(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

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

PCT

(10) International Publication Number
WO 03/103197 A1

(51) International Patent Classification⁷: H04J 3/02, H04Q 11/04, G06F 13/00

(21) International Application Number: PCT/US03/12482

(22) International Filing Date: 23 April 2003 (23.04.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/158,038 30 May 2002 (30.05.2002) US

(71) Applicant: **MOTOROLA, INC.** [US/US]; 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

(72) Inventors: **STOGNER, Darrell, James**; 1870 N.W. 107th Avenue, Plantation, FL 33322 (US). **CONRAD, Alan, P.**; 3106 Charlemagne Lane, St. Charles, IL 60174 (US). **LAMPERT, Chet, A.**; 11251 N.W. 8th Street, Plantation, FL 33325 (US). **LOGALBO, Robert, D.**; 2127 Seaver Lane, Hoffman Estates, IL 60194 (US).

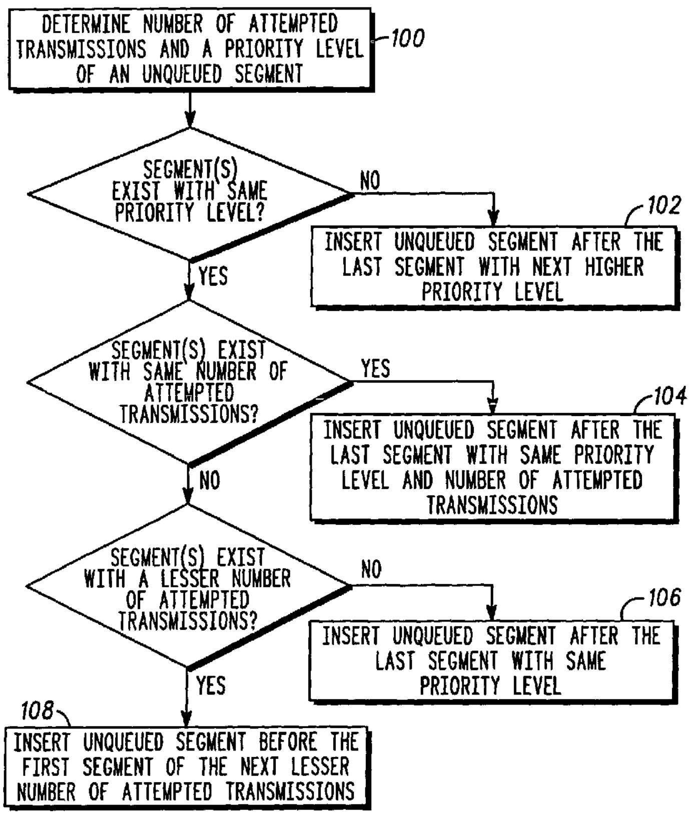
(74) Agents: **DOUTRE, Barbara, R.** et al.; 8000 West Sunrise Blvd., Room 1610, Fort Lauderdale, FL 33322 (US).

(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, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO,

[Continued on next page]

(54) Title: METHODS FOR SEQUENCING DATAGRAM TRANSMISSIONS



(57) Abstract: A method for sequencing datagram transmissions, including, receiving an unqueued segment to be enqueued in a queue, determining a priority level and a number of attempted transmissions for the unqueued segment (100), inserting unqueued segment after the last segment with next higher priority level if there is no segment exists with same priority level (102), inserting unqueued segment after the last segment with same priority level and number of attempted transmissions (104),), inserting unqueued segment after the last segment with same priority level if there is segment exists with no less number of attempted transmissions (106), inserting unqueued segment before the first segment of next lesser number of attempted transmissions with same priority level if there is segment exists with less number of attempted transmissions (108).

WO 03/103197 A1

SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

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.

METHODS FOR SEQUENCING DATAGRAM TRANSMISSIONS

Field of the Invention

The present invention relates generally to methods for sequencing datagram transmissions.

Background of the Invention

Various network protocols have been developed to allow the transfer of information, such as voice, video or application data, between endpoints (either network infrastructure or devices subscribed to the network). In all cases, the endpoints are given a finite amount of resources to accomplish their information transfer. Scheduling is the process by which the endpoint determines the order in which information multiplexed from many sources is sent over the transport.

An example is a cellular phone designed to use the IS-136 or GSM protocol, which is given a fixed number of timeslots per second to transmit audio. In this case, the required throughput is known *a priori* and sufficient bandwidth is guaranteed to support the lone data source. The scheduler simply transfers audio packets on a first in, first out basis without fear that packets will timeout (i.e., expire because they remained unsent for their allotted lifetime).

Another example is the packet data component of the TETRA protocol. In this case, each service context establishes its own private link. These links might provide a different quality of service (e.g., a specific priority level, confirmed or unconfirmed transfer, etc.). Again, in this case, since the data on a link is of the same class, it can be scheduled using a simple first in, first out basis.

As customer needs expand to high data rate applications, networks will evolve towards more universal protocols that can support multiple service types at the same time. A common endpoint in such a network might be supporting a video call with associated audio, and at the same time, transferring files or downloading web pages over the Internet. Efficient designs will forgo the wasteful setup of multiple links or contexts and instead multiplex all information over one, common connection.

