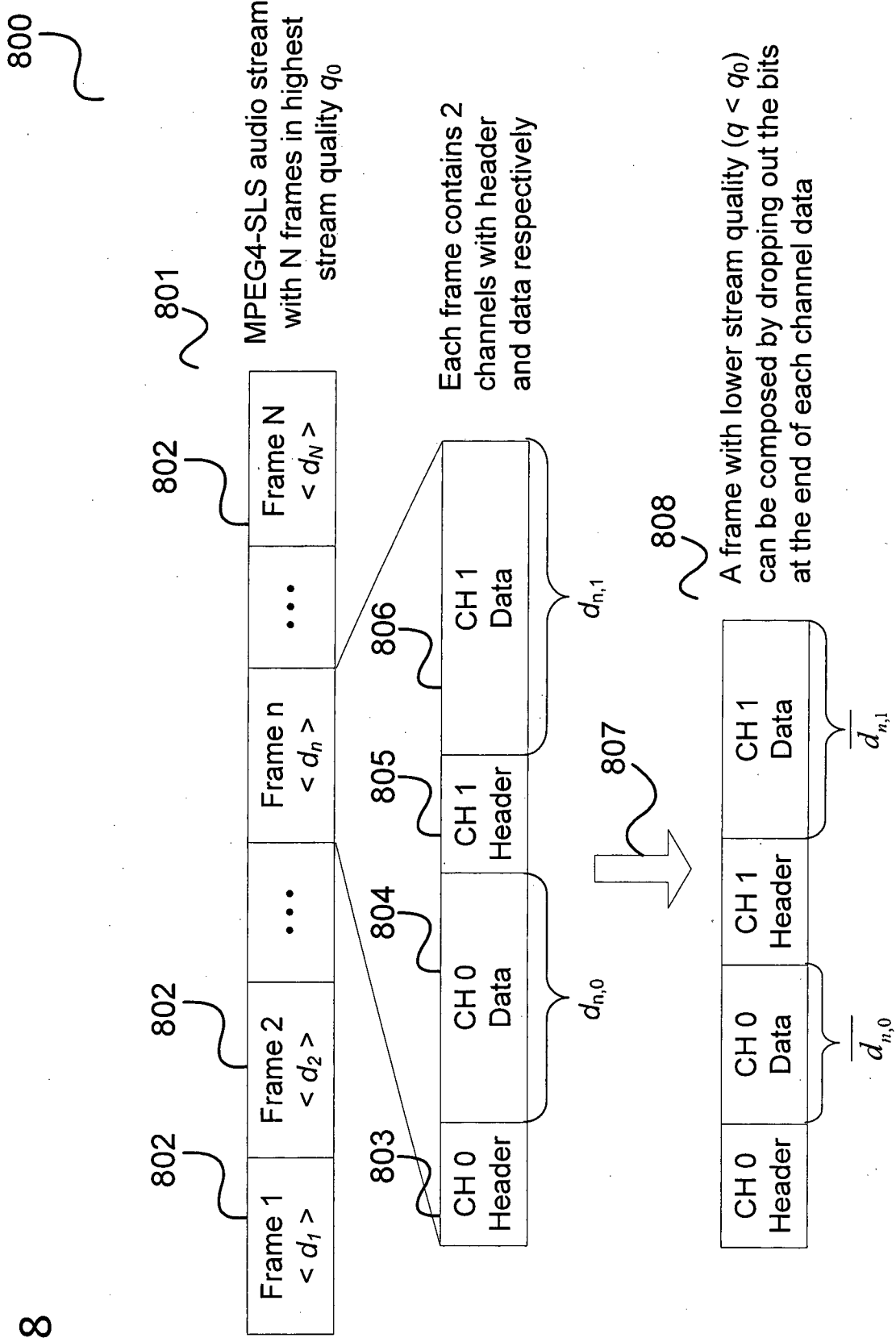


