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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 February 2024 (22.02.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/039735 A1

(51) International Patent Classification:
G06F 9/50 (2006.01) *H04L 67/51* (2022.01)
H04L 67/10 (2022.01)

(21) International Application Number:
PCT/US2023/030375

(22) International Filing Date:
16 August 2023 (16.08.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/399,162 18 August 2022 (18.08.2022) US
63/448,605 27 February 2023 (27.02.2023) US

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

(72) Inventors: **KRISHNA, Renan**; Shoreditch Business Center, 64 Great Eastern Street, London Greater London EC2A 3QR (GB). **BERNARDOS, Carlos Jesus**; Universidad Carlos III de Madrid, Avda. Universidad, 30, 28911 Leganes (ES).

(74) Agent: **SHAO, Yin**; 200 Bellevue Parkway, Suite 300, Wilmington, Delaware 19809 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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, ST, SV, SY, TH,

(54) Title: METHODS AND APPARATUSES FOR TASK DISTRIBUTION IN WIRELESS COMMUNICATIONS

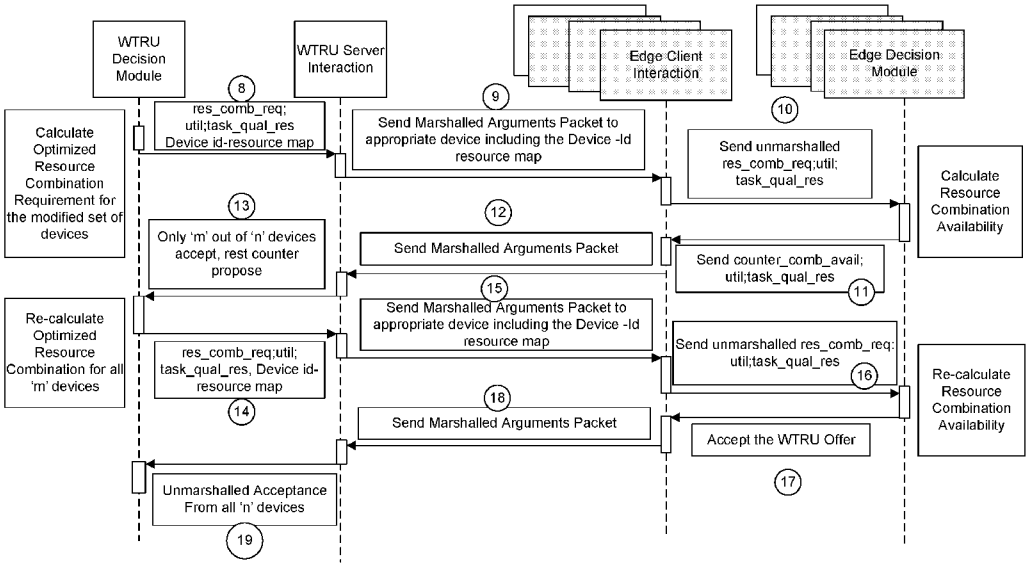


FIG. 12

(57) Abstract: Methods and apparatuses for enhancement of task distribution in wireless communications are provided. In an example, a wireless transmit/receive unit (WTRU) is configured to determine a first resource requirement for a list of tasks associated with one or more Edge devices; transmit, to the one or more Edge devices, a first request indicating the first resource requirement; receive first information indicating a combination of resources being available at the one or more Edge devices based on the first resource requirement; determine, based on the first information, a second resource requirement for the list of tasks; transmit a second request indicating the second resource requirement; and receive second information indicating a decision result associated with the list of tasks based on the second resource requirement.

[Continued on next page]

WO 2024/039735 A1

TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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))*

METHODS AND APPARATUSES FOR TASK DISTRIBUTION IN WIRELESS COMMUNICATIONS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to and the benefit of U.S. Provisional Application No. 63/399,162 filed in the U.S. Patent and Trademark Office on August 18, 2022, and U.S. Provisional Application No. 63/448,605 filed in the U.S. Patent and Trademark Office on February 27, 2023, the entire contents of each of which being incorporated herein by reference as if fully set forth below in their entirety and for all applicable purposes.

SUMMARY

[0002] This disclosure relates to communication networks, wireless and/or wired. For example, one or more embodiments disclosed herein are related to methods and apparatus for enhancement of task distribution(s) in wireless communications. Methods, procedures, and architectures for extended reality (XR) applications such as XR task distribution(s) and/or re-distribution(s) in communication networks are provided.

[0003] In one embodiment, methods, procedures, and components for devices (e.g., WTRUs) and the edge dynamically agree on the distribution of tasks in the face of varying wireless conditions and the fluctuations in the resource requirements of the running application are provided. For example, a method implemented by a wireless transmit/receive unit (WTRU) for wireless communications includes determining a first resource requirement for a list of tasks associated with one or more Edge devices; transmitting, to the one or more Edge devices, a first request indicating the first resource requirement; receiving first information indicating a combination of resources being available at the one or more Edge devices based on the first resource requirement; determining, based on the first information, a second resource requirement for the list of tasks; transmitting a second request indicating the second resource requirement; and receiving second information indicating a decision result associated with the list of tasks based on the second resource requirement.

[0004] In one embodiment, a method implemented by a wireless transmit/receive unit (WTRU) includes transmitting, to an Edge device, a first message including a first resource combination request; receiving, from the Edge device, a second message including information indicating a combination of resources being available at the Edge device based on the first resource combination request and/or a calculation of resources at the Edge device; determining resource combination requirements of an application based on the received second message; transmitting, to the Edge device, a third message including a second resource combination request based on the determined resource combination requirements; and receiving, from the Edge device,

a fourth message including information indicating a decision in response to the second resource combination request.

[0005] In one embodiment, the WTRU comprising a processor, a transmitter, a receiver, and/or memory is configured to implement one or more methods disclosed herein. For example, the WTRU is configured to determine a first resource requirement for a list of tasks associated with one or more Edge devices; transmit, to the one or more Edge devices, a first request indicating the first resource requirement; receive first information indicating a combination of resources being available at the one or more Edge devices based on the first resource requirement; determine, based on the first information, a second resource requirement for the list of tasks; transmit a second request indicating the second resource requirement; and receive second information indicating a decision result associated with the list of tasks based on the second resource requirement.

[0006] In another embodiment, extension to Session Description Protocol (SDP) is discussed. For example, a set of parameters such as System Utility function, combination of resource requirements, and/or associated quality values are exchanged between terminal device(s) (e.g., WTRU) and edge device(s).

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] A more detailed understanding may be had from the detailed description below, given by way of example in conjunction with drawings appended hereto. Figures in such drawings, like the detailed description, are examples. As such, the Figures (FIGs.) and the detailed description are not to be considered limiting, and other equally effective examples are possible and likely. Furthermore, like reference numerals ("ref.") in the FIGs. indicate like elements, and wherein:

[0008] FIG. 1A is a system diagram illustrating an example communications system in which one or more disclosed embodiments may be implemented;

[0009] FIG. 1B is a system diagram illustrating an example wireless transmit/receive unit (WTRU) that may be used within the communications system illustrated in FIG. 1A according to an embodiment;

[0010] FIG. 1C is a system diagram illustrating an example radio access network (RAN) and an example core network (CN) that may be used within the communications system illustrated in FIG. 1A according to an embodiment;

[0011] FIG. 1D is a system diagram illustrating a further example RAN and a further example CN that may be used within the communications system illustrated in FIG. 1A according to an embodiment;

[0012] FIG. 2 is a system diagram illustrating an example of an end-to-end path between a user equipment (UE) and Cloud that is segmented into three distinct tiers and used by current XR systems; and

[0013] FIG. 3 is an approximation algorithm for multiple resources and multiple QoS dimensions.

[0014] FIG. 4 is a system diagram illustrating an exemplary architecture running on a WTRU, according to one or more embodiments;

[0015] FIG. 5 is a system diagram illustrating an exemplary architecture running on an edge device, according to one or more embodiments;

[0016] FIG. 6 is a system diagram illustrating an example of modules running on a WTRU, according to one or more embodiments;

[0017] FIG. 7 is a system diagram illustrating an example of modules running on an edge device, according to one or more embodiments;

[0018] FIG. 8 is a diagram illustrating an architecture to support Extended Reality (XR), according to one or more embodiments;

[0019] FIG. 9 is a diagram illustrating a first example of an architecture to extend the functionality of XR, according to one or more embodiments;

[0020] FIG. 10 is a diagram illustrating a second example of an architecture to extend the functionality of XR, according to one or more embodiments;

[0021] FIG. 11 is a diagram illustrating a first example of sequence of messages in a procedure for Case 1, according to one or more embodiments;

[0022] FIG. 12 is a diagram illustrating a second example of sequence of messages in a procedure for Case 1, according to one or more embodiments;

[0023] FIG. 13 is a diagram illustrating an example of sequence of messages in a procedure for Case 2, according to one or more embodiments;

[0024] FIG. 14 is a diagram illustrating an exemplary packet structure for communication between a WTRU and Edge device(s), according to one or more embodiments; and

[0025] FIG. 15 is a diagram illustrating an example of a packet structure with an SDP message, according to one or more embodiments.

DETAILED DESCRIPTION

[0026] In the following detailed description, numerous specific details are set forth to provide a thorough understanding of embodiments and/or examples disclosed herein. However, it will be understood that such embodiments and examples may be practiced without some or all of the specific details set forth herein. In other instances, well-known methods, procedures, components and circuits have not been described in detail, so as not to obscure the following description. Further, embodiments and examples not specifically described herein may be practiced in lieu of, or in combination with, the embodiments and other examples described, disclosed or otherwise provided explicitly, implicitly and/or inherently (collectively "provided") herein. Although various embodiments are described and/or claimed herein in which an apparatus, system,

device, etc. and/or any element thereof carries out an operation, process, algorithm, function, etc. and/or any portion thereof, it is to be understood that any embodiments described and/or claimed herein assume that any apparatus, system, device, etc. and/or any element thereof is configured to carry out any operation, process, algorithm, function, etc. and/or any portion thereof.

Communications Networks and Devices

[0027] The methods, apparatuses and systems provided herein are well-suited for communications involving both wired and wireless networks. An overview of various types of wireless devices and infrastructure is provided with respect to FIGs. 1A-1D, where various elements of the network may utilize, perform, be arranged in accordance with and/or be adapted and/or configured for the methods, apparatuses and systems provided herein.

[0028] FIG. 1A is a system diagram illustrating 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), zero-tail (ZT) unique-word (UW) discreet Fourier transform (DFT) spread OFDM (ZT UW DTS-s OFDM), unique word OFDM (UW-OFDM), resource block-filtered OFDM, filter bank multicarrier (FBMC), and the like.

[0029] 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) 104/113, a core network (CN) 106/115, 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, any of which may be referred to as a "station" and/or a "STA", may be configured to transmit and/or receive wireless signals and may include (or be) a user equipment (UE), a mobile station, a fixed or mobile subscriber unit, a subscription-based unit, a pager, a cellular telephone, a personal digital assistant (PDA), a smartphone, a laptop, a netbook, a personal computer, a wireless sensor, a hotspot or Mi-Fi device, an Internet of Things (IoT) device, a watch or other wearable, a head-mounted display (HMD), a vehicle, a drone, a medical device and applications (e.g., remote surgery), an industrial device and applications (e.g., a robot and/or other wireless devices operating in an industrial and/or an automated processing chain contexts), a

consumer electronics device, a device operating on commercial and/or industrial wireless networks, and the like. Any of the WTRUs 102a, 102b, 102c and 102d may be interchangeably referred to as a UE.

[0030] The communications systems 100 may also include a base station 114a and/or 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, e.g., to facilitate access to one or more communication networks, such as the CN 106/115, the Internet 110, and/or the networks 112. By way of example, the base stations 114a, 114b may be any of a base transceiver station (BTS), a Node-B (NB), an eNode-B (eNB), a Home Node-B (HNB), a Home eNode-B (HeNB), a gNode-B (gNB), a NR Node-B (NR NB), 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.

[0031] The base station 114a may be part of the RAN 104/113, 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 on one or more carrier frequencies, which may be referred to as a cell (not shown). These frequencies may be in licensed spectrum, unlicensed spectrum, or a combination of licensed and unlicensed spectrum. A cell may provide coverage for a wireless service to a specific geographical area that may be relatively fixed or that may change over time. 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 an embodiment, the base station 114a may include three transceivers, i.e., one for each sector of the cell. In an embodiment, the base station 114a may employ multiple-input multiple output (MIMO) technology and may utilize multiple transceivers for each or any sector of the cell. For example, beamforming may be used to transmit and/or receive signals in desired spatial directions.

[0032] 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, centimeter wave, micrometer wave, infrared (IR), ultraviolet (UV), visible light, etc.). The air interface 116 may be established using any suitable radio access technology (RAT).

[0033] 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 104/113 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).

[0034] In an 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) and/or LTE-Advanced Pro (LTE-A Pro).

[0035] In an embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement a radio technology such as NR Radio Access, which may establish the air interface 116 using New Radio (NR).

[0036] In an embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement multiple radio access technologies. For example, the base station 114a and the WTRUs 102a, 102b, 102c may implement LTE radio access and NR radio access together, for instance using dual connectivity (DC) principles. Thus, the air interface utilized by WTRUs 102a, 102b, 102c may be characterized by multiple types of radio access technologies and/or transmissions sent to/from multiple types of base stations (e.g., an eNB and a gNB).

[0037] In an embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement radio technologies such as IEEE 802.11 (i.e., Wireless Fidelity (Wi-Fi), 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.

[0038] The base station 114b in FIG. 1A may be a wireless router, Home Node-B, Home eNode-B, 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, an industrial facility, an air corridor (e.g., for use by drones), a roadway, and the like. In an 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 an 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 an embodiment, the base station 114b and the WTRUs 102c, 102d may utilize a cellular-based RAT (e.g., WCDMA, CDMA2000, GSM, LTE, LTE-A, LTE-A Pro, NR, etc.) to establish any of a small cell, 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 CN 106/115.

[0039] The RAN 104/113 may be in communication with the CN 106/115, 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. The data may have varying quality of service (QoS) requirements, such as differing throughput requirements, latency requirements, error tolerance requirements,

reliability requirements, data throughput requirements, mobility requirements, and the like. The CN 106/115 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 104/113 and/or the CN 106/115 may be in direct or indirect communication with other RANs that employ the same RAT as the RAN 104/113 or a different RAT. For example, in addition to being connected to the RAN 104/113, which may be utilizing an NR radio technology, the CN 106/115 may also be in communication with another RAN (not shown) employing any of a GSM, UMTS, CDMA 2000, WiMAX, E-UTRA, or Wi-Fi radio technology.

[0040] The CN 106/115 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/or the internet protocol (IP) in the TCP/IP internet protocol suite. The networks 112 may include wired and/or wireless communications networks owned and/or operated by other service providers. For example, the networks 112 may include another CN connected to one or more RANs, which may employ the same RAT as the RAN 104/114 or a different RAT.

[0041] Some or all of the WTRUs 102a, 102b, 102c, 102d in the communications system 100 may include multi-mode capabilities (e.g., the WTRUs 102a, 102b, 102c, 102d may include multiple transceivers 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.

[0042] FIG. 1B is a system diagram illustrating 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/or other elements/peripherals 138, among others. It will be appreciated that the WTRU 102 may include any sub-combination of the foregoing elements while remaining consistent with an embodiment.

[0043] 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 Arrays (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, e.g., in an electronic package or chip.

[0044] 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 an embodiment, the transmit/receive element 122 may be an antenna configured to transmit and/or receive RF signals. In an 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 an embodiment, the transmit/receive element 122 may be configured to transmit and/or 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.

[0045] 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. For example, the WTRU 102 may employ MIMO technology. Thus, in an 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.

[0046] 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 NR and IEEE 802.11, for example.

[0047] 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).

[0048] 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.

[0049] 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.

[0050] The processor 118 may further be coupled to other elements/peripherals 138, which may include one or more software and/or hardware modules/units that provide additional features, functionality and/or wired or wireless connectivity. For example, the elements/peripherals 138 may include an accelerometer, an e-compass, a satellite transceiver, a digital camera (e.g., for photographs and/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, a virtual reality and/or augmented reality (VR/AR) device, an activity tracker, and the like. The elements/peripherals 138 may include one or more sensors, the sensors may be one or more of a gyroscope, an accelerometer, a hall effect sensor, a magnetometer, an orientation sensor, a proximity sensor, a temperature sensor, a time sensor; a geolocation sensor; an altimeter, a light sensor, a touch sensor, a magnetometer, a barometer, a gesture sensor, a biometric sensor, and/or a humidity sensor.

[0051] The WTRU 102 may include a full duplex radio for which transmission and reception of some or all of the signals (e.g., associated with particular subframes for both the uplink (e.g., for transmission) and downlink (e.g., for reception) may be concurrent and/or simultaneous. The full duplex radio may include an interference management unit to reduce and or substantially eliminate self-interference via either hardware (e.g., a choke) or signal processing via a processor (e.g., a separate processor (not shown) or via processor 118). In an embodiment, the WTRU 102 may include a half-duplex radio for which transmission and reception of some or all of the signals (e.g., associated with particular subframes for either the uplink (e.g., for transmission) or the downlink (e.g., for reception)).

[0052] FIG. 1C is a system diagram illustrating the RAN 104 and the CN 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, and 102c over the air interface 116. The RAN 104 may also be in communication with the CN 106.

[0053] The RAN 104 may include eNode-Bs 160a, 160b, 160c, though it will be appreciated that the RAN 104 may include any number of eNode-Bs while remaining consistent with an embodiment. The eNode-Bs 160a, 160b, 160c may each include one or more transceivers for communicating with the WTRUs 102a, 102b, 102c over the air interface 116. In an embodiment, the eNode-Bs 160a, 160b, 160c may implement MIMO technology. Thus, the eNode-B 160a, for example, may use multiple antennas to transmit wireless signals to, and receive wireless signals from, the WTRU 102a.

[0054] Each of the eNode-Bs 160a, 160b, and 160c 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 (UL) and/or downlink (DL), and the like. As shown in FIG. 1C, the eNode-Bs 160a, 160b, 160c may communicate with one another over an X2 interface.

[0055] The CN 106 shown in FIG. 1C may include a mobility management entity (MME) 162, a serving gateway (SGW) 164, and a packet data network (PDN) gateway (PGW) 166. While each of the foregoing elements are depicted as part of the CN 106, it will be appreciated that any one of these elements may be owned and/or operated by an entity other than the CN operator.

[0056] The MME 162 may be connected to each of the eNode-Bs 160a, 160b, and 160c in the RAN 104 via an S1 interface and may serve as a control node. For example, the MME 162 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 162 may provide a control plane function for switching between the RAN 104 and other RANs (not shown) that employ other radio technologies, such as GSM and/or WCDMA.

[0057] The SGW 164 may be connected to each of the eNode-Bs 160a, 160b, 160c in the RAN 104 via the S1 interface. The SGW 164 may generally route and forward user data packets to/from the WTRUs 102a, 102b, 102c. The SGW 164 may perform other functions, such as anchoring user planes during inter-eNode-B handovers, triggering paging when DL data is available for the WTRUs 102a, 102b, 102c, managing and storing contexts of the WTRUs 102a, 102b, 102c, and the like.

[0058] The SGW 164 may be connected to the PGW 166, 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.

[0059] The CN 106 may facilitate communications with other networks. For example, the CN 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 CN 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 CN 106 and the PSTN 108. In addition, the CN 106 may provide the WTRUs 102a, 102b, 102c with access to the other networks 112, which may include other wired and/or wireless networks that are owned and/or operated by other service providers.

[0060] Although the WTRU is described in FIGs. 1A-1D as a wireless terminal, it is contemplated that in certain representative embodiments that such a terminal may use (e.g., temporarily or permanently) wired communication interfaces with the communication network.

[0061] In representative embodiments, the other network 112 may be a WLAN.

[0062] A WLAN in infrastructure basic service set (BSS) mode may have an access point (AP) for the BSS and one or more stations (STAs) associated with the AP. The AP may have an access or an interface to a distribution system (DS) or another type of wired/wireless network that carries traffic into and/or out of the BSS. Traffic to STAs that originates from outside the BSS may arrive through the AP and may be delivered to the STAs. Traffic originating from STAs to destinations outside the BSS may be sent to the AP to be delivered to respective destinations. Traffic between STAs within the BSS may be sent through the AP, for example, where the source STA may send traffic to the AP and the AP may deliver the traffic to the destination STA. The traffic between STAs within a BSS may be considered and/or referred to as peer-to-peer traffic. The peer-to-peer traffic may be sent between (e.g., directly between) the source and destination STAs with a direct link setup (DLS). In certain representative embodiments, the DLS may use an 802.11e DLS or an 802.11z tunneled DLS (TDLS). A WLAN using an Independent BSS (IBSS) mode may not have an AP, and the STAs (e.g., all of the STAs) within or using the IBSS may communicate directly with each other. The IBSS mode of communication may sometimes be referred to herein as an "ad-hoc" mode of communication.

