

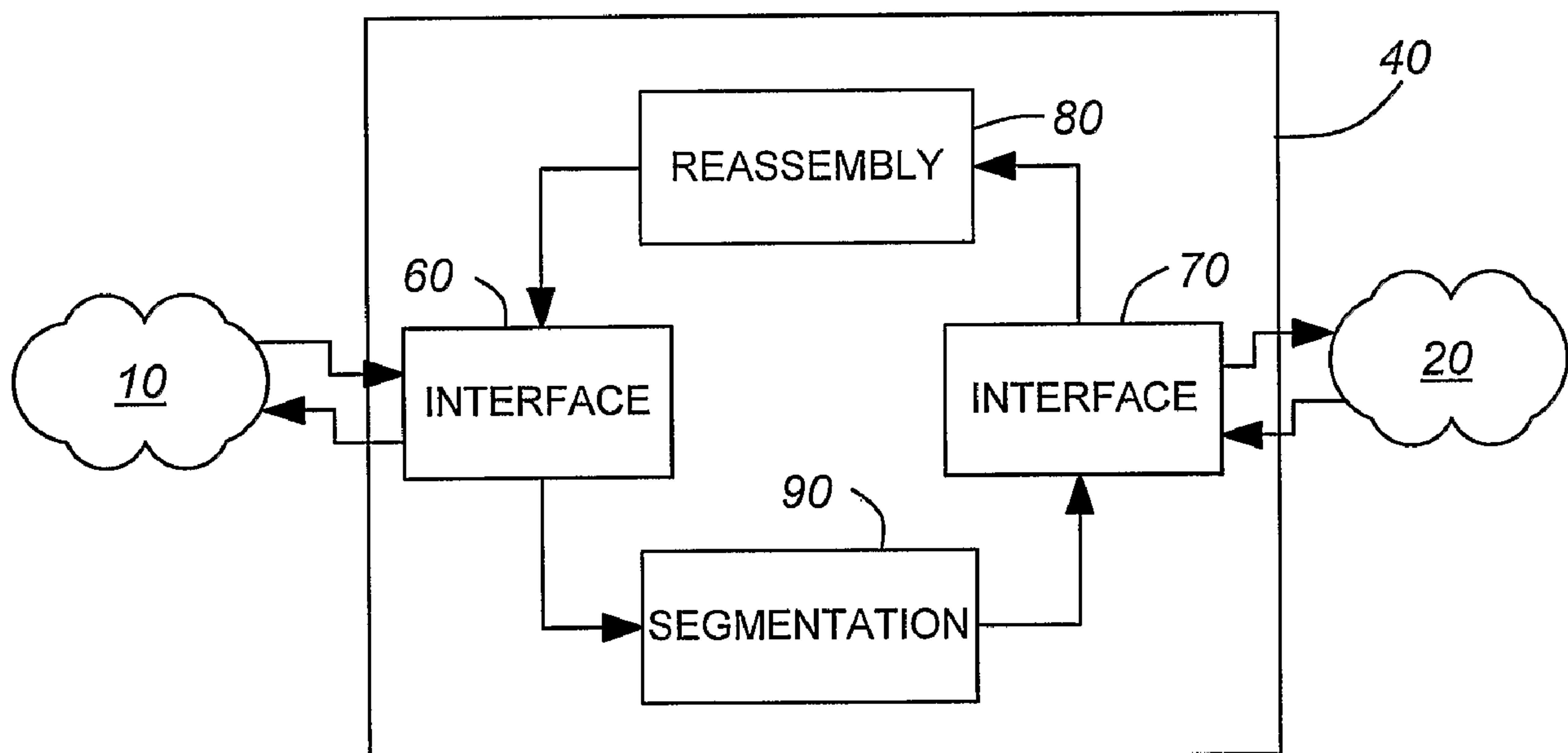


(86) Date de dépôt PCT/PCT Filing Date: 2005/08/03
(87) Date publication PCT/PCT Publication Date: 2006/02/09
(45) Date de délivrance/Issue Date: 2010/11/16
(85) Entrée phase nationale/National Entry: 2007/02/05
(86) N° demande PCT/PCT Application No.: CA 2005/001206
(87) N° publication PCT/PCT Publication No.: 2006/012743
(30) Priorité/Priority: 2004/08/06 (US10/912,200)

(51) Cl.Int./Int.Cl. *H04L 12/56* (2006.01),
H04L 12/66 (2006.01)
(72) Inventeur/Inventor:
WILLIAMS, MATTHEW ROBERT, CA
(73) Propriétaire/Owner:
IPEAK NETWORKS INCORORATED, CA
(74) Agent: CASSAN MACLEAN

(54) Titre : SYSTEM ET PROCEDE PERMETTANT D'AUGMENTER LA VITESSE DE TRANSMISSION DE DONNEES A TRAVERS UN RESEAU DE TRANSPORT

(54) Title: SYSTEM AND METHOD FOR HIGHER THROUGHPUT THROUGH A TRANSPORTATION NETWORK



(57) Abrégé/Abstract:

Systems and methods for transporting data from a source network to a destination network by way of a transport network. Data transmission units (DTUs) from the source network are received at a server logically located between the source and the transport networks. These first DTUs are subdivided into second DTUs and are transmitted to the destination network by way of the transport network. Also transmitted are encoded or extra second DTUs that allow the original first DTUs to be recreated even if some of the second DTUs are lost. These encoded second DTUs may be merely copies of the second DTUs transmitted, parity second DTUs, or second DTUs which have been encoded using erasure correcting coding. At the destination network, the second DTUs are received and are used to recreate the original first DTUs. In the event an insufficient number of second DTUs are received, such that recreating the original first DTU cannot be accomplished, a re-transmission of the second DTUs is requested.



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

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
9 February 2006 (09.02.2006)

PCT

(10) International Publication Number
WO 2006/012743 A1

(51) International Patent Classification⁷: **H04L 12/56**, 12/66

(21) International Application Number:
PCT/CA2005/001206

(22) International Filing Date: 3 August 2005 (03.08.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/912,200 6 August 2004 (06.08.2004) US

(71) Applicant (for all designated States except US): **IPEAK NETWORKS INCORPORATED** [CA/CA]; Suite 219, 300 Earl Grey Drive, Kanata, Ontario K2T 1C1 (CA).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **WILLIAMS, Matthew Robert** [CA/CA]; 7 Insmill Crescent, Kanata, Ontario K2T 1G5 (CA).

(74) Agent: **CASSAN MACLEAN**; Suite 401, 80 Aberdeen Street, Ottawa, Ontario K1S 5R5 (CA).

