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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
02 January 2020 (02.01.2020)

WIPO | PCT

(10) **International Publication Number**
WO 2020/002761 A1

(51) **International Patent Classification:**
H04L 12/24 (2006.01) ***H04W 48/18*** (2009.01)
H04L 29/08 (2006.01) ***H04W 84/04*** (2009.01)
H04W 28/08 (2009.01) ***G06F 9/455*** (2018.01)
H04W 40/32 (2009.01) *H04W 40/20* (2009.01)

(21) **International Application Number:** PCT/FI2019/050395

(22) **International Filing Date:** 22 May 2019 (22.05.2019)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
62/690,277 26 June 2018 (26.06.2018) US

(71) **Applicant: NOKIA TECHNOLOGIES OY** [FI/FI]; Karakaari 7, 02610 Espoo (FI).

(72) **Inventors: KAHN, Colin**; 69 Forest Way, Morris Plains, New Jersey 07950 (US). **CHANDRAMOULI, Devaki**; 3532 Thorp Springs Drive, Plano, Texas 75025 (US).

(74) **Agent: NOKIA TECHNOLOGIES OY** et al.; Ari Aarnio, IPR Department, Karakaari 7, 02610 Espoo (FI).

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

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

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

(54) **Title:** SUPPORTING INTERACTIONS BETWEEN STATELESS NETWORK FUNCTIONS IN A CORE NETWORK

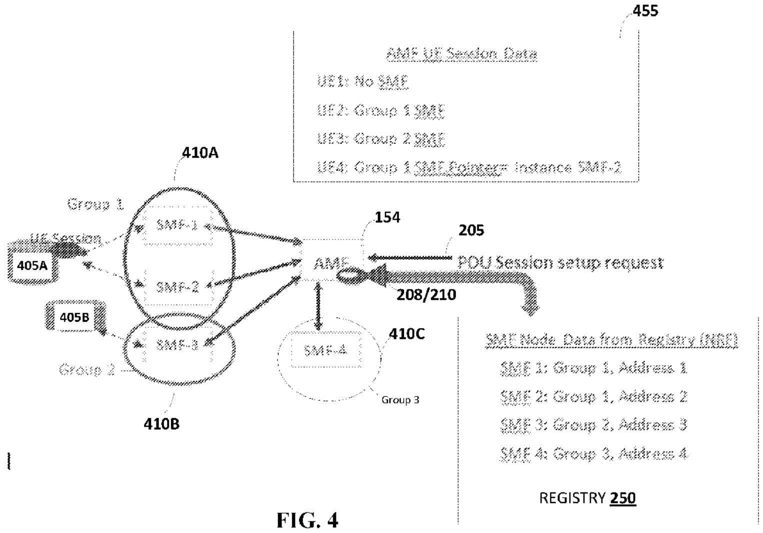

455
AMF UE Session Data
UE1: No SMF
UE2: Group 1 SMF
UE3: Group 2 SMF
UE4: Group 1 SMF Pointer= Instance SMF-2
410A
Group 1
SMF-1
UE Session
405A
SMF-2
405B
SMF-3
Group 2
410B
154
205
AMF
PDU Session setup request
208/210
SMF-4
410C
Group 3
SMF Node Data from Registry (NRF)
SMF 1: Group 1, Address 1
SMF 2: Group 1, Address 2
SMF 3: Group 2, Address 3
SMF 4: Group 3, Address 4
REGISTRY 250

FIG. 4

(57) **Abstract:** Methods and apparatus, including computer program products, are provided for supporting network functions. In some example embodiments, there may be provided a method that includes selecting, for network node, a network function instance from a group of network function instances having a common network function, the selection based on an identifier indicating that the network node and the selected network function instance are part of a compatible superset of different types of network functions; and accessing, by the network node, the selected network function instance. Related systems, methods, and articles of manufacture are also described.

SUPPORTING INTERACTIONS BETWEEN STATELESS NETWORK FUNCTIONS IN A CORE NETWORK

Field

The subject matter described herein relates to cellular networks.

Background

5th generation mobile networks (5G) will evolve to provide higher data rates with lower latency to more devices, when compared to 4G. Moreover, it is anticipated that 5G will provide support for additional services including device-to-device communications, machine-type communications, broadcast services, emergency communications, and Internet of Things communications. The 5G core network along with the 5G radio access network and user equipment will need to support these services, higher data rates, and lower latency.

Summary

Methods and apparatus, including computer program products, are provided for network function selection.

In some example embodiments, there may be provided a method that includes selecting, for network node, a network function instance from a group of network function instances having a common network function, the selection based on an identifier indicating that the network node and the selected network function instance are part of a compatible superset of different types of network functions; and accessing, by the network node, the selected network function instance.

In some variations, one or more of the features disclosed herein including the following features can optionally be included in any feasible combination. The identifier may include a superset identifier indicating compatibility among the different types of network functions and with respect to at least one attribute. The at least one attribute may include connectivity, location, manufacturer, performance, capacity, optional features, and/or supporting a same network slice. The selecting may be further based on a group identifier indicating compatible instances of a network function of a certain type. The selected network function instance may be configured to provide at least an access management function, a session management function, a policy control function, a user plane function, a unified data management function, an authentication server function, a network exposure function, and/or a network slice selection function. The identifier may be obtained from one of a registry or a management system. The network function instance may include a virtualized network function instance. The group may include virtualized network function instances. The virtualized network function instance may be virtualized in a host computer and/or a virtual machine. The network node may include a consuming network function instance.

In some example embodiments, there may be provided a method that includes selecting, for a consuming network service instance, a producing network service instance from a group of producing network service instances offering a common network service, the selecting based on an identifier indicating that the consuming network service instance and producing network service instances are part of a compatible superset of different types of network service instances; and accessing, by the consuming network service instance, the selected network service instance.

In some variations, one or more of the features disclosed herein including the following features can optionally be included in any feasible combination. The identifier may include a superset identifier indicating compatibility among the different types of network services and with respect to at least one attribute. The at least one attribute may include connectivity, location, manufacturer, performance, capacity, optional features, and/or supporting a same network slice. The selecting may be further based on a group identifier indicating the network service instances. The identifier may be obtained from one of a registry or a management system. The selected producing network service instance may be configured to provide at least an access management function service, a session management function service, a policy control function service, a user plane function service, a unified data management function service, an authentication server function service, a network exposure function service, a network resource function service and/or a network slice selection function service.

The above-noted aspects and features may be implemented in systems, apparatus, methods, and/or articles depending on the desired configuration. The details of one or more variations of the subject matter described herein are set forth in the accompanying drawings and the description below. Features and advantages of the subject matter described herein will be apparent from the description and drawings, and from the claims.

Description of Drawings

In the drawings,

FIGs. 1A depicts an example of a portion of a 5G network, in accordance with some example embodiments;

FIG. 1B depicts examples of hosts for providing virtual network functions, in accordance with some example embodiments;

FIG. 2 depicts an example of a process for handling virtualized network functions, in accordance with some example embodiments;

FIG. 3 depicts an example of a registry query for a candidate list of virtualized network functions, in accordance with some example embodiments;

FIG. 4 depicts an example of a system including virtual network functions, in accordance with some example embodiments;

FIG. 5 depicts an example of a portion of a network including groups of virtualized network functions, in accordance with some example embodiments;

FIG. 6 depicts groups of VNF instances, in accordance with some example embodiments;

FIG. 7 depicts an example of an apparatus, in accordance with some example embodiments; and

FIG. 8 depicts another example of an apparatus, in accordance with some example embodiments.

Like labels are used to refer to same or similar items in the drawings.

Detailed Description

As wireless networks evolve towards 5G and beyond, some network functions may be virtualized. Rather than sending a request to a dedicated, specialized network node that performs a single network function (NF) for example, a host, such as a cloud-based computer server, may service the request using an instance of a NF that has been virtually instantiated. This virtualized network function (VNF) may allow what was once performed on the dedicated, specialized network node to be performed as a service on, for example, on general-purpose hosts, which may be hosted on, for example, cloud-based servers.

As this VNF framework evolves, there may be different levels of implementation maturity levels which may need to coexist. For example, VNFs may initially be adapted to work on general purpose hardware that provide a virtualized system in which the application logic is separated from the hardware and associated infrastructure layer operations and maintenance (OA&M). And, the user equipment's (UE) session may be assigned to a specific VNF instance. The evolution may also include network functions that support one or more somewhat independent software services. Software service instances, or the VNF instances supporting the service, may be discovered by a consumer and, once discovered, utilized by the consumer for a UE's session.

The evolution towards VNFs may also include compute and storage separated in the network infrastructure layer, so the VNFs can utilize available resources as needed. For example, the VNFs may store context information, for assigned UE sessions, in storage, such as a database that is separate from the VNF processing. This separation may enable storage resources to be scaled independently of the host's compute resources being used for VNF processing. NFs may, for example, store assigned user sessions in a storage function, such as a database, that is separate from NF's "compute" processing, enabling a release of the processing resources after a transaction is completed. This may also allow efficient use of processing resources, and independent scaling of storage and compute resources.

The VNF evolution may also include VNFs that are semi-stateless. For example, a given UE's session context may be retrieved from storage, and for each occurrence of the UE's activity during a given UE session, the session context may be dynamically assigned to different VNF instances. But during a given occurrence of UE activity, all transactions may be directed to a selected VNF instance. When the UE becomes dormant during the session, the corresponding session information (for example, the context of the session) may be returned to storage, which can be accessed by a certain group of VNF instances. When the UE's activity resumes, another, different VNF instance (which is part of the group) may be selected to resume handling, based on the stored session information, of the transactions for the UE's session.

A group of VNFs may be functionally the same or similar in some respects but currently have different UE session data for different corresponding UEs. For example, a group of VNF instances may all be functionally similar. To illustrate further, a group of VNFs may include VNFs which each provide an access management function (AMF), while another group of VNFs may each provide a session management function (SMF). Each of the VNFs in a given group may be functionally the same or similar, and each of the VNF instances in a group may be configured with different session information for UEs being served (e.g., those UEs with on-going transactions). Each group of VNF instances may also be identified by a group identifier (ID), in accordance with some example embodiments. The group IDs may be assigned by a network management plane and stored at, for example, a registry an example of which is a network repository function (NRF).

In some example embodiments, the different VNF groups may each provide a different type of NFs, such as an access management function (AMF) VNF, a session management function (SMF) VNF, a policy control function (PCF) VNF, and/or other types of network nodes which can be implemented as a VNF.

Some VNFs may be stateless. When stateless, the UE's session(s) may be decoupled from VNFs, so the UE may be dynamically assigned to a given VNF instance only for the duration of a single transaction (which may be on the order of milliseconds, for example). After the transaction is complete, the UE's session information may be returned to a storage, such as a unstructured data storage function (UDSF) accessible by a certain group of VNF instances.

When a subsequent transaction arrives for a UE, a new (for example, another and/or different) VNF instance may be selected from a group of VNFS instances and then configured based on the stored session information, such as the UE's context information. The session information may indicate the UE is authorized to receive/provide a service, include encryption keys for protecting user plane data and control plane messages, indicate network functions currently holding state information for the UE, provide network slice information associated with the UE, include tunnel endpoint information (for example, of a RAN), include QoS for which the UE is authorized, include private group membership and serving area restrictions, and/or include other context or session information.

The stateless VNF instances may be considered stateless from the perspective of the UE sessions, but the VNFs may be stateful in other respects such as having node-level state information. At the node level for example, a VNF may have a state, such as whether the VNF instance is in or out of service, in or out of overload, and/or persistent associations with other functions and platforms (for example, for OA&M and analytics).

To illustrate further, a VNF may be stateless, so that each transaction, even within the same UE session or across different sessions, can be handled by a different VNF instances having access to the session information stored in storage. For stateless VNFs, a transaction may be sufficiently short (for example, on the order of milliseconds) so that the consuming NF may wait if necessary for an ongoing transaction to complete and/or the UE session information to be returned to storage before the consuming NF selects a new VNF instance for the next transaction. The VNF instance may, as noted, be semi-stateless in the sense that the UE state may be held for a period of UE activity (which may be seconds to minutes in duration, for example) before the UE session information is returned to storage. The consuming NF may, as such, require a mechanism to determine if there is an assigned VNF instance. If so, the consuming NF can select the assigned VNF instance before attempting to choose a new VNF instance and retrieve context information from storage for the next transaction.

In some example embodiments, if a VNF instance has a group ID assigned, the VNF instance may correspond to a VNF instance that is stateless or semi-stateless, so any of the VNF instances in the group (which is identified by the group ID) can serve a transaction for a given UE. In some example embodiments, the group ID may be assigned to VNF Instances of the same functional type that also share other properties, such as access to the same stored UE session information. For example, a group ID may be assigned to a group of AMF VNF instances, a group of SMF VNF instances, a group of PCF VNF instances, and/or other types of network node instances implemented as a VNF. The group ID may, for example, indicate that the NF instances in each of these cases may access the same data storage where UE session information is kept.

In some example embodiments, there may be provided a system in which stateful, semi-stateless, and stateless VNF instances can be discovered and selected. The VNFs may be configured to operate with a services-based interface that can be accessed by any authorized consuming NF, and a mechanism using a network resource function (NRF) may also be defined so a consuming NF can discover producing NFs and the services the producing NFs support. Furthermore, an unstructured data storage function (UDSF) may be configured for storage and retrieval of opaque (e.g., not standardized) data by any NF. For example, the UDSF may be used to store UE session information that may be retrieved by any NF that is a member of the same group. The UDSF may be used, for example, in procedures for graceful access management function (AMF) removal from service and for automatic AMF failure recovery. The UDSF may facilitate both these procedures as replacement AMF instances can retrieve a stored UE context independent of the failed AMF.