[0063] When using the 802.11ac infrastructure mode of operation or a similar mode of operations, the AP may transmit a beacon on a fixed channel, such as a primary channel. The primary channel may be a fixed width (e.g., 20 MHz wide bandwidth) or a dynamically set width via signaling. The primary channel may be the operating channel of the BSS and may be used by the STAs to establish a connection with the AP. In certain representative embodiments, Carrier sense multiple access with collision avoidance (CSMA/CA) may be implemented, for example in in 802.11 systems. For CSMA/CA, the STAs (e.g., every STA), including the AP, may sense the primary channel. If the primary channel is sensed/detected and/or determined to be busy by a particular STA, the particular STA may back off. One STA (e.g., only one station) may transmit at any given time in a given BSS.

[0064] High throughput (HT) STAs may use a 40 MHz wide channel for communication, for example, via a combination of the primary 20 MHz channel with an adjacent or nonadjacent 20 MHz channel to form a 40 MHz wide channel.

[0065] Very high throughput (VHT) STAs may support 20 MHz, 40 MHz, 80 MHz, and/or 160 MHz wide channels. The 40 MHz, and/or 80 MHz, channels may be formed by combining contiguous 20 MHz channels.

A 160 MHz channel may be formed by combining 8 contiguous 20 MHz channels, or by combining two non-contiguous 80 MHz channels, which may be referred to as an 80+80 configuration. For the 80+80 configuration, the data, after channel encoding, may be passed through a segment parser that may divide the data into two streams. Inverse fast fourier transform (IFFT) processing, and time domain processing, may be done on each stream separately. The streams may be mapped on to the two 80 MHz channels, and the data may be transmitted by a transmitting STA. At the receiver of the receiving STA, the above-described operation for the 80+80 configuration may be reversed, and the combined data may be sent to a medium access control (MAC) layer, entity, etc.

[0066] Sub 1 GHz modes of operation are supported by 802.11af and 802.11ah. The channel operating bandwidths, and carriers, are reduced in 802.11af and 802.11ah relative to those used in 802.11n, and 802.11ac. 802.11af supports 5 MHz, 10 MHz and 20 MHz bandwidths in the TV white space (TVWS) spectrum, and 802.11ah supports 1 MHz, 2 MHz, 4 MHz, 8 MHz, and 16 MHz bandwidths using non-TVWS spectrum. According to a representative embodiment, 802.11ah may support meter type control/machine-type communications (MTC), such as MTC devices in a macro coverage area. MTC devices may have certain capabilities, for example, limited capabilities including support for (e.g., only support for) certain and/or limited bandwidths. The MTC devices may include a battery with a battery life above a threshold (e.g., to maintain a very long battery life).

[0067] WLAN systems, which may support multiple channels, and channel bandwidths, such as 802.11n, 802.11ac, 802.11af, and 802.11ah, include a channel which may be designated as the primary channel. The primary channel may have a bandwidth equal to the largest common operating bandwidth supported by all STAs in the BSS. The bandwidth of the primary channel may be set and/or limited by a STA, from among all STAs in operating in a BSS, which supports the smallest bandwidth operating mode. In the example of 802.11ah, the primary channel may be 1 MHz wide for STAs (e.g., MTC type devices) that support (e.g., only support) a 1 MHz mode, even if the AP, and other STAs in the BSS support 2 MHz, 4 MHz, 8 MHz, 16 MHz, and/or other channel bandwidth operating modes. Carrier sensing and/or network allocation vector (NAV) settings may depend on the status of the primary channel. If the primary channel is busy, for example, due to a STA (which supports only a 1 MHz operating mode), transmitting to the AP, the entire available frequency bands may be considered busy even though a majority of the frequency bands remains idle and may be available.

[0068] In the United States, the available frequency bands, which may be used by 802.11ah, are from 902 MHz to 928 MHz. In Korea, the available frequency bands are from 917.5 MHz to 923.5 MHz. In Japan, the available frequency bands are from 916.5 MHz to 927.5 MHz. The total bandwidth available for 802.11ah is 6 MHz to 26 MHz depending on the country code.

[0069] FIG. 1D is a system diagram illustrating the RAN 113 and the CN 115 according to an embodiment. As noted above, the RAN 113 may employ an NR radio technology to communicate with the WTRUs 102a, 102b, 102c over the air interface 116. The RAN 113 may also be in communication with the CN 115.

[0070] The RAN 113 may include gNBs 180a, 180b, 180c, though it will be appreciated that the RAN 113 may include any number of gNBs while remaining consistent with an embodiment. The gNBs 180a, 180b, 180c may each include one or more transceivers for communicating with the WTRUs 102a, 102b, 102c over the air interface 116. In an embodiment, the gNBs 180a, 180b, 180c may implement MIMO technology. For example, gNBs 180a, 180b may utilize beamforming to transmit signals to and/or receive signals from the WTRUs 102a, 102b, 102c. Thus, the gNB 180a, for example, may use multiple antennas to transmit wireless signals to, and/or receive wireless signals from, the WTRU 102a. In an embodiment, the gNBs 180a, 180b, 180c may implement carrier aggregation technology. For example, the gNB 180a may transmit multiple component carriers to the WTRU 102a (not shown). A subset of these component carriers may be on unlicensed spectrum while the remaining component carriers may be on licensed spectrum. In an embodiment, the gNBs 180a, 180b, 180c may implement Coordinated Multi-Point (CoMP) technology. For example, WTRU 102a may receive coordinated transmissions from gNB 180a and gNB 180b (and/or gNB 180c).

[0071] The WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c using transmissions associated with a scalable numerology. For example, OFDM symbol spacing and/or OFDM subcarrier spacing may vary for different transmissions, different cells, and/or different portions of the wireless transmission spectrum. The WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c using subframe or transmission time intervals (TTIs) of various or scalable lengths (e.g., including a varying number of OFDM symbols and/or lasting varying lengths of absolute time).

[0072] The gNBs 180a, 180b, 180c may be configured to communicate with the WTRUs 102a, 102b, 102c in a standalone configuration and/or a non-standalone configuration. In the standalone configuration, WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c without also accessing other RANs (e.g., such as eNode-Bs 160a, 160b, 160c). In the standalone configuration, WTRUs 102a, 102b, 102c may utilize one or more of gNBs 180a, 180b, 180c as a mobility anchor point. In the standalone configuration, WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c using signals in an unlicensed band. In a non-standalone configuration WTRUs 102a, 102b, 102c may communicate with/connect to gNBs 180a, 180b, 180c while also communicating with/connecting to another RAN such as eNode-Bs 160a, 160b, 160c. For example, WTRUs 102a, 102b, 102c may implement DC principles to communicate with one or more gNBs 180a, 180b, 180c and one or more eNode-Bs 160a, 160b, 160c substantially simultaneously. In the non-standalone configuration, eNode-Bs 160a, 160b, 160c may serve as a mobility anchor for WTRUs 102a,

102b, 102c and gNBs 180a, 180b, 180c may provide additional coverage and/or throughput for servicing WTRUs 102a, 102b, 102c.

[0073] Each of the gNBs 180a, 180b, 180c 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 UL and/or DL, support of network slicing, dual connectivity, interworking between NR and E-UTRA, routing of user plane data towards user plane functions (UPFs) 184a, 184b, routing of control plane information towards access and mobility management functions (AMFs) 182a, 182b, and the like. As shown in FIG. 1D, the gNBs 180a, 180b, 180c may communicate with one another over an Xn interface.

[0074] The CN 115 shown in FIG. 1D may include at least one AMF 182a, 182b, at least one UPF 184a, 184b, at least one session management function (SMF) 183a, 183b, and at least one Data Network (DN) 185a, 185b. While each of the foregoing elements are depicted as part of the CN 115, it will be appreciated that any of these elements may be owned and/or operated by an entity other than the CN operator.

[0075] The AMF 182a, 182b may be connected to one or more of the gNBs 180a, 180b, 180c in the RAN 113 via an N2 interface and may serve as a control node. For example, the AMF 182a, 182b may be responsible for authenticating users of the WTRUs 102a, 102b, 102c, support for network slicing (e.g., handling of different protocol data unit (PDU) sessions with different requirements), selecting a particular SMF 183a, 183b, management of the registration area, termination of NAS signaling, mobility management, and the like. Network slicing may be used by the AMF 182a, 182b, e.g., to customize CN support for WTRUs 102a, 102b, 102c based on the types of services being utilized WTRUs 102a, 102b, 102c. For example, different network slices may be established for different use cases such as services relying on ultra-reliable low latency (URLLC) access, services relying on enhanced massive mobile broadband (eMBB) access, services for MTC access, and/or the like. The AMF 162 may provide a control plane function for switching between the RAN 113 and other RANs (not shown) that employ other radio technologies, such as LTE, LTE-A, LTE-A Pro, and/or non-3GPP access technologies such as Wi-Fi.

[0076] The SMF 183a, 183b may be connected to an AMF 182a, 182b in the CN 115 via an N11 interface. The SMF 183a, 183b may also be connected to a UPF 184a, 184b in the CN 115 via an N4 interface. The SMF 183a, 183b may select and control the UPF 184a, 184b and configure the routing of traffic through the UPF 184a, 184b. The SMF 183a, 183b may perform other functions, such as managing and allocating UE IP address, managing PDU sessions, controlling policy enforcement and QoS, providing downlink data notifications, and the like. A PDU session type may be IP-based, non-IP based, Ethernet-based, and the like.

[0077] The UPF 184a, 184b may be connected to one or more of the gNBs 180a, 180b, 180c in the RAN 113 via an N3 interface, which may provide the WTRUs 102a, 102b, 102c with access to packet-switched networks, such as the Internet 110, e.g., to facilitate communications between the WTRUs 102a, 102b, 102c and IP-enabled devices. The UPF 184, 184b may perform other functions, such as routing and forwarding

packets, enforcing user plane policies, supporting multi-homed PDU sessions, handling user plane QoS, buffering downlink packets, providing mobility anchoring, and the like.

[0078] The CN 115 may facilitate communications with other networks. For example, the CN 115 may include, or may communicate with, an IP gateway (e.g., an IP multimedia subsystem (IMS) server) that serves as an interface between the CN 115 and the PSTN 108. In addition, the CN 115 may provide the WTRUs 102a, 102b, 102c with access to the other networks 112, which may include other wired and/or wireless networks that are owned and/or operated by other service providers. In an embodiment, the WTRUs 102a, 102b, 102c may be connected to a local Data Network (DN) 185a, 185b through the UPF 184a, 184b via the N3 interface to the UPF 184a, 184b and an N6 interface between the UPF 184a, 184b and the DN 185a, 185b.

[0079] In view of FIGs. 1A-1D, and the corresponding description of FIGs. 1A-1D, one or more, or all, of the functions described herein with regard to any of: WTRUs 102a-d, base stations 114a-b, eNode-Bs 160a-c, MME 162, SGW 164, PGW 166, gNBs 180a-c, AMFs 182a-b, UPFs 184a-b, SMFs 183a-b, DNs 185a-b, and/or any other element(s)/device(s) described herein, may be performed by one or more emulation elements/devices (not shown). The emulation devices may be one or more devices configured to emulate one or more, or all, of the functions described herein. For example, the emulation devices may be used to test other devices and/or to simulate network and/or WTRU functions.

[0080] The emulation devices may be designed to implement one or more tests of other devices in a lab environment and/or in an operator network environment. For example, the one or more emulation devices may perform the one or more, or all, functions while being fully or partially implemented and/or deployed as part of a wired and/or wireless communication network in order to test other devices within the communication network. The one or more emulation devices may perform the one or more, or all, functions while being temporarily implemented/deployed as part of a wired and/or wireless communication network. The emulation device may be directly coupled to another device for purposes of testing and/or may performing testing using over-the-air wireless communications.

[0081] The one or more emulation devices may perform the one or more, including all, functions while not being implemented/deployed as part of a wired and/or wireless communication network. For example, the emulation devices may be utilized in a testing scenario in a testing laboratory and/or a non-deployed (e.g., testing) wired and/or wireless communication network in order to implement testing of one or more components. The one or more emulation devices may be test equipment. Direct RF coupling and/or wireless communications via RF circuitry (e.g., which may include one or more antennas) may be used by the emulation devices to transmit and/or receive data.

[0082] Introduction

[0083] The current state-of-art edge cloud infrastructure can be used to support offloaded Extended Reality (XR) applications. XR applications can be decomposed into tasks to facilitate offloading to the edge. In some examples, computational resources can be allocated to XR applications. An application's multiple dimensions may characterize its Quality of Service (QoS) in some cases.

[0084] As such, new or improved methods, procedures, and architectures may be desired to implement task distribution(s) and/or re-distribution(s) among a WTRU (or UE) and multiple Edge devices. For example, a WTRU and multiple Edge devices may need to dynamically agree on the distribution of tasks in the face of varying wireless conditions and fluctuations in the resource requirements of one or more running applications. In another example, information containing mapping information among multiple edge devices, tasks and resources may be exchanged between the WTRU and multiple edge devices.

[0085] Edge-Cloud infrastructure for XR applications

[0086] FIG. 2 shows how an end-to-end path between a WTRU (or a UE) and Cloud is segmented into three distinct tiers and used by current XR systems. Each tier has common design considerations relevant to that tier.

[0087] Tier 1 represents the three design considerations of compute elasticity, storage permanence and hardware consolidation. The compute elasticity design consideration requires that new machines can be quickly spun-up or old machines be powered down depending on the demand for their resources. The storage permanence design consideration requires redundancy of storage using for example Redundant Array of Independent Disks (RAID), a stability of infrastructure and practices that require data backup and disaster recovery. Finally, the hardware consolidation design consideration requires that operational costs be amortized over a large number of machines in a data center. Commercial examples of services that are provided using Tier 1 infrastructure include but are not limited to Amazon Web Services, Microsoft Azure, and the Google Cloud Platform.

[0088] Tier 2 represents the design consideration of network proximity. This consideration requires that mini data center with a small number of servers be provided closer to the location of UEs. This proximity is quantified in terms of low round-trip-times (RTT) and the availability of high bandwidth connectivity between the UE and a tier 2 server. Tier 2 mini data centers are also known as Edge Computing devices. Examples of devices used to provide tier 2 infrastructure include but not limited to mini data centers, luggable devices and vehicular devices.

[0089] Tier 3 represents the design considerations of mobility and sensing. The mobility design consideration requires that the devices used in this tier are mobile. This places constraints on the weight, size, heat dissipation and the battery life of such devices. The sensing design consideration requires that the device used at tier 3 should support sensors such as GPS, microphones, accelerometers, gyroscopes, and

video cameras. Notice that the tier 3 devices themselves might not be able to provide the necessary processing power necessary to analyze the data from the afore-mentioned sensors. Examples of tier 3 devices include smart phones, Augmented Reality/Virtual Reality (AR/VR) devices, drones, and sensors that are static or on vehicles.

[0090] XR Application Decomposition

[0091] In some embodiments, computationally intensive XR application may be decomposed so that some of the resulting tasks can be offloaded to Edge devices. This offloading can lead to reduced application latency, improve application performance, reduce the energy requirements of the mobile device and finally reduce the heat dissipated by the mobile device.

[0092] In order for an application to run across the two tiers of the UE and the edge described above, it needs to be decomposed into tasks that can run separately on the UE and the Edge. The Cloud is not included in our scheme as the latency required by AR/VR applications is less than 20*ms* making it infeasible for the computationally intensive tasks to be processed at the remote cloud. Rather that processing is done on the Edge in close network proximity to the UE.

[0093] There are various ways in which an XR application such as an AR/VR application can be decomposed into multiple tasks. In the first approach, the programmer can be asked to specify the possible ways of decomposing the application. In the second approach, modern language runtimes can be used to identify candidate decompositions at the granularity of methods. A hybrid approach can also be adopted which combines the two ways where the granularity of decomposition is at the method level and the programmer is asked to specify the methods that can be run on the edge. Finally, an application that has already been decomposed to run on distributed systems middleware can be run separately on the UE and the Edge using those decomposed components.

[0094] As an example, an XR application can be decomposed into the following tasks, some of which might run on the UE and others on an Edge device:

[0095] The user's actions in the game are captured by the **"User Interaction" task** and are sent as "User Commands" to a **"Command Interpretation" task** that receives the user commands from the client and converts them into "User Actions". Next, the "User Actions" are interpreted as changes in the real world by the **"AR/VR Logic" task**. These world changes are then converted into "Rendered Scene" by the "**GPU Rendering" task**. The "GPU Rendering" task then forwards the "Rendered Scene" to a **"Video Encoder" task** which encodes (including compression) the video and sends that video to the **"Video Streaming" task**. Finally, the "Video Streaming" task sends the "Encoded Video" as a "Video Stream" to a "Video Decoder" task that then decodes the video and displays it on the UE for the user.

[0096] Resource allocation for XR applications

[0097] Each decomposed task can require multiple resources. For example, the AR/VR set of tasks discussed above might require resources such as Central Processing Unit (CPU), network bandwidth, battery energy, file cache state, and memory.

[0098] In general, 'n' tasks can require 'm' system resources. Let R_j denote the set of allocation choices for the j^{th} shared resource (e.g., memory). Then the set of all possible resource allocation choices is:

$$R = R_1 \times R_2 \times \ldots\ldots \times R_m$$

[0099] The Quality Dimensions for an Application

[0100] An XR application has multiple quality dimensions. Each of those dimensions can in turn take multiple values. For example, XR requirements can be organized (but not limited to) along the following dimensions:

1) Cryptographic Security (encryption key-length, in bits): 0 (off), 56, 64, 128
2) Data Delivery Reliability, which could be:
   a) maximum packet loss: as a percentage of all packets
   b) expected packet loss: as a percentage of all packets
   c) packet loss occurrence: as a per packet probability of loss
3) Video related quality:
   a) Picture Format: SQCIF, QCIF, MPEG-1, MPEG-2, MPEG-4, MPEG-7, H.264, H.265, H.266.
   b) Color Depth(bits): 1, 3, 8, 16,24
      Black/white, grey scale to higher color
   c) Video Timeliness – frame rate (fps): 1,2, ……..30..
      Low-frame rate cartoon or animation to 3D
4) Audio related quality:
   a) Sampling rate (kHz): 8, 16, 24,44….
      AM, FM, DVD quality….
   b) Sample bit (bits): 8, 16…
   c) Audio timeliness- end-to-end delay (ms): 25, 50, 75, 100…

[0101] Each decomposed task Ti (e.g., GPU rendering task) can be associated with the following components for each of the quality dimensions:

1) Quality Space Q_i for example for the decomposed task T_i say "GPU Rendering" includes: Picture format, color depth, and video timeliness.

2) Quality Index defined as:

$f_{ij} : Q_{ij} \rightarrow \{1,2,\ldots.|Q_{ij}|\}$

where Q_{ij} is the j^{th} component of Q_i

For example, for the decomposed task "GPU Rendering":

Q_{i1} = Picture Format

Q_{i2} = Color Depth

Q_{i3} = Video timeliness (Frame rate)

In one embodiment, each of the Q_{1i}, Q_{2i}, and Q_{3i} can be constructed as shown below.

To construct Q_{i1} = Picture Format, each qualitative value may be mapped to an integer like so:

Picture Format	SQIF	QCIF	MPEG1	MPEG2	MPEG4	MPEG7	H.264	H.265	H.266
Quality Index	1	2	3	4	5	6	7	8	9

Similarly, to construct Q_{i2} = Color Depth, each integer value may be mapped to linear scale starting from 1:

Color Depth (bits)	1	3	8	16	24
Quality Index	1	2	3	4	5

Finally, to construct Q_{i3} = Video timeliness (Frame rate), the following mapping (the ellipses represent numbers from 3 to 29) may be applied:

Video Timeliness (Frame rate)	1	2	…	30
Quality Index	1	2	…	30

3) It may be defined that a dimension-wise utility value for a specify quality by setting $q_{ij} \in Q_{ij}$ for a decomposed task Ti on quality dimension j:

u_{ij}: Q_{ij} -> R

4) Similarly, an application utility can be associated with a decomposed task T_i:

u_i: Q_i -> R

5) A relationship between the set of resources 'R' and the quality space Q_i for a decomposed task may be defined:

r |-$_i$ q

[0102] This relation is a list of all resource allocation combinations that can be used to achieve each quality point 'q'. An R-U graph can be generated for each of the decomposed tasks T_i. Each R-U graph is a multi-dimensional step function. This can be represented as follows:

$$C_i = \begin{pmatrix} Ui1 & \cdots & Uik \\ \vdots & \ddots & \vdots \\ Ri1 & \cdots & Rik \end{pmatrix}$$

which represents the discrete set of utility-resource pairs and is the R-U graph.

[0103] Many Combination of resources can map to the same utility value for a decomposed task: For example, one might compress video files to reduce network bandwidth requirements, while using more CPU processing cycles to perform the compression/decompression operations. Alternatively, one could avoid compression which would require more network bandwidth, but would reduce the processing cycles needed. It is important to note that the two approaches result in different system resource requirements but an identical utility value. Thus:

- Many different combinations of resources required for the decomposed task "GPU rendering" mapping to a utility value of u_1 .
- Similarly many different combinations of resources required for the decomposed task "GPU rendering" mapping to a utility value of u_2 .
- Similarly many different combinations of resources required for the decomposed task "GPU rendering" mapping to a utility value of u_3 .
- And etc.

Note that all the above mappings of different resource combinations to different utility values is the R-U graph Ci. It has been defined above a Quality index Qi for a decomposed task Ti that has as its domain the set of vectors with each vector representing the specific value of the combination of resources.

