OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(12) (19) (CA) **Demande-Application**

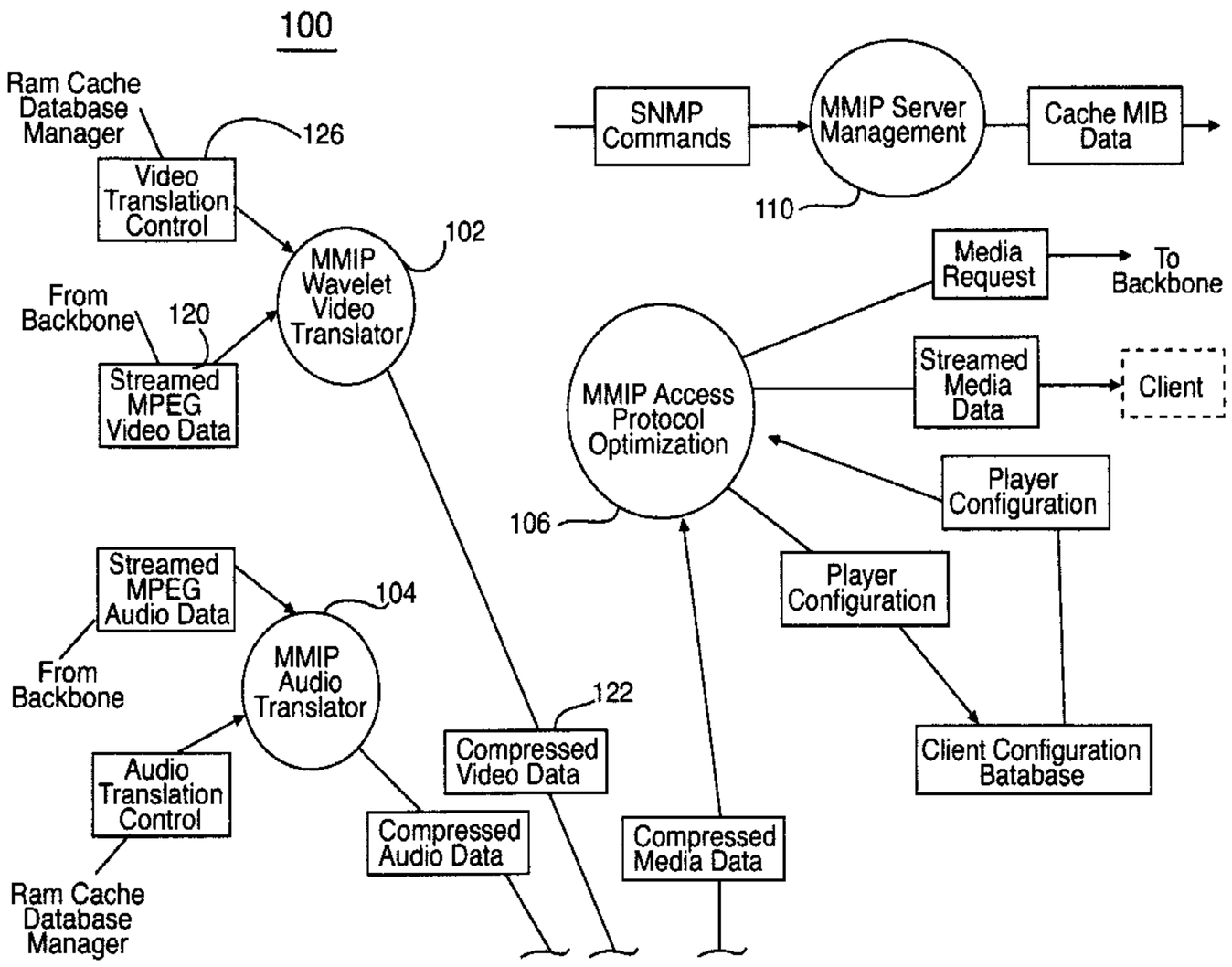
(21) (A1) **2,280,662**
(22) 1999/09/02
(43) 2000/11/21

(72) TOTH, JOE, CA
(72) SCHELLENBERG, JAMES, CA
(72) GRAVES, DAVID, CA
(71) EDGE NETWORKS CORPORATION, CA

(51) Int.Cl.[6] H04L 12/16, H04L 29/06
(30) 1999/05/21 (2,272,590) CA
(30) 1999/07/09 (2,277,373) CA

(54) **SERVEUR DE MEDIA A COMPRESSION EVOLUTIVE MULTIDIMENSIONNELLE DES DONNEES**

(54) **MEDIA SERVER WITH MULTI-DIMENSIONAL SCALABLE DATA COMPRESSION**



(57) A method and system for managing scalable compression of multicast media over an Internet protocol, between a media server and a client. The method includes the steps of determining the client access bandwidth statistics, determining a data buffer status of the client, generating a quantization mask in response to the buffer status, applying the quantization mask to an array of transformed frame data for each frame in a sequence, or group of frames in three-dimensional case, to produce quantized data, performing arithmetic coding on the quantized data to produce a bit stream; and transmitting the bitstream to the client.

Abstract

A method and system for managing scalable compression of multicast media over an Internet protocol, between a media server and a client. The method includes the steps of determining the client access bandwidth statistics, determining a data buffer status of the client, generating a quantization mask in response to the buffer status, applying the quantization mask to an array of transformed frame data for each frame in a sequence, or group of frames in three-dimensional case, to produce quantized data, performing arithmetic coding on the quantized data to produce a bit stream; and transmitting the bitstream to the client.

MEDIA SERVER WITH MULTI-DIMENSIONAL SCALABLE DATA COMPRESSION

The present invention relates to a system and method for providing scalable data compression in a caching server and more particularly to a system and method for optimizing and managing such compression for streaming media edge servers.

BACKGROUND OF THE INVENTION

The Internet today is defined as the interconnection of Internet protocol (IP) - based networks. The Internet protocol stack diagram is represented in terms of the ISO - 7-layer model. Various equipment types and products may be associated with the layer functionality that they service.

The Internet may be viewed as a single integrated network in which various access types are interconnected to various backbone types through edge servers and edge equipment (also called remote access servers or network access servers). There are approximately twenty or more different variations of access paths that can be used to connect the backbone services to a customer (interchangeably referred to as a client). There are six basic access types of connections, namely wireless terrestrial, wireless, satellite, copper, coaxial cable and fiber. In the future, additional access types may be created.

Thus far, the Internet has been a resounding success. Ironically, it is this very success and more specifically, the success of the graphical World Wide Web (the web) that may be its undoing. The number of web subscribers, content providers, and requests by those subscribers for content grows exponentially faster than the capability of the network to meet the demand. The majority of current data transfers involve text and graphics. However, the future of the Internet appears to be evolving towards the transfer of full motion video and audio.

As web sites continue to increase their multimedia content through the integration of audio, video and data, the ability of the web to effectively deliver this media to Internet end users will yield a congestion problem due to the nature of the web. One of the features that has made the web such a success is the ability of one user to access another

user's information regardless of where that information is stored, what type of computer it is stored on, or what kind of application was used to create it. Unfortunately, the same flexibility and ease of use features result in a serious contention issue, since everyone competes for the same available network resources. Streaming technologies for live audio and video over the web have exacerbated this problem even further.

Streaming media is different from the typical transfer of multimedia data. Normally, hyperlinks point to multimedia files that are downloaded in their entirety to a user's local disk before being viewed or played. However, streaming media allows users to watch live video and audio as the file is downloading. Streaming media often requires a continual transfer of large volumes of data. If many people request the same data at the same time, it will lead to bandwidth restrictions or bottlenecks.

Some of the reasons for these bandwidth restrictions or bottlenecks are highlighted in the following description. Since the Internet is IP based, all packets must be evaluated by routers to determine the destination delivery paths, creating traffic congestion, particularly with the increased demand in real-time media, such as video and audio. It has been found that backbone, subnet and router upgrades are not sufficient to increase the Internet throughput to offset the increasing bandwidth requirements of the WWW itself. This problem is further exacerbated by end users having faster access to the ISP POP (Internet Service Provider Point of Presence). Providing "bigger pipes" to the POP simply sends bigger chunks of data onto the web. In another attempted solution, real-time protocols and specialized backbones have been developed. However, these solutions are suitable only for improving transports for scheduled or premium events, but are unsuitable for the proliferation of multimedia content that is expected in the near future. Although improved compression techniques promise to squeeze multimedia files into smaller and smaller sizes, video and audio will continue to require a "big pipe" as a result of the real-time transport requirements.

Oracle Corporation has proposed a solution to the above problem. In this solution, it is proposed that the multimedia data repository is placed closer to the consumer of the multimedia. Thus, servers are deployed at the edge of the web and multimedia data is replicated on these edge servers where the user connection terminates at the POP. Hyperlinks on the web pages become pointers to streaming media servers

that are physically closest to the consumer. The philosophy behind this implementation is that the POP is the logical termination of the user's access point, and thus packets flowing into or out of the POP are only limited by the access speed of the user's connection. Any data packets that flow behind or through the ISP back channel, for example, router, are affected by bottlenecks. Thus, by placing the media repository at the POP and behind the router, the user is insulated from traffic conditions that exist on the Internet at any given time. It is envisioned that content providers and web publishers use a combination of mirroring or caching techniques to replicate data to the edge servers.

A disadvantage of the above scheme is that in the mirroring scheme it requires the content providers and web publishers themselves to stage, propagate, and update the multimedia data to be replicated. In the caching model, if the requested data by the user was not already cached, a dialogue box would inform the user with the approximate time the media would be available and might suggest that they visit other sites in the interim. In general, both situations are unacceptable to most users since most users require instant access to the requested data.

A further improvement on this method and particularly applicable to streaming media, has been proposed by Real Networks which introduced a distributed multi-tier broadcast architecture for the Internet termed the Real Broadcast Network (RBN).

In this solution, access to the RBN server is distributed throughout the Internet backbone. Live feed is transmitted directly to splitters, which are located in the major backbone provider's network. This feed is then retransmitted or "split" from the backbone provider to splitters installed at the ISP site, where it is finally streamed to the user's computer.

Another solution that Real Networks proposes in order to counter the problem of providing high quality media (video and audio) to streaming users while accommodating the various physical connection speeds between the user and the ISP, is to create a scalable stream where the server can reduce the amount of data being sent to keep the client from rebuffering. This approach is generally referred to as video "stream-thinning". The limitations of this approach is that a video or audio file designed to play at one data rate and subsequently scaled down to a lower rate results in an inferior quality level when compared to a video optimized specifically for the lower data rates.

Furthermore, audio codecs cannot usually dynamically send to lower data rates. An approach to address this heterogeneous connection rate environment is to create several files so that when a client connects, the server streams the appropriate file. This has been referred to as "bandwidth negotiation". This process is not dynamic, so if a user's actual throughput changes due to congestion or packet loss, the server cannot adjust. Another difficulty is the increased labor required for coding and then managing the media clip for different bandwidths. The Real Networks solution to these problems in its most recent incarnation is to provide an encoding framework for combining multiple data streams, each at different bit rates into a single file. A sophisticated client server mechanism is provided for detecting changes in bandwidth and translating those changes into combinations of different streams.

