PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/56

A2

(11) International Publication Number: **WO 99/04536**

(43) International Publication Date: 28 January 1999 (28.01.99)

(21) International Application Number: PCT/FI98/00591

(22) International Filing Date: 14 July 1998 (14.07.98)

(30) Priority Data:

972981	14 July 1997 (14.07.97)	FI
973746	22 September 1997 (22.09.97)	FI
980825	9 April 1998 (09.04.98)	FI

(71) Applicant *(for all designated States except US)*: NOKIA TELECOMMUNICATIONS OY [FI/FI]; Keilalahdentie 4, FIN–02150 Espoo (FI).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: MA, Jian [FI/FI]; Pihlajamarjantie 13–15 C1, FIN–02940 Espoo (FI).

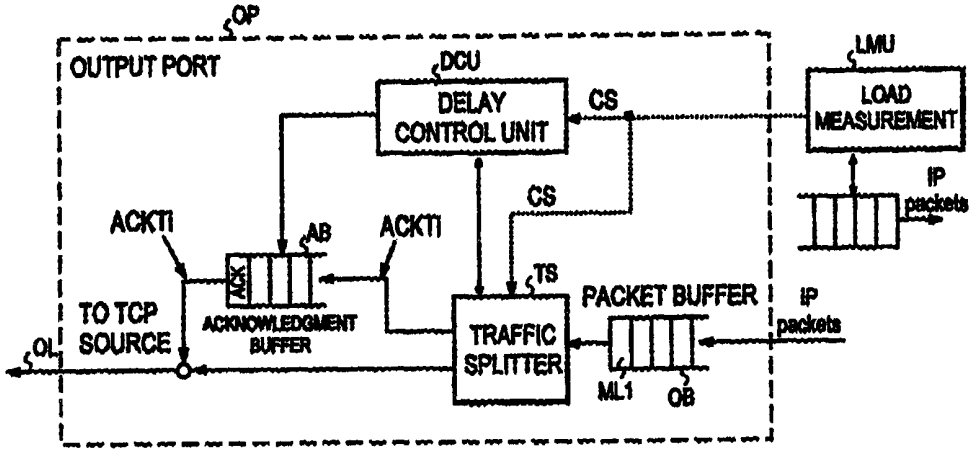
(74) Agent: PATENT AGENCY COMPATENT LTD.; Teollisuuskatu 33, P.O. Box 156, FIN–00511 Helsinki (FI).

(81) Designated States: 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, 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, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, 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).

Published

Without international search report and to be republished upon receipt of that report.

(54) Title: FLOW CONTROL IN A TELECOMMUNICATIONS NETWORK



(57) Abstract

The invention relates to a method for controlling overload in a packet switched network, especially in a network where Transmission Control Protocol (TCP) is used as the transport layer protocol. In order to inform the traffic source at a very early stage that the network is getting overloaded or congested, the acknowledgments traveling towards the source are delayed when the load level in the network exceeds a predetermined value.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

Flow control in a telecommunications network

Field of the invention

This invention relates generally to flow control in a telecommunications network. More particularly, the invention relates to congestion control in a packet switched telecommunications network, especially in a network where Transmission Control Protocol (TCP) is used as a transport layer protocol.

Background of the invention

As commonly known, TCP is the most popular transport layer protocol for data transfer. It provides a connection-oriented reliable transfer of data between two communicating hosts. (Host refers to a network-connected computer, or to any system that can be connected to a network for offering services to another host connected to the same network.) TCP uses several techniques to maximize the performance of the connection by monitoring different variables relating to the connection. For example, TCP includes an internal algorithm for avoiding congestion.

ATM (Asynchronous Transfer Mode), in turn, is a (newer) connection-oriented packet-switching technique which the international telecommunication standardization organization ITU-T has chosen as the target solution of a broadband integrated services digital network (B-ISDN). The problems of conventional packet networks have been eliminated in the ATM network by using short packets of a standard length (53 bytes), known as cells. ATM networks are quickly being adopted as backbones for the various parts of TCP/IP networks (such as Internet).

Although ATM has been designed to provide an end-to-end transport level service, it is very likely that also in the future networks will be implemented in such a way that (a) TCP/IP remains as the de-facto standard of the networks and (b) only part of the end-to-end path of a connection is implemented using ATM. Thus, even though ATM will continue to be utilized, TCP will still be needed to provide the end-to-end transport functions.

The introduction of ATM also means that implementations must be able to support the huge legacy of existing data applications, in which TCP is widely used as transport layer protocol. To migrate the existing upper layer protocols to ATM networks, several approaches to congestion control in ATM networks have been considered in the past.

Congestion control relates to the general problem of traffic management for packet switched networks. Congestion means a situation in which the number of transmission requests at a specific time exceeds the transmission capacity at a certain network point (called a bottle-neck resource). Congestion usually results in overload conditions. As a result, the buffers overflow, for instance, so that packets are retransmitted either by the network or by the subscriber. In general, congestion arises when the incoming traffic to a specific link is more than the outgoing link capacity. The primary function of congestion control is to ensure good throughput and delay performance while maintaining a fair allocation of network resources to users. For the TCP traffic, whose traffic patterns are often highly bursty, congestion control poses a challenging problem. It is known that packet losses result in a significant degradation in TCP throughput. Thus, for the best possible throughput, a minimum number of packet losses should occur.

The present invention relates to congestion control in packet switched networks. For the above-mentioned reasons, most of such networks are, and will in the foreseeable future be, TCP networks or TCP over ATM networks (i.e. networks in which TCP provides the end-to-end transport functions and the ATM network provides the underlying "bit pipes"). In the following, the congestion control mechanisms of these networks are described briefly.

ATM Forum has specified five different service categories which relate traffic characteristics and the quality of service (QoS) requirements to network behavior. These service classes are: constant bit rate (CBR), real-time variable bit rate (rt-VBR), non-real time variable bit rate (nrt-VBR), available bit rate (ABR), and unspecified bit rate (UBR). These service classes divide the traffic between guaranteed traffic and so-called "best effort traffic", the latter being the traffic which fills in the left-over bandwidth after the guaranteed traffic has been served.

One possible solution for the best effort traffic is to use ABR (Available Bit Rate) flow control. The basic idea behind the ABR flow control is to use special cells, so-called RM (Resource Management) cells, to adjust source rates. ABR sources periodically probe the network state (factors such as bandwidth availability, the state of congestion and impending congestion) by sending RM cells intermixed with data cells. The RM cells are turned around at the destination and sent back to the source. Along the way, ATM

switches can write congestion information on these RM cells. Upon receiving returned RM cells, the source can then increase, decrease or maintain its rate according to the information carried by the cells.

In TCP over ATM networks, the source and the destination are interconnected through an IP/ATM/IP sub-network. Figure 1 illustrates a connection between a TCP source A and a TCP destination B in a network, where the connection path goes through an ATM network using ABR flow control. When congestion is detected in the ATM network, ABR rate control becomes effective and forces the edge router R1 to reduce its transmission rate to the ATM network. Thus, the purpose of the ABR control loop is to command the ATM sources of the network to reduce their transmission rate. If congestion persists, the buffer in the router will reach its maximum capacity. As a consequence, the router starts to discard packets, resulting in the reduction of the TCP congestion window (the congestion window concept will be explained in more detail later).

From the point of view of congestion control, the network of Figure 1 comprises two independent control loops: an ABR control loop and a TCP control loop. However, this kind of congestion control, which relies on dual congestion control schemes on different protocol layers, may have an unexpected and undesirable influence on the performance of the network. To put it more accurately, the inner control loop (ABR loop) may cause unexpected delays in the outer control loop (TCP loop).

An alternative approach to support the best effort traffic is to use UBR service with sufficiently large buffers and let the higher layer protocols, such as TCP, handle overload or congestion situations. Figure 2 illustrates this kind of network, i.e. a TCP over UBR network. The nodes of this kind of network comprise packet discard mechanisms which discard packets or cells when congestion occurs. When a packet is discarded somewhere in the network, the corresponding TCP source does not receive an acknowledgment. As a result, the TCP source reduces its transmission rate.

The UBR service employs no flow control and provides no numerical guarantees on the quality of service; it is therefore also the least expensive service to provide. However, because of its simplicity, plain UBR without adequate buffer sizes gives poor performance in a congested network.

To eliminate this drawback, more sophisticated congestion control mechanisms have been proposed. One is the so-called early packet discard

(EPD) scheme. According to the early packet discard scheme, an ATM switch drops entire packets prior to buffer overflow. In this way the throughput of TCP over ATM can be much improved, as the ATM switches need not transmit cells of a packet with corrupted cells, i.e. cells belonging to packets in which at least one cell is discarded (these packets would be discarded during the reassembly of packets in any case). Another advantage of the EPD scheme is that it is relatively inexpensive to implement in an ATM switch. For those interested in the subject, a detailed description of the EPD method can be found, for example, in an article by A. Romanow and S. Floyd, *Dynamics of TCP Traffic over ATM Networks*, Proc. ACM SIGCOMM '94, pp. 79-88, August 1994.

However, the EPD method still deals unfairly with the users. This is due to the fact that the EPD scheme discards complete packets from all connections, without taking into account their current rates or their relative shares in the buffer, i.e. without taking into account their relative contribution to an overload situation. To remedy this drawback, several variations for selective drop policies have been proposed. One of these is described in an article by Rohit Goyal, *Performance of TCP/IP over UBR+*, ATM_Forum/96-1269. This method uses a FIFO buffer at the switch, and performs some per-VC accounting to keep track of the buffer occupancy of each virtual circuit. In this way only cells from overloading connections can be dropped, whereas the underloading connections can increase their throughput.

Despite all the improvements mentioned above, the prior art congestion control methods still have the major drawback that there is no means of giving early warning to the traffic source when excessive load is detected in the network. In other words, the traffic source is not informed quickly of overload so that it could reduce its output rate.

Summary of the invention

The purpose of the invention is to eliminate the above-mentioned drawback and to create a method by means of which it is possible, using a simple implementation, to inform the traffic source at a very early stage that the network is becoming overloaded or congested and to ask the source to slow down its transmission rate. The purpose is also that the method allows the co-operation of TCP and ATM flow control mechanisms in an efficient way.

This goal can be attained by using the solution defined in the independent patent claims.

The basic idea of the invention is to delay the acknowledgments being transferred from the destination towards the sender. This can be done at the same network point where congestion has been detected, or, alternatively, a network point detecting overload or congestion can direct another network point to delay the acknowledgments. Thus, with this invention congestion control is performed on the return path of the connection, whereas prior art systems control traffic on the forward path. Instead of discarding packets or cells on the forward path, the network according to the present invention delays acknowledgments on the return path and thus causes the TCP source to reduce its output rate.