[0104] In effect, the utility function is a multi-dimensional step function if it lets the range of utility function be the set of integers.

[0105] The utility-resource pairs of C_i above can be calculated as follows. To select out the most valued QoS point for each resource value, the following function can be defined:

$$g_i : R \rightarrow \mathbb{R},$$

Such that

$$g_i(r) = \max\{\, u_i(q) \mid r \models_i q \,\}$$

To extract out the quality points associated with the utility value $g_i(r)$ the following function can be used:

$$h_i(r) = \{\, q \in Q_i \mid u_i(q) = g_i(r) \wedge r \models_i q \,\}$$

This function can be used to generate a respective R-U graph for each decomposed task. Each respective R-U graph will be a multidimensional step function.

6) For the overall XR application such as AR/VR with multiple tasks each of which possibly requires multiple resources, a system utility function is defined:

$u: Q_1 \times Q_2 \times \ldots Q_n \rightarrow R$

Then, the system utility function for fair sharing can be defined as

$u = u^*$ where $u^*(q_1, q_2, \ldots q_n) = \min_{i=1..n}(u_i(q_i))$

7) Each decomposed task can specify minimum QoS requirement for each of its relevant QoS dimensions:

$q^{min}_i = (q^{min}_{i\,1}, q^{min}_{i2}, \ldots\ldots\ldots\ldots q^{min}_{idi})$

8) Finally, the mathematical formulation of the problem is as follows:

For a given set of decomposed tasks $T_1, T_2, T_3, \ldots\ldots T_n$ the problem is to assign qualities (q_i) and allocate resources (r_i) to the decomposed tasks such that the system utility 'u' is maximized:

Maximise $u(q_1, q_2, \ldots\ldots\ldots q_n)$

Subject to $q_i >= q^{min}_i$ for $q_i = 0$ and $I = 1,2,\ldots.n$ (QoS constraint)

$\sum r_{ij} <= r^{max}_j$, $j = 1,2,\ldots.n$ (Resource Contraints)

$r \vdash_i q_i$

9) There are many algorithms such as extended dynamic programming for multiple resources, CPLEX Integer Programming, and approximation algorithms that can solve the above problem.

[0106] As an example, one algorithm namely an approximation algorithm may use a local search heuristic.

[0107] As discussed above,

$$C_i = \left\langle \begin{matrix} Ui1 & \cdots & Uik \\ \vdots & \ddots & \vdots \\ Ri1 & \cdots & Rik \end{matrix} \right\rangle$$

represent the discrete set of utility-resource pairs and is the R-U graph. A penalty vector is defined to assign the cost of each resource combination:

$p = (p_1, p_2\ldots\ldots p_m)$ where $p_i \in [1, \infty)$

Using the above, a penalized resource vector can be defined:

$r_p = (r_1.p_1, r_2.p_2\ldots\ldots\ldots r_m.p_m)$

Finally, a scalar metric is defined for each penalized resource vector:

$$r^* = \|r_p\| = \sqrt{(r_{p_1})^2 + \cdots + (r_{p_m})^2}$$

Using r^*, C_i is enhanced as follows:

$$C_{ic} = \left\langle \begin{pmatrix} u_{i1} \\ r_{i1} \\ r^*_{i1} \end{pmatrix}, \ldots, \begin{pmatrix} u_{ik_i} \\ r_{ik_i} \\ r^*_{ik_i} \end{pmatrix} \right\rangle$$

[0108] As an example of an algorithm for resource allocation while considering multiple QoS values, consider the algorithm shown in FIG. 3. The input of this algorithm is the number of tasks n and the R-U graph C_i for each task T_i. The output of this algorithm assigns qualities q[i] and resources r*[i] to each task T_i. In this example, r^c denotes the current remaining resource capacity after some of the available resources have been allocated. To represent the task ids, their associated r-values and u-values of the associated tasks, s.list[i].t , s.list[i].r , s.list[i].u are used. To represent resources currently allocated to decomposed task T_i, r[i] is used.

[0109] Each element in C_{ic} has a compound resource component. The convex_hull_frontier function in the algorithm works on that component. To control the solution refinement steps the parameter ε can be set to different values along with the heuristic result from the procedure initial_penalty and adjust_penalty. The input of this algorithm is the number of tasks n and the R-U graph C_i for each task T_i. The output of this algorithm assigns qualities q[i] and resources r*[i] to each task T_i.

[0110] **Overview**

[0111] An XR system running on a terminal device may deal with changing characteristics of communication (e.g., latency, bandwidth), availability of system resources (e.g., CPU, GPU, TPU, memory) to support high QoS when running energy and computationally intensive XR tasks.

[0112] In this section, the problems with the current software architecture of XR systems running on a WTRU is explained. Next, the shortcomings in the current interaction patterns between WTRU and Edge, for example, in a dynamic-anycast architecture, are explained.

[0113] Some current software architectures may run on terminal devices to support the running of XR systems. We envisage a scenario where XR applications such as AR/VR, multi-player gaming applications are running on the UE. Such applications will need to re-distribute their XR based tasks at a rate that responds to the mobility of the user, to variation in the wireless link properties such as latency and bandwidth and to variation in the operating environment such as availability of resources at the Edge. This introduces complexity that the current XR system architectures do not address. In particular, current architectures such as Google Stadia , GeForce Now, PS Now, Outatime, and/or Kahawi do not support the ability to dynamically re-distribute tasks as the operating environment changes. Additionally, such re-distribution of tasks currently requires a human-in-the loop making the running of such a re-distribution infeasible. Current architectures also do not support an ability to re-distribute based on balancing between the current availability of resources and the current demand. Another problem is that the current architectures do not support dynamic selection of some information leaving out other data points as appropriate for the application to take a decision.

[0114] In some examples, a centralized control of redistribution of tasks may not be feasible as the operating conditions are local to a device and it can be costly because of high latency to send data to a centralized entity in the architecture. Moreover, with multiple participating devices each reacting to a change in its local environment, a centralized control of redistribution of tasks is highly inefficient. Finally, the UE and Edge devices will in general have different ownership and there will be no central entity to control the re-distribution of tasks.

[0115] In some other examples, a WTRU or UE can act as a centralized controller of redistribution of tasks as per its requirements. This is because the WTRU or UE is running an XR application and may have the latest information on the tasks that are running inefficiently.

[0116] There is an interaction pattern between the terminal device(s) and Edge device(s) to support XR systems. The UE and edge devices need to acquire the necessary information about the environment by monitoring and exchanging the appropriate parameters. None of the current solutions specify such parameters to support distribution of XR-based tasks. In addition, current solutions do not provide the mechanisms by which XR system could abstract and understand the environment by matching a monitored value to a state of the environment. Finally, XR systems need machinery to identify the actions that the UE and Edge should take to redistribute tasks which is not provided by current solutions.

[0117] For example, the dynamic-anycast architecture proposes an architecture to support the ability to deal with the dynamic nature of computation and network resources required by various applications such as XR systems. A protocol message structure is proposed, and some fields in this message are defined as follows:

- *Dynacast Service ID (DSID)*: This is an identifier representing a service that the clients use for accessing the necessary resources.
- *Dynacast Binding ID (DBID)*: This is an address to reach a service instance for a given DSID.
- *Metrics*: These are computing, network resources that are associated with DSID and BSID and can be specified as a single number or a tuple of numbers.

[0118] However, the details were proposed to be discussed in future work. As such, it is desired to have details of the parameters required for the signaling and the interaction pattern between the WTRU and the Edge.

[0119] In various embodiments, a WTRU (or a UE) may interact with one or more Edge devices. In some examples, the Cloud may not be included in a scheme discussed as the latency required by AR/VR applications is less than 20ms making it infeasible for the computationally intensive tasks to be processed at the remote cloud. Rather that processing is done on the Edge in close network proximity to the WTRU.

[0120] Architectural Components to Achieve Low Latency

[0121] In one embodiment, arrangement of software components is provided to enable a terminal device to run applications that deliver a low latency experience on a WTRU. In an example, the architecture of the proposed system that is running on a WTRU (or UE) and an Edge server/device to deliver low latency to XR applications is provided. The WTRU architecture as shown in FIG. 4 comprises the following components whose inputs and outputs are discussed below:

[0122] ***AR/VR Logic***: the "User Actions" by the user are interpreted as changes in the user view by the "AR/VR Logic" module of the UE. These changes can be carried as a data structure labelled "User View Changes" to the "Data Aggregation" or the "MPC Controller" module as shown in FIG. 4. In addition, these changes can be carried as a data structure labelled "Current User View Changes" in FIG. 4 to the "Decision Module".

[0123] ***GPU Rendering***: This component's one input can be "Current User View Changes" carried in a data structure as shown in FIG. 4. This data is forwarded by the "Decision Module" and is originally coming from the "AR/VR Logic" module on UE. Another input can be "Predicted User View Changes" carried in a data structure and forwarded by the "Decision Module". FIG. 4 shows that this second input can come from the "MPC Controller" component either on the UE or on the Edge device. This "GPU Rendering" component converts those data structures into appropriate video frames as an output.

[0124] ***Video Decoder***: This module can receive an "Encoded Video" Stream from the network and can decompress it in a video frame format appropriate for rendering a scene on the UE as shown in FIG. 4.

[0125] ***Server interaction Module***: This module provides services that include but are not limited to marshalling arguments into byte arrays and un-marshalling the byte array replies back into a suitable data format, providing authentication services, naming and directory services, programming abstractions etc.

[0126] ***Decision Module***: This module can accept as an input "Current User View Changes" carried in a data structure from the "AR/VR Logic" component on UE as shown in FIG. 4. In addition, as shown in FIG. 4, it can accept "Predicted User View Changes" carried in a data structure from either the "MPC Controller" component of UE or "MPC Controller" of the Edge device. Depending on the decision algorithm, both the "Current User View Changes" data structure and the "Predicted User View Changes" data structure can be output into the "GPU Rendering" module or a call can be made to "Server Interaction Module" procedures.

[0127] In one embodiment, the decision algorithm on the Decision Module selects one out of the following combination of choices: (a) Use low quality video rendering by UE's GPU Rendering module + use low quality prediction by UE's DNN; (b) Use low quality video rendering by UE's GPU Rendering module + use high quality prediction by Edge device's DNN; (c) Use low quality prediction by UE's DNN + use high quality video rendering by Edge device's GPU rendering module; (d) use high quality video rendering by Edge device's GPU rendering module + use high quality prediction by Edge device's DNN. This choice is made in coordination with the Decision Module on the Edge device on the basis of an exchange of telemetry data. In one embodiment, the telemetry data could be the Round-trip time (RTT) between the UE and the Edge device.

[0128] ***MPC Controller***: This module runs a Model-Predictive-Controller (MPC) algorithm which is a classical control policy. It takes predictions labelled "UE Neural Network Data" from the Deep Neural Network (DNN) trained predictor labelled "Trained Neural Network" in Figure 4 (e.g., this prediction could be a point estimate whereas in another embodiment, it could be a probability distribution) . These predictions can be used as an input to optimize an objective Quality of Experience (QoE) function. In one embodiment, the QoE is a function of the round-trip time (RTT). In addition, the MPC Controller can get updates of the user's moves as current "User View Changes" from the "AR/VR Logic" module. The final input can be updates to the numerical parameters of the control algorithm from the Edge device's MPC Controller labelled "Neural

Network Updates" in Figure 4. The output of the MPC Controller is a data structure carrying the "Predicted User View Changes" that can be sent to the "Decision Module".

[0129] ***Data Aggregation***: The module can collect "User View Changes" carried in a data structure from the UE's "AR/VR Logic" module and use this data to train the DNN labelled "Trained Neural Network" as shown in Figure 4. In one embodiment, this training is done once a day.

[0130] ***Trained Neural Network***: This component is a small DNN that can be run on a UE. However, because of its small size, this DNN can only produce approximate predictions based on the inputs from the "Data Aggregation" module of the UE discussed above. Its output is a prediction of the future state of the user's in the environment. In one embodiment, the predictions are in the form of a point estimate. In another embodiment, the predictions are in the form of a probability distribution. These approximate predictions labelled "UE Neural Network Data" in Figure 4 can be used on the UE itself by the MPC Controller. Alternatively, it can be sent to the Edge device using the "Server Interaction Module" procedures.

[0131] FIG.5 shows the components running on the Edge device. This architecture is composed of the following components whose inputs and outputs are discussed below:

[0132] ***GPU Rendering***: Its input can be a data structure carrying "Changes in the User View" or a data structure carrying "Predicted User View Changes" from the Edge's "Decision Module" and originally coming from UE (labelled as "Current/Predicted User View Changes" in Figure 5). Another input can be a data structure carrying "Predicted User View Changes" from the Edge's "MPC Controller" and triggered by the Edge's "Decision Module". The "GPU Rendering" module can convert the data structures into appropriate video frames as an output into the "Video Encoder" component of the Edge.

[0133] ***Client Interaction Module***: This component gets its inputs from the UE. These inputs fall into three categories as already discussed previously: "Current User View Changes", "Predicted User View Changes", and "UE Neural Network Data". This "Client Interaction module" can then forward these inputs to the "Decision Module". In addition, the "Current User View Changes" can also be forwarded to the Edge's "MPC Controller". The "Client Interaction Module" provides services that include but are not limited to marshalling arguments into byte arrays and un-marshalling the byte array replies back into a suitable data format, , authentication services, naming and directory services, programming abstractions etc.

[0134] ***MPC Controller***: This module runs a Model-Predictive-Controller (MPC) algorithm which is a classical control policy. It takes predictions labelled "DNN data from UE" in FIG. 5 from the Deep Neural Network (DNN) trained predictor "Trained Neural Network" on the UE (FIG. 4) which is a low quality prediction. Alternatively, it takes predictions as shown in FIG. 5 from the Deep Neural Network (DNN) trained predictor "Trained Neural network" on the Edge device which combines the low quality prediction from UEs "Trained Neural Network" with Edge's own "Trained Neural Network". The decision to choose the prediction input from these two choices is made by the "Decision Module" on the Edge device discussed below.

[0135] In one embodiment, this prediction could be a point estimate whereas in another embodiment, it could be a probability distribution) as an input to optimize an objective Quality of Experience (QoE) function. In one embodiment, the QoE is a function of the round-trip time (RTT).

[0136] In addition, as shown in FIG.5, the MPC Controller can get updates of the user's view as "Current User View Changes" carried in a data structure from the "AR/VR Logic" module on the UE and sent to the Edge device's "Server Interaction Module" from UE's "Client Interaction Module".

[0137] One output of an MPC Controller can be a data structure carrying the "Predicted User View Changes" that is sent to the "GPU Rendering" component on the Edge device as shown in FIG. 5. The second output can be updated to the numerical parameters of the control algorithm from the Edge device's "MPC Controller" to the UE's "MPC Controller". The final output can be a data structure carrying "Predicted User View Changes" that is sent to the UE via a procedure of the Edge's "Client Interaction Module" as shown in FIG. 5.

[0138] ***Video Encoder***: Its input can be appropriate video frames to be compressed from the "GPU Rendering" module of the Edge. Its output can be a set of compressed video frames using an appropriate encoding.

[0139] ***Video Streaming Module***: This component's input can be a set of compressed video frames using an appropriate encoding. Its output can be a video stream appropriate for transport over the network.

[0140] ***Decision Module***: This module can accept the following inputs from the "Client Interaction Module" of the Edge device: The first input can be "Current User View Changes" carried in a data structure from UE, the second input can be "Predicted User View Changes" carried in a data structure and the third input that can be accepted is the "UE Neural Network Data" which in one embodiment could be a point estimate and in another a probability distribution.

[0141] The outputs are as follows: The module can forward the "Current User View Changes" as a data structure carrying status updates to the "GPU Rendering" module or the "Data Aggregation" module; The module can also forward "Predicted User View Changes" carried in a data structure to the "GPU Rendering" module on the Edge Device. Another output can be the "DNN Data from UE" (which in one embodiment could be a point estimate and in another a probability distribution) to the DNN labelled "Trained Neural Network" in FIG. 5. This same "DNN Data from UE" can also be forwarded to the "MPC Controller" component on the Edge Device as shown in FIG. 5.

[0142] In one embodiment, the decision algorithm on the Decision Module selects one out of the following combination of choices: (a) Use low quality video rendering by UE's GPU Rendering module + use low quality prediction by UE's DNN; (b) Use low quality video rendering by UE's GPU Rendering module + use high quality prediction by Edge device's DNN; (c) Use low quality prediction by UE's DNN + use high quality video rendering by Edge device's GPU rendering module; (d) Use high quality video rendering by Edge device's

GPU rendering module + use high quality prediction by Edge device's DNN. This choice is made in coordination with the Decision Module on the UE on the basis of an exchange of telemetry data. In one embodiment, the telemetry data could be the Round-trip time (RTT) between the UE and the Edge device.

[0143] ***Data Aggregator***: This module can collect current user view changes labelled as "Data for Training" in Figure 5 and carried in a data structure from the Edge device's "Decision Module". It can use this data to train the DNN labelled "Trained Neural Network" in FIG. 5. In one embodiment, this training is done once a day.

[0144] ***Trained Neural Network***: This is a DNN with the following inputs: the first input can be data for training the DNN from the "Data Aggregator" as discussed above. In one embodiment, this training is done once a day. The second input can be "DNN Data from UE" forwarded by the "Decision Module" as shown on FIG. 5. This second input originally coming from the UE's smaller DNN in one embodiment could be an approximate point estimate and in another an approximate probability distribution. The output of this DNN is a prediction which in one embodiment could be an accurate point estimate and in another an accurate probability distribution.

[0145] **Representative procedure and software architecture that run on terminal device(s) and/or Edge device(s) to support XR applications**

[0146] In one embodiment, FIG. 6 shows a proposed architecture on a WTRU (or UE). We have discussed the UE Decision module and the UE Server Interaction Module in the previous section. Referring to FIG. 6, we show how the UE's Decision Module is layered upon the UE Server Interaction module. This UE Decision module depends on services provided by the layers of the "UE Server Interaction Module".

[0147] Our proposal envisages this architecture running on the participating WTRU (or UE) and Edge devices to enable a dynamic redistribution of tasks at a rate that responds to the mobility of the user, to variation in the wireless link properties such as latency and bandwidth and to variation in the operating environment such as availability of resources at the Edge . There is no centralized control of this redistribution of tasks.

[0148] In this example, the functionality of each of the layers of this architecture and how the functionality maps to the problems discussed previously are discussed.

[0149] ***Task Dependency View***: This module maintains a data structure to represent the precedence relations between the distributed tasks.

[0150] ***Task Execution***: This module represents a set of programming modules (e.g. Classes) that implement the tasks running on the device.

[0151] ***UE Decision Module***: This module runs distributed algorithms to support the ability of the architecture to dynamically re-distribute tasks as the operating environment changes. These distributed algorithms ensure that a human-in-the loop is not required for taking an appropriate decision. These

algorithms re-distribute tasks based on balancing between the current availability of resources and the current demand. In addition, the module runs distributed algorithms that support a dynamic selection of some information leaving out other data points as appropriate for the application to take a decision. To run these distributed algorithms, the module interacts with the "UE Decision Module" running on all the other devices by exchanging messages that follow an interaction pattern imposed by our proposed protocol. The interaction pattern and the associated protocol will be addressed later.

[0152] As discussed above, the "UE Decision Module" requires support from the UE Server Interaction Module. In particular, the layers that constitute the "UE Server Interaction Module" in FIG. 6 ensure that the "UE Decision Module" is notified when devices join or leave the running application. In addition, the layers help maintain a consistent state of the current dependencies between the tasks. The layers also enable the "UE Decision Module" to perform a consistent re-distribution of tasks across all the participating UEs and Edge devices. Finally, the layers provide a marshalling/unmarshalling service.

[0153] **Layers of the "UE Server Interaction Module"**

[0154] ***Resource Sharing Manager***: The purpose of this layer is to ensure that the shared resources are accessed in a consistent manner when concurrent requests are made.

[0155] ***Event Dissemination manager***: This layer ensures that events such as device failures, communication failures, new device addition are disseminated in the face of volatile communication links.

[0156] ***Device Volatility Manager***: This layer ensures that a consistent list of devices that are participating in the application is maintained in the face of device and communication failures and addition of new devices.

[0157] ***Marshalling/Unmarshalling Manager***: This layer provides a marshalling service which is the translation of structured data items and primitive data values to an external data message format. In addition, this layer provides an unmarshalling service which consists of the generation of primitive values from their external data representation and the rebuilding of the data structures.

[0158] In another embodiment, a similar architecture runs on the Edge server as shown in FIG. 7. Referring to FIG. 7, we show how the Server's Decision Module is layered upon the Server UE Interaction module. This Server Decision module depends on services provided by the layers of the "Server UE Interaction Module". The layers shown in the edge server have similar functionalities as the corresponding layers on the UE.

[0159] **Representative procedure and interaction pattern between the terminal device(s) and Edge device(s) to support XR applications**

[0160] In this section, we first show how the components in the UE and Edge can be placed in the 3GPP system. Next, we explain the novel interaction pattern between the UE and the Edge device to support the running of XR systems.

[0161] Mapping of the UE and Edge components to 3GPP architecture

[0162] The 3GPP document TR 26.928 presents an architecture to support Extended Reality (XR) as shown in FIG. 8. XR is defined in that document as follows: "Extended reality (XR) refers to all real-and-virtual combined environments and associated human-machine interactions generated by computer technology and wearables. It includes representative forms such as augmented reality (AR), mixed reality (MR), and virtual reality (VR) and the areas interpolated among them."

