



- (51) **International Patent Classification:**
H04L 29/08 (2006.01) **H04W 76/02** (2009.01)

(21) **International Application Number:**
PCT/US2016/065923

(22) **International Filing Date:**
9 December 2016 (09.12.2016)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
62/266,303 11 December 2015 (11.12.2015) US

(71) **Applicant:** **INTERDIGITAL PATENT HOLDINGS, INC.** [US/US]; 200 Bellevue Parkway, Suite 300, Wilmington, Delaware 19809-3727 (US).

(72) **Inventors:** **DE FOY, Xavier**; 1000 Sherbrooke Street West, 10th Floor, Montreal, Québec H3A 3G4 (CA). **REZNIK, Alexander**; 416 S. Main Street, Pennington, New Jersey 08534 (US).

(74) **Agent:** **LOMBARDO, Jonathan G.**; Volpe and Koenig, P.C., United Plaza, Suite 1800, 30 S. 17th Street, Philadelphia, Pennsylvania 19103 (US).

(81) **Designated States** (*unless otherwise indicated, for every kind of regional 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, 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))*

Published:

— with international search report (Art. 21(3))

- (54) Title:** METHOD AND APPARATUS FOR ENABLING THIRD PARTY EDGE CLOUDS AT THE MOBILE EDGE

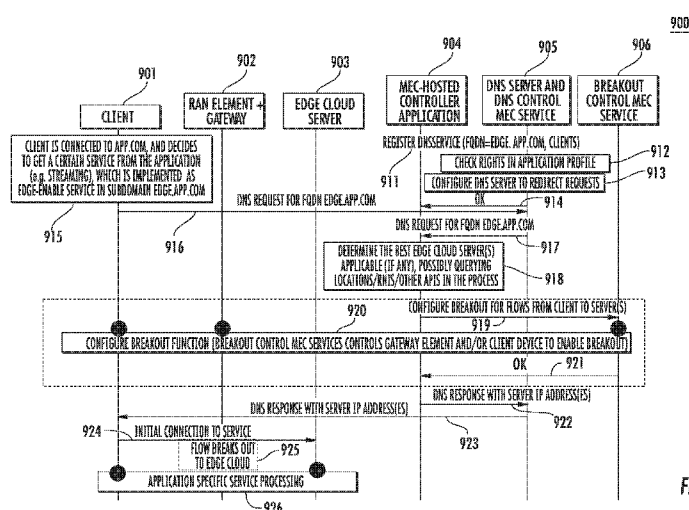


FIG. 9

- (S7) Abstract:** A mobile edge computing (MEC) server receives a content request from a wireless transmit/receive unit (WTRU) such as a domain name service (DNS) request. The MEC server may determine at least one third party edge cloud server to allocate to the content request based on content in the content request, such as a service from an application available at a website, and may trigger creation of an application instance in the at least one third party edge cloud server. The MEC server may configure a breakout function to enable service flows between the WTRU and the at least one third party edge cloud server and may transmit a DNS response to enable application connectivity between the WTRU and the at least one third party edge cloud server. The MEC server may embed a DNS server to redirect content requests to a MEC-hosted controller application hosted on the MEC server.

METHOD AND APPARATUS FOR ENABLING THIRD PARTY EDGE CLOUDS AT THE MOBILE EDGE

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application Serial No. 62/266,303 filed December 11, 2015, the contents of which are hereby incorporated by reference herein.

BACKGROUND

[0002] Edge computing extends cloud computing and services to the edge of a network, for example, using computing nodes deployed inside access networks, mobile devices, or IoT end devices such as sensors and actuators. Edge computing has the potential to provide data, computing, storage, and application services at the network edge using methods similar to cloud computing in remote data centers. The field of edge computing may include developments toward mobile network applications (i.e., mobile edge computing) and IoT-focused applications (i.e., fog computing).

SUMMARY

[0003] Methods and apparatuses for enabling third party edge clouds at the mobile edge are described. A wireless transmit/receive unit (WTRU) may include a transceiver that transmits, to a domain name service (DNS) server, a DNS request for a fully qualified domain name (FQDN) corresponding to an edge-enabled service of an application. Further, the transceiver may receive a DNS response including an internet protocol (IP) address of an edge cloud server for the edge-enabled service on a condition that an edge cloud instance exists.

[0004] In one embodiment, a mobile edge computing (MEC) server may receive a content request from a wireless transmit/receive unit (WTRU). The content request may be, for example, a domain name service (DNS) request. The MEC server may then determine at least one third party edge cloud server to

allocate to the content request based on content in the content request, and may possibly trigger the creation of an application instance in the at least one third party edge cloud server. For example, the content request may be for a service from an application available at a website. The MEC server may then configure a breakout function to enable service flows between the WTRU and at least one third party edge cloud server third party edge cloud server. The MEC server may then transmit a DNS response to enable application connectivity between the WTRU and third party edge cloud server. The MEC server may embed a DNS server to redirect content requests to an MEC-hosted controller application hosted on the MEC server.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] A more detailed understanding may be had from the following description, given by way of example in conjunction with the accompanying drawings wherein:

[0006] FIG. 1A is a system diagram of an example communications system in which one or more disclosed embodiments may be implemented;

[0007] FIG. 1B is a system diagram of an example wireless transmit/receive unit (WTRU) that may be used within the communications system illustrated in FIG. 1A;

[0008] FIG. 1C is a system diagram of an example radio access network and an example core network that may be used within the communications system illustrated in FIG. 1A;

[0009] FIG. 2 is a diagram of an example system for edge cloud providers offering services over a mobile network;

[0010] FIG. 3 is a diagram of another example system for a mobile network operator offering services over edge cloud providers;

[0011] FIG. 4 is a flow diagram of an example process for redirecting a content request to a server in an edge cloud;

[0012] FIG. 5 is a flow diagram of an example process for redirecting a content request to a server in an edge cloud;

[0013] FIG. 6 is a flow diagram of an example MEC-hosted controller application access authorization process;

[0014] FIG. 7 is a flow diagram of an example process for instance management in a MEC-hosted controller application;

[0015] FIG. 8 is a flow diagram of an example process for in-path request routing by a MEC-hosted controller application;

[0016] FIG. 9 is a flow diagram of an example process for in-path request routing by a MEC-hosted controller application with an embedded;

[0017] FIG. 10 is a flow diagram of an example process for in-path request routing by a MEC-hosted controller application with the breakout and DNS configured in advance;

[0018] FIG. 11 is a flow diagram of an example process for in-path request routing by a MEC-hosted controller application with local breakout on a client using a dynamic method of IP routing;

[0019] FIG. 12 is a flow diagram of an example process for in-path request routing by a MEC-hosted controller application with local breakout on a client using an ANDSF-based IP routing;

[0020] FIG. 13 is a diagram of an example schema for an edge cloud managed ANDSF management object; and

[0021] FIG. 14 is a flow diagram of an example process for processing of edge cloud access rights.

DETAILED DESCRIPTION

[0022] FIG. 1A is a diagram of an example communications system 100 in which one or more disclosed embodiments may be implemented. The communications system 100 may be a multiple access system that provides content, such as voice, data, video, messaging, broadcast, etc., to multiple wireless users. The communications system 100 may enable multiple wireless users to access such content through the sharing of system resources, including

wireless bandwidth. For example, the communications systems 100 may employ one or more channel access methods, such as code division multiple access (CDMA), time division multiple access (TDMA), frequency division multiple access (FDMA), orthogonal FDMA (OFDMA), single-carrier FDMA (SC-FDMA), and the like.

[0023] As shown in FIG. 1A, the communications system 100 may include wireless transmit/receive units (WTRUs) 102a, 102b, 102c, 102d, a radio access network (RAN) 104, a core network 106, a public switched telephone network (PSTN) 108, the Internet 110, and other networks 112, though it will be appreciated that the disclosed embodiments contemplate any number of WTRUs, base stations, networks, and/or network elements. Each of the WTRUs 102a, 102b, 102c, 102d may be any type of device configured to operate and/or communicate in a wireless environment. By way of example, the WTRUs 102a, 102b, 102c, 102d may be configured to transmit and/or receive wireless signals and may include user equipment (UE), a mobile station, a fixed or mobile subscriber unit, a pager, a cellular telephone, a personal digital assistant (PDA), a smartphone, a laptop, a netbook, a personal computer, a wireless sensor, consumer electronics, and the like.

[0024] The communications systems 100 may also include a base station 114a and a base station 114b. Each of the base stations 114a, 114b may be any type of device configured to wirelessly interface with at least one of the WTRUs 102a, 102b, 102c, 102d to facilitate access to one or more communication networks, such as the core network 106, the Internet 110, and/or the other networks 112. By way of example, the base stations 114a, 114b may be a base transceiver station (BTS), a Node-B, an eNodeB, a Home Node B, a Home eNodeB, a site controller, an access point (AP), a wireless router, and the like. While the base stations 114a, 114b are each depicted as a single element, it will be appreciated that the base stations 114a, 114b may include any number of interconnected base stations and/or network elements.

[0025] The base station 114a may be part of the RAN 104, which may also include other base stations and/or network elements (not shown), such as a base station controller (BSC), a radio network controller (RNC), relay nodes, etc. The base station 114a and/or the base station 114b may be configured to transmit and/or receive wireless signals within a particular geographic region, which may be referred to as a cell (not shown). The cell may further be divided into cell sectors. For example, the cell associated with the base station 114a may be divided into three sectors. Thus, in one embodiment, the base station 114a may include three transceivers, i.e., one for each sector of the cell. In another embodiment, the base station 114a may employ multiple-input multiple-output (MIMO) technology and, therefore, may utilize multiple transceivers for each sector of the cell.

[0026] The base stations 114a, 114b may communicate with one or more of the WTRUs 102a, 102b, 102c, 102d over an air interface 116, which may be any suitable wireless communication link (e.g., radio frequency (RF), microwave, infrared (IR), ultraviolet (UV), visible light, etc.). The air interface 116 may be established using any suitable radio access technology (RAT).

[0027] More specifically, as noted above, the communications system 100 may be a multiple access system and may employ one or more channel access schemes, such as CDMA, TDMA, FDMA, OFDMA, SC-FDMA, and the like. For example, the base station 114a in the RAN 104 and the WTRUs 102a, 102b, 102c may implement a radio technology such as Universal Mobile Telecommunications System (UMTS) Terrestrial Radio Access (UTRA), which may establish the air interface 116 using wideband CDMA (WCDMA). WCDMA may include communication protocols such as High-Speed Packet Access (HSPA) and/or Evolved HSPA (HSPA+). HSPA may include High-Speed Downlink Packet Access (HSDPA) and/or High-Speed Uplink Packet Access (HSUPA).

[0028] In another embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement a radio technology such as Evolved UMTS

Terrestrial Radio Access (E-UTRA), which may establish the air interface 116 using Long Term Evolution (LTE) and/or LTE-Advanced (LTE-A).

[0029] In other embodiments, the base station 114a and the WTRUs 102a, 102b, 102c may implement radio technologies such as IEEE 802.16 (i.e., Worldwide Interoperability for Microwave Access (WiMAX)), CDMA2000, CDMA2000 1X, CDMA2000 EV-DO, Interim Standard 2000 (IS-2000), Interim Standard 95 (IS-95), Interim Standard 856 (IS-856), Global System for Mobile communications (GSM), Enhanced Data rates for GSM Evolution (EDGE), GSM EDGE (GERAN), and the like.

[0030] The base station 114b in FIG. 1A may be a wireless router, Home Node B, Home eNodeB, or access point, for example, and may utilize any suitable RAT for facilitating wireless connectivity in a localized area, such as a place of business, a home, a vehicle, a campus, and the like. In one embodiment, the base station 114b and the WTRUs 102c, 102d may implement a radio technology such as IEEE 802.11 to establish a wireless local area network (WLAN). In another embodiment, the base station 114b and the WTRUs 102c, 102d may implement a radio technology such as IEEE 802.15 to establish a wireless personal area network (WPAN). In another embodiment, the base station 114b and the WTRUs 102c, 102d may utilize a cellular-based RAT (e.g., WCDMA, CDMA2000, GSM, LTE, LTE-A, etc.) to establish a picocell or femtocell. As shown in FIG. 1A, the base station 114b may have a direct connection to the Internet 110. Thus, the base station 114b may not be required to access the Internet 110 via the core network 106.

[0031] The RAN 104 may be in communication with the core network 106, which may be any type of network configured to provide voice, data, applications, and/or voice over internet protocol (VoIP) services to one or more of the WTRUs 102a, 102b, 102c, 102d. For example, the core network 106 may provide call control, billing services, mobile location-based services, pre-paid calling, Internet connectivity, video distribution, etc., and/or perform high-level security functions, such as user authentication. Although not shown in FIG. 1A, it will be

appreciated that the RAN 104 and/or the core network 106 may be in direct or indirect communication with other RANs that employ the same RAT as the RAN 104 or a different RAT. For example, in addition to being connected to the RAN 104, which may be utilizing an E-UTRA radio technology, the core network 106 may also be in communication with another RAN (not shown) employing a GSM radio technology.

[0032] The core network 106 may also serve as a gateway for the WTRUs 102a, 102b, 102c, 102d to access the PSTN 108, the Internet 110, and/or other networks 112. The PSTN 108 may include circuit-switched telephone networks that provide plain old telephone service (POTS). The Internet 110 may include a global system of interconnected computer networks and devices that use common communication protocols, such as the transmission control protocol (TCP), user datagram protocol (UDP) and the internet protocol (IP) in the TCP/IP internet protocol suite. The networks 112 may include wired or wireless communications networks owned and/or operated by other service providers. For example, the networks 112 may include another core network connected to one or more RANs, which may employ the same RAT as the RAN 104 or a different RAT.

[0033] Some or all of the WTRUs 102a, 102b, 102c, 102d in the communications system 100 may include multi-mode capabilities, i.e., the WTRUs 102a, 102b, 102c, 102d may include multiple transceivers, transmitters, or receivers for communicating with different wireless networks over different wireless links. For example, the WTRU 102c shown in FIG. 1A may be configured to communicate with the base station 114a, which may employ a cellular-based radio technology, and with the base station 114b, which may employ an IEEE 802 radio technology.

[0034] FIG. 1B is a system diagram of an example WTRU 102. As shown in FIG. 1B, the WTRU 102 may include a processor 118, a transceiver 120, a transmit/receive element 122, a speaker/microphone 124, a keypad 126, a display/touchpad 128, non-removable memory 130, removable memory 132, a power source 134, a global positioning system (GPS) chipset 136, and other

peripherals 138. It will be appreciated that the WTRU 102 may include any sub-combination of the foregoing elements while remaining consistent with an embodiment.

[0035] The processor 118 may be a general purpose processor, a special purpose processor, a conventional processor, a digital signal processor (DSP), a plurality of microprocessors, one or more microprocessors in association with a DSP core, a controller, a microcontroller, Application Specific Integrated Circuits (ASICs), Field Programmable Gate Array (FPGAs) circuits, any other type of integrated circuit (IC), a state machine, and the like. The processor 118 may perform signal coding, data processing, power control, input/output processing, and/or any other functionality that enables the WTRU 102 to operate in a wireless environment. The processor 118 may be coupled to the transceiver 120, which may be coupled to the transmit/receive element 122. While FIG. 1B depicts the processor 118 and the transceiver 120 as separate components, it will be appreciated that the processor 118 and the transceiver 120 may be integrated together in an electronic package or chip.

[0036] The transmit/receive element 122 may be configured to transmit signals to, or receive signals from, a base station (e.g., the base station 114a) over the air interface 116. For example, in one embodiment, the transmit/receive element 122 may be an antenna configured to transmit and/or receive RF signals. In another embodiment, the transmit/receive element 122 may be an emitter/detector configured to transmit and/or receive IR, UV, or visible light signals, for example. In another embodiment, the transmit/receive element 122 may be configured to transmit and receive both RF and light signals. It will be appreciated that the transmit/receive element 122 may be configured to transmit and/or receive any combination of wireless signals.

[0037] In addition, although the transmit/receive element 122 is depicted in FIG. 1B as a single element, the WTRU 102 may include any number of transmit/receive elements 122. More specifically, the WTRU 102 may employ MIMO technology. Thus, in one embodiment, the WTRU 102 may include two or

more transmit/receive elements 122 (e.g., multiple antennas) for transmitting and receiving wireless signals over the air interface 116.

[0038] The transceiver 120 may be configured to modulate the signals that are to be transmitted by the transmit/receive element 122 and to demodulate the signals that are received by the transmit/receive element 122. As noted above, the WTRU 102 may have multi-mode capabilities. Thus, the transceiver 120 may include multiple transceivers for enabling the WTRU 102 to communicate via multiple RATs, such as UTRA and IEEE 802.11, for example.

[0039] The processor 118 of the WTRU 102 may be coupled to, and may receive user input data from, the speaker/microphone 124, the keypad 126, and/or the display/touchpad 128 (e.g., a liquid crystal display (LCD) display unit or organic light-emitting diode (OLED) display unit). The processor 118 may also output user data to the speaker/microphone 124, the keypad 126, and/or the display/touchpad 128. In addition, the processor 118 may access information from, and store data in, any type of suitable memory, such as the non-removable memory 130 and/or the removable memory 132. The non-removable memory 130 may include random-access memory (RAM), read-only memory (ROM), a hard disk, or any other type of memory storage device. The removable memory 132 may include a subscriber identity module (SIM) card, a memory stick, a secure digital (SD) memory card, and the like. In other embodiments, the processor 118 may access information from, and store data in, memory that is not physically located on the WTRU 102, such as on a server or a home computer (not shown).

[0040] The processor 118 may receive power from the power source 134, and may be configured to distribute and/or control the power to the other components in the WTRU 102. The power source 134 may be any suitable device for powering the WTRU 102. For example, the power source 134 may include one or more dry cell batteries (e.g., nickel-cadmium (NiCd), nickel-zinc (NiZn), nickel metal hydride (NiMH), lithium-ion (Li-ion), etc.), solar cells, fuel cells, and the like.

[0041] The processor 118 may also be coupled to the GPS chipset 136, which may be configured to provide location information (e.g., longitude and latitude) regarding the current location of the WTRU 102. In addition to, or in lieu of, the information from the GPS chipset 136, the WTRU 102 may receive location information over the air interface 116 from a base station (e.g., base stations 114a, 114b) and/or determine its location based on the timing of the signals being received from two or more nearby base stations. It will be appreciated that the WTRU 102 may acquire location information by way of any suitable location-determination method while remaining consistent with an embodiment.

[0042] The processor 118 may further be coupled to other peripherals 138, which may include one or more software and/or hardware modules that provide additional features, functionality and/or wired or wireless connectivity. For example, the peripherals 138 may include an accelerometer, an e-compass, a satellite transceiver, a digital camera (for photographs or video), a universal serial bus (USB) port, a vibration device, a television transceiver, a hands free headset, a Bluetooth® module, a frequency modulated (FM) radio unit, a digital music player, a media player, a video game player module, an Internet browser, and the like.

[0043] FIG. 1C is a system diagram of the RAN 104 and the core network 106 according to an embodiment. As noted above, the RAN 104 may employ an E-UTRA radio technology to communicate with the WTRUs 102a, 102b, 102c over the air interface 116. The RAN 104 may also be in communication with the core network 106.

[0044] The RAN 104 may include eNodeBs 140a, 140b, 140c, though it will be appreciated that the RAN 104 may include any number of eNodeBs while remaining consistent with an embodiment. The eNodeBs 140a, 140b, 140c may each include one or more transceivers for communicating with the WTRUs 102a, 102b, 102c over the air interface 116. In one embodiment, the eNodeBs 140a, 140b, 140c may implement MIMO technology. Thus, the eNodeB 140a, for

example, may use multiple antennas to transmit wireless signals to, and receive wireless signals from, the WTRU 102a.