FIG 9

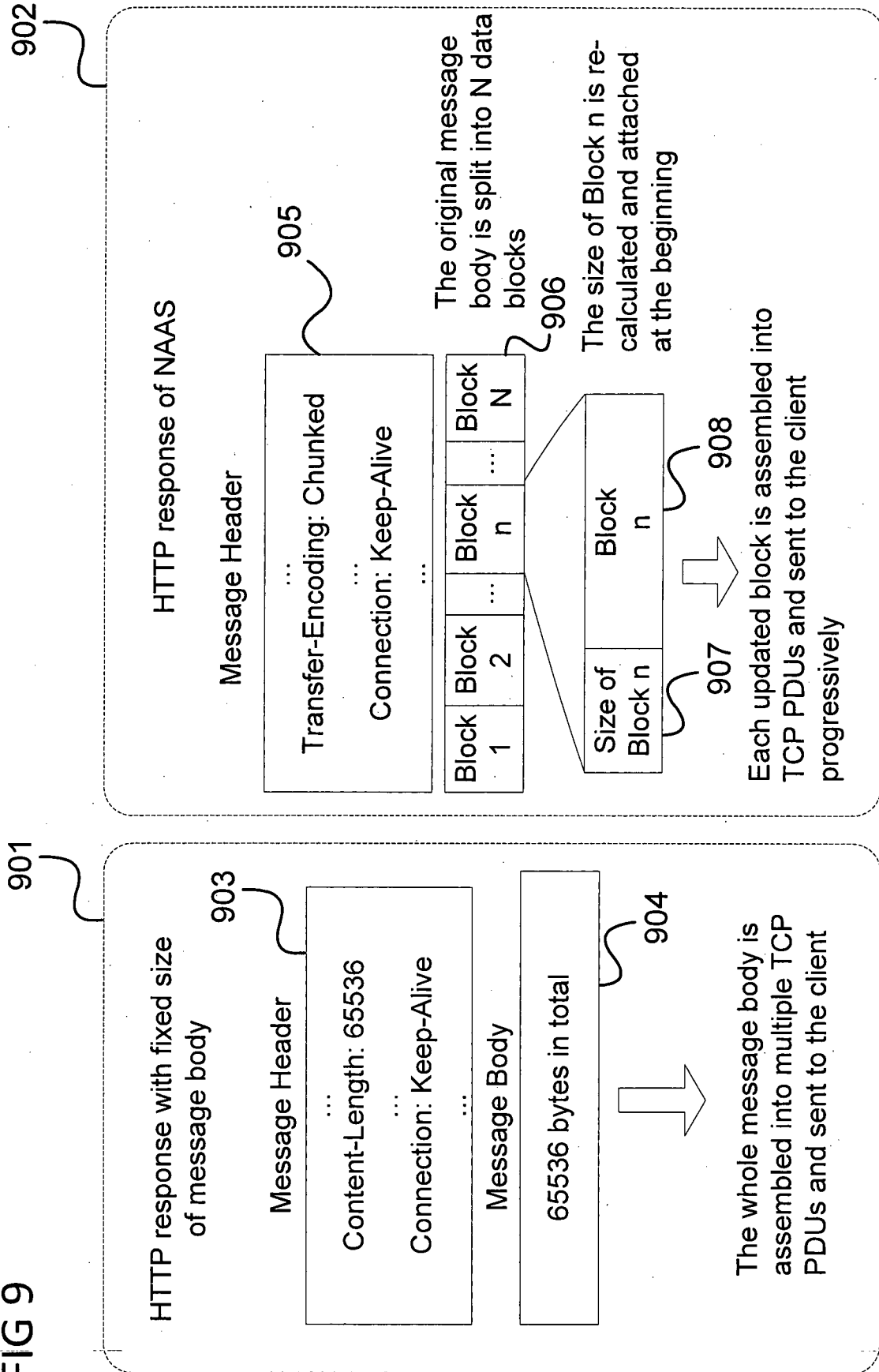