[0163] The AR/VR low-latency requirements are discussed in that document. In particular, the document specifies that 15ms- 20ms are more accurate estimates of acceptable latency in XR applications where the user is moving.

[0164] In this section, we show how our proposals can be used to extend the 3GPP proposals. Referring to FIG. 8, the "5G-XR" client is the receiver of 5G-XR session data that may be accessed through well-defined interfaces/APIs by the 5G-XR Aware Application. This client accesses the 5G-XR AF through the X5 interface (as shown in FIG. 8). This 5G-XR AF is an Application Function similar as defined in 3GPP TS 23.501, clause 6.2.10, dedicated to 5G-XR Services. In addition, the 5G-XR client accesses 5G-XR AS through the X4 interface (as shown in FIG. 8). 5G-XR AS is an Application Server dedicated to 5G-XR Services.

[0165] FIG. 9 shows how our proposed components can be used to extend the functionality of the architecture proposed by the 3GPP TR 26.928. The "5G-XR Client" can be extended by including "Server Interaction Module", "GPU Rendering", and "Video Decoder" components of in FIG. 4.

[0166] According to 3GPP TR 26.928, AI components are considered specific to the particular XR applications for which they are created. As a result, as shown in FIG. 9, "MPC Controller", Data Aggregation" and "Trained Neural Network" can be part of the 5G-XR Application in this embodiment. In addition, "AR/VR Logic" and the "Decision Module" are also a part of the application.

[0167] Similarly, the Trusted DN (Data Network) module can play the role of "Client Interaction Module" as shown in FIG. 9. In addition, the "GPU Rendering", "Video Encoder" and "Video Streaming" are a part of the trusted DN in this embodiment.

[0168] "Decision Module", "MPC Controller", "Data Aggregation", and "Trained Neural Network" components can extend the functionality of the 5G-XR Application Provider's "External Data Network (DN)" as shown in FIG. 9.

[0169] FIG. 10 shows our embodiment of 3GPP 26.928 in more detail. The "XR Engine" in the 3GPP architecture can play the role of "GPU Rendering" and "Video Decoder". The "XR Session Handler" can play the role of our "Server Interaction Module". Finally, the "5GXR Aware Application" can embody our components "AR/VR Logic", "Decision Module", "MPC Controller", "Data Aggregation", and "Trained Neural Network".

[0170] On the server side as shown in FIG. 10, the "5G XR AS" module can embody our components "GPU Rendering" and "Video Encoder" and "Video Streaming"; the "5G XR AF" module can embody our

"Client Interaction Module" and finally the "5GXR Application Provider" module can embody our components "Decision Module", "MPC Controller", "Data Aggregation", and "Trained Neural Network".

[0171] Although the embodiments in FIGs. 8-10 focuses on 5G networks, the proposed ideas in this disclosure can be similarly applied to beyond 5G and 6G cellular networks.

[0172] The interaction pattern to support dynamic agreement to re-distribute XR tasks

[0173] The interaction pattern between a WTRU (or UE) and multiple Edge servers can be organized along the following two dimensions.

[0174] The first one is that the UE sends a combination of resource requirements to 'n' edge servers and only 'm' out of 'n' edge servers accept the offer. The rest counter-propose another combination of resources that is actually available with them. This is parametrized by the value of the system utility function, a list of tasks, the associated quality values that the Edge is required to do and a DeviceID-Task- Resource Map that specifies which Edge device need to provide what resources.

[0175] The second one is that the UE sends a combination of resource requirements to 'm' edge servers and all the edge servers are able to fulfill the requirements. This is also parametrized by the value of the system utility function, a list of tasks, the associated quality values that the Edge is required to do and a DeviceID-Task- Resource Map that specifies which Edge device need to provide what resources.

[0176] In an example, each of the two dimensions in turn is discussed in Case 1 and Case 2 below in details. For example, details of the parameters required for the signaling and the interaction pattern between the UE and the Edge devices for each case are provided.

[0177] ***Case 1***:

[0178] Procedure ***res_comb_req***: for the procedure res_comb_req. for resource combination request:

[0179] It is assumed that there are 'n' edge servers. The UE sends a combination of resource requirements to each of those 'n' edge servers. It is considered the case where 'm' out of 'n' edge servers can fulfill the resource request partially. These 'm' edge servers therefore make a counteroffer with the resource availability that is different(perhaps lower) than that requested by the UE.

[0180] This procedure res_comb_req, requires multiple values that are each an instance of a class that represents an Edge server. This class provides procedures for programmatically accessing the Internet address and port of an Edge machine. We name this class "EdgeRef" for the purpose of our discussion in this document, but it could be named anything as per the implementation's convenience. These multiple values of 'EdgeRef' are sent as an ordered vector.

[0181] The procedure res_comb_req also requires a parameter that identifies the value of the system utility function for which the resource combination requirement is being negotiated between the UE and the Edge server.

[0182] Finally, the procedure res_comb_req needs as a set of parameters the following: a list of tasks, their associated quality value, their associated device ID for the device on which the specific task is to run, and their associated vector containing the actual resource allocations that the associated Edge device is required to do for the associated task. In one embodiment, these parameters could be sent as a byte array. Note that each ordered EdgeRef as described above is mapped to its corresponding list of tasks, their associated quality values, their associated device ID for the device on which the specific task is to run and their associated vector containing the actual resources allocations that the associated Edge device is required to provide for the associated task.

[0183] The Server Interaction Module provides to the caller of res_comb_req procedure a service that marshals arguments into byte arrays and un-marshals the byte array replies back into a suitable data format.

[0184] Note that the procedure res_comb_req does not execute locally within a device. Rather, res_comb_req needs to exchange messages with external devices (e.g. edge servers) to complete its execution.

[0185] ***Procedure getStateOfComputation***:

[0186] This procedure getStateofComputation, requires as a parameter an instance of a class that represents an Edge server. This class provides procedures for programmatically accessing the Internet address and port of an Edge machine. We name this class "EdgeRef" for the purpose of our discussion in this document, but it could be named anything as per the implementation's convenience.

[0187] The procedure getStateOfComputation also requires a parameter that identifies the operation that needs to be invoked. This could be encoded as an Integer.

[0188] Finally, the procedure getStateOfComputation sends the device ID of the containers, microservices, VMs or a combination thereof , running on the corresponding Edge device on which the operation needs to be invoked. For the rest of this document, whenever we use 'VM', it should be read to mean containers, microservices, VMs or a combination thereof. If there is no VM running, this parameter is null.

[0189] Note that the procedure getStateOfComputation does not execute locally within a device. Rather, getStateOfComputation needs to exchange messages with external devices (e.g. edge servers) to complete its execution.

[0190] ***Procedure sendStateOfComputation***:

[0191] This procedure *sendStateofComputation*, requires as a parameter an instance of a class that represents a UE. This class provides procedures for programmatically accessing the Internet address and port of a client machine. We name this class "UE" for the purpose of our discussion in this document, but it could be named anything as per the implementation's convenience.

[0192] The procedure *sendStateOfComputation* also requires a parameter that identifies the operation that needs to be invoked. This could be encoded as an Integer.

[0193] Finally, the procedure sendStateOfComputation sends the state of computation encoded for example as a HashMap along with the device ID of the VM running on the corresponding Edge device which is sending its state of computation. If there is no VM running, this parameter is null.

[0194] Note that the procedure sendStateOfComputation does not execute locally within a device. Rather, sendStateOfComputation needs to exchange messages with external devices (e.g. UEs) to complete its execution.

[0195] Referring to FIG. 11 and FIG. 12, sequence of messages are illustrated that the system needs to support after the execution of the ***res_comb_req*** procedure for Case 1. This figure shows how our procedure can be used to implement a scenario where the UE and multiple Edge devices need to dynamically agree on the distribution of tasks in the face of varying wireless conditions and the fluctuations in the resource requirements of the running application.

[0196] Exemplary steps shown in FIG. 11 include one or more of the following.

[0197] In step 1, it is triggered when the Decision Module wants to remove an underperforming device (say an Edge Server) that is unable to meet the quality level of the task assigned to it. The Decision Module running on the UE sends a query getStateOfComputation to the underperforming device requesting the current state of computation. This query has the underperforming device's EdgeRef and the Device ID of the VM as a parameter (discussed above).

[0198] Next, in step 2 the arguments of the getStateofComputation operation are marshaled into byte arrays by the UE's "UE Server Interaction" component and sent over the network to the Edge device's "Edge Client Interaction" component. These arguments have been discussed above.

[0199] In step 3, the "Edge Client Interaction Module" at the Edge server unmarshalls the byte arrays into appropriate data structures and sends them to the "Edge Decision Module" of the Edge device.

[0200] In step 4, the Edge Decision Module of the Edge device that is underperforming calculates the current state of the computation and encodes it in a data structure for example a HashMap. In addition, as discussed above the device ID of the VM is also sent.

[0201] In step 5, these data structure arguments are marshaled into byte arrays by the Edge's "Edge Client Interaction" component and sent over the network to the client device's "UE Server Interaction" component.

[0202] In step 6, the "UE Server Interaction Module" at the UE unmarshalls the byte arrays into appropriate data structures and sends them to the "UE Decision Module" of the UE. This data structure encodes the state of the computation in the form of, for example, a HashMap. In addition, as discussed above the device ID of the VM is also received.

[0203] In step 7, the UE Decision Module calls the Device Volatility Manager on UE to compute the list of current devices. The UE's Device Volatility Manager in turn, exchanges messages with Multiple Edge Servers each of which is running its own Device Volatility Manager as part of their Edge Client Interaction Module. This updates the UE Device Volatility Manager's list of current devices. This list is then returned to the UE's Decision Module.

[0204] Exemplary steps shown in FIG. 12 include one or more of the following.

[0205] As shown in FIG. 12, in step 8, the UE Decision Module calculates the optimized resource combination requirement for the set of Edge Devices. For example, there might be 'n' devices some of which might be edge servers and others might be VMs running on some of the Edge servers. The UE then calls the res_comb_req procedure provided by the UE's "Server Interaction Module" to the UE's "UE Decision Module" component to send the request to the multiple Edge server for a list of tasks, their associated quality value, their associated device ID for the VM on which the specific task is to run, and their associated vector containing the actual resource allocations that the associated Edge device is required to do for the associated task. The UE's value of the system utility function for this particular combination of resources is also sent.

[0206] Next, in step 9 the arguments of the res_comb_req operation are marshaled into byte arrays by the UE's "UE Server Interaction" component and sent over the network to the Edge device's "Edge Client Interaction" component. These arguments have been discussed above.

[0207] In step 10, the "Edge Client Interaction Module" at the Edge server unmarshalls the byte arrays into appropriate data structures and sends them to the "Edge Decision Module" of the Edge device.

[0208] In step 11, some of the edge devices are unable to meet the resource combination (say n-m devices) request of the UE after calculating the availability of the resource combination. As a result, some of the edge device (say 'm' devices) make a counter-offer to the UE of a combination of resources that are available at those edge devices. The "Edge Decision Module" calls the procedure counter_comb_avail at each of the multiple edge devices with the system utility function value and the list of tasks, their resource combination requirements and their associated quality values, the reference to itself 'EdgeRef' and device ID(s) of the VM. This procedure is provided to the "Edge Decision Module" by the "Edge Client Interaction" module. The re-negotiation is for the value of the utility function corresponding to a combination of resources. So, if internally, the edge server decides to take a running service and downgrade it AND that results in a new value for the utility function, it will trigger a re-negotiation. This utility-based re-negotiation is an example of a scenario when this step is executed.

[0209] In step 12, the arguments of the counter_comb_avail operation are marshaled into byte arrays by the UE's "UE Server Interaction" component and sent over the network to the Edge device's "Edge Client Interaction" component. These arguments have been discussed above.

[0210] In step 13, the "UE Server Interaction Module" at the UE unmarshalls the byte arrays into appropriate data structures and sends them to the "UE Decision Module" of the UE. The UE calculates that only 'm' out of 'n' devices have sent counter-proposals.

[0211] In step 14 in FIG. 12, the UE Decision Module calculates the optimized resource combination requirement for the set of Edge Devices. For example, there might be 'm' devices some of which might be edge servers and others might be VMs running on some of the Edge servers. The UE then calls the res_comb_req procedure provided by the UE's "Server Interaction Module" to the UE's "UE Decision Module" component to send the request to the multiple Edge server for a list of tasks, their associated quality value, their associated device ID for the device on which the specific task is to run, and their associated vector containing the actual resource allocations that the associated Edge device is required to do for the associated task. The UE's value of the system utility function for this particular combination of resources is also sent.

[0212] Next, in step 15, the arguments of the res_comb_req operation are marshaled into byte arrays by the UE's "UE Server Interaction" component and sent over the network to multiple Edge device's "Edge Client Interaction" component. These arguments have been discussed above.

[0213] In step 16, the "Edge Client Interaction Module" at the multiple Edge servers unmarshalls the byte arrays into appropriate data structures and sends them to the "Edge Decision Module" of the Edge devices.

[0214] The various Edge Decision modules forwards their acceptance of the UE offer to the Edge's "Edge Client Interaction" component in step 17.

[0215] Next, in step 18, the data structure arguments encapsulating the various Edge device's acceptance are marshaled into byte arrays by the respective Edge's "Edge Client Interaction" component and sent over the network to the client device's "UE Server Interaction" component.

[0216] Finally, in step 19, the UE receives the acceptance of its offer at its "UE Decision Module" when its "UE Server Interaction" component unmarshalls the byte array received over the network from multiple edge devices.

[0217] In various embodiments, the message sequence described herein may be independent of any algorithm running in the Decision Module on the UE and/or the Edge devices.

[0218] ***Case 2***:

[0219] For the procedure **res_comb_req**. for resource combination request: Now, we consider the case where the UE sends a combination of resource requirements to multiple edge servers and unlike Case 1, the edge server is able to fulfill the requirements. This is also parametrized by the value of the system utility function and a list of tasks with associated quality values that the Edge is required to do.

[0220] For Case 2, in one embodiment, the steps 1 through 7 are same (or similar) as those steps in FIG. 11. FIG. 13 illustrates the remaining steps of a sequence of messages that the system needs to support after the execution of the res_comb_req procedure for Case 2.

[0221] In step 8, as shown in FIG. 13, the UE Decision Module calculates the optimized resource combination requirement for the set of Edge Devices. For example, there might be 'n' devices, some of which might be edge servers and others might be VMs running on some of the Edge servers. The UE then calls the res_comb_req procedure provided by the UE's "Server Interaction Module" to the UE's "UE Decision Module" component to send the request to the multiple Edge server for a list of tasks, their associated quality value, their associated device ID for the VM on which the specific task is to run, and their associated vector containing the actual resource allocations that the associated Edge device is required to do for the associated task. The UE's value of the system utility function for this particular combination of resources is also sent.

[0222] Next, in step 9, the arguments of the res_comb_req operation are marshaled into byte arrays by the UE's "UE Server Interaction" component and sent over the network to the Edge device's "Edge Client Interaction" component. These arguments have been discussed above.

[0223] In step 10, the "Edge Client Interaction Module" at the Edge server unmarshalls the byte arrays into appropriate data structures and sends them to the "Edge Decision Module" of the Edge device.

[0224] In step 11, all of the edge devices are able to meet the resource combination request of the UE after calculating the availability of the resource combination. The "Edge Decision Module" calls the procedure counter_comb_avail at each of the multiple edge devices with the system utility function value which is the same value as sent by UE and the list of tasks, their resource combination requirements and their associated quality values, the reference to itself 'EdgeRef' and device ID(s) of the VM. This procedure is provided to the "Edge Decision Module" by the "Edge Client Interaction" module.

[0225] Next, in step 12, the arguments of the counter_comb_avail operation are marshaled into byte arrays by the UE's "UE Server Interaction" component and sent over the network to the Edge device's "Edge Client Interaction" component. These arguments have been discussed above.

[0226] In step 13, the "UE Server Interaction Module" at the UE unmarshalls the byte arrays from all 'n' device into appropriate data structures and sends them to the "UE Decision Module" of the UE.

[0227] An Example of Protocol Packet Structure

[0228] The packet structure for the communication between the UE and the Edge is shown in FIG. 14. The first field labelled "Data ID" is used to indicate if the packet is being sent from UE or the Edge. The field labelled "Sequence Number" indicates the identifier of the message.

[0229] The field labelled "Remote Device ref" is an instance belonging to the class RemoteDevice. An object belonging to this class has methods that return the internet address and port of the remote device such as an Edge device.

[0230] The field labelled "Utility Value for resource combination" identifies the value of the utility function for which the resource combination requirement is being negotiated between the UE and the Edge server.

[0231] Finally, the last field is an ordered "list of tasks", quality values, device IDs and resource value vectors. In one embodiment, these parameters could be sent as a byte array.

[0232] The fields and data of our proposed packet can be mapped to SDP "Offer" procedure's capability negotiation parameters contained in an SIP protocol's UPDATE method.

[0233] To add new capabilities to SDP, we need to define a new attribute type "a=" at the session level. Attributes at the session level are listed before the first media line in SDP.

[0234] The following shows a proposed new SDP offer procedure's capability attribute:

a=res_comb_req:utility_value task1 quality_value1 deviceID1 resource_value_vector1 task2 quality_value2 deviceID2 resource_value_vector2……taskN quality_valueN deviceIDN resource_value_vectorN

Here "a=res_comb_req:" is the new attribute that that has multiple parameters separated by space. In our proposal, the first parameter is the value of the system utility function. The next set of parameters is a list of tasks, their associated quality values the deviceIDs of VMs and the vector containing the actual resource allocations that the Edge is required to do.

[0235] For example, as discussed above, an XR application can be decomposed into tasks as follows: The user's actions in the game are captured by the "User Interaction" task and are sent as "User Commands" to a "Command Interpretation" task that receives the user commands from the client and converts them into "User Actions". Next, the "User Actions" are interpreted as changes in the real world by the "AR/VR Logic" task. These world changes are then converted into "Rendered Scene" by the "GPU Rendering" task. The "GPU Rendering" task then forwards the "Rendered Scene" to a "Video Encoder" task which encodes (including compression) the video and sends that video to the "Video Streaming" task. Finally, the "Video Streaming" task sends the "Encoded Video" as a "Video Stream" to a "Video Decoder" task that then decodes the video and displays it on the UE for the user.

[0236] In various embodiments, the scope of the methods, procedures and architecture described/proposed herein includes direct (or indirect) applicability to one or more AR/VR use cases in addition to cloud gaming applications.

[0237] Referring to FIG. 15, an example of a packet structure with an SDP message is provided. In this example, the SDP message is nested in the protocol stack as a payload of the SIP packet. The SIP packet itself is carried over the Datagram Transport Layer Security (DTLS) protocol packet. This DTLS payload is in turn carried over UDP packet. Finally, the UDP packet is carried as a payload over an IP packet.

[0238] Conclusion

[0239] Although features and elements are provided 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. The present disclosure is not to be limited in terms of the particular embodiments

described in this application, which are intended as illustrations of various aspects. Many modifications and variations may be made without departing from its spirit and scope, as will be apparent to those skilled in the art. No element, act, or instruction used in the description of the present application should be construed as critical or essential to the invention unless explicitly provided as such. Functionally equivalent methods and apparatuses within the scope of the disclosure, in addition to those enumerated herein, will be apparent to those skilled in the art from the foregoing descriptions. Such modifications and variations are intended to fall within the scope of the appended claims. The present disclosure is to be limited only by the terms of the appended claims, along with the full scope of equivalents to which such claims are entitled. It is to be understood that this disclosure is not limited to particular methods or systems.

[0240] The foregoing embodiments are discussed, for simplicity, with regard to the terminology and structure of infrared capable devices, i.e., infrared emitters and receivers. However, the embodiments discussed are not limited to these systems but may be applied to other systems that use other forms of electromagnetic waves or non-electromagnetic waves such as acoustic waves.

[0241] It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to be limiting. As used herein, the term "video" or the term "imagery" may mean any of a snapshot, single image and/or multiple images displayed over a time basis. As another example, when referred to herein, the terms "user equipment" and its abbreviation "UE", the term "remote" and/or the terms "head mounted display" or its abbreviation "HMD" may mean or include (i) a wireless transmit and/or receive unit (WTRU); (ii) any of a number of embodiments of a WTRU; (iii) a wireless-capable and/or wired-capable (e.g., tetherable) device configured with, inter alia, some or all structures and functionality of a WTRU; (iii) a wireless-capable and/or wired-capable device configured with less than all structures and functionality of a WTRU; or (iv) the like. Details of an example WTRU, which may be representative of any WTRU recited herein, are provided herein with respect to FIGs. 1A-1D. As another example, various disclosed embodiments herein *supra* and *infra* are described as utilizing a head mounted display. Those skilled in the art will recognize that a device other than the head mounted display may be utilized and some or all of the disclosure and various disclosed embodiments can be modified accordingly without undue experimentation. Examples of such other device may include a drone or other device configured to stream information for providing the adapted reality experience.

[0242] In addition, the methods provided 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.