[0045] Each of the eNodeBs 140a, 140b, 140c may be associated with a particular cell (not shown) and may be configured to handle radio resource management decisions, handover decisions, scheduling of users in the uplink and/or downlink, and the like. As shown in FIG. 1C, the eNodeBs 140a, 140b, 140c may communicate with one another over an X2 interface.

[0046] The core network 106 shown in FIG. 1C may include a mobility management entity gateway (MME) 142, a serving gateway 144, and a packet data network (PDN) gateway 146. While each of the foregoing elements are depicted as part of the core network 106, it will be appreciated that any one of these elements may be owned and/or operated by an entity other than the core network operator.

[0047] The MME 142 may be connected to each of the eNodeBs 140a, 140b, 140c in the RAN 104 via an S1 interface and may serve as a control node. For example, the MME 142 may be responsible for authenticating users of the WTRUs 102a, 102b, 102c, bearer activation/deactivation, selecting a particular serving gateway during an initial attach of the WTRUs 102a, 102b, 102c, and the like. The MME 142 may also provide a control plane function for switching between the RAN 104 and other RANs (not shown) that employ other radio technologies, such as GSM or WCDMA.

[0048] The serving gateway 144 may be connected to each of the eNodeBs 140a, 140b, 140c in the RAN 104 via the S1 interface. The serving gateway 144 may generally route and forward user data packets to/from the WTRUs 102a, 102b, 102c. The serving gateway 144 may also perform other functions, such as anchoring user planes during inter-eNodeB handovers, triggering paging when downlink data is available for the WTRUs 102a, 102b, 102c, managing and storing contexts of the WTRUs 102a, 102b, 102c, and the like.

[0049] The serving gateway 144 may also be connected to the PDN gateway 146, which may provide the WTRUs 102a, 102b, 102c with access to packet-

switched networks, such as the Internet 110, to facilitate communications between the WTRUs 102a, 102b, 102c and IP-enabled devices.

[0050] The core network 106 may facilitate communications with other networks. For example, the core network 106 may provide the WTRUs 102a, 102b, 102c with access to circuit-switched networks, such as the PSTN 108, to facilitate communications between the WTRUs 102a, 102b, 102c and traditional land-line communications devices. For example, the core network 106 may include, or may communicate with, an IP gateway (e.g., an IP multimedia subsystem (IMS) server) that serves as an interface between the core network 106 and the PSTN 108. In addition, the core network 106 may provide the WTRUs 102a, 102b, 102c with access to the networks 112, which may include other wired or wireless networks that are owned and/or operated by other service providers.

[0051] Other network 112 may further be connected to an IEEE 802.11 based wireless local area network (WLAN) 160. The WLAN 160 may include an access router 165. The access router may contain gateway functionality. The access router 165 may be in communication with a plurality of access points (APs) 170a, 170b. The communication between access router 165 and APs 170a, 170b may be via wired Ethernet (IEEE 802.3 standards), or any type of wireless communication protocol. AP 170a is in wireless communication over an air interface with WTRU 102d.

[0052] A WTRU, as used in the embodiments described herein and the associated Figures, may include any one of the following: a UE; a STA; a mobile device (i.e., smart phones); a client device or a device running a client application as described herein; an end devices (i.e., sensor, actuator, TV, etc.); or any other device configured to operate in a wireless communication system.

[0053] The following terms are used herein:

[0054] OTT – Over-The-Top

[0055] AP – Access Point

[0056] MEC – Mobile Edge Computing

- [0057] ETSI - European Telecommunications Standards Institute
- [0058] DNS – Domain Name System
- [0059] LIPA – Local IP Access
- [0060] SIPTO - Selective IP Traffic Offload
- [0061] IFOM – IP Flow Mobility
- [0062] API – Application Programming Interface
- [0063] FQDN – Fully Qualified Domain Name
- [0064] HSS – Home Subscriber Service
- [0065] TOF – Traffic Offload Function (or Traffic Rules Control)
- [0066] MNO – Mobile Network Operator
- [0067] MN – Mobile Network
- [0068] VM – Virtual Machines
- [0069] ANDSF – Access Network Discovery and Selection Function
- [0070] The term “MEC Service” is used to designate a service implemented as part of the MEC server platform, and under the control of the network operator. Such a service may provide an API that may be accessed by MEC applications.
- [0071] The term “MEC application” or “MEC-hosted application” may be used to designate a software application that runs over the MEC server platform (e.g., on a VM hosted on an MEC server).
- [0072] There are several drivers for edge computing in the European Telecommunications Standards Institute (ETSI) mobile edge computing (MEC). One driver for edge computing is that Network Operators may be willing to provide additional value added services and bring better performance and Quality of Experience (QoE) to end users, by leveraging the characteristics of their Access Network such as proximity to the end user, awareness of users’ identity, etc. Another major driver for edge computing is the need to complement underpowered IoT devices with computing capability at the edge at the network, in order to enable complex operations or operations involving large amount of data and devices, which would simply not be possible otherwise due to latency

and capacity limitation introduced by backhaul links. Another driver for edge computing comes from the development of cloud computing itself, which leads to further integration of software development and deployment activities, as illustrated by the DevOps model of development, in order to cope with increasing system complexity. This trend, enabled by technologies like network and function virtualization, may also be described as merging network infrastructure with the IT world, and at its core aims to reduce CAPEX and OPEX for the application provider. Edge Computing may be seen as a way to extend this new flexibility out of the data centers into the rest of the Internet and even end user devices, which ultimately facilitate innovation for new classes of applications not well served by Distant Clouds. Large manufacturers and Network Operators are also supporting major initiatives in this area.

[0073] ETSI MEC has not provided a solution for mobile edge computing or a mechanism for third party cloud operators to cooperate with cellular network operators. In particular, ETSI has not defined a mechanism that enables third party cloud operators to provide edge cloud resources close to mobile clients (e.g., one wireless hop away) within the current MEC framework defined in ETSI MEC.

[0074] The ETSI MEC framework describes that MEC servers may hold MEC applications and may expose service application programming interfaces (APIs) to them. Services provided using this model may include a traffic offload function (also referred to as traffic rules control), a radio network information service, communication services or a service registry. However, this model does not consider the possibility of running network applications on a device at home or in an enterprise (unless the network operator places a MEC server there) or on any hardware not under the control of the mobile operator. If the application does not need direct access to mobile network-specific services, such as the radio network information service, this constraint may be unnecessary. Similarly, using this model, third party edge cloud operators may be required to use the network operator MEC feature, which may not hold enough capacity or agility to

adapt to a changing marketplace. Further, this model may pose a problem for the mobile network operators themselves, which would need to maintain all of the edge computing capacity between their end users and the Internet.

[0075] While certain classes of applications may need to remain under strict control of the operator (e.g., applications requiring access to radio information or application optimizing network operations), the vast majority of networked applications do not require such strict control and also involve interacting with a large number of application providers.

[0076] Hosting part of the control plane of edge clouds in mobile networks is described herein. The system described herein provides embodiments in which a third-party edge cloud control plane is integrated within mobile networks. Controller applications may be integrated with the mobile network's DNS servers. Internet Protocol (IP) connectivity may be enabled between clients (i.e. a WTRU as defined herein) and edge clouds connected at various points in the mobile network (e.g., at the eNodeB or other breakout point). A new breakout function may be enabled in mobile networks, which may be managed dynamically. Additionally, edge cloud servers may be local. For example, edge cloud servers may be located at home (i.e. home servers) or in an enterprise (i.e. enterprise server, which may include a server that serves the needs of an enterprise) that may be reached through an interface not managed by the mobile network but that may be made available as edge cloud resources for applications served over the mobile network. These local edge servers may be accessible directly using IP routing or with additional IP routes to be installed on the client device. Integrating access rights with edge cloud usage and tying access rights to client IP addresses and/or content requests are also described herein.

[0077] Enabling a third party edge cloud control plane in mobile networks may be achieved through the use of MEC applications and extensions of sets of APIs for DNS control, breakout control, and edge cloud control

[0078] The embodiments described herein may address the issues identified above by enabling mobile network operators to leverage their

knowledge of user location and context to host third party edge cloud controllers as MEC applications (referred to herein as a “MEC-hosted controller application” or “application”). The edge clouds themselves are typically (but not necessarily) located outside of the cellular network and may be accessed by clients either through the mobile network (e.g., through local breakout) or through another radio access technology (RAT) interface at the client device (i.e. a WTRU as defined herein). The embodiments described herein may include cellular network operators providing additional control to MEC-hosted controller applications through new and/or enhanced MEC APIs. The MEC-hosted controller application may perform functions including but not limited to the following

[0079] The MEC-hosted controller applications may be able to control the behavior of the operator’s Domain Name System (DNS) service. In particular, the MEC-hosted controller application may request that requests for a given fully qualified domain name (FQDN) from a particular client (or set of clients/IP blocks) be resolved as a particular set of IP addresses, which may enable the MEC-hosted controller application to allocate specific application instances to specific clients.

[0080] The MEC-hosted controller application may control local breakout at various points in the network including on the client (e.g., local IP access (LIPA), selective IP traffic offload (SIPTO), IP flow mobility (IFOM) or connection manager on the client) and report usage information back to the MEC-hosted controller application. The breakout API may also enable manipulating IP routing on the client, for example, to add a route to an edge cloud through another interface.

[0081] The MEC-hosted controller application may control the edge cloud either directly through an MEC API or through a cloud controller located in the core network, the Internet, and/or a gateway to such a controller.

[0082] The radio network information service or other MEC services may enable the MEC-hosted controller application to determine the best possible application instances to serve clients by enabling access to information including

but not limited to a location of a client device, signal strength, and other radio information. The radio network information service or other MEC services may be enhanced to provide this information. For example, the traffic offload function (TOF) (also referred to as Traffic Rules Control (TRC)) may be enhanced to report statistics on certain flows that are not to be offloaded.

[0083] The MEC-hosted controller application may be integrated with the mobile network and DNS servers of the mobile network using approaches including but not limited to the following: the MEC-hosted controller may statically configure the local DNS server; the MEC-hosted DNS server may dynamically exchange messages with the local DNS server as part of servicing DNS requests; or the MEC-hosted controller application may implement a DNS server and configure the local DNS server to see it as authoritative for specific FQDNs.

[0084] Breakout and edge cloud control may be collocated in the MEC server, for example, with an independent breakout function at the MEC server. The breakout function may physically forward data packets toward a breakout interface. The breakout control function may exchange control messages with breakout functions including the MEC breakout function described herein, as well as, for example, local gateways at eNodeBs, etc.

[0085] Embodiments are described herein that may handle local edge clouds, which may include reporting IP routing information up to the MEC-hosted controller application, and, if necessary, the MEC-hosted controller application may request new IP routes to be added on the client device. A client may be directed to use an IP-routable local edge cloud.

[0086] Procedures are also described herein whereby the MEC-hosted controller application may trigger the creation of edge cloud service instances, in advance, when possible, or otherwise when requested. Where creation of edge cloud service instances is requested, the client may handle a response from a fallback instance, which may or may not be application-specific. For example, it

may provide a degraded experience for a limited time or may request or trigger the client application to request to wait for some time before obtaining service.

[0087] With regard to access rights and their relationship to client IP addresses, procedures are described herein whereby the MEC-hosted controller application may retrieve a list of private edge clouds that may be accessed by a particular IP address and whereby a new authorization MEC service may be cooperating with the mobile network to map the client IP address with an identity and then obtain subscriber profile information using the identity.

[0088] FIG. 2 is a diagram of an example system 200 for edge cloud providers offering services over a mobile network in accordance with one embodiment, which may be used in combination with any of the embodiments described herein. In the example of FIG. 2, a third party cloud provider may deploy MEC controller applications, including but not limited to the MEC-hosted controller application. The cloud providers may be customers of the mobile network operator and may deploy MEC controller applications and gateways into the mobile network. Further, multiple application providers may be customers of the cloud provider and may deploy their applications over the cloud. These applications may then be deployed in edge clouds by the cloud provider as described herein.

[0089] The example of FIG. 2 includes the main functions and interfaces of the system 200. For simplicity, some interfaces and functions defined in cellular networks and in ETSI MEC are not shown but may also be used in the system 200. Referring to FIG. 2, the system 200 may include mobile network operator 224, cellular edge network 225, cellular core network 233, MEC server 201, MEC-hosted controller application 202 (which may also be referred to as MEC-hosted cloud controller application), DNS control service 209, DNS (proxy) servers 210a and 210b, breakout control service 211, over-the-top (OTT) cloud provider 226, OTT cloud control service 203, client device 227 (which may be a WTRU as defined herein), client application 228, edge cloud control service gateway 204,

edge clouds 205 including local edge cloud 206, first-hop edge cloud 207, and intermediate edge cloud 208.

[0090] In the example of FIG. 2, several interfaces are shown. ECC1 216, as referred to herein, is the control interface (e.g., remote access to OpenStack control services) between OTT edge cloud control service 203 and edge clouds 205. ECC2 217, as referred to herein, is the interface between edge cloud control service gateway 204 and OTT edge cloud control service 203. ECC3 218, as referred to herein, is the interface between edge cloud control service gateway 204 and MEC-hosted controller application 202.

[0091] MEC server 201 may be an edge cloud server hosted at a base station, concentration point, or eNodeB. MEC-hosted controller application 202 is a type of MEC application, which may perform functions including but not limited to redirecting client requests toward application instances in one of the plurality of edge clouds 205. CM 215, as referred to herein, is the interface between MEC applications and MEC application platform services. In the example of FIG. 2, CM 215 provides an interface among MEC-hosted controller application 202, DNS control service 209, and breakout control service 211. MEC APIs may be defined over an interface such as CM 215.

[0092] OTT cloud control service 203 may be a service operated by a OTT cloud provider 226 (e.g., MICROSOFT, AMAZON, GOOGLE) accessible via the Internet 234 and may be tasked with ensuring that applications are instantiated according to certain rules set by the cloud provider, which may be related to a contract and/or settings set by the application provider. OTT cloud control service 203 may manage edge clouds 205 directly (e.g., creating or tearing down application instances), for example, through policies and/or direct control messages. OTT cloud control service 203 may also communicate with MEC applications including but not limited to MEC-hosted controller application 202. This communication may be via a gateway such as edge cloud control service gateway 204. OTT cloud control service 203 may hold application images 223,

which may be deployed into edge network instances. Application images 223 may be virtual machine (VM) images or application containers such as Dockerfiles.

[0093] Edge cloud control service gateway 204 may be a service running in cellular core network 233, which may be an intermediary for communication between MEC-hosted controller applications 202 and OTT cloud control service 203. Edge cloud control service gateway 204 may be implemented as an MEC application. Edge cloud control service gateway 204 may be used for reasons including but not limited to security reasons. For example, edge cloud control service gateway 204 may be used to avoid requiring direct communication between an OTT application and multiple applications. In another example, edge cloud control service gateway 204 may perform aggregation of upstream data from the MEC server 201 and operate as an intermediary for control messages between MEC-hosted controller applications 202 and OTT cloud control service 203.

[0094] Edge clouds 205 may be deployed over a number of existing cloud management technologies, including, for example, OpenStack. Edge clouds 205 may each include control services and compute and storage nodes. In the example of FIG. 2, the edge clouds 205 may differ, for example, by their point of attachment. While edge clouds 205 have separate control, compute and storage functions in the example of FIG. 2, it is also possible for these functions to be collocated in single nodes. For example, a home or enterprise local edge cloud may be composed of computers or gateways running a control agent that may be remotely controlled by a cloud provider. Further, while edge clouds 205 are examples of Infrastructure as a Service clouds (i.e., hosted applications may be packaged as Virtual Machines), some edge clouds 205 may be providing higher layer hosting services (i.e., hosted applications may be packaged as application containers, programs, service configuration profiles).

[0095] As shown in the example of FIG. 2, edge clouds 205 may include local edge cloud 206, first-hop edge cloud 207, and intermediate edge cloud 208. Local edge cloud 206 may include control services (e.g., OpenStack, Nova,

Neutron, etc.) 229c, compute nodes 230c, virtual app instance 231c, and storage nodes 232c. In the example of FIG. 2, client device 227 may connect to local edge cloud 206 via a local connection, such as Ethernet, WiFi, etc. over 0 or more hops 221c. Further, local edge cloud 206 may be accessible from client device 227 through a different access technology than the one or more technologies managed by the mobile network operator 224 (e.g., through a home or enterprise Wi-Fi access point (AP)). Home computing devices, such as home gateways, processors collocated with APs, or game consoles, may be used. Client traffic may reach local edge cloud 206 through IP routing. Techniques such as IFOM may also be used, although it may not be necessary to move flows between interfaces in general.

[0096] First-hop edge cloud 207 may include control services 229b, compute nodes 230b, virtual app instance 231b, and storage nodes 232b. In the example of FIG. 2, client device 227 may connect to first-hop edge cloud 207 via a local connection, such as Ethernet, WiFi, etc. over 0 or more hops 221b. Further, first-hop edge cloud 207 may be accessible through the first hop access point (AP or eNodeB) 226 managed by the mobile network operator 224. Client traffic may reach first-hop edge clouds through a local gateway 222b located on access point (AP or eNodeB) 226 (e.g., following breakout control 213 such as LIPA feature defined for 3GPP-compliant networks).

[0097] Intermediate edge cloud 208 may include control services 229a, compute nodes 230a, virtual app instance 231a, and storage nodes 232a. In the example of FIG. 2, client device 227 may connect to intermediate edge cloud 208 via a local connection, such as Ethernet, WiFi, etc. over 0 or more hops 221a. Further, intermediate edge cloud 208 may be accessible through a deeper point of attachment to the mobile network (e.g., through local gateway 222a or S/P-GW). Client traffic may reach intermediate edge cloud 208 through local gateway 222a (e.g., following breakout control 212 such as SIPTO feature defined for 3GPP compliant networks, such as by using separate bearers for client-edge cloud traffic).

[0098] DNS control service 209 may be a MEC application platform service offering an API to MEC-hosted controller application 202 to control DNS (proxy) server 210b. DNS (proxy) server 210a and DNS (proxy) server 210b may be a DNS server, such as a recursive DNS proxy, located in the cellular edge network 225 and/or the cellular core network 233. One of its roles, especially when located in cellular edge network 225, may be to reduce DNS latency for the client device 227 through caching. This role may be extended to also enable MEC-hosted controller application 202 to perform DNS request routing. CD 219, as referred to herein, provides a control interface between DNS control service 209 and DNS (proxy) server 210a and DNS (proxy) server 210b.

[0099] Breakout control service 211 may be an MEC application platform service offering an API to MEC-hosted controller application 202 to control breakout of data flows at various points in the network. The breakout control functions may include various control functions present throughout the network (e.g., in gateways) and performing control for break out of data flows. Breakout functions may be defined through technologies such as LIPA, SIPTO, and IFOM. Additionally, a program on the client device 227, such as a connection manager, may handle breakout on the client device 227 at the IP routing level. CB 220, as referred to herein, may provide an interface between breakout control service 211 and the breakout control functions such as breakout control 212, 213, and 214. CB 220 may provide an interface to breakout control 214 using IFOM or connection manager feature.

[0100] In alternative embodiments, mobile network operator 224 and OTT cloud provider 226 may be the same entity. Further, the OTT edge cloud control service 203 may be connected with edge cloud control service gateway 204 in several access networks, including other cellular and fixed networks, resulting in a global edge computing reach for applications hosted by the OTT cloud provider 226.

[0101] In alternative embodiments, the example system 200 of FIG. 2 may include a global cloud provider that may offer a seamless experience to the

application provider. The application provider may select an OTT cloud provider and deploy its applications over it.

