

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
11 December 2003 (11.12.2003)

PCT

(10) International Publication Number
WO 03/103236 A1

(51) International Patent Classification⁷: **H04L 12/56**

(21) International Application Number: PCT/US03/15729

(22) International Filing Date: 8 May 2003 (08.05.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/158,291 29 May 2002 (29.05.2002) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200
Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventors: **HUANG, Anguo**; 1688 Yale Drive, Mountain
View, CA 94040 (US). **CAIA, Jean-Michel**; 1675 York
Street, San Francisco, CA 94110 (US). **LING, Jing**;
4117 Stevenson Boulevard, #243, Fremont, CA 94538

(US). **CALDERON, Juan-Carlos**; 38144 Miller Place,
Fremont, CA 94536 (US). **JOSHI, Vivek**; 1063 Morse
Avenue, #14-102, Sunnyvale, CA 94089 (US).

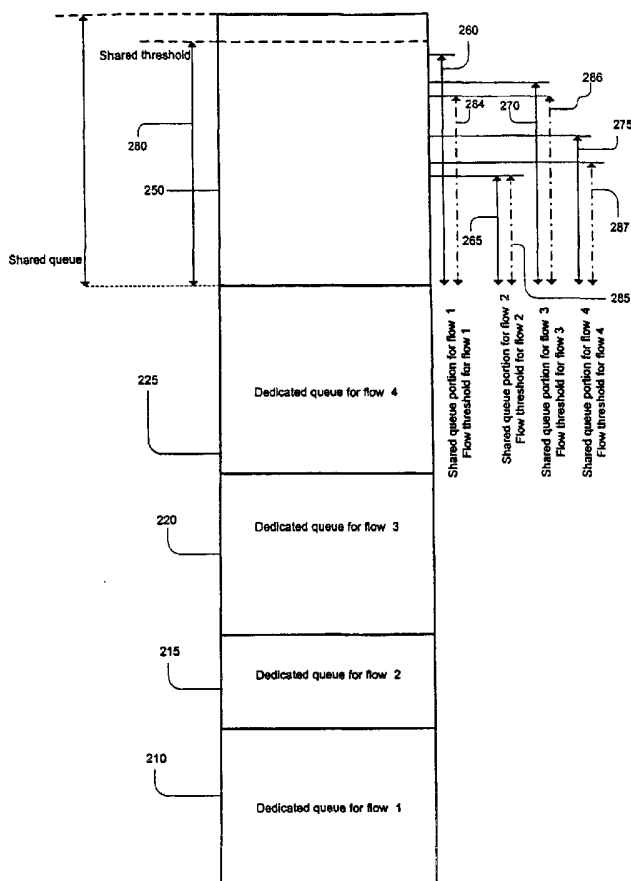
(74) Agents: **HARRIS, Scott, C.** et al.; Fish & Richardson
P.C., 4350 La Jolla Village Drive, Suite 500, San Diego,
CA 92122 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU,
CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH,
GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC,
LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW,
MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD,
SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ,
VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),
Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,

[Continued on next page]

(54) Title: BUFFER MEMORY RESERVATION



(57) Abstract: Network applications may require a guaranteed rate of throughput, which may be accomplished by using buffer memory reservation to manage a data queue used to store incoming packets. Buffer memory reservation reserves a portion of a data queue as a dedicated queue for each flow, reserves another portion of a data queue as a shared queue, and associates a portion of the shared queue with each flow. The amount of the buffer memory reserved by the dedicated queue sizes and the shared queue portion sizes for all of the flows may exceed the amount of physical memory available to buffer incoming packets.

WO 03/103236 A1



ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO,
SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM,
GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

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.

Published:

— *with international search report*

BUFFER MEMORY RESERVATION

BACKGROUND

The following description relates to a digital communication system, and more particularly to a system that includes a high speed packet-switching network that transports packets. High speed packet-switching networks, such as Asynchronous Transfer Mode (ATM), Internet Protocol (IP), and Gigabit Ethernet, support a multitude of connections to different sessions in which incoming packets compete for space in a buffer memory.

Digital communication systems typically employ packet-switching systems that transmit blocks of data called packets. Typically, a message or other set of data to be sent is larger than the size of a packet and must be broken into a series of packets. Each packet consists of a portion of the data being transmitted and control information in a header used to route the packet through the network to its destination.

DESCRIPTION OF DRAWINGS

Fig. 1 is a diagram of a packet-switching network.

Fig. 2 is a diagram of buffer memory used to store incoming packets.

Fig. 3 is a state diagram for a process performed to reserve buffer memory space to store incoming packets.

Figs. 4, 5 and 6 are flow charts illustrating processes for reserving buffer memory space to store incoming packets.

Like reference symbols in the various drawings indicate like elements.

5

DETAILED DESCRIPTION

Fig. 1 shows a typical packet-switching system that includes a transmitting server 110 connected through a communication pathway 115 to a packet-switching network 120 that is connected through a communication pathway 125 to a destination server 130. The transmitting server 110 sends a message through the packet-switching network 120 to the destination server 130 as a series of packets. In the packet-switching network 120, the packets typically pass through a series of servers. As each packet arrives at a server, the server stores the packet briefly in buffer memory before transmitting the packet to the next server. The packet proceeds through the network until it arrives at the destination server 130, which stores the packet briefly in buffer memory 135 as the packet is received.

High-speed packet-switching networks are capable of supporting a vast number of connections (also called flows). Some broadband networks, for example, may support in each line card 256,000 connections through 64 logical ports. Each incoming packet from a flow may be stored in a data

queue in buffer memory upon receipt. If no buffer memory space is available to store a particular packet, the incoming packet is dropped.

Network applications may require a guaranteed rate of throughput, which may be accomplished by using buffer memory reservation to manage a data queue used to store incoming packets. Buffer memory reservation reserves a portion of a data queue as a dedicated queue for each flow, reserves another portion of a data queue as a shared queue, and associates a portion of the shared queue with each flow. The dedicated queue size provided to each flow provides a guaranteed rate of throughput for incoming packets, and the shared queue provides space to buffer packets during periods having peak rates that exceed the guaranteed rate of throughput. The dedicated queue may have one or more reserved memory addresses assigned to it or may be assigned memory space without reference to a particular reserved memory address. Similarly, the shared queue may have one or more reserved memory addresses assigned to it or may be assigned memory space without reference to a particular reserved memory address. The amount of the buffer memory reserved by the dedicated queue portions and the shared queue portion for all of the flows may exceed the amount of physical memory available to buffer incoming packets. However, the amount of buffer memory reserved by the

dedicated queue portions may not exceed the amount of physical memory available to buffer incoming packets.

