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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 February 2018 (22.02.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/034744 A1

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

(21) International Application Number:
PCT/US2017/041306

(22) International Filing Date:
10 July 2017 (10.07.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/241,634 19 August 2016 (19.08.2016) US

(71) Applicant: **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, Redwood Shores, California 94065 (US).

(72) Inventor: **HERRERO, Rolando**; 8 High Street, Derry, New Hampshire 03038 (US).

(74) Agent: **GOLDSMITH, Barry S.**; Miles & Stockbridge P.C., 1751 Pinnacle Drive, Suite 1500, Tysons Corner, Virginia 22102 (US).

(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.

(54) Title: FAST ACCESS TELECOMMUNICATION TUNNEL CLONING

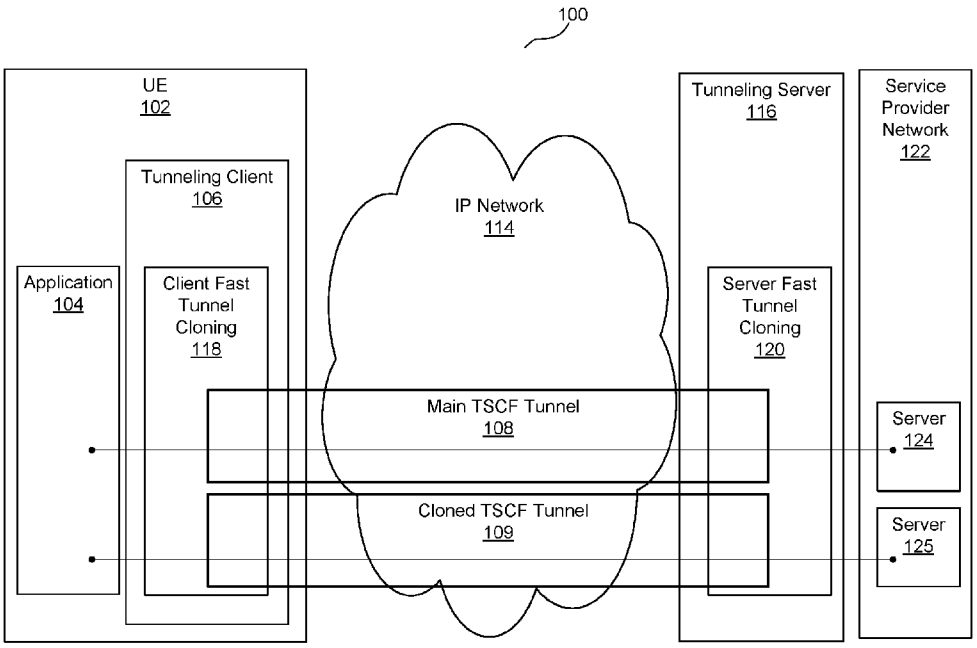


Fig. 1

(57) Abstract: A system establishes a main tunnel between a tunneling client and a tunneling server using a first socket, the main tunnel including a corresponding tunnel identifier and Internet Protocol ("IP") address. The system traverses the encapsulated media over the main tunnel during the telecommunication session and then determines that a cloned tunnel is needed for the telecommunication session. The system establishes a cloned tunnel between the tunneling client and the tunneling server using a second socket that has been marked as a cloned tunnel candidate, where the cloned tunnel includes the corresponding tunnel identifier and IP address of the main tunnel. The system then traverses the encapsulated media over the cloned tunnel instead of the main tunnel during the telecommunication session.

WO 2018/034744 A1

[Continued on next page]

(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))*

FAST ACCESS TELECOMMUNICATION TUNNEL CLONING

FIELD

[0001] One embodiment is directed generally to a communications network, and in particular, to the transmission of encapsulated media within a tunnel over a communications network.

BACKGROUND INFORMATION

[0002] Many enterprise environments have replaced their Public Switched Telephone Network ("PSTN") telephony services with telephony services that use the Internet Protocol ("IP"), commonly known as Voice over IP ("VoIP") or IP Telephony. Since IP Telephony uses an IP network as its backbone, it can provide advanced features such as video conferencing, call recording, and call forwarding.

[0003] Recently, the growing base of mobile data subscribers, the wide availability of Internet access, and the high availability of bandwidth in both fixed and mobile networks has resulted in the popularity of advanced services accessed via the Internet (known as Over-the-Top ("OTT") services). This has caused competitive

service providers to offer OTT services and hence face corresponding challenges as they implement these new services.

SUMMARY

[0004] One embodiment is a system that establishes a main tunnel between a tunneling client and a tunneling server using a first socket, the main tunnel including a corresponding tunnel identifier and Internet Protocol ("IP") address. The system traverses the encapsulated media over the main tunnel during the telecommunication session and then determines that a cloned tunnel is needed for the telecommunication session. The system establishes a cloned tunnel between the tunneling client and the tunneling server using a second socket that has been marked as a cloned tunnel candidate, where the cloned tunnel includes the corresponding tunnel identifier and IP address of the main tunnel. The system then traverses the encapsulated media over the cloned tunnel instead of the main tunnel during the telecommunication session.

BRIEF DESCRIPTION OF THE DRAWINGS

[0001] Fig. 1 is an overview diagram of a network including network elements that implement embodiments of the present invention and/or interact with embodiments of the present invention.

[0002] Fig. 2 is a block diagram of a computer server/system in accordance with an embodiment of the present invention.

[0003] Fig. 3 illustrates example protocol layers in a Tunneled Services Control

Function tunneling configuration for encapsulating media traffic according to an embodiment.

[0004] Fig. 4 includes network elements such as a tunneling client in communication with an application, and a tunneling server in communication with a service provider network, as described herein with reference to Fig. 1.

[0005] Fig. 5 is a flow diagram of a fast access tunnel cloning module of Fig. 2 and/or a tunneling client and a tunneling server of Fig. 1 when creating cloned tunnels for transporting encapsulated media traffic in accordance with embodiments of the present invention.

DETAILED DESCRIPTION

[0006] One embodiment establishes a main telecommunication tunnel for encapsulated media or signaling traffic. During a telecommunication session (e.g., a voice call), upon detection of reduced quality or for any other reason, a second "cloned" tunnel is established, having an identical IP address and as the main tunnel, that may use a different transport layer as the main tunnel and that may connect to a different server as the main tunnel. The traffic is then moved to the second cloned tunnel.

[0007] Fig. 1 is an overview diagram of a network 100 including network elements that implement embodiments of the present invention and/or interact with embodiments of the present invention. Network 100 includes a user equipment ("UE") 102 that performs real-time communications ("RTC") over an Internet Protocol ("IP") network 114 with a service provider network/backbone 122. In RTC, users exchange

information instantly or with insignificant latency. Example applications for RTC include voice and/or video calls, application streaming, softphones, and remote desktop applications. UE 102 may be any device used by an end-user for communications, such as a smartphone, a laptop computer, a tablet, a television, etc.