[0102] FIG. 3 is a diagram of another example system 300 for a mobile network operator offering services over edge cloud providers in accordance with another embodiment, which may be used in combination with any of the embodiments described herein. In the example system 300 of FIG. 3, mobile network operator 324 may interconnect with multiple edge clouds 305 operated by one or more edge cloud providers. A virtual cloud provider may deploy MEC-hosted controller application 302. The mobile network operator 324 may be a customer of the edge cloud providers and may rent edge clouds or slices of edge clouds from these providers. Further, the virtual cloud provider may be a customer of mobile network operator 324 and may deploy MEC-hosted controller application 302 and a gateway into the mobile network. Multiple application providers may be customers of the virtual cloud provider. The multiple application providers may deploy their applications into the virtual cloud (e.g., through an OTT portal). In turn, the virtual cloud provider may deploy the application through one or more mobile networks. The mobile networks may deploy the applications into edge networks under the control of MEC-hosted controller application 302.

[0103] The example system 300 of FIG. 3 includes some of the same functions and interfaces included in the example system 200 in FIG. 2 and defined above. Referring to FIG. 3, the system 300 may include mobile network operator 324, cellular edge network 325, cellular core network 333, MEC server 301, MEC-hosted controller application 302 (which may also be referred to as MEC-hosted cloud controller application), DNS control service 309, DNS (proxy) servers 310a and 310b, breakout control service 311, client device 327 (which may be a WTRU as defined herein), client application 328, and edge clouds 305 including local edge cloud 306, first-hop edge cloud 307, and intermediate edge cloud 308.

[0104] CM 315 provides an interface among MEC-hosted controller application 302, DNS control service 309, and breakout control service 311. CB 320 may provide an interface between breakout control service 311 and the breakout control functions such as breakout control 312, 313, and 314. CB 320 may provide an interface to breakout control 314 using IFOM or connection manager feature.

[0105] The example system 300 of FIG. 3 also includes parent edge cloud control service 335 and edge cloud control service 319. Parent edge cloud control service 335 may be a core network entity managing allocation of resources in the edge clouds under its control. Parent edge cloud control service 335 may hold application images 323.

[0106] Edge cloud control service 319 may be an MEC service providing an API to MEC applications to control and receive information from edge clouds. ECC4 316, as referred to herein, may provide an interface between parent edge cloud control service 335 and edge cloud control service 319. CE 318, as referred to herein, may provide an interface between edge cloud control service 319 and control services in edge clouds 334. CE' 317 may be an interface between the edge cloud and the parent edge cloud control service. CE', as referred to herein, may provide an interface that used instead of, or in complement to, CE 318.

[0107] Local edge cloud 306 may include control services 329c, compute nodes 330c, virtual app instance 331c, and storage nodes 332c. In the example of FIG. 3, client device 327 may connect to local edge cloud 306 via a local connection, such as Ethernet, WiFi, etc. over 0 or more hops 321c. Further, local edge cloud 306 may be accessible from client device 327 through a different access technology than the one or more access technologies managed by the mobile network operator 324 (e.g., through a home or enterprise Wi-Fi access point (AP)). Home computing devices, such as home gateways, processors collocated with APs, or game consoles, may be used. Client traffic may reach local edge cloud 306 through IP routing. Techniques such as IFOM may also be used, although it may not be necessary to move flows between interfaces in general.

[0108] First-hop edge cloud 307 may include control services 329b, compute nodes 330b, virtual app instance 331b, and storage nodes 332b. In the example of FIG. 3, client device 327 may connect to first-hop edge cloud 307 via a local connection, such as Ethernet, WiFi, etc. over 0 or more hops 321b. Further, first-hop edge cloud 307 may be accessible through the first hop access point (AP or eNodeB) 326 managed by the mobile network operator 324. Client traffic may reach first-hop edge clouds through a local gateway 322b located on access point (AP or eNodeB) 326 (e.g., following breakout control 313 such as LIPA feature defined for 3GPP-compliant networks).

[0109] Intermediate edge cloud 308 may include control services 329a, compute nodes 330a, virtual app instance 331a, and storage nodes 332a. In the example of FIG. 3, client device 327 may connect to intermediate edge cloud 308 via a local connection, such as Ethernet, WiFi, etc. over 0 or more hops 321a. Further, intermediate edge cloud 308 may be accessible through a deeper point of attachment to the mobile network (e.g., through local gateway 322a or S/P-GW). Client traffic may reach intermediate edge cloud 308 through local gateway 322a (e.g., following breakout control 312 such as SIPTO feature defined for 3GPP compliant networks, such as by using separate bearers for client-edge cloud traffic).

[0110] In alternative embodiments, the example system 300 of FIG. 3 may include a virtual cloud provider that may mediate this relationship and deploy MEC-hosted controller applications in the mobile network. Global cloud providers may, therefore, play the virtual cloud provider role and may provide a seamless experience for application providers.

[0111] A difference between the example system 200 of FIG. 2 and the example system 300 of FIG. 3 may be viewed in terms of ownership of the edge cloud control plane: the OTT cloud provider in the example system 200 of FIG. 2 and the mobile network operator in the example system 300 of FIG. 3.

[0112] The example system 200 of FIG. 2 may provide flexibility because it may enable multiple third party cloud providers to deploy and manage a growing

number of edge clouds to adjust to the needs of end users. Moreover, in the embodiment of FIG. 2, the edge clouds may be easily shared between clients connected through different access networks.

[0113] The example system 300 of FIG. 3 may be chosen, for example, by mobile network operators that wish to retain tight control over edge clouds and are willing to incur the associated costs.

[0114] A MEC-hosted controller application, as described above with respect to the example system 200 of FIG. 2 and the example system 300 of FIG. 3, that is hosted on a MEC server may access a DNS control API, enabling the application to be involved in how the DNS server responds to client requests. For example, the application may own a domain (e.g., application.com) and may delegate a sub-domain (e.g., edge.application.com) to a cloud provider. The cloud provider may further delegate it to mobile operators' networks where the cloud provider's MEC-hosted controller application is deployed. The DNS servers located in the mobile network may be configured by the cloud provider-controlled MEC-hosted controller applications using, for example, a DNS control service API.

[0115] Two types of DNS control APIs may be used. In an embodiment, the MEC-hosted controller application may have a programmatic interface to a local DNS proxy server and may use it to configure in advance, or to dynamically be called to handle, DNS requests. In another embodiment, the MEC-hosted controller application may act as a DNS server. In this embodiment, the DNS controller API may be used to configure the local DNS proxy server to re-direct certain requests toward the controller's DNS server.

[0116] The APIs and configuration items described in the tables below may be used in any of the embodiments described herein.

[0117] Table 1 describes an example API for DNS control service, for use by the MEC-hosted controller application. Parts of this API may be for a proactive configuration of the DNS, while others may be for dynamic involvement of the

controller application in the DNS request itself. Examples of both types of procedures are described in detail below.

TABLE 1

DNS Control Service API	Description
AssociateFQDN(fqdn, clients, servers) Caller: Application	Request that any future DNS request for the given fqdn and originated from the specified clients be replied with the specified servers. fqdn is the FQDN of the DNS requests that are targeted. Clients is a set of 1 or more IDs of clients, (e.g., can be 1 or more IP address blocks identifying the clients targeted by this association request). Servers is a set of 1 or more server IP addresses that may be used in the reply. If there are more than one, the DNS server may send them in random order in the replies, to ensure load balancing between servers.
RegisterForDNSRequestEvent(fqdn, clients, children=true, hold=true) Caller: Application	Request the DNS control service to send an event to the calling application, whenever a DNS request is received for this FQDN from the specified group of clients (and any child of this FQDN is the optional parameter children is true). Clients can be specified by IP blocks for example. If hold is true, the DNS server will wait for a response from the application before proceeding with sending a response to the requester. Note that in a variant RegisterForDNSRequestEvent could also have a callback function parameter, taking a DNSRequestEvent parameter.
Event: DNSRequestEvent(fqdn, requestId, client) Sender: DNS Control Service	Event data structure sent to an application registered with RegisterForDNSRequestEvent(). Fqdn is the FQDN present in the DNS request.

	RequestId is an ID which may be used later to refer to this particular DNS request. Client is the ID of the DNS client who originated the DNS request. For example it can be an IP address.
ProceedWithDNSResponse(requestId, servers) Caller: Application	In relation to a DNSRequestEvent event, this function requests the DNS Server to respond to the given request with the given set of servers. RequestId is the ID obtained from the event. Servers is a set of 1 or more server IP addresses to send back to the requesting client.

[0118] Table 2 describes MEC-hosted controller application configuration items. For example, the configuration items may be placed in an application profile in the home subscriber service (HSS) or other database in the mobile network and retrieved by the DNS control service when the application connects to the service.

TABLE 2

MEC-Hosted Controller Application Configuration Item	Description
Allowed FQDNs for events	List of FQDN (possibly using wildcards to match hierarchies) for which the MEC-hosted controller application may receive DNS-related events.
Allowed FQDNs for proactive config	List of FQDN (possibly using wildcards to match hierarchies) which the application may associate to servers.
Allowed FQDNs for dynamic config	List of FQDN (possibly using wildcards to match hierarchies) for which the MEC-hosted controller application may be involved in individual DNS request procedures (e.g. for those FQDNs the application can call RegisterForDNSRequestEvent() with the hold parameter set to true).
Allowed Client IP blocks	IP blocks of clients for which the MEC-hosted controller application is authorized to configure/participate in DNS.

Allowed Server IP blocks	IP blocks of servers that the MEC-hosted controller application is authorized to set in the DNS responses.
---------------------------------	--

[0119] Messages sent over the CD interface may vary based on implementation. For example, such messages may be similar to the DNS control service API functions described above.

[0120] The DNS server may support advanced features, such as communicating certain requests to an external program and waiting for a response from that program before responding to the client. Alternatively, the DNS server may be integrated with the DNS service running on the MEC server.

[0121] In an embodiment, a DNS server may be integrated with the MEC-hosted controller application. This may be recommended for a lower impact on existing systems. For example, since the application may embed a local DNS server, the MEC-hosted controller application may enable interactions with the existing DNS server to use the DNS protocol. The DNS control API described above may become internal to the MEC-hosted controller application, and the DNS control service may provide an additional API function to register the MEC-hosted controller application's DNS server with the mobile network's DNS system. Table 3 provides an example the additional API function. In this embodiment, the configuration for the MEC-hosted controller application may be as provided in Table 4.

TABLE 3

DNS Control Service API	Description
RegisterDNSService(fqdn, clients=null) Caller: Application	This ensures that the DNS server controlled by the MEC DNS Control service is configured to redirect DNS queries for the specified FQDN towards the MEC-hosted Controller application, which may embed a DNS server to be able to handle those requests. For example, assuming the DNS server controlled by the MEC DNS Control service is a recursive DNS server, this API function tells it that the calling MEC application is authoritative for this

	FQDN. If clients are passed as parameters, the MEC DNS server may redirect DNS requests from those clients.
--	---

TABLE 4

MEC-hosted controller application Configuration Item	Description
FQDNs	List of FQDN which the application may be considered authoritative for.
Allowed Client IP blocks	IP blocks of clients from which the controller application is authorized to get requests.

[0122] In an embodiment, a MEC application hosted on a MEC server may access a breakout control service, which may enable configuring user and/or in-network devices to direct data flows appropriately toward edge clouds. Table 5 provides an example of a breakout control service API.

TABLE 5

Breakout Control Service API	Description
ruleId = ConfigureBreakout(clients, servers) Caller: Application	This function ensures that flows from specified clients towards specified servers are enabled through the network. The breakout service may have enough knowledge of the network to determine which breakout function should be configured, and communicate the appropriate configuration to this function. Clients is a set of 1 or more client IP addresses (e.g., an IP block). Servers is a set of 1 or more server IP addresses (e.g., an IP block).
ReleaseBreakout(ruleId) Caller: Application	This function reverts a breakout configuration previously made using ConfigureBreakout()
costMap = GetCostMap(clients, servers) Caller: Application	This function retrieves a cost map, which associates a cost to the path between a client or set of clients and a server or set of servers. This function may rely on a pre-known cost map between access points and servers, as well as the

	location of the client(s). This function may additionally collect IP routing information from the client to know of any available edge cloud available locally to the WTRU (e.g., at home or in an enterprise). The caller needs to be aware of the servers available in the area (e.g., through configuration, from OTT controller, etc.)
SetLocalRoute(clients, routes) Caller: Application	This function update the routing table on a client or set of clients to add a route to an edge cloud (e.g., through a Wi-Fi interface while the default route is over the cellular network). The caller needs to be aware of routing information to local edge clouds. This information can for example be registered in OTT controller by local users such as enterprises or home owners.
Statistics From service to application	Traffic statistics can be provided by the service to the application, including: number of current connections towards a given server, number of bytes exchanged between clients and a server, etc. This statistics may be used locally by the MEC-hosted Controller application, and/or forwarded upstream towards the gateway and/or OTT controller, where it can be used for global edge cloud management.
Congestion information From service to application	In-network congestion information may be pre-computed by the Mobile Network Operator in relation to breakout traffic towards edge cloud IP blocks, in order to inform the application without leaking proprietary information on in-network topology. For example, internally the MNO may monitor one or more critical internal links in the breakout path towards an edge cloud mini data center, and then derive a congestion ratio by comparing with a maximum achievable throughput, and associate this congestion ratio with the IP blocks of the servers present in this mini data center. This information may be provided under the form of a cost map similar to what is defined in ALTO.

[0123] Some breakout commands may result in a network stack reconfiguration on the client device (e.g., for directing traffic to a local cloud, which may be, for example, located in devices at home or in the enterprise local area network (LAN), or deeper in the network, such as in the eNodeB). The breakout service may communicate with the control function of LIPA, SIPTO, IFOM and/or other breakout technologies. Available breakout technologies may include, for example, LIPA (e.g., as defined in 3GPP Release 10, where a local IP network may achieve IP connectivity with a WTRU through one or more H(e)NBs of the mobile operator), SIPTO (which may achieve breakout deeper in the network, such as at the radio network controller (RNC) site, IFOM (which may be based upon Dual Stack Mobile IPv6 (DSMIPv6) and may depend on the device being able to use two radios, such as Wi-Fi and cellular, simultaneously, as well as on support in the network, such as a home agent at the P-GW). In an embodiment, the MEC controller application may configure the TOF of the MEC server to redirect selected traffic to a breakout MEC service, which may forward the traffic to an edge cloud. Regular IP routing may be configured on the client, for example, to reach a local edge cloud.

[0124] In an embodiment, certain breakout command messages may be exchanged over the CB interface, including, for example, a command code (CREATE, RELEASE), one or more client IP addresses, one or more server IP addresses, and a timeout.

[0125] Statistics and network congestion information may be input data for cloud controllers. The breakout control service may either provide a programmatic API, or rely on a network service such as an ALTO service, to make this information available to authorized MEC applications. If used, the ALTO service itself may be collocated with the MEC server or maybe a global service in the core network or a third party service.

[0126] Table 6 describes main configuration items that may be defined for a particular MEC-hosted controller application in relation to the breakout control service and includes configurations for the mobile network.

TABLE 6

MEC-Hosted Controller Application Configuration Item	Description
Allowed Client IP blocks	IP blocks of clients for which the controller application is authorized to get cost map and set breakout.
Allowed Server IP blocks	IP blocks of servers that the controller application is authorized to get cost map for and set breakout towards.
Allowed Routes & Clients	Local routes that may be applied into clients, and client IP blocks which can be configured this way.

[0127] As listed above, breakout at the client may be enabled by mobility technologies, such as IFOM, or through regular IP routing. In the embodiment described herein, there may only be a need to statically allocate a flow to an interface, which may be solved using regular IP routing in some cases. IFOM may further enable moving the flow to pass through another interface, which may help implement mobility solutions but may not be required for supporting edge clouds. Moreover, edge cloud mobility may also be handled by the changing edge server, and, therefore, IFOM-based mobility may not be suitable for such flows. Accordingly, a simple IP routing-based solution may be desirable.

[0128] For example, in a case where a mobile device routes IP traffic in the range 192.168.1.0/24 over its Wi-Fi interface, a local edge cloud may include several hosts on this LAN. Upon a DNS request for local.myapp.com, the mobile device may receive a list of IP addresses including 192.168.1.100 and may, therefore, connect to it without requiring any type of active breakout function to be involved. A breakout function may be used to report the device IP routing table to the MEC-hosted controller application so that the application may check, in its list of available edge cloud servers, which ones may be reached without impacting the routing table.

[0129] Furthermore, additional local edge cloud servers may be available and located beyond a router present on the local LAN, for example 192.168.1.0/24. In this case, the MEC-hosted controller application may instruct

the breakout control service to add a route on the client device, for example, in order to reach 192.168.2.0/24, where the edge cloud hosts may be located.

[0130] A breakout MEC feature may be implemented in MEC servers, which may include a mechanism to forward traffic to/from a breakout connection, for example, to a local micro data center. This breakout MEC feature may be one of the possible breakout functions controlled by the breakout control service defined above and may be implemented as an extension of the existing TOF MEC service. A breakout MEC service API may be used by the breakout control MEC service of the breakout MEC feature, such as described in Table 7.

TABLE 7

TOF Breakout API extension	Description
ConfigureBreakout(flowSpec, breakoutPort) Caller: Application or Service (e.g., Breakout Control MEC Service)	This function ensures that a specified flow is forwarded over a given port of the MEC server. This effectively enables breakout connections to be created inside the MEC server. The flow specifications may be a 5 tuple (Protocol, IP source, port, IP destination, port) for example. The breakout port may refer to a physical or virtual port of the MEC server.

[0131] The MEC-hosted controller application may need user traffic statistics for certain flows. For example, the controller may monitor the number of flows toward servers of a given application to be aware of the potential need to use edge clouds. However, TOF monitoring may be used to supplement monitoring in breakout functions and may not be able to replace it in cases, such as first hop edge clouds, when breakout traffic between client to edge cloud servers may not pass through the MEC server. In an embodiment, the TOF API may be extended to enable monitoring of flows. Table 8 provides an example of a TOF monitoring API extension.

TABLE 8

TOF Monitoring API extension	Description
flowId = AddFlowForTrafficMonitoring(flowSpec) Caller: Application	This function may add a flow to be monitored by the MEC server. The flow specifications may be a 5 tuple (Protocol, IP source, port, IP destination, port) for example. It can return a flow ID which may be used to retrieve the traffic statistics for this flow later.
RemoveFlowFromTrafficMonitoring(flowSpec) Caller: Application	This function removed a flow to be monitored by the MEC server.
stats = GetTrafficMonitoringStats(flowId) Caller: Application	This function returns a buffer (or pointer to a buffer) holding counters such as number of bytes received and sent for a give flow.

[0132] To implement this feature, the TOF may be extended with a loopback feature where, instead of being directed toward a local MEC application, specified flows may pass through a filter that collects traffic statistics before forwarding them over the link.

[0133] RNIS may provide an API to radio information. In an embodiment, the controller application may use this API to collect related radio and traffic conditions at the access point.

[0134] The edge cloud control MEC service API may be similar to OpenStack or other cloud management APIs, for example including commands to create VMs or deploy service instances. It may also enable collection of resource usage statistics from the edge cloud.

[0135] The MEC-hosted controller application's first function may be to perform DNS request routing for the domains (e.g., FQDNs) under its control. This may include replying, or configuring the DNS service to reply, to client DNS requests. This may involve several aspects, including taking allocation decisions, creating service instances, and configuring breakout or client IP routing for data flows. Some of these aspects, such as creating service instances, may be

performed partly or completely through another network function, such as the controller gateway or an OTT controller.

[0136] Additionally, and in order to enable the controller gateway and/or an OTT controller to perform their functions, the MEC-hosted controller application may also collect and transmit information such as the number of DNS requests and user data traffic information. To perform its role, the MEC-hosted controller application may maintain a list of available edge cloud servers, along with reachability and access right information, such as provided in Table 9.