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

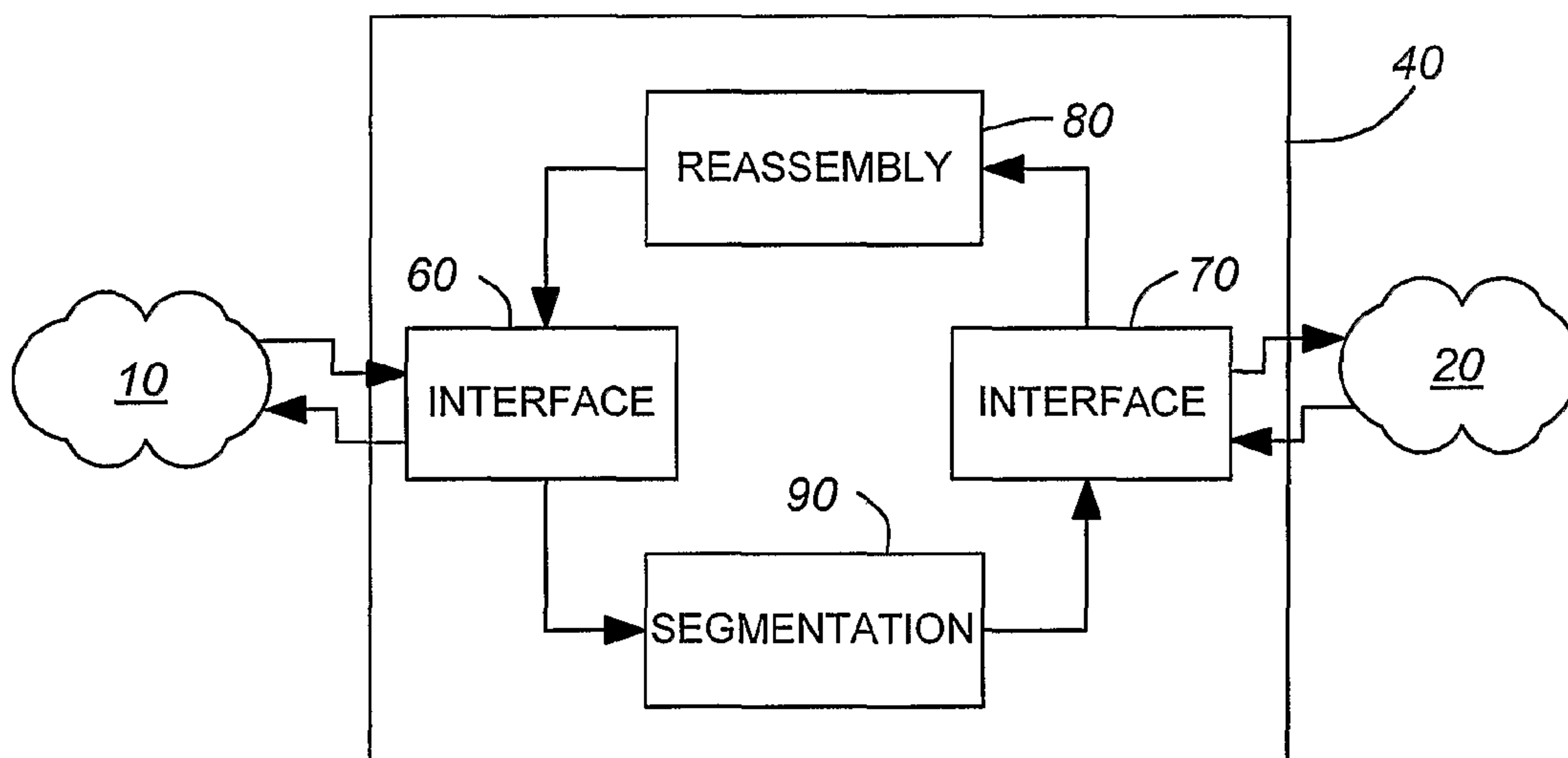
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (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.

(54) Title: SYSTEM AND METHOD FOR HIGHER THROUGHPUT THROUGH A TRANSPORTATION NETWORK



(57) Abstract: Systems and methods for transporting data from a source network to a destination network by way of a transport network. Data transmission units (DTUs) from the source network are received at a server logically located between the source and the transport networks. These first DTUs are subdivided into second DTUs and are transmitted to the destination network by way of the transport network. Also transmitted are encoded or extra second DTUs that allow the original first DTUs to be recreated even if some of the second DTUs are lost. These encoded second DTUs may be merely copies of the second DTUs transmitted, parity second DTUs, or second DTUs which have been encoded using erasure correcting coding. At the destination network, the second DTUs are received and are used to recreate the original first DTUs. In the event an insufficient number of second DTUs are received, such that recreating the original first DTU cannot be accomplished, a re-transmission of the second DTUs is requested.

WO 2006/012743 A1

SYSTEM AND METHOD FOR HIGHER THROUGHPUT THROUGH A TRANSPORTATION NETWORK

Field of the Invention

5 The present invention relates to communication data networks. More specifically, the present invention relates to systems and methods for increasing the throughput of data transmissions through a transport network as seen from the edge of the transport network.

Background to the Invention

10 The growth in the number of businesses using an internal network for geographically remote locations has been phenomenal in the past few years. An office in Dallas can have its own internal communications network separate and distinct from the networks of its sister offices in Los Angeles and Seattle. However, given the need
15 to share data and resources between such distinct locations, high speed and high performance backbone links between such offices are usually leased by the parent company to connect the different networks. Clearly, such backbone links are expensive to lease from their providers. However, the businesses that lease these lines are looking for, among others, good application performance across the transport network
20 or the network that transports the data between the two end networks.

 Most, if not all, transport networks are packet based networks that break up larger pieces of data into smaller packets of data which are then transmitted from a first source network to a third destination network via a second transport network. However, due to congestion and other network limitations, not all packets successfully arrive at
25 the destination network. It is this packet loss that the end networks seek to minimize and, consequently, the expensive high performance transport networks or links are leased.

 What matters to the source and end destination networks is the performance of the transport network. The transport network must, from the point of view of the
30 applications at the end networks, ideally be perfect with no lost packets. However, it would be preferred if such performance can be had for a price lower than the usual costs of leasing high performance transport networks.

Accordingly, there is a need for systems and methods which can be used with low cost communications transport networks to provide end network applications with a high performance view of the transport network.

Currently, two approaches have been tried to address the above situation. In one approach, the end networks employ custom protocols that are not greatly affected by data loss and latency. However, this approach requires extensive retooling as most systems use well accepted protocols such as TCP/IP. Such an approach may also result in chaos as there is no guarantee that such custom protocols will be interoperable with established end and transport networks.

Another approach involves using a custom protocol stack to transfer data across the transport network and initiating new TCP/IP sessions at the destination network. However, this approach is only useful for systems which use the TCP/IP protocol and other protocols, such as UDP/IP, may be affected by the latency and/or loss across the transport network.

It should be noted that the term data transmission unit (DTU) will be used in a generic sense throughout this document to mean units which include transmitted data. Thus, such units may take the form of packets, cells, frames, or any other such units as long as data is encapsulated within the unit. Thus, the term DTU is applicable to any and all packets and frames that implement specific protocols, standards, or transmission schemes.

Summary of the Invention

The present invention provides systems and methods for transporting data from a source network to a destination network by way of a transport network. Data transmission units (DTUs) from the source network are received at a server logically located between the source and the transport networks. These first DTUs are subdivided into second DTUs and are transmitted to the destination network by way of the transport network. Also transmitted are encoded or extra second DTUs that allow the original first DTUs to be recreated even if some of the second DTUs are lost. These encoded second DTUs may be merely copies of the second DTUs transmitted, parity second DTUs, or second DTUs which have been encoded using erasure correcting coding. At the destination network, the second DTUs are received and are used to

recreate the original first DTUs. In the event an insufficient number of second DTUs are received, such that recreating the original first DTU cannot be accomplished, a re-transmission of the second DTUs is requested.

5 In accordance with one aspect of the invention, a server is provided for forwarding data from a first network to a second network. The server includes a first interface module for sending data to and receiving data from the first network and a second interface module for sending data to and receiving data from the second network. A segmenting module segments each first data transmission unit (DTU) of a first type received from the first network into at least two second DTUs of a second type, 10 the segmenting module receiving first DTUs from the first interface module and the segmenting module sending the at least two second DTUs and at least one duplicate second DTU to the second interface module for transmission to the second network. A reassembly module reassembles first DTUs from second DTUs and duplicate second DTUs received the second network and sends reassembled first DTUs to the first 15 interface module for transmission to the first network. The reassembly module receives the second DTUs and duplicate second DTUs from the second network interface and determines if a predetermined total number of second DTUs and duplicate second DTUs have been received to reassemble a first DTU. In the event the predetermined total number of second DTUs and duplicate second DTUs is not received, the 20 reassembly module transmits a retransmit request to a sender server by way of the second interface module. For example, the segmenting module may divide each first DTU into "n" second DTUs and, for each first DTU, may send "m" duplicate second DTUs to the second interface module, and "m" is less than or equal to "n".

25 The segmenting module may buffer and retransmit a requested group of second DTUs by way of the second interface module in the event the segmenting module receives the retransmit request from the reassembly module. The segmenting module may interleave second DTUs from different first DTUs when sending the interleaved second DTUs to the second interface module. At least one of the interface modules may include a filter for filtering DTUs wherein the filter determines which DTUs are to 30 be sent to the segmenting module and the filter may determine a segmentation of each first DTU. A filter may determine if second DTUs generated from the first DTUs are to be re-transmitted, wherein second DTUs which are not to be re-transmitted are tagged

by the server.

Further in accordance with the invention there is provided a system for transmitting data from a first network to a third network by way of a second network. The system includes a first server connecting the first network and the second network, the first server receiving first DTUs of a first type from the first network and subdividing each one of the received first DTUs into second DTUs of a second type, the second DTUs and at least one duplicate second DTU being transmitted to the second network by the first server. A second server connects the second network and the third network, the second server receiving the second DTUs and the at least one duplicate second DTU from the second network, and on determining that a predetermined total number of second DTUs and duplicate second DTUs have been received, the second server recreating the first DTUs from the second DTUs and the duplicate second DTUs, the second server requesting a retransmission of the second DTUs from the first server in the event the predetermined total number of second DTUs and duplicate second DTUs are determined not to have been received, and the second server transmitting the recreated first DTUs to the third network.

