

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
14 December 2000 (14.12.2000)

PCT

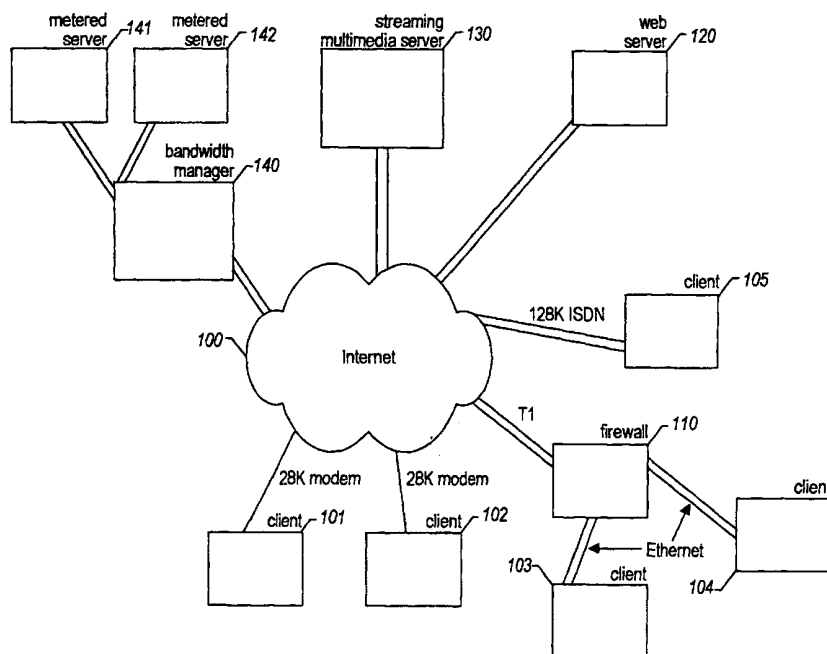
(10) International Publication Number
WO 00/76146 A1

- (51) International Patent Classification⁷: H04L 12/56, 29/06
- (74) Agents: LAWRENZ, Steven, D. et al.; Perkins Coie LLP, Suite 4800, 1201 Third Avenue, Seattle, WA 98101-3099 (US).
- (21) International Application Number: PCT/US00/14897
- (22) International Filing Date: 31 May 2000 (31.05.2000)
- (81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW.
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 09/329,984 9 June 1999 (09.06.1999) US
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- (71) Applicant: WORLDSTREAM COMMUNICATIONS, INC. [US/US]; 12320 8th Street, Bellevue, WA 98005 (US).
- (72) Inventors: FAIRMAN, Jeff; Apartment 207, 109 John Street, Seattle, WA 98109 (US). WILLIAMS, Ken; 8434 N. Mercer Way, Mercer Island, WA 98040 (US). JONES, Thomas; 3721 S. 180th Street A315, Seatac, WA 98188 (US). PALAZON, Eduard; Code Factory, S.L., Rbla. Francesco Macia, 116 1, E-08226 Terrassa, Barcelona (ES).

Published:
— With international search report.

[Continued on next page]

(54) Title: METERED CONTENT DELIVERY



(57) Abstract: The present invention is directed to a facility for managing the transmission of data from a server computer system to a client computer system. The facility obtains data destined for the client computer system, and accesses an indication of the maximum rate at which the transmit data to the client computer system. The facility then transmits the data destined for the client computer system at a rate no greater than the indicated maximum rate.



WO 00/76146 A1



— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

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

METERED CONTENT DELIVERY

TECHNICAL FIELD

The present invention is directed to the field of computer networks, and more particularly, to the field of delivering data using a computer
5 network.

BACKGROUND OF THE INVENTION

The term "multimedia" refers to various different presentation formats that have been developed for presenting data to users of computer systems. These include graphics, audio, and video. Because audio and video
10 are played, or "rendered," over a period of time, instances of these forms of multimedia are called "multimedia sequences." In order to support rendering over a period of time, multimedia sequences are often time-indexed.

While multimedia sequences may be physically delivered on removable media such as CD ROMs and DVDs, they may also be delivered via
15 a network such as the Internet. Where delivered via a network, the data making up a multimedia sequence is transmitted over the network from a server computer system to a client computer system. The data received in the client computer system is used by the client computer system to render the sequence for the benefit of one or more users.

20 Initially, multimedia sequences were delivered to the client computer system in their entirety before being rendered. This approach to delivering multimedia sequences, called "pre-rendering delivery," has been largely superseded by an approach called "streaming delivery." In streaming delivery, the data of the multimedia sequence is transmitted to the client in a
25 form that permits the client to begin rendering the sequence almost

immediately. While rendering proceeds, data for additional portions of the sequence, which are to be rendered after the portion of the sequence presently being rendered, is transmitted to the client.

Streaming delivery has several advantages over pre-rendering
5 delivery. First, for dense media such as video, pre-rendering delivery of even a sequence of moderate length can impose a wait time of many minutes before rendering can begin, while rendering can often begin within seconds using streaming delivery. Also, streaming delivery is much better suited to live
10 sequences, which typically do not have a fixed length, and whose data is often not all available when downloading commences. In a related vein, streaming delivery is also much better suited to live sequences in that it permits live sequences to be rendered in near real time, thus reinforcing their “up to the minute” nature. Further, since streaming delivery permits users to in essence preview a sequence, it enables users to quickly cancel the delivery of unwanted
15 sequences.

Streaming delivery has rigorous bandwidth requirements. While streaming delivery can be configured to use a larger data rate for clients having high-speed connections to their servers, modern streaming delivery systems for audio sequences generally rely on transmitting data at a rate of about 10-15
20 kilobits/second (kbps) to the client computer system. Because streaming delivery schemes commonly utilize protocols that provide delivery verification such as TCP, streaming may consume an even greater data rate where packets containing streaming data are lost during their initial delivery and must be retransmitted.

25 In order to effectively render a streaming multimedia sequence at the client computer system, it is necessary for adequate bandwidth to be consistently available from the server computer system to the client computer system to transmit the data representing the sequence at the required data rate. Unfortunately, because many client computer systems are connected to server
30 computer systems via the Internet, and connected to the Internet via a 28.8 kbps

modem, this leaves only about 10-15 kbps of additional bandwidth for other Internet applications. When other Internet applications, such as web browsing, file downloading, or email retrieval, consume more than this 10-15 kbps bandwidth “remainder” during streaming delivery, the required bandwidth for streaming delivery is no longer available, and the quality of the rendered multimedia sequence can suffer substantially. Indeed, rendering of the multimedia sequence may even be interrupted or terminated. While modern streaming delivery systems typically buffer an amount of data corresponding to a few seconds of the sequence, this buffering usually only enables rendering to continue uninterrupted during brief periods when significant portions of the bandwidth to the client computer system is consumed by other Internet applications.

In view of the rigorous bandwidth requirements of streaming delivery discussed above, and in view of the limited bandwidth available to many client computer systems on the Internet, a system for controlling the bandwidth used to transmit other types of data to the client computer system to reserve adequate bandwidth for the transmission of streaming media data would have significant utility.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a network diagram showing a typical network in which the facility is implemented.

Figure 2 is a high-level block diagram of a typical general-purpose computer system in the network in which portions of the facility operate.

Figure 3 is a high-level block diagram showing typical components of a client computer system in which portions of the facility operate, such as client computer systems 101-105 shown in Figure 1.

Figure 4 is a conceptual diagram showing, at a high level, the processing performed by the bandwidth manager.

Figure 5 is a flow diagram showing the steps preferably performed in a bandwidth manager by the facility in order to receive, packetize, and queue messages.

Figure 6 is a flow diagram showing the conceptual steps that are performed by the facility and the bandwidth manager to transmit messages to clients.

Figure 7 is a flow diagram showing the steps that are preferably actually performed by the facility in order to transmit packets to client computer systems.

Figure 8 shows the queue data structure at a time before the transmission of the packet.

Figure 9 is a data structure diagram showing the state of the queue data structure after a packet has been sent.

Figure 10 is a conceptual diagram showing, at a high level, the processing performed by the client computer system.

Figure 11 is a flow diagram showing the steps preferably performed by the portion of the facility that is in the client computer system in order to process received packets.

Figure 12 is a flow diagram showing the steps preferably performed by the facility to process an uncached message.

Figure 13 is a display diagram showing a user interface preferably displayed by the facility.

Figure 14 is a display diagram showing the user interface after the receipt of an uncached chat message.

Figure 15 is a flow diagram showing the steps preferably performed in the client computer system by the facility to process a message of the cache class.

Figure 16 is a flow diagram showing the steps preferably performed in the client computer system by the facility in order to process a trigger message.

Figure 17 is a display diagram showing a display of contents of the cached message in response to receiving a trigger message.

Figure 18 is a display diagram showing the display of a progress meter in accordance with step 1604 of Figure 16.

5 Figure 19 is a flow diagram showing the steps preferably performed by the facility in order to process a message deletion message.

Figure 20 is a flow diagram showing the steps preferably performed by the facility in order to process a message retention cache clearing message.

10 Figure 21 is a flow diagram showing the steps preferably performed by the facility in order to retrieve chat messages.

Figure 22 is a flow diagram showing the steps preferably performed in the HTTP server in response to an HTTP request from the client for packets for a particular component.

15 Figure 23 is a flow diagram showing the steps preferably performed by the facility in the second implementation of the client in components other than the chat component.

DETAILED DESCRIPTION

The present invention provides a facility for controlling the bandwidth used to transmit data to a recipient computer system (“the facility”).
20 In a preferred embodiment, the facility is implemented in a bandwidth manager computer system (“the bandwidth manager”), which sends data to the recipient computer system at a data rate generally not exceeding a maximum data rate.

The maximum data rate is preferably chosen for the recipient
25 computer system based on both the total data rate at which the recipient computer system can typically receive data, as well as the sorts and sources of data that the recipient computer system is expected to receive. For example, where a recipient computer system is connected to the Internet via a “28.8K” modem, the recipient computer system can generally receive data at a total rate

of about 20 kilobits per second (“kbps”), allowing for error correction and connections at speeds lower than the modem maximum speed. (Recipient computer systems connected to the Internet with higher-speed connections, such as “56.6K” modems, ISDN, xDSL, cable modems, or T1 connections, have higher total data rates.) If the recipient computer system is expected to receive data only from the bandwidth manager, then a maximum data rate of 20 kbps may be set for this recipient computer system. If, on the other hand, the recipient computer system is also expected to receive streaming audio from a streaming multimedia server at the rate of 10 kbps, then a maximum data rate of 10 kbps, or about 100 bytes every 100 ms, may be set for data transmitted from the bandwidth manager to the recipient computer system.

By limiting the rate at which the bandwidth manager transmits data to the recipient computer system to the maximum data rate established for the recipient computer system, the facility ensures that data sent to the recipient computer system through the bandwidth manager can be successfully received by the recipient computer system. Similarly, the facility ensures that data sent to the recipient computer system from computer systems other than the bandwidth manager that is accounted for in the maximum data rate established for the recipient computer system can be successfully received by the recipient computer system. Thus, the facility may be used to reserve bandwidth to the recipient computer system for multimedia streams, web browsing, electronic mail and other forms of messaging, and other network applications of any type.

In certain preferred embodiments of the invention, the bandwidth manager computer system receives messages from one or more message sources. Each received message is addressed to one or more recipient computer systems. The bandwidth manager preferably divides each message into one or more packets of a size based on contents of the message, such as a message type. This process is referred to as “packetizing” the message. The packets that are produced are ultimately sent to each recipient computer system to which the message is addressed, at a rate not exceeding the maximum data rate set for the

recipient computer system. Until these packets are transmitted to the recipient computer system, however, the bandwidth manager retains them in queues that it maintains for each recipient computer system.

The bandwidth manager preferably maintains a set of queues for
5 each recipient computer system to which it is presently configured to forward messages. The queues of a set each have a different priority level. After a message is received and packetized, for each addressee recipient computer system, the bandwidth manager places the produced packets in the queue for the addressee recipient computer system having the proper priority based on the
10 contents of the message, such as a message type contained in the message. When the bandwidth manager transmits packets to a recipient computer system, it first transmits packets stored in the queue for the recipient computer system having the highest priority, then transmits packets stored in the queue for the recipient computer system having the second-highest priority, etc.

15 When the transmitted packets are received in the recipient computer system, they are reassembled into whole messages and processed on the recipient computer system. For example, visual information contained in the messages may be displayed by the recipient computer system. In certain preferred embodiments, the recipient computer system stores some of the
20 received messages in a cache, so that a very short message can later be transmitted to the recipient computer system to quickly display the contents of the cached message. For instance, images whose messages can take several seconds to transmit to a client computer system may be pre-transmitted to the client computer system, then quickly displayed in response to such a "trigger"
25 command. Using the facility's message prioritization scheme, messages containing trigger commands and other small administrative messages may be quickly transmitted to the recipient computer system ahead of other earlier-pending messages.

In order to ensure that data is transmitted to each recipient
30 computer system at a data rate not exceeding the maximum data rate set for the

recipient computer system, the facility utilizes a calculated quantity called “minimum sleep time.” Once a packet has been transmitted to a recipient computer system, the minimum sleep time is the minimum length of time that bandwidth manager must wait before sending the next packet to the recipient
5 computer system to prevent the actual data rate to the recipient computer system from exceeding the maximum data rate for the recipient computer system. The facility calculates minimum sleep time based upon the maximum data rate for the recipient computer system and the size of either the last packet sent to the recipient computer system or the next packet to be sent to the recipient
10 computer system. In general, the facility transmits the next packet to the recipient computer system at or slightly after the minimum sleep time has elapsed.

In order to manage data relating to particular recipient computer systems in the bandwidth manager, the bandwidth manager preferably
15 instantiates a programmatic object for each “active” recipient computer system to which the bandwidth manager is configured to send data. This “client object” preferably contains all of the state information needed to transmit to the recipient computer system messages addressed to it at a rate no larger than the maximum data rate specified for the recipient computer system, such as
20 indications of all packets that need to be transmitted to the recipient computer system and their priorities and origination times; an indication of the last time a packet was sent to the recipient computer system; an indication of the next time a packet should be sent to the recipient computer system; an indication of the maximum data rate for the recipient computer system; and information needed
25 to send a packet to the recipient computer system, such as the network address of the recipient computer system. Each client object preferably also exposes a conditional send method. When the conditional send method is invoked, if the current time is later than the next packet transmission time, then the method transmits the next packet to the recipient computer system. Otherwise, the
30 conditional send method returns the next packet transmission time, so that the

conditional send method can be invoked again at or shortly after the next packet transmission time to transmit the next packet.