In some example embodiments, VNF instances in different functional types of VNF groups (e.g., SMF VNF instances, AMF VNF instances, policy control function (PCF) VNF instances, and/or the like) may be associated with an identifier, such as a superset ID, to identify compatibility among different type of VNFs or different types of groups of VNFs. For example, VNF instances in the same superset (e.g., sharing the same superset ID) may be used together as the VNFs in the same superset may be considered compatible.

In some example embodiments, the VNF instances having the same superset ID may have compatible connectivity, although the VNF instances in the same super set may be compatible in other ways, such as having the same or similar location, supporting the same network slices, having the same optional features and services, from the same vendor, having similar performance and capacity, sharing common compute resources, and/or the like. For example, a SMF VNF instance having the same superset ID as an AMF VNF instance may indicate that these two, different types of VNF instances are compatible at least with respect to connectivity.

FIG. 1A depicts an example of a portion of a 5G wireless network 100, in accordance with some example embodiments. For example, the 5G wireless network 100 may include a user equipment (UE) 150 configured to wirelessly couple to a radio access network (RAN) 152, such as a base station and/or other type of wireless access point. The network 100 may include a core network, which may include an access management function (AMF) 154, a session management function (SMF) 156, a policy control function (PCF) 160, and a user plane function (UPF) 158. The network 100 may also couple to other nodes and networks, such as a data network (DN) 162, a registry 166 such as a network repository function (NRF), and/or other nodes or networks. FIG. 1A also depicts service interfaces, such as N1, N2, N3, N4, N6, N7, N11, N15, and/or the like. The architecture and the service interfaces may be defined in accordance with a standard, such as 3GPP TS 23.501.

The network nodes depicted at FIG. 1A may be implemented as dedicated, specialized network function nodes. However, one or more of the network nodes may be implemented as a virtualized network function (VNF), in accordance with some example embodiments. For a packet data unit session request associated with the UE 150 for example, the AMF 154 may send a request to the SMF 156, which may be implemented as a VNF, in accordance with some example embodiments. For example, a plurality of VNFs may be instantiated at one or more host computers, and some of the VNF instances may be of a type that can provide the SMF network function. To illustrate further, a cloud-based server may provide a group of VNF instances each providing, for example, a SMF, provide a group of VNF instance each providing, for example, an AMF, and/or provide a group of VNF instances each proving other NFs.

FIG. 1B depicts an example of a system 199 including host computers 302-306 for instantiating VNFs, in accordance with some example embodiments. Although some of the examples refer to the SMF being virtualized as an instance of VNF ("VNF instance"), other network functions, such as the AMF, the PCF, the UPF, and/or other types of network functions and/or services, may be virtualized, in accordance with some example embodiments. Moreover, although some of the examples herein refer to a one-to-one mapping between network function and VNFs, a given VNF may support multiple network functions as well. For example, a given VNF may support the AMF and SMF, or the SMF and UPF, or the SMF, AMF, user plane function (UPF), and PCF.

To illustrate by way of the example of FIG. 1B, when a request is received for a network function or service, the requesting NF may send the request to VNFs 310A-F instantiated on host computers 302-306. If a VNF has not been instantiated and/or additional VNFs are needed, a host computer may trigger instantiation of a VNF. To illustrate further, the VNF instances may be instantiated dynamically and/or asynchronously with individual requests (for example, based on the load level of VNFs). The instantiated VNF instance may then respond to the request. As used herein, the sender of a request (which in turn receives a response, for example) may be referred to as a consumer or consuming NF or VNF, while the receiver of the request (which produces a response or action) may be referred to as a producer or producing NF or VNF.

Each of the host computers 302-306 may include at least one processor circuitry and at least one memory circuitry including instructions such as program code which when executed by the processor circuitry causes operations associated with the host computer. The host computers, such as cloud-based servers, may be coupled to a network, such as the Internet, a private network, an intranet, a core 3GPP network, and/or other nodes of a network, such as the 3GPP network. Moreover, each of the host computers may include virtual machines that emulate a physical computer (or aspects of a computer), and the virtual machines may host one or more VNF instances. The host computers may also include a hypervisor to operate these virtual machines. In operation, the system 199 may instantiate one or more VNF instances in a virtual machine. Furthermore, the host computers may include, or be coupled to, a dispatcher 195 for routing requests to certain VNF instances and/or triggering the instantiation of a new VNF instance. For example, a given VNF instance, such as VNF instance 310A, may be identified by an address. The address may include an address of the host computer 302, such as an IP address, a media access control address, and/or other type of address. Alternatively or additionally, the address may include an instance identifier (ID) indicating the specific instance for the VNF 310A, for example. The instance ID may identify the instance of VNF 310A from among other instances at the host computer 302.

In some example embodiments, some of the VNF instances may be grouped. When this is the case, a group ID may be used to refer to at least one VNF instance. When the VNFs are part of the same group, the VNF instances in that group may all have the same group ID. Moreover, the VNF instances in the group may all perform the same or similar network function and/or may have access to a common storage, such as an unstructured data storage function (UDSF) (e.g., a database, containing a UE's session information including the UE context information for a given UE session).

In some example embodiments, the VNF instances in the group may all perform the same or similar network function and/or may all be similarly configured (for example, same or similar version of software, same or similar manufacturer, same or similar network function, same or similar slice, and/or the like). As such, any of the VNF instances in a group may be able to respond to a request given access to the UE's session information stored at the storage. In some example embodiments, the group ID may be provided for a group of VNF instances, when each of the VNF instances in the group are stateless or semi-stateless.

In some example embodiments, an identifier, such as the superset ID, may be used to identify compatibility among different functional types of VNF instances. For example, VNF instances 310A-B, and F may each correspond to an SMF network function, and VNF instances 310C-E may each correspond to an AMF network function. In this example, the SMF instances 310A-

B and F may share the same group ID (e.g., group ID “1”) indicating the same functional type (which in this example of SMF) and other common attributes (e.g., access to the same common storage for UE session data), while VNF instances 310C-E may share the same group ID (e.g., group ID “2”) indicating the same functional type (which in this example of AMF), and other common attributes (e.g., access to the same common storage for UE session data, which may be different from the shared storage used by VNF instances belonging to different groups).

In some example embodiments, SMF VNF instance 310B and AMF instance 310C may be part of the same superset and thus be compatible at least with respect to an attribute such as connectivity, although the compatibility may be with respect to other attributes such as location, manufacturer, performance, capacity, optional features, and/or supporting a same network slice. When this is the case, the SMF VNF instance 310B and the AMF instance 310C may share the same identifier, such as the superset ID, to identify compatibility among different functional types of VNF instances. For example, SMF VNF instance 310B and AMF instance 310C may share the same superset ID indicating compatibility at least with respect to compatible communications/connectivity, so SMF VNF instance 310B may be able to access AMF instance 310C (and/or vice versa).

FIG. 2 depicts an example process 200 for handling network functions, in accordance with some example embodiments. The description of FIG. 2 also refers to FIGs. 1A, 3 and 4.

At 205, a request may be received, in accordance with some example embodiments. For example, a consuming NF (which in the instant example is AMF 154 although other NFs may be used as well) may receive, from the UE 150, a request, which in this example is a session initiation request such as a packet data unit (PDU) session setup request. Although the instant example refers to the VNF being triggered by a session initiation setup request, the process 200 may operate with other types of requests and/or transactions as well.

At 208, a consuming NF, such as the AMF 154, may query a registry 166 to request a list of candidate producing VNF instances, in accordance with some example embodiments. For example, the AMF 154 may need a producing NF, such as a VNF instance configured to provide a SMF 156 type network function. When this is the case, the AMF 154 may send to the registry 166 a request for candidate VNF instances, and the request may include the type of function requested (which in this example is the SMF network function).

Although the example used with respect to FIG. 2 refers to a specific consuming NF, such as AMF 154, and a specific producing VNF providing an SMF network function 156, these are examples as other network functions may serve as consuming and/or producing NFs and/or may be virtualized as a VNF instance, in accordance with some example embodiments. Examples of the other types of NF nodes which can be virtualized as VNF instances include the PCF, the (unified data management) UDM, the authentication server function (AUSF), the network slice selection function (NSSF), the UPF, a network exposure function (NEF), and/or the like.

At 210, the consuming NF, such as the AMF 254, may receive a list of candidate producing VNF instances which may be able to provide the requested type of network function, in accordance with some example embodiments. In the example of FIG. 2, the registry 166 may return a list of candidate VNF instances configured to provide the SMF type network function. The response provided at 210 may also include, in accordance with some example embodiments, other information, such as metadata for the VNFs. In some example embodiments, the metadata may include the addresses of the VNFs, the type of VNF instance (for example, the NF type, such as SMF type), and/or the like. In some example embodiments, the metadata may also include at least one group ID identifying VNF groups (of the same or similar type of VNF functional type) and/or instance IDs identifying specific instances of VNFs. In some example embodiments, the metadata may include an identifier, such as the superset ID, indicating compatibility across different types of VNF instances (or groups of VNF instances).

FIG. 3 depicts an example interaction between the registry 166 and a consuming network function 310, in accordance with some example embodiments. The description of FIG. 3 also refers to FIG. 2.

In the example of FIG. 3, the consuming network function 310 (which in the current example is AMF 154 although other types of consuming NFs may be used as well) may interact with the registry 166, which may be a NRF though other registries may be used as well. The consuming network function 310 may query, at 208, the registry 166 to discover which producing VNF instance(s) offer a given type of service or functionality. For example, the registry may include metadata such as the group ID to identify the same or similar functional types of VNF instances. Alternatively or additionally, the registry may include metadata, such as a superset ID, to identify compatibility across different types of VNF instances and/or groups of VNF instances.

The consuming network function 310 may subscribe to the registry 166 to receive updates, such as when a producing VNF of interest changes (for example, when a VNF instance fails, is added, is modified, is deleted, undergoes a type or configuration change, and/or the like). In the response at 210, the registry may, as noted, respond with the addresses of the VNF instances as well as other metadata, such as the group ID, superset ID, and/or the like. In some example embodiments, the metadata may include a group ID to indicate which VNF instances are part of the same functional group and/or may share other attributes such as access to a common storage database. In some example embodiments, the metadata may include a superset ID to identify compatibility across different types of VNF instances and/or groups of VNF instances.

When the VNFs are part of the same group, the VNF instances in that group may, as noted, have access to a common storage, such as a database, containing a UE's session information including the UE context information for a given UE session. In some example embodiments, a group ID and/or superset ID may be assigned, by a management plane function, to the stateless/semi-stateless groups of VNFs, and this assignment may be stored at the registry 166.

Referring again to FIG. 2, a consuming NF may select, at 212, at least one of the candidate VNF instances, in accordance with some example embodiments. The selection may be based on metadata obtained from the registry 166 as well as other information. To illustrate by way of an example, the consuming NF may base the selection on the type of network function provided by the VNF instance, the instance ID identifying a specific VNF instance, a group ID identifying a specific group of VNF instances, the superset ID to identify compatibility across different types of VNF instances and/or groups of VNF instances, the NF capacity, the services supported by the NF, UE session information, whether the VNF instances are stateful, stateless, or semi-stateless, and/or other information. The group ID may identify (for example, point to) a group of VNF instances; instance ID may identify a specific VNF instance at a host computer; while the superset ID may indicate compatibility among different functional types of VNFs.

To illustrate further, a consuming NF, such as the AMF 254, may take into account the assignment state of the UE session associated with the UE's request at 205. For example, if the there is no VNF instance assigned to the UE session (associated with the UE's request or transaction at 205), the AMF may be free to select any VNF instance. If however the session information indicates a group of VNF instances (which are assigned the same group ID, for example) is assigned to the UE session associated with the UE's request/transaction at 205, the AMF may select any VNF instance in the group of VNF instances. Moreover, the selection may be based on the superset ID to identify compatibility across different types of VNF instances and/or groups of VNF instances. And, if a specific VNF instance in a group of VNFs has been assigned to the UE session associated with the UE's request at 205, the AMF may assign that specific VNF instance from the group of VNFs.

At 214, the consuming NF, such as the AMF 254 in this example, may determine the assignment state for the UE session being handled, in accordance with some example embodiments. For example, the AMF may examine UE session information to see if a pointer (for example, an identity or address) is stored or assigned to the UE's session (as described further below with respect to FIG. 4 at 455). The UE session information may be cached locally at the AMF, obtained from a registry, shared database, and/or other sources as well. The pointer may identify a VNF instance.

FIG. 4 depicts an example of VNF instances grouped 410A-C, in accordance with some example embodiments. In the example of FIG. 4, the VNF instances correspond to the SMFs 1-4, although as noted above other types of network functions, or combinations of network functions can be virtualized and grouped as well.

In the example of FIG. 4, when the consuming NF, such as AMF 154, receives the list of candidate VNF instances, such as SMF 1, SMF 2, SMF 3, and SMF 4. The AMF 154 may also receive, from the registry 166 (or from another node or locally cached), metadata for each of the VNF instances. In the example of FIG. 4, the SMFs 1-2 are part of group 1 410A as shown by the common group ID "1", SMF 3 is part of group 2 410B as shown by the common group ID "2", and SMF 4 is part of group 3 410C as shown by the common group ID "3". The metadata also includes an address, or pointer, for each VNF instance. For example, SMF 1 may have an address comprising (1) an IP address identifying the host computer hosting the VNF instance comprising SMF 1 and/or (2) an instance identifier, indicating at the host computer, the VNF instance comprising SMF 1.

