Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3086895 A1 2019/05/23

(21) 3 086 895

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2018/11/20
(87) Date publication PCT/PCT Publication Date: 2019/05/23
(85) Entrée phase nationale/National Entry: 2020/06/24
(86) N° demande PCT/PCT Application No.: NZ 2018/050164
(87) N° publication PCT/PCT Publication No.: 2019/098860
(30) Priorité/Priority: 2017/11/20 (NZ737496)

(51) Cl.Int./Int.Cl. *H04L 12/46* (2006.01), *H04L 29/06* (2006.01)

(71) Demandeur/Applicant:
MAKO NETWORKS NZ LIMITED, NZ

(72) Inventeur/Inventor:
COPLEY, MICHAEL ALAN, NZ

(74) Agent: GOWLING WLG (CANADA) LLP

(54) Titre : PROCEDE ET SYSTEME DE TRANSMISSION DE DONNEES
(54) Title: METHOD AND SYSTEM FOR TRANSMITTING DATA

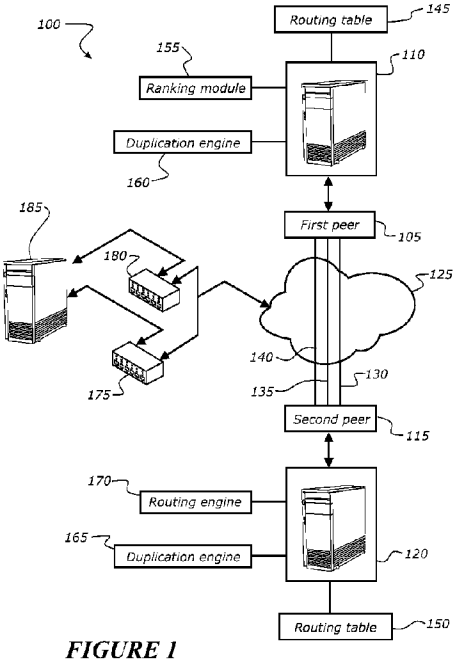

FIGURE 1

(57) Abrégé/Abstract:
An aspect of the invention provides a data network (100) configured to transmit data from a first peer (105). The network comprises a ranking module (155) associated to the first peer, the ranking module configured to measure the data against an importance threshold. A duplication engine (160) is configured, on determining that the data satisfies an importance threshold, to: assemble a first data packet having a first tunnel identifier associated to the first tunnelling session (130), a first primary packet sequence identifier, and a secondary packet sequence identifier; and assemble a second data packet having a second tunnel identifier associated to the second tunnelling session (135), a second primary packet sequence identifier, and the secondary packet sequence identifier. The first peer (105) is configured to: transmit the first data packet from the first peer (105) within the first tunnelling session (130); and transmit the second data packet from the first peer (105) within the second tunnelling session (135).

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 May 2019 (23.05.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/098860 A1

(51) International Patent Classification:
H04L 12/46 (2006.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/NZ2018/050164

(22) International Filing Date:
20 November 2018 (20.11.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
737496 20 November 2017 (20.11.2017) NZ

(71) Applicant: **MAKO NETWORKS LIMITED** [NZ/NZ]; 7-62 Paul Matthews Road, Albany, Auckland, 0632 (NZ).

(72) Inventor; and
(71) Applicant: **COPLEY, Michael Alan** [NZ/NZ]; 8/62 Paul Matthews Road, Albany, Rosedale, Auckland, 0632 (NZ).

(74) Agent: **AJ PARK**; Level 22, Aon Centre, 1 Willis Street, Wellington, 6011 (NZ).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*

(54) **Title:** METHOD AND SYSTEM FOR TRANSMITTING DATA

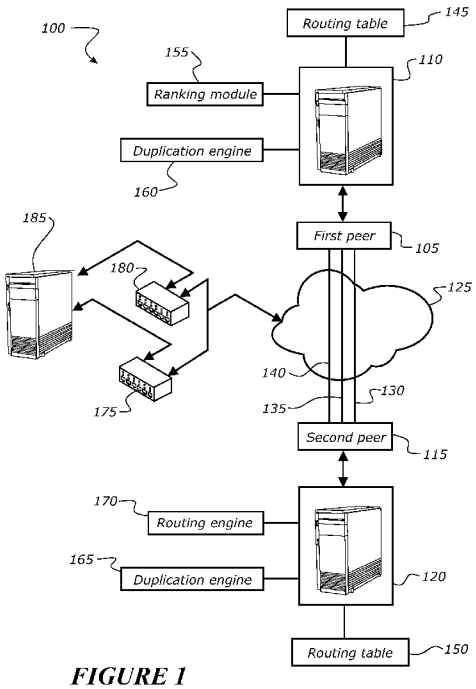

FIGURE 1

(57) **Abstract:** An aspect of the invention provides a data network (100) configured to transmit data from a first peer (105). The network comprises a ranking module (155) associated to the first peer, the ranking module configured to measure the data against an importance threshold. A duplication engine (160) is configured, on determining that the data satisfies an importance threshold, to: assemble a first data packet having a first tunnel identifier associated to the first tunnelling session (130), a first primary packet sequence identifier, and a secondary packet sequence identifier; and assemble a second data packet having a second tunnel identifier associated to the second tunnelling session (135), a second primary packet sequence identifier, and the secondary packet sequence identifier. The first peer (105) is configured to: transmit the first data packet from the first peer (105) within the first tunnelling session (130); and transmit the second data packet from the first peer (105) within the second tunnelling session (135).

WO 2019/098860 A1

METHOD AND SYSTEM FOR TRANSMITTING DATA

FIELD OF INVENTION

The invention generally relates to methods and systems for transmitting data between peers in Virtual Private Networks (VPNs). More particularly the invention relates to VPNs with provision for channel independent tunnel identification and redirection.

BACKGROUND OF THE INVENTION

A Virtual Private Network (VPN) is effectively a tunnel through a network between two cooperating routing processes. The tunnel is established by encapsulating network packets in a layer after encrypting them. The layer in which the packets are encrypted may be of various types dependent on the VPN protocol chosen.

A well-known encryption protocol is IPSec which provides the ability to encrypt either just the payload of a packet (transport mode) or to encrypt a complete packet (tunnel mode).

In the tunnel mode of IPSec, the entire packet is encrypted and/or authenticated usually within a layer level delivery protocol equal to or higher than the payload protocol. Typically, the external protocol is ESP/IP or UDP/IP, whose packets carry no information about the payload they are carrying within it.

Usually the tunnel is established with a tunnelling protocol such as IKE and ISAKMP in which a client process calls a server process to set up a session for a tunnel in which IPSec packets and their payloads are carried within the tunnelling protocol packets.

While these packets carry a tunnel id or security parameter index (SPI) the packet is reliant on the channel over which the packets are provided being correctly identified in order for the packet to be processed. If the channel on which the packets are being sent changes in any way, the packets will not be recognised.

This means that if a tunnel with IP packets carrying tunnelling protocol and IPSec packets is disrupted there is no way for the client end to provide any information to the server end, since any attempt to return information to the originating IP address cannot be related to any tunnel data and has no authentication information. Normally it is

required to re-establish a tunnel anew from the establishing end, something which takes an appreciable time.

Similarly, if the routing of a tunnel changes, for instance by the insertion of a new network behind the remote server, there is no easy method of advising the local end of the change.

None of the several tunnelling protocols (IPSec, SSL, PPTP/L2TP based for the most part) resolve these problems, requiring the complete re-establishment of a new tunnel to recommence the tunnel traffic. Typically this requires a time lapse sufficient that any transaction ongoing will lapse.

Therefore there is a need to provide some method of detecting the interruption of a tunnel and providing a method of rapidly re-establishing the same or another tunnel for the required traffic.

It is an object of at least preferred embodiments of the present invention to address at least some of the aforementioned disadvantages.

An additional or alternative object is to provide a tunnelling protocol capable of replacing or encapsulating IPSec packets which resolves this issue and others caused by the transient nature of a tunnel.

An additional or alternative object is to provide a means of confirming that a tunnel is still operational despite the lack of any user data packets being transferred.

An additional or alternative object is to at least provide the public with a useful choice.

SUMMARY OF THE INVENTION

In accordance with an aspect of the invention, a method of transmitting data between a first peer and a second peer comprises, on determining that the data satisfies an importance threshold: transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and transmitting, from the first peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier. On determining that the data does not satisfy an importance threshold the method includes transmitting, from the first

peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, and a first primary packet sequence identifier.

The term 'comprising' as used in this specification means 'consisting at least in part of'. When interpreting each statement in this specification that includes the term 'comprising', features other than that or those prefaced by the term may also be present. Related terms such as 'comprise' and 'comprises' are to be interpreted in the same manner.

In an embodiment the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

In an embodiment the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the first routing version identifier maintained in computer memory associated to the first peer.

In an embodiment the method further comprises maintaining the first routing version identifier in computer memory within a routing table.

In an embodiment the first data packet comprises one of a network content packet, a keep-alive packet, a routing table packet.

In an embodiment the second data packet comprises a network content packet.

In an embodiment the first data packet comprises a keep-alive packet, the keep-alive packet including the first routing version identifier.

In accordance with a further aspect of the invention, a method of transmitting data between a first peer and a second peer comprises: transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and transmitting, from the first peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier.

In accordance with a further aspect of the invention, a method of resolving data transmitted between a first peer and a second peer comprises receiving, at the second

peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; receiving, at the second peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier; and discarding the first data packet or the second data packet on determining that both the first data packet and the second data packet include the secondary packet sequence identifier.

In an embodiment the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

In an embodiment the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the first routing version identifier maintained in computer memory associated to the second peer.

In an embodiment the first data packet includes a routing version identifier associated to the first data packet, the method further comprising transmitting a routing table packet on determining a mismatch between the first routing version identifier and the routing version identifier associated to the first data packet.