In accordance with an additional embodiment of the present invention, the facility is implemented in an HTTP server in order to operate
5 with a recipient computer system whose interaction with the Internet is limited by a firewall security device that prevents direct interaction with the bandwidth manager. In this embodiment, the recipient computer system periodically sends an HTTP request to the HTTP server. In response to the HTTP request, the HTTP server contacts the bandwidth manager, transfers any packets pending for
10 the recipient computer system in the queues of the bandwidth manager to the HTTP server, and sends some or all of these pending packets to the recipient computer system in an HTTP reply sent at a rate not exceeding the maximum data rate for the recipient computer system. In a further preferred embodiment, the HTTP server sends only some of the pending packets to the recipient
15 computer system, but includes an indication that the recipient computer system should send additional HTTP requests to obtain the remaining pending packets.

In order to fully describe the invention, a more detailed description of the implementation of various embodiments and aspects of the invention follows.

20 Figure 1 is a network diagram showing a typical network in which the facility is implemented. The network shows a number of client computer systems, or "recipient computer systems," 101-105 that are all connected to the Internet 100. Clients 101 and 102 are connected to the Internet 100 via 28.8K modems. Clients 103 and 104 are connected to a security firewall 110 via
25 Ethernet connections, and from there to the Internet 100 via a T1 connection. The client 105 is connected to the Internet 100 via an ISDN connection. The clients 101-105 can receive data via the Internet 100 from servers, such as a web server, or "HTTP server," 120; a streaming multimedia server 130 for providing multimedia sequences using streaming delivery; and metered servers,
30 such as metered servers 141 and 142, which provide information to the clients

through a bandwidth manager 140. The bandwidth manager 140 executes portions of the facility in order to limit the rate at which data is sent from the metered servers to each client to a data rate not larger than a maximum data rate specified for the client.

5 Figure 2 is a high-level block diagram of a typical general-purpose computer system in the network in which portions of the facility operate. The computer system 200 contains a central processing unit (CPU) 210, input/output devices 220, and a computer memory (memory) 230. Among the input/output devices is a network connection 221, through which the
10 computer system 200 may communicate with other connected computer systems, a storage device 222, such as a hard disk drive; and a computer-readable media drive 223, which can be used to install software products, including the facility, which are provided on a computer-readable medium, such as a CD-ROM. The memory 230 preferably contains computer programs and
15 data. In computer systems that act as servers, the memory 230 and/or the storage device 222 preferably contain both data that is served to client computer systems as well as server software such as an HTTP server and/or a streaming multimedia server. While the facility is preferably implemented on computer systems configured as described above, those skilled in the art will recognize
20 that it may also be implemented on computer systems having different configurations.

 Figure 3 is a high-level block diagram showing typical components of a client computer system in which portions of the facility operate, such as client computer systems 101-105 shown in Figure 1. In
25 addition to the components shown in Figure 2 and discussed above, the client computer system 300 preferably includes the following additional input/output devices: a display device 324, such as a video monitor, for displaying visual information; a keyboard 325 for inputting text; a pointing device 326, such as a mouse, for selecting positions within information displayed on the display
30 device; and an audio output device 327, such as speakers, for outputting audio

information. Among the programs 331 stored in the memory 330 is preferably a web browser program that can issue HTTP requests to web servers and display the contents of the resulting HTTP responses. Many such browsers also support JavaScript, an HTTP scripting language. Some browsers further support Java, a largely hardware-independent language. While computer systems such as the one shown is preferably used as client computer systems, those skilled in the art will recognize that client computer systems having different configurations may also be used.

Figure 4 is a conceptual diagram showing, at a high level, the processing performed by the bandwidth manager. At the highest level, the bandwidth manager 410 transforms messages 401 addressed to particular recipient computer systems into message packets that are transmitted to the addressee recipient computer systems at a data rate no greater than the maximum data rate specified for each of the addressee recipient computer systems. As part of this process, the bandwidth manager in step 411 first divides each message into smaller packets. This process is referred to as packetization. In step 412, the bandwidth manager queues the packets making up the message in queues for each addressee. Finally, in step 413, the bandwidth manager dequeues and sends the queued message packets at a rate no greater than the maximum data rate for each addressee recipient computer system. These steps are discussed below in greater detail in conjunction with Figures 5-9.

Figure 5 is a flow diagram showing the steps preferably performed in a bandwidth manager by the facility in order to receive, packetize, and queue messages. In step 501, the facility receives a message. The message may either directly contain content data, or may contain a reference to content data available from the server. Each message also contains a list of addressees, recipient computer systems, or "clients" to which the message is to be transmitted. In step 502, the facility packetizes the message by dividing the message into one or more packets that are each no larger than a target packet

size. The target packet size is preferably specified with respect to certain contents of the message, such as a message type indication contained by the message. In steps 503-507, the facility loops through each client specified to receive the message. In step 504, the facility places the packets of the message on a queue for the current client. The client preferably has several different queues, each having a different priority. The facility preferably selects the appropriate queue by discerning the priority of the message based on the contents of the message, such as an indication of the type of the message contained in the message. These packets may be placed on the queue either directly or by reference. In step 505, if the queue to which the message packets were added, as well as any higher-priority queues, were emptied before the addition of the packets, then the facility continues in step 506, else the facility continues in step 507. In step 506, the facility recalculates the next time at which a packet is to be sent to the client based upon the size of the first new packet. After step 506, the facility continues as step 507. In step 507, the facility loops back to step 503 to process the next addressee client. After step 507, the facility continues in step 501 to process the next received message.

The steps shown in Figure 5 may be performed by one or more threads in the bandwidth manager. In particular, steps 501-502 may be performed in a first thread, and steps 503-507 may be performed in a second thread. Further, as is discussed in greater detail below, steps 505 and 506 relate to an embodiment in which the time at which the next packet is sent is based on the size of the next packet to be sent. In an alternative embodiment in which the time at which the next packet is sent is placed upon the size of the last packet sent, steps 505 and 506 are unnecessary and are omitted.

Figure 6 is a flow diagram showing the conceptual steps that are performed by the facility and the bandwidth manager to transmit messages to clients. Steps 601-605 are repeated each time a next send time is reached for a particular client. In step 602, the facility sends the highest-priority packet contained in the queues for the client. In step 603, the facility updates the last

send time for the client to the current time. In step 604, the facility recalculates the next send time for the client based upon the current time, and either the size of the new highest-priority packet in the queues for the client or on the size of the packet sent in step 602. In step 605, the facility loops back to step 601 to process the next client next send time that is reached.

The processing of step 604 is central to ensuring that data is sent to each client computer system at a rate not exceeding its maximum data rate. The next send time is calculated by adding to the last send time an amount of time equal to the size of a packet addressed to the client computer system divided by the maximum rate for the client computer system. In a first embodiment, the packet whose size is used for this calculation is the packet that was last sent. In a second embodiment of the invention, the packet whose size is used in this calculation is the next packet to be sent. Because the second embodiment relies on an identification of the next packet to be sent, and because new packets that are added to the queues for a client that are of a higher priority than the other queued packets for the client can change the identity of the next packet to be sent, in the second embodiment, the facility preferably recalculates the next send time each time a new packet is added to the queues for a client that becomes the highest-priority packet. Consider the following example comparing the two embodiments mentioned above. In the example, the current client has a maximum data rate of 10 kbps. The facility has just sent a 1,000-bit packet to the client, and will next send a 500-bit packet to the client. In accordance with a first embodiment, the facility adds to the current time a minimum of sleep time of the size of the packet just sent, 1,000 bits, divided by the maximum data rate of 10 Kbps to arrive at a minimum sleep time of .1 seconds. In accordance with a second embodiment, the facility determines the next sent time by adding to the current time a minimum sleep time equal to the size of the next packet to be sent, 500 bits, divided by the maximum data rate for the client of 10 kbps, or .05 seconds.

Figure 7 is a flow diagram showing the steps that are preferably actually performed by the facility in order to transmit packets to client computer systems. In step 701, the facility initializes an earlier next send time variable to a time that is far in the future. In steps 702-707, the facility loops through each client in the active list -- that is, each client whose queues contain outgoing packets. In step 703, if the next send time for the client is later than the current time, then the facility continues in step 706, else the facility continues in step 704. In step 704, the facility sends the highest-priority packet queued for the client. This is the packet having the highest priority value that was least recently received in the bandwidth manager. The facility preferably uses information stored for the client, such as the client's network address, to send this packet to the client. As part of step 704, the facility removes the sent packet from the queues for the client. In step 705, the facility calculates the next send time for the client as discussed above in conjunction with step 604. After step 705, the facility continues in step 706. In step 706, the facility sets the value of the earliest next send time variable to the earlier of (1) the current value of the earliest next send time variable and (2) the next send time for the current client. In step 707, the facility loops back to step 702 to process the next client in the active list. After step 707, the earliest next send time variable contains the earliest next send time variable contains the earliest time at which any client is scheduled to send its next packet. In step 708, the facility sleeps until this earliest next send time. After step 708, the facility wakes and continues in step 701 to repeat the cycle.

In the second embodiment discussed above in which the size of the next packet to be sent is used to calculate minimum sleep time for a client, the sleep of step 708 may be interrupted to send a packet to a client to whose queues a new packet has been added as the new highest priority packet if the new packet is smaller than the former highest priority packet.

In certain embodiments, each active client is represented by a programmatic object. Such a "client object" contains data relating to the state

of the bandwidth manager's efforts on behalf of the client, including the contents of its queues, its last sent time, and its next send time. The client object preferably further implements a conditional send method that, when invoked, provides the functionality of steps 703-705. That is, the conditional send method determines the next send time for the client has been reached, and, if so, sends a packet to the client and recalculates the next send time for the client. If the next send time has not been reached, the conditional send method preferably returns an indication of the next send time, which can be used to schedule a future invocation of the conditional send method.

Figures 8 and 9 are data structure diagrams showing the state preferably maintained by the facility for each active client. Figure 8 shows the queue data structure at a time before the transmission of the packet. It can be seen that the queue data structure 800 contains information for each of a number of active clients. For active client 1, the queue data structure 800 contains five queues: queue 810 for priority 1 packets, queue 815 for priority 2 packets, queue 820 for priority 3 packets, queue 825 for priority 4 packets, and queue 830 for priority 5 packets. The priority 1 packets in queue 810, packets 811, 812, and 813, are of the highest priority (most urgent), while packet 826 in queue 825 is of the lowest priority (least urgent). For each active client, the queue data structure 802 further includes an indication 805 of the last time at which packet was sent to active client 1 and an indication 806 of the next time at which a packet is to be sent to active client 1. The state of the queue data structure in Figure 8 corresponds to a current time of 2:06:12.010.

Figure 9 is a data structure diagram showing the state of the queue data structure after a packet has been sent. Figure 9 corresponds to a current time of 2:06:12.020. It can be seen that, while the priority 1 queue 910 still contains packets 912 and 913, the priority 1 queue 910 no longer contains packet 811 shown in Figure 8. This packet was sent at time 2:06:12.018. After being sent, it was removed from priority 1 queue 910. Further, the last send time 905 and next send time 906 for active client 1 have been updated. The last

send time 905 now indicates the time at which packet 811 was sent, the next send time 906 indicates the time at which the new highest-priority packet 912 will be sent.

The packets pending for a particular client are described above in conjunction with Figures 8 and 9 as being represented in a number of queues, each having a different priority levels. In this description, the next packet selected for transmission to the client is always the packet least recently added to the highest-priority nonempty queue. In a further embodiment of the invention, the facility instead represents the pending packets for a client in a binary tree structure. Each packet is preferably represented as a node in the tree. Nodes representing new packets are added to the tree, and nodes representing sent packets are selected and removed from the tree, based on a two-component ordinality in which packet priority level is the high-order component and packet reference time is the low-order component. Using such a tree to represent pending packets can be both (1) more efficient than fixed queues, since no space is reserved for priorities at which no packets are sent, and (2) more flexible, as the priorities assigned to packets need not be within a predetermined set of possible priorities.

Figure 10 is a conceptual diagram showing, at a high level, the processing performed by the client computer system. The client computer system 1010 receives metered message packets 1001 from the bandwidth manager. In step 1002, the portion of the facility on the client computer system reassembles messages made up by the received metered message packets. In step 1003, based on the kind, or "class" of the message, the facility either continues in step 1004 if the message is of an uncached class, continues in step 1005 if the message is of a cached class, or continues in step 1006 if the message is of a trigger class. For messages of the uncached class, the contents of the message are immediately displayed in step 1004. Uncached messages are those whose content is to be displayed immediately upon delivery, such as chat messages. For messages of the cached class, the facility stores the message in

the cache in step 1005. Cached messages are those whose contents are not to be displayed immediately upon receipt, but are rather stored in the cache until a trigger message is received specifying the display of the contents of the cached message. Finally, if the message is of a trigger class, the trigger message specifies a cached message that is to be displayed in step 1006 in response to receiving the trigger message.

Figure 11 is a flow diagram showing the steps preferably performed by the portion of the facility that is in the client computer system in order to process received packets. In step 1101, the facility receives a packet from the bandwidth manager computer system. In step 1102, if the received packet is the first packet of a new message, then the facility continues in step 1103, else the facility continues in step 1104. In step 1103, the facility adds an entry to an incoming message store for the message. The incoming message store is designed to manage the packets of any messages that have not yet been received in their entirety. In a preferred embodiment, the incoming message store is implemented using a hash table that maps from a message identifier stored in each packet to the packets that have been received containing this message identifier. In one embodiment, the facility further preferably maintains a mapping to the message identifier for received messages from a second identifier for the message used in trigger messages and kill messages. The second identifier is preferably invariant across all the client computer systems, while the message identifier of a particular message may vary across the different client computer systems to which it is transmitted, since the bandwidth manager preferably assigns message identifiers serially for each client. After step 1103, the facility continues in step 1104.

