



IMT-2020

ITU-T Focus Group
IMT-2020 Deliverables
2017



Foreword

As we approach 2020, one of the most important areas of ITU work will be our international standardization of IMT-2020 (5G) systems. ITU is supporting the development of a 5G environment where we will all have access to highly reliable communications; where trusted ICTs will be core to innovation in every industry sector.

Alongside enhanced mobile broadband and the Internet of Things, 5G will support ultra-reliable and low-latency communications for applications such as automated driving, remote medical surgery, collaborative robotics and advanced virtual reality. At this high end of 5G application, in some cases we will demand end-to-end latencies as low as 1 millisecond.

What becomes evident when looking at the ambitious performance targets of 5G systems, and the wide variety of envisioned 5G applications, is that future networks will need to be agile all-round players able to perform a wide array of specialized functions.

5G will make no compromises when it comes to performance. Every application must be able to perform to its full potential and this will demand significant innovation in network architectures and orchestration techniques. In May 2015, ITU established a [Focus Group on network aspects of IMT-2020](#) to address exactly this challenge.

The Focus Group explored how emerging 5G technologies will interact in future networks, studying network softwarization and slicing, 5G architecture and fixed-mobile convergence, end-to-end network management, information-centric networking, and related open-source innovation.

The Focus Group's work made it clear that network softwarization and slicing – underpinning deeply programmable networks able to be 'sliced' into virtual networks with very specialized capabilities – will be fundamental to the dynamic allocation of network resources in the 5G environment, giving networks the agility required to support the specific requirements of any particular 5G application.

The Focus Group concluded its study in December 2016 with the delivery of five draft ITU standards and four draft ITU technical papers to fuel standardization work led by ITU-T Study Group 13.



The success of the Focus Group was in large part thanks to the expert leadership of its Chairman Peter Ashwood-Smith, Huawei, and Vice-Chairmen Yoshinori Goto, NTT; Luca Pesando, Telecom Italia; Namseok Ko, ETRI; Yachen Wang, China Mobile.

Special thanks are owed to the champions of the Focus Group's four working groups: Namseok Ko, ETRI; Akihiro Nakao, University of Tokyo; Olivia Heeyun Choi and Sangwoo Kang, KT Corporation; and Marc Mosko, PARC.

This compendium of the Focus Group's outputs will be of great assistance to the standardization experts supporting the architecture and orchestration innovations critical to the success of 5G.

Chaesub Lee

Director of the ITU Telecommunication
Standardization Bureau

During its lifetime the FG IMT-2020 FG IMT-2020 convened two workshops

- [Workshop on network softwarization, Turin, Italy, 21 September 2015](#)
- [Workshop and Demo Day: Wireline Technology Enablers for 5G took place at ITU headquarters in Geneva, Switzerland, on 7 December 2016. \(Videos of demonstrations\)](#)

Interview with Focus Group leaders:

- [Chairman Peter Ashwood-Smith](#) (Huawei, Canada)
- [Vice-Chairman Yachen Wang](#) (China Mobile)

This publication contains the outputs as delivered by FG IMT-2020 at its last meeting in December 2016. At the time of publication of this document the outputs had been integrated into the work programme of ITU-T SG13 for further development as ITU-T Recommendations.

Table of contents

Foreword	iii
1 Terms and definitions	1
Terms and definitions for IMT-2020 in ITU-T	2
2 High level network architecture for 5G	7
Requirements of IMT-2020 from network perspective	9
Framework of IMT-2020 network architecture.....	35
3 Network softwarization	63
Application of network softwarization to IMT-2020.....	65
IMT-2020 Network Management Requirements	193
Network Management Framework for IMT-2020.....	217
4 Information centric networking (ICN)	255
Application of information centric networking to IMT-2020	256
5 Fixed and Mobile Convergence	287
Requirements of IMT-2020 Fixed Mobile Convergence.....	289
Unified Network Integrated Cloud for Fixed Mobile Convergence.....	299

Control plane

Data plane

Function

architecture

Function

IMT-2020

Logical res

Management

Network virtual

Software-defined net

User

Virtualized Network

Function Virtual reso

ional
onal entity
source
nt plane
alization
tworking
r plane
k
ource

1.

Terms and definitions

Terms and definitions for IMT-2020 in ITU-T

1 Usage of Terms defined elsewhere

The following terms defined elsewhere have referenced by deliverables of FG IMT-2020.

Control plane [ITU-T Y.2011]: The set of functions that controls the operation of entities in the stratum or layer under consideration, plus the functions required to support this control.

Data plane [ITU-T Y.2011]: The set of functions used to transfer data in the stratum or layer under consideration.

Functional architecture [ITU-T Y.2016]: A set of functional entities used to describe the structure of an NGN. These functional entities are separated by reference points, and thus, they define the distribution of functions. The functional entities can be used to describe a set of reference configurations. These reference configurations identify which reference points are visible at the boundaries of equipment implementations and between administrative domains.

Functional entity [ITU-T Y.2012]: An entity that comprises an indivisible set of specific capabilities. Functional entities are logical concepts, while groupings of functional entities are used to describe practical, physical implementations.

IMT-2020 [ITU-R M.2083-0]: systems, system components, and related aspects that support to provide far more enhanced capabilities than those described in Recommendation ITU-R M.1645.

Logical resource [ITU-T Y.3011]: An independently manageable partition of a physical resource, which inherits the same characteristics as the physical resource and whose capability is bound to the capability of the physical resource.

NOTE – "independently" means mutual exclusiveness among multiple partitions at the same level.

Management plane [ITU-T Y.2011]: The set of functions used to manage entities in the stratum or layer under consideration, plus the functions required to support this management.

Network virtualization [ITU-T Y.3011]: A technology that enables the creation of logically isolated network partitions over shared physical networks so that heterogeneous collection of multiple virtual networks can simultaneously coexist over the shared networks. This includes the aggregation of multiple resources in a provider and appearing as a single resource.

Software-defined networking [ITU-T Y.3300]: A set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner.

User plane [ITU-T Y.2011]: A synonym for data plane.

User plane [ITU-T Y.1714]: Refers to the set of traffic forwarding components through which traffic flows.

NOTE – "User plane" is also referred to as "transport plane" in other ITU-T Recommendations.

Virtualized Network Function [ITU-T Y.3321]: A network function whose functional software is decoupled from hardware, and runs on virtual machine(s).

Virtual resource [ITU-T Y.3011]: An abstraction of physical or logical resource, which may have different characteristics from the physical or logical resource and whose capability may be not bound to the capability of the physical or logical resource.

2 New draft Terms defined by FG IMT-2020 works

The following draft terms and definitions are defined and used by deliverables of FG IMT-2020.

Even if the meeting reached consensus on several important terms and definitions, still much work needs to continue to harmonize and further development of relevant terms and definition.

2.1 Terms and definitions reached consensus

back haul: The network path connecting the base station site and the network controller or gateway site.

front haul: The intra-base station transport, in which a part of the base station function is moved to the remote antenna site.

evolved IMT-advanced RATs: The enhanced version of IMT-advanced RATs and they will be supported by IMT-2020 network.

PDU session: Association between the UE and a data network through IMT-2020 that provides a PDU connectivity service. The type of the association includes IP type, Ethernet type and non-IP type.

Management: The functions or operations related to management of the network functions and resources.

NOTE – Overall coordination and adaptation for configuration and event reporting are achieved between network function infrastructure and network management systems. It includes the collection and forwarding of performance measurements and events. Network function lifecycle management is included with network function instance management. Network management system is authorized to exercise control over and /or collect management information from another system. It is tightly connected with BSS/OSS such that the most efficient and effective way to access, control, deploy, schedule and bind resources is chosen as requested by customers.

Network Function: A processing function in a network. It includes but is not limited to network nodes functionality, e.g. session management, mobility management, switching, routing functions, which has defined functional behaviour and interfaces. Network functions can be implemented as a network node on a dedicated hardware or as a virtualized software functions.

Network slice blueprint: A complete description of the structure, configuration and the plans/work flows for how to instantiate and control the Network Slice Instance during its life cycle.

Network softwarization: An overall transformation trend for designing, implementing, deploying, managing and maintaining network equipment and/or network components by software programming, exploiting the natures of software such as flexibility and rapidity all along the lifecycle of network equipment/components, for the sake of creating conditions enabling the re-design of network and services architectures, optimizing costs and processes, enabling self-management and bringing added values in network infrastructures.

Orchestrator: An entity that fulfils orchestration functions.

NOTE – An entity that manages network service lifecycle and coordinates the management of network service life cycle, network function lifecycle and network function infra resources to ensure optimized allocation of the necessary resources and connectivity. It builds and operates each slice suitable for the service requirements. It is also tightly connected to OSS/BSS for service management.

Network Slice Instance: An activated network slice. It is created based on network slice blueprint (or template).

NOTE – A set of managed run-time network functions, and resources to run these network functions, forming a complete instantiated logical network to meet certain network characteristics required by the service instance(s). It provides the network characteristics which are required by a service instance. A network slice instance may also be shared across multiple service instances provided by the network operator. The network slice instance may be composed by none, one or more sub-network instances, which may be shared by another network slice instance.

Service Instance: An instance of an end-user service or a business service that is realized within or by a network slice. Each service is represented by a service instance. Services and service instances would be provided by the network operator or by third parties.

2.2 Terms and definitions need further harmonization

The following terms need more work to reach consensus in the future ITU-T SG13 meeting.

Network slice: A Network slice is a managed group of infrastructure resources, network functions and services. Network slice is programmable and has the ability to expose its capabilities. The behaviour of the network slice realized via network slice instance(s).

NOTE 1 – Network slice enables the operator to create networks customized to provide flexible solutions for different market scenarios, which have diverse requirements, with respect to the functionality, performance and resource separation.

NOTE 2 – Infrastructure resources include any kind of resources (e.g., physical, logical, virtual resource).

Orchestration: An automated arrangement, coordination of complex network systems and functions including middleware for both physical and virtual infrastructures. It is often discussed as having an inherent intelligence or even implicitly autonomic control. Orchestration results in automation with control network systems.

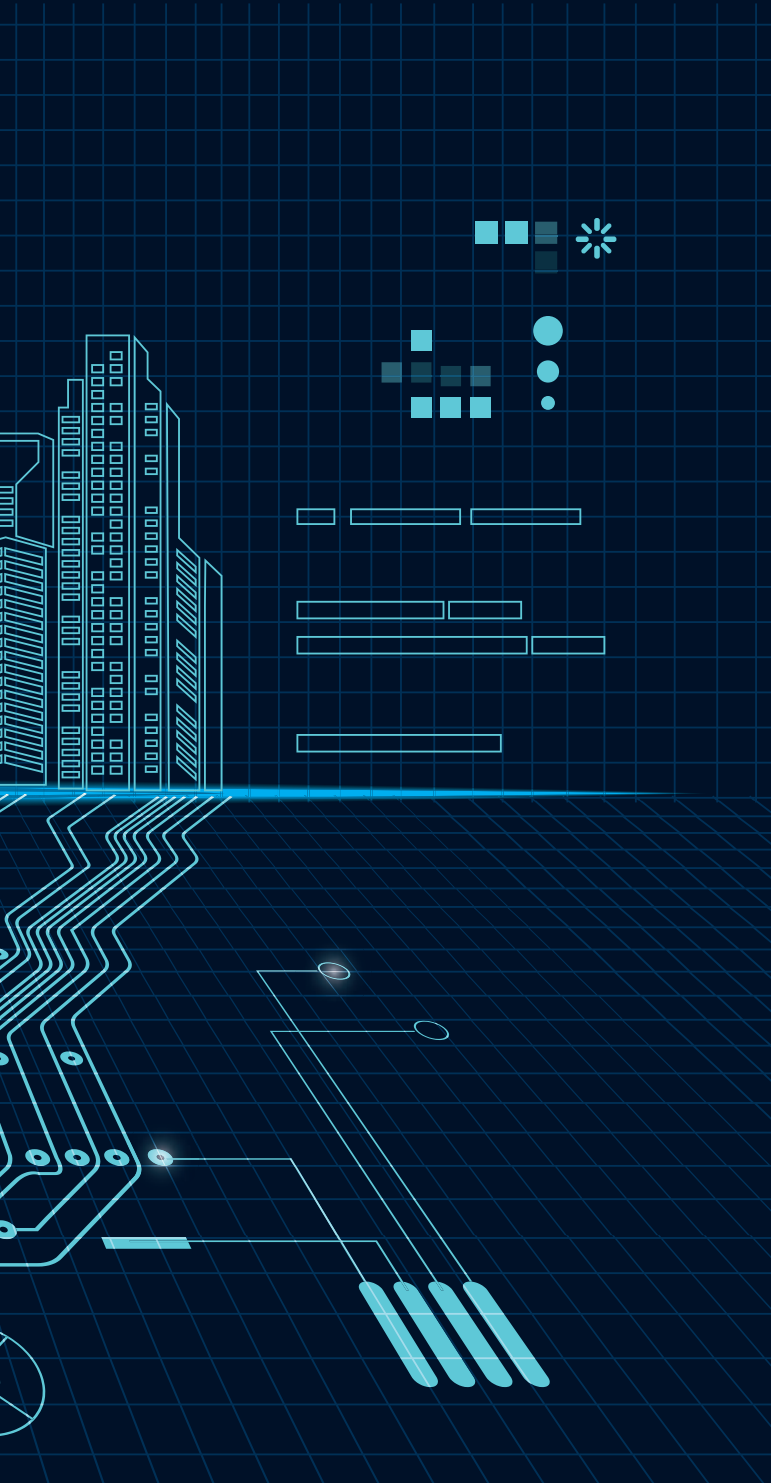
Physical resource: A physical asset for computation, storage and transport (e.g. switch, router, antenna, etc.).

NOTE – Network functions are not regarded as resources.

Slice: As a concept describing system behaviour, slice is a logically isolated set of programmable infrastructure resources (i.e., physical and/or logical resources) to enable functions and services of IMT-2020 network. Slice is created and deleted by order from the orchestrator.







2.

High level network architecture for 5G





work

**Requirements
of IMT-2020 from
network perspective**

ualization

Summary

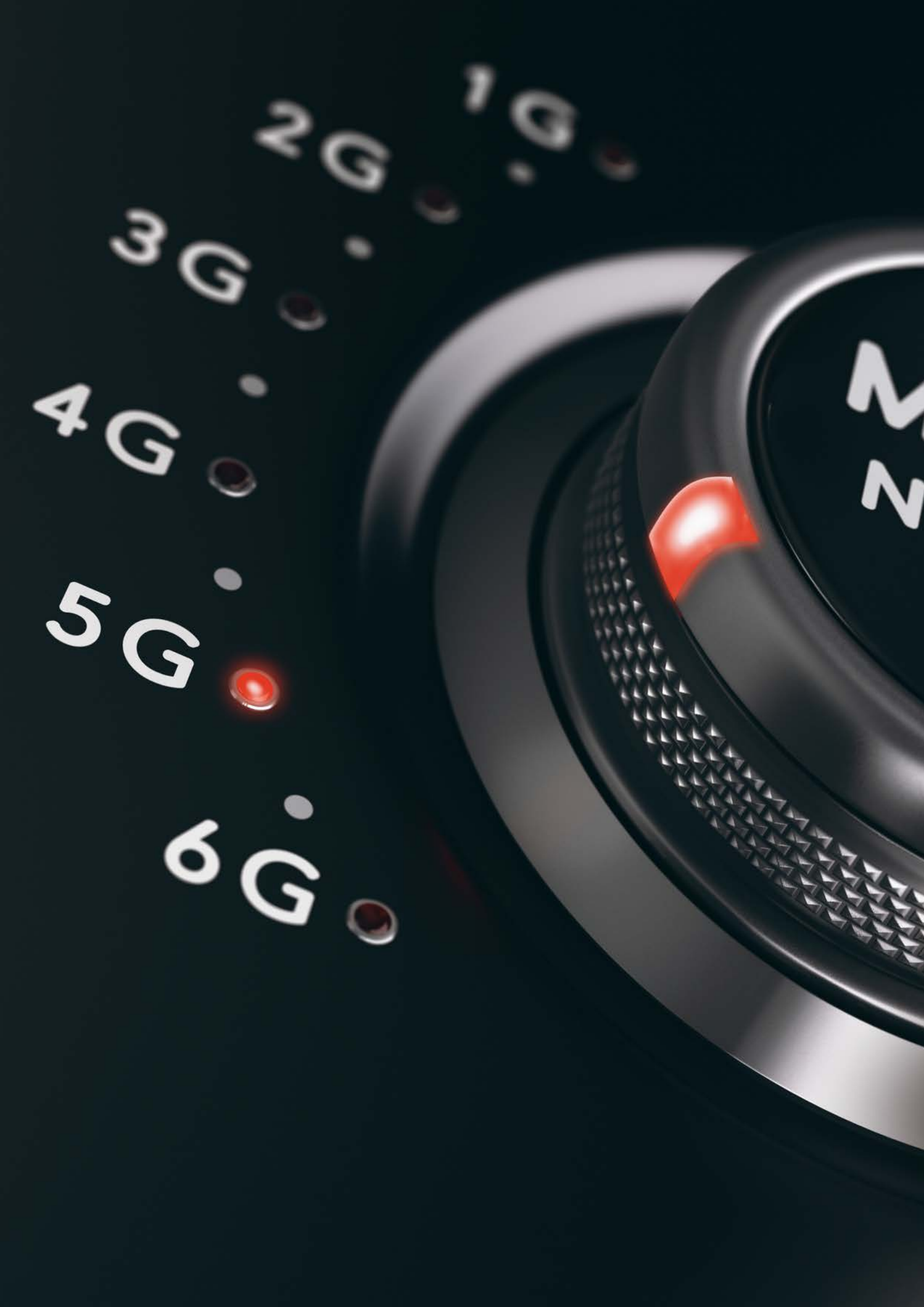
This document describes requirements and capabilities to support emerging services and applications in IMT-2020 from network perspective. Based on the analysis of use cases and business models, design goals as key principles, high level requirements from the view of network operations as well as services for overall non-radio aspects of IMT-2020 networks are specified.

Keywords

5G, IMT-2020, requirements, network slice, orchestration, management

Table of contents

1	Scope
2	References
3	Definitions
3.1	Terms defined elsewhere
3.2	Terms defined in this Recommendation
4	Abbreviations and acronyms
5	Conventions
6	Overview of IMT-2020 from network perspective
7	Design goals
7.1	Service diversity
7.2	Functional flexibility and programmability
7.3	Converged access-agnostic and unified core network
7.4	Separation of control plane and user plane functions
7.5	Distributed network architecture
7.6	In-network data processing
7.7	Unified intelligent network management
7.8	Optimization
7.9	Reliability and security
7.10	Energy efficiency
8	Requirements from service points of view
8.1	Enhanced mobile broadband services
8.2	Enhanced massive machine type communication services
8.3	Ultra-reliable and low latency communication services
9	Requirements from network operation points of view
9.1	Network flexibility and programmability
9.2	Fixed-mobile converged networks
9.3	Enhanced mobility management
9.4	Scalable and incremental installations
9.5	Network capability exposure
9.6	Authentication and Security
9.7	Flexible Signalling
9.8	Numbering, Naming and Addressing
9.9	QoS control
9.10	Context awareness
9.11	Profile management (User, Device, etc.)
9.12	Network management
9.13	Accounting and Charging
9.14	Interworking
Appendix I – Uses cases of IMT-2020	
I-1	Network slicing with shared Core Network instance
Appendix II – Contributors (in Alphabetical Order)	
Appendix III – Acknowledgement	
Bibliography	



N
N

1G

2G

3G

4G

5G

6G

1 Scope

This draft Recommendation describes requirements and capabilities to support emerging services and applications in IMT-2020 from network perspective. Based on the analysis of use cases and business models, design goals as key principles, high level requirements from the view of network operations as well as services for overall non-radio aspects of IMT-2020 networks are specified.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-R M.2083-0] Recommendation ITU-R M.2083-0 (2015), *Framework and overall objectives of the future development of IMT for 2020 and beyond*.

[ITU-T Y.3001] Recommendation ITU-T Y.3001 (2011), *Future networks: Objectives and design goals*.

3 Definitions

<Check in the ITU-T Terms and definitions database on the public website whether the term is already defined in another Recommendation. It may be more consistent to refer to such a definition rather than redefine it>

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 IMT-2020 [ITU-R M.2083-0]: Systems, system components, and related aspects that support to provide far more enhanced capabilities than those described in Recommendation ITU-R M.1645.

3.1.2 logical resource [b-ITU-T Y.3011]: An independently manageable partition of a physical resource, which inherits the same characteristics as the physical resource and whose capability is bound to the capability of the physical resource.

NOTE – "independently" means mutual exclusiveness among multiple partitions at the same level.

3.1.3 network virtualization [b-ITU-T Y.3011]: A technology that enables the creation of logically isolated network partitions over shared physical networks so that heterogeneous collection of multiple virtual networks can simultaneously coexist over the shared networks. This includes the aggregation of multiple resources in a provider and appearing as a single resource.

3.1.4 software-defined networking [b-ITU-T Y.3300]: A set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner.

3.1.5 virtual resource [b-ITU-T Y.3011]: An abstraction of physical or logical resource, which may have different characteristics from the physical or logical resource and whose capability may be not bound to the capability of the physical or logical resource.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 back haul: The network path connecting the base station site and the network controller or gateway site

3.2.2 front haul: The intra-base station transport, in which a part of the base station function is moved to the remote antenna site

3.2.3 evolved IMT-advanced RATs: The enhanced version of IMT-advanced RATs and they will be supported by IMT-2020 network.

3.2.4 network slice: A Network slice is a managed group of infrastructure resources, network functions and services. Network slice is programmable and has the ability to expose its capabilities. The behaviour of the network slice realized via network slice instance(s).

NOTE 1 – Network slice enables the operator to create networks customized to provide flexible solutions for different market scenarios, which have diverse requirements, with respect to the functionality, performance and resource separation.

NOTE 2 – Infrastructure resources include any kind of resources (e.g., physical, logical, virtual resource).

3.2.5 network slice instance: An activated network slice. It is created based on network slice blueprint (or template).

NOTE – A set of managed run-time network functions, and resources to run these network functions, forming a complete instantiated logical network to meet certain network characteristics required by the service instance(s). It provides the network characteristics which are required by a service instance. A network slice instance may also be shared across multiple service instances provided by the network operator. The network slice instance may be composed by none, one or more sub-network instances, which may be shared by another network slice instance.

3.2.6 network softwarization: An overall transformation trend for designing, implementing, deploying, managing and maintaining network equipment and/or network components by software programming, exploiting the natures of software such as flexibility and rapidity all along the lifecycle of network equipment / components, for the sake of creating conditions enabling the re-design of network and services architectures, optimizing costs and processes, enabling self-management and bringing added values in network infrastructures.

3.2.7 orchestration: An automated arrangement, coordination of complex network systems and functions including middleware for both physical and virtual infrastructures. It is often discussed as having an inherent intelligence or even implicitly autonomic control. Orchestration results in automation with control network systems.

3.2.8 orchestrator: An entity that fulfils orchestration functions.

NOTE – An entity that manages network service lifecycle and coordinates the management of network service life cycle, network function lifecycle and network function infra resources to ensure optimized allocation of the necessary resources and connectivity. It builds and operates each slice suitable for the service requirements. It is also tightly connected to OSS/BSS for service management.

3.2.9 physical resource: A physical asset for computation, storage and transport (e.g. switch, router, antenna, etc.).

NOTE – Network functions are not regarded as resources.

3.2.11 slice: As a concept describing system behaviour, slice is a logically isolated set of programmable infrastructure resources (i.e., physical and/or logical resources) to enable functions and services of IMT-2020 network. Slice is created and deleted by order from the orchestrator.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AES	Advanced Encryption Standard
AKA	Authentication and Key Agreement
API	Application Programming Interface
AR	Augmented Reality
CCN	Content Centric Network
CN	Core Network
DPI	Deep Packet Inspection
DSCP	Differentiated Service Code Point
E2E	End-to-End
EPC	Evolved Packet Core
ePDG	evolved Packet Data Gateway
E-UTRAN	Evolved-UMTS Terrestrial Radio Access Network
eSIM	electronic Subscriber Identity Module
ICN	Information Centric Networking
IMT	International Mobile Telecommunications
ISP	Internet Service Provider
KPI	Key Performance Indicator
IMS	IP Multimedia Subsystem
IoT	Internet of Things
LTE	Long Term Evolution
LWA	LTE/WLAN link Aggregation
M2M	Machine-to-Machine
MBH	Mobile Backhaul
MFH	Mobile Fronthaul
MNO	Mobile Network Operator
MTC	Machine Type Communications
MVNO	Mobile Virtual Network Operator
NFV	Network Function Virtualization
OAM	Operation, Administration and Maintenance
QCI	QoS Class Indicator
QoS	Quality of Service
RAT	Radio Access Technologies
SDN	Software Defined Networking
SIM	Subscriber Identity Module
TWAG	Trusted Wireless Access Gateway

UE	User Equipment
UHD	Ultra-High Definition
USIM	Universal Subscriber Identity Module
V2X	Vehicle-to-Everything
VoLTE	Voice over LTE
WLAN	Wireless LAN

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option, and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with this Recommendation.

6 Overview of IMT-2020 from network perspective

According to ITU-R Recommendation M.2083-0, IMT-2020 is described as systems, system components, and related aspects that provide far more enhanced capabilities than those described in Recommendation ITU-R M.1645. IMT-2020 systems will differentiate themselves from fourth generation (4G) systems not only through further evolution in radio performance but also through greatly increased end-to-end flexibility. This end-to-end flexibility will come in large part from the incorporation of softwarization into every component. Well known techniques such as SDN, NFV and cloud computing will together allow unprecedented flexibility in the IMT-2020 system. Such flexibility will enable many new capabilities including network slicing. IMT-2020 is not just an increase in bandwidth from the previous releases of IMT system, but rather a fundamental change to support the following capabilities:

- To support IMT-2020 RATs, evolved IMT-2020 RATs, WLAN, fixed broadband network, satellite, and future radio access networks in access agnostic ways as a single network
- To cope with traffic explosion
- To easily incorporate future emerging services and extend them anywhere in the world
- To provide cost-efficient infrastructure
- To expand the geographic reach of IMT-2020 network

The use cases expected in IMT-2020 are categorized into three representative services: enhanced mobile broadband service, ultra-reliable and low-latency communications, and massive machine type communications. The other services are placed in-between those three service characteristics.

Enhanced mobile broadband services are to allow users to experience high-speed and high-quality multimedia services, e.g., virtual reality, augmented reality, 4K/8K Ultra-High Definition video, and even hologram services, at any time and any place. Ultra-reliable and low-latency communications are to enable delay sensitive and mission critical services such as tactile Internet which requires less than a millisecond end-to-end delay, remote control of medical and industrial robots, and vehicle-to-everything (V2X) communications. The massive machine type communications is to support connections and communications among massive amounts of Internet of Things (IoT) devices.

7 Design goals

Future networks are recommended to fulfil the following objectives [ITU-T Y.3001]:

- Service awareness
- Data awareness
- Environmental awareness
- Social and economic awareness

And in order to realize these objectives, IMT-2020 networks should take into account following design goals as key principles to design IMT-2020 networks.

7.1 Service diversity

The IMT-2020 network should support diversified services accommodating a wide variety of traffic characteristics and behaviours, to support a huge number and wide variety of communication objects, such as user devices, peripheral devices, sensors, and IoT/M2M devices.

7.1.1 Diversity of QoS requirements

The IMT-2020 network should support services that have different end-to-end QoS (data rate, reliability, latency, location accuracy etc.) requirements. There may be some latency critical applications while there are others which are tolerant long end-to-end latency.

7.1.2 Diversity of UE mobility and service continuity

IMT-2020 should support a wide range of mobility and service continuity options. It is expected that mobility requirements for user devices will vary depending on the device and/or application types. Therefore, IMT-2020 should not assume the same mobility support for all devices and application services but rather provide mobility on demand only to those that need it.

The IMT-2020 network should provide mobility that facilitates high-speed and large-scale networks in an environment where a huge number of UE can dynamically move across heterogeneous networks.

7.1.3 Diversity of user data type

IMT-2020 should support the transmission of IP data and non-IP data. In addition, IMT-2020 should support frequent big data and infrequent small data delivery.

7.2 Functional flexibility and programmability

The IMT-2020 network should be flexible, resilient and extensible to cope with various and sometimes even conflicting service requirements adaptably instead of installing a dedicated network for each emerging new service. Thus, the network architecture should support programmable function/service/application allocation and configuration, and dynamic scale-in/-out, etc.

IMT-2020 network architecture should be designed to have a common core which supports on-demand composition of variety of multiple network slices (e.g., converged fixed/mobile network optimized for a particular service).

IMT-2020 network should support virtualization of resources associated with network functions IMT-2020 network should support isolation of any network slice and virtual resource from all others.

Operators of IMT-2020 network should expose network capabilities to 3rd party ISP/ICPs via open APIs to allow agile service creation, flexible and efficient use of the capabilities.

7.3 Converged access-agnostic and unified core network

Up until in the legacy IMT networks, the introduction of a new mobile technology has been accompanied with a new type of packet core network. Therefore, the interworking between the new core network and legacy core networks has always been a technical challenge to overcome.

The IMT-2020 network is envisioned to be an access network-agnostic architecture where core network will be a common unified network for emerging new radio access technologies for IMT-2020 as well as existing fixed and wireless networks (e.g., WLAN). The access technology-agnostic unified core network should be accompanied by common control mechanisms which are decoupled from access technologies.

The IMT-2020 network should support newly-defined RATs for IMT-2020, evolved IMT-advanced RATs, WLAN access networks, fixed broadband access networks, and fixed and mobile satellite networks. IMT-2020 network should support efficient access and management capability for various types of IoT/M2M devices.

7.4 Separation of control plane and user plane functions

The IMT-2020 network should be future-proof as much as possible to accommodate even unforeseeable use cases. The clear separation of control and data planes and the enabling technologies are the basis to make the IMT-2020 network flexible and extensible.

The IMT-2020 network should support highly scalable distributed architecture to avoid signalling congestion and to minimize the signalling overhead for diverse UE/RAT/service requirements.

In order to support distributed network architecture, and optimized routes for application data and signalling data, control plane and user plane functions should be clearly separated with defined interface.

7.5 Distributed network architecture

The IMT-2020 network should be flexible enough to handle the explosive increase of traffic from the new emerging bandwidth-hungry services such as ultra-high definition (UHD) TV, augmented reality (AR), video conferencing, remote medical treatment, etc. The heavily centralized architecture onto an anchor node of existing IMT networks is expected to be changed to cope with the explosion of mobile data traffic. This will require the gateways to the core network are expected to be located closer to the cell sites resulting in distributed network architecture.

The distributed network architecture will bring a significant reduction on backhaul and core network traffic by enabling placing content servers closer to mobile devices and also be beneficial to the latency of the services.

7.6 In-network data processing

The IMT-2020 network should be designed and implemented for optimal and efficient handling of huge amounts of data.

The IMT-2020 network should have mechanisms for promptly retrieving data regardless of their location. In-network data processing is a system that provides with network wide data processing and application services by network nodes.

In IMT-2020, network nodes, where and when required, should provide data processing and application services (i.e., in-network processing), and storage to reduce the network congestion and response time (e.g. for context-aware proximity services, CCNx, on-path big data processing, etc.). ICN and the edge computing are typical examples.

7.7 Unified intelligent network management

The IMT-2020 network should be designed to simplify operations and management of the network with increased complexity due to flexible and extensible network softwarization.

Procedures should be automated as far as possible, with well-defined open interfaces to mitigate multi-vendor interworking problems as well as interoperability (roaming) issues. Standardized management protocol and common OAM protocol are desirable. Also, enhanced end-to-end QoS management and security/privacy models should be designed.

7.8 Optimization

The IMT-2020 network should provide sufficient performance by optimizing network equipment capacity based on service requirement and user demand.

The IMT-2020 network is recommended to provide dynamic data routing mechanisms that respond to changing conditions of network segments.

7.9 Reliability and security

The IMT-2020 network should be designed, operated, and evolved with reliability and resilience, considering congestion and disaster conditions, and to be designed for safety and privacy of their users.

7.10 Energy efficiency

The IMT-2020 network should be designed to reduce UE power consumption and to improve energy efficiency in overall network operation.

In IMT-2020 network, device-level, equipment-level, and network-level technologies should cooperate with each other in achieving a solution for network energy savings.

The architecture of 5G is based on a large number of small cell's densification. One of the major problems with this architecture is the increase in the amount of signalling needed which can reduce the data capacity considerably - it has been suggested by as much as 25%. In addition to this, the base station signalling contributes to the overall energy usage within the system and prevents achievement of the energy reduction KPI's for 5G. In order to address this concept of splitting the control (C) plane from the data (U) plane has been proposed such that the C plane can be delivered via an overlay macro cell. In this architecture, the base station just delivers data on the U plane saving capacity and energy. One proposal is to make the macro cell a satellite cell so that the terrestrial spectrum can be saved. Energy efficiency has been identified as an important metric for 5G. It has been demonstrated that significant reduction in overhead is achieved with the integrated architecture since the U-plane's frame structure can be optimized to suit the local environment.

8 Requirements from service points of view

8.1 Enhanced mobile broadband services

8.1.1 Description

More and more user devices are being equipped with enhanced media consumption capabilities, such as Ultra-High Definition display, multi-view High Definition display, mobile 3D projections, immersive video conferencing, and augmented reality and mixed reality display and interface. This will all lead to a demand for significantly higher data rates in IMT-2020.

The demand for mobile high-definition multimedia also keeps increasing in many areas beyond entertainment, such as medical treatment, safety, and security, which is well reflected to the performance targets for connection density and area traffic capacity in ITU-R Recommendation M.2083-0.

8.1.2 Requirements

[REQ] The network architecture for IMT-2020 is recommended to make IMT-2020 network more flexible and resilient with enhanced capabilities including the upgrades of session or bearer management, more efficient multicast methods, distributed function deployment, network slice, and efficient codecs.

NOTE – Current network architecture is not appropriate to support various bandwidth demands, from ultra-high to extremely low, required for IMT-2020 and beyond.

[REQ] The network architecture for IMT-2020 is recommended to eliminate the needs for separate local mobile specific gateways, additional APN and associated signaling.

NOTE 1 – Local offloading can be used to support efficient traffic distribution. However, local offloading requires a separate local mobile GW in existing IMT networks, which also brings up the needs to support multiple APN connectivity in user devices or APN switching should be initiated, which may result in service disruption of ongoing sessions and operational complexities.

NOTE 2 – Satellite networks can be used to off-load traffic from the terrestrial networks and, in particular, for the video based traffic which is the largest contributor to the spectrum demands. This can be achieved by traffic classification and intelligent routing and will thus reduce the demands on the terrestrial spectrum.

[REQ] The network architecture for IMT-2020 is required to reduce traffic of backhaul and core network.

NOTE – The heavily centralized architecture of existing IMT networks should be changed to cope with the explosion of mobile data traffic. In IMT-2020 networks, therefore, gateways to an IMT-2020 core network can be flexibly located closer to the cell sites, which will bring a significant reduction on backhaul and core network traffic by enabling placing content servers closer to mobile devices and also be beneficial to the latency of the services. The IMT-2020 core networks, therefore, is envisioned to be a distributed network being composed of the multiple distributed gateways. The architectural changes expected from the distributed network, including point-to-point access architecture between a UE and a service network in existing IMT network, should be studied.

8.2 Enhanced massive machine type communication services

8.2.1 Description

In IMT-2020 networks, almost every object that can benefit from being connected is expected to be connected through wired or wireless internet technologies, which will lead to a situation where the number of connected devices exceeds the number of human user devices. These connected “things” can be various ranging from low-complexity devices to highly complex and advanced devices. As more and more things get connected, various services that utilize the connection capabilities of things will appear: smart energy distribution grid system, agriculture, healthcare, vehicle-to-vehicle and vehicle-to-road infrastructure communication.

At least one hundred thousand simultaneous active connections per square kilometre, which will be mostly coming true by the deployment of those massive MTC (machine type communications) devices (e.g., IoT/M2M devices), should be supported in IMT-2020 network. Consistent end-to-end user experience should also be provided even in the presence of that large number of concurrent connections.

8.2.2 Requirements

[REQ] The network architecture for IMT-2020 is required to address architectural issues coming from the massive number of MTC devices.

[REQ] The network architecture for IMT-2020 is required to prevent signalling and user data congestion by massive number of MTC devices.

[REQ] The network architecture for IMT-2020 is required to provide signaling protocols to support various traffic characteristics and communication types of MTC devices (for example, such as short or massive burst traffic, delay sensitive or non-sensitive data, etc.).

[REQ] The network architecture for IMT-2020 is required to simplify the message procedures to prevent signalling congestion and overload traffic.

NOTE – The traffic generated by a very large number of connected devices typically will be a relatively low volume of non-delay-sensitive data; however, the traffic is characterized as intermittent short burst traffic. The main problem in supporting the intermittent short burst traffic is that traffic has to go through the full signaling procedure, which causes the waste of battery life, spectrum and network capacity. The enhancement of current monolithic bearer management and the accompanied signaling in IMT-2020 network should be studied to cope with the issues coming from the increase of terminals.

[REQ] The network architecture for IMT-2020 is required to provide optimised user identification solution for the massive IoT devices.

[REQ] The network architecture for IMT-2020 is recommended to support power saving mode of IoT devices, if IoT device has equipped with power control functions.

8.3 Ultra-reliable and low latency communication services

8.3.1 Description

The new applications with very low latency and real-time constraints are expected to be prevalent in IMT-2020 networks: driverless cars, enhanced mobile cloud services, real-time traffic control optimization, emergency and disaster response, smart grid, e-health, augmented reality, remote tactile control, and tele-protection are some of the examples.

8.3.2 Requirements

[REQ] The network architecture for IMT-2020 is required to increase service/content availability.

NOTE – Service/content availability can be increased by the ability to replicate content and service functions and ability of forwarders for short and long term caching. In addition, delay-tolerant networking aspects such as cache-and-forward is very useful in the last mile where the content objects can be pushed or pulled to/by the end user based on its wireless conditions opportunistically.

[REQ] The network architecture for IMT-2020 is required to provide efficient signaling protocol or system to cope with the limitations on the existing mobile systems.

NOTE – There are various signalling procedures which contribute to the end-to-end connectivity establishment involving all the network components such as radio interface, fronthaul/backhaul and mobile core network. Besides the transport delay through the network components, the signaling which is basically accompanied in the beginning of each new session or transmission may give more serious impacts on the total end-to-end latency.

[REQ] The network architecture for IMT-2020 is required to provide enhanced performance for many diverse applications.

NOTE – Latency studies carried out on many IMT-Advanced deployed network demonstrate the 3GPP specifications provide adequate guidelines, while actual IMT-Advanced network performance varies due to many variables and adjacent ecosystem. Similarly, a network latency model and an end to end latency budget for services should be studied so that it provides optimal performance for many diverse applications in IMT-2020 networks.

9 Requirements from network operation points of view

9.1 Network flexibility and programmability

9.1.1 Description

An IMT-2020 network, as an integrated common core network, will be flexible enough to support extremely variety of requirements in user devices and application services. Therefore, the IMT-2020 network is envisioned as a network where multiple logical network instances tailored to various requirements can be created. As a basic feature to realize this, the separation of control and data planes in IMT-2020 network is needed, which enables the components of an IMT-2020 network to be reconfigured, upgraded or even replaced easily with those of other vendors. NFV is expected to do a significant role in making the IMT-2020 network more flexible by realizing network components as software components. We should note that the reality would not allow all the required network functions to be softwarized mainly because of the performance reason.

The openness given by the separation of control and data planes also makes the network programmable by controlling/steering traffic depending on user specific requirements and some use-cases.

Network slicing allows the operator to provide dedicated logical networks (i.e., network slices) with customer specific functionality. A network slice can span all the domains of network, such as transport network supporting flexible locations of functions, dedicated radio configurations or specific RAT and core network

dedicated to different types of services. Different types of network slices can be composed of not only standardized network functions but also some proprietary functions which are provided by different operators or 3rd parties.

9.1.2 Requirements

[REQ] The network architecture for IMT-2020 is required to provide softwarization capabilities with enhanced performance for wired and wireless (i.e. WLAN, satellite) networks.

[REQ] The network architecture for IMT-2020 is recommended to provide softwarization capabilities with enhanced performance for mobile access networks (i.e., newly-defined RATs for IMT-2020, evolved IMT-advanced RATs).

NOTE – The softwarization has been initially designed for wired networks and it may not be optimized for wireless (e.g. WLAN, satellite) and mobile networks. The mobile network optimized softwarization should be studied considering the best use of existing features to overcome the performance issues which may arise in some of SDN solutions.

[REQ] The network architecture for IMT-2020 is required to support programmability of network functions in data plane for easier provisioning of new emerging services.

NOTE – The requirements in data plane in an IMT-2020 network will be depending on the various service characteristics of new emerging services such as ICN. In-network data processing and service provisioning are the capabilities to cope with the diverse requirements in data plane. However, the capabilities have not studied from mobile network perspective. This also applies to a satellite network segment. The capabilities to support the easier provisioning of new emerging services in IMT-2020 network architecture should be studied.

[REQ] The network architecture for IMT-2020 is required to support the separation of control and data plane functions in network.

[REQ] The network architecture for IMT-2020 is required to create, operate and manage network slice.

NOTE – Network slice can be created dynamically to form a complete and fully operational network customized to cater for different diverse market scenarios. The operator should be able to compose network slices, i.e. sets of network functions (e.g. potentially from different vendors), resources to run these network functions and policies and configurations, e.g. for hosting multiple enterprises or MVNOs etc.

[REQ] The network architecture for IMT-2020 is required to protect negative impact in one network slice offered by other network slices.

NOTE – Dynamic control of slice resources (bandwidth, CPU, etc.) may need to be considered based on priority of slices.

[REQ] The network architecture for IMT-2020 is recommended to maintain the established QoS of the network slice after creation regardless of the status of other network slices.

NOTE – The isolation between different network slices should be considered in order to prevent data communication in one slice to negatively impact services offered by other slices.

[REQ] The network architecture for IMT-2020 is required to support elasticity of network slice in term of capacity with no negative impact on the services of this slice or other slice.

[REQ] The network architecture for IMT-2020 is required to support that the 3rd parties can create and manage a network slice configuration via suitable APIs, within the limits set by the network operator.

[REQ] The network architecture for IMT-2020 is required to be guaranteed the overall QoS of a network slice instance served for both operators and third parties (MVNOs, enterprises, service providers, content providers, etc.) on shared infrastructure.

[REQ] The network architecture for IMT-2020 is required to identify certain terminals and subscribers to be associated with a particular network slice.

NOTE – Users can obtain services from one or more specific network slices simultaneously based on subscription and/or UE context and/or local policy. A UE may access multiple slices simultaneously via a single RAN, and may provide network slice selection assistance information to the network.

[REQ] The network architecture for IMT-2020 is required to have the capability to meet the service-specific security assurance requirement in a single network slice, rather than the whole network slices.

[REQ] The network architecture for IMT-2020 is recommended that network functions are virtualized and it should support dynamic scale-in /scale-out per operator's policies.

9.2 Fixed-mobile converged networks

9.2.1 Description

The use of multiple heterogeneous radio access networks, including WLAN networks, and their interworking with each other in existing IMT networks are becoming prevalent with various approaches: LTE/WLAN link Aggregation (LWA), interworking with ePDG or TWAG (trusted wireless access gateway), etc. The trend is also expected to be continued in IMT-2020 networks, but with more advanced and efficient ways. The multi-connectivity through the multiple available radio access networks improves the robustness of the network as well as the throughput performance. Especially, the dual connectivity of an existing IMT-network and a new IMT-2020 network will help ensure the smooth introduction of IMT-2020 networks.

The IMT-2020 network is envisioned to be an access network-agnostic architecture whose core network will be a common unified core network for emerging new radio access technologies for IMT-2020 as well as existing fixed and wireless networks (e.g., WLAN). The access technology-agnostic unified core network should be accompanied by common control mechanisms which are decoupled from access technologies.

9.2.2 Requirements

[REQ] The network architecture for IMT-2020 is required to support that the signaling on different radio access networks is independent resulting in inevitable duplications in signaling for attachment, authentication, and mobility in each radio access networks.

[REQ] The network architecture for IMT-2020 is required to support newly-defined RATs for IMT-2020, evolved IMT-advanced RATs, WLAN access networks, and fixed broadband access networks.

[REQ] The network architecture for IMT-2020 is required to minimize access dependency in order to allow independent evolutions of core network and access networks.

[REQ] The network architecture for IMT-2020 is required to support simultaneous multi-RAT connectivity, optimization and resiliency of diversified MFH & MBH for the extreme traffic/connection density, enhanced multi-RAT coordination to ensure seamless user experience while mobile, standardized interfaces for multi-operator/shared use of infrastructure, etc.

[REQ] The network architecture for IMT-2020 is required to support more flexible and optimized multi-RAT interworking architecture, although traffic steering and the selection of best access technology in existing IMT networks is considered.

[REQ] The network architecture for IMT-2020 is required to support the multi-connectivity through the multiple available radio access networks that improves the robustness of the network as well as the throughput performance.

[REQ] The network architecture for IMT-2020 is required to consider fixed broadband access networks as an access network of IMT-2020 to interwork with other radio access networks.

NOTE – A converged access-agnostic core (i.e., where identity, mobility, security, etc. are decoupled from the access technology), which integrates fixed and mobile core, is envisioned as a direction of IMT-2020. Therefore, the IMT-2020 network architecture should be studied to support a true fixed and mobile convergence ensuring a seamless user experience within the fixed and mobile domains.

[REQ] The network architecture for IMT-2020 is required to support a unified end-to-end network management to ensure compatibility and flexibility for the operation and management of an IMT-2020 network.

NOTE – Various network management protocols in different network domains make it difficult to support unified network operations over multiple network domains.

[REQ] The network architecture for IMT-2020 is required to support standard OAM protocols for fault management and performance management between network equipment which may be commonly used across the IMT-2020 network.

NOTE – OAM protocols are not standardized in some parts of IMT networks such as fronthaul network.

9.3 Enhanced mobility management

9.3.1 Description

IMT-2020 should support a wide range of mobility options. Only around 30 % of total subscribers are actually mobile even in the current IMT networks according to some of MNOs's observation. Therefore, IMT-2020 should not assume the same mobility support for all devices and application services but rather provide mobility on demand only to those that need it. While mobility is not required for some stationary devices such as smart meters and CPE devices, but we also need to provide mobility for high-speed trains running at 500 km/h. Another mobility scenario to account for is when the RAN itself (UEs and base station) is mobile such as on a ship or plane. The service continuity levels also vary: seamless mobility, nomadic mobility, mobility for sporadic transmission, etc.

9.3.2 Requirements

[REQ] The network architecture for IMT-2020 is required to support enhanced mobility management such as "context-aware mobility" considering device types, application characteristics, etc.

NOTE 1 – It is expected that mobility requirements for user devices will be variable depending on the device and/or application types. Many user devices are stationary, e.g., smart meters and CPE, even in mobile networks while fast handover is a key feature of most mobile devices and some applications may address the mobility by setting up a new connection automatically with the help of buffering.

NOTE 2 – The signaling procedure in the existing IMT networks is heavy and not optimized for some of emerging new services such as IoT.

[REQ] The network architecture for IMT-2020 is required to support the mobility management aligning with those architectural changes that the IMT-2020 core network is envisioned to be a flat distributed network which is composed of the multiple distributed gateways to cope with traffic explosion and latency requirements of applications.

[REQ] The network architecture for IMT-2020 is required to support seamless and consistent user experience while moving across different access networks, and also steer mobile devices to choose the most suitable access technology in a seamless way.

9.4 Scalable and incremental installations

9.4.1 Description

It is cost effective to add on only that which is need in the location and capacity.

9.4.2 Requirements

[TBD]

9.5 Network capability exposure

9.5.1 Description

IMT2020 network is expected to accommodate various services and continues to define requirements for key service categories, i.e. massive IoT, critical communications, and enhanced mobile broadband, respectively. To allow the 3rd party to access information regarding services provided by the network (e.g., connectivity information, QoS, mobility, etc.) and to dynamically customize the network capability for different diverse use cases within the limits set by the operator, IMT2020 network should provide suitable access/exchange of network/connectivity information (e.g. via APIs) to the 3rd party.

The operator can provide some basic network capabilities (e.g. QoS policy) to the 3rd party. However, with the architecture evolution and new features (e.g. separation of control plane and data plane, network slicing and customize functions on-demand) brought by IMT-2020 network, it is necessary to look into the requirements and solutions for a common capability exposure framework and new network capabilities.

9.5.2 Requirements

[REQ] The network architecture for IMT-2020 is required to support the unified exposure for network capability by the IMT-2020 centralized control functions.

[REQ] The network architecture for IMT-2020 is required to support acquiring information (e.g., network/connectivity information) and allowing the 3rd party to access this information.

NOTE 1 – Identify the network information that can be provided to 3rd party ISPs/ICPs to enable more customized and efficient service provision.

NOTE 2 – For any identified information, provide the mechanism to enable the operator network to acquire information and to allow the 3rd party to access this information.

[REQ] The network architecture for IMT-2020 is required to support the capability of creating the dedicated network, i.e., the capability of composing functions on-demand into a dedicated slice, according to the requirement of the 3rd party.

NOTE 1 – It is envisioned to be flexible enough for realizing network components as software components, and simplifying the process of creating the dedicated network.

[REQ] The network architecture for IMT-2020 is required to support exposure of application registry and running capability to provide support for running the 3rd party applications.

NOTE – For example, it enables the access to the service close to UE and meets their latency requirements with respect to the UE. From the above requirements, solutions for this key issue will assume that the next generation system should be able to expose network capabilities to the 3rd party applications located within the operator domain close to the edge of the network, or outside the operator domain. The network capabilities which are suggested to take into considerations, but not limited to, are QoS Enforcement, Charging Control, Congestion Management, Service Chaining, Network slicing capability, and Application Hosting Close to the Network Edge, etc.

9.6 Authentication and Security

9.6.1 Description

It is foreseeable that the authentication based SIM/eSIM, e.g. AKA (Authentication and Key Agreement), still will be one of the most important authentication methods in IMT-2020. The current IMT networks leverage the USIM to access both the IMT network and IMT service network (e.g., VoLTE IMS) by AKA authentication, which leads to transforming from distributing to centralizing of the generation and storage of the authentication parameters.

A mass of IoT and M2M devices will access new IMT-2020 network architecture. A large number of devices requires much lower cost, power consumption, and complexity than smart phones. The current encryption algorithms, e.g. AES-128, are too complex to these devices. IoT devices are also restricted by the area of circuit board, so they have low capabilities on computing, storage, communication and battery.

Therefore, efficient authentication mechanism and lightweight cryptography algorithm for data protection should be considered to reduce the demands on device hardware and power consumption.

9.6.2 Requirements

[REQ] The network architecture for IMT-2020 is required to support unified authentication framework for the different access systems with various subscriber identity types.

NOTE – The SIM/eSIM based authentication mechanisms in current IMT network need to be taken into account but backward compatibility is not required to encourage development of efficient unified authentication framework.

[REQ] The network architecture for IMT-2020 is required to support efficient authentication mechanism and lightweight security algorithm for low complexity, low power consumption, and low data-rate IoT devices.

9.7 Flexible Signalling

9.7.1 Description

In the existing IMT networks, all different types of traffic are treated in a uniform procedure by a monolithic bearer management and its accompanied signaling protocol. The characteristics of traffic are expected to vary significantly from devices to devices and from applications to applications in IMT-2020 networks. For example, as the number of IoT devices is expected to increase, the intermittent short burst traffic from massive number of devices will cause signaling bottleneck. Meanwhile, a full-fledged signaling may be still essential for the devices/applications in which the support of mobility is more critical.

9.7.2 Requirements

[REQ] The network architecture for IMT-2020 is required to support flexible signaling architecture.

9.8 Numbering, Naming and Addressing

For further study.

9.9 QoS control

9.9.1 Description

The IMT-2020 system is expected to be able to provide the required QoS for a variety of different services, and different traffic characteristics. The services and traffic characteristics will also have differences in performance requirements in terms of lower latency, higher data rate, and higher mobility. Furthermore, some UEs can access more than one service simultaneously with diverse characteristics. In order to solve these performance issues in a resource efficient manner, QoS control is able to differentiate their handling according to service and device types.

Also, as the IMT-2020 system is expected to support multiple access technologies (e.g., New Radio Technology, E-UTRAN, WLAN, etc.), QoS control of core network is able to be access-agnostic.

In existing EPC, QoS control only covers RAN and core network, but for the IMT-2020 system E2E QoS control (e.g., RAN, core network, and transport network) is needed to support proper QoS interworking (e.g., mapping QCI to DSCP) between the networks that packet traverses.

Further, for existing EPC, QoS granularities in traffic treatment only covers Bearers per ADN, but for the IMT-2020 system QoS granularities need to be finer (e.g., per-packet flow, per-aggregated packet flow, etc.) in order to support extremely different performance requirements (e.g., ultra-low latency, ultra-high bandwidth).

The session management function is able to provide connectivity between UE and APN by selecting UP functions that achieve demanding user experience. Therefore, the relevant QoS information is able to be informed for session management function to select the proper UP functions that traffic traverses.

In EPC, network-initiated QoS control is used for network to initiate the signal to set up a dedicated bearer with a specific QoS, which is triggered by an application function or a DPI function. The signal is carried over standardized interfaces (Rx and/or Gx). For the IMT-2020 system user-initiated QoS control is needed to provide the specific QoS parameters (e.g., maximum bit rate, guaranteed bit rate) in order to support extreme performance requirements (e.g., ultra-low latency, ultra-high bandwidth).

The separation of control plane and user plane is expected to be one of basic principles for IMT-2020 system. Under this condition, in order to quickly provide E2E QoS authorization, a decision-making point of QoS control would be centralized in control plane while an enforcement point of QoS control would be distributed in user plane.

9.9.2 Requirements

[REQ] The network architecture for IMT-2020 is required to support access-agnostic QoS control from core network perspective.

[REQ] The network architecture for IMT-2020 is required to support E2E QoS as well as proper QoS interworking.

NOTE – QoS interworking is being still supported in the existing networks (e.g., QoS mapping between QCI and DSCP). When the levels of granularities between the two networks are different, aggregation may be needed.

[REQ] The network architecture for IMT-2020 is recommended to support a finer level of QoS granularities in order to provide service with different performance requirements.

[REQ] The network architecture for IMT-2020 is recommended to use QoS information along with other information such as slice ID, location, etc. by session management in selecting proper UP functions.

[REQ] The network architecture for IMT-2020 is required to support an implicit and/or an explicit request of packet forwarding treatment by network and/or user respectively.

NOTE – The packet forwarding treatment represents, for example, scheduling weights, admission thresholds, queue management thresholds, link layer protocol configuration, etc. [b-3GPP TS 23.203] describes more explanation in detail.

[REQ] The network architecture for IMT-2020 is required to support E2E QoS authorization by providing a centralized QoS decision-making point and a distributed QoS enforcement point within one operator.

9.10 Context awareness

For further study.

9.11 Profile management (User, Device, etc.)

For further study.

9.12 Network management

9.12.1 Description

Multiple network management protocols in different network domains make it difficult to support unified network operations over multiple network domains. A unified end-to-end network management should be considered to ensure compatibility and flexibility for the operation and management of an IMT-2020 network.

The IMT-2020 network should facilitate an integrated access network management for multiple radio access networks including WLAN and fixed access networks. The integrated multi-RAT and fixed network management should support seamless and consistent user experience while moving across different access networks, and also steer mobile devices to choose the most suitable access technology in a seamless way. In addition, simultaneous multiple connections for a mobile device to multiple RATs and fixed access networks

should be supported to increase user experienced data rate through the integrated management of multi-RAT and fixed access networks.

OAM protocols are not standardized in some parts of IMT networks such as the front haul network. Standard OAM protocols should be studied for fault management and performance management between network equipment that may be commonly used across the IMT-2020 network.

9.12.2 Requirements

[REQ] The network architecture for IMT-2020 is required to support a unified end-to-end network management to ensure compatibility and flexibility for the operation and management of an IMT-2020 network.

[REQ] The network architecture for IMT-2020 is required to support an integrated access network management for multiple radio access networks including WLAN and fixed access networks.

[REQ] The network architecture for IMT-2020 is required to support standard OAM protocols for fault management and performance management between network equipment that may be commonly used across the IMT-2020 network.

9.13 Accounting and Charging

9.13.1 Description

Charging is one of the important necessary functions for the network and service providers. The charging requirements of IMT-2020 network are expected to be similar to existing networks such as NGN, 3G and LTE etc., and the architecture of IMT-2020 should be taken into account to collect charging data.

There are various charging models that need to be taken into account to design architecture of IMT-2020 network to support different charging policies and requirements of the network operators and service providers including third party service providers. To meet requirements of third party service providers such as MVNO, charging function to provide charging data to third party would be required. The function for providing charging data can be realized as an open API for third party and it will be one of the capability exposure feature of IMT-2020 network.

The charging models includes volume based charging, time based charging, session based charging, application based charging, and no charging, etc. Also, it may possible to provide combined charging model that consists of each charging model (e.g., time and volume based charging).

Also, charging capability for IMT-2020 network needs to be applied to each network slice independently. Different charging model can be applied to each network slice at the cost of complexity.

9.13.2 Requirements

[REQ] The network architecture for IMT-2020 is required to provide charging function that supports online or offline charging.

[REQ] The network architecture for IMT-2020 is required to support various charging models based on the charging policy of network operators and/or service providers.

NOTE – The charging models includes volume based charging, time based charging, session based charging, application based charging, no charging, and combined charging, etc.

[REQ] The network architecture for IMT-2020 is recommended to provide charging function that hand over charging data to third party.

NOTE – If charging function to hand over charging data is supported, it will be realized as an open API for third party that is one of network capability exposure features of IMT-2020 network.

9.14 Interworking

9.14.1 Description

As new IMT-2020 network doesn't get better coverage than existing LTE network in preliminary stage, LTE network is still needed to guarantee services when UE leaves out of IMT-2020 coverage to existing LTE network coverage as shown in figure1. So IMT-2020 core network is required to support interworking with existing LTE network, and including as an anchor to manage handover action between IMT-2020 and existing LTE network.