As shown in Fig. 2, the buffer memory for a data queue 200 used to store incoming packets is apportioned into queues 210, 215, 220, and 225 dedicated to each flow and a shared queue 250. For brevity, Fig. 2 illustrates only a small portion of data queue 200. The portion of the shared queue 250 associated with each flow is shown by arrows 260, 265, 270, 275. Eighty percent of the shared queue size is associated with a first flow in portion 260, forty eight percent of the shared queue size is associated with a second flow in portion 265, seventy-five percent of the shared queue size is associated with a third flow in portion 270, and fifty-five percent of the shared queue size is associated with a fourth flow in portion 275. The sum of the sizes of the dedicated queues 210, 215, 220, 225 and the sizes of the shared queue portions 260, 265, 270, 275 exceeds the amount of physical memory available to store incoming packets.

The unused portion of the data queue 200 may decrease during the time period from when the determination is made that space is available in the data queue to store a particular incoming packet to when the particular incoming packet is stored. Such a decrease in the unused portion of the data queue may prevent the particular incoming packet

from being stored, and may result in the incoming packet being dropped.

A shared threshold 280 that is less than the size of the shared queue may reduce the number of incoming packets that are dropped because of such a data queue increase. The shared threshold 280 may be set to a value that is less than or equal to the size of the shared queue 250, with the actual value of the threshold being selected based on a balance between the likelihood of dropping packets (which increases as the shared threshold increases) and the efficiency with which the shared queue is used (which decreases as the shared threshold decreases). In addition, a flow threshold 284-287 that is less than or equal to the size of the shared queue portion 260, 265, 270, 275 associated with the flow may be set for each flow.

The size of the dedicated queues used in buffer memory reservation implementations may be the same for all flows or may vary between flows. An implementation may use the same flow threshold values for all flows, may vary the flow threshold values between flows, or may use no flow thresholds.

Fig. 3 illustrates a state diagram 300 for execution of buffer memory reservation on a processor. After receiving an incoming packet, the processor may store the incoming packet from a flow in the dedicated queue associated with

the flow (state 310), may store the incoming packet in the shared queue (state 320), or may drop the packet (state 330).

The processor stores the incoming packet from a flow in the dedicated queue associated with the flow (state 310) if space is available in the dedicated queue for the packet (transitions 342, 344, 346). For a particular flow, the processor remains in state 310 (transition 342) until the dedicated queue for the flow is full.

When space is not available in the dedicated queue (transition 348), the incoming packet may be stored in the shared queue (state 320) if space is available in the shared queue portion for the flow and in the shared queue (transition 350). Space must be available both in the shared queue portion for the flow and the shared queue because the physical memory available to the shared queue may be less than the amount of space allocated to the sum of the shared queue portions for all of the flows. When there is no space available to store the incoming packet in the shared queue or the dedicated queue (transitions 354, 356), the incoming packet is dropped from the flow of packets (state 330). The processor continues to drop incoming packets until space becomes available in the shared queue (transition 352) or the dedicated queue (transition 346).

Referring to Fig. 4, a process 400 uses the size of the incoming packet to determine whether space is available in the shared queue portion for a flow. The implementation of the process 400 in Fig. 4 uses a shared threshold for the
5 shared queue that is equal to the size of the shared queue and does not associate a flow threshold with the flow from which the incoming packets are received.

The process 400 begins when a processor receives an incoming packet from a flow (410). The processor determines
10 whether the unused portion of the dedicated queue size for the flow is greater than or equal to the packet size (420). If so, the processor stores the packet in the dedicated queue for the flow (430) and waits to receive another incoming packet from a flow (410).

15 If the processor determines that the unused portion of the dedicated queue size is less than the packet size (e.g., space is not available to store the packet in the dedicated queue for the flow), the processor determines whether the size of the unused portion of the shared queue portion for
20 the flow is greater than or equal to the packet (440), and, if not, drops the packet (450). The packet is dropped because neither the dedicated queue for the flow nor the shared queue portion for the flow has sufficient space available to store the packet. After dropping the packet,

the processor waits to receive another incoming packet (410).

If the processor determines that the size of the unused portion of the shared queue portion for the flow is greater than or equal to the packet size, the processor determines whether the used portion of the shared queue is less than or equal to the shared threshold (460). If so, the processor stores the packet in the shared queue (470) and waits to receive another incoming packet from a flow (410). If the processor determines that the used portion of the shared queue size is greater than the shared threshold, the processor drops the packet (450) and waits to receive an incoming packet from a flow (410).

Referring to Fig. 5, a process 500 uses a flow threshold to determine whether space is available in the shared queue portion for a flow. The process 500 uses a shared threshold for the shared queue that is less than the size of the shared queue and associates with each flow a flow threshold that is less than the size of the shared queue portion associated with the flow.

The process 500 begins when a processor receives an incoming packet from a flow (510), determines whether the dedicated queue for the flow has space available for the packet (520), and, when space is available, stores the incoming packet in the dedicated queue for the flow (530).

If space is not available in the dedicated queue for the flow (520), the processor determines whether the used portion of the shared queue portion is less than or equal to the flow threshold (540). This is in contrast to the
5 implementation described with respect to Fig. 4, where the processor determines whether the shared queue portion has space available based on the size of the incoming packet and does not use a flow threshold.

If the flow threshold is satisfied, the processor
10 determines whether the used portion of the shared queue is less than or equal to the shared threshold (550). The processor stores the packet in the shared queue (560) only if the used portions of the shared queue portion and the shared queue are less than or equal to their respective
15 thresholds. Otherwise, the processor drops the incoming packet (570). The processor then waits for an incoming packet (510) and proceeds as described above.

Referring to Fig. 6, a process 600 assigns a probability of being accepted into the shared queue to a
20 particular received packet and accepts the received packet into the shared queue when the particular packet has a higher probability of being accepted than the probabilities assigned to other incoming packets that are competing for buffer memory space.

The process 600 begins when a processor receives an incoming packet from a flow (610), determines whether the dedicated queue for the flow has space available for the packet (620), and, when space is available, stores the incoming packet in the dedicated queue for the flow (630).
 5 If space to store the packet is not available in the dedicated queue for the flow, the processor determines whether the used portion of the shared queue portion for the flow is less than or equal to the flow threshold (640) and
 10 determines whether the used portion of the shared queue is less than or equal to the shared threshold (650). Based on those determinations, the processor may drop the packet or store the packet in the shared queue as set forth in the table below.