[0008] In performing RTC, UE 102 communicates signaling and media traffic with respective servers 124, 125 (additional servers may be included, not shown) in service provider network 122. Signaling traffic may be communicated according to an application layer protocol such as the Session Initiation Protocol ("SIP"). SIP is configured to be independent of the underlying transport layer. Accordingly, SIP can run on different transport protocols, such as the Transmission Control Protocol ("TCP" as described in, for example, Internet Engineering Task Force ("IETF") request for comments ("RFC") 793 and RFC 675), the User Datagram Protocol ("UDP" as described in, for example, IETF RFC 768), etc.

[0009] Network 100 further includes a tunneling server 116 that, together with a tunneling client 106 within UE 102, provides functionality for establishing and managing one or more tunnels for performing RTC according to the Tunneled Services Control Function ("TSCF") standard as described in, for example, 3rd generation partnership program ("3GPP") technical report ("TR") 33.830 V0.5.0, the disclosure of which is hereby incorporated by reference in its entirety. In one embodiment, tunneling client 106 and tunneling server 116 establish a main TSCF tunnel 108 that is compliant with TSCF tunnel management (e.g., tunnel initialization, maintenance, termination, etc., as defined by, e.g., 3GPP TR 33.830 V0.5.0), and TSCF tunnel transport protocols are

supported for the negotiation of TSCF tunnel 108 between tunneling client 106 and tunneling server 116, and subsequently establishes a second cloned TSCF tunnel 109, as further described below.

[0010] The TSCF standard provides client side and server side network elements for establishing managed tunnels for performing RTC (e.g., tunneling client 106 and tunneling server 116 in Fig. 1). It also provides two types of outer layer tunneling transports: a stream-based outer layer tunneling transport via TCP or Transport Layer Security ("TLS"), and a datagram-based outer layer tunneling transport via UDP or Datagram Transport Layer Security ("DTLS").

[0011] TLS is a cryptographic protocol as provided in, for example, IETF RFC 2246, RFC 4346, RFC 5246, and/or RFC 6176. DTLS is a protocol that provides communications privacy for datagram protocols. TCP and TLS provide reliable, ordered and error-checked delivery of the inner layer traffic, but introduce undesirable latency that is detrimental to RTC applications over a communication network that experiences impairments. On the other hand, UDP and DTLS do not guarantee reliable delivery, thus minimizing latency and being desirable for RTC.

[0012] In some embodiments, IP network 114 may be a restrictive network in that it may include security devices (e.g., firewalls, proxies, etc.) that allow traffic of only a certain transport protocol (e.g., only TCP, only UDP, etc.). Accordingly, tunneling client 106 and tunneling server 116 may establish and manage TSCF tunnels 108, 109 such that UE 102 may use them to traverse such security devices and connect to tunneling server 116 to reach servers 124, 125 in service provider network 122.

[0013] The TSCF standard further provides control messages for exchanging configuration information between tunneling client 106 and tunneling server 116. According to the TSCF standard, control messages are of a "request/response" type, and a control message response for a request includes either a corresponding reply or an error code indicating why the request cannot be honored by the receiving end. TSCF control messages use a Type Length Value ("TLV") encoding. TLV is a variable length concatenation of a unique type and a corresponding value.

[0014] Each TSCF control message includes a control message ("CM") header at the beginning, including a "CM_Version" field identifying the version of the header and indicating the outer transport protocol of a TSCF tunnel, a "CM_Indication" field identifying whether the message is a control message or not, a "Reserved" field reserved for future use, a "CM_Type" field identifying the type of the control message (e.g., whether it is a request or a response, the corresponding functionality, etc.), a "TLV_Count" field indicating the number of TLVs that follow or are appended to the header in the corresponding control message, a "Tunnel Session ID" ("TSID") field including a tunnel session identifier ("ID") or ("TID") assigned by tunneling server 116 to uniquely identify TSCF tunnel 108 (and subsequent cloned tunnels), and a "Sequence" field that is incremented per message, as described in, for example, 3GPP TR 33.830 V0.5.0.

[0015] In one embodiment, in order to establish main TSCF tunnel 108, tunneling client 106 sends a "configuration request" message to tunneling server 116 to obtain configuration information for TSCF tunnel 108. In a "configuration request" message,

the TSID header field bits are set to 1 (i.e., FFFF...). In response, tunneling server 116 assigns a TSID to a TSCF tunnel and sends a "configuration response" message back to tunneling client 106. The "configuration response" message includes the TSID assigned by tunneling server 116 to TSCF tunnel 108. The subsequent messages between tunneling client 106 and tunneling server 116 include this assigned TSID in their headers.

[0016] In one embodiment, if a control message is communicated between tunneling client 106 and tunneling server 116 and does not include the expected TSID, the control message is dropped and the corresponding TSCF tunnel is terminated. Alternatively, in one embodiment, tunneling client 106 may send a "configuration release request" message to tunneling server 116 to terminate a TSCF tunnel. In response to such a "configuration release request" message, tunneling server 116 sends a "configuration release response" message to tunneling client 106. At this time, TSCF tunnel 108 is terminated.

[0017] In one embodiment, UE 102 executes an application 104 that may be a SIP based RTC application relying on a library such as the software development kit ("SDK") provided by the Tunneled Session Management ("TSM") solution from Oracle Corp. The TSM solution employs a client/server architecture using session border controllers ("SBCs") and client applications, such as application 104, that may be developed using the SDK. The client applications initiate secure communications sessions with the service provider over the Internet. The session border controllers (e.g., implemented by tunneling server 116) at the edge of the network terminate and

control the tunnels before passing the secure traffic into the service core of service provider network 122. In one embodiment, SDKs are implemented by a client fast tunnel cloning module 118 and/or a server fast tunnel cloning module 120.

[0018] The SDKs in general provide additional APIs beyond "standard" TSCF APIs in order to implement the functionality disclosed herein. One embodiment provides TSCF SDKs that support an application programming interface ("API") so that application 104 can enable the fast tunnel cloning functionality. The TSCF SDK provides a Berkeley software distribution ("BSD")-like socket API that can be used to send and receive encapsulated media using the tsc_sendto and tsc_recvfrom functions, respectively.

[0019] Fig. 2 is a block diagram of a computer server/system (i.e., system 10) in accordance with an embodiment of the present invention. System 10 can be used to implement any of the network elements shown in Fig. 1 as necessary in order to implement any of the functionality of embodiments of the invention disclosed in detail below. Although shown as a single system, the functionality of system 10 can be implemented as a distributed system. Further, the functionality disclosed herein can be implemented on separate servers or devices that may be coupled together over a network. Further, one or more components of system 10 may not be included. For example, for the functionality of tunneling server 116 of Fig. 1, system 10 may be a server that in general has no need for a display 24 or one or more other components shown in Fig. 2.

[0020] System 10 includes a bus 12 or other communication mechanism for

communicating information, and a processor 22 coupled to bus 12 for processing information. Processor 22 may be any type of general or specific purpose processor. System 10 further includes a memory 14 for storing information and instructions to be executed by processor 22. Memory 14 can be comprised of any combination of random access memory ("RAM"), read only memory ("ROM"), static storage such as a magnetic or optical disk, or any other type of computer readable medium. System 10 further includes a communication device 20, such as a network interface card, to provide access to a network. Therefore, a user may interface with system 10 directly, or remotely through a network, or any other method.