While the above attempts to address the solution of bandwidth negotiation and stream thinning, it still suffers from the limitation in that multiple streams corresponding to different bit rates must still be composed at the server end. For example, if ten different streams are to be composed each ranging from 1 megabit per second to 50 kilobits per second, then all ten streams are composed at the server end on the backbone to the POP. Thus, a 1.8-megabit per second stream is sent down the backbone. At the POP, ten different caches are now required. The POP then forwards the appropriate bit stream to the user depending on the user's access capability. It may be seen that in this solution, the user is provided with a relatively consistent stream. However, it still does not alleviate the problem of backbone congestion since multiple streams must all be transmitted along the backbone.

An improvement in the current architecture is described in the subject applicants pending Canadian application Serial no. 2,272,590 filed May 21, 1999 and titled "System and Method for Streaming Media over an Internet Protocol". In this architecture communication between a client and a continuos media server is implemented by the media server composing data to be transmitted into a backbone common format; the server transmitting the backbone common format data to the client POP; converting at the POP the backbone common format data into a plurality of access common format data for transmission to ones of a plurality of clients. In this system a single high quality data stream may be transmitted to an edge server. The edge server or POP filters the data to

the respective client bandwidth capabilities. In a further embodiment of this architecture the edge server may also utilize trans-compression techniques to adapt the received data for filtering to the respective client bandwidth capabilities.

Traditionally, image compression methods may be classified as those methods, which reproduce the original data exactly, that is, "lossless compression" and those, which trade a tolerable divergence from the original data for greater compression, that is, "lossy compression". Typically, lossless methods have a problem that they are unable to achieve a compression of much more than 70%. Therefore, where higher compression ratios are needed, lossy techniques have been developed. In general, the amount by which the original media source is reduced is referred to as the compression ratio. Compression technologies have evolved over time to adapt to the various user requirements. Historically, compression technology focused on telephony, where sound wave compression algorithms were developed and optimized. These algorithms all implemented a one-dimensional (1D) transformation, which increased the 1D entropy of the data in the transformed domain to allow for efficient quantization and 1D data coding.

Compression technologies then focused on two-dimensional (2D) data such as images or pictures. At first, the 1D audio algorithms were applied to the line data of each image to build up a compressed image. Research then progressed to the point today where the 1D algorithms have been extended to implement a two dimensional (2D) transformation, which increases the 2D entropy to allow for efficient quantization and 2D data coding.

Currently, state of the art technology requires compression of moving pictures or video. In this area, research is focused on applications of 2D image coding algorithms to a multitude of images which comprise video (frames) and apply motion compensation techniques to take advantage of correlation between frame data. For example, United States Patent No. RE 36015, re-issued December 29, 1998, describes a video compression system which is based on the image data compression system developed by the motion picture experts group (MPEG) which uses various groups of field configurations to reduce the number of binary bits used to represent a frame composed of odd and even fields of video information.

In general, MPEG systems integrate a number of well known data compression techniques into a single system. These include motion compensated predictive coding, discrete cosine transformation (DCT), adaptive quantization and variable length coding (VLC). The motion compensated predictive coding scheme processes the video data in groups of frames in order to achieve relatively high levels of compression without allowing the performance of the system to be degraded by excessive error propagation. In these group of frame processing schemes, image frames are classified into one of three types: the intraframe (I-Frame), the predicted frame (P-Frame) and the bi-directional frame (B-Frame). A 2D DCT is applied to small regions such as blocks of 8 x 8 pixels to encode each of the I-Frames. The resulting data stream is quantized and encoded using a variable length code such as amplitude run length Huffman code to produce the compressed output signal. As may be seen, this quantization technique still focuses on compressing single frames or images which may not be the most effective means of compression for current multimedia requirements. Also, for low bit rate applications, MPEG suffers from 8 x 8 blocking artifacts known as tiling. Furthermore, these second-generation compression approaches as described above, have reduced the media of data requirements for video by as much as 100:1. Typically, these technologies are focused on the following approaches: wavelet algorithms and vector quantization.

The wavelet algorithms are implemented with efficient significance map coding such as EZW and line detection with gradient vectors depending on the application's final reconstructed resolution. The wavelet algorithms operate on the entire image and have efficient implementation due to finite impulse response (FIR) filter realizations. All wavelet algorithms decompose an image into coarser, smooth approximations with low pass digital filtering (convolution) on the image. In addition, the wavelet algorithms generate detailed approximations (error signals) with high pass digital filtering or convolution on the image. This decomposition process can be continued as far down the pyramid as a designer requires where each step in the pyramid has a sample rate reduction of two. This technique is also known as spatial sample rate decimation or down sampling of the image where the resolution is one half in the next sub-band of the pyramid.

In vector quantization (VQ), algorithms are used with efficient codebooks. A single frame from a video stream is divided into macroblocks of 8x8 or 16x16 pixels, each macroblock thus has either 64 or 256 states which are input to a codebook (look-up table) to produce N unique output codes where N is much less than 64. The VQ algorithm codebooks are based on macroblocks (8 x 8 or 16 x 16) to compress image data. These algorithms also have efficient implementations. However, they suffer from blocking artifacts (tiling) at low bit rates (high compression ratio). The codebooks have a few codes to represent a multitude of bit patterns where fewer bits are allocated to the bit patterns in a macro block with the highest probability.

As discussed earlier, these current techniques are limited when applied to third generation compression requirements, that is, compression ratios approaching 1000:1. That is, wavelet and vector quantization techniques as discussed above still focus on compressing single frames or images which may not be the most effective for third generation compression requirements.

A vastly improved compression technique over the current techniques is described in a pending Canadian Patent Application filed July 9, 1999 and titled "Multi-Dimensional Data Compression", also assigned to the subject applicants. The technique as described in this application may be applied to video data signals. The method comprises the steps of selecting a sequence of image frames in a video stream, applying a three dimensional transform to the selected sequence to produce a transformed output, and then encoding the transformed output to produce a compressed stream output.

Given the current Internet architecture and compression, there is a need for a multicast media over Internet protocol (MMIP) caching bandwidth manager implemented in an edge server which uses the above described compression techniques for efficiently streaming media over the client access link and which adapts to the client access bandwidth.

SUMMARY OF THE INVENTION

In accordance with this invention there is provided a method for managing scalable compression of multicast or unicast media over an Internet protocol, between a media server and a client, the method comprising the steps of:

(a) determining the client access bandwidth statistics and client hardware capabilities;

(b) determining a data buffer status of the client;

(c) generating a quantization mask in response to the buffer status;

(d) applying the quantization mask to an array of transformed frame data for each frame in a sequence, or group of frames in 3D case to produce quantized data;

(e) performing entropy coding on the quantized data to produce a bit stream; and

(f) transmitting the bitstream to the client.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features of the preferred embodiments of the invention will become more apparent in the following detailed description in which reference is made to the appended drawings wherein:

Figure 1 is a schematic diagram of an Internet architecture;

Figure 2 is a schematic system diagram of an edge or gateway server located at an ISP according to an embodiment of the invention;

Figure 3 is a schematic system diagram of a client according to an embodiment of the invention;

Figure 4 is a schematic functional block diagram of a caching bandwidth manager of figure 2;

Figure 5 is a schematic functional block diagram of a client player of figure 3;

Figure 6 is a schematic flow diagram of the bandwidth manager operation;

Figure 7 is a schematic flow diagram of a bandwidth optimizer according to an embodiment of the present invention;

Figure 8 is a schematic flow diagram of a quantizer according to an embodiment of the present invention;

Figure 9 is a schematic diagram of a real-time frame by frame quantizer for a 2D wavelet case; and

Figure 10 is a schematic diagram of a real-time frame by frame quantizer for a 3D spectral case.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following description, like numerals refer to like structures in the drawings. Referring to figure 1, a general Internet architecture as it currently exists is shown generally by numeral 20. The architecture comprises a backbone network 22 which is defined as the interconnection equipment concerned with connecting local web sites to local POPs and an access network 24 that is defined as the interconnection between the local POPs and the consumers. The Web sites 26 host both digital and analog content from various Content providers 28, which are in-turn connected via a global and national Internet infrastructure 30 to the local access Internet infrastructure 32. The consumer or viewer 34 connects to the national Internet infrastructure 30 i.e. at the POP, by one or more access links 33. Cache 36 sites are provided between the global and national Internet infrastructure 30 and the local access Internet infrastructure 32. The cache sites 36 are normally the demarcation between the backbone network 22 and the access network 24. Backbone web sites 26 typically do not consider the needs of various types of access 33 employed by clients 34 and various qualities of access links in their consideration of web content. It is normally the responsibility of the web content provider 28 to customizing the web site content for different access links.

Therefore the present invention leverages off the existing architecture, but implements a content compression architecture that uses technology in the data link level to application level of the ISO model to optimize the access from the local POP to the customer 34.

Referring to figure 2, a system block diagram of local access server 37 is shown, while in figure 3, a client or consumer 34-system block diagram according to an embodiment of the present invention is shown. In this case the local access link 32 is a 56k modem.

Referring to figure 4, a schematic diagram of the functional blocks of a multicast media over an internet protocol (MMIP) caching bandwidth manager implemented in the local access server or edge server 37 is indicated generally at numeral 100. The manager 100 functions as a network edge server/gateway and is compliant with the IETF working groups protocol recommendations for streaming media. The manager 100 includes an m-dimension wavelet or spectral video translator module 102, an audio translator module

104, an access protocol optimization module 106, a cache database manager module 108, and a server management module 110.

In general the function performed by the manager 100 is to stream media over IP to a client player, shown in the schematic system diagram of figure 3 and the corresponding functional diagram of figure 5, by implementing a controlled compression or extended client filtering. In controlled compression the manager 100 receives compressed video from standard COTS video servers and then translates the video stream using an m-dimensional wavelet or spectral codec. The manager 100 is also capable of receiving m-dimensional compressed video streams over IP from a originating media server in accordance with an embodiment of the present invention.

The caching bandwidth manager 100 receives streamed MPEG data compliant with RTP and sends translated streamed media data in wavelet format with header compression that is optimized for the access link 33. In addition the manager receives end user access configuration data in the form of client video/audio capabilities, access link capabilities, and media requests. The caching bandwidth manager utilizes the configuration data to send MPEG server requests to indicate when a user requests media from the WWW.

The caching bandwidth manager 100 performs the functions of an edge server or traditional gateway for converting protocols between the backbone network and the access network.