[0243] Variations of the method, apparatus and system provided above are possible without departing from the scope of the invention. In view of the wide variety of embodiments that can be applied, it should be understood that the illustrated embodiments are examples only, and should not be taken as limiting the scope of the following claims. For instance, the embodiments provided herein include handheld devices, which may include or be utilized with any appropriate voltage source, such as a battery and the like, providing any appropriate voltage.

[0244] Moreover, in the embodiments provided above, processing platforms, computing systems, controllers, and other devices that include processors are noted. These devices may include at least one Central Processing Unit ("CPU") and memory. In accordance with the practices of persons skilled in the art of computer programming, reference to acts and symbolic representations of operations or instructions may be performed by the various CPUs and memories. Such acts and operations or instructions may be referred to as being "executed," "computer executed" or "CPU executed."

[0245] One of ordinary skill in the art will appreciate that the acts and symbolically represented operations or instructions include the manipulation of electrical signals by the CPU. An electrical system represents data bits that can cause a resulting transformation or reduction of the electrical signals and the maintenance of data bits at memory locations in a memory system to thereby reconfigure or otherwise alter the CPU's operation, as well as other processing of signals. The memory locations where data bits are maintained are physical locations that have particular electrical, magnetic, optical, or organic properties corresponding to or representative of the data bits. It should be understood that the embodiments are not limited to the above-mentioned platforms or CPUs and that other platforms and CPUs may support the provided methods.

[0246] The data bits may also be maintained on a computer readable medium including magnetic disks, optical disks, and any other volatile (e.g., Random Access Memory (RAM)) or non-volatile (e.g., Read-Only Memory (ROM)) mass storage system readable by the CPU. The computer readable medium may include cooperating or interconnected computer readable medium, which exist exclusively on the processing system or are distributed among multiple interconnected processing systems that may be local or remote to the processing system. It should be understood that the embodiments are not limited to the above-mentioned memories and that other platforms and memories may support the provided methods.

[0247] In an illustrative embodiment, any of the operations, processes, etc. described herein may be implemented as computer-readable instructions stored on a computer-readable medium. The computer-readable instructions may be executed by a processor of a mobile unit, a network element, and/or any other computing device.

[0248] There is little distinction left between hardware and software implementations of aspects of systems. The use of hardware or software is generally (but not always, in that in certain contexts the choice between hardware and software may become significant) a design choice representing cost versus efficiency tradeoffs. There may be various vehicles by which processes and/or systems and/or other technologies described herein may be affected (e.g., hardware, software, and/or firmware), and the preferred vehicle may vary with the context in which the processes and/or systems and/or other technologies are deployed. For example, if an implementer determines that speed and accuracy are paramount, the implementer may opt for a mainly hardware and/or firmware vehicle. If flexibility is paramount, the implementer may opt for a mainly software implementation. Alternatively, the implementer may opt for some combination of hardware, software, and/or firmware.

[0249] The foregoing detailed description has set forth various embodiments of the devices and/or processes via the use of block diagrams, flowcharts, and/or examples. Insofar as such block diagrams, flowcharts, and/or examples include one or more functions and/or operations, it will be understood by those within the art that each function and/or operation within such block diagrams, flowcharts, or examples may be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof. In an embodiment, several portions of the subject matter described herein may be implemented via Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), digital signal processors (DSPs), and/or other integrated formats. However, those skilled in the art will recognize that some aspects of the embodiments disclosed herein, in whole or in part, may be equivalently implemented in integrated circuits, as one or more computer programs running on one or more computers (e.g., as one or more programs running on one or more computer systems), as one or more programs running on one or more processors (e.g., as one or more programs running on one or more microprocessors), as firmware, or as virtually any combination thereof, and that designing the circuitry and/or writing the code for the software and or firmware would be well within the skill of one of skill in the art in light of this disclosure. In addition, those skilled in the art will appreciate that the mechanisms of the subject matter described herein may be distributed as a program product in a variety of forms, and that an illustrative embodiment of the subject matter described herein applies regardless of the particular type of signal bearing medium used to actually carry out the distribution. Examples of a signal bearing medium include, but are not limited to, the following: a recordable type medium such as a floppy disk, a hard disk drive, a CD, a DVD, a digital tape, a computer memory, etc., and a transmission type medium such as a digital and/or an analog communication medium (e.g., a fiber optic cable, a waveguide, a wired communications link, a wireless communication link, etc.).

[0250] Those skilled in the art will recognize that it is common within the art to describe devices and/or processes in the fashion set forth herein, and thereafter use engineering practices to integrate such described

devices and/or processes into data processing systems. That is, at least a portion of the devices and/or processes described herein may be integrated into a data processing system via a reasonable amount of experimentation. Those having skill in the art will recognize that a typical data processing system may generally include one or more of a system unit housing, a video display device, a memory such as volatile and non-volatile memory, processors such as microprocessors and digital signal processors, computational entities such as operating systems, drivers, graphical user interfaces, and applications programs, one or more interaction devices, such as a touch pad or screen, and/or control systems including feedback loops and control motors (e.g., feedback for sensing position and/or velocity, control motors for moving and/or adjusting components and/or quantities). A typical data processing system may be implemented utilizing any suitable commercially available components, such as those typically found in data computing/communication and/or network computing/communication systems.

[0251] The herein described subject matter sometimes illustrates different components included within, or connected with, different other components. It is to be understood that such depicted architectures are merely examples, and that in fact many other architectures may be implemented which achieve the same functionality. In a conceptual sense, any arrangement of components to achieve the same functionality is effectively "associated" such that the desired functionality may be achieved. Hence, any two components herein combined to achieve a particular functionality may be seen as "associated with" each other such that the desired functionality is achieved, irrespective of architectures or intermedial components. Likewise, any two components so associated may also be viewed as being "operably connected", or "operably coupled", to each other to achieve the desired functionality, and any two components capable of being so associated may also be viewed as being "operably couplable" to each other to achieve the desired functionality. Specific examples of operably couplable include but are not limited to physically mateable and/or physically interacting components and/or wirelessly interactable and/or wirelessly interacting components and/or logically interacting and/or logically interactable components.

[0252] With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

[0253] It will be understood by those within the art that, in general, terms used herein, and especially in the appended claims (e.g., bodies of the appended claims) are generally intended as "open" terms (e.g., the term "including" should be interpreted as "including but not limited to," the term "having" should be interpreted as "having at least," the term "includes" should be interpreted as "includes but is not limited to," etc.). It will be further understood by those within the art that if a specific number of an introduced claim recitation is intended, such an intent will be explicitly recited in the claim, and in the absence of such recitation no such

intent is present. For example, where only one item is intended, the term "single" or similar language may be used. As an aid to understanding, the following appended claims and/or the descriptions herein may include usage of the introductory phrases "at least one" and "one or more" to introduce claim recitations. However, the use of such phrases should not be construed to imply that the introduction of a claim recitation by the indefinite articles "a" or "an" limits any particular claim including such introduced claim recitation to embodiments including only one such recitation, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an" (e.g., "a" and/or "an" should be interpreted to mean "at least one" or "one or more"). The same holds true for the use of definite articles used to introduce claim recitations. In addition, even if a specific number of an introduced claim recitation is explicitly recited, those skilled in the art will recognize that such recitation should be interpreted to mean at least the recited number (e.g., the bare recitation of "two recitations," without other modifiers, means at least two recitations, or two or more recitations). Furthermore, in those instances where a convention analogous to "at least one of A, B, and C, etc." is used, in general such a construction is intended in the sense one having skill in the art would understand the convention (e.g., "a system having at least one of A, B, and C" would include but not be limited to systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together, etc.). In those instances where a convention analogous to "at least one of A, B, or C, etc." is used, in general such a construction is intended in the sense one having skill in the art would understand the convention (e.g., "a system having at least one of A, B, or C" would include but not be limited to systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together, etc.). It will be further understood by those within the art that virtually any disjunctive word and/or phrase presenting two or more alternative terms, whether in the description, claims, or drawings, should be understood to contemplate the possibilities of including one of the terms, either of the terms, or both terms. For example, the phrase "A or B" will be understood to include the possibilities of "A" or "B" or "A and B." Further, the terms "any of" followed by a listing of a plurality of items and/or a plurality of categories of items, as used herein, are intended to include "any of," "any combination of," "any multiple of," and/or "any combination of multiples of" the items and/or the categories of items, individually or in conjunction with other items and/or other categories of items. Moreover, as used herein, the term "set" is intended to include any number of items, including zero. Additionally, as used herein, the term "number" is intended to include any number, including zero. And the term "multiple", as used herein, is intended to be synonymous with "a plurality".

[0254] In addition, where features or aspects of the disclosure are described in terms of Markush groups, those skilled in the art will recognize that the disclosure is also thereby described in terms of any individual member or subgroup of members of the Markush group.

[0255] As will be understood by one skilled in the art, for any and all purposes, such as in terms of providing a written description, all ranges disclosed herein also encompass any and all possible subranges and combinations of subranges thereof. Any listed range can be easily recognized as sufficiently describing and enabling the same range being broken down into at least equal halves, thirds, quarters, fifths, tenths, etc. As a non-limiting example, each range discussed herein may be readily broken down into a lower third, middle third and upper third, etc. As will also be understood by one skilled in the art all language such as "up to," "at least," "greater than," "less than," and the like includes the number recited and refers to ranges which can be subsequently broken down into subranges as discussed above. Finally, as will be understood by one skilled in the art, a range includes each individual member. Thus, for example, a group having 1-3 cells refers to groups having 1, 2, or 3 cells. Similarly, a group having 1-5 cells refers to groups having 1, 2, 3, 4, or 5 cells, and so forth.

[0256] Moreover, the claims should not be read as limited to the provided order or elements unless stated to that effect. In addition, use of the terms "means for" in any claim is intended to invoke 35 U.S.C. §112, ¶ 6 or means-plus-function claim format, and any claim without the terms "means for" is not so intended.

CLAIMS

What is claimed is:

1. A method implemented by a wireless transmit/receive unit (WTRU) for wireless communications, the method comprising:

determining a first resource requirement for a list of tasks associated with one or more Edge devices;

transmitting, to the one or more Edge devices, a first request indicating the first resource requirement;

receiving first information indicating a combination of resources being available at the one or more Edge devices based on the first resource requirement;

determining, based on the first information, a second resource requirement for the list of tasks;

transmitting a second request indicating the second resource requirement; and

receiving second information indicating a decision result associated with the list of tasks based on the second resource requirement.

2. The method of claim 1, wherein the combination of resources being available at the one or more Edge devices is further based on a calculation of resources at the one or more Edge devices.

3. The method of any one of the preceding claims, wherein the decision result indicates an acceptance of the second request.

4. The method of any one of the preceding claims, wherein the first request was marshaled into byte arrays by the WTRU before transmission.

5. The method of any one of the preceding claims, wherein the first request comprises a first set of data structure arguments.

6. The method of any one of the preceding claims, wherein the second request comprises a second set of data structure arguments.

7. The method of any one of the preceding claims, wherein the determining the second resource requirement comprises calculating the second resource requirements based on the combination of resources being available at the one or more Edge devices.

8. The method of any one of the preceding claims, wherein the determining the second resource requirement comprises recalculating the first resource requirements based on the combination of resources being available at the one or more Edge devices.

9. The method of any one of the preceding claims, wherein the combination of resources being available at the one or more Edge devices is different from the first resource requirement.

10. The method of any one of the preceding claims, wherein the second resource requirement is different from the first resource requirement.

11. A wireless transmit/receive unit (WTRU) for wireless communications, the WTRU comprising circuitry, including a transmitter, a receiver, a processor and memory, configured to:

determine a first resource requirement for a list of tasks associated with one or more Edge devices;

transmit, to the one or more Edge devices, a first request indicating the first resource requirement;

receive first information indicating a combination of resources being available at the one or more Edge devices based on the first resource requirement;

determine, based on the first information, a second resource requirement for the list of tasks;

transmit a second request indicating the second resource requirement; and

receive second information indicating a decision result associated with the list of tasks based on the second resource requirement.

12. The WTRU of claim 11, wherein the combination of resources being available at the one or more Edge devices is further based on a calculation of resources at the one or more Edge devices.

13. The WTRU of claim 11, wherein the decision result indicates an acceptance of the second request.

14. The WTRU of any one of claims 11-13, wherein the WTRU is further configured to marshal the first request into byte arrays before transmission.

15. The WTRU of any one of claims 11-14, wherein the first request comprises a first set of data structure arguments.

16. The WTRU of any one of claims 11-15, wherein the second request comprises a second set of data structure arguments.

17.The WTRU of any one of claims 11-16, wherein, when determining the second resource requirement, the WTRU is further configured to calculate the second resource requirements based on the combination of resources being available at the one or more Edge devices.

18.The WTRU of any one of claims 11-16, wherein, when determining the second resource requirement, the WTRU is further configured to recalculating the first resource requirements based on the combination of resources being available at the one or more Edge devices.

19.The WTRU of any one of claims 11-18, wherein the combination of resources being available at the one or more Edge devices is different from the first resource requirement.

20.The WTRU of any one of claims 11-19, wherein the second resource requirement is different from the first resource requirement.

21.A method implemented by a wireless transmit/receive unit (WTRU) for wireless communications, the method comprising:

transmitting, to an Edge device, a first message including a first resource combination request;

receiving, from the Edge device, a second message including information indicating a combination of resources being available at the Edge device based on 1) the first resource combination request and/or 2) a calculation of resources at the Edge device;

determining resource combination requirements of an application based on the second message;

transmitting, to the Edge device, a third message including a second resource combination request based on the determined resource combination requirements; and

receiving, from the Edge device, a fourth message including information indicating a decision result in response to the second resource combination request.

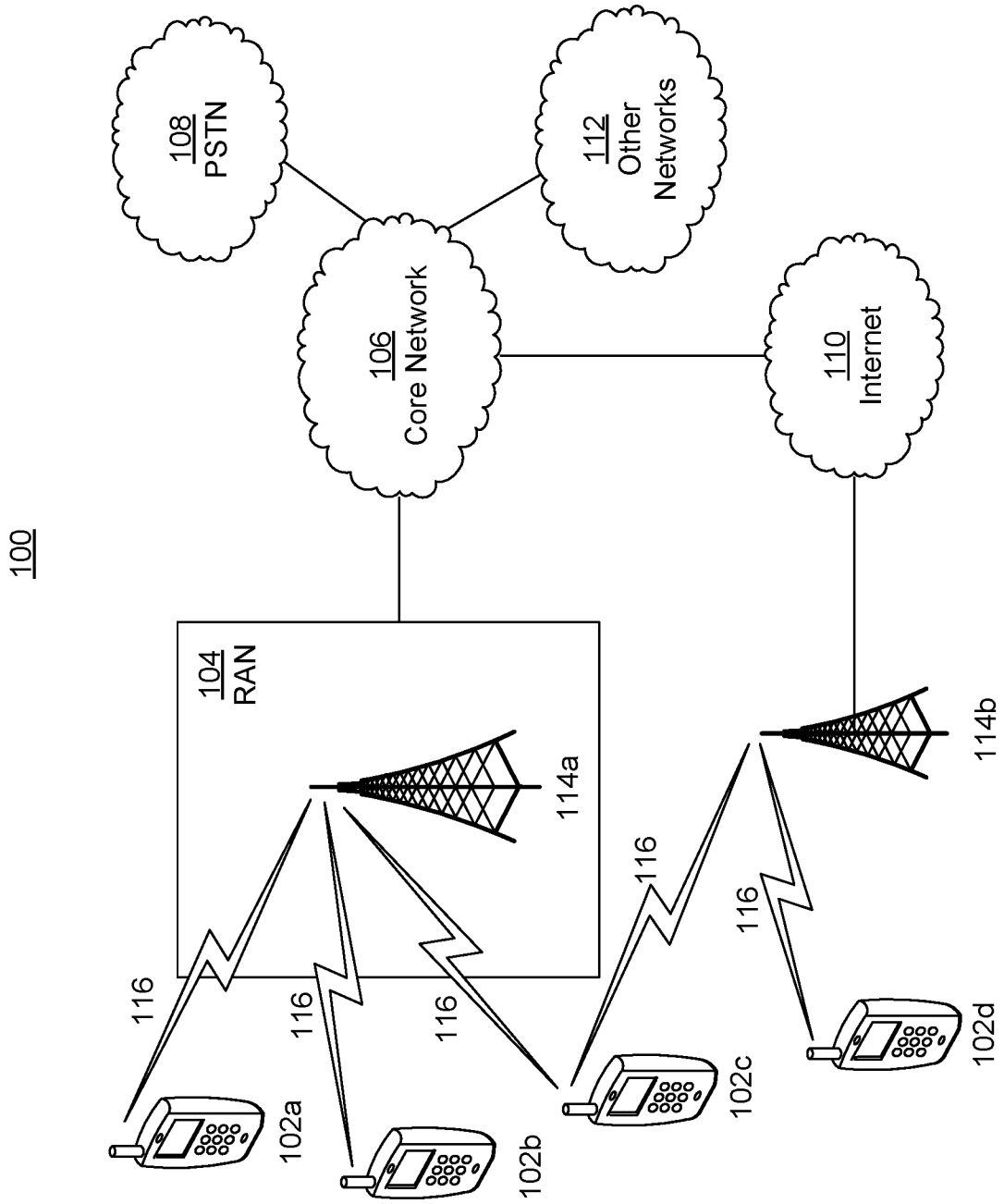

100
108 PSTN
112 Other Networks
106 Core Network
110 Internet
104 RAN
114a
114b
116
102a
102b
102c
102d

FIG. 1A

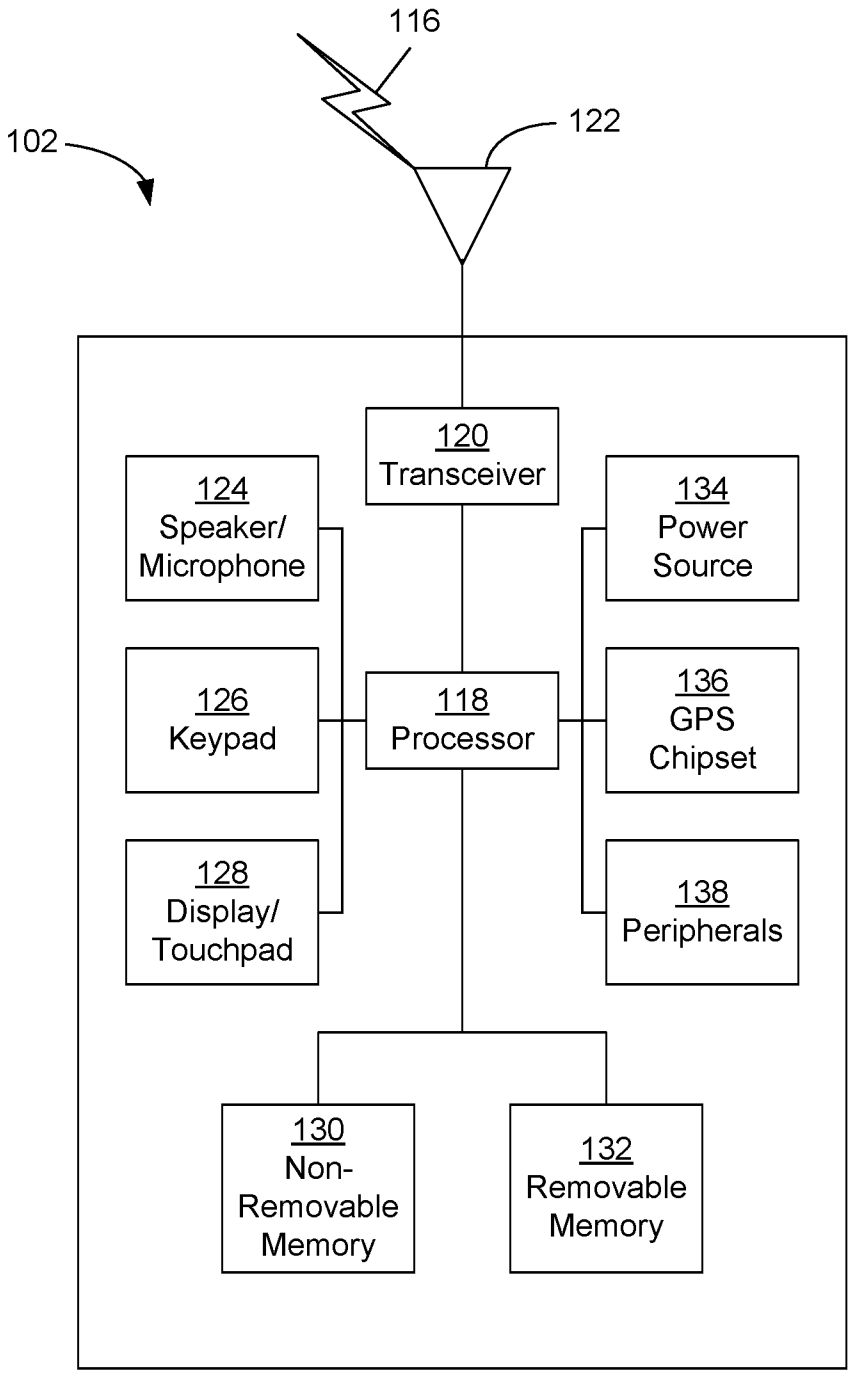

116
122
102
120
Transceiver
124
Speaker/
Microphone
134
Power
Source
126
Keypad
118
Processor
136
GPS
Chipset
128
Display/
Touchpad
138
Peripherals
130
Non-
Removable
Memory
132
Removable
Memory