[0021] Computer readable medium may be any available media that can be accessed by processor 22 and includes both volatile and nonvolatile media, removable and non-removable media, and communication media. Communication media may include computer readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism, and includes any information delivery media.

[0022] Processor 22 may further be coupled via bus 12 to a display 24, such as a Liquid Crystal Display ("LCD"). A keyboard 26 and a cursor control device 28, such as a computer mouse, may further be coupled to bus 12 to enable a user to interface with system 10 on an as needed basis.

[0023] In one embodiment, memory 14 stores software modules that provide functionality when executed by processor 22. The modules include an operating system 15 that provides operating system functionality for system 10. The modules

further include a fast access tunnel cloning module 16 for creating cloned tunnels for transporting encapsulated media traffic, and all other functionality disclosed herein. In one example embodiment, fast access tunnel cloning module 16 may implement tunneling server 116 of Fig. 1 or tunneling client 106 in conjunction with one or more remaining elements of Fig. 2. System 10 can be part of a larger system, such as added functionality to the "Acme Packet 6300" session border controller from Oracle Corp. Therefore, system 10 can include one or more additional functional modules 18 to include the additional functionality. A database 17 is coupled to bus 12 to provide centralized storage for fast access tunnel cloning module 16 and additional functional modules 18.

[0024] In a tunneling configuration, encapsulated (i.e., in a packet/frame) media is typically communicated according to the real-time transport protocol ("RTP" as provided, for example, in IETF RFC 3550). In a TSCF tunneling configuration, RTC (e.g., speech, video, etc.) may be subject to two levels of transport: one at the outer tunnel layer typically according to TCP/TLS, and another at the inner tunnel layer typically according to UDP. Fig. 3 illustrates example protocol layers in a TSCF tunneling configuration 300 for encapsulating media traffic according to an embodiment. In TSCF tunneling configuration 300, compressed media (e.g., speech, video, etc.) is communicated according to RTP at the application layer, and is transported via an inner UDP at the inner transport layer within an inner IP at the inner network layer. The inner layers are within an outer TCP/TLS at the outer transport layer which is in turn within an outer IP at the outer network layer. In one embodiment, since most IP networks block

any outer traffic that is not stream-based, TCP/TLS is used at the outer transport layer of TSCF tunnel 108 to guarantee delivery.

[0025] To support low latency and reliable delivery of traffic, in certain situations it is desired to dynamically switch the transport type and/or the path of data associated with a given inner socket. For example, it is sometimes necessary for inner socket TCP based signaling traffic to follow a different path than inner socket TCP based instant messaging traffic. Known solutions provided for TSCF, such as Dynamic Datagram Tunnels ("DDT"), fail to address this problem for two reasons: (1) they are intended for media only; and (2) they rely on datagram transport only.

[0026] Referring again to Fig. 1, in order to provide mechanisms of low latency and reliable firewall traversal, as well as forward error correction ("FEC"), it may be necessary to establish or "clone" one or more additional tunnels (e.g., cloned tunnel 109) that, relying on different transport layer and path diversity, carry the same internal IP address as that of the main tunnel (e.g., main tunnel 108). To minimize packet loss and latency the cloned tunnels need to be established as fast as possible, and simultaneously be able to route traffic in an efficient way according to the needs of application 104. Therefore, embodiments include a transparent mechanism for dynamic main tunnel cloning based on transport and routing requirements.

[0027] In one embodiment, in order to implement, a new socket option is added to the "tsc_setsockopt" API to allow application 104 to enable a new transport by means of one or more cloned tunnel 109. The cloned tunnels in one embodiment have an internal IP address that is identical to that of main tunnel 108 and created upon socket

binding.

[0028] In one embodiment, a new TSCF CM configuration request parameter is established to enable cloned tunnel creation. If a cloned tunnel cannot be established or until it is established, all traffic that belongs to its associated socket will traverse main tunnel 108. Further, server 116 and client 106 route traffic that belongs to specific sockets through their associated cloned tunnels.

[0029] In one embodiment, cloned tunnel 109 is terminated by the infrastructure as soon as their associated sockets are removed by application 104. Cloned tunnels 109 are terminated if main tunnel 108 is removed.

[0030] Fig. 4 is an example message sequence diagram 400, including the sequence of messages exchanged between application 104, tunneling client 106 and tunneling server 116 of Fig. 1, for fast access tunnel cloning according to some embodiments. Fig. 4 includes network elements such as tunneling client 106 in communication with application 104, and tunneling server 116 in communication with service provider network 122, as described herein with reference to Fig. 1.

[0031] In response to a tunnel creation request 401 from application 104, TSCF tunnel 108 (i.e., "main" tunnel) is established by executing a "tsc_ctrl_new_tunnel" API. Internally, tunneling client 106 generates a tunnel configuration request CM at 402 that is responded to by tunneling server 116 with a tunnel configuration response CM at 403, having a corresponding TID and IP address.

[0032] Application 104 then creates and binds socket #1 (i.e., a first socket) by executing a tsc_socket API at 420 and a tsc_bind API at 421.

[0033] Application 104 then calls a tsc_sendto API at 422 and 423 to send two frames (1 and 2) at 404, 405 using socket #1 over main tunnel 108.

[0034] At 424, application 104 creates socket #2 (i.e., a second socket) by executing a tsc_socket API and then marks socket #2 as a candidate for transport in a cloned tunnel by executing a tsc_setsockopt API at 425. Application 104 may create socket #2, and determine that a cloned tunnel is needed, based on many possible factors, including determining that the quality of an existing telecommunication session (e.g., a telephone call) using main tunnel 108 has sufficiently degraded. For example, using a measurement of packet loss, jitter, etc., the quality can be determined.

[0035] In one embodiment, degradation of the telecommunication session is measured through Medium Opinion Score ("MOS") values obtained by the TSCF infrastructure using probe packets that are looped back at the tunnel server. The mechanism measures latency, jitter and packet loss to calculate a quality score that is then reported to application 104. Application 104 can then use a cloned tunnel to improve communications.

[0036] At 426, application 104 binds socket #2 by calling a tsc_bind API. Internally, tunneling client 106 establishes a cloned tunnel 109 by generating a tunnel configuration request CM at 407 including a cloning request that is responded to by tunneling server 116 with a tunnel configuration response CM at 410. The established clone tunnel has the same corresponding TID and internal IP address as the main tunnel established at 402, 403. In one embodiment, the cloned tunnel is established without the use of TSCF service messages. TSCF service messages are a type of

control message ("CM") that are used to enable services once a tunnel has been established.

[0037] Application 104 then calls a tsc_sendto API at 427 and 428 to send two frames (3 and 4) at 411, 413 using socket #2 over cloned tunnel 109. The frames can go to the same server in service provider network 122 (e.g., server 124) as the frames in the main tunnel, or can go to a different server as the frame in the main tunnel (e.g., main tunnel 108 frames go to server 124, and clone tunnel 109 frames go to server 125).