In an embodiment the routing table packet includes the first routing version identifier.

In an embodiment the method further comprises maintaining the first routing version identifier in computer memory within a routing table.

In an embodiment the first data packet comprises a keep-alive packet.

In accordance with a further aspect of the invention, a data network configured to transmit data from a first peer comprises a ranking module associated to the first peer, the ranking module configured to measure the data against an importance threshold; and a duplication engine configured, on determining that the data satisfies an importance threshold, to: assemble a first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and assemble a second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier. The first peer is configured to: transmit the first data packet from the first peer within the first

tunnelling session; and transmit the second data packet from the first peer within the second tunnelling session.

In an embodiment the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

In an embodiment the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the first routing version identifier maintained in computer memory associated to the first peer.

In an embodiment the network further comprises a routing table stored in computer memory within which the first routing version identifier is maintained.

In accordance with a further aspect of the invention, a data network configured to resolve data received from a first peer comprises a second peer configured to: receive a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier, and receive a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier; and a duplication engine configured to discard the first data packet or the second data packet on determining that both the first data packet and the second data packet include the secondary packet sequence identifier.

In an embodiment the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

In an embodiment the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the first routing version identifier maintained in computer memory associated to the second peer.

In an embodiment the first data packet includes a routing version identifier associated to the first data packet, the data network further comprising a routing engine configured to transmit a routing table packet on determining a mismatch between the first routing version identifier and the routing version identifier associated to the first data packet.

In an embodiment the routing table packet includes the first routing version identifier.

In an embodiment the data network further comprises a routing table stored in computer memory within which the first routing version identifier is maintained.

In accordance with a further aspect of the invention, a computer readable medium has stored thereon computer-executable instructions that, when executed by a processor, cause the processor to perform a method of transmitting data between a first peer and a second peer. The method comprises, on determining that the data satisfies an importance threshold: transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and transmitting, from the first peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier. The method further comprises, on determining that the data does not satisfy an importance threshold, transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, and a first primary packet sequence identifier.

In an embodiment the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

In an embodiment the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier. The method further comprises maintaining the first routing version identifier in computer memory associated to the first peer.

In an embodiment the method further comprises maintaining the first routing version identifier in computer memory within a routing table.

In an embodiment the first data packet comprises one of a network content packet, a keep-alive packet, a routing table packet.

In an embodiment the second data packet comprises a network content packet.

In an embodiment the first data packet comprises a keep-alive packet, the keep-alive packet including the first routing version identifier.

In accordance with a further aspect of the invention, a computer readable medium has stored thereon computer-executable instructions that, when executed by a processor, cause the processor to perform a method of transmitting data between a first peer and a

second peer. The method comprises: transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and transmitting, from the first peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier.

In accordance with a further aspect of the invention, a computer readable medium has stored thereon computer-executable instructions that, when executed by a processor, cause the processor to perform a method of resolving data transmitted between a first peer and a second peer. The method comprises: receiving, at the second peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; receiving, at the second peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier; and discarding the first data packet or the second data packet on determining that both the first data packet and the second data packet include the secondary packet sequence identifier.

In an embodiment the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

In an embodiment the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier. The method further comprises maintaining the first routing version identifier in computer memory associated to the second peer.

In an embodiment the first data packet includes a routing version identifier associated to the first data packet. The method further comprises transmitting a routing table packet on determining a mismatch between the first routing version identifier and the routing version identifier associated to the first data packet.

In an embodiment the routing table packet includes the first routing version identifier.

In an embodiment the method further comprises maintaining the first routing version identifier in computer memory within a routing table.

In an embodiment the first data packet comprises a keep-alive packet.

The terms 'component', 'module', 'system', 'interface', and/or the like as used in this specification in relation to a processor are generally intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a controller and the controller can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

The term 'connected to' as used in this specification in relation to data or signal transfer includes all direct or indirect types of communication, including wired and wireless, via a cellular network, via a data bus, or any other computer structure. It is envisaged that they may be intervening elements between the connected integers. Variants such as 'in communication with', 'joined to', and 'attached to' are to be interpreted in a similar manner. Related terms such as 'connecting' and 'in connection with' are to be interpreted in the same manner.

The term 'channel' as used in this specification includes one or more of the list comprising network connection type, network IP address, and network port. An example of a network connection type includes 'UDP'. An example of a network IP address includes 203.0.113.1. An example of a network port includes '2453'.

Within this specification the terms 'server' and 'client' are interchangeable in that they are roles rather than design factors in hardware. Similarly 'local' and 'remote' do not distinguish differences in hardware or function but serve to differentiate relative locations.

The invention in one aspect comprises several steps. The relation of one or more of such steps with respect to each of the others, the apparatus embodying features of construction, and combinations of elements and arrangement of parts that are adapted to affect such steps, are all exemplified in the following detailed disclosure.

To those skilled in the art to which the invention relates, many changes in construction and widely differing embodiments and applications of the invention will suggest themselves without departing from the scope of the invention as defined in the appended claims. The disclosures and the descriptions herein are purely illustrative and are not intended to be in any sense limiting. Where specific integers are mentioned herein which have known equivalents in the art to which this invention relates, such known equivalents are deemed to be incorporated herein as if individually set forth.

In addition, where features or aspects of the invention are described in terms of Markush groups, those persons skilled in the art will appreciate that the invention is also thereby described in terms of any individual member or subgroup of members of the Markush group.

As used herein, '(s)' following a noun means the plural and/or singular forms of the noun.

As used herein, the term 'and/or' means 'and' or 'or' or both.

It is intended that reference to a range of numbers disclosed herein (for example, 1 to 10) also incorporates reference to all rational numbers within that range (for example, 1, 1.1, 2, 3, 3.9, 4, 5, 6, 6.5, 7, 8, 9, and 10) and also any range of rational numbers within that range (for example, 2 to 8, 1.5 to 5.5, and 3.1 to 4.7) and, therefore, all sub-ranges of all ranges expressly disclosed herein are hereby expressly disclosed. These are only examples of what is specifically intended and all possible combinations of numerical values between the lowest value and the highest value enumerated are to be considered to be expressly stated in this application in a similar manner.

In this specification where reference has been made to patent specifications, other external documents, or other sources of information, this is generally for the purpose of providing a context for discussing the features of the invention. Unless specifically stated otherwise, reference to such external documents or such sources of information is not to be construed as an admission that such documents or such sources of information, in any jurisdiction, are prior art or form part of the common general knowledge in the art.

In the description in this specification reference may be made to subject matter which is not within the scope of the appended claims. That subject matter should be readily identifiable by a person skilled in the art and may assist in putting into practice the invention as defined in the presently appended claims.

Although the present invention is broadly as defined above, those persons skilled in the art will appreciate that the invention is not limited thereto and that the invention also includes embodiments of which the following description gives examples.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred forms of the method and system for transmitting data will now be described by way of example only with reference to the accompanying figures in which:

Figure 1 shows a block diagram of an embodiment of a VPN;

Figure 2 shows a method for establishing a tunnel between a first peer and a second peer;

Figure 3 shows a method for transmitting a data packet from a peer;

Figure 4 shows a method for assembling the data packet as part of the method of figure 3;

Figure 5 shows a method of resolving data packets received at a peer;

Figure 6 shows an example of a method for processing a data packet when received by a peer as described above with reference to figure 5; and

Figure 7 shows an example of a computing device configured to implement at least part of the network of figure 1.

DETAILED DESCRIPTION

Figure 1 shows a Virtual Private Network (VPN) 100 in accordance with an embodiment of the invention. The network includes a first peer 105 connected to a first peer local network 110. The first peer 105 may also be referred to as an initiator peer.

In an embodiment a second peer 115 is connected to a second peer local network 120. In an embodiment network 120 is remote to network 110. The second peer 115 may also be referred to as a responder peer.

First peer 105 and second peer 115 are connected to each other via the internet 125 or other suitable network or combination of networks, so as to exchange data between peer 105 and peer 115. Examples of suitable networks include fibre networks, Wi-Fi networks, and cellular networks.

In an embodiment the network 100 further includes a plurality of tunnelling sessions or tunnels. Examples of tunnels are indicated at 130, 135 and 140. It will be appreciated that at least one of the tunnels has a path between peer 105 and peer 115 that involves a combination of networks.

In an embodiment the first peer 105 and/or the second peer 115 includes a server connected to a router (not shown). The router facilitates connection to network 125 via a VPN.

In an embodiment the first peer 105 and/or the second peer 115 includes a smartphone, tablet, wearable device or similar device running a VPN directly from the device rather than through a router. Such a device may be configured to include two or more connections over which data packets may be exchanged. Examples of connections include Wi-Fi network and cellular network connections.

In an embodiment a routing table 145 is maintained in computer memory associated to the first peer 105. Routing table 150 is maintained in computer memory associated to the second peer 115.

The operation of the tunnels 130, 135 and 140 and the routing tables 145 and 150 are further described below. References below to tunnels between peers include references to tunnels between routers associated to respective peers.

In an embodiment there are at least two tunnels between first peer 105 and second peer 115. In an embodiment there exist multiple tunnels between first peer 105 and second peer 115. For example, there could be two tunnels, three tunnels, or more than three tunnels.

In an embodiment a first tunnel has a first path or channel between first peer 105 and second peer 115. A second tunnel has a second path or channel between first peer 105 and second peer 115. In an embodiment the first path is different to the second path. For example, the first path in an embodiment comprises a fibre channel while the second path comprises a cellular channel.

In an embodiment the tunnels are associated to respective paths or channels, the paths or channels are different to each other. The use of different paths or channels for different tunnels has the potential to provide increased reliability of the network as the failure of one path or channel simply means that other paths or channels are available within which to exchange data between the first peer 105 and the second peer 115.A first duplication engine 160 and a ranking module 155 are associated to first peer 105. The operation of first duplication engine 160 and ranking module 155 is further described below. In summary the ranking module determines which of the data to be transmitted from first peer 105 is important or critical. The first duplication engine 160 creates duplicate packets of important or critical data to be transmitted within different tunnels.