The invention offers an inexpensive solution for giving the TCP source an early warning of impending overload or congestion in the network. It is also important to note that the transport protocol TCP itself does not have to be amended in any way. To put the invention into use, a congestion control algorithm must be introduced into the network, but for this purpose many existing control algorithms in the TCP over UBR can be utilized with only slight modifications.

Moreover, by means of the present invention the variations in the output rate of the TCP source can be smoothed, which in turn results in better bandwidth utilization. Further, because the amount of variation is lessened, the buffer capacity requirements are also reduced.

According to one preferred embodiment of the invention, load level information from an ATM network point is transported in RM cells to a node providing access to the ATM network, and the acknowledgments are delayed in said access node on the basis of the information contained in the RM cells. In this way the TCP and the ATM flow control mechanisms can be made dependent on each other so that they function efficiently together.

By means of the invention the performance of connections can be significantly improved, especially in large latency networks.

Brief description of the drawings

In the following, the invention and its preferred embodiments are described in closer detail with reference to examples shown in the appended drawings, wherein

Figure 1 illustrates a TCP connection path through an ABR-based ATM sub-network,

Figure 2 illustrates a TCP connection path through a UBR-based ATM sub-network,

Figure 3 illustrates the flow control loop according to the present invention in a TCP over ATM network,

Figure 4a illustrates one possible implementation of the new method in an IP switch,

Figure 4b is a time diagram showing the significant moments of the implementation of Figure 4a,

Figure 5 is a flow chart illustrating a method for determining delay values,

Figure 6a illustrates another possible implementation of the delaying of acknowledgments in a switch,

Figure 6b illustrates an alternative way of using acknowledgment buffers,

Figure 7a illustrates one way of applying the method to an IP network,

Figure 7b illustrates another way of applying the method to an IP network,

Figure 8a illustrates one way of applying the method to an ATM network,

Figure 8b illustrates another way of applying the method to an ATM network,

Figure 9 illustrates the interworking of the TCP and ATM flow control loops according to a preferred embodiment of the invention,

Figure 10 illustrates an example of packet transfer between the traffic source and the traffic destination in a known TCP network,

Figure 11 illustrates an example of packet transfer between the traffic source and the traffic destination in a TCP network utilizing the method according to the invention,

Figure 12 is a flow diagram illustrating a further embodiment of the method, and

Figure 13 illustrates one possible implementation of the method according to Figure 12 in an IP switch.

Detailed description of the invention

Figure 3 illustrates the basic principle of the invention by showing a connection between two user terminals (A and B) in a TCP over ATM network, i.e. the user terminals using TCP as a transport layer protocol. In addition to the access nodes (AN1 and AN2) of the user terminals, only one intermediate

node (N1) and the transmission lines (TL1, TL2) connecting the nodes are shown.

The TCP connection between the hosts A and B starts out, the same as any other TCP connection, with a negotiation between the hosts to open the connection. This initial negotiation is called a three-way handshake, as three opening segments are transmitted during this handshake phase. The term "segment" refers to the unit of information passed by TCP to IP (Internet Protocol). IP headers are attached to these TCP segments to form IP datagrams, i.e. TCP segments are transferred to the receiver within IP datagrams which is the information unit used by IP. During the initial handshaking process, the hosts inform each other for example of the maximum segment size they will accept. This is done to avoid fragmentation of the TCP segments, as fragmentation would slow down the performance of the TCP connection considerably.

After the initial handshake has been completed, the hosts begin to send data by means of the TCP segments. Each uncorrupted TCP segment, including the handshaking segments, is acknowledged. To illustrate the basic idea of the invention, let us assume that host A sends one TCP segment to host B. At the network layer, host A adds an IP header to this TCP segment to form an IP datagram. This datagram is converted into standard ATM cells in an access node AN1 located at the edge of the ATM network ANW. The cells of the datagram are then routed through the ATM network to the access node AN2 of host B. This access node reconstructs the original IP datagram from the arriving cells and sends the IP datagram to host B. Host B removes the IP header to reveal the TCP segment. If the segment is received correctly, host B sends an acknowledging TCP segment ACK1 back to host A. Up till now the network has operated in a known manner.

The load of the network is monitored in the access node AN1, for example, by monitoring the occupancy of one or more of the buffers buffering the traffic to the ATM network. If overload is detected, i.e. if buffer occupancy exceeds a predefined level, a congestion notification CM is sent inside the node to delay the acknowledgments traveling at that moment through the switch towards the traffic sources. Thus, also our exemplary acknowledgment (ACK1) is delayed when passing through access node AN1, provided that node AN1 experiences overload during that particular time period.

TCP is one of the few transport protocols that natively has a congestion control mechanism. The solution of the invention relies on this known TCP control mechanism, i.e. no other control mechanisms are needed in the source or in the destination. Therefore, this mechanism is described briefly in the following.

TCP congestion control is based on two variables: the receiver's advertised window (Wrcvr) and the congestion window (CNWD). The receiver's advertised window is maintained at the receiver as a measure of the buffering capacity of the receiver, and the congestion window is maintained at the sender as a measure of the capacity of the network. The TCP source can never send more segments than the minimum of the receiver's advertised window and the congestion window.

The TCP congestion control method comprises two phases: slow start and congestion avoidance. A variable called SSTHRES (slow start threshold) is maintained at the source to distinguish between the two phases. The source starts to transmit in the slow start phase by sending one TCP segment, i.e. the value of CWND is set to one in the beginning. When the source receives an acknowledgment, it increments CWND by one, and, as a consequence, sends two more segments. In this way the value of CWND doubles every round trip time during the slow start phase, as each segment is acknowledged by the destination terminal. The slow start phase ends and the congestion avoidance phase begins when CWND reaches the value of SSTHRES.

If a packet is lost in a TCP connection, the source does not receive acknowledgment and times out. The source sets SSTHRES to half the CWND value when the packet was lost. More precisely, SSTHRES is set to max{2, min{CWND/2, Wrcvr}}, and CWND is set to one. As a result, the source enters the congestion avoidance phase. During the congestion avoidance phase, the source increments its CWND by 1/CWND every time a segment is acknowledged.

As the invention does not in any way change the above-described known TCP congestion control mechanism, the latter is not described in more detail here. Anyone interested in the matter can find more detailed information from several books describing the field. (For example, see W. Richard Stevens, TCP/IP Illustrated Volume 1, The protocols, Addison-Wesley, 1994, ISBN 0-201-63346-9)

According to the invention, when overload or congestion is detected at a network point, one or more acknowledgments traveling towards the source on the return path are delayed. In this way the TCP source, which operates in the manner described above, automatically starts to slow down its transmission rate, or at least it does not increase its transmission rate as quickly as it otherwise would. This is because the delay slows down the rate at which the source increases the size of its congestion window.

Figure 4a illustrates this principle by showing an example in which the acknowledgments are delayed at the output port OP of an IP switch. A load measurement unit LMU measures the load level of the switch by measuring the fill rates (occupancies) of the buffers buffering the traffic passing through the switch in the forward direction. It is to be noted that the load level can be determined in any known manner.

The IP datagrams passing through the switch in the backward direction are first routed to their correct output port. At this port the received datagrams are stored in a FIFO-type output buffer OB.

A traffic splitter TS reads the stored packets out from the output buffer, one packet at a time from the first memory location ML1 of the buffer. The traffic splitter operates in the following ways.

If the congestion signal CS from the load measurement unit indicates that the load of the switch is below a predefined level, the traffic splitter forwards all the datagrams (packets) directly to the outgoing link OL, irrespective of whether they include acknowledgments or not.

On the other hand, if the congestion signal CS indicates that the load level has reached a predefined level, the traffic splitter starts to read the acknowledgment bit of each TCP header inside each IP datagram. If this bit is valid, i.e. if the datagram includes an acknowledgment, the traffic splitter forwards the packet to an acknowledgment buffer AB. If the bit is not valid, the traffic splitter forwards the packet directly to the outgoing link OL. Thus, only packets including an acknowledgment are delayed.

In the acknowledgment buffer, each IP datagram is delayed for a certain period. The length of the period is preferably directly proportional to the current load level measured by the unit LMU. After the delay period for each outgoing acknowledgment packet has elapsed, the packet is sent to the outgoing link.

If ACKTi denotes the moment in time when a packet with an acknowledgment is output from the traffic splitter to the acknowledgment buffer and ACKTo denotes the moment in time when a packet is output from the acknowledgment buffer to the link, ACKTo can be defined as follows:

$$ACKTo(j) = ACKTi(j) + d_j,\ j=1,2,...$$

where j is the packet sequence number, and d_j is the value of the delay associated with a packet with sequence number j.

Figure 4b illustrates the moments when the packets leave the traffic splitter and the acknowledgment buffer, respectively. It is assumed that excessive load is detected after ACKTo(7) (until then the acknowledgments have not been delayed). If the congestion signal received by the delay control unit DCU indicates that the level of the load has exceeded a predefined value, the delay control unit executes an algorithm defining how long the next packet to be transferred to the link should be delayed. The calculated value may depend on one or more parameters, such as the current traffic rate, the current buffer occupancy, or the previous delay value (d_{j-1}). As can be seen from Figure 4b, the value of the delay may vary from one packet to another.

Figure 5 is a flow chart illustrating an example of the algorithm which is executed by the delay control unit for each packet to be read out from the acknowledgment buffer AB.

If congestion is detected, the delay value d_j for the current packet to be read out from the acknowledgment buffer (i.e. the length of the time that the current packet is stored in the buffer) is calculated by the following formula:

$$d_j = ad_{j-1} + (1-a)d_M \quad (1)$$

where d_{j-1} is the delay value of the previous packet, d_M is a measured delay value, and a is a smoothing factor (preferably a<0,5<1). The measured delay is the actual delay as measured between the moment when a packet is received by the acknowledgment buffer and the moment when that packet is read out from the acknowledgment buffer. This delay can be measured as a mean value over a certain period of time or over a certain number of packets. The delay control unit can perform this measurement.

If congestion is detected, and if $d_{j-1}=0$ and $d_M=0$, i.e. if the previous packet was not delayed and there have been no packets in the acknowledgment buffer AB for a certain predefined preceding period, the delay value d_j of the current packet gets the value of a predefined delay parameter $d_{initial}$, i. e. $d_j = d_{initial}$.

When the switch recovers from congestion or an overload state, the delay control unit calculates the delay value d_j with the formula:

$$d_j = ad_{j-1} - (1-a)d_M \quad (2).$$

The purpose of the second term in formula (1) is to smoothly increase the delay when congestion is detected, and in formula (2) to smoothly decrease the delay when the network is recovering from congestion.

