



(51) International Patent Classification:
H04L 12/24 (2006.01) *H04W 52/02* (2009.01)

(21) International Application Number:
PCT/US2018/044063

(22) International Filing Date:
27 July 2018 (27.07.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/539,925 01 August 2017 (01.08.2017) US
62/539,936 01 August 2017 (01.08.2017) US

(71) Applicant: INTEL IP CORPORATION [US/US]; 2200
Mission College Boulevard, Santa Clara, California 95054
(US).

(72) Inventors: CHOU, Joey; 8825 N. 85th Place, Scottsdale,
Arizona 85258 (US). YAO, Yizhi; 2352 W. Desertbroom
Pl., Chandler, Arizona 85248 (US).

(74) Agent: POLIS, Brian, C.; Eschweiler & Potashnik, LLC,
629 Euclid Avenue, Suite 1000, Cleveland, Ohio 44114
(US).

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

(54) Title: TECHNIQUES RELATED TO INTERFACE BETWEEN NEXT GENERATION NODEB CENTRAL UNIT AND NEXT
GENERATION NODEB DISTRIBUTED UNIT

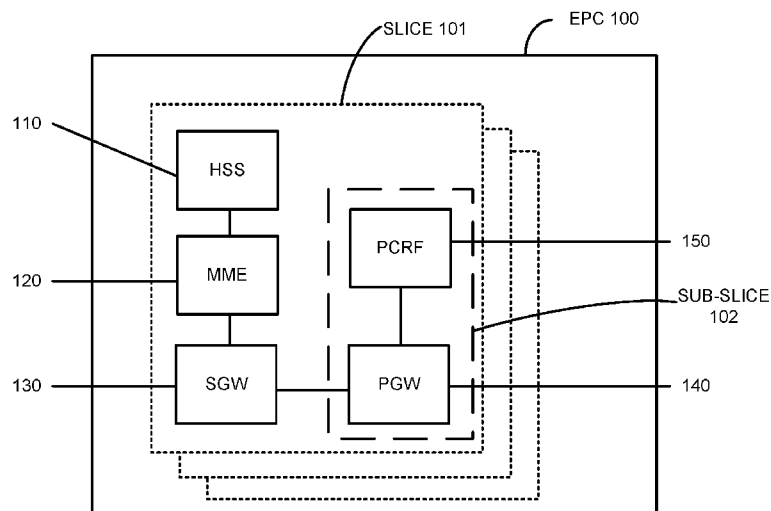


FIG. 1

(57) Abstract: Techniques for employing network function virtualization in connection with a gNB (next generation NodeB) having split functionality are discussed. A first set of embodiments discussed herein can facilitate operation of gNB-CU and gNB-DU in connection with reference point Itf-N. A second set of embodiments discussed herein can facilitate techniques for updating latency and bandwidth requirements of the interface between gNB-CU and gNB-DU. Various embodiments discussed herein can belong to the first set of embodiments, the second set of embodiments, or both.



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

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

TECHNIQUES RELATED TO INTERFACE BETWEEN NEXT GENERATION NODEB CENTRAL UNIT AND NEXT GENERATION NODEB DISTRIBUTED UNIT

REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Applications No. 62/539,925 filed August 1, 2017, entitled "UPDATING FORWARDING GRAPH FOR RAN VIRTUALIZED NETWORK FUNCTION", and No. 62/539,936 filed August 1, 2017, entitled "MANAGING THE INTERFACE BETWEEN NEXT GENERATION NODEB CENTRAL UNIT AND NEXT GENERATION NODEB DISTRIBUTED UNIT OF A NEXT GENERATION RADIO ACCESS NETWORK", the contents of which are herein incorporated by reference in their entirety.

FIELD

[0002] The present disclosure relates to core network technology of a communication network, and more specifically to techniques for employing network function virtualization in connection with a gNB (next generation NodeB) having split functionality.

BACKGROUND

[0003] Network Function Virtualization (NFV) involves the replacement of physical network nodes with Virtual Network Functions (VNFs) implemented via Virtualized Resources (VRs) that perform the same function as the physical node. In 5G, the NG-RAN is split into gNB-CU and gNB-DU, where gNB-CU is a Virtualized Network Function (VNF) that is deployed in the cloud, gNB-DU is implemented in the vertical hardware containing radio functionalities to interface to User Equipment (UE).

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] **FIG. 1** is a diagram illustrating components of a network in accordance with some embodiments.

[0005] **FIG. 2** is a block diagram illustrating components, according to some example embodiments, able to read instructions from a machine-readable or computer-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein.

[0006] **FIG. 3** is a diagram of an example architecture that facilitates support of lifecycle management by a 3GPP (Third Generation Partnership Project) management system, according to various aspects described herein.

[0007] **FIG. 4** is a block diagram of a system employable by a Network Manager (NM) that facilitates employing network function virtualization in connection with a gNB (next generation NodeB) having split functionality, according to various aspects described herein.

[0008] **FIG. 5** is a block diagram of a system employable by a network Element Manager (EM) that facilitates employing network function virtualization in connection with a gNB having split functionality, according to various aspects described herein.

[0009] **FIG. 6** is a block diagram of a system employable by a Network Function Virtualization (NFV) Orchestrator (NFVO) that facilitates employing network function virtualization in connection with a gNB having split functionality, according to various aspects described herein.

[0010] **FIG. 7** is a diagram of an example architecture for a gNB comprising a virtualized gNB-CU (Core Unit) and non-virtualized gNB-DU(s) (Distributed Unit(s)) along with the relation between the gNB-CU and CN (Core Network), according to various aspects discussed herein.

[0011] **FIG. 8** is a diagram of the structure of a NSD (Network Service Descriptor), according to various aspects discussed herein.

[0012] **FIG. 9** is a diagram of the Containment/Naming and Association relation for a gNB with functional split, according to various aspects discussed herein.

[0013] **FIG. 10** is a diagram of the inheritance relations for a gNB with functional split, according to various aspects discussed herein.

[0014] **FIG. 11** is a flow diagram illustrating an example method of NSD on-board, according to various aspects discussed herein.

[0015] **FIG. 12** is a flow diagram illustrating an example method for NSD update, according to various aspects discussed herein.

DETAILED DESCRIPTION

[0016] The present disclosure will now be described with reference to the attached drawing figures, wherein like reference numerals are used to refer to like elements throughout, and wherein the illustrated structures and devices are not necessarily drawn to scale. As utilized herein, terms “component,” “system,” “interface,” and the like are

intended to refer to a computer-related entity, hardware, software (e.g., in execution), and/or firmware. For example, a component can be a processor (e.g., a microprocessor, a controller, or other processing device), a process running on a processor, a controller, an object, an executable, a program, a storage device, a computer, a tablet PC and/or a user equipment (e.g., mobile phone, etc.) with a processing device. By way of illustration, an application running on a server and the server can also be a component. One or more components can reside within a process, and a component can be localized on one computer and/or distributed between two or more computers. A set of elements or a set of other components can be described herein, in which the term “set” can be interpreted as “one or more.”

[0017] Further, these components can execute from various computer readable storage media having various data structures stored thereon such as with a module, for example. The components can communicate via local and/or remote processes such as in accordance with a signal having one or more data packets (e.g., data from one component interacting with another component in a local system, distributed system, and/or across a network, such as, the Internet, a local area network, a wide area network, or similar network with other systems via the signal).

[0018] As another example, a component can be an apparatus with specific functionality provided by mechanical parts operated by electric or electronic circuitry, in which the electric or electronic circuitry can be operated by a software application or a firmware application executed by one or more processors. The one or more processors can be internal or external to the apparatus and can execute at least a part of the software or firmware application. As yet another example, a component can be an apparatus that provides specific functionality through electronic components without mechanical parts; the electronic components can include one or more processors therein to execute software and/or firmware that confer(s), at least in part, the functionality of the electronic components.

[0019] Use of the word exemplary is intended to present concepts in a concrete fashion. As used in this application, the term “or” is intended to mean an inclusive “or” rather than an exclusive “or”. That is, unless specified otherwise, or clear from context, “X employs A or B” is intended to mean any of the natural inclusive permutations. That is, if X employs A; X employs B; or X employs both A and B, then “X employs A or B” is satisfied under any of the foregoing instances. In addition, the articles “a” and “an” as used in this application and the appended claims should generally be construed to

mean “one or more” unless specified otherwise or clear from context to be directed to a singular form. Furthermore, to the extent that the terms “including”, “includes”, “having”, “has”, “with”, or variants thereof are used in either the detailed description and the claims, such terms are intended to be inclusive in a manner similar to the term “comprising.”

[0020] As used herein, the term “circuitry” may refer to, be part of, or include an Application Specific Integrated Circuit (ASIC), an electronic circuit, a processor (shared, dedicated, or group), and/or memory (shared, dedicated, or group) that execute one or more software or firmware programs, a combinational logic circuit, and/or other suitable hardware components that provide the described functionality. In some embodiments, the circuitry may be implemented in, or functions associated with the circuitry may be implemented by, one or more software or firmware modules. In some embodiments, circuitry may include logic, at least partially operable in hardware.

[0021] Embodiments described herein may be implemented into a system using any suitably configured hardware and/or software. **FIG. 1** illustrates components of a network in accordance with some embodiments. In various aspects, part(s) or all of one or more of the components illustrated in connection with FIG. 1 can be implemented as virtual network functions (VNFs) in connection with various aspects described herein. An Evolved Packet Core (EPC) network 100 is shown to include a Home Subscriber Server (HSS) 110, a Mobility Management Entity (MME) 120, a Serving Gateway (SGW) 130, a Packet Data Network (PDN) Gateway (PGW) 140, a Policy and Charging Rules Function (PCRF) 150.

[0022] The HSS 110 comprises one or more databases for network users, including subscription-related information to support the network entities' handling of communication sessions. For example, the HSS 110 may provide support for routing/roaming, authentication, authorization, naming/addressing resolution, location dependencies, etc. The EPC network 100 may comprise one or several HSSs 110, depending on the number of mobile subscribers, on the capacity of the equipment, on the organization of the network, etc.

[0023] The MME 120 is similar in function to the control plane of legacy Serving General packet radio service (GPRS) Support Nodes (SGSN). The MMEs 120 manage mobility aspects in access such as gateway selection and tracking area list management. The EPC network 100 may comprise one or several MMEs 120

[0024] The SGW 130 terminates the interface toward an Evolved UMTS (Universal Mobile Telecommunications System) Terrestrial Radio Access Network (E-UTRAN), and routes data packets between the E-UTRAN and the EPC network 100. In addition, the SGW 130 may be a local mobility anchor point for inter-eNodeB handovers and also may provide an anchor for inter-3GPP mobility. Other responsibilities may include lawful intercept, charging, and some policy enforcement.

[0025] The PGW 140 terminates an SGi interface toward the PDN. The PGW 140 routes data packets between the EPC network 100 and external networks, and may be a node for policy enforcement and charging data collection. The PCRF 150 is the policy and charging control element of the EPC network 100. In a non-roaming scenario, there may be a single PCRF in the Home Public Land Mobile Network (HPLMN) associated with a User Equipment's (UE) Internet Protocol Connectivity Access Network (IP-CAN) session. In a roaming scenario with local breakout of traffic, there may be two PCRFs associated with a UE's IP-CAN session: a Home PCRF (H-PCRF) within a HPLMN and a Visited PCRF (V-PCRF) within a Visited Public Land Mobile Network (VPLMN). The PCRF 150 may be communicatively coupled to an application server (alternatively referred to as application function (AF)). Generally, the application server is an element offering applications that use Internet Protocol (IP) bearer resources with the core network (e.g., UMTS Packet Services (PS) domain, Long Term Evolution (LTE) PS data services, etc.). The application server may signal the PCRF 150 to indicate a new service flow and selecting the appropriate Quality of Service (QoS) and charging parameters. The PCRF 150 may provision this rule into a Policy and Charging Enforcement Function (PCEF) (not shown) with the appropriate traffic flow template (TFT) and QoS class of identifier (QCI), which commences the QoS and charging as specified by the application server.

[0026] The components of the EPC 100 may be implemented in one physical node or separate physical nodes. In some embodiments, Network Functions Virtualization (NFV) is utilized to virtualize any or all of the above described network node functions via executable instructions stored in one or more computer readable storage mediums (described in further detail below). A logical instantiation of the EPC network 100 may be referred to as a network slice 101. A logical instantiation of a portion of the EPC network 100 may be referred to as a network sub-slice 102 (e.g., the network sub-slice 102 is shown to include the PGW 140 and the PCRF 150).

[0027] FIG. 2 is a block diagram illustrating components, according to some example embodiments, able to read instructions from a machine-readable or computer-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein. Specifically, FIG. 2 shows a diagrammatic representation of hardware resources 200 including one or more processors (or processor cores) 210, one or more memory/storage devices 220, and one or more communication resources 230, each of which are communicatively coupled via a bus 240. For embodiments where node virtualization (e.g., NFV) is utilized, a hypervisor 202 may be executed to provide an execution environment for one or more network slices/sub-slices to utilize the hardware resources 200.

[0028] The processors 210 (e.g., a central processing unit (CPU), a reduced instruction set computing (RISC) processor, a complex instruction set computing (CISC) processor, a graphics processing unit (GPU), a digital signal processor (DSP) such as a baseband processor, an application specific integrated circuit (ASIC), a radio-frequency integrated circuit (RFIC), another processor, or any suitable combination thereof) may include, for example, a processor 212 and a processor 214. The memory/storage devices 220 may include main memory, disk storage, or any suitable combination thereof.

[0029] The communication resources 230 may include interconnection and/or network interface components or other suitable devices to communicate with one or more peripheral devices 204 and/or one or more databases 206 via a network 208. For example, the communication resources 230 may include wired communication components (e.g., for coupling via a Universal Serial Bus (USB)), cellular communication components, Near Field Communication (NFC) components, Bluetooth® components (e.g., Bluetooth® Low Energy), Wi-Fi® components, and other communication components.

[0030] Instructions 250 may comprise software, a program, an application, an applet, an app, or other executable code for causing at least any of the processors 210 to perform any one or more of the methodologies discussed herein. The instructions 250 may reside, completely or partially, within at least one of the processors 210 (e.g., within the processor's cache memory), the memory/storage devices 220, or any suitable combination thereof. Furthermore, any portion of the instructions 250 may be transferred to the hardware resources 200 from any combination of the peripheral devices 204 and/or the databases 206. Accordingly, the memory of processors 210, the

memory/storage devices 220, the peripheral devices 204, and the databases 206 are examples of computer-readable and machine-readable media.

[0031] Referring to **FIG. 3**, illustrated is a diagram of an example architecture that facilitates generation and communication of performance measurements related to virtualized resources, according to various aspects described herein. The system illustrated in FIG. 3 comprises a Network Manager (NM) 302, Operation Support Systems (OSS)/Business Support Systems 304, network Element Manager (EM) 306, Domain Manager (DM) 308, Network Function Virtualization (NFV) Management and Orchestration (MANO) components (NFV Orchestrator (NFVO) 310, VNF Manager (VNFM) 312, and Virtualized Infrastructure Manager (VIM) 314), a set of Virtualized Network Functions (VNFs) 316, virtualized by Virtualized resources (VRs) of a NFV Infrastructure (NFVI) 318 (which can comprise a hypervisor such as hypervisor 202 and hardware resources such as hardware resources 200), and optionally one or more Network Entities (NEs) 320i that can implement Physical Network Functions (PNFs). The lines between these entities indicate reference points or other communication connections that can facilitate data exchange between these entities (some of which are labeled, such as reference point ltf-N, etc.).

[0032] In 5G (Fifth Generation), the gNB (next generation Node B) is split into Central Unit (CU) and Distributed Unit (DU), wherein the gNB-CU can be a Virtualized Network Function (VNF) that can be deployed in the cloud, and the gNB-DU can be implemented in the vertical hardware comprising radio functionalities to interface to a UE (User Equipment).

[0033] Various embodiments discussed herein can facilitate operation of gNB(s) split between gNB-CU(s) and gNB-DU(s) in connection with a NR (New Radio) RAN (Radio Access Network). A first set of embodiments discussed herein can facilitate operation of gNB-CU and gNB-DU in connection with a network resource model. A second set of embodiments discussed herein can facilitate techniques for updating latency and bandwidth requirements of the interface between gNB-CU and gNB-DU. Various embodiments discussed herein can belong to the first set of embodiments, the second set of embodiments, or both.

[0034] Referring to **FIG. 4**, illustrated is a block diagram of a system 400 employable by a Network Manager (NM) that facilitates techniques for employing network function virtualization in connection with a gNB (next generation NodeB) having split functionality, according to various aspects described herein. System 400 can comprise

one or more processors 410 (e.g., which can comprise one or more of processor(s) 210, etc., each of which can comprise processing circuitry and one or more interfaces for send/receive data to/from each other or other circuitry, such as a memory interface to send/receive data to/from memory 430, a communication circuitry interface to send/receive data to/from communication circuitry 420, etc.), communication circuitry 420 (which can facilitate communication of data via one or more reference points, networks, etc., and can comprise communication resource(s) 230, etc.), and memory 430 (which can comprise any of a variety of storage mediums and can store instructions and/or data associated with at least one of the one or more processors 410 or communication circuitry 420, and can comprise memory/storage device(s) 220 and/or cache memory of processor(s) 410, etc.). In some aspects, the one or more processors 410, the communication circuitry 420, and the memory 430 can be included in a single device, while in other aspects, they can be included in different devices, such as part of a distributed architecture. As described in greater detail below, system 400 can be employed by a NM to facilitate implementation of embodiment(s) of a first and/or second set of embodiments discussed herein, in various embodiments.

[0035] Referring to **FIG. 5**, illustrated is a block diagram of a system 500 employable by a network Element Manager (EM) that facilitates techniques for employing network function virtualization in connection with a gNB (next generation NodeB) having split functionality, according to various aspects described herein. System 500 can comprise one or more processors 510 (e.g., which can comprise one or more of processor(s) 210, etc., each of which can comprise processing circuitry and one or more interfaces for send/receive data to/from each other or other circuitry, such as a memory interface to send/receive data to/from memory 530, a communication circuitry interface to send/receive data to/from communication circuitry 520, etc.), communication circuitry 520 (which can facilitate communication of data via one or more reference points, networks, etc., and can comprise communication resource(s) 230, etc.), and memory 530 (which can comprise any of a variety of storage mediums and can store instructions and/or data associated with at least one of the one or more processors 510 or communication circuitry 520, and can comprise memory/storage device(s) 220 and/or cache memory of processor(s) 510, etc.). In some aspects, the one or more processors 510, the communication circuitry 520, and the memory 530 can be included in a single device, while in other aspects, they can be included in different devices, such as part of a distributed architecture. As described in greater detail below, system 500 can be

employed by an EM to facilitate implementation of embodiment(s) of a first and/or second set of embodiments discussed herein, in various embodiments.