In step 1104, the facility adds the received packet to the entry in the incoming message store for the message. In step 1105, if the incoming message store now contains the complete message, then the facility continues in step 1106, else the facility continues in step 1101 to receive the next incoming packet. In step 1106, the facility removes the complete message from the

incoming message store and processes the message based upon a message type indication contained in the message. The processing of step 1106 is discussed in greater detail below in conjunction with Figures 12, 15, and 16.

A first type, or “class” of message that is processed by the facility is uncached messages. As mentioned above, uncached messages are those whose contents are immediately displayed upon receipt of a message. Figure 12 is a flow diagram showing the steps preferably performed by the facility to process an uncached message. In step 1201, the facility displays the contents of the message. The steps then conclude.

Figures 13 and 14 show the processing of an uncached message. Figure 13 is a display diagram showing a user interface preferably displayed by the facility. In the user interface, the client window 1300 contains several visual components, each of which can be designated to receive messages addressed to the client computer system. Components 1321, 1322, and 1323 are all image components that can display images received in messages. Component 1310 is a chat window that can display chat lines 1311-1314 contained on chat messages received by the client computer system. The user may also type a new chat line into field 1331, and send it to other chat participants by pressing button 1332.

Figure 14 is a display diagram showing the user interface after the receipt of an uncached chat message. It can be seen that, while Figure 14 is similar to Figure 13, an additional chat line 1415 has been added to chat lines 1311-1314 shown in Figure 13. The facility preferably does so immediately in response to receiving the last packet of a message containing chat line 1415.

Figure 15 is a flow diagram showing the steps preferably performed in the client computer system by the facility to process a message of the cached class. Such messages include those that contain images. In steps 1501, the facility stores the message in a message retention cache. The message retention cache is preferably implemented using a hash table that maps from the

secondary message identifier of the message to the contents of the message. After step 1501, these steps conclude.

Figure 16 is a flow diagram showing the steps preferably performed in the client computer system by the facility in order to process a trigger message. Each trigger message preferably includes the secondary identifier of the cached message whose contents are to be displayed. In step 1601, if the message identified by the secondary identifier contained in the trigger message is stored in the message retention cache, then the facility continues in step 1606, else the facility continues in step 1602. In step 1602, if a portion of the identified message has been received and is stored in the incoming message store, then the facility continues in step 1604, else the facility continues in step 1603. In step 1603, the facility waits for the receipt of the first packet of the identified message. This waiting step preferably involves periodically checking the incoming message store for the arrival of the first packet at an interval such as .5 seconds. After step 1603, the facility continues in step 1604. In step 1604, until the last packet of the identified message is received, the facility displays a progress meter indicating the partial extent to which the identified message has been received. This progress meter is preferably displayed at the location in the user interface at which the contents of the identified message will eventually be displayed. In step 1605, once the last packet of the identified message is received in the client computer system, the facility moves the identified message to the message retention cache. In step 1606, the facility displays the identified message contained in the message retention cache. After step 1606, these steps conclude. Note that the identified message is preferably not deleted from the message retention cache at this time, permitting the identified message to be redisplayed again in the future when another trigger message is received identifying this cached message identifying this cached message.

Figures 17 and 18 show the processing of a cached message. Figure 17 is a display diagram showing the display of contents of the cached

message in response to receiving a trigger message. By comparing Figure 17 to Figure 14, it can be seen that, in response to receiving a message containing image 1724, then receiving a trigger command identifying image 1724 stored in the message retention cache, the facility has displayed image 1724 in place of
5 image 1422.

Figure 18 is a display diagram showing the display of a progress meter in accordance with step 1604 of Figure 16. By comparing Figure 18 to Figure 17, it can be seen that the display of image 1724 has been replaced with a progress meter 1825 in response to receiving a trigger message for a cached
10 message that has not yet been received in its entirety. As additional packets for this message are received, the facility updates the progress meter to indicate that more of the image has been downloaded. As soon as the last packet of the message is received, the facility preferably replaces the display of the progress meter 1825 with the display of the contents of the new image.

15 Figures 19 and 20 describe the processing of additional types of messages used to maintain the contents of the message retention cache. Figure 19 is a flow diagram showing the steps preferably performed by the facility in order to process a message deletion message. A message deletion message is preferably sent to remove a cached message from the message
20 retention cache that is no longer needed. Such messages identify a single cached message to be deleted from the message retention cache, preferably by the secondary identifier of the message. In step 2001, the facility deletes the cached message identified by the message deletion message from the message retention cache. After step 1901, these steps conclude.

25 Figure 20 is a flow diagram showing the steps preferably performed by the facility in order to process a message retention cache clearing message. A message retention cache clearing message is preferably sent in order to remove all of the messages in the cache to make space for new cached messages. In step 2001, the facility deletes from the message retention cache all
30 of the cached messages contained there. After step 2001, these steps conclude.

The approach described above for receiving and processing packets in the client computer system assumes both that (1) Java, or another fairly sophisticated Internet message-processing computer language may be executed on the client computer system, and (2) that the described metered message packets are able to travel unimpeded from the bandwidth manager computer system to the client computer system. These assumptions are not true in the case of every potential client computer system. As for assumption (1), many web browser programs presently in use do not provide for the execution of programs in Java or a similar language. Further, some client computer systems, such as client computer systems 103 and 104 shown in Figure 1, are connected to the Internet via a security firewall that intercepts certain types of packets transmitted to the client computer systems. In order to deliver messages to client computer systems having any of the above limitations, the facility preferably provides a second client implementation for use in these systems. The second client implementation utilizes simple HTTP requests generated by JavaScript scripts supported by most web browser programs presently in use. The second client implementation preferably uses a HTTP server or "web server," such as web server 120 shown in Figure 1 as an intermediary between the bandwidth manager and the client computer system. Because most security firewalls are configured to permit the flow of HTTP traffic between the clients and Internet web servers, it is generally possible to send HTTP requests from the client computer system to the web server, and to send HTTP responses from the web server to the client computer system.

The second client implementation preferably constitutes a compound HTML document that is loaded into the browser of the client computer system. The compound HTML document contains several frames each corresponding to one of the components 1310, 1321, 1322, and 1323 shown in Figure 13. Each frame in turn is controlled by a JavaScript script that issues HTTP requests to the web server to retrieve the contents for the frame.

In the second client implementation, designed to support chat lines and images, the messages containing retrieval of all information from the bandwidth manager is based on the retrieval of chat lines, which are generally sent to the client more frequently than new images. Figure 21 is a flow diagram showing the steps preferably performed by the facility in order to retrieve chat messages. These steps are preferably implemented in the JavaScript script executing in the frame for the chat component. The facility preferably repeats steps 2101-2109 every five seconds, or at another suitable interval for retrieving chat messages. In step 2102, the facility sends to the HTTP server an HTTP request for client packets containing the client identifier of the client and the component identifier of the chat component. In step 2103, the facility receives from the HTTP server any chat packets addressed to the client computer system. In step 2104, the facility displays the received chat packets in the chat component, as is shown in Figure 14. In step 2105, the facility further receives from the HTTP server the component identifiers of any components for which HTTP server also has packets for the client computer system. In steps 2106-2108, the facility loops through each received component identifier. In step 2107, the facility sends a message from the chat component and frame to the components and frames identified by the current component identifier. In step 2108, the facility loops back to step 2106 to process the next received component identifier. In step 2109, after five seconds has elapsed, the facility continues in step 2101 to check for additional chat messages at the HTTP server. While the chat component preferably also contains functionality for sending chat messages generated by the user of the client computer system, these steps are not shown here.

Figure 22 is a flow diagram showing the steps preferably performed in the HTTP server in response to an HTTP request from the client for packets for a particular component. These steps are preferably implemented in the HTTP server computer system using a CGI script executing on the HTTP server. These steps receive as parameters a client identifier for the client

computer system that sends a request, and a component identifier indicating the component whose packets are to be retrieved. In step 2201, if the component identifier is the component identifier of the chat component, then the facility continues in step 2202, else the facility continues to step 2206. In step 2202, the facility retrieves all pending packets from the binding manager's queues for the client having the client identifier. The retrieved packets may include packets for the chat component, packets for the image components, or packets for other components. In step 2203, the facility caches any packets retrieved in step 2202 that specify components other than the chat component. In step 2204, the facility sends the packets retrieved in step 2202 that are chat packets to the client at or below the maximum data rate for the client. Step 2204 is preferably formed in the manner described above in conjunction with Figures 6 and 7. In step 2205, the facility sends to the client additional HTTP responses containing the component identifiers specified for the retrieved packets that were cached in step 2203. After step 2205, these steps conclude. In step 2206, where the request contains a component identifier other than the component identifier for the check component, the facility sends any cache packets specifying this component identifier to the client at or below the maximum data rate for the client. The facility preferably deletes the cache packets from the cache at this point. After step 2206, these steps conclude.

Figure 23 is a flow diagram showing the steps preferably performed by the facility in the second implementation of the client in components other than the chat component. These steps are preferably performed by a JavaScript script controlling each of the frames containing the non-chat components. In step 2301, the facility receives a message from the chat component indicating that the HTTP server has cached packets specifying this component. In step 2302, the facility sends an HTTP request to the HTTP server for packets containing the client identifier of the client computer system and the component identifier of this component. In step 2303, the facility receives from a HTTP server an HTTP response containing the component

packets. In step 2304, the facility displays the received component packets in the frame. After step 2304, the facility continues in step 2301 to receive the next message in the chat component.

While the present invention has been shown and described with
5 reference to preferred embodiments, it will be understood by those skilled in the art that various changes or modifications in form and detail may be made without departing from the scope of the invention. For example, the functionality provided by the bandwidth manager may be distributed over multiple computer systems of various types. In particular, multiple bandwidth
10 managers may be used for the same or different sets of metered servers. Where multiple bandwidth manager computer systems are used, they may be allocated in accordance with the geographic distribution of metered servers, the geographic distribution of clients, or in accordance with other factors. Also, while the preferred embodiment of the invention is described in conjunction
15 with messages containing chat and image data, the facility may be used to deliver information of all types. Further, the portion of the facility that operates on the client computer system may be straightforwardly adapted to a variety of different types of client computer systems and browser computer programs executing thereon.

CLAIMS

We claim:

1 1. A method in a computer system for limiting the rate at
2 which a client computer system receives data from one or more source
3 computer systems to a maximum rate, comprising:

4 receiving bodies of data each directed to the client computer
5 system and each from a source computer system;

6 adding the contents of each received body of data to one of a
7 plurality of queues based upon contents of the body of data, each queue having
8 a level of priority relative to the other queues; and

9 periodically transferring a quantum of data to the client computer
10 system from the highest-priority queue containing data, each transfer occurring
11 at a time late enough to limit the rate at which the client computer system
12 receives data transferred from the queues based on the size of the transferred
13 quantum of data.

1 2. The method of claim 1, further comprising, for each
2 received body of data:

3 determining a type associated with the body of data; and

4 selecting one of the plurality of queues to add the contents of the
5 received body of data to based upon the determined type.

1 3. The method of claim 1, further comprising, for each
2 received body of data:

3 determining a type associated with the body of data; and

4 dividing the received body of data into quanta using a target
5 quantum size associated with the determined type,

6 and wherein one quantum of data is transferred at a time.

1 4. The method of claim 1 wherein the queues are together
2 represented in an image tree, and wherein the contents of each received body of
3 data is added to the binary tree.

1 5. The method of claim 1, further comprising, in the client
2 computer system:

3 receiving one or more first quanta of data comprising a body of
4 data containing information for presentation on the client computer system;

5 receiving, separate from the first quanta of data, a second
6 quantum of data specifying the presentation of the information contained in the
7 first quanta of data; and

8 in response to receiving the second quantum of data, presenting
9 the information contained in the first quanta of data.

1 6. The method of claim 5 wherein the first quanta of data
2 comprises a multimedia artifact, and wherein the multimedia artifact is rendered
3 in response to receiving the second quantum of data.

1 7. The method of claim 5, further comprising:

2 receiving a third quantum of data specifying the deletion of the
3 body of data comprised by the first quantum of data; and

4 in response to the receipt of the third quantum of data, deleting
5 the body of data comprised by the first quantum of data.

1 8. A computer-readable medium whose contents cause a
2 server computer system to managing the transmission of data from the server
3 computer system to a client computer system by:

4 obtaining data destined for the client computer system;

5 accessing an indication of the maximum rate at which to transmit
6 data to the client computer system; and
7 transmitting the data destined for the client computer system at a
8 rate no greater than the indicated maximum rate.

1 9. The computer-readable medium of claim 8 wherein data
2 destined for the client computer system is obtained from a plurality of sources.

1 10. The computer-readable medium of claim 8 wherein a total
2 maximum data rate and a protected data rate smaller than the total maximum
3 data rate are both specified for the client computer system, and wherein the
4 contents of the computer-readable medium further cause the server computer
5 system to:

6 store the indication of the maximum rate at which to transmit data
7 to the client computer system, the indication indicating that the maximum rate
8 at which to transmit data to the client computer system is a rate no larger than
9 the difference of the total maximum data rate and the protected data rate,
10 such that additional data may be received by the client computer
11 system at least the protected data rate at the same time that the client computer
12 system is receiving data from the server computer system at up to the indicated
13 rate.

1 11. The computer-readable medium of claim 10 wherein the
2 protected data rate is a rate associated with a type of streaming media,
3 such that an instance of streaming media of the type of streaming
4 media may be received by the client computer system at least the rate associated
5 with the type of streaming media at the same time that the client computer
6 system is receiving data from the server computer system at up to the indicated
7 rate.

1 12. The computer-readable medium of claim 8 wherein the
2 contents of the computer-readable medium further cause the computer system to
3 generate the indication of the maximum rate at which to transmit data to the
4 client computer system, by:

5 for each of a plurality of distinct data rates,
6 attempting to transmit data to the recipient computer
7 system at the current data rate;

8 determining whether the attempt to transmit data to the
9 recipient computer system at the current data rate produced satisfactory results;
10 and

11 determining the maximum rate based upon the highest data
12 rate at which it is determined that the attempt to transmit data to the recipient
13 computer system produced satisfactory results.