Figure 6a shows the solution according to Figure 4a in a shared buffer switch architecture. In the embodiment of Figure 6a, all the packets are buffered in a shared buffer SB prior to routing each packet to the correct output port OP_i of the switch. In other respects, the embodiment of Figure 6a correspond to the embodiment shown in Figure 4a. The traffic splitters TS_i (i=1..n) can also form a single unit which reads one packet at a time from the shared buffer and delivers the packet to the correct port. The delay control unit DCU (not shown in Figure 6a) can also be implemented as a common unit for all the output ports.

In the embodiments of Figures 4a and 6a, the acknowledgment buffer contains packets from several connections, and all the packets are delayed according to the same delay algorithm. Alternatively, the packets may be stored on a per-connection basis at each output port, i.e. the data packets of each IP connection (or each TCP connection) can be stored in a separate buffer. In these cases each buffer can be a FIFO-type buffer, as the packets of a single queue do not have to be re-sequenced even though different connections were delayed in different ways. Also the relative share of each connection in the forward buffer can be determined through measurement of the load level, and the connections can be delayed on the basis of the measured values. In this way the acknowledgments of connections loading the network more heavily can be delayed longer. Figure 6b illustrates this alternative embodiment in which the output port has a buffer unit BFU, including separate queues for at least some of the connections.

If connection-specific buffers are not used, and if different connections are delayed in different ways, the buffers can, for example, be shift-register type memories, allowing the re-sequencing of packets so that packets of underloading connections can pass packets of overloading connections.

As mentioned earlier, the congestion control method in accordance with the invention can be utilized in packet networks. This means that the network comprises user terminals, network access points providing access to

the network, and switches. The user terminals act as traffic sources and destinations, i.e. as points transmitting and receiving data. The switches can be packet switches or ATM switches. An access point can be, for example, a router, or an access point can carry out packet assembling/reassembling, routing or switching. The delaying of acknowledging packets is preferably carried out at the access points, but it can also be carried out in the switches within the network, as described later.

Figures 7a and 7b show two different ways of implementing the invention in an IP network. In the embodiment of Figure 7a, the congestion detection as well as the delaying of acknowledgments are carried out within the access switch IPS1, which provides access to the IP network. In the embodiment of Figure 7b, congestion detection is carried out in the access node, whereas the delaying of acknowledgments is carried out in the TCP/IP protocol stack of the user terminal UT. Congestion notifications CS are transmitted to the user terminal, where the packets with acknowledgments are delayed in the above-described manner prior to their being sent to the TCP source.

Figures 8a and 8b show two different ways of implementing the invention in association with an ATM network. In the embodiment of Figure 8a, the congestion detection and the delaying of acknowledgments are carried out in the access node AN. The access node can be divided into an interface card unit ICU and an ATM switch ASW. The interface card unit includes the ATM Adaptation Layer (AAL) functions for the segmentation and reassembly of the IP datagrams. Congestion is monitored in the ATM switch part of the node by monitoring, for example, the fill rates (occupancies) of the buffers buffering the subscriber traffic towards the network. Congestion notifications are transferred to the interface card unit, where the reassembled IP packets are delayed in the above-described manner. In the embodiment of Figure 8b, congestion is monitored in switch ASW, whereas the acknowledging packets are delayed in the TCP/IP protocol stack of the user terminal UT.

The embodiments of Figures 7a and 8a are the more advantageous ones because it is much more economical to implement the delaying of acknowledgments in a single access node than in several terminals located in user premises. Furthermore, it is naturally preferable that the user terminals need not be amended in any way to put the invention into use.

As mentioned earlier, one network element in the connection path can command another network element of the same path to perform the de-

laying. Figure 9 illustrates this principle in a TCP over ATM network by showing a connection between two user terminals (A and B), using TCP as a transport layer protocol. In addition to the access nodes (ANS and AND) of the user terminals, only one intermediate ATM node (N1) and the transmission lines connecting the nodes are shown. It is assumed that the network nodes have channels in two directions; a forward channel and a backward channel. In order to simplify the description, we assume that the data packets are sent from terminal A to terminal B via access node ANS, one or more ATM switches, and access node AND (forward direction), while the acknowledgments are returned from terminal B to terminal A via access node AND, one or more ATM switches, and access node ANS (backward direction). As indicated above, the access nodes can be divided into an interface card unit ICU and an ATM switch ASW. The interface card unit includes the ATM Adaptation Layer (AAL) functions for the segmentation and reassembly of the IP datagrams. As in the example of Figure 8a, the delaying of acknowledgments is performed in the interface card unit. However, in this case congestion is not monitored in the ATM switch part of the access node, but in an ATM switch located further within the ATM network. In Figure 9, said ATM switch, which commands the access node to delay the acknowledgments, is switch N1.

In the network of Figure 9, ABR flow control occurs between a sending end-system (ANS) and a receiving end-system (AND). As regards the RM cell flow in this bidirectional ABR connection, each termination point is both the sending and the receiving end-system. As shown in Figure 9, for the forward information flow from access node ANS to access node AND, there is a control loop consisting of two RM cell flows, one in the forward direction and the other in the backward direction. Access node ANS generates forward RM cells, which are turned around by access node AND and sent back to access node ANS as backward RM cells. These backward RM cells carry feedback information provided by the network nodes and/or the access node AND. A network node within the ATM network, such as node N1, can:

- insert feedback control information directly into RM cells when they pass the node in the forward or backward direction,
- indirectly inform the source about congestion by setting the EFCI bit (Explicit Forward Congestion Indication) in the headers of data cells (i.e. user cells) traveling in the forward direction. In this case, the access node AND updates the backward RM cells according to this congestion information,

- generate backward RM cells.

Thus, there are at least three different ways of controlling the delaying of the acknowledgments in the access node from within the network.

In RM cells, the congestion information can be inserted, for example, in the 45 octet long "Function Specific Fields", or in the subsequent "Reserved" part having a length of 6 bits. The traffic parameters forwarded to the user of ABR capability via RM cells are described in item 5.5.6.3 of the ITU-T specification I.371, and the structure of an RM cell in item 7.1 of said specification, where an interested reader can find a more detailed description of RM cells.

The EFCI bit, in turn, is the middlemost bit in the 3 bit wide PTI (Payload Type Indicator) field in the ATM cell header.

According to this preferred embodiment of the invention, when overload or congestion is detected at an ATM network node, the corresponding access node receives backward RM cells containing the congestion information. On the basis of this information, the ATM switch part of the access node adjusts its output rate towards the ATM network, and the flow control mechanism delays the acknowledgments traveling towards the traffic source on the backward channel. In this way the TCP source automatically starts to slow down its transmission rate, or at least it does not increase its transmission rate as quickly as it otherwise would. As mentioned earlier, this is because the delay slows down the rate at which the source increases the size of its congestion window.

In the above-described way the end-to-end ABR flow control can be performed without changing the interworking TCP protocol. In other words, the interworking of the ATM and TCP flow control loops can be implemented in an inexpensive way.

Figures 10 and 11 are time lines illustrating the exchange of segments between a TCP source and a TCP destination. The source is shown on the left side and the destination on the right side. The transmission and reception events have been marked with numbers starting from 3.

Figure 10 gives an example of how the source and the destination behave in a conventional network, i.e. in a network without the implementation of the inventive method on the return path of the connection. At first, the source is in the slow start phase. Let us assume that the load of the network increases gradually, and, as a result, packet P10 transmitted at number 21 is

lost in the overloaded network point. After this, the source still sends packets, as the acknowledgments it receives are in sequence. At number 37 the source finally notices that the acknowledgment number received was out of sequence and stops transmitting.

At 41 the timer of the source goes off and the source retransmits packet P10. Simultaneously the source moves over to the congestion avoidance phase.

Figure 11 gives an example of the data exchange when the network utilizes the present invention. Here, overload is detected after the destination has transmitted the seventh acknowledgment (ACK7). As a result, this and the subsequent acknowledgments (ACK8...ACK11) are delayed in the network.

As can be seen from the figure, already at number 24 the source begins to slow down its output rate, continuing in the slow start phase. As shown, the conventional network behaves in a more uneven way, i.e. first the source sends a lot of packets, and when congestion is detected, no packets are sent. On the contrary, a network implementing the present invention behaves in a much smoother way. This is because delaying the acknowledgments prevents the source from incrementing its congestion window as quickly as in the known network. Because of this, the buffering capacity of the access network can be diminished.

The above-described method can also be used together with other flow control mechanisms. As the above-described method needs a long acknowledgment buffer, if the congestion situation lasts for a long time, it may in some applications be advantageous to combine it with another mechanism which takes care of the more severe congestion situations. According to a further embodiment of the invention, the delaying of acknowledgments is used together with a method which is otherwise similar to the above method but which generates duplicate acknowledgments, instead of delaying acknowledgments. By duplicating the acknowledgments the TCP source can be made to slow down its output rate, i.e. duplication has a same kind of effect on the TCP source as delaying. This is based on the fast retransmission and fast recovery algorithms which the source automatically performs after receiving a certain number (typically three) of duplicate acknowledgments. These algorithms are nowadays widely implemented in different TCP versions. According to the algorithms the source performs, after receiving a certain number of duplicate acknowledgments, a retransmission of what appears to be the miss-

ing segment, without waiting for a retransmission timer to expire (the fast retransmission algorithm). After this the source performs congestion avoidance, instead of slow start, in order not to reduce the data flow abruptly (the fast recovery algorithm).

Figure 12 is a flow chart illustrating the combinatory method. If congestion is not detected on the forward path, the acknowledgments are forwarded without delay and with the incoming acknowledgment number. If the load measurement detects that the load level on the forward path exceeds a predetermined value (phase 111), it is tested (phase 112) whether the fill rate of the acknowledgment buffer has exceeded a predetermined value. If this is the case, duplicate acknowledgments are generated. Otherwise acknowledgments are only delayed. Thus, if congestion occurs only slightly and for a short period, delaying of acknowledgments is performed. However, should there be a more severe congestion situation, the system always moves over to generate duplicate acknowledgments. This means that a network node sends towards the source M successive acknowledgments in which the acknowledgment number, which indicates the next sequence number that the destination expects to receive, are equal to each other.

Figure 13 illustrates how this preferred embodiment is implemented in the node of Figure 4a.

As mentioned above in connection with Figure 4a, the IP datagrams passing through the switch in the backward direction are first routed to their correct output port. At this port the received datagrams are stored in a FIFO-type output buffer OB.

The traffic splitter reads the stored packets out from the output buffer, one packet at a time from the first memory location ML1 of the buffer.

If the congestion signal CS1 from the load measurement unit LMU indicates that the load of the switch on the forward path is below a predefined level, the traffic splitter forwards all the datagrams (packets) directly to the outgoing link OL, irrespective of whether they include acknowledgments or not.