[0036] Referring to **FIG. 6**, illustrated is a block diagram of a system 600 employable by a Network Function Virtualization (NFV) Orchestrator (NFVO) that facilitates techniques for employing network function virtualization in connection with a gNB (next generation NodeB) having split functionality, according to various aspects described herein. System 600 can comprise one or more processors 610 (e.g., which can comprise one or more of processor(s) 210, etc., each of which can comprise processing circuitry and one or more interfaces for send/receive data to/from each other or other circuitry, such as a memory interface to send/receive data to/from memory 630, a communication circuitry interface to send/receive data to/from communication circuitry 620, etc.), communication circuitry 620 (which can facilitate communication of data via one or more reference points, networks, etc., and can comprise communication resource(s) 230, etc.), and memory 630 (which can comprise any of a variety of storage mediums and can store instructions and/or data associated with at least one of the one or more processors 610 or communication circuitry 620, and can comprise memory/storage device(s) 220 and/or cache memory of processor(s) 610, etc.). In some aspects, the one or more processors 610, the communication circuitry 620, and the memory 630 can be included in a single device, while in other aspects, they can be included in different devices, such as part of a distributed architecture. As described in greater detail below, system 600 can be employed by a NFVO to facilitate implementation of embodiment(s) of a first and/or second set of embodiments discussed herein, in various embodiments.

Managing the Interface between gNB-CU and gNB-DU of a NG (Next Generation)-RAN

[0037] RAN3 (RAN (Radio Access Network) WG3 (Working Group #3)) has selected Option 2 (based on centralized PDCP (Packet Data Convergence Protocol)/RRC (Radio Resource Control) and distributed RLC (Radio Link Control)/MAC (Medium Access Control)/PHY (Physical layer)) for Higher-Layer Functionality split between CU and DU for normative work in Rel-15 (3GPP (Third Generation Partnership Project) Release 15).

[0038] An operator can manage the CU and DU over reference point Itf-N, for example, to manage the coverage, capacity and/or frequency bands of a specific DU, or to manage a feature (e.g., mobility, SON (Self-Organizing Network)) of a specific CU.

Management of the CU and DU over reference point ltf-N should remain valid regardless of whether: (1) CU and DU are provided by the same vendor or not; and/or (2) DU is managed directly by EM or through CU.

[0039] In order to support these capabilities, the CU and DU can be modelled in the network resource model. A first set of embodiments discussed herein can facilitate operation of a gNB comprising virtualized and non-virtualized portions.

[0040] The use cases and capabilities for establishing the relation between the virtualized part and the non-virtualized part of a gNB are addressed in clauses 4.1.1 and 5.1 of draft 3GPP TR (Technical Report) 32.864, however, there are no existing potential solutions for these use cases. Various embodiments of the first set of embodiments can facilitate solutions for uses cases for establishing the relation between the virtualized part and the non-virtualized part of a gNB.

[0041] The use cases and capabilities on establishing relation between a CN (Core Network) NF (Network Function) and the gNB are addressed in clauses 4.1.2 and 5.1 of draft TR 32.864, however, there are no existing potential solutions for these use cases. Various embodiments of the first set of embodiments can facilitate solutions for uses cases for establishing the relation between the CN NF and the gNB.

[0042] The use cases and capabilities on instantiating NS (Network Service(s)) comprising CN VNF and gNB are addressed in clauses 4.2.2 and 5.2 of draft TR 32.864. Various embodiments of the first set of embodiments can facilitate solutions for these uses cases and capabilities for instantiating NS comprising CN VNF and gNB.

[0043] The use cases and capabilities on update of transport network requirements are addressed in clauses 4.2.3 and 5.2 of draft TR 32.864. Various embodiments of the first set of embodiments can facilitate solutions for these uses cases and capabilities for update of transport network requirements.

[0044] Referring to **FIG. 7**, illustrated is a diagram showing an example architecture for a gNB comprising a virtualized gNB-CU 720 and non-virtualized gNB-DU(s) 710_i along with the relation between the gNB-CU 720 and CN 730, according to various aspects discussed herein.

[0045] Referring to **FIG. 8**, illustrated is a diagram showing the structure of the NSD (Network Service Descriptor), according to various aspects discussed herein.

NRMs (Network Resource Models) for gNB with Functional Split

[0046] To enable an NM (e.g., employing system 400) to manage the gNB-CU, gNB-DU, and the F1 interface between gNB-CU and gNB-DU over ltf-N (e.g., to manage: the coverage, capacity or frequency bands of a gNB-DU; a specific feature (e.g., mobility, SON) of a gNB-CU; or the transport network requirements between gNB-CU and gNB-DU), the gNB-CU, gNB-DU and end point of F1 interface can be modelled over ltf-N.

[0047] Referring to **FIG. 9**, shows the Containment/Naming and Association relation for a gNB with functional split, according to various aspects discussed herein. The gNB-CU can be modelled as IOC (Information Object Class) GNBcuFunction (gNB-CU Function), the gNB-DU can be modelled as IOC GNBduFunction (gNB-DU Function), the end point of F1 interface is modelled as IOC EP_F1 (End Point of F1). The IOC GNBcuFunction and IOC GNBduFunction can be comprised within the IOC ManagedElement (managed element). The IOC GNBcuFunction can comprise one or more than one IOC EP_F1, which can be associated with the IOC GNBduFunction, respectively.

[0048] Referring to **FIG. 10**, illustrated is a diagram showing the inheritance relations for a gNB with functional split, according to various aspects discussed herein. To support the virtualization of gNB-CU, the IOC GNBcuFunction can be inherited from the IOC ManagedFunction that comprises the VNF related attribute(s). The gNB-DU part of the gNB is non-virtualized, so the IOC GNBduFunction can be inherited from the IOC Top. The IOC EP_F1 can be inherited from IOC EP_RP.

Potential Solutions on Configuration Management

Potential Solution on Establishing Relation Between Virtualized Part and Non-Virtualized Part of a gNB

[0049] This section provides a potential solution to support the use case and capabilities for establishing a relation between the virtualized part (gNB-CU) and non-virtualized part (gNB-DU) of a gNB.

[0050] In various embodiments, a NM (e.g., employing system 400) can establish the relation by: (1) creating (e.g., via processor(s) 410) the MOI (Managed Object Instance) of the end point of the F1 interface between the gNB-CU and gNB-DU (e.g., EP_F1) and (2) configuring (e.g., via processor(s) 410) the end point MOI to associate with the far end MOI of gNB-DU (e.g., gNBduFunction) or the far end MOI of gNB-CU (e.g., gNBcuFunction).

[0051] One or more EMs (e.g., employing system(s) 500) can configure (e.g., via processor(s) 410) the gNB-CU and/or the gNB-DU to establish the relation. Depending on the embodiment, the EM of the eNB-CU and the EM of the gNB-DU can be the same or different.

Potential Solution on Establishing Relation between the CN NF and the gNB

[0052] This section provides a potential solution to support the use case and capabilities for establishing a relation between the CN NF and the gNB.

[0053] In various embodiments, a NM (e.g., employing system 400) can establish the relation by (1) creating (e.g., via processor(s) 410) the MOI of end point of the reference point between the CN NF and the gNB CU and (2) configuring (e.g., via processor(s) 410) the end point MOI to associate with the MOI of the CN NF.

[0054] An EM (e.g., employing system 500) can configure (e.g., via processor(s) 510) the CU to establish the relation with the CN NF.

[0055] In various aspects, the CN NF can be configured to establish the relation based on the RAN3 definition of the reference point between the CN NF and the gNB.

Potential Solution for Instantiating NS comprising CN VNF and gNB

[0056] This section provides a potential solution to support the use case and capabilities for instantiating the NS comprising the CN VNF and the gNB.

[0057] In various embodiments, a NM (e.g., employing system 400) can request (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over reference point Os-Ma-nfvo) a NFVO (e.g., employing system 600, which can receive the request via communication circuitry 620 and process it via processor(s) 610) to create (e.g., via processor(s) 610) a NS identifier for the NSD that can reference to the VNFD (VNF Descriptor) of the VN NF, the VNFD of the virtualized part of the gNB (e.g., gNB-CU), and/or the PNFD (PNF (Physical Network Function) Descriptor) of the non-virtualized part of the gNB (e.g., gNB-DU) if any.

[0058] The NFVO can respond to the NM (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) about the successful creation with the parameter nsInstanceld (NS instance identifier).

[0059] The NM can request the NFVO to instantiate the NS identified by nsInstanceld (e.g., via a request generated by processor(s) 410, sent via

communication circuitry 420 over the Os-Ma-nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the parameter additionalParamForVnf (additional parameter for VNF) providing the information for the CN NF and virtualized part of the gNB, parameter pnfInfo (PNF information) providing the information of the non-virtualized part of the gNB, and possibly other parameters.

[0060] The NFVO can send the NS lifecycle change notification to the NM (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the start of the NS instantiation procedure.

[0061] The NFVO can instantiate (e.g., via processor(s) 610) the NS containing the CN VNF and the gNB, based on the information provide by NM and one or more of the information provided in the NSD, VNF package, and PNFD.

[0062] NFVO can sends the NS Lifecycle Change notification to NM indicating the result of NS instantiation (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

Potential solution for updating transport network requirements

Potential solution for updating latency and bandwidth requirements

[0063] This section provides a potential solution to support the Use Case and capabilities on update of transport network requirements.

[0064] This solution can apply to scenarios wherein both the latency and bandwidth requirements are updated.

[0065] The NM can invoke (e.g., via processor(s) 410) an Update NSD operation to request the NFVO to update the NSD (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), wherein the parameter nsdInfoId (NSD information identifier) can indicate the NSD to be updated, and the parameter NSD can indicate the updated NSD, which contains the following attributes to be updated:

- 1) vnffgd (VNFFG (VNF Forwarding Graph) Descriptor) that can comprise the following attributes:
 - a) vnffgdId (VNFFGD identifier) that uniquely identifies a VNFFGD;

- b) vnfdId (VNFD identifier) that identifies the VNFD of the constituent VNF realizing the virtualized part of gNB;
 - c) pnfdId (PNFD identifier) that identifies the PNFD of the constituent PNF realizing the non-virtualized part of gNB;
 - d) virtualLinkDesc (virtual link descriptor) that can comprise the following attribute:
 - i) virtualLinkDf (Virtual link deployment flavor) that can comprise the following attribute:
 - (1) qoS (quality of service) that can comprise the latency attribute, which can indicate the latency requirement for the F1 interface.
 - e) cpdPoolId (Connection Point Descriptor Pool identifier) that references to a pool of descriptors of connection points attached to the constituent VNF(s) (realizing the virtualized part of gNB) and PNF(s) (realizing the non-virtualized part of gNB).
 - f) nfpd (network forwarding path descriptor) (optional) that specifies the network forwarding path associated to the VNFFG.
- 2) nsDf (NS deployment flavor) that can comprise the following attributes:
- a) virtualLinkProfile (virtual link profile) that can comprise the following attributes:
 - i) maxBitrateRequirements (maximum bitrate requirements) indicating the maximum bandwidth requirement for the F1 interface;
 - ii) minBitrateRequirements (minimum bitrate requirements) attributes, indicating the minimum bandwidth requirement for the F1 interface;
 - b) nsInstantiationLevel (NS instantiation level) to indicate the NsLevel(s) (NS level(s) within the NS deployment flavor, and each NsLevel that contain the following attributes:
 - i) virtualLinkToLevelMapping (virtual link to level mapping) that can comprise the bitRateRequirements (bitrate requirements) indicating the bandwidth requirement for the F1 interface.

[0066] The NFVO can validate (e.g., via processor(s) 610) whether the bitRateRequirements is in the range between minBitrateRequirements and maxBitrateRequirements, and can return the nsdInfoId (NSD information identifier) indicating the successful NSD update to NM if the validation went through (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); otherwise, the NFVO can return an out of range error to NM indicating the bitRateRequirements is out of the range between

minBitrateRequirements and maxBitrateRequirements (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0067] The NM can invoke (e.g., via processor(s) 410) the Update NS operation to request the NFVO to update the latency and bandwidth requirements for the F1 interface (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the following parameters: (a) updateType (Update Type) = “AssocNewNsdVersion” (“Associate New NSD version”) to associate the NS instance with the updated NSD; and (b) assocNewNsdVersionData (Associate new NSD version data) to indicate the NSD with which the NS instance will be associated.

[0068] The NFVO can send the NS lifecycle change notification to the NM indicating the start of NS update procedure (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0069] The NFVO can update the NS (e.g., via processor(s) 610), based on the information provide by NM and the information provided in the updated NSD.

[0070] The NFVO can send the NS Lifecycle Change notification to the NM indicating the result of NS update (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

Potential solution for updating bandwidth requirement

[0071] This section provides a potential solution to support the Use Case and capabilities on update of transport network requirements.

[0072] This solution can apply to scenarios wherein only the bandwidth requirement is updated.

[0073] The NM can invoke (e.g., via processor(s) 410) the Update the NSD operation to request the NFVO to update the NSD (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610),

with the parameter nsdInfold indicating the NSD to be updated, and the parameter NSD indicating the updated NSD, which can comprise the following attributes to be updated:

- 1) nsDf that can comprise the following attributes:
 - a) virtualLinkProfile that can comprise the following attributes:
 - i) maxBitrateRequirements, indicating the maximum bandwidth requirement for the F1 interface;
 - ii) minBitrateRequirements, indicating the minimum bandwidth requirement for the F1 interface;
 - b) nsInstantiationLevel that can indicate the NsLevel(s) within the NS deployment flavor, wherein each NsLevel can comprise the following attributes:
 - i) virtualLinkToLevelMapping that can comprise the bitRateRequirements indicating the bandwidth requirement for the F1 interface.

[0074] The NFVO can validate (e.g., via processor(s) 610) whether the bitRateRequirements is in the range between minBitrateRequirements and maxBitrateRequirements, and can return the nsdInfold indicating the successful NSD update to NM if the validation went through (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); otherwise, the NFVO can return an out of range error to the NM indicating the bitRateRequirements is out of the range between minBitrateRequirements and maxBitrateRequirement (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0075] The NM can invoke (e.g., via processor(s) 410) the Update NS operation to request the NFVO to update the bandwidth requirement for the F1 interface (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the following parameters: (a) updateType = "ChangeNsDf" ("change NS deployment flavor") to change the deployment flavor for the NS instance.

[0076] The NFVO can send the NS lifecycle change notification to NM indicating the start of NS update procedure (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0077] The NFVO can update the NS (e.g., via processor(s) 610), based on the information provided by the NM and the information provided in the updated NSD.

[0078] The NFVO can send a NS Lifecycle Change notification to the NM indicating the result of the NS update (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

Potential solution for updating latency requirement

[0079] This section provides a potential solution to support the Use Case and capabilities on update of transport network requirements.

[0080] This solution can apply to scenarios wherein only the latency requirement is updated.

[0081] The NM can invoke (e.g., via processor(s) 410) the Update NSD operation to request NFVO to update the NSD (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the parameter nsdInfold indicating NSD to be updated, and the parameter NSD indicating the updated NSD, which comprises the following attributes to be updated:

- 1) vnffgd that can comprise the following attributes:
 - a) vnffgdId that uniquely identifies a VNFFGD;
 - b) vnfdId that identifies the VNFD of the constituent VNF realizing the virtualized part of gNB;
 - c) pnfdId that identifies the PNFD of the constituent PNF realizing the non-virtualized part of gNB;
 - d) virtualLinkDesc that can comprise the following attribute:
 - i) virtualLinkDf that can comprise the following attribute:
 - (1) qoS that can comprise the latency attribute, indicating the latency requirement for the F1 interface.
 - e) cpdPoolId that references to a pool of descriptors of connection points attached to the constituent VNF(s) (realizing the virtualized part of gNB) and PNF(s) (realizing the non-virtualized part of gNB);
 - f) nfpd (optional) that specifies the network forwarding path associated to the VNFFG.

[0082] NFVO can update the Vnffgd (e.g., via processor(s) 610), and can return the nsdInfold indicating the updated NSD to the NM (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0083] The NM can invoke (e.g., via processor(s) 410) the Update NS operation to request NFVO to update the latency requirement for the F1 interface (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the following parameters: (a) updateType = "UpdateVnffg" update the VNFFG, including the VL between gNB-CU and gNB-DU for the NS instance; and (b) updateVnffg to indicate the updated VNFFGD for the NS instance.

[0084] The NFVO can send the NS lifecycle change notification to NM indicating the start of NS update procedure (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0085] The NFVO can update the NS (e.g., via processor(s) 610), based on the information provide by NM and the information provided in the updated NSD.

[0086] The NFVO can send the NS Lifecycle Change notification to the NM indicating the result of the NS update (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

VNNFGD Onboard and Update

[0087] A second set of embodiments discussed herein can facilitate techniques for updating latency and bandwidth requirements of the interface between gNB-CU and gNB-DU.

NG-RAN Deployment

[0088] Deploying a NG-RAN that comprises gNB-CU and gNB-DU can be performed by an operator as follows: (1) deploy the gNB-DU with RF equipment; (2) on-board the NSD (Network Service Descriptor) for gNB-CU VNF with the latency and bandwidth requirements for the interface between the gNB-CU and gNB-DU; and (3) instantiate the NS based on the on-boarded NSD referencing the VNFD of the gNB-CU and/or the PNFD of the gNB-DU.

[0089] Referring to **FIG. 11**, illustrated is a flow diagram showing an example method of NSD on-board, according to various aspects discussed herein.

[0090] At 1101, the NM (e.g., employing system 400) can request the NFVO (e.g., employing system 600) to on-board the NSD with the virtualLinkDesc and nsDf attributes to specify the latency and/or bandwidth requirements, respectively, for the interface between gNB-CU and gNB-DU (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610).

[0091] The nsDf attribute can define the bandwidth requirement in terms of minBitrateRequirements, maxBitrateRequirements and bitRateRequirement, wherein bitRateRequirement should be in the range between minBitrateRequirements and maxBitrateRequirements of the specific NS deployment flavour. At 1102, the NFVO can validate (e.g., via processor(s) 610) the NS deployment flavor (e.g., as indicated via the nsDf attribute) to determine if the bitRateRequirement is in the range between minBitrateRequirements and maxBitrateRequirements.

[0092] At 1103, if bitRateRequirement is within the range between minBitrateRequirements and maxBitrateRequirements, then the NFVO can on-board (e.g., via processor(s) 610) the NSD, and can return the nsdInfold to the NM (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0093] Otherwise, at 1104, if bitRateRequirement is not within the range between minBitrateRequirements and maxBitrateRequirements, the NFVO can return an out of range error (of the bitRateRequirement parameter) to the NM to indicate the NSD on-board operation failed (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0094] Referring to **FIG. 12**, illustrated is a flow diagram showing an example method for NSD update, according to various aspects discussed herein. The method of FIG. 12 can be employed by an operator in various scenarios to update the transport network requirements (e.g., latency and bandwidth) of a NG-RAN.