In the system DTUs may be received by a filter for determining which received DTUs are to be subdivided based on at least one criterion selected from the group comprising: a destination IP address of the incoming DTU; a source IP address of the incoming DTU; a source TCP/UDP port of the incoming DTU; a destination TCP/UDP port of the incoming DTU; header information in the incoming DTU; data in the incoming DTU; a size or length of the incoming DTU; and, an outgoing port or ingress port of the incoming DTU. DTUs which are not to be subdivided may be tagged by the server. A filter may also determine which received DTUs are suitable for re-transmission based on the foregoing criterion.

The invention further provides a method of forwarding data transmissions from a first network to a third network by way of a second network wherein the first network produces a first data transmission unit (DTU) of a first type and the first DTU is segmented into second DTUs of a second type. Second DTUs and at least one duplicate second DTU are received from the second network and, based on how many second DTUs and duplicate second DTUs have been received, a determination is made whether a predetermined total number of second DTUs and duplicate second DTUs

have been received to recreate a first DTU. In the event the predetermined total number of second DTUs and duplicate second DTUs have not been received, re-transmission of the second DTUs and duplicate second DTUs is requested and the foregoing determination is repeated. In the event the predetermined total number of second DTUs and duplicate second DTUs have been received, the first DTU is recreated and the recreated first DTU is transmitted to the third network. The second DTUs and duplicate second DTUs received from the second network may be interleaved with second DTUs and duplicate second DTUs from different first DTUs.

Further, a method forwarding data transmissions from a first network to a third network via a second network is provided by the invention by which first data transmission units (DTUs) of a first type are received from the first network and for each first DTU of a first type, the first DTU is segmented into second DTUs of a second type and the second DTUs are transmitted to the second network. At least one duplicate second DTU is transmitted to the second network, wherein the at least one duplicate second DTU allows the first DTU to be recreated in the event not all second DTUs are received at the second network. In the event a predetermined total number of second DTUs and duplicate second DTUs have not been received to recreate the first DTU, re-transmission of the second DTUs and the at least one duplicate DTU is requested. Second DTUs and duplicate second DTUs from different first DTUs may be interleaved prior to transmitting them.

Brief Description of the Drawings

A better understanding of the invention will be obtained by considering the detailed description below, with reference to the following drawings in which:

Figure 1 is a block diagram of an environment in which the invention may be practiced;

Figure 2 is a block diagram illustrating the components in a server used in Figure 1;

Figure 3 is a block diagram illustrating the components in another server used in Figure 1;

Figure 3A is a block diagram of a server similar to that illustrated in Figure 3 but implementing at least one filter;

Figure 4 is a flowchart illustrating the steps executed when first DTUs are received and transformed into second DTUs for transmission to the transport network; and

Figure 5 is a flowchart illustrating the steps executed when second DTUs are received from the transport network and transformed into first DTUs for transmission to an end network.

Detailed Description

Referring to Fig 1, a block diagram of an environment in which the invention may be practised is illustrated. A first network 10 communicates with a second network 20 which, in turn, communicates with a third network 30. The first network 10 and the second network 20 communicate through server 40 while the second network 20 communicates with the third network 30 through server 50. Should the first network 10 wish to communicate with the third network 30 (perhaps to create a larger wide area network), then the second network 20 becomes a transport network as it merely provides a conduit between the first network 10 and the third network 30. As noted above, applications residing in the two end networks, the first network 10 and the third network 30, merely require that the transport network provide a virtually perfect conduit with no lost DTUs and, preferably, minimal latency.

The present invention seeks to provide a means for insulating the end networks from the vagaries of the transport network. The servers 40, 50 insulate the end networks 10, 30 from dealing with the transport network 20 by handling the transmission of DTUs through the transport network 20. The end networks may use any communications protocols between them and the servers 40, 50 may use its own protocol between them. The servers 40, 50 receive DTUs in one protocol from the end networks and repackages these DTUS into new DTUs prior to transmitting them through the transport network. However, instead of merely repackaging the DTUs from the end networks, the servers 40, 50 also subdivides them into preferably smaller DTUs. Also, the servers 40, 50 encodes extra DTUs which will allow the recreation/reconstruction of the original DTUs from the end networks even though some of the DTUs which were transmitted through the transport network were lost.

Once the DTUs transmitted through the transport network are received at the

other end, the original DTUs from the end network are recreated. This is done by reordering the received DTUs and, if some DTUs were lost during transmission, using the extra DTUs which were transmitted as well. In the event the extra received DTUs are insufficient to recreate the original DTUs from the end network, then the receiving
 5 server may request a re-transmission of the DTUs previously sent.

For ease of explaining the above process, DTUs originating from or being transmitted to the end networks will be referred to as first DTUs and as being of a first type of DTU. DTUs being transmitted across and/or received from the transport network will be referred to as second DTUs and as being of a second type of DTU.

10 Referring to Fig 2, a block diagram of the modules in server 40 is illustrated along with the data flow associated with the networks server 40 communicates with. As can be seen, server 40 has a first interface 60, a second interface 70, a reassembly module 80, and a segmentation module 90. The first interface 60 sends and receives first DTUs to and from an end network. In Fig 2, the end network is illustrated as the
 15 first network 10. The second interface 70 of the server 40 sends and receives second DTUs to and from the transport network. In Fig 2, the transport network is illustrated as being the second network 20.

Regarding the modules in the server 40, the reassembly module 80 receives second DTUs from the second interface 70 and produces first DTUs for transmission
 20 to the first network 10 by way of the first interface 60. The segmentation module 90, on the other hand, receives first DTUs from the first interface 60 and produces second DTUs for transmission to the transport network 20 by way of the second interface 70.

Once the second DTUs have been transmitted through the transport network, they are received by the server at the other end. To illustrate this, Fig 3 illustrates the
 25 modules and data flow through the server 50 which communicates with the transport network and the other end network. In Fig 3, the transport network is illustrated as the second network 20 and the third network 30 is illustrated as the end network.

The modules in the server 50 are the same as those in server 40 and, in fact, have the same functions. The second interface 70A in server 50 also communicates
 30 with the transport network and sends and receives second DTUs. The first interface 60A also communicates with an end network (in this case the third network 30) and sends and receives first DTUs. The reassembly module 80A receives second DTUs

and produces first DTUs while the segmentation module 90A receives first DTUs and produces second DTUs.

As noted above, first DTUs are the DTUs used by the end networks while second DTUs are used by the transport network and the servers 40, 50 when transmitting data to and from each other. The second DTUs are derived from the first DTUs. The first DTUs may be divided into smaller units and each smaller unit may be the payload in a second DTU. As such, it is quite possible that each second DTU may be smaller than the original first DTU from which it was derived. As an example, a 10kb first DTU may be subdivided into 5 2 kb units. These may be the payload of 5 second DTUs, each of which may be smaller than 10kb. This function of creating second DTUs from first DTUs is accomplished by the segmentation modules 90, 90A.

Once the second DTUs are created by the segmentation module 90, 90A, to assist in recreating the original first DTU from which the second DTUs were derived from, the segmentation module also creates extra or encoded second DTUs. These extra second DTUs are derived from the second DTUs derived from the first DTU. The concept behind the extra second DTUs is to assist the reassembly modules 80, 80A in recreating or reassembling the original first DTU in the event one or more second DTUs are lost during their transmission through the transport network.

The extra second DTUs may take many forms. In perhaps the simplest embodiment, the extra second DTUs are merely copies of selected second DTUs previously sent. As an example, if a first DTU was divided into 4 second DTUS (DTU1, DTU2, DTU3, and DTU4), then the extra second DTUs may be copies of DTU2 and DTU3. As such, if DTU 2 or DTU3 were lost during the transmission, then the original first DTU can still be recreated. Whichever of the second DTUs are to be duplicated and how many extra second DTUs are to be used are left to the discretion of the system administrator or to the setup of the system. Depending on how lossy the transport network may be, more or less redundancy may be built in by including more or less duplicate second DTUs. In the simplest case, all the second DTUs can be duplicated to ensure that, in essence, two copies of each second DTU is sent to the destination server.

Another type of extra second DTU would be parity DTUs. As is well known in the art, a parity DTU can be created using the XOR function. The bits of the different

second DTUs created from the original first DTU can be XOR'd to result in bit values which can be stored in an extra second DTU. If any one of the second DTUs (not including the extra second DTU) is lost during transmission, the other received second DTU and the extra second DTU can be used to recreate the lost second DTU.