The joining of all information sources onto a common connection will introduce numerous problems that have yet to be encountered. For example, packets on a link are typically identified using a finite number sequence that wraps around in a modulo manner. When confirmed and unconfirmed packets share the same link, the opportunity arises for a confirmed packet to block the transmission of a rapid arrival unconfirmed packet stream (e.g., from an audio or video CODEC). That is, the packet number of the fast arriving unconfirmed packets might wrap around to the packet number held by the confirmed packet, which has yet to resolve. This might happen due to the confirmed packet being a low priority or requiring multiple attempts to successfully transfer.

In previous systems, such ambiguity was not an issue. A technique common to the art and known as windowing was used to limit the number of packets in flight on a confirmed link at any given moment. Packets on an unconfirmed link could be sent in order to prevent any packet number wrapping. It is a necessity that any new protocol also be capable of avoiding packet number ambiguity. Other desirable traits would be to minimize throughput delay, guarantee the quality of service and provide support for as many quality levels as needed.

Thus, there exists a need for sequencing datagram transmissions.

Brief Description of the Figures

A preferred embodiment of the invention is now described, by way of example only, with reference to the accompanying figures in which:

FIG. 1 illustrates a flowchart of the enqueuing rules in accordance with the present invention;

FIG. 2 illustrates an example of placing an unqueued segment into a queue, wherein the unqueued segment has a different priority level than the other segments previously entered in the queue, in accordance with the present invention;

FIG. 3 illustrates an example of placing an unqueued segment into the queue, wherein the unqueued segment has the same priority level and number of attempted transmission of another segment in the queue, in accordance with the present invention;

FIG. 4 illustrates an example of placing an unqueued segment into the queue, wherein the unqueued segment has the same priority level of another segment in the queue, however, has a higher number of attempted transmissions, in accordance with the present invention;

FIG. 5 illustrates an example of placing an unqueued segment into the queue, wherein the unqueued segment has the same priority level of another segment in the queue, however, has a lower number of attempted transmissions, in accordance with the present invention; and

FIG. 6 illustrates an example of placing an unqueued segment into the queue, wherein the unqueued segment causes an interruption of the transmission of an unconfirmed datagram, in accordance with the present invention.

Detailed Description of the Preferred Embodiment

The present invention utilizes at least one queue to sequence datagram transmissions. A datagram is a grouping of information (e.g., an IP packet, an ATM cell, a frame relay frame, etc.). The datagram is typically divided into segments small enough to fit over the transport, although in some cases, a segment may contain one or more whole datagrams. Preferably, all segments for a datagram are queued at the same time; all segments for a selective retry, however, are not necessarily queued together. The present invention classifies datagrams as confirmed or unconfirmed. The transmitter of a confirmed datagram requires acknowledgement/negative acknowledgment of each segment belonging to the confirmed datagram from the receiver of the segment; as a result, individual confirmed segments that were not successfully received at the receiver may be retransmitted. The transmitter of the unconfirmed datagram, however, does not require acknowledgement/negative acknowledgment of the unconfirmed segments from the receiver of the segments; as a result, the preferred embodiment of the present invention assumes that unconfirmed datagrams are transmitted in their entirety once the first segment belonging to the unconfirmed datagram has been transmitted; however, the present invention is not limited to this assumption (as depicted in an example below). It should also be noted that unconfirmed datagrams could be re-transmitted under a guise of a different identifier (as explained below in the enqueuing rules).

In the preferred embodiment of the present invention, segments are enqueued based on priority, and within a priority level, the segments with a higher number of attempted transmissions have precedence over the segments with a lower number of attempted transmissions; typically, segments are dequeued when the segment is transmitted or discarded. Enqueuing is the act of taking segments and placing them into at least one queue according to at least a portion of a set of enqueuing rules of the present invention (described in detail below); dequeuing is the act of removing segments from at least one queue in any fashion or for any reason. In the preferred embodiment of the present invention, the act of enqueuing segments is performed by the logical link control layer ("LLC"), however, enqueuing can be performed by other components as well. Optionally, the type of service provided can be requested (e.g., maximum reliability, minimum delay, etc.).

As shown in FIG. 1, in the preferred embodiment, when a segment is to be enqueued, the number of attempted transmissions and the priority level of the unqueued segment is determined 100. The priority level implies the position of the segment in the queue; in other words, the higher priority segments (e.g., priority level 1) are transmitted sooner than the lower priority segments (e.g., priority level 3). Once the unqueued segment is received by the LLC, the rules to place the unqueued segment into at least one queue are as follows:

1a. If no segment in the queue exists with the same priority level as the unqueued segment, the unqueued segment is inserted after the last segment of all segments in the queue having a higher priority level 102;

1b. If a segment in the queue exists with the same priority level and the same number of attempted transmissions as the unqueued segment, the unqueued segment is inserted after the last segment having the same priority level and the same number of attempted transmissions 104;

1c. If a segment in the queue exist with the same priority level, and without the same number of attempted transmissions, but with a greater number of attempted transmissions than the unqueued segment, the unqueued segment is inserted immediately after all of the segments having the same priority level and a greater number of attempted transmissions 106;