[0038] At 429, application 104 closes socket #2 by executing a tsc_close API at 429. Internally, tunneling client 106 generates a tunnel configuration release request CM at 415 that is responded to by tunneling server 116 with a tunnel configuration release response CM at 417.

[0039] As a result of the functionality of Fig. 4, embodiments allow for the creation of an unlimited number of cloned tunnels that rely on any transport type, such as a stream or datagram, and can access any preconfigured server that is part of service provider network 122, or located elsewhere. Therefore, embodiments allow for dynamically changing paths while maintaining inner layer parameters such as the internal IP address of main tunnel 108.

[0040] One embodiment supports fast access tunnel cloning by expanding TSCF to include a "clone" TLV value that is used to that is used indicate that the new tunnel is a clone of the main tunnel indicated in the TID of the configuration request CM (i.e., at 407 of Fig. 4).

[0041] Table 1 below provides an example TSCF TLV for providing fast access tunnel cloning functionality according to some embodiments.

TLV TYPE		SEMANTICS	SHORT/ LONG FORMAT	VALUE TYPE	LENGTH
NAME	VALUE				
Clone	54	0: original tunnel 1: cloned tunnel	Short	8-bit number	1 byte

Table 1

[0042] In one embodiment, if a given socket is to run traffic over a cloned tunnel, client application 104 sets the right option using the tsc_setsockopt API (i.e., at 425 of Fig. 4) as shown in the following pseudo-code.

```
int transport = TSC_NEW_TRANSPORT_UDP;
int result = tsc_setsockopt(socket, SOL_SOCKET, SO_TSC_NEW_TRANSPORT,
                (char *)&transport, sizeof(int));
```

where the "transport" variable above indicates the desired transport type of cloned tunnel 109 according to the following possible definitions in one embodiment, where the transport types are either UDP or TCP:

```
#define TSC_NEW_TRANSPORT_UDP 0
#define TSC_NEW_TRANSPORT_TCP 1
```

[0043] If "tsc_setsockopt" returns -1 the option was not set correctly. If it returns 0 it was set correctly but the functionality is not active until the cloned tunnel is negotiated. A new notification "tsc_notification_new_transport" can be used to notify the

client about this activation. The following pseudo-code shows how the notification is enabled and what the notification callback function looks like:

```
tsc_notification_enable(handle, tsc_notification_new_transport, new_transport_notification, NULL);

void new_transport_notification(tsc_notification_data *notification)
{
   tsc_notification_new_transport_info_data *new_transport_data = (tsc_notification_new_transport
(info_data *)notification->data;

   if (new_transport_data && new_transport_data->available == tsc_bool_true) {
      if (new_transport_data->enabled == tsc_bool_true) {
         printf("new transport enabled notification on socket %d\n", new_transport_data->socket);
      } else {
         printf("new transport notification playing on socket %d\n", new_transport_data->socket);
      }
   } else {
      printf("new transport notification not allowed on socket %d\n", new_transport_data->socket);
   }
}
```

The fourth NULL parameter in “tsc_notification_enable” is an opaque/private data pointer that can be recovered in the “tsc_notification_data” structure upon callback.

[0044] One embodiment that is implemented using an SBC, such as the Acme Packet 6300 from Oracle Corp., provides a configuration object “tscf-interface” that

includes a parameter “assigned-services” that includes a keyword “cloned_tunnels”. Table 2 below provides an example of the tscf-interface configuration object according to one embodiment.

Parameter Name	Extensible markup language (“XML”) tag	Data Type and Default	Value Ranges/Possible Values	Required or Optional (for feature to work)?
assigned-services	assigned-services	String: Blank	***cloned_tunnels*** to enable cloned tunnels	Optional

Table 2

[0045] The following functionality provides an example interface configuration for providing fast access tunnel cloning according to one embodiment.

```
tscf-interface
    realm-id              access
    state                 enabled
    max-tunnels           1000
    local-address-pools   lp
    assigned-services     SIP, cloned_tunnels
    tscf-port
        address                 192.168.1.100
        port                    4888
        transport-protocol      TCP
        tls-profile
        rekey-profile
```

[0046] The following is an example extensible markup language (“XML”) functionality for providing fast access tunnel cloning according to one embodiment:

```
<tscfInterface realmID='access'
    state='enabled'
    maxTunnels='1000'
```

```
assignedServices='SIP,cloned_tunnels
options=''
lastModifiedBy='admin@console'
lastModifiedDate='2015-01-05 05:01:10'
objectId='33'>
<key>access</key>
<localAddressPool name='lp'/>
<tscfPort address='192.168.1.100
    port='4888'
    transProtocol='TCP'
    tlsProfile=''
    rekeyProfile=''
</tscfInterface>
```

[0047] Fig. 5 is a flow diagram of fast access tunnel cloning module 16 of Fig. 2 and/or tunneling client 106 and tunneling server 116 of Fig. 1 when creating cloned tunnels for transporting encapsulated media traffic in accordance with embodiments of the present invention. In one embodiment, the functionality of the flow diagram of Fig. 5 is implemented by software stored in memory or other computer readable or tangible medium, and executed by a processor. In other embodiments, the functionality may be performed by hardware (e.g., through the use of an application specific integrated circuit ("ASIC"), a programmable gate array ("PGA"), a field programmable gate array ("FPGA"), etc.), or any combination of hardware and software.

[0048] At 502, for a telecommunication session, a main tunnel is established between tunneling client 106 and tunneling server 116 using a first socket. In one embodiment, the main tunnel is a TSCF based tunnel that has a corresponding tunnel

ID and IP address.

[0049] At 504, encapsulated media traverses the main tunnel. In another embodiment, other types of data such as signaling traffic may traverse the main tunnel.

[0050] At 506, application 104 determines that a cloned tunnel is needed. For example, a measurement of a quality of a session involving the encapsulated media of 504 may move below a predetermined threshold (e.g., a voice call has degraded, and as a result the cloned tunnel is established in the middle of the voice call). Tunneling server 116 or tunneling client 106 can also determine if a cloned tunnel is needed by receiving a determination from application 104.

[0051] At 508,using a second socket that has been marked as a cloned tunnel candidate, a cloned tunnel is established between tunneling client 106 and tunneling server 116. In one embodiment, the cloned tunnel is a TSCF based tunnel that has the same corresponding tunnel ID and IP address as the main tunnel of 502. The cloned tunnel may have a different transport layer as the main tunnel (e.g., UDP instead of TCP), and the data in the frames may go to a different server of the service provider network as the main tunnel. For example, each tunnel (e.g., main tunnel 108 and cloned tunnel 109) may terminate at two different interfaces (e.g., tunnel servers 124, 125) that propagate the information down the network to the untunneled side. Since each tunnel server is associated to a different interface they are also associated to different networks with different impairments.