Each of the elements of the caching manager as shown in figure 4 will be described in detail below. Thus referring back to figure 4, the m-dimensional wavelet video translator module 102 receives streamed MPEG video data 120 from an MPEG media server (not shown) that is preferably compliant with RFC2250. The MPEG data is decompressed back into the luminance and chromanence frame data values. This uncompressed frame data is then re-compressed using controlled compression according to an embodiment of the present invention, as described below. All the wavelet significance maps or sub-bands defined by the wavelet pyramidal decomposition, as described in copending Canadian Patent application titled "Multi-Dimensional Data Compression", incorporated herein by reference, is stored as compressed video data 122 in a media database 124 along with timestamp headers that are compliant with the

standards definition in RTP. The wavelet video translator module 102 is enabled by a video translation control signal 126 generated by the edge server when the access configuration data indicates that there is an MPEG media request being made by the end user.

The wavelet stream is stored in the media database 124 as N level deep pyramidal multiresolution sub-bands coded with the embedded zerotrees of wavelet coefficients (EZW) compression algorithm. The highly correlated lowest resolution significance map or sub-band in the tree is processed with an algorithm such as the Discrete Cosine Transform (DCT), and then entropy coded with Huffman coding or arithmetic coding. The EZW algorithm is used to code all the other subbands or children of the lowest resolution subband that is coded by the DCT. This technique will result in efficient compression of the principle components of the video stream by a method, which closely approximates the optimal Karhunen-Loeve transformation.

The Audio Translator module 104 receives streamed MPEG audio data from the Media Server that is compliant with RFC2250. The compressed audio stream may be uncompressed back to sample data values and this uncompressed stream efficiently re-compressed with timestamps using the audio wavelet codec. The audio translator module 104 sends compressed audio data to the media database for efficient streaming to the client player over the access link.

An efficient application of the EZW transformation algorithm is in providing progressive video over various access link bandwidths. The Access Protocol Optimization module 106 uses access optimization protocols, such as the controlled compression and extended client filtering to read Compressed Media Data from the Media Database and Stream Media Data in the form of time synchronized media wavelet coefficients to the Client Player based on the available bandwidth and the MMIP Client Player configuration. For low speed access links, the PPP (RFC1661) and the Serial Line Internet Protocol (SLIP) shall be supported with 10:1 IPv4 header compression compliant to RFC1144 and RFC2508/RFC2509 respectively. These IETF recommendations discuss lossless header compression algorithms to reduce the redundancies in the header addresses and timestamps by using difference products. RFC2507 for non serial links for header compression in mobile IP, etc will result in approximately 15:1 IPv6 compression

of the header information. As an example, at 50 packets per seconds, the UDP/IPv4 headers consume 11.2 kbits/s and UDP/IPv6 headers consume 19.2 kbits/s when uncompressed. Using RFC2507, the overhead can be reduced to approximately 1.7 KBPS. In addition, up to 2:1 lossless compression can be achieved for the packet payload if the web data is not already in an encrypted or compressed format. An algorithm is used here to detect if the compression of the web data results in data expansion. Data expansion is typical when a compression algorithm is applied to encrypted data. The Access Protocol Optimization module supports RTP according to RFC1889 and RTSP according to RFC2326 to stream media to the Client Player.

The Access Protocol Optimization module 106 implements algorithms to efficiently perform Controlled Compression and Extended Client Filtering in media streaming to the MMIP Client Player over the access link bandwidth. The algorithms implemented include the following functions:

1) Stream the multiresolution subbands to the Client Player utilizing RTP/UDP/IP.

2) Stream the multiresolution subbands to the Client Player utilizing RTP/TCP/IP for streaming of media through firewalls that do not support UDP port assignments.

3) Stream only M of a total of N multiresolution subbands starting from the lowest resolution, where $M \leq N$. The algorithm for which M subbands to stream to the Client Player is based on access infrastructure and bandwidth available (i.e.: twisted pair, wireless, satellite) to maintain the best subjective image quality according to the human visual systems logarithmic sensitivity to light intensity and sensitivity to abrupt spatial changes.

4) Selectively stream only an area of each of the M subbands in 3). The algorithm will determine the shape of the area to stream of the M subbands and vary it from all the lowest subband coefficients to a small geometrical area in the center of the highest resolution subband. The algorithm to control the rate of change of the geometrical area from low to high resolution subbands in the center of the image area is optimized based on the access infrastructure, bandwidth available, and client capability.

5) Selectively eliminate the highest resolution subbands in 4) as the geometrical area approaches a single coefficient in the significance map that will enhance the coding efficiency of the EZW algorithm.

6) Gradual spatial subsampling by decimation of the subbands in 5) to stream data from CIF to QCIF, etc. to maintain full motion video over the access links available bandwidth.

7) YUV color space subsampling to reduce the compressed data rate and to approach a gray scale video while retaining full frame rate video (Y, Cr, Cb subband map tagging).

8) Variable number of frames between key frames based on scene information and use of motion compensation between frames to deliver progressive video between key frames when scene does not change.

9) Frame rate decimation below 30 Hz to delivery gray scale video over low speed access infrastructures (i.e. Personal Digital Assistants).

10) Congestion algorithms optimized for the access infrastructure such as graceful degradation of video while maintaining the audio quality and audio degradation only after the frame rate is zero (RFC2001/RFC2581 identifies TCP congestion control algorithms).

11) IP Packet length Segmentation and Reassembly (SAR) algorithms optimized for the access link bandwidth (i.e. approximately 700 bytes).

12) Optimization algorithms such as header compression and payload compression operating over layers 2 to 7 of the OSI 7 layer network model to optimize the access link bandwidth.

13) Client player buffer size and buffer fullness continuous feedback loop with an algorithm to optimize streaming to the client to avoid overflow or underflow (i.e.: 15 second client player buffer depth for a client configured with 64 Meg RAM 80% free).

14) Scene change detection with variable key frames between video frames (1-90). Use of a Distributed Keyframe to balance loading over access links with lower overall burstiness of traffic and lower delay to display scene change. This provides for progressive video for talk shows, distance education, etc.

Referring to figure 6, a flow chart showing the operation of the caching bandwidth manager 100 is shown generally by numeral 200. Normally the process

begins with the edge server receiving a request from the client for streaming media content 202. The caching bandwidth manager runs two loops, a non-real-time loop 204 and a real-time loop 206. In the non-real-time loop, the bandwidth manager determines the bandwidth characteristics of the client 208. The characteristics are used to generate a quantization mask 210 to be applied by the real time loop 206 on a frame by frame basis, or group of frames in 3D cases, for creating a bitstream to the client. Thus in the real time loop 206, (by real- time is generally meant time in the order of the frame rate) the bandwidth manager receives the transformed pixel data and performs coding thereon using the quantization mask 210. The bitstream generated by this process is then sent 212 to the client. The client performs the inverse operation 214 on the bitstream to generate an appropriate display.

Referring to Figure 7, a flow chart showing the operation of the bandwidth optimizer is indicated generally at numeral 400. Initially, an edge server receives a client request. The client request is buffered and the server determines the access bandwidth statistics of the client. The client buffer status is determined, as to whether the client buffer is approaching an underflow condition or approaching an overflow condition. If the client buffer status indicates that the buffer is approaching an overflow condition, the bandwidth optimizer reduces video quality by changing the co-efficient array quantization. Spatial decimation and temporal decimation follow this. In addition the bandwidth optimizer may perform colorspace reduction and decrease subjective quality based on the human visual system sensitivity. Next the values are updated in the array for the N x N quantization mask and the coded bit rate decreased to the client. The next step is then to update the new N x N quantization in the real time quantization process.

If the client buffer state is determined to be approaching an underflow condition the bandwidth optimizer improves video quality by controlling the co-efficient array quantization. Next spatial interpolation and temporal interpolation are performed as above. Also color space expansion is performed followed by an increase in subject subjective quality using a human visual system sensitivity profile. This is similar to the sensitivity profile described above for the overflow condition however in this case the optimizer moves in the direction to increase the quality of the image. Next the N x N

quantization mask is updated to increase the coded bit rate to the client. It may be noted, that the quantization mask is not restricted to N x N but my be a N x M or N x M x Z.

In both instances, the outputs are used to update the new N x N quantization mask in the real-time quantization process.

Referring to Figure 8, a flow chart showing the real time quantization process is shown generally by numeral 500. It may be shown that the quantization process is repeated for each frame, or group of frames in 3D cases, transformed. The quantization process begins by inputting a transformed pixel data of size N x N. It is next determined whether a new real-time quantization mask is available. If so the system updates the quantizer co-efficient mask from the bandwidth optimizer as described with reference to Figure 7. This quantization mask is then applied to the N x N array of transformed frame data as shown schematically in Figure 9.

Next the quantizer performs entropy (for example arithmetic) coding on the quantized data, the resulting bit stream is then sent to the client.

Referring to Figure 9, a schematic representation of a real-time frame by frame quantization with independent co-efficient quantization is shown for a 2D wavelet case. As shown, a quantization mask of size N x N 604 is created and then applied to the data 602 on a frame by frame basis. The purpose of the quantization mask is for real-time selective wavelet sub-band quantization to reject certain sub-bands or portions of sub-bands for NxN, N x M or N x M x P cases. For example an AND is performed on a corresponding element in the transformed coefficient array 602 with the quantization mask 604. In this scenario the independent value for each quantization mask array element will truncate the corresponding transformed coefficient value in bit positions in the quantization mask array elements that are zero. Referring to the first array element in the quantization mask, which is represented as F4(Hex) when ANDed with the coefficient value of 92(Hex) results in a coefficient value of 90(hex) or in other words less video resolution. On the other hand for an increase in resolution a value of FF(Hex) in the quatization mask will result in a coefficient value of 92(hex).

Referring to Figure 10, a real time group of frames quantization with independent co-efficient quantization for a 3D spectral case is shown generally by numeral 700. In

this case, rather than a 2D quantization mask being created, a 3D quantization mask block is created and applied to an entire sequence of frames.

The cache database manager module 108 receives Access Configuration Data from the user. When the Access Configuration Data indicates an MPEG media request has been made by the end user, the Cache Database Manager sends an MPEG Server request to the MPEG Media Server to enable Streamed MPEG Data to be sent over IP on the WWW. The Cache Database Manager is also be capable of sending Media Requests to enable other Compressed Media Data comprised of audio, video, or data for transmission over IP on the WWW based on the Access Configuration Data received from the user.

Thus an embodiment of the present invention provides a method and system for optimizing the quality of media displayed at a client that is connected to an IP network.

Furthermore, although the invention has only been described in detail relating to media transmissions over an IP network, it may be extended to other forms of data transmission. For example, cable television companies typically transmit over 6MHz channels. Using today's MPEG compression technology, they are generally only able to accommodate from four to six television stations per channel. Using the controlled compression described herein, different types of television shows can be compressed with different compression rates. Therefore, in the cable television industry each 6 MHz channel statistically will be able to accommodate much more television shows. In a similar fashion, the subject of the present invention may be applied to other industries, such as broadcast television, jukeboxes, personal electronic devices and the like.