1d. If a segment in the queue exist with the same priority level, and without the same number of attempted transmissions, but with a lesser number of attempted transmissions than the unqueued segment, the unqueued segment is inserted immediately before all of the segments having the same priority level and a lesser number of attempted transmissions 108;

2. if the unqueued segment is enqueued in front of an unconfirmed datagram, and at least one segment of the unconfirmed datagram has been transmitted before all the segments of the unconfirmed datagram have been transmitted, one of three sub-rules must apply:

a. the remaining segments of the interrupted unconfirmed datagram in the queue are discarded;

b. the remaining segments of the interrupted unconfirmed datagram in the queue are transmitted; or

c. the entire unconfirmed datagram is re-enqueued at the same location in the queue as the partially transmitted datagram, but with a different identifier.

It should be obvious to a person of ordinary skill in the art that the enqueuing rules described above, or any portion thereof, does not need to be implemented, or additional rules may be added, and still remain within the spirit and scope of the present invention. For example, the following additional rules may be added:

- if an unqueued unconfirmed segment is to be enqueued with the same priority level as a confirmed segment previously enqueued for its first attempted transmission, the unconfirmed segment is enqueued in front of the previously enqueued confirmed segment (i.e., precedence is given to an unconfirmed segment over a confirmed segment having the same priority level and number of attempted transmissions, or vice versa);
- queue overflow may be addressed by discarding segments previously queued (e.g., segment(s) with the lowest priority level and the lowest number of

attempted transmissions, segments that are time-sensitive and that have remained in the queue longer than other segments that are time-sensitive, etc.); and/or

- transmitting segments of a datagram multiple times for reliability, and scheduling multiple transmission attempts of a segment/datagram sequentially; for example, with a two segment unconfirmed datagram requiring two transmission attempts, examples of sequential transmission attempts include, but are not limited to, segment 1, segment 1, segment 2, segment 2, or segment 1, segment 2, segment 1 segment 2.

Let us now turn our attention to examples of the enqueuing rules of the present invention. The following examples illustrated in FIGS. 2-6 utilize a single queue for enqueuing all segments. Other embodiments of the present invention, however, may utilize a plurality of queues to enqueue the segments (e.g., a separate queue for each priority level, etc.). Further, FIGS. 2-5 will focus on enqueuing segments belonging to datagram E (i.e., the unqueued segments) according to the enqueuing rules of the present invention and determining the positions of the segments belonging to datagram E relative to the other segments previously queued.

FIG. 2 illustrates an example of enqueuing rule 1a of the preferred embodiment of the present invention. As illustrated in FIG. 2, the LLC has previously received and enqueued segments belonging to datagrams A, B, C and D based on priority. As shown, datagram A is an unconfirmed datagram having a priority level of 1, and segments A_0, A_1, and A_2 are enqueued first; datagram B is an unconfirmed datagram having a priority level of 2, and segments B_0, B_1 and B_2 are enqueued second; datagram C is an unconfirmed datagram having a priority level of 3, and segments C_0, C_1 and C_2 are enqueued fourth; and datagram D is a confirmed datagram having a priority level of 3, and segments D_3, D_4, and D_5 are enqueued third. Since datagrams C and D both have the same priority level (i.e., priority level 3), the LLC gives precedence to the datagrams having the higher number of attempted transmission, thus the LLC enqueues segments D_3, D_4 and D_5 before segments C_0, C_1 and C_2 because datagram D is on its second transmission attempt whereas datagram C is on its first transmission attempt. It should be noted that only a portion of datagram D (i.e., D_3, D_4 and D_5) is being retransmitted for a second time, which indicates segments D_0, D_1 and D_2 were successfully transmitted during the first transmission attempt; this is possible since datagram D is a confirmed datagram.

Also shown in FIG. 2 are segments belonging to datagram E. Datagram E is a confirmed datagram having a priority level of 4; datagram E is divided into segments E_0, E_1, E_2, E_3 and E_4. According to this example, datagram E (i.e., the unqueued segments of datagram E) was received after the segments for datagrams A, B, C, and D have been enqueued. Since segments E_0, E_1, E_2, E_3 and E_4 have the lowest priority level of all the segments enqueued, these segments will be transmitted last. Moreover, since no other segments in the queue exist with the same priority level as the segments belonging to datagram E, precedence within a priority level does not need to be determined based on attempted transmissions.

It should be noted that the examples illustrated in FIGS. 3, 4, and 5 focus on enqueuing prioritizations for priority level 3; in these upcoming examples, enqueuing prioritizations for priority 1 and 2 remain the same as that described in FIG. 2 since only a single datagram is classified into each of these priority levels (i.e., datagrams A and B, respectively). Moreover, segments belonging to datagram C and D have already been enqueued according to the enqueuing rules of the present invention, and their order in the queue relative to other segments previously queued will not change (for example, in these figures, the segments belonging to datagram D will always precede the segments belonging to datagram C); as a result, enqueuing prioritization will not be addressed with respect to segments belonging to datagrams C and D in FIGS. 3-5. As stated above, FIGS. 3-5 will focus on enqueuing segments belonging to datagram E (i.e., the unqueued segments of datagram E) according to the enqueuing rules of the present invention and determining the positions of the segments belonging to datagram E relative to the other segments previously queued, particularly within priority level 3.