The registry information 250 shows that the SMF 1 and SMF 2 are both in group 1, so SMF 1 and SMF 2 can be used interchangeably as SMF 1 and SMF 2 may be stateless or semi-stateless VNFs. However, the SMF 3 and the SMF 4 may be of any type (for example, stateful, stateless, or semi-stateless). As SMF 3 and the SMF 4 have their own unique group IDs, a session may be bound to SMF 3 or SMF 4, when established.

The VNFs in the same group may, as noted, have access to a common storage such as a database in order to obtain session information. For example, group 1 410A including SMFs 1 and 2 have access to database 405A which stores session information including UE context information for a UE session, while group 2 410B including SMF 3 has access to database 405B.

At FIG. 4, the AMF 154 may select an SMF VNF instance for each UE 1-4 with the indicated session information at 455. In the case of UE 1 (which has no assigned SMF VNF instance), the AMF may pick any SMF VNF instance as the AMF is handling a new session for UE 1. Regarding UE 2, it has had a session previously handled by the group 1 SMF VNF instances. As the session information 455 for UE 2 does not include a pointer to a specific instance ID in group 1, the AMF may select any SMF VNF instance within group 1 (for example, SMF 1 or

SMF 2) for further interaction involving UE 2's session. When the SMF VNF instance receives a message from the AMF, the SMF VNF instance will retrieve the SMF session information for UE 2 from the common storage, such as database 405A. Regarding UE 3, UE 3 has had a session previously handled by the group 2 SMF VNF instance. In the example of FIG. 4, SMF 3 is the only group 2 SMF VNF instance, so this SMF VNF instance must be selected by the AMF. Regarding UE 4, the session information identifies group 2 and a pointer identifying SMF 2 VNF instance. In this example, the AMF must send all messages associated with the transaction to SMF 2 VNF instance. When the transaction or period of UE 4 activity ends, the SMF 2 VNF instance may release the UE session information to the storage 405A, and notify AMF that UE 4 session information should be updated to remove the pointer to SMF 2 VNF instance.

Referring again to FIG. 2, the consuming NF such as AMF 154 may, at 214, need to take into account the assignment state as indicated by the UE's session information (for example, session information 455) associated with the UE.

If there is no VNF instance or VNF group assigned to the UE's session as indicated by the UE's session information, any VNF instance may be selected from the candidate list (yes at 216 and 218). This assignment may also take into account the superset ID, in accordance with some example embodiments. Moreover, a pointer identifying the identity or location of the selected VNF instance (and/or group) may be recorded, at the consuming NF, for the UE session being handled. For example, the AMF 254 may add the pointer to the UE request before forwarding the request, store the pointer locally at the AMF as session information, and/or store the pointer at a registry. The pointer may be any type of identifier of location or identity. For example, the pointer may comprise an IP address, fully qualified domain name, MAC address, and/or other indication of the identity or location of a group ID and/or VNF instance.

If there is a VNF instance and VNF group assigned to a UE session as indicated by the UE's session information, the assigned VNF instance (which may be designated in a pointer in UE session information) may be selected (yes at 220 and 222).

If only a VNF group is assigned to a UE session as indicated by the UE's session information, any VNF instance from the VNF group assigned to the UE session may be selected (230 and 232). For example, any VNF instance in the VNF group indicated by the VNF group ID may be selected. In some example embodiments, a superset ID may be taken into account as well. Moreover, a pointer to the selected VNF instance may be recorded for the UE session being handled.

Referring again to 218, the consuming NF may then forward, at 242, the UE session request to the VNF instance selected at 218, in accordance with some example embodiments. For example, AMF 254 may forward a session initiation request 205 to the selected VNF instance. At 244, the selected VNF instance may (if needed) then retrieve, from storage, UE session information (e.g., content information) and then proceed to process the UE request 205.

Referring again to 222, the consuming NF may then forward, at 250, the UE session request to the VNF instance selected at 222, in accordance with some example embodiments. For example, the AMF 254 may forward a session initiation request 205 to the VNF instance selected at 222. This selected VNF instance may then process, at 252, the UE request 205. If the selected VNF has been previously assigned, the selected VNF may not need, at 252, to access storage for UE context information as the selected VNF may already have the session or context information to handle the UE's request. If the UE context information is needed at 252, the select VNF may access that information from storage as noted at 244, for example.

Referring again to 232, the consuming NF may then forward, at 260, the UE session request to the VNF instance selected at 232, in accordance with some example embodiments. At 262, the selected VNF instance may then retrieve, from storage, UE content information and then proceed to process the UE session request.

In accordance with some example embodiments, the consuming NF such as AMF 154 may release the pointer to the VNF, when there is a change or failure in the handling the UE session.

Referring again to 212, initially (for example, prior to the first selection of a producer VNF instance by the consuming NF), the pointer in the UE session information may not be known or populated in the UE's session information or a message, for example. When this is the case, the consuming NF may pick any producing VNF instance within a corresponding VNF group. If a superset ID has been assigned to the consuming NF, the consuming NF may pick any producing VNF instance within the group that also has the same superset ID. The address for the picked producing VNF instance may then be used as the pointer that is used to populate a transaction or request message, and the pointer may be maintained for the transaction(s) with the producing VNF instance. If later the selected producing VNF instance wants to allow the consuming NF to pick a different producing VNF instance, the producing VNF instance may indicate to the consuming NF to release the pointer. At this point, the prior/old producing VNF instance (requesting the release) may then store its UE session information in storage, such as database 405A or other type of data storage function accessible by the VNFs in the old producing VNF's group. If the consuming NF detects that a producing VNF has failed, the consuming NF may autonomously release the pointer. This allows a new VNF instance from the producing VNF's group to be selected for the next transaction. The producing VNF may release the pointer for a specific UE only, release the pointer for all UEs, and/or release the pointer for sessions served by the producing VNF.

As noted, the VNF instances may be configured to provide other types of network functions. In the case of a PCF for example, the VNF instances in the same group may be able to access the same database including UE Policy and Charging Control Sessions. In the case of a network function comprising unified data management (UDM, see, for example, 3GPP TS 23.501), the VNF instances comprising UDM front-ends in the same group may be able to access the same unified data repository/shared data layer where subscriber data is stored. In the case of the SMF, the VNF instances in the same group may be able to control sessions on the same UPFs. In the case of the AMF, the VNF instances in the same group may have connectivity to the same RAN and/or AN nodes.

In some example embodiments, the producing VNF instance (which in the instant example is a stateless VNF instance, such as the SMF instance) may store session information, such as the UE context, in storage at the end of a transaction. For subsequent transactions, the AMF (which serves as the consuming NF in this example) may use the previously selected producer NF (for example, a serving NF) instance as indicated by a pointer stored in the session information/UE context of the AMF.

In some example embodiments, the producing VNF instance (which in this example is a semi-stateless VNF instance, such as an SMF instance) may store the session information, such as the UE context, in storage at the end of a given period of UE activity (which may be determined based on local policies of the producer NF). And, the SMF instance may inform consuming NFs that the consuming NF may pick a new VNF, when the UE's activity resumes. The SMF instance in this example may inform the consumer NF to remove the association between the UE context and the producer NF instance pointer. At the beginning of a new activity period, a consuming NF may select any NF from the group of NF instances identified by a NF group ID (for example, SMF Group ID). If a superset ID has been assigned to the consuming NF, the consuming NF may pick any producing NF instance within the group that also has the same superset ID. During periods of activity, the consuming NF may pick the currently selected producer NF as indicated by the pointer in its own context.

In some example embodiments, the producing VNF instance (which in this example is a semi-stateless, dedicated VNF instance such as an SMF instance) may store the UE context in store at the end of a period of UE activity (determined based on local policies of the producer NF, for example). The consuming NF may pick the currently selected producer NF as indicated by a pointer in its own context.

In some example embodiments, the producing VNF instance (which in this example is a stateful VNF instance such as an SMF instance) may store the UE session information or context locally. For subsequent transactions, the consuming NF uses the previously known producing VNF (which is serving the consuming NF) pointer to select the NF and forward the request.

From the consuming NF perspective such as the AMF 154, the consuming NF may know whether a pointer to a producing VNF instance has been populated with a valid address, such as the instance identifier. When the pointer is populated, the consuming NF is interacting directly with the producing VNF instance at the specified address. When a pointer is not populated but the group is populated, the consuming NF may select (which may be based on the group ID and/or superset ID) any VNF instance from the group and populate the pointer with the corresponding address of the VNF instance. Similarly, the registry 166 may not know explicitly the type of NF (stateless, semi-stateless, and/or stateful). An identifier, such as a unique ID or special group ID, may indicate that a substitute producing VNF instance cannot be chosen once the producing VNF instance has been selected.

In some example embodiments, the consuming NF may store both the VNF group ID and a pointer to the VNF (if populated) as part of the UE session information such as the UE context data. If the producing VNF is stateless, or semi-stateless with an idle UE, then the pointer may not be assigned to a specific VNF instance, so consuming NF can select any VNF (with the same superset ID if assigned) from the VNF group. But if the producing VNF is holding a UE state, then the VNF pointer will be populated with the specific VNF instance address.

In some example embodiments, a producing VNF instance that is stateless may request the consuming NF to remove the pointer at the end of each transaction. For example, the producing VNF instance, such as the SMF instance of FIG. 4, may send a message to the consuming NF, such as the AMF, to remove the pointer from the UE context, so that the consuming NF can remove the corresponding pointer to the producing VNF instance.

In some example embodiments, the registry 166 may be configured with an indication to not keep a pointer towards the producing VNF instance, when a transaction has been completed. When the consuming NF query's the registry for example, the metadata (which is returned to the consuming VNF) may include this indication to not keep the pointer after a transaction for certain stateless VNFs.

If demand for a service or network function increases, additional producing VNFs may be instantiated by a management plane. The additional VNFs may be registered with the registry 166, which in turn may send push notifications to inform consuming NFs that a new producer from a VNF group of interest is available. On the other hand, if demand for service wanes, producing VNFs may be removed from service by simply blocking new transactions. For example, the registry may remove a producing VNF from the registry, so that the producing VNF can no longer be discovered by consuming NFs. If a consuming NF has already discovered the removed producing VNF, the registry may push a message to the consuming NF regarding the removal. As ongoing transactions end or UEs become idle, the session information/UE context data in the producing VNF may be returned to a shared storage. Since no new transactions are started, the producing VNF can quickly become idle, after which it can be deactivated and removed from service.

Although a stateless VNF may eliminate the need for a Diameter routing agent or similar function to associate VNFs with UE contexts, semi-stateless VNFs may need this Diameter routing agent or similar function for non-Diameter protocols. For example, when a UE becomes active and an ongoing consumer-producer interaction has been initiated (for example, between the AMF and SMF), a new consuming NF may want to find the same producing VNF instance that currently has the UE context. For new transactions to semi-stateless VNFs, the producer indicated by the VNF pointer may need to be validated. If a pointer to another VNF instance in the same group already exists, the transaction may be redirected / forwarded to the relevant service producer VNF instance and the consuming NF told to replace the selected pointer with the current pointer to the relevant service producing VNF instance).

As noted above, a superset of NF instances among different NF types, such as AMF, SMF, PCF, and/or the like, may be used to identify compatible NF instances, in accordance with some example embodiments. For example, different types of VNF instances (which may be assigned to different VNF groups and/or have different group IDs) may be in the same superset. The different types of VNF instances in the same superset may be used together as the VNFs in the same superset are compatible at least with respect to connectivity and/or are otherwise compatible with respect to, for example, same location, supporting the same network slices, and/or the like.

In some example embodiments, a superset ID may, as noted, be used to identify compatible NF instances spanning different NF types, such as AMF, SMF, PCF, and/or other types of network functions. For example, NF instances in the same superset may be used together as the VNFs in the same superset are compatible at least with respect to connectivity and/or are otherwise compatible with respect to, for example, same geolocation, supporting the same network slices, having the same optional features and services, from the same vendor, having similar performance and capacity, sharing common compute resources.

To illustrate, a superset may be used as follows. When instantiated, VNF instance may be assigned, in accordance with some example embodiments, to a superset. This assignment may be based on the VNFs connectivity or other attributes, such as geolocation, supporting the same network slices, and/or the like. The assignment of the VNF instance may be performed by associating the VNF instance to a superset ID, although the identifier may take other forms as well. For a given VNF instance, the superset ID may be registered in a network node, such as the NRF. The registration may be performed by the VNF instance itself (e.g., self-registration), although another node or the management plane may handle the registration. This registration may enable the superset ID to be discoverable by other devices, such as consumer NFs or VNFs that seek to select a producer NF or VNF. Whenever a consumer VNF selects a producer VNF, the consumer VNF may choose a producer VNF that has the same superset ID as the consumer VNF. In this way, the superset ID may be used to ensure compatibility when selecting different functional types of VNF instances.

With respect to the consumer VNF selecting a producer VNF, this may take place under normal conditions/operation after, for example, a stateless producer VNF releases a UE session context to a UDSF. However, the consumer VNF selecting a producer VNF may also take place under a variety of other circumstances, such as after scale-in or other scenarios when a producer VNF is removed from service for example, during recovery from producer NF failure, and/or the like.