A second duplication engine 165 is associated to second peer 115. The second duplication engine 165 detects duplicate packets created by the duplication engine 160. Second duplication engine 165 discards sufficient duplicate data packets so as to remove duplicates.

Routing engine 170 is associated to second peer 115. The operation of the routing engine 170 is further described below. In summary the routing engine 170 detects and resolves changes in routes or paths between first peer 105 and second peer 115.

In an embodiment the network 100 includes VPN routers 175 and 180 each capable of acting as VPN concentrators and connected to a local network represented by server 185.

In an embodiment, each of the VPN routers 175 and 180 is a remotely managed router managed from network 185 via the internet 125 and via VPN tunnels (not shown) set up initially from default settings in the routers, but eventually by individual VPN tunnels allocated to each router.

In an embodiment each router 110, 120, 175, 180 is configured to manage a plurality of VPN tunnels. Some of the tunnels may tunnel through other VPN routers for instance to reach a secure location inside a large local network.

Figure 2 shows a method 200 for establishing a tunnel between the first peer 105 and the second peer 115. First peer 105 receives 205 a channel associated to second peer 115. First peer 105 connects to second peer 115 using security provisions over the channel associated to second peer 115.

First peer 105 and second peer 115 negotiate 215 the details of a VPN session. In an embodiment the details include one of more of a session key with an identifier, an identifier for the new tunnel, an initial primary packet sequence identifier for the first 20 packets, details of the channel over which the connection will be made.

In an embodiment the negotiated session details are stored 220 in a session table associated to the first peer 105 and/or stored in a session table associated to the second peer 115.

The tunnel is established 225 between the first peer 105 and second peer 115. In an embodiment the method 200 is repeated so as to establish a plurality of tunnels between the first peer 105 and the second peer 115.

In an embodiment, when the initiator/first peer 105 establishes a session, the first peer 105 indicates to the responder/second peer 115 that it supports a protocol that permits duplicate data packets to be sent over multiple tunnels. This avoids the need for the responder/second peer 115 to be preconfigured with knowledge of the shared sequence/packet duplication. The responder/second peer 115 will also then selectively duplicate user data heading from it to the initiator.

The duplication of data packets is described in more detail below.

Figure 3 shows an example method 300 for transmitting a data packet from a peer. The method 300 is described in relation to the first peer 105. It will be appreciated that the same method is used to transmit a data packet from the second peer 115.

The first peer 105 assembles 305 a first data packet. Process 305 is described in more detail below.

The first peer 105 checks 310 to ascertain whether or not the data to be transmitted is critical or important. In an embodiment the data is checked to determine whether it meets an importance threshold. The importance threshold is typically user-defined.

In an embodiment the importance threshold is user-defined for example specifying in configuration files how to mark traffic as critical.

One example of an importance threshold is a purpose or application associated to the transfer of data between first peer 105 and second peer 115. For example, anything in real time, in particular Voice over IP (VoIP) and video calls, is critical. Other applications such as card payment transactions may also be deemed critical. On the other hand, applications such as web traffic or file uploads are not critical.

In this example the importance threshold test comprises 'Does the application associated to a data packet match at least one predefined application selected from voice, video and card payment transactions? If so, the data packet satisfies the importance threshold and is deemed critical. If not, the data packet does not satisfy the importance threshold'.

Another example of an importance threshold test comprises 'Does the application associated to a data packet involve operate in real time? If so, the data packet satisfies the importance threshold and is deemed critical. If not, the data packet does not satisfy the importance threshold'.

In an embodiment the importance threshold is set as IP traffic with an 'expedited forwarding' (EF) DSCP value set. In this example the threshold test comprises 'Does the Internet Protocol (IP) header associated to a data packet have a DSCP EF flag set? If so, the data packet satisfies the importance threshold and is deemed critical. If not, the data packet does not satisfy the importance threshold'.

In an embodiment the importance threshold is set by defining values of a 5-tuple including IP protocol, sender IP, sender port, destination IP, destination port.

In this example the threshold test comprises 'Does the value of at least one of the IP protocol, sender IP, sender port, destination IP, and destination port associated to a data packet match a predefined value? If so, the data packet satisfies the importance threshold and is deemed critical. If not, the data packet does not satisfy the importance threshold'.

In this example another example of threshold test comprises 'Does the value of each of the IP protocol, sender IP, sender port, destination IP, and destination port associated to a data packet match respective predefined values? If so, the data packet satisfies the importance threshold and is deemed critical. If not, the data packet does not satisfy the importance threshold'.

If the first data packet 305 contains critical data the first peer 105 assembles 315 a second data packet. In an embodiment the second data packet is a duplicate of the first data packet.

Whether or not the first data packet 305 contains critical data, the first peer 105 transmits 320 the first data packet over a first tunnel. In an embodiment the second peer 115 receives the first data packet as will be further described below.

In an embodiment the first data packet 305 includes a first tunnel identifier associated to the first tunnel. The first tunnel identifier signals to all components of the network 100 that the first data packet is to be, and has been, transmitted within the first tunnel. An example of the first tunnel is tunnel 130.

In an embodiment the first data packet further includes a first primary packet sequence identifier. As described above negotiation 215 of the details of a VPN session includes determining an initial primary packet sequence identifier in addition to a tunnel identifier.

In an embodiment the first primary packet sequence identifier is unique within the first tunnel. It is derived by incrementing either an initial primary packet sequence identifier or a previous primary packet sequence identifier.

Where the first data packet contains critical data, the first peer 105 transmits 325 the second data packet over a second tunnel.

In an embodiment the second data packet includes a second tunnel identifier associated to the second tunnel. The second tunnel identifier signals to all components of the network 100 that the second data packet is to be, and has been, transmitted within the second tunnel. An example of the second tunnel is tunnel 135.

In an embodiment the second data packet further includes a second primary packet sequence identifier. In an embodiment the second primary packet sequence identifier is unique within the second tunnel. It is derived by incrementing either an initial primary packet sequence identifier within the second tunnel or a previous primary packet sequence identifier associated to the second tunnel.

In an embodiment the first data packet and the second data packet both include a common secondary packet sequence identifier. The presence of a secondary packet sequence identifier within a data packet indicates to all components of the network 100 that the data packet is a duplicate. The value of the common secondary packet sequence identifier indicates to all components of the network 100 that a pair of data packets are duplicates of each other.

In an embodiment the first peer 105 as initiator peer determines an initial value for the secondary packet sequence identifier. The secondary packet sequence identifier is incremented each time a sending peer has a critical data packet requiring duplication. An example of an initial value for the secondary packet sequence identifier is the value '1'. An example of incrementing the secondary packet sequence identifier includes adding the value '1' to the current value of the secondary packet sequence identifier.

Figure 4 shows a method for assembling 305 the data packet described above with reference to figure 3.

The data to be transmitted is fetched 400 from the first peer 105 and formed into at least one data packet. The first peer 105 has available to it details of the tunnel through which data packets encapsulating the fetched data will be sent. The data route through the tunnel is verified 405 in relation to a data packet. If the data route cannot be verified then the packet is discarded.

Identifiers required for the data packet are retrieved 410 from session details stored in a session table maintained in computer memory. Session details maintained in the session table include information about a peer such as its IP address, encryption/decryption keys, a current primary packet sequence identifier, and a current secondary packet sequence identifier. In an embodiment the identifiers include one or more of a session identifier, a tunnel identifier, a primary packet sequence identifier.

In an embodiment the payload of the data packet is encrypted 415 with a key using at least some of the above identifiers.

The correct peer channel for the tunnel is then retrieved 420 as the address retrieved from the last incoming packet.

In an embodiment a user defines in advance which channels to send critical data over. For example a default setting may comprise send data packets over all available channels between first peer 105 and second peer 115.

The user may wish to only send over specific channels. In one example the user might have available a first channel comprising a fibre internet connection, a second channel comprising a fibre internet connection, and a third channel comprising a cellular connection as a backup. Cellular networks typically charge by the amount of traffic sent. The user may wish to send duplicated critical data over the first channel and the second channel but not the third channel in an effort to minimise cost.

The data packet is then transmitted 320 as described above with reference to figure 3. In an embodiment the data packet is transmitted to the second peer 115.

Figure 5 shows an example method 500 of resolving data packets received at a peer. The method 500 is described in relation to the second peer 115. It will be appreciated that the same method is used to transmit a data packet from the first peer 105.

A first data packet is received 505 at the second peer 115. The second peer 115 performs processing on the first data packet as will be further described below.

In an embodiment the first data packet includes an indicator of packet type. Examples of packet types include encapsulated network content packets, keep-alive packets, and routing table packets.

The first data packet is checked 510 to determine whether the first data packet is a keep-alive packet. A keep-alive packet contains information that is extracted from a peer. The keep-alive packet is either responded to or used to update a peer state.

In an embodiment a keep-alive packet is transmitted intermittently during normal use and also when a tunnel is not otherwise being used. The keep-alive packet may include a first routing version identifier.

Where the first data packet is a keep-alive packet, the route associated to the first data packet is verified 515. The first routing version identifier represents a version of a routing table associated to the first peer 105. The first routing version identifier is checked against a routing version identifier maintained in routing table 150 associated to second peer 115.

There may be periods when there is no traffic through a tunnel, and therefore no information as to whether a given tunnel is actually working. One function of the keep-alive packets is to detect that a tunnel is operational.

Where there is a match between the routing version identifiers in the first data packet and the routing table 150 respectively, the second peer 115 transmits a simple response data packet that confirms the tunnel is still operational.