Used portion of the shared queue portion less than or equal to flow threshold	Used portion of the shared queue less than or equal to shared threshold	Assign probability to packet (optional)	Storage result
Yes	Yes		Store packet in shared queue
Yes	No	Assign higher probability to packet	Store packet in shared queue if packet probability is higher than competing packets; else drop packet
No	Yes	Assign lower probability to packet	Store packet in shared queue if packet probability is higher than competing packets; else drop packet
No	No		Drop packet

15

The packet is dropped (660) when the used portion of the shared queue portion is greater than the flow threshold

and the used portion of the shared queue is greater than the shared threshold.

The packet is stored in the shared queue (670) when the used portion of the shared queue portion is less than or equal to the flow threshold and the used portion of the shared queue size is less than or equal to the shared threshold.

If neither of those two conditions exist, the processor assigns the packet a higher probability of being stored in the shared queue (680) when the used portion of the shared queue portion is less than or equal to the flow threshold and the used portion of the shared queue is greater than the shared threshold. The processor assigns the packet a lower probability of being stored in the shared queue (685) when the used portion of the shared queue portion is greater than the flow threshold and the used portion of the shared queue is less than or equal to the shared threshold. The processor then determines whether the probability assigned to the packet is greater than the probability assigned to other incoming packets that are competing for buffer memory space (690). If so, the processor stores the packet in the shared queue (670; otherwise, the packet is dropped (660).

Buffer memory reservation helps to provide a guaranteed rate of throughput for incoming packets and to avoid buffer congestion. Buffer memory reservation techniques provide a

variety of parameters that can be used to manage the network application, including a shared threshold, a flow threshold for each flow, a dedicated queue for each flow, a shared queue, and a shared queue portion for each flow. Some
5 implementations may predesignate parameters, while other implementations may vary the parameters based on current network conditions.

The benefits of buffer memory reservation for packet applications are applicable to other implementations of
10 packet-switching networks that use fixed-length or variable-length packets.

Implementations may include a method or process, an apparatus or system, or computer software on a computer medium. It will be understood that various modifications
15 may be made without departing from the spirit and scope of the following claims. For example, advantageous results still could be achieved if steps of the disclosed techniques were performed in a different order and/or if components in the disclosed systems were combined in a different manner
20 and/or replaced or supplemented by other components.

WHAT IS CLAIMED IS:

1. A buffer memory management method for a packet-switching application, the method comprising:

associating each of a plurality of flows of packets with a dedicated queue and a particular portion of a shared queue to provide a size of a combination of the dedicated
5 queues and the shared queue portions for all of the flows exceeding an amount of physical memory available to buffer packets, and

accepting a particular packet from a particular flow of
10 packets into the dedicated queue associated with the particular flow if a size of an unused portion of the dedicated queue associated with the particular flow is greater than or equal to a size of the particular packet.

2. The method of claim 1 wherein the size of the
15 dedicated queue varies for different flows.

3. The method of claim 1 wherein the size of the dedicated queue is the same for all flows.

4. The method of claim 1 further comprising:
setting a shared threshold that is less than or equal
20 to a size of the shared queue, and

accepting a particular packet from a particular flow of packets into the shared queue if the particular packet is not accepted by the dedicated queue associated with the particular flow, a size of an unused portion of the shared queue portion associated with the particular flow is greater
5 than or equal to the size of the particular packet, and a size of a used portion of the shared queue is less than or equal to the shared threshold.

5. The method of claim 4 further comprising dropping
10 a particular packet from a particular flow of packets if the particular packet is not accepted by the dedicated queue associated with the particular flow and a size of a used portion of the shared queue is greater than the shared threshold.

15 6. The method of claim 4 further comprising dropping a particular packet from the particular flow of packets if the particular packet is not accepted by the dedicated queue associated with the particular flow and the size of the unused portion of the shared queue portion associated with
20 the particular flow is less than the size of the particular packet.

7. The method of claim 4 further comprising:

associating each flow of packets with a flow threshold,
and

dropping a particular packet from the particular flow
of packets if the particular packet is not accepted by the
5 dedicated queue associated with the particular flow and a
size of the used portion of the shared queue portion
associated with the particular flow is greater than the flow
threshold associated with the particular flow.

8. The method of claim 1 further comprising:

10 associating each received packet with a probability of
being accepted into the shared queue,

accepting a particular packet from a particular flow of
packets into the shared queue if the particular packet is
not accepted by the dedicated queue associated with the
15 particular flow and the probability associated with the
particular packet is greater than the probability associated
with one or more other received packets that have not been
accepted by the dedicated queues associated with the flows
of the received packets, and

20 dropping a particular packet from a particular flow of
packets if the particular packet is not accepted into the
dedicated queue associated with the particular flow and the
particular packet is not accepted into the shared queue.

9. The method of claim 8, wherein the shared threshold is less than the size of the shared queue, the method further comprising:

associating each flow of packets with a flow threshold;

5 associating a particular packet from a particular flow of packets with a first probability if:

the particular packet is not accepted by the dedicated queue associated with the particular flow,

10 the size of the used portion of the shared queue is greater than the shared threshold, and

the size of the used portion of the shared queue portion is less than or equal to the flow threshold associated with a particular flow; and

15 associating a particular packet from a particular flow of packets with a second probability if:

the particular packet is not accepted by the dedicated queue associated with the particular flow,

the size of the used portion of the shared queue is less than or equal to the shared threshold, and

20 the size of the used portion of the shared queue portion is greater than the flow threshold associated with the particular flow;

wherein the first probability is less than the second probability.

10. A computer readable medium or propagated signal having embodied thereon a computer program configured to cause a processor to implement buffer memory management for a packet-switching application, the computer program
5 comprising code segments for causing a processor to:

associate each of a plurality of flows of packets with a dedicated queue and a particular portion of a shared queue to provide a size of a combination of the dedicated queues and the shared queue portions for all of the flows exceeding
10 an amount of physical memory available to buffer packets,
and

accept a particular packet from a particular flow of packets into the dedicated queue associated with the particular flow if a size of an unused portion of the
15 dedicated queue associated with the particular flow is
greater than or equal to a size of the particular packet.

11. The medium of claim 10 wherein the size of the dedicated queue varies for different flows.

12. The medium of claim 10 wherein the size of the
20 dedicated queue is the same for all flows.

13. The medium of claim 10 further comprising code segments for causing a processor to:

set a shared threshold that is less than or equal to the shared queue size, and

accept a particular packet from a particular flow of packets into the shared queue if the particular packet is not accepted by the dedicated queue associated with the particular flow, a size of an unused portion of the shared queue portion associated with the particular flow is greater than or equal to the size of the particular packet, and a size of a used portion of the shared queue is less than or equal to the shared threshold.