On the other hand, if the congestion signal CS1 indicates that the load level has reached a predefined level, the traffic splitter starts to read the acknowledgment bit of each TCP header inside each IP datagram. If this bit is valid, i.e. if the datagram includes an acknowledgment, the traffic splitter forwards the packet to an acknowledgment buffer AB. If the bit is not valid, the traffic splitter forwards the packet directly to the outgoing link OL. Thus, only

packets including an acknowledgment are delayed.

In the acknowledgment buffer, each IP datagram is delayed for a certain period. The length of the period is preferably directly proportional to the current load level measured by the unit LMU. After the delay period for each outgoing acknowledgment packet has elapsed, the packet is sent to the outgoing link.

The load measurement unit LMU also measures the fill rate of the acknowledgment buffer AB. If this fill rate exceeds a predetermined value, the load measurement unit sends the control unit CU a second congestion signal CS2 indicating that the control unit should now begin to produce duplicate acknowledgments. The duplication can be done for example by modifying the acknowledgment number of the acknowledgments in the packet buffer OB. The traffic splitter is also informed to direct all the traffic directly to the output link. The command can be given either by the load measurement unit or by the control unit.

Although the invention has been described here in connection with the examples shown in the attached figures, it is clear that the invention is not limited to these examples, as it can be varied in several ways within the limits set by the attached patent claims. The following describes briefly some possible variations.

As indicated above, a prerequisite for a user terminal is that it acknowledges correctly received (i.e. uncorrupted) data units. Therefore, the idea can in principle be applied to any other protocol which sends acknowledgments and slows down its output rate if the acknowledgments are delayed. The formula used for calculating the absolute delay value can also vary in many ways. The measurement unit can inform about the load level in many ways; as an ON/OFF type information, or more than one bit can be used to indicate the value of the measured load. The signal (CS) informing about the load level can also include information on the particular connections that should be subject to delaying of acknowledgments. User terminals can also have wireless access to the network.

Claims

1. A method for controlling overload in a packet switched network comprising traffic sources (A), traffic destinations (B) and network nodes (AN, N1), the method comprising the steps of

- sending data units from a traffic source to a traffic destination,
- sending an acknowledgment from the destination to the source, if a data unit is received correctly at the destination, and
- measuring load level in at least one network node,

characterized by

delaying the acknowledgments traveling towards the source when the measured load level exceeds a predetermined value.

2. A method according to claim 1, characterized in that the acknowledgments are delayed in the same network node where the load level is measured.

3. A method according to claim 1, characterized in that the acknowledgments are delayed in a different network node than where the load level is measured.

4. A method according to claim 3, characterized in that the acknowledgments are delayed in an access node (AN, ANS, AND) providing the traffic sources and destinations access to the network, and the load level is measured in at least one network node (N1) located within the network.

5. A method according to claim 4, wherein the network between the access nodes is an ATM network, characterized by

- transporting load level information in RM cells to the access node, and
- delaying the acknowledgments on the basis of the information contained in the RM cells.

6. A method according to claim 1, characterized in that the acknowledgments are delayed in at least one network node by

- storing at least part of the data packets traveling in a first direction through the node in a first buffer,
- reading data packets out from the first buffer in such a way that (a) packets including an acknowledgment are transferred into a second buffer, and (b) packets failing to include an acknowledgment are transferred directly to an outgoing link (OL),

- determining a delay value for each packet in the second buffer, and

- reading out a packet from the second buffer to the outgoing link when the delay value determined for said packet has elapsed.

7. A method according to claim 6, characterized in that the first and second buffers are used at each output port of the first direction.

8. A method according to claim 6, characterized in that the delay value is determined using the same determination rule for all packets in the second buffer.

9. A method according to claim 8, characterized in that the delay value for a packet is determined on the basis of the delay value of the preceding packet and of the delay value measured over a certain preceding period.

10. A method according to claim 1, characterized in that only acknowledgments belonging to selected connections are delayed if the measured load level exceeds a predetermined value.

11. A method according to claim 1, wherein said data units travel along a forward path from the traffic source to the traffic destination and said acknowledgments travel along a backward path from the destination to the source, characterized by the steps of

- measuring load level both on the forward path and on the backward path,

- delaying acknowledgments when the load level on the forward path is higher than a first predetermined value and the measured load level on the backward path is lower than a second predetermined value, and

- transmitting duplicate acknowledgments when the load level on the forward path is higher than the first predetermined value and the measured load level on the backward path is higher than the second predetermined value.

12. Packet switched telecommunications network comprising

- nodes interconnected by transmission lines (TL1, TL2),

- user terminals (UT) connected to the nodes, said user terminals acting as traffic sources which send data packets and as traffic destinations which receive data packets, and

- measuring means (LMU) for measuring current load level in a node,

c h a r a c t e r i z e d in that the network further comprises

- delaying means (AB, DCU), operably connected to the measuring means (LCU) for delaying data packets carrying acknowledgments from a destination towards a source.

13. A network according to claim 12, c h a r a c t e r i z e d in that at least one node comprises both the measuring means and the delaying means.

14. A network according to claim 13, c h a r a c t e r i z e d in that said at least one network node is an access node connecting at least one user terminal to the network.

15. An IP network according to claim 13, wherein the network nodes switch IP packets, c h a r a c t e r i z e d in that said at least one network node can be any one or more of the network nodes.

16. A TCP over ATM network according to claim 12, c h a r a c t e r i z e d in that the delaying means are connected to the measuring means by an RM cell flow, said RM cells carrying information on the load level.