Routes may not remain constant and there is therefore a need to advise the peers at each end of a tunnel of any changes which occur at either end. On detecting a mismatch between the routing version identifiers in the first data packet and the routing table 150 respectively, the second peer 115 transmits a routing table packet.

A routing table packet is a packet that is sent, for example, when the second peer 115 detects from reading a keep-alive packet that the routing table for this tunnel is in error. In an embodiment the second peer 115 returns a routing table packet containing the actual current version of its routing table and an array carrying the entries in the routing table.

If the first data packet is a routing table data packet 520, for example received at the second peer 115 from the first peer 105, the routing table associated to the second peer 115 is updated by updating the routing table 150. This change has the effect of changing the version number.

In an embodiment the method includes resolving 525 duplicates. Keep-alive packets and routing table packets would not normally contain critical data so would not usually be duplicated. Other types of packets are checked to detect the presence of a shared secondary packet sequence identifier within two or more data packets received at the second peer 115.

Where two or more data packets include a shared secondary packet sequence identifier one or more of the data packets is discarded so that only one data packet remains having the secondary packet sequence identifier.

Figure 6 shows an example of a method 600 for processing a first data packet when received by the second peer 115 as described above with reference to figure 5.

The channel on which the data packet was received is identified 605. The data packet header is examined to identify 610 that the tunnel identifier is that of an agreed upon tunnel.

A security check is performed 615 on the data packet. In an embodiment the security check comprises a comparison of a hash check of the complete packet with a hash derived from parameters in the packet header. Examples of parameters include a protocol version, a packet type, a tunnel id and a sequence id. If the data packet is found to be not intact it is dropped 620.

Examples of checks carried out on the data packet include one or more of:

- verifying that the tunnel identifier corresponds to an existing tunnel;
- verifying that the payload is encrypted if the packet type requires encryption;
- verifying that the sequence identifier is greater than the last verified sequence identifier but less than a specified maximum count greater;
- verifying that the integrity check value (ICV) is correct, the value including as factors at least the data payload, the protocol version, the tunnel identifier, the sequence identifier and the packet type.

If the data packet is intact the payload within the data packet is decrypted 625, using a key and initialisation vector that is partly dependent on the contents of the header of the data packet.

The first peer channel is checked by comparing the received channel with that negotiated and stored in routing table 150. On detecting a change 630 of the first peer channel, a security check is performed 635. If the change in channel satisfies a security check and verified as valid, the return path is updated 640 in the routing table.

On detecting no change in first peer channel, or on updating a change in first peer channel, the data within the data packet is relayed 645 to the second peer.

An example format for a data packet is set out below:

Byte	Name	Data Type	Description
0	protocol version	unsigned 1-byte integer	Protocol version
1	packet type	unsigned 1-byte integer	Some packet types may not carry encrypted payloads
2-4	tunnel id	unsigned 4-byte integer	Locally scoped identifier for the end of the tunnel
5-8	primary packet sequence ID	unsigned 4-byte integer	Sequence assisting in receipt confirmation check and one salt for the encryption

9-12	secondary packet sequence ID	unsigned 4-byte integer	Sequence assisting in critical data receipt and deduplication. Optional.
13	key id	unsigned 1-byte integer	Specifies which session key to use when decrypting the payload and checking the sequence id
14-30	aes-gcm tag	opaque 16-byte value	Used by the authenticated encryption with associated data (AEAD) algorithm to detect tampering
31-N	payload	arbitrary length encrypted payload	The encrypted payload itself, with padding where applicable

The above data packet includes at byte 0 the version of the protocol, at byte 1 the packet type which may typically indicate encapsulated network content, a keep-alive packet, a routing table packet or some other option.

Bytes 2-4 hold the tunnel identifer for this end of the tunnel, the scope being easily large enough to cater for the several thousand tunnels expected of a large local network using a VPN concentrator.

Bytes 5-8 provide a recycling primary packet sequence identifier for each tunnel packet. Bytes 9-12 are used to represent a secondary packet sequence identifier where data packets are duplicated. Byte 13 is the key identifying the session identifier.

Bytes 14-30 provide a data check value derived by calculation from at least the payload content, the protocol version, the tunnel identifier, the sequence identifier and the packet type. If this value does not match that calculated using the same items at the receiving end the packet is deemed corrupt or contrived.

The bytes from 31 onwards contain the payload, which is typically an encrypted network packet, but may be a packet with other content, specified in the packet type field.

The content of differing packet types is treated differently. A data packet contains information to be transferred onwards to a destination client, a keep-alive packet contains information which is extracted by the local peer and either responded to or used to update the local peer state, a routing packet will contain information which is again extracted by the local peer and may be used to update or confirm routing at the remote peer. Other packet types may be provided for.

An example of a data format for a keep-alive packet is set out below:

Byte	Name	Data Type	Description
0-3	keep-alive sequence id	unsigned 4-byte integer	
4-5	timestamp	unsigned 2-byte integer	Contains a partial timestamp of the time the request was made
6	remote-routing table version	unsigned 1-byte integer	The version of the local peer's routing table that the remote peer knows about.

In an embodiment the keep-alive packet is a packet which is sent encrypted as the payload content of a data packet intermittently during normal use and also when the VPN tunnel is not otherwise being used.

Bytes 0-3 represent a keep-alive sequence identifier which is a recycling integer. Bytes 4-5 represent a timestamp indicating when the packet was created.

Byte 6 represents a remote routing table version which is the version of the local routing table which is currently recorded as held by the initiating peer. The routing table version comprises a single byte and hence will wrap around comparatively frequently.

An example data format for routing table packet is shown below:

Field	Data Type	Description
routing-table-version	unsigned 1-byte integer	The version of the sender's routing table
routing-table-entry[ ]	variable length	zero or more routing table entries

The routing table packet is a packet which is sent as the payload content of data packet when, for example, a remote peer detects from reading a keep-alive packet that the routing table for this tunnel is in error.

In an embodiment the routing table packet includes a 1-byte integer representing an actual current version of its routing table and a variable length array representing a plurality of routing table entries in the routing table.

Figure 7 shows an embodiment of a suitable computing device to implement embodiments of one or more components of the network 100 of figure 1. These components include one or more of the first peer 105, the routing table 145, the ranking module 155, the duplication engine 160, the second peer 115, the routing table 150, the duplication engine 165, the routing engine 170.

The computing device of Figure 7 is an example of a suitable computing device. It is not intended to suggest any limitation as to the scope of use or functionality of the operating environment.

Example computing devices include, but are not limited to, personal computers, server computers, hand-held or laptop devices, mobile devices, multiprocessor systems, consumer electronics, mini computers, mainframe computers, and distributed computing environments that include any of the above systems or devices. Examples of mobile devices include mobile phones, tablets, and Personal Digital Assistants (PDAs).

Although not required, embodiments are described in the general context of 'computer readable instructions' being executed by one or more computing devices. In an embodiment, computer readable instructions are distributed via tangible computer readable media.

In an embodiment, computer readable instructions are implemented as program modules. Examples of program modules include functions, objects, Application Programming Interfaces (APIs), and data structures that perform particular tasks or implement particular abstract data types. Typically, the functionality of the computer readable instructions is combined or distributed as desired in various environments.

Shown in figure 7 is a system 700 comprising a computing device 705 configured to implement one or more embodiments described above. In an embodiment, computing device 705 includes at least one processing unit 710 and memory 715. Depending on the exact configuration and type of computing device, memory 715 is volatile (such as RAM, for example), non-volatile (such as ROM, flash memory, etc., for example) or some combination of the two.

A server 720 is shown by a dashed line notionally grouping processing unit 710 and memory 715 together.

In an embodiment, computing device 705 includes additional features and/or functionality.

One example is removable and/or non-removable additional storage including, but not limited to, magnetic storage and optical storage. Such additional storage is illustrated in Figure 7 as storage 725. In an embodiment, computer readable instructions to implement one or more embodiments provided herein are maintained in storage 725. In an embodiment, storage 725 stores other computer readable instructions to implement an operating system and/or an application program. Computer readable instructions are loaded into memory 715 for execution by processing unit 710, for example.

Memory 715 and storage 725 are examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, Digital Versatile Disks (DVDs) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computing device 705. Any such computer storage media may be part of device 705.

In an embodiment, computing device 705 includes at least one communication connection 740 that allows device 705 to communicate with other devices. The at least one communication connection 740 includes one or more of a modem, a Network Interface Card (NIC), an integrated network interface, a radio frequency transmitter/receiver, an infrared port, a USB connection, or other interfaces for connecting computing device 705 to other computing devices.

In an embodiment, the communication connection(s) 740 facilitate a wired connection, a wireless connection, or a combination of wired and wireless connections. Communication connection(s) 740 transmit and/or receive communication media.

Communication media typically embodies computer readable instructions or other data in a "modulated data signal" such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" includes a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal.

In an embodiment, device 705 includes at least one input device 745 such as keyboard, mouse, pen, voice input device, touch input device, infrared cameras, video input devices, and/or any other input device. In an embodiment, device 705 includes at least one output device 750 such as one or more displays, speakers, printers, and/or any other output device.

Input device(s) 745 and output device(s) 750 are connected to device 705 via a wired connection, wireless connection, or any combination thereof. In an embodiment, an input device or an output device from another computing device is/are used as input device(s) 745 or output device(s) 750 for computing device 705.

In an embodiment, components of computing device 705 are connected by various interconnects, such as a bus. Such interconnects include one or more of a Peripheral Component Interconnect (PCI), such as PCI Express, a Universal Serial Bus (USB), firewire (IEEE 13104), and an optical bus structure. In an embodiment, components of

computing device 705 are interconnected by a network. For example, memory 715 in an embodiment comprises multiple physical memory units located in different physical locations interconnected by a network.