In some example embodiments, the superset ID may be assigned on a per VNF instance basis (e.g. each VNF instance is assigned a superset ID). Alternatively or additionally, the superset ID may be assigned on a per VNF group (e.g., each group of VNFs may be assigned the same value of the superset ID). Alternatively or additionally, the superset ID may be assigned on a per-service instance basis, rather than per NF instance. When assigned on a per service instance basis, this may accommodate scenarios where an operator may limit the scope of a service rather than NF connectivity. A producing NF may offer one or more services to a consuming NF. For example, an AMF may offer communications, event exposure, and/or mobile termination (MT) services. An SMF may offers PDU session and event exposure services. An NRF may offer NF management and NF discovery services. A consuming NF or service may select a producer service using group and/or superset identifiers in the same manner as described herein for NFs. When this is the case, a service instance, rather than an NF instance, is selected, and the group and superset IDs may be defined per-service rather than per-NF instance.

In the case of a per NF instance assignment, the superset ID may capture (e.g., include or be linked to) VNF instance attributes not captured by the group ID but may be relevant for VNF selection. For example, the group ID may be assigned to all VNF instances that share the same UDSF, but connectivity between those instances and other VNFs may vary as explained further below with respect to FIG. 5.

In the case of the per NF group assignment, this may be used to ensure compatibility between all VNF instances within a group and VNF instances of a different type in another group as shown further below with respect to FIG. 6.

In some example embodiments, compatible VNF groups may be given the same group ID values as an alternative to the superset ID noted above. When this is the case, the group ID takes into account a superset of compatibility. When this is the case, the group ID may be assigned to a compatible group of VNFs. For example, a group of AMF instances having a group ID of 1 may be compatible with other VNF groups having a group ID of 1, such as SMF VNF group having a group ID of 1, PCF VNF group having a group ID of 1, UDM VNF group having a group ID of 1, and/or the like. However, this use of the group ID may be limited in the sense that different functional types of VNFs share the same group ID. While a separate superset ID (which may be used in addition to the group ID) may enable the flexibility of different VNF types to be grouped differently (e.g., more flexible way of grouping each VNF depending on their functionality).

The AMF VNF grouping may also depend on the radio access network (RAN) and UE associations, while SMF VNF grouping may depend on UPF association, and UDM VNF grouping may depend on the unified data repository (UDR) association, and/or the like. To illustrate further, a first PCF VNF instance (PCF 1) and a second PCF instance (PCF 2) may belong to different groups of VNFs (e.g., group 1 and 2, respectively) as these PCFs do not have access to a common UDSF containing UE session data. However, these PCFs may be acceptable for a first AMF instance (AMF 1) to choose either the PCF 1 or PCF 2, when making an initial group selection. In this example, AMF 1, PCF 1 and PCF 2 may have the same superset ID, but not share the same group IDs as at least PCF 1 and PCF 2 have different group IDs.

FIG. 5 depicts an example of a portion of a network including groups of VNFs coupled to corresponding UDSFs, in accordance with some example embodiments.

In the example of FIG. 5, VNF groups (e.g., group 1, group 2, and group 6) may be defined so that the VNF instances assigned to a given group are functionally the same or similar with respect to the same type of network function, offering the same service, sharing the same UDSF, being from the same manufacturer, supporting the same network slice(s), being geographically located in close proximity (e.g., to ensure latency of signaling is reduced and/or enabling negligible performance impact to the system), and/or otherwise share a common attribute.

At 505, a network node, such as an AMF 507A, may pick an VNF instance and/or a VNF group. For example, the AMF 507A may pick SMF group 1 509 having SMF VNF instances and pick PCF group 2 510 having PCF VNF instances to, for example, complete a protocol data unit (PDU) session set up. The AMF may select the groupings when the UE registers with the network, during a PDU session establishment setup, and/or at other times as well.

At 520, an SMF, such as SMF 512 (which is a member of SMF group 1 509) may store session information in, for example, storage, such as a UDSF or other type of storage. The storage of the session information may enable subsequent transactions to select another SMF instance from within SMF group 1 509 and configure a SMF VNF instance using the session information. The UDSF may or may not be the same as that used by other NFs in other groups. For example, SMF instances belonging to another group 4 (not shown) may have a separate UDSF from the UDSF used by group 1 509 SMFs. Similarly, AMF instances in group 6 507A may use a different UDSF from the UDSF used by SMF instances in Group 1 509.

At 530, the PCF 532 (which in this example is a member of group 2 510) may pick any SMF instance, such as SMF instance 534, from group 1 509. When there is an AF or PCF initiated transaction for example, the PCF 532 may pick the SMF instance 534 from group 1 509. This selection may be similar to the selection noted above with respect to process 200 above. However, in the example of FIG. 5, not all of the the SMF instances in group 1 509 have the needed connectivity to certain AMFs. For example, SMF 534 does not have connectivity to AMF 507A. This lack of connectivity for the group 1 SMF 534 illustrates the need for an identifier, such as the superset ID that groups VNF instances based on a given attribute, such as connectivity and/or the like, across the different functional types of VNFs.

Returning to the previous example of FIG. 5, SMF 536 does have connectivity to AMF 507A, while PCF 538 has connectivity to SMF 536. In this example, PCF 538, SMF 536, and AMF 507A may be part of the same superset and thus may be assigned the same identifier, such as the superset ID, to indicate compatibility at least with respect to connectivity.

At 540, the use of an identifier, such as the superset ID, may enable VNF instances in different types of VNF groups (and/or having different types with respect to network function) to be selected in a manner that ensures compatibility with respect to at least connectivity. In the example of FIG. 5, VNF instances 542 A-F all share the same superset ID 1 even across different network function types of VNF instances, while VNF instances 544A-F all share the same superset ID 2 even across different network function types of NF instances. In this way, the PCF VNF instance 544F can pick an SMF VNF instance 544C knowing, based on the superset ID, that the selected SMF instance is compatible (which in this example is connectivity) to the appropriate AMF 544B, for example. The same methodology, as noted above with respect to the superset ID and the connectivity example given above may be used to choose NF instances that are geographically in close proximity, are from the same vendor, offer a proprietary service or otherwise have a specific attribute of importance.

In the example of FIG. 5, all the SMF instances in group 1 509 offer the same services and have access to the same UDSF, but the group 1 SMF instances may differ in other ways, such as their ability to provide connectivity to a previously selected AMF. In some example embodiments, an identifier, such as the superset ID, may be assigned to each VNF instance or group of VNF instances. This identifier may indicate a certain type of compatibility (e.g., with respect connectivity) shared among groups of NF instances. The VNF instances in an assigned superset ID group share, in this example, compatibility with respect to connectivity, so these VNF instances can connect to each other. In the example of FIG. 5, the initially selected VNFs at PDU session setup can share a common superset ID. When a new VNF needs to be selected from within a group (in this case an SMF), compatibility between VNFs is assured by selecting a VNF instance from the same superset, as shown at 540.

FIG. 6 depicts groups of VNF instances, in accordance with some example embodiments. In the example of FIG. 6, all of the NF instances in a given group, such as group 1 610A, all perform the same or similar network function and/or may all be similarly configured (for example, same or similar version of software, same or similar manufacturer, same or similar network function, same or similar slice, and/or the like). Moreover, the VNF instances within the given group may all be compatible with respect to other attributes as well such as connectivity. For example, the SMF VNF instances of group 2 610B are compatible with the AMF instances in group 1 610A as both groups 610A-B are in the same superset as evident by the superset group ID 1. As such, an SMF VNF instance in group 2 610B has connectivity to any of the AMF VNF instances in group 1 610A. Likewise, an AMF instance in group 2 610C is compatible and, in this example, has connectivity to any of the SMF instances in groups 2 or 3 610B or 610B as both groups are part of the same superset as evident by the superset ID of 1. And, AMF instance in group 3 620A is compatible and, in this example, has connectivity to any of the SMF instances in group 4 620B as both groups are part of the same superset as evident by the superset ID of 2.

Thus, as shown in the example of FIG. 6, Group 1 and 2 AMFs 610A/610C and Group 2 and 3 SMFs 610B/610D are assigned to superset 1, and group 3 AMFs 620A and group 4 SMFs 620B are assigned to superset 2. In this example, when an AMF in group 1 needs to initially select an SMF, the AMF may choose any SMF instance in group 2 and group 3. The group 1 AMF may not select an instance in SMF group 4 as SMF group 4 belongs to a different superset. Moreover, when an AMF in group 3 needs to initially select an SMF, the AMF may choose an SMF from group 4, but the AMF may not select an instance in SMF groups 2 or 3.

Referring again to FIGs. 4, if a pointer to a producer NF or service is assigned, then the consuming node may choose a producer NF or service indicated by the pointer. However, if the pointer is not assigned but the group ID of the producer NF or service is assigned and superset ID is not assigned, then the consuming node may pick a producer instance (or, for example, a service instance) indicated by the group ID. When the pointer and group of the producer is not assigned but the superset ID of the producer is assigned, the consuming node may pick the producing NF or service instance with a superset ID that matches the consumer's superset ID. If the pointer is not assigned but the producer's group ID and superset ID are assigned, then the consuming node may pick the producing NF (or service instance) indicated by both group ID and superset ID. And if the pointer, group ID, and superset ID are not assigned, the consuming node may pick any producer NF or service.

In some example embodiments, a network function instance may be selected for a network node. This selection may be from a group of network function instances having a common network function. This selection may be based on an identifier indicating that the network node and the selected network function instance are part of a compatible superset of different types of network functions. The selected network function may then be accessed by the network node. Referring to the example at FIG. 5, the SMF 544C may be selected for the PCF 544F. SMF 544C may be selected from a group of SMFs having a common network function (which in this example is SMF). And, the selection here is based on the superset ID indicating that the network node, such as the PCF 544F, and the selected network function, such as SMF instance 544C are part of a compatible superset (e.g., superset 2 in the example of FIG. 5) of different types of network functions. The network node, such as PCF 544F, may then access the selected network function instance.

As noted, the NF instance may be implemented as a set of services in accordance with for example, 3GPP TS 23.501, section 7.2. When this is the case, a consuming network service may select a producing network service instance from a group of producing network service instances offering a common network service. Examples of network services include an access management function service, a session management function service, a policy control function service, a user plane function service, a unified data management function service, an authentication server function service, a network exposure function service, a Network Resource Function service and/or a network slice selection function service. And, the selection may be based on an identifier, such as the superset ID, indicating that the consuming network service instance and producing network service instance are part of a compatible superset of different types of network service instances. The consuming network service may then access the selected network service instance.

FIG. 7 depicts a block diagram illustrating a computing system 700, in accordance with some example embodiments. The computing system 700 may be used to implement the host computers 302, 304, 306 and the like at FIG. 7. Alternatively or additionally, the computing system may be used to provide the virtualized network functions (VNFs), in accordance with some example embodiments.

As shown in FIG. 7, the computing system 700 can include a processor 710, a memory 720, a storage device 730, and input/output devices 740. The processor 710, the memory 720, the storage device 730, and the input/output devices 740 can be interconnected via a system bus 750. The processor 710 may be capable of processing instructions for execution within the computing system 700. Such executed instructions can implement host computers 302, 304, 306 a network node including a home subscriber server, a serving call session controller function, a serving call session controller function instance, and/or other types of nodes, NFs, and/or VNFs. In some example embodiments, the processor 710 may host one or more virtual machines, each of which may include a network node, such as a serving call session control function instance as well as other types of nodes, NFs, and/or VNFs. The processor 710 may be capable of processing instructions stored in the memory 720 and/or on the storage device 730 to display graphical information for a user interface provided via the input/output device 740.

The memory 720 may be a computer readable medium such as volatile or non-volatile that stores information within the computing system 700. The memory 720 can store instructions, such as computer program code. The storage device 730 may be capable of providing storage for the computing system 700. The storage device 730 can be a hard disk device, an optical disk device, or a tape device, or other suitable storage mechanism. The input/output device 740 provides input/output operations for the computing system 700. In some example embodiments, the input/output device 740 includes a keyboard and/or pointing device. In various implementations, the input/output device 740 includes a display unit for displaying graphical user interfaces. Alternatively or additionally, the input/output device 740 may include wireless and/or wired interface to enable communication with other devices, such as other network nodes. For example, the input/output device 740 can include an Ethernet interface, a WiFi interface, a cellular interface, and/or other wired and/or wireless interface to allow communications with one or more wired and/or wireless networks and/or devices.

FIG. 8 illustrates a block diagram of an apparatus 10, in accordance with some example embodiments.

The apparatus 10 may represent a user equipment, such as the user equipment 110. For example, the apparatus 10 may be configured as user equipment 150. Alternatively or additionally, one or more portions of the apparatus 10 may be used to implement a network node, such as a base station, a core network node, and/or other types of nodes, NFs, and/or VNFs.

The apparatus 10 may include at least one antenna 12 in communication with a transmitter 14 and a receiver 16. Alternatively transmit and receive antennas may be separate. The apparatus 10 may also include a processor 20 configured to provide signals to and receive signals from the transmitter and receiver, respectively, and to control the functioning of the apparatus. Processor 20 may be configured to control the functioning of the transmitter and receiver by effecting control signaling via electrical leads to the transmitter and receiver. Likewise, processor 20 may be configured to control other elements of apparatus 10 by effecting control signaling via electrical leads connecting processor 20 to the other elements, such as a display or a memory. The processor 20 may, for example, be embodied in a variety of ways including circuitry, at least one processing core, one or more microprocessors with accompanying digital signal processor(s), one or more processor(s) without an accompanying digital signal processor, one or more coprocessors, one or more multi-core processors, one or more controllers, processing circuitry, one or more computers, various other processing elements including integrated circuits (for example, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), and/or the like), or some combination thereof. Accordingly, although illustrated in FIG. 8 as a single processor, in some example embodiments the processor 20 may comprise a plurality of processors or processing cores.