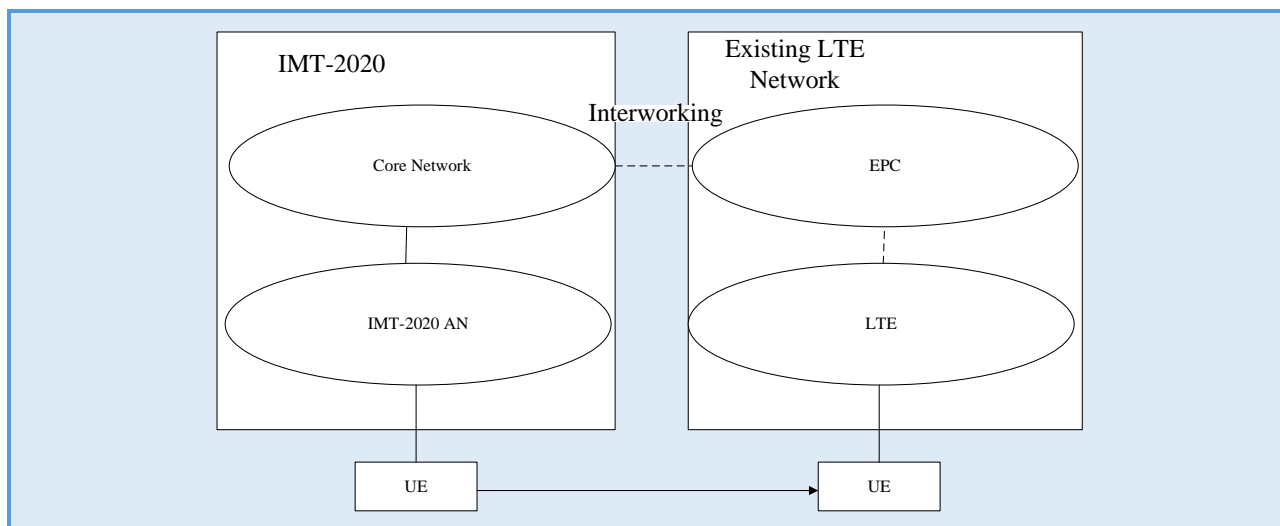


Figure 1 – Interworking scenario

9.14.2 Requirements

[REQ] The network architecture for IMT-2020 is required to support interworking with existing LTE network.

[REQ] The network architecture for IMT-2020 is required to support mobility management, session management and service continuity as an anchor when handover from IMT-2020 to existing LTE network.

[REQ] The network architecture for IMT-2020 is required to support end-to-end QoS management, charging control when interworking between IMT-2020 and existing LTE network.

Appendix I

Use cases of IMT-2020

(This appendix does not form an integral part of this Recommendation.)

I-1 Network slicing with shared Core Network instance

Network slice instance is a set of network functions, and resources to run these network functions, forming a complete instantiated logical network to meet certain network characteristics. The function in a network slice instance can include RAN functions and Core Network (CN) functions. In IMT-2020 architecture, there are two types of core network slice instance listed as follows:

- Dedicated CN instance:** All the network functions belong to this CN instance are deployed only serving for or associated with one network slice instance, even though the functions are the same with ones of other network slice instances, e.g. CN instance A.
- Shared CN instance:** Different shared CN instances can share part of network functions, e.g. CN instance B and C.

Therefore, there are two kinds of network slicing use cases considering two types of CN shown in Fig. I-1.

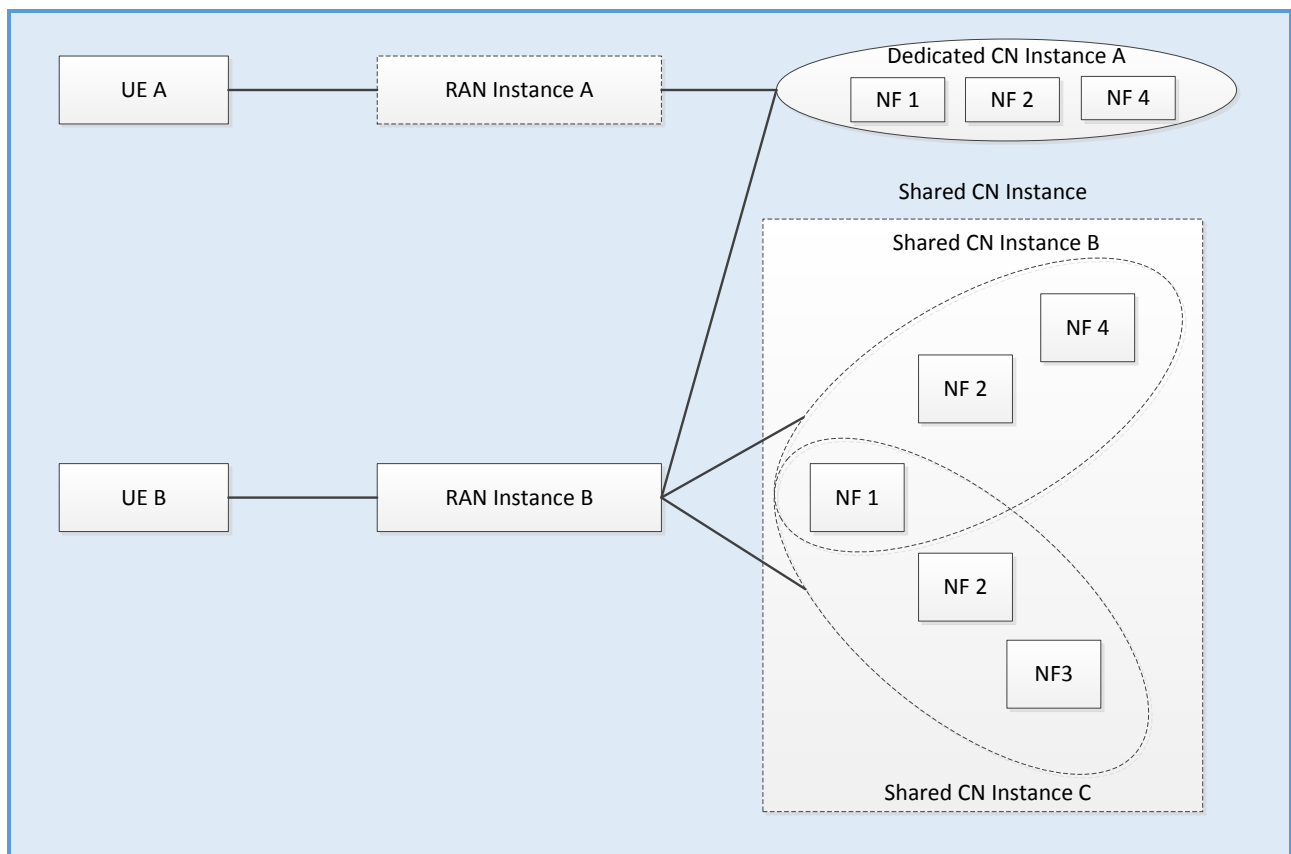


Figure I-1 - Network slicing with shared CN instance

As shown in Figure I-1, a CN instance is made up of multiple network functions. For the dedicated CN instance, all the network functions are dedicated to the CN instance. For the shared CN instance, part of the network functions can be shared among different CN instances. There exists something difference in the identification and selection mechanisms of CN instance. For dedicated CN instance, network slice selection function may be not necessary. However, for the shared CN instance, the shared network function is required to find out the next following network function in the corresponding CN instance if need. For shared CN instance, how to realize the isolation of shared network functions among different CN instances is should also be considered.

Appendix II

Contributors (in Alphabetical Order)

(This appendix does not form an integral part of this Recommendation.)

This is the list of all contributors who submitted any written form of comments or contributions.

–	Bugaba Simon	Uganda Communications Commission
–	Byung Jun Ahn	ETRI
–	Donna Bethea-Murphy	Inmarsat
–	Fidelis Onah	Nigerian Communications Commission
–	Gang Fu	China Unicom
–	Haru Alhassan	Nigerian Communications Commission
–	Hui Cai	China Mobile
–	Jeong Yun Kim	ETRI
–	Jinlan Ma	China Telecom
–	John Grant	Nine Tiles
–	Miao Xue	China Unicom
–	Namseok Ko	ETRI
–	Noik Park	ETRI
–	Nura Falalu	Nigerian Communications Commission
–	Marco Carugi	NEC Corporation
–	Shin-Gak Kang	ETRI
–	Stanley Russo	SES
–	Thomas Hayford	National Communication Authority, Ghana
–	Wei Chen	China Mobile
–	Weixing Wang	Nokia Networks
–	Wenyi Li	China Telecom
–	Xiayu Li	CATR
–	Xiaowen Sun	China Mobile
–	Yachen Wang	China Mobile
–	Yilin Lin	China Telecom
–	Ying Cheng	China Unicom

Appendix III

Acknowledgement

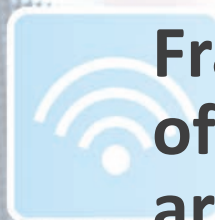
(This appendix does not form an integral part of this Recommendation.)

- This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R7116-16-1001, 5G Core Network Technologies Standards)
- This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0132-16-1005, Development of Wired-Wireless Converged 5G Core Technologies)
- This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R7117-16-0127, Development of Standard for Service Adaptive Dynamic Network Slicing)

Bibliography

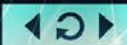
- [b-3GPP TS 23.203] 3GPP TS 23.203 V14.2.0 (2016), *Policy and charging control architecture (Release 14)*
- [b-FG IMT2020-Gap] ITU-T TD 208 (PLEN/13) (2015), *Report on Standards Gap Analysis*
- [b-ITU-R M.1645] Recommendation ITU-R M.1645 (2003), *Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000*
- [b-ITU-T Y.3011] Recommendation ITU-T Y.3011 (2012), *Framework of network virtualization for future networks*
- [b-ITU-T Y.3300] Recommendation ITU-T Y.3300 (2014), *Framework of software-defined networking*





Framework of IMT-2020 network architecture





enter your search here...



10100010111
011101010110001
101110011010
0110010100111
00011101
010

CONNECTION



BUSINESS

MARKETING
BANKING
ACCESS



00010101000111
0010111010101101
11010110011010
01010110010100111
1100100011101
1101010



PROJECT



NEWS



SHARE



SHOP



SOCIAL NETWORK



MEDIA



EMAIL



000101010010100010111
001011101010110001
110101110011010
01010110010100111
1100100011101
1101010



CLOUD



ENCRYPTION

000101010010100010111
001011101010110001
110101110011010
01010110010100111
1100100011101
1101010



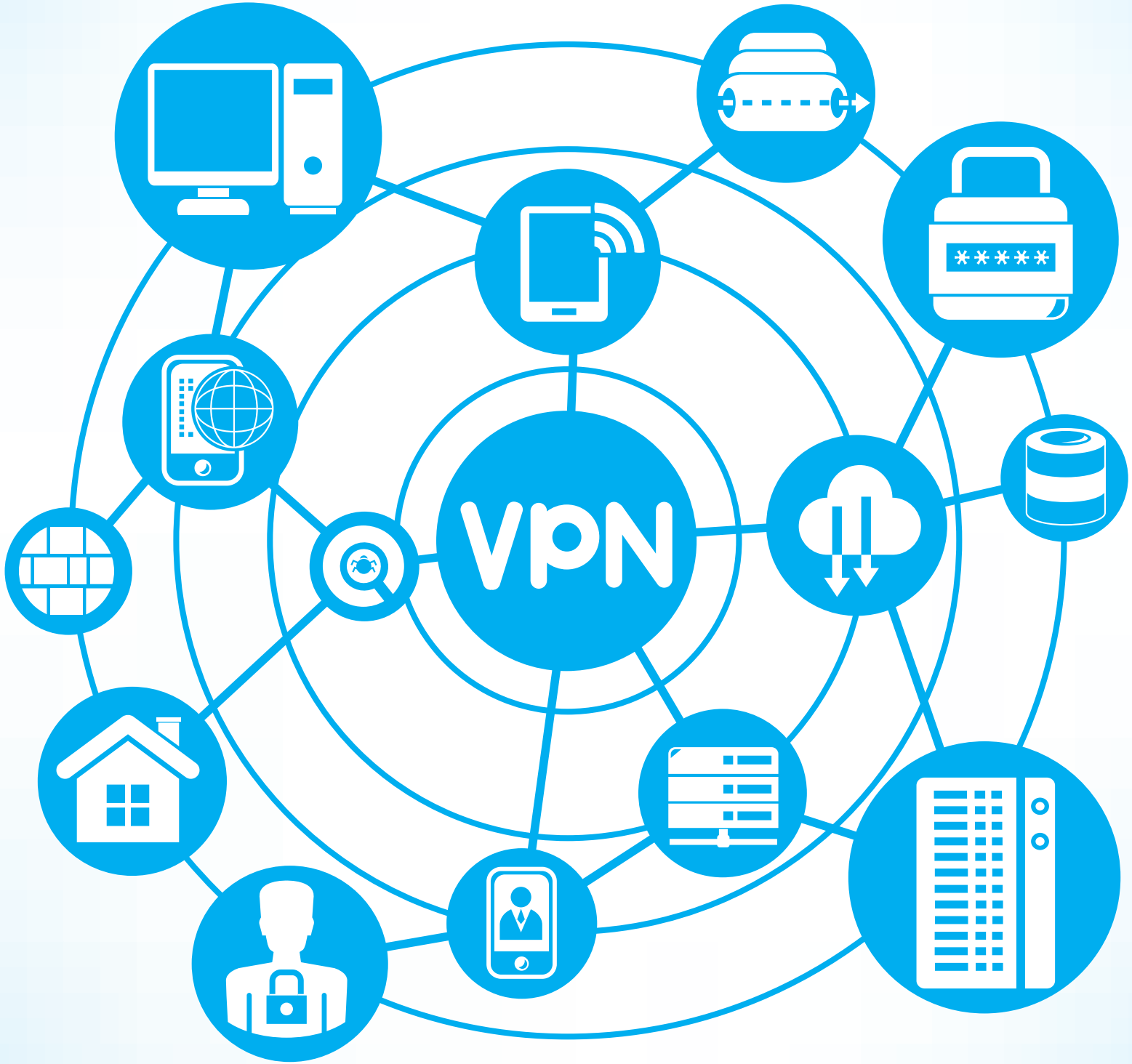
SIGNAL METER



CHAT

Table of Contents

1	Scope
2	References
3	Terms and definitions
3.1	Terms defined elsewhere
3.2	Terms defined in this Recommendation
4	Abbreviations and acronyms
5	Overview of IMT-2020 network architecture framework
5.1	General framework and architectural principles
6	Functional architecture of IMT-2020 network
6.1	Design principles
6.2	End-to-end functional architecture of IMT-2020
6.3	Functional entities
6.4	Reference points
Appendix I – Living list on Common IMT-2020 Network Architecture Diagram	
I.1	Living list document #1 – Common architecture framework proposed in [b-ITU-T IMT-I-225]
I.2	Living list document #2 – Common architecture framework proposed in [b-ITU-T IMT-I-247]
I.3	Living list document #3 – Common architecture framework proposed in [b-ITU-T IMT-I-250]
I.4	Living list document #4 – Common architecture framework proposed in [b-ITU-T IMT-I-255]
Appendix II – Contributors (in Alphabetical Order)	
Appendix III – Acknowledgement	
Bibliography	



1 Scope

This draft Recommendation provides the framework of IMT-2020 network architecture. Following an overview of this IMT-2020 network architecture framework, a high-level functional architecture of IMT-2020 network, including functional entities and reference points, is specified.

2 References

The following ITU-T Recommendations and other references contain provisions, which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editors indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is published regularly.

NOTE – The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-R M.2083-0] Recommendation ITU-R M.2083-0 (2015), *Framework and overall objectives of the future development of IMT for 2020 and beyond*.

[ITU-T Y.3001] Recommendation ITU-T Y.3001 (2011), *Future networks: Objectives and design goals*.

3 Terms and definitions

3.1 Terms defined elsewhere

3.1.1 control plane [ITU-T Rec. Y.2011 (10/2004)]: The set of functions that controls the operation of entities in the stratum or layer under consideration, plus the functions required to support this control.

3.1.2 data plane [ITU-T Rec. Y.2011 (10/2004)]: The set of functions used to transfer data in the stratum or layer under consideration.

3.1.3 functional architecture [ITU-T Rec. Y.2016 (08/2009)]: A set of functional entities used to describe the structure of an NGN. These functional entities are separated by reference points, and thus, they define the distribution of functions. The functional entities can be used to describe a set of reference configurations. These reference configurations identify which reference points are visible at the boundaries of equipment implementations and between administrative domains.

3.1.4 functional entity [ITU-T Rec. Y.2082 (08/2013)]: An entity that comprises an indivisible set of specific capabilities. Functional entities are logical concepts, while groupings of functional entities are used to describe practical, physical implementations.

3.1.5 logical resource [ITU-T Y.3011]: An independently manageable partition of a physical resource, which inherits the same characteristics as the physical resource and whose capability is bound to the capability of the physical resource.

3.1.6 management plane [ITU-T Rec. Y.2011 (10/2004)]: The set of functions used to manage entities in the stratum or layer under consideration, plus the functions required to support this management.

3.1.7 user plane [ITU-T Rec. Y.2011 (10/2004)]: A synonym for user plane.

3.1.8 virtual resource [ITU-T Y.3011]: An abstraction of physical or logical resource, which may have different characteristics from the physical or logical resource and whose capability may not be bound to the capability of the physical or logical resource.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 network function: A processing function in a network. It includes but is not limited to network nodes functionality, e.g. session management, mobility management, switching, routing functions, which has defined functional behaviour and interfaces. Network functions can be implemented as a network node on a dedicated hardware or as a virtualized software functions.

3.2.2 network slice: A complete end-to-end logically partitioned network providing dedicated telecommunication services and network capabilities. The behaviour of the network slice is realized via network slice instance(s).

3.2.3 network slice blueprint: A complete description of the structure, configuration and the plans/work flows for how to instantiate and control the Network Slice Instance during its life cycle.

3.2.4 network slice instance: An activated network slice. It is created based on network slice blueprint (or template).

NOTE – A set of managed run-time network functions, and resources to run these network functions, forming a complete instantiated logical network to meet certain network characteristics required by the service instance(s). It provides the network characteristics which are required by a service instance. A network slice instance may also be shared across multiple service instances provided by the network operator. The network slice instance may be composed by none, one or more sub-network instances, which may be shared by another network slice instance.

3.2.5 PDU session: Association between the UE and a data network through IMT-2020 that provides a PDU connectivity service. The type of the association includes IP type, Ethernet type and non-IP type.

3.2.6 physical resource: A physical asset for computation, storage or transport including radio access. Network functions are not regarded as resources.

4 Abbreviations and acronyms

This deliverable defines the following abbreviations:

AN	Access Network
CN	Core Network
CP	Control Plane
CPE	Customer Premises Equipment
CritC	Critical Communication
eMBB	enhanced Mobile BroadBand
IMT-2020	International Mobile Telecommunication 2020
IoT	Internet of Things
MMCF	Mobility Management Control Function
NAS	Non-Access Stratum
NF	Network Function
NFI	Network Function Instance
NS	Network Slice
NSI	Network Slice Instance
NSSF	Network Slice Selection Function
PCRF	Policy and Charging Rule Function

PGW	Packet data network GateWay
QoS	Quality of Service
QSCF	QoS Control Function
RAT	Radio Access Technology
SDN	Software Defined Network
SGW	Serving GateWay
SIDB	Subscriber Information DataBase
SMCF	Session Management Control Function
TUPF	Terminating User Plane Function
UACF	Unified Authentication Control Function
UE	User Equipment
UPGW	User Plane GateWay function
UP	User Plane
VPN	Virtual Private Network

5 Overview of IMT-2020 network architecture framework

5.1 General framework and architectural principles

IMT-2020 shall provide converged network-computing capabilities to enable variety of services including eMBB, massive IoT, autonomous vehicles, tactile Internet, etc. The scope of the IMT-2020 includes not only mobile technologies but also fixed communication, cloud, and services. Further, IMT-2020 will allow the provisioning of both existing and new services independently of the network and the access type used.

It is expected that network systems will consist of distributed multiple physical and/or virtual network functions that may be deployed on-demand basis over the infrastructure in the operator's network, and be able to support diverse service requirements.

By virtue of softwarization and pervasive deployment of computing infrastructure with connectivity and storage, network function instances can be distributed on an on-demand basis. This brings to the IMT-2020 network architecture a shift in networking paradigms: a transition from deployment of network entities with fixed functionalities (e.g., PGW, SGW, etc.) to deployment of network functions as necessary.

IMT-2020 network shall provide network services demanding diverse requirements, by using network functions instantiated at right place. IMT-2020 infrastructure will provide required infrastructure resources to instantiate the network functions. Network operators can provision and operate many different network slices according to their business strategies.

Network slicing enables the operator to create logically partitioned networks customised to provide optimized solutions for different market scenarios. These scenarios demand diverse requirements in terms of service characteristics, required functionality, performance and isolation issues.

The functional architecture of IMT-2020 network shall provide a complete set of network functions required to support all IMT-2020 services. On the other hand, a network slice is comprised only of the necessary network functions. They are collected from a complete set of network functions in the IMT-2020 network functional architecture, and orchestrated for the particular services and purposes.

The general framework of IMT-2020 can be represented by two separate architecture levels, i.e. the 'IMT-2020 network slice life-cycle management' level and the 'network slice instances' level as shown in Figure 5-1. Functions for creating and managing network slice instances and the functions instantiated in the network slice instance are mapped to respective architecture level.

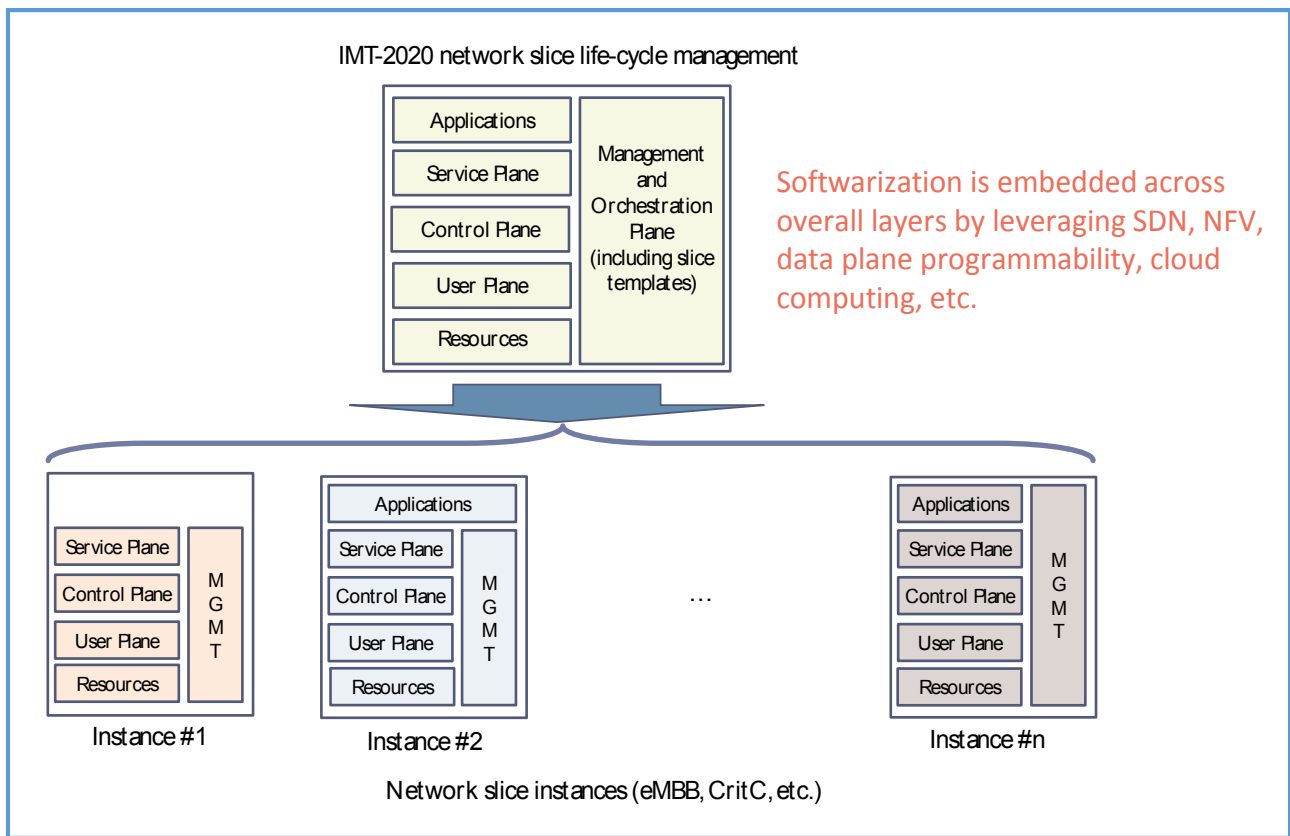


Figure 5-1 – Conceptual IMT-2020 non-radio network architecture

5.1.1 IMT-2020 network slice life-cycle management level architecture

This level architecture provides functionalities required for the life cycle management of network slice instances.

In this architecture level:

- Applications are slice life cycle management related ones for use by network operators;
- Service, control, and user planes are not library functions to instantiate any slices, but rather the architectural planes to be used to manage life cycle of network slice instances. For example, functions in the user plane of this architecture level provide required user plane functionalities, e.g. related to the manipulation of SDN or VPN;
- Resources provide the physical and virtual infrastructure required to instantiate network slices;
- Management and orchestration plane provides global management and orchestration functions including slice blue print or templates that are used to create slice instances.

5.1.2 Network slice instance level architecture

In this architecture level:

- Network functions (NFs) in control plane and user plane operate on instantiated network slices, i.e. network slice instances. Fundamental control plane network functions include NSSF (Network Slice Selection Function), MMCF (Mobility Management Control Function), SMCF (Session Management Control Function), PCRF (Policy and Charging Rule Function), and UACF (Unified Authentication Control Function), etc. as shown in Figure 6-1;
- Some Network Function instances (NFIs) can be shared by multiple NSIs (Network Slice Instances) and commonly used by those NSIs. For example, an UE may access multiple NSIs simultaneously. In such case, those NSIs share some control plane (CP) NFIs, e.g. mobility management function, authentication function, etc. Those shared CP NFIs are 'Common Function Instances' in Figure 5-1;

- Control plane of each NSI is composed of common CP NFIs and NSI specific CP NFIs. Control plane of a NSI may have no common CP NFIs or no NSI specific CP NFIs;
- Some NSIs may have applications used by end-users. Applications in the NSI are managed by management plane. And the applications are created from the resource in the slice management and orchestration level architecture, based on the template or blue print.
- As new services emerge in the future, interconnection model between network functions needs to allow easy insertion of new network functions while maintaining the existing network function interconnection path without increasing complexity of the system.

6 Functional architecture of IMT-2020 network

6.1 Design principles

From [b-ITU IMT-O-042] and [b-3GPP TR 22.861], a large number of requirements and principles of IMT-2020 system are identified in order to design a system architecture in terms of evolutionary architecture, vertical industries' demand and multiple access, etc. The following principles and characteristics are considered for the design of IMT-2020 network functional architecture framework:

- Separation of control plane (CP) and user plane (UP) functions, allowing independent scalability and evolution, where CP dynamically configures UP functions (i.e. activates various operations of the user plane functions as needed). The IMT-2020 network should support highly scalable distributed architecture to avoid signalling congestion and to minimize the signalling overhead for diverse UE/RAT/service requirements. In order to support distributed network architecture, and optimized routes for application data and signalling data, Control/User-plane functions should be clearly separated with defined interface;
- Flexible deployment of UP and CP functions, i.e. centralized or distributed;
- Access network (AN) agnostic common core network (CN) design (e.g., AN-CN functional division and a common interface between them);
- Efficient support of different levels of UE mobility. It is expected that mobility requirements for user devices will vary depending on the device and/or application types. Many user devices are stationary, e.g., smart meters and CPE, even in mobile networks while fast handover is a key feature of most mobile devices and some applications may address the mobility by setting up a new connection automatically with the help of buffering. Therefore, IMT-2020 should not assume the same mobility support for all devices and application services but rather provide mobility on demand only to those that need it.
- Support of services that have different latency requirements between the UE and the Data Network. In IMT-2020 networks, gateways to a core network can be flexibly located closer to the cell sites, which will bring a significant reduction on back haul and core network traffic (e.g., with placing content servers closer to UE, services latency leads to be reduced). Furthermore, The IMT-2020 network should support services that have different end-to-end QoS (data rate, reliability, latency, location accuracy etc.) requirements. There may be some latency critical applications while there are others which are tolerant long end-to-end latency.
- Support of network slicing. An IMT-2020 network, as an integrated common core network, will be flexible enough to support extremely variety of requirements in user devices and application services. Therefore, the IMT-2020 network is envisioned as a network where multiple logical network instances tailored to various requirements can be created. Network slicing allows the operator to provide dedicated logical networks (i.e., network slices) with customer specific functionality. The architecture shall allow different network configurations in different network slices. A network slice can span all the domains of network, such as transport network supporting flexible locations of functions, dedicated radio configurations or specific RAT and core network dedicated to different types of services. Different types of network slices can be composed of not only standardized network functions but also some proprietary functions that are provided by different operators or 3rd parties. Modular function design to enable flexible network slicing is desirable (e.g., separation of mobility management (MM) and session management (SM) control functions);

- Support of network capability exposure;
- Support of unified authentication framework;
- Abstraction of the transport domain to support various transport technologies.
- Leveraging existing techniques such as Network Function Virtualization and Software Defined Networking):
 - SDN enables separation of Control plane and Data plane functions differently from existing network equipment. It enables network architecture to be flexible and programmable.
 - NFV is expected to play a significant role in making the IMT-2020 network more flexible by realizing network components as software components. It leads to reduce total cost of ownership, improve operational efficiency, energy efficiency, and simplicity and flexibility for offering new services.

6.2 End-to-end functional architecture of IMT-2020

6.2.1 Overview of end-to-end IMT-2020 functional architecture

6.2.2 Functional architecture of IMT-2020 core network

Figure 6-1 shows the functional architecture framework of IMT-2020 core network in network slice instance level.

Each network function in the figure may be comprised of one or more functional entities.

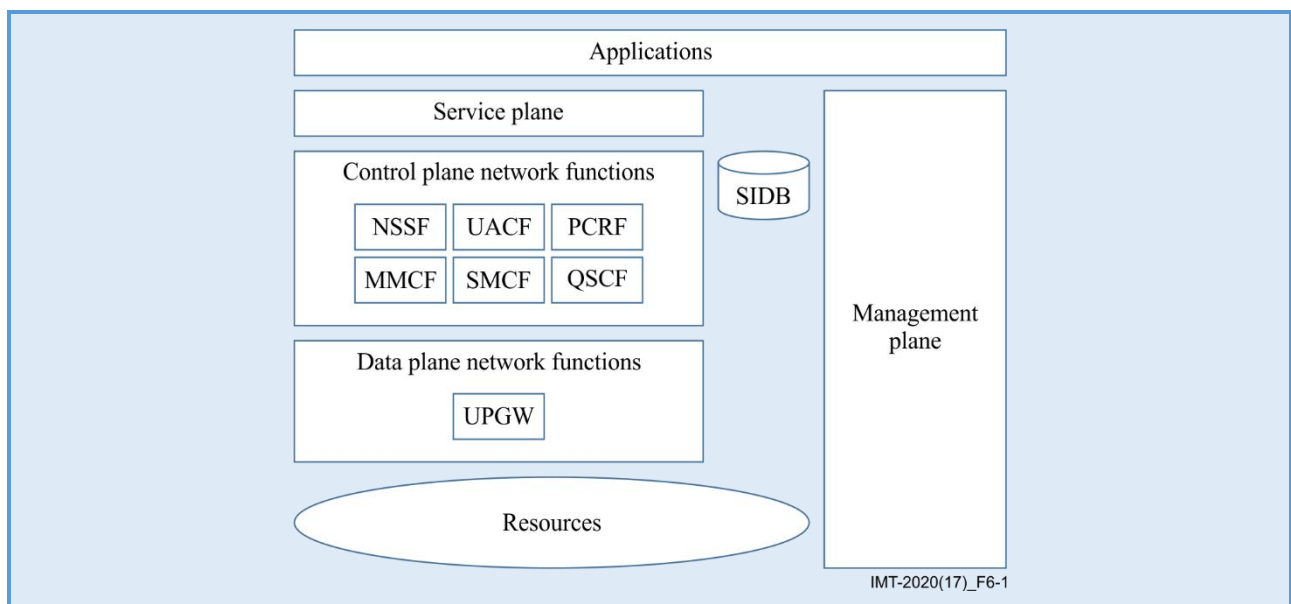


Figure 6-1 – Framework of IMT-2020 core network functional architecture

6.2.3 Network function descriptions

6.2.3.1 Control plane network functions

NSSF (Network Slice Selection Function)

- Selects core network slice instance to accommodate the UE.

NOTE – It is assumed that UE can indicate required network slice type to the core network and then the core network selects a particular network slice instance for the UE. Further clarifications on how UE and CN interact to select a NSI are required.

MMCF (Mobility Management Control Function)

- Provides mobility support, including session continuity, for UEs connected to IMT-2020 core via mobile and/or fixed access networks.
 - When a UE is simultaneously connected to the same core network over heterogeneous access networks, the UE is served by a single MMCF via interworking function (IWF).
- Performs access specific mobility control, e.g. UE reachability management, mobility restriction, etc.
- Performs access common mobility control, e.g. UE registration and location management, etc.
- Routes NAS (Non-Access Stratum) signalling messages (e.g., session management related) to corresponding session management control function.
 - A UE with multiple established PDU sessions may be served by different instances of SMCF. The MMCF selects the SMCF functions for the PDU sessions. MMCF may select different SMCF instances for different PDU sessions.
- Terminates RAN-CP interface and NAS (including ciphering and integrity protection)
- Access Authentication

SMCF (Session Management Control Function)

- Responsible for the setup of the IP or non-IP traffic connectivity (i.e. PUD session) for the UE as well as controlling the user plane for that connectivity (e.g., selection/re-selection of user plane network functions and user path, enforcement of policy and charging rules, etc.).
 - Multiple SMCF can serve one UE in a network slice instance, e.g. to support different data networks.
 - Data plane network functions are selected/re-selected based on UE location, UE subscription profile, session and service continuity mode selected for the PDU session, data network, user plane network function load distribution, etc.
- Allocates UE IP address (including optional authorization).
- Provides session continuity with respect to UE mobility.
- Terminates session management related NAS messages
- Initiates access network specific session management information
 - When SMCF needs to send NAS session management signalling to a UE, it provides information allowing the MMCF to retrieve the corresponding UE NAS signalling context.
- Handles policy and charging rules

UACF (Unified Authentication Control Function)

- Responsible for the authentication of the identity (e.g., user identity) that is presented to the network, when a UE requests to receive service(s) from the network and network slices.

SIDB (Subscriber Information DB function)

- Provides and manages user subscription data, policy data (e.g., on QoS and charging), session/user related context and state.

PCRF (Policy and Charging Rule Function)

- Provides dynamic policies rules to CP network functions for QoS enforcement, charging control, traffic routing, etc.

QSCF (QoS Control Function)

- Enables E2E QoS in proper granularity (e.g., per UE, per flow, or per PDU session) with QoS parameters (e.g., maximum bit rate, guaranteed bit rate, priority level, etc.).
- Controls user plane network functions for the QoS enforcement.

6.2.3.2 Data plane network functions

Data plane network functions support operations such as forwarding to other user plane functions /control plane functions, processing transport traffic, performing user plane gateway functions, etc. Multiple user plane functions per session can be activated and configured by the control plane (e.g., SMCF) as needed for a given service.

UPGW (User Plane Gateway function)

- Access/Trunking media gateway functionality which provides interworking between the IP-based transport and non-IP based access.
- Terminating User Plane Function (TUPF)
- Session anchor to provide IP and non-IP PDU session (e.g., IP anchor, tunnelling, etc.)
- Access border gateway functionality between an access network and a core network
- Interconnection border gateway functionality between transport networks in different domains
- Packet routing & forwarding
- Traffic handling (e.g., QoS enforcement)
- Optional functionalities such as Packet inspection, Lawful intercept (UP collection), etc.

6.3 Functional entities

NOTE – Plans are for the functional entities to be developed by the ITU-T Study Group 13 in 2017-2018.

6.4 Reference points

NOTE – Plans are for the Reference points to be developed by the ITU-T Study Group 13 in 2017-2018.

Appendix I

Living list on Common IMT-2020 Network Architecture Diagram

(This appendix does not form an integral part of this Recommendation.)

The following I.X clauses provide four different inputs discussed for the common architecture framework during FG IMT-2020 work. Even though they were the basis for the discussion concerning the consolidation of the Clause 5.1 and related contents. It was agreed to keep them in this living list for further consideration purpose.

I.1 Living list document #1 - Common architecture framework proposed in [b-ITU-T IMT-I-225]

NOTE – Following is the proposed text for IMT-2020 network architecture framework in [b-ITU-T IMT-I-225]. For more details of the proposal, refer to [b-ITU-T IMT-I-225].

I.1.1 General framework and architectural principles

IMT-2020 shall provide converged network-computing capabilities to enable variety of services including eMBB, massive IoT, autonomous vehicles, tactile Internet, etc. The scope of the IMT-2020 includes not only mobile technologies but also fixed communication, cloud, and services. Accordingly, IMT-2020 is defined as systems, system components, and related aspects that support far more enhanced capabilities than those described in IMT-2000.

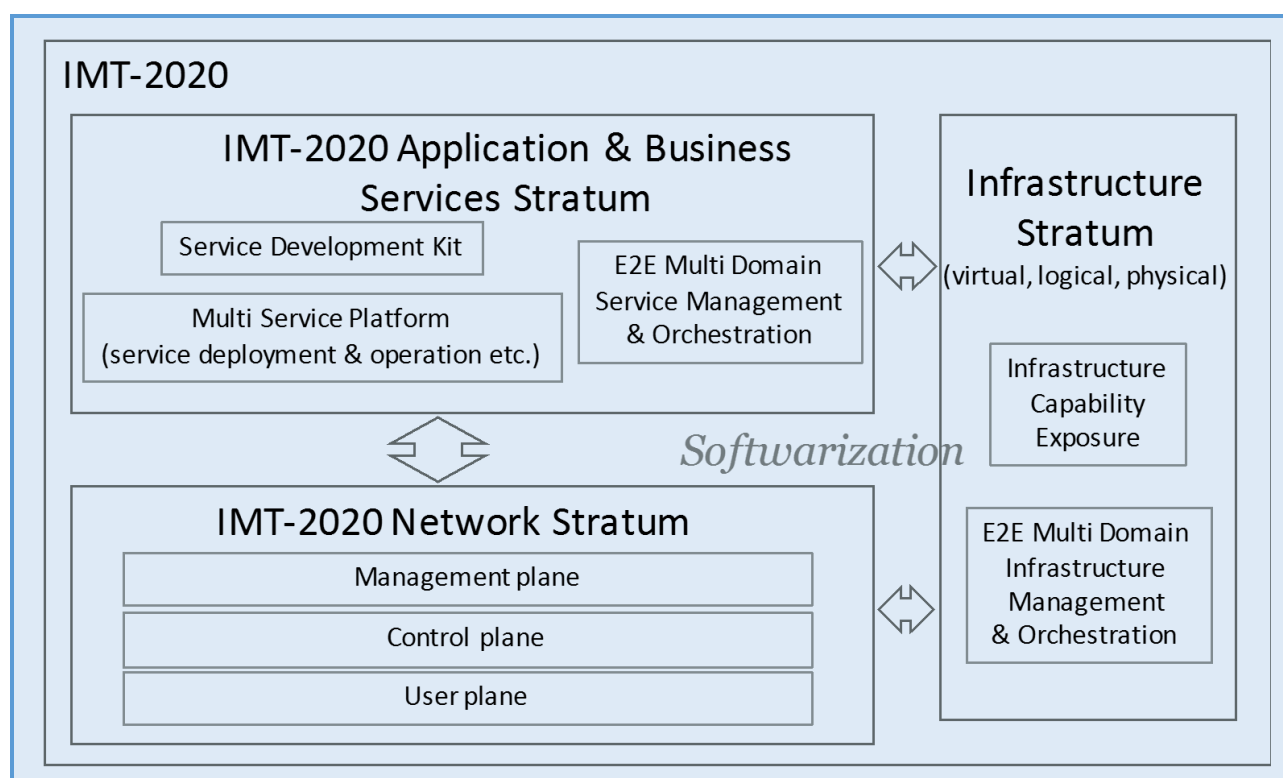


Figure 5-1 – General framework of IMT-2020 architecture

Figure 5-1 shows the general framework of IMT-2020 architecture.

For IMT-2020, there is an increased need on service customization by the service providers whereby some of them will offer their customers the possibility to customize their own services through service related APIs (Application Programming Interfaces) in order to support the creation, provisioning and management of services. IMT-2020 will allow the provisioning of both existing and new services independently of the network

and the access type used. And functional entities are likely to be distributed on-demand basis over the infrastructure. Therefore, in the IMT-2020 architectures, there shall be a clear decoupling of service, network, and infrastructure, allowing them to be offered separately and to evolve independently. The separation is represented by three strata in the figure.

In general, each stratum will have its own set of roles, players and administrative domains (refer to [b-ITU-T Y.110]). The roles involved in service(s) provision are independent from those involved in network connectivity provision. Each stratum needs to be treated separately from a technical point of view. In the figure, Multi-domain orchestration refers to the automated management of services and resources in multi-technology (multiple domains involving different cloud and networking technology) and multi-operator (multiple administrative domains) environments [b-ITU-T IMT-O-047].

I.1.1.1 IMT-2020 network stratum

By virtue of softwarization and pervasive deployment of computing infrastructure, network function instances can be distributed on-demand basis. This brings IMT-2020 network architecture a shift in networking paradigms: a transition from deployment of network entities with fixed functionalities (e.g., PGW, SGW, etc.) to deployment of network functions as necessary.

IMT-2020 network stratum provides network services requested by service stratum using network functions instantiated at right place. IMT-2020 infrastructure stratum provides required infrastructure resources to instantiate the network functions.

The functional architecture of IMT-2020 network stratum shall provide a complete set of network functions required to support all IMT-2020 services. Reference points and information flows between interacting functions will be defined in the architecture, too. As new services emerge in the future, interconnection model between network functions needs to allow easy insertion of new network functions while maintaining the existing network function interconnection path without increasing complexity of the system.

Network operators can provision and operate many different network slices according to their business strategies. A network slice is comprised of only necessary network functions. They are collected from a complete set of network functions in the IMT-2020 network functional architecture, and orchestrated for the particular service and purpose.

Life cycle management including orchestration and instantiation of network slices is performed by management plane network functions (e.g., NSMF). Management plane is also responsible for the life cycle management of network functions.

Network functions in control plane and user plane operate on instantiated network slices, i.e. network slice instances. Fundamental control plane network functions include ANCF (Access Network Control Function), NSCF (Network Slice Control Function), MMCF (Mobility Management Control Function), SMCF (Session Management Control Function), PCRF (Policy and Charging Rule Function), and AAAP (Authentication Function), etc.

Figure 5-3 shows IMT-2020 network architecture framework. High level functional architecture of IMT-2020 is defined in clause 6. Each network function is comprised of one or more functional entities.

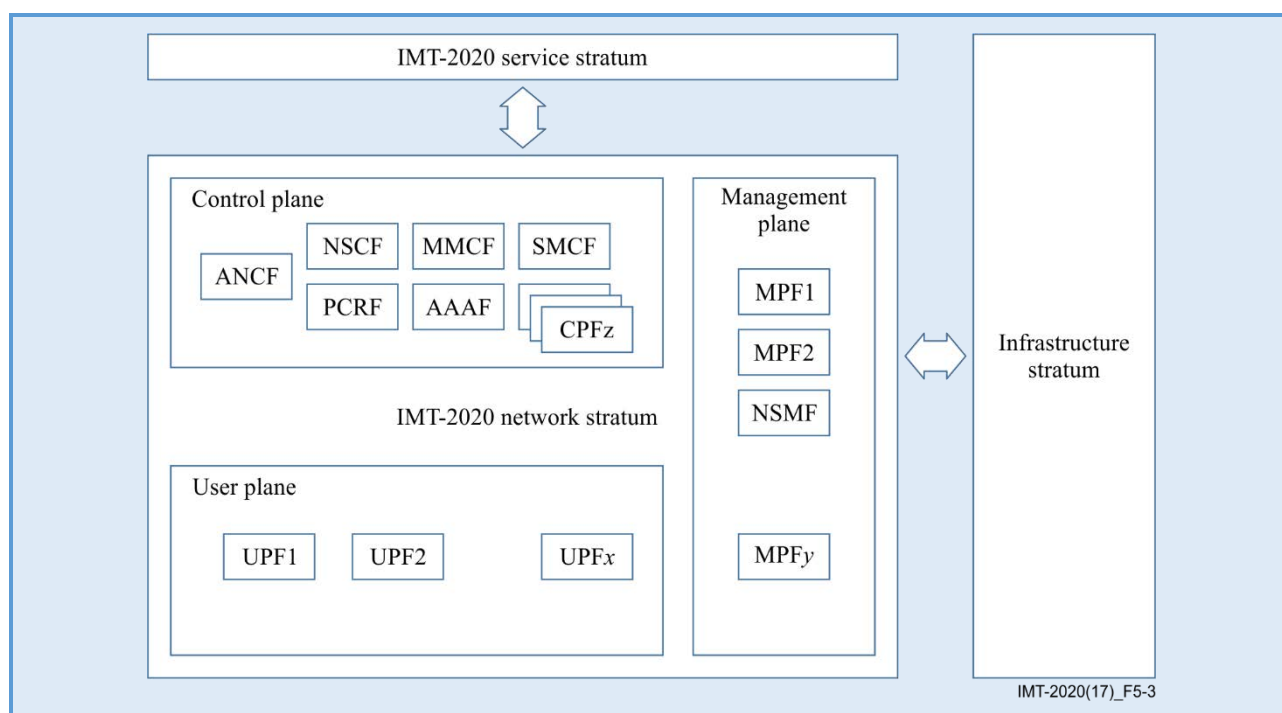


Figure 5-3 – IMT-2020 network architecture framework

I.2 Living list document #2 - Common architecture framework proposed in [b-ITU-T IMT-I-247]

NOTE – Following is the proposed text for IMT-2020 network architecture framework in [b-ITU-T IMT-I-247]. For more details of the proposal, refer to [b-ITU-T IMT-I-247].

I.2.1 General framework and architectural principles

According to [IMT-2020 network requirement], following high level architectural principles are considered for the IMT-2020 network:

- Support network slicing to accommodate various services with typical service categories, i.e. enhanced mobile broadband, massive IoT, ultra-reliability and low latency communications, and autonomous vehicles
- Minimize access and core network dependencies, design converged core network to adapt various access network
- Separation of control and user plane functions in core network side, allowing independent scalability and evolution, allowing flexible deployment
- Design the network function and interconnection with leveraging NFV and SDN technology
- Support network capability exposure to enable rich value-added services
- Design uniform network management system which cover both IMT-2020 system and legacy 2G, 3G and LTE networks.

Figure 1 shows the IMT-2020 high-level framework diagram, which is comprised of four planes and associated relationships to meet the above architectural principles.

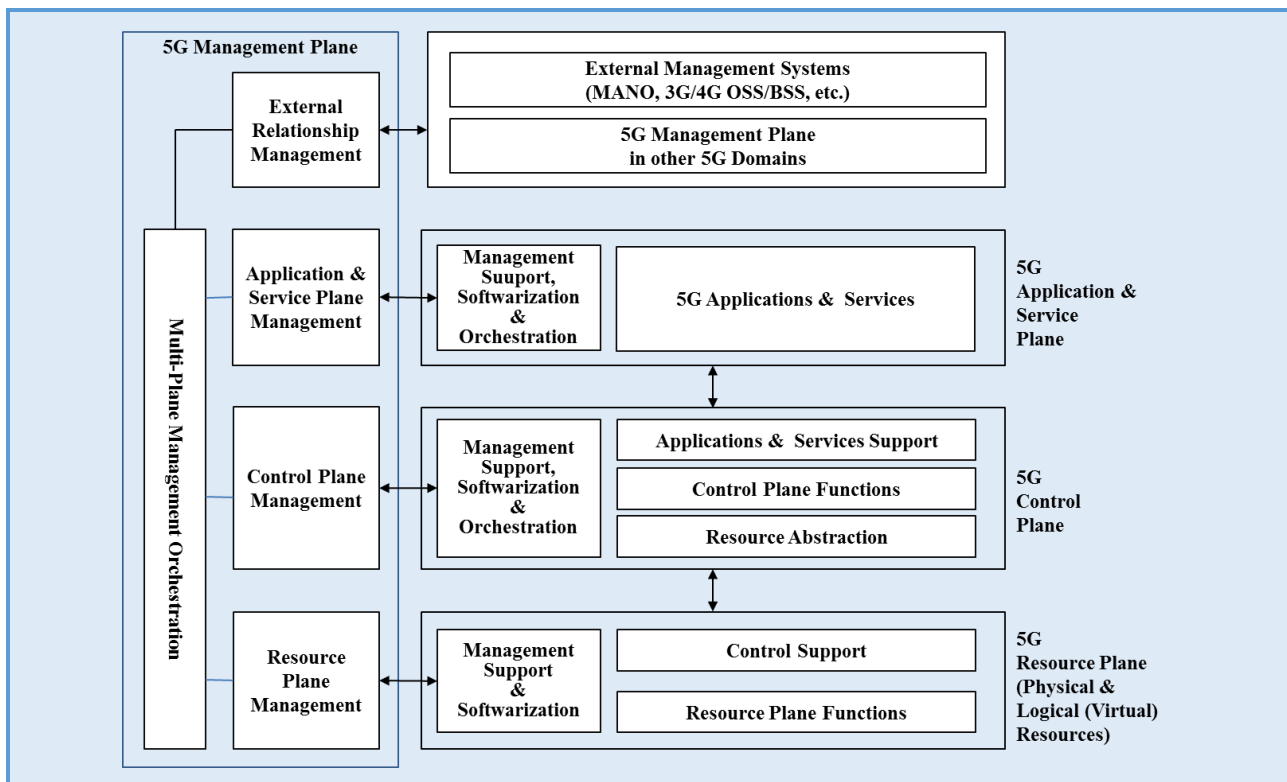


Figure 1 – IMT-2020 general framework diagram

The descriptions on the planes are as follows:

- **5G Application and Service Plane**

The 5G Application and Service Plane (5G-ASP) enables a service-aware behaviour of 5G network in a programmatic manner via various APIs which exposes softwarized network capabilities. It consists of application & service plane management support/softwarization/orchestration functional component that interacts with 5G multi-layer management function and multiple 5G application functional components that interacts with 5G Control Plane via standard interfaces.

- **ASP Management Support, Softwarization, & Orchestration**

The ASP Management Support, Softwarization, and Orchestration (ASP-MSSO) softwarizes and orchestrates 5G applications with the support of Multi-Layer Management.

It includes applications code repository management, application lifecycle management (creation, modification, and deletion), performance monitoring of applications to meet SLA requirements, detection, isolation, and correction of application faults, and security management of 3rd party applications (authentication and identity management).

It also provides the capabilities for provisioning of ASP resources (resources that are used to run ASP applications) based on the requests received from the Management Plane. It coordinates provisioning of ASP service resources and network resources in Resource Layer in cross-layer manner and also performs atomic transactions to ensure the integrity of the requested provisioning service. ASP-MSSO provides capabilities for managing of ASP resources. It manages the capacity and performance related to each ASP service. ASP-MSSO obtains resource quality of service information in order to measure and record key performance indicators (KPIs) for the ASP service. Capacity for an ASP service is allocated or de-allocated based on these KPIs.

In case when virtual networks have been created by Softwarization capability, they are built with the support of the Control Layer, and may use the same Application Layer, or to attach an ASP instance to each virtual network.

- **5G Control Plane**

The 5G Control Plane (5G-CP) provides programmable means to control the behaviour of 5G resources (such as data transport and processing), following requests received from the 5G-ASP and according to 5G-MP policies. 5G-CP is operating on resources provided by the 5G Resource Layer and exposes an abstracted view of the 5G access and core network to the 5G Application and Service Plane. The 5G-CP interacts with the 5G Resource Plane (5G-RP), 5G-ASP, and 5G-MP via standard interfaces.

- **Applications and Services Support**

The Control Plane Applications and Services Support (CP-AAS) provides a standard interface to the 5G-ASP for accessing 5G network information and requesting application-specific 5G network behaviours. The information exposed to 5G-ASP is abstracted by means of information and data models. In case of network virtualization by softwarization capability, the CP-AAS may expose a subset of network resources which can be used for exclusive for the virtual network.

- **Management Support, Softwarization, and Orchestration**

The Control Plane Management Support, Softwarization, and Orchestration (CP-MSSO) provides management support, softwarization, and orchestration capabilities of 5G-CP and the management, if delegated by 5G-MP, of network resources based on the policies provided by 5G-MP in multi-domain and/or heterogeneous 5G access and core network environment. It also provides the capabilities for provisioning CP functions and managing a particular CP function and its scalability and performance based on the KPIs. It keeps track of the overall state of allocated and available resources in the 5G-CP.

It orchestrates CP functions based on service policies (e.g., a placement rule which aims to avoid single points of failure) and softwarizes some CP functions as virtual functions if required.

The CP-MSSO also provides capabilities for connecting multiple 5G access and core network domains in order to make inter-domain operations. In such case it is responsible for establishing the communication path(s) required, and for passing appropriate identity and credentials with requests made among these domains.

- **Control Plane Functions**

The Control Plane Functions (CP-F) provides a set of programmable control and optionally management functions (if delegated by 5G-MP) covering e.g., discovery of physical and virtual network topologies, network element configuration, and traffic flows forwarding management, on-demand path computation, monitoring of responsible resources, mobility management, mobile edge including distributed EPC control, cloud control, etc.

- **Resource Abstraction**

The Control Plane Resource Abstraction (CP-RA) provides capabilities to support unified programmability of resources. Common information and data models of underlying resources are provided so that the developers of CP functions can simplify their program logics without the need for a detailed knowledge of the underlying network resource technologies. These models provide a detailed, abstracted view of both, physical or virtualized network resources. The Resource Abstraction can create multiple virtual resources by using a single physical resource or can create a composite virtual resource as an aggregation of several virtual or physical resources. In cases when the Resource Plane provides itself the abstracted view of its resources to the 5G-CP, the CP-RA can be ignored.

- **5G Resource Plane**

The 5G Resource Plane (5G-RP) is where the physical or virtual network elements perform transport and/or processing of data packets according to 5G-CP decisions. Such data processing can be hardwired as it is the case at the present or programmable which is supported by user plane deep programmability capabilities. These decisions, the information about network resources, and resource management policies/requests are exchanged via standard interfaces.

– **Control Support**

The Resource Plane Control Support (RP-CS) interacts with 5G-CP via standard interface. It provides data model, or information model, of the 5G network resources, which are to be abstracted in the Control Plane. It may perform resource abstraction, instead of CP, in case when it is supported by the underlying resource technology.

The RP-CS is programmable. It enables to update and/or modify the Data Transport and/or the Data Processing capabilities supported by user plane deep programmability functionality. For example, a new protocol or a new set of interface specifications may be added for the purpose of enhancing functionalities of the Data Transport and/or the Data Processing functional components. For another example, software bugs may be fixed through the update.

– **Resource Plane Functions**

The Resource Plane Functions provide data forwarding and data routing functionalities. The control of the data forwarding functionality is provided by 5G-CP. 5G-RP routing rules can be customized by the 5G-CP for SDN applications needs. Data forwarding and data routing functionalities are extensible and programmable. Examples of functional extension include enhancing the existing data transport capabilities and incorporating new data transport capabilities. It also provides data processing functionalities to examine and manipulate data. They enable to alter format and/or payload of data packets/frames and to adjust sending of data packets/frames as specified by SDN applications. Data processing functionalities are also extensible and programmable. Examples of functional extension include enhancing the existing data processing capabilities and incorporating new data processing capabilities, e.g., new transcoding algorithms.

– **Management Support and Softwarization**

The Resource Plane Management Support and Softwarization (RP-MSS) provides resources description, i.e. vendor, software version, and their status (e.g., CPU load, used RAM memory or storage). It may include a management support functionality that performs some local management operations if delegated by 5G-MP. This functionality can be used to support technology dependent resource discovery, for programmable, local monitoring of the Resource Plane entity (in order to limit the amount of exchanged data or to focus on a specific issue) or to implement autonomic management behaviour. It provides capabilities of softwarization of physical resources into virtual resources. It also provides lifecycle management of all RP software-based resources with the support of 5G-MP.

• **5G Management Plane**

The 5G Management Plane (5G-MP) provides functionalities for managing the functionalities of other planes, i.e., 5G-ASP, 5G-CP and 5G-RP. It interacts with these layers using standard interfaces. 5G-MP interoperates with 3rd party management systems, for example for billing, customer care, statistics collection or dynamic service provisioning, therefore a standard interface to the external operator management systems (e.g., MANO, 3G/4G NMS/BSS, etc.) also exists. It is also responsible for the orchestration (if needed) of dynamically deployed management tasks (services) in a multi-plane management nature of 5G-MP and coordinated (orchestrated) reconfiguration of resources of the 5G-RP.

5G-MP includes functionalities for supporting fault management, configuration management, accounting management, performance management and security management (FCAPS) as described in [ITU-T M.3400]. Examples of such functionalities are equipment inventory, fault isolation, performance optimization, and initial configuration of 5G-RP, 5G-CP and 5G-ASP. It is also responsible for the lifecycle management of 5G software-based components of all 5G planes. By considering energy and environmental constraints, 5G-MP provides energy efficient operations of virtual and physical resources used for the implementation of all planes. The functionality can be realized by energy-aware algorithms supported by resource status monitoring and analytics.

5G-MP functions include Resource Plane Management (RPM), Control Plane Management (CPM), Application and Service Plane Management (ASPM), Multi-Plane Management Orchestration (MPMO), and External Relationship Management (ERM).

5G-MP can delegate some management operations, specifically those which require intensive exchange of data with 5G-CP to be performed directly by 5G-CP (for example, autonomous management operations). Delegated operations are performed by management support functions within each plane.