It will be appreciated that storage devices used to store computer readable instructions may be distributed across a network. For example, in an embodiment, a computing device 755 accessible via a network 760 stores computer readable instructions to implement one or more embodiments provided herein. Computing device 705 accesses computing device 755 in an embodiment and downloads a part or all of the computer readable instructions for execution. Alternatively, computing device 705 downloads portions of the computer readable instructions, as needed. In an embodiment, some instructions are executed at computing device 705 and some at computing device 755.

In an embodiment a client application 785 enables a user experience and user interface. In an embodiment, the client application 785 is provided as a thin client application configured to run within a web browser. The client application 785 is shown in figure 7 associated to computing device 755. It will be appreciated that application 785 in an embodiment is associated to computing device 705 or another computing device.

The foregoing description of the invention includes preferred forms thereof. Modifications may be made thereto without departing from the scope of the invention.

CLAIMS:

1. A method of transmitting data from a first peer, the method comprising:

on determining that the data satisfies an importance threshold:

transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and

transmitting, from the first peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier; and

on determining that the data does not satisfy an importance threshold:

transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, and a first primary packet sequence identifier;

wherein the presence of the secondary packet sequence identifier within the first data packet indicates that the first data packet is a duplicate; and

wherein the presence of the secondary packet sequence identifier within the second data packet indicates that the second data packet is a duplicate.

2. The method of claim 1 wherein the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

3. The method of claim 1 or claim 2 wherein the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the first routing version identifier maintained in computer memory associated to the first peer.

4. The method of claim 3 further comprising maintaining the first routing version identifier in computer memory within a routing table.

5. The method of any one of the preceding claims wherein the first data packet comprises one of a network content packet, a keep-alive packet, a routing table packet.

6. The method of any one of the preceding claims wherein the second data packet comprises a network content packet.

7. The method of any one of the preceding claims wherein the first data packet comprises a keep-alive packet, the keep-alive packet including the first routing version identifier.

8. A method of transmitting data between a first peer and a second peer comprising:

transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and

transmitting, from the first peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier;

wherein the presence of the secondary packet sequence identifier within the first data packet indicates that the first data packet is a duplicate; and

wherein the presence of the secondary packet sequence identifier within the second data packet indicates that the second data packet is a duplicate.

9. A method of resolving data transmitted between a first peer and a second peer comprising:

receiving, at the second peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier;

receiving, at the second peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier; and

discarding the first data packet or the second data packet on determining that both the first data packet and the second data packet include the secondary packet sequence identifier.

10. The method of claim 9 wherein the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

11. The method of claim 9 or claim 10 wherein the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the first routing version identifier maintained in computer memory associated to the second peer.

12. The method of any one of claims 9 to 11 wherein the first data packet includes a routing version identifier associated to the first data packet, the method further comprising transmitting a routing table packet on determining a mismatch between the first routing version identifier and the routing version identifier associated to the first data packet.

13. The method of claim 12 wherein the routing table packet includes the first routing version identifier.

14. The method of claim 12 or claim 13 further comprising maintaining the first routing version identifier in computer memory within a routing table.

15. The method of any one of claims 9 to 14 wherein the first data packet comprises a keep-alive packet.

16. A data network configured to transmit data from a first peer, the network comprising:

a ranking module associated to the first peer, the ranking module configured to measure the data against an importance threshold;

a duplication engine configured, on determining that the data satisfies an importance threshold, to:

assemble a first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and

assemble a second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier; and

the first peer configured to:

transmit the first data packet from the first peer within the first tunnelling session; and

transmit the second data packet from the first peer within the second tunnelling session;

wherein the presence of the secondary packet sequence identifier within the first data packet indicates that the first data packet is a duplicate; and

wherein the presence of the secondary packet sequence identifier within the second data packet indicates that the second data packet is a duplicate.

17. The network of claim 16 wherein the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

18. The network of claim 16 or claim 17 wherein the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the first routing version identifier maintained in computer memory associated to the first peer.

19. The network of claim 18 further comprising a routing table stored in computer memory within which the first routing version identifier is maintained.

20. A data network configured to resolve data received from a first peer, the network comprising:

a second peer configured to:

receive a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier, and

receive a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier; and

a duplication engine configured to discard the first data packet or the second data packet on determining that both the first data packet and the second data packet include the secondary packet sequence identifier.

21. The data network of claim 20 wherein the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

22. The data network of claim 20 or claim 21 wherein the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the first routing version identifier maintained in computer memory associated to the second peer.

23. The data network of any one of claims 20 to 22 wherein the first data packet includes a routing version identifier associated to the first data packet, the data network further comprising a routing engine configured to transmit a routing table packet on determining a mismatch between the first routing version identifier and the routing version identifier associated to the first data packet.

24. The data network of claim 23 wherein the routing table packet includes the first routing version identifier.

25. The data network of claim 23 or claim 24 further comprising a routing table stored in computer memory within which the first routing version identifier is maintained.

26. A computer readable medium having stored thereon computer-executable instructions that, when executed by a processor, cause the processor to perform a method of transmitting data from a first peer, the method comprising:

on determining that the data satisfies an importance threshold:

transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and

transmitting, from the first peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier; and

on determining that the data does not satisfy an importance threshold:

transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, and a first primary packet sequence identifier;

wherein the presence of the secondary packet sequence identifier within the first data packet indicates that the first data packet is a duplicate; and

wherein the presence of the secondary packet sequence identifier within the second data packet indicates that the second data packet is a duplicate.

27. The computer readable medium of claim 26, wherein the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

28. The computer readable medium of claim 26 or claim 27 wherein the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the method further comprising maintaining the first routing version identifier in computer memory associated to the first peer.

29. The computer readable medium of claim 28, the method further comprising maintaining the first routing version identifier in computer memory within a routing table.

30. The computer readable medium of any one of claims 26 to 29 wherein the first data packet comprises one of a network content packet, a keep-alive packet, a routing table packet.

31. The computer readable medium of any one of claims 26 to 30 wherein the second data packet comprises a network content packet.

32. The computer readable medium of any one of claims 26 to 31 wherein the first data packet comprises a keep-alive packet, the keep-alive packet including the first routing version identifier.

33. A computer readable medium having stored thereon computer-executable instructions that, when executed by a processor, cause the processor to perform a method of transmitting data between a first peer and a second peer, the method comprising:

transmitting, from the first peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier; and

transmitting, from the first peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier

associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier;

wherein the presence of the secondary packet sequence identifier within the first data packet indicates that the first data packet is a duplicate; and

wherein the presence of the secondary packet sequence identifier within the second data packet indicates that the second data packet is a duplicate.

34. A computer readable medium having stored thereon computer-executable instructions that, when executed by a processor, cause the processor to perform a method of resolving data transmitted between a first peer and a second peer, the method comprising:

receiving, at the second peer, a first data packet within a first tunnelling session, the first data packet having a first tunnel identifier associated to the first tunnelling session, a first primary packet sequence identifier, and a secondary packet sequence identifier;

receiving, at the second peer, a second data packet within a second tunnelling session, the second data packet having a second tunnel identifier associated to the second tunnelling session, a second primary packet sequence identifier, and the secondary packet sequence identifier; and

discarding the first data packet or the second data packet on determining that both the first data packet and the second data packet include the secondary packet sequence identifier.

35. The computer readable medium of claim 34 wherein the first primary packet sequence identifier is unique within the first tunnelling session and/or the second primary packet sequence identifier is unique within the second tunnelling session.

36. The computer readable medium of claim 34 or claim 35 wherein the first tunnelling session is associated to a first route, the first route associated to a first routing version identifier, the method further comprising maintaining the first routing version identifier in computer memory associated to the second peer.

37. The computer readable medium of any one of claims 34 to 36 wherein the first data packet includes a routing version identifier associated to the first data packet, the method further comprising transmitting a routing table packet on determining a

mismatch between the first routing version identifier and the routing version identifier associated to the first data packet.

38. The computer readable medium of claim 37 wherein the routing table packet includes the first routing version identifier.

39. The computer readable medium of claim 37 or claim 38, the method further comprising maintaining the first routing version identifier in computer memory within a routing table.

40. The computer readable medium of any one of claims 37 to 39 wherein the first data packet comprises a keep-alive packet.