[0052] In one embodiment, the cloned tunnel is established without the use of TSCF service messages. In one embodiment, a second, different server is first

determined and then the corresponding second socket is selected. In one embodiment the cloned tunnel has the same inner layer of the main tunnel and a different outer layer. For example, two application sockets #1 and #2 that run UDP traffic can be transported on two different tunnels. Traffic of socket #1 may be in a TCP transport tunnel and traffic of socket #2 may be in a UDP transport tunnel. For example, in one embodiment, socket #1 has the following four layers: outer IP + outer TCP + inner IP + inner UDP, and socket #2 has the following four layers: outer IP + outer UDP + inner IP + inner UDP.

[0053] At 510, additional encapsulated media traverses the cloned tunnel instead of the main tunnel for the telecommunication session.

[0054] As disclosed, for a telecommunication session, a main tunnel is established using a first socket. As some point during the session due to, for example, degraded quality, a cloned tunnel is established using a second socket that is marked for a cloned tunnel. The cloned tunnel has the same tunnel ID and IP address as the main tunnel. The cloned tunnel is then used for transporting encapsulated media instead of the main tunnel.

[0055] Several embodiments are specifically illustrated and/or described herein. However, it will be appreciated that modifications and variations of the disclosed embodiments are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

WHAT IS CLAIMED IS:

1. A method of transmitting encapsulated media during a telecommunication session, the method comprising:

establishing a main tunnel between a tunneling client and a tunneling server using a first socket, the main tunnel comprising a corresponding tunnel identifier and Internet Protocol (IP) address;

traversing the encapsulated media over the main tunnel during the telecommunication session;

determining that a cloned tunnel is needed for the telecommunication session;

establishing a cloned tunnel between the tunneling client and the tunneling server using a second socket that has been marked as a cloned tunnel candidate, wherein the cloned tunnel comprises the corresponding tunnel identifier and IP address of the main tunnel; and

traversing the encapsulated media over the cloned tunnel instead of the main tunnel during the telecommunication session.

2. The method of claim 1, wherein the main tunnel and the cloned tunnel are established according to a tunneled services control function (TSCF) standard.

3. The method of claim 1, wherein the main tunnel terminates at a first interface associated with a first server and the cloned tunnel terminates at a second interface associated with a second server that is different than the first server.

4. The method of claim 2, wherein the cloned tunnel is established without the use of TSCF service messages.

5. The method of claim 1, wherein the main tunnel and the cloned tunnel comprise one of either a Transmission Control Protocol (TCP) transport or a User Datagram Protocol (UDP) transport.

6. The method of claim 1, wherein the main tunnel and the cloned tunnel comprise a same inner layer and a different outer layer.

7. The method of claim 1, further comprising traversing signaling traffic over the cloned tunnel instead of the main tunnel.

8. The method of claim 1, wherein the determining that the cloned tunnel is needed for the telecommunication session comprises determining a level of quality degradation of the telecommunication session.

9. A computer readable medium having instructions stored thereon that, when executed by a processor, cause the processor to transmit encapsulated media during a telecommunication session, the transmitting comprising:

establishing a main tunnel between a tunneling client and a tunneling server using a first socket, the main tunnel comprising a corresponding tunnel identifier and

Internet Protocol (IP) address;

traversing the encapsulated media over the main tunnel during the telecommunication session;

determining that a cloned tunnel is needed for the telecommunication session;

establishing a cloned tunnel between the tunneling client and the tunneling server using a second socket that has been marked as a cloned tunnel candidate, wherein the cloned tunnel comprises the corresponding tunnel identifier and IP address of the main tunnel; and

traversing the encapsulated media over the cloned tunnel instead of the main tunnel during the telecommunication session.

10. The computer readable medium of claim 9, wherein the main tunnel and the cloned tunnel are established according to a tunneled services control function (TSCF) standard.

11. The computer readable medium of claim 9, wherein the main tunnel terminates at a first interface associated with a first server and the cloned tunnel terminates at a second interface associated with a second server that is different than the first server.

12. The computer readable medium of claim 10, wherein the cloned tunnel is established without the use of TSCF service messages.

13. The computer readable medium of claim 9, wherein the main tunnel and the cloned tunnel comprise one of either a Transmission Control Protocol (TCP) transport or a User Datagram Protocol (UDP) transport.

14. The computer readable medium of claim 9, wherein the main tunnel and the cloned tunnel comprise a same inner layer and a different outer layer.

15. The computer readable medium of claim 9, the transmitting further comprising traversing signaling traffic over the cloned tunnel instead of the main tunnel.

16. The computer readable medium of claim 9, wherein the determining that the cloned tunnel is needed for the telecommunication session comprises determining a level of quality degradation of the telecommunication session.

17. A user equipment device comprising:

an application; and

a tunneling client;

the tunneling client configured to establish a main tunnel between a tunneling client and a tunneling server using a first socket, the main tunnel comprising a corresponding tunnel identifier and Internet Protocol (IP) address;

the tunneling client configured to traverse the encapsulated media over the main

tunnel during the telecommunication session;

the application configured to determining that a cloned tunnel is needed for the telecommunication session;

the tunneling client configured to establish a cloned tunnel between the tunneling client and the tunneling server using a second socket that has been marked as a cloned tunnel candidate, wherein the cloned tunnel comprises the corresponding tunnel identifier and IP address of the main tunnel; and

the tunneling client configured to traverse the encapsulated media over the cloned tunnel instead of the main tunnel during the telecommunication session.

18. The user equipment device of claim 17, wherein the main tunnel and the cloned tunnel are established according to a tunneled services control function (TSCF) standard.

19. The user equipment device of claim 17, wherein the main tunnel terminates at a first interface associated with a first server and the cloned tunnel terminates at a second interface associated with a second server that is different than the first server.

20. The user equipment device of claim 18, wherein the cloned tunnel is established without the use of TSCF service messages.