14. The medium of claim 13 further comprising code segments for causing a processor to drop a particular packet from a particular flow of packets if the particular packet is not accepted by the dedicated queue associated with the particular flow and a size of a used portion of the shared queue is greater than the shared threshold.

15. The medium of claim 13 further comprising code segments for causing a processor to drop a particular packet from the particular flow of packets if the particular packet is not accepted by the dedicated queue associated with the particular flow and the size of the unused portion of the shared queue portion associated with the particular flow is less than the size of the particular packet.

16. The medium of claim 13 further comprising code segments for causing a processor to:

associate each flow of packets with a flow threshold, and

5 drop a particular packet from the particular flow of packets if the particular packet is not accepted by the dedicated queue associated with the particular flow and a size of the used portion of the shared queue portion associated with the particular flow is greater than the flow
10 threshold associated with the particular flow.

17. The medium of claim 10 further comprising code segment for causing a processor to:

associate each received packet with a probability of being accepted into the shared queue,

15 accept a particular packet from a particular flow of packets into the shared queue if the particular packet is not accepted by the dedicated queue associated with the particular flow and the probability associated with the particular packet is greater than the probability associated
20 with one or more other received packets that have not been accepted by the dedicated queues associated with the flows of the received packets, and

drop a particular packet from a particular flow of packets if the particular packet is not accepted into the dedicated queue associated with the particular flow and the particular packet is not accepted into the shared queue.

5 18. The medium of claim 17, wherein the shared threshold is less than the shared queue size, the medium further comprising code segments for causing a processor to:
associate each flow of packets with a flow threshold;
associate a particular packet from a particular flow of
10 packets with a first probability if:

the particular packet is not accepted by the dedicated queue associated with the particular flow,
the size of the used portion of the shared queue is greater than the shared threshold, and

15 the size of the used portion of the shared queue portion is less than or equal to the flow threshold associated with a particular flow; and

associate a particular packet from a particular flow of packets with a second probability if:

20 the particular packet is not accepted by the dedicated queue associated with the particular flow,
the size of the used portion of the shared queue is less than or equal to the shared threshold, and

the size of the used portion of the shared queue portion is greater than the flow threshold associated with the particular flow;

wherein the first probability is less than the second
5 probability.

19. An apparatus for buffer memory management in a packet-switching application, the apparatus including a processor and memory connected to the processor, wherein the processor comprises one or more components to:

10 associate each of a plurality of flows of packets with a dedicated queue, and a particular portion of a shared queue to provide a size of a combination of the dedicated queues and the shared queue portions for all of the flows exceeding an amount of physical memory available to buffer
15 packets, and

accept a particular packet from a particular flow of packets into the dedicated queue associated with the particular flow if a size of an unused portion of the dedicated queue associated with the particular flow is
20 greater than or equal to a size of the particular packet.

20. The apparatus of claim 19 wherein the size of the dedicated queue varies for different flows.

21. The apparatus of claim 19 wherein the size of the dedicated queue is the same for all flows.

22. The apparatus of claim 19, the processor being further comprises one or more components to:

5 set a shared threshold that is less than or equal to a size of the shared queue, and

 accept a particular packet from a particular flow of packets into the shared queue if the particular packet is not accepted by the dedicated queue associated with the
10 particular flow, a size of an unused portion of the shared queue portion associated with the particular flow is greater than or equal to the size of the particular packet, and a size of a used portion of the shared queue is less than or equal to the shared threshold.

15 23. The apparatus of claim 22, the processor being further comprising one or more components to drop a particular packet from a particular flow of packets if the particular packet is not accepted by the dedicated queue associated with the particular flow and a size of a used
20 portion of the shared queue is greater than the shared threshold.

24. The apparatus of claim 22, the processor being further comprising one or more components to drop a particular packet from the particular flow of packets if the particular packet is not accepted by the dedicated queue
5 associated with the particular flow and the size of the unused portion of the shared queue portion associated with the particular flow is less than the size of the particular packet.

25. The apparatus of claim 22, the processor further
10 comprising one or more components to:

associate each flow of packets with a flow threshold, and

drop a particular packet from the particular flow of packets if the particular packet is not accepted by the
15 dedicated queue associated with the particular flow and a size of the used portion of the shared queue portion associated with the particular flow is greater than the flow threshold associated with the particular flow.

26. The apparatus of claim 19, the processor further
20 comprising one or more components to:

associate each received packet with a probability of being accepted into the shared queue,

accept a particular packet from a particular flow
of packets into the shared queue if the particular packet is
not accepted by the dedicated queue associated with the
particular flow and the probability associated with the
5 particular packet is greater than the probability associated
with one or more other received packets that have not been
accepted by the dedicated queues associated with the flows
of the received packets, and

drop a particular packet from a particular flow of
10 packets if the particular packet is not accepted into the
dedicated queue associated with the particular flow and the
particular packet is not accepted into the shared queue.

27. The apparatus of claim 26, wherein the shared
threshold is less than the size of the shared queue, the
15 processor being further comprising one or more components
to:

associate each flow of packets with a flow
threshold;

associate a particular packet from a particular
20 flow of packets with a first probability if:

the particular packet is not accepted by the
dedicated queue associated with the particular flow,

the size of the used portion of the shared
queue is greater than the shared threshold, and

the size of the used portion of the shared queue portion is less than or equal to the flow threshold associated with a particular flow; and

associate a particular packet from a particular
5 flow of packets with a second probability if:

the particular packet is not accepted by the dedicated queue associated with the particular flow,

the size of the used portion of the shared queue is less than or equal to the shared threshold, and

10 the size of the used portion of the shared queue portion is greater than the flow threshold associated with the particular flow;

wherein the first probability is less than the second probability.

15 28. A system for buffer memory management in a packet-switching application, the system comprising:

a traffic management device;

a port coupled to a transmission channel; and

a link between the traffic management device and the
20 port,

wherein the traffic management device is comprised of one or more components to:

associate each of a plurality of flows of packets with a dedicated queue and a particular portion of a shared queue

to provide a size of a combination of the dedicated queues and the shared queue portions for all of the flows exceeding an amount of physical memory available to buffer packets, and

5 accept a particular packet from a particular flow of packets into the dedicated queue associated with the particular flow if a size of an unused portion of the dedicated queue associated with the particular flow is greater than or equal to a size of the particular packet.