1 13. The computer-readable medium of claim 12 wherein the
2 step of determining the maximum rate selects as the maximum rate the highest
3 data rate at which it is determined that the attempt to transmit data to the
4 recipient computer system produced satisfactory results.

1 14. The computer-readable medium of claim 12 wherein the
2 step of determining whether the attempt to transmit data to the recipient
3 computer system at the current data rate produced satisfactory results
4 determines whether the attempt to transmit data to the recipient computer
5 system at the current data rate incurred any transmission errors.

1 15. The computer-readable medium of claim 12 wherein the
2 step of determining whether the attempt to transmit data to the recipient
3 computer system at the current data rate produced satisfactory results
4 determines whether the attempt to transmit data to the recipient computer

5 system at the current data rate incurred less than a threshold number of
6 transmission errors.

1 16. The computer-readable medium of claim 12 wherein the
2 plurality of data rates is selected in accordance with a binary search of a range
3 of acceptable data rates.

1 17. The computer-readable medium of claim 12 wherein the
2 plurality of data rates is selected by:

3 beginning with an initial data rate; and

4 selecting each subsequent data rate by choosing a data rate lower
5 than the current data rate if it is determined that the attempt to transmit data to
6 the recipient computer system at the current data rate did not produce
7 satisfactory results, and choosing a data rate higher than the current data rate if
8 it is determined that the attempt to transmit data to the recipient computer
9 system at the current data rate produced satisfactory results.

1 18. A computer memory containing a programmatic object
2 representing a data recipient computer system, comprising:

3 one or more dynamic data members specifying units of data to be
4 transmitted to the data recipient computer system; and

5 one or more dynamic data members specifying a next
6 transmission time at which one of the units of data may be transmitted to the
7 data recipient computer system without exceeding a maximum rate at which
8 data is to be transmitted to the data recipient computer system.

1 19. The computer memory of claim 18 wherein the
2 programmatic object further comprises:

3 a method that, when invoked, transfers a unit of data specified by
4 data members of the object to the data recipient computer system if the time of
5 invocation is not earlier than the next transmission time.

1 20. The computer memory of claim 19 wherein the method
2 returns the next transmission time if the time of invocation is earlier than the
3 next transmission time.

1 21. The computer memory of claim 19 wherein the method
2 also deletes the transferred unit of data and updates the next transmission time
3 system if the time of invocation is not earlier than the next transmission time.

1 22. The computer memory of claim 19 wherein the data
2 members specifying units of data to be transmitted include a binary tree
3 indicating an order in which the units of data are to be transmitted.

1 23. The computer memory of claim 19 wherein the binary tree
2 contains a node for each unit of data to be transmitted, and wherein the binary
3 tree is organized in accordance with both a priority level of each unit of data
4 and a reference line indication of each unit of data.