The apparatus 10 may be capable of operating with one or more air interface standards, communication protocols, modulation types, access types, and/or the like. Signals sent and received by the processor 20 may include signaling information in accordance with an air interface standard of an applicable cellular system, and/or any number of different wireline or wireless networking techniques, comprising but not limited to Wi-Fi, wireless local access network (WLAN) techniques, such as Institute of Electrical and Electronics Engineers (IEEE)

802.11, 802.16, 802.3, ADSL, DOCSIS, and/or the like. In addition, these signals may include speech data, user generated data, user requested data, and/or the like.

For example, the apparatus 10 and/or a cellular modem therein may be capable of operating in accordance with various first generation (1G) communication protocols, second generation (2G or 2.5G) communication protocols, third-generation (3G) communication protocols, fourth-generation (4G) communication protocols, fifth-generation (5G) communication protocols, Internet Protocol Multimedia Subsystem (IMS) communication protocols (for example, session initiation protocol (SIP) and/or the like. For example, the apparatus 10 may be capable of operating in accordance with 2G wireless communication protocols IS-136, Time Division Multiple Access TDMA, Global System for Mobile communications, GSM, IS-95, Code Division Multiple Access, CDMA, and/or the like. In addition, for example, the apparatus 10 may be capable of operating in accordance with 2.5G wireless communication protocols General Packet Radio Service (GPRS), Enhanced Data GSM Environment (EDGE), and/or the like. Further, for example, the apparatus 10 may be capable of operating in accordance with 3G wireless communication protocols, such as Universal Mobile Telecommunications System (UMTS), Code Division Multiple Access 2000 (CDMA2000), Wideband Code Division Multiple Access (WCDMA), Time Division-Synchronous Code Division Multiple Access (TD-SCDMA), and/or the like. The apparatus 10 may be additionally capable of operating in accordance with 3.9G wireless communication protocols, such as Long Term Evolution (LTE), Evolved Universal Terrestrial Radio Access Network (E-UTRAN), and/or the like. Additionally, for example, the apparatus 10 may be capable of operating in accordance with 4G wireless communication protocols, such as LTE Advanced, 5G, and/or the like as well as similar wireless communication protocols that may be subsequently developed.

It is understood that the processor 20 may include circuitry for implementing audio/video and logic functions of apparatus 10. For example, the processor 20 may comprise a digital signal processor device, a microprocessor device, an analog-to-digital converter, a digital-to-analog converter, and/or the like. Control and signal processing functions of the apparatus 10 may be allocated between these devices according to their respective capabilities. The processor 20 may additionally comprise an internal voice coder (VC) 20a, an internal data modem (DM) 20b, and/or the like. Further, the processor 20 may include functionality to operate one or more software programs, which may be stored in memory. In general, processor 20 and stored software instructions may be configured to cause apparatus 10 to perform actions. For example, processor 20 may be capable of operating a connectivity program, such as a web browser. The connectivity program may allow the apparatus 10 to transmit and receive web content, such as location-based content, according to a protocol, such as wireless application protocol, WAP, hypertext transfer protocol, HTTP, and/or the like.

Apparatus 10 may also comprise a user interface including, for example, an earphone or speaker 24, a ringer 22, a microphone 26, a display 28, a user input interface, and/or the like, which may be operationally coupled to the processor 20. The display 28 may, as noted above, include a touch sensitive display, where a user may touch and/or gesture to make selections, enter values, and/or the like. The processor 20 may also include user interface circuitry configured to control at least some functions of one or more elements of the user interface, such as the speaker 24, the ringer 22, the microphone 26, the display 28, and/or the like. The processor 20 and/or user interface circuitry comprising the processor 20 may be configured to control one or more functions of one or more elements of the user interface through computer program instructions, for example, software and/or firmware, stored on a memory accessible to the processor 20, for example, volatile memory 40, non-volatile memory 42, and/or the like. The apparatus 10 may include a battery for powering various circuits related to the mobile terminal, for example, a circuit to provide mechanical vibration as a detectable output. The user input interface may comprise devices allowing the apparatus 20 to receive data, such as a keypad 30 (which can be a virtual keyboard presented on display 28 or an externally coupled keyboard) and/or other input devices.

As shown in FIG. 8, apparatus 10 may also include one or more mechanisms for sharing and/or obtaining data. For example, the apparatus 10 may include a short-range radio frequency (RF) transceiver and/or interrogator 64, so data may be shared with and/or obtained from electronic devices in accordance with RF techniques. The apparatus 10 may include other short-range transceivers, such as an infrared (IR) transceiver 66, a BluetoothTM (BT) transceiver 68 operating using BluetoothTM wireless technology, a wireless universal serial bus (USB) transceiver 70, a BluetoothTM Low Energy transceiver, a ZigBee transceiver, an ANT transceiver, a cellular device-to-device transceiver, a wireless local area link transceiver, and/or any other short-range radio technology. Apparatus 10 and, in particular, the short-range transceiver may be capable of transmitting data to and/or receiving data from electronic devices within the proximity of the apparatus, such as within 10 meters, for example. The apparatus 10 including the Wi-Fi or wireless local area networking modem may also be capable of transmitting and/or receiving data from electronic devices according to various wireless networking techniques, including 6LoWpan, Wi-Fi, Wi-Fi low power, WLAN techniques such as IEEE 802.11 techniques, IEEE 802.15 techniques, IEEE 802.16 techniques, and/or the like.

The apparatus 10 may comprise memory, such as a subscriber identity module (SIM) 38, a removable user identity module (R-UIM), an eUICC, an UICC, and/or the like, which may store information elements related to a mobile subscriber. In addition to the SIM, the apparatus 10 may include other removable and/or fixed memory. The apparatus 10 may include volatile memory 40 and/or non-volatile memory 42. For example, volatile memory 40 may include

Random Access Memory (RAM) including dynamic and/or static RAM, on-chip or off-chip cache memory, and/or the like. Non-volatile memory 42, which may be embedded and/or removable, may include, for example, read-only memory, flash memory, magnetic storage devices, for example, hard disks, magnetic tape, optical disc drives and/or media, non-volatile random access memory (NVRAM), and/or the like. Like volatile memory 40, non-volatile memory 42 may include a cache area for temporary storage of data. At least part of the volatile and/or non-volatile memory may be embedded in processor 20. The memories may store one or more software programs, instructions, pieces of information, data, and/or the like which may be used by the apparatus for performing operations disclosed herein including, for example, process 200 and/or the like.

The memories may comprise an identifier, such as an international mobile equipment identification (IMEI) code, capable of uniquely identifying apparatus 10. The memories may comprise an identifier, such as an international mobile equipment identification (IMEI) code, capable of uniquely identifying apparatus 10. In the example embodiment, the processor 20 may be configured using computer code stored at memory 40 and/or 42 to control and/or provide one or more aspects disclosed herein (see, for example, process 200 and/or other operations/functions disclosed herein).

Some of the embodiments disclosed herein may be implemented in software, hardware, application logic, or a combination of software, hardware, and application logic. The software, application logic, and/or hardware may reside on memory 40, the control apparatus 20, or electronic components, for example. In some example embodiment, the application logic, software or an instruction set is maintained on any one of various conventional computer-readable media. In the context of this document, a "computer-readable medium" may be any non-transitory media that can contain, store, communicate, propagate or transport the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer or data processor circuitry, with examples depicted at FIG. 8, computer-readable medium may comprise a non-transitory computer-readable storage medium that may be any media that can contain or store the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer.

Without in any way limiting the scope, interpretation, or application of the claims appearing below, a technical effect of one or more of the example embodiments disclosed herein may be enhanced flexibility in instantiating network functions in a network.

The subject matter described herein may be embodied in systems, apparatus, methods, and/or articles depending on the desired configuration. For example, the base stations and user equipment (or one or more components therein) and/or the processes described herein can be implemented using one or more of the following: a processor executing program code, an application-specific integrated circuit (ASIC), a digital signal processor (DSP), an embedded processor, a field programmable gate array (FPGA), and/or combinations thereof. These various implementations may include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device. These computer programs (also known as programs, software, software applications, applications, components, program code, or code) include machine instructions for a programmable processor, and may be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the term “computer-readable medium” refers to any computer program product, machine-readable medium, computer-readable storage medium, apparatus and/or device (for example, magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions. Similarly, systems are also described herein that may include a processor and a memory coupled to the processor. The memory may include one or more programs that cause the processor to perform one or more of the operations described herein.

Although a few variations have been described in detail above, other modifications or additions are possible. In particular, further features and/or variations may be provided in addition to those set forth herein. Moreover, the implementations described above may be directed to various combinations and subcombinations of the disclosed features and/or combinations and subcombinations of several further features disclosed above. Other embodiments may be within the scope of the following claims.

If desired, the different functions discussed herein may be performed in a different order and/or concurrently with each other. Furthermore, if desired, one or more of the above-described functions may be optional or may be combined. Although various aspects of some of the embodiments are set out in the independent claims, other aspects of some of the embodiments comprise other combinations of features from the described embodiments and/or the dependent claims with the features of the independent claims, and not solely the combinations explicitly set out in the claims. It is also noted herein that while the above describes example embodiments, these descriptions should not be viewed in a limiting sense. Rather, there are several variations and modifications that may be made without departing from the scope of some of the embodiments as defined in the appended claims. Other embodiments may be within the scope of the following claims. The term “based on” includes “based on at least.” The use of the phase “such as” means “such as for example” unless otherwise indicated.

CLAIMS

1. A method comprising:

selecting, for a network node, a network function instance from a group of network function instances having a common network function, the selection based on an identifier indicating that the network node and the selected network function instance are part of a compatible superset of different types of network functions; and

accessing, by the network node, the selected network function instance.

2. The method of claim 1, wherein the identifier comprises a superset identifier indicating compatibility among the different types of network functions and with respect to at least one attribute.

3. The method of claim 2, wherein the at least one attribute comprises connectivity, location, manufacturer, performance, capacity, optional features, and/or supporting a same network slice.

4. The method of any of claims 1-3, wherein the selecting is further based on a group identifier indicating compatible instances of a network function of a certain type.

5. The method of any of claims 1-4, wherein the selected network function instance is configured to provide at least an access management function, a session management function, a policy control function, a user plane function, a unified data management function, an authentication server function, a network exposure function, a network resource function, and/or a network slice selection function.

6. The method of any claims 1-5, wherein the identifier is obtained from one of a registry or a management system.

7. The method of any of claims 1-6, wherein the network function instance comprises a virtualized network function instance, and wherein the group comprises virtualized network function instances.

8. The method of claim 7, wherein the virtualized network function instance is virtualized in a host computer and/or a virtual machine, and wherein the network node comprises a consuming network function instance.

9. A method comprising:

selecting, for a consuming network service instance, a producing network service instance from a group of producing network service instances offering a common network service, the selecting based on an identifier indicating that the consuming network service instance and producing network service instance are part of a compatible superset of different types of network service instances; and

accessing, by the consuming network service instance, the selected network service instance.

10. The method of claim 9, wherein the identifier comprises a superset identifier indicating compatibility among the different types of network services and with respect to at least one attribute.

11. The method of claim 10, wherein the at least one attribute comprises connectivity, location, manufacturer, performance, capacity, optional features, and/or supporting a same network slice.

12. The method of claim 9, wherein the selecting is further based on a group identifier indicating compatible instances of a network function of a certain type.

13. The method of any claims 9-12, wherein the identifier is obtained from one of a registry or a management system.

14. The method of any of claims 9-13, wherein the selected producing network service instance is configured to provide at least an access management function service, a session management function service, a policy control function service, a user plane function service, a unified data management function service, an authentication server function service, a network exposure function service, a network resource function service and/or a network slice selection function service.

15. An apparatus comprising:

at least one processor; and

at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to at least:

select a network function instance from a group of network function instances having a common network function, the selection based on an identifier indicating that the network node and the selected network function instance are part of a compatible superset of different types of network functions; and access the selected network function instance.

16. The apparatus of claim 15, wherein the identifier comprises a superset identifier indicating compatibility among the different types of network functions and with respect to at least one attribute.

17. The apparatus of claim 16, wherein the at least one attribute comprises connectivity, location, manufacturer, performance, capacity, optional features, and/or supporting a same network slice.

18. The apparatus of any of claims 15-17, wherein the selection is further based on a group identifier indicating compatible instances of a network function of a certain type.

19. The apparatus of any of claims 15-18, wherein the selected network function instance is configured to provide at least an access management function, a session management function, a policy control function, a user plane function, a unified data management function, an authentication server function, a network exposure function, a network resource function, and/or a network slice selection function.

20. The apparatus of any claims 15-19, wherein the identifier is obtained from one of a registry or a management system.

21. The apparatus of any of claims 15-20, wherein the network function instance comprises a virtualized network function instance, and wherein the group comprises virtualized network function instances.

22. The apparatus of claim 21, wherein the virtualized network function instance is virtualized in a host computer and/or a virtual machine, and wherein the apparatus comprises a network node and/or a consuming network function instance.

23. An apparatus comprising:

at least one processor; and

at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to at least:

select a producing network service instance from a group of producing network service instances offering a common network service, the selection based on an identifier indicating that a consuming network service instance and producing network service instance are part of a compatible superset of different types of network service instances; and

access the selected network service instance.