5 Performing an XOR function on the received second DTUs and the extra second DTU will recreate the missing second DTU.

It should be noted that the extra second DTUs may be encoded using other erasure correcting codes. As an example, if n second DTUs are generated for a single first DTU, m extra second DTUs may be generated to allow the lost second DTUs to be

10 recreated. As noted above, the m extra second DTUs may be viewed as "redundant" second DTUs and, if mere duplication is used, $m \leq n$ with complete duplication being achieved at $m=n$. However, if erasure correcting codes are used, it is possible to encode the redundant information in such a way that more than one second DTU could be lost and the reassembly modules can still reconstruct the lost second DTUs. Well-

15 known methods and coding techniques such as Reed-Solomon, Forward Erasure Correction techniques, and Bose-Chaudhuri-Hochquenghem (BCH) codes, and a multitude of others may be used.

While the extra second DTUs should assist in counteracting the effects of losing some second DTUs, losing too many second DTUs cannot be completely compensated

20 for. As such, losing a number of second DTUs past a preset and predetermined threshold level may cause the reassembly modules to request a re-transmission of a package or group of second DTUs. As an example, if the extra second DTUs can recover from a 25% loss of DTUs and there are 4 second DTUs generated from a single first DTU, then the loss of a single second DTU will not trigger a re-transmission

25 request. However, with the loss of 2 second DTUs, a 50% loss, the reassembly module may request a re-transmission. The preset and predetermined threshold is ideally related to the error or loss correcting capability of the coding used for the extra or redundant second DTUs. The reassembly modules can keep track of the number of second DTUs received for each first DTU that has been segmented as the reassembly

30 modules will need to properly sequence the payloads of the second DTUs. However, as will be discussed below, some applications may not be amenable to re-transmission. In such cases, re-transmission would not be requested. It should also be noted that,

if there is an insufficient number of second DTUs to recreate the first DTUs, the system may just drop the second DTUs it did receive, after enough time has elapsed to declare the first DTU dead.

As can be noted from the above, the reassembly modules will need to decode, unwrap, and reassemble the second DTUs received. The second DTUs received will need to be tracked to determine if there is a sufficient number of them to be compensated for by the extra second DTUs or if a retransmission is to be requested. If there is a sufficient number and if some second DTUs have been lost, then the reassembly modules will need to recreate or reconstruct the missing second DTUs or, if not the actual lost second DTUs, the contents of those missing second DTUs will need to be reconstructed. As noted above, this process depends upon the coding used and the overall strategy employed. This decoding and error correction process is well-known to those versed in this art.

Once the required number of second DTUs have been received, their payloads are extracted and used to reconstruct the original first DTU from which was derived the second DTUs. This may be as simple as concatenating the payloads of the second DTUs to result in the reconstructed first DTU. However, as noted above regarding the decoding, the reconstruction process will depend upon the process used to segment or divide the original first DTU. With the original first DTU reconstructed, it can be forwarded to the interface module that communicates with the end network.

Regarding the segmentation modules, these modules perform the task of segmenting or dividing the first DTUs and “repackaging” the segments into second DTUs. The segmentation modules also encode the extra second DTUs as discussed above. The second DTUs, both those derived from the first DTU and the encoded extra second DTUs, are then passed on to the interface module that communicates with the transport network. To facilitate the re-transmission of the second DTUs in the event the reassembly module of the other end of the transport network requests it, the segmentation module should, preferably, buffer the last few second DTUs transmitted. As an example, if 5 first DTUs have been segmented into 20 second DTUs and 5 extra second DTUs, the segmentation module may buffer the last 3 sets of second DTUs corresponding to the last 3 first DTUs encoded. Thus, 12 second DTUs and 3 extra second DTUs can be buffered by the segmentation module.

To spread the risk of losing multiple second DTUs across different segmented first DTUs, the segmentation module may be configured to transmit second DTUs in an interleaved manner. Thus, instead of sequentially sending groups of second DTUs such that each group corresponds to a single first DTU, second DTUs from different first DTUs can be interleaved with one another. To illustrate the point, it can be assumed that first DTUs A, B, and C are respectively segmented into second DTUs DTU-A1, DTU-A2, DTU-A3, DTU-B1, DTU-B2, DTU-B3, DTU-C1, DTU-C2, and DTU-C3. Instead of sending these second DTUs as being grouped according to which first DTU they were derived from (i.e. group A is DTU-A1, DTU-A2, DTU-A3, and group B is DTU-B1, DTU-B2, DTU-B3, and so on), they can be interleaved. If second DTUs are sent in groups of 3 DTUs, then the first group can be DTU-A1, DTU-B1, and DTU-C1. Another group can be DTU-A2, DTU-B2, and DTU-C2, and so on. Using this scheme, if a group is lost, then a whole first DTU is not lost – only one third of the 3 first DTUs is lost. Depending on the coding and strategy employed, this type of loss may be recoverable.

Referring to Figure 3A, a variant 50A of the server 50 in Figure 3 is illustrated. In this variant 50A, the interfaces 60B, 70B each have a filter 65, 75 which filters incoming DTUs. Since some data streams are not amenable to compression or delay, the DTUs of these data streams may not be suitable for segmentation, re-transmission, or both. As an example, voice data packets or DTUs for a voice based application (e.g. VoIP) would not be suitable for either re-transmission or segmentation. For such DTUs which are unsuitable for re-transmission, the DTUs are tagged as being unsuitable for re-transmission. These DTUs, if suitable for segmentation, are thus segmented and transmitted through the transport network 20. Once received at the other end of the transport network 20, the receiving interface 60B filters it by way of filter 65 so that, if there are insufficient second DTUs received, no re-transmission is to be requested of the transmitting server. In such a situation, the first DTU from which the second DTUs are derived is declared dead.

For DTUs which are unsuitable for segmentation, the DTUs are tagged as being unsuitable for segmentation. These DTUs, suitably tagged, are directly sent to the other interface in the server. Thus, if an incoming first DTU is to be tagged as not to be segmented, it is received by the interface 60B and filtered by filter 65. The tagged DTU

is then sent to the other interface 70B to be sent through the transport network 20. At the other end of the transport network 20, another interface receives the tagged DTU and, because a corresponding filter recognizes the tag, directs the DTU straight to the other interface in the server.

5 The tagging of the DTUs can be done by changing a specific value in the DTU header or by concatenating a specific header or header values to the DTU being tagged.

10 The filter may also determine which incoming or first DTUs to segment based on any criteria such as source or destination ports, data contained in the DTU, or any one of or any combination of the following criteria:

- a destination IP address of the incoming DTU
- a source IP address of the incoming DTU
- a source TCP/UDP port of the incoming DTU
- a destination TCP/UDP port of the incoming DTU
- 15 - header information in the incoming DTU
- data in the incoming DTU
- a size or length of the incoming DTU
- an outgoing port or ingress port of the incoming DTU.

20 To take advantage of the invention's feature of guarding against lost or dropped DTUs, the system may generate at least one second DTU for every x first DTUs. This can be especially useful if the first DTUs are not segmented or otherwise repackaged. The second DTU generated can act as insurance against the dropping of the first DTUs being transmitted over the transport network. As an example, if DTU1, DTU2, DTU2 are first DTUs to be transported across the transport network without being segmented, 25 then DTU4 can be a generated second DTU derived from the DTU1, DTU2, and DTU3 units. Thus, if one of DTU1, DTU2, or DTU3 is lost, then DTU4 can be used to recover/recreate the lost DTU. This scheme is especially applicable if the first DTUs are of a fixed size and are generated at a regular rate.

30 The filter explained above may also be used for other ends than those listed above. The filter may, by detecting the size of the incoming first DTUs, determine the number of second DTUs into which each first DTU is to be segmented. By adjusting the filter settings, such as having a minimum or a maximum size for each second DTU,

various efficiencies in the system can be exploited. As an example, if a large number of second DTUs is desired, a small second DTU size can be set as a large number of second DTUs can decrease the overhead for the system. If the first DTUs are large in size, increasing the number of second DTUs into which each first DTU is segmented decreases the size of the extra second DTUs generated, thereby reducing the overhead. However, if the first DTUs are of a small size, the overhead for each second DTU generated may be larger than the size of the extra header, thereby causing overall overhead to increase.