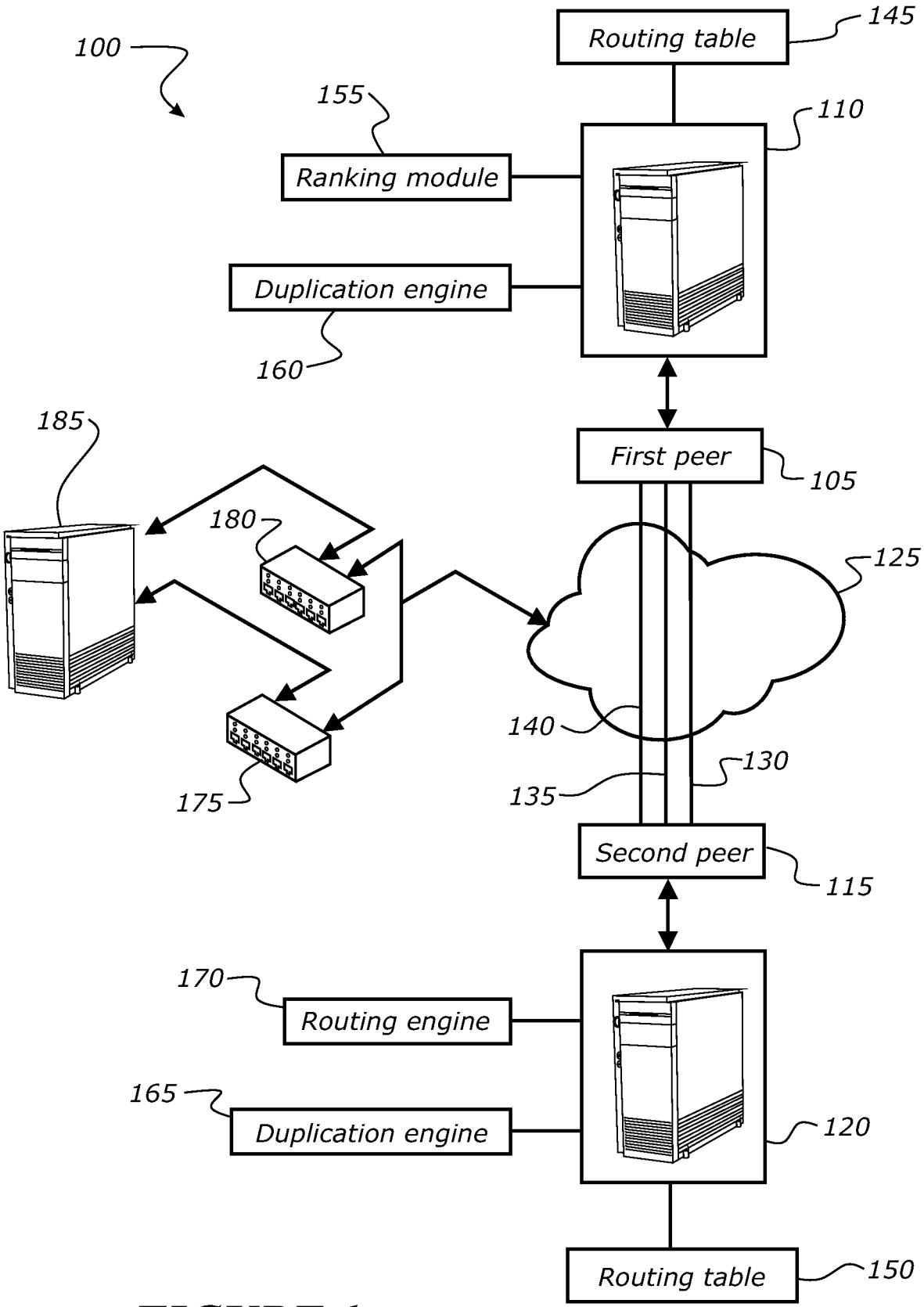
100
Routing table
145
155
110
Ranking module
Duplication engine
160
185
First peer
105
180
125
140
130
135
175
Second peer
115
170
Routing engine
165
Duplication engine
120
Routing table
150

FIGURE 1

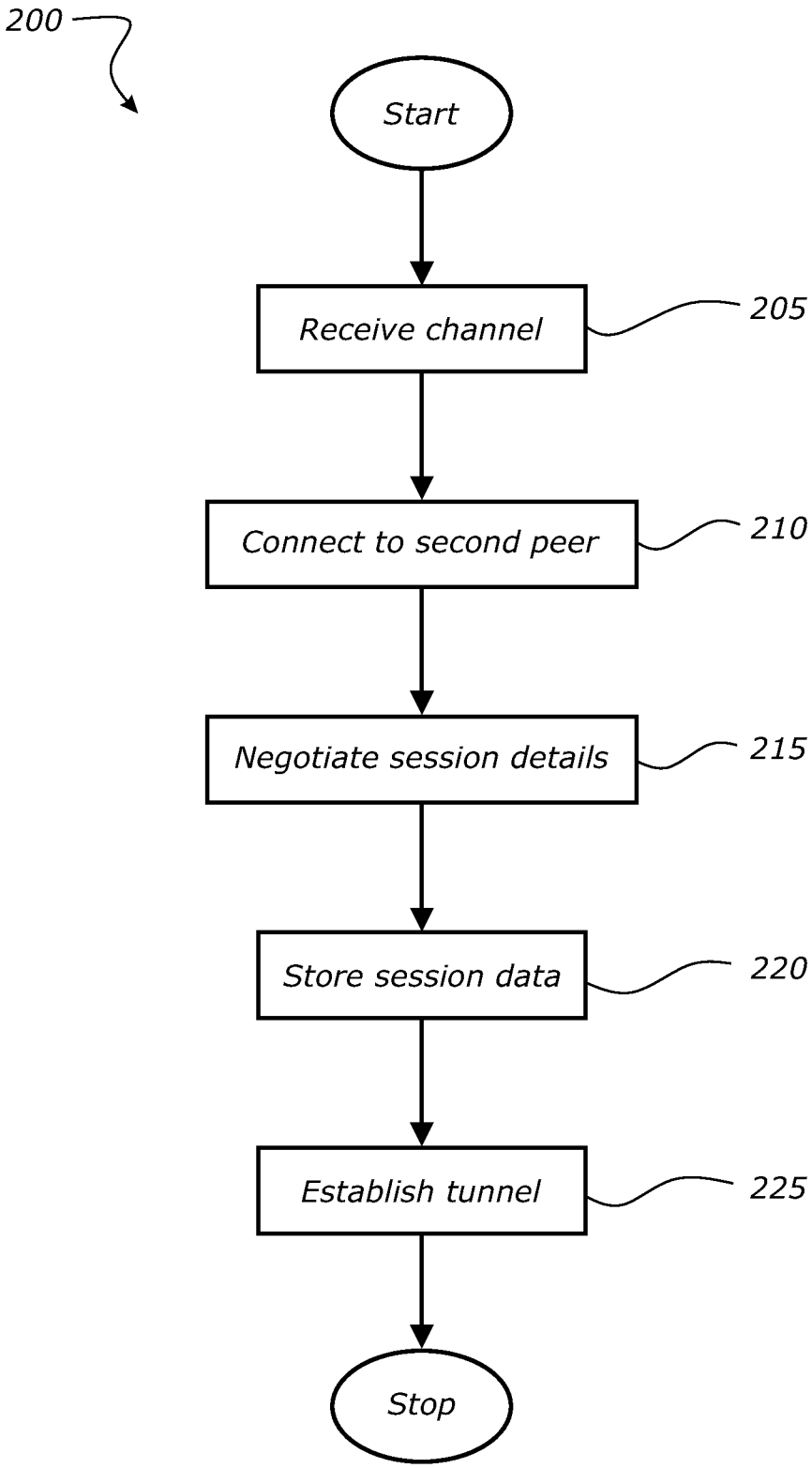
200
Start
Receive channel
205
Connect to second peer
210
Negotiate session details
215
Store session data
220
Establish tunnel
225
Stop

FIGURE 2

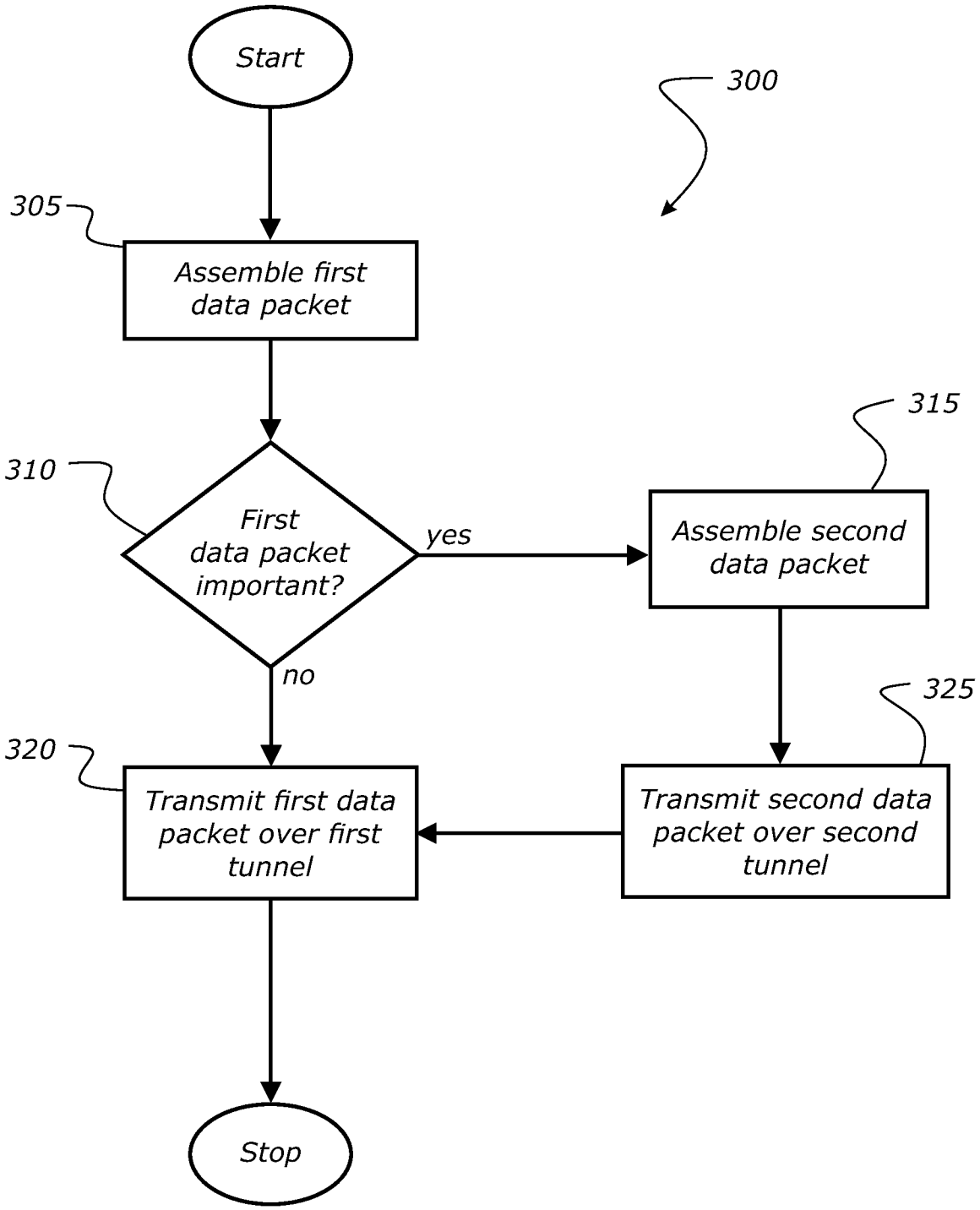
Start
300
305
Assemble first data packet
315
310
First data packet important?
yes
Assemble second data packet
no
325
320
Transmit first data packet over first tunnel
Transmit second data packet over second tunnel
Stop