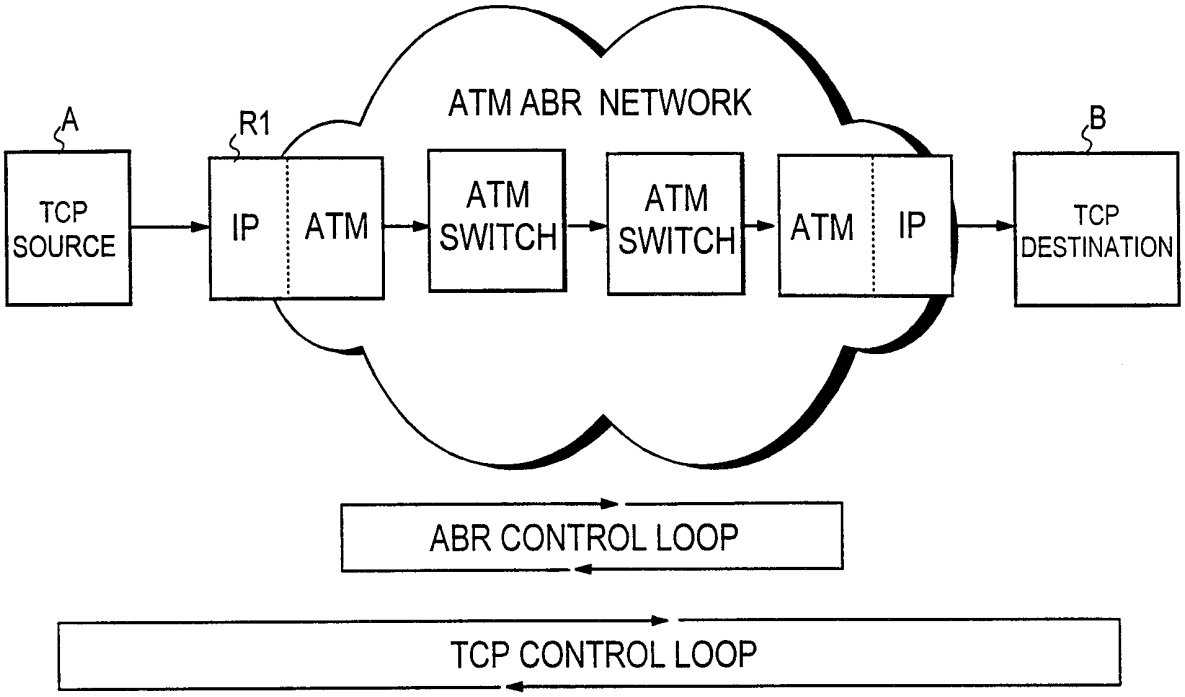

A
R1
ATM ABR NETWORK
B
TCP SOURCE
IP
ATM
ATM SWITCH
ATM SWITCH
ATM
IP
TCP DESTINATION
ABR CONTROL LOOP
TCP CONTROL LOOP

FIG. 1

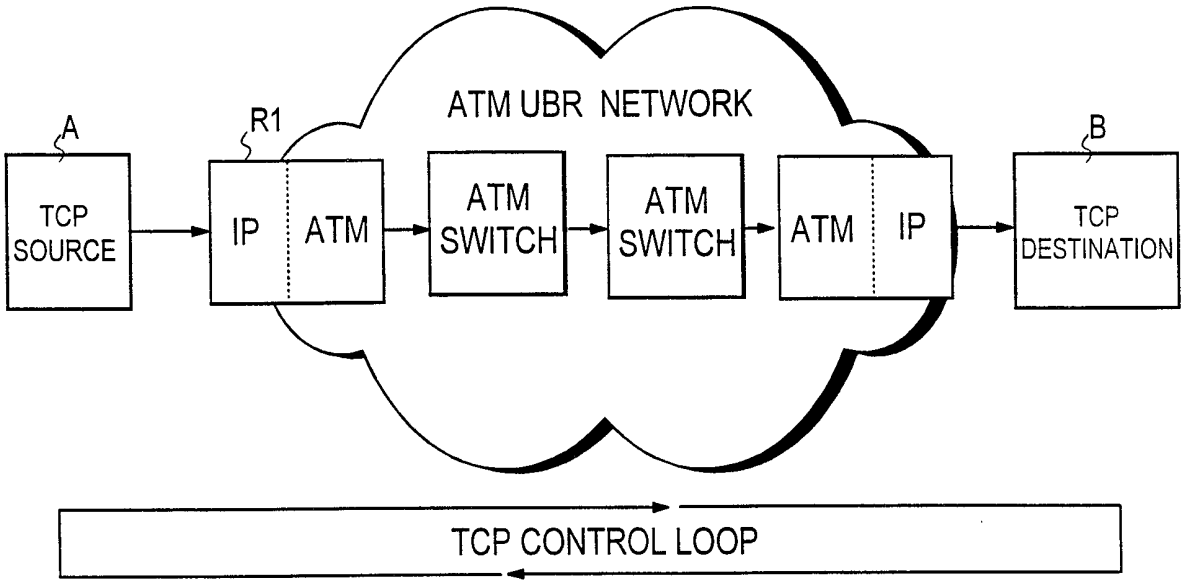

A
R1
ATM UBR NETWORK
B
TCP SOURCE
IP
ATM
ATM SWITCH
ATM SWITCH
ATM
IP
TCP DESTINATION
TCP CONTROL LOOP

FIG. 2

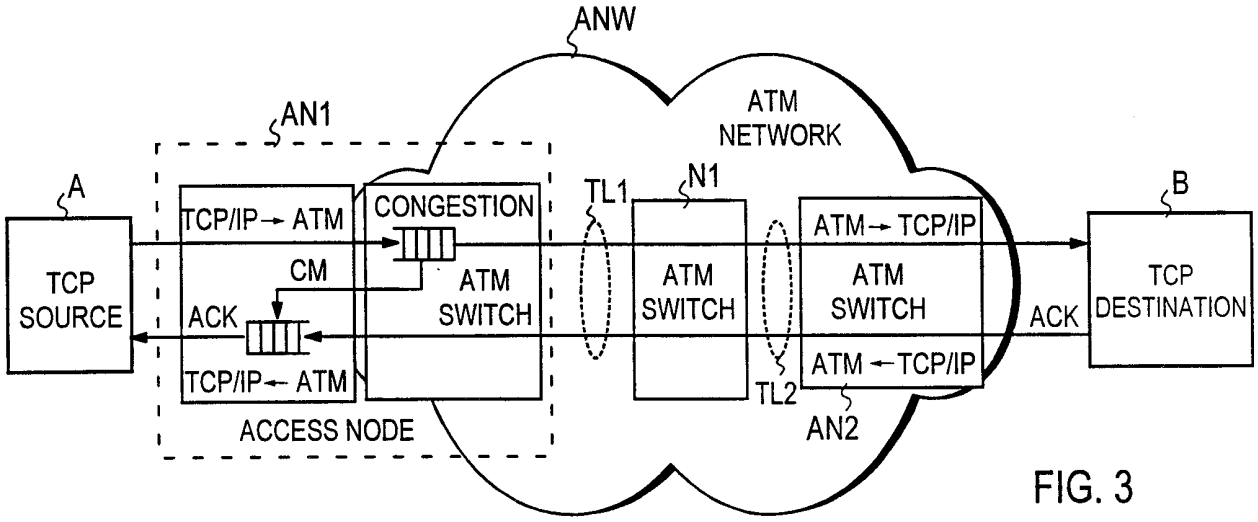


FIG. 3

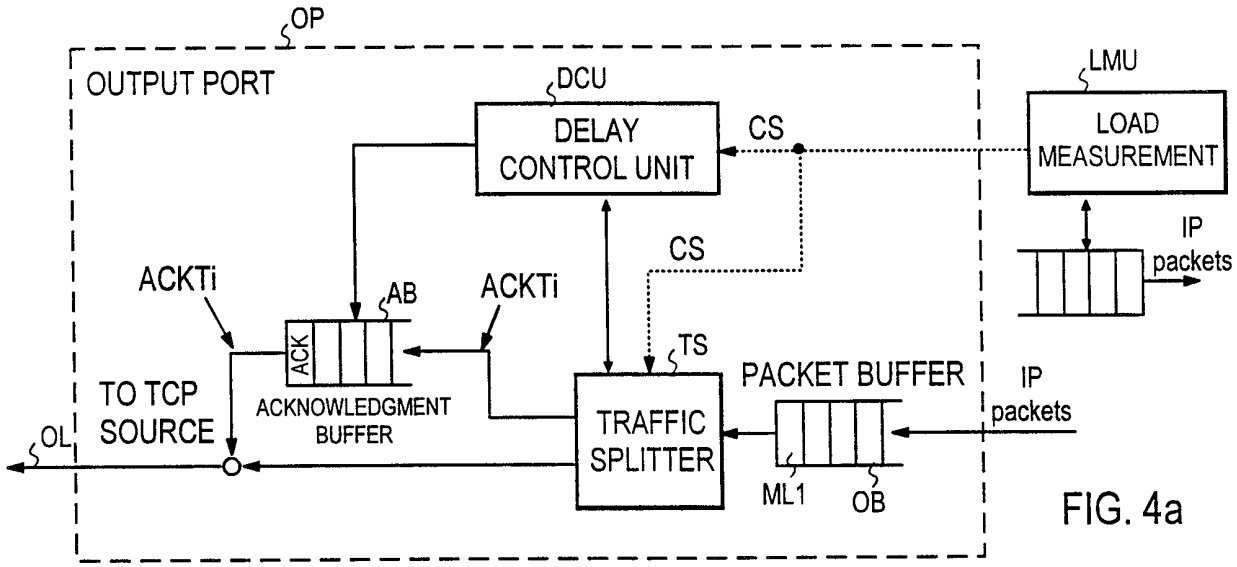


FIG. 4a

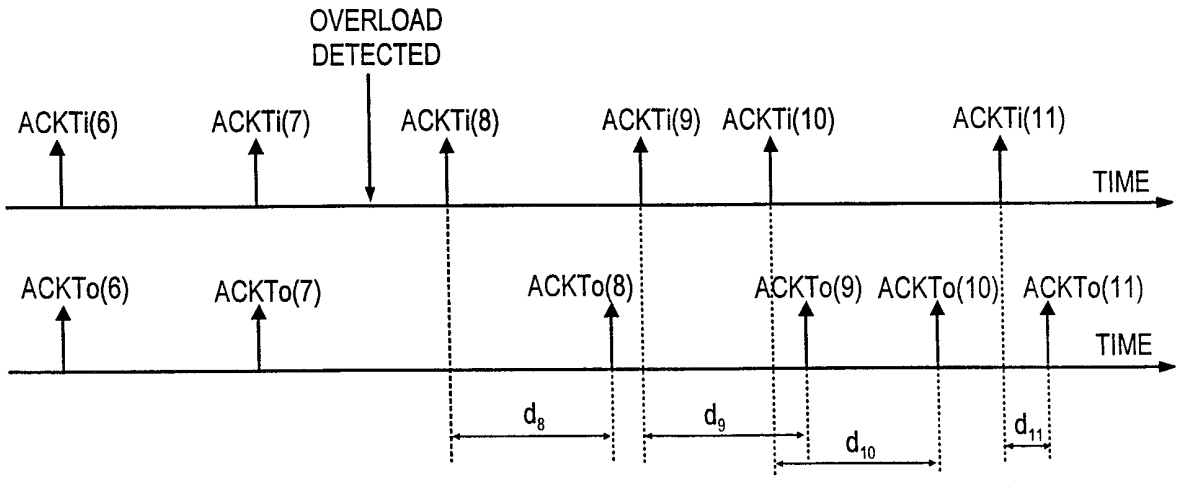


FIG. 4b

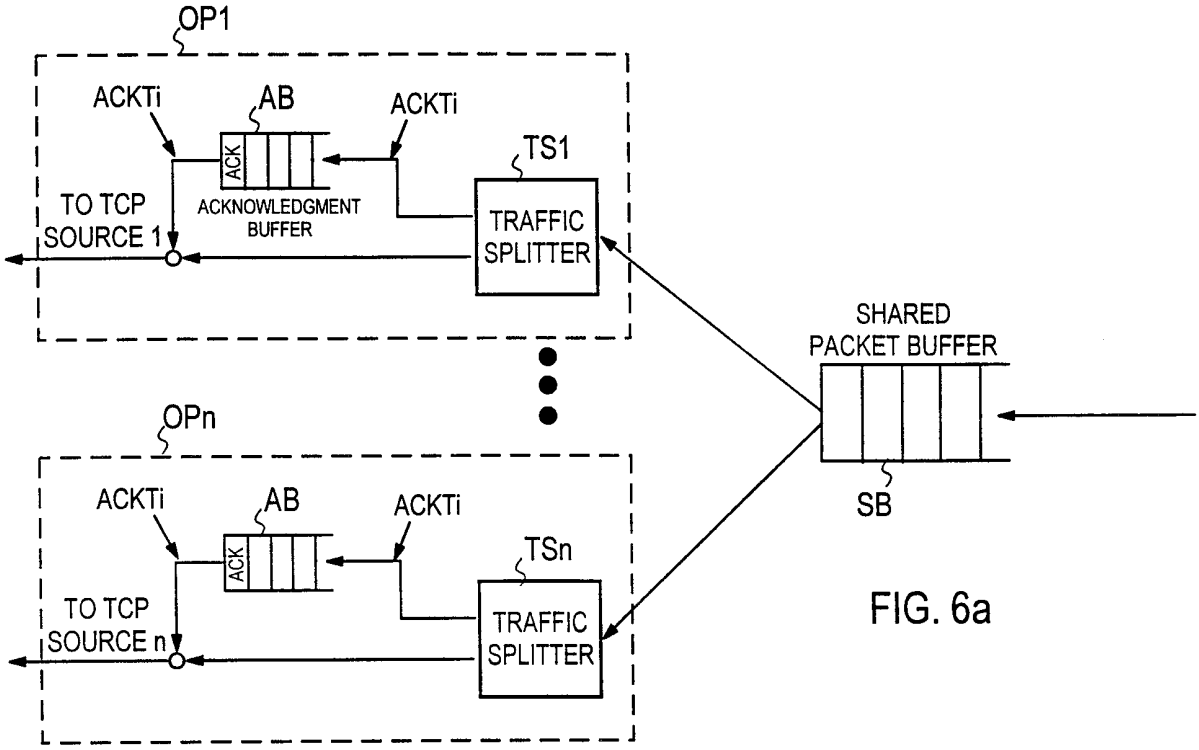

OP1
ACKTi
AB
ACKTi
TS1
ACK
TO TCP SOURCE 1
ACKNOWLEDGMENT BUFFER
TRAFFIC SPLITTER
SHARED PACKET BUFFER
OPn
ACKTi
AB
ACKTi
TSn
ACK
TO TCP SOURCE n
TRAFFIC SPLITTER
SB