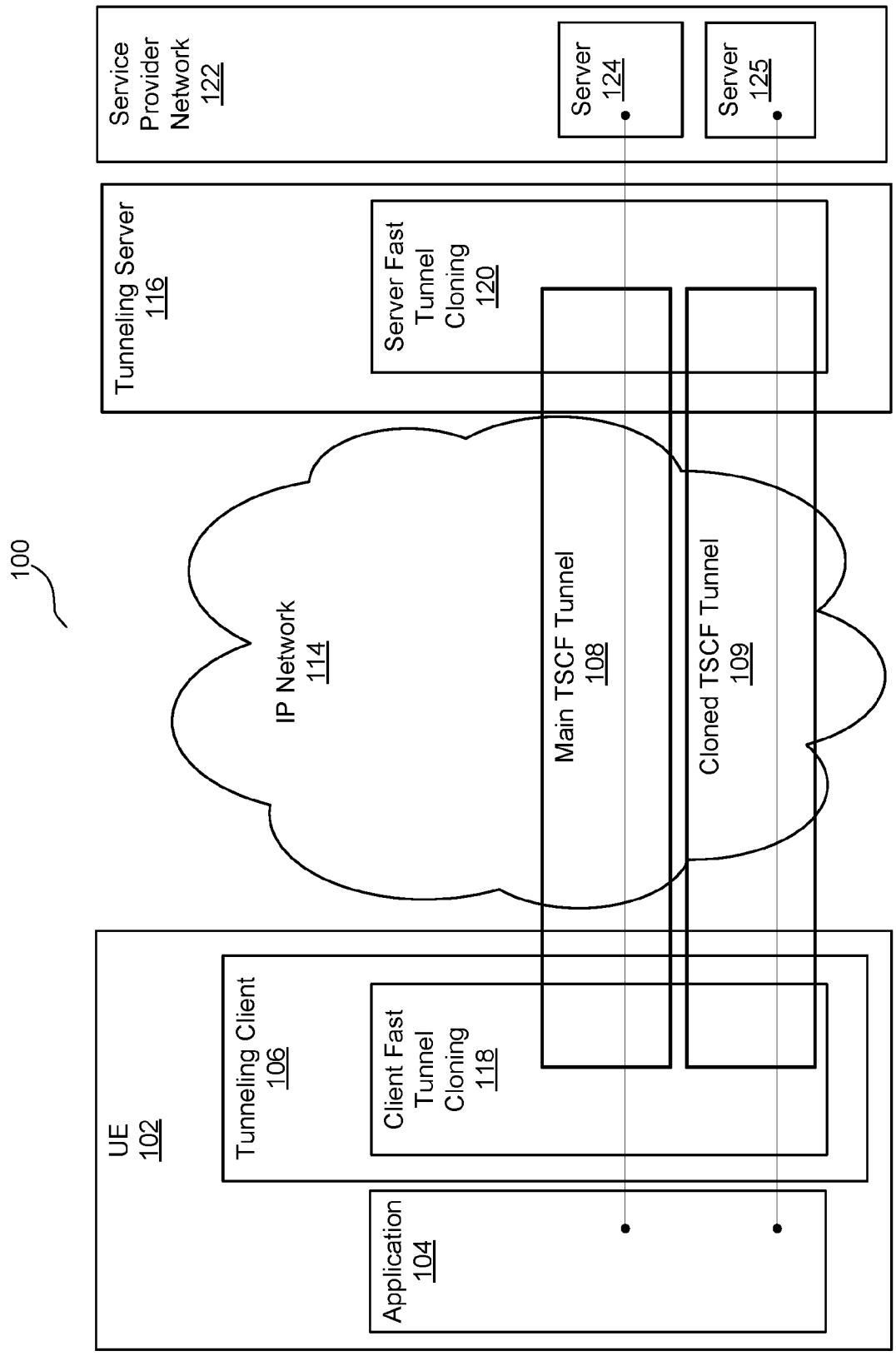

100
UE
102
Application
104
Tunneling Client
106
Client Fast
Tunnel
Cloning
118
IP Network
114
Main TSCF Tunnel
108
Cloned TSCF Tunnel
109
Tunneling Server
116
Server Fast
Tunnel
Cloning
120
Service
Provider
Network
122
Server
124
Server
125

Fig. 1

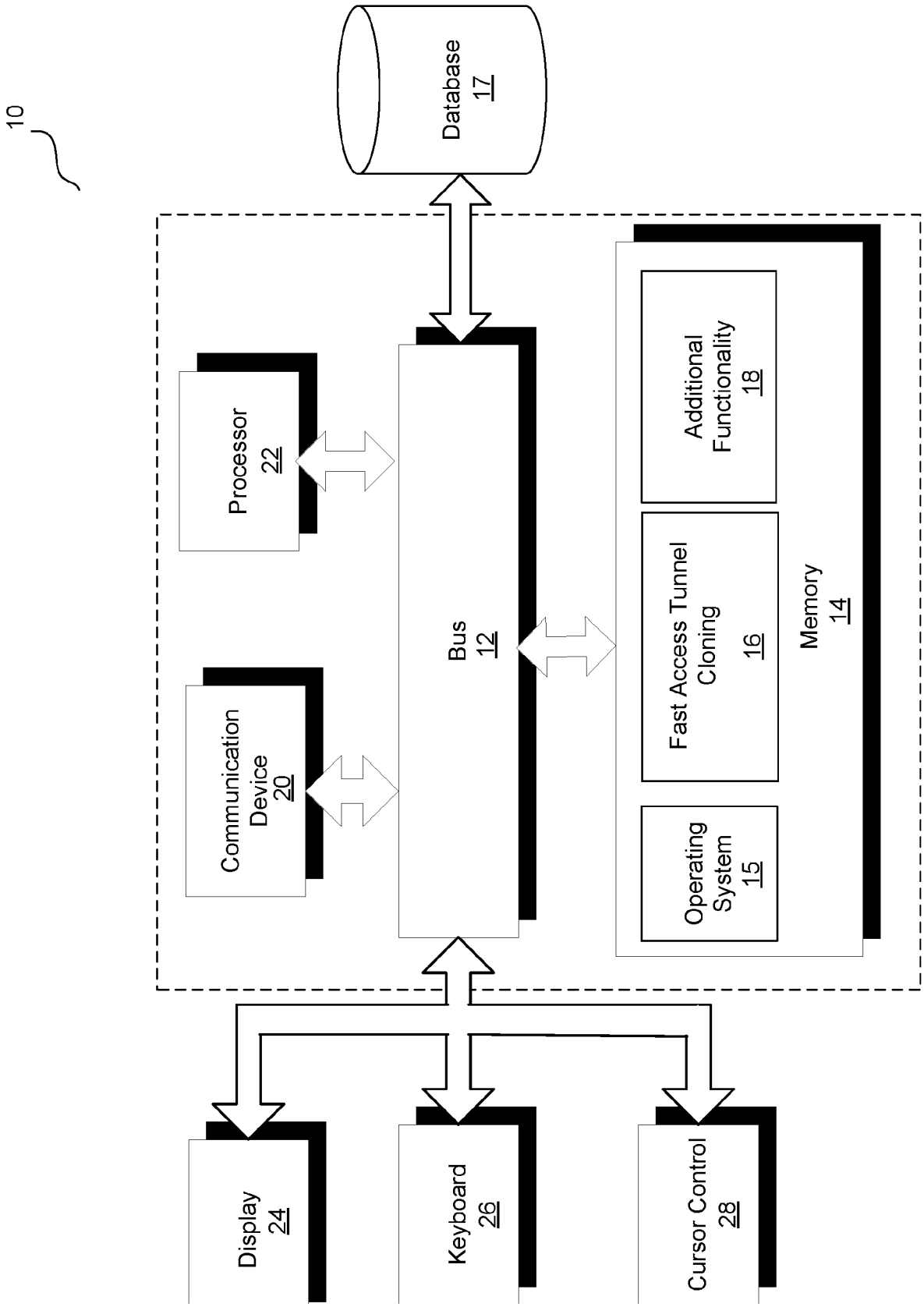

10
Database
17
Processor
22
Bus
12
Additional
Functionality
18
Fast Access Tunnel
Cloning
16
Memory
14
Operating
System
15
Communication
Device
20
Display
24
Keyboard
26
Cursor Control
28