24. The apparatus of claim 23, wherein the identifier comprises a superset identifier indicating compatibility among the different types of network services and with respect to at least one attribute.

25. The apparatus of claim 24, wherein the at least one attribute comprises connectivity, location, manufacturer, performance, capacity, optional features, and/or supporting a same network slice.

26. The apparatus of claim 23, wherein the selection is further based on a group identifier indicating compatible instances of a network function of a certain type.

27. The apparatus of any claims 23-26, wherein the identifier is obtained from one of a registry or a management system.

28. The apparatus of any of claims 9-13, wherein the selected producing network service instance is configured to provide at least an access management function service, a session management function service, a policy control function service, a user plane function service, a unified data management function service, an authentication server function service, a network exposure function service, a network resource function service and/or a network slice selection function service.

29. An apparatus comprising:

means for selecting, for a network node, a network function instance from a group of network function instances having a common network function, the selection based on an identifier indicating that the network node and the selected network function instance are part of a compatible superset of different types of network functions; and

means for accessing, by the network node, the selected network function instance.

30. The apparatus of claim 29 further comprising means for performing a method of any of claims 2-8.

31. An apparatus comprising:

means for selecting, for a consuming network service instance, a producing network service instance from a group of producing network service instances offering a common network service, the selecting based on an identifier indicating that the consuming network service instance and producing network service instance are part of a compatible superset of different types of network service instances; and

means for accessing, by the consuming network service instance, the selected network service instance.

32. The apparatus of claim 31 further comprising means for performing a method of any of claims 10-14.

33. A non-transitory computer-readable storage medium including program code which when executed by at least one processor causes operations comprising:

selecting, for a network node, a network function instance from a group of network function instances having a common network function, the selection based on an identifier indicating that the network node and the selected network function instance are part of a compatible superset of different types of network functions; and

accessing, by the network node, the selected network function instance.

34. A non-transitory computer-readable storage medium including program code which when executed by at least one processor causes operations comprising:

selecting, for a consuming network service instance, a producing network service instance from a group of producing network service instances offering a common network service, the selecting based on an identifier indicating that the consuming network service instance and producing network service instance are part of a compatible superset of different types of network service instances; and

accessing, by the consuming network service instance, the selected network service instance.

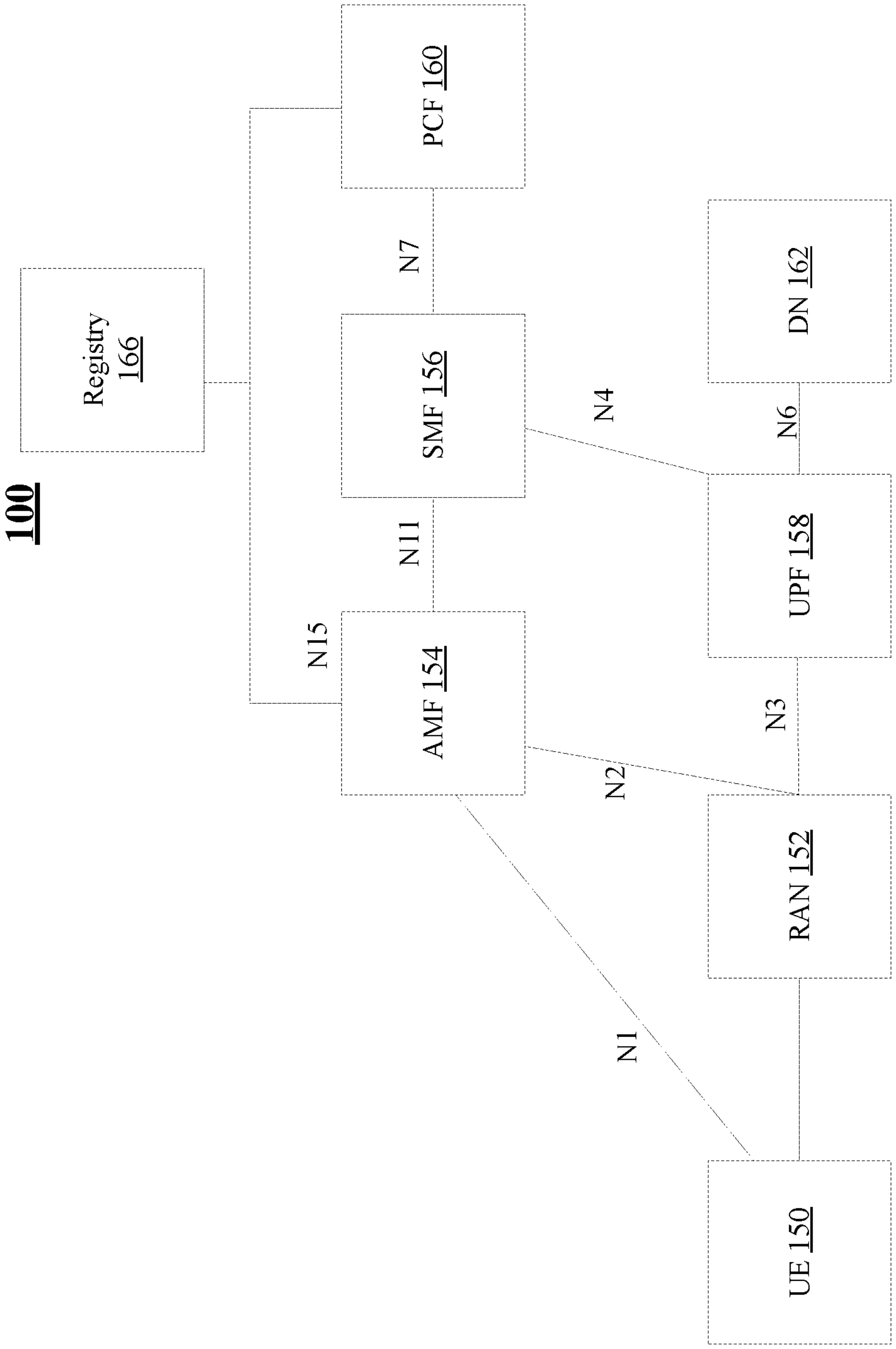

100
Registry 166
PCF 160
N7
SMF 156
N11
AMF 154
N15
N4
N2
N1
DN 162
N6
UPF 158
N3
RAN 152
UE 150

FIG. 1A

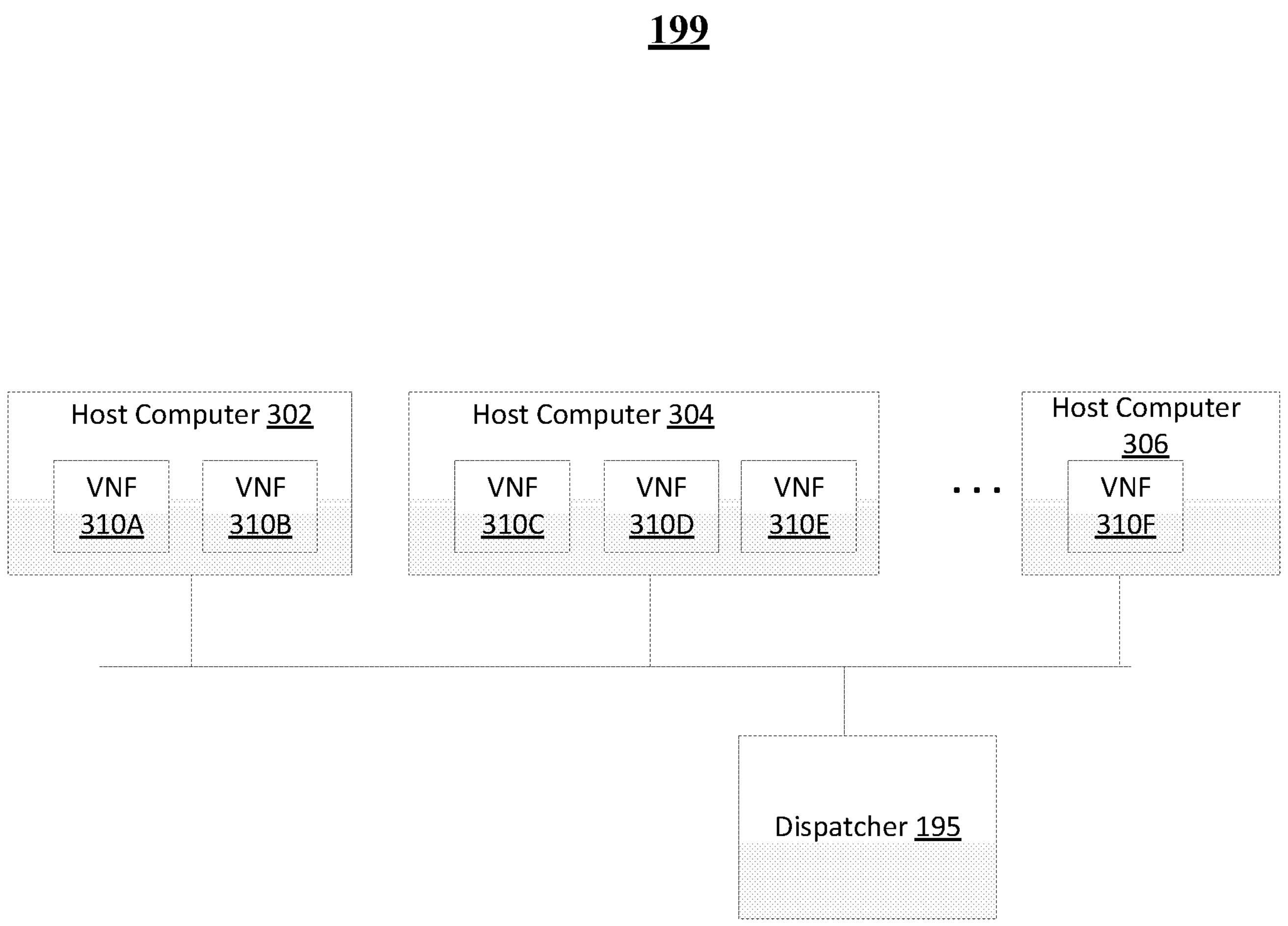

199
Host Computer 302
VNF 310A
VNF 310B
Host Computer 304
VNF 310C
VNF 310D
VNF 310E
. . .
Host Computer 306
VNF 310F
Dispatcher 195

FIG. 1B

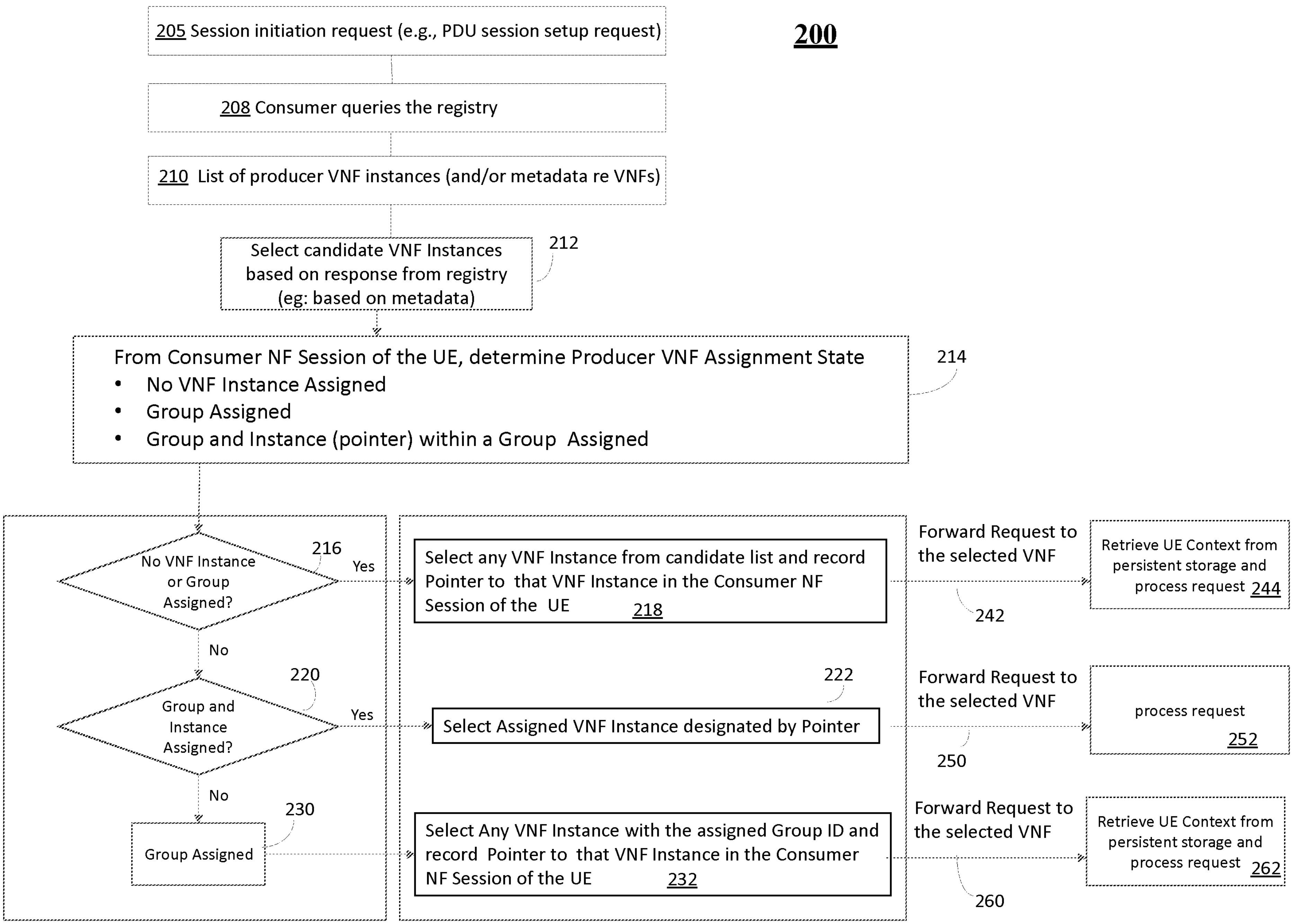

200
205 Session initiation request (e.g., PDU session setup request)
208 Consumer queries the registry
210 List of producer VNF instances (and/or metadata re VNFs)
Select candidate VNF Instances based on response from registry (eg: based on metadata)
212
From Consumer NF Session of the UE, determine Producer VNF Assignment State
• No VNF Instance Assigned
• Group Assigned
• Group and Instance (pointer) within a Group Assigned
214
No VNF Instance or Group Assigned?
216
Yes
No
Group and Instance Assigned?
220
Yes
No
Group Assigned
230
Select any VNF Instance from candidate list and record Pointer to that VNF Instance in the Consumer NF Session of the UE 218
222
Select Assigned VNF Instance designated by Pointer
Select Any VNF Instance with the assigned Group ID and record Pointer to that VNF Instance in the Consumer NF Session of the UE 232
Forward Request to the selected VNF
242
Forward Request to the selected VNF
250
Forward Request to the selected VNF
260
Retrieve UE Context from persistent storage and process request 244
process request 252
Retrieve UE Context from persistent storage and process request 262