While the system above describes a point to point connectivity, the system can be extended to a point to multipoint connectivity. The filter in the system may be used for this extension. The filter can, based on the destination information in the first DTUs header, direct the resulting second DTUs to the proper multiple destinations. Similarly, the input port from which the incoming first DTU entered may be used as the filter criterion. As an example, the filter could ensure that if a first DTU came from port x1, then any second DTUs or DTUs from that port are redirected to multiple, predetermined destinations.

Should a user of the system wish, the system may be configured to control or limit the system throughput. Such a limitation would be useful for limiting the system usage by some subscribers. Other subscribers, perhaps those at a higher service level or at a higher subscriber level, may have their access to the system be unencumbered.

To limit the throughput, a preset limit of x bytes per unit time may be implemented. By using a timer in conjunction with a count of the number of bytes being sent to the transport network, the system can keep track of a given throughput. Once the throughput has been exceeded, a delay can be built in during which no data is transported. Once the delay has been factored in, the transmission can resume. Multiple methods of applying the timer, delay, and byte counting can be implemented. To determine which data streams are to be delayed, again the filter discussed above may be used.

It should be noted that the system may also use well-known encryption and compression methods. The encryption may be applied to either the incoming first DTUs or the outgoing second DTUs although it may be more advantageous to apply any encryption to the incoming first DTUs.

For compression, any of the well-known compression methods may be applied. As with encryption, compression methods may be applied to either the incoming first DTUs or the outgoing second DTUs. However, again as with encryption, it may be more advantageous to apply compression to the incoming first DTUs. If both
 5 compression and encryption are to be applied, compression should be applied first as most encryption methods make it difficult to compress data once encrypted.

For a server transmitting second DTUs across the transport network, the steps in the process it executes are illustrated in Fig 4. The process begins with step 100, that of receiving a first DTU from the source end network. After being received, the first
 10 DTU is then divided or segmented (step 110) and the segments are packaged into second DTUs (step 120). Once the second DTUs are created, the extra or redundant second DTUs are encoded and created (step 130). The second DTUs are then buffered (step 140) and then transmitted to the transport network (step 150). Decision 160 then determines if a re-transmission request has been received. IF such a request
 15 is received, then the decision flow returns to step 150, that of transmitting the second DTUs. If no request is received within a set period of time, the connector 170 returns the control flow to step 100.

If a filter is used in the system as illustrated in Fig 3A, once the first DTUs are received, the decision 105 determines if the incoming DTU is to be segmented or not.
 20 If the incoming DTU is to be segmented, then the logic flow continues to step 110, that of segmenting the DTU. The filter can, as noted above, determine how many segments the DTU is to be divided into, how the segmented DTUs are to be routed, etc., during step 105. On the other hand, if the incoming DTU is not to be segmented, then the logic flow moves to step 130, that of generating extra second DTUs (if desired) to
 25 provide a safeguard against the loss of any DTUs.

Referring to Fig 5, an illustration of the steps executed by a server receiving second DTUs from the transport network is presented. The process begins at step 180 as the server receives second DTUs from the transport network. Decision 190 then determines if all the second DTUs have been received to reconstruct the first DTU from
 30 which the second DTUs were received. IF they have been received, then step 200 is that of recreating the original first DTU. Once the first DTU have been recreated, the recreated DTU is transmitted to the destination end network (step 205) and the control

flow moves back to step 180 by way of connector 210.

Returning to decision 190, if not all the second DTUs have been received, then decision 220 determines if the extra second DTUs have been received. If the extra second DTUs have been lost as well, then a re-transmission is requested (step 230).

5 After the re-transmission request, the control flow returns to step 180 by way of connector 210. In the event the extra second DTUs have been received, decision 240 determines if sufficient extra second DTUs and second DTUs have been received to reconstruct the original first DTU. If an insufficient amount has been received, then connector 250 moves the control flow to step 230, that of requesting a re-transmission.

10 If a sufficient amount has been receive, then step 260 is that of decoding the extra second DTUs to recreate or reconstruct the missing second DTUs. Connector 270 then returns the control flow to step 200, that of recreating the first DTU from the received second DTUs.

If the system implements as filter as in Figure 3A, step 185 in Figure 5 is that of

15 filtering the incoming second DTUs. This step would ensure that incoming DTUs which have not been segmented are sent directly to their output ports or are otherwise properly routed. Similarly, if second DTUs are unsuitable for re-transmissions and have been tagged as such, then these second DTUs can be directed used to recreate the first DTUs (step 200). Otherwise, the logic flow continues to decision 190.

20 Embodiments of the invention may be implemented in any conventional computer programming language. For example, preferred embodiments may be implemented in a procedural programming language (e.g. "C") or an object oriented language (e.g. "C++"). Alternative embodiments of the invention may be implemented as pre-programmed hardware elements, other related components, or as a combination

25 of hardware and software components.

Embodiments can be implemented as a computer program product for use with a computer system. Such implementation may include a series of computer instructions fixed either on a tangible medium, such as a computer readable medium (e.g., a diskette, CD-ROM, ROM, or fixed disk) or transmittable to a computer system,

30 via a modem or other interface device, such as a communications adapter connected to a network over a medium. The medium may be either a tangible medium (e.g., optical or electrical communications lines) or a medium implemented with wireless

techniques (e.g., microwave, infrared or other transmission techniques). The series of computer instructions embodies all or part of the functionality previously described herein. Those skilled in the art should appreciate that such computer instructions can be written in a number of programming languages for use with many computer architectures or operating systems. Furthermore, such instructions may be stored in any memory device, such as semiconductor, magnetic, optical or other memory devices, and may be transmitted using any communications technology, such as optical, infrared, microwave, or other transmission technologies. It is expected that such a computer program product may be distributed as a removable medium with accompanying printed or electronic documentation (e.g., shrink wrapped software), preloaded with a computer system (e.g., on system ROM or fixed disk), or distributed from a server over the network (e.g., the Internet or World Wide Web). Of course, some embodiments of the invention may be implemented as a combination of both software (e.g., a computer program product) and hardware. Still other embodiments of the invention may be implemented as entirely hardware, or entirely software (e.g., a computer program product).

A person understanding this invention may now conceive of alternative structures and embodiments or variations of the above all of which are intended to fall within the scope of the invention as defined in the claims that follow.

What is claimed is:

1. A server for forwarding data from a first network to a second network, the server comprising:

a first interface module for sending data to and receiving data from said first network;

a second interface module for sending data to and receiving data from said second network;

a segmenting module for segmenting each first data transmission unit (DTU) of a first type received from said first network into at least two second DTUs of a second type, said segmenting module receiving first DTUs from said first interface module and said segmenting module sending said at least two second DTUs and at least one duplicate second DTU to said second interface module for transmission to said second network;

a reassembly module reassembling first DTUs from second DTUs and duplicate second DTUs received said second network, said reassembly module determining if a predetermined total number of second DTUs and duplicate second DTUs have been received to reassemble a first DTU, said reassembly module sending reassembled first DTUs to said first interface module for transmission to said first network, said reassembly module receiving said second DTUs and duplicate second DTUs from said second network interface, and, in the event the predetermined total number of second DTUs and duplicate second DTUs is not received by said reassembly module, said reassembly module transmits a retransmit request to a sender server by way of said second interface module.

2. A server according to claim 1 wherein in the event said segmenting module receives the retransmit request from the reassembly module, said segmenting module retransmits a requested group of second DTUs by way of said second interface module.

3. A server according to claim 1 wherein said segmenting module interleaves second DTUs from different first DTUs when sending the interleaved second DTUs to said second interface module.

4. A server according to claim 2 wherein said segmenting module buffers second DTUs sent to said second interface module.
5. A server according to claim 1 wherein said segmenting module divides each first DTU into n second DTUs and for each first DTU, sends m duplicate second DTUs to said second interface module, and m is less than or equal to n .
6. A server according to claim 1 wherein at least one of said interface modules includes a filter for filtering DTUs wherein said filter determines which DTUs are to be sent to said segmenting module.
7. A server according to claim 6 wherein said filter determines a segmentation of each first DTU.
8. A server according to claim 6 wherein said filter determines if second DTUs generated from said first DTUs are to be re-transmitted.
9. A server according to claim 8 wherein second DTUs which are not to be re-transmitted are tagged by said server.
10. A system for transmitting data from a first network to a third network by way of a second network, the system comprising:
 - a first server connecting said first network and said second network, said first server receiving first DTUs of a first type from said first network and subdividing each one of said first DTUs into second DTUs of a second type, said second DTUs and at least one duplicate second DTU being transmitted to said second network by said first server;
 - a second server connecting said second network and said third network, said second server receiving said second DTUs and said at least one duplicate second DTU from said second network, and on determining that a predetermined total number of second DTUs and duplicate second DTUs have been received, said second server recreating said first DTUs from said second DTUs and said duplicate second DTUs,

said second server requesting a retransmission of said second DTUs from said first server in the event the predetermined total number of second DTUs and duplicate second DTUs are determined not to have been received, and said second server transmitting said recreated first DTUs to said third network.