TABLE 9

Name & Type	IP addresses	ID & Reachability	Access Rights & Auth
Local Edge Cloud 1	192.168.2.11, 192.168.2.12, 192.168.2.13	Type = "Local IP Routing" with additional info: LAN: 192.168.2.0/24 Router R1 interfaces: 192.168.2.1/24, 192.168.1.1/24, 192.168.3.1/24 ID = identity allocated by cloud provider	Private Application whitelist/blacklist Clients IP address whitelist/blacklist
1st-Hop Edge Cloud 2	10.10.10.{10-99}	Type = "Breakout at eNodeB" ID = identity of local gateway collocated with eNodeB	Public Application whitelist/blacklist Clients IP address whitelist/blacklist

[0137] This list may be obtained from the OTT/parent/gateway cloud service and may have been originally obtained from the network operators, such as a mobile network operator (MNO) for in-network breakout edge clouds and a home/enterprise network administrator for home/enterprise local edge clouds. For private access rights, the end user subscriber profile hosted by the mobile network may include the identities of edge clouds that a particular end user may access. The MEC-hosted controller application may query the mobile network for this information as described further below.

[0138] The OTT edge cloud control service may be similar to current cloud provider services. This service may include cloud management for data centers and may be extended to manage edge clouds, such as micro data centers or even home or enterprise edge clouds. To illustrate this last example, home or enterprise users may register devices to be used as edge cloud resources by the cloud provider. The user may need to install a management program provided by the cloud provider on each edge cloud device. The user may also configure, for example through an OTT management portal, usage limitations for these devices, for example in terms of white list or black list of applications, application types, end user identities, and end user location areas. For example, the cloud provider may lower or waive management service fees if the user accepts to provide computing capacity to other users.

[0139] The OTT edge cloud control service may, therefore, control edge clouds and may especially instantiate application instances, retrieve load usage, etc. and, more generally, may orchestrate the usage of edge cloud resources. On the other side, the OTT edge cloud control service may also communicate with the MEC-hosted control applications (through a gateway) on the other side. This communication may involve providing instance location and load information to MEC-hosted control applications and obtain client and traffic information back from the MEC-hosted control applications. Communication with MEC-hosted control applications may be mediated by a gateway, which is described below.

[0140] The parent edge cloud control service may be similar to the OTT edge cloud control service, with the difference being that the parent edge cloud control service may be under the control of the mobile network operator and may not require a gateway to mediate communication with the edge controller (it can be considered as a combination of an OTT edge cloud control service and the gateway).

[0141] The role of the edge cloud control service gateway in the system illustrated in FIG. 2 may be to mediate communication between the OTT edge

cloud control service and the edge controller, for example, for security purposes, and to mediate communication between different edge cloud controllers.

[0142] For example, the edge cloud control service gateway may aggregate information. For example, it may include an ALTO server to convey instance usage information down to MEC-hosted controller applications to enable them to choose the most appropriate server or servers for a client. It may also aggregate traffic usage information collected from MEC-hosted controller applications and present the aggregated information to the OTT cloud controller. It may also enable federating the MEC-hosted controller applications and enable supporting client mobility by mediating transfer of client-related information from one MEC-hosted controller application to another.

[0143] The edge cloud control service gateway may be an MEC application, for example, running on a MEC server in the core network. This may make it possible for different cloud providers to develop and use this gateway application. It may also be possible for certain application providers to write and sell such a gateway to cloud providers that are unwilling to develop and maintain it. Alternatively, a gateway function may be standardized in order to offer more guarantees to the mobile network operator that the gateway function will not impair network operation.

[0144] In some embodiments, third party application developers may augment an existing application with edge cloud acceleration, for example, with the authorization of the application provider. The resulting application may be deployed and used in one or both of the systems illustrated in FIGs. 2 and 3.

[0145] For example, an FQDN may be allocated to a portion of the original application, REST API, which may be run in an edge cloud (e.g., local.myapp.com) so that DNS-based redirection may be used. A REST API is used in the example, but it could be another type of service API, such as SOAP or proprietary protocols. In this example, the means of redirection is DNS-based, so any form of IP-based API may be used and partitioned using different FQDNs.

[0146] The application provider may propose a client API and backend API. The client API may be used by application clients and may be, for example, served at myapp.com and edge.myapp.com. The backend API may be a separate API enabling building an edge cloud-ready application implementing the edge-enabled portion of the client API. The application provider may provide access to the backend API only to trusted parties.

[0147] For example, the backend API of, for example myapp.com, accessed at for example, backend.myapp.com, may provide access to video files for transfer, while the edge cloud-ready portion of the client API (for example, edge.myapp.com) may enable streaming videos. For example, an application developer specialized in edge clouds may develop a distributed caching application that on one side uses the backend.myapp.com API served by the cloud application and on the other side serves the edge.myapp.com API to local clients. The edge cloud application may be uploaded to an edge cloud service (e.g., implemented as defined above).

[0148] FIG. 4 is a flow diagram of an example process 400 for redirecting a content request to a server in an edge cloud, which may be performed by an MEC server running an MEC-hosted controller application and which may be used in combination with any of the embodiments described herein. As described herein, the MEC server may be hosted at a base station eNodeB, or other node in a network. While each step of the process 400 in FIG. 4 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other. Referring to FIG. 4, the MEC server may receive a content request (which may also be referred to herein as a service request) from a WTRU 401. The content request may be, for example, a DNS request. The MEC server may then determine at least one third party edge cloud server to allocate to the content request and if needed trigger the creation of an application service instance on the server 402 based on content in the content request. For example, the content request may be for a service from an application available at a website such as edge.app.com. The

MEC server may determine a plurality of third party edge cloud servers to allocate to the content request. The MEC server may then configure a breakout function to enable service flows between the WTRU and third party edge cloud server 403. The MEC server may then transmit a DNS response to enable application connectivity between the WTRU and third party edge cloud server 404.

[0149] FIG. 5 is a flow diagram of an example process 500 for redirecting a content request to a server in an edge cloud, which may be performed by a DNS server embedded in an MEC server running an MEC-hosted controller application and which may be used in combination with any of the embodiments described herein. While each step of the process 500 in FIG. 5 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other. Referring to FIG. 5, the DNS server may receive a content request (which may also be referred to herein as a service request) from a WTRU 501. The content request may be, for example, a DNS request. The DNS server may then redirect the content request to the MEC-hosted controller application 502 based on content in the content request. For example, the content request may be for a service from an application available at a website such as edge.app.com. The DNS server may then receive at least one IP address of a third party edge cloud server 503 to allocate to the content request. The DNS server may receive a plurality of IP addresses of a plurality of third party edge cloud servers to allocate to the content request. The DNS server may then transmit a DNS response with the at least one IP address of the third party edge cloud server to enable application connectivity between the WTRU and third party edge cloud server 504.

[0150] Content or service requests and responses, as referred to in any of the embodiments described herein, may be transported over the DNS protocol. Alternatively, content or service requests and responses may be transported over another protocol (e.g., over HTTP, HTTP/2 or the QUIC protocol). DNS Servers

and DNS infrastructure are referred to in for exemplary purposes, to refer to service location servers and infrastructure that enable locating services points of presence from service identifiers.

[0151] In some embodiments, DNS and breakout control may be controlled by the network operator. FIG. 6 is a flow diagram of an example MEC-hosted controller application access authorization process 600, which may be used in combination with any of the embodiments described herein. While each step of the process 600 in FIG. 6 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other. Referring to FIG. 6, MEC-hosted applications profile service 604 may provision MEC-hosted controller application profiles in profile service by the network operation or application provider under the control of the network operator 611. The MEC-hosted controller application profiles may be provisioned in a database maintained by the network operator, which is referred to herein as the MEC-hosted applications profile service (MAPS). MEC-hosted controller application 601 may be deployed on an MEC server 612 and then MEC-hosted controller application 601 may start 613. When the MEC-hosted controller application starts 613, it may connect to certain MEC services, including DNS control MEC service 602 and breakout control MEC services 603. Additional services may behave in a similar fashion such that, upon an application having a successful authentication, the service may obtain a service-specific application profile from the MAPS.

[0152] In the example of FIG. 6, MEC-hosted controller application 601 may make an initial connection (with an application ID and credentials) 614 with DNS control MEC service 602, which may then verify the credentials 615. DNS control MEC service 602 may send a request to get the application profile 616 to MEC-hosted applications profile service 604, which may respond by sending the application profile portion related to the DNS control service 617 to DNS control MEC service 602. DNS control MEC service 602 may then store the registration

in a local state associated with the connected application 618 and send an OK message 619 to MEC-hosted controller application 601.

[0153] MEC-hosted controller application 601 may make an initial connection (with an application ID and credentials) 620 with breakout control MEC services 603, which may then verify the credentials 621. Breakout control MEC services 603 may send a request to get the application profile 622 to MEC-hosted applications profile service 604, which may respond by sending the application profile portion related to breakout control service 623 to breakout control MEC services 603. Breakout control MEC services 603 may then store the registration in a local state associated with the connected application 624 and send a reply 625 (shown with an OK message) to MEC-hosted controller application 601.

[0154] MEC-hosted controller application 601 may then connect to additional MEC services 626. MEC-hosted controller application 601 then becomes operational 627.

[0155] FIG. 7 is a flow diagram of an example process 700 for instance management in a MEC-hosted controller application, which may be used in combination with any of the embodiments described herein. While each step of the process 700 in FIG. 7 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other. The content and service request routing procedures described herein may use application instances that are up and running before a client initiates the service by sending the content or service request (e.g., a DNS request). A client may first connect to the application and log in. The application session may be conducted over HTTP or any other protocol and may first resolve the application domain name (e.g., application.com). After logging in and navigating through the application, the client may then use an edge-enabled portion of the application (e.g., edge.application.com). This FQDN may be delegated through the DNS system by the application provider to a cloud

provider, which may itself redirect requests toward mobile networks where the cloud provider's MEC-hosted controller applications are deployed.

[0156] Creating a new service instance may take up to several minutes (e.g., starting a VM on a cloud server may take around a minute or longer, depending on the operating system and cloud). Accordingly, it may not be possible to start a new instance upon a DNS request in time for the DNS response. Accordingly, the MEC-hosted controller application may be configured to trigger the creation ahead of time, for example based on a trigger event, such as a DNS request for a particular FQDN. The application edge instance may then stay active for a given time and be removed after a certain time of inactivity (e.g., 1 hour).

[0157] It may be possible, therefore, that in some fringe cases a DNS request to the edge-enabled portion of the application (e.g., edge.application.com) may be received and no instance may be available in a nearby edge cloud. In the example, a user may log into the application, stay connected for an hour and then start streaming (assuming streaming is the edge-enabled feature served at the edge-enabled portion of the application (e.g., edge.application.com)). In this example, an edge instance may be started when the user starts the application and then torn down after a time of inactivity. Therefore, a second mechanism may be used (e.g., the trigger event to create the instance is the first DNS request for edge.application.com).

[0158] In this second mechanism, the MEC-hosted controller application may be configured to reply with a well-known IP address or alias (CNAME) to a fallback server. The fallback server may, for example, tell the client application to retry after a given time. In this case, the DNS response may have a short time out to ensure that when the application retries, a new DNS request will be sent by the client. The fallback server may be the same for many applications and may be provided by the cloud provider. It may, for example, be an HTTP server returning 503 Service Unavailable. Clients may be aware of the potential reason and retry the request later. Alternatively, the fallback server may be application

specific, in which case the application provider may implement any needed behavior. For example, the fallback server may deliver a low resolution video stream or other degraded application experience.

[0159] To enable these methods, the MEC-hosted controller application may use a DNS control API function for registering for DNS request events (e.g., RegisterForDNSRequestEvent defined above) and/or an OTT/gateway controller API or edge cloud control API function to create a new instance of an application in a specific edge cloud or in a set of edge clouds and/or receive usage information relative to existing edge cloud service instances.

[0160] The example process 700 of FIG. 7 may be used with both the above approaches, where a trigger FQDN may be set to application.com while the managed FQDN is edge.application.com (first case), or the trigger FQDN may be the same as the managed FQDN edge.application.com (second case).

[0161] Referring to FIG. 7, the MEC-hosted controller application may configure DNS control service for events for triggering one or more FQDNs and one or more edge-cloud managed FQDNs 701. The MEC-hosted controller application may then configure the DNS service to reply with one or more short lived IP addresses of fallback servers 702. The MEC-hosted controller application may then wait to receive a DNS Request event for any of those one or more FQDNs 703. The MEC-hosted controller application may then receive an event 704 and create at least one instance in an edge cloud (using an edge control service API or gateway/OTT service API) 705. The MEC-hosted controller application may then configured the DNS service to reply with one or more IP addresses of servers 706. The MEC-hosted controller application may then monitor edge cloud service usage using the OTT/Gateway Controller API or edge cloud control API 707.

[0162] If remaining instances were not used for a given time and there is no minimum number of instance specified in the application profile 710, the MEC-hosted controller application may configure the DNS service to reply with one or more short lived IP addresses of the fallback server and remove remaining

instances 711. The MEC-hosted controller application may then wait to receive a DNS Request event for any of those one or more FQDNs 703.

[0163] If average/peak instances load is too low 709, then the MEC-hosted controller application may remove some instances unless they are at a minimum specified in the application profile 712. The MEC-hosted controller application may then monitor edge cloud service usage using the OTT/Gateway Controller API or edge cloud control API 707.

[0164] If average/peak instances load is too high 708, then the MEC-hosted controller application may add some instances unless they are at a maximum specified in the application profile 713. The MEC-hosted controller application may then monitor edge cloud service usage using the OTT/Gateway Controller API or edge cloud control API 707.

[0165] FIG. 8 is a flow diagram of an example process 800 for in-path request routing by a MEC-hosted controller application, which may be used in combination with any of the embodiments described herein. While each step of the process 800 in FIG. 8 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other.

[0166] In the example of FIG. 8, the MEC-hosted controller application 804 may register to receive DNS events from the DNS control MEC service by sending, for example, `RegisterForDNSRequestEvent(fqdn=edge.app.com,clients,hold=true)` 811 to DNS server and DNS control MEC service 805. The hold parameter may be set to true to indicate that the DNS control service will wait for a response to the event from MEC-hosted controller application 804 before proceeding with sending the response. This may ensure that MEC-hosted controller application 804 has the opportunity to set one or more server IP addresses to send back in the DNS response.

[0167] DNS server and DNS control MEC service 805 may check that this application has the right to perform such calls by, for example, checking the

rights in the application profile 812 that was obtained in accordance with any of the methods described herein. If the application has the appropriate rights, the DNS server and DNS control MEC service 805 may store the registration information in a local state 813 and reply 814 (shown with an OK message) to MEC-hosted controller application 804. Depending on implementation details, the DNS control service may or may not need to communicate this registration information to the DNS server (not shown in the example of FIG. 8). In one example, the DNS server may communicate requests to the DNS control service (e.g., this may make sense when they are collocated), and, therefore, there may be no need for such a message. In other examples, such as when they are not collocated, it may make sense for the DNS control service to indicate the registration information to the DNS server, which may send events to the DNS control service for DNS requests pertaining to FQDNs that have a registration.

[0168] Client 801 (which may be a WTRU as defined herein) receiving service from an application may use an edge cloud-enabled feature. In this example, client 801 may be connected to an application (e.g., app.com) and may decide to get a certain service from the application such as streaming, which is implemented as an edge-enabled service in the subdomain (e.g., edge.app.com) 815. Client 801 may then send DNS request for FQDN edge-enabled service in the subdomain (e.g., edge.app.com) 816 to DNS server and DNS control MEC service 805. When this request is received, the DNS server may forward it to the DNS control service. DNS server and DNS control MEC service 805 may check whether the application is registered for dynamic DNS for this client and FQDN 817. Based on the registration record, DNS server and DNS control MEC service 805 may send the event 818 to MEC-hosted controller application 804, which in the example of FIG. 8 is shown as calling registered callback with eventDNSRequest(client, FQDN).

[0169] MEC-hosted controller application 804 may then determine one or more edge cloud servers applicable, if any, by in an example querying location/RNIS/other APIs in the process and if needed trigger creation of an

application instance in edge cloud 819. MEC-hosted controller application 804 may use available information to determine the one or more servers to allocate to the request. The event parameters (FQDN, client IP) along with information including but not limited to: live per-application instance location and load information (e.g., obtained from the OTT controller via the gateway and/or an ALTO server), client attachment point and related radio network information (e.g., RNIS MEC service), and mobile network congestion information and/or cost map for links between client IP blocks and server IP blocks (e.g., from the breakout API and/or ALTO and/or OTT controller).

[0170] Once the one or more servers is selected, MEC-hosted controller application 804 may configure breakout flows from the client 801 to the selected one or more servers 820. A breakout function may then be configured 821 (breakout control MEC services 806 may control a gateway element and/or client 801 to enable the breakout). Breakout control MEC services 806 may use its knowledge of the mobile network topology to determine which actual breakout function should be configured (e.g., here, the local gateway is collocated with the eNodeB), perform this configuration, and then return a response to the caller.

[0171] On-demand breakouts may be reverted to at some point to, for example, avoid filling routing tables. A solution may be to use a timeout at a breakout control MEC services 806 itself (e.g., a local gateway, a connection manager, etc.). Entries in a table may be associated with a timeout after which a local breakout function may remove it. For example, for breakout performed by a connection manager at client 801, the connection manager may be informed by the breakout command message of the validity duration of the local breakout and then remove the entry after this time elapsed. To handle cases where breakout control MEC services 806 does not support a timeout, breakout control MEC services 806 may issue a breakout REMOVE command after the timeout elapse. The timeout is longer than the DNS response timeout for edge cloud servers, in order to ensure that client 801 will need to re-send a DNS request before connecting a same edge server again.

[0172] Breakout control MEC services 806 may then reply 822 (shown with an OK message) to MEC-hosted controller application 804. MEC-hosted controller application 804 may reply to the DNS event by sending one or more server IP addresses 823 to DNS server and DNS control MEC service 805. DNS server and DNS control MEC service 805 may send a DNS response with one or more server IP addresses 824 to client 801. Client 801 may make an initial connection via RAN element and gateway 802 to the service 825 with edge cloud server 803 and flows break out to the edge cloud 826. Application specific service processing 827 may be performed.

[0173] FIG. 9 is a flow diagram of an example process 900 for in-path request routing by a MEC-hosted controller application with an embedded DNS, which may be used in combination with any of the embodiments described herein. Embedding a DNS server (responsible for handling requests for managed and possibly trigger FQDNs passing through the MEC server) is an efficient design choice that may be used in any of the embodiments described herein. While each step of the process 900 in FIG. 9 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other. In the example of FIG. 9, the MEC-hosted controller application 904 may register for DNS service from the DNS control MEC service by sending, for example, RegisterDNSService (fqdn=edge.app.com,clients) 911 to DNS server and DNS control MEC service 905. DNS server and DNS control MEC service 905 may check that this application has the right to perform such calls by, for example, checking the rights in the application profile 912 that was obtained in accordance with any of the methods described herein. If the application has the appropriate rights, the DNS server and DNS control MEC service 905 may configure the DNS server to redirect requests 913 and reply 914 (shown with an OK message) to MEC-hosted controller application 904.