Moving ahead to FIG. 3. FIG. 3 illustrates an example of enqueuing rule 1b of the preferred embodiment of the present invention. The example in FIG. 3 is very similar to that illustrated in FIG. 2, however, datagram E now has a priority level of 3, which is the same priority level as datagrams C and D. Further, it should be noted that in FIG. 3, datagram E is on its second attempt to transmit segments E0, E1, and

E4. Since three datagrams have a priority level of 3 in this example (i.e., datagrams C, D, and E) , segments belonging to datagram E are enqueued in the following manner:

- Segments belonging to the second transmission attempt of datagram D have precedence over segments belonging to the second transmission attempt of datagram E because, even though both datagrams D and E are on their second transmission attempts, the second transmission attempt of datagram D was enqueued before the second transmission attempt of datagram E; and
- Segments belonging to the second transmission attempt of datagram E have precedence over segments belonging to the first transmission attempt of datagram C because datagram E has a higher number of attempted transmissions than datagram C.

Next, FIG. 4 illustrates an example of enqueuing rule 1c and is similar to the example illustrated in FIG. 3, however, the number of attempted transmissions of datagram E has increased from two attempts to three attempts. As a result, the enqueuing prioritization of segments belonging to datagram E is as follows:

- Segments belonging to the third transmission attempt of datagram E has precedence over segments belonging to the first transmission attempt of datagram C and the second transmission attempt of datagram D because datagram E has the highest number of attempted transmissions.

FIG. 5 is an example of enqueuing rule 1d and is also similar to the example illustrated in FIG. 3, however, the number of attempted transmissions of datagram E has decreased from two attempts to one attempt. As a result, the enqueuing prioritization of the segments belonging to datagram E is as follows:

- Segments belonging to the second transmission attempt of datagram D have precedence over segments belonging to the first transmission attempt of datagram E because datagram D has a higher number of attempted transmissions than datagram E; and

- Segments belonging to the first transmission attempt of datagram C now have precedence over segments belonging to the first transmission attempt of datagram E because, even though datagrams C and E have the same number of attempted transmissions, the first transmission attempt of datagram C was enqueued before the first transmission attempt of datagram E.

Enqueuing rule 2 is the example illustrated in FIG.6 and is dissimilar from the examples illustrated in FIGS. 2-5. In FIGS. 2-5, enqueuing the unqueued datagram (i.e., datagram E) did not disrupt the transmission of an unconfirmed segment since there were segments in the queue that were of a higher priority than the segments of the unqueued datagram, thus preventing the immediate transmission of the segments of the unqueued datagram. In FIG. 6, however, the enqueuing of the unqueued datagram (i.e., in this example, datagram C) causes disruption of the attempted transmission of an unconfirmed segment (i.e., in this example, A_5). As illustrated in FIG. 6, datagrams A and B have a priority level of 3 and are both unconfirmed datagrams. Prior to enqueuing segments belonging to datagram C, segment A_5 is the next segment to be transmitted. Before or during the transmission of segment A_5, datagram C is received having a priority level of 1. Since datagram C has a priority level higher than any other datagram in the queue, including the remaining segments of datagram A, the LLC enqueues the segments belonging to datagram C to be transmitted next. As a result, the transmission of unconfirmed datagram A is interrupted because now the next segment to be transmitted is segment C_0 as opposed to segment A_5. Since the transmission of unconfirmed datagram A was interrupted (i.e., a segment from a second datagram (i.e., datagram C) was transmitted before all the segments of a first datagram was transmitted), all the segments belonging to unconfirmed datagram A remaining in the queue are discarded, and the entire unconfirmed datagram A is re-enqueued with a different identifier (i.e., in this case, datagram A'). It is important to note that datagrams A and A' are identical, however, a different identifier is provided when re-enqueued to avoid the problem of datagram ambiguity.

It should be noted that if the enqueuing of an unqueued datagram causes disruption of the attempted transmission of a confirmed datagram, the remaining segments belonging to the confirmed datagram in the queue will still be transmitted; in the preferred embodiment, the remaining segments belonging to the confirmed datagram will be transmitted after transmission of the unqueued datagram, and any other datagram/segement that has been enqueued in front of the remaining segments belonging to the confirmed datagram. As noted above, transmitting the confirmed segments belonging to the confirmed datagram at a later time is allowable because the transmitter receives an acknowledgment/negative acknowledgment from the receiver as to which segments were received/not received.

At this point, the enqueuing rules and examples of enqueuing prioritization have been described. Let us now turn the discussion to the receiver of the segments. In the preferred embodiment, when the receiver receives a segment from a single transmitter, the segment is determined to be an unconfirmed or a confirmed data segment. When the receiver determines that an unconfirmed segment from datagram X has been received, the receiver waits for the remaining segments of datagram X until at least one of the following events occur:

1. a segment from datagram Y is received;
2. a trigger occurs (e.g., an elapsed amount of time, frequency change, etc.) that informs the receiver to no longer wait for the remaining segments; and/or
3. all of the unconfirmed segments belonging to datagram X are received.