Although the invention has been described with reference to certain specific embodiments, various modifications thereof will be apparent to those skilled in the art without departing from the spirit and scope of the invention as outlined in the claims appended hereto.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A method for managing scalable compression of multicast or unicast media over an Internet Protocol, between a media server and a client, said method comprising the steps of:

(a) determining the client access bandwidth statistics and client hardware capabilities;

(b) determining a data buffer status of the client;

(c) generating an m-dimensional quantization mask in response to said buffer status;

(d) applying said quantization mask to an array of transformed frame data for each frame in a sequence to produce quantized data;

(e) performing entropy coding on said quantized data to produce a bit stream; and

(f) transmitting said bitstream to said client.

2. A method as defined in claim 1, further comprising the steps of enhancing dynamic video quality by application of extended client filtering or controlled compression of said bit stream.

3. A method as defined in claim 1, wherein said quantization mask is applied to coefficient data generated using wavelets and said quantization mask has real time independent control of the resolution of each coefficient in the m-dimensional transformed pixel array.

4. A method as defined in claim 1, wherein said quantization mask is applied to coefficient data generated using spectral transforms and said quantization mask has real time independent control of the resolution of each coefficient in the m-dimensional transformed pixel array.

5. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by streaming multiresolution subbands to a client player utilizing RTP/UDP/IP.

6. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by streaming multiresolution subbands to a client player utilizing RTP/TCP/IP for streaming of media through firewalls that do not support UDP port assignments.

7. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by streaming only M of a total of N multiresolution subbands starting from the lowest resolution, where $M \leq N$, and wherein an algorithm for determining which M subbands for streaming to a client player is based on access infrastructure and bandwidth available for maintaining the best subjective image quality according to the human visual systems, logarithmic sensitivity to light intensity, and sensitivity to abrupt spatial changes.

8. A method as defined in claim 7, wherein only an area of each of said M subbands are selectively streamed, and an algorithm determines the shape of the area to varies it from the lowest subband coefficients to a small geometrical area in the center of the highest resolution subband, said algorithm is optimized based on the access infrastructure, bandwidth available, and client capability.

9. A method as defined in claim 8, wherein the highest resolution subbands are selectively eliminated as the geometrical area approaches a single coefficient in the significance map that will enhance the coding efficiency of an EZW algorithm.

10. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by decimating subbands for gradual spatial subsampling for streaming data from CIF to QCIF while maintaining full motion video over an access link's available bandwidth.

11. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by YUV color space subsampling for reducing the

compressed data rate and for approaching gray scale video while retaining full frame rate video.

12. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by utilizing a variable number of frames between key frames based on scene information and use of motion compensation between frames for delivering progressive video between key frames when a scene does not change.

13. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by frame rate decimation below 30 Hz for delivering gray scale video over low speed access infrastructures.

14. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by utilizing congestion algorithms optimized for the access infrastructure while maintaining the audio quality and audio degradation only after the frame rate is zero.

15. A method as defined in claim 14, wherein said congestion algorithm is a graceful degradation of video.

16. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by utilizing IP packet length Segmentation And Reassembly (SAR) algorithms optimized for the access link bandwidth.

17. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by optimizing algorithms such as header compression and payload compression operating over layers 2 to 7 of the ISO 7 layer network model for optimizing the access link bandwidth.

18. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by implementing a client player buffer size and buffer

fullness continuous feedback loop with an algorithm for optimizing streaming to the client for avoiding overflow or underflow.

19. A method as defined in claim 2, wherein said extended client filtering or controlled compression is accomplished by utilizing scene change detection with variable key frames between video frames, and the use of a distributed keyframe for balancing loading over access links with lower overall burstiness of traffic and lower delay to display scene change.

20. A method for broadcasting media over a fixed channel, wherein controlled compression or extended client filtering is used for generating a bitstream with dynamic control of the compression of said bitstream.

21. A method as defined in claim 20, wherein the broadcast media is cable television.

22. A method for dynamically controlling video quality of a transmission through controlled compression or extended client filtering within a primary media server for multicast or unicast transmission over an Internet Protocol.

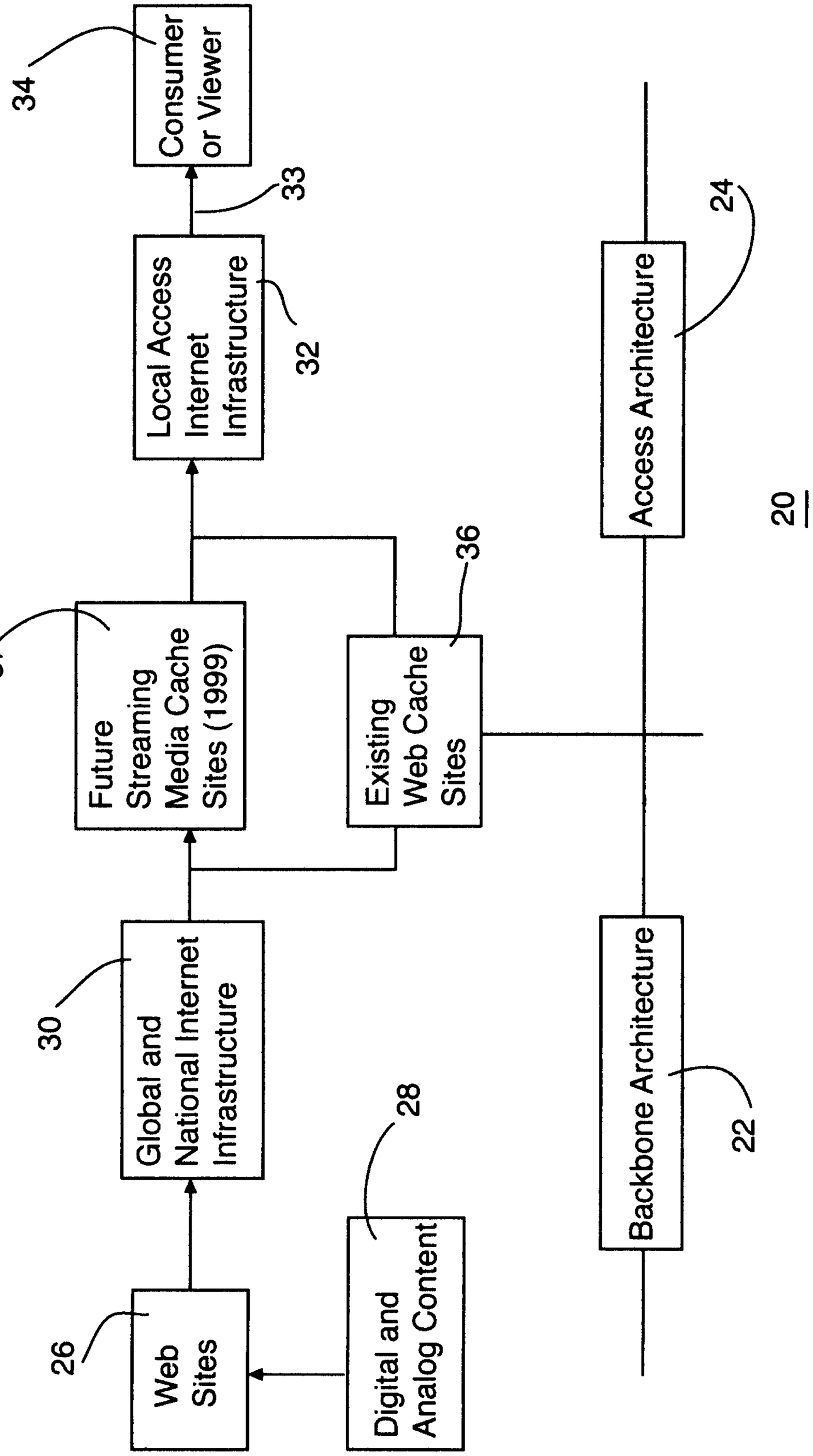
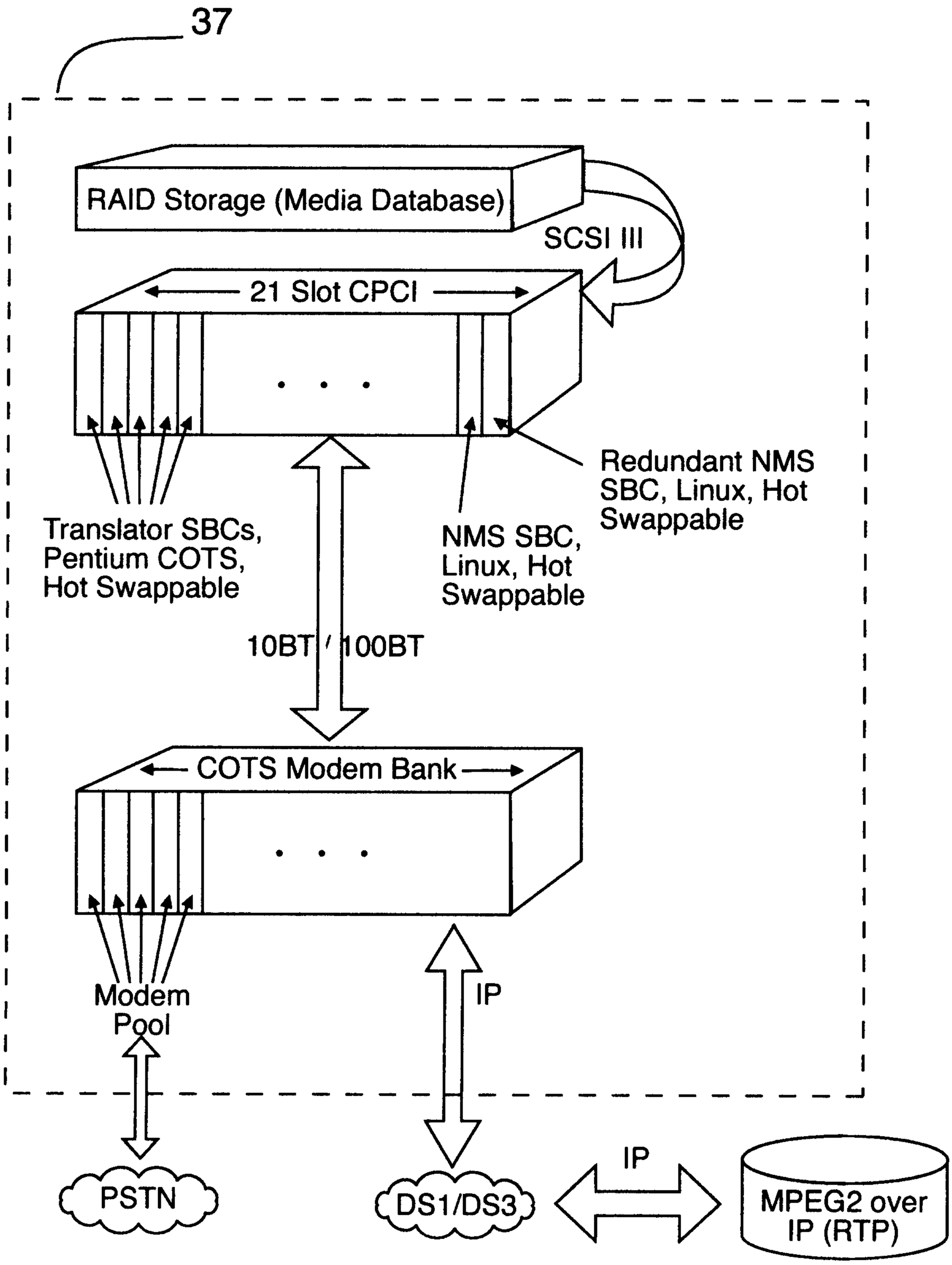