– **Application and Service Plane Management**

The Application and Service Plane Management (ASPM) provides management functionality for managing resources of 5G-ASP. It includes FCAPS of resources in the application layer. It is also used in a cross-layer orchestration driven by 5G-ASP.

5G-specific Application and Service Plane management includes applications code repository management, applications lifecycle management (creation, modification, and deletion), performance monitoring of virtualized applications to meet the SLA requirements, detection, isolation, , recovery of application faults and security management of 3rd party applications (authentication, identity management).

– **Control Layer Management functional component**

The Control Plane Management (CPM) functional component includes management of resources used to deploy control layer functions (hardware, software platforms, links connecting control plane with other planes) in order to ensure high availability and scalability of CP, performance, fault and security management of control traffic generated between CP entities and RP or ASP entities. It can bootstrap CP entities or their components, monitor performance of CP entities in terms of reliability, utilization, detection, root cause analysis, and correction of faults of CP; detection, isolation, and control of CP related traffic and authentication and authorization management functionality of 5G-CP entities. The policy management can include business, technical, security, privacy and certification policies that apply to CP-services and their usage by 5G applications. The CPM also provides energy-aware the CP resource management.

– **Resource Plane Management**

Resource Plane Management (RPM) functional component is responsible for the management of physical and/or virtual resources. RPM provides capabilities for discovering and bootstrapping of virtual and physical resources to make them ready for operation. RPM functional component include support for FCAPS and RP orchestration (for example coordinated resource reconfiguration) that is provided by Multi-Plane Management Orchestration (MPMO) functionality. MPMO provides the capabilities for provisioning of RP resources and RPM keeps track of the overall state of allocated and available resources of RP. It is also responsible for managing the configuration relationship between virtual and physical control layer resources, performance co-relation between virtual and physical control layer resources, and faults by considering the relationship between virtual and physical control layer resources, and finally isolation, control of anomaly to selected SDN network resource, and finally authentication and authorization of the network resources. By providing the overall resource status information monitored as an input to RPM, energy efficient resource management capability can be realized by, e.g., turning off of unused resources.

– **Multi-Plane Management Orchestration**

The Multi-Plane Management Orchestration (MPMO) provides functionalities for supporting the lifecycle management of 5G application/network services across the entire 5G operator's domain and orchestrates multi-layer resource management. It coordinates management operations among application, control, and resource planes, especially the relationship among virtualized and physical resources across multiple layers.

– **External Relationship Management**

The External Relationship Management (ERM) provides management functionality to interwork with external management entities. It plays a role of the representative interface of 5G management toward the external management entities such as MANO, 2G, 3G, and LTE OSS/BSS. Its main functionality includes abstraction of 5G management information, request/reply of management operations with external management entities, external SDN policy management, data analytics, charging and DevOps operations.

It also provides another important management capability to interwork with 5G-MP in other domains. This interaction allows inter-slice management and hierarchical/recursive 5G management capabilities for scalability purposes.

I.3 Living list document #3 - Common architecture framework proposed in [b-ITU-T IMT-I-250]

NOTE – Following is the proposed text for IMT-2020 network architecture framework in [b-ITU-T IMT-I-250]. For more details of the proposal, refer to [b-ITU-T IMT-I-250].

I.3.1 IMT 2020 Global Pictures - Logical and Physical Architectures - Revisited

The following figure summarizes the IMT-2020 **Network Elements abstractions** in the form of a high-level logical design and applicable plane specific functionality.

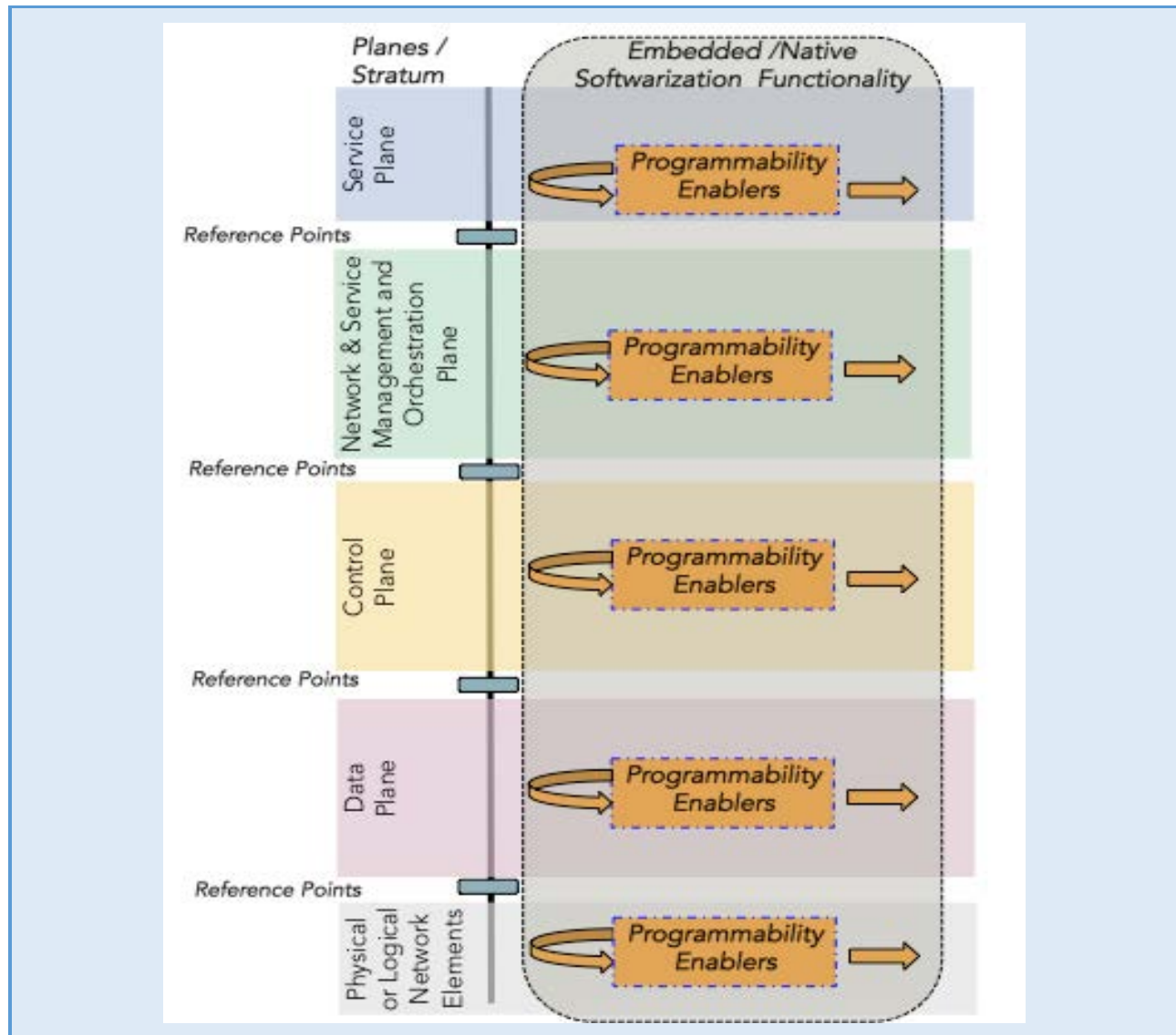


Figure 1 – Proposal: IMT-2020 Network Element Viewpoint

The following figure summarizes the IMT-2020 **Network abstractions** in the form of a high-level logical network architecture and applicable plane specific functionality.

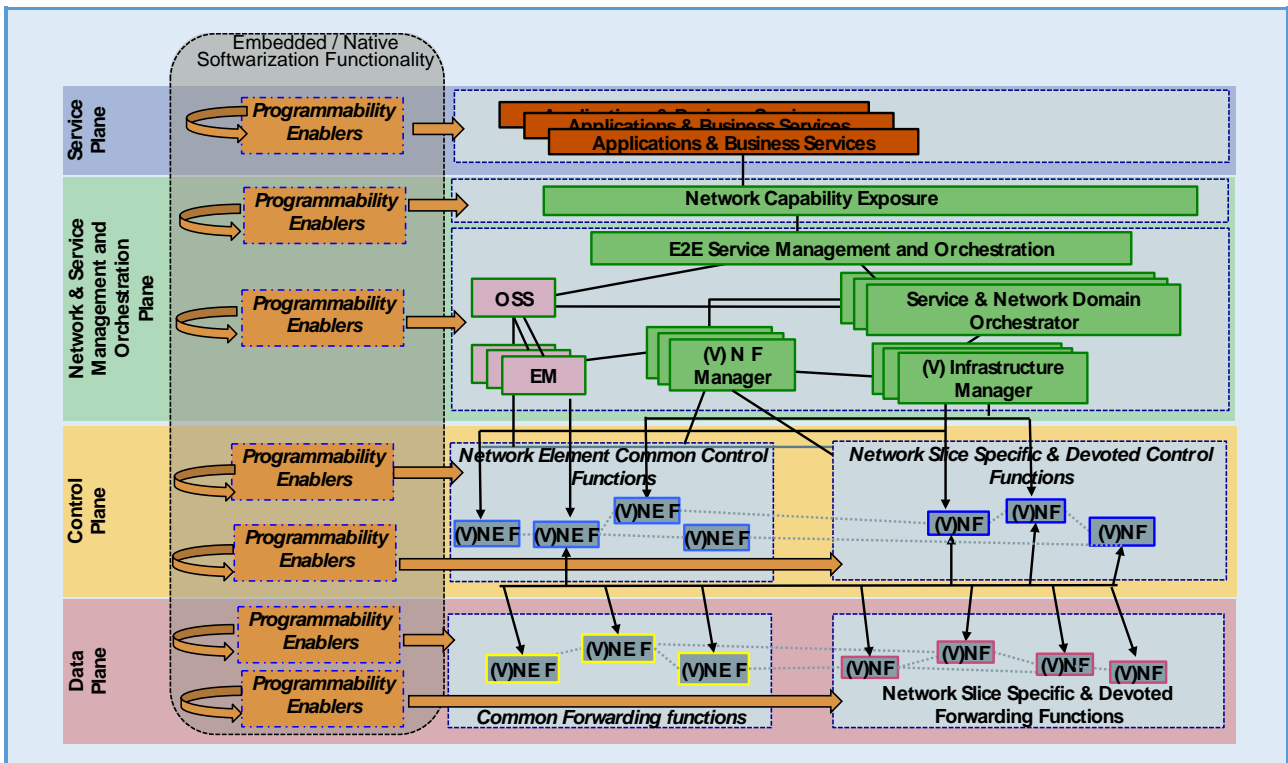


Figure 2 – 5G Logical Network Viewpoint

The following figure summarizes the IMT-2020 Network abstractions in the form of a high-level logical network architecture and applicable plane specific functionality with the separation of Service Management from Network Management.

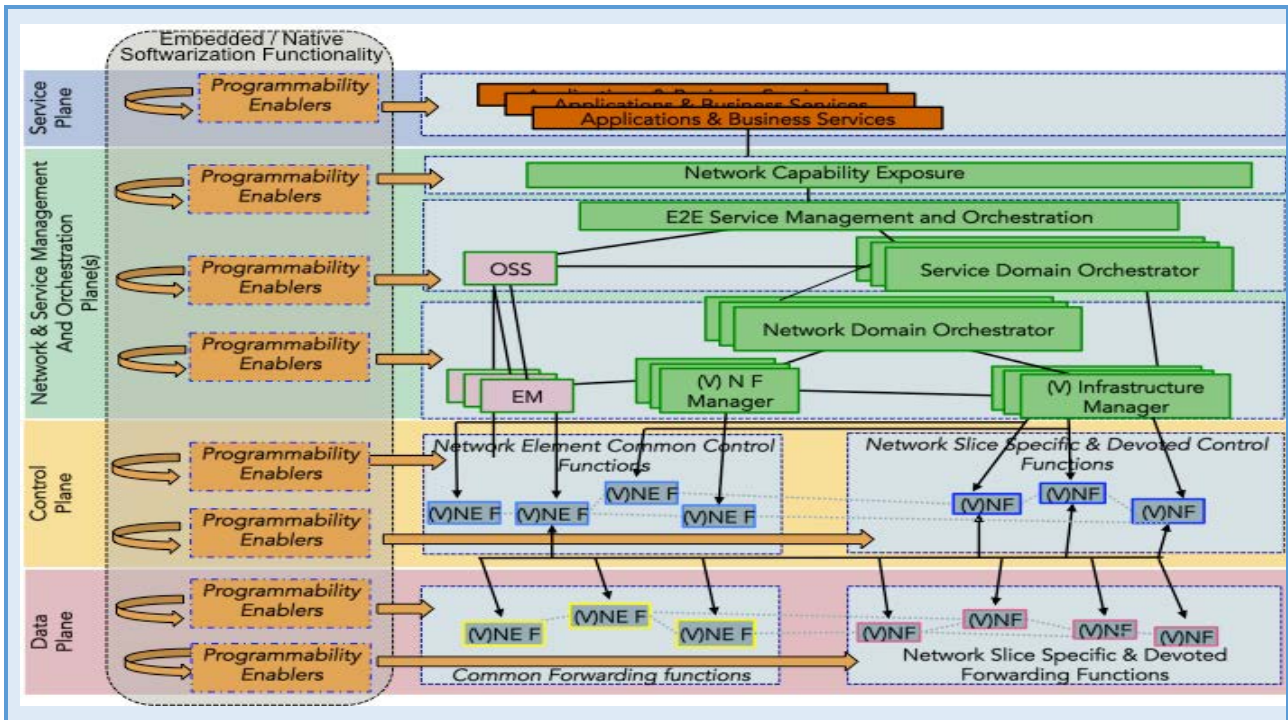


Figure 3 – 5G Logical Network with separation of Service Management from Network Management

The following figure summarizes the IMT-2020 Network logical and physical abstractions in the form of a high-level schematic for 5G Network segments and applicable plane specific functionality.

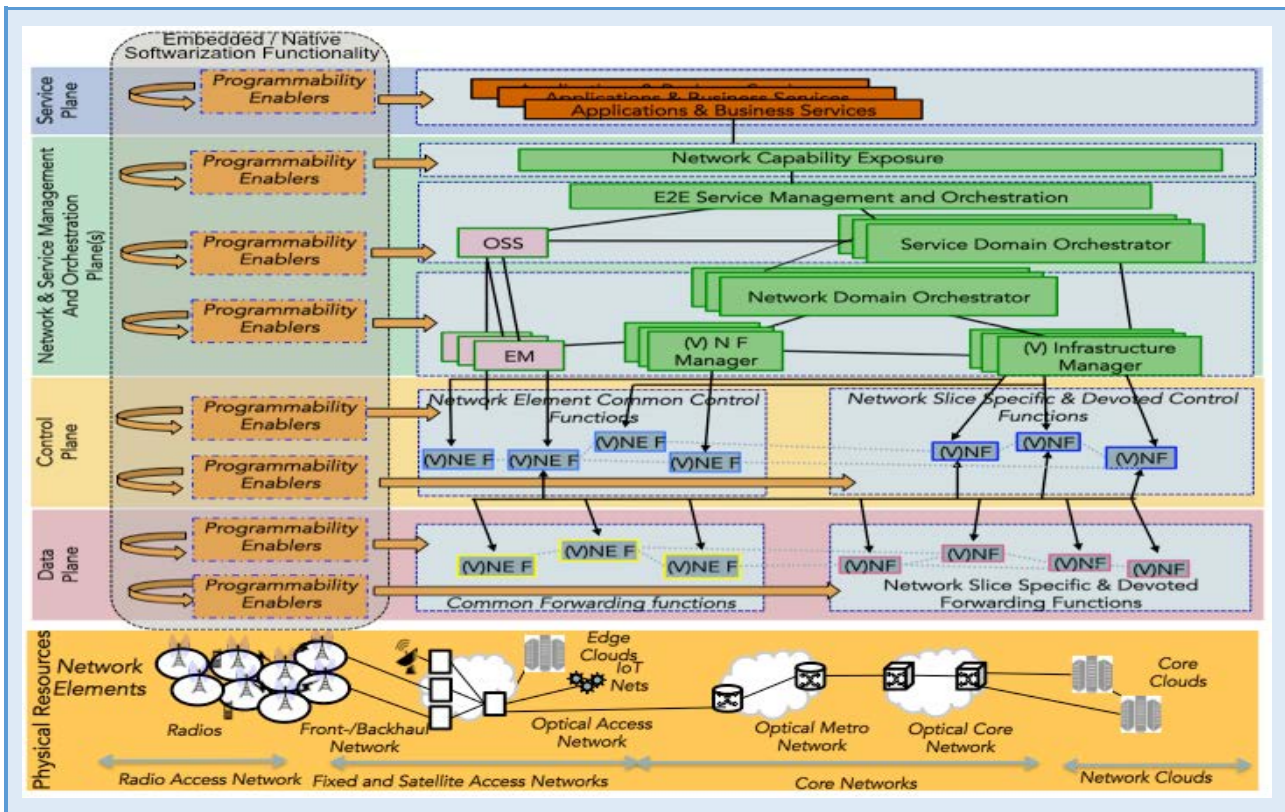


Figure 4 – 5G Logical and Physical Network Segments Viewpoint

1.3.2 Taxonomy of IMT 2020 Network Functionality

The followings represent a taxonomy of IMT 2020 network functionality in separate planes (application / service plane, Management planes, control plane, forwarding plane) with additional embedded native softwarization functionality. Although separately specified, the planes functionality is not completely independent; key items in each are identified as related to items in the other planes. Each plane substantially uses foundational concepts. However, the planes are sufficiently independent to simplify reasoning about the complete system specification.

Application and Business Service Plane Functionality – It defines and implements the business processes of the services along specific value chains. A service in the 5G context is a piece of software that performs one or more functions, provides one or more APIs to applications or other services of the same or different planes to make usage of those functions, and returns one or more results. Services can be combined with other services, or called in a serialized manner to create a new service. An application in the 5G context is a piece of software that utilizes the underlying services to perform a function. Application operation can be parameterized, for example, by passing certain arguments at call time, but it is meant to be a standalone piece of software; an App does not offer any interfaces to other applications or services.

Multi-Service Orchestration and Management (Sub) Plane Functionality – The functions and interfaces in this plane are used to set up and manage groups of network instances and/or nodes. More specifically, the setup consists of creating/installing/arranging/deactivation/coordinating NFs and interfaces according to the available physical and virtual resources. It also comprises the set of functions associated with the network operations, such as fault management, performance management and configuration management. It further includes Slice – Service Mapper functions, Resources, Domain and Service Orchestration functions, Service Information Management functions and Network Capability Discovery functions. It also includes the lifecycle management of individual network functions and mobile network instances as a whole. In current mobile networks, this role is often performed by the Operations Support System (OSS). The idea is to enable the

creation, operation, and control of multiple dedicated communication service networks running on top of a 5G E2E infrastructure.

Integrated Network Management & Operations (Sub)Plane – It enables the creation, deactivation, operation, control and coordination (orchestration) of dedicated management functions operating on top of a 5G E2E infrastructure; and the collection of resources required for managing the overall operation and coordination of individual network devices. It further includes E2E Network segments management, FCAPS functionality, Monitoring operations, Network Information Management, In-network data and operations processing and Multi domains management operations.

Control Plane Functionality – The collection of functions responsible for controlling one or more network functions. Control Plane instructs network devices, network elements, and network functions with respect to processing elementary data units (packets, frames, symbols, bits, etc.) of the user/data/forwarding plane. The control of (virtual) network functions include Control of Network (Virtual) functions, Control of Orchestration functions, Control of Mobility functions, Cloud Control functions, Mobile Edge Computing Control functions and adaptors to different enforcement functions. The control of (virtual) network functions is generally 5G-applicable, and they are separated from the control and enforcements functions which are network segment-specific. The control plane interacts primarily with the forwarding plane and, to a lesser extent, with the management plane.

Forwarding Plane / Data Plane Functionality – The collection of resources across all network devices responsible for forwarding traffic.

Softwarization Embedded / Native Functionality – It includes triggering, instantiating and running of programming functions in all network elements, network segments, network functions and network slices. Enables the provisioning and operation of software and service networks. It facilitates the operation of end-to-end heterogeneous networking and distributed cloud platforms, including physical and logical resources and devices. It includes functions for designing, implementing, deploying, managing and maintaining network equipment, network components and/or network functions and /or network services by programming. It further includes functions for the provision of software and service networks, application driven network softwarization, programmability of Software Networks, dynamic deployment of new network and management services (i.e. which could be executed in data, control, management, service plane), network capability exposure, and E2E slice provisioning. It includes functions for dynamic programmability of (1) network devices; (2) network (virtual) functions; (3) slices, (4) network services and applications; (5) user plane; (6) control plane; (7) management plane. The software utilizes features such as flexibility and rapidity all along the lifecycle of network equipment/components/services, in order to create conditions that enable the re-design of network and services architectures, optimize costs and processes, allow self-management and bring added value to network infrastructures.

Softwarization functionality affects and changes the other planes functions through a set of Programmability recursive methods and APIs including: (1) allowing the functionality of some of their network elements to be dynamically programmable. The behaviour of network elements and resources can then be customized and changed through such programming interfaces ; (2) enabling the fast, flexible setting-up of new network services, new slice-services, new software networks and new management services by dynamic programmability of the network resources executed as groups of virtual machines in the user plane, control plane, management plane and service plane in all segments of the network; (3) enabling dynamic re-deployment and/or dynamic changes to elasticity characteristics for network services, slice services, software networks and management services.; (4) enabling injection of executable code into the execution environments of network elements in order to create the new functionality at run time; (5) enable trusted third parties (some end users, operators, and service providers) to inject application-specific services (in the form of code) into the network and/or slices. Applications may utilize this network support in terms of optimized network resources and, as such, they are becoming network aware.

I.4 Living list document #4 - Common architecture framework proposed in [b-ITU-T IMT-I-255]

NOTE – Following is the proposed text for IMT-2020 network architecture framework in [b-ITU-T IMT-I-255]. For more details of the proposal, refer to [b-ITU-T IMT-I-255].

I.4.1 General framework and architectural principles

According to [IMT-2020 network requirement], following high level architectural principles are considered for the IMT-2020 network:

- Support network slicing to accommodate various services with typical service categories, i.e. enhanced mobile broadband, massive IoT, ultra-reliability and low latency communications, and autonomous vehicles.
- Minimize access and core network dependencies, design converged core network to adapt various access network.
- Separation of control and user plane functions in core network side, allowing independent scalability and evolution, allowing flexible deployment.
- Design the network function and interconnection with leveraging NFV and SDN technology.
- Support network capability exposure to enable rich value-added services.
- Design uniform network management system which cover both IMT-2020 system and legacy 2G, 3G and LTE networks.

Figure 1 shows the IMT-2020 general framework diagram, which is comprised of 5 different domains and corresponding sub-networks or sub-planes.

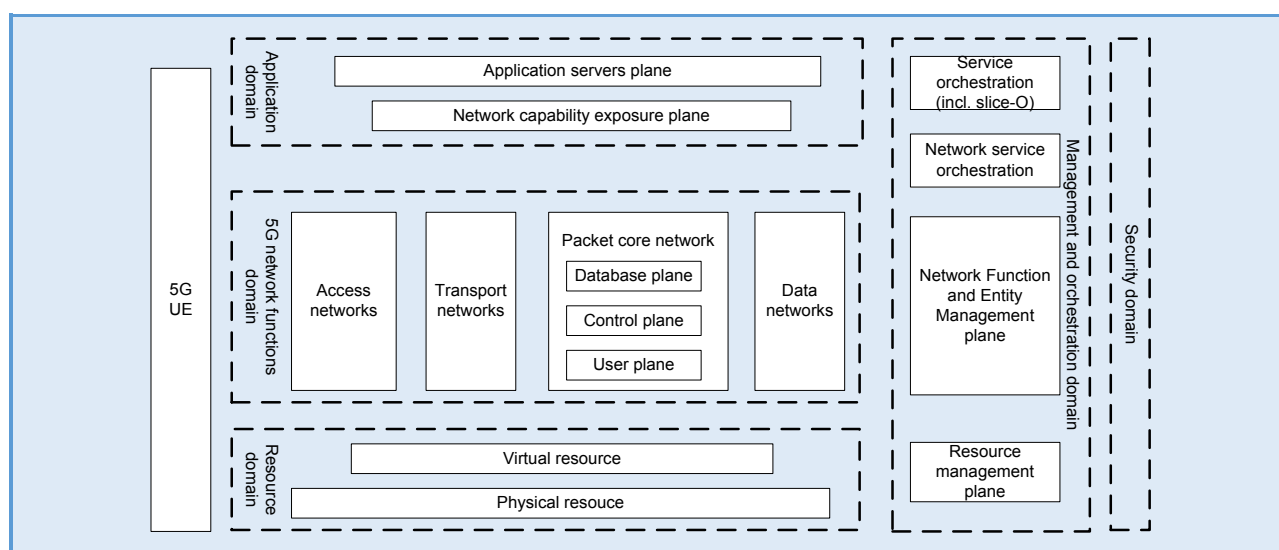


Figure 1 – IMT-2020 general framework diagram

Resource domain: consists of a set of virtual or physical resource (compute, networking, storage) which support running of 5G network functions.

5G Network functions domain: consists of access network (including 5G RAN, evolved LTE RAN, WiFi access network and fixed access network), transport network (including fronthaul, backhaul and backbone/core network), packet core network (including control plane functions, user plane functions and database planes functions), data network (including IMS or service function chains etc), which support 5G service execution.

Management and orchestration domain: consists of resource management plane, network functions and network element management plane, network service orchestration, and service orchestration plane (including network slice orchestration), which support E2E management and orchestration of 5G network and service.

Application domain: consists of network capability exposure plane and application servers plane, which enable various value-added services.

Security domain: consists of security functions which ensure the security of whole 5G system.

Appendix II

Contributors (in Alphabetical Order)

(This appendix does not form an integral part of this Recommendation.)

This is the list of all contributors who submitted any written form of comments or contributions.

–	Alex GALIS	University College London
–	Byung Jun AHN	ETRI
–	Jeong Yun KIM	ETRI
–	Marco CARUGI	NEC
–	Namseok KO	ETRI
–	Peter ASHWOOD-SMITH	Huawei Technologies
–	Shin-Gak KANG	ETRI
–	Taesang CHOI	ETRI
–	Weixing WANG	Nokia Networks

Appendix III

Acknowledgement

(This appendix does not form an integral part of this Recommendation.)

- This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R7116-16-1001, 5G Core Network Technologies Standards).
- This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0132-16-1005, Development of Wired-Wireless Converged 5G Core Technologies).
- This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R7117-16-0127, Development of Standard for Service Adaptive Dynamic Network Slicing).
- This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0115-16-0001, 5GCHAMPION – 5G Communication with a Heterogeneous, Agile Mobile network in the PyeongChang wInter Olympic competition).
- This work was partially supported by the EU H2020 5G PPP projects: 5GEX (“5G Multi-Domain Exchange”; <https://www.5gex.eu>) and SONATA (“Service Programing and Orchestration for Virtualized Software Networks”; <http://sonata-nfv.eu/>).

Bibliography

- [b-3GPP TR 22.861] 3GPP TR 22.861 V14.1.0 (2016), *FS_SMARTER – massive Internet of Things (Release 14)*
- [b-ITU-T IMT-I-225] ITU-T FG IMT-2020 Input Document IMT-I-225 (2016), *Proposed text for clause 5.1 of Framework of IMT-2020 network architecture*
- [b-ITU-T IMT-I-247] ITU-T FG IMT-2020 Input Document IMT-I-247 (2016), *Proposal for IMT-2020 high-level architectural framework*
- [b-ITU-T IMT-I-250] ITU-T FG IMT-2020 Input Document IMT-I-250 (2016), *Revisited IMT-2020 network – Global Pictures & Framework*
- [b-ITU-T IMT-I-255] ITU-T FG IMT-2020 Input Document IMT-I-255 (2016), *Proposal for IMT-2020 framework diagram*
- [b-ITU-T IMT-O-042] ITU-T FG IMT-2020 Output Document IMT-O-042 (2016), *Requirements of IMT-2020 from network perspective*
- [b-ITU-T IMT-O-047] ITU-T FG IMT-2020 Output Document IMT-O-047 (2016), *Network management framework for IMT-2020*
- [b-ITU-T Y.110] Recommendation ITU-T Y.110 (1998), *Global Information Infrastructure principles and framework architecture*





3.

Network softwarization



A close-up photograph of a computer keyboard. In the upper left, a brass key with a rectangular notch is resting on the keys. In the lower left, a red card with the words "SOURCE" and "WARE" in white, pixelated capital letters is partially visible. A person's finger is pressing down on a key to the right of the card. The keyboard keys are dark grey or black. The text "pg dn" is visible on a key in the upper right. The overall image has a technical and digital feel.

Application of network softwarization to IMT-2020

SOURCE
WARE

5G



Table of Contents

1	Scope
2	References
3	Definitions
3.1	Terms defined elsewhere
3.2	Terms defined in this document
4	Abbreviations and acronyms
5	Conventions
6	Network Softwarization for IMT-2020
6.1	Standardization activities for network softwarization
6.2	Open source projects for network softwarization
6.3	Prototyping activities for network softwarization
6.4	Work in-progress research projects
7	Vertical extension of slicing
7.1	Network slicing for FH/BH
7.2	The review of the gaps described in Phase 1 Report and usecases
7.3	Data Plane Programmability
8	Horizontal extension of slicing
8.1	Capability exposure and APIs
8.2	End-to-end slice
9	Application scenario of network softwarization
9.1	Scenario 1: Edge computing
9.2	Scenario 2: ICN on a Slice
9.3	Scenario 3: LTE in a slice
9.4	Scenario 4: Network Slicing
9.5	Scenario 5: Satellite integration in the 5G Ecosystem
Appendix I – LTE in Slice for Softwarization	
I.1	Overview
I.2	Discussion
I.3	Components
I.4	SIM card programming
I.5	Lab pictures
Appendix II – FlexE	
II.1	IP/Ethernet based Mobile Backhaul
II.2	OAM Functions for FlexE
II.3	FlexE-3 Resizing of FlexE Connection
Contributors (in Alphabetical Order)	

1 Scope

This report describes the aspects relevant to the application of network softwarization to IMT-2020, including standardization activities, open source projects, prototyping activities, work in-progress research projects, vertical extension of slicing, horizontal extension of slicing, and application scenario.

2 References

This document refers to the following document in regard to the conceptual IMT-2020 non-radio network architecture, which is depicted in Fig. 5-1 of the following document and understood as commonly applicable to all the deliverables of FG IMT-2020.

[IMT-O-043] Framework of IMT-2020 network architecture.

This document refers to the following document for an understanding of the terms used hereafter. Note that the understanding and the resulting definitions of terms provided in the following document may change in the expected future work of completing it.

[IMT-O-040] Terms and definitions for IMT-2020 in ITU-T

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this document

None.

4 Abbreviations and acronyms

This document uses the following abbreviations and acronyms:

SDN	Software Defined Networking
NFV	Network Virtualization
LINP	Logically Isolated Network Partition

5 Conventions

None.

6 Network Softwarization for IMT-2020

6.1 Standardization activities for network softwarization

6.1.1 Standardization activities at 3GPP SA2

The “Study on Architecture for Next Generation System” (Rel-14) was started at SA2 #112 meeting and officially approved as 3GPP TR 23.799 at the 3GPP TSG SA #70 plenary in December 2015. The latest version (i.e. v1.2.0 November 2016) of the NextGen TR 23.799 can be found on the following links for more detailed information.

- ftp://ftp.3gpp.org/tsg_sa/WG2_Arch/Latest_SA2_Specs/Latest_draft_S2_Specs/
- http://www.3gpp.org/ftp/tsg_sa/WG2_Arch/Latest_SA2_Specs/Latest_draft_S2_Specs/

The objective of the study is to design a system architecture for “5G”, which is called as the next generation mobile network or NextGen aiming to support at least the new RAT(s), the evolved E-UTRA, non-3GPP accesses and minimize access dependencies.

The study is being done based on the following studies on the requirements for the next generation mobile networks which have been carried in 3GPP SA1.

- 3GPP SA1 (FS_SMARTER): TR 22.891
 - Feasibility Study on New Services and Markets Technology Enablers (Rel-14)
- 3GPP SA1 (FS_SMARTER-mIoT): TR 22.861
 - SMATER – Massive Internet of Things (Rel-14)
- 3GPP SA1 (FS_SMARTER-CRIC): TR 22.862
 - SMATER – Critical Communications (Rel-14)
- 3GPP SA1 (FS_SMARTER-eMBB): TR 22.863
 - SMATER – Enhanced Mobile Broadband (Rel-14)
- 3GPP SA1 (FS_SMARTER-NEO): TR 22.864
 - SMATER – Network Operation (Rel-14)

A total of 30 high-level architectural requirements have been derived as a guidance for the architecture study. The following requirements, among others, are directly and indirectly related to network softwarization.

- Support a separation of Control plane and User plane functions
- Leverage techniques (e.g., Network Function Virtualization and Software Defined Networking) to reduce total cost of ownership, improve operational efficiency, energy efficiency, and simplicity and flexibility for offering new services
- Support network slicing
- Support network capability exposure

Based on the requirements, they are also coming up with key issues to be solved and their solutions; as of now (i.e. September 2016), a total of 22 key issues are being discussed. The following key issues are related to network softwarization.

- Support of network slicing
- Network function granularity and interactions between them
- 3GPP architecture impacts to support network capability exposure
- Architecture impacts when using virtual environments

6.1.1.1 Terminologies

Network Capability: Is a network provided and 3GPP specified feature that typically is not used as a separate or standalone "end user service", but rather as a component that may be combined into a telecommunication service that is offered to an "end user".

NOTE – For example, the location service is typically not used by an "end user" to simply query the location of another UE. As a feature or network capability it might be used e.g. by a tracking application, which is then offering as the "end user service". Network capabilities may be used network internally and/or can be exposed to external users, which are also denoted a 3rd parties.

Network Function: In this TR, Network function is a 3GPP adopted or 3GPP defined processing function in a network, which has defined functional behaviour and 3GPP defined interfaces.

NOTE – A network function can be implemented either as a network element on a dedicated hardware, or as a software instance running on a dedicated hardware, or as a virtualised function instantiated on an appropriate platform, e.g. on a cloud infrastructure.

Network Slice Template (NST): is a logical representation of the Network Function(s) and corresponding resource requirements necessary to provide the required telecommunication services and network capabilities.

Network Slice Instance (NSI): is an instance created from a Network Slice Template (NST).

Network Slice: is a concept describing a system behaviour which is implemented via Network Slice Instance(s).

NextGen Core Network: A core network specified in the present document that connects to a NextGen access network.

NextGen Access Network (NG AN): It refers to a NextGen RAN or a Non-3GPP access network and interfaces with the next generation core.

NextGen System (NG System): It refers to NextGen system including NextGen Access Network (NG AN) and NextGen Core.

6.1.1.2 Key issues and solutions

6.1.1.2.1 Key issue: Support of network slicing

Network slicing is studied to provide operators flexibility to create networks customised according to diverse requirements from the perspectives of functionality, performance, isolation, etc. They refer to the definitions and terminologies on Network Slicing from 3GPP SA1 and NGMN while they still open to other industry organisations for more inputs.

Among others, discussion through several meeting includes sharing of network functions among different network slices, selection of network slice of a UE, how to enable a UE to simultaneously access multiple network slices, etc. Please refer to Clauses 5.1 and 6.1 in NextGen TR 23.799 for the detailed non-exhaustive list of solutions for this key issue.

Table 6.1.1-1 – Work tasks for network slicing

Work Task ID	Work Task(s)	Work Task Description
NS_WT_#1	Network Slice Instance Selection and Association	1) Initial network slice instance selection to support UE's service establishment and re-selection to support UE mobility and other scenarios that are TBD, NOTE – More scenarios beyond the mobility need to be identified that may trigger network slice instance re-selection. 2) Network slice instance identification, 3) authorization for UE association with network slice instance 4) Network assistance information support for UE network slice instance association with corresponding PLMN
NS_WT_#2	Network Slicing Isolation	1) Security isolation 2) Resource isolation 3) OAM support isolation (e.g., Usage and Fault isolation etc.) NOTE – Whether all items listed here are within the scope of SA2 is FOR FURTHER STUDY.

Table 6.1.1-1 – Work tasks for network slicing

Work Task ID	Work Task(s)	Work Task Description
NS_WT_#3	Network Slicing Architecture	1) Identifying impacted network functions and interfaces to support one or more network slice instances on top of a shared RAN and a shared infrastructure. 2) Identifying the common functions (if any) that need to be available in the core network and/or RAN to enable network slicing 3) Identifying the approach to enable UE to associate with multiple slices simultaneously.
NS_WT_#4	Network Slicing Roaming support	1) Determination what visiting and home Network Function(s) are required to support roaming.
NS_WT_#5	Network Slicing terminology & definitions	1) If Network Slice Instance is agreed to apply E2E system, then, we should consider new terminology for Access slice instance and Core slice instances.

The following bullets are the current status of agreements on the network slicing (source: SA2 #118)

- 1) The network slice is a complete logical network (providing Telecommunication Services and Network Capabilities) including AN and CN. Whether RAN is sliced is up to RAN WGs to determine.
 - a) AN can be common to multiple network slices.
 - b) Network slices may differ for features supported and Network Functions optimisations use cases.
 - c) Networks may deploy multiple Network slice instances delivering exactly the same optimisations and features as per but dedicated to different groups of UEs, e.g. as they deliver a different committed service and/or because they may be dedicated to a customer.
- 2) A UE may provide network slice selection assistance information (NSSAI) consisting of a set of parameters to the network to select the set of RAN and CN part of the network slice instances (NSIs) for the UE.
 - a) The NSSAI can have standard values or PLMN specific values. The NSSAI (which is used to select the CCNF) is a collection of SM-NSSAIs (see sub-bullet 2c for the SM-NSSAI definition), each allowing the network to select a particular slice.

NOTE – whether a single value which is a representation of a collection of the SM-NSSAIs could also be used as NSSAI is to be assessed in normative work.

- b) The UE may store a Configured and/or Accepted NSSAI per PLMN.

The Configured NSSAI is a NSSAI configured by default in a UE to be used in a PLMN before any interaction with the PLMN ever took place.

The Accepted NSSAI is the NSSAI used by the UE after the PLMN has accepted an Attach Request from the UE. The Attach Accept message includes the Accepted NSSAI. The accepted NSSAI may be updated by MM procedures (see below).

- c) If the UE has been provided a Configured or Accepted NSSAI for the ID of the PLMN that the UE accesses, the UE provides this NSSAI in RRC and NAS as described below.

Each SM-The NSSAI in the NSSAI may include:

- Slice/Service type (SST), which refers to the expected network behaviour in terms of features and services.
- Information that complements the Slice/Service type(s) to allow further differentiation for selecting from the potentially multiple network slice instances that all comply with the indicated slice/service type(s). This information is referred to as Slice Differentiator (SD).

NOTE – The abbreviation SM-NSSAI does not imply it is used only in SM procedures nor that it only carries SM information. E.g. it may be used to help in AMF selection as part of the NSSAI.

An SM-NSSAI can include both a Slice/Service Type and Slice Differentiator or just the Slice/Service Type.

The RAN routes the initial access to a CCNF using the NSSAI (see bullet 4 for CCNF definition).

- d) If the UE did not receive any Accepted NSSAI for the ID of the PLMN that the UE accesses, the UE provides the Configured NSSAI in RRC and NAS, if the UE has been provided with a Configured NSSAI. The RAN uses the NSSAI for routing the initial access to a CCNF. If the UE doesn't store any NSSAI (Accepted or Configured) for the ID of the PLMN that the UE accesses, the UE provides no NSSAI in RRC and NAS, and the RAN sends NAS signalling to a default CCNF (see bullet 4 for CCNF definition).
- e) After (initial) slice selection, upon successful attachment the UE is provided with a Temp ID that is provided by the UE in RRC during subsequent accesses to enable the RAN to route the NAS message to the appropriate CCNF, as long as the Temp ID is valid. In addition the serving PLMN may return an Accepted NSSAI that the UE stores for the PLMN ID of the serving PLMN. The Accepted NSSAI includes the SM-NSSAI values of the slices the UE is accepted to use by the network.
- f) For a "Service Request" the UE is registered/updated and has a valid temp ID, which is sufficient in the RAN to route the request to the serving Common CP NF. It is assumed that the slice configuration doesn't change within the UE's registration areas.
- g) For enabling routing of a TA update request the UE includes always Accepted NSSAI and a complete Temp ID in RRC. If the RAN is aware of and can reach the CCNF which is associated with the Temp ID, then RAN forwards the request to the CCNF. Otherwise, RAN selects a suitable CCNF based on the Accepted NSSAI and forwards the request to the selected CCNF. If the RAN is not able to select a CCNF based on the Accepted NSSAI, then the request is sent to a default CCNF.
- h) The UE shall include in a PDU session establishment Request a SM-NSSAI which, shall enable the selection of an SMF, alongside the DNN.
- i) In order for RAN to select a proper resource for supporting network slicing in RAN, RAN may need to be aware of the network slices. How the RAN is aware of this is up to RAN WGs to determine.
- 3) If a network deploys network slicing, then it may use UE provided network slice selection assistance information to select a network slice. In addition, the UE capabilities and UE subscription data may be used.
- 4) A UE may access multiple slices simultaneously via a single RAN. In such case, those slices share some control plane functions, e.g. AMF and Network Slice Instance Selection Function. These common functions are collectively identified as CCNF (Common Control Network functions).
- 5) The CN part of network slice instance(s) serving a UE is selected by CN not RAN.
- a) The NSSF is one function of CCNF, which is a set of NFs including the AMF and the NSSF, and is used to select the NSI for the UE.

- 6) With reference to Annex D: move forward with Group B type of solution in rel-15 (Group C is subsumed under Group B). Group A is not pursued in R15.
- 7) It shall be possible to handover a UE from a slice in NGC to a DCN in EPC. There is not necessarily a one-to-one mapping between slice and DCN.
- 8) The UE need to be able to associate an application with one out of multiple parallel established PDU sessions. Different PDU sessions may belong to different slices.
- 9) The UE may cause the network to change the set of network slices it is using by submitting in an MM procedure the value of a new NSSAI. The final decision is up to the network.

The network, based on local policies, subscription changes and/or UE mobility, can change the set of network slices that are being used by a UE by providing the UE a notification of Accepted NSSAI change. This then triggers a UE initiated MM procedure including in RRC and NAS Signalling the new value of the new accepted NSSAI the network has provided.

Change of set of slices used by a UE (whether UE or Network initiated), may lead to CCNF change subject to operator policy.

NOTE – The scenarios and which MM procedures to use when such UE triggered network slice change is to be used is to be determined during normative phase.

NOTE – Changing the set of network slices accessible by the UE will result in termination ongoing PDU sessions with the original set of network slices if these slices are no longer used (Some slices are still retained, potentially).
- 10) The Network subscription data includes information about what slices a UE is allowed to access.
 - a) The Information in the subscription includes both SST and SD information for each slice the UE is allowed to access (SD information is present if applicable for a slice). This is also the SM-NSSAI for the slice.
 - b) The subscription data include information on whether a slice is a default slice (i.e. the UE shall be using this slice when it is attached to the network). When a UE is initially attached to the network without providing any NSSAI, the CN should use a default NSSAI, composed of the SM-NSSAI values stored in the UE subscription with a flag indication they are to be considered default slices, to determine the default initial network slice(s set) to serve the UE.
- 11) During the initial Attach procedure, in case the CCNF to serve the UE is to be redirected, the CCNF, which first receives the initial Attach request, shall be able to redirect the initial Attach request to another CCNF via the RAN or via direct signalling between the initial CCNF and the target CCNF. The redirection message sent by the CCNF may include an information about the new CCNF to serve the UE.
- 12) For a UE that is already registered, the system shall support a redirection of a UE from its serving CCNF to a target CCNF.
 - a) The operator policy determines whether redirection between CCNFs is allowed.
 - b) When the network decides to redirect the UE due to NSSAI change, the network send the updated/new NSSAI to the UE using an MM procedure and an indication for the UE to initiate an MM procedure with the updated/new NSSAI. The UE then initiates an MM procedure with the updated/new NSSAI.
- 13) The network operator may provision the UE with network slice selection policy (NSSP). The NSSP includes one or more NSSP rules each one associating an application with a certain SM-NSSAI. A default rule may also be included which matches all applications and contains a default SM-NSSAI. The UE uses the NSSP to associate UE applications with SM-NSSAIs. When a UE application associated with a specific SM-NSSAI requests data transmission, then:
 - a) If the UE has one or more PDU sessions established with this specific SM-NSSAI, the UE routes the data of this application in one of these PDU sessions, unless other conditions in the UE prohibit the use of these PDU sessions. If the application provides a DNN, then the UE considers also this DNN to determine which PDU session to use.

- b) If the UE does not have a PDU session established with this specific SM-NSSAI, the UE requests a new PDU session with this SM-NSSAI and with the DNN that may be provided by the application.
- 14) The CCNF selects an SMF in a network slice instance based on SM-NSSAI, DNN and other information e.g. UE subscription and local operator policies. The selected SMF establishes a PDU session based on SM-NSSAI and DNN.
- 15) For roaming scenarios, the slice specific network functions beyond CCNF in VPLMN and HPLMN are selected based on the SM-NSSAI provided by the UE as following.
 - a) If standard SM-NSSAI is used, then selections of slice specific NF instances are done by each PLMN based on the standard SM-NSSAI.
 - b) If non-standard SM-NSSAI is used, the VPLMN maps the SM-NSSAI of HPLMN to a SM-NSSAI of VPLMN based on roaming agreement (including mapping to a default SM-NSSAI of VPLMN). The selection of slice specific NF instance in VPLMN are done based on the SM-NSSAI of VPLMN, and the selection of any slice specific NF instance in HPLMN are based on the SM-NSSAI of HPLMN.

NOTE – whether the SM-NSSAI can be directly used as input to network function instance selection procedure defined in KI 7# should be decided in normative phase.

Regarding support of network slicing, SA2 scope in TS phase 1 is only limited to NextGen Core Network. Whether RAN is sliced is up to RAN WGs to determine.

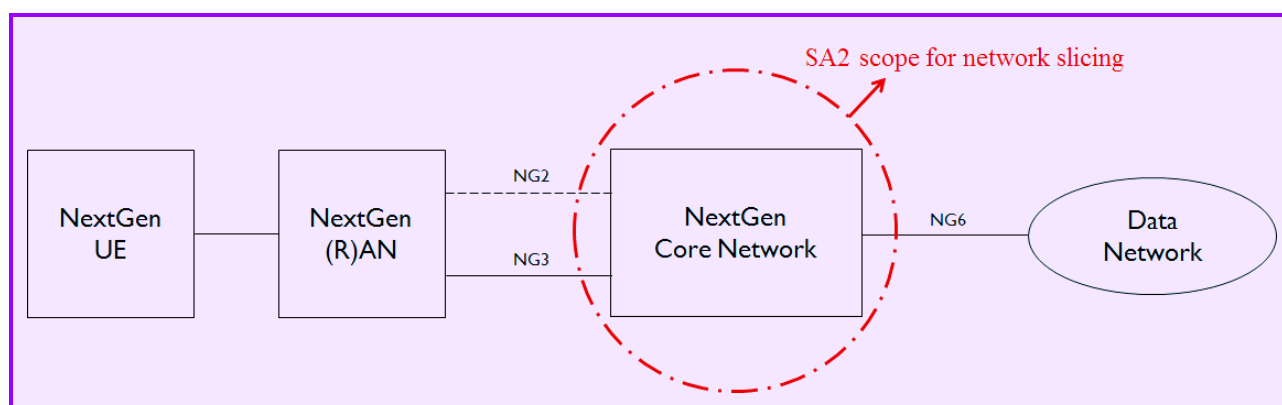


Figure 6.1.1-1 – SA2 scope for support of network slicing in TS phase 1

6.1.1.2.1.1 Solution 1.1: Slice selection solution update

(Source: Nokia, Alcatel-Lucent Shanghai Bell)

This is a solution for key issue on support for network slicing. In this solution the concept of Non-Autonomous Core Network Slice is introduced, whereby a UE related NG2 and NAS signalling are handled by a common Frontend as shown in Figures 6.1.1-1. Figure 6.1.1-1 is depicting the concept of Non-Autonomous Network Core network Slice. A Non-Autonomous Core network slice is sharing with other slices for the same UE the NG2 and NAS signalling handling, supported by a common front end for the UE.

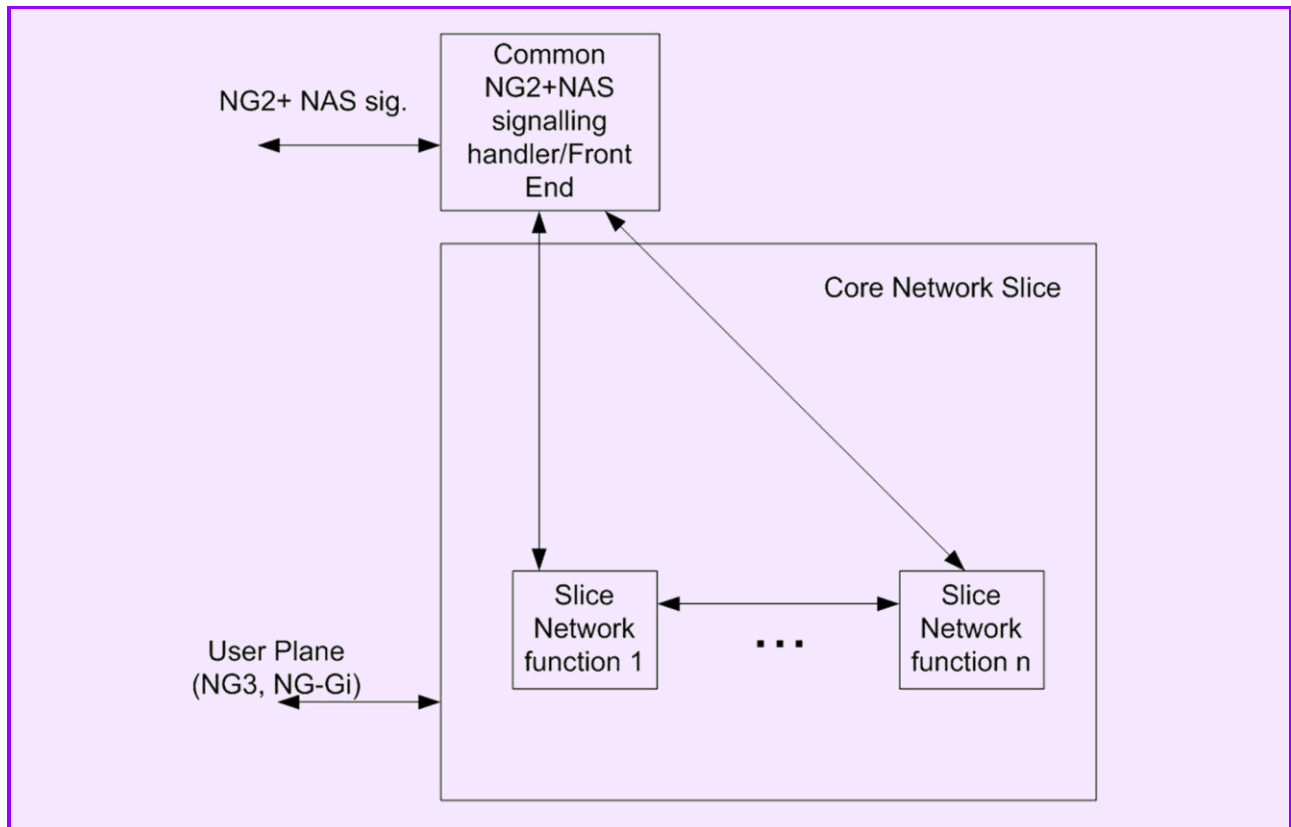


Figure 6.1.1-2 – Non-Autonomous Core Network Slice concept

With the Non-Autonomous Core network Slice concept, once the UE is assigned to a Frontend during the attach procedure, all the signalling is directed to this frontend based on the UE Temporary ID.

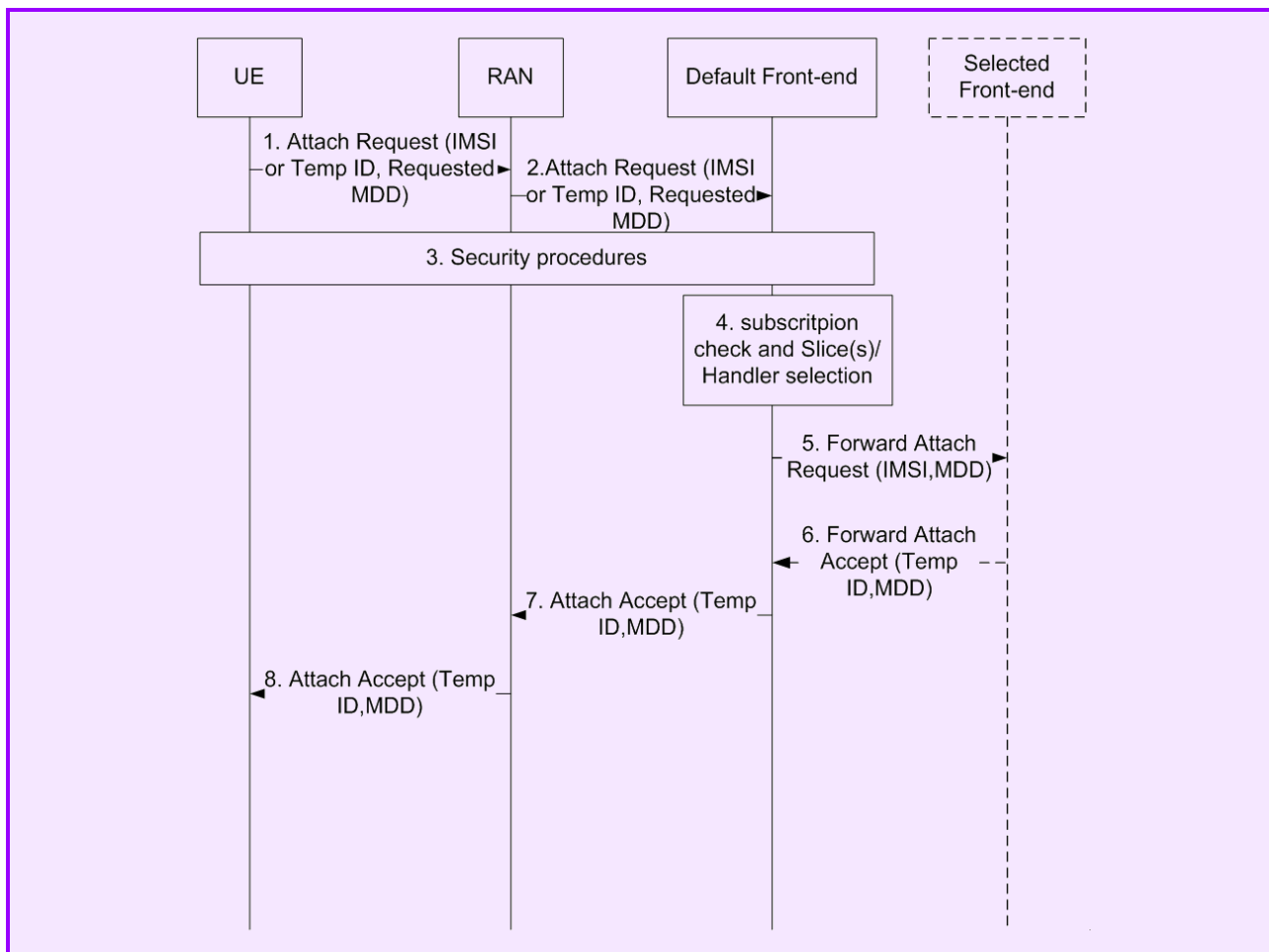


Figure 6.1.1-3 – Initial access

- 1) The UE sends an attach Request including the IMSI if a Temporary ID for the UE is not available. If the Temporary ID is available, the UE includes the RRC layer message at least the Routing field of the Temporary ID, so that the RAN can route the message to a suitable handler in the core.
If the UE requests an initial set of slices, it can do so by indicating a Requested MDD. The Requested MDD may also be included in the RRC layer to further enhance the routing of the Attach in the event the Temporary ID was not available to the UE or the Temporary ID was not assigned by the same PLMN the UE is in or not belonging to the network. The MDD in the RRC layer may also be included to enable the access to a suitable RAN slice.
- 2) The RAN forwards the Attach Request to the Core based on the routing criteria outlined in step 1. If the IMSI is present a default handler is selected. Otherwise the Front end (i.e. the per UE common NG2+NAS Signalling Handler) associated to the Temporary ID is used if available in the Serving PLMN. If not a default front end is used. A UE should, based on configuration, not attach with a Temporary ID that does not belong to the current PLMN.
- 3) The NAS Handler may execute security procedures.
- 4) If the UE is successfully validated, its subscription data is checked and the Handler decides the initial set of slices the UE can use based on an evaluation of the Requested MDD, subscribed MDD and UE capabilities. The following applies:
 - If the UE did not provide the Requested MDD, the network assigns the UE to the default slice(s)
 - If the UE did provide the Requested MDD, the network assigns the UE to the slices the UE is authorized to use among the requested slices
 - If some Default Slice was missing from the requested MDD, the UE is also assigned to these slices

- 5) If the UE is not suitably handled by the (Default) Front-end where the Attach Request was routed to, this front end requests to assign the UE to a new front end that is more optimal (or less loaded) for the selected slices. Then, it forwards the attach request to it with an indication it is a forwarded attach and the IMSI and MDD indicated are respectively validated and already reflecting the Slice Assignment at step 4. If not this step is skipped and the procedure continues from step 7.
- 6) The Selected front end binds itself to the selected slices for the UE and then sends back the Forwarded Attach Accept message with Temporary ID and the accepted MDD for subsequent Usage by the UE.
- 7) If the steps 5 and 6 were executed, the (Default) Handler sends to the RAN the Attach Accept in a NAS message with the content copied from the message in step 6. Otherwise the (Default) Handler binds itself to the selected slices for the UE and then sends Attach Accept message with Temporary ID, the accepted MDD for subsequent Usage by the UE. The (Default) Handler includes the MDD in the NG2 transport.
- 8) The RAN forwards the Attach Accept received in step 7 to the UE

6.1.1.2.1.2 Solution 1.2: UE slice association/overload control procedure

(Source: Huawei, HiSilicon)

In this solution it is assumed that any slicing of a PLMN is not visible to the UEs at the radio interface. So in this case, a slice routing and selection function is needed to link the radio access bearer(s) of a UE with the appropriate core network instance. The solution is comparable to what is introduced with the DÉCOR feature. The solutions do not make any assumption on any potential RAN internal slicing. The main characteristics is that the RAN appears as one RAT+PLMN to the UE and any association with network instance is performed network internally, without the network slices being visible to the UE.