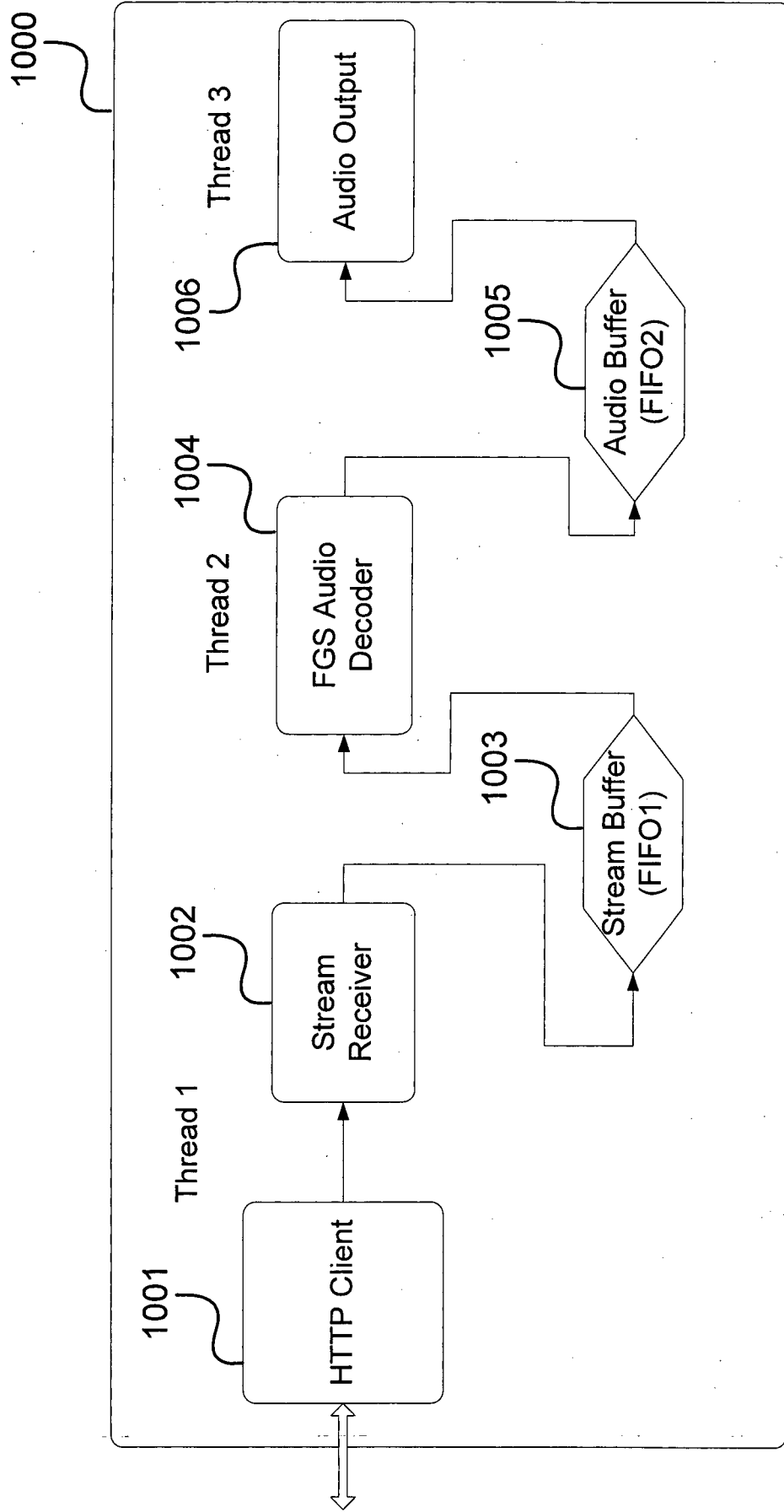