Consumer or Viewer
34
33
Local Access Internet Infrastructure
32
Access Architecture
24
20
Future Streaming Media Cache Sites (1999)
37
Existing Web Cache Sites
36
Global and National Internet Infrastructure
30
Backbone Architecture
22
Digital and Analog Content
28
Web Sites
26

Figure 1

Figure 2


37
RAID Storage (Media Database)
SCSI III
21 Slot CPCI
Translator SBCs,
Pentium COTS,
Hot Swappable
NMS SBC,
Linux, Hot
Swappable
Redundant NMS
SBC, Linux, Hot
Swappable
10BT / 100BT
COTS Modem Bank
Modem
Pool
IP
PSTN
DS1/DS3
IP
MPEG2 over
IP (RTP)

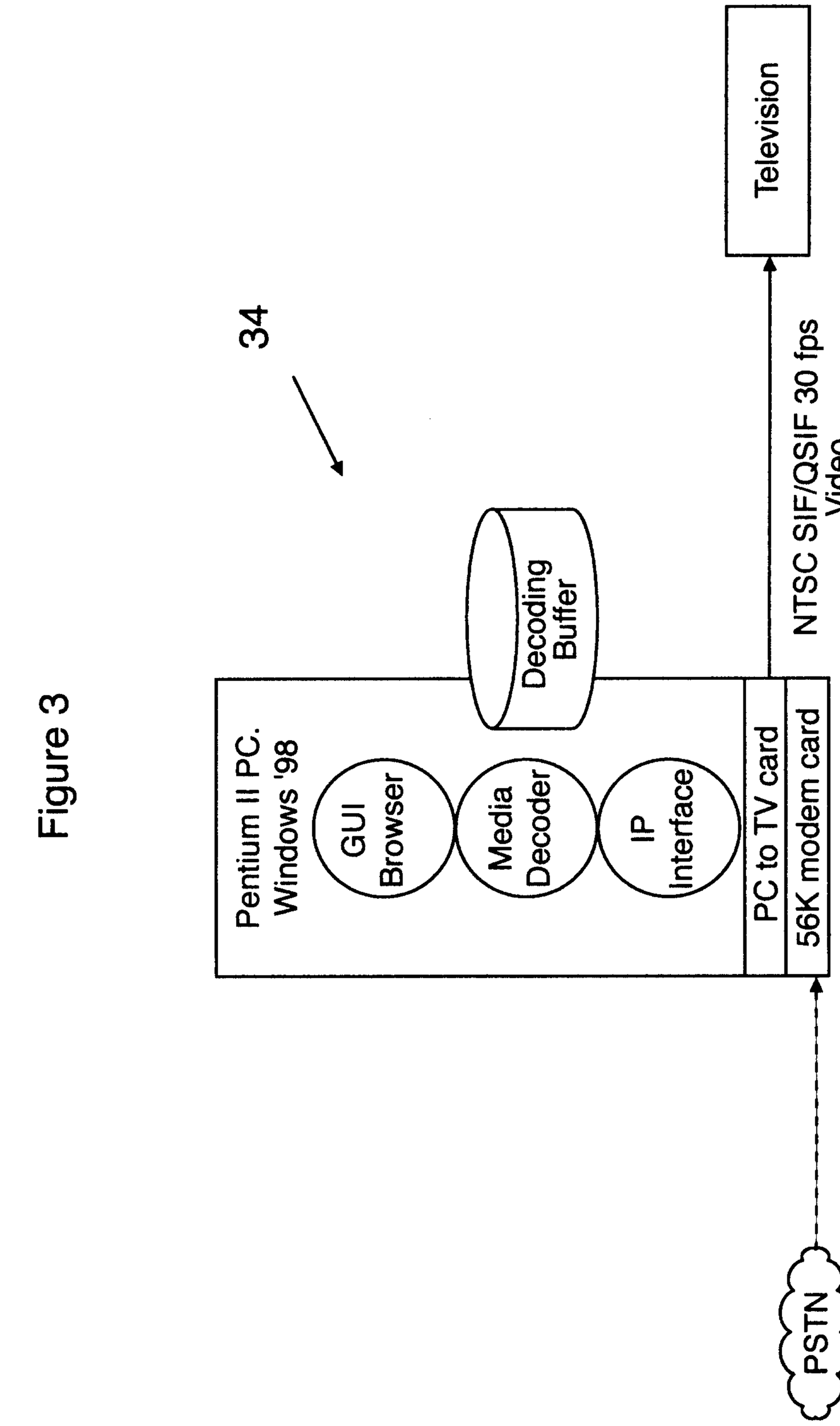

Television
NTSC SIF/QSIF 30 fps Video
34
Decoding Buffer
Pentium II PC. Windows '98
GUI Browser
Media Decoder
IP Interface
PC to TV card
56K modem card
PSTN

Figure 3

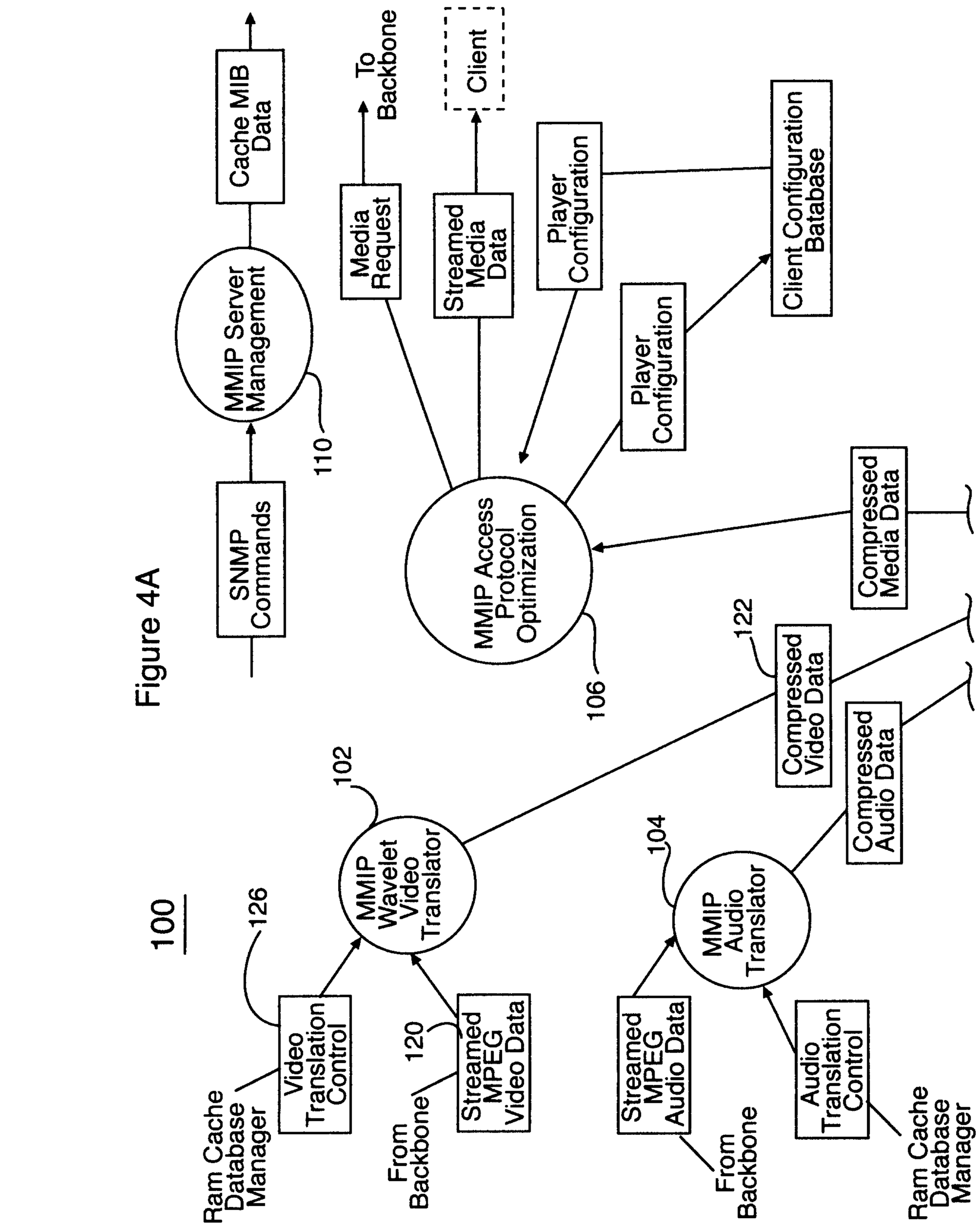
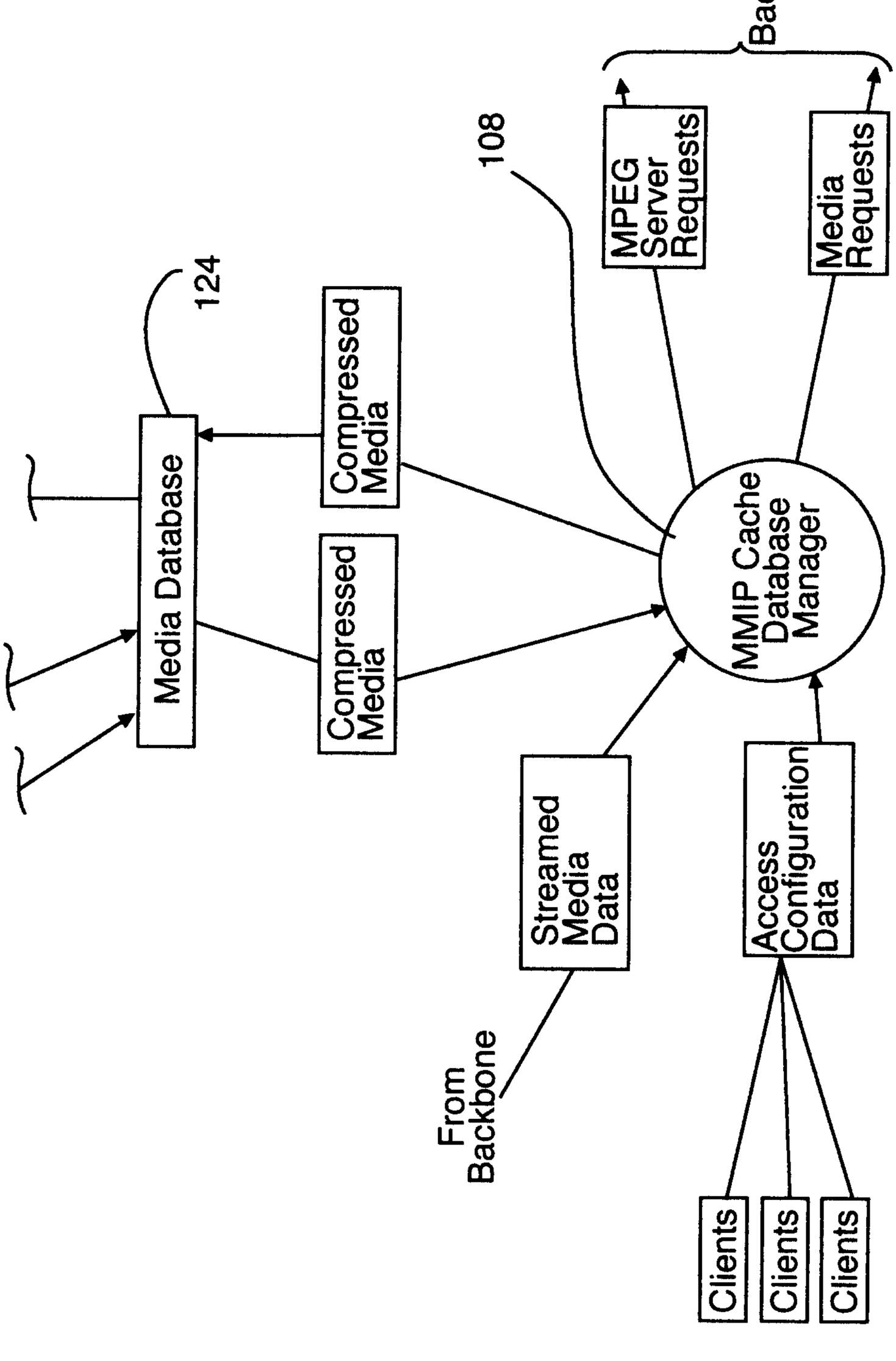