11. A system according to claim 10 wherein a filter receives DTUs from said first network and determines which received DTUs are to be subdivided based on at least one criterion selected from the group comprising:

- a destination IP address of the incoming DTU
- a source IP address of the incoming DTU
- a source TCP/UDP port of the incoming DTU
- a destination TCP/UDP port of the incoming DTU
- header information in the incoming DTU
- data in the incoming DTU
- a size or length of the incoming DTU
- an outgoing port or ingress port of the incoming DTU.

12. A system according to claim 10 wherein a filter receives DTUs from said first network and determines which received DTUs are suitable for re-transmission.

13. A system according to claim 12 wherein said filter determines which received DTUs are suitable for re-transmission based on at least one criterion selected from the group comprising:

- a destination IP address of the incoming DTU
- a source IP address of the incoming DTU
- a source TCP/UDP port of the incoming DTU
- a destination TCP/UDP port of the incoming DTU
- header information in the incoming DTU
- data in the incoming DTU
- a size or length of the incoming DTU
- an outgoing port or ingress port of the incoming DTU.

14. A system according to claim 11 wherein DTUs which are not to be subdivided are tagged by said server.

15. A method of forwarding data transmissions from a first network to a third network by way of a second network, said first network producing a first data transmission unit (DTU) of a first type, said first DTU being segmented into second DTUs of a second type, the method comprising:

a) receiving from said second network second DTUs and at least one duplicate second DTU;

b) determining if a predetermined total number of second DTUs and duplicate second DTUs have been received to recreate a first DTU, said determining step being based on how many second DTUs and duplicate second DTUs have been received;

c) in the event the predetermined total number of second DTUs and duplicate second DTUs have not been received, requesting a re-transmission of said second DTUs and duplicate second DTUs and re-executing step b);

d) in the event the predetermined total number of second DTUs and duplicate second DTUs have been received, recreating said first DTU; and

e) transmitting said recreated first DTU to said third network.

16. A method according to claim 15 wherein said second DTUs and duplicate second DTUs received from said second network are interleaved with second DTUs and duplicate second DTUs from different first DTUs.

17. A method of forwarding data transmissions from a first network to a third network via a second network, the method comprising:

a) receiving first data transmission units (DTUs) of a first type from said first network;

b) for each first DTU of a first type, segmenting said first DTU into second DTUs of a second type;

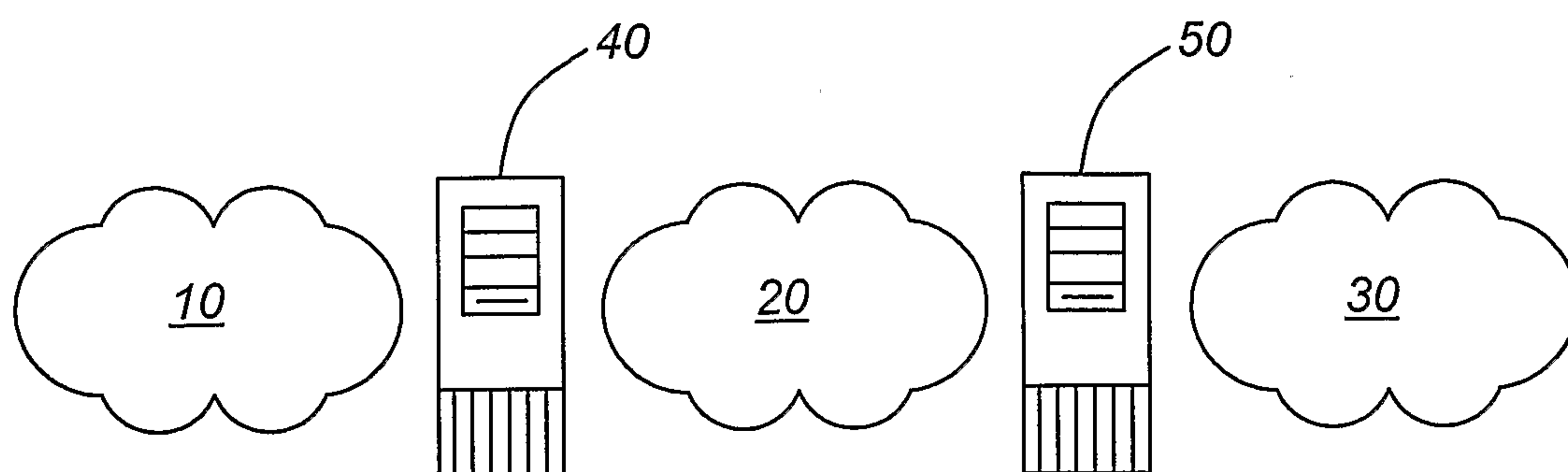
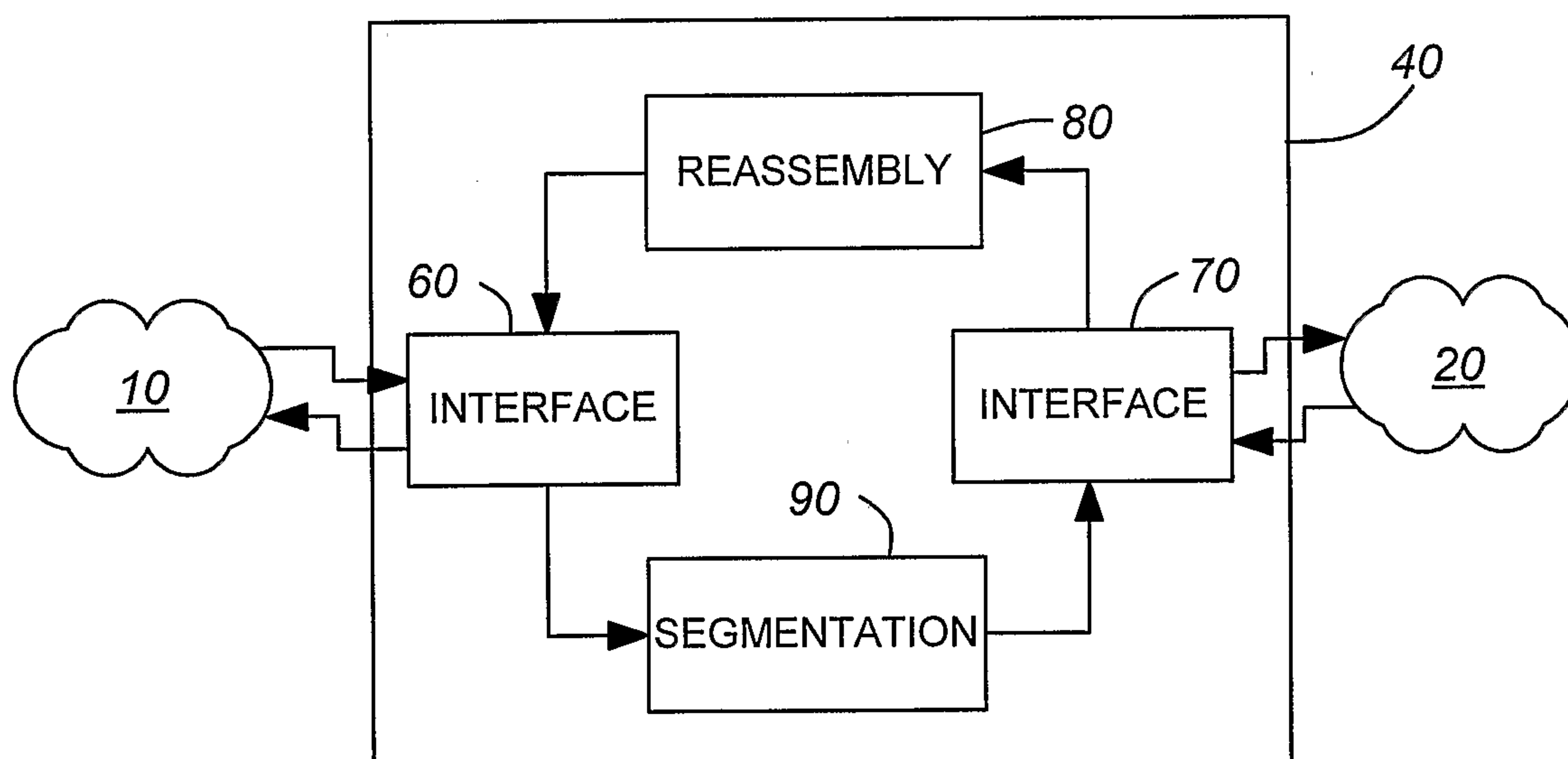
c) transmitting said second DTUs to said second network;

d) transmitting at least one duplicate second DTU to said second network, wherein said at least one duplicate second DTU allows said first DTU to be recreated