FIG. 2

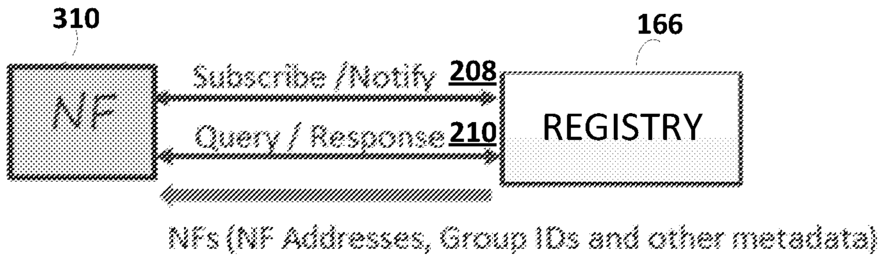

310
NF
Subscribe /Notify 208
Query / Response 210
166
REGISTRY
NFs (NF Addresses, Group IDs and other metadata)

FIG. 3

400

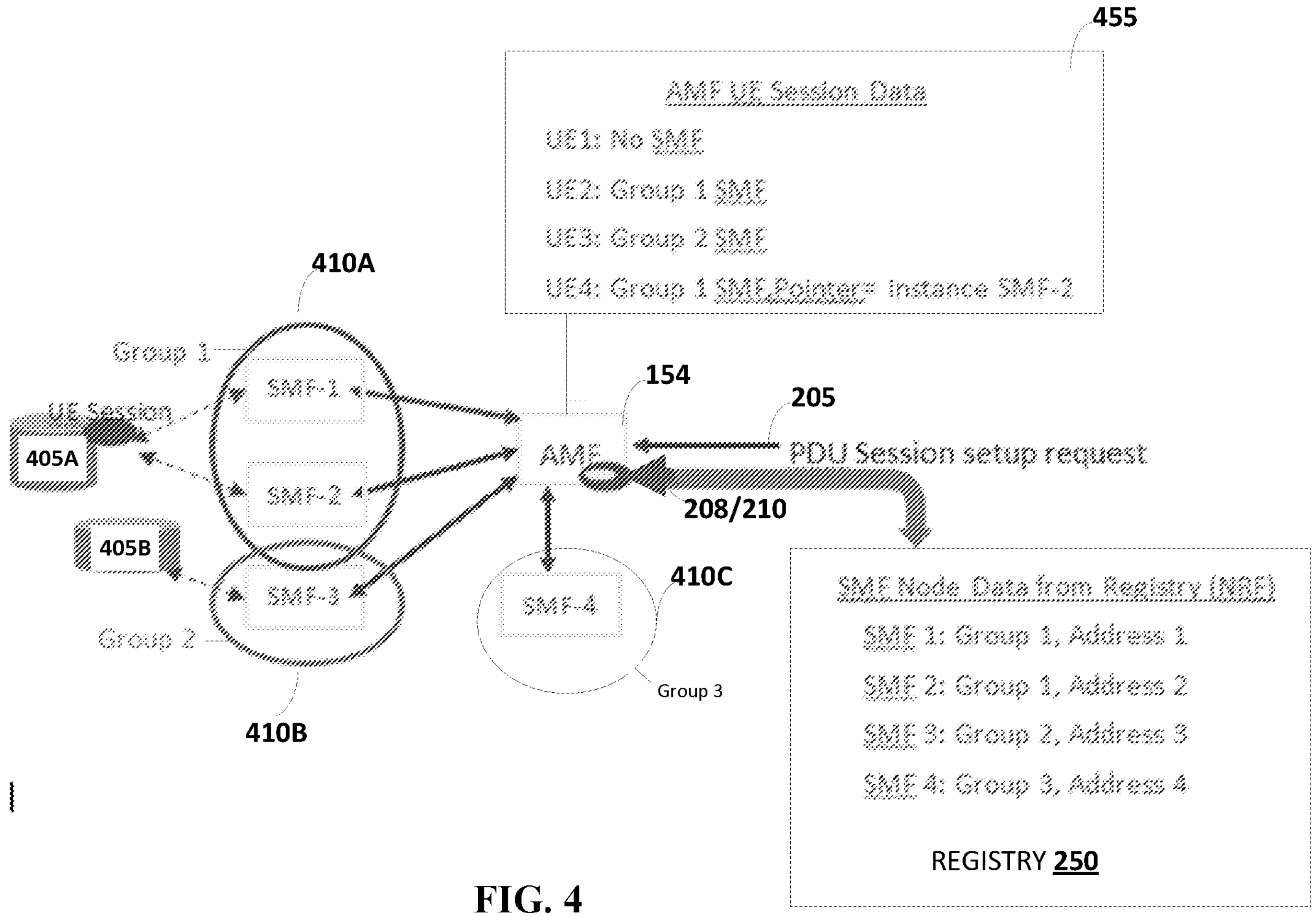
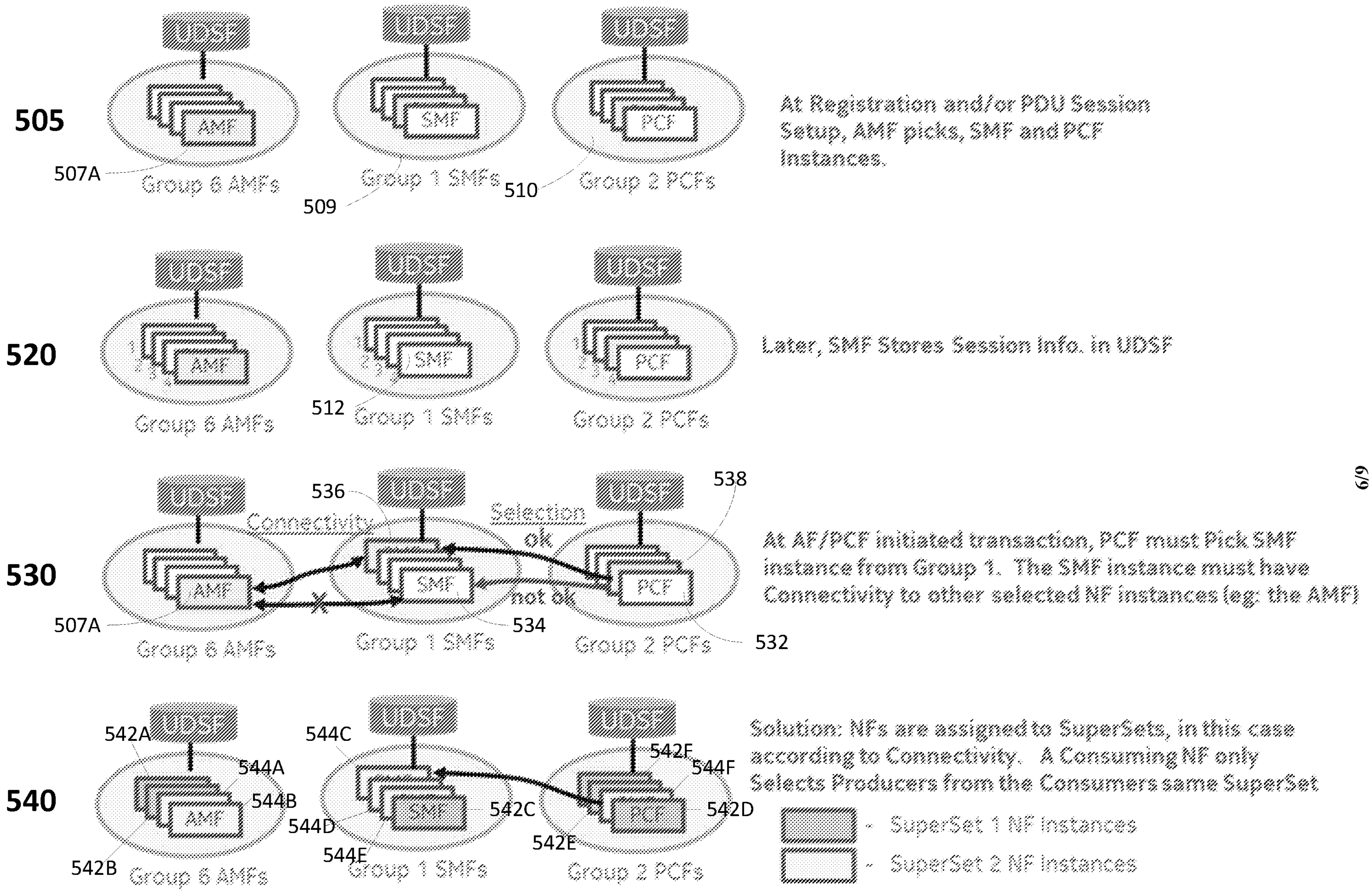

455
AMF UE Session Data
UE1: No SMF
UE2: Group 1 SMF
UE3: Group 2 SMF
UE4: Group 1 SMF Pointer= instance SMF-2
410A
Group 1
SMF-1
UE Session
405A
SMF-2
405B
SMF-3
Group 2
410B
154
AMF
205
PDU Session setup request
208/210
SMF-4
410C
Group 3
SMF Node Data from Registry (NRF)
SMF 1: Group 1, Address 1
SMF 2: Group 1, Address 2
SMF 3: Group 2, Address 3
SMF 4: Group 3, Address 4
REGISTRY 250

FIG. 4



At Registration and/or PDU Session Setup, AMF picks, SMF and PCF Instances.

Later, SMF Stores Session Info. in UDSF

At AF/PCF initiated transaction, PCF must Pick SMF instance from Group 1. The SMF instance must have Connectivity to other selected NF instances (eg: the AMF)

Solution: NFs are assigned to SuperSets, in this case according to Connectivity. A Consuming NF only Selects Producers from the Consumers same SuperSet