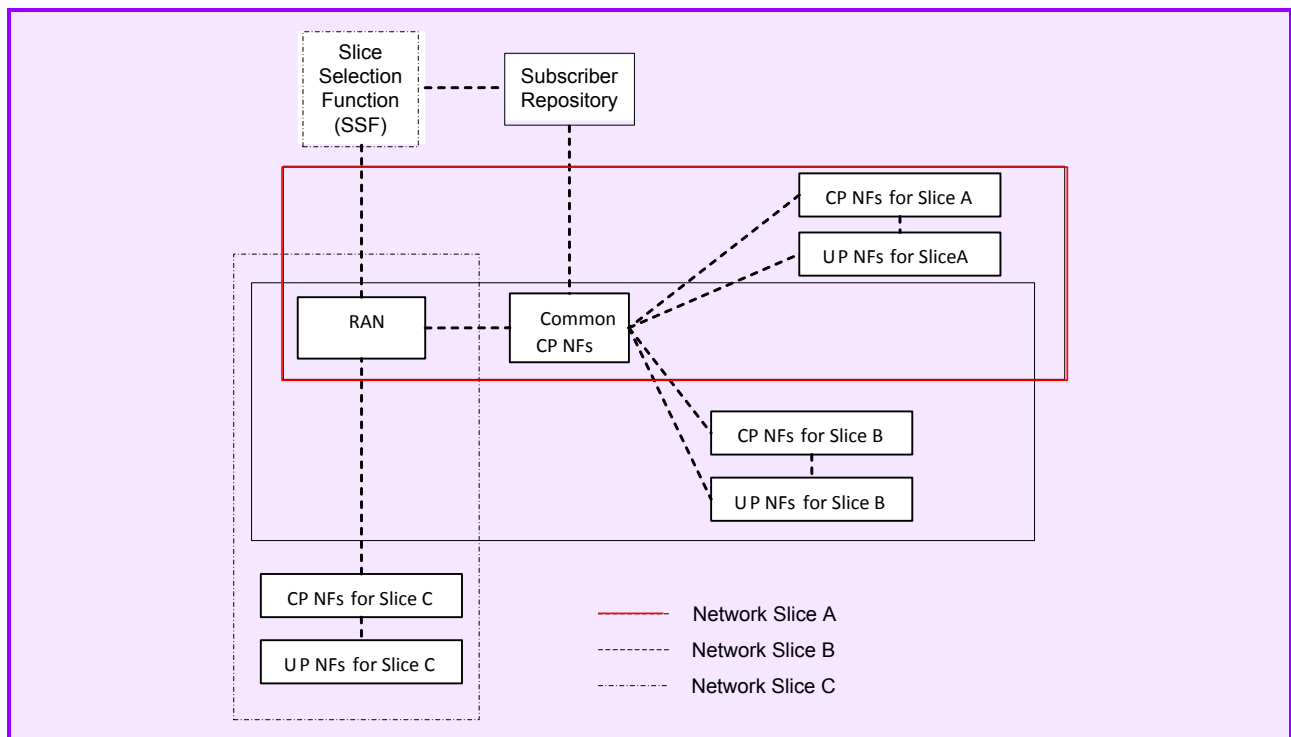


Figure 6.1.1-4 – Control plane interfaces for network slicing with common and slice specific functions

Each network slice instance has a network slice specific instance ID (NSI-ID). When there are common CP NFs, the NSI-ID is a combination of a common CN NF ID and a slice specific ID.

6.1.1.2.2 Key issue: Network function granularity and interactions between them

A network generation operator's network is expected to consist of multiple physical and/or virtual network functions to support diverse service requirements. To achieve flexibility on that network environment, the next generation system capabilities should be supported such as dynamic deployment of network functions and function re-usability based on the architecture principles that allow flexible network function deployment, ease of interfacing, flexible chaining, co-location of network functions.

Please refer to Clauses 5.7 and 6.7 in NextGen TR 23.799 for the detailed non-exhaustive list of solutions for this key issue.

Table 6.1.1-2 – Work tasks for interconnection NF

Work Task ID	Work Task(s)	Work Task Description
INF_WT_#1	Network function interconnection reference model	<ol style="list-style-type: none"> 1. State assumption and/or applicability of the proposed reference model 2. Depict how the network functions are connected using the proposed reference model 3. Identifying a mechanism for a NF instance to interconnect to its peer NFs instances (e.g., via provisioning, selection and discovery etc.)
INF_WT_#2	Sample call flow	<ol style="list-style-type: none"> 1. Describe sample call flows, e.g. network function A sending a req/rsp to network function B

Interim agreements for Key issue #7 “Function Granularity and Interconnection of them” are as follows:

- 1) Any two NFs interacts with each other directly while avoiding the functional and signalling impact on unrelated NF.

NOTE 1 – This does not preclude to pass information via a third NF if two NFs do not interact directly, e.g. if MM received subscription information from SDB then it can pass it to SM if there is an interaction between MM and SM (e.g., during PDU connection establishment procedure).

- 2) In order to facilitate utilization of the capability (s) of one NF the capability (s) of NFs are exposed as a service to other NF, wherever applicable, (e.g., by following the guidelines defined in Annex E). As such the NF provides a service based interface to other NFs.

NOTE 2 – It is expected that SA2 will specify the end to end signalling flow and then deduce the services and functionalities that one NF supports, and CT WGs define the data model of service interface, i.e. information elements included in service interface. For more detail refer to Annex X.

NOTE 3 – To support different variants of a service and to enable the invoking NF to discover the expected service, the service need be uniquely identified.

- 3) Network functions within the NG Control plane unless determined otherwise during the normative phase, shall exhibit service based interfaces for services that can be re-used by multiple network functions. This will be evaluated on a case by case basis when specifying the procedure. The NG1, NG2, NG4 interface are not considered to support the service based interface.
- 4) The NF selection and discovery shall be supported to enable NF selection and discovery, including:
 - The NF selection and discovery function maintains the function profile of the deployed NF instances, e.g. the type of the NF, network slice related information which the NF belongs to.
 - When deploying/removing one NF instance, the information of the NF instance is updated.
 - One NF shall be able to utilize the type of the NF (e.g., SMF, PCF, etc.) and other service parameters (including network slice related parameters) to discover the expected NF instance (s), and the NF selection and discovery function provides the IP address or the FQDN of NF instance(s) to the NF.

NOTE 4 – The network slice related information/parameters are determined in the key issue #1.

NOTE 5 – whether it utilizes the NF Repository function or an enhancement of the DNS server to reach this functionality is left for CT WG to determine.

- 5) To support "stateless" NFs (where the "compute" resource is decoupled from the "storage" resource that stores state as opaque data), 3GPP will specify (possibly by referencing) interfaces from NFs to a data storage function. NFs may use data storage function to store opaque data.
- 6) Exposure of information as a service is supported as follows:
 - Network functions may expose structured information (e.g., UE related information) to other network functions as a capability. Exposure of information can trigger actions in the consuming NF.

NOTE 6 – What information is to be exposed will be determined during the normative phase.

- A network function may use information exposed by other network functions.
- A network function may expose information that it received from another NF.

The Network Exposure Function receives information from other network functions (based on exposed capabilities of other network functions). It may store the received information as structured data using a standardized interface to a data storage network function (interface to be defined by 3GPP). The stored information can be "re-exposed" by the NEF to other network functions and used for other purposes such as analytics.

6.1.1.2.2.1 Solution 2.1: IRF based network function interconnection model solution update

(Source: Cisco)

This solution is applicable only to the interconnection of the network functions. The network function's definition and its functionalities are assumed to be defined by solutions to other key issues.

Assuming that only one of the control plane access network function is required to interface with only one of the control plane core network function, the interconnection model between them can be point-to-point interface based, e.g. control plane core network function 1 interfaces with control plane access network function 1 over a point-to-point interface between them. However, if it is decided to allow multiple control plane access network functions to interface directly with multiple control plane core network functions then the principles of this solution can be extended to interconnect control plane access network and core network functions as well.

Similarly, it is assumed that the user plane core network functions are required to interface only with their respective control plane core network functions. And hence the interconnection model between them can be point-to-point interface based. However, if it is decided to allow any control plane function to interface with any user plane function then the principles of this solution can be extended to interconnect control plane and user plane core network functions as well.

The solution assumes that multiple control plane core network functions are required to interact with multiple other control plane core network functions in the next generation core network architecture.

NOTE – Unless explicitly specified, the term "network function" refers to "control plane core network function" in this solution. Just for the sake of understanding the "control plane core network functions" in the EPC context are MME, control plane of PGW, PCRF, HSS, control plane of TDF, TSSF, RCAF, SCEF etc.

This solution is applicable only to the interconnection of the network functions. The network function's definition and its functionalities are assumed to be defined by solutions to other key issues.

Assuming that only one of the control plane access network function is required to interface with only one of the control plane core network function, the interconnection model between them can be point-to-point interface based, e.g. control plane core network function 1 interfaces with control plane access network function 1 over a point-to-point interface between them. However, if it is decided to allow multiple control plane access network functions to interface directly with multiple control plane core network functions then the principles of this solution can be extended to interconnect control plane access network and core network functions as well.

Similarly, it is assumed that the user plane core network functions are required to interface only with their respective control plane core network functions. And hence the interconnection model between them can be point-to-point interface based. However, if it is decided to allow any control plane function to interface with any user plane function then the principles of this solution can be extended to interconnect control plane and user plane core network functions as well.

The solution assumes that multiple control plane core network functions are required to interact with multiple other control plane core network functions in the next generation core network architecture.

NOTE – Unless explicitly specified, the term "network function" refers to "control plane core network function" in this solution. Just for the sake of understanding the "control plane core network functions" in the EPC context are MME, control plane of PGW, PCRF, HSS, control plane of TDF, TSSF, RCAF, SCEF etc.

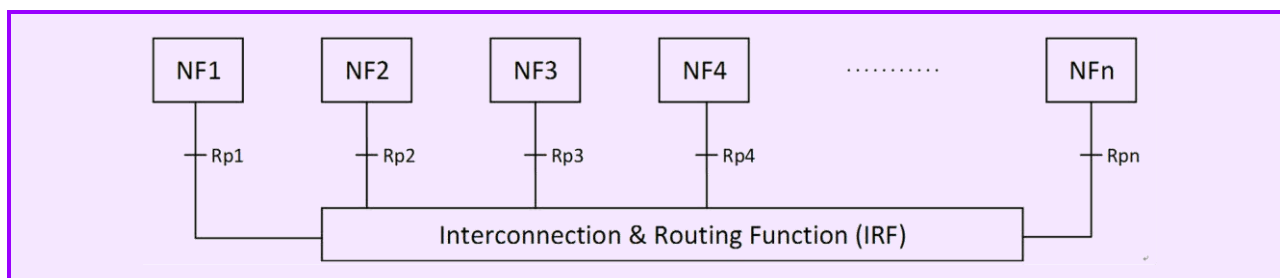


Figure 6.1.1-5 – Non-roaming reference model for the interconnection of network functions

Below is a sample call flow of the management of the binding between the UE's session and its corresponding serving NF, at the IRF.

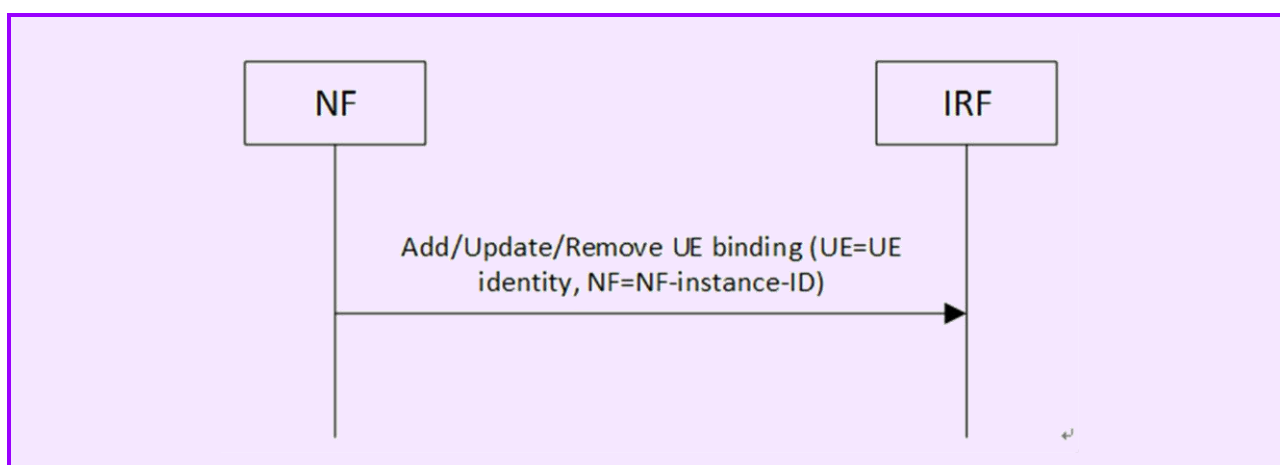


Figure 6.1.1-6 – Sample call flow of UE-NF binding management at IRF

- 1) When a UE's session is created at an NF, e.g. during the procedures such as attach, new PDU session establishment, relocation etc., the NF updates the IRF by sending "Add UE binding" message. The IRF creates new binding in its binding repository.

For updating the existing binding, the NF sends "Update UE binding" message to the IRF. Correspondingly, the IRF updates the binding repository. This could take place if the NF changes its instance for an existing UE's session, e.g. due to scale-in, scale-out or restoration feature.

When the existing UE's session is released by the NF, e.g. during relocation or PDU session release procedures, the NF sends "Remove UE binding" message to the IRF. The IRF clears its binding repository for the UE's session.

NOTE 1 – The above messages can be sent independently or by piggybacking on other relevant messages, e.g. While sending "PDU session establishment" message to NF2, NF1 can piggyback "Add UE binding" message for creating binding between the UE's session and NF1's serving instance, at the IRF.

6.1.1.2.2 Solution 2.2: NFs discovery solution

(Source: Huawei, HiSilicon, China Mobile)

One general model for the interconnection of CP NFs is depicted as the following figure:

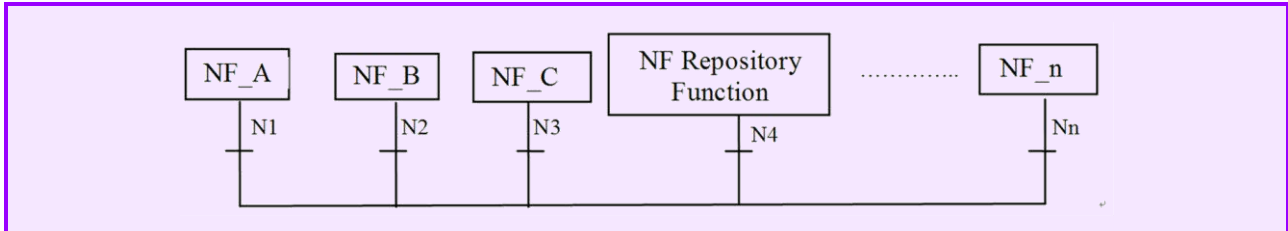


Figure 6.1.1-7 – The general model of the interconnection of CP NFs

The high level principles are utilized in the general model of the interconnection of CP NFs

- The NF (Function Consumer) determines the function type of the NF (Function Provider) that it needs to access;
- The NF (Function Consumer) shall be able to obtain the instance of the NF (Function Provider) after performing NF discovery;
- The NFs (Function Consumer) shall be able to obtain the function of the NF (Function Provider) via the same interface provided by the NF (Function Provider);
- The general communication protocol is used to transfer the message between the NF (Function Consumer) and the NF (Function Provider);
- The NF (Function Provider) authorizes the functionality that one NF (Function Consumer) can access per UE granularity;
- The interface which one NF provides its functionality to other NFs is not changed per the interconnected NFs.

The NF discovery procedure in one PLMN is depicted in the following figure:

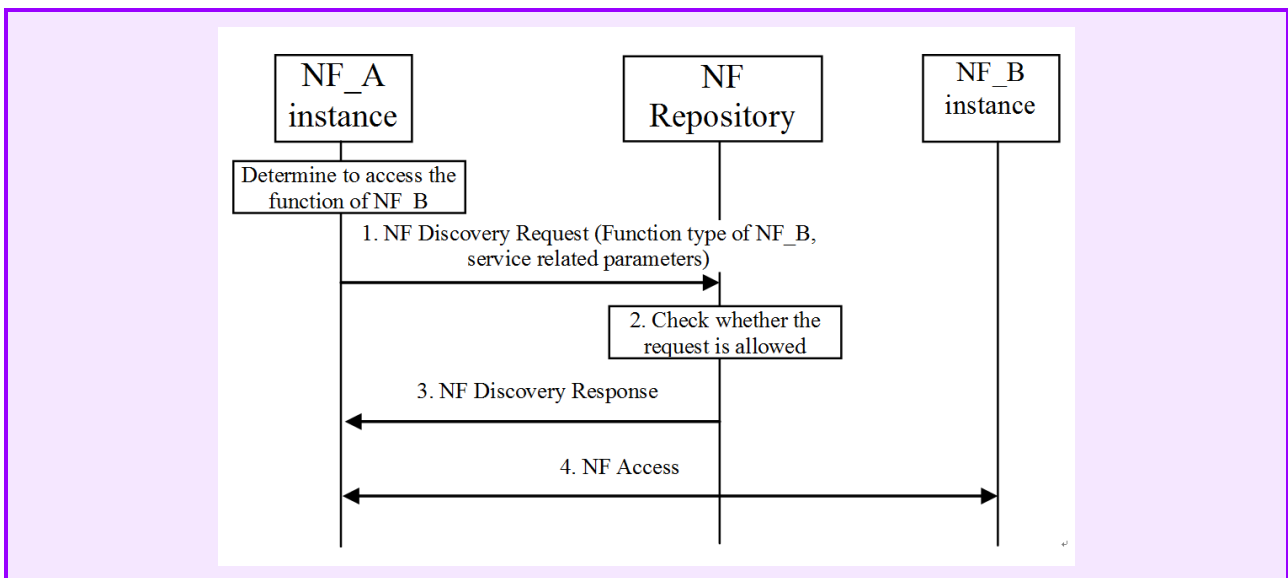


Figure 6.1.1-8 – NF discovery procedure in one PLMN

- 1) Per user's request the NF_A needs to access the functionality provided by the NF_B. NF_A sends NF Discovery Request including the Function Type of NF_B and/or the service related parameters, e.g. Authentication type in case that NF_B is Authentication NF and handles one special type of Authentication mechanism. The service related parameters help to select the special capability NF. The Function type is used to discover the related NF instance.
- 2) NF Repository Function check if NF_A is authorised to access NF_B based on criteria of the NF permission list.
- 3) If the request is allowed, NF Repository Function provides the candidate of NF_B's instance to NF_A by NF Discovery Response message based on i.e. the load level of NB's instance. Further NF Repository Function can store the discovery request of NF_A, and it notices the alternative instance of NF_B to NF_A when detecting that the original instance of NB_B can't continue to provide its function.
- 4) NF_A access the NF_B instance based on the interface supported by NF_B, and the generic communication protocol is used to transfer the messages between NF instances.

6.1.1.2.2.3 Solution 2.3: NF communication solution

(Source: ZTE)

This section addresses key issue on Network Function interconnection. In the following description, the Network Function refers to Network Function Instance.

The Network Function Repository Function stores the local Network Function information, including the IP address of the Network Function, Network Function type. The Network Function doesn't query the target Network Function from DNS server but instead from NF repository Function. When Network Function is instantiated, it sends registration to NF Repository Function. When the Network Function is tear down, it sends deregistration to NF Repository Function. When Network Function needs to communicate with the other Network Function, it asks the IP address of the target Network Function from the NF Repository Function. No interconnection between the NF Repository Functions.

If NFs are registered in the same Network Function Repository, direct communication is used. See the following figure for direct communication. The NGx/NGy/NGz interface is per Network Function pair.

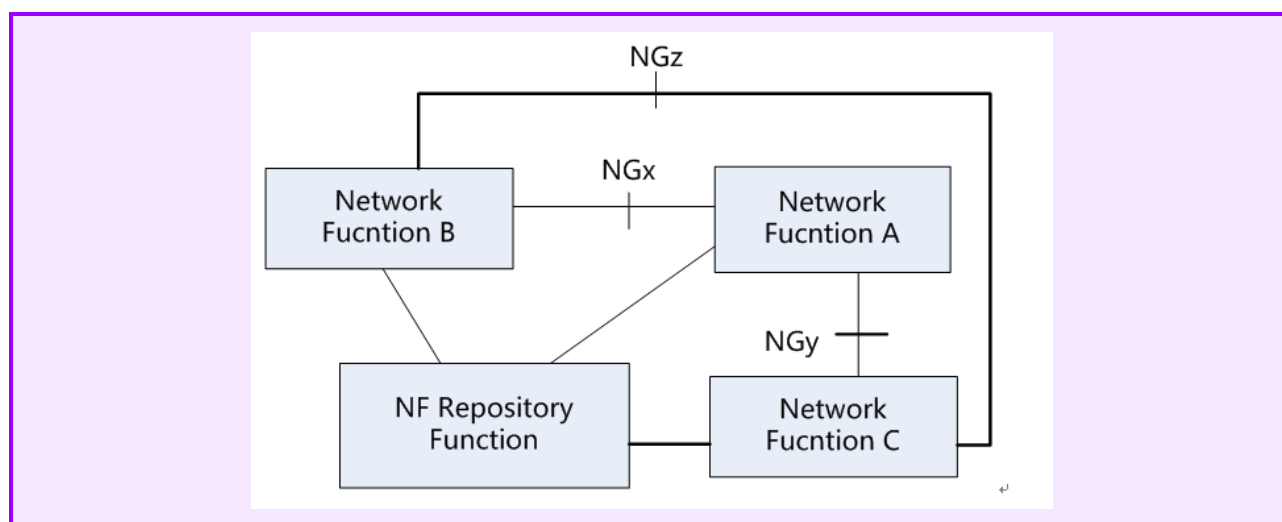


Figure 6.1.1-9 – Direct NF communication

The following is the message flow example to establish direct connectivity between two Network Functions.

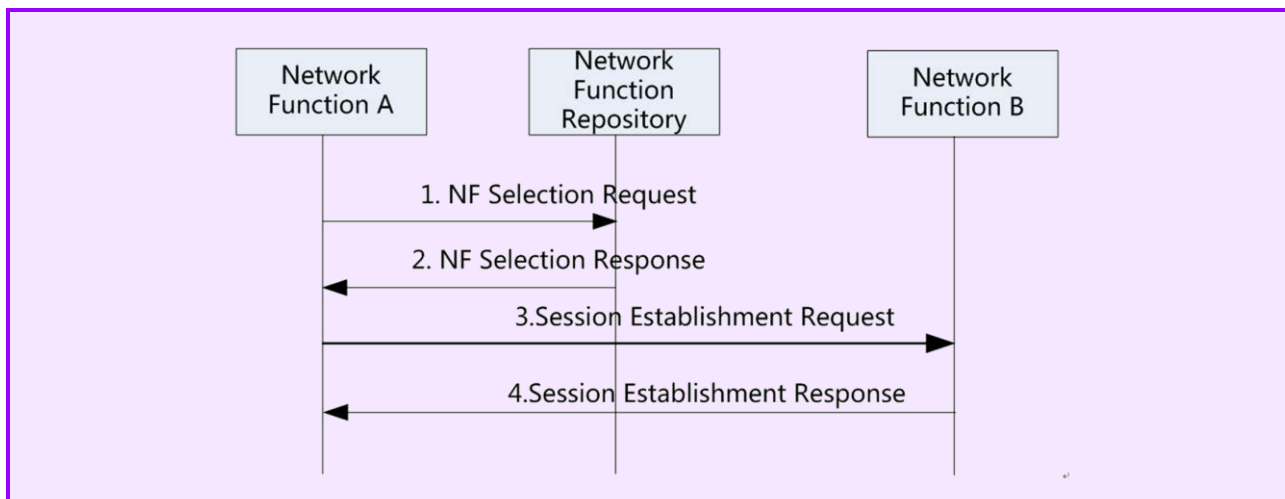


Figure 6.1.1-10 – Message flow to establish session between two Network Functions

- 1) The Network Function A decides to establish a new session with Network Function B. It sends Function Discovery Request message to Network Function Repository Function, including the Requested Network Function Type.
- 2) The Network Function Repository Function determines if a local Network Function B should be selected or not. If local Network function B is selected it sends Function Discovery Response message including a routable IP address of Network Function B. If the Network Function Repository Function decides not to select a local Network Function B it sends Function Discovery Response message including a routable IP address of Interconnection Function A.
- 3) If the Network Function Repository Function response with a routable IP address of the Network Function B, it sends Session Establishment Request message to Network Function B directly, including necessary information for session establishment.
- 4) The Network Function B establishes the session context per request and sends Session Establishment Response to Network Function A directly.

6.1.1.2.3 Key issue: 3GPP architecture impacts to support network capability exposure

The network capability exposure function is to allow 3rd party/UE to access information regarding services provided by the network (e.g., connectivity information, QoS, mobility, etc.) and to dynamically customize the network capability for different diverse use cases within the limits set of functions by the operator. The next generation system should provide suitable access/exchange of network/connectivity information (e.g., via APIs) to the 3rd party/UE.

Please refer to Clauses 5.9 in NextGen TR 23.799 for the detailed non-exhaustive list of solutions for this key issue. One thing that may need to keep tracking of is that the network capability exposure function itself does not seem to include control capabilities differently from our discussion in Focus Group.

6.1.1.2.4 Key issue: Architecture impacts when using virtual environments

The NextGen system is expected to support deployments in virtualized environments. This key issue will determine the need for and architecture impacts due to load rebalancing and load migration in the context of:

- scaling of a network function instance, and
- dynamic addition or removal of a network function instance.

An agreement is as follows:

- 1) The architecture should support mechanisms to avoid issues caused by the persistence ("stickiness") of UE-specific associations on at least NG2.

NOTE 1 – Solutions should be developed during normative phase.

NOTE 2 – Other reference points may be considered.

NOTE – Load rebalancing and load migration across network function instances assumes multiple active instances of a network function. Potential issues resulting from load rebalancing and load migration to be addressed may include:

- UE signalling overhead.

6.1.1.3 Architecture(s) for the Next Generation System

An agreement is as follows (excerpt from TR 23.799. please verify all specifics with 3GPP documents necessarily. for example, architecture principles, non-roaming reference architecture, roaming reference architectures, reference points and functionality description, and so on):

- 1) In Rel-15, AMF and SMF functions should be standardized as separate functions with standardized interactions.
- 2) NAS MM and SM protocol messages terminate in AMF and SMF respectively. This is independent of whether SM protocol terminates in the H-SMF or V-SMF.
- 3) NAS SM messages are routed by AMF.
- 4) Subscription profile data in NextGen is managed according to a user data convergence approach:
 - A common user data repository (UDR) stores the subscription data, and this can be present within the UDM.
 - UDM Front End and PCF have access to this common UDR by implementing application Front Ends to access relevant subscription data.

NOTE 1 – The terminology used above corresponds to the user data convergence approach defined in EPC UDC architecture in TS 23.335.

NOTE 2 – The application logic of UDM Front End, e.g. for location management and subscription update notification, as well as the application logic of PCF is to be further detailed during normative phase.

- 5) The SEAF and SCMF are supported by the AMF.
- 6) The AUSF is defined as a separate NF.
- 7) Each NF can interact with each other directly.
- 8) The architecture does not describe an intermediate function between control plane functions but it does not preclude the use of an intermediate function for routing and forwarding of messages (e.g., like a DRA) between control plane functions, which may be identified for specific cases in the deployments and should not require further work in stage 2.

6.1.1.3.1 Consolidated architecture option 1

(Source: Cisco Systems, Inc., ATT, Sprint, etc.)

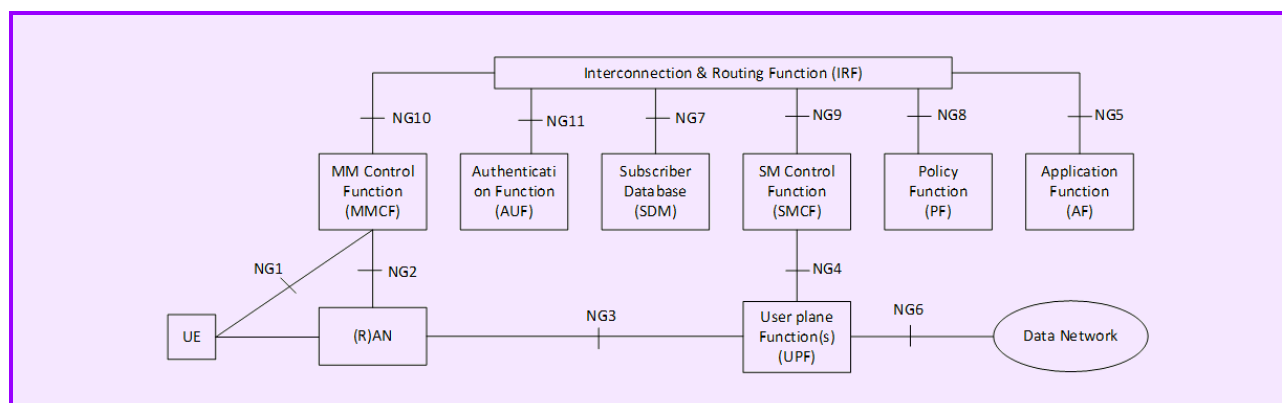


Figure 6.1.1-11 – Non-roaming reference architecture

Following is the nature of the functionality supported by the NFs in the reference architecture.

- **Subscriber DB (SDM):** Management of subscription profile.
- **SM Control Function (SMCF):** Allocation of UE IP address, User plane function selection, enforcement of policy and charging rules, etc. The exact set of functionality will be based on the conclusion of key issues 2, 3, 4, 6.
- **MM Control Function (MMCF):** Terminates control plane interface, NG2, from the Access Network carrying both access specific and access common information. Also terminates the NG1 interface. Performs access specific, e.g. UE reachability management, mobility restriction, etc., and access common, e.g. UE registration management, etc., functionality. The exact set of functionality will be based on the conclusion of key issues 1, 2, 3, 4, 18.

NOTE 1 – Generic name needs to be used to reflect the fact that MMCF is common to all accesses.

- **Policy Function (PF):** Provides dynamic policies for QoS enforcement, charging control, traffic routing, etc. The exact set of functionality will be based on the conclusion of key issue 10.
- **Application Function (AF):** Requests dynamic policies and/or charging control.
- **Authentication Function (AUF):** Performs authentication process with the UE. The exact set of functionality will be based on the conclusion of key issue 12.
- **User-plane Function(s) (UPF):** Supports user-plane operations (forwarding to other user-plane functions/a data networks/the control-plane, bitrate enforcement, service detection, etc.). Multiple user-plane functions per session can be activated and configured by the control-plane as needed for a given user-plane scenario. The exact set of functionality will be based on the conclusion of key issue 4.

6.1.1.3.2 Consolidated architecture option 2

(Source: China Mobile, etc.)

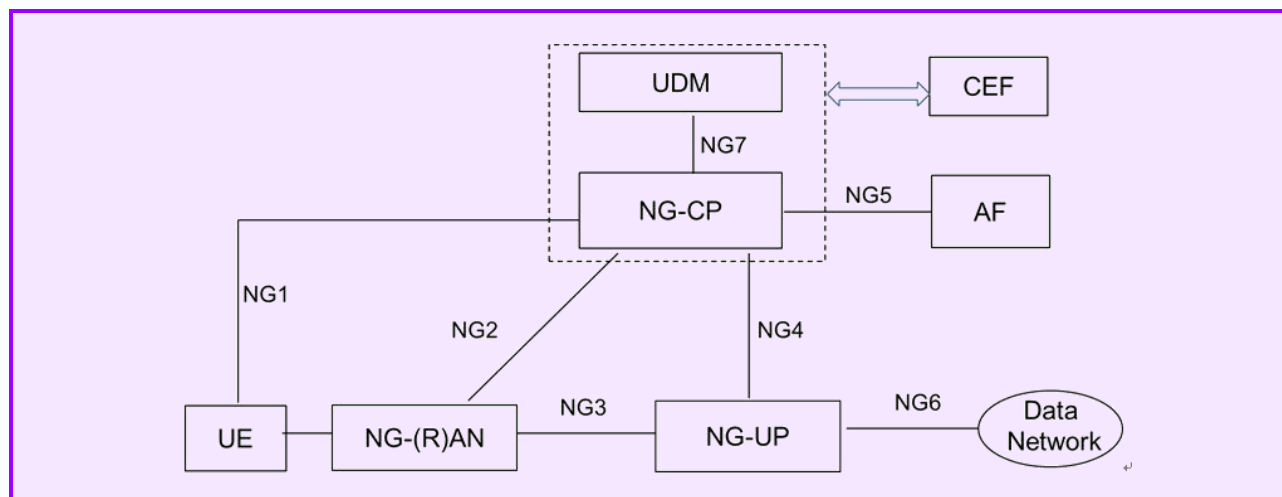


Figure 6.1.1-12 – Non-Roaming reference Architecture

The NextGen architecture consists of the following functions:

NG Core Control Plane functions (NG-CP):

- Authentication & Authorization
- Mobility Management
- Session Management
- Policy Control
- Charging Management (control part of enforcement of charging)

- Lawful intercept (control part of enforcement of LI)
- NSSF: network slice selection function

NG Core User Plane functions (NG-UP):

- Packet routing & forwarding
- Traffic handling (e.g., QoS enforcement)
- Mobility anchor
- Session anchor
- Packet inspection
- Lawful intercept (UP collection)

NG User Data Management (UDM)

- Store user subscription data, policy data (e.g., on QoS and charging), session/user related context and state in a unified data layer. Such a unified data layer is to reduce redundant state information in multiple network functions. It also aims at "stateless" network functions, i.e., state and context information could be easily relocated and restored to benefit from virtualization.

Capability Exposure Function (CEF)

- Exposure information/capability to the 3rd party and/or UEs in proper manner for efficient service provision

6.1.1.3.3 Consolidated architecture option 3

(Source: Nokia, Ericsson, Verizon, etc.)

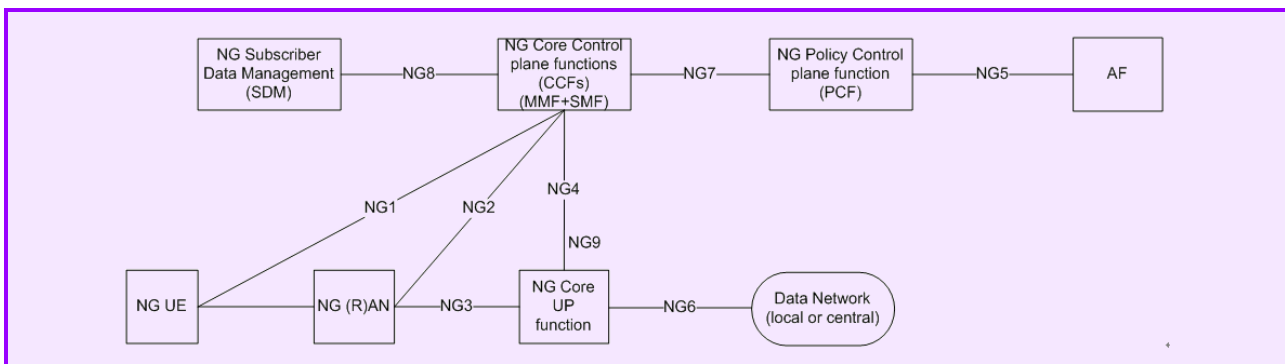


Figure 6.1.1-13 – Non-roaming reference architecture

The 5G Reference Architecture consist of the following functions:

- NG Subscriber Data Management (NG SDM)
- NG Policy Control function (NG PCF)
- NG Core Control functions (NG CCFs)
- NG Core User plane function (NG UPF)
- NG RAN
- NG UE
- Data network, e.g. operator services, Internet access or 3rd party services.

The **NG Core Control functions** include the following functionality:

- Termination of RAN CP interface
- Termination of NAS
- Access Authentication

- NAS Ciphering and Integrity protection
- Mobility management
- Session Management
- UE IP address allocation & management (incl optional Authorization)
- Selection of UP function
- Termination of interfaces towards Policy control and Charging functions
- Policy & Charging rules handling, incl control part of enforcement and QoS
- Lawful intercept (CP and interface to LI System)

NOTE 5 – Not all of the CCF functions are required to be supported in an instance of CCFs of a network slice

The **NG Core User plane functions** include the following functionality:

- Anchor point for Intra-/Inter-RAT mobility (when applicable)
- External PDU session point of interconnect (e.g., IP).
- Packet routing & forwarding
- QoS handling for User plane
- Packet inspection and Policy rule enforcement
- Lawful intercept (UP collection)
- Traffic accounting and reporting

NOTE 6 – Not all of the UPF functions are required to be supported in an instance of user plane function of a network slice.

The **NG Policy function** includes the following functionality:

- Supports unified policy framework to govern network behavior.
- Provides policy rules to control plane function(s) to enforce them.

6.1.1.3.4 Consolidated architecture option 4

(Source: Deutsche Telekom AG)

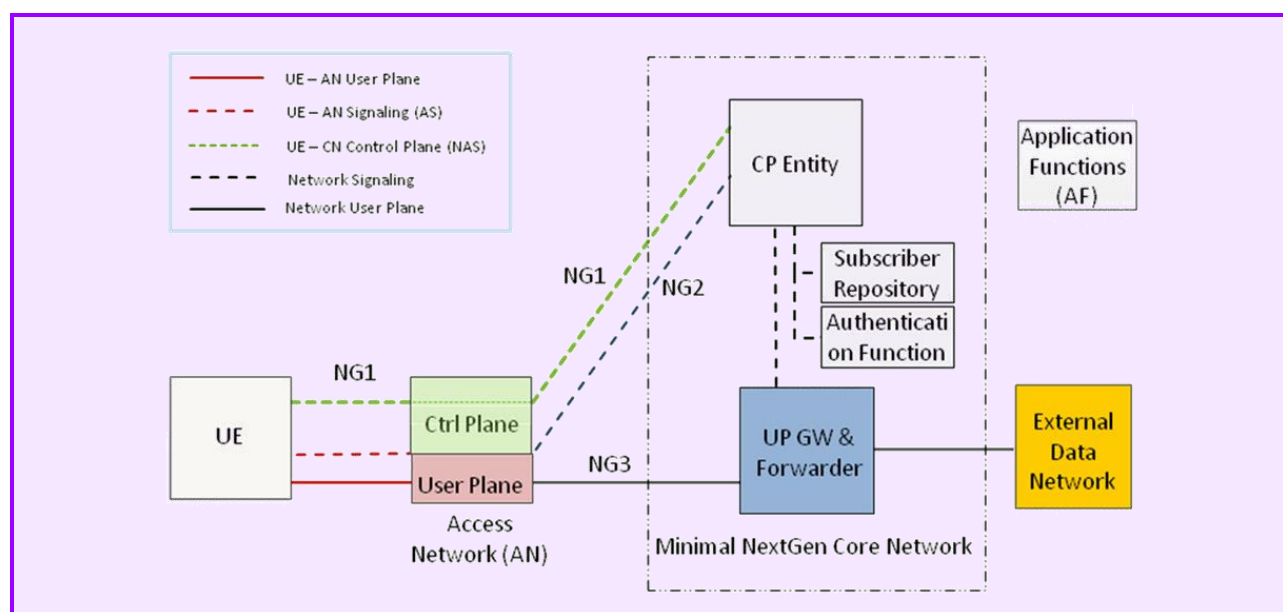


Figure 6.1.1-14 – Baseline NextGen system

A minimal system should be specified as the baseline that can serve the most basic use cases. Additional functionality can then be added on demand as needed to serve more demanding use cases.

The communication (CP and UP) between UE and AN as well as any AN internal communication is access specific. All other communication can be designed in an access independent fashion:

- The CP entity authenticates the subscriber and obtains the subscription information interacting with an access independent subscriber repository and authentication function
- Following successful authentication, the CP entity informs the UP GW & Forwarder about the permitted UP flows for this UE, based on the information obtained from the subscriber profile
- The User Plane GW & Forwarder function routes service data flows between AN and external Data Network (e.g., Internet)

Additional CN functions (QoS, charging, etc.) beyond this baseline system are needed in more demanding usage scenarios. Such additional CN control plane functions are enabled on demand, as required by the specific customer use case.

It is thereby possible to configure e.g. the following CP functions per UE / flow:

- Quality of Service
- Charging
- Policy Control
- Mobility Management
- Session/Service Continuity (with or without intra-AN mobility, with or without AN change)

CN user plane functions associated with the active CP functions can be chained into the user plane path as shown above.

NOTE – the above functions are examples only and are neither intended to define the overall functionality of the network nor the granularity of the network functions.

NOTE – AF may be shared with external network, or exist exclusively within or outside of the operator domain.

6.1.1.4 Gap Analysis (should be updated after further investigation)

- A network slice is an E2E concept including CN parts of the slice and RAN parts of the slice (3GPP SA2 Interim Agreement)
 - RAN slicing at 3GPP SA2 is out of scope
 - RAN WGs at 3GPP have agreed not including slicing in phase 1 (Rel-15) due to many consideration
- The right level of NF granularity is scope of phase 2 (Rel-16)
 - The level of inter-dependency between network functions
 - Need for independent scalability of individual network functions
 - Need for deployment of individual network functions within or across operator network (e.g., PLMN) boundaries
 - Need for supporting centralized or geographically distributed deployments
 - Based on above, identification of the network functions for the next generation architecture and definition of the set of functionalities supported by each of them
- Network capability exposure and APIs is scope of phase 2 (Rel-16)
 - Define network capability exposure framework
 - Identify the mechanisms and interfaces to expose network capabilities to the 3rd party and/or UEs
 - Identify the network information that can be provided to 3rd party ISPs/ICPs and to the UE to enable more customized and efficient service provision

- How to create the network slice based on the requirement of the 3rd party or customize network function on-demand
 - Above all bullets are not scope of 3GPP SA2
- Definition of requirements or guideline for global or public slice type (e.g., emergency, safety, disaster, etc.)
- Possible work of ITU-T (e.g., not GSMA) from the viewpoint of wired and wireless convergence

Table 6.1.1-3 – Gap analysis

Issues related on Network Softwarization	Other SDO's Activities	Issues relevant to ITU-T	Description
Network Function Definition and Granularity	ETSI-NFV		
Network Function Software Architecture	ETSI-NFV		
Service Chaining/Blueprint/Network Slice Creation/Type		high	
Blueprint Deployment/Network Slice Instantiation	SA5		
Network Slice Usage	SA2 NSI selection		
Network Slice Management	SA5		
Session Management	SA2 SM		
Network Function Discovery and Interconnection	SA2 KI7		
Network Function Scale-in/out/migration	SA2 KI19		
Network Capability Exposure	SA2 KI9	high	
Network Slice Interworking between PLMNs	SA2 roaming	high	

6.1.1.5 Future Plan for Phase 1 Normative Work

(excerpt from doc #S2-167232 (NextGen WID-v9) at SA2 #118 meeting. please verify all specifics necessarily. for example, supporting features, and so on)

The objective of this work item is to develop the Stage 2 normative specification of Phase 1 of the 5G system based on the conclusions captured in TR 23.799. Phase 1 specifies a deployable 5G architecture that supports features including: network slicing, use of virtual environments, service-based architecture, network capability exposure, network discovery and selection, and so on.

Phase 1 architecture also serves as a foundational architecture for enhancements in future releases that would support additional features.

A set of new specifications will describe the 5G System:

- System Architecture for 5G System: Specifies the overall system architecture reference model including network functions and description of high level functions.
- Procedures for 5G System: Specifies the procedures and flows to capture the interactions between network functions, access network(s) and UE for the listed features.

Expected Output and Time scale:

- Work item rapporteur(s): China Mobile, Tao Sun (suntao@chinamobile.com)

Table 6.1.1-4 – Future plan for phase 1 normative work

Spec No	Title	Presented for information at plenary #	Approved at plenary #	Comments
TS 23.xxx	System Architecture for 5G System	TSG SA #77 (Sep, 2017)	TSG SA #77 (Sep, 2017)	Editor: Nokia, Devaki Chandramouli (devaki.chandramouli@nokia.com)
TS 23.xxx	Procedures for 5G System	TSG SA #77 (Sep, 2017)	TSG SA #78 (Dec, 2017)	Editor: Ericsson, Peter Hedman (peter.hedman@ericsson.com)

6.1.2 Standardization activities at ETSI ISG NFV

ETSI ISG NFV has been active since 2012, with its NFV vision that “An open ecosystem for NFV enables rapid service innovation for network operators and service providers” and that “Innovation in end-to-end services is enabled by software-based deployment and operationalization of virtualized network function (and network services) on independently deployed and operated NFV infrastructure platforms”. In the 2013-2014 time frame, it conducted extensive pre-standardization study and published the Release 1 specifications, regarding the NFV framework architecture, use cases, terminology and so forth. In 2016, it published Release 2 specifications and reports, including functional requirements, interface and information model of reference points for the management and orchestration function block, called NFV-MANO. In addition, the stage 3 specifications for the reference points owned by ETSI ISG NFV has been defined and will be included in Release 2.

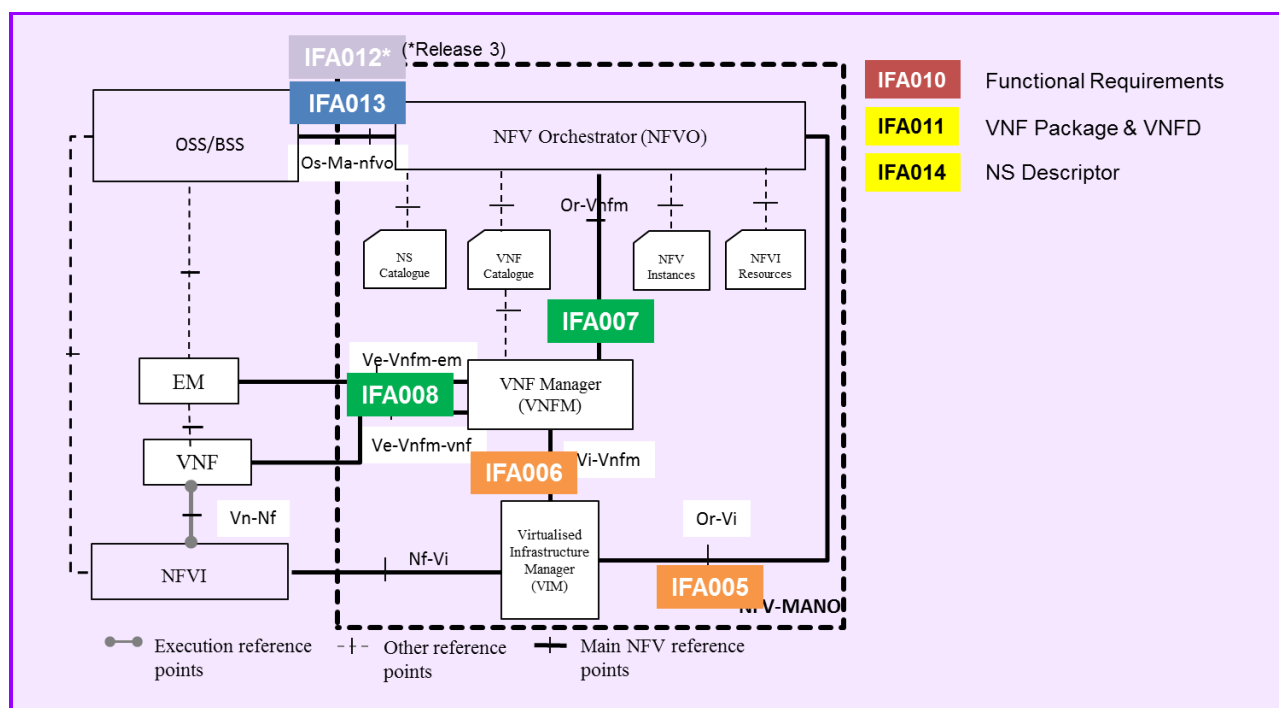


Figure 6.1.2-1 – ETSI NFV Group Specifications related to MANO interfaces

NFV is a technology enabler which is relevant to a large number of use cases including what were identified for IMT-2020, and consequently, to different stakeholders and organizations: consuming upstream technologies and enabling an ecosystem for other technologies to make use of it. ETSI ISG NFV has been leading harmonization of information models applicable to NFV, focusing on SDN and other related technologies. The alignment has been studied for the information models defined by parties such as 3GPP, ATIS, BBF, DMTF, ETSI NFV, IETF, ITU-T, MMF, OASIS/TOSCA, OCC, ODL, ONE, OPNFV, TM Forum.

NFV Release 3 features are defined matching market demand and associated work items have been studied including, Charging, billing and accounting, Policy management, End-to-end management, Multi-site NFV services, VNF lifecycle management and so forth.

6.1.3 Standardization activities at ETSI ISG MEC

The future broadband networks need to cover wide range of key use cases from ultra-low latency services to massive IoT. In order to respond to demands on expected throughput, latency, scalability and programmability, ETSI established an Industry Specification Group on Mobile Edge Computing in 2014.

ETSI ISG MEC develops a standardized and open environment that offers distributed cloud-computing capabilities and an IT service environment to application developers and content providers. By February 2016, the group has finalized three stage 2 specifications: Terminology, Technical Requirements and the Framework and Reference Architecture. Currently the ISG works on specifications of the following work items:

- MEC platform Application Enablement;
- MEC API principles and guidelines;
- MEC Services APIs for Radio Network Information, Location, UE identity and Bandwidth management;
- MEC system, host and platform management;
- MEC lifecycle and policy management;
- MEC UE application interface;
- Deployment of MEC in a NFV environment;
- End-to-end Mobility

By offering distributed cloud-computing capabilities and exposure to real-time radio network and context information, the MEC environment is characterized by:

- **Ultra-low latency:** Mobile Edge services can be run close to the end user devices to provide the lowest possible latency
- **Proximity:** Being close to the source of information, Mobile Edge Computing is particularly useful to capture key information for analytics and big data
- **High Bandwidth:** Mobile Edge location at the edge of the network combined with the use of real time radio network information can be used to optimize the bandwidth for the applications
- **Location awareness:** Mobile Edge can leverage the low-level signalling information to determine the location of each connected device
- **Real time insight into radio network and context information:** Real-time network data can be used by the applications and services to offer context-related services.

MEC can bring significant improvement of mobile user's Quality of Experience on latency or QoS sensitive services. The example use cases include Edge Video Orchestration, Mobile Video Throughput guidance, Augmented reality, Intelligent Video Analytics, Vehicle-to-Infrastructure (V2I) communication etc.

MEC enables the implementation of mobile edge applications as software-only entities that run on top of a virtualisation infrastructure, which is located in or close to the network edge.

Figure 1 depicts the MEC framework according to ETSI GS MEC 003 specification.

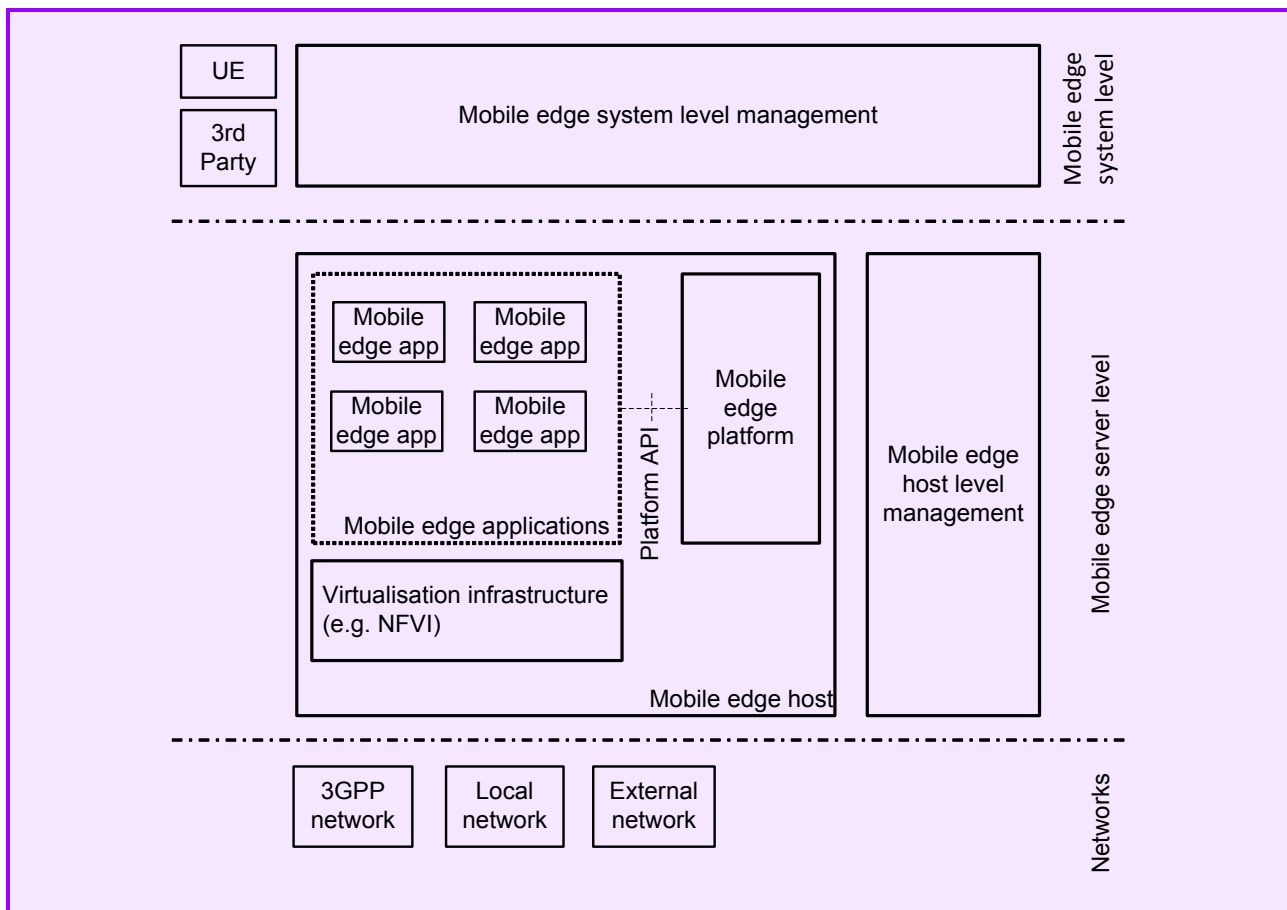


Figure 6.1.3-1 – Mobile Edge Computing Framework

The mobile edge system consists of the **mobile edge host** and the **mobile edge management** necessary to run mobile edge applications within an operator network or a subset of an operator network.

The **mobile edge host** including the following:

- **Virtualisation infrastructure** that provides compute, storage, and network resources for the purpose of running mobile edge platform and mobile edge applications.
- **Mobile edge platform** is the collection of essential functionality required to run mobile edge applications on a particular virtualisation infrastructure and enable them to provide and consume mobile edge services. The mobile edge platform can also provide services.
- **Mobile edge applications** are instantiated on the mobile edge server based on configuration or requests validated by the mobile edge management.

The **mobile edge management** comprises the mobile edge system level management and the mobile edge host level management.

- The **mobile edge system level management** includes the mobile edge orchestrator as its core component, which has an overview of the complete mobile edge system.
- The **mobile edge host level management** comprises the mobile edge platform manager and the virtualisation infrastructure manager, and handles the management of a particular mobile edge platform and the applications running on it.

References

- [6.1.3-1] Draft ETSI GS MEC 002 V0.5.1 (2016-02): Mobile-Edge Computing (MEC); Technical Requirements.
- [6.1.3-2] Draft ETSI GS MEC 003 V0.3.2 (2016-02): Mobile-Edge Computing (MEC); Framework and reference architecture.
- [6.1.3-3] ETSI, "Mobile-Edge Computing – Introductory Technical White Paper," 2014.

6.1.4 Standardization activities at ETSI NTECH AFI WG

The AFI working group in ETSI's NTECH Technical Committee focuses on Autonomic Management and Control (AMC) for network and services, with a comprehensive work programme. Its main deliverable is a reference model for a Generic Autonomic Network Architecture (GANA). The GANA model defines a generic AMC framework and structure within which to specify and design autonomics-enabling functional blocks for any network architecture and its management architecture. The NTECH-AFI work programme comprises also an implementation guide for the GANA reference model, and GANA instantiations onto various reference architectures defined by standardization organizations such as 3GPP, BBF, IEEE, ITU-T and other. For example, the generic model has been instantiated onto the 3GPP mobile backhaul and core network (EPC) architectures as reported in TR 103.404 0.

6.1.4.1 Autonomic Management & Control (AMC) Reference Model

AMC is about Decision-making-Elements (DEs) as autonomic functions (logics that dynamically configure their associated managed entities in respective closed control-loops) with cognition introduced in the management plane as well as in the control plane (whether these planes are distributed or centralized).

Cognition (learning, analysing and reasoning used to effect advanced adaptation) in DEs, enhances DE logic and enables DEs to manage and handle even the unforeseen situations and events detected in the network.

Control refers to the control-loop logic kernel of the DE, capable of dynamically adapting network resources and parameters or services in response to changes in network goals/policies, context changes and challenges in the network environment that affect service availability, reliability and quality.

DEs realize self-* features of a functionality or system (self-configuration, self-optimization, etc.) as a result of the decision-making behaviour of a DE that performs dynamic and adaptive management and control of its associated Managed Entities (MEs) and their configurable and controllable parameters. Such a DE can be embedded in a Network Element (NE) or higher at a specific layer of the outer overall network and services management and control architecture—thereby creating AMC architecture. An NE may be physical or virtualized (such as in the case of the NFV paradigm).