[0095] At 1201, the NM can request the NFVO to update the NSD (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed

by processor(s) 610) with the following attributes to specify the latency and bandwidth requirements, respectively, for the interface between gNB-CU and gNB DU: (a) virtualLinkDesc, which can comprise the latency requirement; (b) VNFFDG (VNF Forwarding Graph), which can comprise the virtualLinkDesc; and (c) nsDf, which can comprise the bandwidth requirement via the bitRateRequirement parameter, and the range of acceptable bandwidth requirements (e.g., via the minBitrateRequirements and maxBitrateRequirements parameters).

[0096] At 1202, the NFVO can validate (e.g., via processor(s) 610) whether the bandwidth requirement is within the range. If the bandwidth requirement is within the range, the NFVO can update the NSD (e.g., via processor(s) 610), and can return the nsdInfo to indicate the NSD has been successfully updated (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410). Otherwise, if the bandwidth requirement is not within the range, the NFVO can return an out of range error (of the bitRateRequirement parameter) to the NM to indicate the NSD on-board operation was failed (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0097] At 1203, if the NSD has been successfully updated, the NM can request the NFVO to update the NS with the NSD being successfully updated (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610).

[0098] At 1204, the NFVO can return a notification to the NM to indicate that the NS update has been started (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[0099] At 1205, the NFVO can update the NS (e.g., via processor(s) 610).

[00100] At 1206, the NFVO can return a notification to the NM to indicate the result of the NS update (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

Life Cycle Management Use Cases

NSD On-boarding of NS Deployment Flavor

Issues

[00101] The interface between the gNB-CU and the gNB-DU has specific latency and bandwidth requirements, based on the functional split options. The NSD information element comprises the virtualLinkDesc and nsDf attributes that are used in NSD On-boarding operation to specify the latency and bandwidth requirements for the interface between gNB-CU and gNB DU, respectively.

Pre-conditions

[00102] (a) The operator decides to on-board the NSD and (b) The underlying transport network requirements for the selected CU-DU functional split option are known.

Description

[00103] The NM requests NFVO to on-board the NSD with the virtualLinkDesc and nsDf attributes to specify the latency and bandwidth requirements, respectively, for the interface between gNB-CU and gNB DU (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610).

[00104] The nsDf attribute defines the bandwidth requirement in terms of minBitrateRequirements, maxBitrateRequirements and bitRateRequirement, wherein bitRateRequirement should be in the range between minBitrateRequirements and maxBitrateRequirements.

[00105] The NFVO can on-boards the NSD (e.g., via processor(s) 610), and can return the nsdInfold to NM indicating the NSD on-board operation was successful in scenarios wherein the maxBitrateRequirements is in the range (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-Nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); otherwise, the NFVO can return an out of range error to the NM to indicate the NSD on-board operation failed (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-Nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

Post-conditions

[00106] The NSD has been on-boarded successfully or not, based on whether the bandwidth information is valid.

NSD Update of NS Deployment Flavor*Issues*

[00107] The interface between gNB-CU and gNB-DU has specific latency and bandwidth requirements, based on the functional split options. The NSD information element contains the virtualLinkDesc and nsDf attributes that can be used in NSD update operation to change the latency and bandwidth requirements for the interface between gNB-CU and gNB DU, respectively.

[00108] The NSD can only be updated if the bandwidth information contained in the nsDf attribute are valid.

Pre-conditions

[00109] (a) the operator decides to on-board the NSD and (b) the underlying transport network requirements for the selected CU-DU functional split option are known.

Description

[00110] The NM can request NFVO to update the NSD with the virtualLinkDesc and nsDf attributes (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610) to specify the latency and bandwidth requirements, respectively, for the interface between gNB-CU and gNB DU.

[00111] The nsDf attribute defines the bandwidth requirement in terms of minBitrateRequirements, maxBitrateRequirements and bitRateRequirement, wherein bitRateRequirement should be in the range between minBitrateRequirements and maxBitrateRequirements.

[00112] The NFVO can update (e.g., via processor(s) 610) the NSD, and can returns the nsdInfold to the NM (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-Nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the NSD update operation was successful if maxBitrateRequirements is in the range; otherwise,

the NFVO can return an out of range error to the NM to indicate the NSD update operation was failed (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-Nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

Post-conditions

[00113] The NSD has been updated successfully or not, based on whether the bandwidth information is valid.

[00114] REQ-VRAN_Mgmt_LCM-CON-x: the 3GPP management system should be able to receive the result of the NSD On-boarding operation (e.g., success if bandwidth parameter is within the range, error with cause to indicate the bandwidth parameter is out of the range) from the ETSI MANO system.

[00115] REQ-VRAN_Mgmt_LCM-CON-y: the 3GPP management system should be able to receive the result of the NSD update operation (e.g., success if bandwidth parameter is within the range, error with cause to indicate the bandwidth parameter is out of the range) from the ETSI MANO system.

Potential Solution for On-boarding NSD for the gNB

[00116] This section provides a potential solution to support the Use Case and capabilities for on-boarding the NSD for the gNB.

[00117] This section can apply to scenarios wherein the following items can have been on-boarded at the NFVO: (a) the VNF package of the virtualized part of the gNB; (b) the PNFD of the non-virtualized part of the gNB; (c) VNF package(s) of other VNF(s) (besides the virtualized part of gNB) if they are constituents of the NS; and (d) the PNFD of the other PNF(s) (besides the non-virtualized part of gNB) if they are the constituents of the NS.

[00118] The NM can request the NFVO to invoke the On-board NSD operation to on-board the NSD (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), which can comprise the following attributes:

- 1) Reference to the VNFD of the virtualized part of the gNB;
- 2) Reference to the PNFD of the non-virtualized part of the gNB;
- 3) Reference to the VNFD of other constituent VNF(s) (if any) of the NS;

- 4) NFD of the other PNF(s) (besides the non-virtualized part of the gNB)
- 5) Reference to the PNFD of other constituent PNFs (if any) of the NS;
- 6) vnffgd that can comprise the following attribute:
 - a) vnffgdId that uniquely identifies a VNFFGD;
 - b) vnfdId that identifies the VNFD of the constituent VNF realizing the virtualized part of the gNB;
 - c) pnfdId that identifies the PNFD of the constituent PNF realizing the non-virtualized part of the gNB;
 - d) virtualLinkDesc that comprises the following attribute:
 - i) virtualLinkDf that comprises the following attribute:
 - (1) qpS that comprises the latency attribute, indicating the latency requirement for the F1 interface.
 - e) cpdPoolId that references to a pool of descriptors of connection points attached to the constituent VNFs (realizing the virtualized part of the gNB) and PNFs (realizing the non-virtualized part of the gNB).
 - f) nfpd (optional) that specifies the network forwarding path associated to the VNFFG.
- 7) nsDf that comprises the following attribute:
 - a) virtualLinkProfile that comprises the following attributes:
 - i) maxBitrateRequirements indicating the maximum bandwidth requirement for the F1 interface;
 - ii) minBitrateRequirements attributes, indicating the minimum bandwidth requirement for the F1 interface;
 - b) nsInstantiationLevel to indicate the NsLevel(s) within the NS deployment flavour, and each NsLevel that comprises the following attributes:
 - i) virtualLinkToLevelMapping that comprises the bitRateRequirements indicating the bandwidth requirement for the F1 interface.

[00119] The NFVO can validate (e.g., via processor(s) 610) whether the bitRateRequirements is in the range between minBitrateRequirements and maxBitrateRequirements, and can return the nsdInfoId indicating the successful NSD on-boarding to NM (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) if the validation went through; otherwise, the NFVO can return an out of range error to NM indicating the

bitRateRequirements is out of the range between minBitrateRequirements and maxBitrateRequirements (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

Potential Solution for Adding VNFFGD for the gNB

[00120] This section provides a potential solution to support the Use Case and capabilities for adding VNFFGD for the gNB.

[00121] This section can apply to scenarios wherein the NSD can have been on-boarded, but not the VNFFGD.

[00122] The NM can request the NFVO to invoke the Update NSD operation to add the VNFFGD for the gNB (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the following parameters:

- 1) nsdInfol indicating the NSD to be updated;
- 2) NSD indicating the NSD that contains the VNFFGD to be added. The VNFFGD comprises the following attribute:
 - a) vnffgdId that uniquely identifies a VNFFGD;
 - b) vnfdId that identifies the VNFD of the constituent VNF realizing the virtualized part of the gNB;
 - c) pnfdId that identifies the PNFD of the constituent PNF realizing the non-virtualized part of the gNB;
 - d) virtualLinkDesc that comprises the following attribute:
 - i) virtualLinkDf that comprises the following attribute:
 - (1) qoS that comprises the latency attribute, indicating the latency requirement for the F1 interface.
 - e) cpdPoolId that references to a pool of descriptors of connection points attached to the constituent VNF(s) (realizing the virtualized part of gNB) and PNF(s) (realizing the non-virtualized part of gNB).
 - f) nfpd (optional) that specifies the network forwarding path associated to the VNFFG.

[00123] The NFVO can update (e.g., via processor(s) 610) the NSD with the added VNFFGD, and can respond to the NM about the successful NSD update (e.g., via a

response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410), with parameter nsdInfold indicating the updated NSD.

Potential Solution for Adding VNFFG(s) for the gNB

[00124] This section provides a potential solution to support the Use Case and capabilities for adding VNFFGs for the gNB.

[00125] This section can apply to scenarios wherein the NSD has been on-boarded with the vnffgd and nsDf, which comprises information relevant to latency and bandwidth of the virtual link, respectively, have been on-boarded at NFVO, and a NS based on this NSD has been instantiated.

[00126] If the NSD has been on-boarded without the associated nsDf, the NM can request the NFVO to update the NSD to include the following attributes (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), and the NFVO can respond to the NM (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) about the successful NSD update with the parameter nsdInfold indicating the updated NSD:

- 1) nsDf that comprises the following attributes:
 - a) virtualLinkProfile that comprises the following attributes:
 - i) maxBitrateRequirements indicating the maximum bandwidth requirement for the F1 interface;
 - ii) minBitrateRequirements attributes, indicating the minimum bandwidth requirement for the F1 interface;
 - b) nsInstantiationLevel to indicate the NsLevel(s) within the NS deployment flavor, and each NsLevel that comprises the following attributes:
 - i) virtualLinkToLevelMapping that contain the bitRateRequirements indicating the bandwidth requirement for the F1 interface.

[00127] NM can invoke the Update NS operation to request the NFVO to add VNFFGs connecting the virtualized part and non-virtualized part of the gNB (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed

by processor(s) 610), with the following parameters: (a) updateType = "AddVnffg"; and (b) addVnffg that comprises the vnffgdld and vnflInstanceId of the virtualized part of gNB to create the VNFFG instance in the NS.

[00128] The NFVO can send the NS lifecycle change notification to NM indicating the start of NS update procedure (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00129] The NFVO can update the NS (e.g., via processor(s) 610), based on the information provide by NM and the information provided in the updated NSD.

[00130] The NFVO can send the NS Lifecycle Change notification to NM indicating the result of the NS update (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

Additional Embodiments

[00131] Examples herein can include subject matter such as a method, means for performing acts or blocks of the method, at least one machine-readable medium including executable instructions that, when performed by a machine (e.g., a processor with memory, an application-specific integrated circuit (ASIC), a field programmable gate array (FPGA), or the like) cause the machine to perform acts of the method or of an apparatus or system for concurrent communication using multiple communication technologies according to embodiments and examples described.

[00132] A first example embodiment in accordance with the first set of embodiments described herein (e.g., related to management of the ltf-N reference point in connection with a functional split of a gNB) comprises a Network Manager (NM, e.g., employing system 400) comprising one or more processors (e.g., processor(s) 410) configured to: send a request to an Element Manager (EM, e.g., employing system 500) to establish the relation between gNB-CU (gNB Central Unit) and gNB-DU (gNB Distributed Unit) (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the ltf-N reference point, received via communication circuitry 520, and processed by processor(s) 510); and receive a response from EM about the establishment result (e.g., a response generated by processor(s) 510, sent via communication circuitry 520 over the ltf-N reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00133] A second example embodiment in accordance with the first set of embodiments can comprise the first example embodiment, wherein the request is made by creating and/or configuring (e.g., via processor(s) 510) a Management Object Instance (MOI) of an Information Object Class (IOC) modeling the end point of the reference point between gNB-CU and gNB-DU.

[00134] A third example embodiment in accordance with the first set of embodiments can comprise any of the first to the second example embodiments, wherein the Information Object Class (IOC) is EP_F1.

[00135] A fourth example embodiment in accordance with the first set of embodiments can comprise any of the first to the third example embodiments, wherein the MOI of the IOC modeling the end point is comprised within a MOI of an IOC modeling one of the gNB-CU or the gNB-DU.

[00136] A fifth example embodiment in accordance with the first set of embodiments can comprise any of the first to the fourth example embodiments, wherein the MOI of the IOC modeling the end point of the reference point between gNB-CU and gNB-DU is associated with a MOI of an IOC modeling one of the gNB-DU or the gNB-CU.

[00137] A sixth example embodiment in accordance with the first set of embodiments can comprise an EM (e.g., employing system 500) that comprises one or more processors (e.g., processor(s) 510) is configured to: receive a request from a NM (e.g., employing system 400) to establish the relation between gNB-CU and gNB-DU (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the ltf-N reference point, received via communication circuitry 520, and processed by processor(s) 510); configure (e.g., via processor(s) 510) the gNB-CU and/or gNB-DU to establish the relation; and send a response to the NM about the establishment result (e.g., a response generated by processor(s) 510, sent via communication circuitry 520 over the ltf-N reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00138] A seventh example embodiment in accordance with the first set of embodiments can comprise a Network Manager (NM) (e.g., employing system 400) comprising one or more processors (e.g., processor(s) 410) configured to: send a request to an Element Manager (EM) to establish a relation between gNB-CU (gNB Central Unit) and Core Network (CN) Network Function (NF) (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the ltf-N reference point, received via communication circuitry 520, and processed by processor(s) 510); and

receive a response from EM about the establishment result (e.g., a response generated by processor(s) 510, sent via communication circuitry 520 over the ltf-N reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00139] An eighth example embodiment in accordance with the first set of embodiments can comprise the seventh example embodiment, wherein the request is made by creating and/or configuring (e.g., via processor(s) 410) a Management Object Instance (MOI) of an Information Object Class (IOC) modeling the end point of the reference point between the gNB-CU and the CN NF.

[00140] A ninth example embodiment in accordance with the first set of embodiments can comprise any of the seventh to the eighth example embodiments, wherein the MOI of the IOC modeling the end point is comprised within a MOI of an IOC modeling one of the gNB-CU or CN NF.

[00141] A tenth example embodiment in accordance with the first set of embodiments can comprise any of the seventh to the ninth example embodiments, wherein the MOI of the IOC modeling the end point of the reference point between gNB-CU and CN NF is associated with a MOI of an IOC modeling one of the CN NF or the gNB-CU.

[00142] An eleventh example embodiment in accordance with the first set of embodiments can comprise an EM (e.g., employing system 500) comprising one or more processors (e.g., processor(s) 510) configured to: receive a request from a NM (e.g., employing system 400) to establish the relation between gNB-CU and CN NF (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the ltf-N reference point, received via communication circuitry 520, and processed by processor(s) 510); configure (e.g., via processor(s) 510) the gNB-CU and/or CN NF to establish the relation; and send a response to NM about the establishment result (e.g., a response generated by processor(s) 510, sent via communication circuitry 520 over the ltf-N reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00143] A twelfth example embodiment in accordance with the first set of embodiments can comprise any of the first to the eleventh example embodiments, wherein the IOC modeling the reference point between gNB-CU and gNB-DU, and the IOC modeling the reference point between gNB-CU and CN NF are inherited from an IOC EP_RP.

[00144] A thirteenth example embodiment in accordance with the first set of embodiments can comprise a Network Manager (NM) (e.g., employing system 400)

comprising one or more processors (e.g., processor(s) 410) configured to: send a first request to a Network Function Virtualization Orchestrator (NFVO) (e.g., employing system 600) to create the NS identifier for the Network Service Descriptor (NSD) that references to one or more of the Virtualized Network Function Descriptor (VNFD) of the CN NF, the VNFD of the virtualized part of the gNB, and the Physical Network Function Descriptor (PNFD) of the non-virtualized part of gNB, if any (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610); receive the nsInstancelId from NFVO (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); and/or send a second request to the Network Function Virtualization Orchestrator (NFVO) to instantiate the NS identified by the said nsInstancelId, with the parameter additionalParamForVnf providing the information for the CN NF and virtualized part of the gNB, and optionally with the parameter pnfnInfo providing the information of the non-virtualized part of the gNB (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610); receive the NS lifecycle change notification from NFVO indicating the start of NS instantiation procedure NFVO (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); and receive the NS Lifecycle Change notification from NFVO indicating the result of NS instantiation (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00145] A fourteenth example embodiment in accordance with the first set of embodiments can comprise a NFVO (e.g., employing system 600) comprising one or more processors (e.g., processor(s) 610) configured to: receive a first request (e.g., generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610) from a NM (e.g., employing system 400) to create the NS identifier for the Network Service Descriptor (NSD) that references to one or more of the Virtualized Network Function Descriptor (VNFD) of the CN NF, VNFD of the virtualized part of gNB,

and the Physical Network Function Descriptor (PNFD) of the non-virtualized part of gNB if any; create the NS identifier and save the NS identifier to nsInstancelId (e.g., via processor(s) 610); send the nsInstancelId to NM (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); and/or receive a second request (e.g., generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610) from the NM to instantiate the NS identified by the said nsInstancelId, with the parameter additionalParamForVnf providing the information for the CN NF and virtualized part of the gNB, and optionally with the parameter pnflInfo providing the information of the non-virtualized part of the gNB; send the NS lifecycle change notification to NM indicating the start of NS instantiation procedure (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); and send the NS Lifecycle Change notification to NM indicating the result of NS instantiation (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00146] A fifteenth example embodiment in accordance with the first set of embodiments can comprise a Network Manager (NM) (e.g., employing system 400) comprising one or more processors (e.g., processor(s) 410) configured to: invoke an Update NSD operation to a Network Function Virtualization Orchestrator (NFVO) (e.g., employing system 600) to update the NSD (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), which comprises the following information:

- 1) vnffgd (VNF Forwarding Graph Descriptor) that can comprise the following attributes:
 - a) vnffgdId (VNFFGD identifier) that uniquely identifies a VNFFGD;
 - b) vnfdId (VNFD identifier) that identifies the VNFD of the constituent VNF realizing the virtualized part of gNB;
 - c) pnfdId (PNFD identifier) that identifies the PNFD of the constituent PNF realizing the non-virtualized part of gNB;

- d) virtualLinkDesc (virtual link descriptor) that can comprise the following attribute:
 - i) virtualLinkDf (Virtual link deployment flavor) that can comprise the following attribute:
 - (1) qoS (quality of service) that can comprise the latency attribute, which can indicate the latency requirement for the F1 interface.
 - e) cpdPoolId (Connection Point Descriptor Pool identifier) that references to a pool of descriptors of connection points attached to the constituent VNF(s) (realizing the virtualized part of gNB) and PNF(s) (realizing the non-virtualized part of gNB).
 - f) nfpd (network forwarding path descriptor) (optional) that specifies the network forwarding path associated to the VNFFG.
- 2) nsDf (NS deployment flavor) that can comprise the following attributes:
- a) virtualLinkProfile (virtual link profile) that can comprise the following attributes:
 - i) maxBitrateRequirements (maximum bitrate requirements) indicating the maximum bandwidth requirement for the F1 interface;
 - ii) minBitrateRequirements (minimum bitrate requirements) attributes, indicating the minimum bandwidth requirement for the F1 interface;
 - b) nsInstantiationLevel (NS instantiation level) to indicate the NsLevel(s) (NS level(s) within the NS deployment flavor, and each NsLevel that contain the following attributes:
 - i) virtualLinkToLevelMapping (virtual link to level mapping) that can comprise the bitRateRequirements (bitrate requirements) indicating the bandwidth requirement for the F1 interface;

receive the response (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) from NFVO indicating one of that the Update NSD operation is successful with nsdInfold or that the Update NSD operation failed due to an out of range error (of the bitRateRequirements parameter); and/or invoke (e.g., via processor(s) 410) the Update NS operation to request the NFVO to associate the NS with the newly updated NSD if the Update NSD operation was successful (e.g., wherein the request can be generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the following parameters: (a) updateType = "AssocNewNsdVersion" to associate the NS instance with the updated NSD and (b) assocNewNsdVersionData to indicate the NSD with which the NS instance will be

associated; receive the NS lifecycle change notification (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) from NFVO indicating the start of NS update procedure; and receive the NS Lifecycle Change notification (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) from NFVO indicating the result of NS update.