Fig. 2

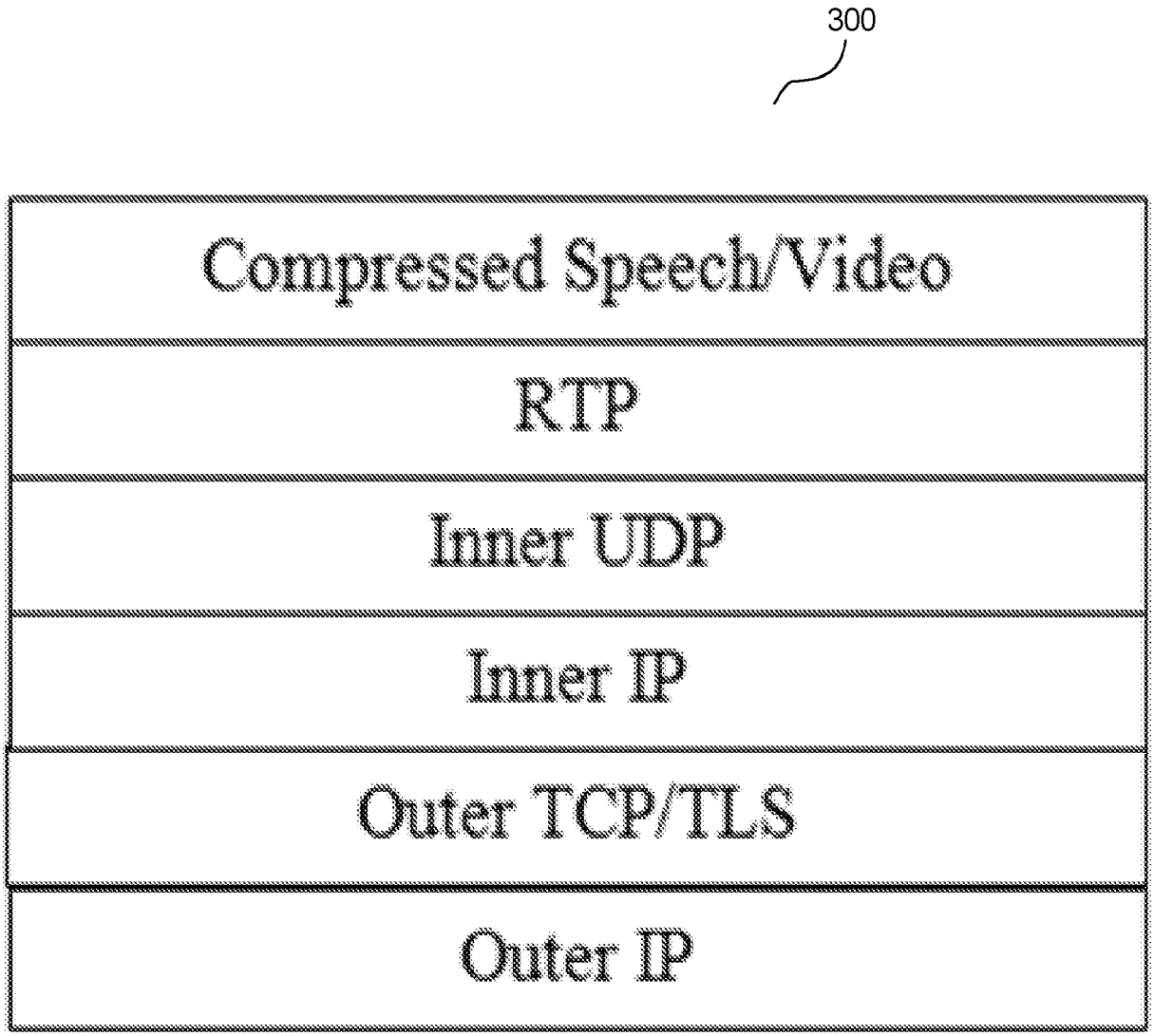

300
Compressed Speech/Video
RTP
Inner UDP
Inner IP
Outer TCP/TLS
Outer IP

Fig. 3

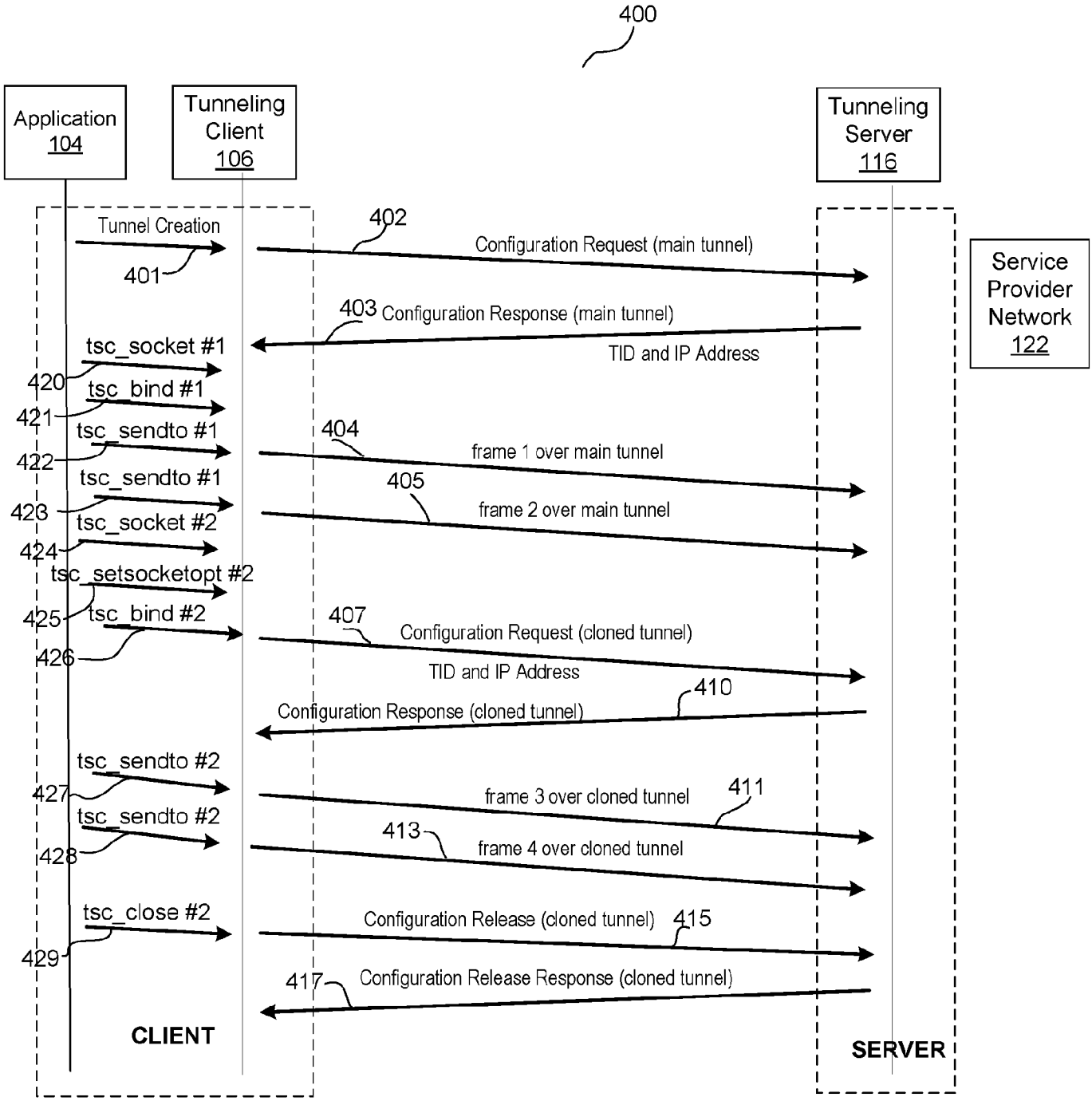
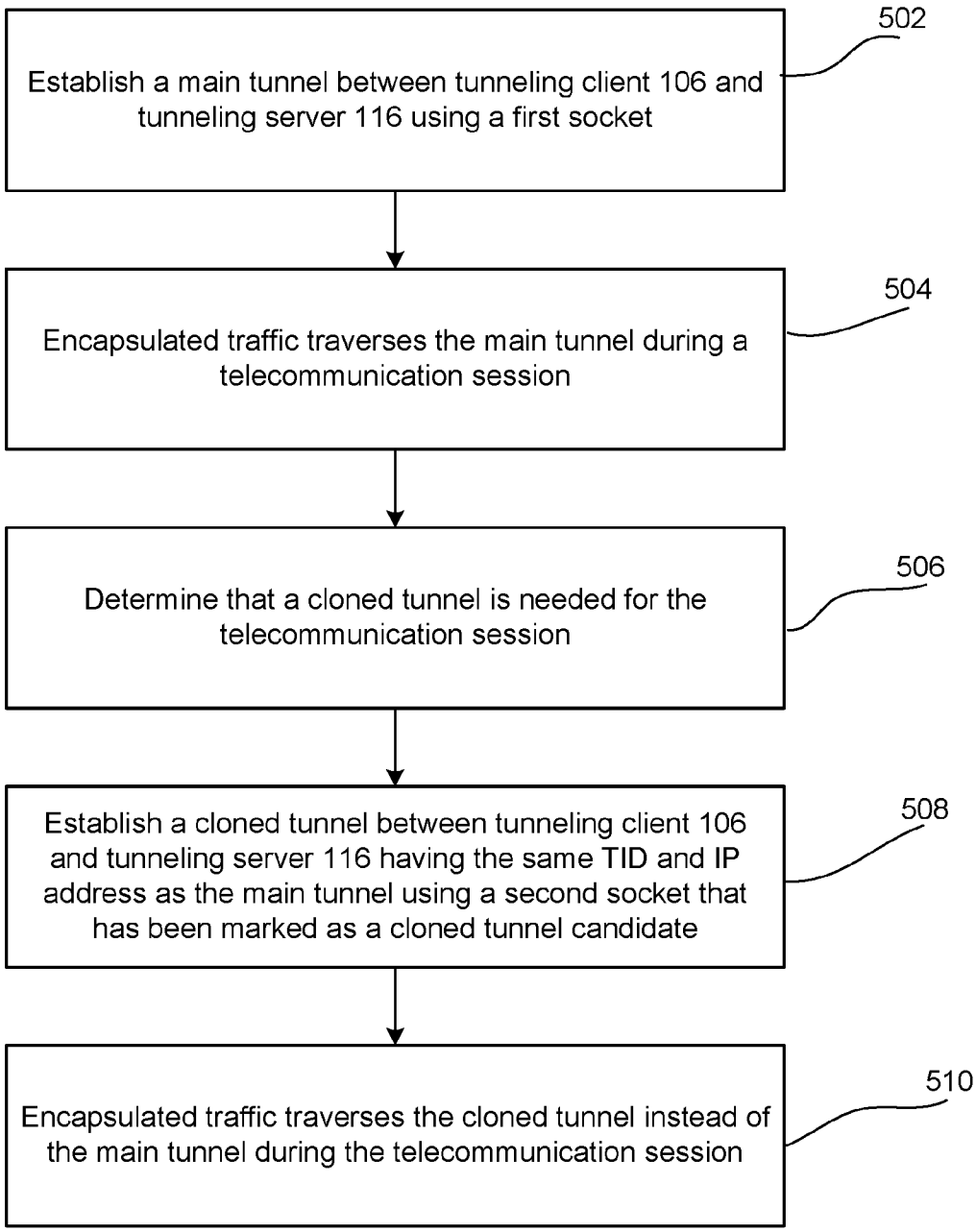

400
Application 104
Tunneling Client 106
Tunneling Server 116
Service Provider Network 122
Tunnel Creation
401
402
Configuration Request (main tunnel)
403 Configuration Response (main tunnel)
TID and IP Address
tsc_socket #1
420
tsc_bind #1
421
tsc_sendto #1
422
404
frame 1 over main tunnel
tsc_sendto #1
423
405
frame 2 over main tunnel
tsc_socket #2
424
tsc_setsocketopt #2
425
tsc_bind #2
426
407
Configuration Request (cloned tunnel)
TID and IP Address
410
Configuration Response (cloned tunnel)
tsc_sendto #2
427
frame 3 over cloned tunnel
411
tsc_sendto #2
428
413
frame 4 over cloned tunnel
tsc_close #2
429
Configuration Release (cloned tunnel)
415
417
Configuration Release Response (cloned tunnel)
CLIENT
SERVER

Fig. 4


Establish a main tunnel between tunneling client 106 and tunneling server 116 using a first socket
502
Encapsulated traffic traverses the main tunnel during a telecommunication session
504
Determine that a cloned tunnel is needed for the telecommunication session
506
Establish a cloned tunnel between tunneling client 106 and tunneling server 116 having the same TID and IP address as the main tunnel using a second socket that has been marked as a cloned tunnel candidate
508