From an architecture perspective, ETSI/NTECH AMC Framework and 3GPP Hybrid-SON (Self Organising Network) model are compatible with each other. Indeed, they share common design principles on enabling implementers of autonomics algorithms to combine centralized control and distributed control of network resources, parameters and services (more details on the compatibility can be found in the ETSI White Paper No.16 [Ref.6.1.4-7]). Indeed, a control-loop in the AMC Framework can be based on a distributed model (for fast control-loops). In this case the DE is embedded in the nodes (physical or virtualized), whereas in a centralized model (for slow control-loops), the DE is outside of the NEs. Both kinds of control-loops act towards a global goal to ensure a stable state of the network. A DE can negotiate with another DE to realize dynamic adaptation of network resources and parameters, or services, via reference points defined in the ETSI/NTECH AMC framework.

This aspect of interworking, complementary, hierarchical and nested control-loops leads to the notion of global network autonomics, a result of interworking DEs as collaborative manager components that perform AMC of their associated MEs and their configurable and controllable parameters.

From an implementation perspective, a DE, as a software module or an executable behavioural specification that enhances management and control intelligence capabilities, may be (re)-loaded or replaced in nodes and in the network centralized management and control plane. This is directly related to the notion of software-driven networks (also referred to as software-empowered networks).

Indeed, DEs (software components) are meant to empower the networks and the management and control planes to realize self-* properties: auto-discovery of information/resources/capabilities/services; self-configuration; self-protection; self-diagnosis; self-repair/heal; self-optimization; self-organization behaviours; as well as self-awareness.

6.1.4.2 Instantiation of the Autonomic Management & Control reference model

Autonomics-enabled implementation-oriented architectures are a result of GANA instantiations onto various reference architectures defined by standardization organizations such as: 3GPP (e.g., ETSI TR 103.404 that has also been presented and discussed with 3GPP SA2 and SA5, Broadband Forum (BBF) (a TR on autonomics in the BBF architectures to be produced in early 2017), IEEE, ITU-T, and other SDOs.

6.1.4.3 ETSI NTECH Call for Autonomics Proof of concept

ETSI NTECH AFI has produced a GANA Proof of Concept (PoC) Framework specification 0 and is now calling for PoC projects. One of the PoCs to be launched in 2016 is a 5G PoC that will be used to demonstrate the complementarities of paradigms of AMC, SDN, NFV, E2E Service Orchestration and Big-Data Analytics for AMC in addressing the dynamics, flexibility in network & service compositions, and intelligence expected in 5G networks. The 5G Network Slicing PoC of ETSI-NTECH/AFI will demonstrate 5G Network Slice Creation, Management and Orchestration use case empowered by intelligence brought by AMC. This is a very generic NGMN 5G use case.

6.1.4.4 Standardization efforts in Joint SDO/Fora Industry Harmonization Initiative for Unified Standards for AMC, SDN, NFV, E2E Service Orchestration and Big-Data Analytics for AMC

This initiative was created by SDOs/Fora in 2014, and it starts demonstrating the success factors for effective cross-SDO collaboration.

Over 12 SDOs / Fora are engaged by the initiative, including ETSI TC NTECH and ITU-T SG13, SG2 and JCA on SDN standards.

Figure 6.1.4-5 below extracted from this report illustrates what the Industry is expecting in terms of standard harmonisation.

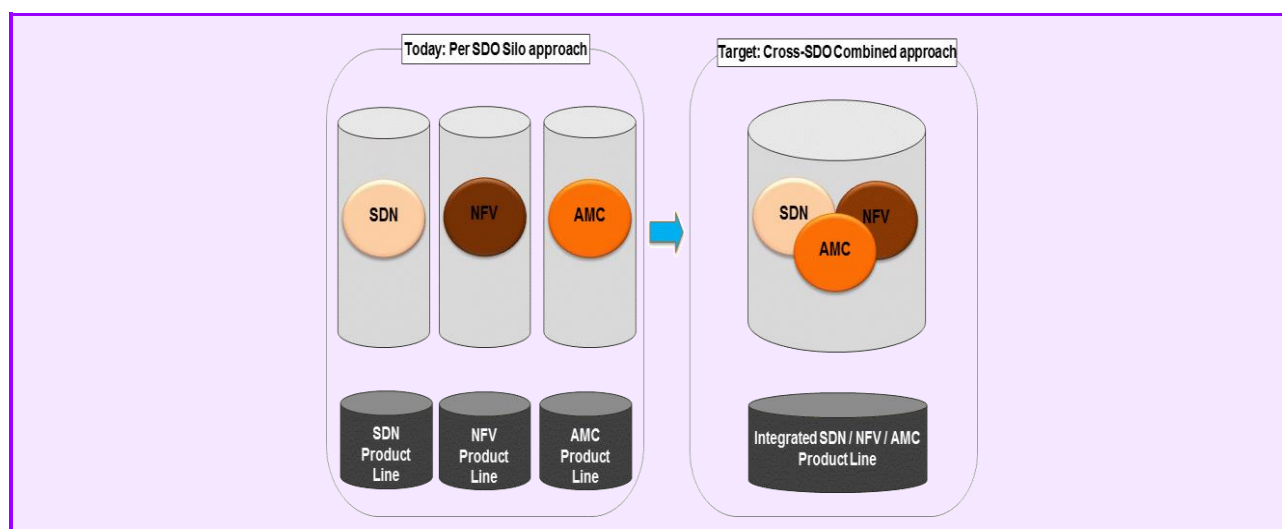


Figure 6.1.4-5 – Cross SDO Combined approach SDN NFV AMC

The Initiative on Joint SDOs/Fora Industry Harmonization on Unified Standards for AMC, SDN, NFV, E2E Service Orchestration and Big-Data Analytics for AMC – all viewed in their combined roles as Software-Oriented Enablers for 5G, is driven by a joint SDOs/Fora White Paper that is identifying the items requiring standard harmonization, to reduce “silos”, duplication of work and divergence [Ref.6.1.4-2].

References

- [6.1.4-1] ETSI NTECH AFI WG: Evolution of Management towards Autonomic Future Internet (AFI)), <https://portal.etsi.org/TBSiteMap/NTECH/NTECHAFIToR.aspx>.
- [6.1.4-2] Workshop Report from the Initiative: Joint SDOs/Fora Industry Harmonization for Unified Standards on AMC, SDN, NFV, E2E Orchestration, Software-oriented enablers for 5G: Joint SDOs/Fora Workshop Report from the workshop held on 4th June 2015 at the TMForum Live 2015 can be downloaded from here: http://projects.sigma-orionis.com/eciao/wp-content/uploads/2015/07/Report-on-Joint-SDOs-Industry-Harmonization-for-Unified-Standards-on-AMC_SDN_NFV_E2E-Orchestration_ver3.01.compressed.pdf.
- [6.1.4-3] ETSI NTECH AFI WG Work Programme, http://webapp.etsi.org/WorkProgram/Report_WorkItem.asp?WKI_ID=42951.
- [6.1.4-4] Draft ETSI TR 103 404 “Network Technologies (NTECH);Autonomic network engineering for the self-managing Future Internet (AFI); Autonomicity and Self-Management in the Backhaul and Core network parts of the 3GPP Architecture” V0.0.10 (2016-09), https://docbox.etsi.org/NTECH/Open/stf%20501_gana_3gpp_draft_tr%20103%20404_v0010_ntech%20approved.pdf.
- [6.1.4-5] ETSI TS 103 371 NTECH AFI Proofs of Concept Framework <http://ntechwiki.etsi.org/>.
- [6.1.4-6] Standardisation Task Force Autonomic and Self-Managed Networks Phase 2 (BBF), https://portal.etsi.org/stfs/Cfe/CL16_3279.doc.
- [6.1.4-7] ETSI White Paper no. 16: The Generic Autonomic Networking Architecture Reference Model for Autonomic Networking, Cognitive Networking and Self-Management of Networks and Services, http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp16_gana_Ed1_20161011.pdf.

6.1.5 Standardization activities at IETF on Detnet

6.1.5.1 Introduction

The Deterministic Networking (Detnet) Working Group in IETF was established in October, 2015. The work focuses on creating multi-hop path over Layer 2 bridged and layer 3 routed segment with deterministic properties in latency, packet loss, jitter, and high reliability, which triggers by the need to support time sensitive services in IP network, usually these kinds of services require bounds on latency, loss and etc. The mission-critical IoT services in 5G such as factory automation, smart grid are typical use cases of Detnet. To support these critical packet flows, it is required to guarantee the absolute maximum and minimum end to end latency across the network, and a better packet loss ratio than the traditional requirements. In addition, the latency and packet loss requirements in transport network in 5G cellular radio network also become more stringent compared with the cellular radio network today. To create the path with bounded latency and packet loss requirements in the IP transport network, Detnet may be a potential solution. For instance, Detnet can be applied to provide with multi-hop path between eNBs with deterministic latency, since CoMP(coordinated multipoint transmission/reception) identified as a technology in 5G can benefit from the coordinated scheduling of different cells only if the signalling delay between eNBs is within 1-10ms. The first step of Detnet is to find the solution for networks that are under a single administrative control or within a closed group of administrative control. The solution proposed by the Detnet will be applicable not only to campus-wide networks, but also to the private WAN.

6.1.5.2 The Work Scope of Detnet WG

Detnet work covers the overall architecture, data plane technologies, the data plane flow information model, yang models, problem statement and vertical requirements as needed, etc. The work in the overall architecture will consist of the data plane, OAM, time synchronization, control plane and manage plane. The data plane will work on the method to identify the flow and to forward the packets across the multi-hop deterministic path over the layer 3 network. The data plane flow information model will be used by the reservation protocol and yang data model. New yang models will be used for device configuration, states reporting, and advertising the deterministic network elements information to the control plane. Problem statement work will establish the deployment environment and deterministic network requirements. Vertical requirements part will analyze the details of the deterministic properties for various services.

6.1.5.3 Standardization activities states of Detnet WG

At present Detnet has four WG drafts, the Problem Statement document states issues that need to be solved, the Use Cases document describes the usecases which will apply Detnet technologies, the Architecture document defines the detnet architecture and explores the techniques used for extreme low data loss rates and bounded latency, the Data Plane document identifies existing IP and MPLS, and other encapsulations that run over IP and/or MPLS data plane technologies that can be considered as the base line solution for deterministic networking data plane definition.

The Problem Statement document states that Detnet must propose a new model first to integrate determinism in IT technology, afterwards, signalling used to establish multi-hop path which meets deterministic requirements will be specified, and the tagging elements to be used identify the flows need to be specified as well. Multi-hop path can be end to end from a host to another host, it also can be the deterministic part of the whole path. To establish the deterministic path, one needs to compute path in advance. Path computation can be based on the centralized or the distributed. Centralized path computation can combine with PCE to achieve the global optimization. The draft proposes many aspects that need to be clarified based on this model, such as how to install path computation on the network entities, how to establish the path, how to make controller interact with network devices etc.. Distributed path computation can be an enhancement to RSVP-TE. To enable RSVP-TE function, some issues such as improving CSPF to compute constrained path, defining flow identification and so on need to be resolved.

The Use Cases document specifies the requirements such as bandwidth and latency to establish multi-hop deterministic path for diverse industries. Various industry applications are included, such as professional audio and video, electrical utilities, building automation system, gaming and so on. Using professional audio and video as an example, when transmitting the professional audio and video on the packet -based network, it's required to eliminate congestion by guaranteeing the bandwidth to achieve the uninterrupted stream playback, to meet the tolerance requirements of audio and video synchronize with bounded latency to make the sound matched to the movements in the video. For packet-based transport network in cellular radio, the allocated time that allows is 100us between the remote radio heads and the baseband processing unite, and the transport loss assume almost zero, because errors can result in a reset of the radio interface, which can reduce the throughput. To support the mobile services in 5G, the path with deterministic properties in latency and loss needs to be established.

The Detnet architecture draft proposes three key points, which are reserving the data plane resources for Detnet flows, creating fixed path for Detnet, and applying replication and elimination at different points for high packet delivery ratio. The draft emphasises that the Detnet flow can't exclude the traditional non-Detnet flow presence, and to support Detnet flows, one doesn't need to configure the special interface. Detnet uses QoS parameters such as maximum and minimum end to end latency, and possibility of packet loss under any assumptions to evaluate the deterministic properties. In the draft, Detnet architecture is composed of an application plane, a control plane, and a network plane, which echoes the architecture of Software-defined Networking (SDN). Detnet architecture is presented in Figure 1 below, which is cited from draft-ietf-detnet architecture. Note that the number of PCE or intermediate node or NIC in the Figure 6.1.5-1 is just for illustration.

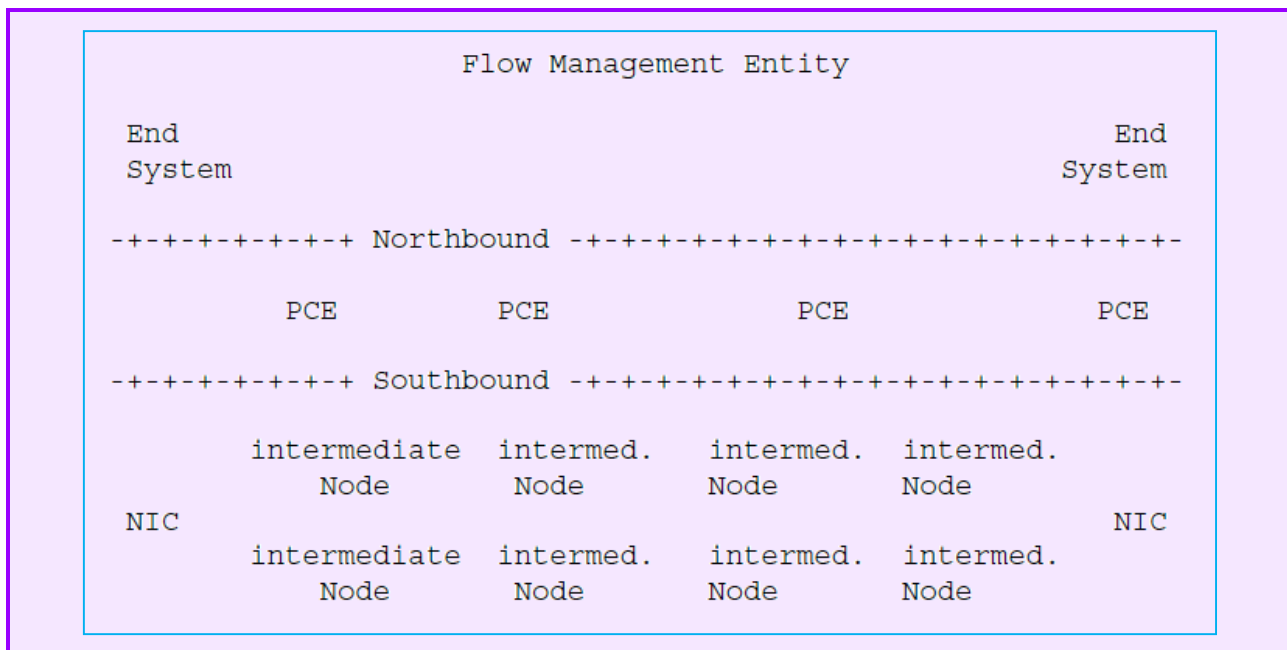


Figure 6.1.5-1 – Detnet architecture

The application plane consists of user agent which is a special application to interact with the end users or providers. In addition, application plane will implement the request for Detnet services from providers via FME. The control plane most importantly includes PCE which will play the core role of computing the path. The control plane will communicate with the application plane through the northbound interface. The network plane consists of all network devices, such as intermediate nodes like IP routers and switches, and NIC of the host. The intermediate nodes report their capability and physical resources to the control plane, and update the PCE topology data bases through the southbound interface. Then PCE computes the path and sets the path up for each flow

For the data plane of Detnet, draft-ietf-detnet-dp-alt analyses the existing IP, MPLS and other encapsulations which can be considered as the baseline for detnet data plane definition. The DetNet data plane is logically divided into two layers: service layer and transport layer. The DetNet service layer provides adaptation of DetNet services, it is composed of a shim layer to carry deterministic flow specific attributes, which are needed during forwarding and for service protection. The DetNet transport layer operates below and supports the DetNet Service layer and optionally provides congestion protection for DetNet flows. The draft lists a set of criteria to help to evaluate the potential data plane options. According to the draft, the alternatives of service layer and transport layer are shown in the below figure cited from draft-ietf-detnet-dp-alt.

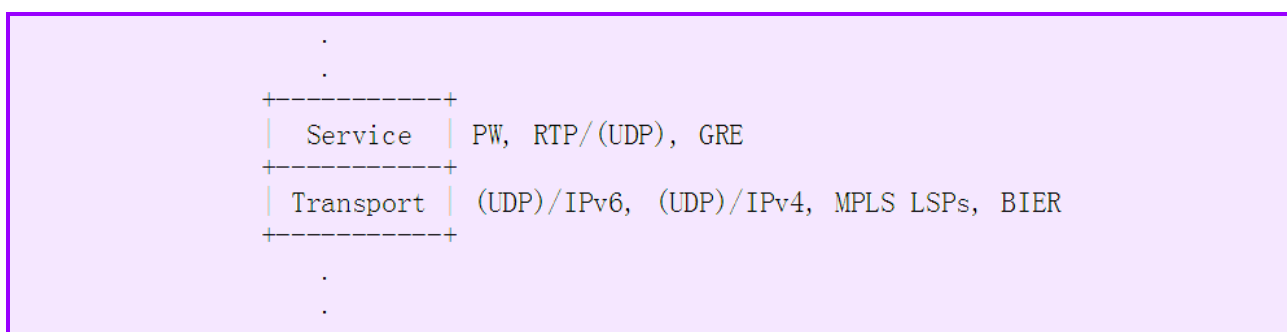


Figure 6.1.5-2 – Detnet data plane alternatives

6.1.6 Standardization activities at IETF on SFC

The delivery of end-to-end services often requires to traverse various service functions along a predetermined path. The term “Service Function Chaining” has emerged to describe the deployment of composite services that are constructed from one or more service function. Modern networks require to have the agility and flexibility to dynamically chain functions together based on network or business requirements—without having to manually reconfigure network components.

By leveraging SDN and NFV techniques, Service Function Chaining allows operator to develop new services by intelligently chaining multiple functions within the network, as well as to reduce time to market and lower operational costs for new service deployment.

The IETF SFC architecture is based on a split control plane and user plane architecture and is aligned with the principles of SDN, which is illustrated in Figure 6.1.6-1.

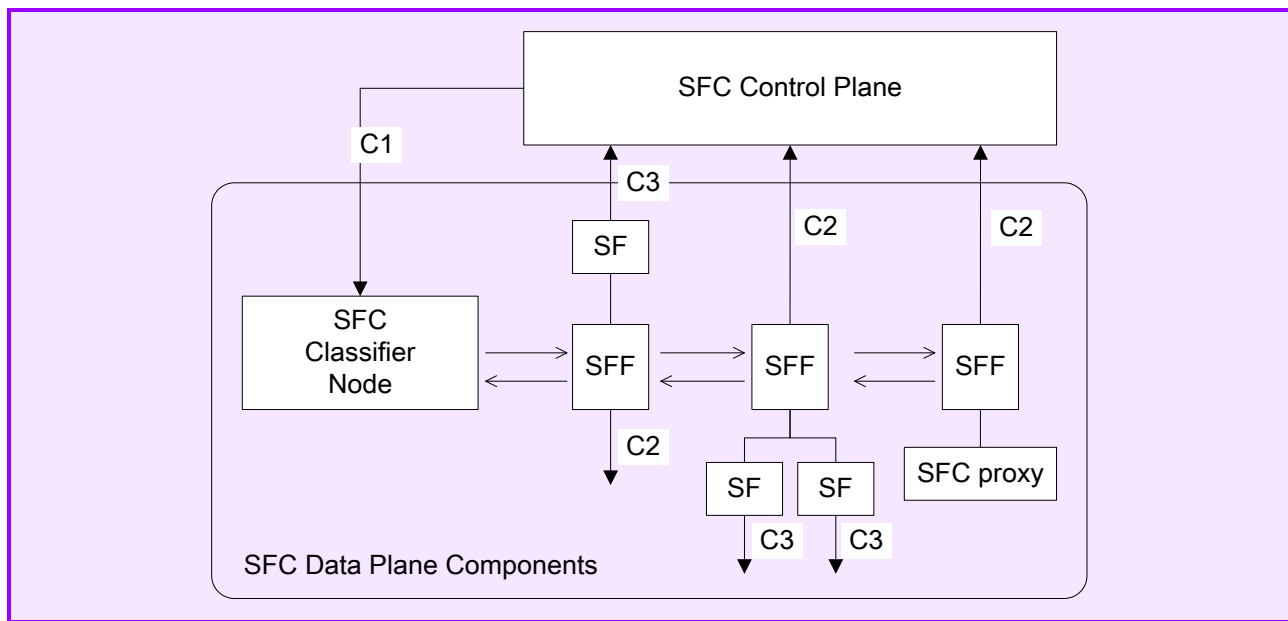


Figure 6.1.6-1 – SFC architecture Overview

The control plane receives traffic steering policies that take into consideration the topology of the data plane domain, as well as information from the data plane (e.g., load of elements, status, etc.) and constructs the service function paths. They are translated into forwarding rules, which are transferred by the control plane to the user plane and specifically to the classifier nodes and the SFFs. The classifier nodes are capable of performing classification using information up to Layer 7, whereas the SFFs have classification capabilities up to Layer 4. Both classifier nodes and SFFs are capable of handling the Network Service Header (NSH). The NSH also provides a mechanism for metadata exchange along a Service Function path. The IETF proposes Representational State Transfer (REST) based Cx (e.g., C1, C2) interfaces.

6.1.7 Standardization activities of Flexible Ethernet at OIF

6.1.7.1 Introduction of Flexible Ethernet

The Flex Ethernet (a.k.a. FlexE) implementation agreement (IA 1.0) is defined by OIF ([Ref.6.1.7-1]). The specification defines basic operation guidelines, configuration, and functions to provide a generic mechanism for supporting a variety of Ethernet MAC rates that may or may not correspond to any existing Ethernet PHY rate.

The general capabilities supported by the FlexE implementation agreement are as follows:

- Bonding of Ethernet PHYs
- Sub-rates of Ethernet PHYs

- Channelization within a PHY or a group of bonded PHYs

Enhancement of FlexE capability (FlexE2.0 project at OIF) is currently under consideration at OIF, and the improvement of FlexE 2.0 includes areas in flexibility, granularity, interoperability, and inclusion of Ethernet 200G/400G (currently under development by IEEE).

6.1.7.2 Terms and Definitions

FlexE	Flexible Ethernet.
FlexE Group	Refers to a group of from 1 to n bonded Ethernet PHYs.
FlexE client	An Ethernet flow based on a MAC data rate that may or may not correspond to any Ethernet PHY rate.
Ethernet PHY	Refers to Ethernet physical layer.
FlexE shim	Refers to the layer that maps or demaps the FlexE clients carried over a FlexE group.

6.1.7.3 Abbreviations and Acronyms

FlexE	Flexible Ethernet
PHY	OSI physical layer
PTN	Packet-based Transport Network
SDN	Software Defined Network

6.1.7.4 Overview of FlexE Data Plane

FlexE technology provides a generic mechanism to support varies of Ethernet MAC rates (e.g., 25G, 50G, 125G, etc.) that do not correspond to any standards based Ethernet PHY rate (e.g., 40G, 100G, etc.). The Ethernet MAC rate reflects a user or application based data flow, which may be larger or less than that of an Ethernet PHY. In the former case, FlexE allows that an Ethernet traffic flow, called FlexE client, to be carried over on more than one and bonded Ethernet PHYs, and in the latter case, to be carried within one Ethernet PHY (called sub-rating), or a group of bonded Ethernet PHYs (called channelization).

The following sections describe some general concepts and characteristics of FlexE data plane. These characteristics and related requirements are on FlexE interfaces.

6.1.7.4.1 FlexE Group

A FlexE group consists of 1 to n bonded Ethernet 100G PHYs, refer to Figure 6.1.7-1 (source is from OIF [Ref.6.1.7-1]). Per [Ref.6.1.7-1], $1 < n < 255$, but the initial implementation will limit the range as $1 < n < (4 \text{ or } 8)$. Each PHY is assigned a unique number within the group and made known to each end of the group.

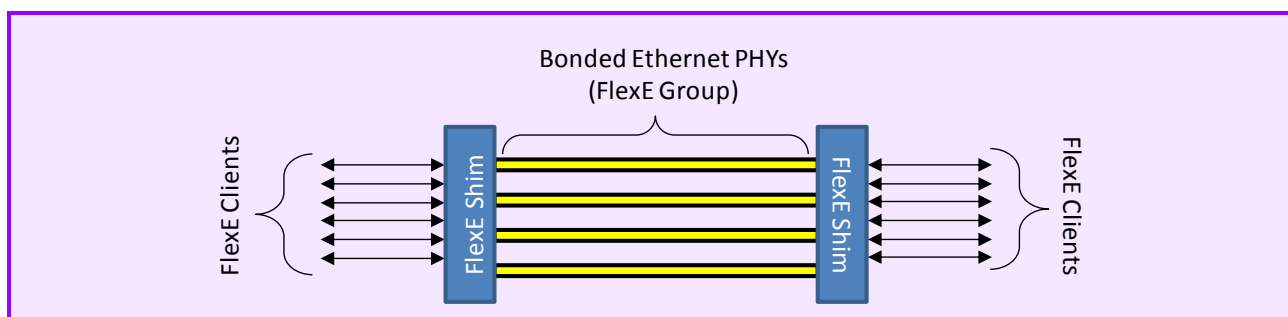


Figure 6.1.7-1 – General Structure of FlexE

6.1.7.4.2 FlexE Client

A FlexE client is an Ethernet data flow, encoded as a 64B/66B according to Figure 82-5 of IEEE 802.3 [Ref.6.1.7-2]. The implementation agreement at OIF [Ref.6.1.7-1] requires that Ethernet MAC operating at a rate of 10, 40, or $m \times 25$ Gb/s ($m \geq 1$).

6.1.7.4.3 FlexE Shim

FlexE shim is a layer on the FlexE interface, and it performs functions including the following:

- 1) On transmission, the FlexE shim maps FlexE clients (Ethernet packets) over the FlexE group in the form of 64B/66B. This function is also called *FlexE mux*.
- 2) On reception, the FlexE shim demaps FlexE clients from the FlexE group in the form of 64B/66B to Ethernet packets. This function is also called *FlexE demux*.

6.1.7.4.4 FlexE Calendar

Per [Ref.6.1.7-1], the FlexE mechanism operates using a calendar which assigns 66B block positions on sub-calendars on each PHY of the FlexE group to each of the FlexE clients. The calendar has a granularity of 5G, and has a length of 20 slots per 100G of FlexE group capacity. Two calendar configurations are supported: an “A” and a “B” calendar configuration. At any given time, one of the calendar configurations is used for mapping the FlexE clients into the FlexE group and demapping the FlexE clients from the FlexE group. Refer to the illustration of Figure 6.1.7-2 (source is from OIF [Ref.6.1.7-1]).

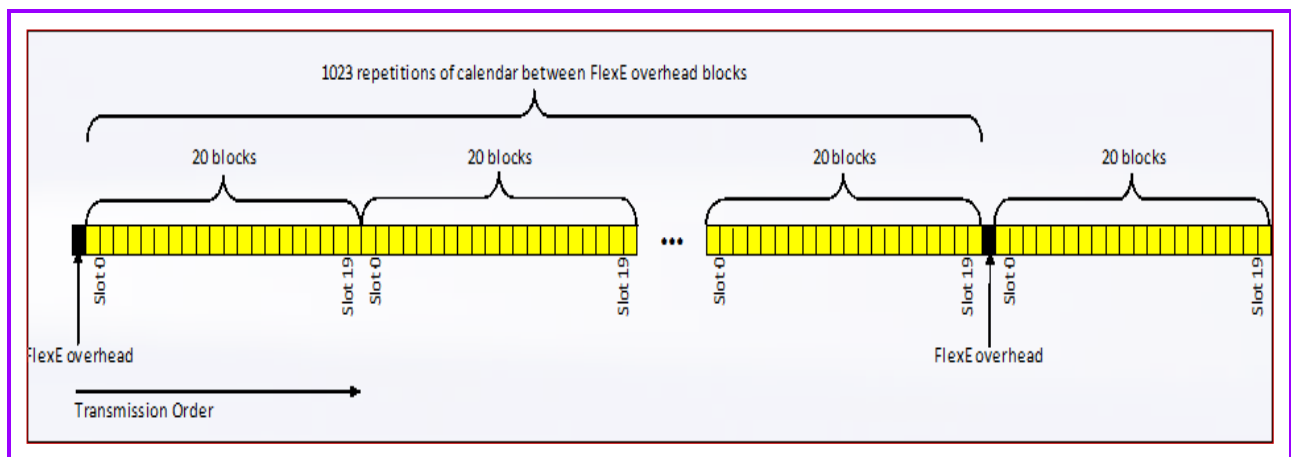


Figure 6.1.7-2 – Illustration of FlexE Calendar and Overhead on each Ethernet PHY

6.1.7.4.5 FlexE Overhead

The alignment of the data from the PHYs of the FlexE group is accomplished by the insertion of FlexE overhead into the stream of 66B blocks carried over the group. The FlexE overhead is delineated by a 66B block which can be recognized independently of the FlexE client data.

6.1.7.4.6 Capabilities of FlexE

FlexE has three general capabilities [Ref.6.1.7-1] as follows:

- 1) Bonding Ethernet PHYs
FlexE allows bonding multiple Ethernet PHYs as a larger pipe to carry a FlexE data flow. E.g., a data flow of MAC rate 200G can be carried over on two bonded Ethernet 100G PHYs.
- 2) Sub-rating of Ethernet PHYs
FlexE allows a data flow with MAC rate less than that of an Ethernet PHY be carried over. E.g., a data flow of MAC rate 50G can be carried over on a single Ethernet 100G PHY.

3) Channelization

FlexE allows several data flows be carried over a single Ethernet PHY or a group of bonded PHYs, with the MAC rate of each flow be mapped to a FlexE *connection*. E.g., a data flow at MAC rate 150G and two other data flows at MAC rate 25G each can be carried over on two bonded Ethernet 100G PHYs.

6.1.7.5 OAM

Section 7.5.2 of [Ref.6.1.7-1] describes two cases for the FlexE Demux fault handling.

First, if the intra-PHY skew exceeds the skew tolerance of the implementation, the FlexE clients will not be demapped from the incoming PHYs, and this local fault condition must be communicated to the transmitting side.

Second, if one or more of the PHYs of the FlexE group has failed (e.g., loss of signal, failure to achieve block lock or alignment lock, high BER, etc.), this is treated as a local fault condition that must be communicated to the transmitting side.

In both cases, the receiving side (transmitting side in the above) would receive the fault information as Remote PHY Fault (RPF), as described in Section 7.3.8 of [Ref.6.1.7-1].

Other OAM aspects for FlexE are elaborated in Appendix FlexE-2.

6.1.7.6 Resizing of FlexE Connection

The bandwidth on an existing FlexE connection, i.e., the Ethernet MAC rates (e.g., 25G, 50G, 125G, etc.) sometimes need to be increased or decreased based on requirement from customers or applications, and this is called resizing of FlexE Connection.

As explained in Section 6.1.7.3.4, there are two FlexE calendar configurations available, and so while one is active, the other one can be (re-)configured with smaller (downsizing) or larger (upsizing) MAC rates followed by a switch operation so the FlexE connection is now running on the second configuration with a resized MAC rate.

Figure 6.1.7-3 illustrates the FlexE calendar configuration and assignment for “A” and “B” corresponding to the two FlexE clients, and the *switch* operation. The two FlexE clients are with MAC rate 10G and 25G, respectively, on “A” configuration actively. And after the switch, the two FlexE clients are with MAC rate 35G and 20G, respectively, on “B” configuration actively. The switch from one active calendar configuration to another can be coordinated between the FlexE mux and the FlexE demux using the Calendar Request (CR) bit sent from the FlexE mux to the FlexE demux, and the Calendar Acknowledge (CA) bit sent from the FlexE demux to the FlexE mux, refer to [Ref.6.1.7-1] for detail.

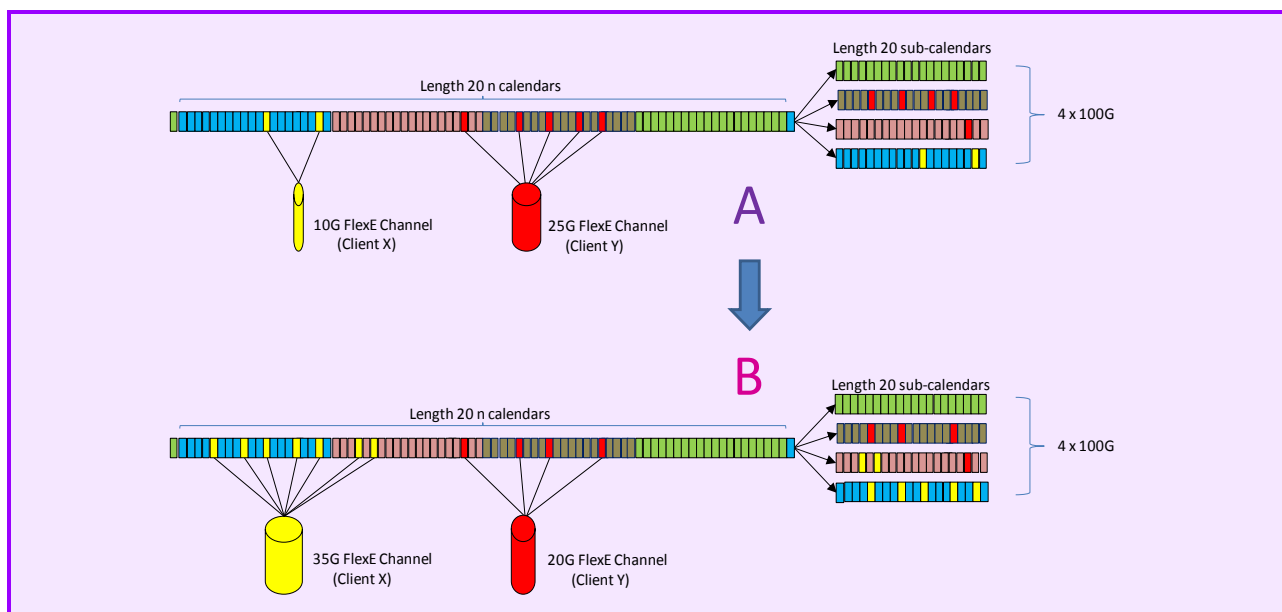


Figure 6.1.7-3 – FlexE Calendar Configuration Switching from A to B

6.1.7.7 Overview - FlexE Application and Networking Slicing

The essential ability of FlexE is to use a dedicated data pipe, supported by underlined hardware, to carry packet data flow for a specific user group or application, separated from others and with extremely low latency, guaranteed bandwidth, deterministic performance, traffic quality, privacy and security. This ability is much appreciated and valuable especially for mobile backhaul networks, enterprise customers and home networks with rigid requirement on data plane including bandwidth, delay, delay variation, and security.

FlexE technology can be used in 5G backhaul networks for slicing purpose. Network slicing is a key architectural approach for 5G, particularly for accommodating new and diversified business demands of the 5G era in a cost-efficient way. Slicing enables the deployment of multiple logical, self-contained networks on a common infrastructure platform concurrently. NGMN [Ref.6.1.7-3] defines slicing as an end-to-end concept, including core network and access network.

From the technical infrastructure perspective, slicing requires the partitioning and assignment of a set of resources that can be used in an isolated and disjunctive manner. A set of such dedicated resources can be called a slice instance. The ability of FlexE that creates end-to-end and dedicated data path provides a powerful networking slicing mechanism, which can be deployed in IP/MPLS networks including 5G backhaul networks.

An end-to-end FlexE connection in an IP/MPLS network can be viewed as a specific networking slicing instance since it owns a piece of dedicated resource provided by Ethernet end-to-end.

Figure 6.1.7-4 illustrates an end-to-end FlexE connection in an IP/MPLS network, where at least some nodes in the network are FlexE-capable. When a node is FlexE-capable, it has interfaces that contain FlexE shim. If all nodes in the network are FlexE-capable, a FlexE connection can be established end-to-end between pairs of Access Nodes across the network. If only some nodes in the network are FlexE-capable, but with proper connectivity, end-to-end FlexE connections can still be established between some pairs of Edge Nodes. In either case, it is an end-to-end FlexE connection in physical context between two peer Access Nodes.

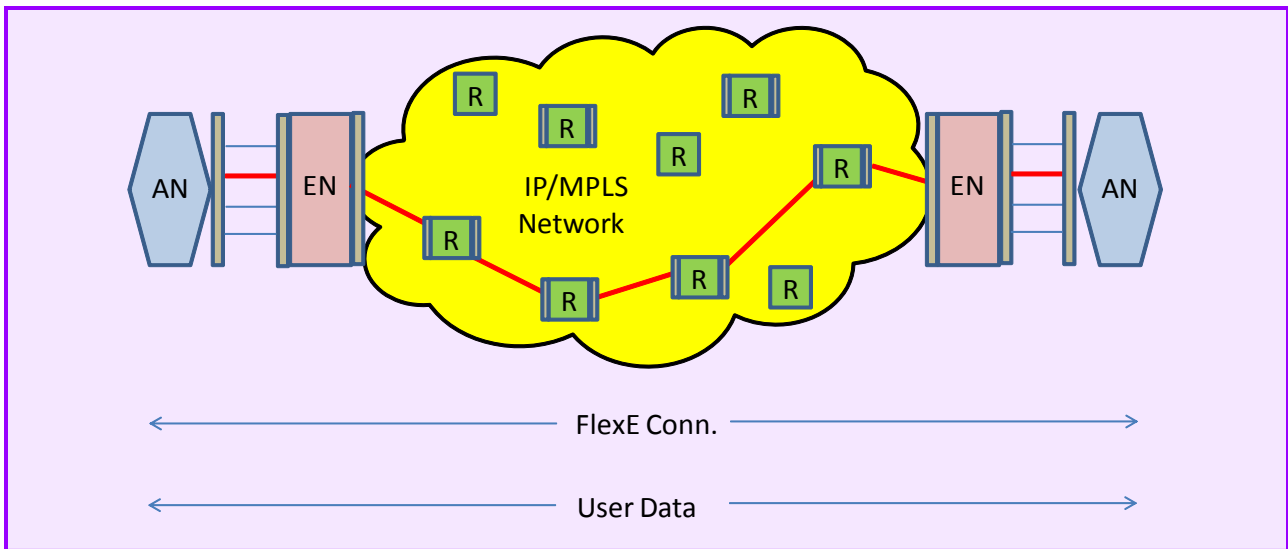


Figure 6.1.7-4 – End-to-End FlexE Connection in IP/MPLS Network

To establish an end-to-end FlexE connection, hop-by-hop provisioning¹ is required so that at each hop on the connection, the FlexE client, i.e., the Ethernet data flow, on each side of the node is specified correctly. In doing so, a network node (R) has the freedom to select and specify any matching FlexE interfaces in pair. Some examples are shown in Figure 6.1.7-5, where the FlexE (0) of 20G (sub-rate in a 40G PHY) can be mapped to anyone of the following:

- 1) FlexE (1): Same as FlexE (0), i.e., 20G (sub-rate in a 40G PHY).
- 2) FlexE (2): 20G, sub-rate in a 100G PHY.
- 3) FlexE (3): 20G, bonding two 10G PHY.
- 4) FlexE (4): 20G, bonding a 40G PHY and a 10G PHY, with sub-rate 15G and 5G respectively.

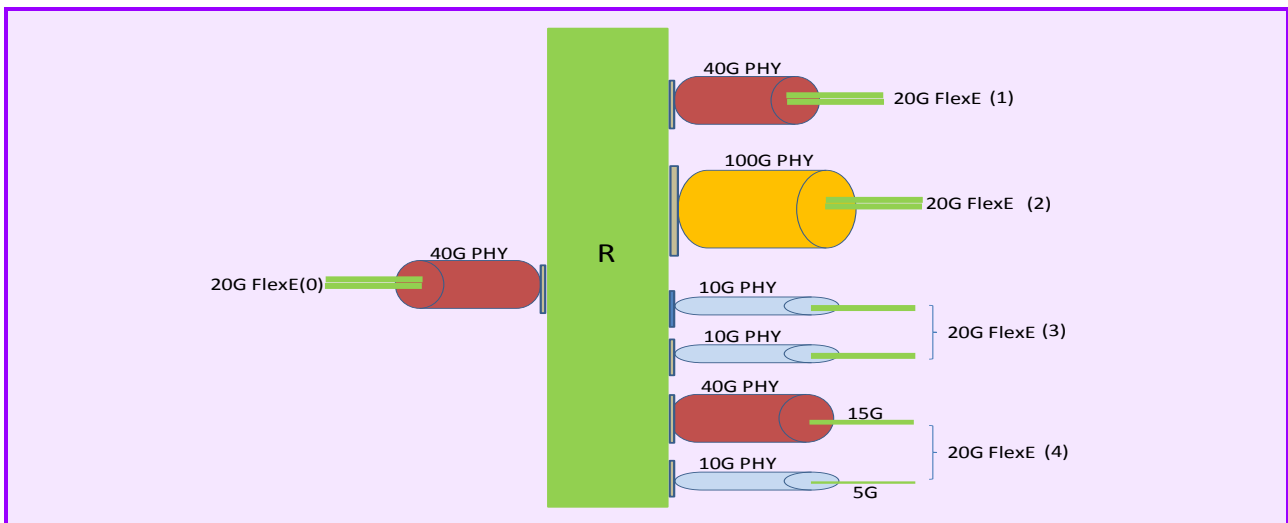


Figure 6.1.7-5 – Example - Client Data Relaying between FlexE Interfaces

¹ End-to-end FlexE connection can be achieved by hop-by-hop manual configuration or via management station, but also can be accomplished by existing control plane protocols such as GMPLS with enhancement. In addition, the connection can also be managed using SDN technology with much agility.

At each hop in this model, a FlexE connection is actually *relayed* or *switched*, i.e., a stream of FlexE client enters the node through an ingress FlexE shim and exits the node through an egress FlexE shim on the same node (refer to Section 7.2.3 of [Ref.6.1.7-1]).

References

- [6.1.7-1] OIF-FLEXE-01.0, “Flex Ethernet 1.0 Implementation Agreement”, OIF, 2016.
- [6.1.7-2] IEEE802.3, “IEEE Standard for Ethernet”, IEEE, 2012.
- [6.1.7-3] NGMN Alliance document, “Description of Network Slicing Concept”, NGMN, 2016, <http://www.ngmn.org/>.

6.2 Open source projects for network softwarization

6.2.1 O3 Project

The O3 project [Ref.6.2.1-1] has been conducted by NEC, NTT, NTT Communications, Fujitsu, and Hitachi, and has delivered an SDN design and Operation guideline, a network orchestration and control framework OSS (e.g., ODNOS), SDN-enabled WAN node control technologies (e.g., MLO: Multi-Layer Orchestrator as a packet/optical integrated control), and SDN software switch OSS (e.g., Lagopus). As our published OSS, Lagopus enables almost 20Gbps flow switching with over 1M flows, ODNOS provides network abstraction model and operation capabilities such as network aggregation, network federation, and network slicing in multi-layer, multi-domain, and multi-service networking environment. MLO provides a system-orchestration function among multiple layers (e.g., between a packet network and an optical network). Technologies and OSS delivered by O3 Project will contribute to end-to-end slicing also for 5G/IoT environment.

References

- [6.2.1-1] O3 Project, <http://www.o3project.org/en/index.html>.

6.2.2 OpenAirInterface

EURECOM has recently created the OpenAirInterface (OAI) Software Alliance (OSA) – www.openairinterface.org – as a distinct legal entity, which aims to provide an ecosystem for the core (EPC) and access-network (EUTRAN) of 3GPP cellular systems with the possibility of interoperating with closed-source equipment in either portion of the network. In the context of the evolutionary path towards 5G, there is clearly the need for open-source tools to ensure a common R&D and prototyping framework for rapid proof-of-concept designs. The current strategic partners include Orange Labs, Nokia Bell Labs, Ecom and TCL-Alcatel. Many top research centers from around the globe are also contributing to this effort. The EPC software is known as openairCN while the access-network software goes under the name of openair5G. The combination of these two software packages currently provides a standard-compliant implementation of a subset of Release 10 LTE for UE, eNB, RRH, MME, HSS, SGW and PGW on standard Linux-based computing equipment (Intel x86 and ARM-based architectures) and a variety of off-the-shelf radio heads as well as some commercially-deployable radio solutions. These components can be used as virtual network functions (VNF) in both fully-virtual OpenStack or bare-metal deployments. Orchestration software such as Canonical Juju can be used to deploy OAI over different cloud platforms. The openair5G software is freely distributed by the OSA under the terms stipulated by a new open-source license catering to the intellectual property agreements used in 3GPP which allows contributions from 3GPP members holding patents on key procedures used in the standard. openairCN is distributed with a standard Apache V2.0 license. The team is working closely with ETSI to harmonize the software license with the intellectual property policy of 3GPP. Some key areas for 5G development including network slicing, SDN concepts in both the core and access networks as well as functional splitting between the radio-head components and access-stratum processing in data-centers (CloudRAN) in order to optimize fronthaul bandwidth.

6.2.3 OPNFV

OPNFV – Open Platform for NFV – is a Linux Foundation Collaborative Project, OPNFV was envisioned to address a number of challenges anticipated by network operators who are pursuing their own NFV strategies. It is the only public forum with a broad community where the industry can pull together the components of

the platform(s) to see if they work together. It is also an incubation environment to try out new software features or hardware components. Establishment of OPNFV resulted from the realization that an open reference platform was needed to validate key NFV concepts, leverage the growing open source community, and accelerate the development and ultimately the adoption of NFV products and services.

OPNFV initially addressed the NFV Infrastructure (NFVI) and Virtualized Infrastructure Manager (VIM). For 2016, OPNFV expanded the scope to address Management and Orchestration (MANO) as well.

OPNFV projects range from those engaging directly with upstream projects, to internal projects that may be classified as system features development (such as service function chaining or high availability), validation and testing (including function, system, or performance testing), tools development (such as installers and controllers) and documentation. Internal projects proposals, priorities, and scope, are motivated by the community, and overseen by the OPNFV Technical Steering Committee.

OPNFV envisions a twice-yearly release cycle, adapting to the release cadence of the upstream projects. Each release select new upstream functionalities that are verified against a Management and Orchestration (MANO) VIM environment, and could ultimately include VNF lifecycle management (VNFM) and NFV Orchestration (NFVO). The goal is not that of defining a complete standard framework for NFV, but rather to provide and qualify a set of common building blocks for specific functions that could even not be all open.

The first release of OPNFV, named Arno, was issued in June 2015 and integrates the results from the upstream projects OpenStack (VIM), KVM (hypervisor), Ceph, (distributed storage), OpenDaylight (SDN Controller framework), and Open vSwitch (software switch), and other communities/blocks. Arno introduced the OPNFV development environment: Continuous Integration, automated deployment and testing, documentation, and tooling. Arno has been demonstrated running on platforms from multiple vendors across the x86 and ARM processor architectures.

In February 2016, OPNF second release – Bramaputra – has been completed. It significantly supports the upstream projects' outcome, addresses multiple technology components across the ecosystem, and makes advances in stability, performance and automation. This second release is lab-ready and improves stability, validation and documentation; platform-level testing of NFV functionality is included. A rigorous upstream integration process allowed including the latest code from partner communities and over 30 accepted projects have contributed.

Brahmaputra offers many deployment scenarios that include additional SDN controllers, installers, deployment options, and carrier-grade features. Continuous integration mechanisms provide a stable framework for deploying and testing.

A recent upgraded release, Brahmaputra 3.0, includes key enhancements to SDN distributed routing, BGP VPN support, Service Function Chaining (SFC), and other Layer 3 infrastructure support. Much of this is addressed via the OPNFV "SDNVPN" project, which has reached deployment with Brahmaputra 3.0 SR.

References

[6.2.3-1] OPNFV: <http://www.opnfv.org>.

6.2.4 ONOS

6.2.4.1 Overview

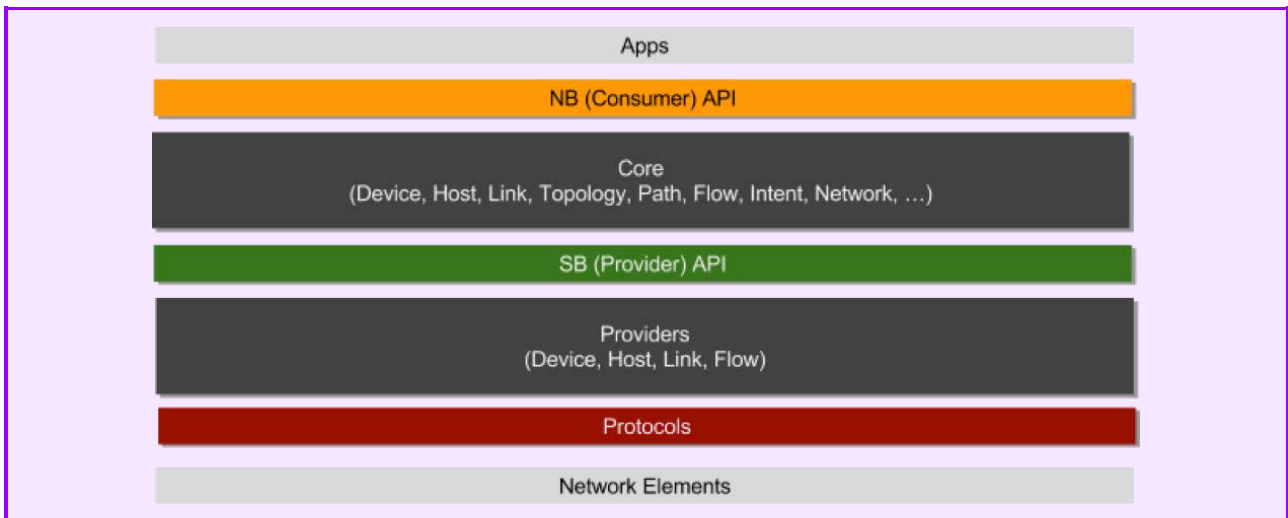
The ONOS provides the control plane for a software-defined network (SDN), managing network components, such as switches and links, and running software programs or modules to provide *communication services* to end hosts and neighboring networks.

ONOS can run as a distributed system across multiple servers

The ONOS kernel and core services, as well as ONOS applications, are written in Java as bundles that are loaded into the KarafOSGi container. OSGi is a component system for Java that allows modules to be installed and run dynamically in a single JVM. Since ONOS runs in the JVM, it can run on several underlying OS platforms such as Ubuntu or OS X.

6.2.4.2 ONOS System Tiers

ONOS is architected with tiers of functionality



6.2.4.3 ONOS Sub Systems

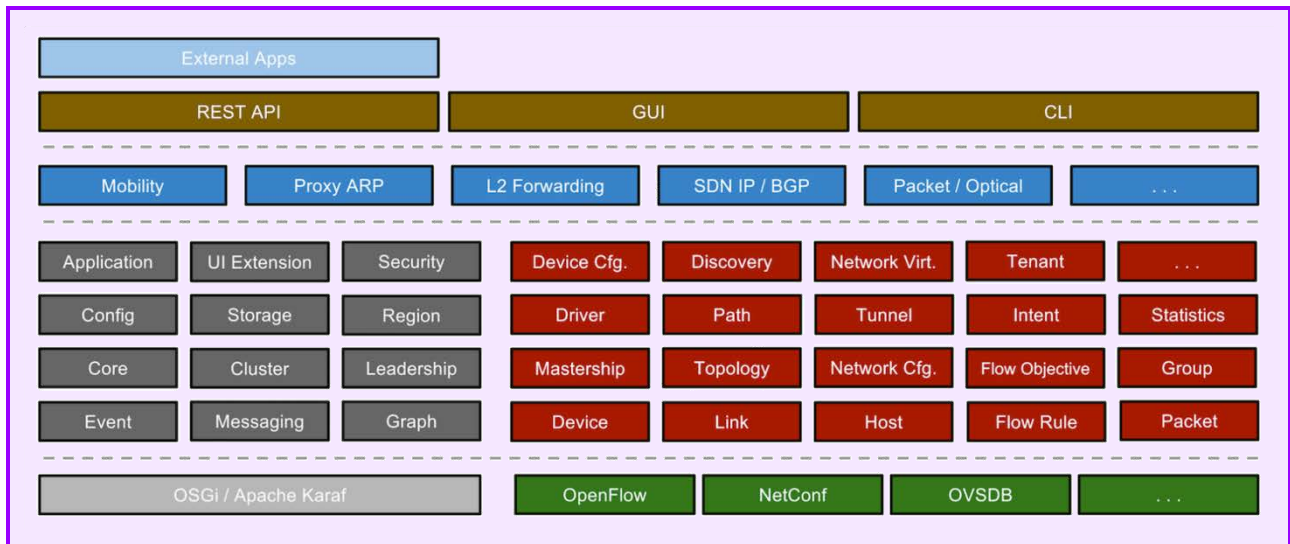
ONOS defines several primary services:

Device Subsystem - Manages the inventory of infrastructure devices.

- *Link Subsystem* - Manages the inventory of infrastructure links.
- *Host Subsystem* - Manages the inventory of end-station hosts and their locations on the network.
- *Topology Subsystem* - Manages time-ordered snapshots of network graph views.
- *PathService* - Computes/finds paths between infrastructure devices or between end-station hosts using the most recent topology graph snapshot.
- *FlowRule Subsystem* - Manages inventory of the match/action flow rules installed on infrastructure devices and provides flow metrics.
- *Packet Subsystem* - Allows applications to listen for data packets received from network devices and to emit data packets out onto the network via one or more network devices.

6.2.4.4 ONOS Architecture

The various subsystems that are part of ONOS are mentioned below



Each of a subsystem's components resides in one of the three main tiers, and can be identified by one or more Java Interfaces that they implement.

6.2.5 OPEN-O

OPEN-O is a Linux Foundation Collaborative Project aiming to build an open source, carrier grade orchestration platform. The primary goal of OPEN-O is to establish an open framework to orchestrate end-to-end composite services across legacy networks, along with emerging SDN/NFV infrastructure. Operators can significantly improve service agility and velocity to increase revenues through OPEN-O, while simultaneously reducing the overall costs.

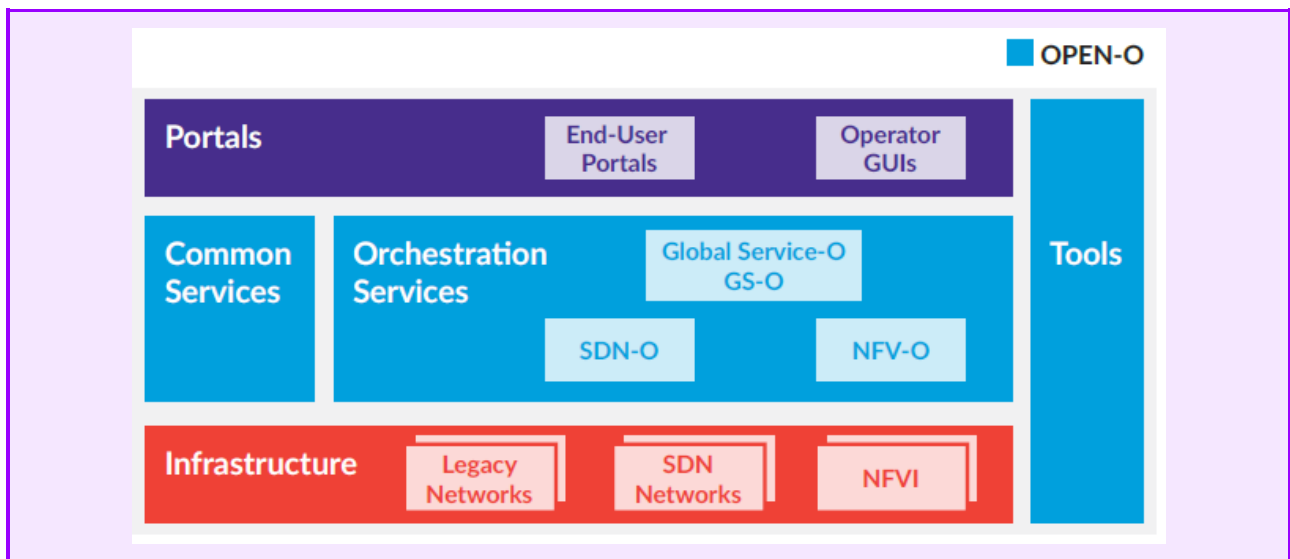


Figure 6.2.5-1 – OPEN-O Architecture

The OPEN-O has proposed an architecture which is aligned with the ETSI NFV Architecture Framework and Management and Orchestration (MANO). OPEN-O consists of a hierarchy of 3 orchestration modules:

- Global Service-O-(GS-O): service-level functional block that enables end-to-end service composition and delivery.
- NFV-O: Responsible for NFV orchestration for diverse Virtualized Network Functions (VNFs) across a wide range of VNF Managers (VNFM) and Virtual Infrastructure Managers (VIMs).

- SDN-O: Oversees network connectivity and network virtualization over an SDN and/or legacy infrastructure typically in conjunction with an SDN Controller (OpenDaylight, ONOS, etc.) and Element Management System (EMS).

Moreover, a set of Common Services are provided for reuse across GS-O, NFV-O, and SDN-O, which includes policy management, security, analytics, logging, and other management capabilities.

OPEN-O also uses YANG and TOSCA for service models, and common, intent-based REST northbound APIs based on the industry standards. OPEN-O is envisioned to operate across diverse NFV Infrastructure, SDN networks, network elements and technologies, and legacy technologies.

OPEN-O announces the first release 1.0, called “SUN”, on November 7, 2016 in only five months since the initial formation of the project.

References

[6.2.3-1] OPEN-O: <https://www.open-o.org/>.

6.3 Prototyping activities for network softwarization

6.3.1 A generalized Operating Platform for Network softwarization

SDN and NFV are rapidly paving the way to the «softwarization» of network functions and services, which are becoming essentially like «applications» executed on distributed logical resources (e.g., Virtual Machines, Containers, etc.).

Network softwarization (which will find first concrete exploitations in the 5G) will deeply integrate IT and Network heterogeneous resources and capabilities (e.g., processing, storage and networking), from the Cloud up to the Edge-Fog computing nodes, devices and even the terminals (e.g., smart phone, but also robots and “things”).