[00147] A sixteenth example embodiment in accordance with the first set of embodiments can comprise a NFVO (e.g., employing system 600) comprising one or more processors (e.g., processor(s) 610) configured to: receive an Update NSD operation from NM (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610) to update the NSD that comprises the following information:

- 1) vnffgd (VNF Forwarding Graph Descriptor) that can comprise the following attributes:
 - a) vnffgdId (VNFFGD identifier) that uniquely identifies a VNFFGD;
 - b) vnfdId (VNFD identifier) that identifies the VNFD of the constituent VNF realizing the virtualized part of gNB;
 - c) pnfdId (PNFD identifier) that identifies the PNFD of the constituent PNF realizing the non-virtualized part of gNB;
 - d) virtualLinkDesc (virtual link descriptor) that can comprise the following attribute:
 - i) virtualLinkDf (Virtual link deployment flavor) that can comprise the following attribute:
 - (1) qoS (quality of service) that can comprise the latency attribute, which can indicate the latency requirement for the F1 interface.
 - e) cpdPoolId (Connection Point Descriptor Pool identifier) that references to a pool of descriptors of connection points attached to the constituent VNF(s) (realizing the virtualized part of gNB) and PNF(s) (realizing the non-virtualized part of gNB).
 - f) nfpd (network forwarding path descriptor) (optional) that specifies the network forwarding path associated to the VNFFG.
- 2) nsDf (NS deployment flavor) that can comprise the following attributes:
 - a) virtualLinkProfile (virtual link profile) that can comprise the following attributes:

- i) maxBitrateRequirements (maximum bitrate requirements) indicating the maximum bandwidth requirement for the F1 interface;
- ii) minBitrateRequirements (minimum bitrate requirements) attributes, indicating the minimum bandwidth requirement for the F1 interface;
- b) nsInstantiationLevel (NS instantiation level) to indicate the NsLevel(s) (NS level(s) within the NS deployment flavor, and each NsLevel that contain the following attributes:
 - i) virtualLinkToLevelMapping (virtual link to level mapping) that can comprise the bitRateRequirements (bitrate requirements) indicating the bandwidth requirement for the F1 interface;

validate (e.g., via processor(s) 610) whether the bitRateRequirements is in the range between minBitrateRequirements and maxBitrateRequirements; and return the nsdInfo indicating the successful NSD update to NM if the validation went through (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); or return an out of range error to NM indicating the bitRateRequirements is out of the range between minBitrateRequirements and maxBitrateRequirements (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); and/or receive the Update NS operation (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610) to associate the NS with the newly updated NSD, with the following parameters: (a) updateType = "AssocNewNsdVersion" to associate the NS instance with the updated NSD and (b) assocNewNsdVersionData to indicate the NSD with which the NS instance will be associated; send the NS lifecycle change notification to NM indicating the start of NS update procedure (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); and send the NS Lifecycle Change notification to NM indicating the result of NS update (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00148] A seventeenth example embodiment in accordance with the first set of embodiments can comprise a Network Manager (NM, e.g., employing system 400) comprising one or more processors (e.g., processor(s) 410) configured to: invoke the Update NSD operation to a Network Function Virtualization Orchestrator (NFVO, e.g., employing system 600) to update the NSD (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610) which comprises the following information:

- 1) vnffgd that can comprise the following attributes:
 - a) vnffgdId that uniquely identifies a VNFFGD;
 - b) vnfdId that identifies the VNFD of the constituent VNF realizing the virtualized part of gNB;
 - c) pnfdId that identifies the PNFD of the constituent PNF realizing the non-virtualized part of gNB;
 - d) virtualLinkDesc that can comprise the following attribute:
 - i) virtualLinkDf that can comprise the following attribute:
 - (1) qoS that can comprise the latency attribute, indicating the latency requirement for the F1 interface.
 - e) cpdPoolId that references to a pool of descriptors of connection points attached to the constituent VNFs (realizing the virtualized part of gNB) and PNFs (realizing the non-virtualized part of gNB);
 - f) nfpd (optional) that specifies the network forwarding path associated to the VNFFG.

receive the response of Update NSD operation from NFVO (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); and/or invoke the Update NS operation to request NFVO to update the VNFFG containing the VL connecting the gNB-CU and gNB-DU (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the following parameters: (a) updateType = "UpdateVnffg" update the VNFFG, including the VL between gNB-CU and gNB-DU for the NS instance and (b) updateVnffg to indicate the updated VNFFGD for the NS instance; receive the NS lifecycle change notification from NFVO (e.g., generated by processor(s) 610, sent

via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the start of NS update procedure; and receive the NS Lifecycle Change notification from NFVO (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the result of NS update.

[00149] An eighteenth example embodiment in accordance with the first set of embodiments can comprise a NFVO (e.g., employing system 600) comprising one or more processors (e.g., processor(s) 610) configured to: receive an Update NSD operation from NM (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610) to update the NSD that comprises the following information:

- 1) vnffgd that can comprise the following attributes:
 - a) vnffgdld that uniquely identifies a VNFFGD;
 - b) vnfdld that identifies the VNFD of the constituent VNF realizing the virtualized part of gNB;
 - c) pnfdld that identifies the PNFD of the constituent PNF realizing the non-virtualized part of gNB;
 - d) virtualLinkDesc that can comprise the following attribute:
 - i) virtualLinkDf that can comprise the following attribute:
 - (1) qoS that can comprise the latency attribute, indicating the latency requirement for the F1 interface.
 - e) cpdPoolld that references to a pool of descriptors of connection points attached to the constituent VNFs (realizing the virtualized part of gNB) and PNFs (realizing the non-virtualized part of gNB);
 - f) nfpd (optional) that specifies the network forwarding path associated to the VNFFG.

update the NSD (e.g., via processor(s) 610); return the nsdInfold (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the successful NSD update; and/or receive the Update NS operation to update the VNFFG containing the VL connecting the gNB-CU and gNB-DU (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420

over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the following parameters: (a) updateType = "UpdateVnffg" update the VNFFG, including the VL between gNB-CU and gNB-DU for the NS instance and (b) updateVnffg to indicate the updated VNFFGD for the NS instance; send the NS lifecycle change notification to NM (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the start of NS update procedure; and send the NS Lifecycle Change notification to NM (e.g., via a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the result of NS update.

[00150] A nineteenth example embodiment in accordance with the first set of embodiments can comprise a Network Manager (NM, e.g., employing system 400) comprising one or more processors (e.g., processor(s) 410) configured to: invoke the Update NSD operation to Network Function Virtualization Orchestrator (NFVO, e.g., employing system 600) to update the NSD (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), which comprises the following information:

- 1) nsDf that can comprise the following attributes:
 - a) virtualLinkProfile that can comprise the following attributes:
 - i) maxBitrateRequirements, indicating the maximum bandwidth requirement for the F1 interface;
 - ii) minBitrateRequirements, indicating the minimum bandwidth requirement for the F1 interface;
 - b) nsInstantiationLevel that can indicate the NsLevel(s) within the NS deployment flavor, wherein each NsLevel can comprise the following attributes:
 - i) virtualLinkToLevelMapping that can comprise the bitRateRequirements indicating the bandwidth requirement for the F1 interface.

receive the response from NFVO (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the Update NSD operation is successful with nsdInfold or indicating the Update NSD operation is

failed due to out of range error (of the bitRateRequirements parameter); and/or invoke the Update NS operation to request NFVO to change the NS deployment flavor if the Update NSD operation was successful (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the following parameters: (a) updateType = "ChangeNsDf" to change the deployment flavor for the NS instance and (b) receive the NS lifecycle change notification from NFVO indicating the start of NS update procedure; and receive the NS Lifecycle Change notification (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) from NFVO indicating the result of NS update.

[00151] A twentieth example embodiment in accordance with the first set of embodiments can comprise a NFVO (e.g., employing system 600) comprising one or more processors (e.g., processor(s) 610) configured to: receive an Update NSD operation from NM to update the NSD (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610) that comprises the following information:

- 1) nsDf that can comprise the following attributes:
 - a) virtualLinkProfile that can comprise the following attributes:
 - i) maxBitrateRequirements, indicating the maximum bandwidth requirement for the F1 interface;
 - ii) minBitrateRequirements, indicating the minimum bandwidth requirement for the F1 interface;
 - b) nsInstantiationLevel that can indicate the NsLevel(s) within the NS deployment flavor, wherein each NsLevel can comprise the following attributes:
 - i) virtualLinkToLevelMapping that can comprise the bitRateRequirements indicating the bandwidth requirement for the F1 interface.

validate (e.g., via processor(s) 610) whether the bitRateRequirements is in the range between minBitrateRequirements and maxBitrateRequirements; and return the nsdInfold indicating the successful NSD update to NM if the validation went through (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); or return an out of range error to NM indicating the

bitRateRequirements is out of the range between minBitrateRequirements and maxBitrateRequirements (e.g., via a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); and/or receive the Update NS operation to update the bandwidth requirements for the F1 interface (e.g., via a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), with the following parameter: updateType = "ChangeNsDf" to change the deployment flavor for the NS instance; send the NS lifecycle change notification (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) to NM indicating the start of NS update procedure; and send the NS Lifecycle Change notification (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) to NM indicating the result of NS update.

[00152] A first example embodiment in accordance with the second set of embodiments can comprise a Network Function Virtualization Orchestrator (NFVO, e.g., employing system 600) comprising one or more processors (e.g., processor(s) 610) configured to: receive a request from NM (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610) to on-board a NSD; on-board the NSD (e.g., via processor(s) 610) if the attributes in the NSD are valid (e.g., as determined by processor(s) 610); and send a response to NM indicating the result (e.g., success or failure) of the NSD on-board operation (e.g., a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00153] A second example embodiment in accordance with the second set of embodiments can comprise a NM (e.g., employing system 400) comprising one or more processors (e.g., processor(s) 410) configured to: send a request to NFVO to on-board a NSD (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610); and receive a response to NM indicating the result

(e.g., success or failure) of the NSD on-board operation (e.g., a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00154] A third example embodiment in accordance with the first set of embodiments can comprise any of the first to second example embodiments, wherein the NSD to be on-boarded can comprise one or more of the following attributes: (a) a virtual link descriptor that comprises the latency requirement of the interface between gNB-CU and gNB-DU; (b) a VNF forwarding graph descriptor that comprises the latency requirement of the interface between gNB-CU and gNB-DU; and/or (c) a NS deployment flavor that comprises the bandwidth requirement and the range defined by the maximum bandwidth and minimum bandwidth values of the interface between gNB-CU and gNB-DU.

[00155] A fourth example embodiment in accordance with the first set of embodiments can comprise any of the first to third example embodiments, wherein the NFVO sends a response (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) with a NSD identifier to indicate the success of NSD on-boarding, and on-boards the NSD, if the bandwidth requirement is within the range.

[00156] A fifth example embodiment in accordance with the first set of embodiments can comprise any of the first to third example embodiments, wherein the NFVO sends a response (e.g., generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) to indicate NSD on-boarding has failed with an error of out of range if the bandwidth requirement is out of the range.

[00157] A sixth example embodiment in accordance with the first set of embodiments can comprise any of the first to second example embodiments, wherein NFVO on-boards NSD that only contains virtual link descriptor and/or VNF forwarding graph descriptor, and sends a response with NSD identifier to indicate the success of NSD on-boarding.

[00158] A seventh example embodiment in accordance with the first set of embodiments can comprise a NFVO (e.g., employing system 600) comprising one or more processors (e.g., processor(s) 610) configured to: receive a request from a NM to update a NSD (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry

620, and processed by processor(s) 610); update the NSD (e.g., via processor(s) 610) if the attributes in the NSD are valid (e.g., as determined by processor(s) 610); send a response to the NM (e.g., a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the result (e.g., success or failure) of the NSD update operation; receive a request from NM to update a NS (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), based on the NSD on-boarded; send a notification to the NM (e.g., a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) to indicate the start of NS update operation; update the NS according to the NSD (e.g., via processor(s) 610); and send a notification to NM to indicate the result of NS update operation (e.g., a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00159] An eighth example embodiment in accordance with the first set of embodiments can comprise a NM (e.g., employing system 400) comprising one or more processors (e.g., processor(s) 410) configured to: send a request to a NFVO to update a NSD (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610); receive a response from the NFVO (e.g., a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) indicating the result (e.g., success or failure) of the NSD update operation; send a request to the NFVO to update a NS (e.g., a request generated by processor(s) 410, sent via communication circuitry 420 over the Os-Ma-Nfvo reference point, received via communication circuitry 620, and processed by processor(s) 610), based on the NSD on-boarded; receive a notification from NFVO indicating the start of NS update operation (e.g., a notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410); and receive a notification from NFVO to indicating the result of NS update operation (e.g., a

notification generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410).

[00160] A ninth example embodiment in accordance with the first set of embodiments can comprise any of the seventh to eighth example embodiments, wherein the NSD to be updated can comprise one or more the following attributes: (a) a virtual link descriptor that comprises the latency requirement of the interface between gNB-CU and gNB-DU; (b) a VNF forwarding graph descriptor that comprises the latency requirement of the interface between gNB-CU and gNB-DU; and/or (c) a NS deployment flavor that comprises the bandwidth requirement and the range defined by the maximum bandwidth and minimum bandwidth values of the interface between gNB-CU and gNB-DU.

[00161] A tenth example embodiment in accordance with the first set of embodiments can comprise any of the seventh to ninth example embodiments, wherein the NFVO can send a response (e.g., a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) with a NSD identifier to indicate the success of the NSD update, and can update the NSD (e.g., via processor(s) 610), if the bandwidth requirement is within the range.

[00162] An eleventh example embodiment in accordance with the first set of embodiments can comprise any of the seventh to ninth example embodiments, wherein the NFVO can send a response (e.g., a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) to indicate the NSD update has failed with an error of out of range if the bandwidth requirement is out of the range.

[00163] A twelfth example embodiment in accordance with the first set of embodiments can comprise any of the seventh to ninth example embodiments, wherein if the NSD to be updated does not contain a bandwidth requirement, the NFVO can update (e.g., via processor(s) 610) the virtual link descriptor and/or the VNF forwarding graph descriptor, and can send a response (e.g., a response generated by processor(s) 610, sent via communication circuitry 620 over the Os-Ma-nfvo reference point, received via communication circuitry 420, and processed by processor(s) 410) with a NSD identifier to indicate the success of NSD update.

[00164] A thirteenth example embodiment in accordance with the first set of embodiments can comprise any of the seventh to eighth example embodiments, wherein the NSD update can comprise adding (e.g., via processor(s) 610) a VNF forwarding graph descriptor for connecting the gNB-CU VNF and gNB-DU.

[00165] A fourteenth example embodiment in accordance with the first set of embodiments can comprise any of the seventh to eighth example embodiments, wherein the NS update can comprise adding (e.g., via processor(s) 610) a VNF forwarding graph to the NS to connect the gNB-CU VNF and gNB-DU.

[00166] A fifteenth example embodiment in accordance with the first set of embodiments can comprise any of the seventh, eighth, or fourteenth example embodiments, wherein the NS update request comprises a vnffgdId that identifies the VNFFGD to be used to create the VNFFG instance, and a vnfInstanceId that identifies one or more VNF instance(s) to be contained to the VNFFG instance.

[00167] A sixteenth example embodiment in accordance with the first set of embodiments can comprise any of the first or thirteenth example embodiments, wherein the bandwidth requirement is defined by bitRateRequirements attribute for a NS level within the NS deployment flavor.

[00168] A seventeenth example embodiment in accordance with the first set of embodiments can comprise any of the first or thirteenth example embodiments, wherein the bandwidth requirement range is defined by maxBitrateRequirements and minBitrateRequirements attributes for the NS deployment flavor.

[00169] Example 1 is an apparatus configured to be employed within a NM (Network Manager), comprising: a memory interface; and processing circuitry configured to: invoke an update NSD (NS (Network Service) Descriptor) operation to request a NFVO (NFV (Network Function Virtualization) Orchestrator) to update a NSD via a first request comprising a nsdInfoId (NSD information identifier) parameter that indicates the NSD to be updated and a NSD parameter that comprises a vnffgd (VNFFG (VNF (Virtual Network Function) FG (Forwarding Graph)) Descriptor) attribute to be updated and a nsDf (NS Deployment Flavor) attribute to be updated; receive a response from the NFVO indicating the NSD was successfully updated, wherein the response comprises the nsdInfoId parameter; invoke a update NS operation to request the NFVO to associate a NS with the NSD via a second request comprising an updateType (update type) parameter equal to AssocNewNsdVersion (associate new NSD version) to associate the NS with the NSD and a assocNewNsdVersionData (associate new NSD

version data) parameter to indicate the NSD to associate with the NS; receive a first NS lifecycle change notification from the NFVO that indicates a start of a NS update procedure; receive a second NS lifecycle change notification from the NFVO that indicates a result of the NS update procedure; and send the nsdInfold parameter to a memory via the memory interface.

[00170] Example 2 comprises the subject matter of any variation of any of example(s) 1, wherein the vnffgd parameter comprises: a vnffgdId (VNFFGD Identifier) attribute that uniquely identifies a VNFFGD, a vnfdId (VNFD (VNF Descriptor) Identifier) attribute that identifies a VNFD of a constituent VNF for a virtualized part of a gNB (next generation NodeB), a pnfdId (PNFD (PNF (Physical Network Function) Descriptor) Identifier) attribute that identifies a PNFD of a constituent PNF for a non-virtualized part of the gNB, and a cpdPoolId (Connection Point Descriptor Pool Identifier) attribute that references to a pool of descriptors of connection points to one or more constituent VNFs realizing the virtualized part of the gNB and one or more PNFs realizing the non-virtualized part of the gNB.