10 29. The system of claim 28 wherein the traffic management device is further comprised of one or more components to:

 set a shared threshold that is less than or equal to a size of the shared queue, and

15 accept a particular packet from a particular flow of packets into the shared queue if the particular packet is not accepted by the dedicated queue associated with the particular flow, a size of an unused portion of the shared queue portion associated with the particular flow is greater
20 than or equal to the size of the particular packet, and a size of a used portion of the shared queue is less than or equal to the shared threshold.

30. The system of claim 29 wherein the traffic management device is further comprised of one or more components to:

drop a particular packet from a particular flow of
5 packets if the particular packet is not accepted by the dedicated queue associated with the particular flow and a size of a used portion of the shared queue is greater than the shared threshold, and

drop a particular packet from the particular flow of
10 packets if the particular packet is not accepted by the dedicated queue associated with the particular flow and the size of the unused portion of the shared queue portion associated with the particular flow is less than the size of the particular packet.

100

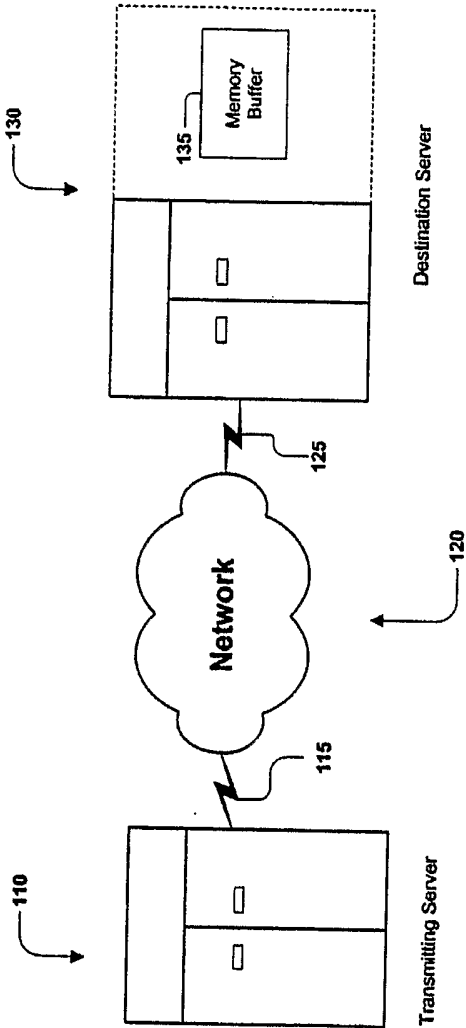


Fig. 1

200

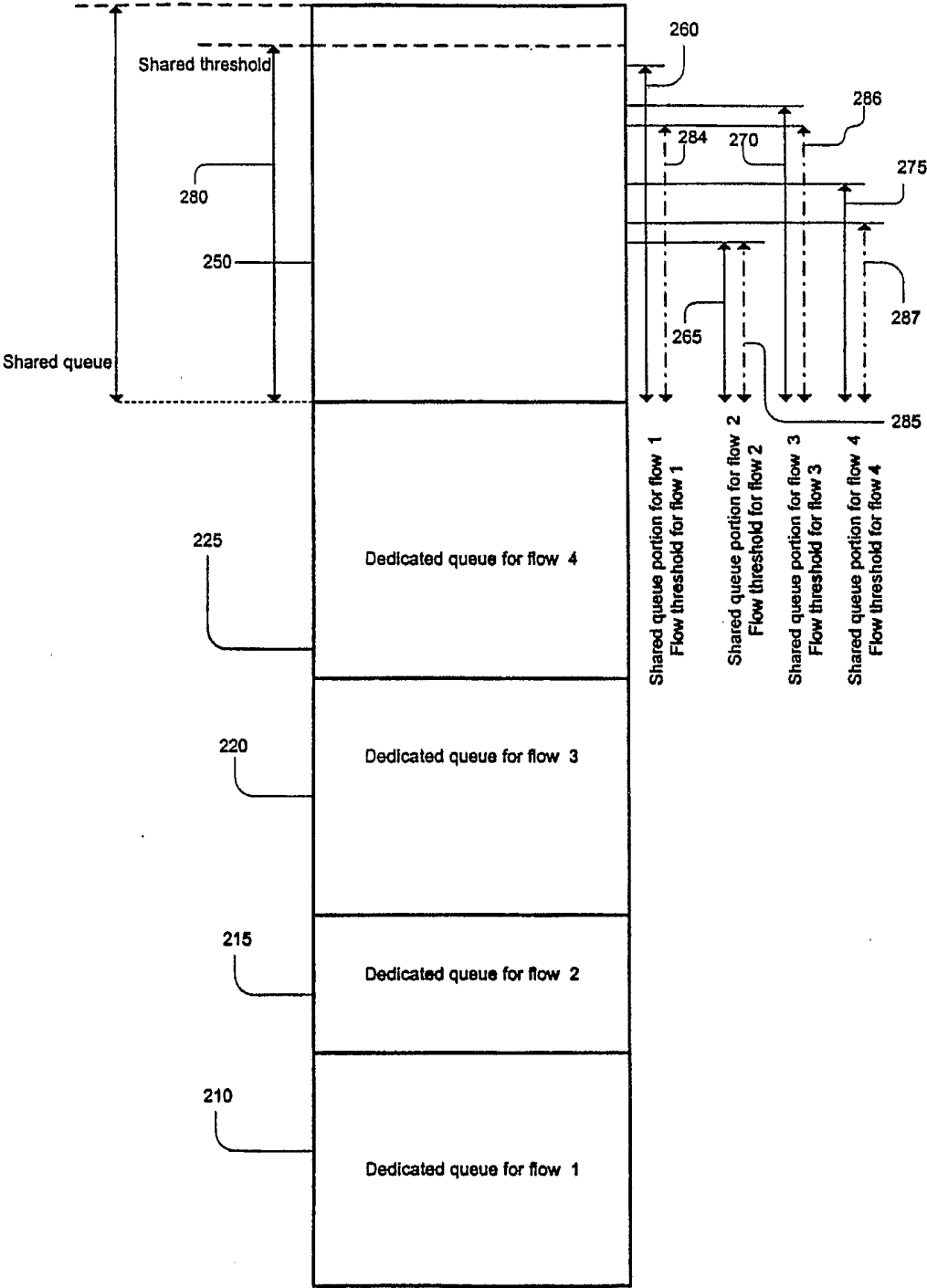


Fig. 2

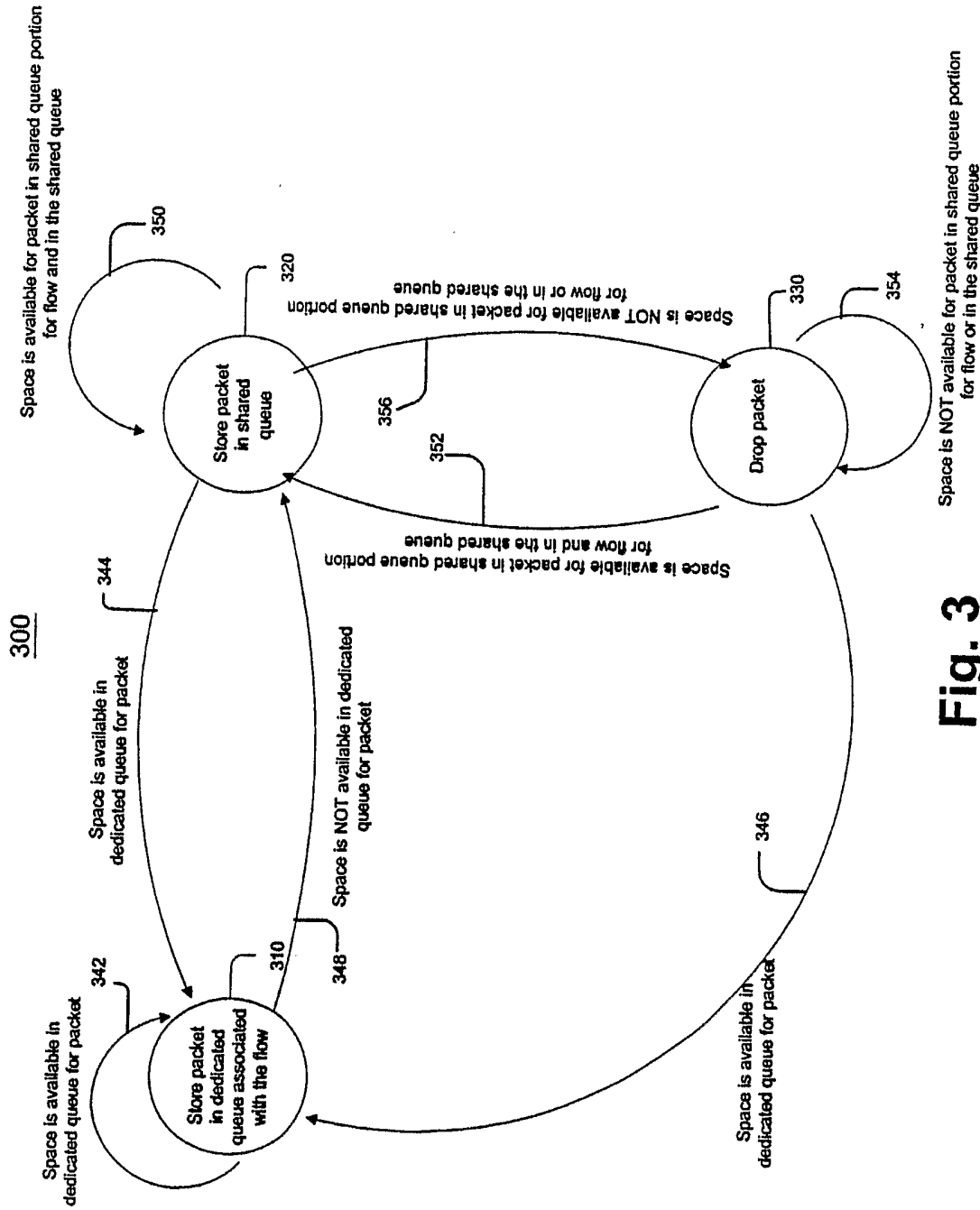
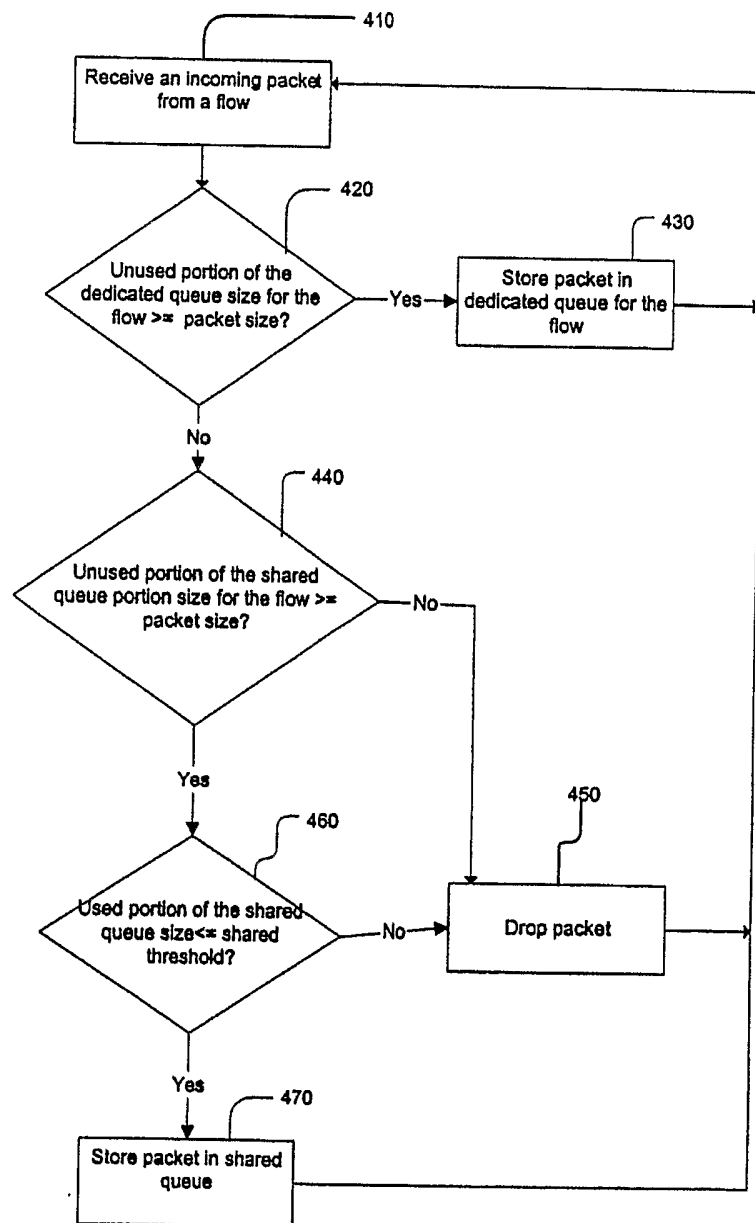
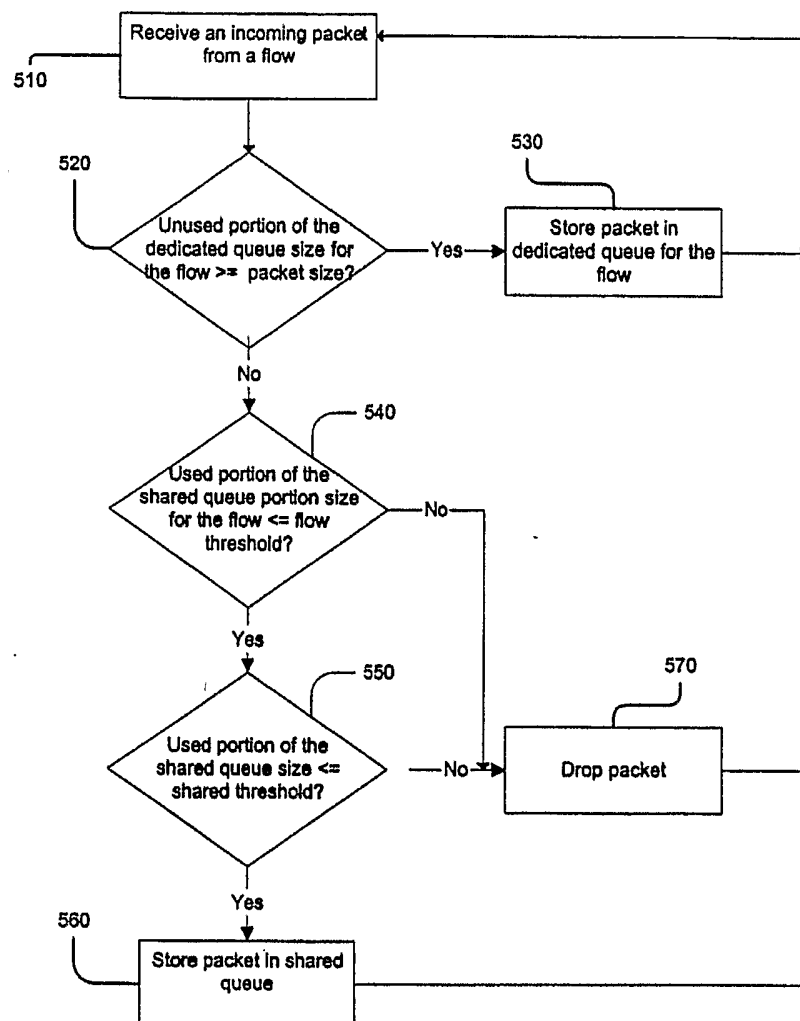