100
SNMP Commands
MMIP Server Management
110
Cache MIB Data
To Backbone
Client
Media Request
Streamed Media Data
Player Configuration
Player Configuration
Client Configuration Batabase
MMIP Access Protocol Optimization
106
Compressed Media Data
122
Compressed Video Data
Compressed Audio Data
102
MMIP Wavelet Video Translator
104
MMIP Audio Translator
126
Ram Cache Database Manager
Video Translation Control
From Backbone
120
Streamed MPEG Video Data
Streamed MPEG Audio Data
From Backbone
Audio Translation Control
Ram Cache Database Manager

Figure 4A

Figure 4B


To Backbone
108
MPEG Server Requests
Media Requests
124
Media Database
Compressed Media
Compressed Media
MMIP Cache Database Manager
Streamed Media Data
Access Configuration Data
From Backbone
Clients
Clients
Clients

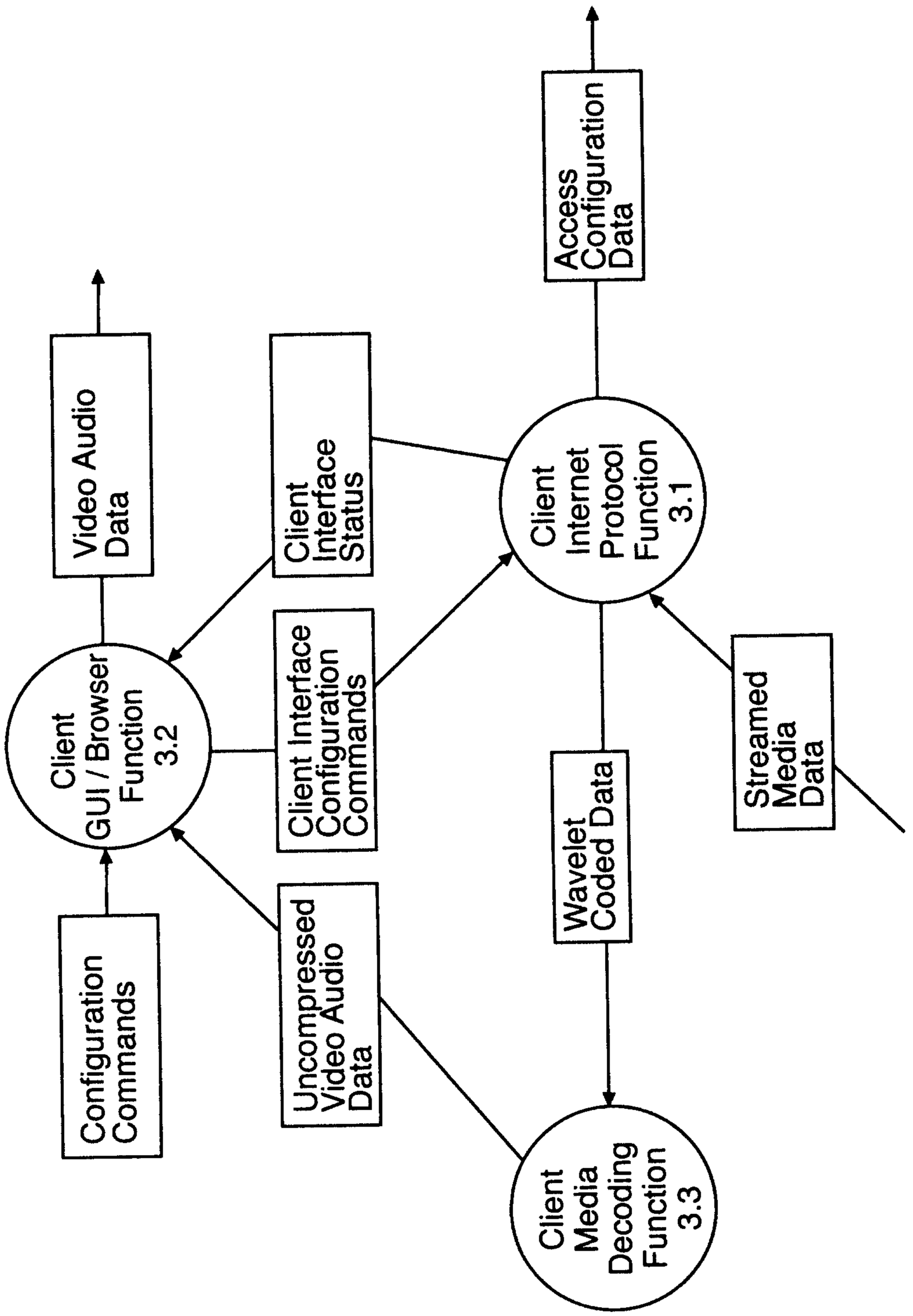

Access Configuration Data
Video Audio Data
Client Interface Status
Client Internet Protocol Function 3.1
Client GUI / Browser Function 3.2
Client Interface Configuration Commands
Streamed Media Data
Configuration Commands
Uncompressed Video Audio Data
Wavelet Coded Data
Client Media Decoding Function 3.3