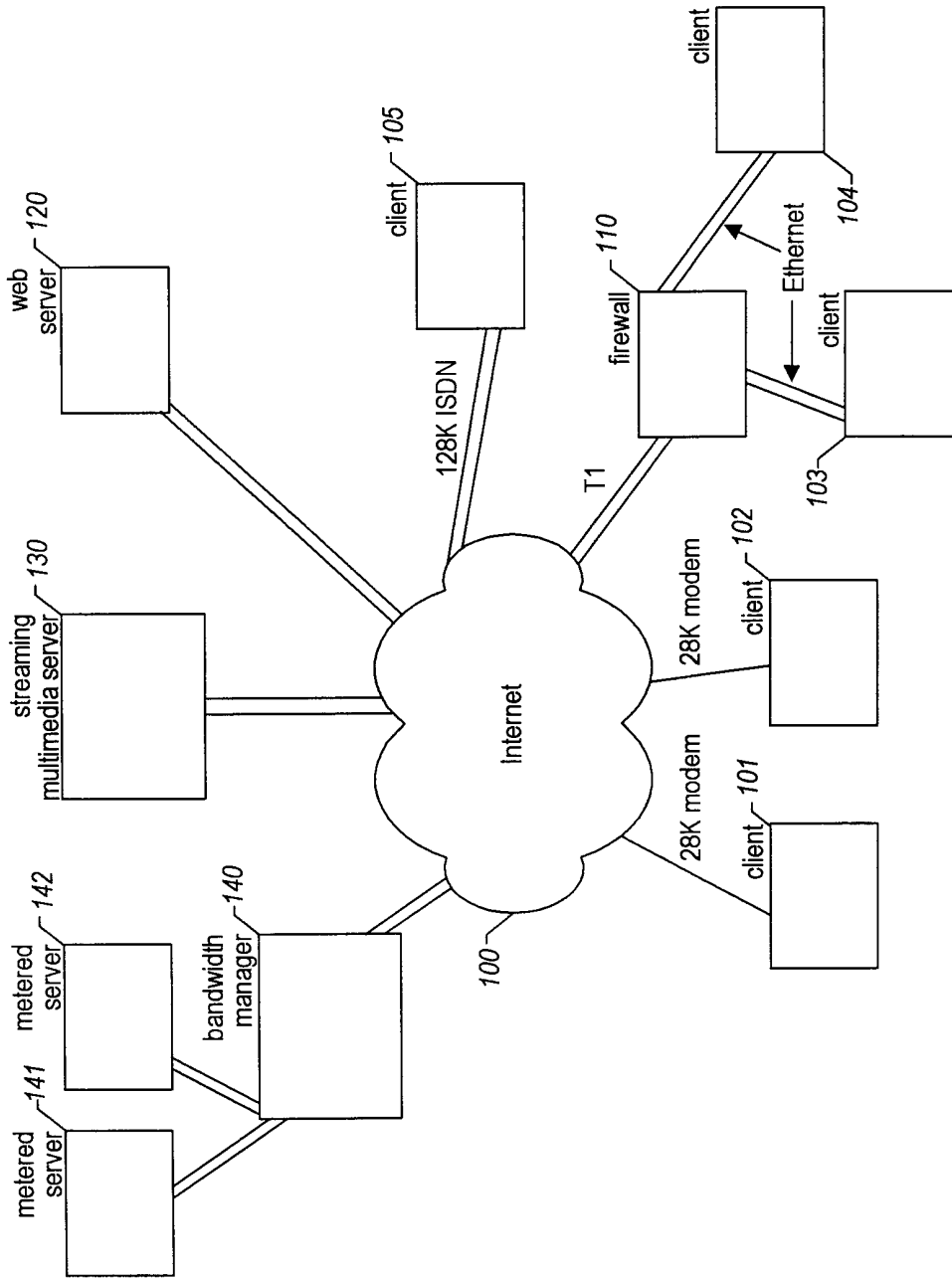


Fig. 1

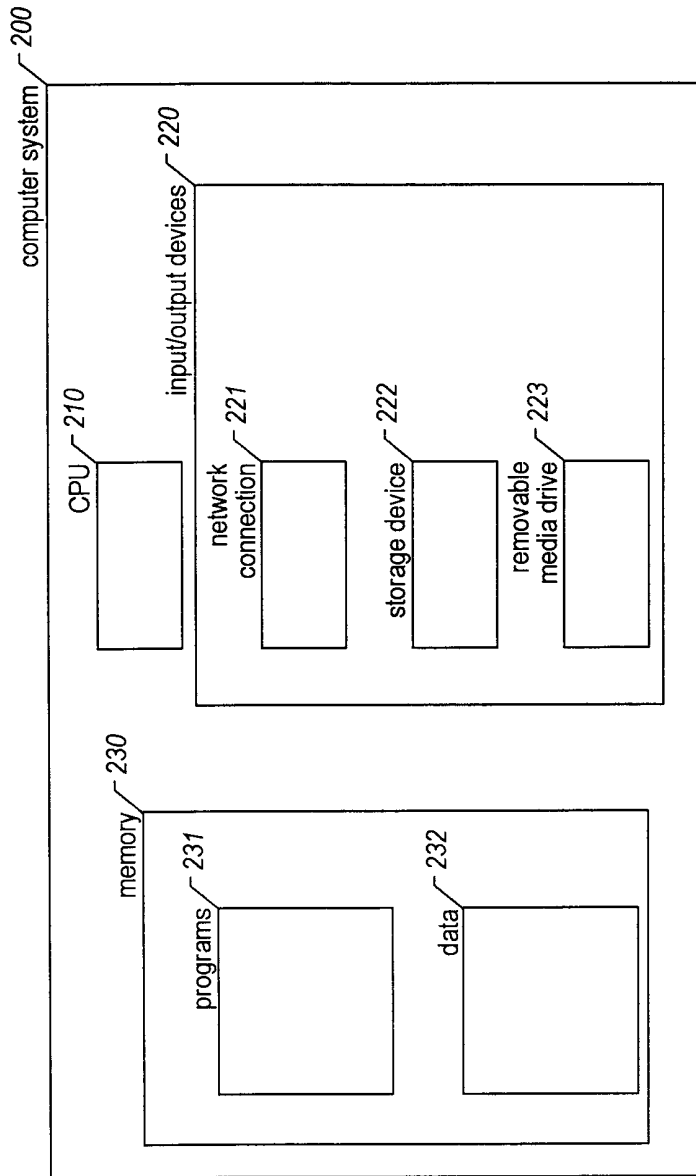


Fig. 2

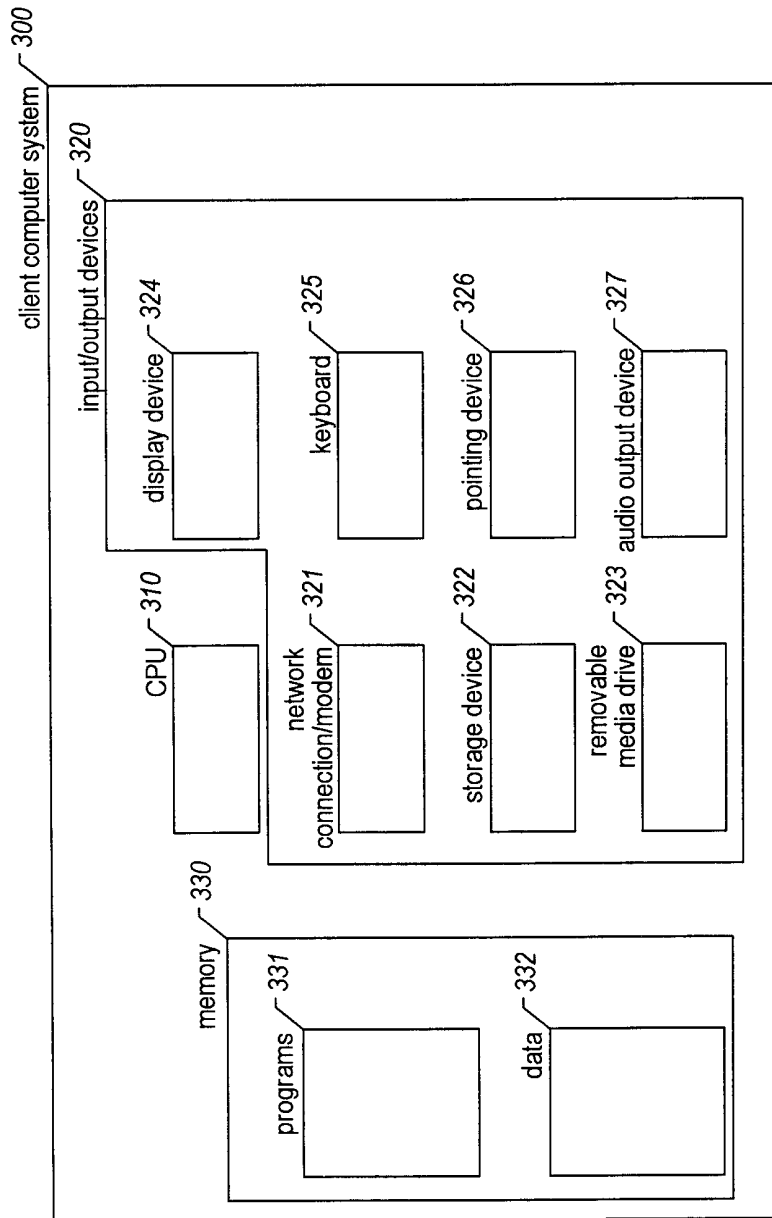


Fig. 3

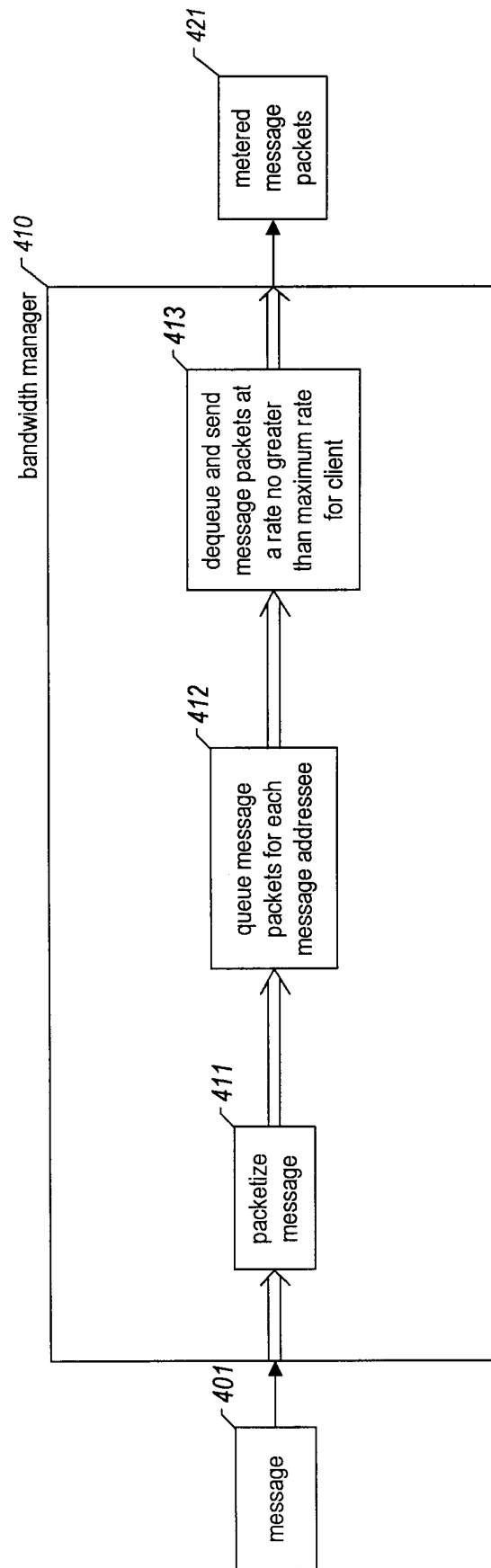


Fig. 4

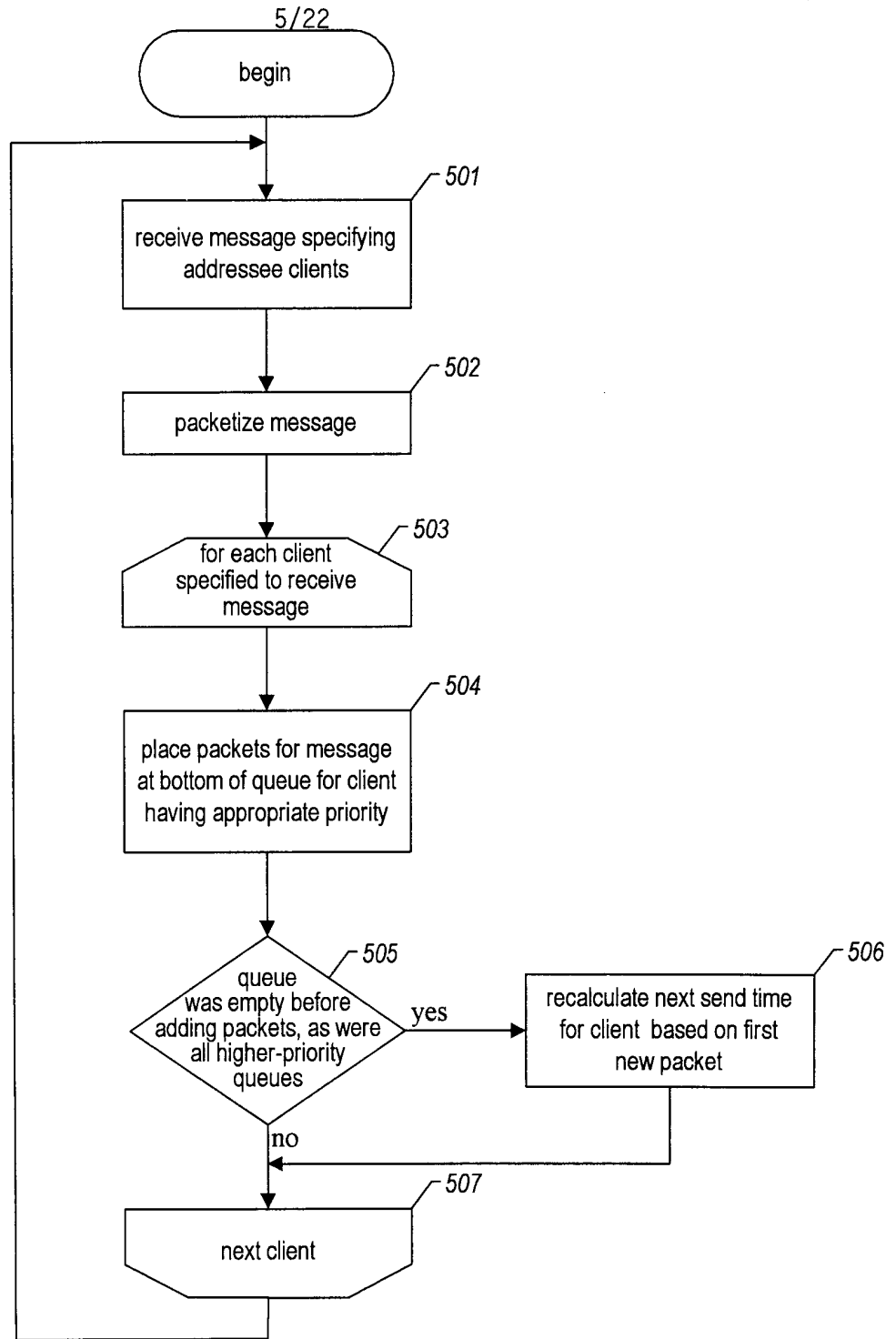