FIGURE 3

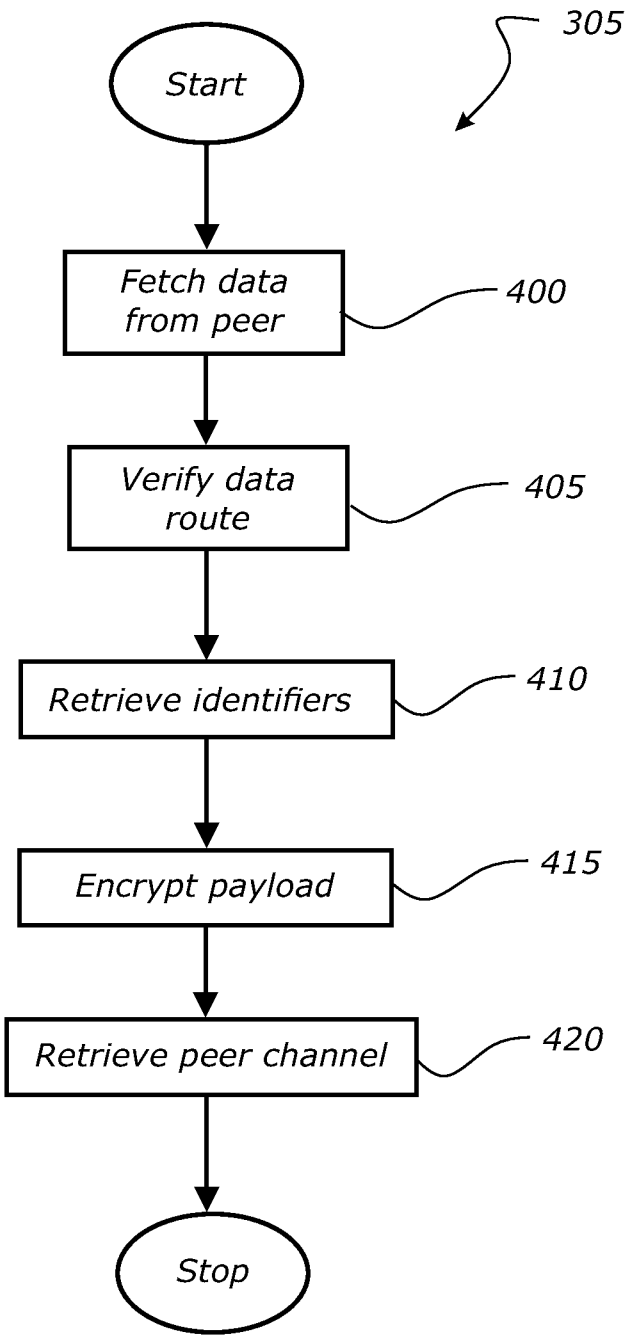
305
Start
Fetch data from peer
400
Verify data route
405
Retrieve identifiers
410
Encrypt payload
415
Retrieve peer channel
420
Stop

FIGURE 4

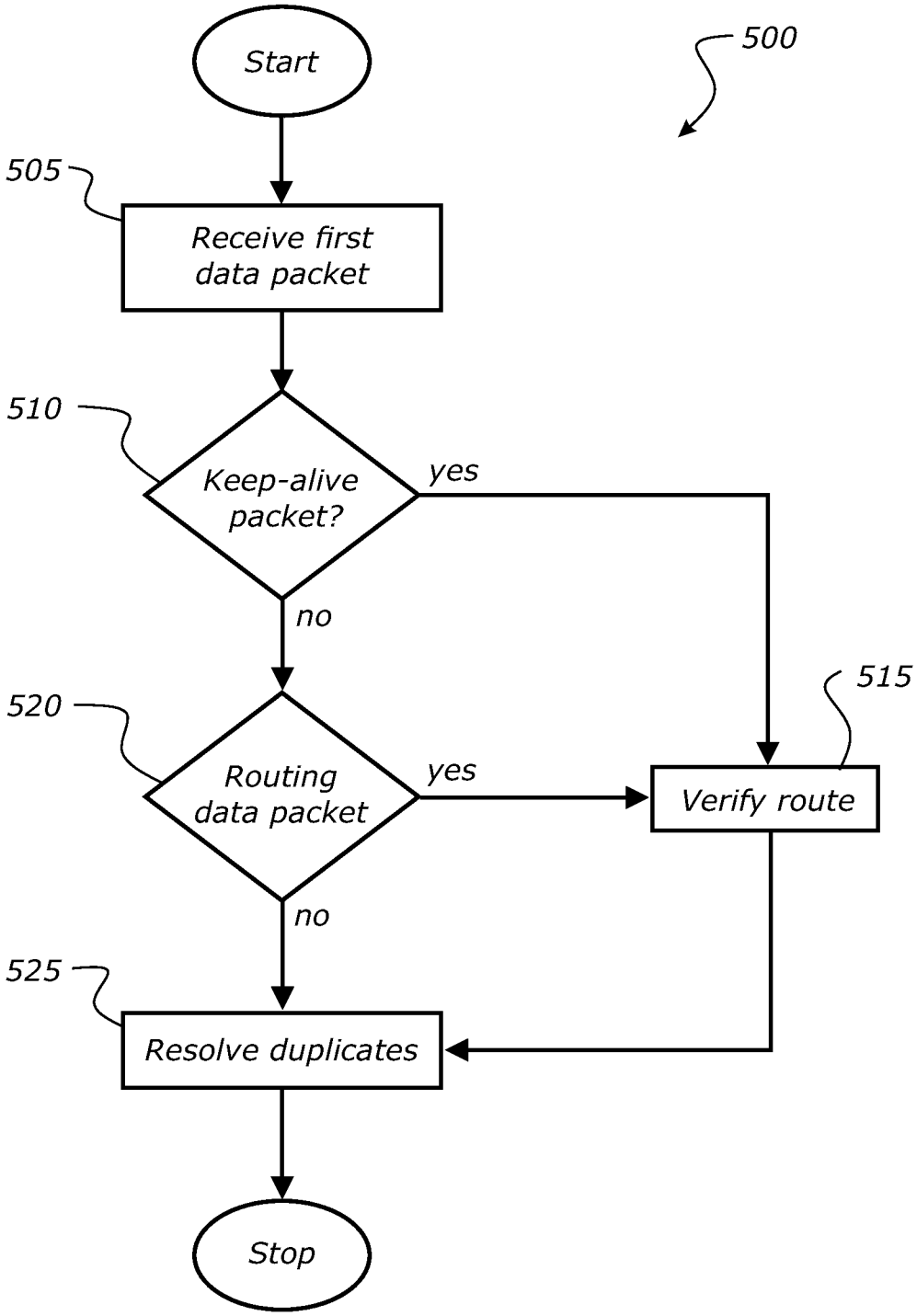
Start
500
505
Receive first data packet
510
Keep-alive packet?
yes
no
515
520
Routing data packet
yes
Verify route
no
525
Resolve duplicates
Stop

FIGURE 5

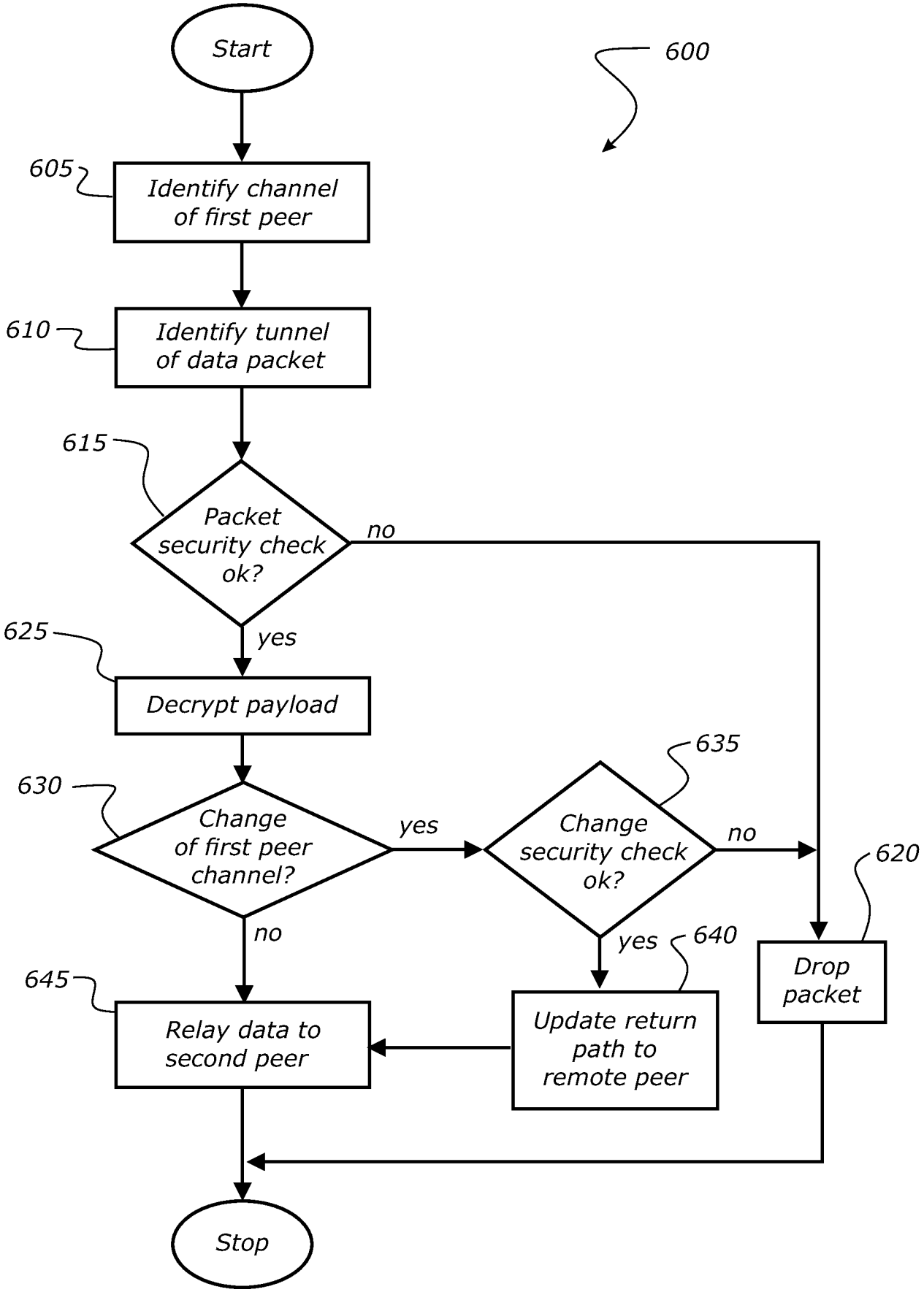
Start
600
605
Identify channel of first peer
610
Identify tunnel of data packet
615
Packet security check ok?
no
yes
625
Decrypt payload
630
Change of first peer channel?
yes
635
Change security check ok?
no
620
Drop packet
no
yes
640
645
Relay data to second peer
Update return path to remote peer
Stop