FIG 10



INTERNATIONAL SEARCH REPORT

International application No  
PCT/SG2012/000038

A. CLASSIFICATION OF SUBJECT MATTER  
 INV. H04N21/24 H04N21/2662 H04L29/06  
 ADD. H04N21/233 H04N21/643

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
 H04N H04L

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
 EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages  | Relevant to claim No. |
|-----------|---|-----------------------|
| X         | CHEN J ET AL: "An innovative adaptive streaming system for MPEG-4 scalable to lossless audio",<br>PROCEEDINGS OF THE EIGHTH IASTED INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING, PATTERN RECOGNITION AND APPLICATIONS [AND] PROCEEDINGS OF THE TWELFTH IASTED INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND IMAGING : FEBRUARY 16 - 18, 2011, INNSBRUCK, 16 February 2011 (2011-02-16), pages 294-299, XP008152417, DOI: 10.2316/P.2011.721-011<br>abstract<br>section 1<br>section 3.3<br>section 4.2<br>figures 1,3, 6, 7<br>-----<br>-/-- | 1-19                  |

Further documents are listed in the continuation of Box C.

See patent family annex.

\* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

|  |  |
|--|--|
| Date of the actual completion of the international search<br><br>4 June 2012 | Date of mailing of the international search report<br><br>12/06/2012 |
|--|--|

|  |   |
|--|---|
| Name and mailing address of the ISA/<br>European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Authorized officer<br><br>Arpaci, Mutlu |
|--|---|

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/SG2012/000038

| C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT |  |                       |
|--|--|-----------------------|
| Category*  | Citation of document, with indication, where appropriate, of the relevant passages   | Relevant to claim No. |
| A  | FECHEYR-LIPPENS A: "A Review of HTTP Live Streaming",<br>INTERNET CITATION,<br>25 January 2010 (2010-01-25), pages 1-37,<br>XP002638990,<br>Retrieved from the Internet:<br>URL:http://issuu.com/andruby/docs/http_live_streaming<br>[retrieved on 2011-05-24]<br>abstract   | 8                     |
| A  | JIANPING ZHOU ET AL: "Scalable audio streaming over the internet with network-aware rate-distortion optimization",<br>MULTIMEDIA AND EXPO, 2001. ICME 2001. IEEE INTERNATIONAL CONFERENCE ON, ADVANCED DISTRIBUTED LEARNING,<br>22 August 2001 (2001-08-22), pages 567-570, XP032177045,<br>DOI: 10.1109/ICME.2001.1237783<br>ISBN: 978-0-7695-1198-6<br>abstract  | 1-19                  |
| A  | KRASIC C., WALPOLE, J., FENG, W.:<br>"Quality-Adaptive Media Streaming by Priority Drop",<br>NOSSDAV'03, ACM, 2 PENN PLAZA, SUITE 701 - NEW YORK USA,<br>1 June 2003 (2003-06-01), pages 112-121,<br>XP040150147,<br>abstract  | 1-19                  |
| A  | WO 2010/111261 A1 (AZUKI SYSTEMS INC [US]; MA KEVIN J [US]; XU JIANGUO [US]; NG TUNG [US]) 30 September 2010 (2010-09-30)<br>abstract  | 1-19                  |
| A  | PANTOS R ET AL: "HTTP Live Streaming; draft-pantos-http-live-streaming-06.txt",<br>HTTP LIVE STREAMING;<br>DRAFT-PANTOS-HTTP-LIVE-STREAMING-06.TXT,<br>INTERNET ENGINEERING TASK FORCE, IETF;<br>STANDARDWORKINGDRAFT, INTERNET SOCIETY (ISOC) 4, RUE DES FALAISES CH- 1205 GENEVA, SWITZERLAND,<br>no. 6, 1 April 2011 (2011-04-01), pages 1-24, XP015075138,<br>[retrieved on 2011-04-01]<br>section 1 | 8                     |
|  | -----<br>-/--  |                       |

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/SG2012/000038

| C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT |  |                       |
|--|--|-----------------------|
| Category*  | Citation of document, with indication, where appropriate, of the relevant passages   | Relevant to claim No. |
| A  | FIELDING DAY SOFTWARE J GETTYS ONE LAPTOP<br>PER CHILD J MOGUL HP H FRYSTYK MICROSOFT L<br>MASINTER ADOBE SYSTEMS P LEACH MICROSOFT T<br>BE: "Hypertext Transfer Protocol --<br>HTTP/1.1; draft-lafon-rfc2616bis-04.txt",<br>20071118,<br>no. 4, 18 November 2007 (2007-11-18),<br>XP015054229,<br>ISSN: 0000-0004<br>section 3.6.1<br><br>----- | 8                     |

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/SG2012/000038

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|--|------------------|-------------------------|------------------|
| WO 2010111261 A1                       | 30-09-2010       | CA 2759880 A1           | 23-09-2010       |
|  |                  | US 2012004960 A1        | 05-01-2012       |
|  |                  | US 2012005364 A1        | 05-01-2012       |
|  |                  | US 2012005365 A1        | 05-01-2012       |
|  |                  | WO 2010111261 A1        | 30-09-2010       |
| -----                                  |                  |                         |                  |