When the receiver determines that a confirmed segment has been received, it assumes that no more than *n* confirmed datagrams are in flight at any given time (where *n* is the window size), the following rules apply:

1. The receiver must maintain all received segments for confirmed datagram X until:

a. all segments have been properly received, in which case the receiver re-assembles the confirmed datagram and passes it onto the application; or

b. the final transmission attempt for the confirmed datagram X has failed, in which case the receiver discards all of the received segments from confirmed datagram X;

2. After each transmission attempt has been completed, the receiver acknowledges all successfully received segments.

While the invention has been described in conjunction with specific embodiments thereof, additional advantages and modifications will readily occur to those skilled in the art. The invention, in its broader aspects, is therefore not limited to the specific details, representative apparatus, and illustrative examples shown and described. Various alterations, modifications and variations will be apparent to those skilled in the art in light of the foregoing description. Thus, it should be understood that the invention is not limited by the foregoing description, but embraces all such alterations, modifications and variations in accordance with the spirit and scope of the appended claims.

CLAIMS

We claim:

1. A method comprising:

receiving an unqueued segment to be enqueued in a queue, wherein the queue comprises at least one segment;

determining a priority level of the unqueued segment;

determining a number of attempted transmissions for the unqueued segment; and

if a segment in the queue exists with the same priority level and the same number of attempted transmissions as the unqueued segment, inserting the unqueued segment after a last segment in the queue having the same priority level and the same number of attempted transmissions.

2. The method of claim 1 wherein the queue is a plurality of queues, and wherein the unqueued segment is enqueued into one of the plurality of queues.

3. The method of claim 1 wherein the unqueued segment and at least one segment in the queue comprises at least one of the following: a complete datagram, and a portion of a datagram.

4. The method of claim 1 further comprising, if the step of inserting results in a queue overflow, discarding the segment having a lowest priority level and lowest number of attempted transmissions.

5. The method of claim 1 further comprising, if the step of inserting results in a queue overflow, discarding the segment in the queue that is time-sensitive and that has been in the queue longer than other segments that are time-sensitive.

6. A method comprising:

receiving an unqueued segment to be enqueued in a queue, wherein the queue comprises at least one segment;

determining a priority level of the unqueued segment;

determining a number of attempted transmissions for the unqueued segment; and

if a segment in the queue exist with the same priority level, and without the same number of attempted transmissions, but with a greater number of attempted transmissions than the unqueued segment, inserting the unqueued segment after all of the segments in the queue having the same priority level and a greater number of attempted transmissions.

7. The method of claim 6 wherein the queue is a plurality of queues, and wherein the unqueued segment is enqueued into one of the plurality of queues.

8. The method of claim 6 wherein the unqueued segment and at least one segment in the queue comprises at least one of the following: a complete datagram, and a portion of a datagram.

9. The method of claim 6 further comprising, if the step of inserting results in a queue overflow, discarding the segment having a lowest priority level and lowest number of attempted transmissions.

10. The method of claim 6 further comprising, if the step of inserting results in a queue overflow, discarding the segment in the queue that is time-sensitive and that has been in the queue longer than other segments that are time-sensitive.