FIG. 6a

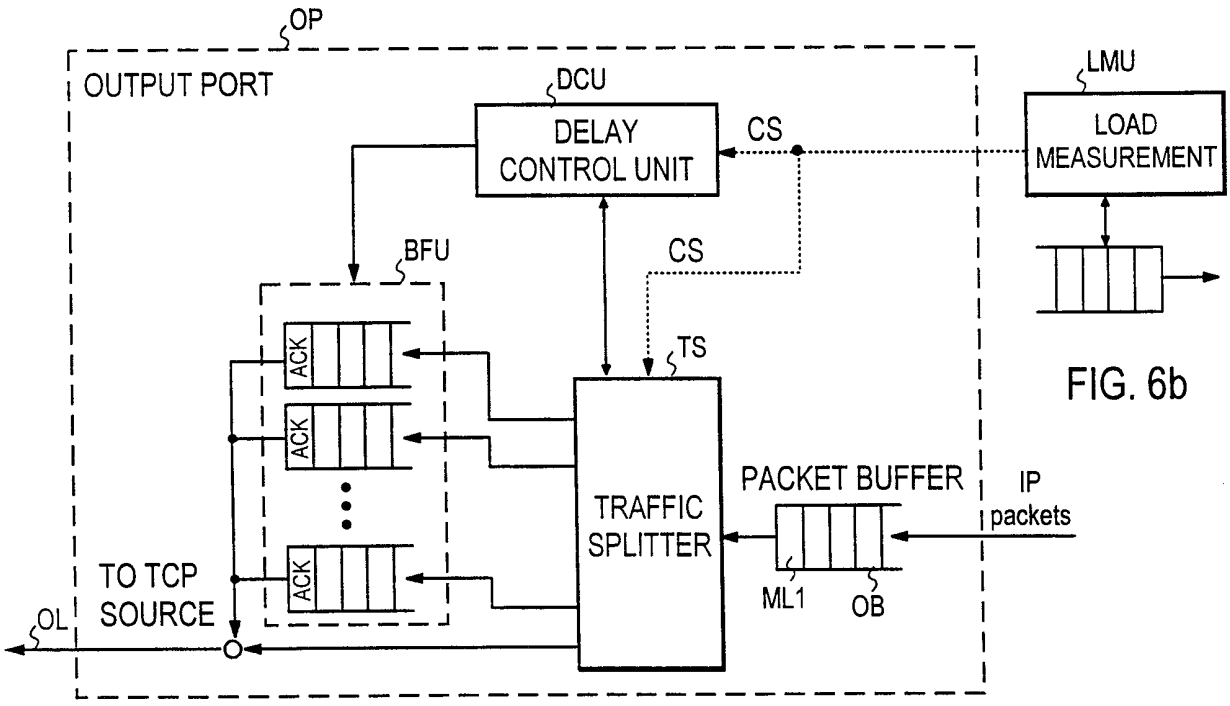

OP
OUTPUT PORT
DCU
DELAY CONTROL UNIT
CS
LMU
LOAD MEASUREMENT
BFU
CS
ACK
ACK
ACK
TS
TRAFFIC SPLITTER
PACKET BUFFER
IP packets
TO TCP SOURCE
OL
ML1
OB

FIG. 6b

FIG. 5

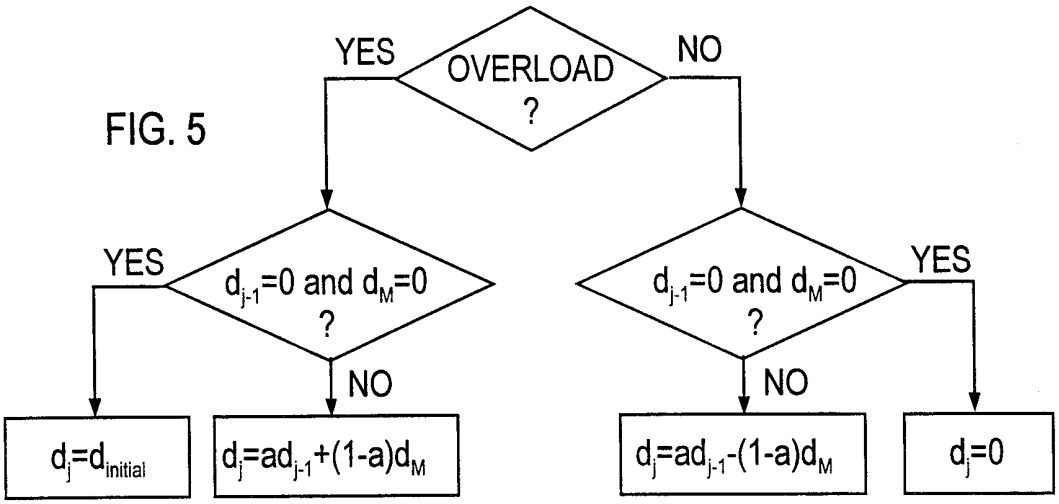


FIG. 7a

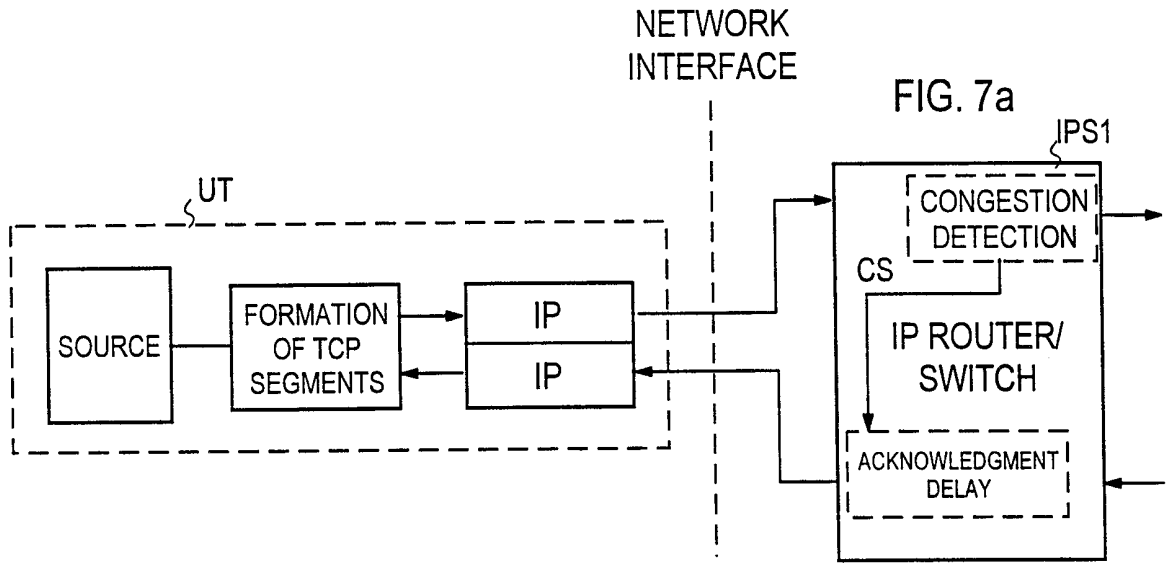
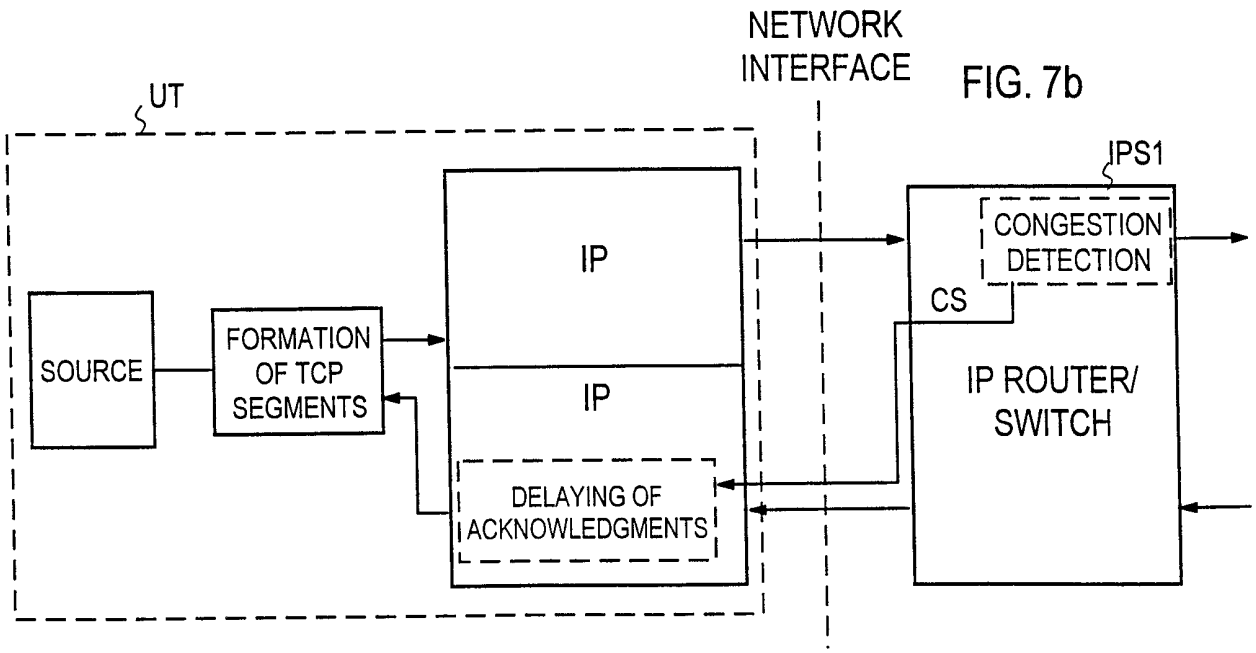


FIG. 7b



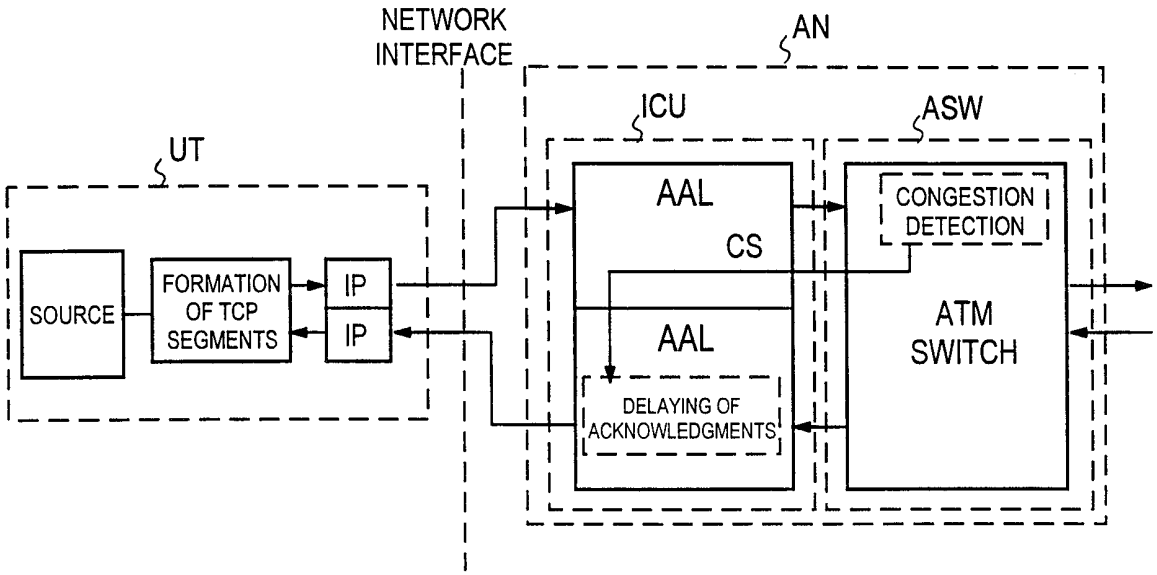

NETWORK INTERFACE
AN
ICU
ASW
UT
AAL
CS
CONGESTION DETECTION
SOURCE
FORMATION OF TCP SEGMENTS
IP
IP
AAL
ATM SWITCH
DELAYING OF ACKNOWLEDGMENTS