Fig. 3

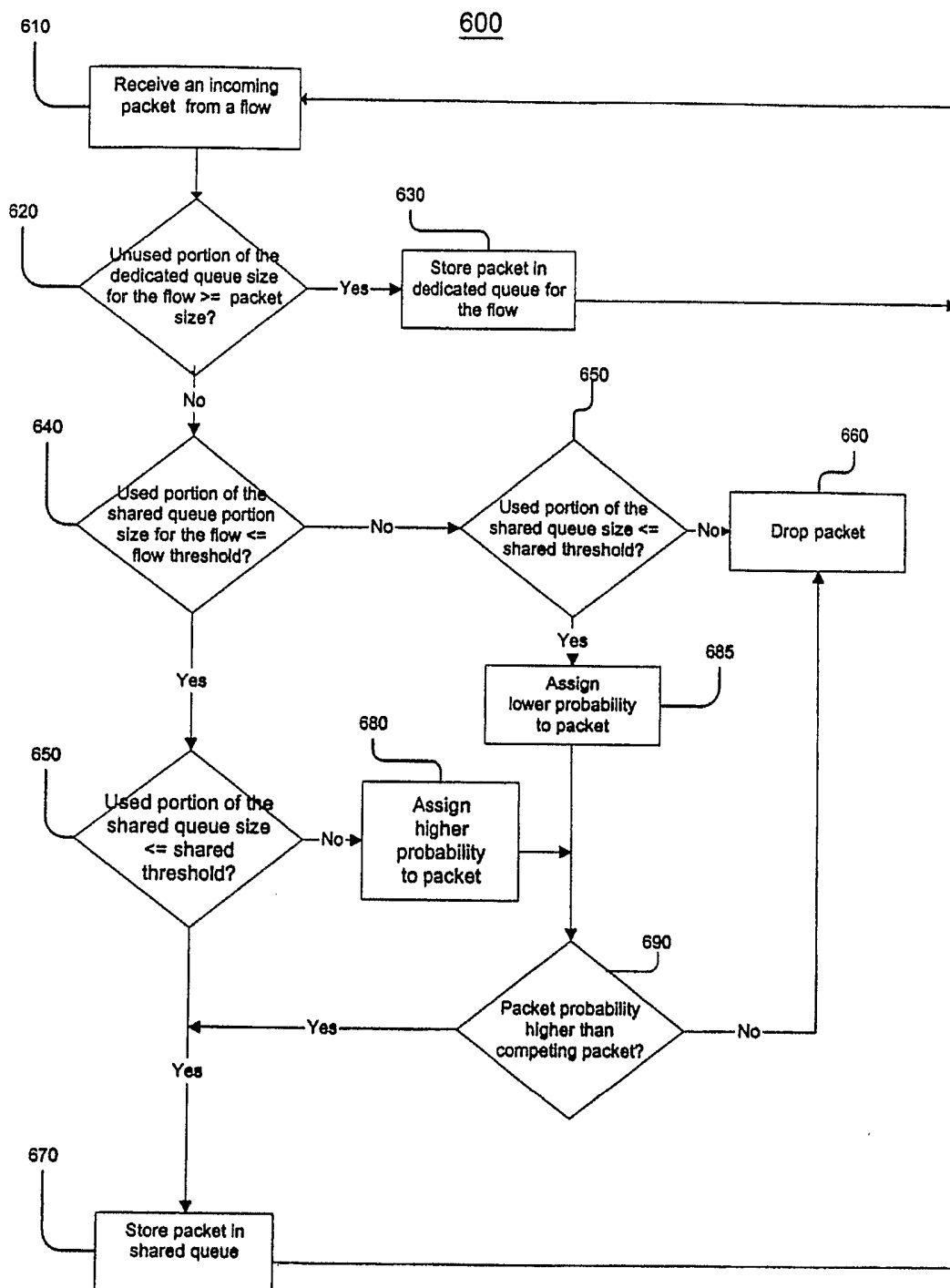
4/6

400**Fig. 4**

5/6

500**Fig. 5**

6/6

**Fig. 6**

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 03/15729

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/56

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

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)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 867 663 A (CALDARA STEPHEN A ET AL) 2 February 1999 (1999-02-02) column 2, line 11 -column 9, line 57 figure 2	1-30
Y	EP 0 706 298 A (AT & T CORP) 10 April 1996 (1996-04-10) figure 3	1, 10, 19, 28
A	page 2, line 36 -page 5, line 12	2-9, 11-18, 20-27, 29, 30
	----- -/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier 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