Encapsulated traffic traverses the cloned tunnel instead of the main tunnel during the telecommunication session
510

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/041306

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/46 H04L29/06
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2015/085664 A1 (SACHDEV VINEET [US] ET AL) 26 March 2015 (2015-03-26) the whole document	1-20
A	US 2016/112372 A1 (KATZ HENRY [US] ET AL) 21 April 2016 (2016-04-21) the whole document	1-20
A	EP 2 826 210 A1 (ERICSSON TELEFON AB L M [SE]) 21 January 2015 (2015-01-21) the whole document	1-20
A	EP 2 908 491 A1 (HOB GMBH & CO KG [DE]) 19 August 2015 (2015-08-19) the whole document	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

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

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 September 2017	09/10/2017

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Jimenez Hernandez, P

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2017/041306

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015085664 A1	26-03-2015	US 2015085664 A1	26-03-2015
		US 2016165480 A1	09-06-2016
		US 2016192232 A1	30-06-2016
US 2016112372 A1	21-04-2016	NONE	
EP 2826210 A1	21-01-2015	CN 104170329 A	26-11-2014
		EP 2826210 A1	21-01-2015
		US 2015043350 A1	12-02-2015
		WO 2013135753 A1	19-09-2013
EP 2908491 A1	19-08-2015	EP 2908491 A1	19-08-2015
		US 2015229490 A1	13-08-2015