Fig. 5

6/22

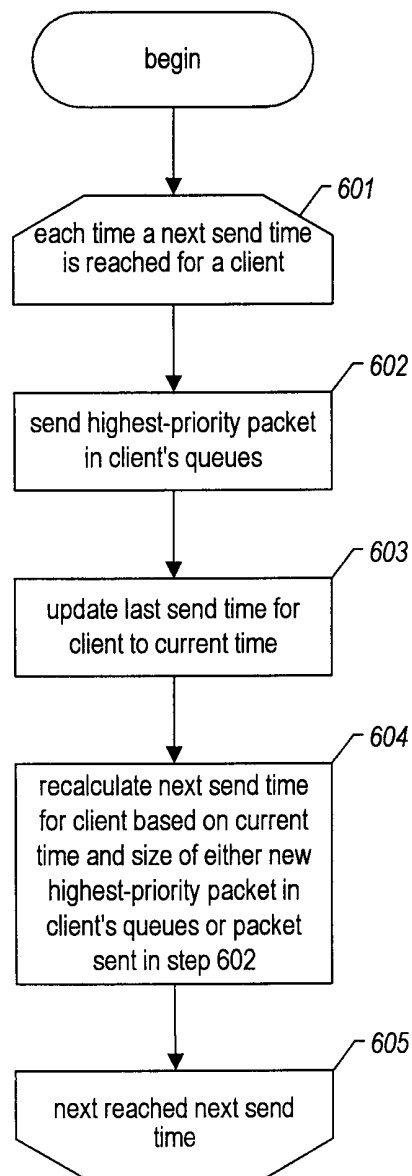


Fig. 6

7/22

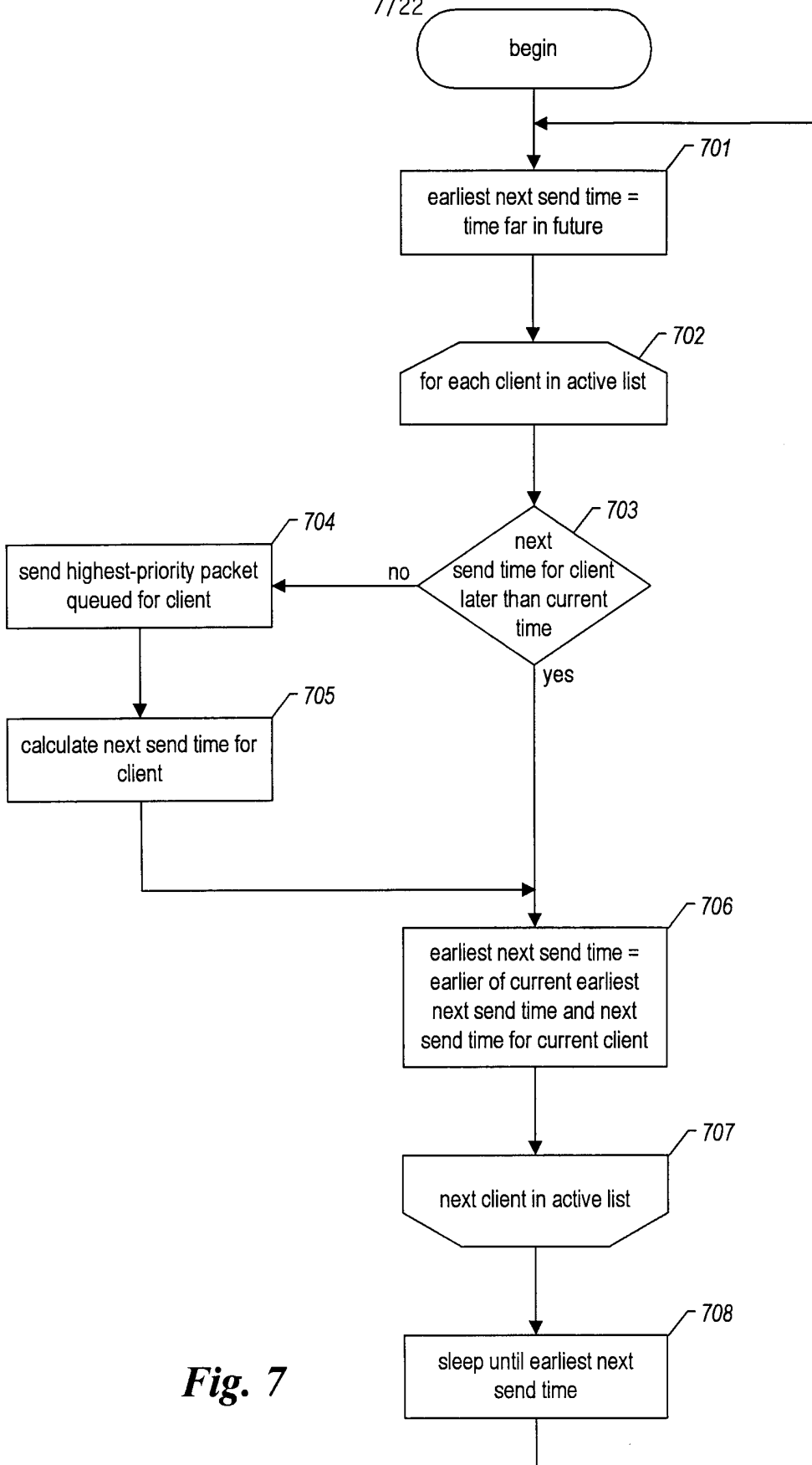


Fig. 7

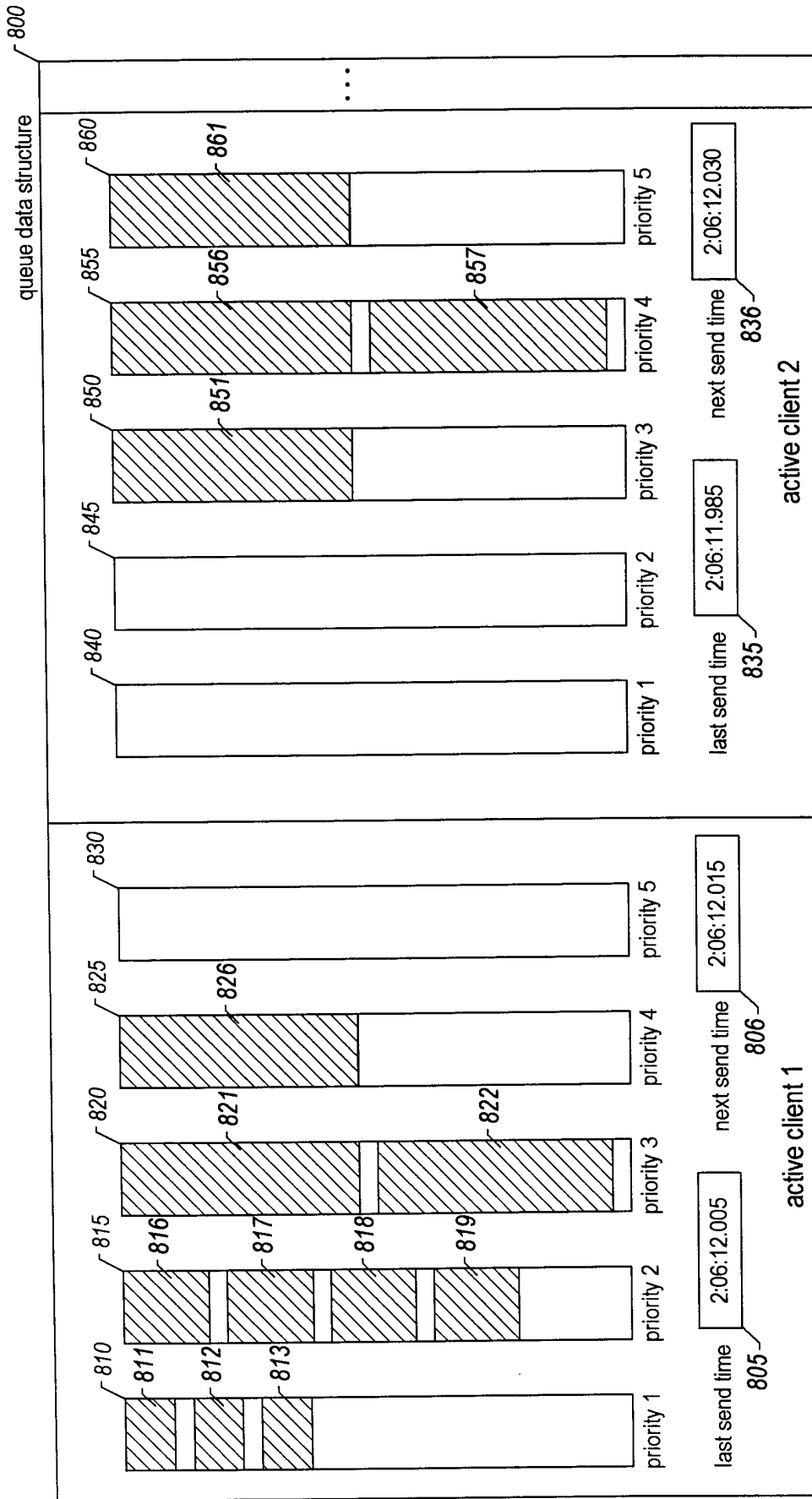


Fig. 8

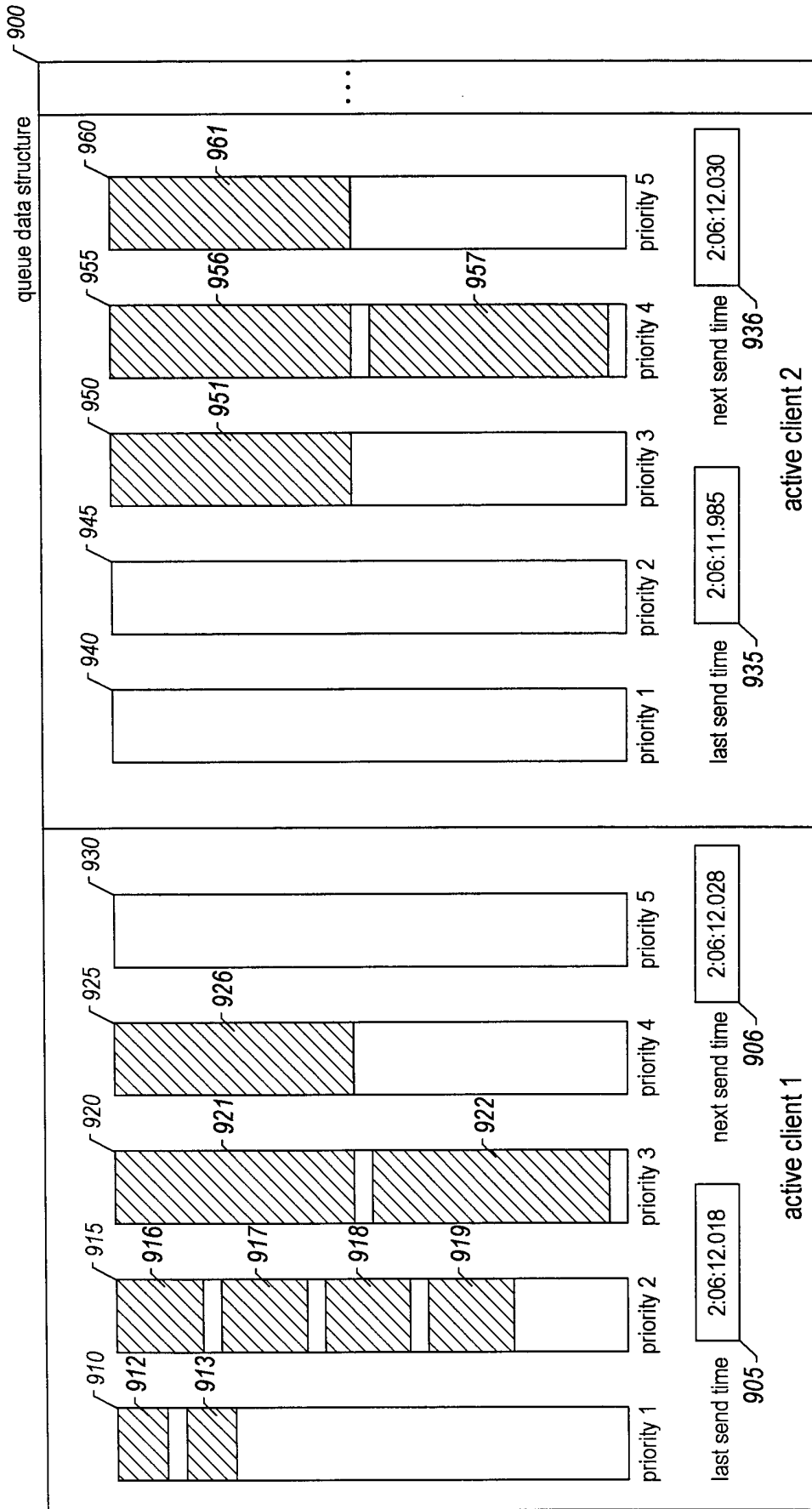


Fig. 9