17 September 2003

Date of mailing of the international search report

25/09/2003

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

Oteo Mayayo, C

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 03/15729

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 01 63858 A (MITEL SEMICONDUCTOR V N INC) 30 August 2001 (2001-08-30) page 3, line 1 -page 3, line 10	1, 10, 19, 28
A	page 8, line 15 -page 12, line 22	2-9, 11-18, 20-27, 29, 30
A	----- WO 97 03549 A (CALDARA STEPHEN A ;ASCOM NEXION INC (US); HAUSER STEPHEN A (US); M) 6 February 1997 (1997-02-06) the whole document	1-30
A	----- LIN A Y M ET AL: "PRIORITY QUEUEING STRATEGIES AND BUFFER ALLOCATION PROTOCOLS FOR TRAFFIC CONTROL AT AN ATM INTEGRATED BROADBAND SWITCHING SYSTEM" IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE INC. NEW YORK, US, vol. 9, no. 9, 1 December 1991 (1991-12-01), pages 1524-1536, XP000267541 ISSN: 0733-8716 the whole document -----	1-30

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 03/15729

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 5867663	A	02-02-1999	AU 6500796 A	18-02-1997
			AU 6500896 A	18-02-1997
			AU 6500996 A	18-02-1997
			AU 6501096 A	18-02-1997
			AU 6501496 A	18-02-1997
			AU 6501696 A	18-02-1997
			AU 6501796 A	18-02-1997
			AU 6501996 A	18-02-1997
			AU 6502096 A	18-02-1997
			AU 6502496 A	18-02-1997
			AU 6502596 A	18-02-1997
			AU 6502696 A	18-02-1997
			AU 6502796 A	18-02-1997
			AU 6503196 A	18-02-1997
			AU 6503296 A	18-02-1997
			AU 6503396 A	18-02-1997
			AU 6503496 A	18-02-1997
			AU 6503596 A	18-02-1997
			AU 6503696 A	18-02-1997
			AU 6503796 A	18-02-1997
			AU 6549196 A	18-02-1997
			AU 6549296 A	18-02-1997
			AU 6648496 A	18-02-1997
			AU 6648796 A	18-02-1997
			AU 6712496 A	18-02-1997
			AU 6712596 A	18-02-1997
			AU 6761896 A	18-02-1997
			AU 6762096 A	18-02-1997
			EP 0845181 A1	03-06-1998
			EP 0839420 A1	06-05-1998
			EP 0839419 A2	06-05-1998
			EP 0872086 A1	21-10-1998
			EP 0839421 A2	06-05-1998
			EP 0839422 A1	06-05-1998
			JP 11510323 T	07-09-1999
			JP 2000501897 T	15-02-2000
			JP 11510324 T	07-09-1999
			JP 2000501900 T	15-02-2000
			JP 2000501901 T	15-02-2000
			JP 2002516038 T	28-05-2002
			JP 2001519973 T	23-10-2001
			JP 11510003 T	31-08-1999
			JP 2000501902 T	15-02-2000
			JP 11510004 T	31-08-1999
			JP 11510005 T	31-08-1999
			JP 11510006 T	31-08-1999
			JP 11511303 T	28-09-1999
			JP 11510007 T	31-08-1999
			JP 11510008 T	31-08-1999
EP 0706298	A	10-04-1996	US 5541912 A	30-07-1996
			CA 2156654 A1	05-04-1996
			EP 0706298 A2	10-04-1996
			JP 3347926 B2	20-11-2002
			JP 8251233 A	27-09-1996
WO 0163858	A	30-08-2001	AU 3704301 A	03-09-2001
			CN 1416633 T	07-05-2003

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 03/15729

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0163858	A	EP 1258115 A1	20-11-2002
		TW 490964 B	11-06-2002
		WO 0163858 A1	30-08-2001
		US 2001051992 A1	13-12-2001
WO 9703549	A 06-02-1997	AU 6500796 A	18-02-1997
		AU 6500896 A	18-02-1997
		AU 6500996 A	18-02-1997
		AU 6501096 A	18-02-1997
		AU 6501496 A	18-02-1997
		AU 6501696 A	18-02-1997
		AU 6501796 A	18-02-1997
		AU 6501996 A	18-02-1997
		AU 6502096 A	18-02-1997
		AU 6502496 A	18-02-1997
		AU 6502596 A	18-02-1997
		AU 6502696 A	18-02-1997
		AU 6502796 A	18-02-1997
		AU 6503196 A	18-02-1997
		AU 6503296 A	18-02-1997
		AU 6503396 A	18-02-1997
		AU 6503496 A	18-02-1997
		AU 6503596 A	18-02-1997
		AU 6503696 A	18-02-1997
		AU 6503796 A	18-02-1997
		AU 6549196 A	18-02-1997
		AU 6549296 A	18-02-1997
		AU 6648496 A	18-02-1997
		AU 6648796 A	18-02-1997
		AU 6712496 A	18-02-1997
		AU 6712596 A	18-02-1997
		AU 6761896 A	18-02-1997
		AU 6762096 A	18-02-1997
		EP 0845181 A1	03-06-1998
		EP 0839420 A1	06-05-1998
		EP 0839419 A2	06-05-1998
		EP 0872086 A1	21-10-1998
		EP 0839421 A2	06-05-1998
		EP 0839422 A1	06-05-1998
		JP 11510323 T	07-09-1999
		JP 2000501897 T	15-02-2000
		JP 11510324 T	07-09-1999
		JP 2000501900 T	15-02-2000
		JP 2000501901 T	15-02-2000
		JP 2002516038 T	28-05-2002
		JP 2001519973 T	23-10-2001
		JP 11510003 T	31-08-1999
		JP 2000501902 T	15-02-2000
		JP 11510004 T	31-08-1999
		JP 11510005 T	31-08-1999
		JP 11510006 T	31-08-1999
		JP 11511303 T	28-09-1999
		JP 11510007 T	31-08-1999
		JP 11510008 T	31-08-1999
		JP 11510009 T	31-08-1999