FIG. 1B

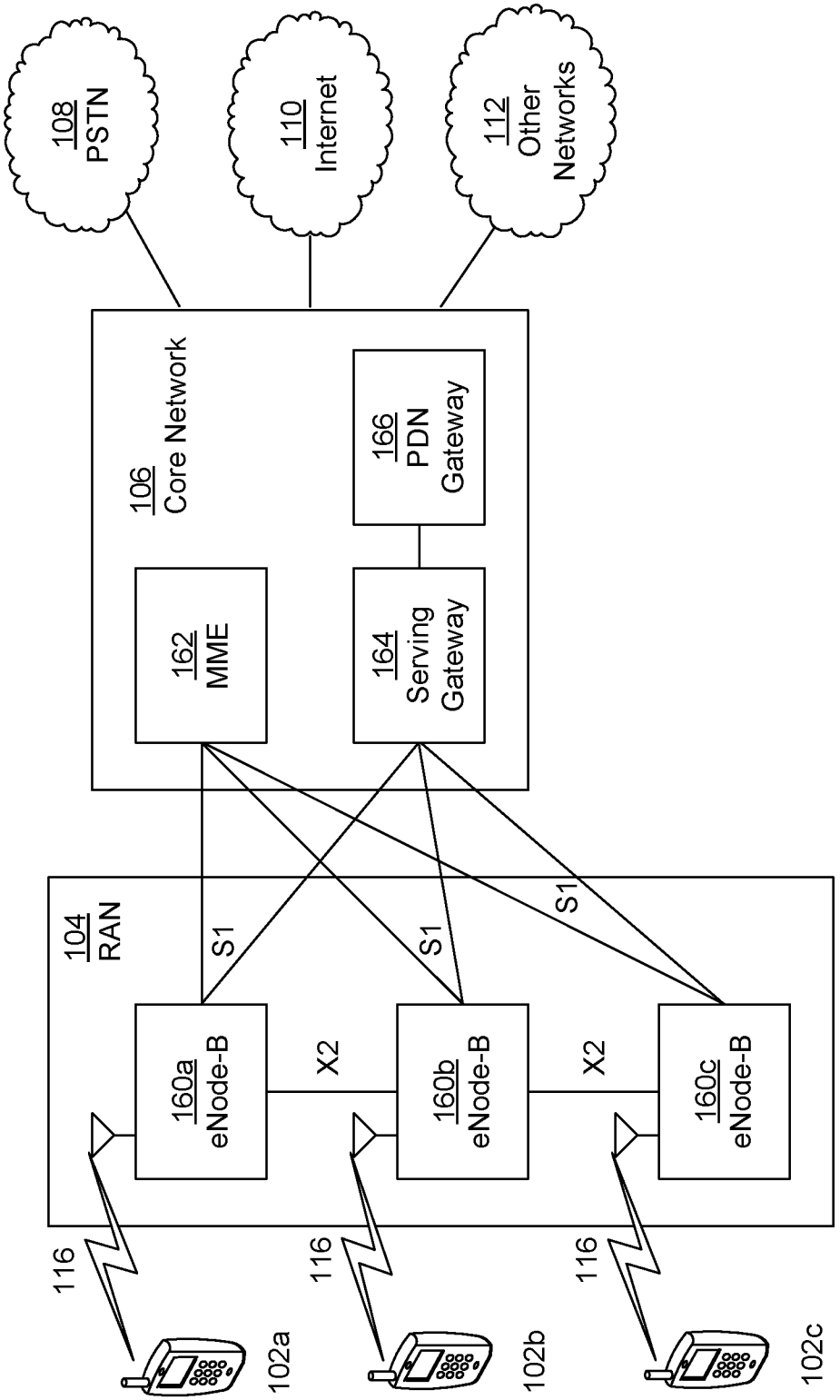

108 PSTN
110 Internet
112 Other Networks
106 Core Network
166 PDN Gateway
162 MME
164 Serving Gateway
104 RAN
S1
S1
S1
160a eNode-B
160b eNode-B
160c eNode-B
X2
X2
116
116
116
102a
102b
102c

FIG. 1C

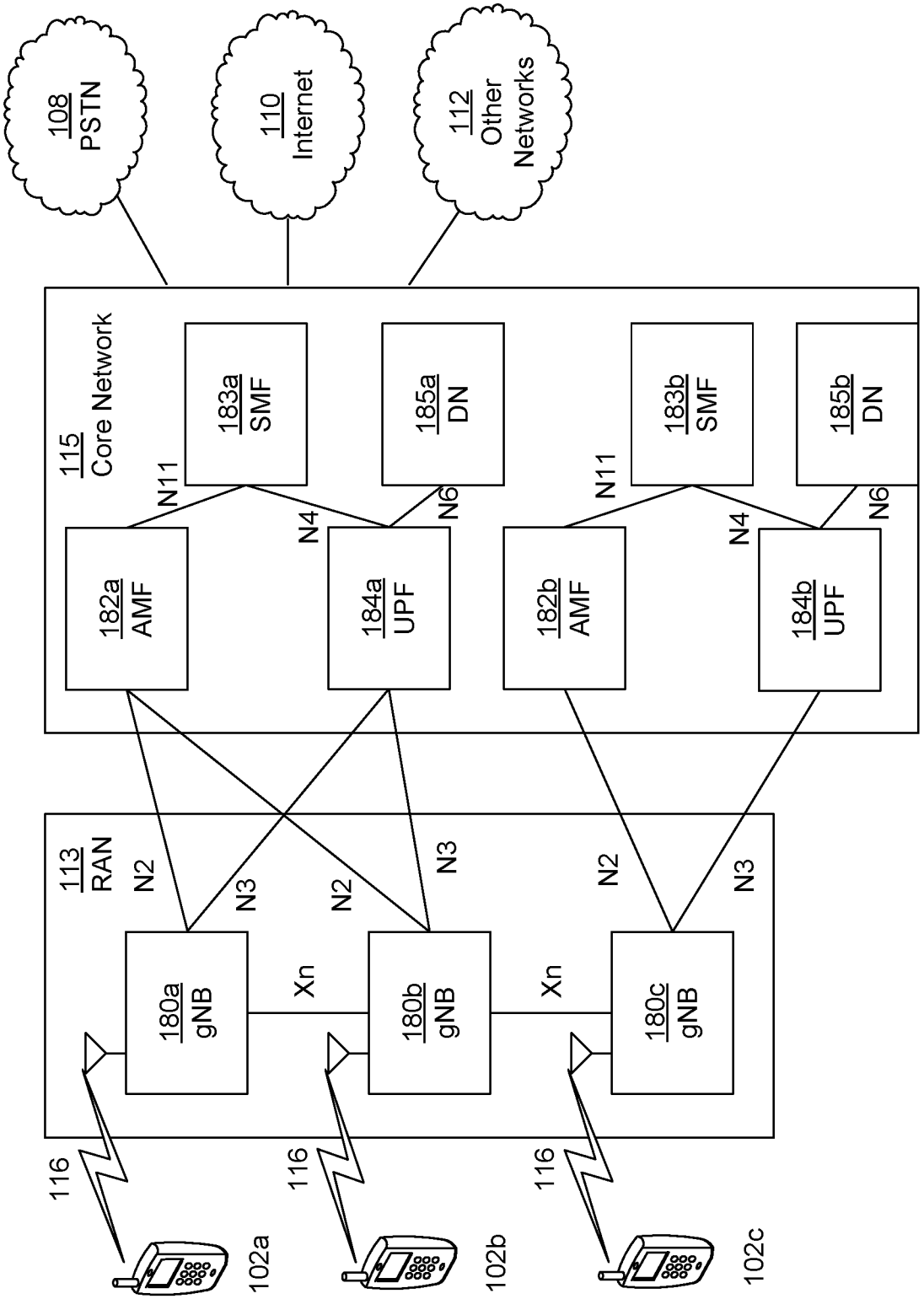

108 PSTN
110 Internet
112 Other Networks
115 Core Network
183a SMF
185a DN
183b SMF
185b DN
N11
N6
N4
182a AMF
184a UPF
182b AMF
184b UPF
113 RAN
N2
N3
180a gNB
Xn
180b gNB
180c gNB
116
102a
102b
102c

FIG. 1D

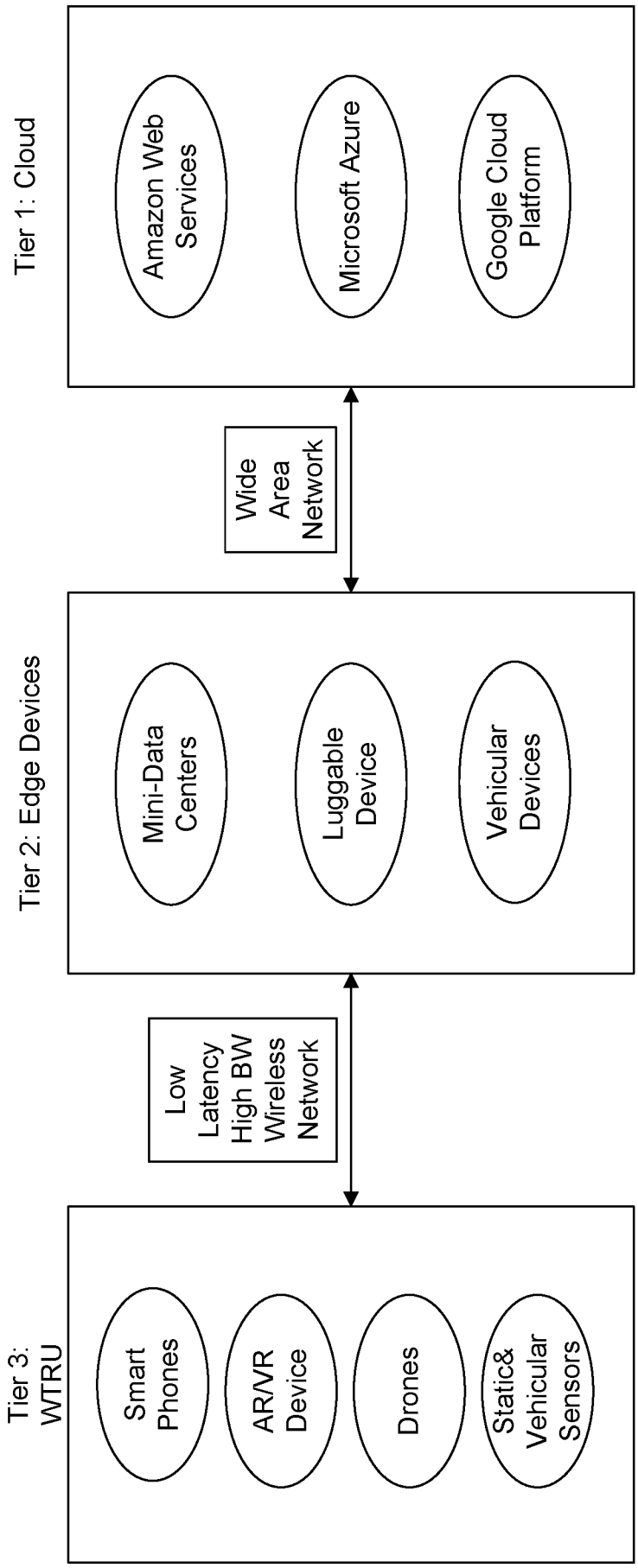

Tier 1: Cloud
Amazon Web Services
Microsoft Azure
Google Cloud Platform
Wide Area Network
Tier 2: Edge Devices
Mini-Data Centers
Luggable Device
Vehicular Devices
Low Latency High BW Wireless Network
Tier 3: WTRU
Smart Phones
AR/VR Device
Drones
Static& Vehicular Sensors

FIG. 2

amrmd1(n,C_1, . . . , Cn, Ɛ)

```
1.  u*  := 0
2.  p  := initial_penalty (C1,…,Cn,r^max)
3.  repeat := true
4.  while repeat do
5.     repeat := false
6.     for i = 1 to n do
7.        Cic := compound_resource (Ci,p)
8.      for i = 1 to n do
9.         C'ic :=0 convex_hull_frontier(Cic)
10.        r[i] := 0                    // vector assignment
11.        u[i] := 0
12.        stop[i] := 0
13.     s_list= merge(C'1c,…, C'nc)
14.     r^c := r^max
15.     for j = 1 to |s_list| do
16.       i := s_list[j].t
17.      if (stop[i]) then
18.         break
19.       b := s_list[j].r - r[i]          //vector subtraction
20.       if (b ≤ r^c) then
21.         r^c :=  r^c - b
22.         r[i] : = s_list[j].r
23.         u[i] : = s_list[j].u         //update allocation of Ti
24.       else
25.           stop[i] := 1
26.      u :=0
27.      for i = 1 to n do
28.       u := u+u*[i]
29.      if ((u – u*) > Ɛ) then
30.         repeat := true
31.         u* := u
32.      for i = 1 to n do
33.       u*[i] := u[i]
34.       r*[i] := r[i]
35.      p := adjust_penalty (p,C1,…,Cn,r^c,r^max)
36.  for i = 1 to n do
37.    q[i] := hi(r*[i])              // see Equation  (6)
38.    return u*,q[1],…,q[n], r*[1],…,r*[i]
```

FIG. 3

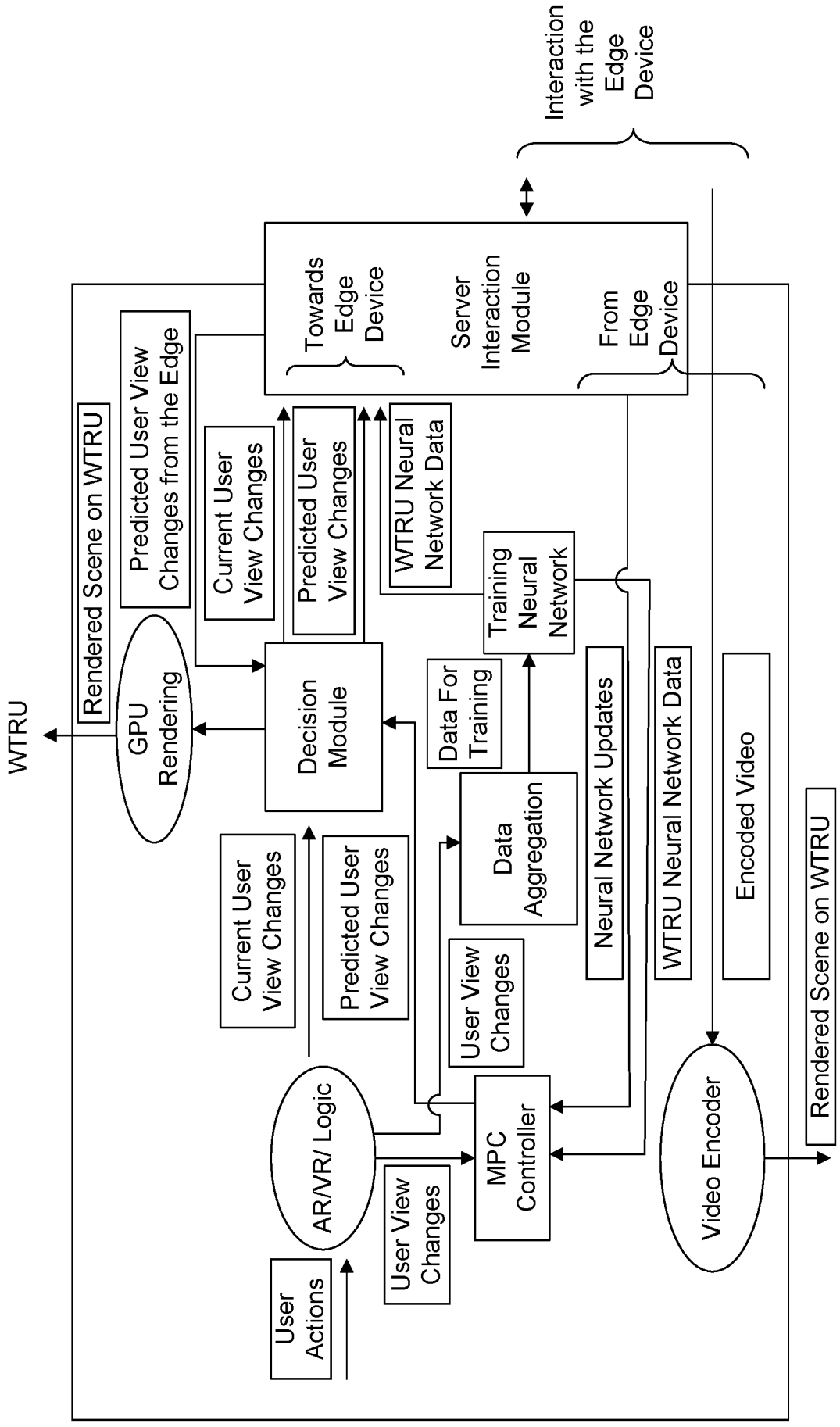

Interaction with the Edge Device
Towards Edge Device
Server Interaction Module
From Edge Device
Predicted User View Changes from the Edge
Current User View Changes
Predicted User View Changes
WTRU Neural Network Data
Rendered Scene on WTRU
WTRU
GPU Rendering
Decision Module
Training Neural Network
Data For Training
Data Aggregation
Neural Network Updates
WTRU Neural Network Data
Encoded Video
Current User View Changes
Predicted User View Changes
User View Changes
AR/VR/ Logic
MPC Controller
Video Encoder
Rendered Scene on WTRU
User Actions
User View Changes

FIG. 4

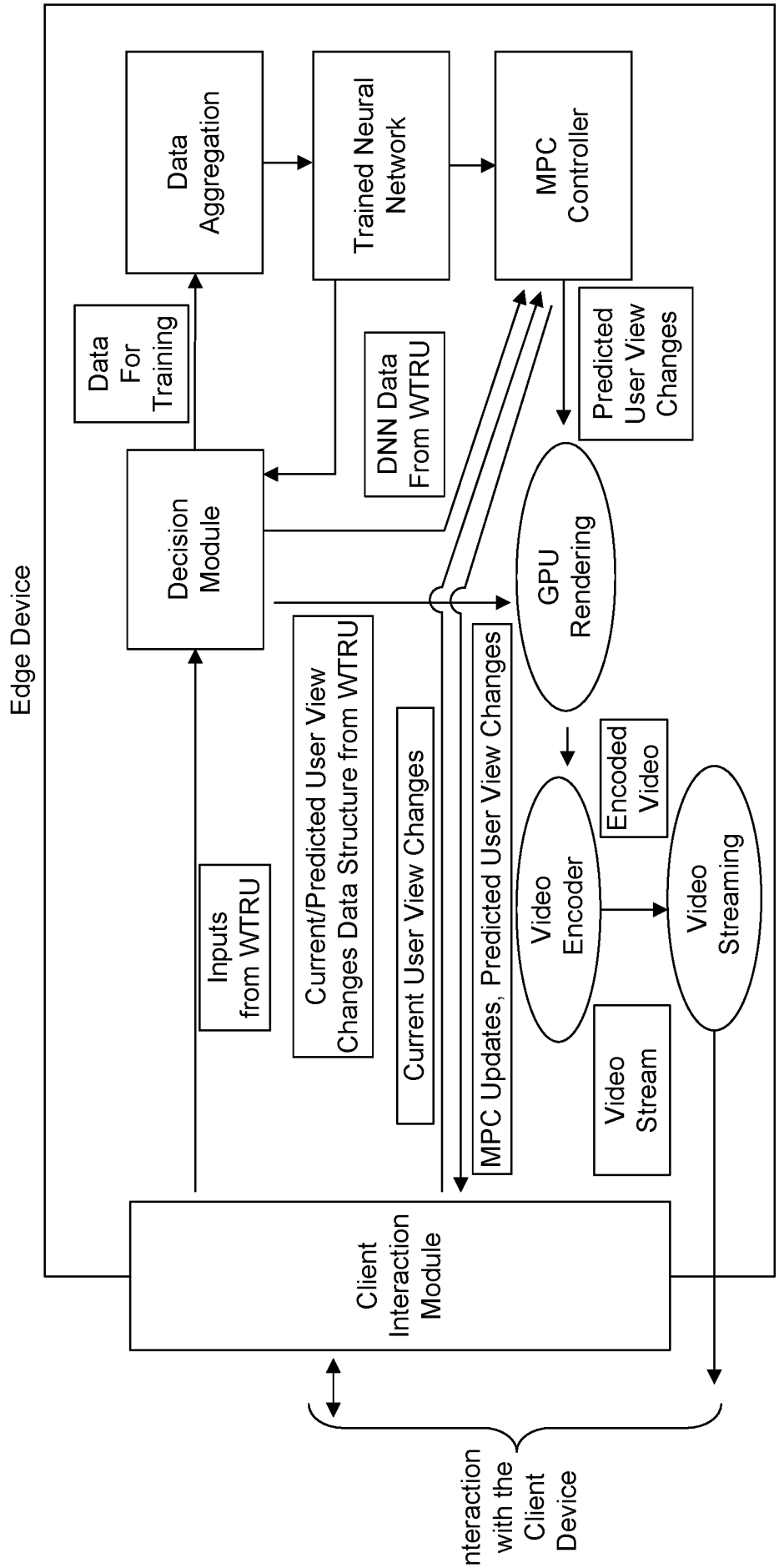

Edge Device
Data Aggregation
Trained Neural Network
MPC Controller
Data For Training
DNN Data From WTRU
Predicted User View Changes
Decision Module
GPU Rendering
Inputs from WTRU
Current/Predicted User View Changes Data Structure from WTRU
Current User View Changes
MPC Updates, Predicted User View Changes
Encoded Video
Video Encoder
Video Streaming
Video Stream
Client Interaction Module
Interaction with the Client Device