This evolution will bring in parallel the emergence of the X-as-a-Service paradigm: i.e., SDN Controllers, Virtual Network Functions, management and control functions and end-Users’ applications will be all modeled with a unified approach as “services” which are executed on top of logical resources.

In view of this, the new levels of complexity and dynamism of infrastructures subjected to softwarization will require the introduction of proper levels of abstractions, APIs and, above all, automated processes for orchestration and service provisioning.

Today there is a high level of fragmentation, in international fora, bodies, projects and initiatives developing systems, platforms and solutions for management/control/orchestration of future 5G infrastructures. Moreover, it is not predictable today which of said platform(s) will be widely accepted and deployed, and how they will evolve. This context is creating the need for a generalised Operating Platform (OP), defined as an “over-arching and agnostic orchestration space” running on top of currently available (and future) control and orchestrations architectures (e.g., ONOS, ODL, OpenStack, MANO etc.) and capable of “hooking” all of them by leveraging on standard universal set of abstractions (which are still under study).

The concept of a generalized Operating Platform for network softwarization is a lightweight distributed software frameworks operating on top of diverse types of terminals (e.g., robots, machines, smart things capable of executing services tasks and storing local data), through the Network (e.g., Edge Clouds), up to the Cloud Computing (e.g., centralized Data Centers).

6.3.1.1 Towards a unified model for services

Network softwarization will bring a unifying service modeling whereby SDN services (e.g., controllers), NFV services (e.g., Virtual Network Functions), and Cloud services are seen as “application” executed on virtualized resources. In this sense, services are executed in one of more infrastructures “slices”.

A “slice” can be defined as a set of logical resources (e.g., Virtual Machines or Containers) interconnected by a set of virtual links (e.g., Virtual Networks).

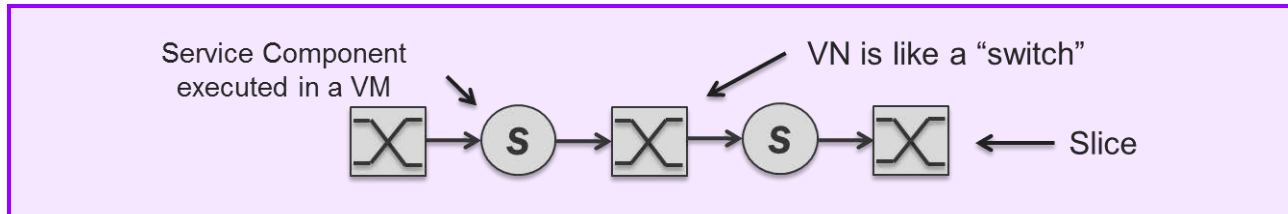


Figure 6.3.1-1 – Example: a service made of a number of service components (S) executed on Virtual Machines (VM) interconnected via Virtual Network (VN)

According the emerging unifying model, a generic service:

- is made of service components or building block;
- can be composed with other services (e.g., service chain, or more articulated service logics);
- provides a function (both “global” and “local”);
- exports APIs (e.g., REST);
- is available anywhere and anytime (location-time independent);
- is scalable, elastic, and resilient.

TOSCA (Topology and Orchestration Specification for Cloud Applications) [Ref.6.3.1-5] is a standard from OASIS that targets interoperable deployment and lifecycle management of cloud services. TOSCA uses the concept of service templates to describe cloud workloads as a topology template. The topology template describes the structure of a service as a set of node templates and relationship templates modelling the relations as a directed graph. Node templates and relationship templates (linking different nodes) in fact specify properties and operations (via interfaces) to manipulate the service components.

In ETSI NFV [Ref.6.3.1-6], a Network Service (NS) is a “*composition of Network Functions and it is defined by its functional and behavioural specification*”, being a Network Function (NF) a functional block within a network infrastructure that has well-defined external interfaces and well-defined functional behaviour.

In IETF, [Ref.6.3.1-7] the term Service Function Chaining (SFC) is used “*to describe the definition and instantiation of an ordered list of instances of such service functions, and the subsequent “steering” of traffic flows through those service functions*”.

Moreover, the YANG declarative data modelling language can be used both to describe deployable instances of a service (e.g., a VNF) and to configure a network device/element at run time.

Eventually, TOSCA and NETCONF/YANG can be considered as complementary instruments: deployment templates may trigger the NETCONF/YANG configurations during the instantiation of services, whilst in the Operations OSS can take over configurations at run time [Ref.6.3.1-8].

6.3.1.2 Overall architecture of generalized Operating Platform

The following figure is showing a representation of the Overall architecture of the generalised Operating Platform for Network softwarization.

In synthesis, the Operating Platform will exploit a “generalised orchestration space” which performs end-to-end services orchestration capabilities (e.g., related the logics of higher services) as well as orchestration capabilities of the ETSI NFVO. The so called infrastructure controller reported in the figure includes the capabilities of the ETSI VNFM.

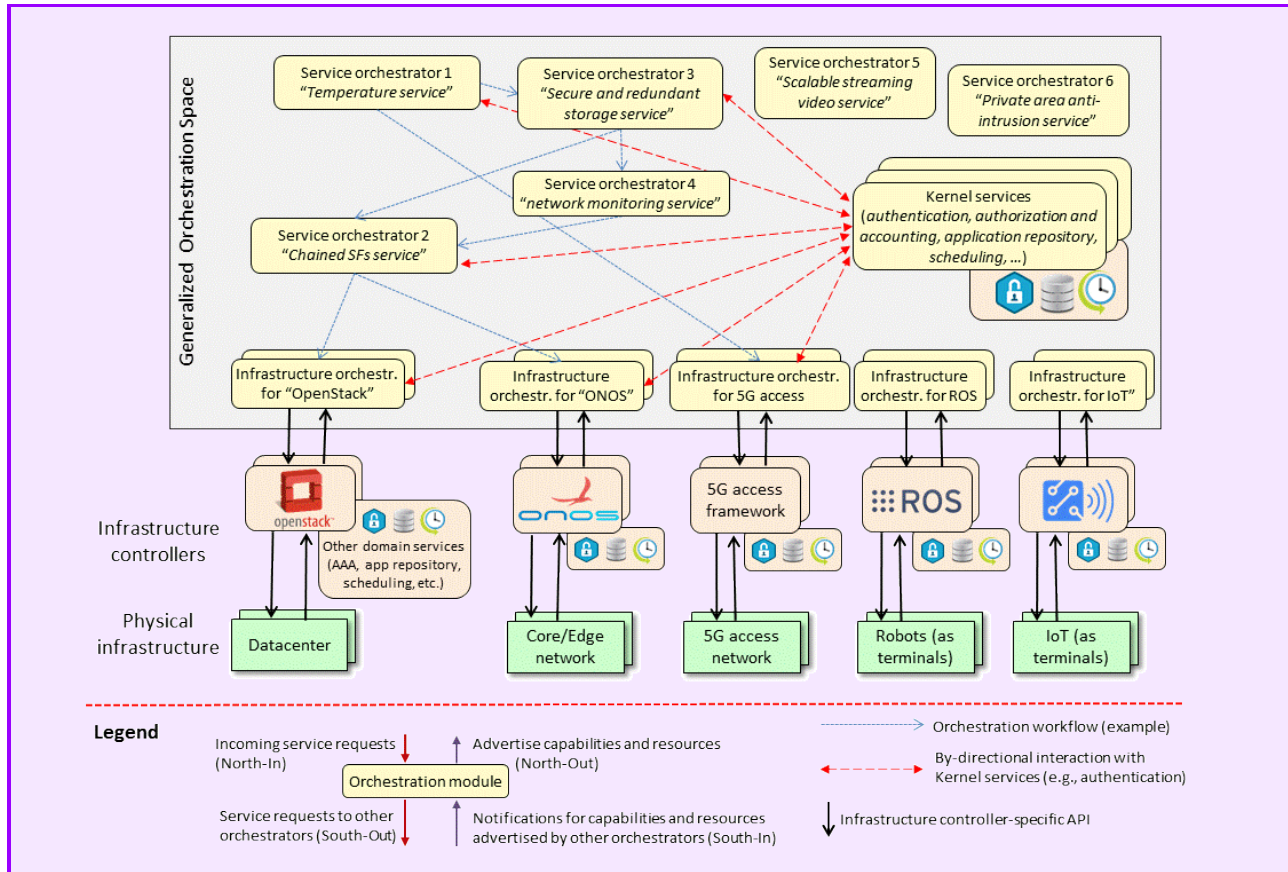


Figure 6.3.1-2 – Overall architecture of the generalised Operating Platform for Network softwarization

Service Orchestrators interact each other via a set of communication/interaction primitives, of which pub-sub represents just one example.

Legacy/proprietary system could be integrated provided that they are exporting towards the “generalized orchestration space” the descriptions of their own domain through APIs through their controllers. Specifically, the domain description can be exported according to YANG data model (both configuration and capabilities). Domain description includes the computational and storage resources that can be provided by the system (e.g., available number of CPUs, available RAM), connectivity capabilities, and the descriptions of other specific services that can be offered.

It should be mentioned that the state of the art of technologies and innovative architectures for orchestration is registering progresses also in the context of several 5GPPP H2020 projects. Notably two examples are SONATA [Ref.6.3.1-9] and 5G Exchange (5GEx) [Ref.6.3.1-10]. However the architecture above described (figure 2) is beyond this state of the art. The concept of "generalized" orchestration space (i.e., highly distributed orchestrators communicating/interacting with certain set of communication primitives) is extending the layering approach proposed in [Ref.6.3.1-9]; moreover this "generalized" orchestration space is expected to be "fully agnostic" with respect to any other available orchestration and control solutions

available today or tomorrow. Same applies also for the comparison with the 5GEx software architecture which it is aiming at cross-domain orchestration of services over multiple administrations or over multi-domain single administrations: in fact, being the "generalized" orchestration space based on an highly distributed framework of orchestrators (basically in peering) the resulted level of flexibility and extensibility is maximised with respect to traditional hierarchical approaches.

6.3.1.3 Demo of a prototype of the generalized Operating Platform

The generalized OP for network softwarization is intended as an "over-arching and agnostic framework" which will perform infrastructure-agnostic orchestration, referring this to the possibility to support the continuous onboarding of new capabilities (e.g., new types of network infrastructures) and resources (e.g., links with more bandwidth), (i) without affecting any already active service and (ii) by allowing new services to take advantages from the new features. This enable the orchestration architecture to be future-proof, being able to support the continuous evolution of infrastructure-level components when they offer new capabilities, add support for new technological domains, or replace an existing infrastructure controller with another software (e.g., an existing ONOS SDN controller is replaced with OpenDaylight).

This prototype demo has shown a simplified version of a generalised OP that is based on a continuous advertisement of capabilities and resources from underlying infrastructure-layer domains, which allows the orchestration to adapt its service logic to exploit the most up-to-date capabilities. The feasibility will be shown to setup a complex NFV service across multiple domains, such as two OpenStack instances connected by an SDN network, where all the service functions (e.g., NAT, firewall, etc.) are launched in the datacenter and the intermediate SDN network is used only to connect all the different components together. However, when the SDN network advertises also the capability to host a given set of network applications (e.g., a NAT), the orchestrator will adapt its service logic and it will instantiate part of the service in the datacenter (e.g., as virtual machines), part in the SDN domain (e.g., as ONOS applications), hence enabling more aggressive optimization strategies in the overarching orchestrator.

References

- [6.3.1-1] D. Soldani, A. Manzalini, et al. "Software defined 5G networks for anything as a service [Guest Editorial]." *Communications Magazine*, IEEE 53.9 (2015): 72-73.
- [6.3.1-2] A. Manzalini, N. Crespi, "An Edge Operating System enabling Anything-as-a-Service", to appear in *IEEE Communication Magazine's* feature Topic: Semantics for Anything-as-a-Service, March 2016.
- [6.3.1-3] D. Soldani, A. Manzalini. "A 5G Infrastructure for Anything-as-a-Service." *Journal of Telecommunications System & Management* 3.2 (2014).
- [6.3.1-4] "FG IMT-2020: Report on Standards Gap Analysis", ITU, TD 208 (PLEN/13), SG-13.
- [6.3.1-5] OASIS TOSCA, "Simple Profile for Network Functions Virtualization (NFV) Version 1.0", available at <http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/tosca-nfv-v1.0.html>.
- [6.3.1-6] ETSI GS NFV 003 V1.2.1 (2014-12), "Network Functions Virtualisation (NFV) Terminology for Main Concepts in NFV" available at https://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf.
- [6.3.1-7] IETF RFC 7498, "Problem Statement for Service Function Chaining", available at <https://tools.ietf.org/html/rfc7498>.
- [6.3.1-8] C. Chappell "Deploying Virtual Network Functions: the complementary roles of TOSCA and NETCONF/YANG", Heavy Reading White Paper (in behalf of CISCO and Alcatel Lucent) <http://www.heavyreading.com/>
- [6.3.1-9] H2020 SONATA project website: <http://sonata-nfv.eu/>.
- [6.3.1-10] H2020 5G Exchange (5GEx) project website: <https://www.5gex.eu/>.

6.3.2 CTTC 5G End-to-End experimental Platform

The fifth generation of mobile networks technology (5G) is not only about the development of new radio interfaces or waveforms. It also deals with the design of end-to-end converged network and cloud infrastructure to facilitate both traditional human-based and emerging Internet of Things (IoT) services. This converged infrastructure, illustrated in Fig. 1, is composed of: i) end-to-end heterogeneous network segments covering radio and fixed access, metro aggregation, and core transport with heterogeneous wireless and optical technologies; ii) massive distributed cloud computing and storage infrastructures; and iii) large amounts of heterogeneous smart devices and terminals for traditional mobile broadband services (e.g., smartphones, tablets, etc.) and IoT services (e.g., sensors, actuators, robots, cars, drones, etc.).

Conducting real-life demonstrations of such a complex system is not easy. CTTC is working in the development of the first-known 5G end-to-end experimental platform for testing advanced end-to-end IoT and mobile services. The approach consists in integrating various existing experimental facilities already available at CTTC which cover activities from the PHY layer to the application/service layer for mobile networks. The building blocks of this demonstration platform are shown in Fig.2. These facilities cover complementary technologies ranging from terminals, sensors and machines, to radio access networks, aggregation/core networks, and cloud/fog computing. Specifically, the five existing experimental facilities involved are: i) the ADRENALINE Testbed® for wired fronthaul/backhaul (SDN-enabled packet aggregation and optical core network, distributed cloud and NFV services in core and metro data-centers); ii) the EXTREME Testbed® and LENA LTE-EPC protocol stack emulator for wireless fronthaul/backhaul and mobile core and RAN (SDN-enabled wireless HetNet and backhaul, edge data center and distributed computing nodes for cloud and NFV services); iii) the GEDOMIS® testbed for LTE/5G PHY real-time prototyping based on FPGAs and software-defined radio (SDR), and the CASTLE testbed (a highly configurable software tool allowing LTE/5G/Satellite PHY layer development and testing); iv) an LTE/5G analog front-end μ wave & mmwave (antenna, power amplifier, filter, mixer, etc.) including digital pre-distortion (SHAPER), and energy harvesting devices for the IoT; and v) the IoTWorld Testbed integrating sensors, actuators, and wireless/wired gateways.

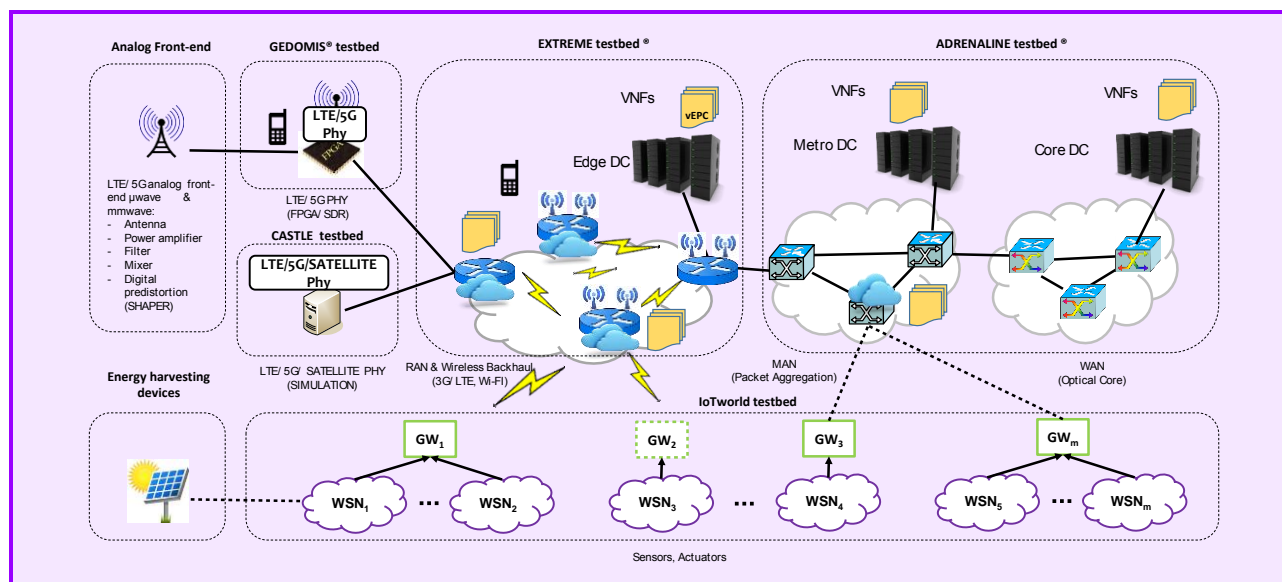


Figure 6.3.2-1 – CTTC 5G End-to-End experimental Platform

ADRENALINE Testbed

The ADRENALINE Testbed (<http://networks.cttc.es/ons/adrenaline/>) encompasses multiple interrelated although independent components and prototypes, to offer end-to-end services, interconnecting users and applications across a wide range of heterogeneous networks technologies for the development and test of 5G services. Different components span IT and networking domains, and allow researchers, system vendors and operators to evaluate experimentally, in conditions close to production systems, all aspects related to cloud computing in distributed environments with multiple geographically split data centers, while jointly managing storage, computing and networking resources.

ADRENALINE includes a multi-technology control plane for multilayer (packet over optical) networks, which manages the networking resources and covers the long-haul core transport and aggregation segments. In brief, a control plane is software that automates the processes involved in the provisioning of networking services, such as optical lightpaths, or Ethernet/MPLS-TP/IP connectivity services. The design of the ADRENALINE control plane follows broad Software Defined Networking (SDN) principles, such as stacking components in a hierarchical setting with different levels of abstraction. Network connectivity services are provisioned by an overarching control orchestration. In particular, at a given domain and layer, the control plane can be based on the GMPLS technology and protocols -- a distributed system in which a dedicated controller is responsible for each node autonomously -- or follow SDN/OpenFlow principles, with a centralized controller that manages all the aspects of a network, dynamically configuring networks according to users' application needs. GMPLS control planes can be augmented with a Path Computation Element (PCE), which is an application or service that assumes specific tasks and responsibilities of the control plane such as computing optimal routes or acting as a central point for connection management (Active Stateful PCE). End-to-end Network Orchestration (to provide an overarching control regardless of the number of domains) is enabled with extensive usage of the Application-Based Network Operation architecture and framework, using the services of the ADRENALINE control plane. End-to-end network virtualization services are performed by a Virtual Network Controller, which is able to provide abstracted multi-layer network views to customers, ensuring security, isolation and independent SDN control (i.e., Customer SDN controllers).

As mentioned, in the all-interconnected context in which end-to-end 5G services may span heterogeneous cloud-computing and networking technologies, ADRENALINE includes an SDN Integrated IT and Network Orchestrator (SINO). A SINO is a centralized system able to coordinate, from a high-level view, cloud and network service management aspects in modern multi-tenant environments which provides the platform to run user applications and virtualized network functions (VNF Manager). A NFV orchestrator is also provided in to deploy end-to-end VNF through VNF Forwarding Graphs. The Cloud Computing service manager is implemented in terms of a modified OpenStack software, one of the top open-source distributed cloud computing systems.

LENA (LTE/EPC network simulator/emulator)

CTTC is the main developer and maintainer of the LTE module of the popular network simulator ns-3 (<http://networks.cttc.es/mobile-networks/software-tools/lena/>). LTE models have been entirely developed at CTTC in close consultation with a small cell vendor (Ubiquisys, now part of Cisco) and are built around industrial small cell forum APIs. As a result, the models are product-oriented and algorithms designed on the simulator can be easily or directly reused in the product.

PHY is simulated through a link-to-system (L2S) interface, which can be easily upgraded to 5G air interfaces, while models from MAC to Application are high fidelity and directly refer to 3GPP standards. New updates are regularly released and include aspects of up to Release 13. Ns-3 is openly available and it offers the opportunity for reproducible research and collaborative development.

Recently, in the context of a grant funded by the Wi-Fi Alliance, the simulator has been extended at CTTC to support the LTE-U and LAA paradigms, with LTE-U Forum and Release 13 compliant implementations, for operation in unlicensed 5 GHz band and fair coexistence with Wi-Fi. Finally, ns-3 also has some unique features at higher layers, including a real-time emulation mode, which allows reusing the code to operate on testbeds, and a capability to compile the source code of real applications and the Linux network kernel for direct use in the simulations. Various such setups have been or are being developed in research projects,

including evaluation of novel RAN functional splits (5GPPP Flex5Gware), mobile layer and transport network layer integration towards an integrated end-to-end network control, or mobile network layer virtual network function deployment (e.g., vEPC) (FP7 COMBO).

EXTREME Testbed

The EXTREME Testbed (EXperimental Testbed for Research on Mobile nEtworks) (http://networks.cttc.cat/mobile-networks/extreme_testbed/) constitutes an experimentation platform that is continuously enhanced and extended with state-of-the-art network management tools, prototyping tools and communication technologies. It can be thought of as a meta-testbed, as it provides a framework for fast deployment of proof-of-concepts. More specifically, its goal is to deploy and run experiments as close as possible to the way one runs simulations. It features a series of administrative and experimentation tools for experiment execution and control over a generic purpose NFV-oriented infrastructure. It features a series of general purpose nodes that can be configured as network nodes or data center servers. Host and guest operating systems can be dynamically loaded to adapt to the need of a given experiment. Integration of OpenStack as virtual infrastructure manager and various flavors of SDN frameworks (e.g., OpenDaylight, Ryu) enables evaluating all sorts of SDN-NFV integration scenarios.

In addition of the generic framework, it also enables the design and integration of more targeted testbeds. For instance, a heterogeneous 802.11ac and millimeter wave mesh network is deployed at the CTTC premises to evaluate an SDN/NFV-managed all-wireless transport network and joint RAN and transport orchestration. Given the distributed computing power deployed throughout the building, this testbed can also be seen as a distributed cloud testbed. Therefore, NFV and MEC use cases are being deployed over this testbed, which will allow evaluating the availability and reliability of these deployments.

Integration with other CTTC research tools and testbeds, such as LENA or the optical networking ADRENALINE Testbed®, as done in projects like FP7 COMBO and 5GPPP 5G-Crosshaul also enables the creation of end-to-end IT and network infrastructures featuring optical and wireless technologies, mobile network layer and transport network layer, and access, aggregation, and core segments. All this orchestrated based on SDN/NFV principles.

GEDOMIS Testbed

The GEDOMIS® testbed (<http://technologies.cttc.es/phycom/gedomis/>) is an ideal platform to develop, test and validate the PHY-layer of modern wireless communication systems covering the prototyping and verification requirements of advanced solutions that target base stations, smart antennas, MIMO systems, Software-Defined Radio (SDR), geolocation, cognitive radio and high-speed test and measurement campaigns. In the past it has been used to develop and test real-time systems based on the IEEE 802.11, IEEE 802.16 and 3GPP rel. 9 standards. GEDOMIS® is able to host PHY-layer prototypes of multi BSs and multi User Equipments (UEs). GEDOMIS® has been used in the past in numerous occasions to implement, test and validate the PHY-layer of various wireless communication systems. The implemented R&D projects were funded either through public competitive calls (at regional, national or European-level) or from direct contract with industrial players. It is worth to lay particular emphasis on two of them, due to their demanding and challenging development and verification cycle. Likewise it is demonstrated the upper bounds capabilities of the use-cases that can be implemented and tested in GEDOMIS®.

CASTLE Testbed

CASTLE is a tool for researchers and industry to test, play and develop over different standards, directly from the cloud, remotely and without installing any software or requiring dedicated hardware. With CASTLE, it is possible to transmit and receive waveforms of different standards over the air and process them locally or remotely. CASTLE is offered as licensed service (free, trial or paid) to CTTC staff or industrial partners via licensing system. CASTLE aims to be the tool where the researchers can develop their own algorithms without the need to implement, modify or extend the standard. CASTLE does it for them. CASTLE provides primitives that interface with different procedures. Researchers can use from top level primitives (such as waveform generation) to bottom level primitives (such as modulators). First, researchers construct their particular scenarios, generate and process waveforms, obtain metrics and analyze results. Researchers do not have

to pay attention to standards' issues. CASTLE offers an API in MATLAB and C++. For Windows, Linux and OSX. CASTLE also allows the management through web interface, inspired in OpenStack. The external access is achieved by means of life-limited license granted by CTTC. With this access, the researcher can use the Test Bed remotely by the web interface and/or the use of an API, which interfaces between the researcher's local machine and remote CASTLE Test Bed. The communication between the researcher and the CTTC is ciphered using a 128-bit AES standard.

IoTWorld Testbed

IoTWorld (<http://iotworld.cttc.es/>) is an End-to-End testbed for the Internet of Things. The main focus is on Wireless Communications systems and data analytics. The testbed has been deployed in two different neighbor buildings: in a laboratory, in an isolated room, and in a real office environment. IoTWORLD is a unique testbed for the Internet of Things. It features: Heterogeneity of wireless technologies, scalable design, integration with 5G technologies and end-user involvement. Different sensors and actuators are connected to a set of gateways, either with a direct connection or via multiple hops. These gateways are then connected to the Internet, providing the capability to retrieve and store data in the cloud, among other functionalities, such as data fusion, compression and analytics. An innovative software middleware has been developed for these gateways. This software makes the integration of new wireless technologies very simple, thus overcoming the heterogeneity barrier. The data gathered by the sensors is stored in the cloud. From there, it is possible to have access to them from a web interface or from a smartphone application. Actuators connected to the IoTWORLD testbed can also have access to these data. IoTWORLD permits to obtain valuable information from the data measurements by means of data analytics on the edge. For this purpose, software defined networking is a key enabler to realize a flexible communication between the different computing entities.

References

- [6.3.2-1] The video of the demonstration presented during Mobile World Congress 2016 is available at: <http://5g-crosshaul.eu/5g-crosshaul-present-at-mwc2016-cttc-demo/>. More videos will be uploaded to <http://networks.cttc.cat> video galleries.
- [6.3.2-2] CTTC website: <http://www.cttc.cat>.
- [6.3.2-3] CTTC Divisions for additional videos and publications: <http://www.cttc.es/research-development/division-departments/>.

6.3.3 Testbed prototyping for network softwarization by NICT

RISE: A multi-tenancy SDN testbed, where multiple SDN-controlled user networks (i.e., tenants) simultaneously work on a SDN-controlled physical network consisting of OpenFlow switches and physical computation servers [Ref.6.3.3-1]. The RISE orchestrator makes a slice consisting of a number of virtual machines and flow spaces of OpenFlow switches and assigns it for each user request. Each user network is then empowered to control the OpenFlow switches and use VMs that are distributedly deployed in JGN-X, a nation-wide R&D network testbed in Japan. VMs can be used such as hosts, network function elements, SDN controllers in the network. Multiple tenants use the RISE simultaneously.

JOSE: An infrastructure as a service specialized for IoT services [Ref.6.3.3-2]. It is intended for multiple large-scale field trials using functional elements such as sensor networks, storage resources, computation resources and networks that connect the functional elements. 20,000 VMs and 1 PByte of storage are provided. The JOSE manager can setup a 200-VM class field trial environment by approximately 13 minutes including VM start-up time, storage setup, VM set up, and user setup.

HIMALIS TB: An ID-based communication testbed on JGN-X. HIMALIS [Ref.6.3.3-3] supports mobility and heterogeneous communication and is compliant with Recs. ITU-T Y.3032 and Y. 3034. The capability of HIMALIS gateways are installed in several routers of JGN-X as distributed mobility anchor (MA) points that avoid single point of failures like a case of Proxy Mobile IP. MAs may be intended to be installed close to BBUs. A registry system and number of mobile hosts (user equipment in other words) are installed in VMs in JGN-X.

Virtualized WiFi: A WiFi network testbed deployed at the NICT Headquarters in Tokyo [Ref.6.3.3-4]. The virtualized WiFi can control the connectivity of a target service by provisioning dedicated base station (BS) resources to the target service or the target WiFi slice. All the decisions on BS selection and handover are separated from the BSs and terminals and put together into a centralized controller, while consistent layer-2 paths for the terminals in the backhaul OpenFlow network are also cooperatively configured. The Virtualized WiFi has potential to connect to core of other virtualized networks.

Abbreviations and Acronyms

- RISE (Research Infrastructure for large-Scale network Experiments)
- JOSE (Japan-wide Orchestrated Smart/Sensor Environment)
- HIMALIS (Heterogeneity Inclusion and Mobility Adaptation through Locator ID Separation)
- SDN (Software-Defined Networking)
- VM (Virtual Machine)
- BS (Base Station)
- WiFi (Wireless Fidelity)
- MA (Mobility Anchor)
- JGN-X (JGN eXtreme)

References

- [6.3.3-1] RISE: A Wide-Area Hybrid OpenFlow Network Testbed, IEICE Transactions on Communications, Vol. E96-B, No. 1, Jan. 2013.
- [6.3.3-2] JOSE: An Open Testbed for Field Trials of Large-scale IoT Services, NICT Journal, Vol. 62 No. 2, Mar. 2016.
- [6.3.3-3] An ID/locator split architecture for future networks, IEEE Communications Magazine, Vol. 48, No. 2, Feb. 2010.
- [6.3.3-4] WiFi Network Virtualization to Control the Connectivity of a Target Service, IEEE Transactions on Network and Service Management, Vol. 12, Issue 2, June 2015.

6.3.4 A network slicing prototype for 5G

6.3.4.1 5G Network Slicing Architecture Model

Network slicing is a key feature in 5G network which enables the operator to create networks customized to provide optimal solutions for different market scenarios, which have diverse requirements, with respect to the functionality, performance and resource isolation. We intend to standardize the slice information model, the interfaces for slice management and slice control as well as the intra-slice/inter-slice interface definitions through the introduction of this section. And this section is well aligned with the end-to-end slice described in Section 8.2, as well as the network slicing scenario in Section 9.4.

The slicing architecture should be easily extensible for various requirements in 5G network. A basic network slice has direct relationships with network resources, network services, network functions, and likely other network slices. The slice virtualizes its resources and provides the service with the resources, and implements various network control/mgmt functions, as shown in

Figure 6.3.4-1. Network resources are utilized to realize network services. The slice model is recursive and extensible both horizontally and vertically to support different scenarios. The construction of network slice is recursive, which means a network slice may be a virtual resource in another network slice. The construction of network slice is also scalable by hierarchical abstraction and the proprietary engineering methods. As an example, the connectivity slice can be constructed of access network slices and transport network slices, which only need to expose certain abstracted interfaces to the connectivity slice as the virtual resources without revealing all the details. As a result, there will be interactions among slices, either in a vertical/hierarchical level or horizontal/peer level.

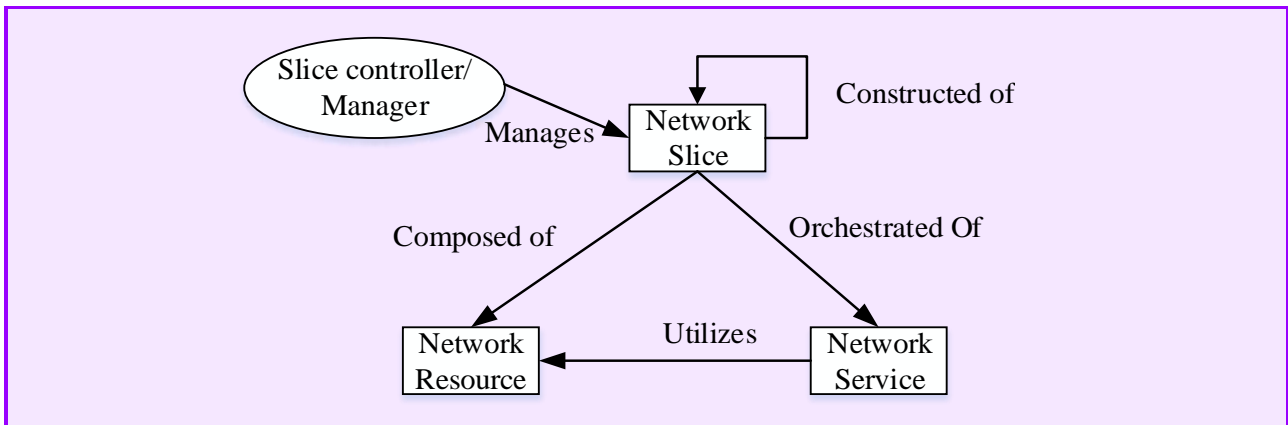


Figure 6.3.4-1 – Basic Network Slice Model

A network slice provides a number of services with dedicated/shared, or physical/virtualized network resources, and implements network functions to control/manage these services and resources. Different slices may provide the same types of services with different QoS. A slice controller/mgr manages the virtual resources and orchestrates them for the user's service requests. A user/application uses services in one slice (likely more than one slice), different services can also share a slice if policy permits it. The service of the upper level slice is an orchestration of the services provided by the lower level sub-slices (intra-domain or inter-domain).

Each network slice can have a set of physical/virtual resources and functions. Vertically, the resources of the upper level slice are composed by the lower level sub-slices grouped in different context (e.g., domains, dependencies, ownerships, share-ability, etc.). The lowest level of resources of the network is constructed by physical resources. A two-level network slice architecture example is illustrated in Figure 6.3.4-2.

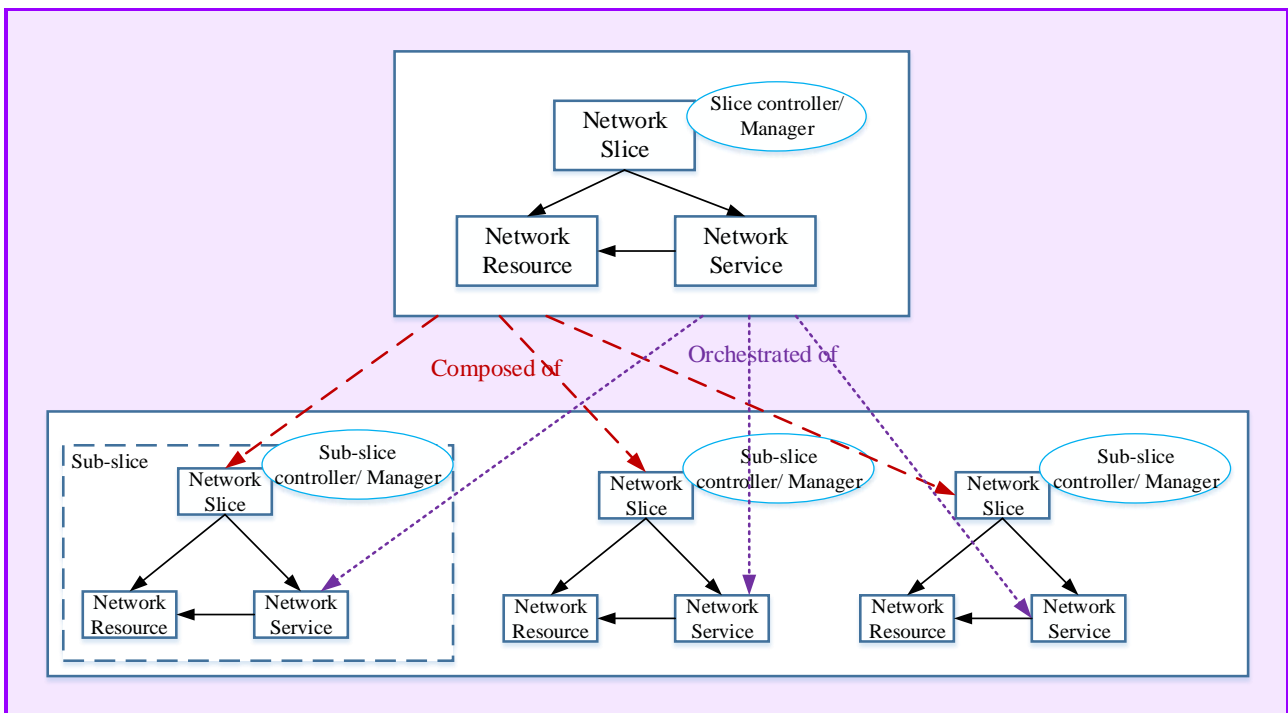


Figure 6.3.4-2 – Two-tier Network slicing example

The network slicing architecture model depicted in this section is aligned with 3GPP's three layer slicing architecture (shown in Figure 6.3.4-3) [3GPP TR 23.799 v0.4.0, 2016-04].

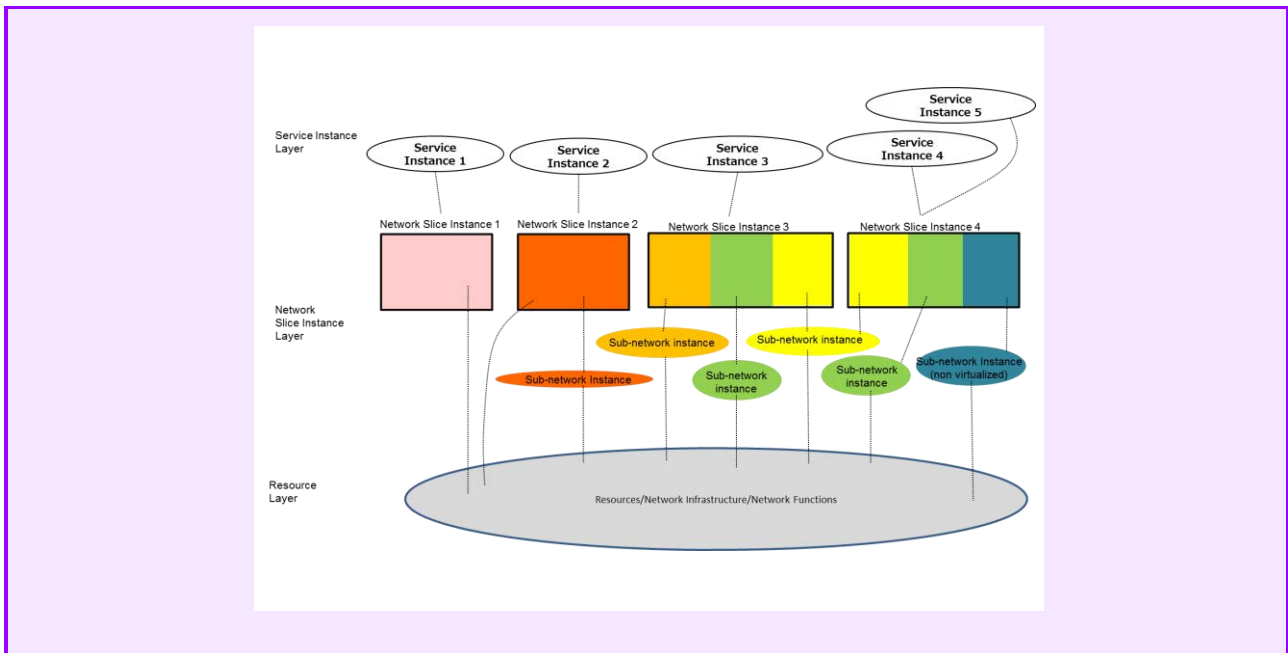


Figure 6.3.4-3 – 3GPP network slicing architecture

6.3.4.2 5G Network Slice Informational Entities

Network slice informational entities are represented by object classes. Each class can have a set of attributes and a set of operations to manipulate the attributes. The controller and NMS can access and manipulate the same object with different operations. Open APIs can be defined between the controller, NMS manager and the service/resource entities. It's an implementation matter whether the controller and NMS should be separated or integrated.

An information model can be used to describe 5G slice entities and their relationships. Figure 6.3.4-4 is an example of showing the resource, service and functional entities of network slices. As an example, a network resource can comprise network equipment, network topology, log record and etc. The network resource tree can be expanded and deeper with sub resources as shown in Figure 6.3.4-4. For example, the logRecord can comprise alarmRecord, which may comprise sliceFaultAlarmRecord for slice fault management. This logRecord object and its subresource alarmRecord object are referred from the management information model defined in ITU X.720. Note that all those examples are for illustrative purpose only. Figure 6.3.4-4 is neither complete nor accurate.

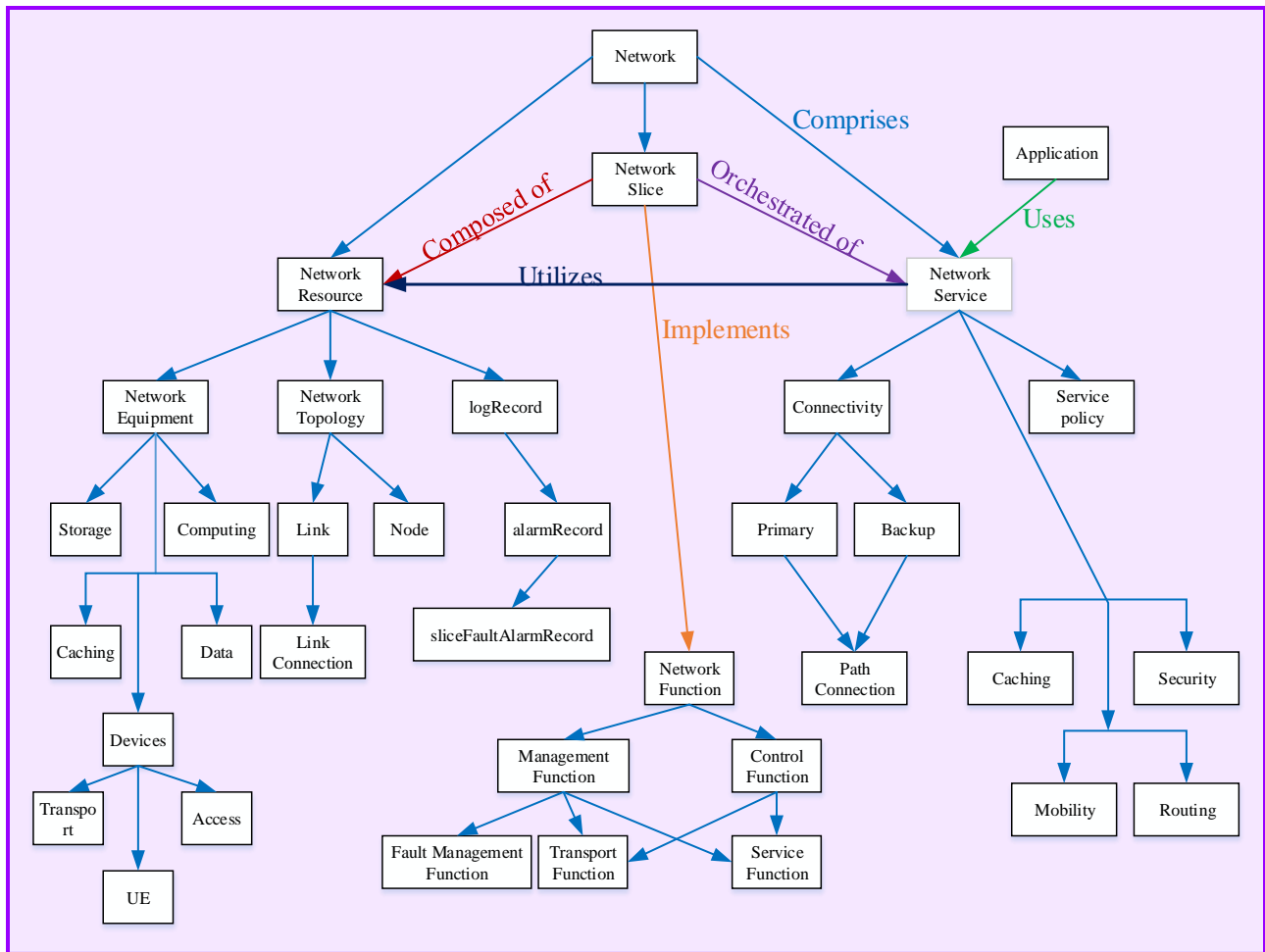


Figure 6.3.4-4 – An example of the slice entities

6.3.4.3 5G Network slicing: Services, Resources and Functions

5G slices are comprised of services, resources and related control/mgmt functions. For a given application scenario, the slices are dynamically created and maintained. Based on the context of the service control/mgmt scope, the slices can be established and managed via a hieratical approach or a federation approach, or a hybrid of the both. Fig 5 is an example of 5G slice engineering model which shows a layered view of end-to-end slices for two video conferencing applications. These applications are multi-party (i.e., multi-source and multi-destination, MS/MD), real-time and interactive, and mobile, over a 5G network connectivity including radio interface, front haul, back haul, MEC access, video service access point, and IP/MPLS/Optical transport backbone network. This example shows the layered slices with a hybrid model of hieratical and federation among slices, based on their context-oriented relationships such as connectivity dependency, geographical topology, mgmt ownership scope, temporal order of the service logics, and the service delivery and demarcation functions, etc.

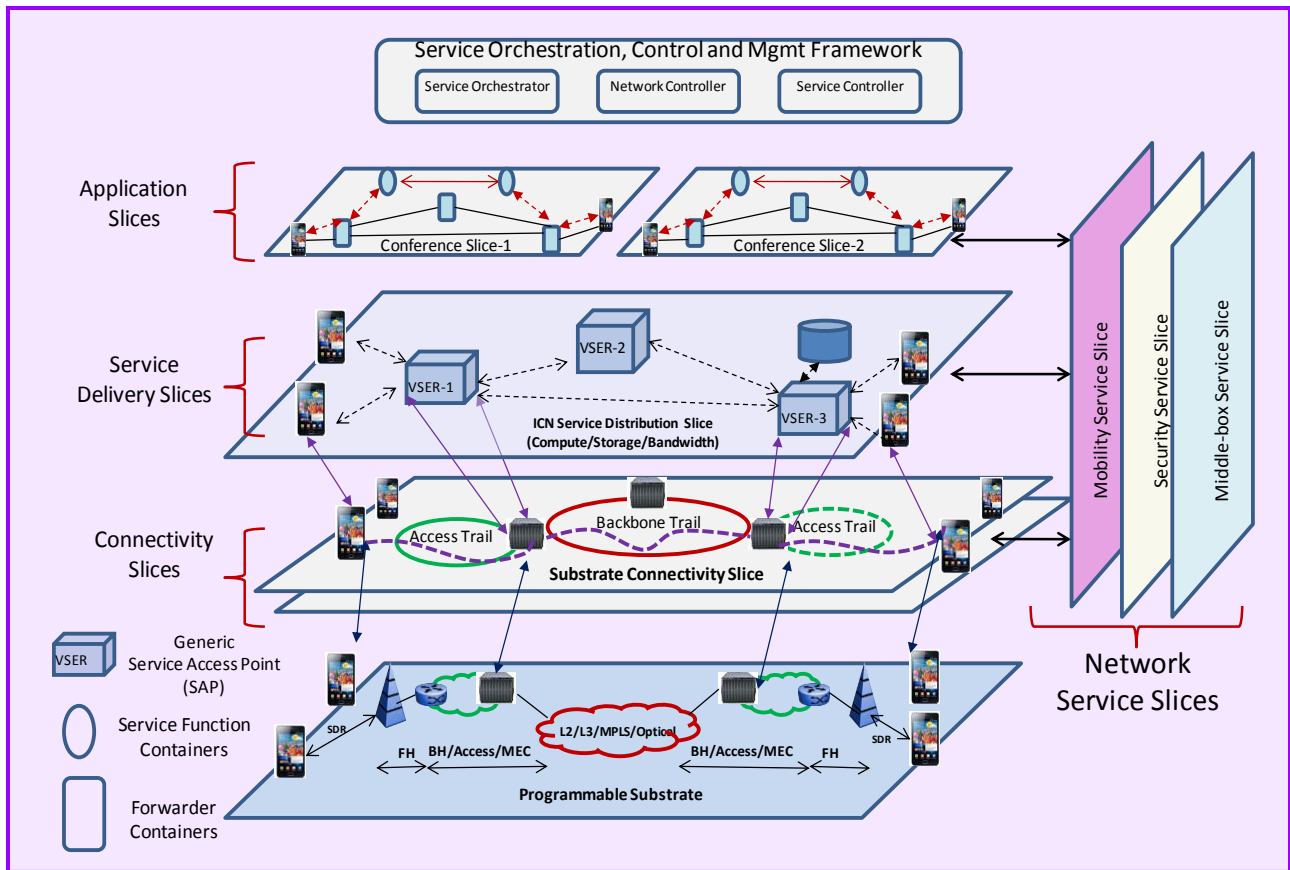


Figure 6.3.4-5 – Layered slices for MS/MD video conferencing applications

In this example, Service orchestration and Mgmt framework provides all the control/mgmt functions for each slice, via a distributed or a centralized manner (or hybrid). In 5G context, this framework can be implemented over open source platforms with the standardized south/north bound interfaces and inter-domain/intra-domain interfaces.

The connectivity slice in Fig 5 can be further decomposed as IP connectivity and channelized optical/packet connectivity with respect to the front haul, back haul and the backbone domains (Figure 6.3.4-6).

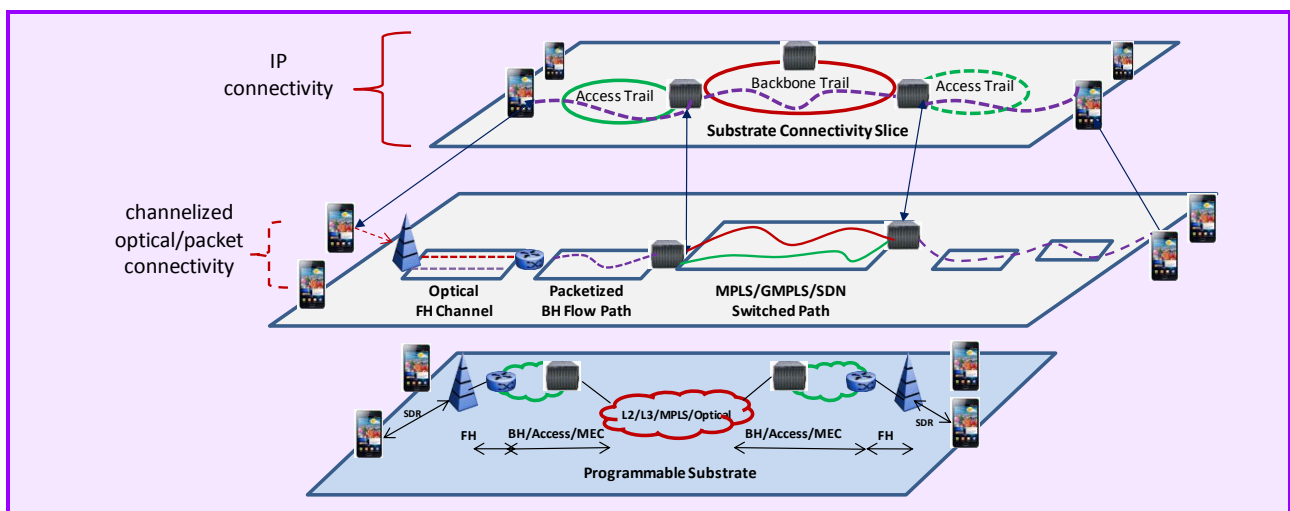


Figure 6.3.4-6 – An example of two-tier connectivity slices

6.4 Work in-progress research projects

6.4.1 5G!Pagoda Project by Aalto University

5G!Pagoda represents the next evolution step in softwarized networks as supported by NFV, SDN and aimed at by the 5G network evolution. The top objectives of 5G!Pagoda are i) the development of a scalable 5G slicing architecture towards supporting specialized network slices composed on multi-vendor network functions, through the development of ii) a scalable network slice management and orchestration framework for distributed, edge dominated network infrastructures, and convergent software functionality for iii) lightweight control plane and iv) data plane programmability and their integration, customization, composition and run-time management towards different markets in Europe and Japan. 5G!Pagoda will develop a coherent architecture enabling research and standardization coordination between Europe and Japan. The proposed developments integrate with a common SDN/NFV based architecture and will additionally provide punctual and highly important developments of the software network architecture. The developments address the next steps of the evolution beyond the immediate NFV standardization and developments, enabling the graceful integration within end-to-end network slices of various highly customized software components, remotely controlling the data path, with specific network function flexibility and network function placement support and easy to manage through a convergent set of scalable orchestration APIs. Besides the technological aspects, 5G!Pagoda will develop a coherent proof of concept with two playground nodes, one in Japan and one in Europe, using a uniform network orchestration and a set of in-slice software features enabling the transparent exchange of knowledge and practical implemented components for dynamic deployment and execution of virtual network functions and applications. The testbed will allow practical demonstration of the functionality and will enable the development of an aligned 5G-oriented standardization roadmap for Japan and Europe



The diagram illustrates the project organization for 5G!Pagoda. At the top, it features the EU flag and the text "EU-Japan Collaboration Project Proposal". Below this, the "5G!Pagoda" logo is prominently displayed in red, with the tagline "A network slice for every service" underneath. To the right, a stylized "5G! PAGODA" logo is shown. The central text describes the project's goal: "Federating Japanese and European 5G Testbeds to Explore Relevant Standards and Align Views on 5G Mobile Network Infrastructure Supporting Dynamic Creation and Management of Network Slices for Different Mobile Services." This is followed by its Japanese equivalent: "サービスに応じたスライス動的生成・管理機能の実証と標準化を目的とする日欧連携 5G 移動通信基盤テストベッド". Contact information is provided for coordinators Tarik Taleb and Akihiro Nakao, including their email addresses and phone numbers. The bottom section displays a grid of logos for the participating organizations: Aalto-yhteisö, Fraunhofer FOKUS, Ericsson, EURECOM, KDDI R&D LABS, The University of Tokyo, Hitachi, Orange, Mandat International, UDG, NEC Networks & System Integration Corporation, and Waseda University.

Figure 6.4.1-1 – Project Organization

References

- [6.4.1-1] 5G! Pagoda website: <http://www.5g-pagoda.eu>.
- [6.4.1-2] Akihiro Nakao, "Application Driven Network Softwarization", Keynote Speech, IEEE NetSoft 2016, <http://sites.ieee.org/netsoft/>.
- [6.4.1-3] Tarik Taleb, Akihiro Nakao, et al., "IF-07: Towards 5G: Mobile Network Softwarization", <http://icc2016.ieee-icc.org/content/industry-panels#IF-07>.
- [6.4.1-4] Tarik Taleb, "Towards 5G: On Network Softwarisation", Keynote Speech, IEEE HPSR 2016 <http://hpsr2016.ieee-hpsr.org>.
- [6.4.1-5] Akihiro Nakao, "Software Defined Data Plane and Applications", Keynote Speech, IEEE HPSR 2016, <http://hpsr2016.ieee-hpsr.org>.
- [6.4.1-6] Tarik Taleb, "Towards 5g : on network softwarization", IEEE PIMRC, <http://www.ieee-pimrc.org/tutorials.html>.

6.4.2 5GPPP H2020 Crosshaul project

6.4.2.1 Introduction

The emerging 5G system requirements, in terms of high capacity, low latency, high efficiency, flexibility, and scalability (i.e. economies of scale), will blur the boundaries between fronthaul and backhaul. Key enablers at the heart of the 5G system, such as SDN, NFV, high capacity transmission media, are commonly envisaged across both fronthaul and backhaul.

It is therefore envisioned that the 5G transport solution will be one where fronthaul and backhaul are truly and flexibly integrated together, opening the transport domain as a service for novel and quickly deployable network applications. Such is the vision of the European H2020 5G PPP (Public Private Partnership) project 5G-Crosshaul [Ref.6.4.2-1], which targets a unified end-to-end packet-based transport design that can address the anticipated challenges of cost, efficiency and scalability. Such solution envisions a seamless integration of existing and emerging fronthaul and backhaul technologies into a converged SDN/NFV-based framework capable of supporting 5G system architectures and performance requirements.

6.4.2.2 Development Focus and Research Challenges of 5G-Crosshaul Project

The 5G-Crosshaul project addresses the following key research challenges, all relevant to IMT-2020 [Ref.6.4.2-2]:

- Develop new or expand existing physical and link-layer technologies such as mmWave and optical wired and wireless to support, at sustainable cost, the 5G requirements in terms of: *capacity, network density, link distance, link budget, energy efficiency, latency, synchronization, cost, and simplified operation*. An example is given by new DWDM technologies (100 Gbit/s direct detection transceivers, reconfigurable wavelength add/drop multiplexers, etc.) based on integrated photonics, which dramatically reduce the cost compared to the devices used in current DWDM metro networks.
- Develop a flexible common frame format that can support the various protocol stack functional splits for the fronthaul profiles envisaged in the 5G RAN (including CPRI) as well as the backhaul traffic, while guaranteeing the individual requirements of each type of traffic in terms of bandwidth and bounded latency. The project concluded that a modular multi-layer switching architecture is necessary to combine the bandwidth advantages of packet statistical multiplexing with time deterministic switching features. Hence, both packet and circuit framing solutions are being studied, the former evolving from Ethernet, the latter based on OTN or alternative frame formats tailored on the Crosshaul network segment (short distances, limited number of add/drop nodes, symmetric uplink and downlink delay, low latency, low jitter).

- Develop a programmable common control infrastructure that leverages the SDN (Software Defined Networking) principles towards a common handling, monitoring, and configuration of the heterogeneous set of technologies (optical, wireless and copper) which compose the integrated fronthaul and backhaul transport network. At this purpose, to be scalable the South Bound interface must hide any unnecessary technology detail to upper control layers but has to expose the minimal set of parameters to enable the orchestration of heterogeneous technologies and resources (networking and processing).
- A certain number of Virtual Network Functions (VNF) are instantiated, connected, and combined over the underlying backhaul/fronthaul substrate, to build operational transport networks. The integrated orchestration of such technologies allows for the dynamic deployment, chaining, and movement of VNFs in order to meet the network demands in a cost-effective way.
- Design context-aware algorithms for concurrent management and orchestration of processing and networking resources, including techniques to reduce the energy consumption of the different elements associated to it jointly across the access and transport domains.
- Enable (recursive) multi-tenant support for sharing the underlying heterogeneous infrastructure in a homogeneous way whilst fulfilling the widely varying requirements of 5G traffic. Indeed, 5G-Crosshaul advocates the introduction of a Partitioner/Slicer Component at each forwarding element to empower the concept of multi-tenancy, allowing multiple parallel tenants controlling the network without affecting each other. It also considers novel techniques on path provisioning and handover for high-capacity, multi-operator, mobile hotspots backhauled via multiple base stations concurrently.