Figure 5

Figure 6

200

Server Flow Chart

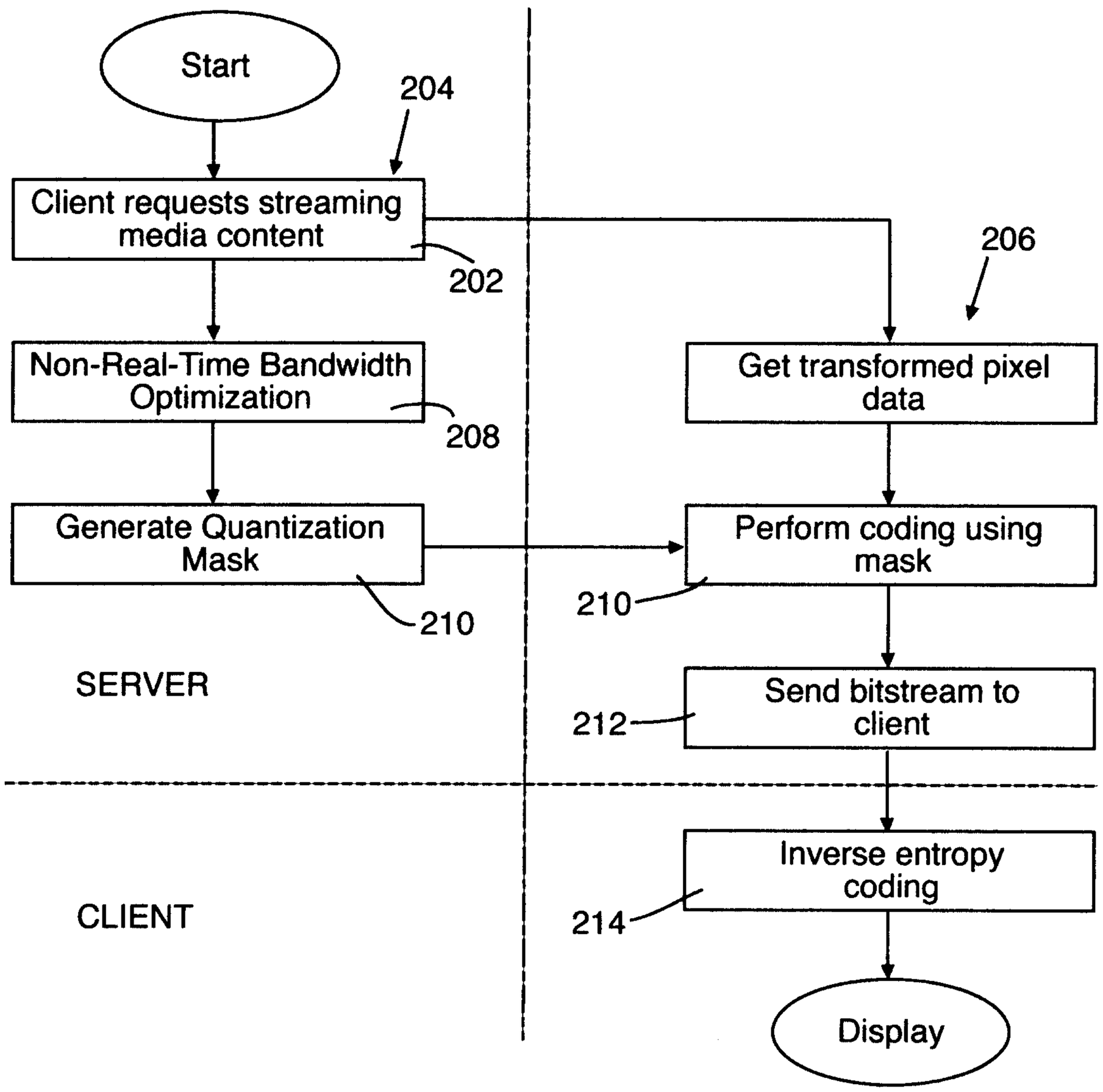
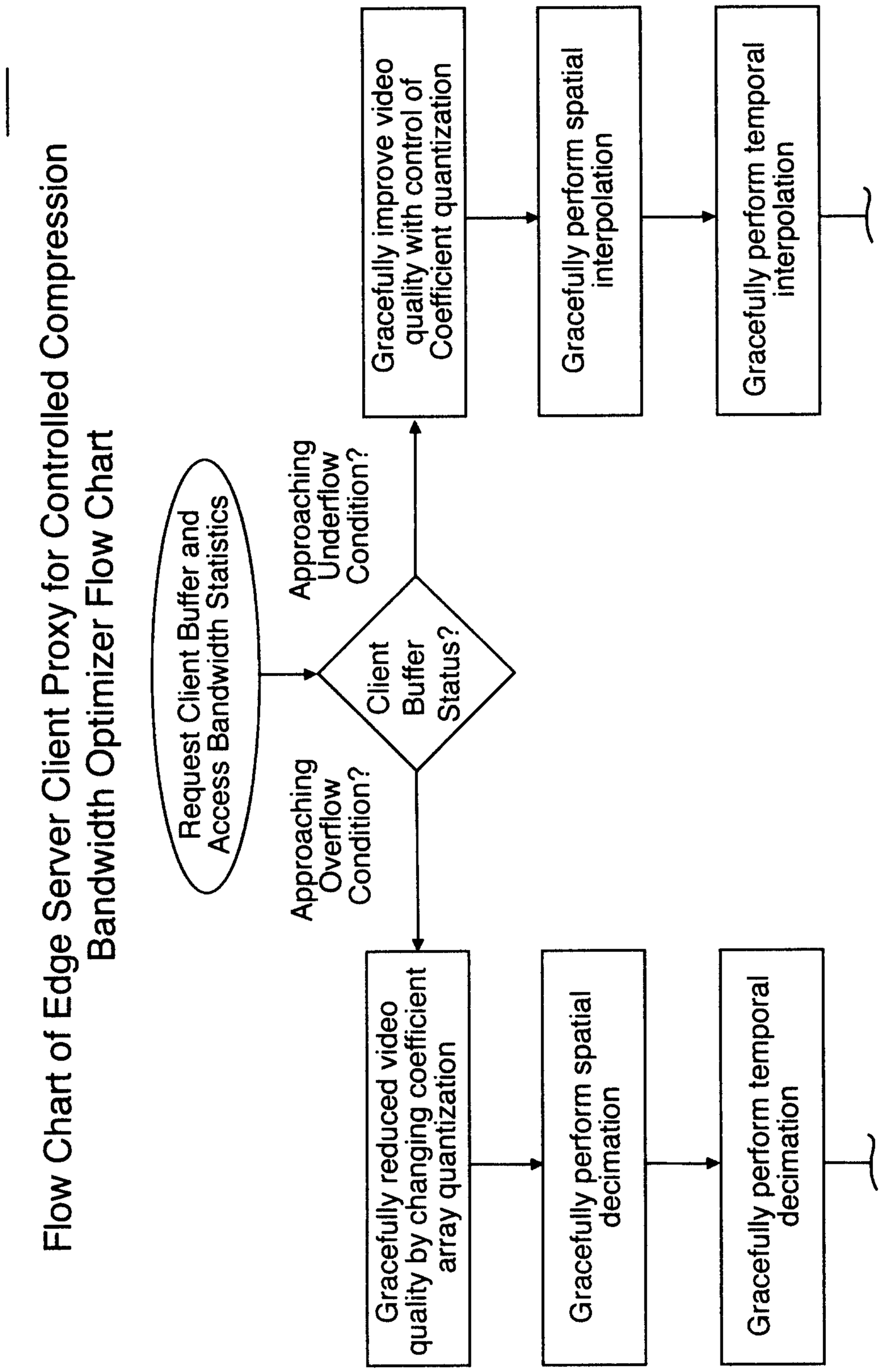


Figure 7A

400

Flow Chart of Edge Server Client Proxy for Controlled Compression
Bandwidth Optimizer Flow Chart



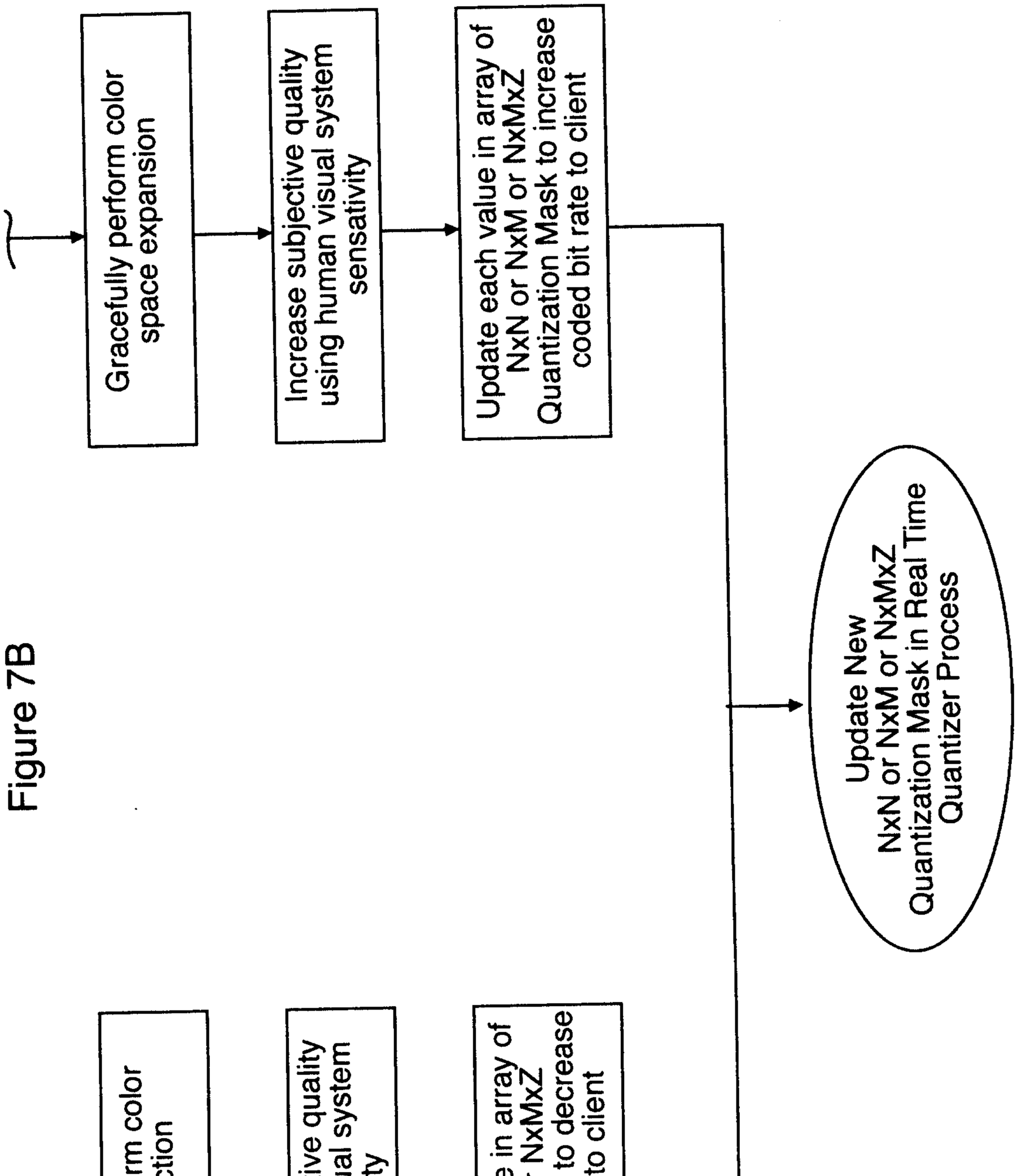

Gracefully perform color space expansion
Increase subjective quality using human visual system sensativity
Update each value in array of NxN or NxM or NxMxZ Quantization Mask to increase coded bit rate to client
Update New NxN or NxM or NxMxZ Quantization Mask in Real Time Quantizer Process
Gracefully perform color space reduction
Decrease subjective quality using human visual system sensativity
Update each value in array of NxN or NxM or NxMxZ Quantization Mask to decrease coded bit rate to client