▨ - SuperSet 1 NF instances

□ - SuperSet 2 NF instances

FIG. 5

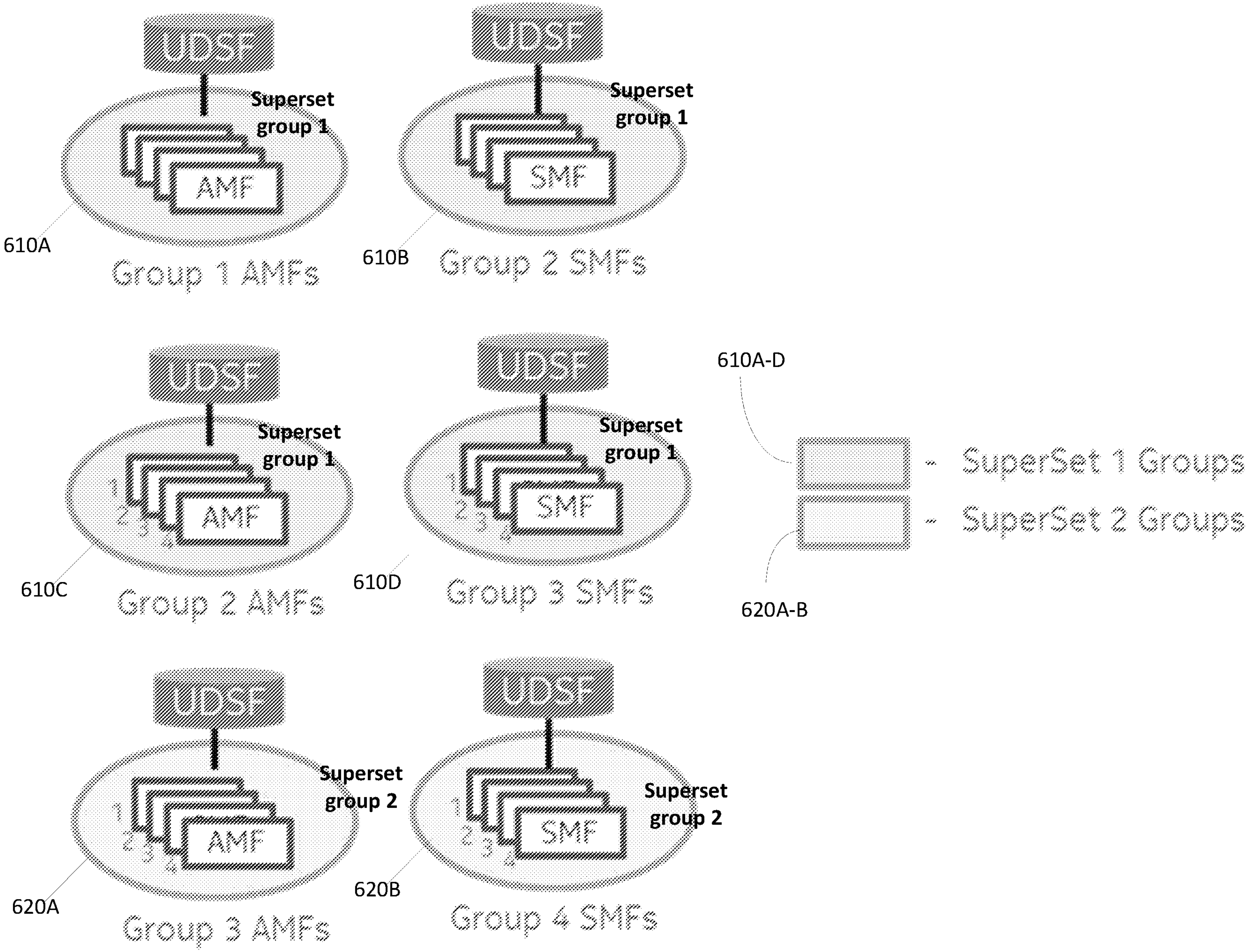

UDSF
Superset group 1
AMF
610A
Group 1 AMFs
UDSF
Superset group 1
SMF
610B
Group 2 SMFs
UDSF
Superset group 1
1
2
3
4
AMF
610C
Group 2 AMFs
UDSF
Superset group 1
1
2
3
4
SMF
610D
Group 3 SMFs
610A-D
~ SuperSet 1 Groups
~ SuperSet 2 Groups
620A-B
UDSF
Superset group 2
1
2
3
4
AMF
620A
Group 3 AMFs
UDSF
Superset group 2
1
2
3
4
SMF
620B
Group 4 SMFs

FIG. 6

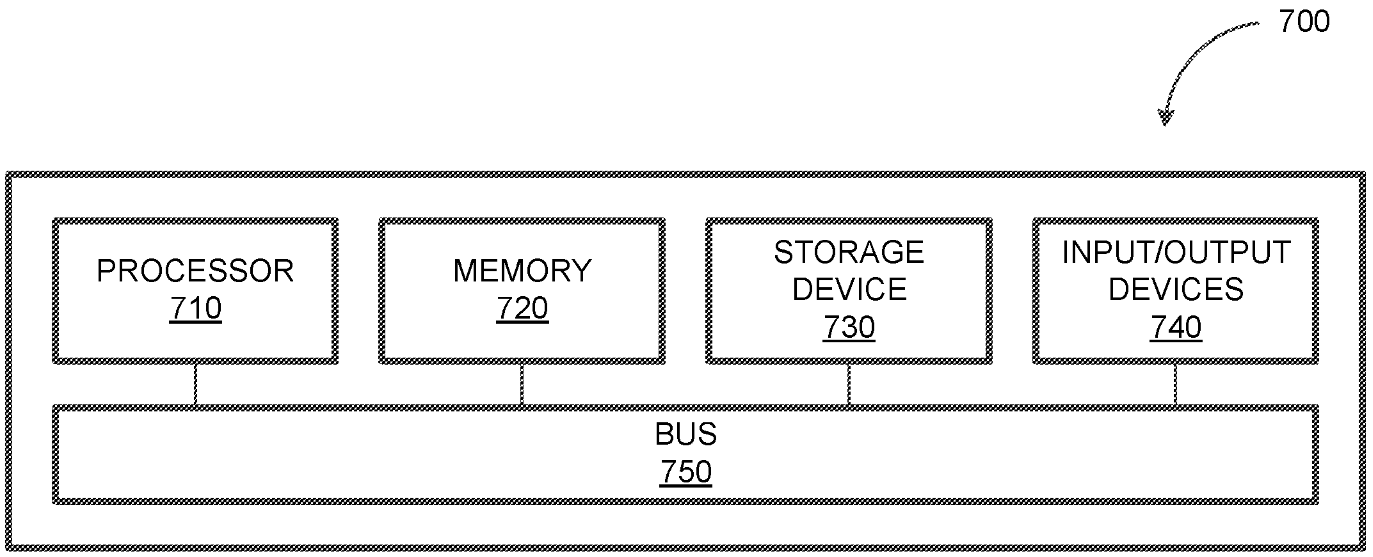

700
PROCESSOR
710
MEMORY
720
STORAGE
DEVICE
730
INPUT/OUTPUT
DEVICES
740
BUS
750

FIG. 7

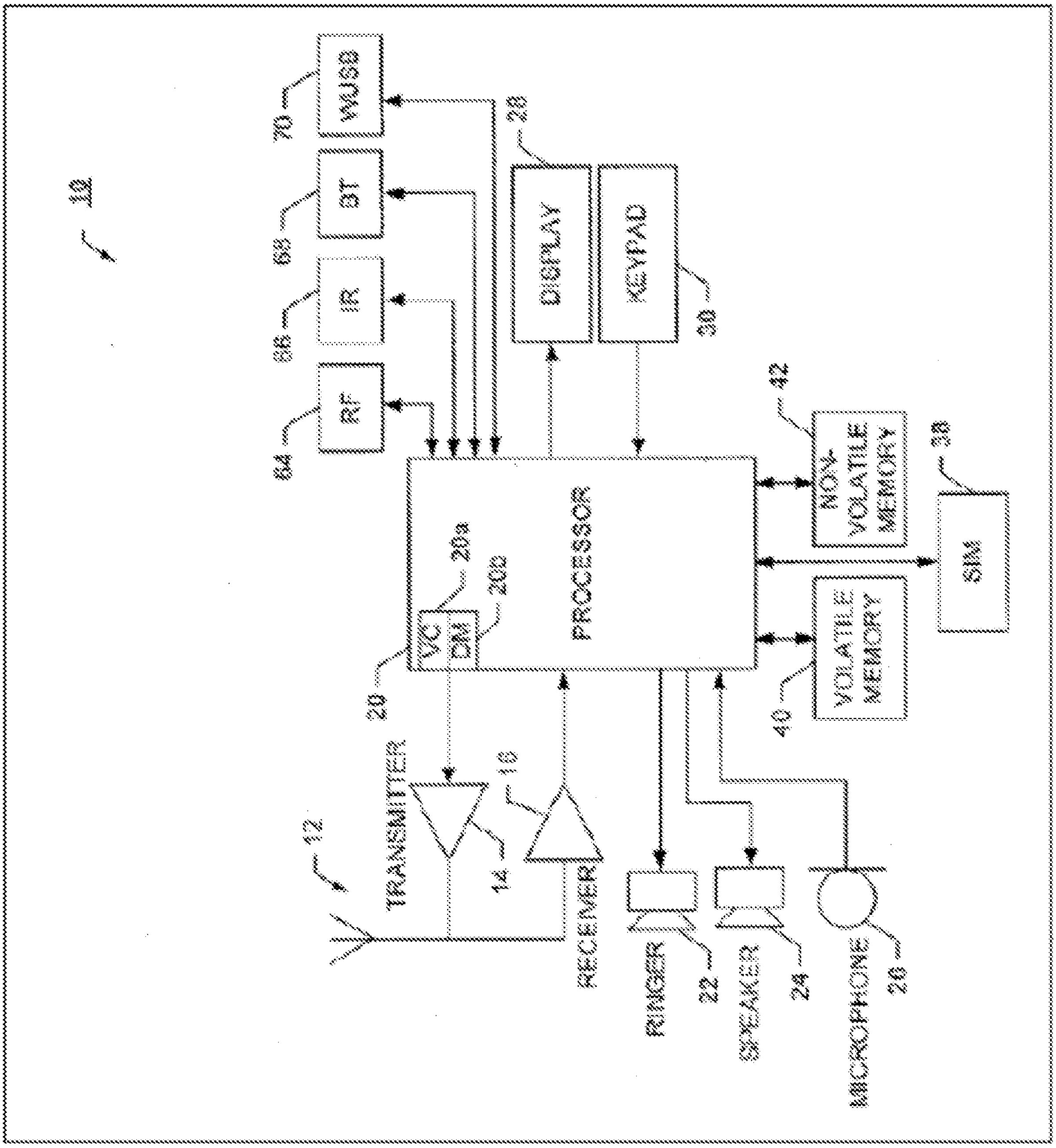

10
12
TRANSMITTER
RECEIVER
14
16
20
VC
DM
20a
20b
PROCESSOR
RINGER
22
SPEAKER
24
MICROPHONE
26
DISPLAY
28
KEYPAD
30
NON-VOLATILE MEMORY
42
SIM
38
VOLATILE MEMORY
40
RF
64
IR
66
BT
68
WUSB
70

FIG. 8

A. **CLASSIFICATION OF SUBJECT MATTER**

See extra sheet

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)

IPC: H04L, H04W, G06F

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

FI, SE, NO, DK

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

EPODOC, EPO-Internal full-text databases, Full-text translation databases from Asian languages, WPIAP, XP3GPP, XPAIP, XPCPVO, XPESP, XPETSI, XPI3E, XPIEE, XPIETF, XPIOP, XPIPCOM, XPJPEG, XPMISC, XPOAC, XPRD, XPTK, COMPDX, INSPEC, TDB, NPL, PRH-Internal, Internet, ACM, Springer

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2018093168 A1 (LG ELECTRONICS INC [KR]) 24 May 2018 (24.05.2018) Figs. Figs. 7-8, 11, 16, 18 & abstract [online] EPOQUENET EPODOC & machine translation of KR 20190082897 A, which belongs to the same family as WO 2018093168 A1, into English by the TXPMTKEA/EPO [online] [retrieved on 2019-08-15] the whole document, in particular paragraphs [0223]-[0229], [0262], [0313], [0316]-[0318], [0324], [0337], [0661]-[0662]	1-34
X	US 2018139107 A1 (SENARATH NIMAL GAMINI [CA] et al.) 17 May 2018 (17.05.2018) the whole document, in particular Figs. 2, 5-6, 11; paragraphs [0007]-[0012], [0029], [0060], [0068]-[0075]	1, 9, 15, 23, 29, 31, 33-34

☒ 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 16 August 2019 (16.08.2019)	Date of mailing of the international search report 20 August 2019 (20.08.2019)
Name and mailing address of the ISA/FI Finnish Patent and Registration Office FI-00091 PRH, FINLAND Facsimile No. +358 29 509 5328	Authorized officer Yrjö Raivio Telephone No. +358 29 509 5000

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2018067780 A1 (CONVIDA WIRELESS LLC [US]) 12 April 2018 (12.04.2018) the whole document, in particular Figs. 6, 10-11, 16; paragraphs [0057], [0081]-[0087], [0186], [0215]-[0219]	1, 9, 15, 23, 29, 31, 33-34
X	US 2017303259 A1 (LEE SEUNG-IK [KR] et al.) 19 October 2017 (19.10.2017) the whole document, in particular Figs. 1-4; paragraphs [0054], [0074], [0084]-[0089], [0094], [0099]	1, 9, 15, 23, 29, 31, 33-34
A	HUAWEI, HISILICON. S2-185389. Solutions for MEANS (Mutually Exclusive Access to Network Slices). 28 May - 1 June 2018, Newport Beach, USA: 3GPP TSG-SA WG2 Meeting #127bis, [online], 2018-05-22, [retrieved on 2019-08-15]. Retrieved from <https://www.3gpp.org/ftp/tsg_sa/WG2_Arch/TSGS2_127BIS_Newport_Beach/Docs/S2-185389.zip>, The whole document.	1-34
A	NGMN. 5G End-to-End Architecture Framework v2.0. [online], 2018-02-26, [retrieved on 2019-08-14]. Retrieved from <https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2018/180226_NGMN_E2EArchFramework_v2.0.0.pdf>, The whole document.	1-34
A	EP 3255839 A1 (NEC CORP [JP]) 13 December 2017 (13.12.2017) the whole document, in particular Fig. 7B; paragraphs [0083]-[0085]	1-34

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2019/050395

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
WO 2018093168 A1	24/05/2018	CN 109964509 A KR 20190082897 A	02/07/2019 10/07/2019
US 2018139107 A1	17/05/2018	CN 109952796 A WO 2018086622 A1	28/06/2019 17/05/2018
WO 2018067780 A1	12/04/2018	CN 109952741 A US 2019238425 A1	28/06/2019 01/08/2019
US 2017303259 A1	19/10/2017	US 10142994 B2 KR 20170119296 A US 2019098618 A1	27/11/2018 26/10/2017 28/03/2019
EP 3255839 A1	13/12/2017	WO 2016125554 JP 6451750 B2 US 2018048523 A1 WO 2016125554 A1	09/11/2017 16/01/2019 15/02/2018 11/08/2016

CLASSIFICATION OF SUBJECT MATTER

IPC
H04L 12/24 (2006.01)
H04L 29/08 (2006.01)
H04W 28/08 (2009.01)
H04W 40/32 (2009.01)
H04W 48/18 (2009.01)
H04W 84/04 (2009.01)
G06F 9/455 (2018.01)
H04W 40/20 (2009.01)