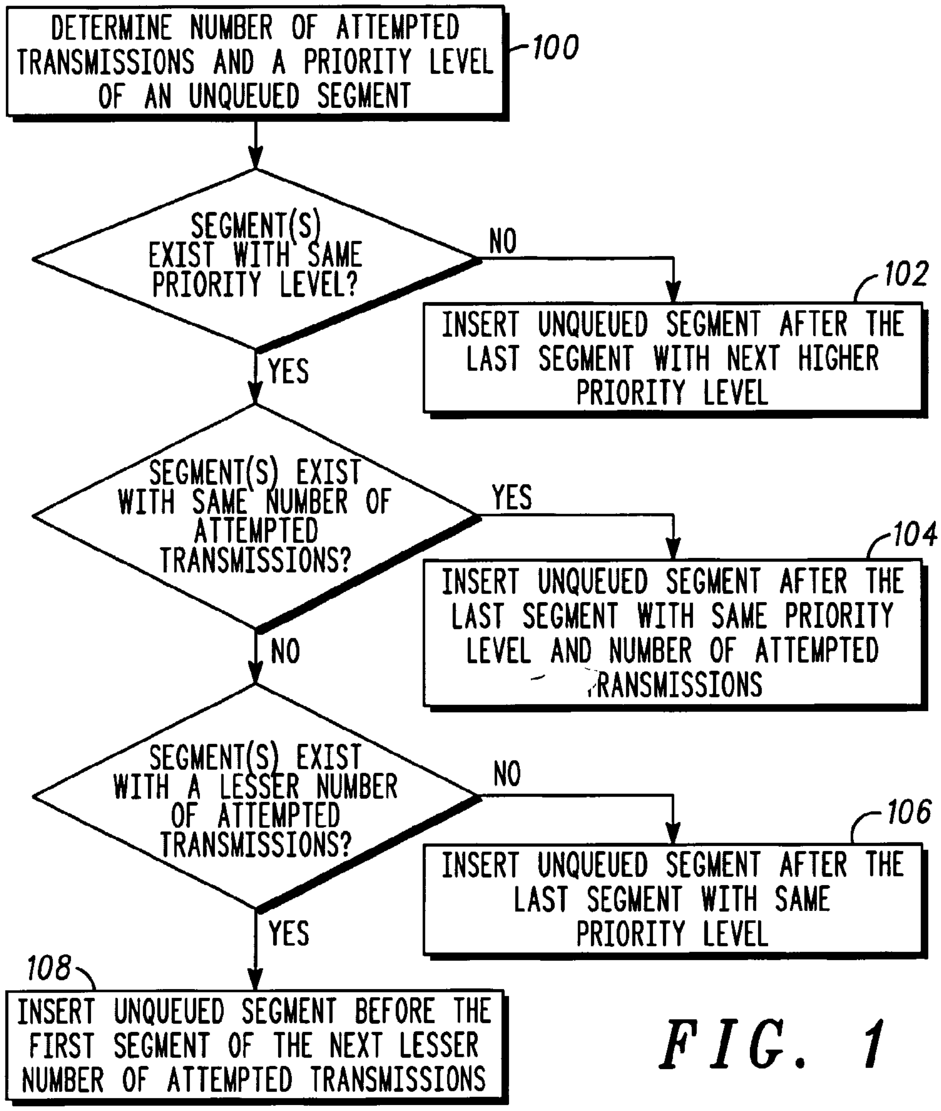

DETERMINE NUMBER OF ATTEMPTED TRANSMISSIONS AND A PRIORITY LEVEL OF AN UNQUEUED SEGMENT
100
SEGMENT(S) EXIST WITH SAME PRIORITY LEVEL?
NO
102
INSERT UNQUEUED SEGMENT AFTER THE LAST SEGMENT WITH NEXT HIGHER PRIORITY LEVEL
YES
SEGMENT(S) EXIST WITH SAME NUMBER OF ATTEMPTED TRANSMISSIONS?
YES
104
INSERT UNQUEUED SEGMENT AFTER THE LAST SEGMENT WITH SAME PRIORITY LEVEL AND NUMBER OF ATTEMPTED TRANSMISSIONS
NO
SEGMENT(S) EXIST WITH A LESSER NUMBER OF ATTEMPTED TRANSMISSIONS?
NO
106
INSERT UNQUEUED SEGMENT AFTER THE LAST SEGMENT WITH SAME PRIORITY LEVEL
YES
108
INSERT UNQUEUED SEGMENT BEFORE THE FIRST SEGMENT OF THE NEXT LESSER NUMBER OF ATTEMPTED TRANSMISSIONS

FIG. 1

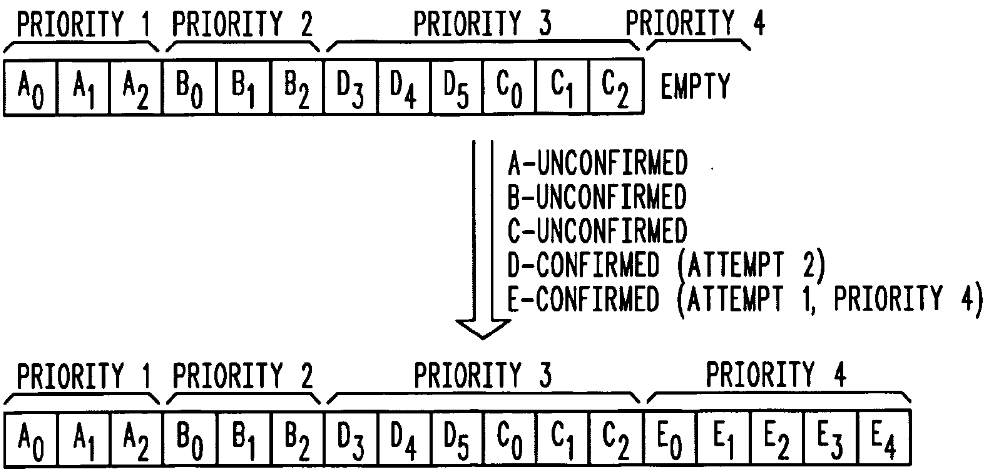

PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
A0 A1 A2 B0 B1 B2 D3 D4 D5 C0 C1 C2
EMPTY
A-UNCONFIRMED
B-UNCONFIRMED
C-UNCONFIRMED
D-CONFIRMED (ATTEMPT 2)
E-CONFIRMED (ATTEMPT 1, PRIORITY 4)
PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
A0 A1 A2 B0 B1 B2 D3 D4 D5 C0 C1 C2 E0 E1 E2 E3 E4