[0174] Client 901 (which may be a WTRU as defined herein) receiving service from an application may use an edge cloud-enabled feature. In this

example, client 901 may be connected to an application (e.g., app.com) and may decide to get a certain service from the application such as streaming, which is implemented as an edge-enabled service in the subdomain (e.g., edge.app.com) 915. Client 901 may then send DNS request for FQDN the edge-enabled service in the subdomain (e.g., edge.app.com) 916 to DNS server and DNS control MEC service 905. DNS server and DNS control MEC service 905 may then redirect this DNS request for FQDN edge.app.com 917 to MEC-hosted controller application 904

[0175] MEC-hosted controller application 904 may then determine the one or more edge cloud servers applicable, if any, by for example querying location/RNIS/other APIs in the process 918. MEC-hosted controller application 904 may use available information to determine the one or more servers to allocate to the request. The event parameters (FQDN, client IP) along with information including but not limited to: live per-application instance location and load information (e.g., obtained from the OTT controller via the gateway and/or an ALTO server), client attachment point and related radio network information (e.g., RNIS MEC service), and mobile network congestion information and/or cost map for links between client IP blocks and server IP blocks (e.g., from the breakout API and/or ALTO and/or OTT controller).

[0176] Once the one or more servers is selected, MEC-hosted controller application 904 may configure breakout flows from the client 901 to the selected one or more servers 919. A breakout function may then be configured 920 (breakout control MEC services 906 may control a gateway element and/or client 901 to enable the breakout). Breakout control MEC services 906 may use its knowledge of the mobile network topology to determine which actual breakout function should be configured (e.g., here, the local gateway is collocated with the eNodeB), perform this configuration, and then return a response to the caller.

[0177] On-demand breakouts may be reverted to at some point to, for example, avoid filling routing tables. In an embodiment, a timeout is used at breakout control MEC services 906 (e.g., local gateway, connection manager,

etc.). Entries in a table may be associated with a timeout after which a local breakout function may remove it. For example, for breakout performed by a connection manager at client 901, the connection manager may be informed by the breakout command message from breakout control MEC services 906 of the validity duration of the local breakout and then remove the entry after this time elapsed. To handle cases where breakout control MEC services 906 does not support a timeout, breakout control MEC services 906 may issue a breakout REMOVE command after the timeout elapse. The timeout is longer than the DNS response timeout for edge cloud servers, in order to ensure that client 901 will need to re-send a DNS request before connecting a same edge server again.

[0178] Breakout control MEC services 906 may then reply 921 (shown with an OK message) to MEC-hosted controller application 904. MEC-hosted controller application 904 may reply to the DNS event by sending the one or more server IP addresses 922 to DNS server and DNS control MEC service 905. DNS server and DNS control MEC service 905 may send a DNS response with one or more server IP addresses 923 to client 901. Client 901 may make an initial connection via RAN element and gateway 902 to the service 924 with edge cloud server 903 and flows break out to the edge cloud 925. Application specific service processing 926 may be performed.

[0179] FIG. 10 is a flow diagram of an example process 1000 for in-path request routing by a MEC-hosted controller application with the breakout and DNS configured in advance, which may be used in combination with any of the embodiments described herein. While each step of the process 1000 in FIG. 10 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other. MEC-hosted controller application 1004 may in advance configure breakout for enabling client 1001 IP blocks to get access to edge clouds available through break out at eNodeBs, MEC servers, and other local gateways. As a result, there may be no need to dynamically configure breakout on demand. The

procedures described above may be used with the additional breakout configuration portion.

[0180] MEC-hosted controller application 1004 may also configure the DNS service in advance to ensure that certain IP blocks are directed toward a given set of servers for a given FQDN. For example, if clients whose DNS traffic is passing through a particular DNS proxy server are to be served by 1 or more servers in an edge cloud, then the FQDN edge.example.com may be associated with these servers' IP addresses in this DNS proxy server. DNS proxy servers may return associated IP addresses in the response, in rotating order for each subsequent DNS request/response, which may ensure load balancing between the servers.

[0181] In the example of FIG. 10, MEC-hosted controller application 1004 may learn about the managed edge cloud micro data center and servers from, for example, OTT/parent/gateway/ALTO servers 1011. MEC-hosted controller application 1004 may configure breakout flows from IP blocks of possible clients to IP blocks of edge servers, which may result in several calls to ConfigureBreakout 1012.

[0182] A breakout function may then be configured 1013 (breakout control MEC services 1006 may control a gateway element and/or client 1001 to enable the breakout). Breakout control MEC service 1006 may send a reply 1014 (shown with an OK message) to MEC-hosted controller application 1004. MEC-hosted controller application 1004 may learn about available application instances from, for example, OTT/gateway and/or request creation of one or more application instances in edge clouds based on OTT/Gateway 1015. MEC-hosted controller application 1004 may collect a list of available application instances' IP addresses from OTT/gateway 1016. MEC-hosted controller application 1004 may send the associateFQDN(fqdn=edge.app.com, clients,servers) 1017. DNS server and DNS control MEC service 1005 may check that this application has the right to perform such calls by, for example, checking the rights in the application profile 1018 that was obtained in accordance with any of the methods described herein.

[0183] If the application has the appropriate rights, the DNS server and DNS control MEC service 1005 may configure the DNS server 1019 and reply 1020 (shown with an OK message) to MEC-hosted controller application 1004. Client 1001 (which may be a WTRU as defined herein) may be connected to app.com and may decide to get a certain service from the application such as streaming, which is implemented as an edge-enabled service in the subdomain edge.app.com 1021. Client 1001 may then send DNS request for FQDN edge.app.com 1022 to DNS server and DNS control MEC service 1005. DNS server and DNS control MEC service 1005 may send a DNS response with one or more server IP addresses 1023 to client 1001. Client 1001 may make an initial connection via RAN element and gateway 1002 to the service 1024 with edge cloud server 1003 and flows break out to the edge cloud 1025. Application specific service processing 1026 may be performed.

[0184] FIG. 11 is a flow diagram of an example process 1100 for in-path request routing by a MEC-hosted controller application with local breakout on a client using a dynamic method of IP routing, which may be used in combination with any of the embodiments described herein. While each step of the process 1100 in FIG. 11 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other. In this example of a request routing procedure, edge cloud servers 1103 may be home or enterprise devices, such as game consoles, PCs or laptops.

[0185] In the example of FIG. 11, an end user may register home or enterprise devices with OTT/parent/cloud provider/gateway 1107 as edge cloud servers and install agents on them to put them under control of the cloud provider, and the end user may configure authorized clients and applications 1108. Certain devices may have pre-installed agents at purchase time to simplify the process. Some application instances may be started on one or more user-provided edge servers based on the configuration of the user 1109.

[0186] MEC-hosted controller application 1104 may then learn about user provided edge servers from OTT/parent/gateway/ALTO and learn as about any available application instances and/or requests creation of application instances 1110. It may also request for some instance to be created, for example upon a trigger event as defined above. MEC-hosted controller application 1104 may then configure DNS by either registering for DNS events (for the in-path request routing variant) or pre-configuring the DNS server as described above 1111. The in-path request routing variant may be used if there is a need to dynamically configure IP routing on the client, as described further below.

[0187] Client 1101 (which may be a WTRU as defined herein) may be connected to app.com and may decide to get a certain service from the application such as streaming, which is implemented as an edge-enabled service in the subdomain edge.app.com 1112. Client 1101 may then send via RAN 1102 a DNS request for FQDN edge.app.com 1113 to DNS server and DNS control MEC service 1105, which may forward the request 1114 to MEC-hosted controller application 1104. MEC-hosted controller application 1104 may determine that local edge clouds may be available 1115. MEC-hosted controller application 1104 may query 1116 breakout control MEC service 1106 for initial information or updated information regarding IP routing information on client 1101. Breakout control MEC service 1106 may query 1117 client 1101 for initial information or updated information regarding IP routing information. Client 1101 may then send the IP routing information 1118 to breakout control MEC service 1106, which may forward it 1119 to MEC-hosted controller application 1104.

[0188] When the IP routing information is available, MEC-hosted controller application 1104 may determine 1120 which service instance and edge cloud server 1103 to select for client 1101. Edge cloud server 1103 may be accessible using the existing IP routing table in the client device, or it may alternatively use an additional route. MEC-hosted controller application 1104 may configure breakout for flows 1121 from client 1101 to edge cloud server 1103. Breakout control MEC service 1106 may send a request 1122 to a connection

manager to add an IP route to edge server 1103, which may reply 1123 (shown with an OK message) to breakout control MEC service 1106. Breakout control MEC service 1106 may then reply 1124 (shown with an OK message) to MEC-hosted controller application 1104, which replies 1125 (shown with an OK message) to DNS server and DNS control MEC service 1105. Alternatively, IP routing on client 1101 may be preconfigured to ensure that edge cloud server 1103 is reachable through the Wi-Fi interface, for example. This alternate pre-configuration may be provided as rules through the access network discovery and selection function (ANDSF) function, for example.

[0189] DNS server and DNS control MEC service 1105 may send a DNS response with one or more server IP addresses 1126 to client 1101. Edge cloud server 1103 IP address is accessible 1128 from client 1101 over a WiFi interface, and client 1101 may make an initial connection to the service 1127 with edge cloud server 1103. Application specific service processing 1129 may be performed.

[0190] In some embodiments, local edge cloud information may be entered into a policy server, such as the ANDSF server. The information may have been provided to the cloud service provider first, which then transmits this information to the mobile operator for configuring the ANDSF server and possibly also to other databases. In some examples, the information may be directly provided by the user to the mobile network operator, such as through the user portal.

[0191] The ANDSF server may be used to distribute local edge routing information both to MEC-hosted controller application 1204 and to client 1201 (which may be a WTRU as defined herein). There may be, therefore, no need for direct communication to take place at DNS request time, which may be more efficient. On the other side, the ANDSF-based alternative may have a higher impact in terms of administration and configuration.

[0192] FIG. 12 is a flow diagram of an example process 1200 for in-path request routing by a MEC-hosted controller application with local breakout on a client using an ANDSF-based IP routing, which may be used in combination with

any of the embodiments described herein. While each step of the process 1200 in FIG. 12 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other. In this example of a request routing procedure, edge cloud servers 1203 may be home or enterprise devices, such as game consoles, PCs or laptops.

[0193] In the example of FIG. 12, an end user may register home or enterprise devices with OTT/parent/cloud provider/gateway 1207 as edge cloud servers and install agents on them to put them under control of the cloud provider, and the end user may configure authorized clients and applications 1210. Certain devices may have pre-installed agents at purchase time to simplify the process. The mobile network's (MN's) ANDSF and/or other MN database 1206 may be provisioned 1211 with information from OTT/parent/cloud provider/gateway 1207 (local edge cloud IP blocks, IP routing reachability). Some application instances may be started on one or more user-provided edge servers based on the configuration of the user 1212.

[0194] MEC-hosted controller application 1204 may then learn about user provided edge servers from ANDSF and/or other MN database 1206 and may add them to its local list of available edge cloud servers 1213. MEC-hosted controller application 1204 may learn about available application instances and/or requests creation of application instances 1214 from OTT/parent/cloud provider/gateway 1207. MEC-hosted controller application 1204 may then configure DNS either by registering for DNS events (for the in-path request routing variant) or by pre-configuring the DNS server as described above 1215. The in-path request routing variant may be used if there is a need to dynamically configure IP routing on the client, as described further below. Client 1201 may obtain a policy or policy information from ANDSF and may configure its IP routing table to be able to reach edge clouds provided in the policy 1216.

[0195] Client 1201 may be connected to app.com and may decide to get a certain service from the application such as streaming, which is implemented as

an edge-enabled service in the subdomain edge.app.com 1217. Client 1201 may then send via RAN 1202 a DNS request for FQDN edge.app.com 1218 to DNS server and DNS control MEC service 1205. MEC-hosted controller application 1204 may be involved dynamically or not depending on the DNS configuration method chosen 1219.

[0196] DNS server and DNS control MEC service 1205 may send a DNS response with one or more server IP addresses 1220 to client 1201. Edge cloud server 1203 IP address is accessible 1222 from client 1201 over a WiFi interface, and client 1201 may make an initial connection to the service 1221 with edge cloud server 1203. Application specific service processing 1223 may be performed.

[0197] FIG. 13 is a diagram of an example schema 1300 for an edge cloud managed ANDSF management object, which may be used in combination with any of the embodiments described herein. In the example of FIG. 13, rectangles denote XML nodes, regular text denotes leaves, and <X>+ is a special node placeholder for multiple nodes. Referring to FIG. 13, <X>+ 1302 may indicate that there may be 0 or more EdgeCloud elements defined in the document. EdgeCloud element 1301 may contain routing information that may make it possible for the client device to set up its routing tables to reach any edge cloud listed in the policy as long as a router to this edge cloud is reachable through the current IP routing table. The example of FIG. 13 shows EdgeCloudId 1303, RoutingInfo 1304, <X>+ 1305, IP Block 1306, Router 1307, <X>+ 1308, IP address 1309, and RulePriority 1310.

[0198] In both of the previous local breakout scenarios, the client may alternatively discover edge clouds through router advertisements or other existing service discovery mechanisms. For example, the OTT cloud operator may expose a discovery REST API, which the client device's connection manager may use to discover nearby edge clouds. The same REST API may be used by a MEC-hosted controller application for initial discovery of available resources.

[0199] FIG. 14 is a flow diagram of an example process 1400 for processing of edge cloud access rights, which may be used in combination with any of the

embodiments described herein. While each step of the process 1400 in FIG. 14 is shown and described separately, multiple steps may be executed in a different order than what is shown, in parallel with each other, or concurrently with each other.

[0200] To grant access to resources on private edge clouds, the MEC-hosted controller application may need to be able to retrieve an identity associated with a client and then query the mobile network for edge cloud access rights for this client. Each user profile in HSS/HLR may hold service specific information where the service may be the particular cloud device. In one embodiment, an authentication MEC service (e.g., a service operated by the MNO and offering an API to MEC-hosted applications) may be queried for service specific authorization information. The MEC-hosted controller application may use this service to retrieve whitelists/blacklists of edge clouds for specific clients, for example based on their IP addresses. The service may first retrieve the identity of the subscriber from the system (e.g., from an S-GW) based on the IP address. Then the service may query subscriber information from the HSS/HLR and reply to the MEC-hosted controller application with the desired information extracted from the user profile.

[0201] Referring to FIG. 14, MEC-hosted controller application 1404 may perform DNS setup or registration 1410. MEC-hosted controller application may register 1411 with an authentication service and provide the cloud service ID of the operator of the MEC-hosted controller application 1404. HSS/HLR 1408 may have user profiles that have, or are associated with, service specific sections associated with a cloud service identified with a “cloud service ID”. Such a section is filled, for example, based on input from the cloud provider, for example, with whitelist/blacklist of edge clouds for each user 1412.

[0202] Client 1401 (which may be a WTRU as defined herein) receiving service from an application may use an edge cloud-enabled feature. In this example, client 1401 may be connected to app.com and may decide to get a certain service from the application such as streaming, which is implemented as

an edge-enabled service in the subdomain edge.app.com 1413. Client 1401 may then send, via RAN element and gateway 1402, DNS request for FQDN edge.app.com 1414 to DNS server and DNS control MEC service 1405. DNS server and DNS control MEC service 1405 may forward the request 1415 to MEC-hosted controller application 1404. MEC-hosted controller application 1404 may send an edge cloud user profile request (client IP) 1416 to authorization MEC service 1407, which may then retrieve the ID related to the WTRU IP address 1417 from S-GW or other entity holding IP/ID mapping 1409. S-GW or other entity holding IP/ID mapping 1409 may then retrieve the user profile service-specific section attached to the “cloud service ID” using the ID 1418 which is sent 1418 to authorization MEC service 1407 and forwarded to HSS/HLR 1408. Authorization MEC service 1407 may then send a blacklist or whitelist of edge clouds 1421.

[0203] MEC-hosted controller application 1404 may then determine the one or more edge cloud servers applicable, if any, by in an example querying location/RNIS/other APIs in the process 1422. MEC-hosted controller application 1404 may use available information to determine the one or more servers to allocate to the request. The event parameters (FQDN, client IP) along with information including but not limited to: live per-application instance location and load information (e.g., obtained from the OTT controller via the gateway and/or an ALTO server), client attachment point and related radio network information (e.g., RNIS MEC service), and mobile network congestion information and/or cost map for links between client IP blocks and server IP blocks (e.g., from the breakout service API and/or from an ALTO server and/or from an OTT service).

[0204] Once one or more servers is selected, MEC-hosted controller application 1404 may configure breakout flows from the client 1401 to the selected one or more servers 1423. A breakout function may then be configured 1424 (breakout control MEC services 1406 may control a gateway element and/or client 1401 to enable the breakout). Breakout control MEC services 1406 may use

its knowledge of the mobile network topology to determine which actual breakout function should be configured (e.g., a local gateway collocated with an eNodeB), perform this configuration, and then return a response to the caller.

[0205] On-demand breakouts may later be removed in order to, for example, avoid filling routing tables. A solution may be to implement a timeout-based mechanism inside a breakout control MEC service 1406 (e.g., local gateway, connection manager, etc.). Entries may be associated with a timeout after which a local breakout function may remove it. For example, for breakout performed by a connection manager at client 1401, the connection manager may be informed by the breakout command message of the validity duration of the local breakout and then remove the entry after this time elapsed. To handle cases where breakout control MEC services 1406 does not support a timeout, breakout control MEC services 1406 may issue a breakout REMOVE command after the timeout elapsed. The timeout is longer than the DNS response timeout for edge cloud servers, in order to ensure that client 1401 will need to re-send a DNS request before connecting a same edge server again.

[0206] Breakout control MEC services 1406 may then reply 1425 (shown with an OK message) to MEC-hosted controller application 1404. MEC-hosted controller application 1404 may reply to the DNS event by sending the one or more server IP addresses 1426 to DNS server and DNS control MEC service 1405. DNS server and DNS control MEC service 1405 may send a DNS response with one or more server IP addresses 1427 to client 1401. Client 1401 may make an initial connection to the service 1428 with edge cloud server 1403 and flows break out to the edge cloud 1429. Application specific service processing 1430 may be performed.

[0207] Although features and elements are described above in particular combinations, one of ordinary skill in the art will appreciate that each feature or element can be used alone or in any combination with the other features and elements. In addition, the methods described herein may be implemented in a computer program, software, or firmware incorporated in a computer-readable

medium for execution by a computer or processor. Examples of computer-readable media include electronic signals (transmitted over wired or wireless connections) and computer-readable storage media. Examples of computer-readable storage media include, but are not limited to, a read only memory (ROM), a random access memory (RAM), a register, cache memory, semiconductor memory devices, magnetic media such as internal hard disks and removable disks, magneto-optical media, and optical media such as CD-ROM disks, and digital versatile disks (DVDs). A processor in association with software may be used to implement a radio frequency transceiver for use in a WTRU, UE, terminal, base station, RNC, or any host computer.

*

*

*

CLAIMS

What is claimed is:

1. A method for use in a server, the method comprising:
receiving, by the server, a content request from a wireless transmit/receive unit (WTRU);
determining, by the server, at least one third party edge cloud server to allocate to the content request based on content in the content request;
configuring, by the server, a breakout function to enable service flows between the WTRU and third party edge cloud server; and
transmitting, by the server, a DNS response to enable application connectivity between the WTRU and third party edge cloud server.
2. The method of claim 1, wherein the content request is a DNS request.
3. The method of claim 1, wherein a DNS server is embedded in the server.
4. The method of claim 1, wherein the content request is for a service from an application available at a website.
5. The method of claim 1, wherein the breakout function is preconfigured.
6. The method of claim 1, wherein the at least one third party edge cloud server is determined using policy information in an access network discovery and selection function (ANDSF) server.
7. The method of claim 1, further comprising:

triggering, by the server, creation of an application instance.

8. A server, the server comprising:

a receiver configured to receive a content request from a wireless transmit/receive unit (WTRU);

a processor configured to determine at least one third party edge cloud server to allocate to the content request based on content in the content request;

the processor further configured to configure a breakout function to enable service flows between the WTRU and third party edge cloud server; and

a transmitter configured to transmit a DNS response to enable application connectivity between the WTRU and third party edge cloud server.