[00171] Example 3 comprises the subject matter of any variation of any of example(s) 2, wherein the vnffgd parameter further comprises a nfpd (Network Forwarding Path Descriptor) attribute that specifies a network forwarding path associated with a VNFFGD of the NS.

[00172] Example 4 comprises the subject matter of any variation of any of example(s) 1-3, wherein the first request comprises a virtualLinkDf (Virtual Link Deployment Flavor) attribute that comprises a qos (Quality of Service) attribute, wherein the qos attribute comprises a latency attribute that indicates a latency requirement for a F1 interface.

[00173] Example 5 comprises the subject matter of any variation of any of example(s) 1-3, wherein the nsDf attribute comprises a virtualLinkProfile (Virtual Link Profile) attribute and a nsInstantiationLevel (NS Instantiation Level) attribute that indicates one or more NsLevel (NS Level) attributes within a deployment flavor of the NS.

[00174] Example 6 comprises the subject matter of any variation of any of example(s) 5, wherein the virtualLinkProfile attribute comprises a maxBitrateRequirements (Maximum Bitrate Requirements) attribute and a minBitrateRequirements (Minimum Bitrate Requirements) attribute, the maxBitrateRequirements attribute indicating a maximum bandwidth requirement for an interface between a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB, and the minBitrateRequirements attribute indicating a minimum bandwidth requirement for an

interface between the virtualized part of the gNB and the non-virtualized part of the gNB.

[00175] Example 7 comprises the subject matter of any variation of any of example(s) 5, wherein each NsLevel attribute of the one or more NsLevel attributes comprises a virtualLinkToLevelMapping (Virtual Link to Level Mapping) attribute that comprises a bitRateRequirements attribute that indicates a bandwidth requirement for an interface between a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB.

[00176] Example 8 is an apparatus configured to be employed within a NFVO (Network Function Virtualization Orchestrator), comprising: a memory interface; and processing circuitry configured to: receive a first request from a NM (Network Manager) to update a NSD, wherein the first request comprises a nsdInfolld (NSD information identifier) parameter that indicates the NSD to be updated and a NSD parameter that comprises a vnffgd (VNFFGD (VNF (Virtual Network Function) FG (Forwarding Graph) Descriptor)) attribute to be updated and a nsDf (NS Deployment Flavor) attribute to be updated; send a response to the NM indicating the NSD was successfully updated, wherein the response comprises the nsdInfolld parameter; receive a second request from the NM to associate a NS with the NSD, wherein the second request comprises an updateType (update type) parameter equal to AssocNewNsdVersion (associate new NSD version) to associate the NS with the NSD and a assocNewNsdVersionData (associate new NSD version data) parameter to indicate the NSD to associate with the NS; send a first NS lifecycle change notification to the NM that indicates a start of a NS update procedure; update the NS based on information provided by the NM and information provided in the NSD; send a second NS lifecycle change notification to the NM that indicates a result of the NS update procedure; and send the nsdInfolld parameter to a memory via the memory interface.

[00177] Example 9 comprises the subject matter of any variation of any of example(s) 8, wherein the vnffgd parameter comprises: a vnffgdld (VNFFGD Identifier) attribute that uniquely identifies a VNFFGD, a vnfdld (VNFD (VNF Descriptor) Identifier) attribute that identifies a VNFD of a constituent VNF for a virtualized part of a gNB (next generation NodeB), a pnfdld (PNFD (PNF (Physical Network Function) Descriptor) Identifier) attribute that identifies a PNFD of a constituent PNF for a non-virtualized part of the gNB, and a cpdPoolld (Connection Point Descriptor Pool Identifier) attribute that references to a pool of descriptors of connection points to one or more constituent VNFs

realizing the virtualized part of the gNB and one or more PNFs realizing the non-virtualized part of the gNB.

[00178] Example 10 comprises the subject matter of any variation of any of example(s) 9, wherein the vnffgd parameter further comprises a nfpd (Network Forwarding Path Descriptor) attribute that specifies a network forwarding path associated with a VNFFG of the NS.

[00179] Example 11 comprises the subject matter of any variation of any of example(s) 8-10, wherein the first request comprises a virtualLinkDf (Virtual Link Deployment Flavor) attribute that comprises a qoS (Quality of Service) attribute, wherein the qoS attribute comprises a latency attribute that indicates a latency requirement for a F1 interface.

[00180] Example 12 comprises the subject matter of any variation of any of example(s) 8-10, wherein the nsDf attribute comprises a virtualLinkProfile (Virtual Link Profile) attribute and a nsInstantiationLevel (NS Instantiation Level) attribute that indicates one or more NsLevel (NS Level) attributes within a deployment flavor of the NS.

[00181] Example 13 comprises the subject matter of any variation of any of example(s) 12, wherein the virtualLinkProfile attribute comprises a maxBitrateRequirements (Maximum Bitrate Requirements) attribute and a minBitrateRequirements (Minimum Bitrate Requirements) attribute, the maxBitrateRequirements attribute indicating a maximum bandwidth requirement for an interface between a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB, and the minBitrateRequirements attribute indicating a minimum bandwidth requirement for an interface between the virtualized part of the gNB and the non-virtualized part of the gNB.

[00182] Example 14 comprises the subject matter of any variation of any of example(s) 12, wherein each NsLevel attribute of the one or more NsLevel attributes comprises a virtualLinkToLevelMapping (Virtual Link to Level Mapping) attribute that comprises a bitRateRequirements attribute that indicates a bandwidth requirement for an interface between a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB.

[00183] Example 15 is an apparatus configured to be employed within a NM (Network Manager), comprising: a memory interface; and processing circuitry configured to: send a request to a NFVO (NFV (Network Function Virtualization) Orchestrator) to invoke an

update NSD (NS (Network Service) Descriptor) operation to add a VNFFGD (VNF (Virtualized Network Function) FG (Forwarding Graph) Descriptor) for a gNB (next generation NodeB), wherein the request comprises a nsdInfold parameter that indicates a NSD to be updated and a NSD parameter, wherein the NSD parameter indicates the NSD and comprises the VNFFGD to be added; receive a response from the NFVO that indicates the NSD was successfully updated, wherein the response comprises the nsdInfold parameter; and send the nsdInfold parameter to a memory via the memory interface.

[00184] Example 16 comprises the subject matter of any variation of any of example(s) 15, wherein the NSD parameter comprises a vnffgd (VNFFGD) attribute that comprises: a vnffgdId (VNFFGD Identifier) that uniquely identifies the VNFFGD, a vnfdId (VNFD (VNF Descriptor) Identifier) attribute that identifies a VNFD of a constituent VNF realizing a virtualized part of a gNB, a pnfdId (PNFD (PNF (Physical Network Function) Descriptor) Identifier) attribute that identifies a PNFD realizing a non-virtualized part of the gNB, and a cpdPoolId (Connection Point Descriptor Pool Identifier) attribute that references to a pool of descriptors of connection points to one or more constituent VNFs realizing the virtualized part of the gNB and one or more PNFs realizing the non-virtualized part of the gNB.

[00185] Example 17 comprises the subject matter of any variation of any of example(s) 16, wherein the vnffgd parameter further comprises a nfpd (Network Forwarding Path Descriptor) attribute that specifies a network forwarding path associated with a VNFFG of the NS.

[00186] Example 18 comprises the subject matter of any variation of any of example(s) 15-17, wherein the first request comprises a virtualLinkDf (Virtual Link Deployment Flavor) attribute that comprises a qoS (Quality of Service) attribute, wherein the qoS attribute comprises a latency attribute that indicates a latency requirement for a F1 interface.

[00187] Example 19 comprises the subject matter of any variation of any of example(s) 15-17, wherein the NSD has been on-boarded.

[00188] Example 20 is an apparatus configured to be employed within a NFVO (Network Function Virtualization Orchestrator), comprising: a memory interface; and processing circuitry configured to: receive a request from a NM (Network Manager) to invoke an update NSD (NS (Network Service) Descriptor) operation to add a VNFFGD (VNF (Virtualized Network Function) FG (Forwarding Graph) Descriptor) for a gNB (next

generation NodeB), wherein the request comprises a nsdInfold parameter that indicates a NSD to be updated and a NSD parameter, wherein the NSD parameter indicates the NSD and comprises the VNFFGD to be added; update the NSD by adding the VNFFGD for the gNB; send a response to the NM that indicates the NSD was successfully updated, wherein the response comprises the nsdInfold parameter; and send the nsdInfold parameter to a memory via the memory interface.

[00189] Example 21 comprises the subject matter of any variation of any of example(s) 20, wherein the NSD parameter comprises a vnffgd (VNFFGD) attribute that comprises: a vnffgdId (VNFFGD Identifier) that uniquely identifies the VNFFGD, a vnfdId (VNFD (VNF Descriptor) Identifier) attribute that identifies a VNFD of a constituent VNF realizing a virtualized part of a gNB, a pnfdId (PNFD (PNF (Physical Network Function) Descriptor) Identifier) attribute that identifies a PNFD realizing a non-virtualized part of the gNB, and a cpdPoolId (Connection Point Descriptor Pool Identifier) attribute that references to a pool of descriptors of connection points to one or more constituent VNFs realizing the virtualized part of the gNB and one or more PNFs realizing the non-virtualized part of the gNB.

[00190] Example 22 comprises the subject matter of any variation of any of example(s) 21, wherein the vnffgd parameter further comprises a nfpd (Network Forwarding Path Descriptor) attribute that specifies a network forwarding path associated with a VNFFG of the NS.

[00191] Example 23 comprises the subject matter of any variation of any of example(s) 20-22, wherein the first request comprises a virtualLinkDf (Virtual Link Deployment Flavor) attribute that comprises a qoS (Quality of Service) attribute, wherein the qoS attribute comprises a latency attribute that indicates a latency requirement for a F1 interface.

[00192] Example 24 comprises the subject matter of any variation of any of example(s) 20-22, wherein the NSD has been on-boarded.

[00193] Example 25 is an apparatus configured to be employed within a NM (Network Manager), comprising: a memory interface; and processing circuitry configured to: invoke an update NS (Network Service) operation for a NS to send a request to a NFVO (NFV (Network Function Virtualization) Orchestrator) to add one or more VNFFGs (VNF (Virtual Network Function) FGs (Forwarding Graphs)) connecting a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB, wherein the request comprises an updateType (update type) parameter equal to AddVnffg (Add

VNFFG) and an addVnffg parameter, wherein the addVnffg parameter comprises a vnffgld (VNFFG Identifier) parameter of the virtualized part of the gNB and a vnflInstanceId (VNF instance identifier) parameter of the virtualized part of the gNB; receive a first NS lifecycle change notification from the NFVO that indicates a start of an update procedure for the NS; receive a second NS lifecycle change notification from the NFVO that indicates a result of the update procedure for the NS; and send the vnffgld parameter and the vnflInstanceId parameter to a memory via the memory interface.

[00194] Example 26 comprises the subject matter of any variation of any of example(s) 25, wherein the NS has been instantiated.

[00195] Example 27 comprises the subject matter of any variation of any of example(s) 25-26, wherein a NSD (NS Descriptor) of the NS has been updated with a vnffgd (VNFFG Descriptor) parameter and a nsDf (NS Deployment Flavor) parameter, wherein the vnffgd parameter comprises information relevant to a latency of a virtual link between the virtualized part of the gNB and the non-virtualized part of the gNB, and wherein the nsDf parameter comprises information relevant to a bandwidth of the virtual link.

[00196] Example 28 is an apparatus configured to be employed within a NFVO (Network Function Virtualization Orchestrator), comprising: a memory interface; and processing circuitry configured to: receive a request from a NM (Network Manager) to add to a NS (Network Service) one or more VNFFGs (VNF (Virtual Network Function) FGs (Forwarding Graphs)) connecting a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB, wherein the request comprises an updateType (update type) parameter equal to AddVnffg (Add VNFFG) and an addVnffg parameter, wherein the addVnffg parameter comprises a vnffgld (VNFFG Identifier) parameter of the virtualized part of the gNB and a vnflInstanceId (VNF instance identifier) parameter of the virtualized part of the gNB; send a first NS lifecycle change notification to the NM that indicates a start of an update procedure for the NS; update the NS based on information provided by the NM and information provided in an updated NSD; send a second NS lifecycle change notification to the NM that indicates a result of the update procedure for the NS; and send the vnffgld parameter and the vnflInstanceId parameter to a memory via the memory interface.

[00197] Example 29 comprises the subject matter of any variation of any of example(s) 28, wherein the NS has been instantiated.

[00198] Example 30 comprises the subject matter of any variation of any of example(s) 28-29, wherein a NSD (NS Descriptor) of the NS has been updated with a vnffgd (VNFFG Descriptor) parameter and a nsDf (NS Deployment Flavor) parameter, wherein the vnffgd parameter comprises information relevant to a latency of a virtual link between the virtualized part of the gNB and the non-virtualized part of the gNB, and wherein the nsDf parameter comprises information relevant to a bandwidth of the virtual link.

[00199] Example 31 is an apparatus configured to be employed within a NM (Network Manager), comprising: a memory interface; and processing circuitry configured to: send a first request to a NFVO (Network Function Virtualization Orchestrator) to create a NS (Network Service) identifier for a NSD (NS Descriptor) that references a VNFD (Virtual Network Function (VNF) Descriptor) of a CN (Core Network) NF (Network Function), a VNFD of a virtualized part of a gNB (next generation NodeB), and a PNFD (Physical Network Function (PNF) Descriptor) of a non-virtualized part of the gNB; receive a response from the NFVO indicating successful creation of the NS identifier, wherein the response comprises a nsInstanceld (NS instance identifier) parameter; send a second request to the NFVO to instantiate the NS identified by the nsInstanceld parameter, wherein the second request comprises an additionalParamForVnf (additional parameter for VNF) parameter that provides information for the CN NF and the virtualized part of the gNB, and wherein the second request comprises a pnfInfo (PNF information) parameter that provides information for the non-virtualized part of the gNB; receive a first NS lifecycle change notification from the NFVO that indicates the start of a NS instantiation procedure; receive a second NS lifecycle change notification from the NFVO that indicates a result of the NS instantiation procedure; and send the nsInstanceld parameter to a memory via the memory interface.

[00200] Example 32 comprises the subject matter of any variation of any of example(s) 31, wherein the second request further comprises one or more other parameters.

[00201] Example 33 is an apparatus configured to be employed within a NFVO (Network Function Virtualization Orchestrator), comprising: a memory interface; and processing circuitry configured to: receive a first request from a NM (Network Manager) to create a NS (Network Service) identifier for a NSD (NS Descriptor) that references a VNFD (Virtual Network Function (VNF) Descriptor) of a CN (Core Network) NF (Network Function), a VNFD of a virtualized part of a gNB (next generation NodeB), and a PNFD

(Physical Network Function (PNF) Descriptor) of a non-virtualized part of the gNB; send a response to the NM indicating successful creation of the NS identifier, wherein the response comprises a nsInstanceld (NS instance identifier) parameter; receive a second request from the NM to instantiate the NS identified by the nsInstanceld parameter, wherein the second request comprises an additionalParamForVnf (additional parameter for VNF) parameter that provides information for the CN NF and the virtualized part of the gNB, and wherein the second request comprises a pnfInfo (PNF information) parameter that provides information for the non-virtualized part of the gNB; send a first NS lifecycle change notification from the NFVO that indicates the start of a NS instantiation procedure; instantiate the NS comprising the CN NF, the virtualized part of the gNB, and the non-virtualized part of the gNB, based on information provided by the NM and information provided in the NSD, a VNF package, and the PNFD; send a second NS lifecycle change notification from the NFVO that indicates a result of the NS instantiation procedure; and send the nsInstanceld parameter to a memory via the memory interface.

[00202] Example 34 comprises the subject matter of any variation of any of example(s) 33, wherein the second request further comprises one or more other parameters.

[00203] Example 35 comprises an apparatus comprising means for executing any of the described operations of any other example(s) described herein.

[00204] Example 36 comprises a machine readable medium that stores instructions for execution by a processor to perform any of the described operations of any other example(s) described herein.

[00205] Example 37 comprises an apparatus comprising: a memory interface; and processing circuitry configured to: perform any of the described operations of any other example(s) described herein.

[00206] The above description of illustrated embodiments of the subject disclosure, including what is described in the Abstract, is not intended to be exhaustive or to limit the disclosed embodiments to the precise forms disclosed. While specific embodiments and examples are described herein for illustrative purposes, various modifications are possible that are considered within the scope of such embodiments and examples, as those skilled in the relevant art can recognize.

[00207] In this regard, while the disclosed subject matter has been described in connection with various embodiments and corresponding Figures, where applicable, it is

to be understood that other similar embodiments can be used or modifications and additions can be made to the described embodiments for performing the same, similar, alternative, or substitute function of the disclosed subject matter without deviating therefrom. Therefore, the disclosed subject matter should not be limited to any single embodiment described herein, but rather should be construed in breadth and scope in accordance with the appended claims below.

[00208] In particular regard to the various functions performed by the above described components or structures (assemblies, devices, circuits, systems, etc.), the terms (including a reference to a “means”) used to describe such components are intended to correspond, unless otherwise indicated, to any component or structure which performs the specified function of the described component (e.g., that is functionally equivalent), even though not structurally equivalent to the disclosed structure which performs the function in the herein illustrated exemplary implementations. In addition, while a particular feature may have been disclosed with respect to only one of several implementations, such feature may be combined with one or more other features of the other implementations as may be desired and advantageous for any given or particular application.

CLAIMS

What is claimed is:

1. An apparatus configured to be employed within a NM (Network Manager), comprising:

a memory interface; and

processing circuitry configured to:

invoke an update NSD (NS (Network Service) Descriptor) operation to request a NFVO (NFV (Network Function Virtualization) Orchestrator) to update a NSD via a first request comprising a nsdInfold (NSD information identifier) parameter that indicates the NSD to be updated and a NSD parameter that comprises a vnffgd (VNFFG (VNF (Virtual Network Function) FG (Forwarding Graph)) Descriptor) attribute to be updated and a nsDf (NS Deployment Flavor) attribute to be updated;

receive a response from the NFVO indicating the NSD was successfully updated, wherein the response comprises the nsdInfold parameter;

invoke a update NS operation to request the NFVO to associate a NS with the NSD via a second request comprising an updateType (update type) parameter equal to AssocNewNsdVersion (associate new NSD version) to associate the NS with the NSD and a assocNewNsdVersionData (associate new NSD version data) parameter to indicate the NSD to associate with the NS;

receive a first NS lifecycle change notification from the NFVO that indicates a start of a NS update procedure;

receive a second NS lifecycle change notification from the NFVO that indicates a result of the NS update procedure; and

send the nsdInfold parameter to a memory via the memory interface.