FIG. 2

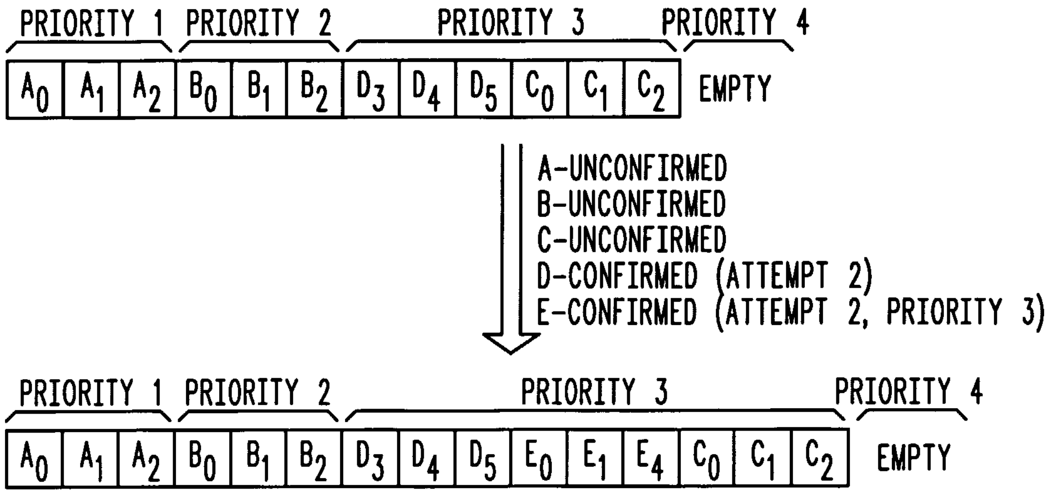

PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
A0 A1 A2 B0 B1 B2 D3 D4 D5 C0 C1 C2
EMPTY
A-UNCONFIRMED
B-UNCONFIRMED
C-UNCONFIRMED
D-CONFIRMED (ATTEMPT 2)
E-CONFIRMED (ATTEMPT 2, PRIORITY 3)
PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
A0 A1 A2 B0 B1 B2 D3 D4 D5 E0 E1 E4 C0 C1 C2
EMPTY

FIG. 3

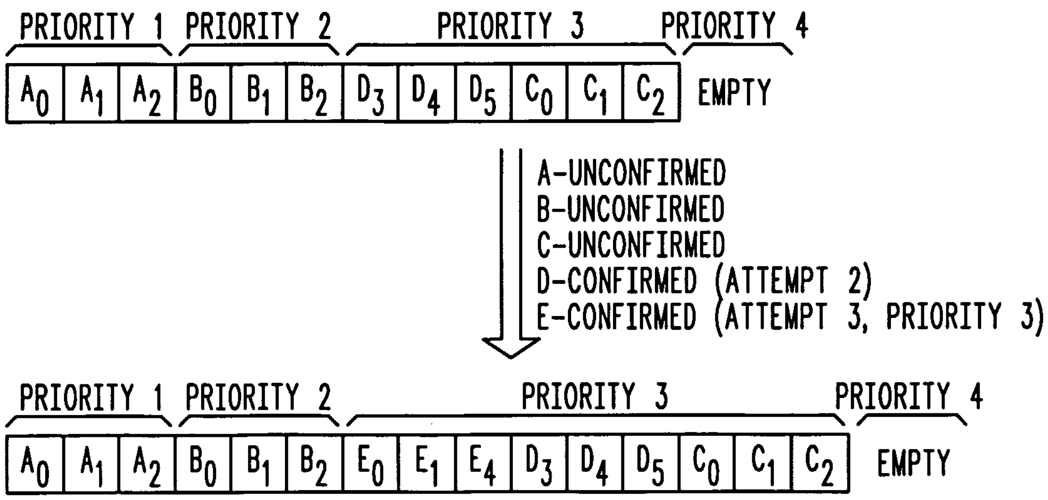

PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
A0 A1 A2 B0 B1 B2 D3 D4 D5 C0 C1 C2
EMPTY
A-UNCONFIRMED
B-UNCONFIRMED
C-UNCONFIRMED
D-CONFIRMED (ATTEMPT 2)
E-CONFIRMED (ATTEMPT 3, PRIORITY 3)
PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
A0 A1 A2 B0 B1 B2 E0 E1 E4 D3 D4 D5 C0 C1 C2
EMPTY