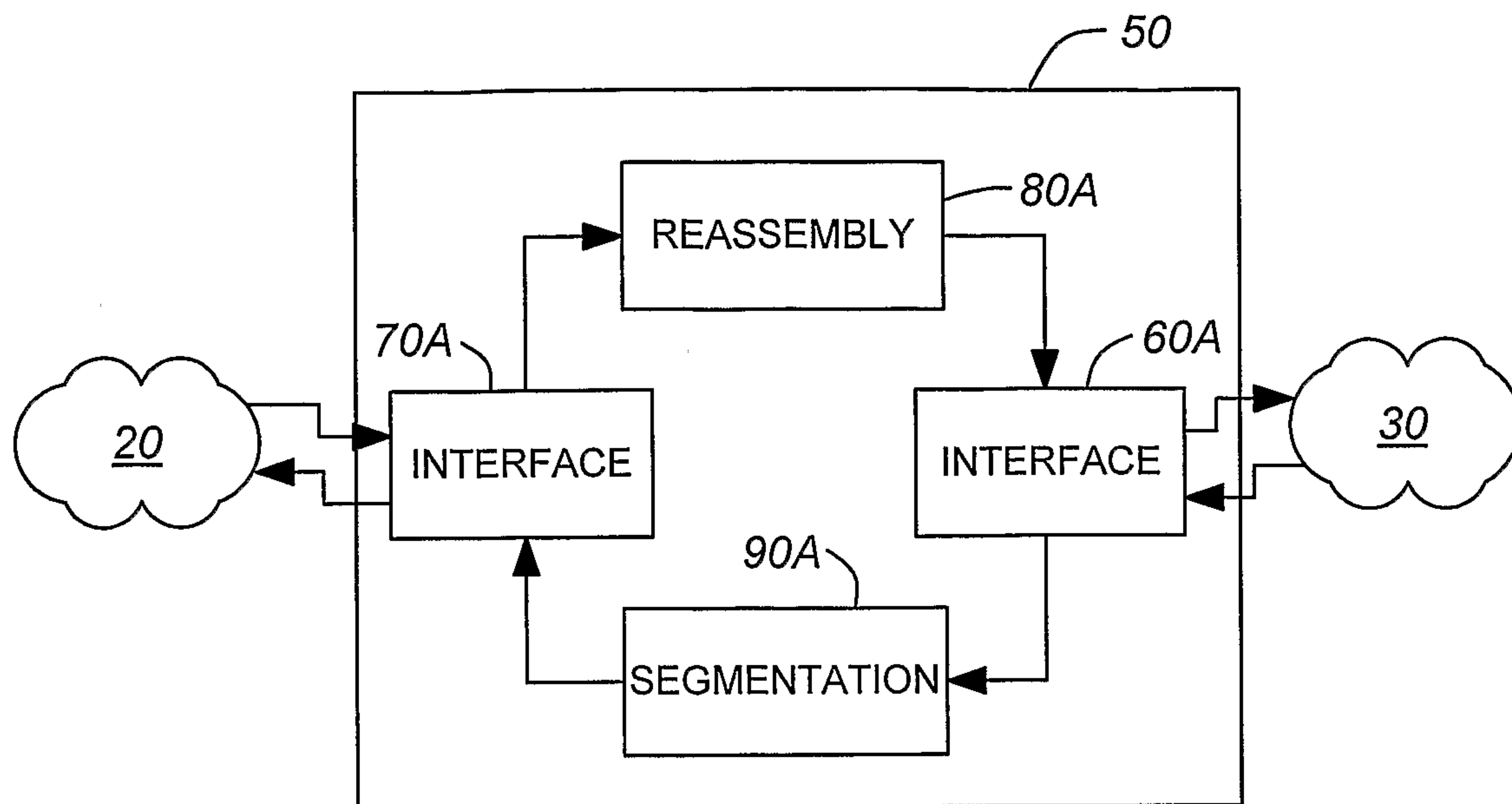
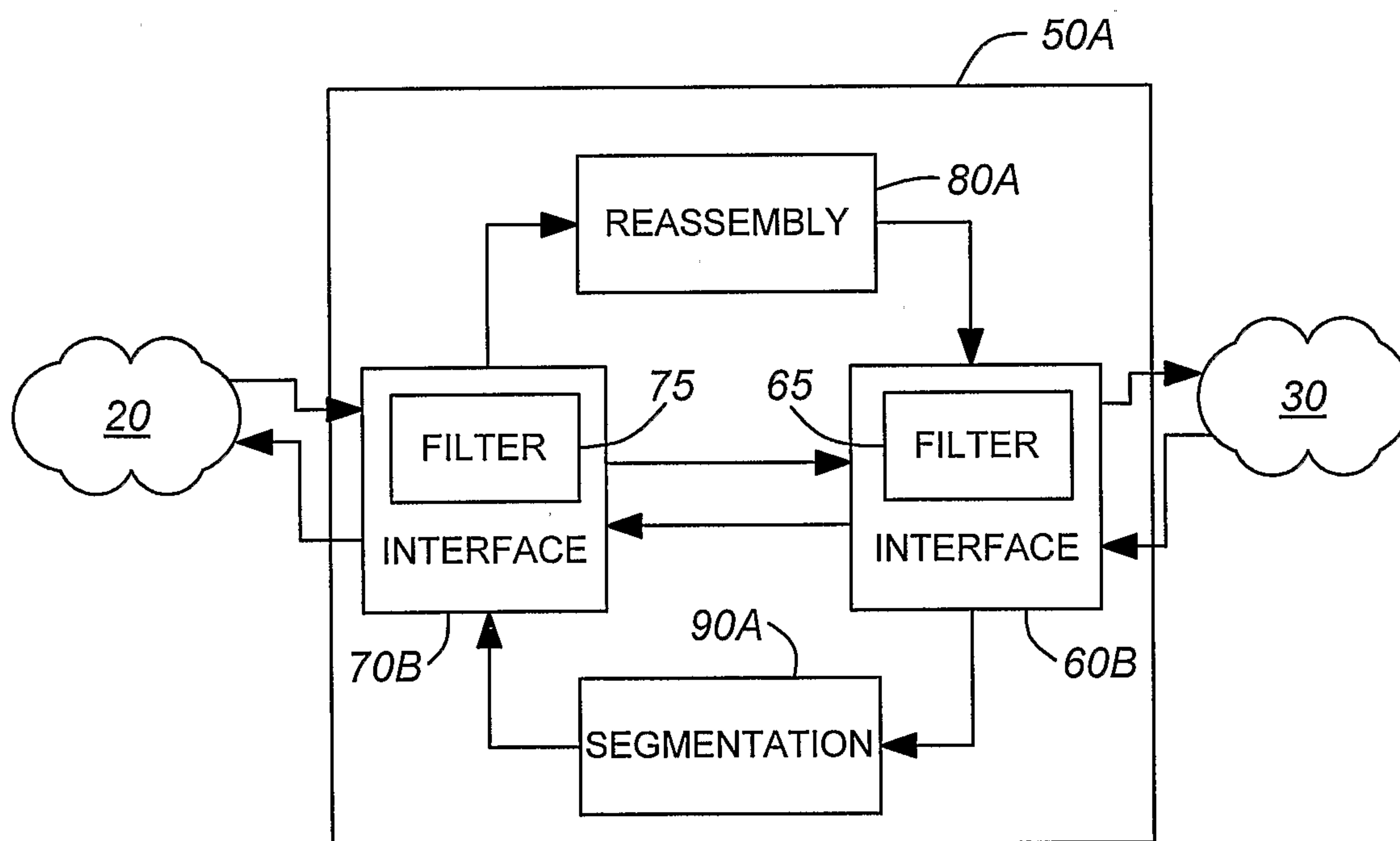
in the event not all second DTUs are received at said second network; and

e) in the event a predetermined total number of second DTUs and duplicate second DTUs have not been received to recreate said first DTU, requesting a re-transmission of said second DTUs and said at least one duplicate DTU.