FIG. 5

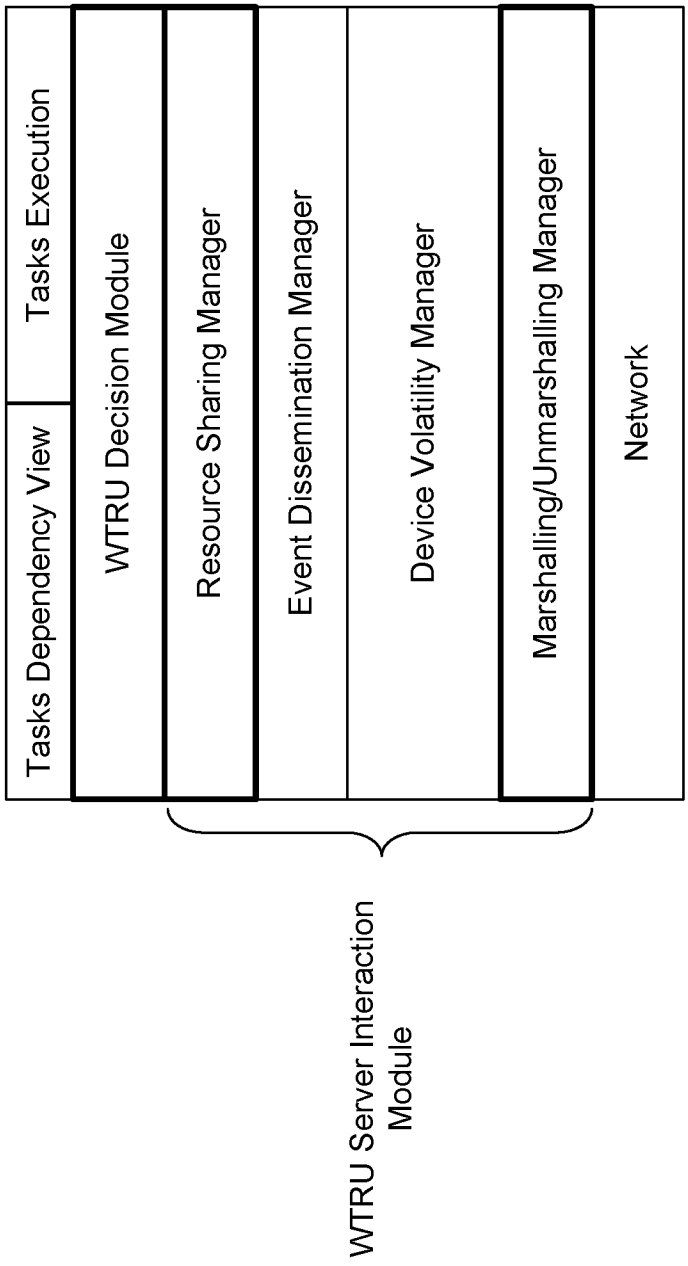

Tasks Dependency View
Tasks Execution
WTRU Decision Module
Resource Sharing Manager
Event Dissemination Manager
Device Volatility Manager
Marshalling/Unmarshalling Manager
Network
WTRU Server Interaction Module

FIG. 6

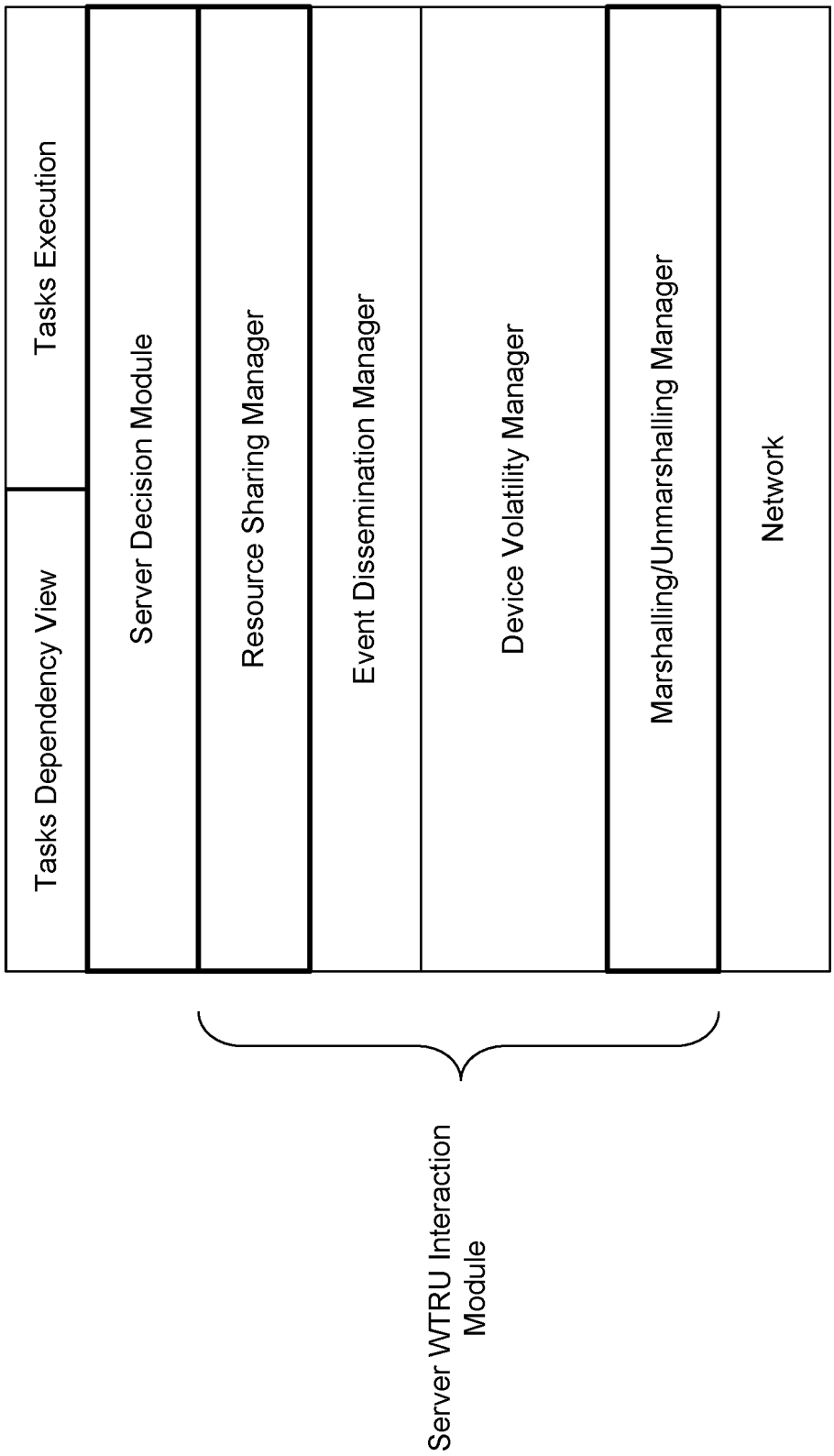

Tasks Dependency View
Tasks Execution
Server Decision Module
Resource Sharing Manager
Event Dissemination Manager
Device Volatility Manager
Marshalling/Unmarshalling Manager
Network
Server WTRU Interaction Module

FIG. 7

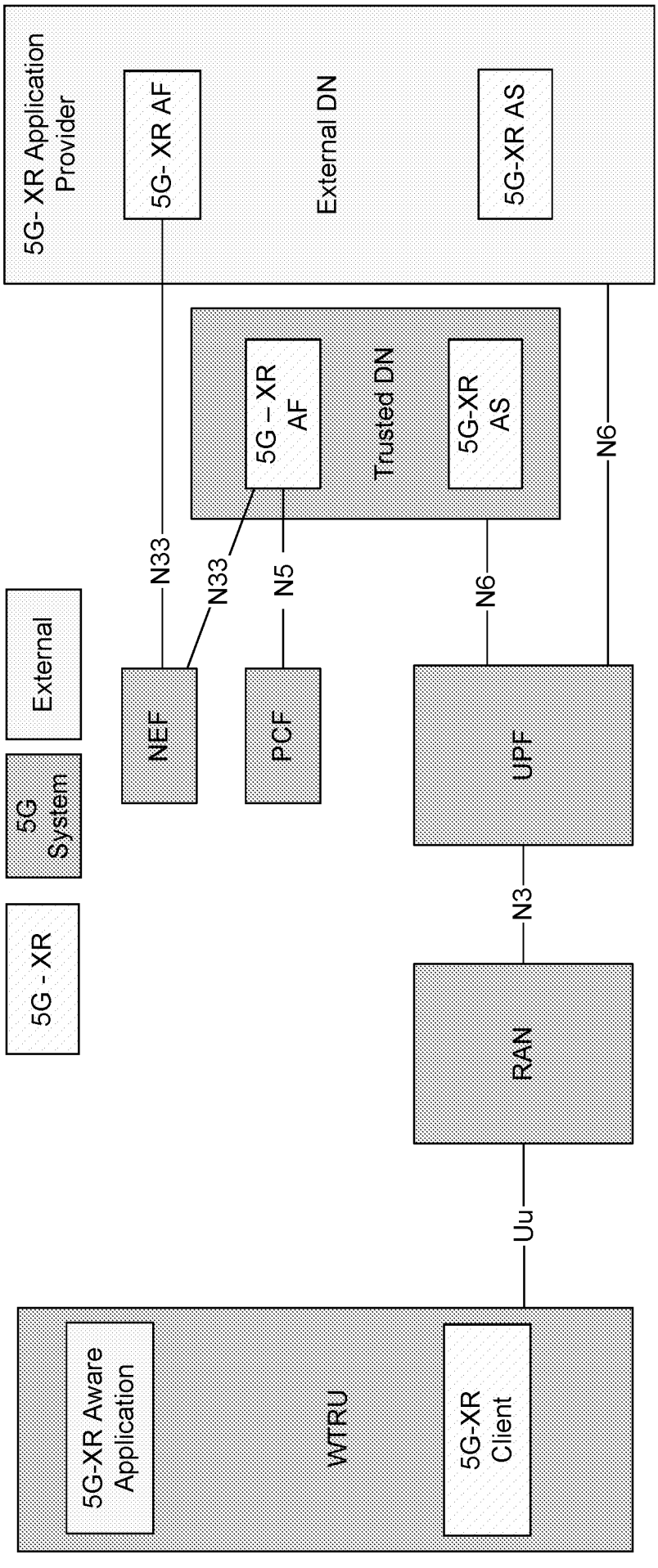

5G- XR Application Provider
5G- XR AF
External DN
5G-XR AS
Trusted DN
5G – XR AF
5G-XR AS
N33
N33
N5
N6
N6
External
5G System
5G - XR
NEF
PCF
UPF
N3
RAN
Uu
5G-XR Aware Application
WTRU
5G-XR Client

FIG. 8

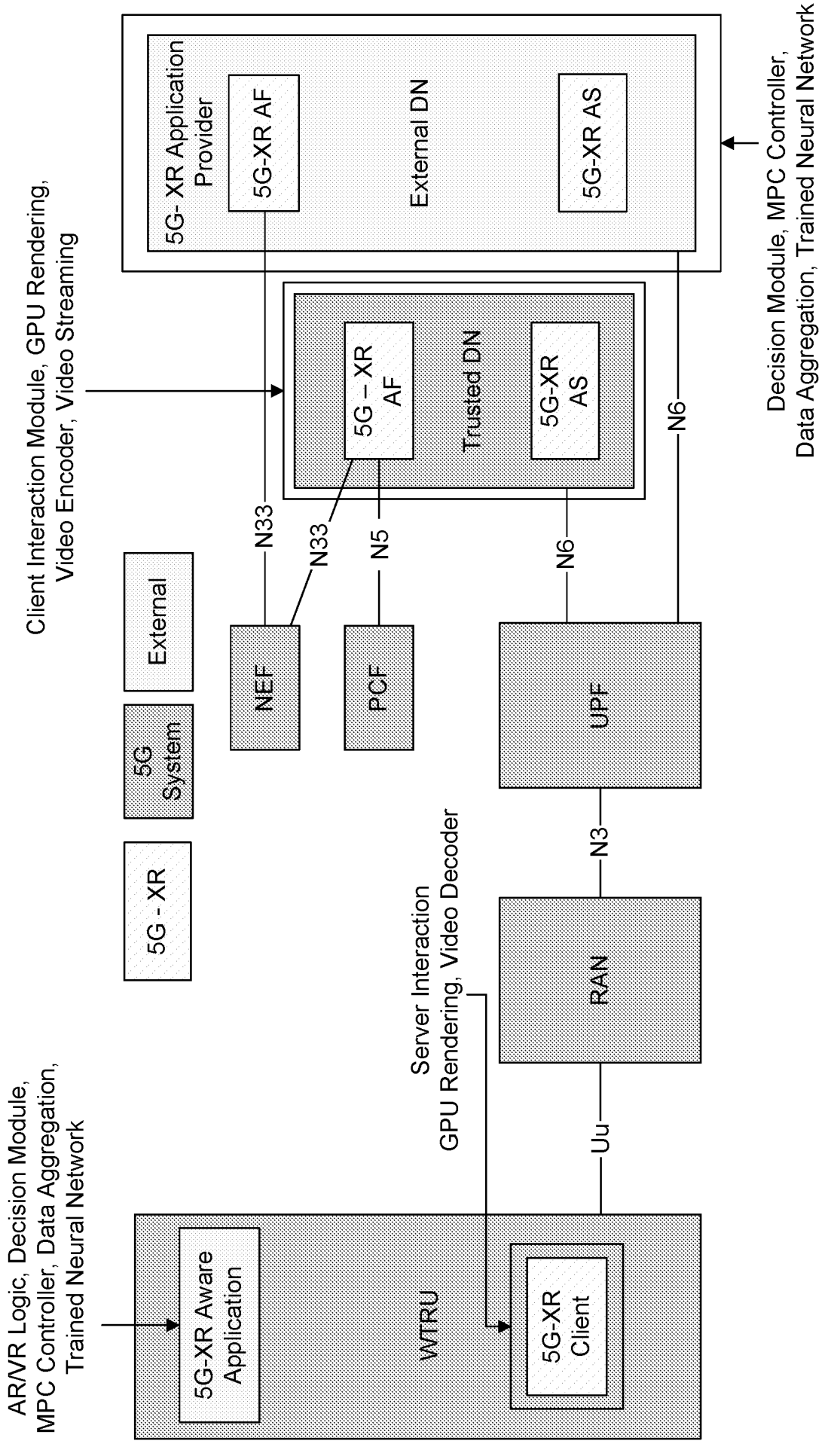

AR/VR Logic, Decision Module,
MPC Controller, Data Aggregation,
Trained Neural Network
5G-XR Aware Application
WTRU
5G-XR Client
Uu
RAN
N3
UPF
N6
Server Interaction
GPU Rendering, Video Decoder
NEF
PCF
N33
N33
N5
N6
5G – XR AF
Trusted DN
5G-XR AS
5G- XR Application Provider
5G-XR AF
External DN
5G-XR AS
Client Interaction Module, GPU Rendering,
Video Encoder, Video Streaming
Decision Module, MPC Controller,
Data Aggregation, Trained Neural Network
5G - XR
5G System
External

FIG. 9

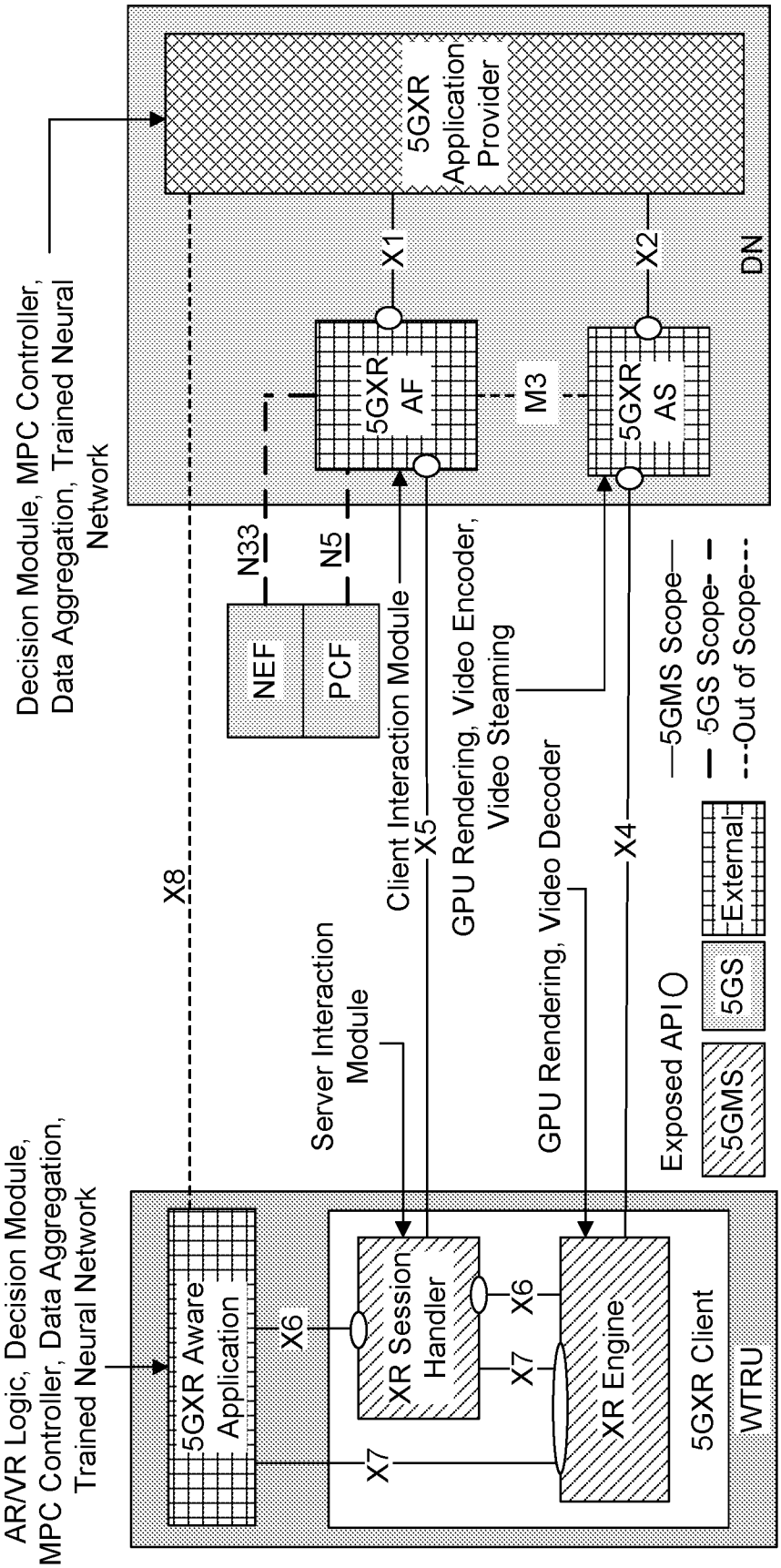

AR/VR Logic, Decision Module, MPC Controller, Data Aggregation, Trained Neural Network
5GXR Aware Application
X6
X7
XR Session Handler
X6
X7
XR Engine
5GXR Client
WTRU
X8
Server Interaction Module
Client Interaction Module
X5
GPU Rendering, Video Encoder, Video Steaming
GPU Rendering, Video Decoder
X4
Exposed API
5GMS
5GS
External
5GMS Scope
5GS Scope
Out of Scope
Decision Module, MPC Controller, Data Aggregation, Trained Neural Network
N33
N5
NEF
PCF
5GXR AF
M3
5GXR AS
X1
X2
5GXR Application Provider
DN