FIG. 4

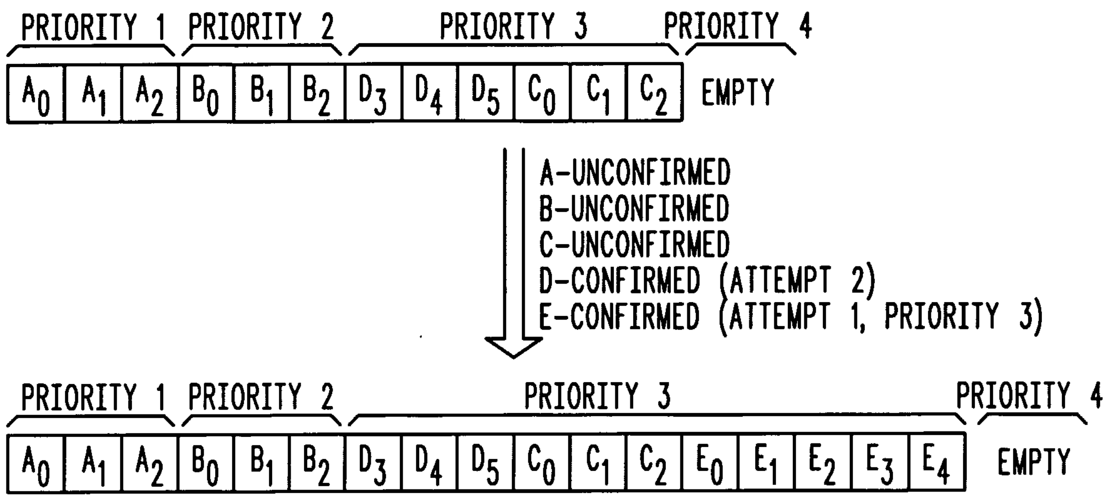

PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
A0 A1 A2 B0 B1 B2 D3 D4 D5 C0 C1 C2
EMPTY
A-UNCONFIRMED
B-UNCONFIRMED
C-UNCONFIRMED
D-CONFIRMED (ATTEMPT 2)
E-CONFIRMED (ATTEMPT 1, PRIORITY 3)
PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
A0 A1 A2 B0 B1 B2 D3 D4 D5 C0 C1 C2 E0 E1 E2 E3 E4
EMPTY

FIG. 5

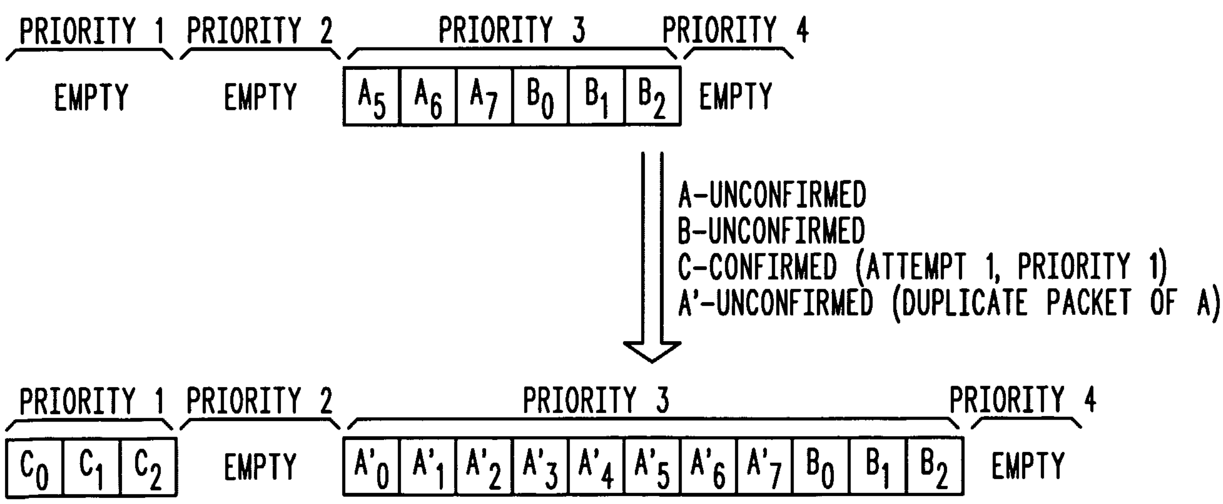

PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
EMPTY
EMPTY
A5 A6 A7 B0 B1 B2
EMPTY
A-UNCONFIRMED
B-UNCONFIRMED
C-CONFIRMED (ATTEMPT 1, PRIORITY 1)
A'-UNCONFIRMED (DUPLICATE PACKET OF A)
PRIORITY 1
PRIORITY 2
PRIORITY 3
PRIORITY 4
C0 C1 C2
EMPTY
A'0 A'1 A'2 A'3 A'4 A'5 A'6 A'7 B0 B1 B2
EMPTY

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US03/12482

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : H04J 3/02; H04Q 11/04; G06F 13/00
US CL : 370/230, 395.4, 412, 429; 709/232; 710/29, 36
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)
U.S. : 370/229, 230, 231, 235, 389, 395.1, 395.4, 395.42, 412, 428, 429; 709/232, 234, 238; 710/29, 36, 39, 40, 52

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EAST search terms: (queu$3 or enqueu$3), priority adj level, overflow, discard$3, transmission near4 attempt$3

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,361,255 A (DIAZ et al) 01 November 1994, see col. 3, lines 40-59, col. 6, line 33 to col. 8, line 18, col. 9, line 58 to col. 10, line 56.	1-10
X	US 5,974,465 A (WONG) 26 October 1999, see col. 2, lines 23-61, col. 3, line 56 to col. 5, line 57.	1-10
A	US 4,914,653 A (BISHOP et al) 03 April 1990, see entire document.	1-10
A, P	US 6,401,147 B1 (SANG et al) 04 June 2002, see entire document.	1-10
A, P	US 6,487,212 B1 (ERIMLI et al) 26 November 2002, see entire document.	1-10

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

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

"E" earlier application or patent 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
02 July 2003 (02.07.2003)

Date of mailing of the international search report
24 JUL 2003

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
Facsimile No. (703)305-3230

Authorized officer
Alpus H. Hsu
Eugenia Logan
Telephone No. (703)305-4700