9. The server of claim 8, wherein the content request is a DNS request.

10. The server of claim 8, wherein a DNS server is embedded in the server.

11. The server of claim 8, wherein the content request is for a service from an application available at a website.

12. The server of claim 8, wherein the breakout function is preconfigured.

13. The server of claim 8, wherein the at least one third party edge cloud server is determined using policy information in an access network discovery and selection function (ANDSF) server.

14. The server of claim 8, further comprising:

the processor further configured to trigger creation of an application instance.

15. A method for use in a server, the method comprising:
receiving a content request from a wireless transmit/receive unit (WTRU);
redirecting the content request to a mobile edge computing (MEC) – hosted controller application;
determining, via the MEC-hosted controller application, at least one third party edge cloud server to allocate to the content request based on content in the content request;
configuring, via the MEC-hosted controller application, a breakout function to enable service flows between the WTRU and third party edge cloud server; and
transmitting, by the server, a DNS response to enable application connectivity between the WTRU and third party edge cloud server.
16. The method of claim 15, wherein the content request is a DNS request.
17. The method of claim 15, wherein a DNS server is embedded in the server.
18. The method of claim 15, wherein the content request is for a service from an application available at a website.
19. The method of claim 15, wherein the breakout function is preconfigured.
20. The method of claim 15, wherein the at least one third party edge cloud server is determined using policy information in an access network discovery and selection function (ANDSF) server.

21. The method of claim 15, further comprising:
triggering, via the MEC-hosted controller application, creation of an application instance.

100

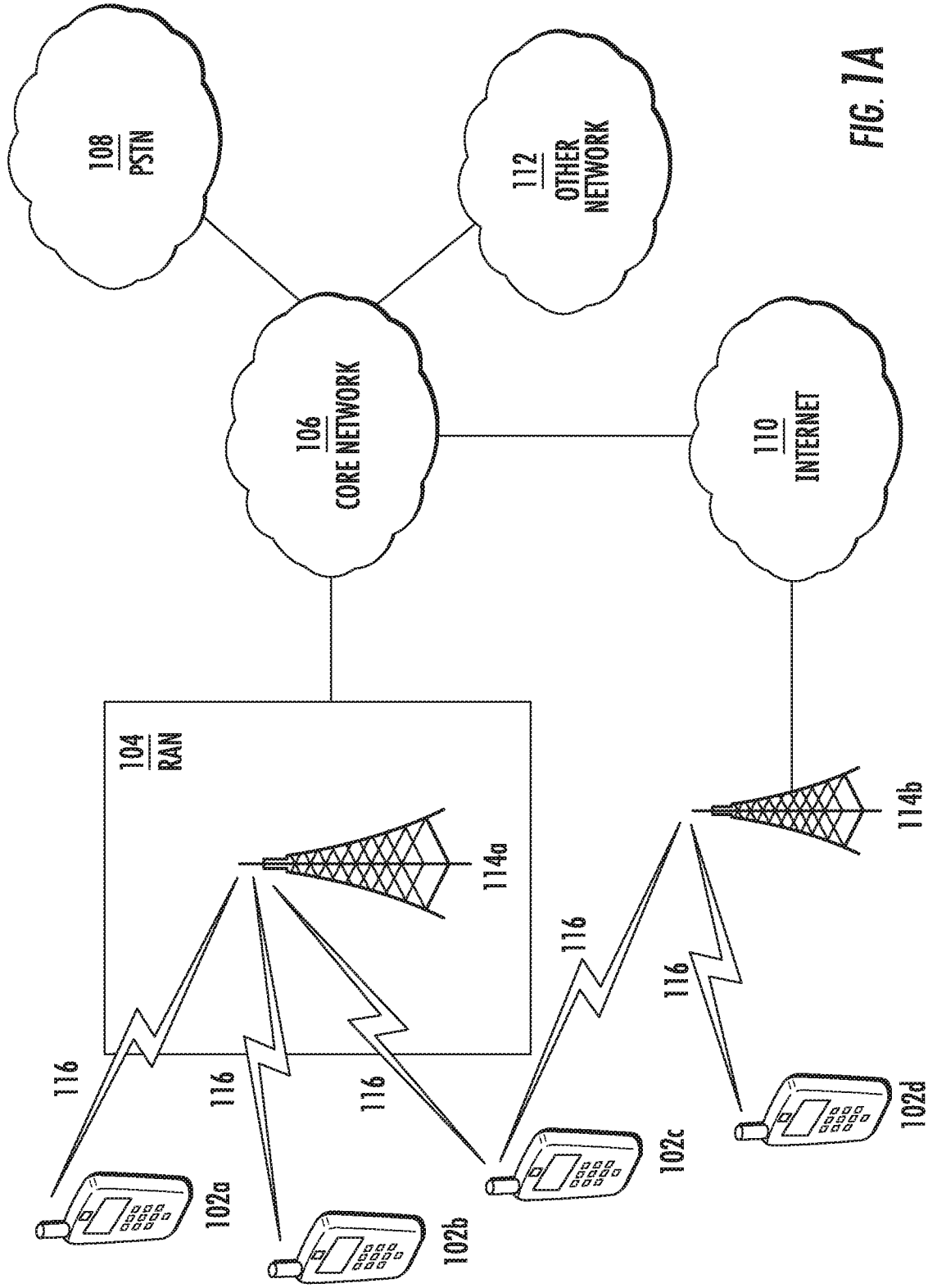
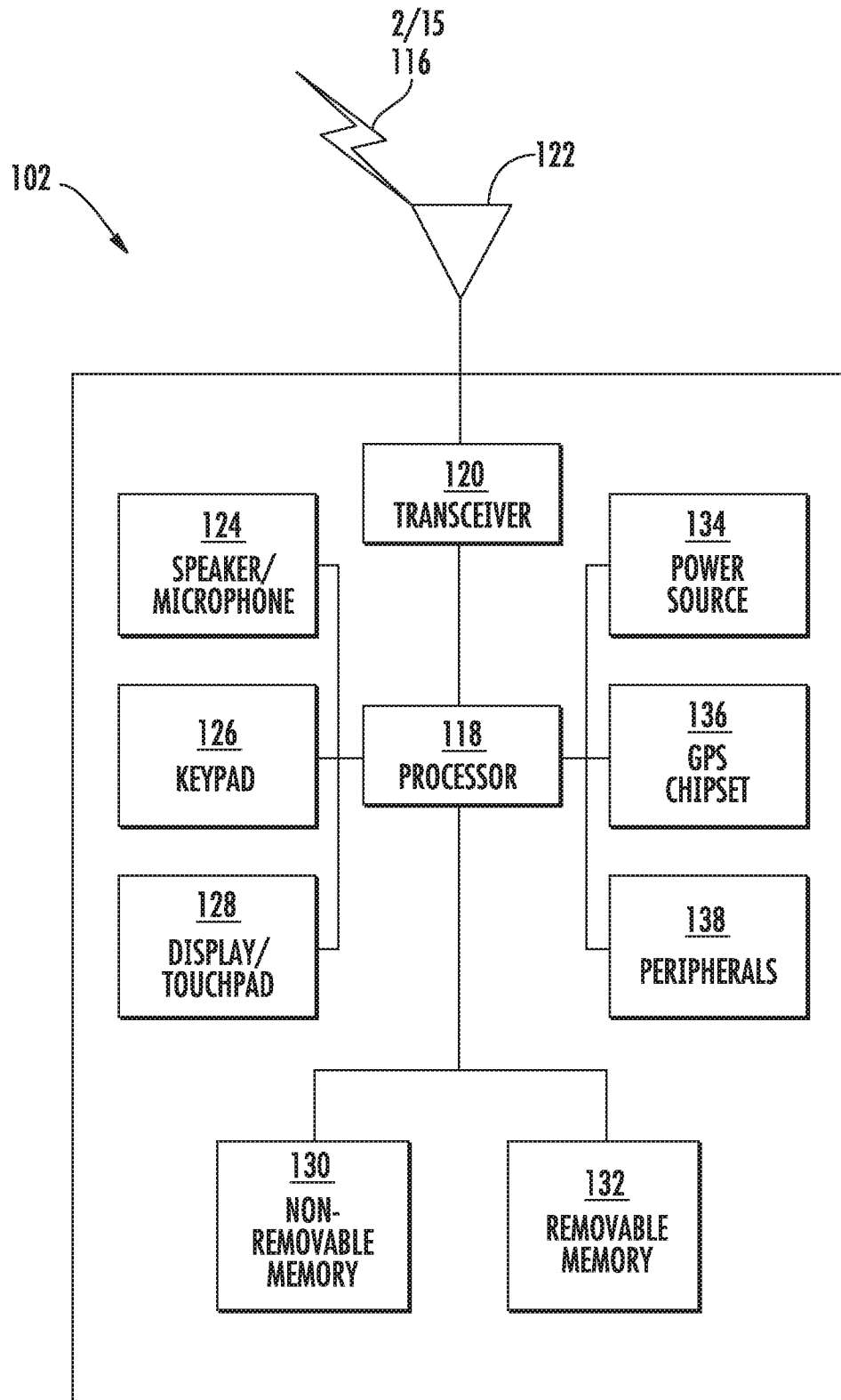