2. The apparatus of claim 1, wherein the vnffgd parameter comprises: a vnffgdId (VNFFGD Identifier) attribute that uniquely identifies a VNFFGD, a vnfdId (VNFD (VNF Descriptor) Identifier) attribute that identifies a VNFD of a constituent VNF for a virtualized part of a gNB (next generation NodeB), a pnfdId (PNFD (PNF (Physical Network Function) Descriptor) Identifier) attribute that identifies a PNFD of a constituent

PNF for a non-virtualized part of the gNB, and a cpdPoolId (Connection Point Descriptor Pool Identifier) attribute that references to a pool of descriptors of connection points to one or more constituent VNFs realizing the virtualized part of the gNB and one or more PNFs realizing the non-virtualized part of the gNB.

3. The apparatus of claim 2, wherein the vnffgd parameter further comprises a nfpd (Network Forwarding Path Descriptor) attribute that specifies a network forwarding path associated with a VNFFG of the NS.

4. The apparatus of any of claims 1-3, wherein the first request comprises a virtualLinkDf (Virtual Link Deployment Flavor) attribute that comprises a qoS (Quality of Service) attribute, wherein the qoS attribute comprises a latency attribute that indicates a latency requirement for a F1 interface.

5. The apparatus of any of claims 1-3, wherein the nsDf attribute comprises a virtualLinkProfile (Virtual Link Profile) attribute and a nsInstantiationLevel (NS Instantiation Level) attribute that indicates one or more NsLevel (NS Level) attributes within a deployment flavor of the NS.

6. The apparatus of claim 5, wherein the virtualLinkProfile attribute comprises a maxBitrateRequirements (Maximum Bitrate Requirements) attribute and a minBitrateRequirements (Minimum Bitrate Requirements) attribute, the maxBitrateRequirements attribute indicating a maximum bandwidth requirement for an interface between a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB, and the minBitrateRequirements attribute indicating a minimum bandwidth requirement for an interface between the virtualized part of the gNB and the non-virtualized part of the gNB.

7. The apparatus of claim 5, wherein each NsLevel attribute of the one or more NsLevel attributes comprises a virtualLinkToLevelMapping (Virtual Link to Level Mapping) attribute that comprises a bitRateRequirements attribute that indicates a bandwidth requirement for an interface between a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB.

8. An apparatus configured to be employed within a NFVO (Network Function Virtualization Orchestrator), comprising:

a memory interface; and

processing circuitry configured to:

receive a first request from a NM (Network Manager) to update a NSD, wherein the first request comprises a nsdInfold (NSD information identifier) parameter that indicates the NSD to be updated and a NSD parameter that comprises a vnffgd (VNFFGD (VNF (Virtual Network Function) FG (Forwarding Graph) Descriptor)) attribute to be updated and a nsDf (NS Deployment Flavor) attribute to be updated;

send a response to the NM indicating the NSD was successfully updated, wherein the response comprises the nsdInfold parameter;

receive a second request from the NM to associate a NS with the NSD, wherein the second request comprises an updateType (update type) parameter equal to AssocNewNsdVersion (associate new NSD version) to associate the NS with the NSD and a assocNewNsdVersionData (associate new NSD version data) parameter to indicate the NSD to associate with the NS;

send a first NS lifecycle change notification to the NM that indicates a start of a NS update procedure;

update the NS based on information provided by the NM and information provided in the NSD;

send a second NS lifecycle change notification to the NM that indicates a result of the NS update procedure; and

send the nsdInfold parameter to a memory via the memory interface.

9. The apparatus of claim 8, wherein the vnffgd parameter comprises: a vnffgdId (VNFFGD Identifier) attribute that uniquely identifies a VNFFGD, a vnfdId (VNFD (VNF Descriptor) Identifier) attribute that identifies a VNFD of a constituent VNF for a virtualized part of a gNB (next generation NodeB), a pnfdId (PNFD (PNF (Physical Network Function) Descriptor) Identifier) attribute that identifies a PNFD of a constituent PNF for a non-virtualized part of the gNB, and a cpdPoolId (Connection Point Descriptor Pool Identifier) attribute that references to a pool of descriptors of connection points to

one or more constituent VNFs realizing the virtualized part of the gNB and one or more PNFs realizing the non-virtualized part of the gNB.

10. The apparatus of claim 9, wherein the vnffgd parameter further comprises a nfpd (Network Forwarding Path Descriptor) attribute that specifies a network forwarding path associated with a VNFFG of the NS.

11. The apparatus of any of claims 8-10, wherein the first request comprises a virtualLinkDf (Virtual Link Deployment Flavor) attribute that comprises a qoS (Quality of Service) attribute, wherein the qoS attribute comprises a latency attribute that indicates a latency requirement for a F1 interface.

12. The apparatus of any of claims 8-10, wherein the nsDf attribute comprises a virtualLinkProfile (Virtual Link Profile) attribute and a nsInstantiationLevel (NS Instantiation Level) attribute that indicates one or more NsLevel (NS Level) attributes within a deployment flavor of the NS.

13. The apparatus of claim 12, wherein the virtualLinkProfile attribute comprises a maxBitrateRequirements (Maximum Bitrate Requirements) attribute and a minBitrateRequirements (Minimum Bitrate Requirements) attribute, the maxBitrateRequirements attribute indicating a maximum bandwidth requirement for an interface between a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB, and the minBitrateRequirements attribute indicating a minimum bandwidth requirement for an interface between the virtualized part of the gNB and the non-virtualized part of the gNB.

14. The apparatus of claim 12, wherein each NsLevel attribute of the one or more NsLevel attributes comprises a virtualLinkToLevelMapping (Virtual Link to Level Mapping) attribute that comprises a bitRateRequirements attribute that indicates a bandwidth requirement for an interface between a virtualized part of a gNB (next generation NodeB) and a non-virtualized part of the gNB.

15. An apparatus configured to be employed within a NM (Network Manager), comprising:

a memory interface; and

processing circuitry configured to:

send a request to a NFVO (NFV (Network Function Virtualization) Orchestrator) to invoke an update NSD (NS (Network Service) Descriptor) operation to add a VNFFGD (VNF (Virtualized Network Function) FG (Forwarding Graph) Descriptor) for a gNB (next generation NodeB), wherein the request comprises a nsdInfold parameter that indicates a NSD to be updated and a NSD parameter, wherein the NSD parameter indicates the NSD and comprises the VNFFGD to be added;

receive a response from the NFVO that indicates the NSD was successfully updated, wherein the response comprises the nsdInfold parameter; and

send the nsdInfold parameter to a memory via the memory interface.

16. The apparatus of claim 15, wherein the NSD parameter comprises a vnffgd (VNFFGD) attribute that comprises: a vnffgdId (VNFFGD Identifier) that uniquely identifies the VNFFGD, a vnfdId (VNFD (VNF Descriptor) Identifier) attribute that identifies a VNFD of a constituent VNF realizing a virtualized part of a gNB, a pnfdId (PNFD (PNF (Physical Network Function) Descriptor) Identifier) attribute that identifies a PNFD realizing a non-virtualized part of the gNB, and a cpdPoolId (Connection Point Descriptor Pool Identifier) attribute that references to a pool of descriptors of connection points to one or more constituent VNFs realizing the virtualized part of the gNB and one or more PNFs realizing the non-virtualized part of the gNB.

17. The apparatus of claim 16, wherein the vnffgd parameter further comprises a nfpd (Network Forwarding Path Descriptor) attribute that specifies a network forwarding path associated with a VNFFG of the NS.

18. The apparatus of any of claims 15-17, wherein the first request comprises a virtualLinkDf (Virtual Link Deployment Flavor) attribute that comprises a qoS (Quality of Service) attribute, wherein the qoS attribute comprises a latency attribute that indicates a latency requirement for a F1 interface.

19. The apparatus of any of claims 15-17, wherein the NSD has been on-boarded.

20. An apparatus configured to be employed within a NFVO (Network Function Virtualization Orchestrator), comprising:
- a memory interface; and
 - processing circuitry configured to:
 - receive a request from a NM (Network Manager) to invoke an update NSD (NS (Network Service) Descriptor) operation to add a VNFFGD (VNF (Virtualized Network Function) FG (Forwarding Graph) Descriptor) for a gNB (next generation NodeB), wherein the request comprises a nsdInfold parameter that indicates a NSD to be updated and a NSD parameter, wherein the NSD parameter indicates the NSD and comprises the VNFFGD to be added;
 - update the NSD by adding the VNFFGD for the gNB;
 - send a response to the NM that indicates the NSD was successfully updated, wherein the response comprises the nsdInfold parameter; and
 - send the nsdInfold parameter to a memory via the memory interface.
21. The apparatus of claim 20, wherein the NSD parameter comprises a vnffgd (VNFFGD) attribute that comprises: a vnffgdId (VNFFGD Identifier) that uniquely identifies the VNFFGD, a vnfdId (VNFD (VNF Descriptor) Identifier) attribute that identifies a VNFD of a constituent VNF realizing a virtualized part of a gNB, a pnfdId (PNFD (PNF (Physical Network Function) Descriptor) Identifier) attribute that identifies a PNFD realizing a non-virtualized part of the gNB, and a cpdPoolId (Connection Point Descriptor Pool Identifier) attribute that references to a pool of descriptors of connection points to one or more constituent VNFs realizing the virtualized part of the gNB and one or more PNFs realizing the non-virtualized part of the gNB.
22. The apparatus of claim 21, wherein the vnffgd parameter further comprises a nfpd (Network Forwarding Path Descriptor) attribute that specifies a network forwarding path associated with a VNFFG of the NS.
23. The apparatus of any of claims 20-22, wherein the first request comprises a virtualLinkDf (Virtual Link Deployment Flavor) attribute that comprises a qoS (Quality of

Service) attribute, wherein the qoS attribute comprises a latency attribute that indicates a latency requirement for a F1 interface.