Figure 7B

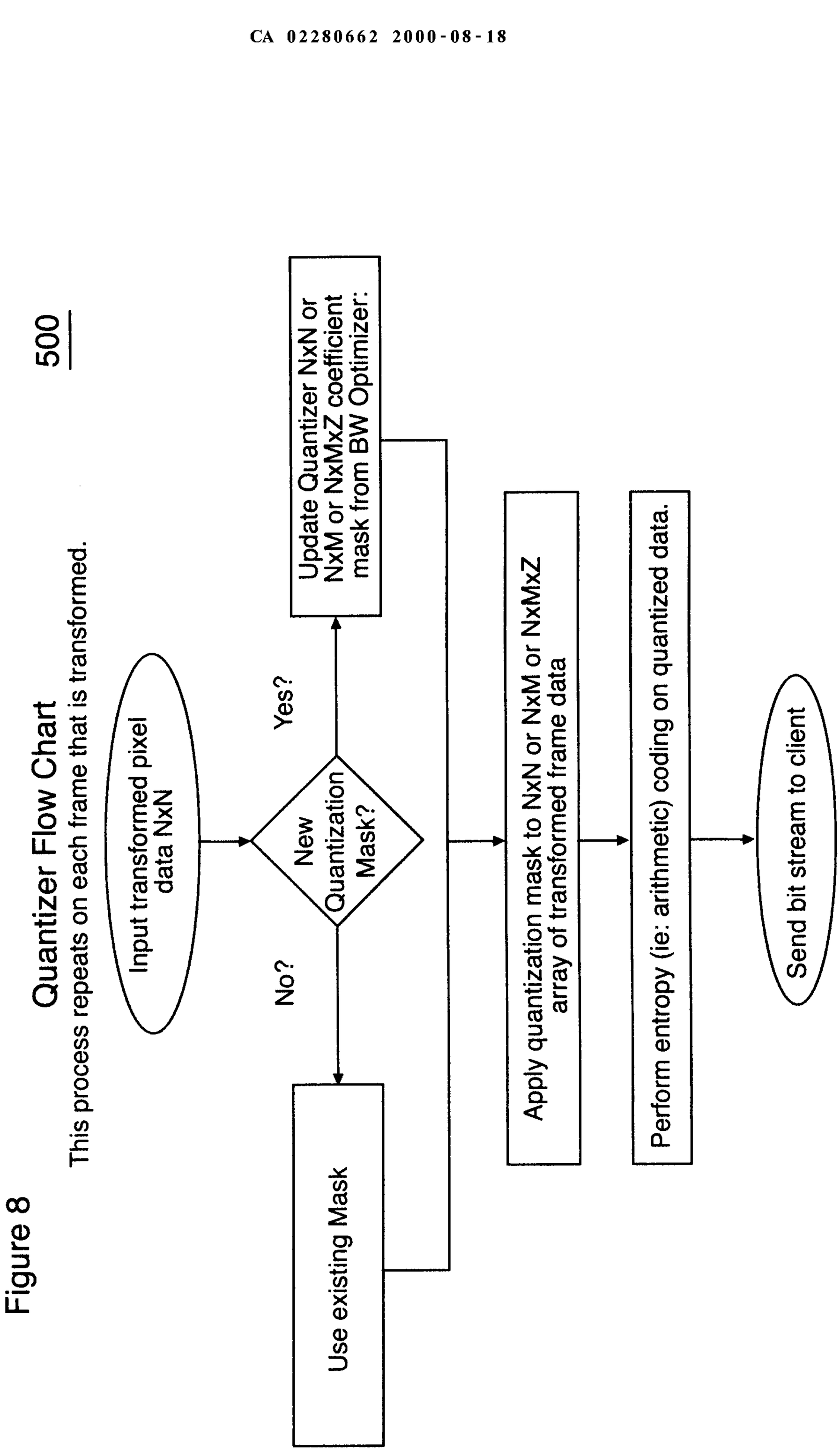

500
Quantizer Flow Chart
This process repeats on each frame that is transformed.
Input transformed pixel data NxN
New Quantization Mask?
Yes?
No?
Update Quantizer NxM or NxMxZ coefficient mask from BW Optimizer:
Use existing Mask
Apply quantization mask to NxN or NxM or NxMxZ array of transformed frame data
Perform entropy (ie: arithmetic) coding on quantized data.
Send bit stream to client

Figure 8

Figure 9

600

2 D Wavelet Scenero:

92	00	10	10	00	00	00	00
26	15	10	10	00	00	00	00
35	00	00	00	00	87	00	00
10	05	00	00	34	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

602

&

&

&

&

Transformed Frame Data
(Coefficients)

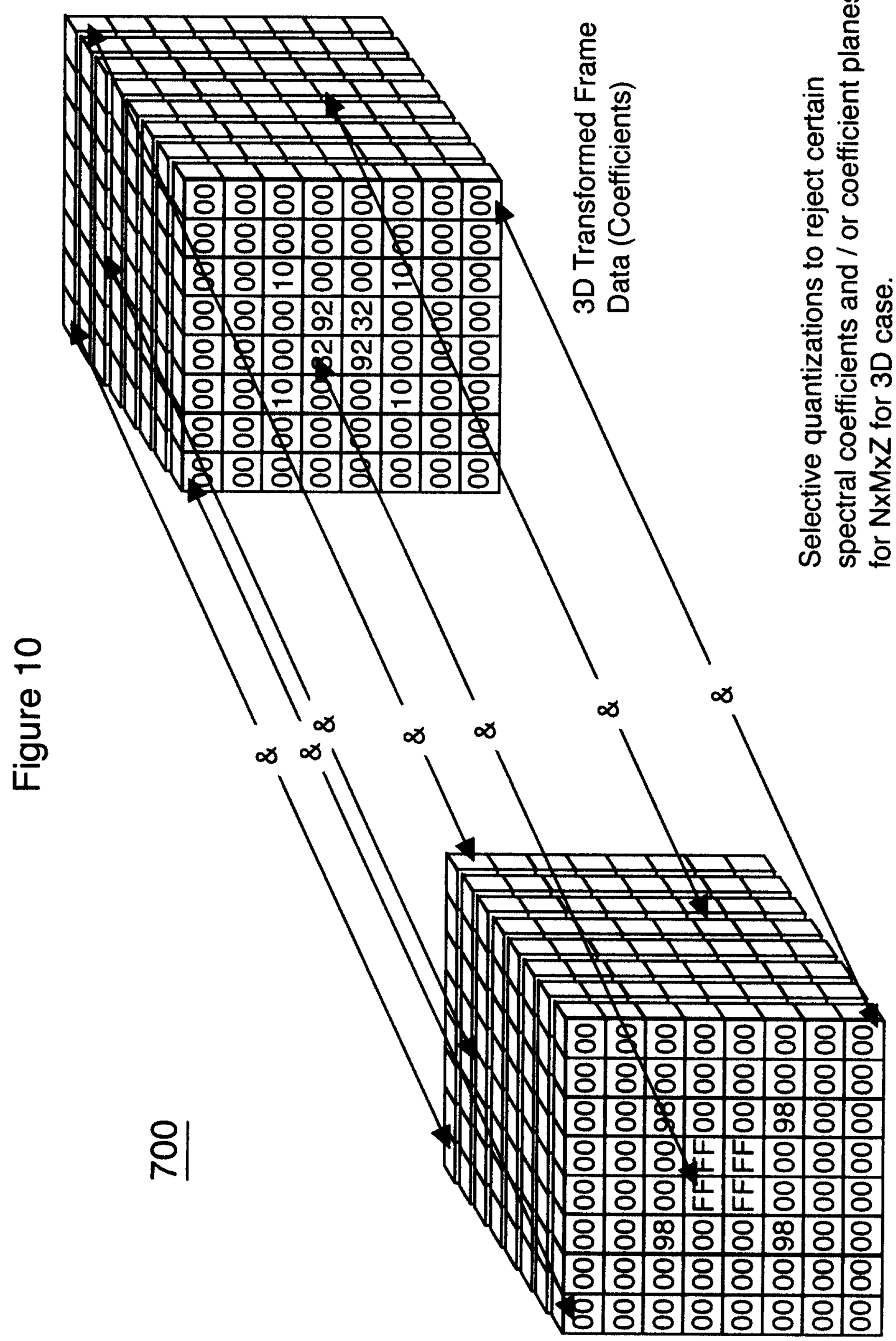
Quantization Mask

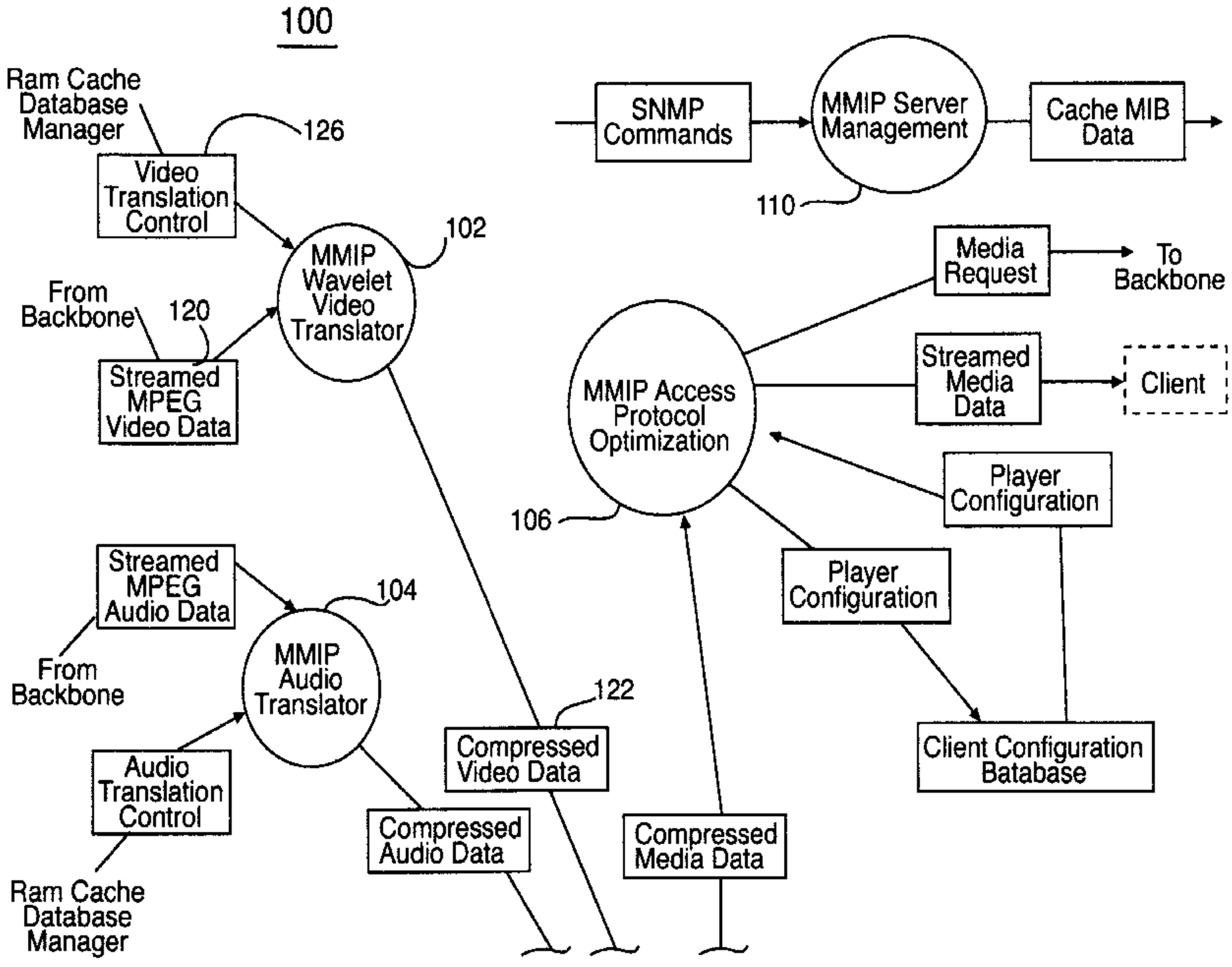
F4	F4	F4	F4	FF	FF	FF	FF
F4	F4	F4	F4	FF	FF	FF	FF
F4	F4	F4	F4	FF	FF	FF	FF
F4	F4	F4	F4	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF

604

Selective wavelet subband quantizations to reject certain subbands and / or portions of subbands for NxN and / or NxM, or NxMxZ for 3D case.

Figure 10


700
3D Quantization Mask
3D Transformed Frame Data (Coefficients)
Selective quantizations to reject certain spectral coefficients and / or coefficient planes for NxMxZ for 3D case.


100
Ram Cache Database Manager
126
Video Translation Control
102
MMIP Wavelet Video Translator
From Backbone
120
Streamed MPEG Video Data
SNMP Commands
MMIP Server Management
Cache MIB Data
110
Media Request
To Backbone
MMIP Access Protocol Optimization
Streamed Media Data
Client
106
Player Configuration
Player Configuration
Client Configuration Batabase
Streamed MPEG Audio Data
104
From Backbone
MMIP Audio Translator
122
Compressed Video Data
Audio Translation Control
Compressed Audio Data
Compressed Media Data
Ram Cache Database Manager