FIGURE 6

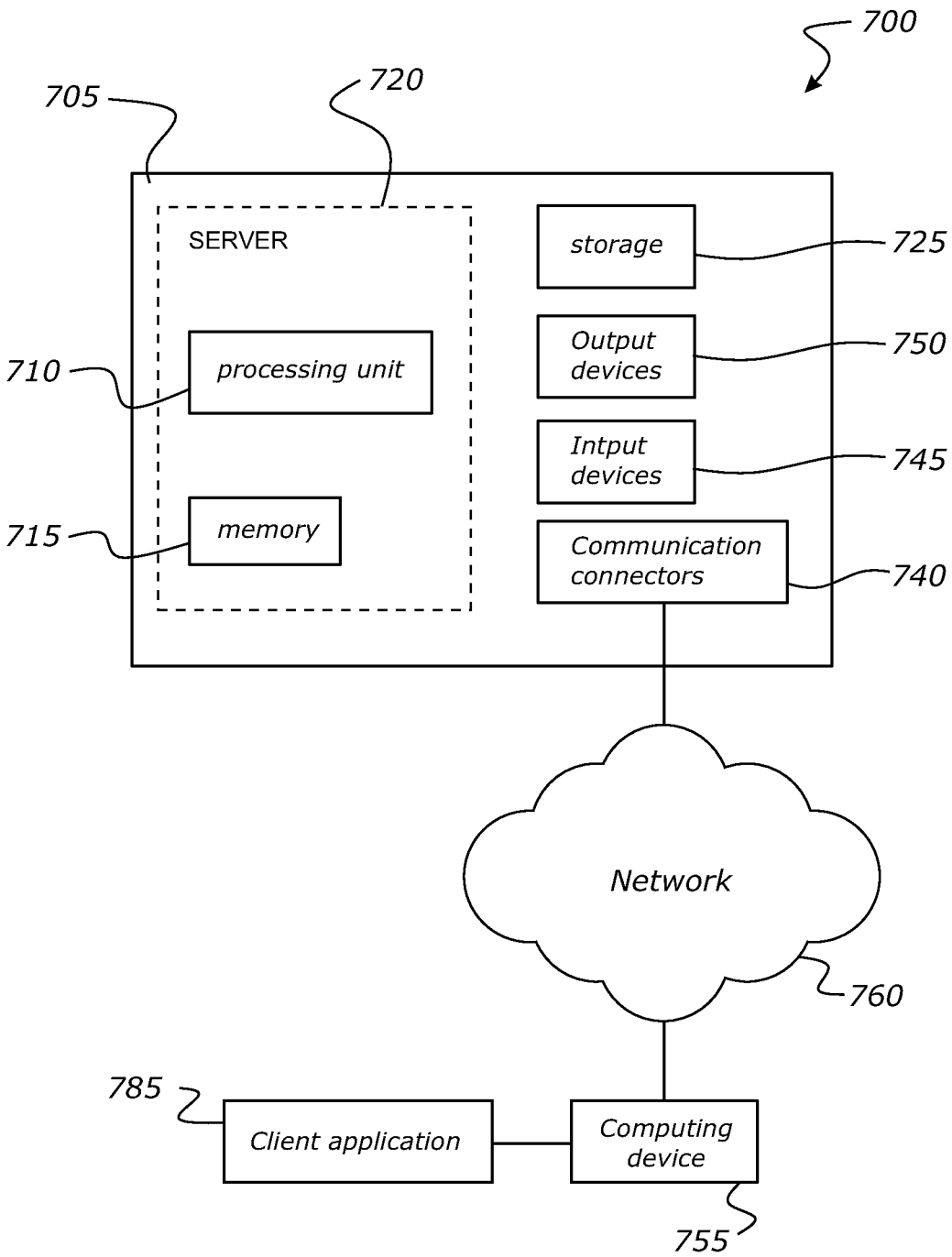
700
705
720
SERVER
storage
725
Output devices
750
processing unit
710
Intput devices
745
memory
715
Communication connectors
740
Network
760
785
Client application
Computing device
755

FIGURE 7

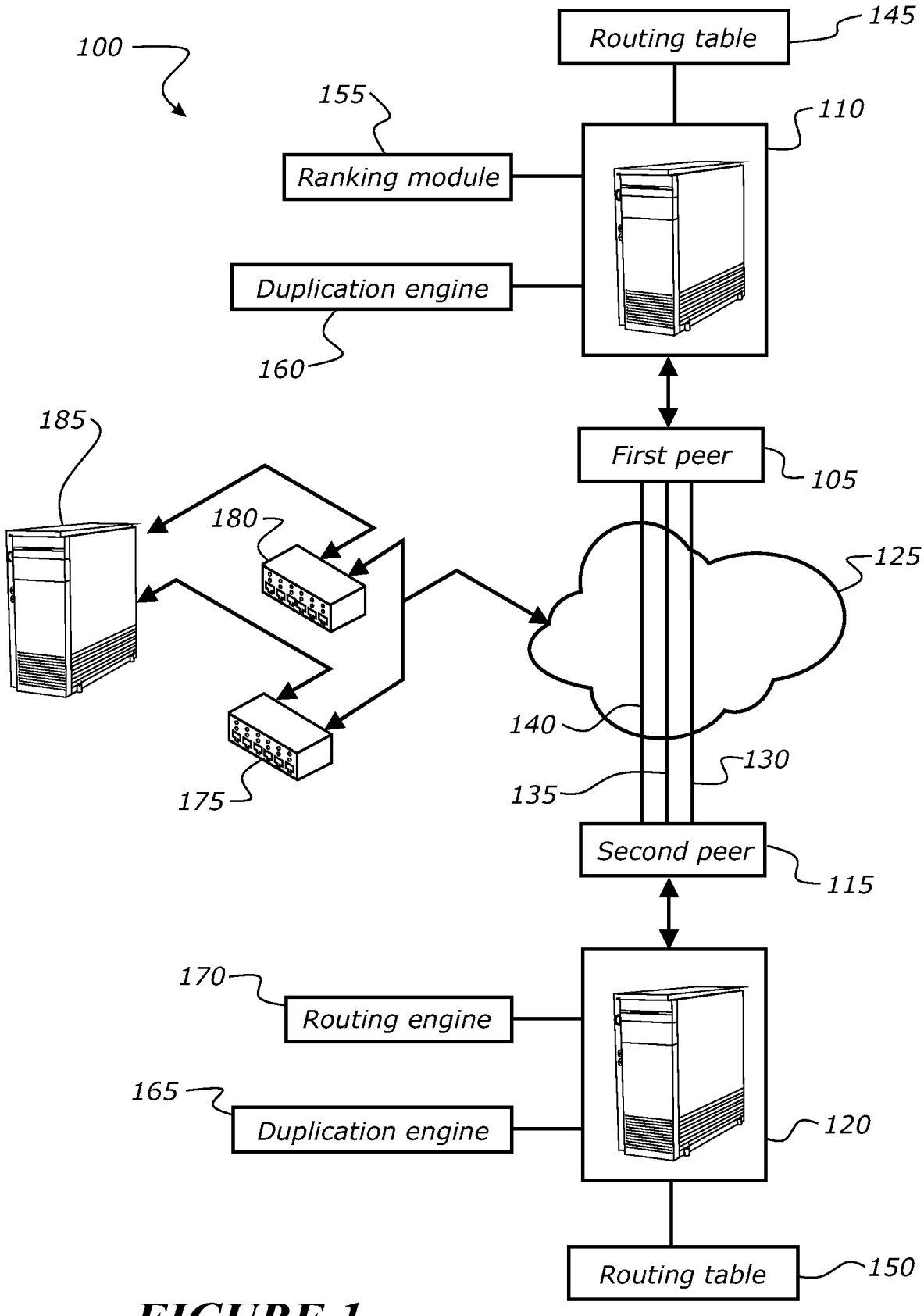
100
Routing table
145
155
110
Ranking module
Duplication engine
160
185
First peer
105
180
125
140
130
135
175
Second peer
115
170
Routing engine
165
Duplication engine
120
Routing table
150

FIGURE 1