FIG. 8a

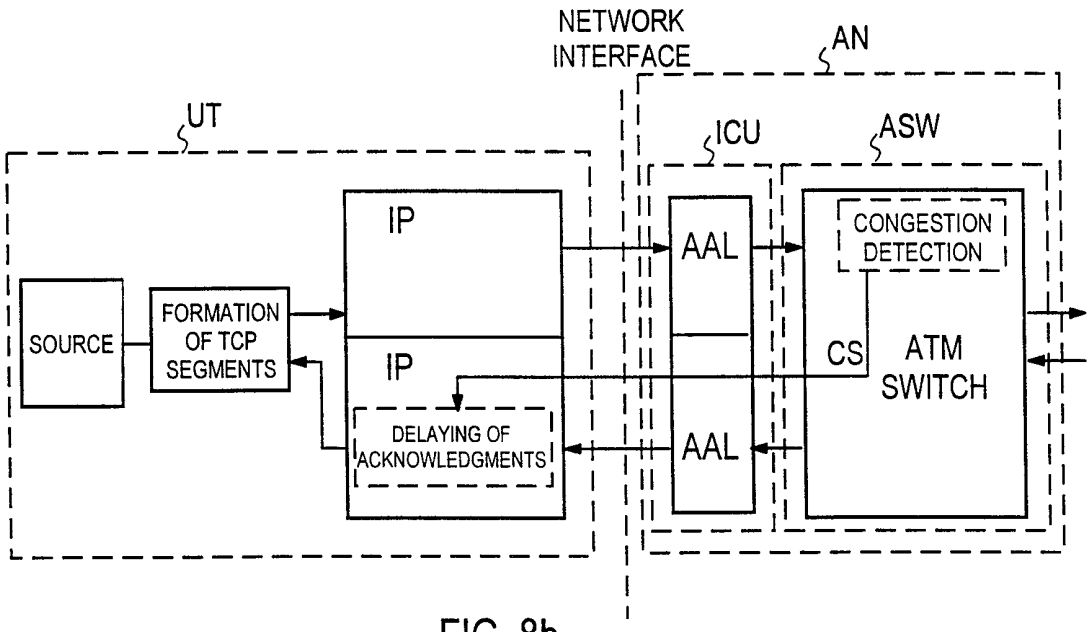

NETWORK INTERFACE
AN
UT
ICU
ASW
IP
AAL
CONGESTION DETECTION
SOURCE
FORMATION OF TCP SEGMENTS
IP
CS
ATM SWITCH
DELAYING OF ACKNOWLEDGMENTS
AAL

FIG. 8b

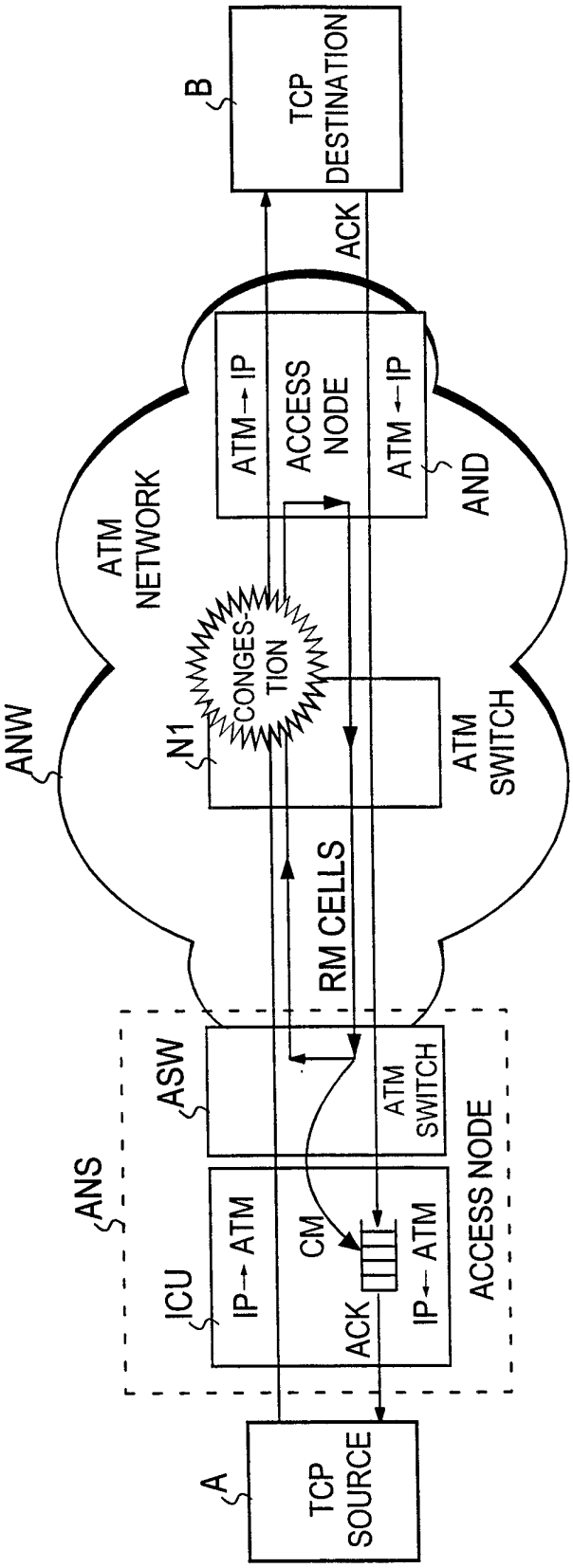

B
TCP DESTINATION
ACK
ATM→IP
ACCESS NODE
ATM←IP
AND
ATM NETWORK
CONGES-TION
ANW
N1
ATM SWITCH
RM CELLS
ASW
ATM SWITCH
ANS
ICU
IP→ATM
CM
ACK
IP←ATM
ACCESS NODE
A
TCP SOURCE

FIG. 9

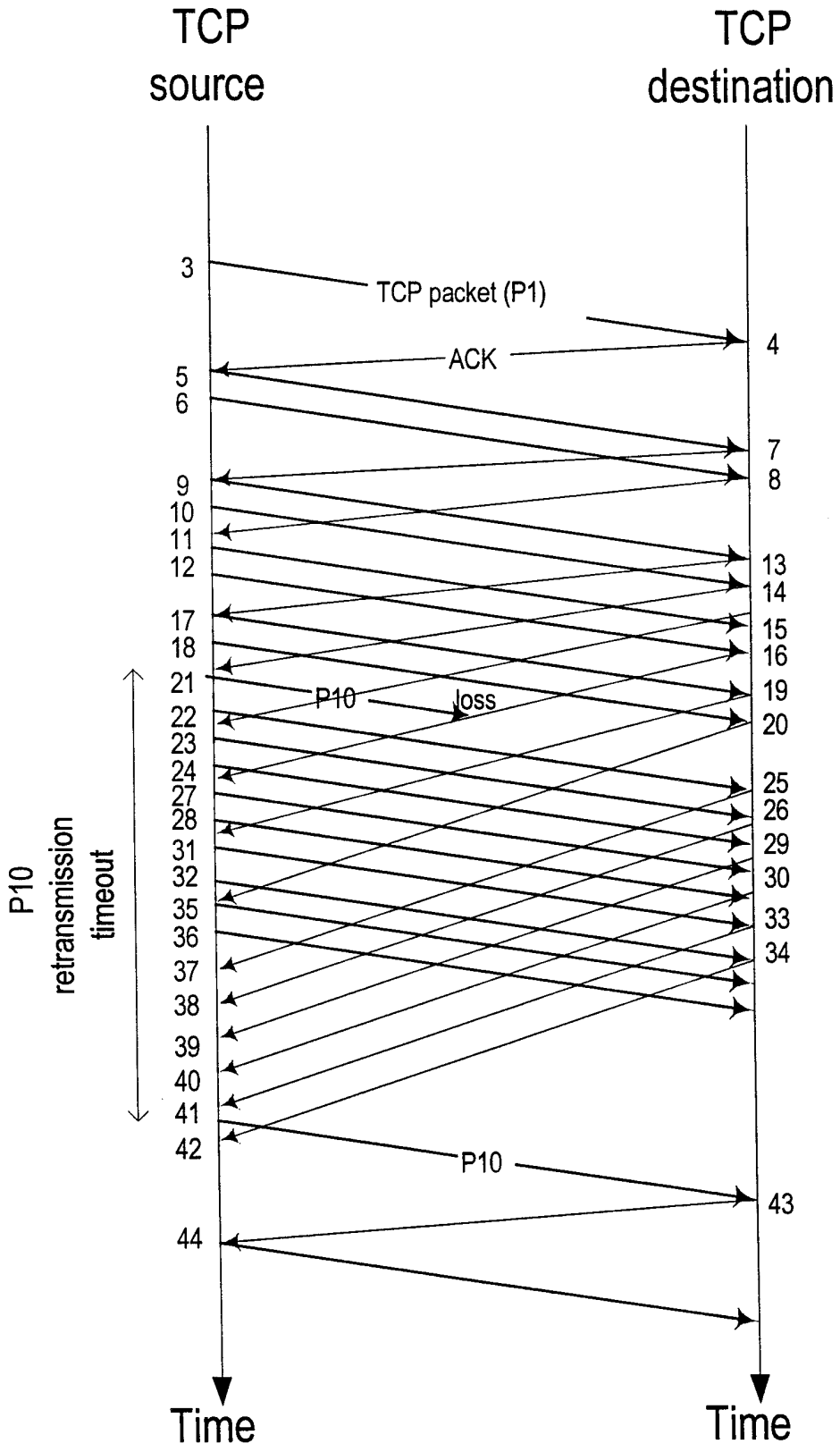

TCP source
TCP destination
3
TCP packet (P1)
4
ACK
5
6
7
8
9
10
11
12
13
14
15
16
17
18
21
P10
loss
19
20
22
23
24
25
27
26
28
29
31
30
32
35
33
36
34
37
38
39
40
41
42
P10
43
44
P10 retransmission timeout
Time
Time