6.4.2.3 5G-Crosshaul transport network architecture

The 5G-Crosshaul transport network architecture is tentatively illustrated in Figure 6.4.2-1. The architecture framework aligns with the architecture in ETSI NFV ISG and embraces the SDN concept including: 1) decoupled data and control planes, 2) logically centralized control, and 3) exposure of abstract resources and state to applications. The SDN framework considered aligns with open-source projects ONOS and OpenDaylight. Data switching for fronthaul and backhaul traffic is primarily packet-based through the Crosshaul packet forwarding elements (XPFE), but for some particular fronthaul traffic requiring extremely low latency circuit-switching is provided through Crosshaul circuit switching elements (XCSE). The 5G-Crosshaul project also advances in the definition and integration of link layer technologies, capable of accommodating the bandwidth requirements of 5G.

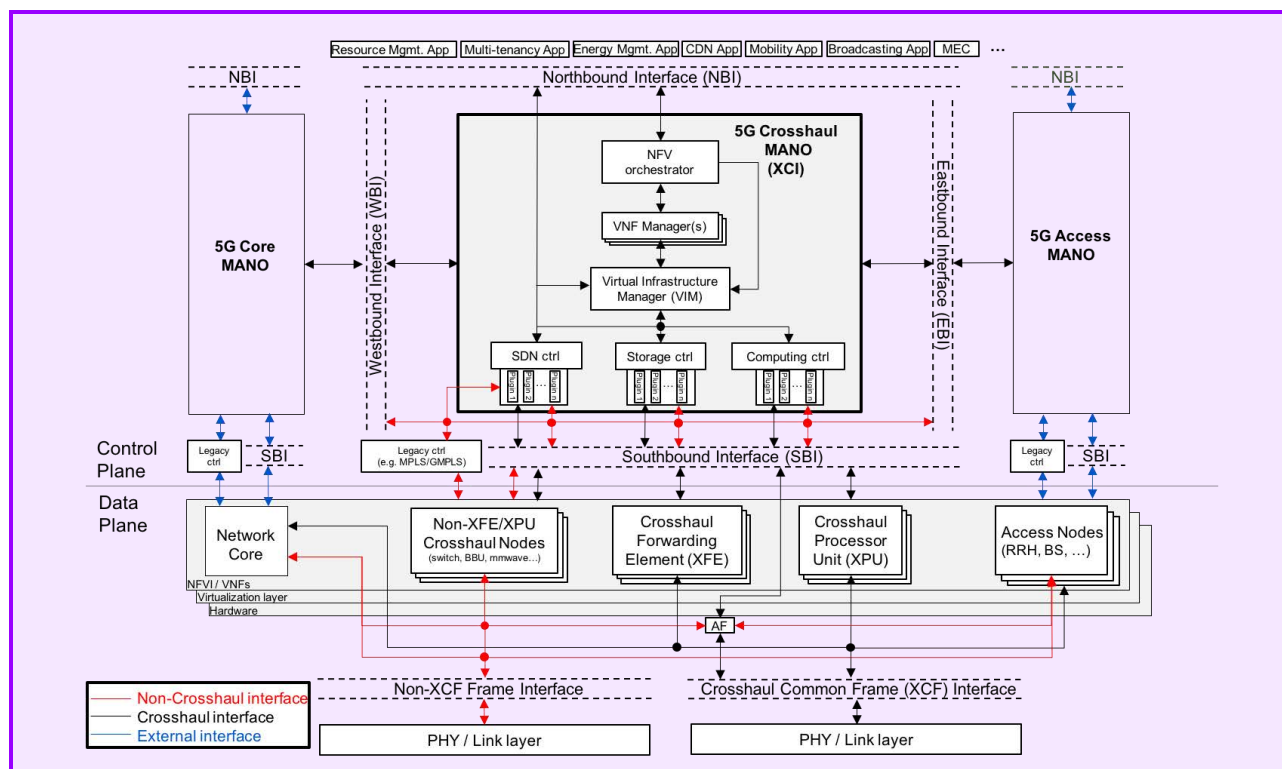


Figure 6.4.2-1 – 5G-Crosshaul transport network architecture

Applications running on top of the Crosshaul Control Infrastructure (XCI) are also being developed, with focus on aspects related to network re-configuration, energy efficiency, media distribution, and mobility management. Examples of these are applications include: 1) Resource Manager Application (RMA) for dynamic network reconfiguration, 2) Energy Management and Monitoring Application (EMMA) for optimisation of energy consumption by activating and deactivating network elements depending on the context, 3) CDN Management Application (CDNMA) and the TV Broadcasting Application (TVBA) for media distribution, and 4) Mobility Management Application (MMA) for mobility management optimization even in the most challenging scenarios (e.g., high-speed trains).

6.4.2.4 5G-Crosshaul components and PoC

The envisioned 5G-Crosshaul transport network solution will consist of high-capacity switches and heterogeneous transmission links (e.g., fibre or wireless optics, high-capacity copper, mmWave) interconnecting Remote Radio Heads, 5G PoAs (e.g., macro and small cells), cloud-processing units (mini data centres), and points-of-presence of the core networks of one or multiple service providers. This transport network will flexibly interconnect distributed 5G radio access and core network functions, hosted on in-network cloud nodes, through the implementation of: (i) a control infrastructure using a unified, abstract network model for control plane integration (Crosshaul Control Infrastructure, XCI); (ii) a unified data plane encompassing innovative high-capacity transmission technologies and novel deterministic-latency switch architectures (Crosshaul Packet Forwarding Element, XFE).

A holistic approach for converged Fronthaul and Backhaul under common SDN/NFV-based control is targeted through key components depicted in Figure 6.4.2-2;

- XCF: Common Frame capable of transporting the mixture of various Fronthaul and backhaul traffic
- XFE: Forwarding Element for forwarding the CrossHaul traffic in the XCF format under the XCI control
- XPU: Processing Unit for executing virtualized network functions and/or centralized access protocol functions
- XCI: Control Infrastructure that is SDN-based and NFV-enabled for executing the orchestrator's resource allocation decisions
- Novel network apps on top to achieve certain KPIs or services

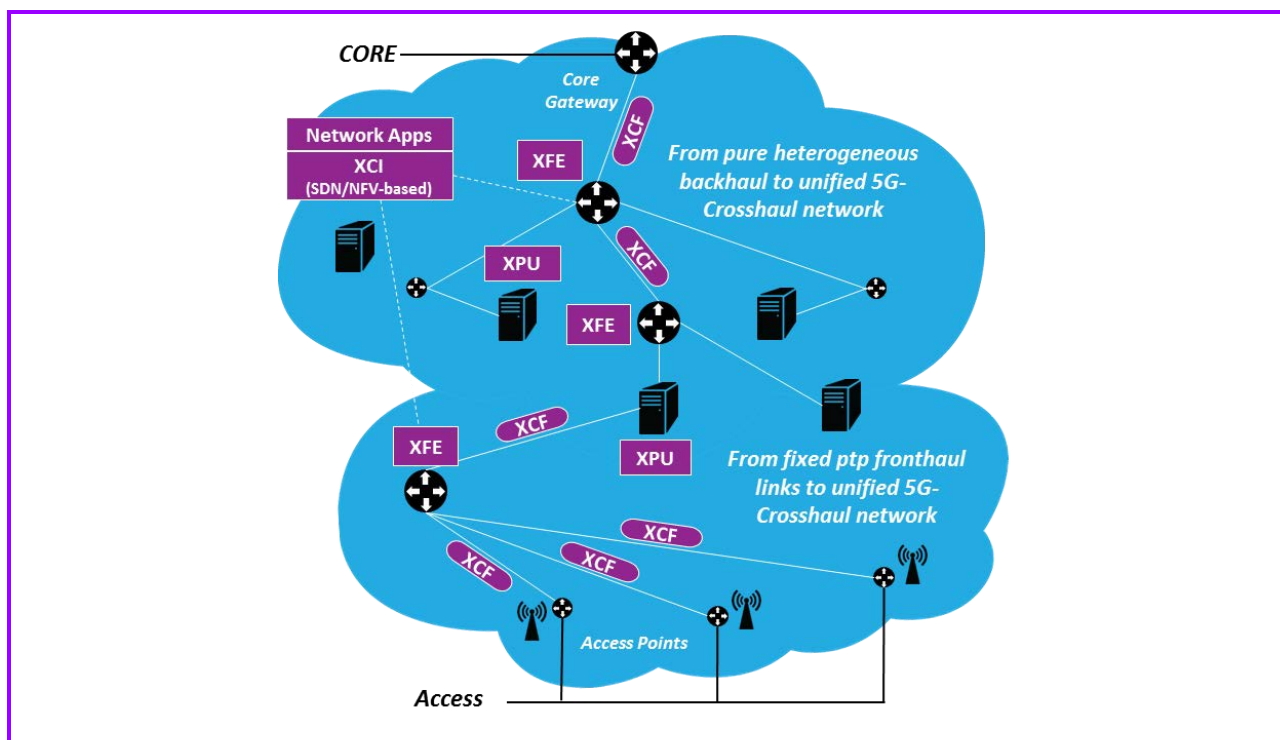


Figure 6.4.2-2 – 5G-Crosshaul components

The validation of the project design will be through proof of concept components integrated together and demonstrated in four testbeds in Berlin, Barcelona, 5TONIC/Madrid, and Taiwan, respectively. Together, all trials will be carried out to verify the accomplishment of the required Key Performance Indicators (KPIs), as well as the challenging objectives described in the proposal.

References

- [6.4.2-1] 5G-Crosshaul website: <http://5g-crosshaul.eu/>
- [6.4.2-2] 5G-Crosshaul deliverable “Detailed analysis of the technologies to be integrated in the XFE based on previous internal reports from WP2/3”, <http://5g-crosshaul.eu/wp-content/uploads/2015/05/D2.1-Detailed-analysis-of-the-technologies-to-be-integrated-in-the-XFE-based-on-previous-internal-reports-from-WP23.pdf>

6.4.3 H2020 5GPPP 5G-XHaul project

6.4.3.1 Introduction

The next generation of mobile networks is supposed to cover a versatile scope of use cases such as massive machine type communications, vehicular communications, or rich multimedia content exchange. The support of these new services by the new radio network, in combination with the ever increasing pressure for cost efficient network design, is expected to have a very strong impact on the transport network.

5G-XHaul aims at developing a converged optical and wireless transport network solution supporting the envisaged 5G end-user and operator use cases, and the corresponding 5G-XHaul transport classes. The envisaged solution has

- dynamically programmable mmWave and sub 6 GHz wireless transceivers,
- time shared optical networks (TSON) and wavelength division multiplexing – passive optical networks (WDM-PON),
- and a software defined cognitive control plane as its main pillars (cf. Figure 6.4.3-6).

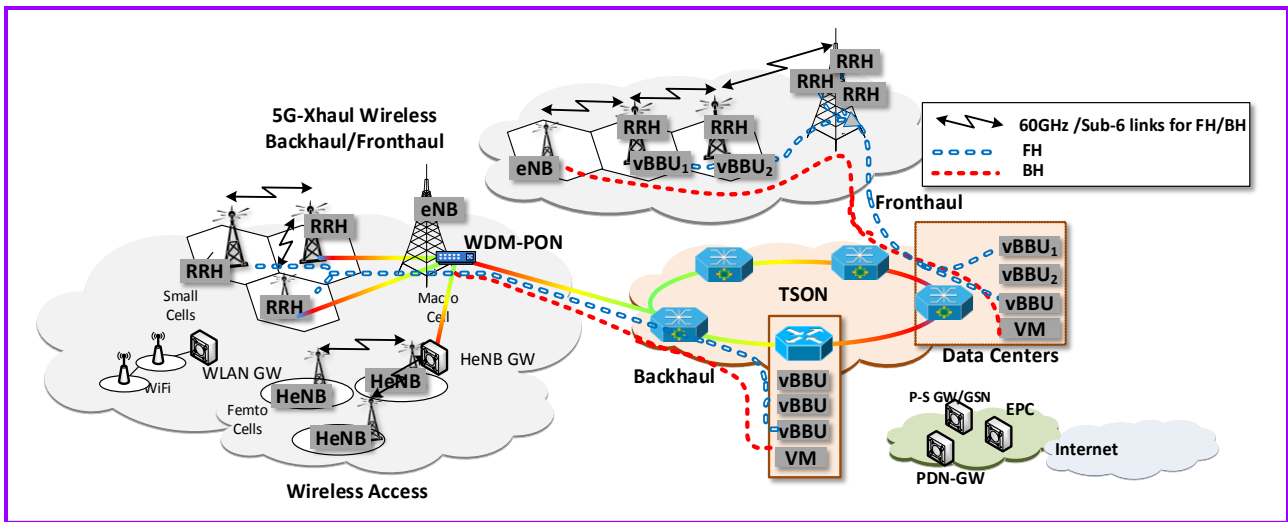


Figure 6.4.3-6 – 5G-XHaul network deployment

6.4.3.2 Development Focus and Research Challenges of 5G-XHaul

In the wireless domain, 5G-XHaul leverages on the SDN controlled mm-wave/Sub-6 GHz wireless transport. While the state-of-the-art wireless transport solutions reach multi Gpbs data rates with ranges of up to 1 km, they are typically realized using relatively expensive technologies. In this context, new mm-wave antenna arrays and beamforming chipsets based on vector modulators will be developed, enabling flexible, robust, and low cost point to multipoint (P2PM) wireless backhauling (BH) and fronthauling (FH) for an ultra-dense network of small cells.

The innovations in the optical transport technology include further developments of the TSON concept with protocol convergence through Ethernet access, very granular bandwidth allocation through a TDM frame/flexigrid. On the other hand, WDM-PON will provide increased data rates up to 25+ Gbps, and an SDN controlled interface.

Finally, 5G-XHaul brings together a heterogeneous set of wireless and optical transport technologies under a logically centralised control plane following the SDN architecture. In particular, the 5G-XHaul control plane will allow the users of the 5G-XHaul transport system to define transport slices specifying how a set of distributed physical or virtual functions are connected through the 5G-XHaul infrastructure. In addition, the 5G-XHaul control plane features a North Bound Interface (NBI) that allows the tenants (in other words, users of the 5G-XHaul transport system, which receive connectivity services over the transport network such as, e.g. virtual 5G network operators) to independently control the functions in their transport slice.

6.4.3.3 5G-XHaul PoC

The most promising technologies in 5G-XHaul will be evaluated using three testbeds:

- The NITOS wireless testbed at the University of Thessaly, Greece, will generate realistic small cell topologies and test SDN control elements.
- The TSON testbed at the University of Bristol, UK, will validate TSON aggregation and core network topologies, and test the TSON SDN control elements.
- Finally, the developed components will be integrated in a city wide wireless-optical testbed in Bristol, UK to validate the end-to-end solution mimicking a real-world scenario (cf. Figure 6.4.3-7).

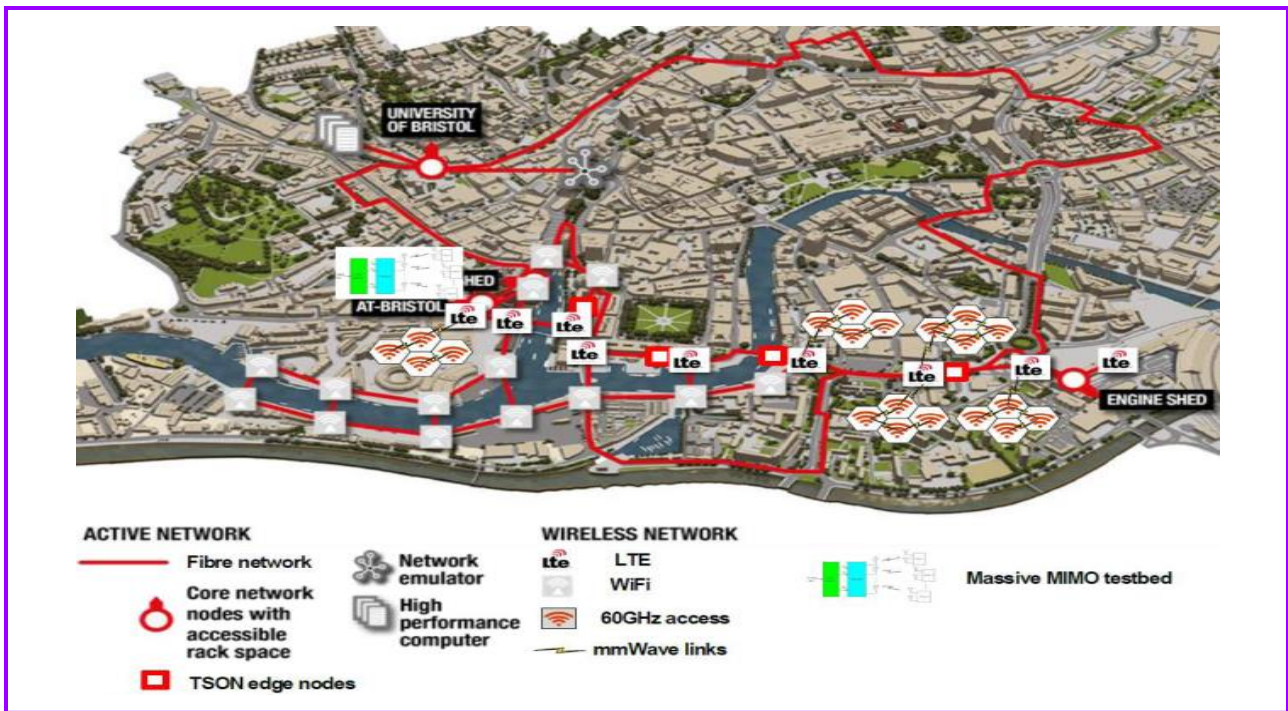


Figure 6.4.3-7 – BIO testbed, Bristol, UK

References

- [6.4.3-1] 5G-XHaul project website: <http://www.5g-xhaul-project.eu/>.
- [6.4.3-2] E. Grass et al. "Dynamically Reconfigurable Optical-Wireless Backhaul/Fronthaul with Cognitive Control Plane for Small Cells and Cloud-RANs", EuCNC 2015, Paris, France, June 2015.
- [6.4.3-3] 5G-XHaul Deliverable D2.1, "Requirements, Specification, and KPIs Document", March 2016.
- [6.4.3-4] A. Tzanakaki et al. "5G Infrastructures Supporting End-User and Operational Services: The 5G-XHaul Architectural Perspective", IEEE ICC 2016, Workshop on 5G Architecture, Kuala Lumpur, Malaysia, May 2016.
- [6.4.3-5] J. Gutiérrez et al. "5G-XHaul: A Converged Optical and Wireless Solution for 5G Transport Networks", Transactions on Emerging Telecommunication Technologies, Wiley, 2016.
- [6.4.3-6] ONF, SDN Architecture, Issue 1, ONF TR-502, June 2014, available at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf.

6.4.4 H2020 5G SONATA project

6.4.4.1 Introduction

SONATA [Ref.6.4.4-1] aims at increasing the flexibility and programmability of 5G networks with a novel Service Development Kit and a novel modular Service Platform and Service Orchestrator; it will bridge the gap between telecom business needs and operational management systems. The scope of SONATA is represented by a new Service Development Kit, the Management System and the Service Platform including: a customizable Service Orchestrator, a Resource Orchestrator, a Service Information Base along with various Enablers; we use here the ETSI reference model as a terminological framework. In fact, while ETSI's division into Service and Resource orchestrator can be mapped onto SONATA's service platform, SONATA is much more flexible in this regard and allows not only replacing these orchestration aspects individually, but even to change the division of work between these functions if so desired. This is achieved by SONATA's microservices-based architecture; e.g., the resource orchestrator corresponds by and large to SONATA's default placement plugin.

6.4.4.2 Development Focus and Research Challenges of 5G SONATA Project

SONATA [Ref.6.4.4-2] [Ref.6.4.4-3] advocates a consistent view of 5G network and compute functions, encompassing a wide conceptual range of such functionality. SONATA functionality covers the Multi-service Control layer and partially Integrated Management and Operation layer and the Application and Business Services Layer. SONATA is also capable of incorporating widely heterogeneous physical resources: various access networks (esp., radio), aggregation & core networks, software networks, data centre networks and mobile edge computing clouds.

A multi-service Control layer is responsible for the creation, operation, and control of multiple dedicated communication network services running on top of a common infrastructure. SONATA's functionality for this layer includes: infrastructure abstraction, infrastructure capability discovery, catalogues and repositories, large number of service and resource orchestration functions as plugins, information management functionality and enablers for automatic re-configuration of running services.

The Business Function Layer maintains 5G application-related functions, organized in Repositories, and DevOps tools necessary for the creation and deployment of services. SONATA's functionality for this layer includes DevOps functionality: Catalogues, Monitoring data analysis tools, Testing tools, Packaging tools, Editors and basic functionality for Application & Service programmability.

Figure 6.4.4-1 depicts the way in which SONATA manages various underlying systems. The core idea is to endow SONATA with several infrastructure abstractions, each of which is custom-tailored to the particular needs of the underlying infrastructure. This allows both simple and complex of Virtual Infrastructure Managers (VIM) to be used, as well as entire networks or single, e.g., OpenStack instances to be integrated.

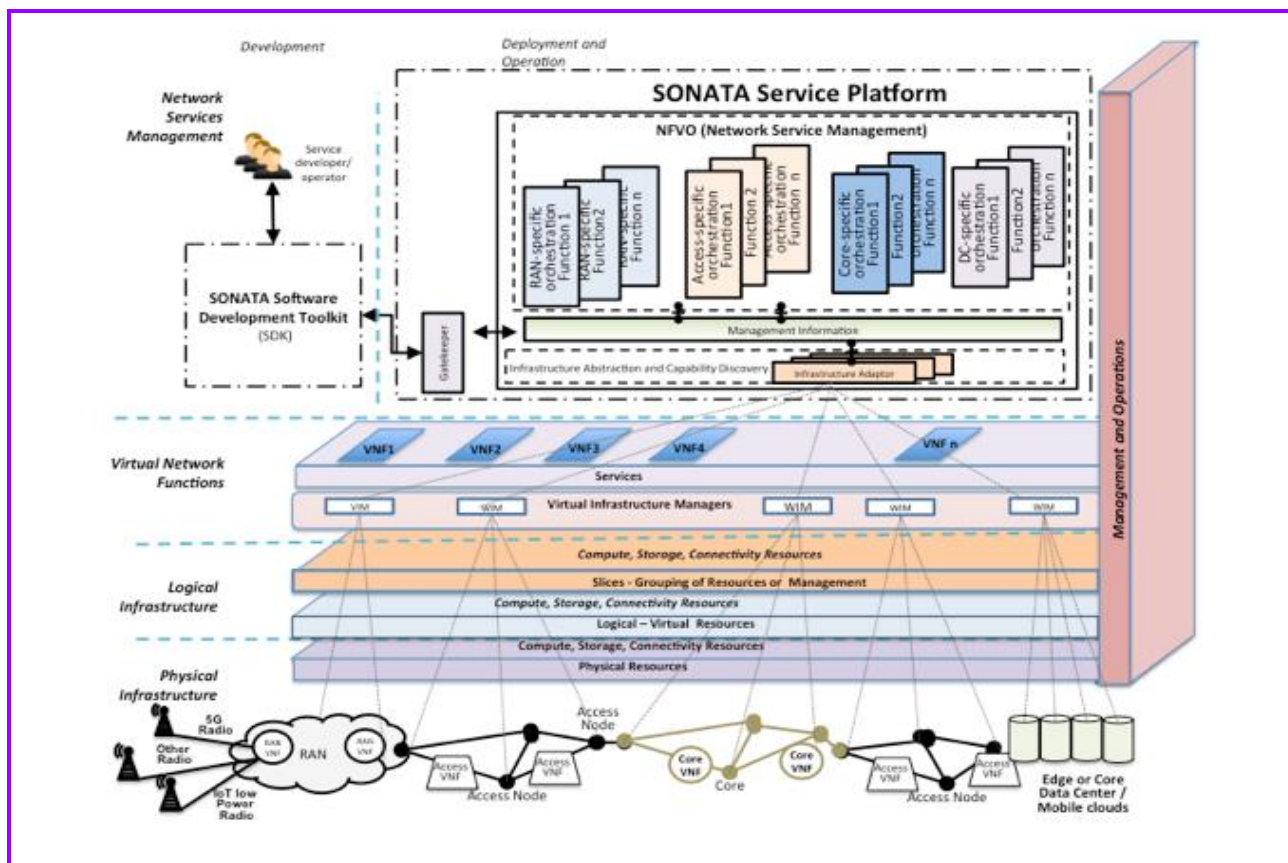


Figure 6.4.4-1 – SONATA's relationship to heterogeneous underlying infrastructures

In summary, SONATA's main contribution to 5G Networking is efficient integration of service programmability, domain orchestration functionality and DevOps functionality. This will maximize the predictability, efficiency, security, and maintainability of development and operational processes around virtualized network functions and chain services

6.4.4.3 5G SONATA architecture

The high-level service deployment procedure is illustrated in the service platform. Each VIM/WIM provides the controlling service platform a view of the available resources and capabilities of its underlying infrastructure/network. A gatekeeper module in the service platform is responsible for processing the incoming requests. The service platform receives the service packages implemented and created with the help of SONATA's SDK and is responsible for placing, deploying, provisioning, scaling, and managing the services on existing cloud infrastructures. For this purpose, it has modules for orchestrating and managing the complete service chain, as well as managing on the VNF level. All artefacts needed to deploy the service can be fetched from catalogues and repositories. The platform can also provide direct feedback about the deployed services to the SDK, for example, monitoring data about a service or its components. SONATA's service platform is designed with full customization possibility, providing flexibility and control to both operators and developers. The core mechanism for this is a microservices-based plug-in architecture (Figure 6.4.4-2): all functionality that is to be provided by an orchestrator is assigned to specific plug-ins, all of which are connected to a message bus that ensures correct delivery semantics of all control messages between these plug-ins. Some of these plug-ins implement exactly one function per orchestrator (e.g., the conflict resolver plug-in, which ensures that any possible resource conflicts between services are resolved in a consistent, service-neutral fashion).

Other plug-ins can be customized by the deployed service itself with caretaking code: these plugins then act as an executive (akin to Microsoft Window's operating system concept) for this service-specific code.

The service developer can ship the service package to the service platform together with service- or function-specific caretaking code, expressing and realizing requirements and preferences. Such caretaking code is referred to in SONATA as Service-Specific Managers (SSM) and Function-Specific Managers (FSM), respectively. SSMs and FSMs can influence the Service and VNF lifecycle management operations, e.g., by specifying desired placement or scaling behaviour. This grants the developer increased flexibility, control and resilience of their service.

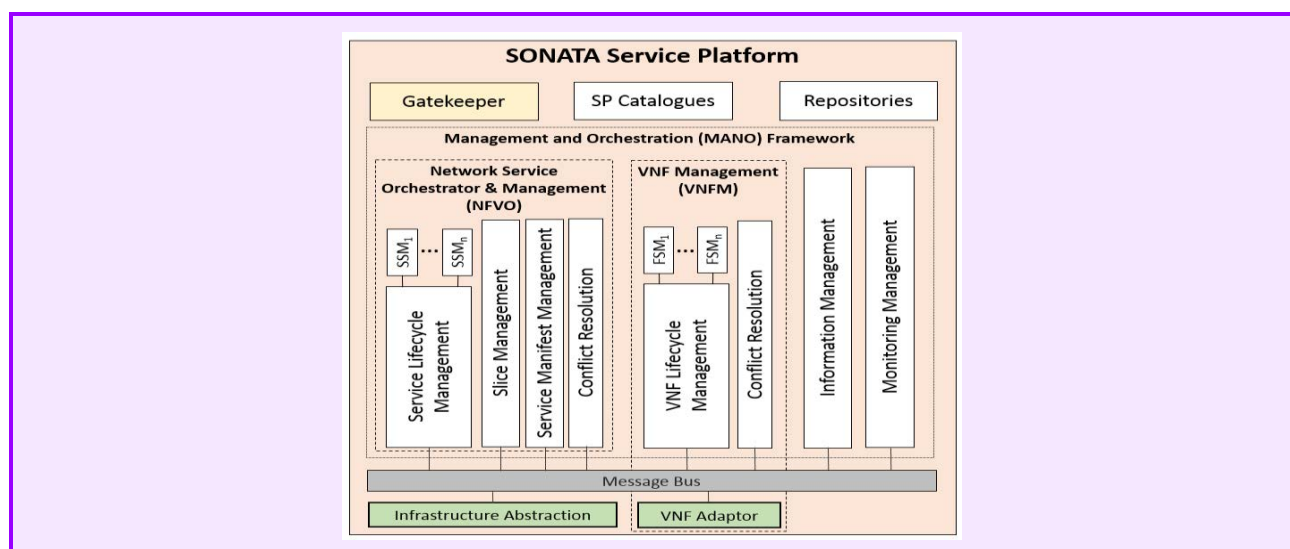


Figure 6.4.4-2 – Main plugins into SONATA's architecture transport network architecture

6.4.4.4 5G SONATA components and PoC

A recursive structure can be defined as a design, rule or procedure that is (partially) explained using a simplified version of itself. In a network service context, this recursive structure can either be a specific part of a network service or a repeated part of the deployment platform. Although different challenges can be thought of, the general idea of reusing existing patterns could reduce complexity and even add more flexible possibilities for extending the service. In Figure 3 recursive orchestration is shown as an SONATA service platform delegating the requested service to another instance of a SONATA platform using a dedicated infrastructure adaptor.

Reclusiveness also leads to an easier management of scalability. Monolithic software entities are prone to performance limitations from a certain workload onwards. Scaling by delegating parts of the service to multiple instances of the same software block is a natural way to handle more complex and larger workloads or service graphs. If this reclusiveness is taken into account from the beginning of the development, the advantages of this approach will come at a minimal cost.

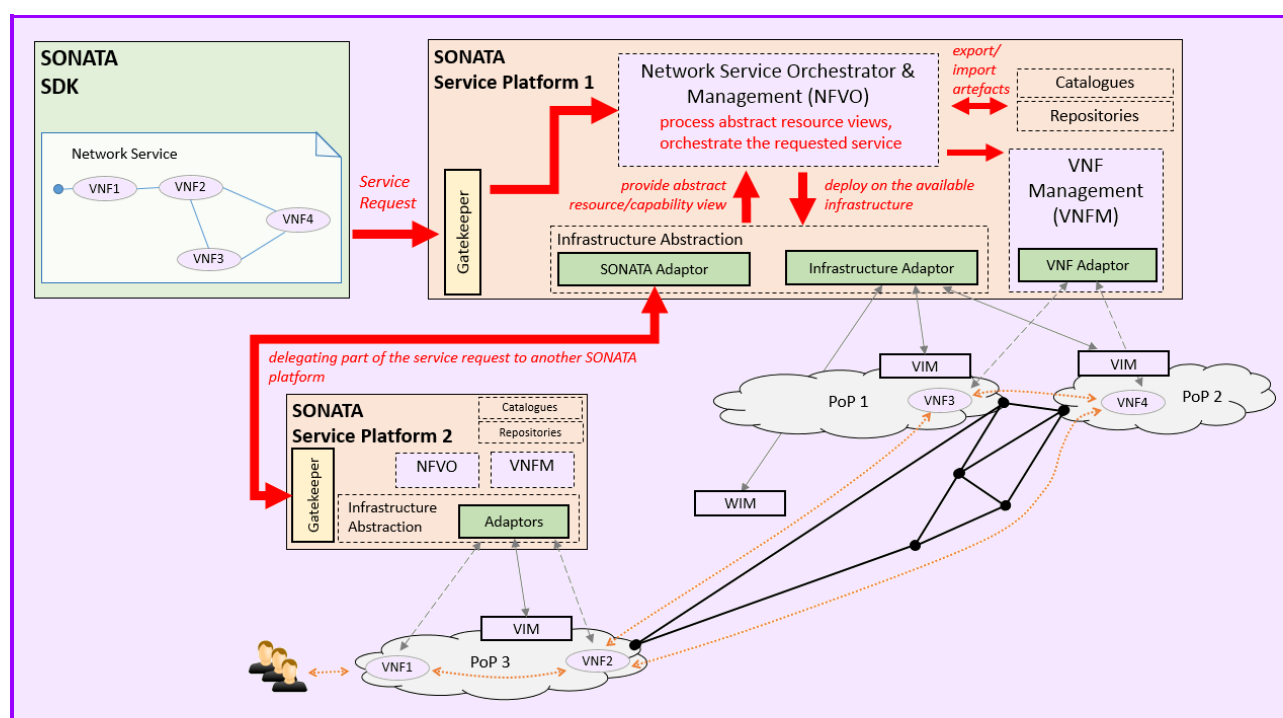


Figure 6.4.4-3 – Service Deployment using the SONATA framework

SONATA Service platform and SDK as open solutions are available for download at [Ref.6.4.4-1].

References

- [6.4.4-1] "SONATA project website," <http://sonata-nfv.eu/>.
- [6.4.4-2] H. Karl, S. Dräxler, M. Peuster, A. Galis, M. Bredel, A. Ramos, J. Martrat, M. S. Siddiqui, S. van Rossem, W. Tavernier, G. Xilouris "Development and Operations (DevOps) for Network Function Virtualization" - Wiley Transactions on Emerging Telecommunications Technologies; – August 2016 [http://onlinelibrary.wiley.com/journal/10.1002/\(ISSN\)2161-3915](http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)2161-3915).
- [6.4.4-3] "Views on 5G Architecture" A. Galis et al - white paper 5G PPP Association July 2016 <https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-July-2016.pdf>.

6.4.5 5GPPP H2020 5GEx project

6.4.5.1 Introduction

NFV does not define orchestration explicitly. Its meaning may be inferred from the NFVO definition Network Functions Virtualisation Orchestrator (NFVO): functional block that manages the Network Service (NS) lifecycle and coordinates the management of NS lifecycle, VNF lifecycle (supported by the VNFM) and NFVI resources (supported by the VIM) to ensure an optimized allocation of the necessary resources and connectivity. Where lifecycle management is defined as: a set of functions required to manage the instantiation, maintenance and termination of a VNF or NS. NFV orchestration is seen as a single concentrated functional block, without delegation. The NFV orchestrator may consider resource availability and load when it responds to a new demand, and may rebalance capacity as needed, including creating, deleting, scaling and migrating VNFs.

Although SDN does not formally define orchestration, the meaning of the concept is apparent from the SDN controller that is expected to coordinate a number of interrelated resources, often distributed across a number of subordinate platforms, and sometimes to assure transactional integrity as part of the process. This is commonly called orchestration. An orchestrator is sometimes considered to be an SDN controller in its own right, but the reduced scope of a lower level controller does not eliminate the need for the lower level SDN controller to perform orchestration across its own domain of control. A provisional definition of (SDN) orchestration might be: the continuing process of allocating resources to satisfy contending demands in an optimal manner. The idea of optimal would include at least prioritized customer SLA commitments, and factors such as customer endpoint location, geographic or topological proximity, delay, aggregate or fine-grained load, monetary cost, fate-sharing or affinity. The word continuing incorporates recognition that the environment and the service demands constantly change over the course of time, so that orchestration is a continuous, multi-dimensional optimization feedback loop. The orchestration process is often discussed as having an inherent intelligence and implicitly autonomic control. Orchestration is also guaranteeing the adequate service performance during the service delivery despite concurrent resource usage among users and service outages.

6.4.5.2 Development Focus and Research Challenges of 5GEx Project

The followings are representing the main expected developments in the 5GEx [Ref.6.4.5-1]:

- Resource Orchestration: automated management of resources in one or multiple Resource Domains to host an NF or a topology of NFs. A resource orchestrator only deals with resource level abstraction and does not understand the service that the NF or topology of NFs deliver.
- Service Orchestration: automated management of a service slice that form a service requested by a customer (network service, cloud service, online service...); a service orchestrator understands the service that the service slice delivers.
- Multi-domain orchestration: automated management of services and resources in multi-technology (multiple domains involving different cloud and networking technology) and multi-operator (multiple administrative domains) environments.

6.4.5.3 5GEx architecture

Figure 6.4.5-1 presents the reference architectural framework [Ref.6.4.5-2] [Ref.6.4.5-3] for organizing the components and interworking interfaces involved in end-to-end management and orchestration in multi-domain environments.

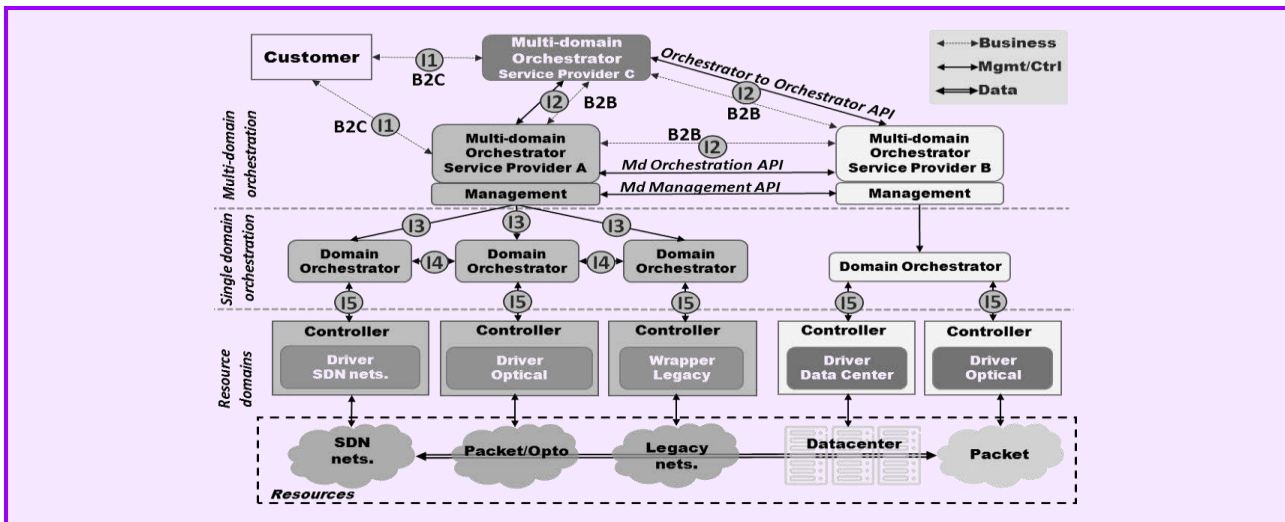


Figure 6.4.5-1 – E2e Management and Orchestration: reference architectural framework

At the lower layer in Figure 6.4.5-1 there are Resource Domains, exposing resource abstraction on interface I5. In the middle layer, Domain Orchestrators perform Resource Orchestration and/or Service Orchestration exploiting the abstractions exposed on I5 by Resource Domains. Interface I4 allows coordination between Domain Orchestrators.

A Multi-domain Orchestrator (MdO) coordinates resource and/or service orchestration at multi-domain level, where multi-domain may refer to multi-technology (orchestrating resources and/or services using multiple Domain Orchestrators) or multi-operator (orchestrating resources and/or services using Domain Orchestrators belonging to multiple administrative domains). The MdO interacts with Domain Orchestrators via interface I3 APIs to orchestrate resources and services within the same administrative domain. The MdO interacts with other MdOs via interface I2 APIs (business-to-business, B2B) to request and orchestrate resources and services across administrative domains. Finally, the MdO exposes on interface I1 service specification APIs (Customer-to-Business, C2B) that allow business customers to specify their service requirements.

The framework considers also MdO service providers, such as C in Figure 6.4.5-1, which does not own resource domains but operate a multi-domain orchestrator level to trade resources and services.

6.4.5.4 5GEx components and PoC

Figure 5 depicts the component-based MdO architecture [Ref.6.4.5-2], including its functional blocks and interfaces to local domain orchestrators and to MdO modules in other administrative domains. MdO modules are grouped in four major functional areas: Exchange of Information and Control (EoIC), Catalogues, Exchange of Functions (EoF), and Exchange of Resources (EoR).

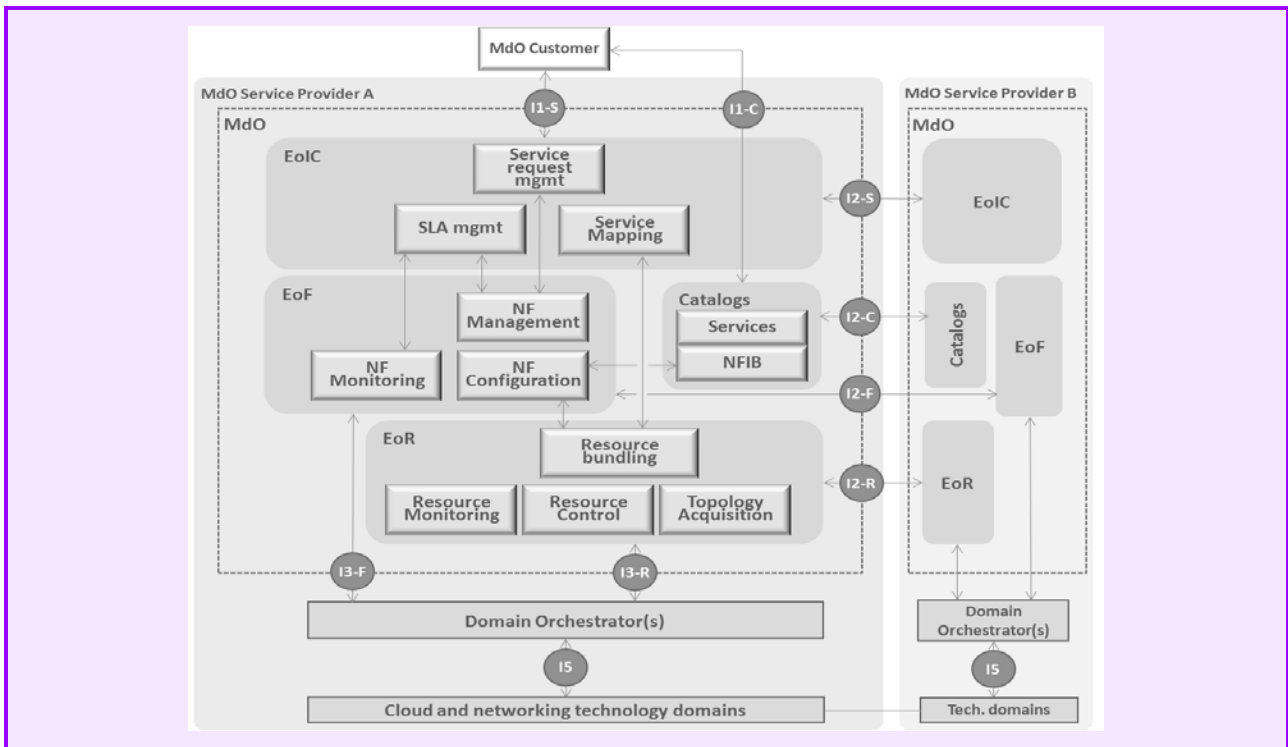


Figure 6.4.5-2 – MDO Functional Architecture Proposal

A. Exchange of Information and Control (EoIC)

The EoIC comprises of functional modules that operate buyer-supplier operations at service level, both for customers on interface I1, and for MdOs belonging to other administrative domains on interface I2. Moreover, the EoIC includes the modules that perform service mapping to topologies of NFs, or service slices, and SLA management.

The Service Request Management module exposes a northbound interface (I1-S) through which an Mdo customer sends the initial request for services. It handles command and control functions to instantiate service slices. Such functions include requesting the instantiation, configuration and interconnection of NFs, as specified by the service graph created by the Service Mapping module, to other Mdo modules in the EoF functional area. It is also responsible for providing SLA templates and SLA management instructions to the SLA Management module in order to assess if the requested service SLA is fulfilled. Finally, it is also acknowledging the result of the service instantiation request to the Mdo customer.

B. Catalogues

The modules exposing repositories of available services and available NFs to customers and to MdOs in other administrative domains are part of the Catalogues functional area.

The service catalogue exposes available services to customers on interface I1-C and to other MdO service operators on interface I2-C. Services are described by service templates, which include a service graph (SG) of NFs, service SLA options, price information and deployment instructions. NFs could either be a basic service component, as described in the NF Information Base (NFIB), or recursively refer to services in the service catalogue. The pricing information of a service can be described as a function of the requirements on the overall graph (i.e. number and location of the end devices) and the functional and non-functional requirements of its component NFs. Service templates are advertised across MdOs in different administrative domains using interface I2-C. For instance, the MdO service provider A in Figure 6.4.5-2 can request services and/or NFs offered by the MdO service provider B and exposed over interface I2-C to provision a certain service to its customers.

C. Exchange of Functions (EoF)

The EoF functional area includes modules that deal with the instantiation, management, configuration and monitoring of NFs. The NF Management module performs lifecycle management operations on individual NFs, which are listed in the NFIB, over interfaces I3-F and I2-F. Performing a lifecycle operation on a given NF may imply reconfigurations of the abstract resources on which it is deployed and/or changes in its operational status (active, inactive, terminated, etc.). Fault management tasks are also handled by this module, such as collecting alarms and notifications from the NF monitoring module. Fault management diagnoses failures in NFs and attempts to repair them. The NF management module provides support for service re-orchestration, performing operations like scaling in/out and migration on individual NFs over interface I3-F and interface I2-F for NFs deployed by other MdO.

D. Exchange of Resources (EoR)

The EoR modules perform resource orchestration, exposing resource slices to modules in EoIC and EoF. Four modules fall in this functional area, dealing with abstract resources and interfacing with underlying domain orchestrators for their realization. The Resource Topology Acquisition module keeps an updated global view of the underlying infrastructure topology exposed by domain orchestrators using interface I3-R for its own domain and interface I2-R for resources in other administrative domains (collected by the respective EoR modules through the corresponding I3-R interface). The topology information provided by the domain orchestrator, or by EoR in other MdOs, is an abstract and limited view of the domain infrastructure resources. For instance, the global view of the infrastructure resources topology gathered by this module may only contain information on aggregates of resources by type, e.g. cloud computing, networking, storage, and geographical location. The topology information is consumed by the service mapping module in EoIC in order to derive a service deployment plan (what are the domain orchestrators chosen to deploy the requested service and what resources are required from them) and accurate pricing information.

The Resource Bundling module aggregates resources belonging to different resource domains, implementing resource slices that may include abstract resources exposed by multiple domain orchestrators, even belonging to other administrative domains.

5GEx platform components as open solutions are available for download at [Ref.6.4.5-1].

References

- [6.4.5-1] "5G Exchange (5GEx) project website": <https://www.5gex.eu/>.
- [6.4.5-2] R. Guerzoni, I. Vaishnavi, D. Perez, A. Galis, et al—" Analysis of End-to-End Multi-Domain Management and Orchestration Frameworks for Software Defined Infrastructures: an Architectural Survey" - August 2016 Wiley Transactions on Emerging Telecommunications Technologies; [http://onlinelibrary.wiley.com/journal/10.1002/\(ISSN\)2161-3915](http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)2161-3915).
- [6.4.5-3] "Views on 5G Architecture" A. Galis et al - white paper 5G PPP Association July 2016 <https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-July-2016.pdf>.

6.4.6 Inter-operator cooperation for the deployment of microcells in private areas

6.4.6.1 Introduction

5G will be a heterogeneous system comprised of radio networks employing various radio access technologies. Different parts of 5G will operate in various parts of the spectrum ranging with completely different propagation characteristics, coverage areas, and interference environments (interference-limited or transmission power limited). Also, very high data rate requirements of IMT-2020 may require extremely low number of users per cell.

We can assume that in this situation different deployment scenarios and business models could be beneficial for operators and end-users. In lower frequency bands including current 4G allocations traditional deployment scenarios may continue to be used, where each operator has individual frequency allocation for exclusive use.

With the increase in carrier frequency, cells will become smaller and more cells per area will be required. After some threshold is passed, it may become economically inefficient for each operator to deploy its own small cells. This may prompt sharing of radio equipment of such small cells by different operators.

Also, with the increase in carrier frequency, propagation loss can become very strong. As a result, network capacity will gradually shift from inter-cell interference limited (where frequency reuse needs to be carefully planned) to transmission power or noise limited, where same frequency can be used by neighbour cells almost without any degradation. This may allow using the same frequency allocations in higher frequency bands by several operators.

Finally, 5G is aiming at serving various types of users. Currently, three groups of use cases have been identified: Enhanced Mobile Broadband, Massive Internet of Things, and Critical Communications. There are tens of use cases within each group and such number will only grow. Different use cases will be served by different network slices. It is also possible that different small cells will be tailored to supporting different network slices for serving different users and/or applications. This will make even more difficult for each operator to deploy all required types of small cells.

This project is focused on designing technical solutions that will provide innovative and cost-efficient ways to address mentioned issues with the focus on three key directions:

- Cooperation between different network operators
- RAN and CN sharing
- Spectrum sharing (in higher frequency bands).

The project is focused on CN design, that is, it is not aiming at designing a new radio technology or RAN.

6.4.6.2 Representative deployment scenario

A representative deployment scenario is shown in Figure 6.4.6-1.

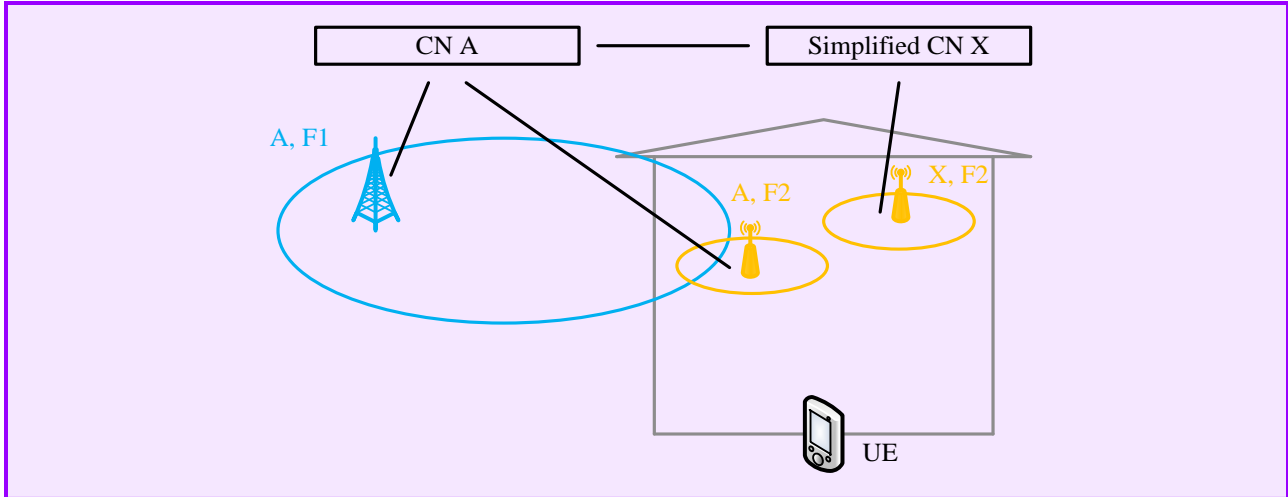


Figure 6.4.6-1 – Network sharing for flexible and cost-efficient 5G coverage in residential, commercial, and industrial properties

Operator A is one of the large-area network operators. It has macro cells operating in lower frequency allocations F1 and small cells operating in higher frequency allocations F2. In this scenario it is assumed that macro cells are used for control plane only.

UE is located inside a local area, for example, inside residential, commercial, or industrial property. It is highly unlikely that each of the large-area network operators will have full coverage by small cells in frequency F2 in all such residential, commercial, or industrial properties. One way to provide such coverage is to introduce operator X.

Operator X is local area network operator (for example, inside residential, commercial, or industrial property) operating small cells in frequency F2 inside such local area. It can be part of a company that owns or maintains the local area. Operator X is aiming at providing specific services inside the local area and is not visible outside it.

The motivation for using operator X is as follows. Small cells of operator X may have better coverage inside this specific local area than small cells of operator A. And/or small cells of operator X are capable of supporting one or several sets of QoS requirements tailored for specific classes of applications compared to micro cells of operator A (for example, tailored for high data rate or low latency applications).

6.4.6.3 Current considerations

Two options to provide a connection to a UE in the deployment scenario described in 6.4.6.2 are currently considered as shown in Figures 6.4.6-2 and 6.4.6-3. These options are motivated by an assumption that operator X will not have fully capable core network supporting all functions. As a result, for some applications operator X will have to outsource control to the core network of operator A. Also, it shall be noted that each option is for one application only. If several applications are running on a UE, a combination of two options can be used.

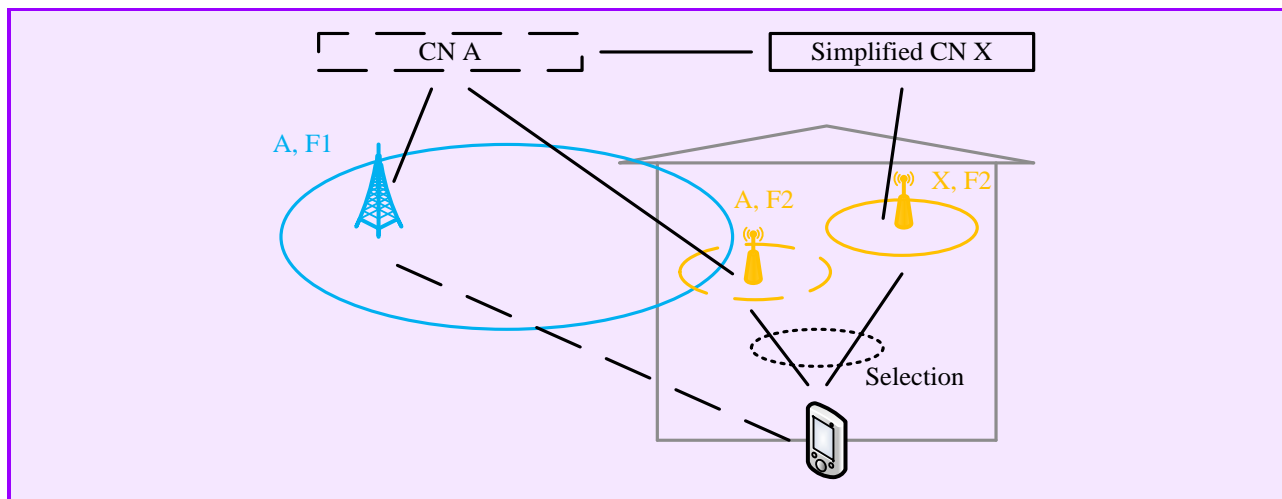


Figure 6.4.6-2 – Connection option 1

In option 1, the macro cell of the large area operator A is used for control information. Small cell of the large area operator A or small cell of the local area operator X is used for data. Selection is based, for example, on coverage, required QoS, etc. for a specific network slice.

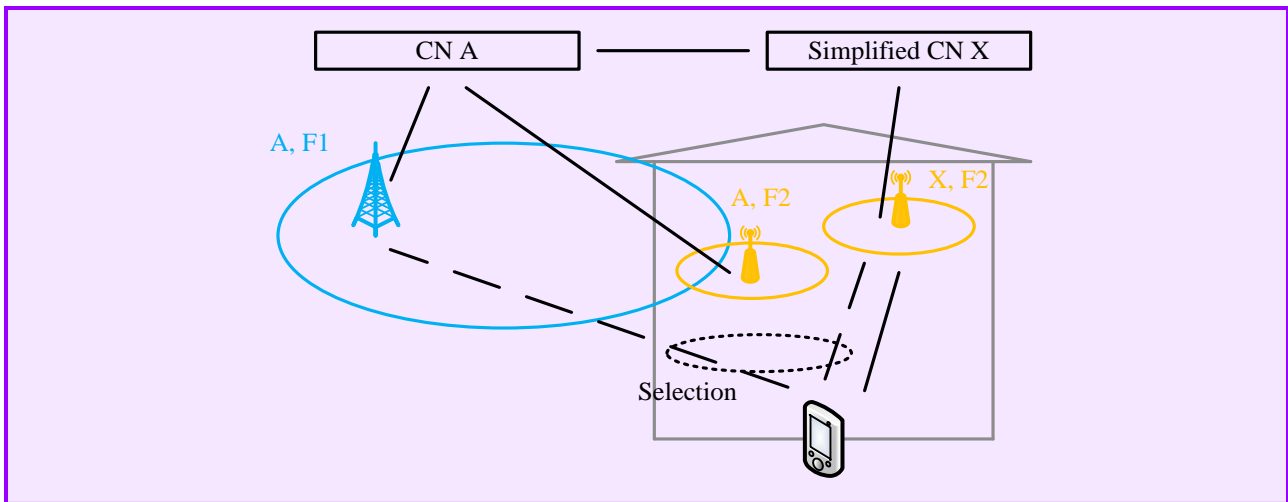


Figure 6.4.6-3 – Connection option 2

In option 2, small cell of the local area operator X is used for data. Macro cell of the large area operator A or small cell of the local area operator X is used for control information. Selection is based, for example, on required core network functionality for a specific network slice.

6.4.7 5G3ALT project by ATR

6.4.7.1 Introduction

“Research and development for realizing 5G mobile communications” or 5G3ALT is a four-year project kicked off in April 2015, that aims to develop technologies for 5G mobile networks. This project is funded by Ministry of Internal Affairs and Communications (MIC) in Japan and is divided among several research centers, companies and universities. The basic plan of this project is described in MIC website [Ref. 6.4.7-1]. As a part of 5G3ALT, Advanced Telecommunications Research Institute international (ATR), Kyoto, Japan, is conducting “research and development on control schemes for utilizations of multiple mobile communication networks” [Ref. 6.4.7-2].

The mission of the 5G3ALT project by ATR is to develop a framework so that multiple mobile communications networks can perform similar to a single network. In addition, the technology of terminals that are capable of communication in multiband and in multi mobile network environment is to be developed. Such a terminal would communicate over a link and with a network that can provide the requested services. In the subsequent subsections, the part of 5G3ALT that is developed in ATR is described.