18. A method according to claim 17 further including the step of interleaving second DTUs and duplicate second DTUs from different first DTUs prior to transmitting them.

**FIG. 1****FIG. 2**

2/4

**FIG. 3****FIG. 3A**

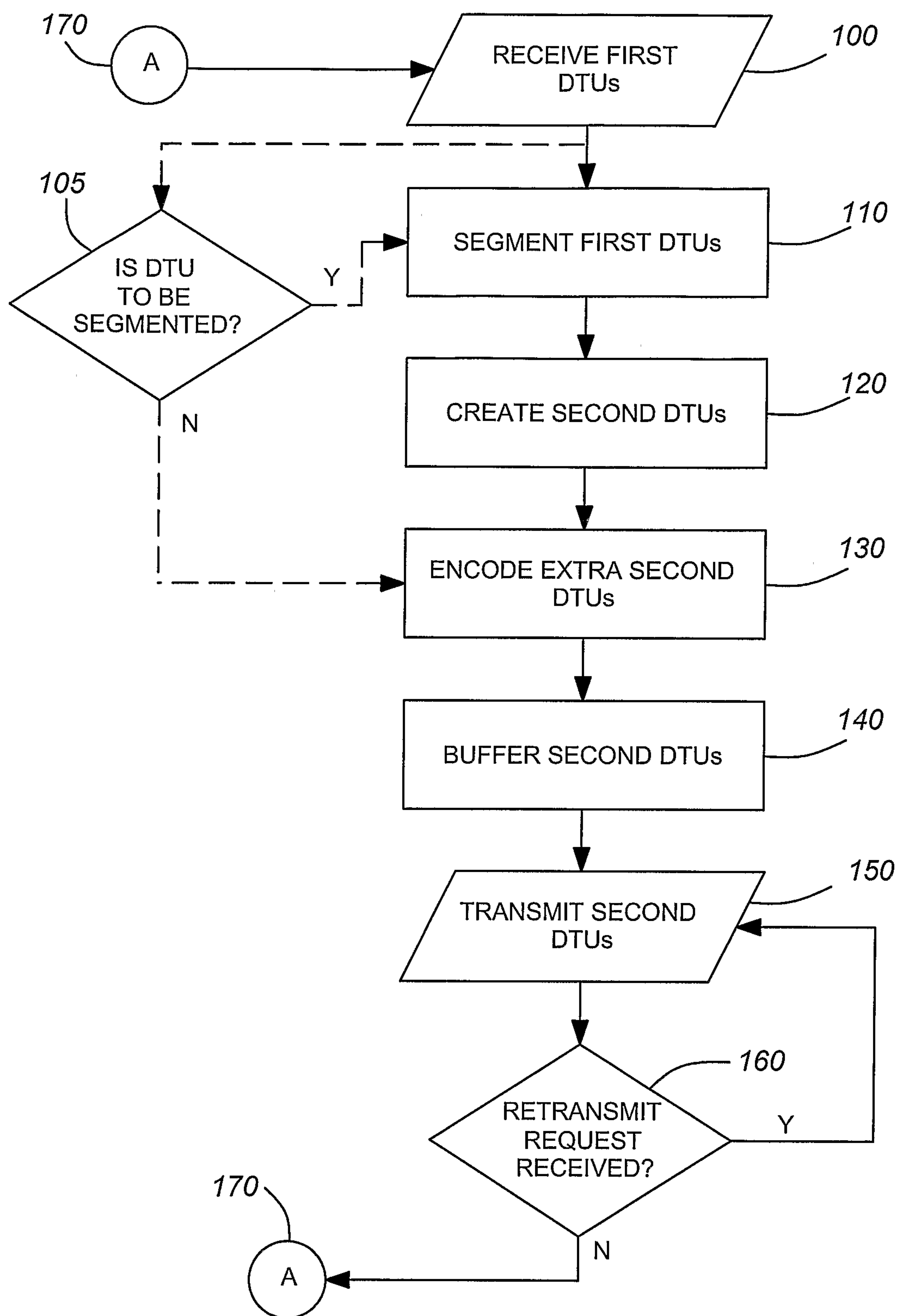


FIG. 4

4/4

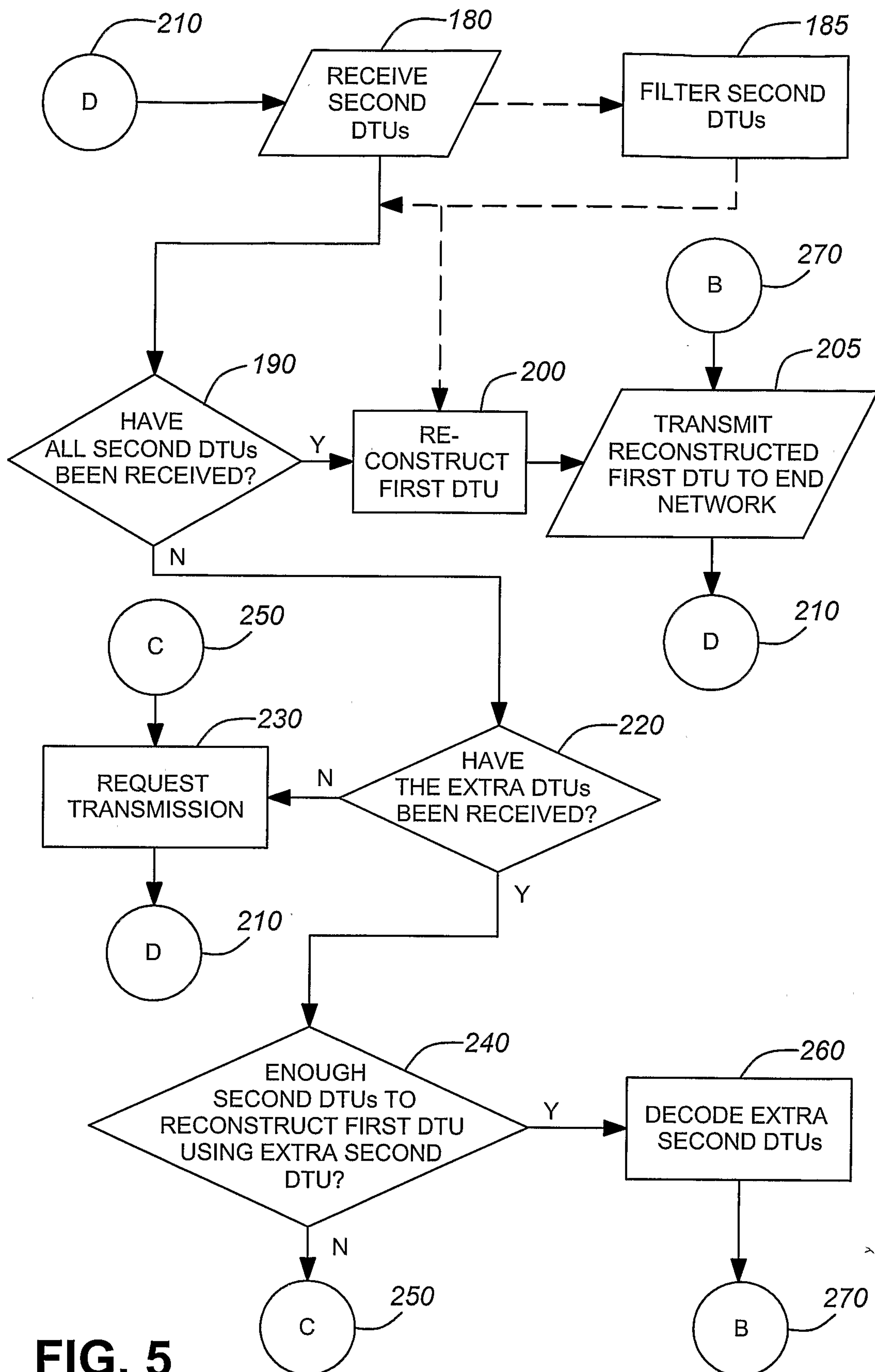


FIG. 5