24. The apparatus of any of claims 20-22, wherein the NSD has been on-boarded.

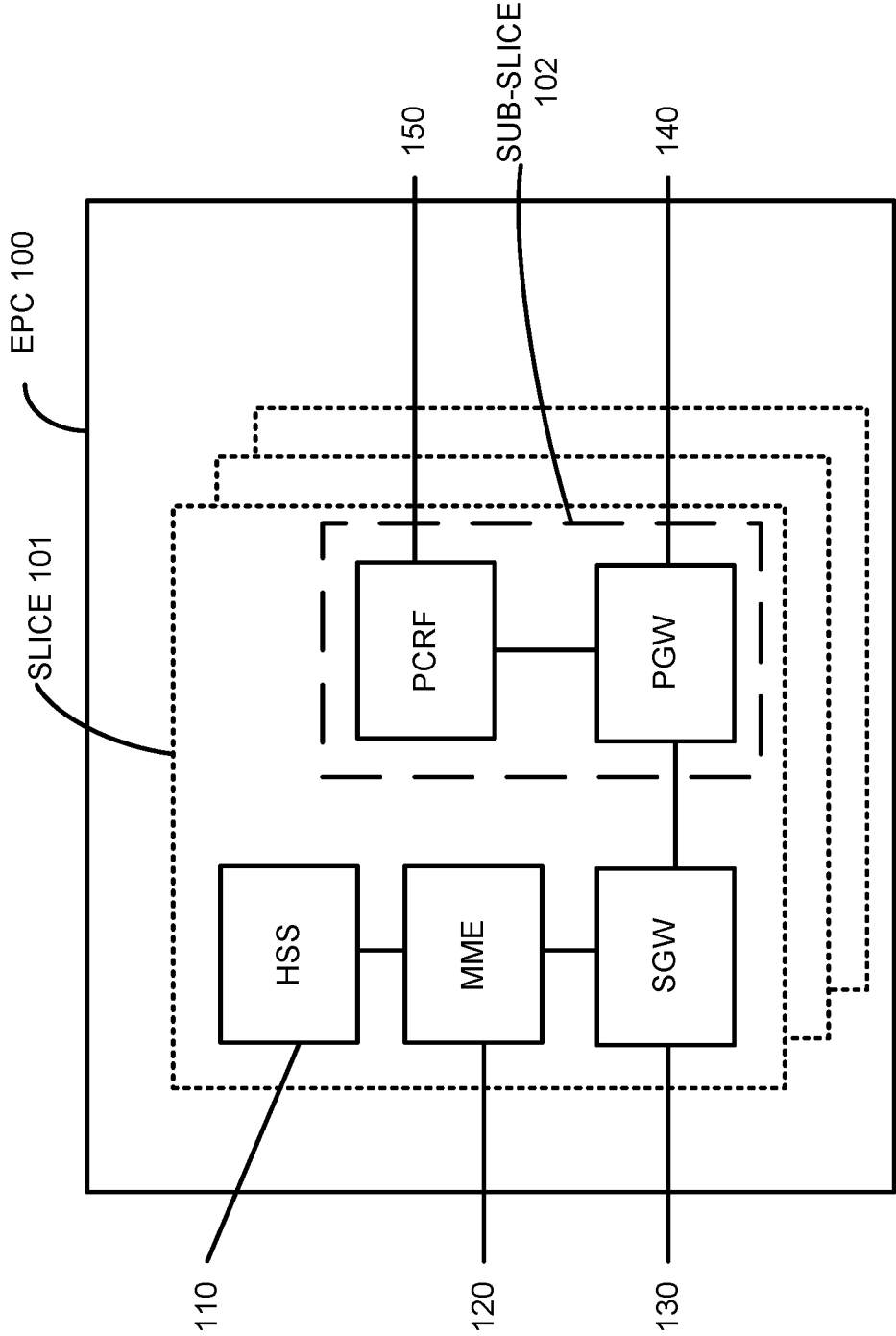


FIG. 1

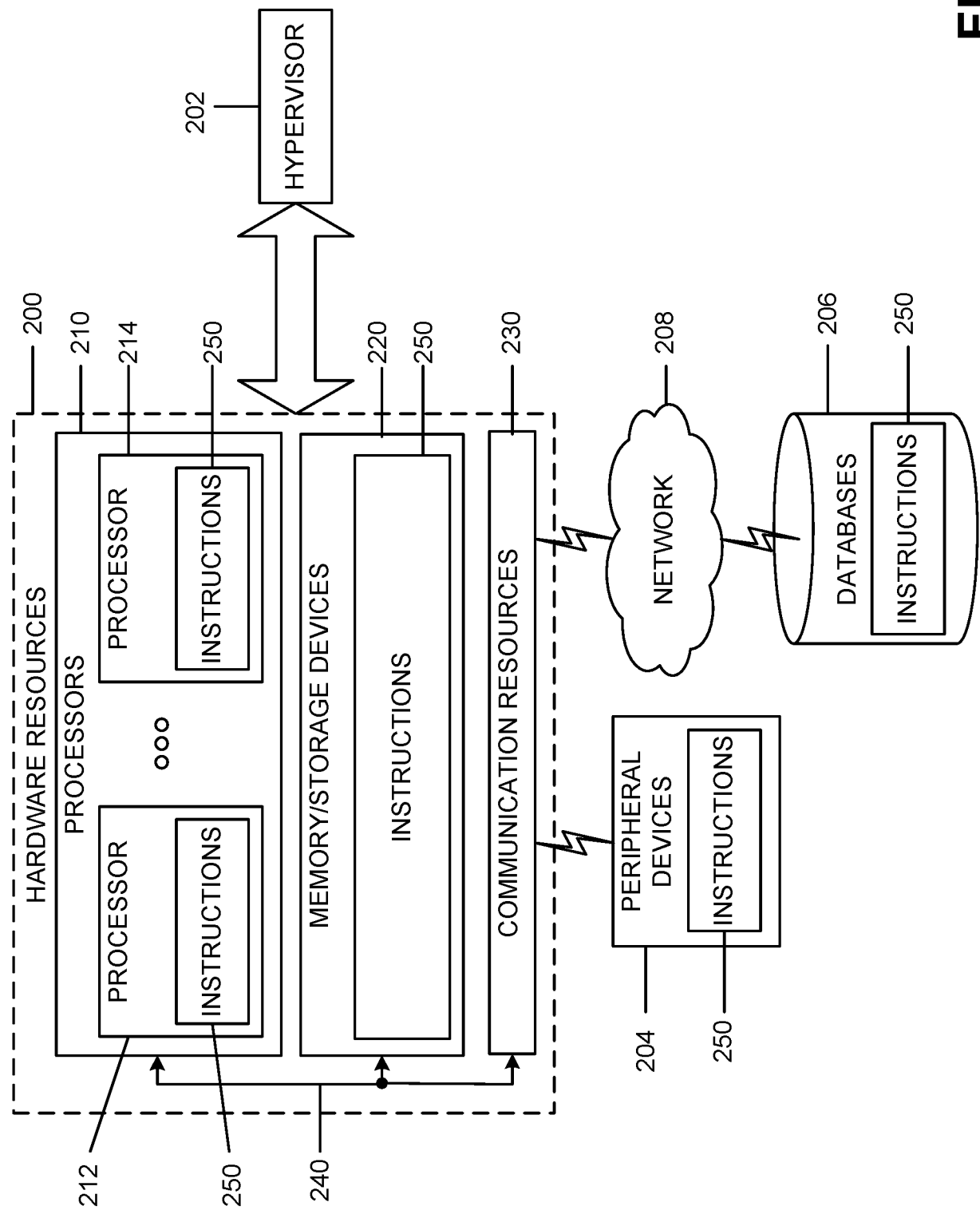


FIG. 2

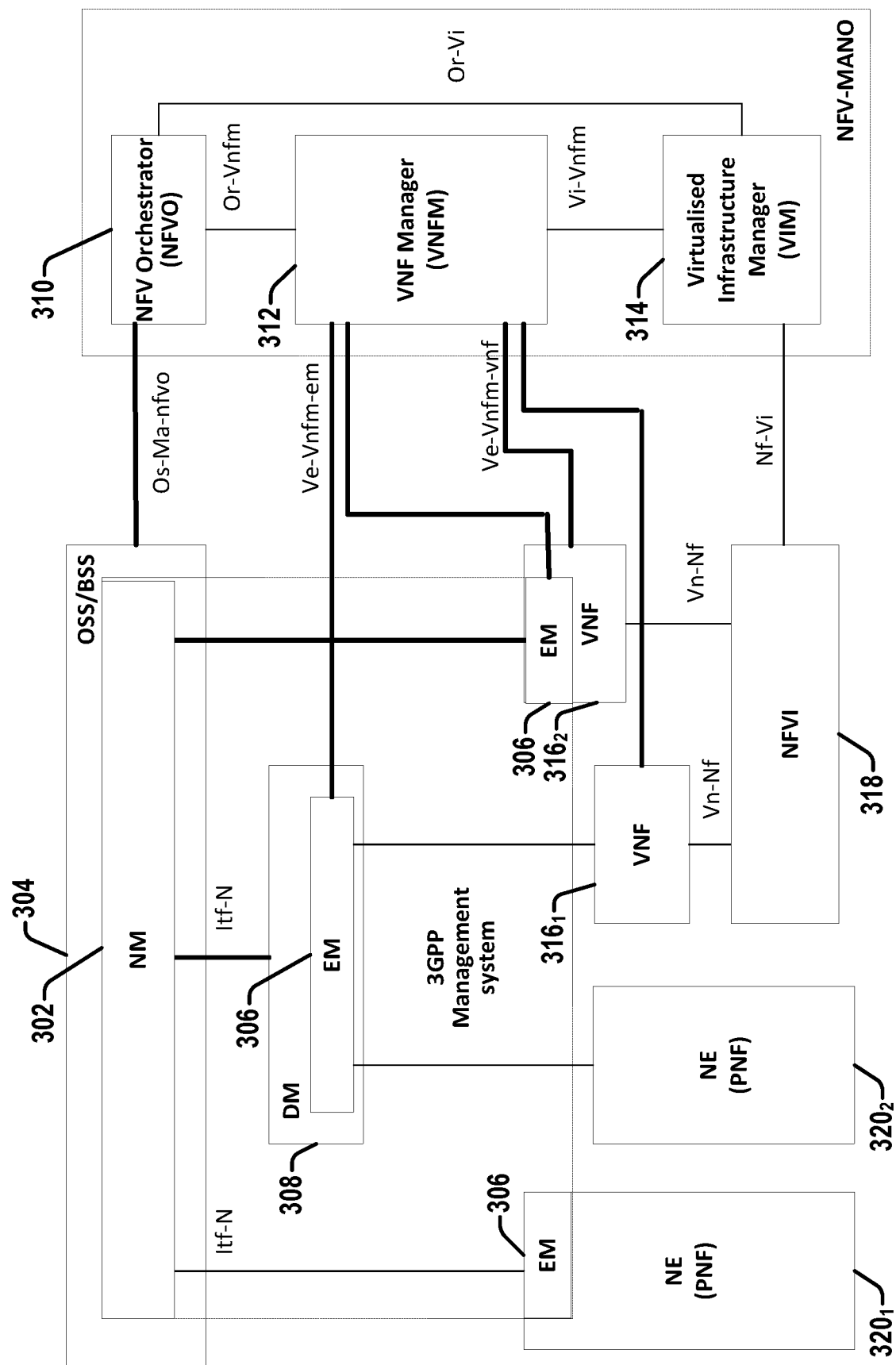


FIG. 3

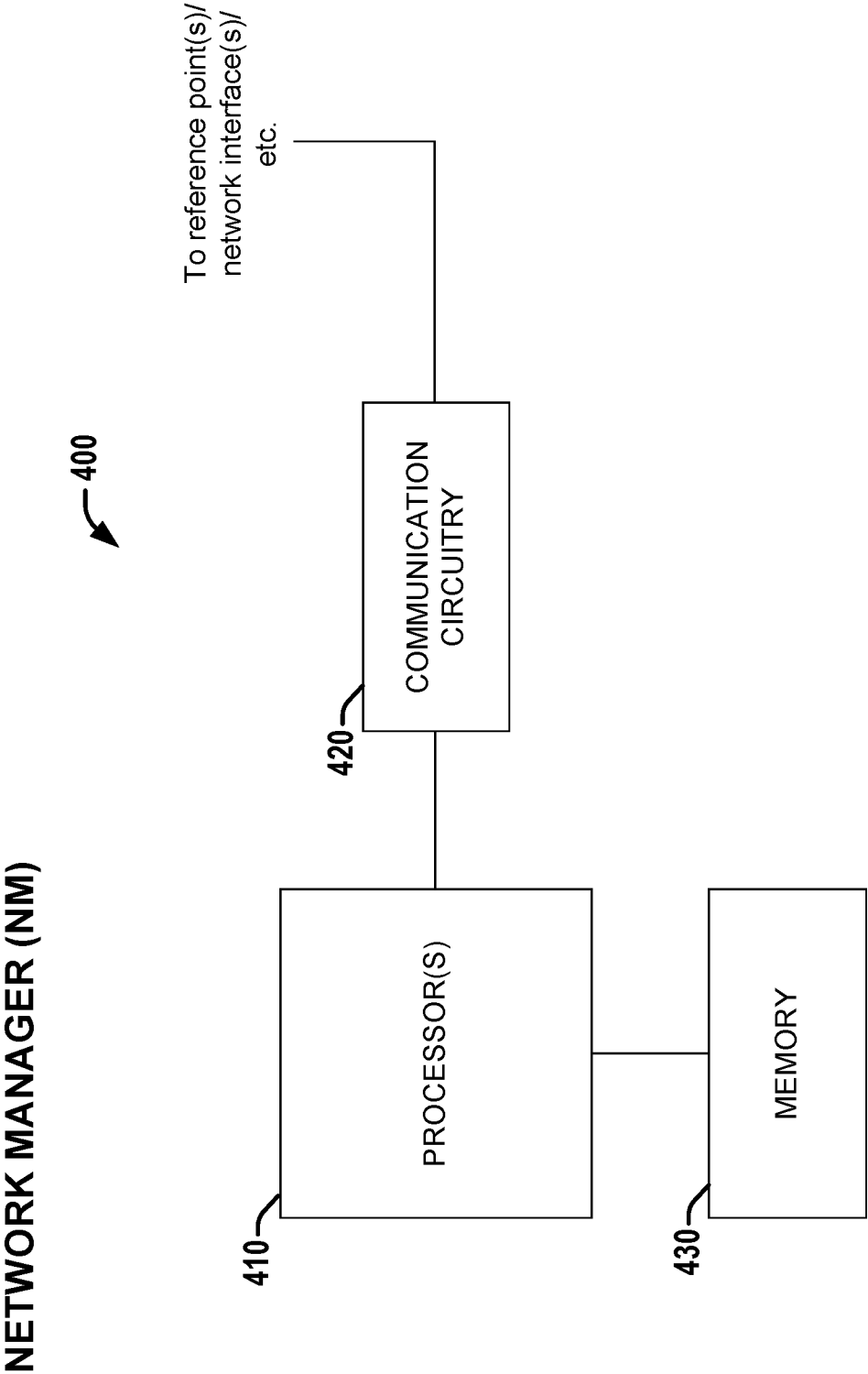


FIG. 4

ELEMENT MANAGER (EM)