FIG. 10

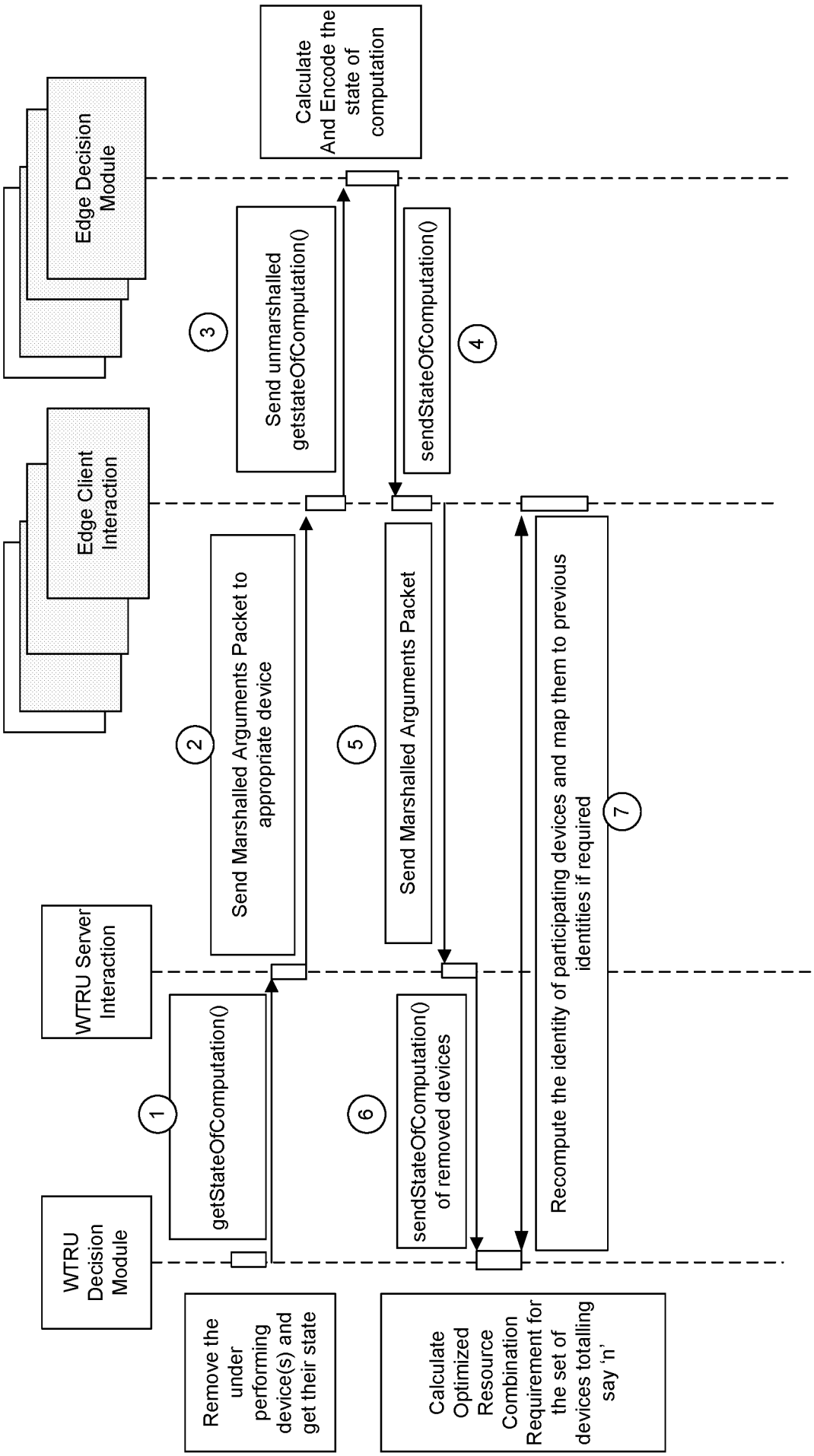

WTRU Decision Module
WTRU Server Interaction
Edge Client Interaction
Edge Decision Module
Remove the under performing device(s) and get their state
Calculate Optimized Resource Combination Requirement for the set of devices totalling say 'n'
1
getStateOfComputation()
2
Send Marshalled Arguments Packet to appropriate device
3
Send unmarshalled getstateOfComputation()
Calculate And Encode the state of computation
4
sendStateOfComputation()
5
Send Marshalled Arguments Packet
6
sendStateOfComputation() of removed devices
7
Recompute the identity of participating devices and map them to previous identities if required

FIG. 11

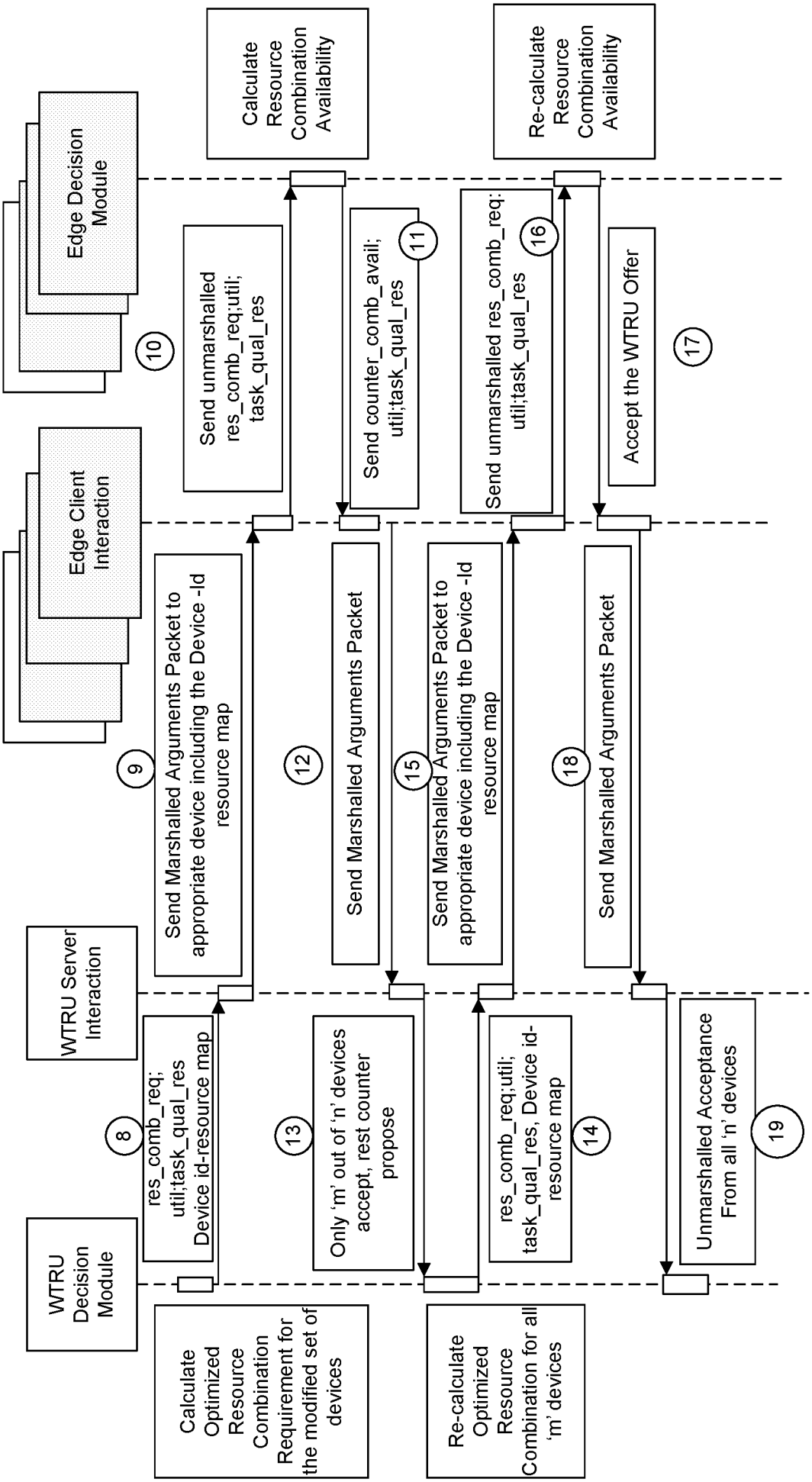

WTRU Decision Module
WTRU Server Interaction
Edge Client Interaction
Edge Decision Module
Calculate Optimized Resource Combination Requirement for the modified set of devices
Re-calculate Optimized Resource Combination for all 'm' devices
Calculate Resource Combination Availability
Re-calculate Resource Combination Availability
8
res_comb_req; util;task_qual_res Device id-resource map
9
Send Marshalled Arguments Packet to appropriate device including the Device -Id resource map
10
Send unmarshalled res_comb_req;util; task_qual_res
11
Send counter_comb_avail; util;task_qual_res
12
Send Marshalled Arguments Packet
13
Only 'm' out of 'n' devices accept, rest counter propose
14
res_comb_req;util; task_qual_res, Device id-resource map
15
Send Marshalled Arguments Packet to appropriate device including the Device -Id resource map
16
Send unmarshalled res_comb_req: util;task_qual_res
17
Accept the WTRU Offer
18
Send Marshalled Arguments Packet
19
Unmarshalled Acceptance From all 'n' devices

FIG. 12

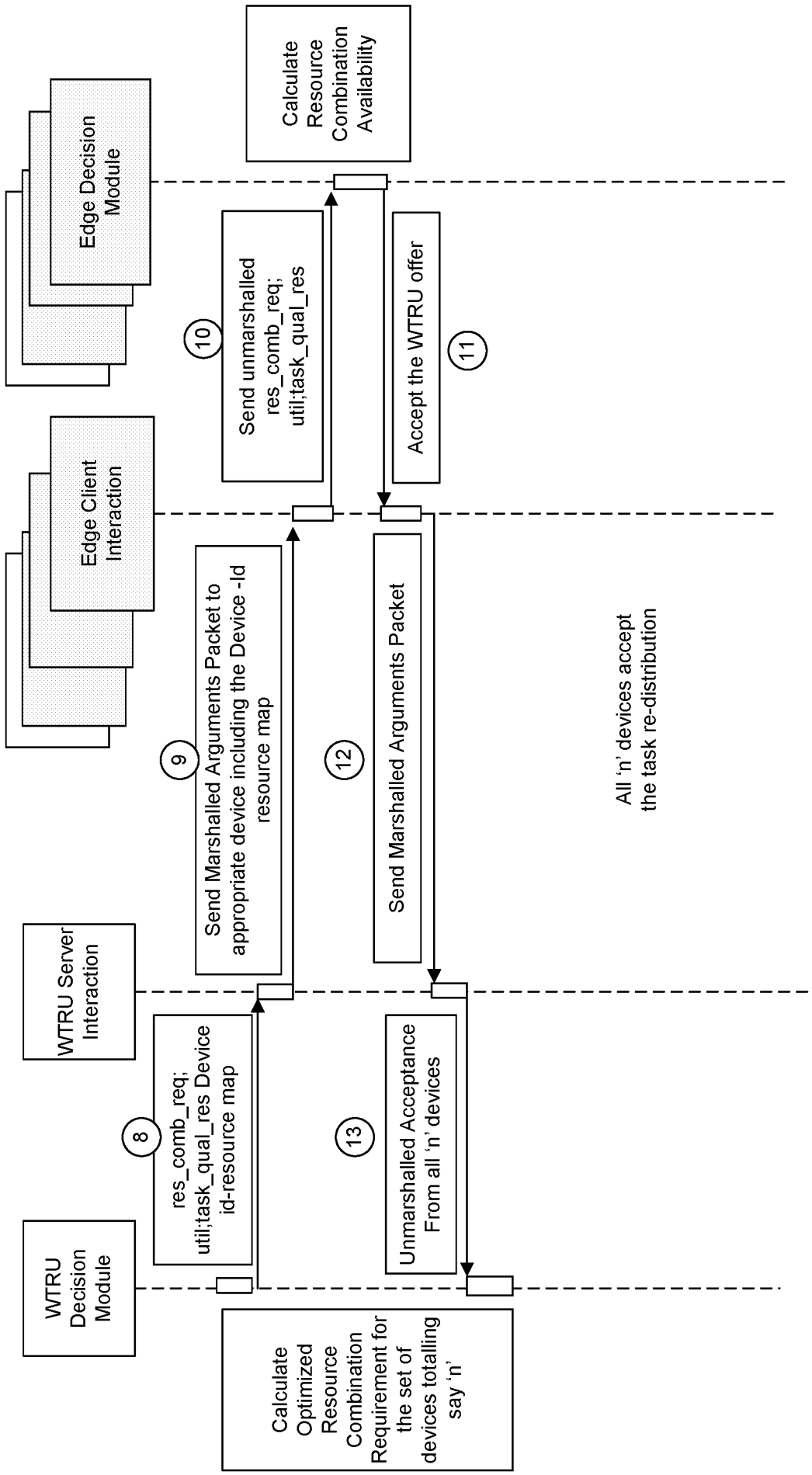

WTRU Decision Module
WTRU Server Interaction
Edge Client Interaction
Edge Decision Module
Calculate Optimized Resource Combination Requirement for the set of devices totalling say 'n'
8
res_comb_req; util;task_qual_res Device id-resource map
9
Send Marshalled Arguments Packet to appropriate device including the Device -Id resource map
10
Send unmarshalled res_comb_req; util;task_qual_res
Calculate Resource Combination Availability
11
Accept the WTRU offer
12
Send Marshalled Arguments Packet
13
Unmarshalled Acceptance From all 'n' devices
All 'n' devices accept the task re-distribution

FIG. 13

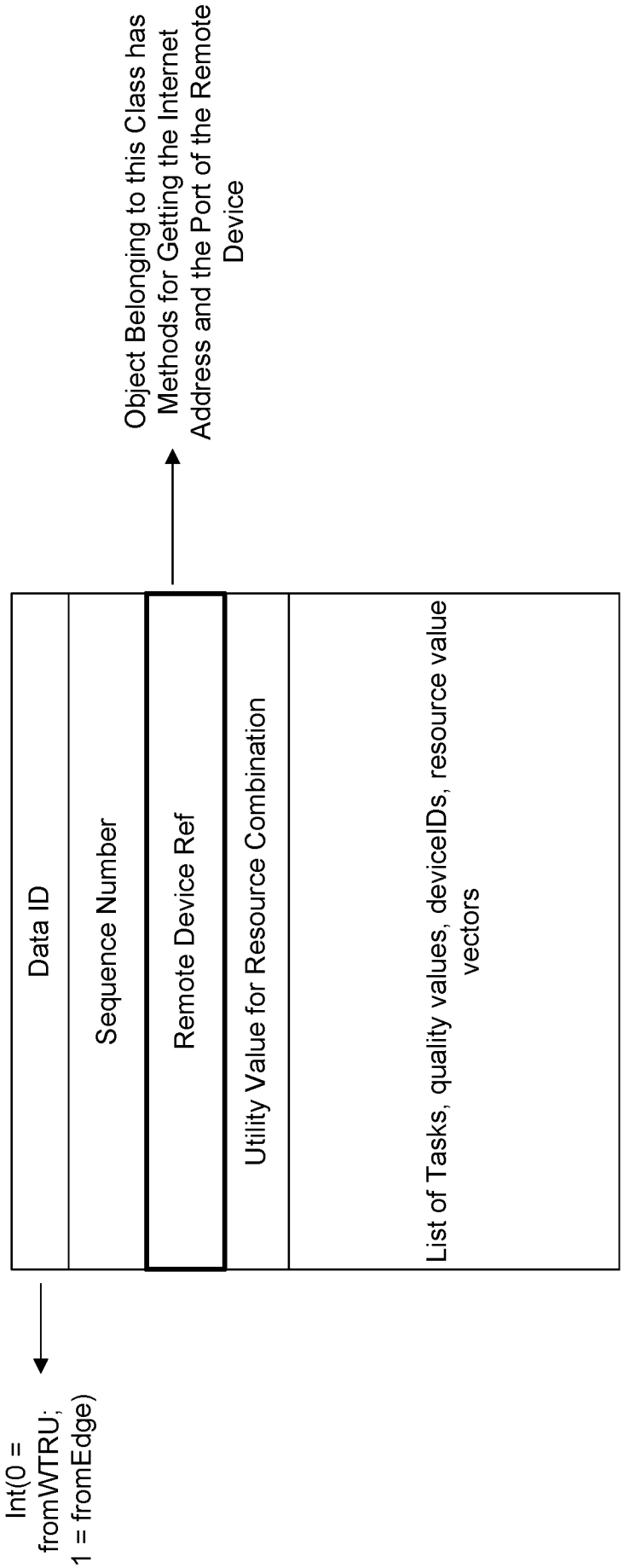

Int(0 = fromWTRU; 1 = fromEdge)
Data ID
Sequence Number
Remote Device Ref
Utility Value for Resource Combination
List of Tasks, quality values, deviceIDs, resource value vectors
Object Belonging to this Class has Methods for Getting the Internet Address and the Port of the Remote Device

FIG. 14

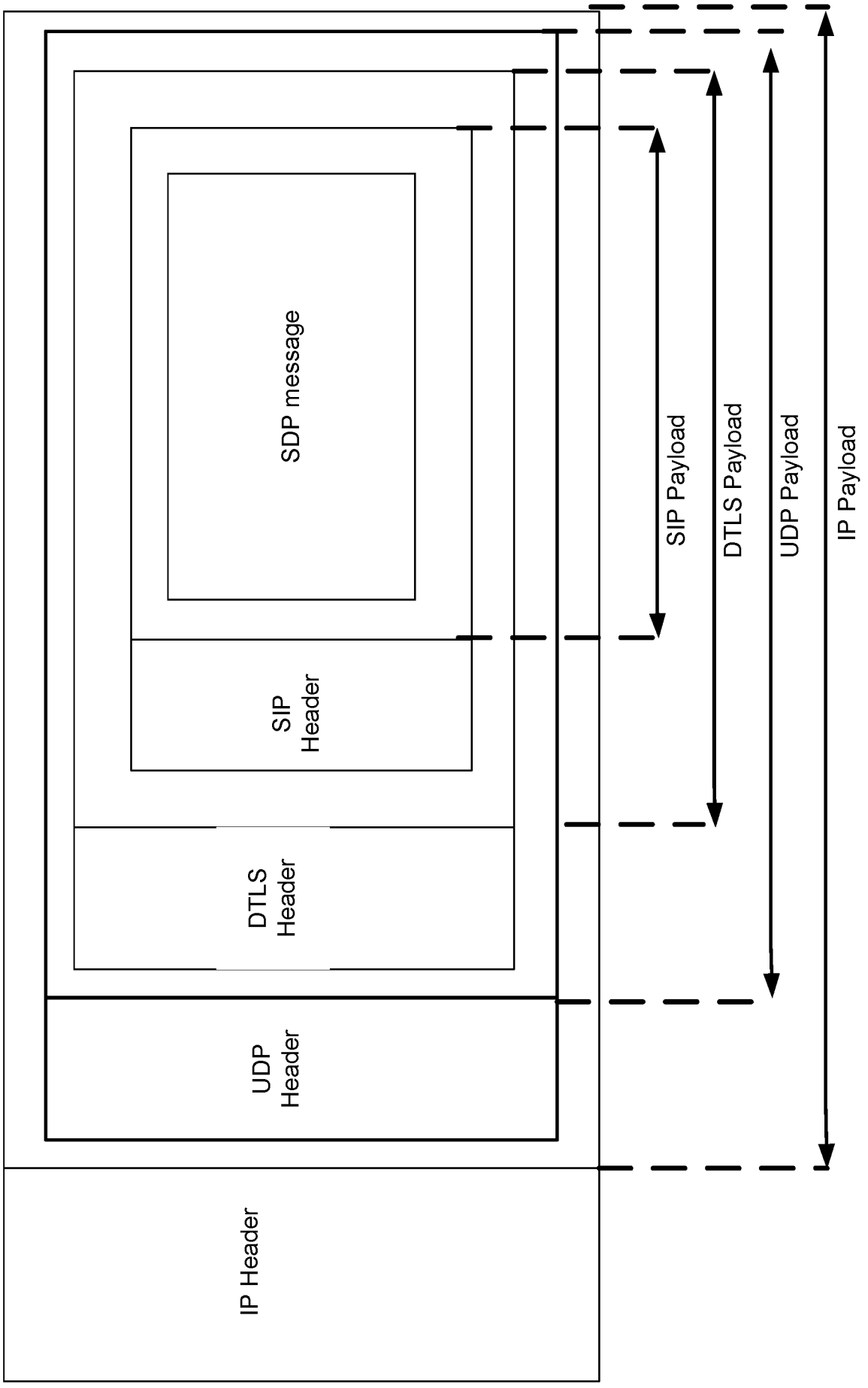

SDP message
SIP Header
DTLS Header
UDP Header
IP Header
SIP Payload
DTLS Payload
UDP Payload
IP Payload

FIG. 15

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/030375

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/50 H04L67/10 H04L67/51
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)
G06F H04L

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2021/014303 A1 (GUIM BERNAT FRANCESC [ES] ET AL) 14 January 2021 (2021-01-14) figures 1,2,4,5,8,10A, 10B, 11, 13, 14, 22, 25, 27B paragraph [0054] paragraph [0087] paragraph [0130] - paragraph [0140] paragraph [0161] - paragraph [0164] paragraph [0175] - paragraph [0176] paragraph [0338] -----	1-21
A	US 2021/286655 A1 (LIU LU [US] ET AL) 16 September 2021 (2021-09-16) figures 5,6,7 paragraph [0066] - paragraph [0069] paragraph [0075] - paragraph [0077] paragraph [0091] -----	1-21

-/--

[x] Further documents are listed in the continuation of Box C.

[x] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 October 2023	27/10/2023

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

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/030375

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2018/255123 A1 (DESAI NIRMIT V [US] ET AL) 6 September 2018 (2018-09-06) figures 3,4,7,13 paragraph [0055] paragraph [0068] - paragraph [0073] paragraph [0120] - paragraph [0124]	1-21

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2023/030375

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2021014303 A1	14-01-2021	CN 114338680 A	12-04-2022
		DE 102021207160 A1	31-03-2022
		US 2021014303 A1	14-01-2021
US 2021286655 A1	16-09-2021	EP 3799634 A1	07-04-2021
		US 2021286655 A1	16-09-2021
		WO 2020009875 A1	09-01-2020
US 2018255123 A1	06-09-2018	NONE	