FIG. 1A

**FIG. 1B**

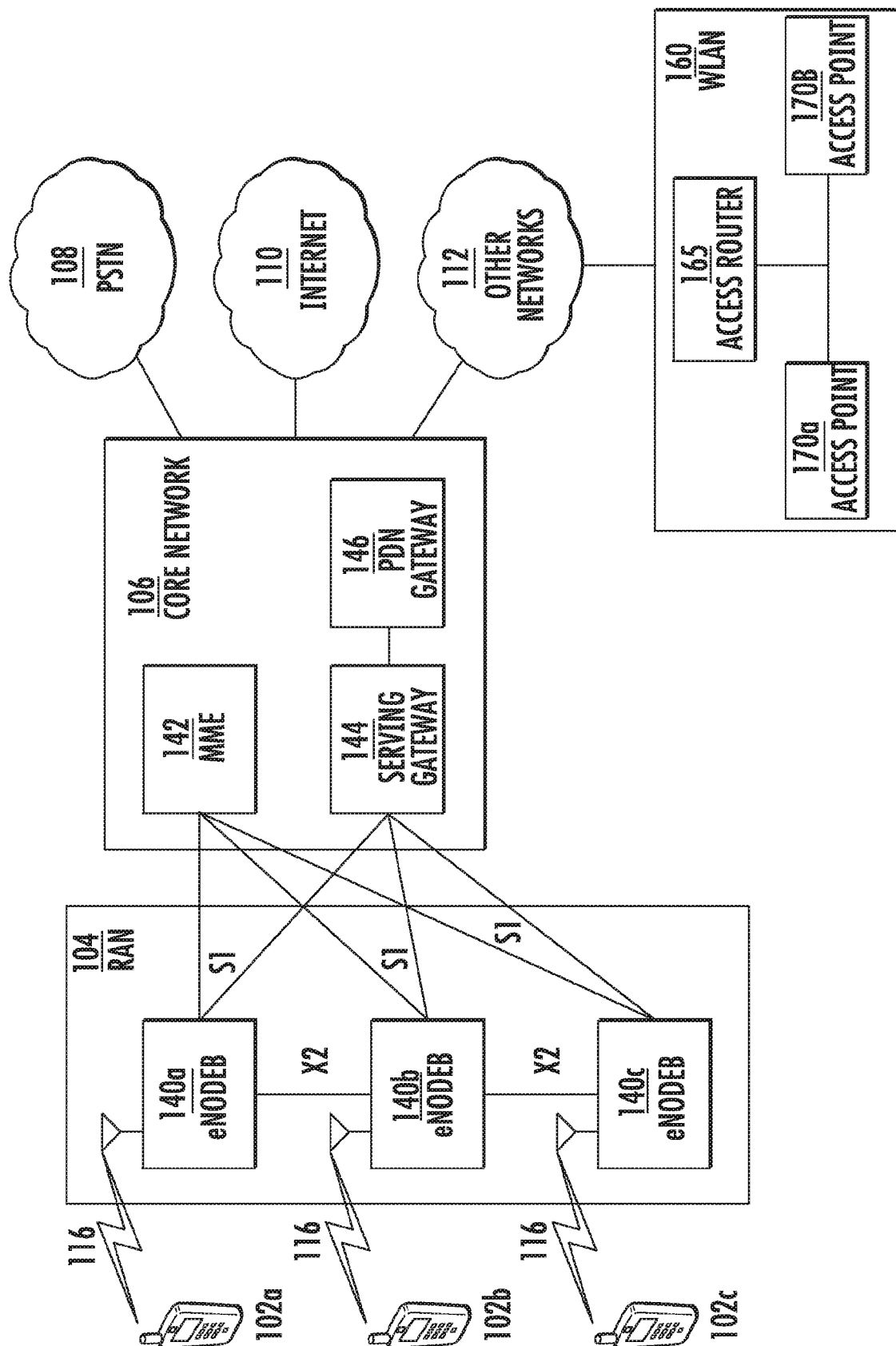


FIG. 1C

4/15

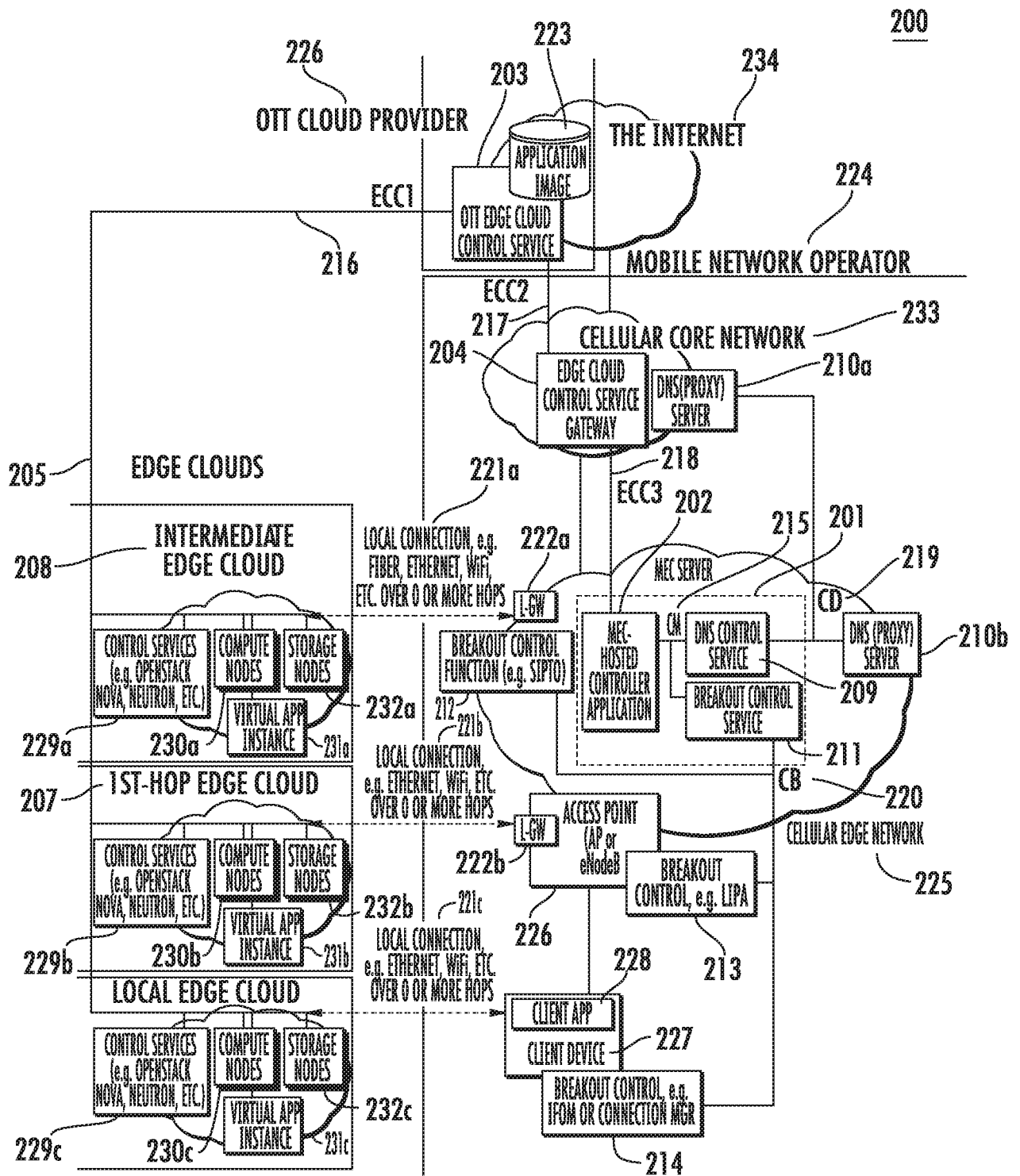


FIG. 2

5/15

300

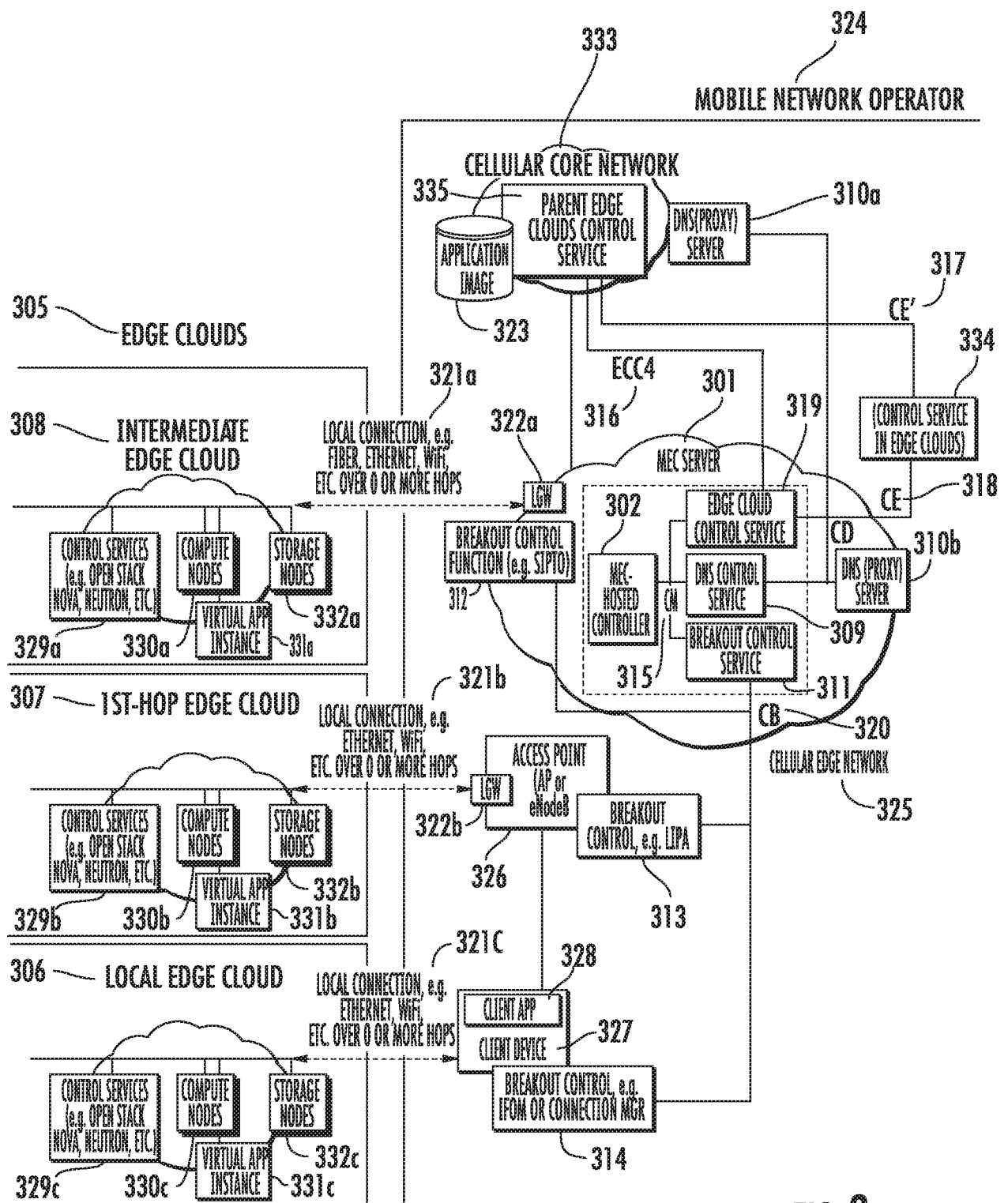
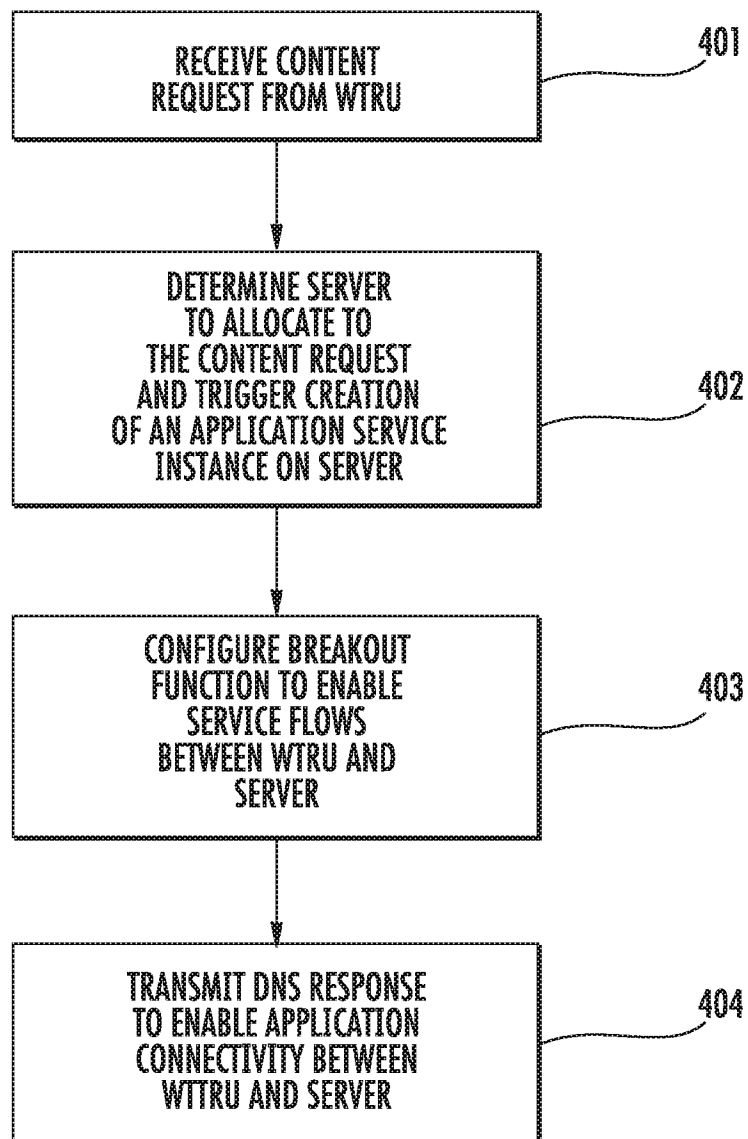