FIG. 10

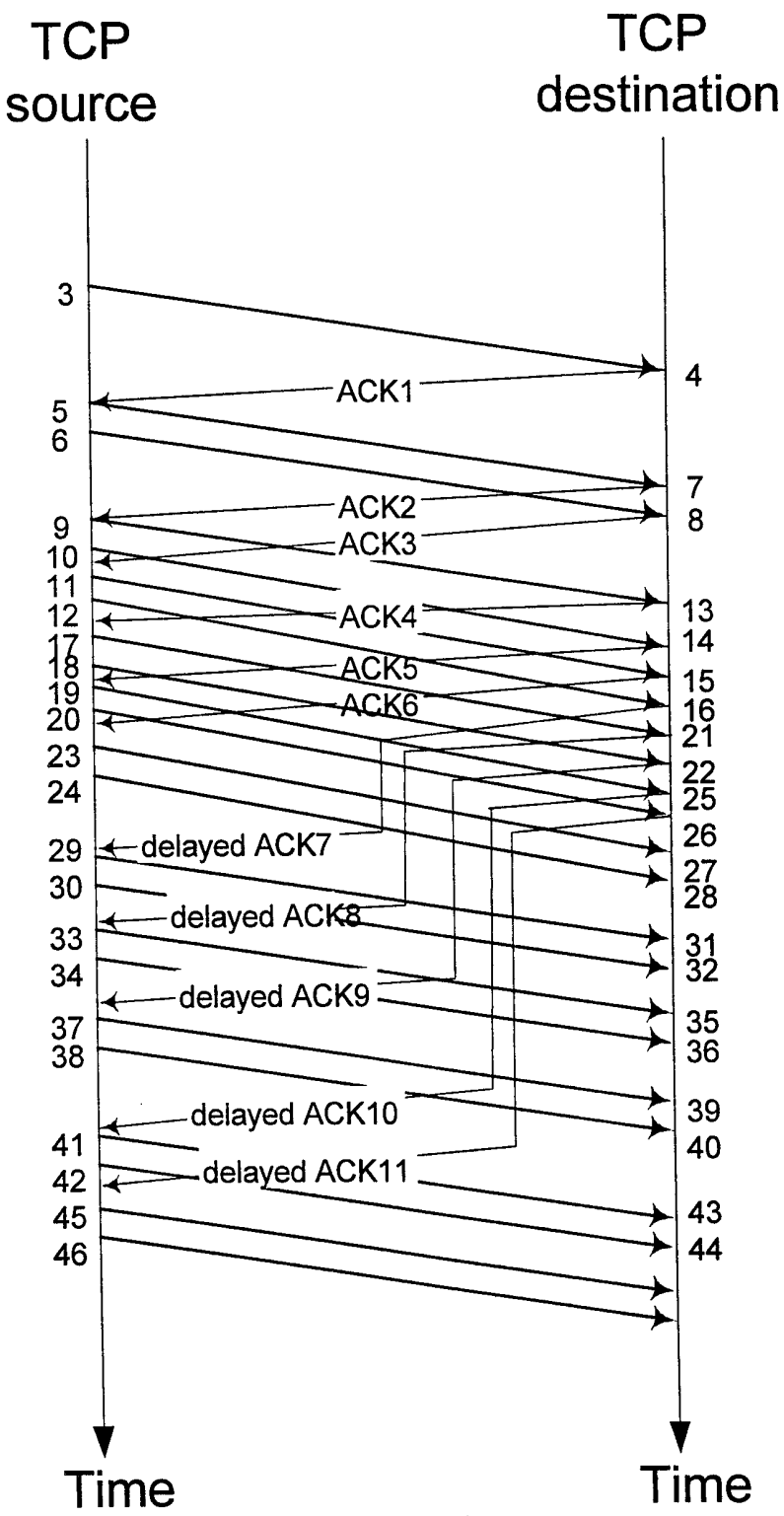

TCP source
TCP destination
3
4
ACK1
5
6
7
ACK2
8
9
ACK3
10
11
12
ACK4
13
14
ACK5
15
16
ACK6
21
22
25
26
27
28
29
delayed ACK7
30
delayed ACK8
31
32
33
34
delayed ACK9
35
36
37
38
39
delayed ACK10
40
41
delayed ACK11
42
43
44
45
46
Time
Time

FIG. 11

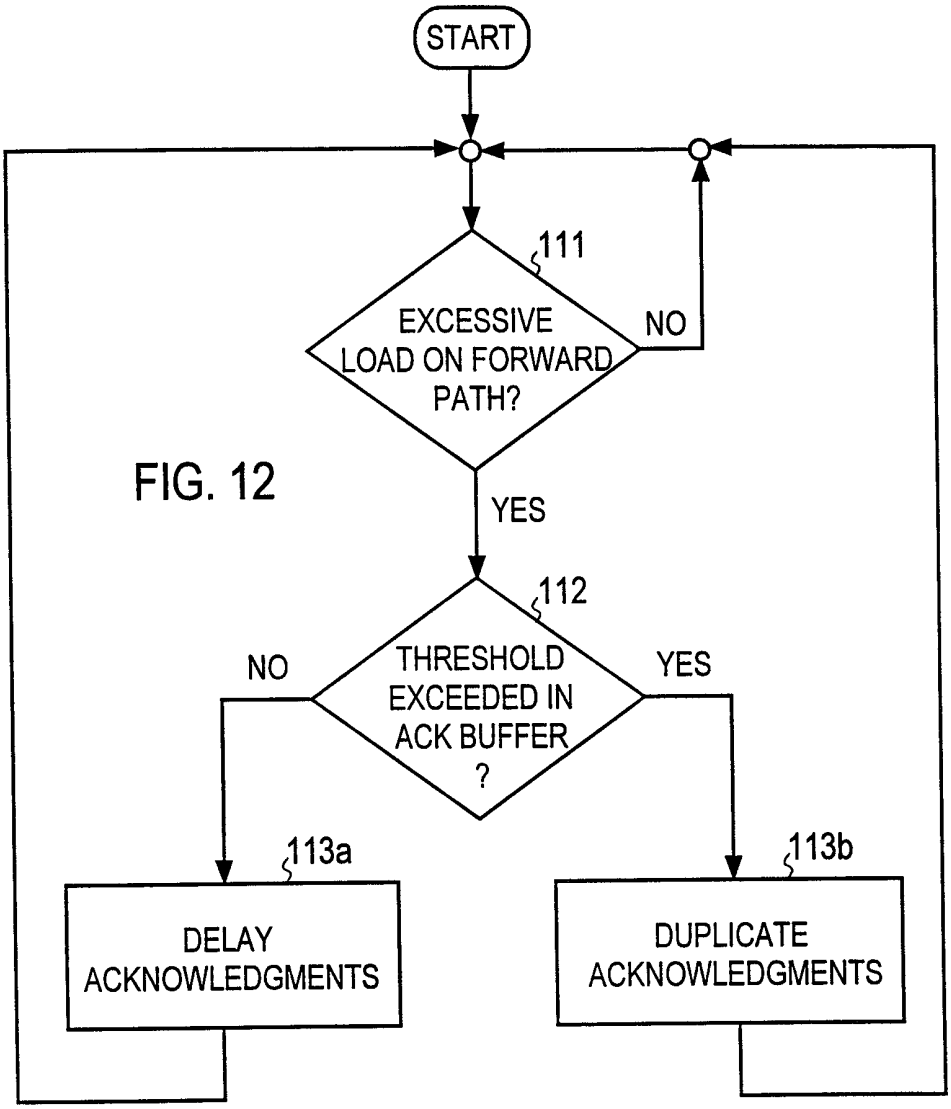

START
111
EXCESSIVE LOAD ON FORWARD PATH?
NO
YES
112
THRESHOLD EXCEEDED IN ACK BUFFER ?
NO
YES
113a
DELAY ACKNOWLEDGMENTS
113b
DUPLICATE ACKNOWLEDGMENTS

FIG. 12

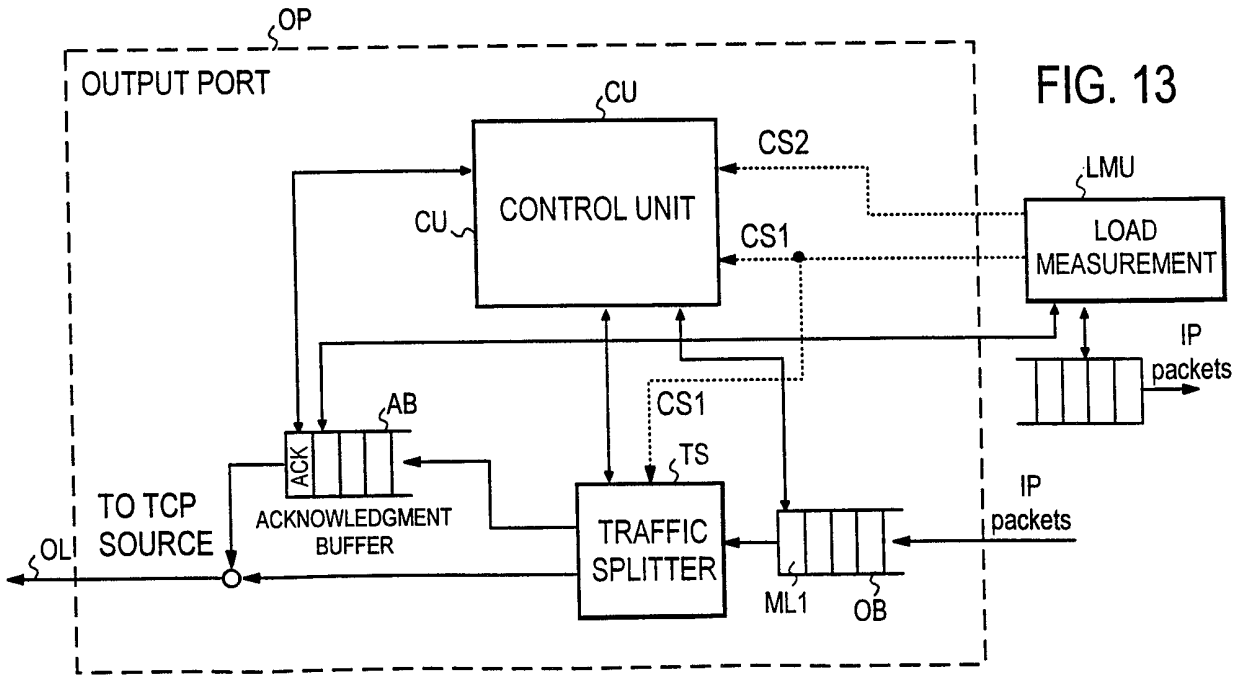

OP
OUTPUT PORT
CU
CONTROL UNIT
CS2
LMU
LOAD MEASUREMENT
CS1
IP packets
AB
ACK
ACKNOWLEDGMENT BUFFER
TS
TRAFFIC SPLITTER
TO TCP SOURCE
OL
ML1
OB
IP packets

FIG. 13