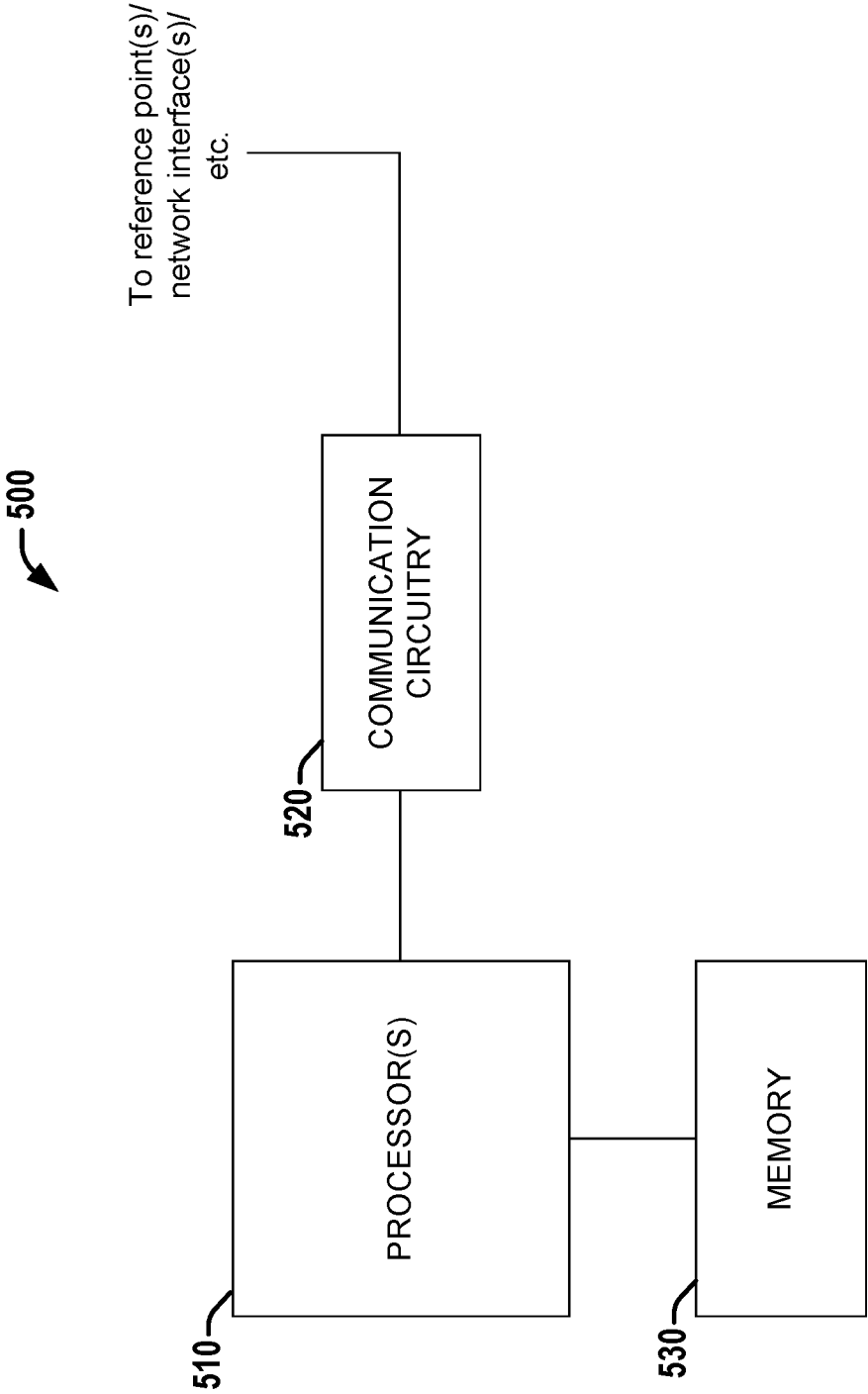


FIG. 5

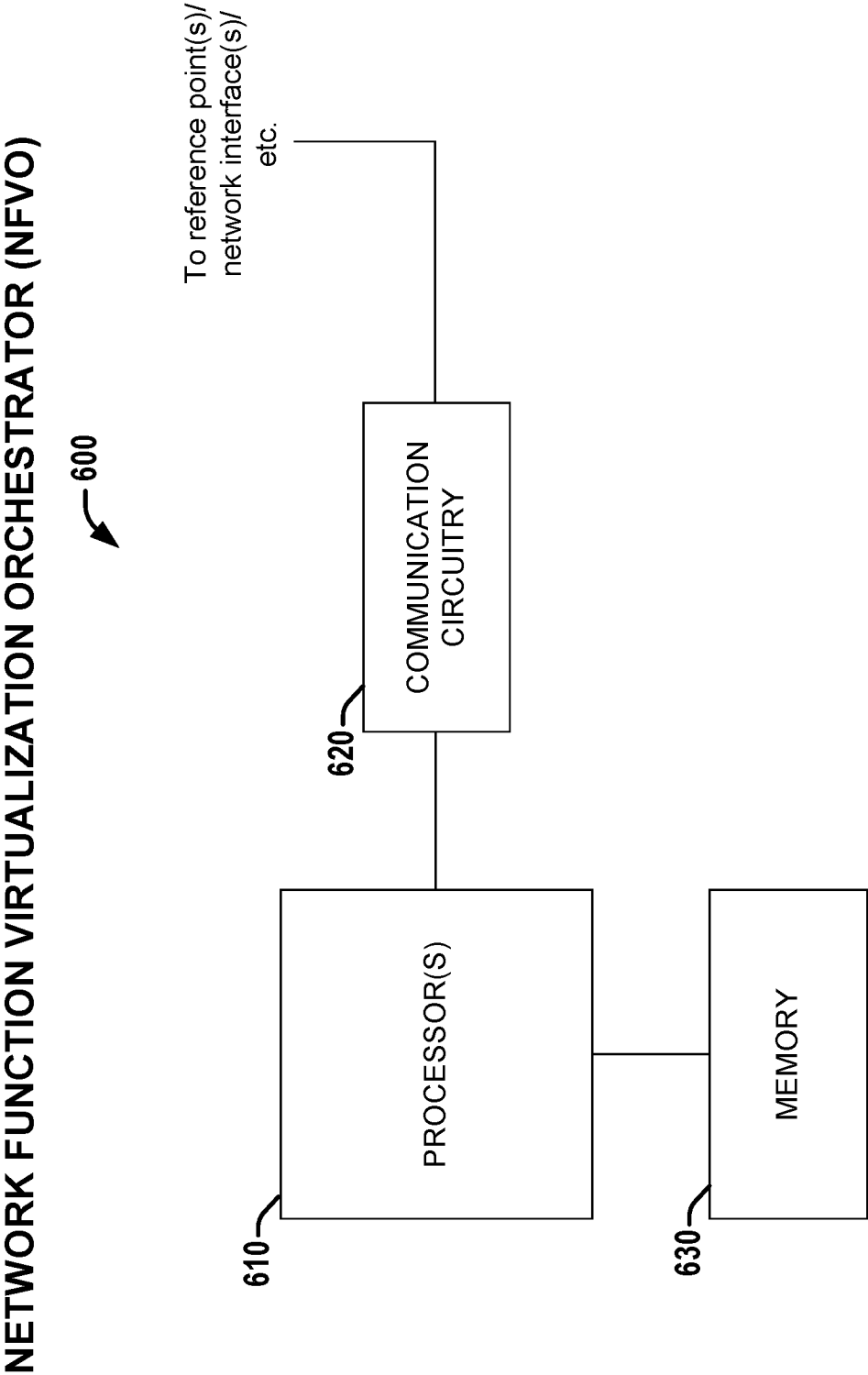


FIG. 6

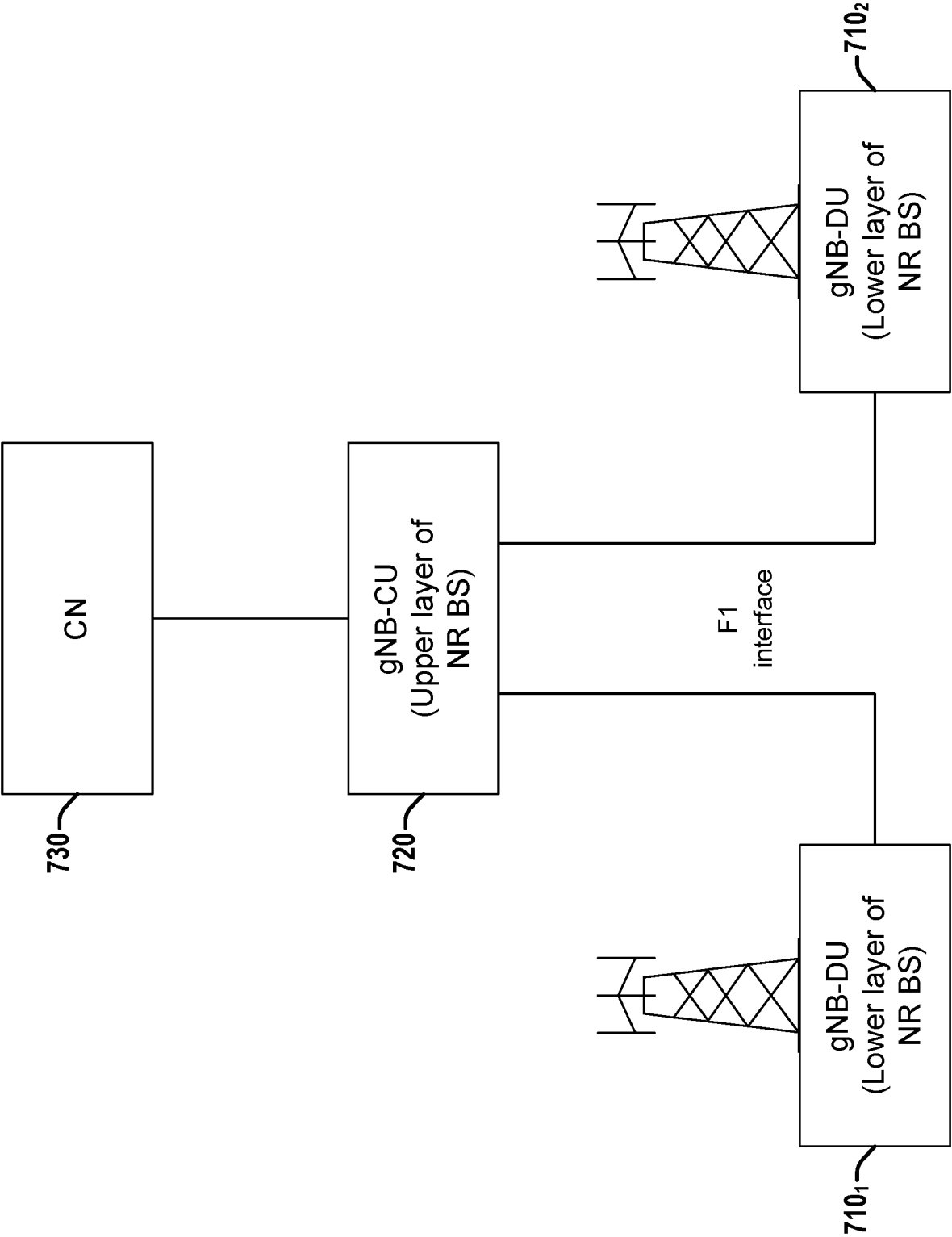


FIG. 7

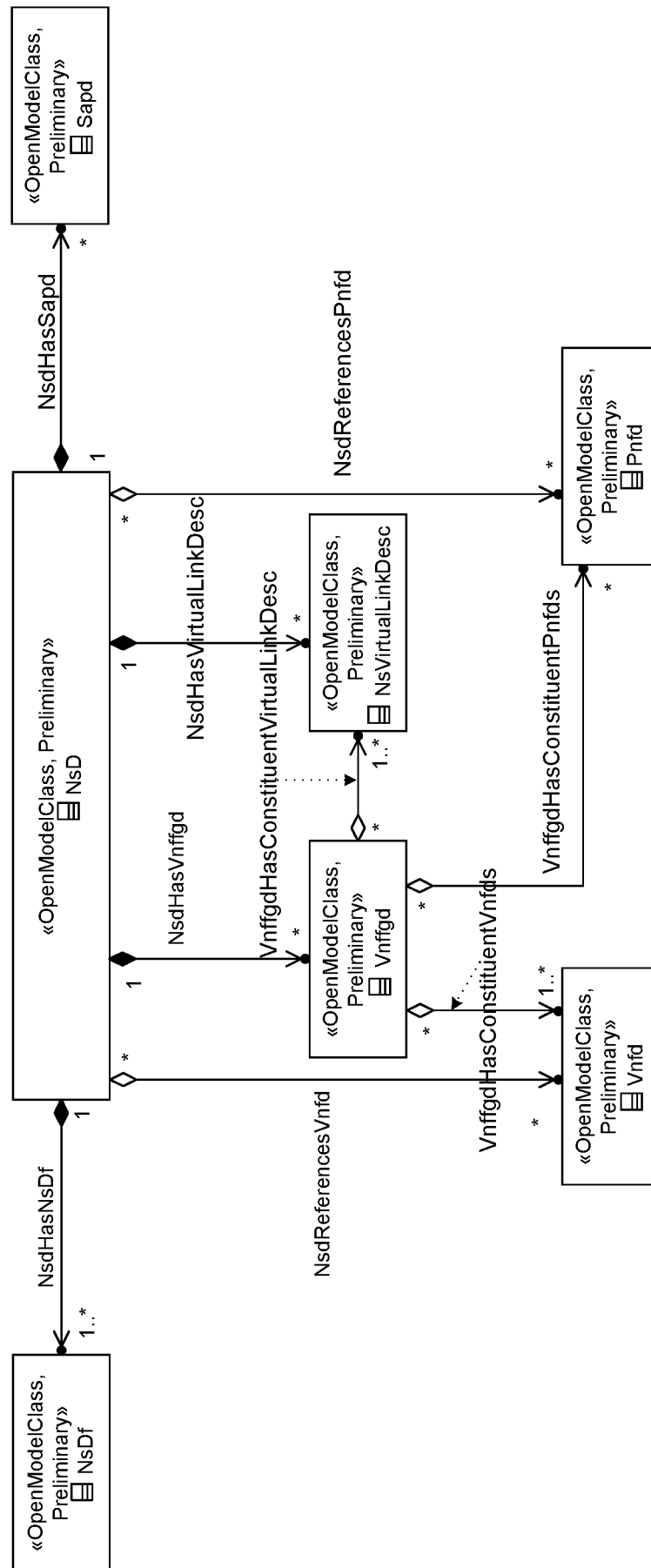


FIG. 8

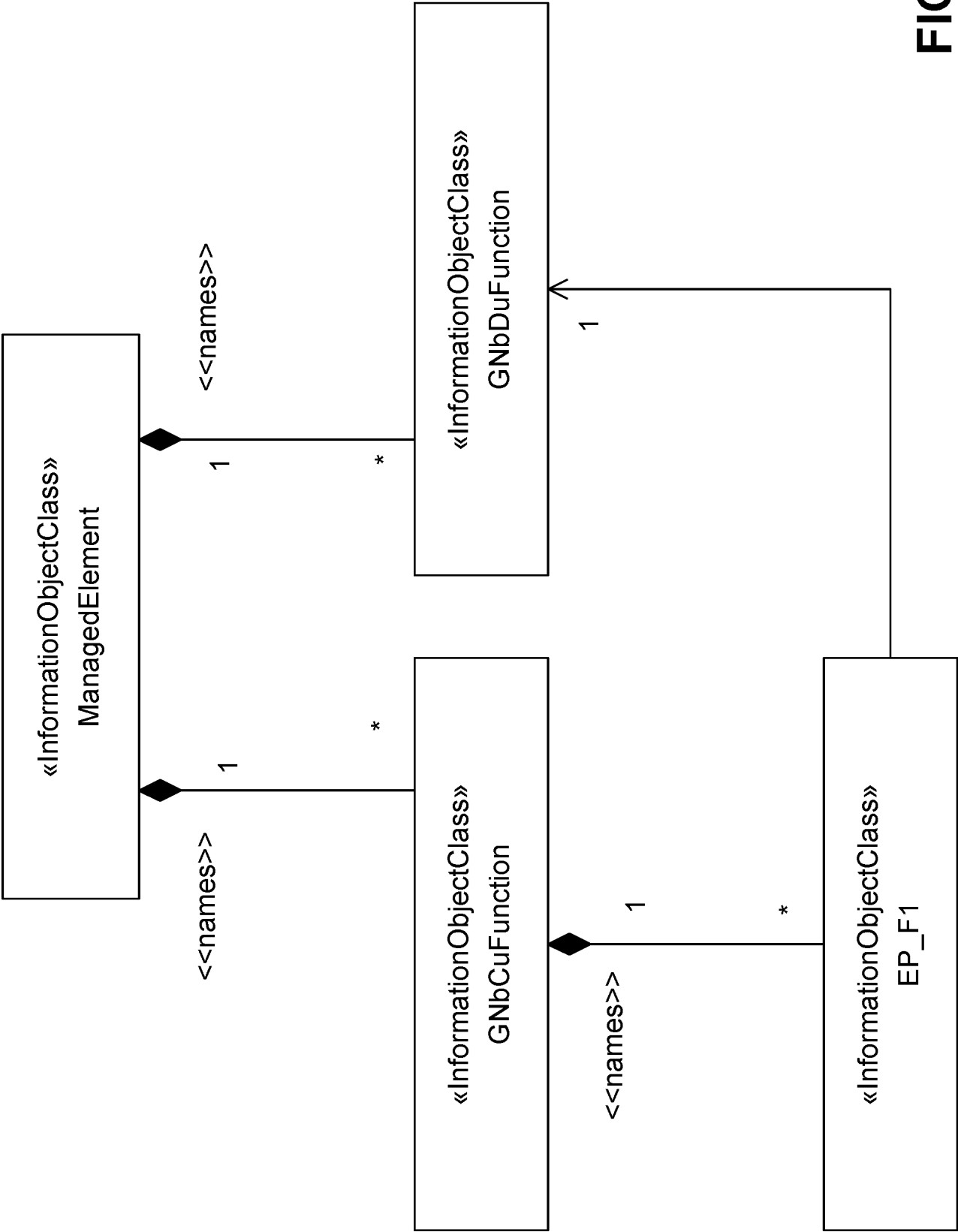


FIG. 9

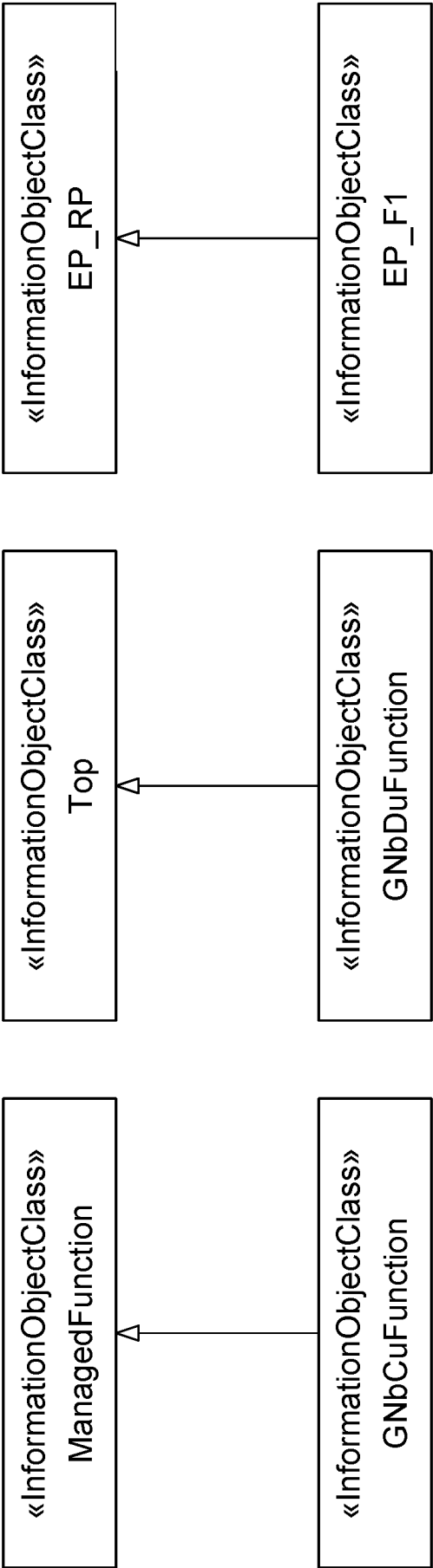


FIG. 10

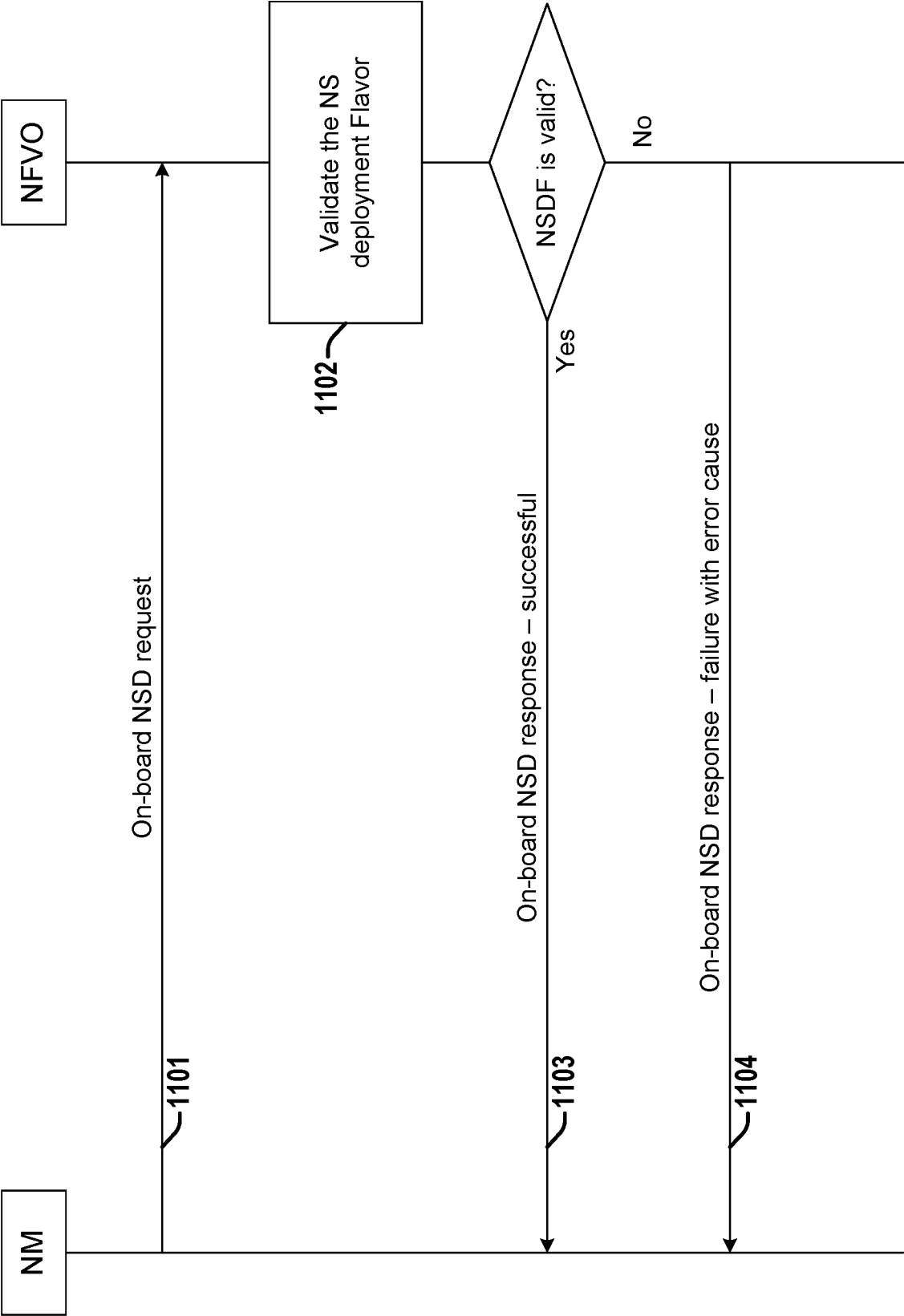


FIG. 11

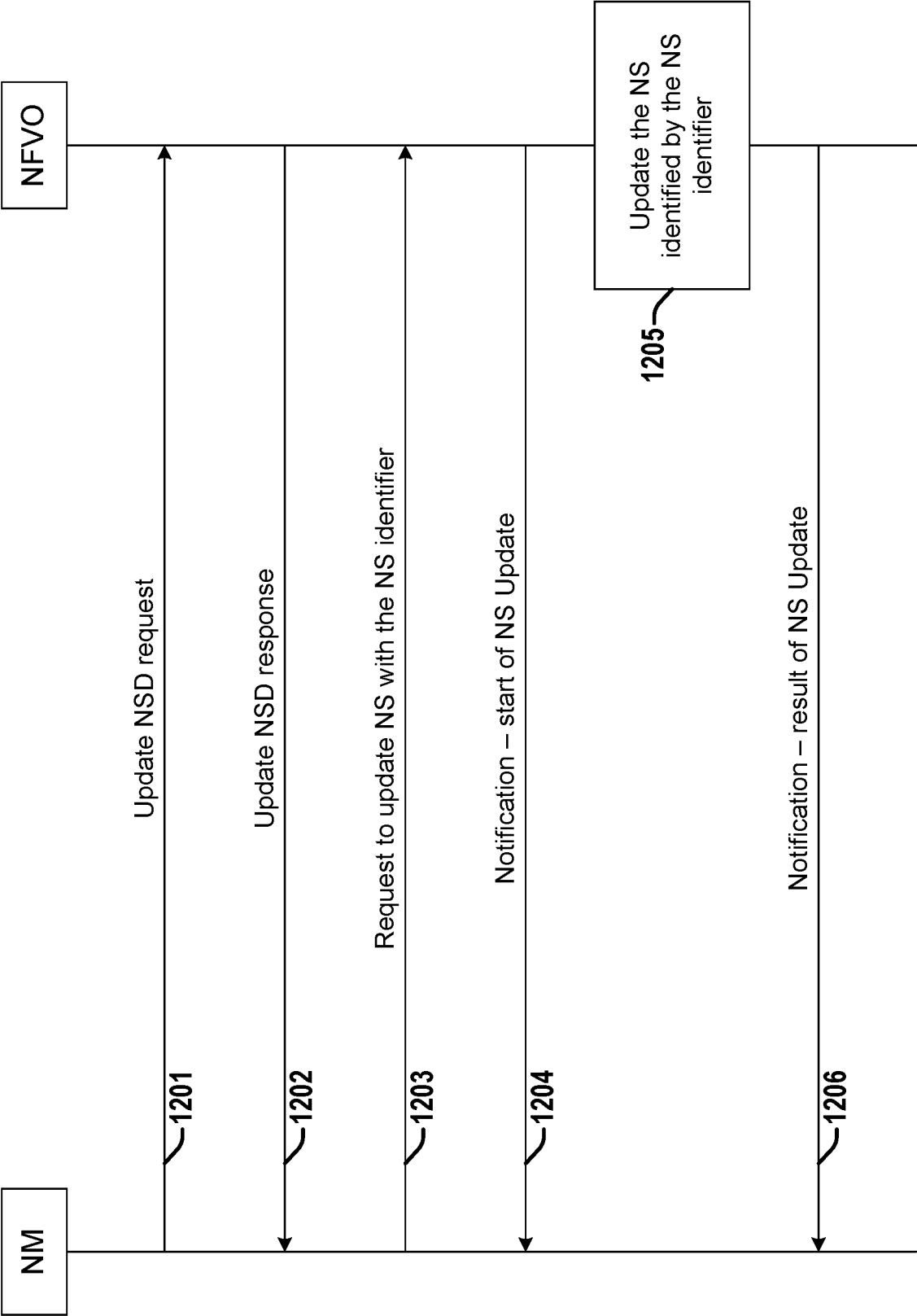


FIG. 12

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/044063

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/24 H04W52/02
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L H04W

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	"3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Life Cycle Management (LCM) for mobile networks that include virtualized network functions; Procedures (Release 14)", 3GPP STANDARD ; TECHNICAL SPECIFICATION ; 3GPP TS 28.526, 3RD GENERATION PARTNERSHIP PROJECT (3GPP), MOBILE COMPETENCE CENTRE ; 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS CEDEX ; FRANCE, vol. SA WG5, no. V14.0.0, 14 March 2017 (2017-03-14), pages 1-36, XP051290465, [retrieved on 2017-03-14]	1,5,8, 12,15, 19,20,24
Y	* chapter 4.4.5 * * chapter 4.5.6 * figures 4.4.5.1-1 -/-	2,3,6,7, 9,10,13, 14,16,



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

4 October 2018

Date of mailing of the international search report

11/10/2018

Name and mailing address of the ISA/

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

Authorized officer

Bertsch, Andreas

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2018/044063

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	figures 4.5.6-1 -----	17,21,22
	INTEL: "pCR TR 32.864 add UC on NS update to instantiate a new VNF instance or add a PNF instance", 3GPP DRAFT; S5-171074 PCR TR 32 864 ADD UC ON UPDATING THE NS INSTANCE FOR GNB, 3RD GENERATION PARTNERSHIP PROJECT (3GPP), MOBILE COMPETENCE CENTRE ; 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS , vol. SA WG5, no. Porto; 20170116 - 20170120 15 January 2017 (2017-01-15), XP051218125, Retrieved from the Internet: URL:http://www.3gpp.org/ftp/Meetings_3GPP_SYNC/SA5/Docs/ [retrieved on 2017-01-15] * chapter 4.1 Life Cycle Management use cases * * chapter 4.1.y Updating a NS instance to add the PNF that is part of a gNB * * chapter 4.1.x Updating a NS instance to add the VNF that is part of a gNB * -----	2,3,6,7, 9,10,13, 14,16, 17,21,22
A	"Network Functions Virtualisation (NFV); Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification", ETSI GROUP SPECIFICATION, EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (ETSI), 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS ; FRANCE , vol. NFV IFA, no. V2.1.1 13 October 2016 (2016-10-13), pages 1-127, XP014279837, Retrieved from the Internet: URL:http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/013/02.01.01_60/gs_NFV-IFA013v020101p.pdf [retrieved on 2016-10-13] * chapter 7.2.5 * * chapter 7.3.5 * ----- -/--	1-24

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/044063

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	"LTE; Telecommunication management; Life Cycle Management (LCM) for mobile networks that include virtualized network functions; Requirements (3GPP TS 28.525 version 14.1.0 Release 14)", ETSI TECHNICAL SPECIFICATION, EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (ETSI), 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS ; FRANCE , vol. 3GPP SA, no. V14.1.0 26 July 2017 (2017-07-26), pages 1-49, XP014291792, Retrieved from the Internet: URL:http://www.etsi.org/deliver/etsi_ts/128500_128599/128525/14.01.00_60/ts_128525v140100p.pdf [retrieved on 2017-07-26] * chapter 6.4.3.5 * -----	1-24
A	EP 3 094 049 A1 (HUAWEI TECH CO LTD [CN]) 16 November 2016 (2016-11-16) paragraph [0006] - paragraph [0052] figure 1 -----	1-24

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2018/044063

Patent document cited in search report	Publication date	Patent family member(s)	Publication date	
EP 3094049	A1	16-11-2016	CN 105379191 A	02-03-2016
			EP 3094049 A1	16-11-2016
			KR 20160113250 A	28-09-2016
			US 2016337172 A1	17-11-2016
			WO 2015113278 A1	06-08-2015