FIG. 3

6/15

400**FIG. 4**

7/15

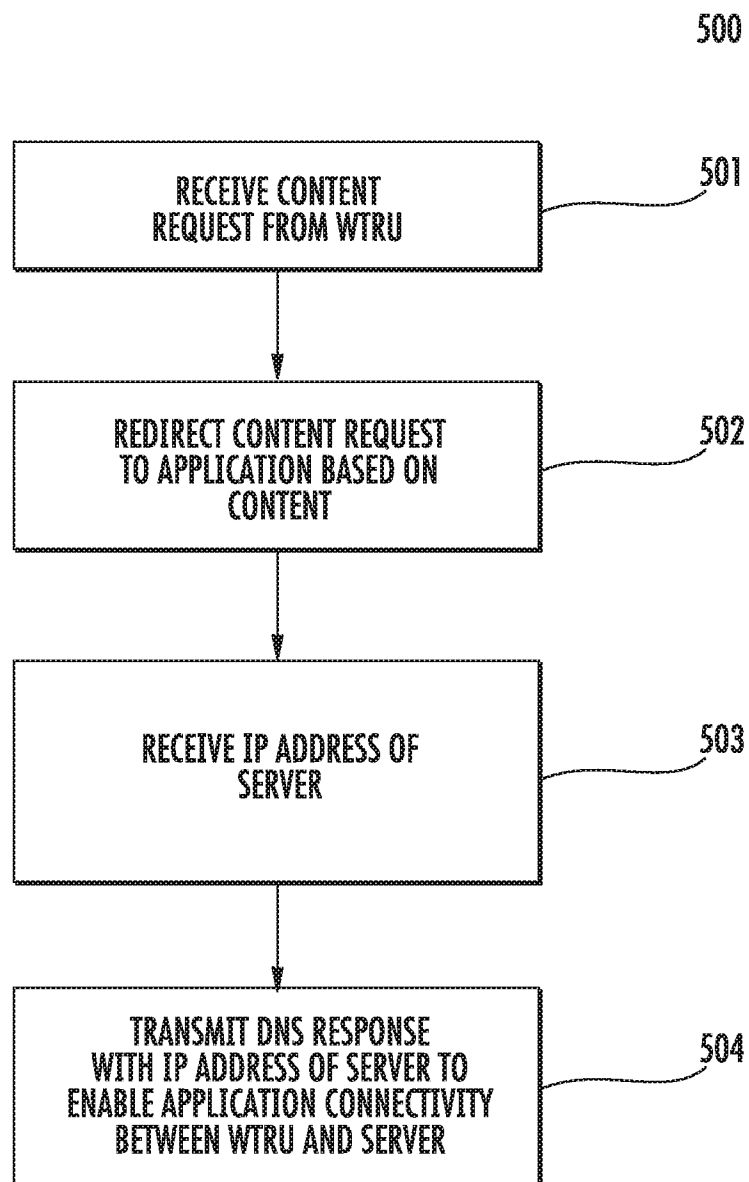


FIG. 5

8/15

600

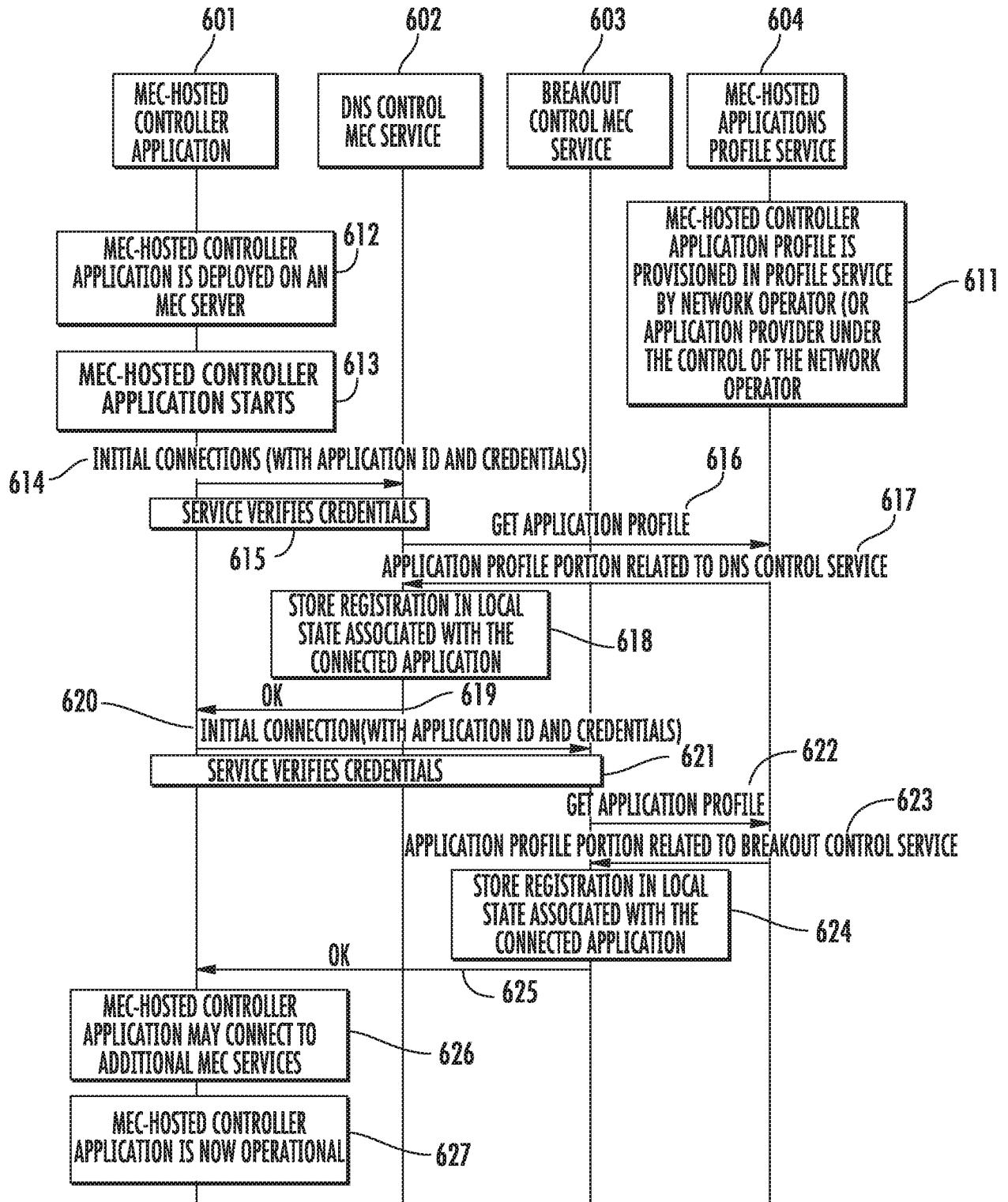


FIG. 6

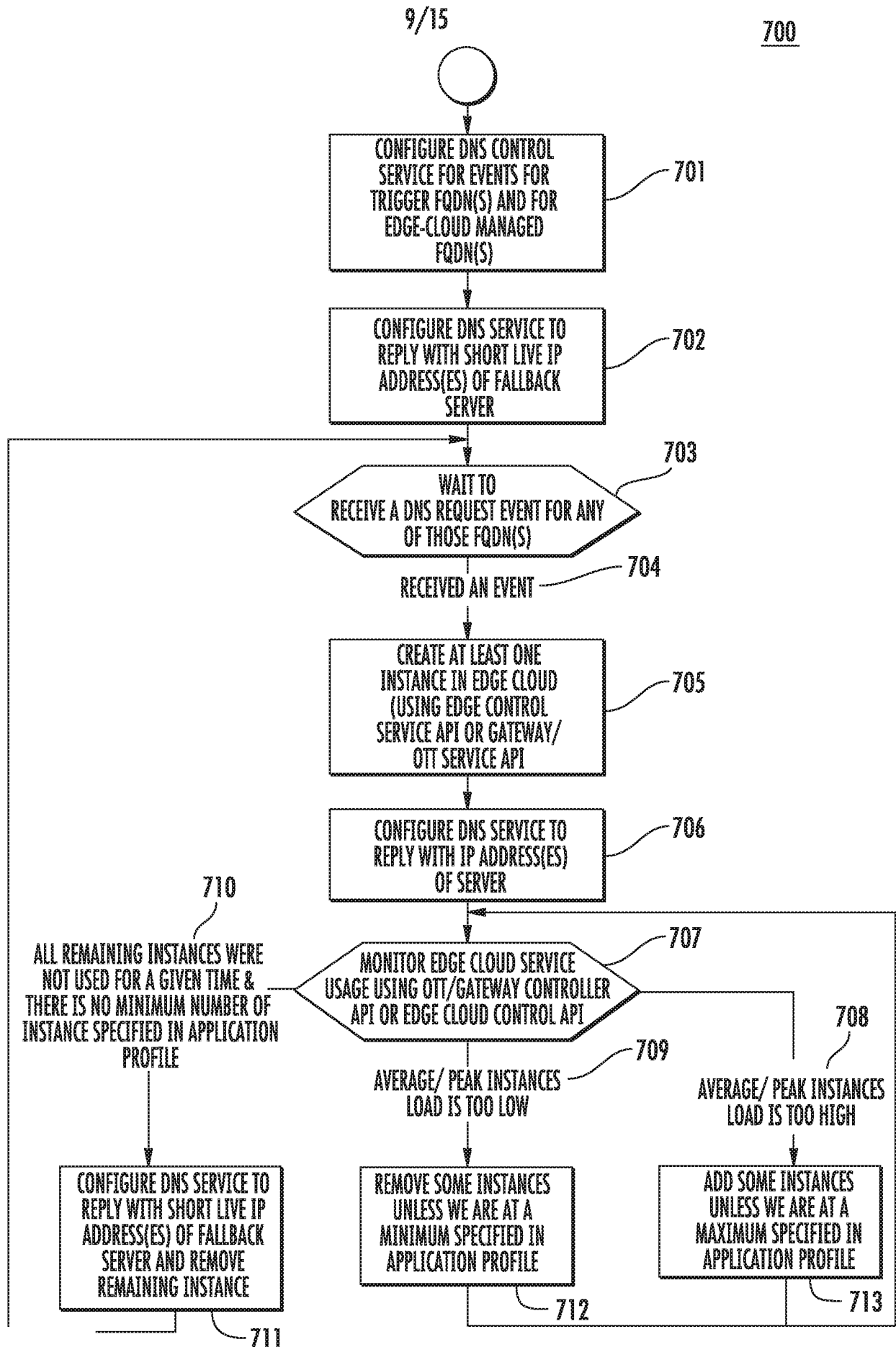


FIG. 7

10/15

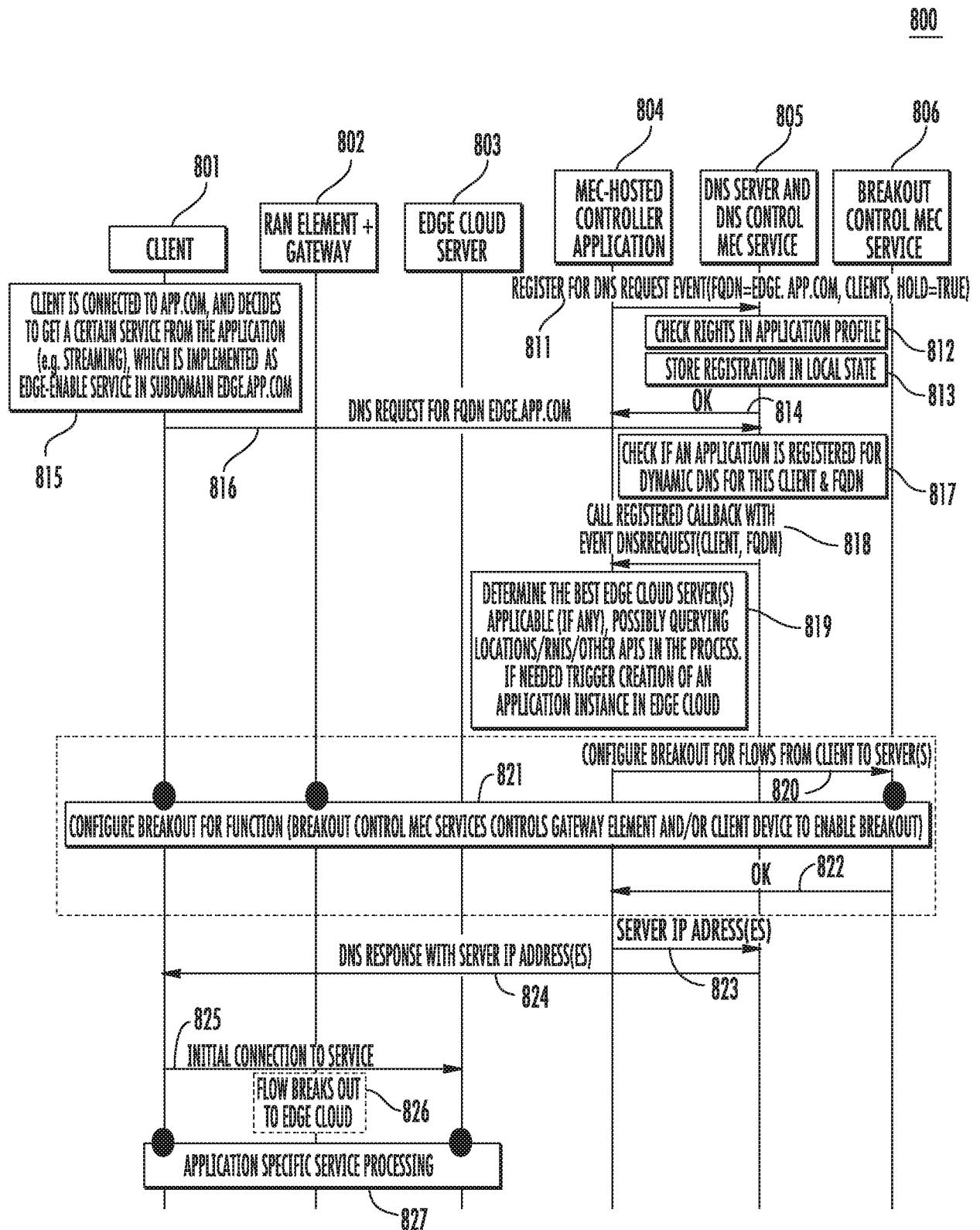


FIG. 8

900

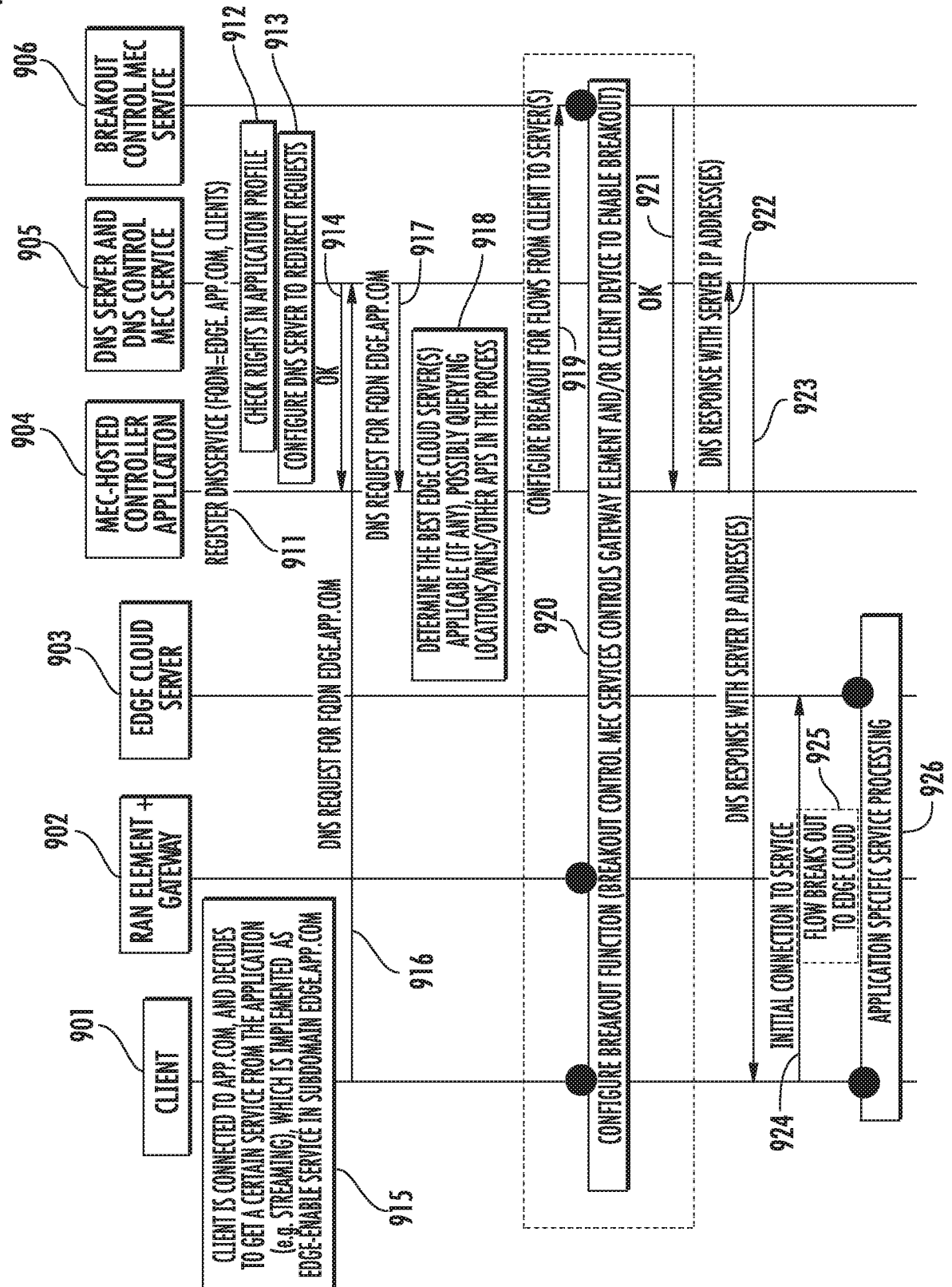


FIG. 9

12/15

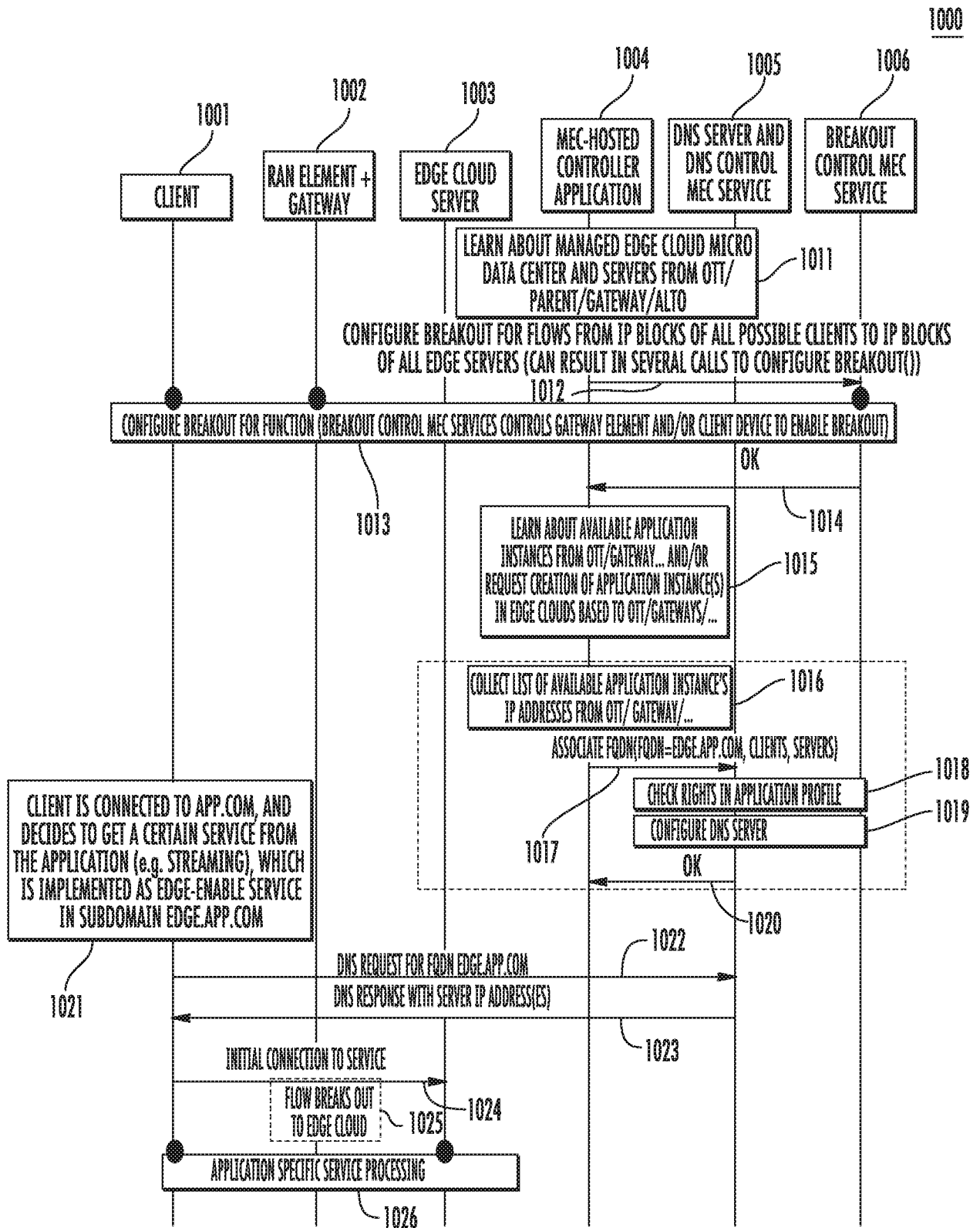


FIG. 10

13/15

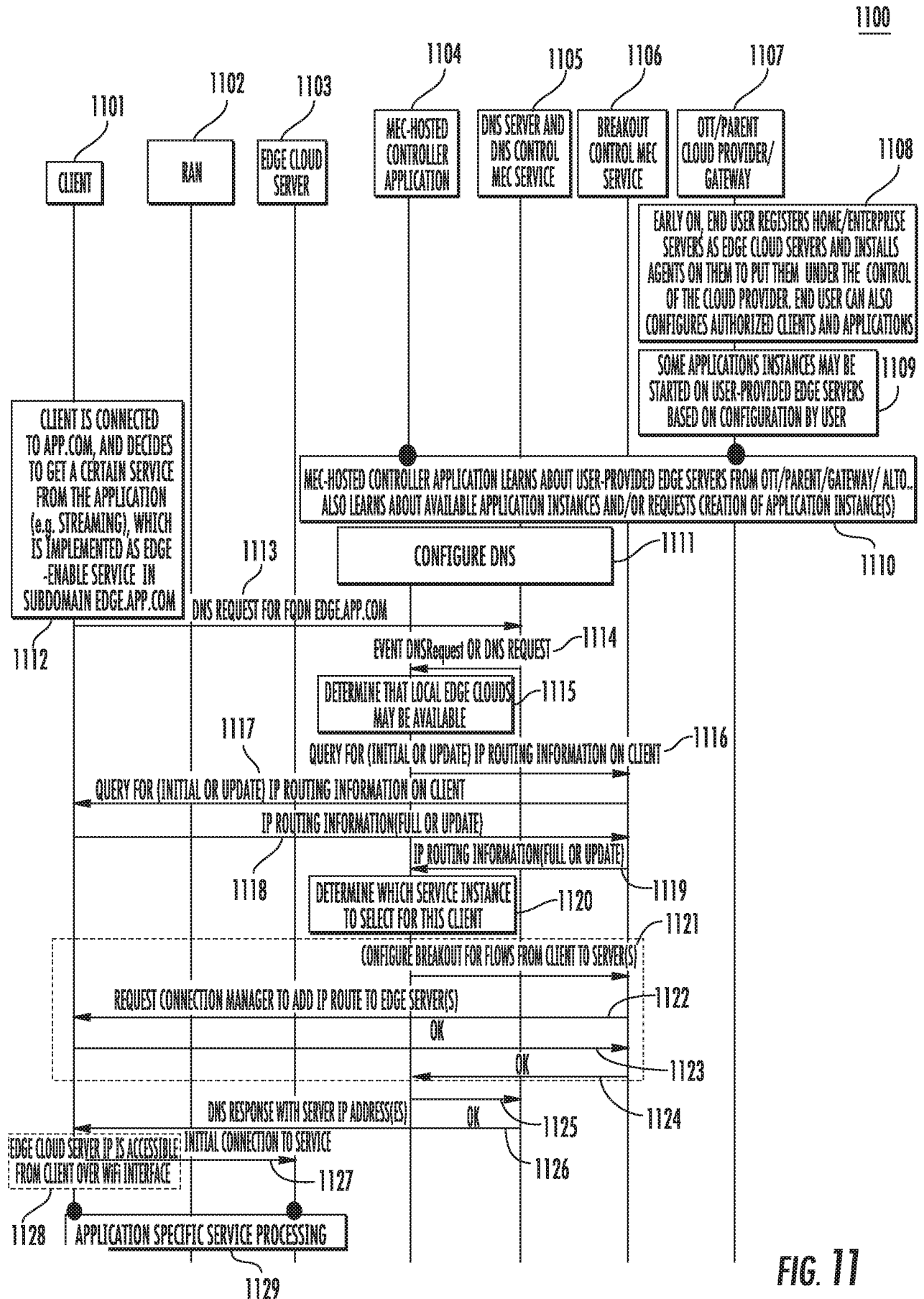


FIG. 11

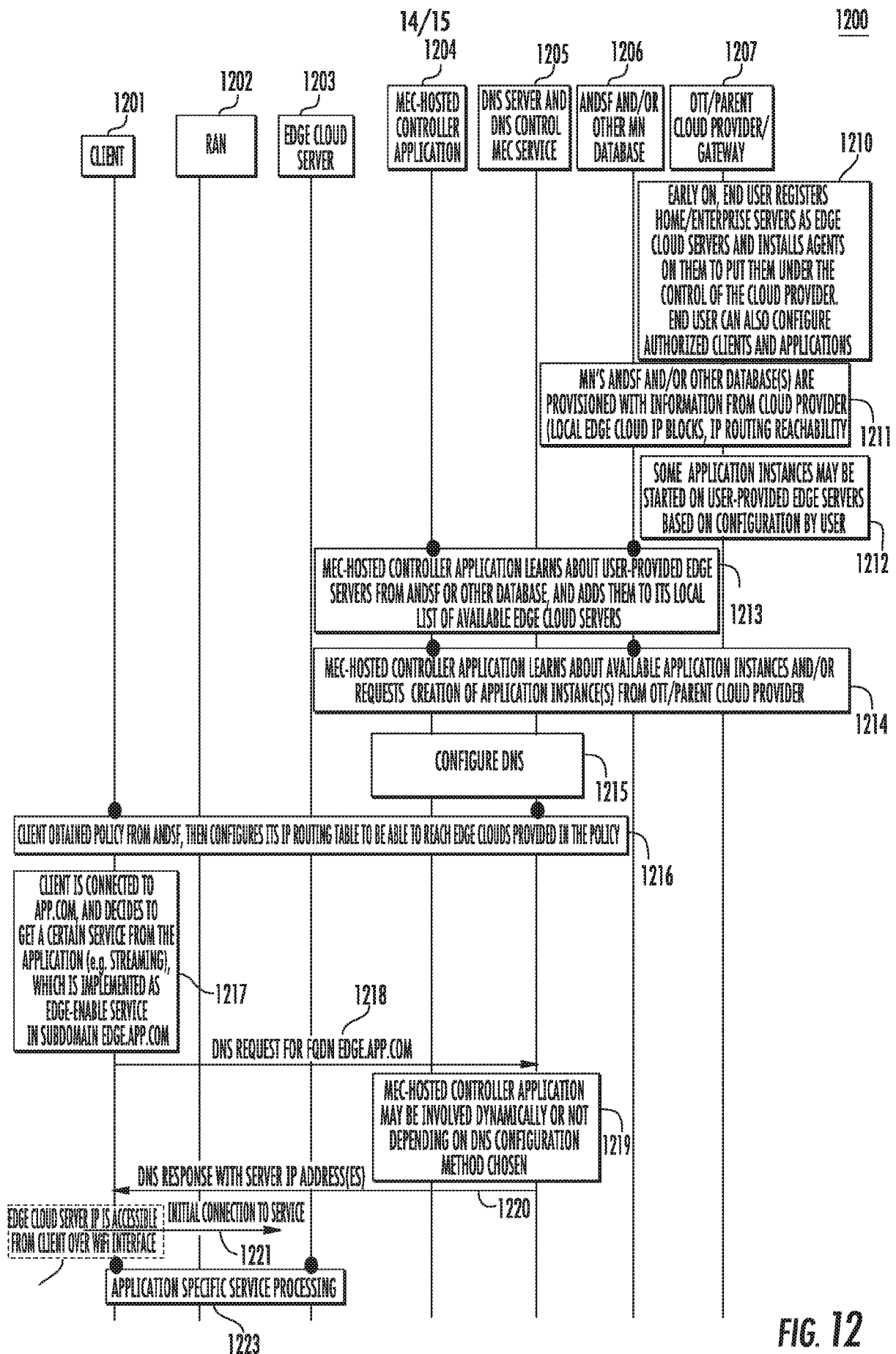


FIG. 12

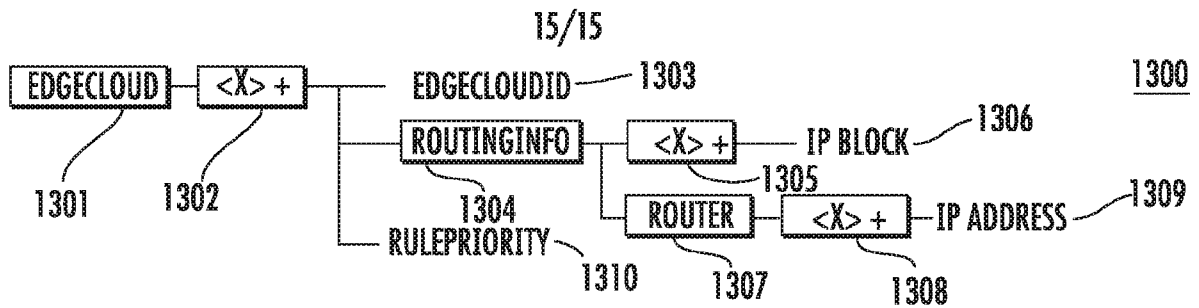


FIG. 13

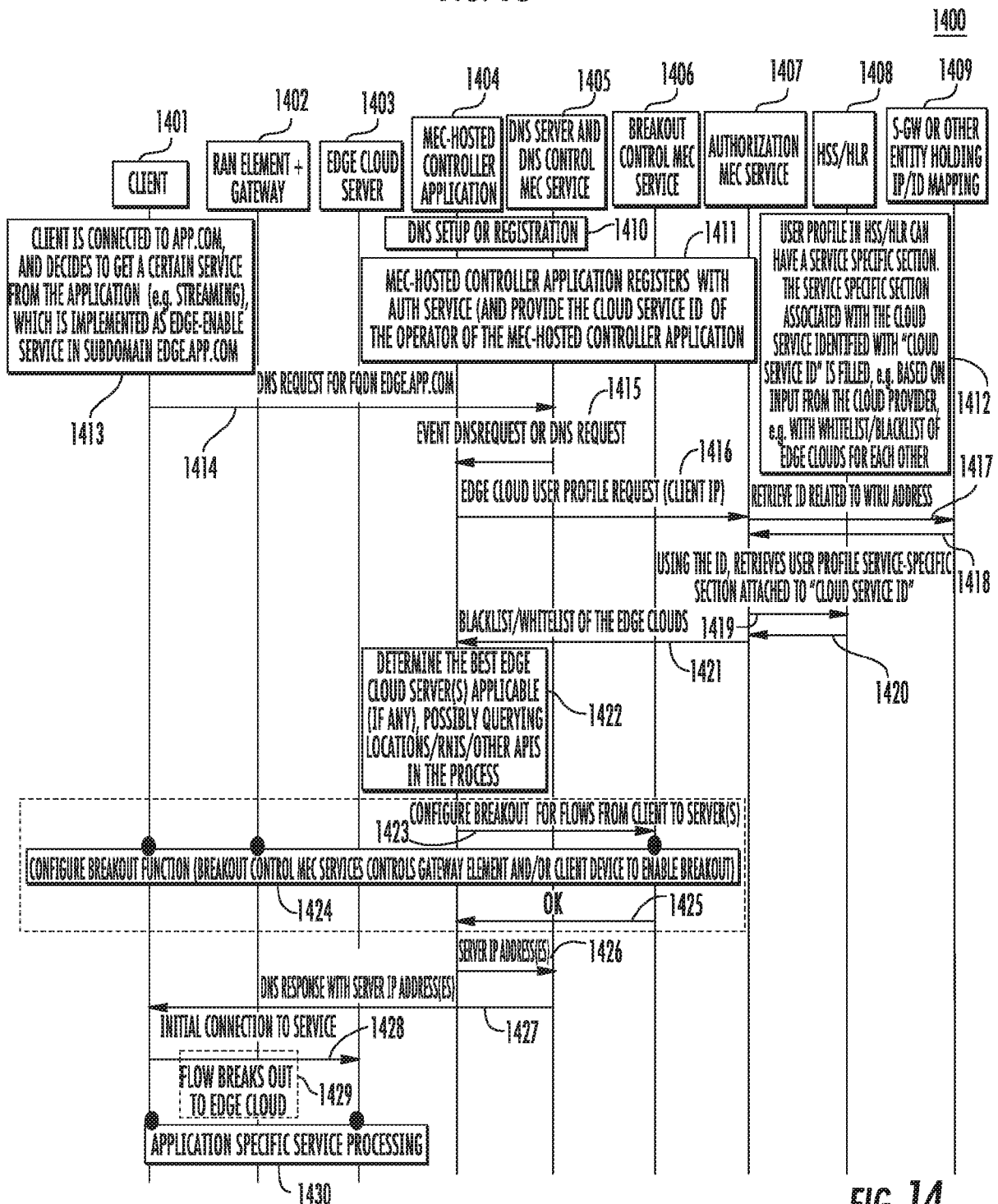


FIG. 14

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2016/065923A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/08 H04W76/02
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 H04W

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.
X	"Mobile-Edge Computing (MEC); Technical Requirements; Draft ETSI GS MEC 002", ETSI DRAFT; DRAFT ETSI GS MEC 002, EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (ETSI), 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS ; FRANCE, vol. ISG - MEC, no. V0.4.2, 15 September 2015 (2015-09-15), pages 1-41, XP014250208, [retrieved on 2015-09-15] page 11 page 15 - page 16 page 22 - page 23 page 27 - page 28 ----- -/-	1-21



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

14 March 2017

Date of mailing of the international search report

20/03/2017

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Wéry, Arnaud

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2016/065923

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	"Mobile-Edge Computing; Mobile-edge Computing - Introductory Technical White Paper V1 18-09-14", ETSI DRAFT; MOBILE-EDGE COMPUTING - INTRODUCTORY TECHNICAL WHITE PAPER V1 18-09-14, EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (ETSI), 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS ; FRANCE, vol. LI, 16 December 2014 (2014-12-16), pages 1-36, XP014232529, [retrieved on 2014-12-16] section 5 -----	1-21
A	US 2012/113893 A1 (DAMOLA AYODELE [SE]) 10 May 2012 (2012-05-10) paragraph [0066] - paragraph [0079] -----	1-21

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2016/065923

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012113893 A1	10-05-2012	CN 103181148 A	26-06-2013
		EP 2638688 A1	18-09-2013
		US 2012113893 A1	10-05-2012
		US 2016112217 A1	21-04-2016
		WO 2012064235 A1	18-05-2012