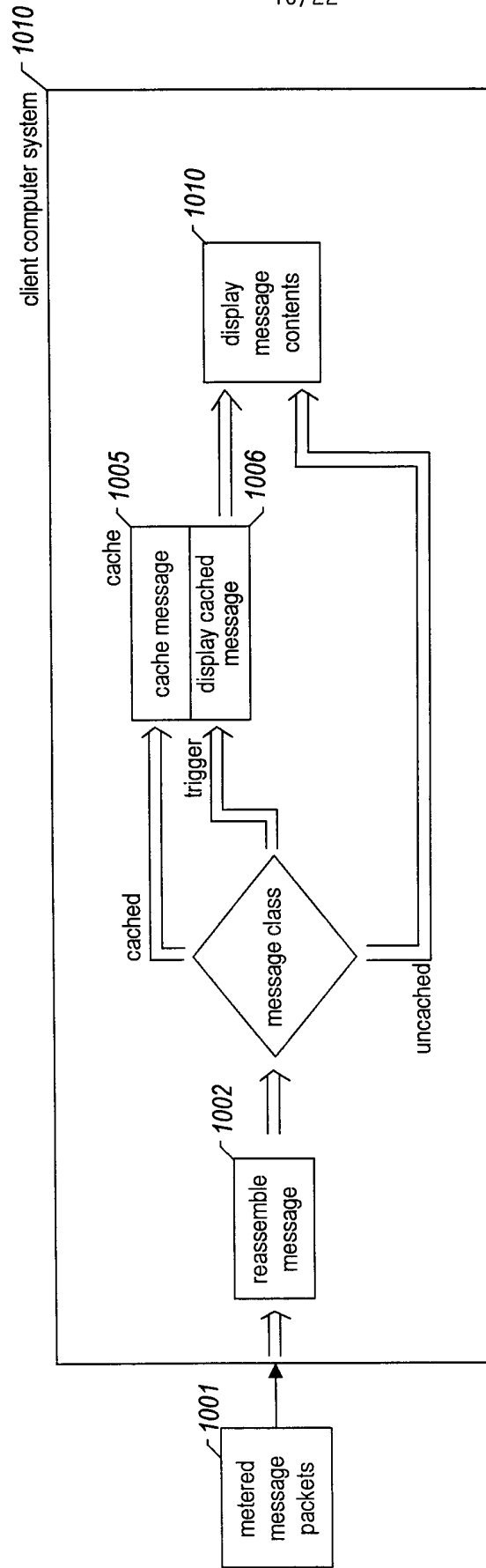


Fig. 10

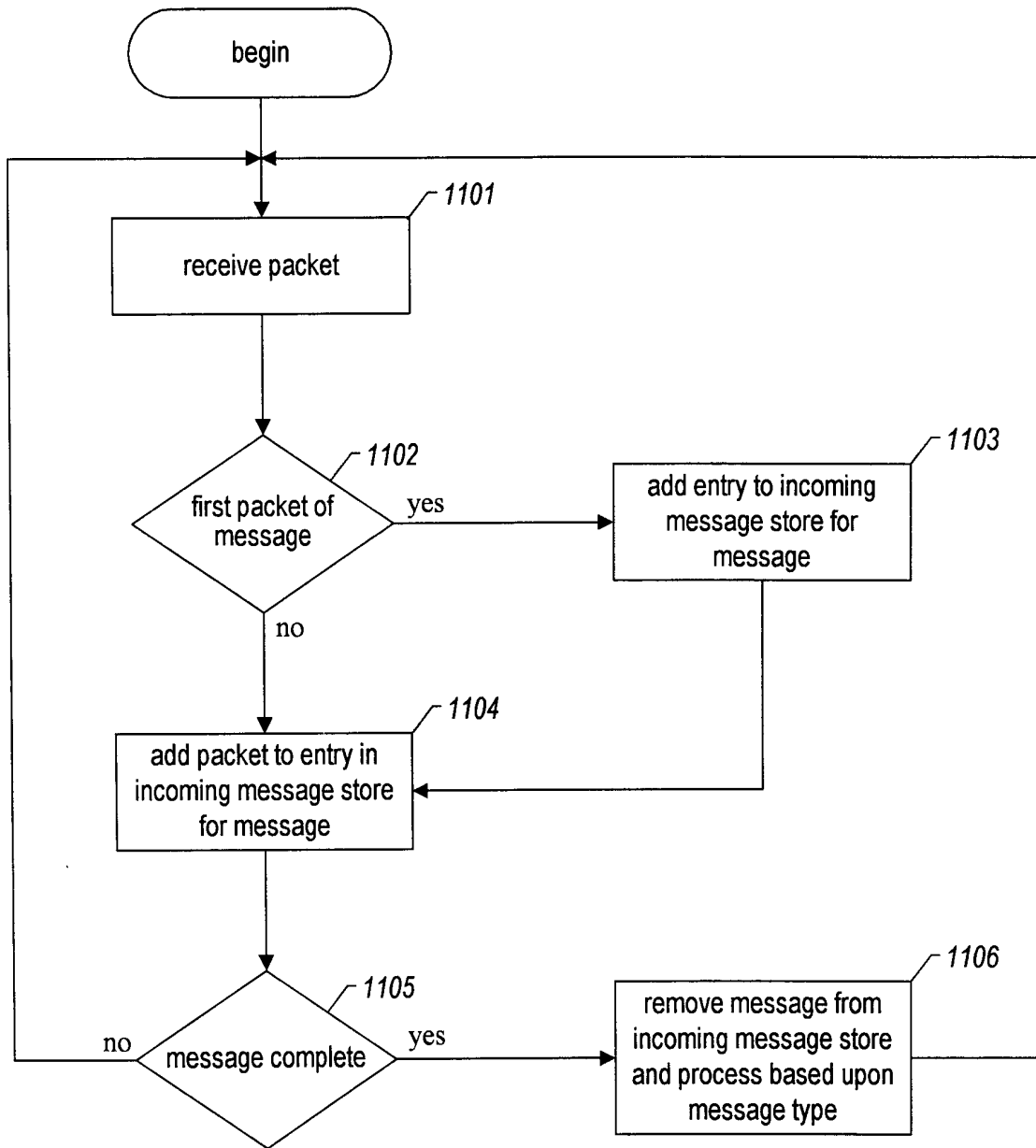


Fig. 11

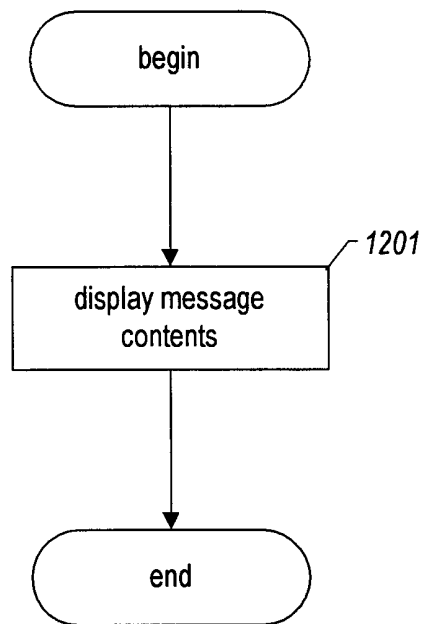


Fig. 12

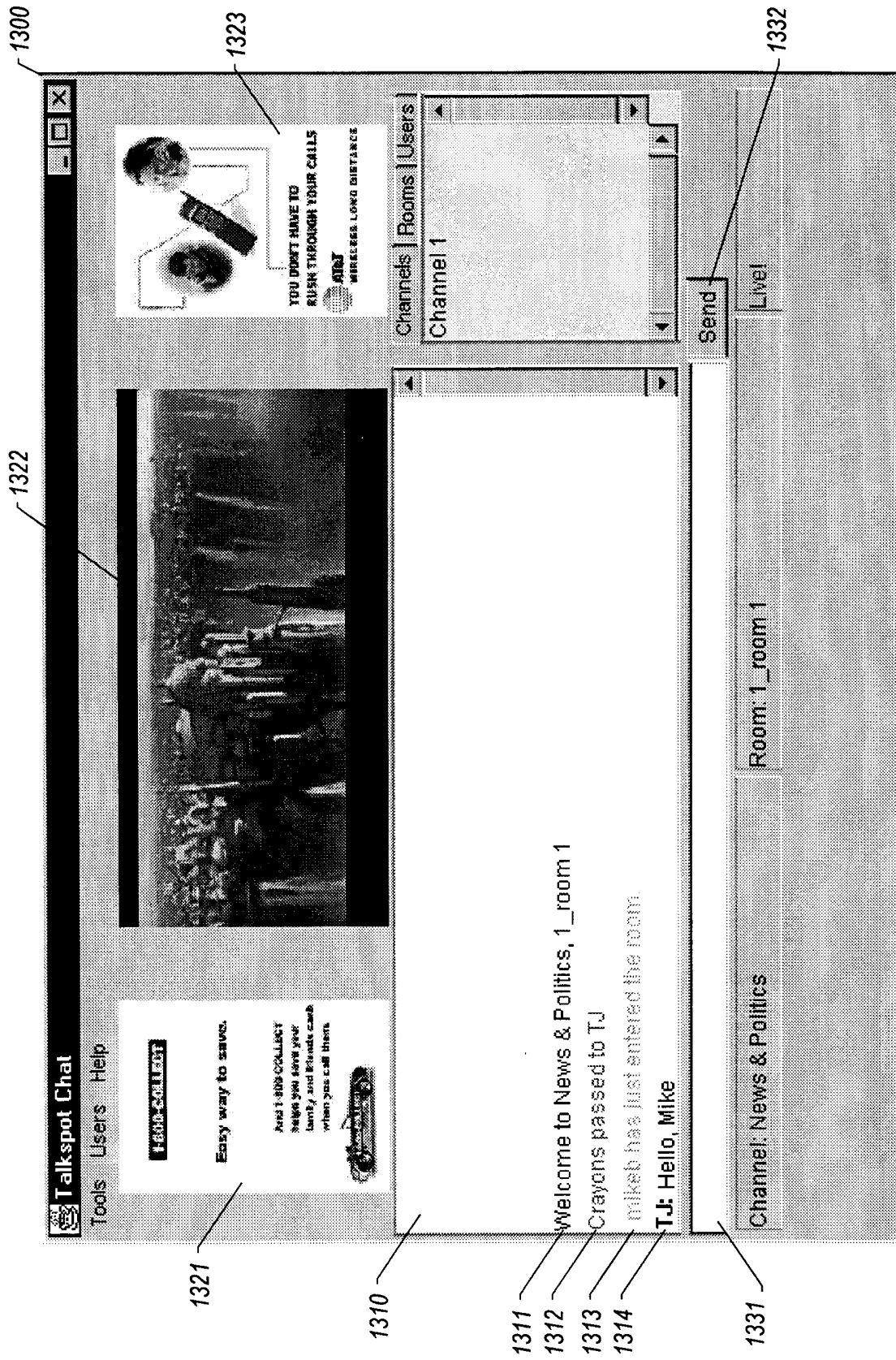
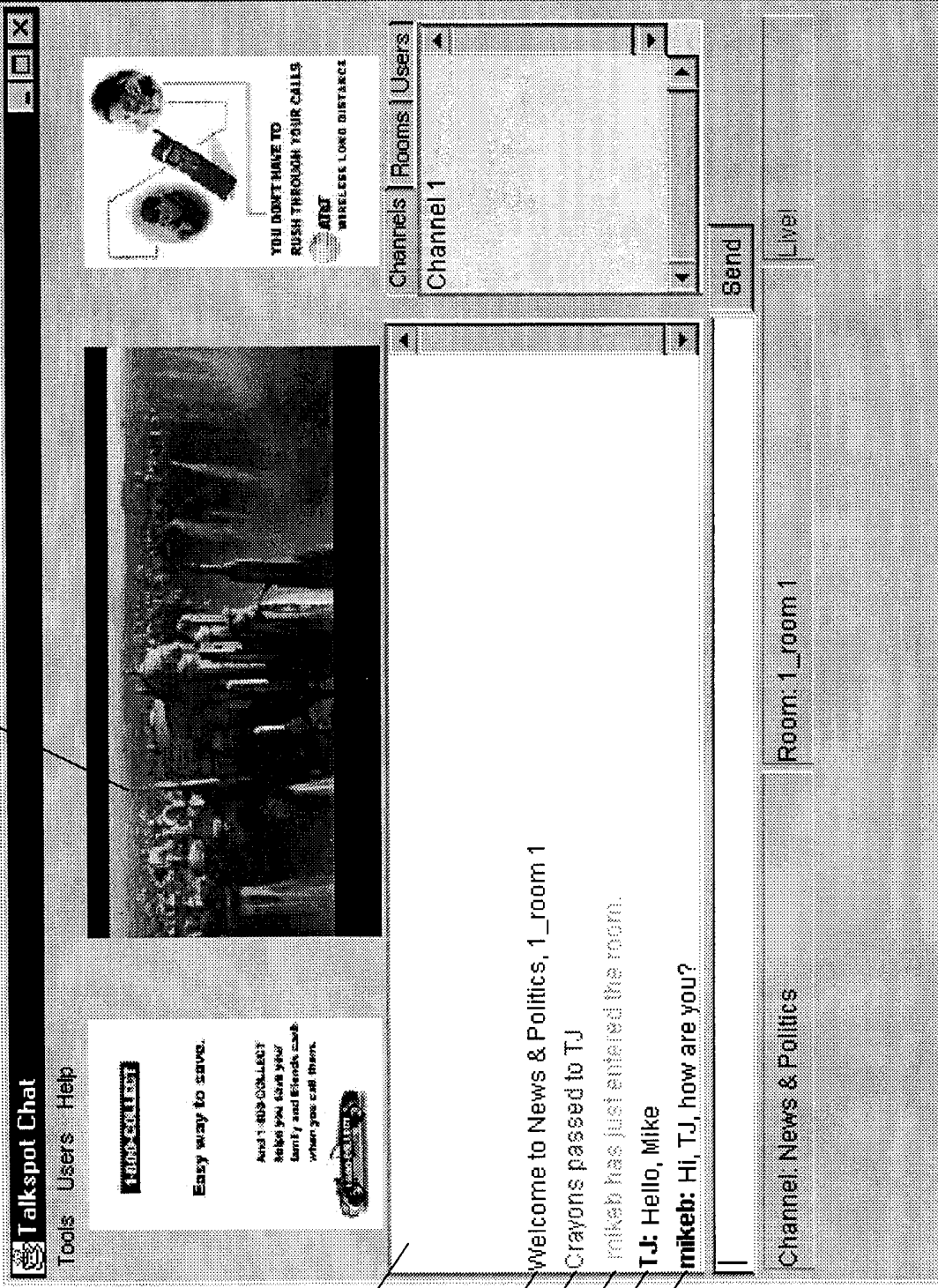


Fig. 13

1400

1422



1410

- 1411
- 1412
- 1413
- 1414
- 1415

Fig. 14

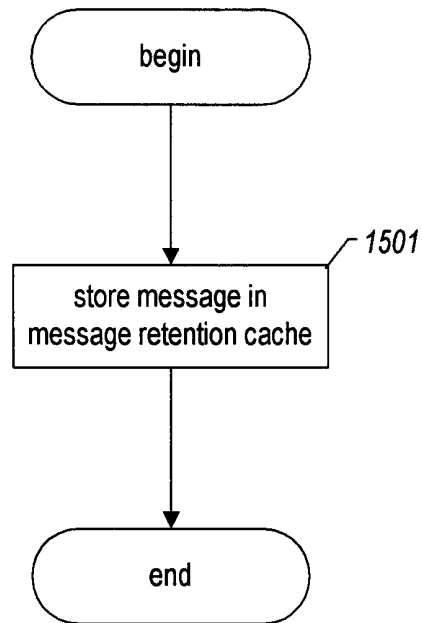


Fig. 15

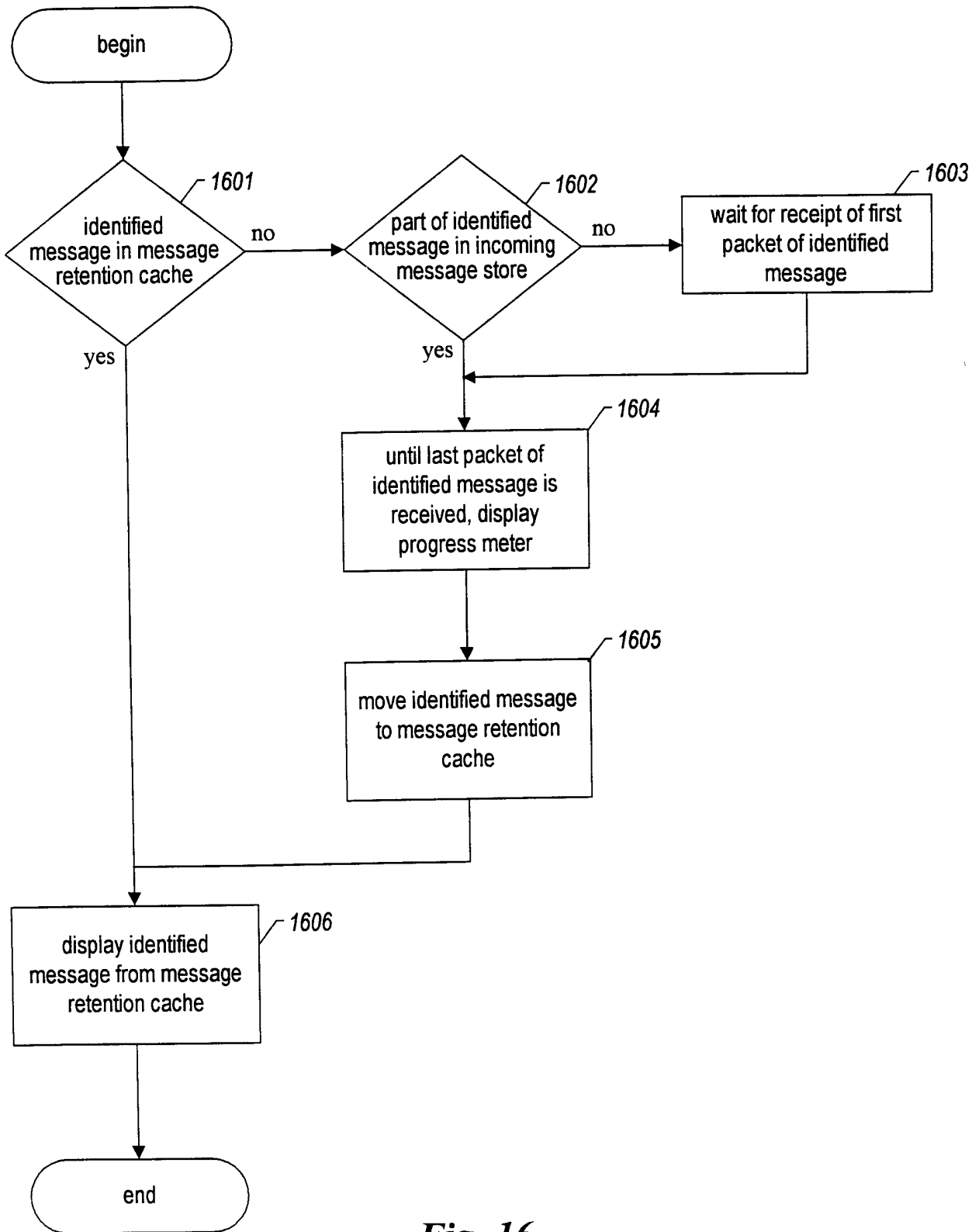


Fig. 16

1700

1724

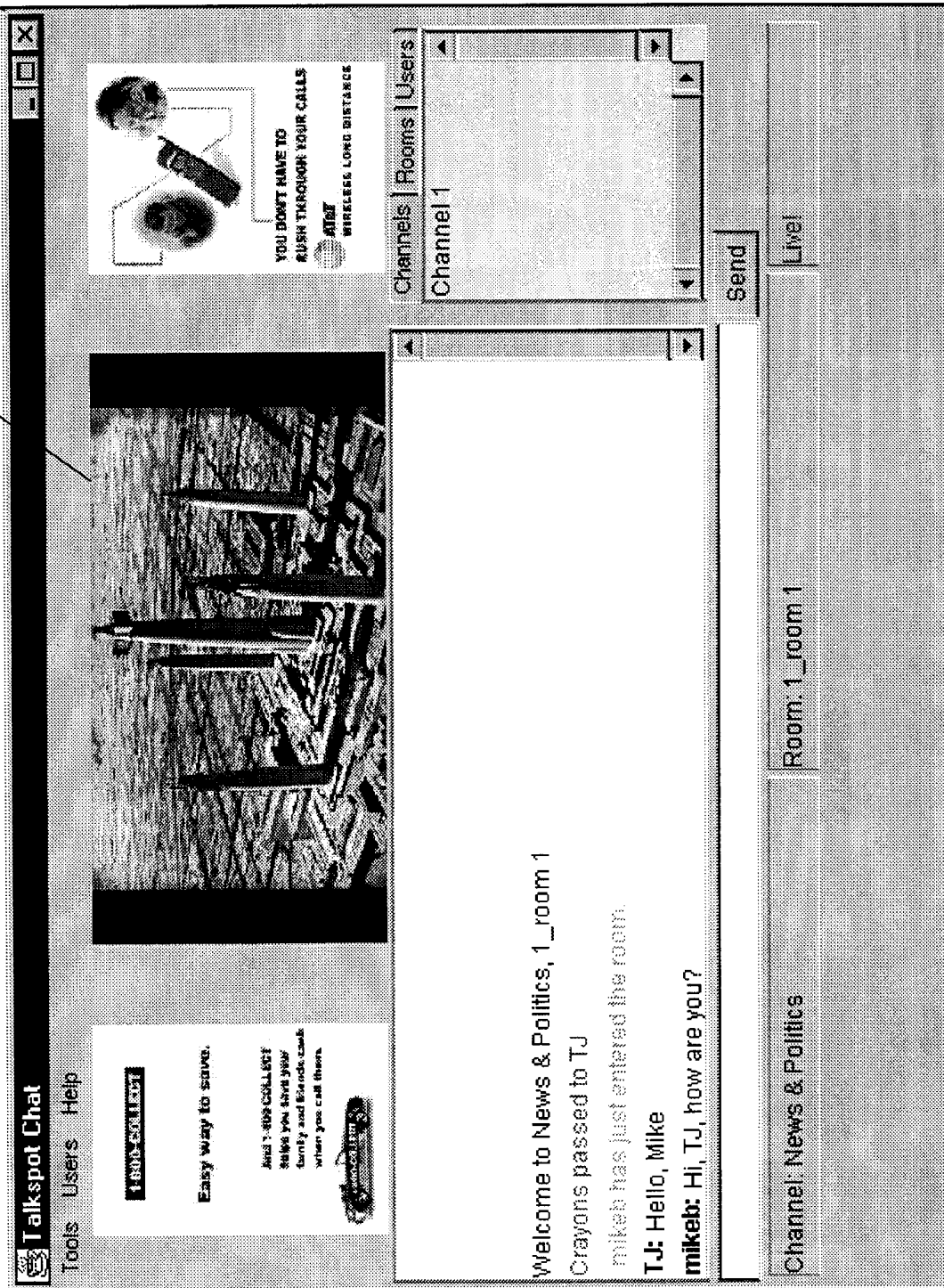


Fig. 17

1800

1825

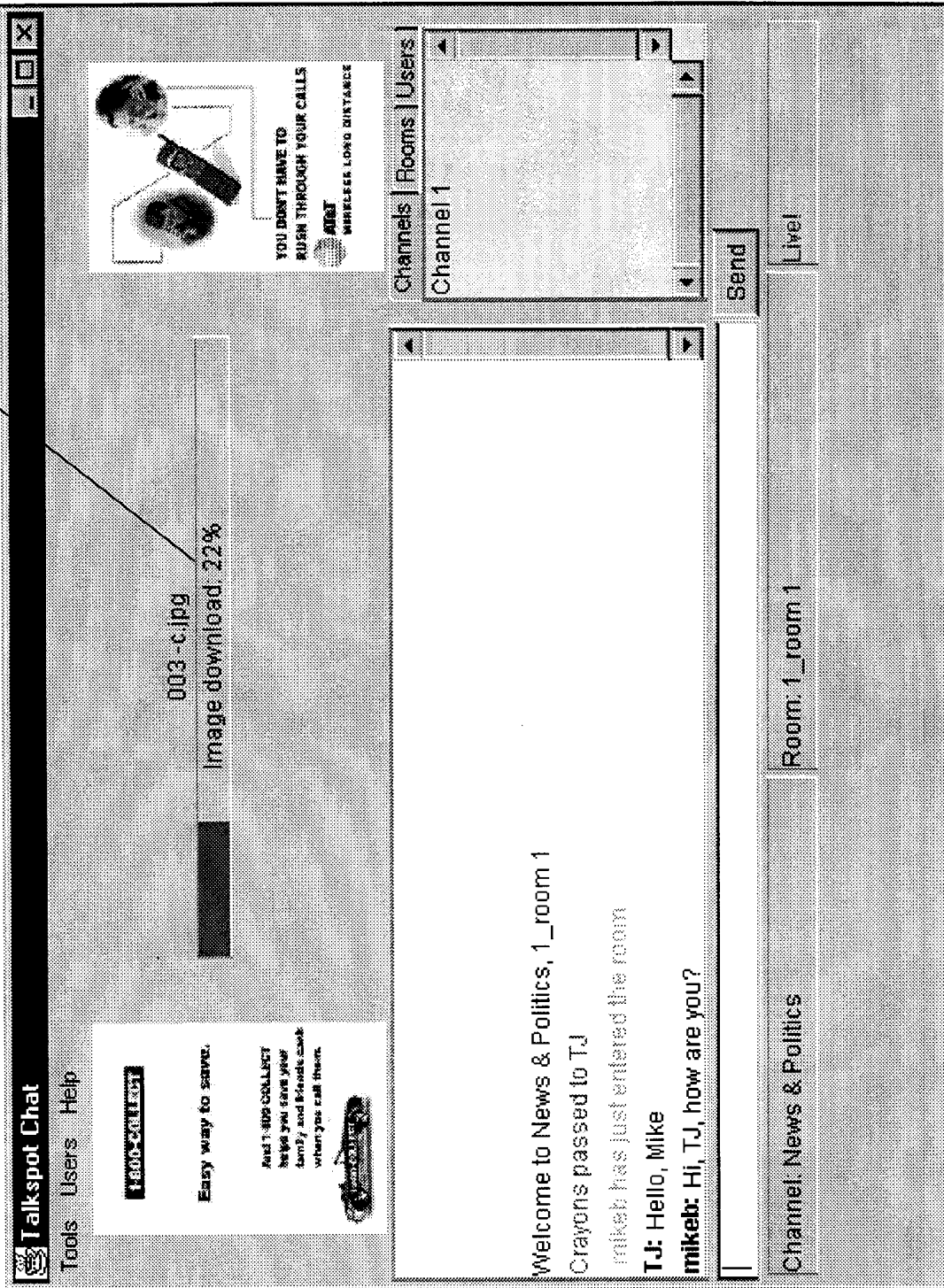


Fig. 18

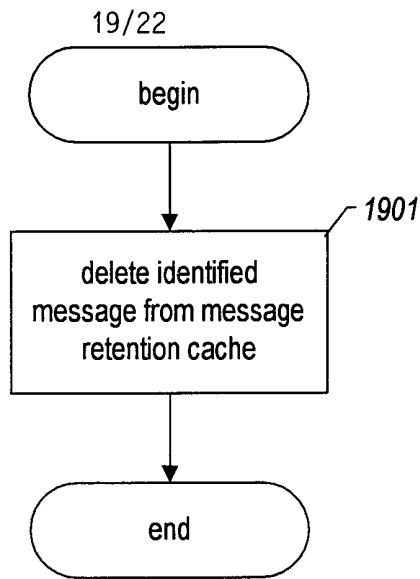


Fig. 19

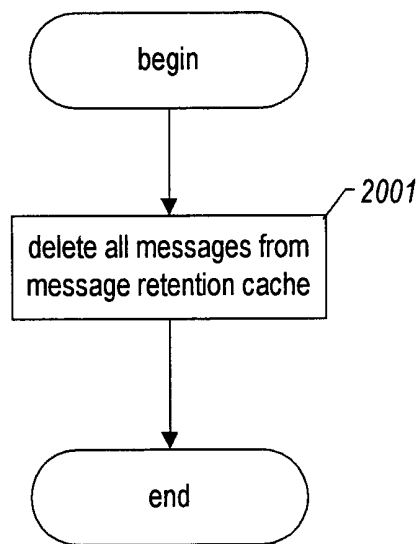


Fig. 20

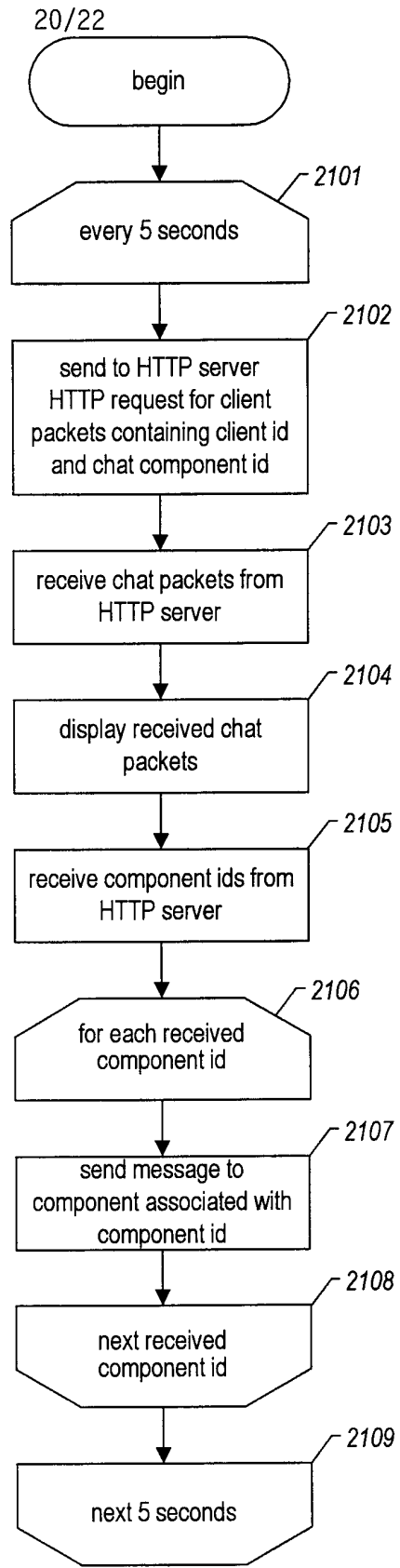


Fig. 21

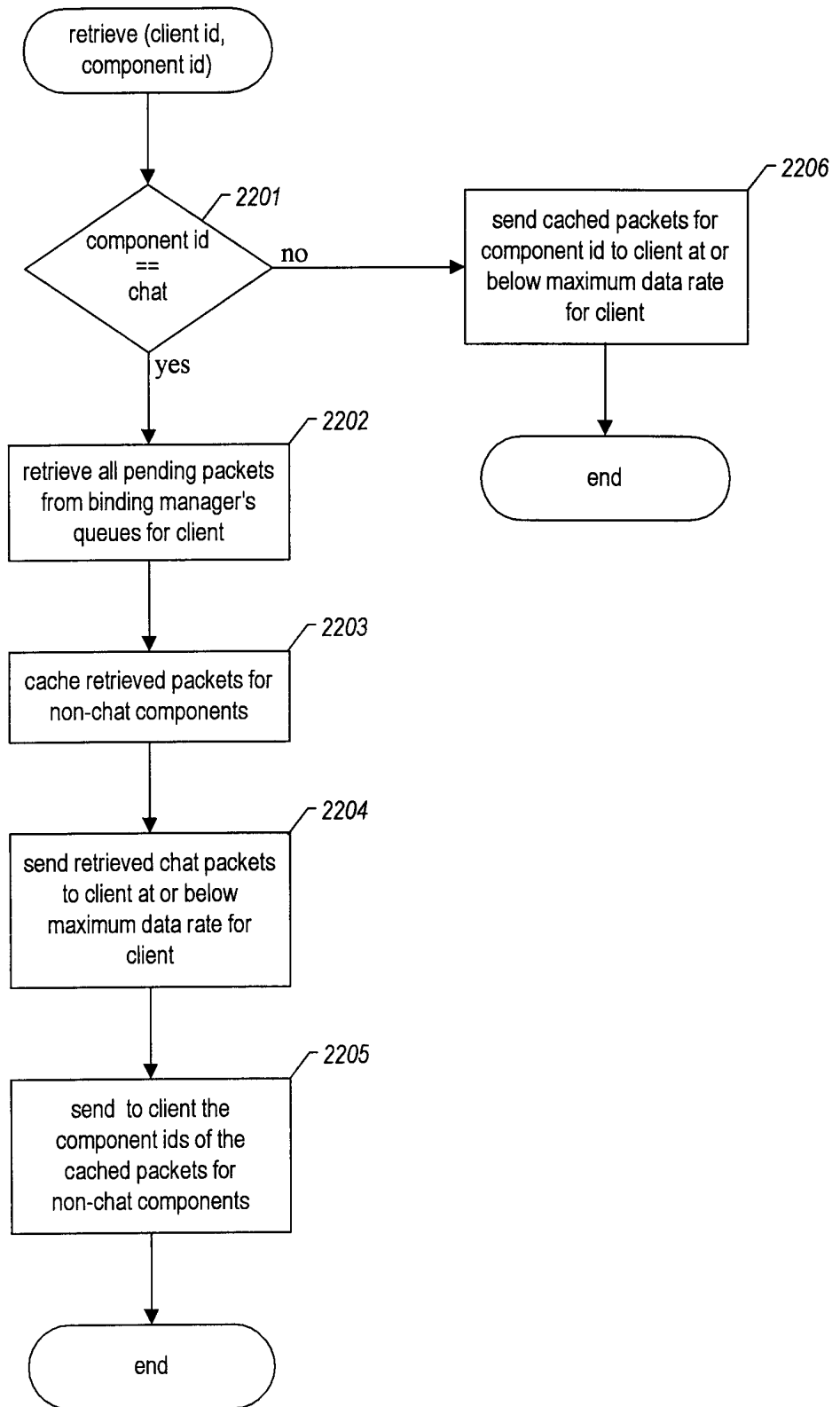


Fig. 22

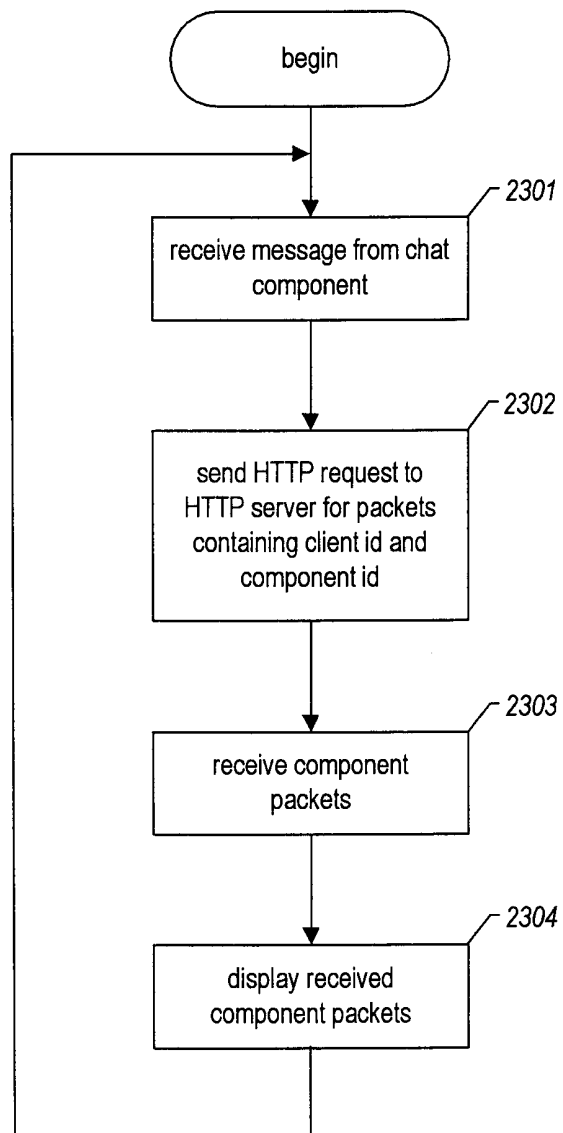


Fig. 23

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/14897

A. CLASSIFICATION OF SUBJECT MATTER IPC 7 H04L12/56 H04L29/06				
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) IPC 7 H04L G06F				
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 practical, search terms used) EPO-Internal, WPI Data, PAJ				
C. DOCUMENTS CONSIDERED TO BE RELEVANT				
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
X Y A	US 5 832 232 A (DANNEELS GUNNER D) 3 November 1998 (1998-11-03) column 3, line 63 -column 7, line 36 --- US 5 799 002 A (KRISHNAN MURALI R) 25 August 1998 (1998-08-25) column 1, line 45 - line 61 column 4, line 34 -column 6, line 25 --- WO 99 04345 A (TIBCO SOFTWARE INC) 28 January 1999 (1999-01-28) page 73, line 14 -page 75, line 26 -----	8 1-3,5,9 4,6,7, 10,12,18 1-3,5,9		
A	WO 99 04345 A (TIBCO SOFTWARE INC) 28 January 1999 (1999-01-28) page 73, line 14 -page 75, line 26 -----	1-3,5,9		
<input type="checkbox"/> Further documents are listed in the continuation of box C.				
<input checked="" type="checkbox"/> Patent family members are listed in annex.				
° Special categories of cited documents :				
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none; vertical-align: top;"> "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document 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 </td> <td style="width: 50%; border: none; vertical-align: top;"> "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 </td> </tr> </table>			"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document 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
"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document 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 <p style="text-align: center; font-weight: bold;">8 November 2000</p>		Date of mailing of the international search report <p style="text-align: center; font-weight: bold;">15/11/2000</p>		
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Authorized officer <p style="text-align: center; font-weight: bold;">RAMIREZ DE AREL., F</p>		

INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern: al Application No

PCT/US 00/14897

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5832232 A	03-11-1998	AU 5365198 A EP 1008058 A WO 9827496 A	15-07-1998 14-06-2000 25-06-1998
US 5799002 A	25-08-1998	NONE	
WO 9904345 A	28-01-1999	US 6038601 A AU 8578898 A EP 0996893 A	14-03-2000 10-02-1999 03-05-2000