6.4.7.2 Background and motivation

5G is expected to cover a wide range of use cases with diverse requirements. To achieve this objective, 5G will take advantage of diverse technologies and various frequency bands. However, it is extremely costly for a single mobile network operators (MNO) to provide a universal coverage for various applications and requirements. Network consolidation and virtualization concepts are introduced to realize 5G in a cost-effective way. Consolidating infrastructure and sharing network resources reduces the cost of providing various services for MNOs.

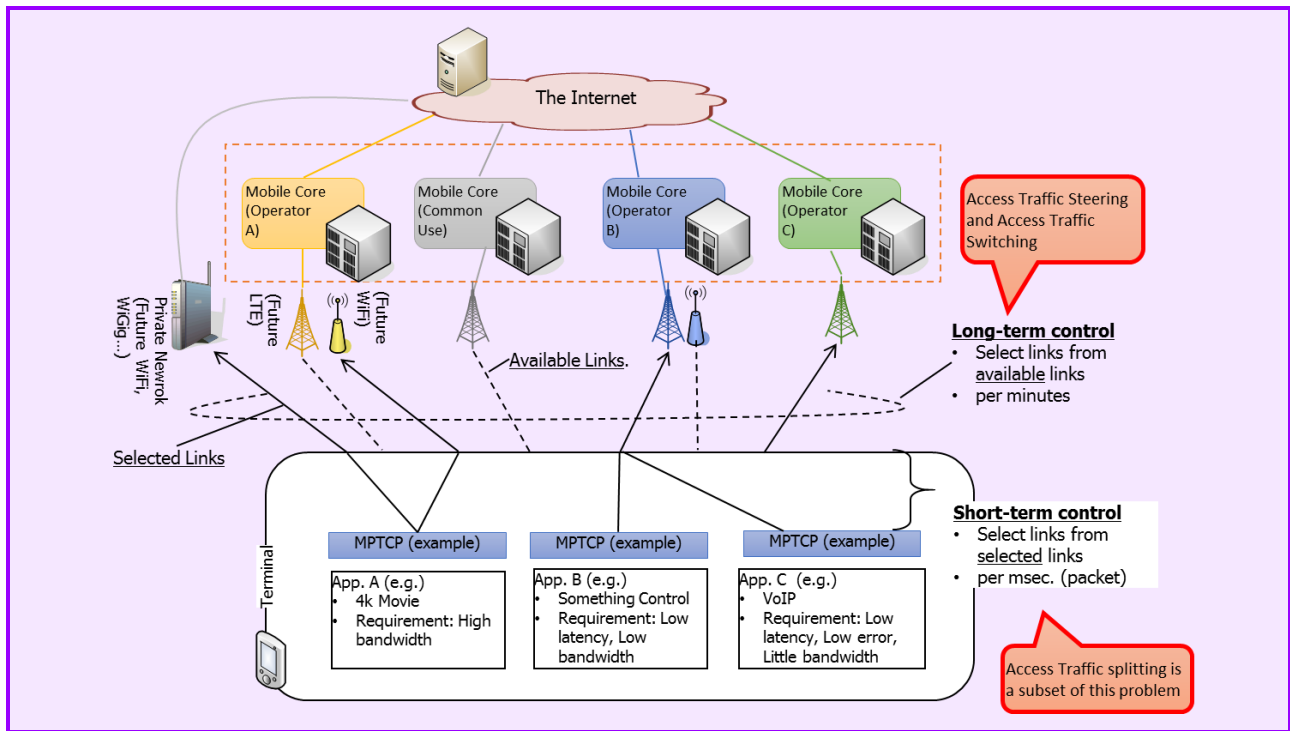


Figure 6.4.7-1 – The high-level architecture of the multi-operator heterogeneous network that is addressed in 5G3ALT project by ATR

Network sharing can be implemented in several ways. The most well-known one is the roaming service, in which the two networks are completely detached, and one provider provides service to the other provider's subscribers when they are out of the coverage of their original service providers. Another way is to completely merge several networks together. The third sharing method is an intermediate approach that falls between the two above mentioned network sharing systems. In this approach, RAN and network infrastructure are shared among two or more operators while the operators are still distinct. This approach preserves the individuality of operators while letting them to benefit from other providers' infrastructures. However, it brings about challenges in resource management and link assignment for subscribers. The focus of 5G3ALT project is on addressing such challenges and developing a framework for an efficient network sharing.

The high-level architecture of the system that is addressed in 5G3ALT project is shown in Figure 6.4.7-1.

6.4.7.3 Objectives and challenges

The main focus of 5G3ALT project is on the resource management in a multi-operator heterogeneous network. The resource management is handled in two levels: long-term and short-term [Ref. 6.4.7-3]. Long-term control is responsible for selecting and associating one or more base stations (BS) or access points (AP) to a service for a subscriber, while short-term control decides which of the associated links is to be used for each packet transmission.

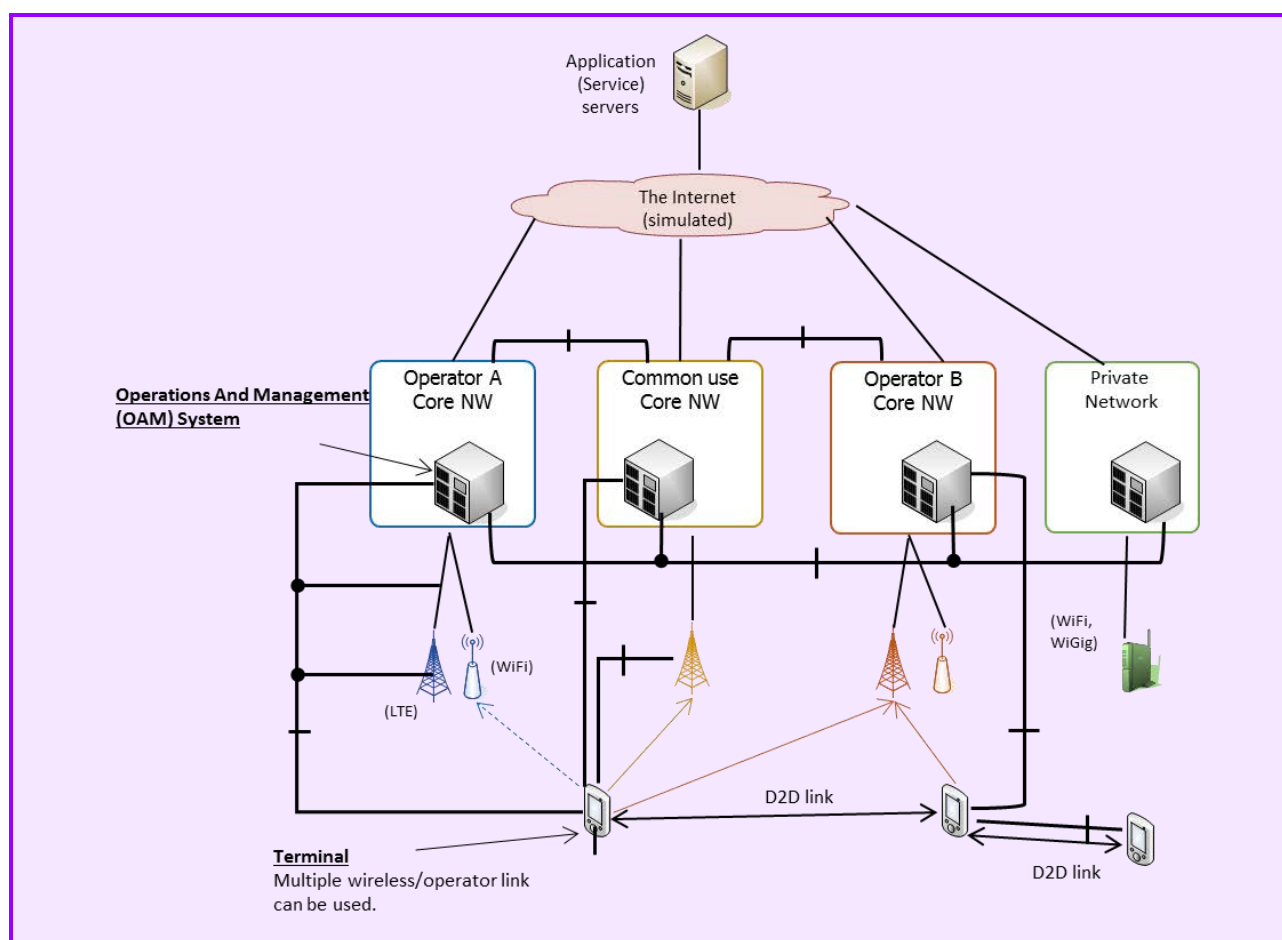


Figure 6.4.7-2 – An example of interfaces and network architecture for resource management in a multi-operator heterogeneous network

5G3ALT intends to provide a solution that considers the QoS requirements of each application and responses accordingly. To be more specific, during the long-term control phase, BSs and/or APs are to be selected/associated considering the particular requirements of each service, such as data rate, latency, mobility, or reliability. The idea that the operators run heterogeneous networks in general, complicates this issue. Network capabilities need to be shared among operators to some extent. Also, the requirements of the requested service should be considered. Various interfaces are needed to convey these requirements as well as networks' capabilities and availabilities. A general example is shown in Figure 6.4.7-2.

The goal of 5G3ALT can be summarized in two parts. The first part is to design the architecture and required interfaces to convey necessary information for management. The flow of the necessary control protocol is also developed in this part. The second part is to design the management algorithm that makes the long-term/short-term control decisions according to service requirements and networks' capabilities.

6.4.7.4 Related key issues in 3GPP standardization

5G3ALT project follows the trend in 3GPP standardization and aligns its developed system to standard specifications while actively participating in standardization process. Following the study item FS_SMARTER in 3GPP [Ref. 6.4.7-4], the work item (WI) "New Study on Architecture and Security for Next Generation System" has been approved in 3GPP TSG SA#71 (March 2016) [Ref. 6.4.7-5]. This WI consists of two work items: FS_NextGen (SA2) and FS_NSA (SA3). FS_NextGen (SA2) impacts the technical report TR 23.799 [Ref. 6.4.7-6]. This technical report lists a number of key issues and several solutions for each issue. Among these key issues 5G3ALT project is mainly concerned with the followings:

1. Key issue 2: QoS framework (Clause 5.2 of [Ref. 6.4.7-3])
2. Key issue 20: Traffic Steering, Switching and Splitting between 3GPP and non-3GPP Accesses (Clause 5.20 of [Ref. 6.4.7-3])
3. Key Issue 17: 3GPP architecture impacts to support network discovery and selection (Clause 5.17 of [Ref. 6.4.7-3])
4. Key Issue 19: Architecture impacts when using virtual environments (Clause 5.19 of [Ref. 6.4.7-3])
5. Key Issue 9: 3GPP architecture impacts to support network capability exposure (Clause 5.9 of [Ref. 6.4.7-3])

References

- [6.4.7-1] Basic Plan of 5G3ALT project by Ministry of Internal Affairs and Communications, Japan (in Japanese), http://www.soumu.go.jp/main_content/000349194.pdf.
- [6.4.7-2] 5G3ALT project by ATR (in Japanese), <https://www.acr.atr.jp/acrmain/?p=191>.
- [6.4.7-3] R. Rezagah, N. Kawanishi, H. Shinbo, "Considerations of Resource Association in 5G Communication Systems," Proc. of the 2016 IEICE Society Conference, B-17-22, Sep. 2016.
- [6.4.7-4] 3GPP TR 22.891 V14.2.0 (2016-09), Feasibility Study on New Services and Markets Technology Enablers, Stage 1, (Release 14).
- [6.4.7-5] TD SP-160227, New Study on Architecture and Security for Next Generation System, 3GPP TSG SA Meeting #71, Gothenburg, Sweden, 9-11 March 2016.
- [6.4.7-6] Technical report 3GPP TR 23.799 V1.1.0 (2016-10) Study on Architecture for Next Generation System (Release 14).

7 Vertical extension of slicing

This section provides interface between orchestrator and virtual resources management entity on mobile fronthaul and backhaul. This section includes reference point of the interface, basic procedure of the virtual resource management, and functional requirements of the interface to control virtual resource from orchestrator.

In the description of the functional requirements, results of Gap analysis with transport SDN by ONF are reflected.

7.1 Network slicing for FH/BH

The basic concept of the Network Softwarization is "Slicing (LINP)" [Ref.7.1-1].

For the useful network slicing services, extensions for the slicing to the vertical axis and horizontal axis have to be considered. In Section 7, vertical extension is described. Main contents of this subsection are functional requirements of interface on reference points between the Orchestrator and FH/BH Control Functions, as the following order.

In the subsection 7.1.1, the conceptual implementation model is described, which includes the desirable reference point to be standardized. Addition to this, the detailed definitions are described to clarify the functionalities in the figure.

In the subsection 7.1.2, the information messages are described, which are exchanged at the reference points.

In the subsection 7.1.3, the basic sequence is described to exchange the information messages.

In the subsection 7.1.4, the lists of parameters are described, which are included in information messages exchanged at the reference points. These parameters are considered through the reviewing of the requirements for Gaps described in the Phase 1 report [Ref.7.1-1]. The detailed descriptions for the functions are also described.

The following subsections review the Gaps and required functions for the solutions to the Gaps

7.1.1 Slice and a conceptual implementation model

The slice is a logically isolated network partition [Ref.7.1-2]. The subsection 7.1.1 describes the conceptual implementation model with the functionalities for the network slicing.

For this purpose, the following four functionalities are required for the network slicing.

- 1) Create slice
- 2) Update slice
- 3) Read slice status
- 4) Delete slice

The diagram shown in Figure 7.1-1 is an assumed conceptual implementation model to provide the slice with the functionalities.

Figure 7.1-2 shows the relation between the assumed model and 5GMF model.

Figure 7.1-3 shows the relation between the assumed conceptual implementation model and the model of the Conceptual IMT-2020 non-radio network architecture [Ref.7.1-3]. In Figure 7.1-1, the shown functionalities are as follows.

“(Network & Service) Orchestrator”

(Network & Service) Orchestrator is an end-to-end slice controller including horizontally and vertically extended slice. Orchestrator controls and manages resources vertically via slice element controllers. It builds and operates each slice suitable for the service requirements.

Slice Element Controller (SEC)

SEC is installed for each control domain in vertically extended slice. Control domain is defined according to geographic location and control function. Each SEC converts requirements from orchestrator into virtual resources and manages virtual resources of slice elements. SEC also exchanges information of virtual resources with slice elements via virtual resource interface.

Physical Resource Controller (PRC)

PRC exchanges information of virtual resource with SEC via virtual resource interface and interconverts between virtual resource and physical resource. The PRC orders physical functions (ex. switches) to allocate physical resource via physical resource interface.

Network or Network Equipment(s)

A group of Network Equipment(s) and the control system(s).

Physical Function

Network Equipment(s) with a specific function

Slice

Slices are the logically isolated network partitions configured by the network elements ordered from the orchestrator.

Back haul(BH)

Back haul is defined as the network path connecting the base station site and the network controller or gateway site in [Ref.7.1-1]. Network path means transport excluding base station, network controller and gateway.

Front haul(FH)

Front haul is defined as the intra-base station transport, in which a part of the base station function is moved to the remote antenna site in [Ref.7.1-1]. Intra-base station transport is not including BBU and RRH.

For the scope of this section, main description targets are as follows,

- The definition of the functions of “Slice Element Controller (SEC)”
- The definition of the reference points (Interfaces) between “SEC” and “PRC” of FH/BH with the Command types, functions and etc. The definitions of others, namely excepting FH/BH, are not included.

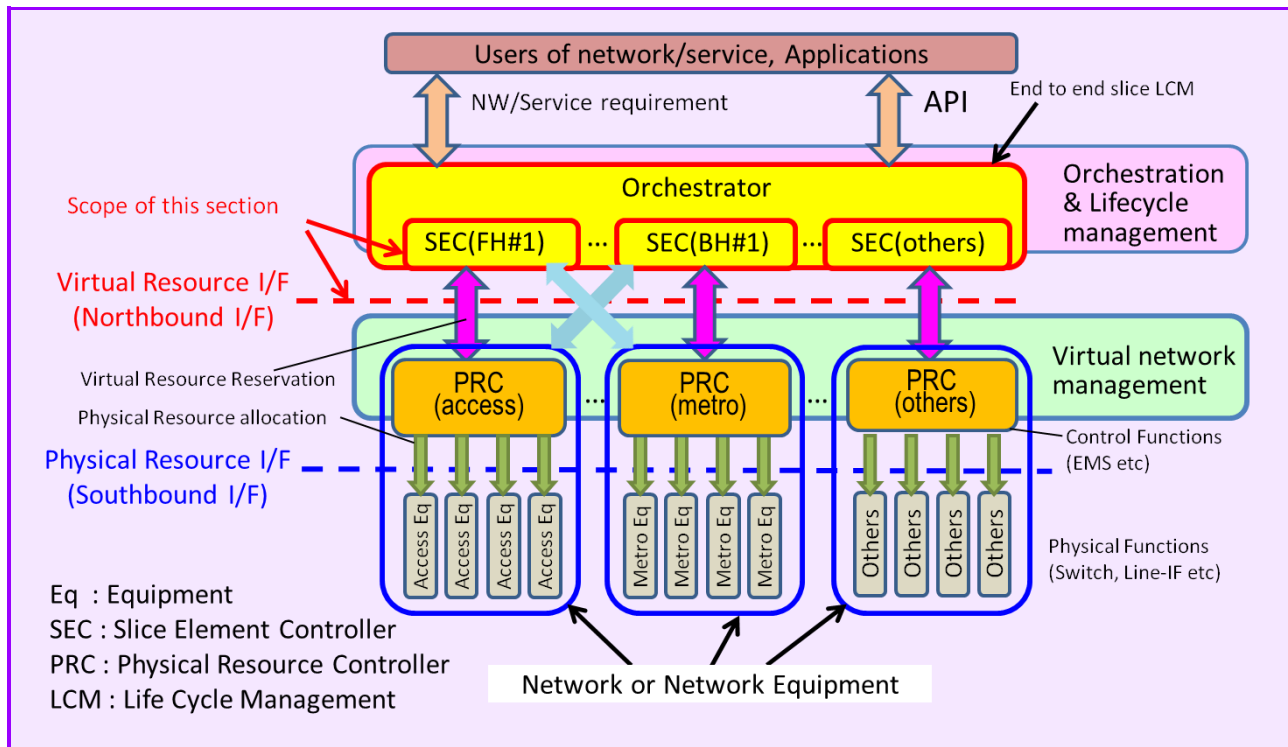


Figure 7.1-1 – A Conceptual Implementation Model for Network Softwarization and Sliced Network Service

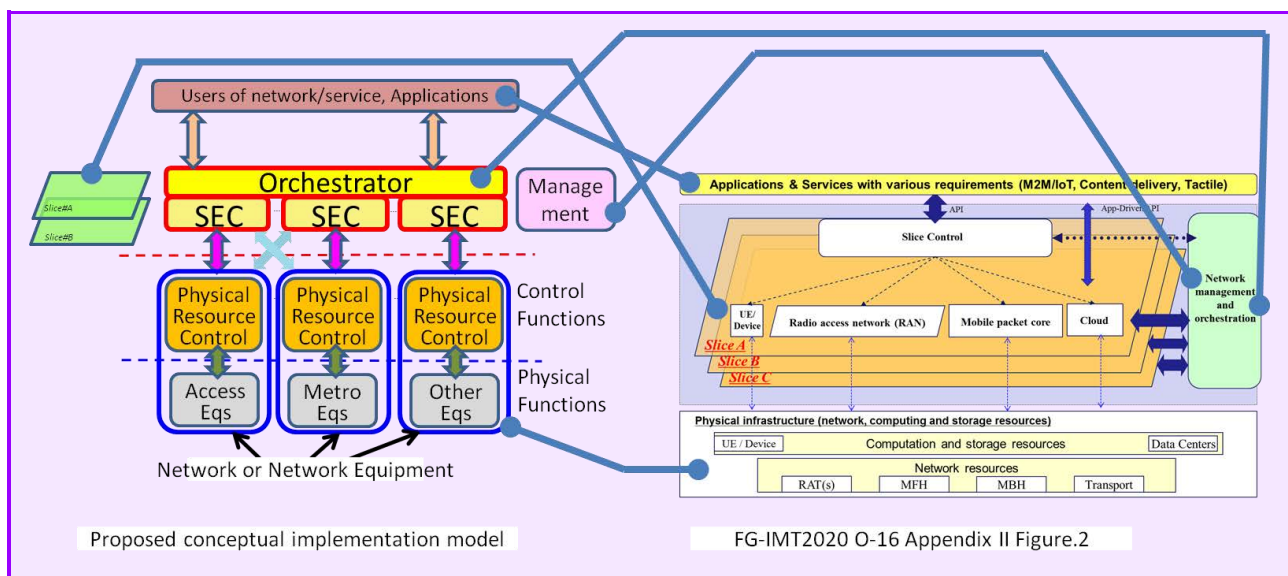


Figure 7.1-2 – Assumed Conceptual Implementation Model and 5GMF model

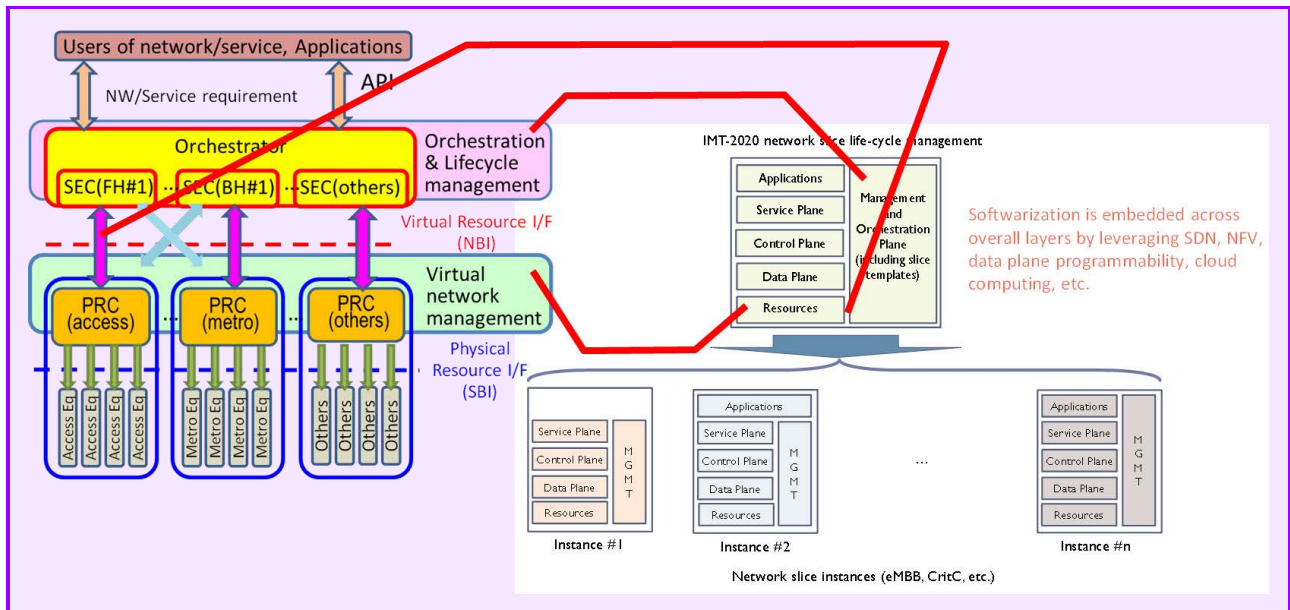


Figure 7.1-3 – Assumed Conceptual Implementation Model and the Model of Conceptual IMT-2020 non-radio network architecture [Ref.7.1-3]

7.1.2 The information messages exchanged at the reference point

7.1.2.1 Message categories at the reference point

In this subsection, the information messages exchanged at reference point which means the virtual resource interface between SEC and PRC are described especially for the FH/BH. The functionalities provided by the other regions will be described in the other subsections or documents.

Each SEC converts requirements from orchestrator into virtual resources and manages virtual resources of slice elements. SEC also exchanges information of virtual resources with slice elements via virtual resource interface.

At the virtual resource interface, the information messages controlling and managing virtual resources are exchanged. The information messages are as follows. The names of the example messages are tentatively given.

Information messages between SEC and PRC

a) Messages related with the “Create”

These messages are prepared to create the new virtual links, new virtual network services or new some other items for virtual resource level services.

One example of this message is “Virtual Resource Reservation” message. According to the requirement from orchestrator, SEC sends this message to PRC to reserve the virtual resource of PRC. For the response of this message, “Virtual Resource Reservation Complete” message is prepared. The PRC sends SEC this message to inform the result of the physical resource reservation.

b) Messages related with the “Update”

These messages are prepared to update (including change value of parameters or modify of characteristics) the described items for the virtual resource level services.

One example of this message is “Virtual Resource Modify” message. According to the requirement from orchestrator, SEC sends this message to PRC to change the value of reserved virtual resource of PRC like “Change the bandwidth from 1Gbit/sec to 2Gbit/sec”. For the response of this message, “Virtual Resource Modify Complete” message is prepared. The PRC sends SEC this message to inform the result of change value of the physical resources.

c) Messages related with the “Read”

These messages are prepared to read (including get the notification) the status, value of parameters or characteristics of virtual resources.

One example of this message is “Virtual Resource Status Request” message. To manage the utilization of virtual resource of PRC, SEC sends this message to PRC to request informing virtual resource status of PRC. For the response of this message, “Virtual Resource Status” message is prepared. The PRC sends SEC this message to inform the status of virtual resource.

d) Message related with the “Delete”

These messages are prepared to delete the created virtual resource level services which are described in “a)” of this subsection.

One example of this message is “Virtual Resource Release” message According to the request from orchestrator, SEC sends this message to PRC to release the virtual resource of PRC. For the response of this message, “Virtual Resource Release Complete” message is prepared. The PRC sends SEC this message to inform the result of virtual resource release message.

For the life cycle management of network slicing, a lot of scenarios may be considered. The following is one of them. When the allocated resource has to be changed the value (for example bandwidth), there are two scenarios as follows. The key point is “continuously existing or once deleted”.

Scenario 1: Once the present slice is deleted, the new one is set up with the new value.

Scenario 2: The slice continues to exist; only the value or parameters of resource are changed.

For the Scenario 1, the messages related with the “delete” and “create” will be used. For the Scenario 2, the messages related with the “update” will be used.

7.1.2.2 Expected additional functionalities and related document at the reference point.

In the above messages, only a few examples are introduced. The other related things for these four categories have been considered by the other SDOs as described in the transport SDN API document [Ref.7.1-4].

Related document (Transport SDN)

The APIs in the document and categories considered by the authors of this clause are shown in Appendix of this clause.

Additional Functionalities

They must be useful to consider the set of the messages on the proposed reference point. However, some of the functionalities should be added. They will be obtained from the gap analysis between the Phase 1 report and related documents [Ref.7.1-4].

(a) Functionalities for the low power operation

It seems that the transport SDN do not include the function to operate with the low power modes. The functionalities should be prepared that the Orchestrator can read the information of the capabilities for the low power modes [Ref.7.1-5] from the Physical Resource Controller if the modes are implemented. In this case, information which the Orchestrator can consider the trade-off to use the modes also should be included.

(b) Functionalities for updating the resource allocation using statistical traffic information

The transport SDN includes the function to allocate the resources with utilization ratio of the (allocated resource / physical link resources); however it seems that it does not include the functions to allocate/update the resources using statistical traffic information. For example, if these functions increase the allocated resource automatically when the traffic is increased, the functions can prevent the overflow by an unexpected traffic increase. This functionality must work well especially in the case when a set of the some virtual paths share a resource.

7.1.3 The basic procedure used for the exchange of information at the reference point

This subsection describes basic procedure used for the exchange of information at the reference point.

Figure 7.1-4 shows the procedure to exchange the “Virtual Resource Reservation (/ Complete)” message, “Virtual Resource Release (/ Complete)” message: and “Virtual Resource Status Request (/ Status)” message: as described in subsection 7.1.2.

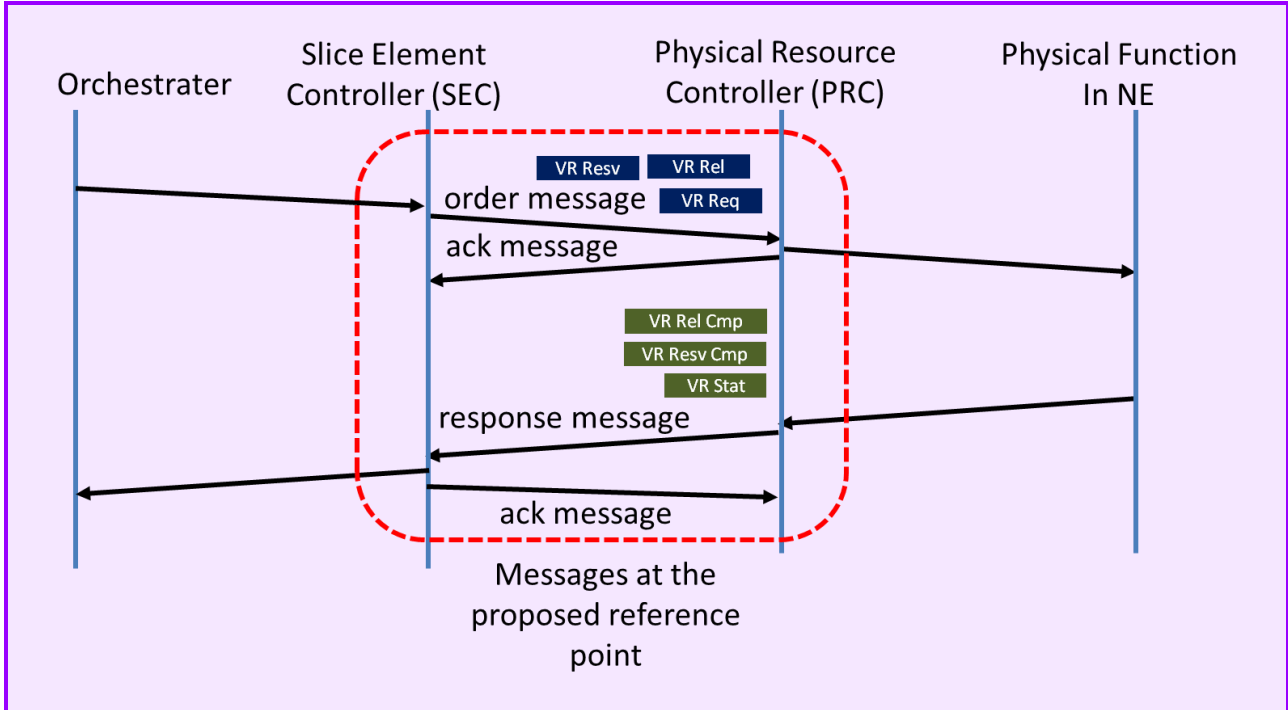


Figure 7.1-4 – Procedure for the virtual resource reservation / release / status message

7.1.4 The list of the parameters included in information messages

In this subsection, the candidates of parameters included in the information messages at the reference point are listed; the reference point is the point between the SEC and PRC as described in subsection 7.1.2. The list of parameters and purposes are shown in Table 7.1-1.

The parameters are also considered in the document [Ref.7.1-4].

Table 7.1-1 – The parameters for message at the reference point

Message	Parameters							
	Slice ID	Virtual Link ID	Section (Send node and Receive. node)	Signal format (SDH, 10GbE, etc.)	Bandwidth (G, M, K bit/sec)	Service Class (pseud wire, VBR, VBR+ABR, etc.)	Signal quality (BER, Packet Loss Ratio, etc.)	Delay and jitter (mili, u, nano, pico sec)
Virtual resource reservation message as the setup of the virtual link.	M	M	M	M	M	O	O	O
Virtual resource release message as the delete of the virtual link.	M	M						
Virtual resource status request message	M	M	M	O	O	O	O	O

M: mandatory, O: option

References

- [7.1-1] ITU-T SG13 TD 208, “Report on Standards Gap Analysis”,
<http://www.itu.int/en/ITU-T/focusgroups/imt-2020/Documents/T13-SG13-151130-TD-PLN-0208%21%21MSW-E.docx>.
- [7.1-2] ITU-T Y.3012, “Requirements of network virtualization for future networks”.
- [7.1-3] IMT-O-047, “Draft Recommendation: Network Management Framework for IMT-2020”, Section 7, Figure 4, the 8th meeting for ITU-T FG on IMT-2020, Geneva, 05-09, Dec., 2016.
- [7.1-4] ONF TR-527, “Functional Requirements for Transport API”, June 10, 2016.
- [7.1-5] 5GMF White Paper ver. 1.01, “5G Mobile Communications Systems for 2020 and beyond”, Section 12.4.1.3: 3) Efficiency of fronthaul and Section 12.4.2.2: Dynamic control of NW resources and path optimization, Jul., 4, 2016.

Appendix

Summary of the functional requirement for the transport SDN by ONF and categories considered by the FG IMT-2020 Network Softwarization Working Group

From the ONF (Open Networking Foundation), “Functional Requirements for Transport API (ONF TR-527)” [Ref.7.1-4] has been released. Table 7.1-A-1 shows the relation between the functionalities or data type of their API and categories defined and mapped by the authors of this document.

Table 7.1-A-1 – Functionalities for transport SDN by ONF and their categories (Part 1)

	Categories considered by the authors			
	Create	Update	Read	Delete
Topology Retrieval APIs				
TAPI_FR 1: Get Topology List			X	
TAPI_FR 2: Get Topology Details			X	
TAPI_FR 3: Get Node Details			X	
TAPI_FR 4: Get Link Details			X	
TAPI_FR 5: Get Node Edge Point Details			X	
Connectivity Service				
TAPI_FR 6: Get Service End Point List			X	
TAPI_FR 7: Get Service End Point Details			X	
TAPI_FR 8: Get Connectivity Service List			X	
TAPI_FR 9: Get Connectivity Service Details			X	
TAPI_FR 10: Get Connection Details			X	
TAPI_FR 11: Get Connection End Point Details			X	
Connectivity Request APIs				
TAPI_FR 12: Create Connectivity Service	X			
TAPI_FR 13: Update Connectivity Service		X		
TAPI_FR 14: Delete Connectivity Service				X
Path Computation Request APIs				
TAPI_FR 15: Compute P2P Path				
TAPI_FR 16: Optimize P2P Path				
Virtual Network Retrieval APIs				
TAPI_FR 17: Get Virtual Network Service List			X	
TAPI_FR 18: Get Virtual Network Service Details			X	
Virtual Network Request APIs				
TAPI_FR 19: Create Virtual Network Service	X			
TAPI_FR 20: Delete Virtual Network Service				X

Table 7.1-A-2 – Functionalities for transport SDN by ONF and their categories (Part 2)

	Categories considered by the authors			
Categories considered by the authors	Create	Update	Read	Delete
Notification Subscription and Filtering APIs				
TAPI_FR 21: Discover Supported Notification Types			X	
TAPI_FR 22: Create Notification Subscription			X	
TAPI_FR 23: Modify Notification Subscription			X	
TAPI_FR 24: Delete Notification Subscription			X	
TAPI_FR 25: Suspend Notification Subscription			X	
TAPI_FR 26: Resume Notification Subscription			X	
TAPI_FR 27: Retrieve Notification Records			X	
Notification Message Types				
TAPI_FR 28: Object Creation Notification			X	
TAPI_FR 29: Object Deletion Notification			X	
TAPI_FR 30: Attribute Value Change Notification			X	
TAPI_FR 31: State Change Notification			X	

Table 7.1-A-3 – Functionalities for transport SDN by ONF and their categories (Part 3)

	Categories considered by the authors			
	Create	Update	Read	Delete
TAPI Data Types				
TAPI_FR 32: Layer Protocol Name			X	
TAPI_FR 33: Capacity (Fixed Bandwidth)			X	
TAPI_FR 34: Capacity (Profile)			X	
TAPI_FR 35: Administration State			X	
TAPI_FR 36: Operational State			X	
TAPI_FR 37: Lifecycle State			X	
TAPI_FR 38: Port Role			X	
TAPI_FR 39: Port Direction			X	
TAPI_FR 40: Termination Direction			X	
TAPI_FR 41: Service End Point TRI format			X	
TAPI_FR 42: Service Type			X	
TAPI_FR 43: Connectivity Constraints			X	
TAPI_FR 44: Virtual Network Service Constraints			X	
TAPI_FR 45: Traffic Matrix			X	
TAPI_FR 46: Path Optimization Constraint			X	
TAPI_FR 47: Path Objective Function			X	
TAPI_FR 48: Notification-Header			X	
TAPI_FR 49: Notification-Type			X	
TAPI_FR 50: Object-Type			X	
TAPI_FR 51: Notification-Source-Indicator			X	

7.2 The review of the gaps described in Phase 1 Report and usecases

7.2.1 Review of the gaps described in Phase 1 Report and required functionalities on the Virtual Resource interface.

In the FG-IMT-2020 Phase 1 report [Ref.7.1-1], 21 gaps are pointed out. In these gaps, some of them are discussed in SG15 or other SDOs. In the Network Softwarization working group in the FG-IMT-2020, 11 gaps should be discussed (That is agreed in the Seoul meeting on March 2016).

In this subsection, the relation between the gaps described in the Phase 1 report [Ref.7.1-1] and functionalities controlled by the information(s) are reviewed. Then, it is clarified whether the gaps can be resolved by the set of the described functionalities or not. The detailed descriptions are described as follows.

7.2.1.1 Large capacity transmission (D.7.1-1)

In the IMT2020 network, large capacity virtual link with the some order made functionalities are required. Therefore, to establish the large capacity transmission, the information(s) at the reference points should be able to include the section of the virtual link, bandwidth, signal format and some optional indexes. Section 7.1 with Table 7.1-1 describes to include these functions.

7.2.1.2 Power saving by sleep or rate control (D.7.4)

For power saving by sleep or rate control, power consumption mode should be ordered from Network & Service Orchestrator to Control Functions. The following information(s) can be used for power saving such as service class, bandwidth and signal quality. Addition to them, the mechanism can be used for the more useful or detailed power saving which is described in the previous subsection. These information(s) will be ordered from network & Service Orchestrator to Physical Resource Controller Functions as described in Section 7.1 with Table 7.1-1.

7.2.1.3 PON as a virtual digital wireline service (D.7.5-1)

When PON would like to be used for the system to provide the virtual digital wireline (VDW) service, the section for VDW, bandwidth, and signal format should be mandatory ordered. The reliability or the other options are also used.

To establish slice for virtual digital wireline service using PON, the parameter of service classes such as pseud wire, VBR, ABR are required to share the virtual resource with other transport link. These information(s) will be ordered from network & Service Orchestrator to Physical Resource Controller Functions as described in Section 7.1 with Table 7.1-1.

7.2.1.4 Reliability and resiliency (D.7.6)

To provide the slices with reliability or resiliency, the reliability mode should be ordered from network & Service Orchestrator to Control Functions. In the document [Ref.7.1-4] describes the functions that can be associated to an connection, such as protection switching is referenced as secondary entities through the associated LTP (Logical Termination Point) instance in Section 4.6 of [Ref.7.1-4].

7.2.1.5 Diversified types of terminals (D.7.7-1)

In the IMT-2020 network, deferent slices will be able to be prepared for the different types of terminals. Each slice will have the suitable virtual link with the suitable sections, bandwidth and signal format. These information(s) will be ordered from Orchestrator to Physical Resource Controller as described in Section 7.1 with Table 7.1-1.

7.2.1.6 Diversified types of traffic (D.7.7-2)

In the IMT-2020 network, deferent slices will be able to be prepared for the different types of traffic. Each slice will have the suitable virtual link with the suitable sections, bandwidth and signal format. These information(s) will be ordered from Orchestrator to Physical Resource Controller as described in Section 7.1 with Table 7.1-1.

7.2.1.7 Diversified types of network operator (D.7.7-2)

In the IMT-2020 network, deferent slices will be prepared for the different types of network operators. Each slice will have the suitable virtual link with the suitable sections, bandwidth and signal format according to the operator's strategy. These information(s) will be ordered from Orchestrator to Physical Resource Controller as described in Section 7.1 with Table 7.1-1.

7.2.1.8 Diversified types of RAN (D.7.7-4)

In the IMT-2020 network, deferent slices will be able to be prepared for the different types of RAN. Each slice will have the suitable virtual link with the suitable sections, bandwidth and signal format according to the RAN. These information(s) will be ordered from Orchestrator to Physical Resource Controller as described in Section 7.1 with Table 7.1-1.

7.2.1.9 Radio over packet (D.8.5)

In the IMT-2020 network, both of the present and new radio over packet signals will be used simultaneously. The virtual link suitable for the each radio over packet signals should be prepared. For this purpose, the information(s) for the virtual link will be ordered from Orchestrator to Physical Resource Controller. These information(s) will include the section, bandwidth and signal format for the virtual link as described in Section 7.1 with Table 7.1-1.

7.2.1.10 Coordination of power saving across MFH/MBH/Radio System (D.9.2.1)

In the phase 1 report O-016, "Coordination of power saving across MFH/MBH/Radio System" is described. In the system, flow routes or radio systems (IMT-2020, WiFi, or stations) to the UE are selected to minimize the power consumption. For this purpose, the sections for virtual links, bandwidth, and the signal format for the FH/BH will be ordered from Orchestrator to Physical Resource Controller as described in Section 7.1 with Table 7.1-1.

7.2.1.11 Power saving by resource optimization (D.9.2.2)

For power saving by resource optimization, power consumption mode should be ordered from Network & Service Orchestrator to Control Functions. The following information(s) can be used for power saving such as service class, bandwidth and signal quality. Addition to them, the mechanism can be used for the more useful or detailed power saving which is described in the previous subsection. These information(s) will be ordered from network & Service Orchestrator to Physical Resource Controller Functions as described in Section 7.1 with Table 7.1-1.

7.2.2 The review the relation between the Use Cases and functionalities controlled by the information(s)

In this subsection, it is clarified whether the functions listed in Section 7.2.1 are enough or not to realize three typical use cases of FH/BH slicing showing below.

Use Case 1: Dynamic Cell Allocation

The first example adoption of the network slicing is the dynamic resource allocation for small cells on the FH. Figure 7.2-1 shows an example of this use case. In this case, when there are some policy groups, resources like link bandwidth to the small cells can be allocated dynamically to satisfy the expected group service policies. The overview and benefits are shown in Table 7.2-1.

In Figure 7.2-1, a dedicated slice, “Slice 4” is prepared for the “high mobility Service”. For this slice, the limited numbers of dedicated paths are allocated between Base Station (BS) 4 and the Remote Heads (RHs). The end points of RHs are changed according to the locations of User Equipment’s (UE). For this mechanism, the listed functionalities are required as shown in the column of the “challenges to be clarified” on Table 7.2-1

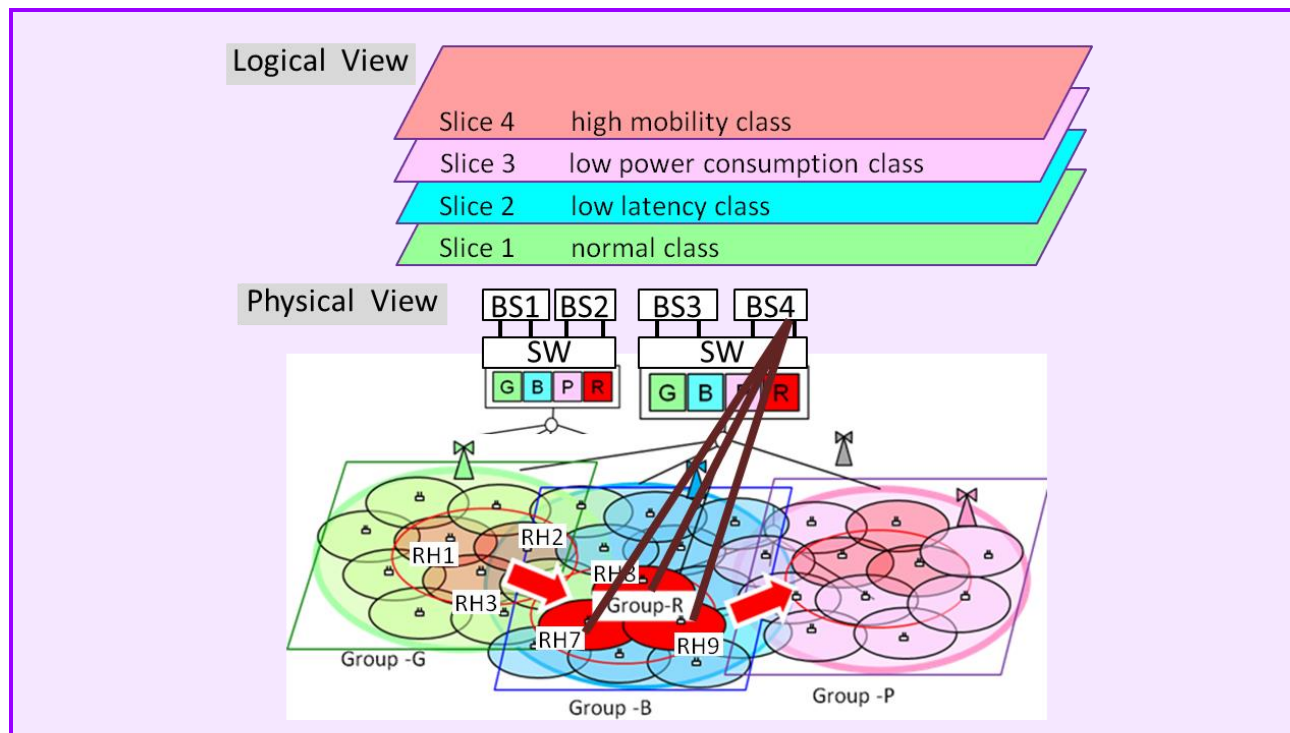


Figure 7.2-1 – Dynamic Resource Allocation for Small Cells

Table 7.2-1 – Dynamic Resource Allocation for Small Cells

Items	Descriptions
Overview	When there are some policy groups, resource like link bandwidth or the small cells are allocated dynamically to satisfy the expected group service policies.
Benefits	<ul style="list-style-type: none"> – Optimization of network resource allocation or configuration for the variety of services – Reduction of power consumption
Challenges to be clarified	<ul style="list-style-type: none"> – Power saving by sleep or rate control (D.7.4) – PON as the virtual digital wireline service (D.7.5-1) – Diversified types of terminals, traffic, network operator, and RAN (D.7.7-1 to 4) – Coordination of power saving across FH/BH/Radio System (D.9.2.1) – Power saving by resource optimization (D.9.2.2)

As the all challenges shown in Table 7.2-1 are covered in Table 7.1-1, it can be said that the functions listed in 7.2.1 are mostly enough to realize this use case.

Use Case 2: Elastic OADM Ring

The second example adoption of the network slicing on the FH/BH is the elastic OADM ring. Figure 7.2-2 shows an example of this use case. In this case, elastic OADM can provide the bandwidth per wavelength to adjust the capacity of mobile backhaul. The wavelength with suitable bandwidth is given to each slice. The overview, benefits and challenges to be clarified are shown in Table 7.2-2.

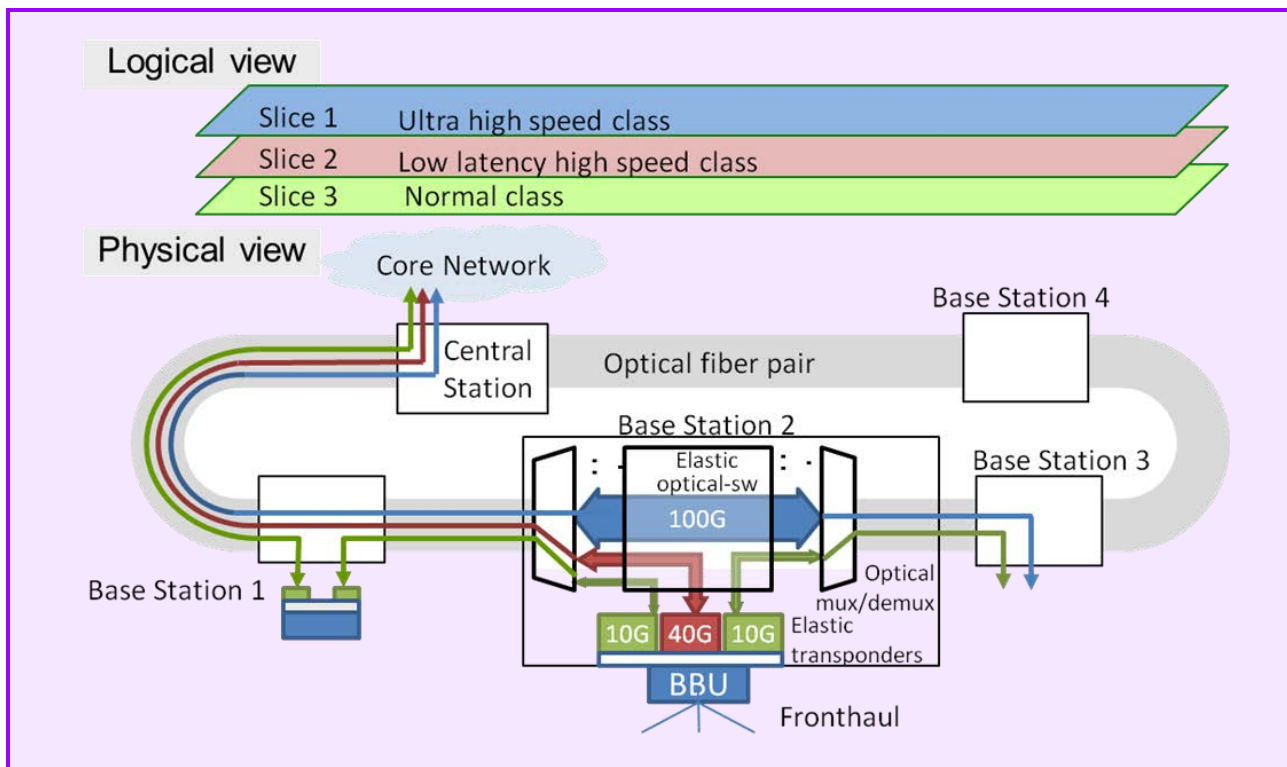


Figure 7.2-2 – Elastic OADM Ring

Table 7.2-2 – Elastic OADM Ring

Items	Descriptions
Overview	In this case, elastic OADM can provide the bandwidth per wavelength to adjust the capacity of mobile backhaul. The wavelength with suitable bandwidth is given to each slice.
Benefits	– The optical bandwidth even over 40 Gbps can be flexibly allocated to each slice on demand.
Challenges to be clarified	– Large capacity transmission (D.7.1) – Power saving by sleep or rate control (D.7.4) – Reliability and resiliency (D.7.6)

As the all challenges shown in Table 7.2-2 are covered in Table 7.1-1, it can be said that the functions listed in 7.1 are mostly enough to realize this use case.

Use Case 3: Virtual BBUs

The third example adoption of the network slicing on the FH/BH is the virtual BBU. Figure 7.2-3 shows an example of this use case. In this case, various generations BBUs and various operators BBUs can be implemented by software on the same hardware. The overview, benefits and challenges to be clarified are shown in Table 7.2-3.

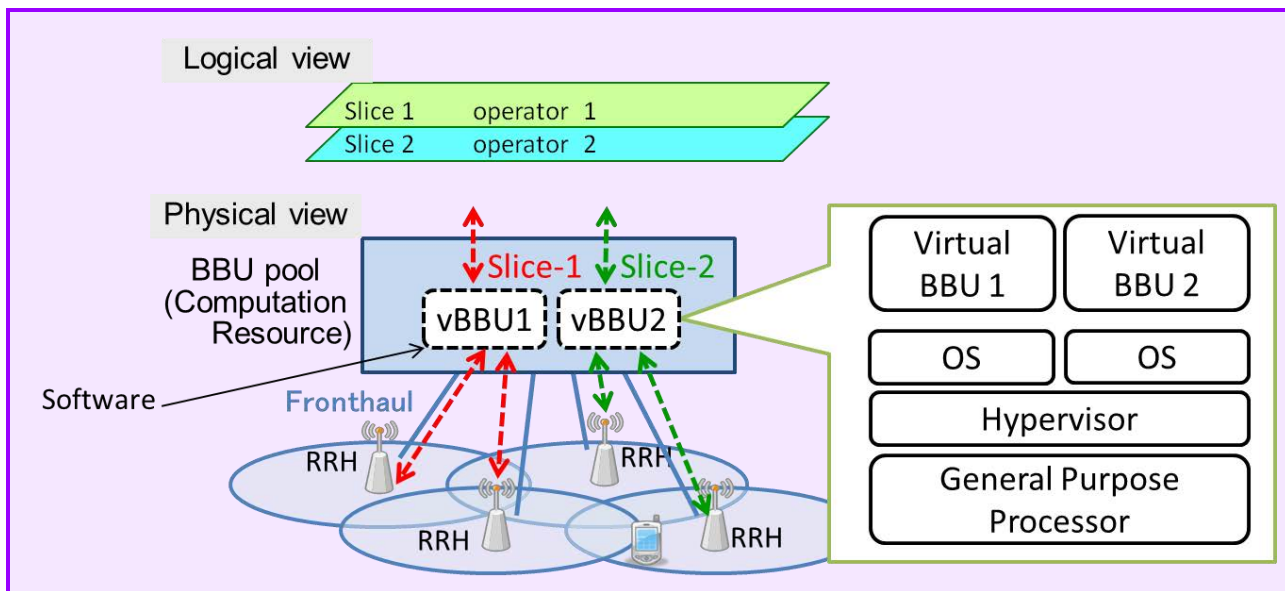


Figure 7.2-3 – Virtual BBUs

Table 7.2-3 – Virtual BBUs

Items	Descriptions
Overview	Various generations BBUs and various operators BBUs can be implemented by software on the same hardware.
Benefits	<ul style="list-style-type: none"> – Computation resources can be flexibly allocated to each virtual BBU on demand. – The BBU function can be dynamically reconfigured and adjusted.
Challenges to be clarified	<ul style="list-style-type: none"> – Diversified types of terminals, traffic, network operator, and RAN (D.7.7-1 to 4) – Coordination of power saving across FH/BH/Radio System (D.9.2.1) – Power saving by resource optimization (D.9.2.2)

While the challenges shown in Table 7.2-3 are partially covered in Table 7.1-1, it needs management/control for the computation resources in addition. This is for further study.

7.3 Data Plane Programmability

7.3.1 Background [Ref.7.3-1]

Although the synergy between SDN and NFV has only recently been discussed, they have been proposed separately. While SDN primarily focuses on the programmability on the control of networking, NFV aims at implementing data processing functions in software on top of virtual machines (VMs) that exist today as hardware network appliances. The clear distinction of the focuses of SDN taking care of networking and NFV of computation may allow scalable construction of programmable infrastructure, since data packets can be programmatically redirected by SDN and can be programmatically processed by NFV.

However, we observe two limitations in this model of separation of SDN and NFV, leaving an interesting research area as a gap between them. First, SDN often defines predetermined interface, so called southbound interface or SBI, mainly for the sake of standardization purpose. It is control plane software including controllers that can be programmed in software above SBI (and not to mention, above so called north-bound interface or NBI), but data plane that implements data forwarding and redirection often remains to be implemented in hardware as in, e.g., OpenFlow switches. If we could arbitrarily define data plane by software, i.e., software-defined data plane, in carefully designed sandboxes such as virtual machines inside network equipment, we should be able to enhance the data plane functionalities, e.g., those related to OAM, and publish the SBI for controllers to use them. Second, NFV is so far limited to implementing network appliances in software, and deals neither with crafting new protocols nor with OAM functionalities,

which are largely considered as SDN's responsibility. However, as mentioned above, SDN's data plane is not so much flexibly programmable.

We posit that data plane programmability may bring more innovations in future networking, in network softwarization, especially in the SDN and NFV areas applied to 5G mobile networking. We expect at least three benefits enabled by deeply programmable data plane, (1) enhanced interaction between applications and networks, (2) enhanced flexibility and optimization of network functions, and (3) rapid development of new network protocols such as ICN.

7.3.2 Challenges in Data Plane Programmability [Ref.7.3-2]

To enable flexibly and deeply programmable, we post there are three major technical challenges, (1) ease of programming, (2) reasonable and predictable performance and (3) isolation among multiple concurrent logics

First, we must consider lowering the barrier to entry for programming network functions. Network softwarization is considered as cost-effective solutions, but the premise is that we need lots of programmers for creating network functions, let alone data plane functionalities. Therefore, one of the most important challenges to resolve is how we can accommodate programmers of various levels of skills and thus increase the entire number of programmers. There could be lots of kinds of programming models for defining programmable data plane, such as FPGA, Intel Data Plane Development Kit (DPDK), Network Processors with many cores, but we need to carefully select these platforms in terms of ease of programming and debugging.

Second, performance is another challenge in programmable networking. It is often the case with programmable network equipment that there is trade-off between the programmability, i.e., how simply and flexibly we can program and the performance, i.e., how fast we can execute programs. Especially, software solutions are mostly susceptible to performance degradation, although it is highly flexible and can be quickly designed and implemented. In the light of this observation, we believe that we should select the platform with high flexibly but reasonable and at least predictable performance.

And finally, we believe the capability of programming multiple concurrent logics on top of a single physical programmable environment is significant. We can virtualize the physical hardware resources and provision necessary amount of virtual resources per logic to achieve programming of multiple logics on top of isolated virtual resources. Isolation of resources plays a very important role.

7.3.3 Introduction to FLARE [Ref.7.3-1, 7.3-2 and 7.3-5]

In FLARE [Ref.7.3-1] architecture, we attempt to resolve all three major technical challenges enumerated in Clause 7.3.2, namely (1) ease of programming, (2) reasonable and predictable performance and (3) isolation among multiple concurrent logics, in realizing software-defined data plane programmability. We introduce Toy-Block networking programming model to enable drag and drop programming in FLARE to resolve (1). Also, in order to achieve (2), we combine a hybrid of computation resources especially design a hierarchical structure of high-frequency small-number- core processors and low-frequency many-core processors. And finally, for (3), we employ a lightweight resource virtualization technique called resource container for isolation of multiple logics. For the best isolation, we decide to partition many cores into groups and deploy a resource container per group [Ref.7.3-2].

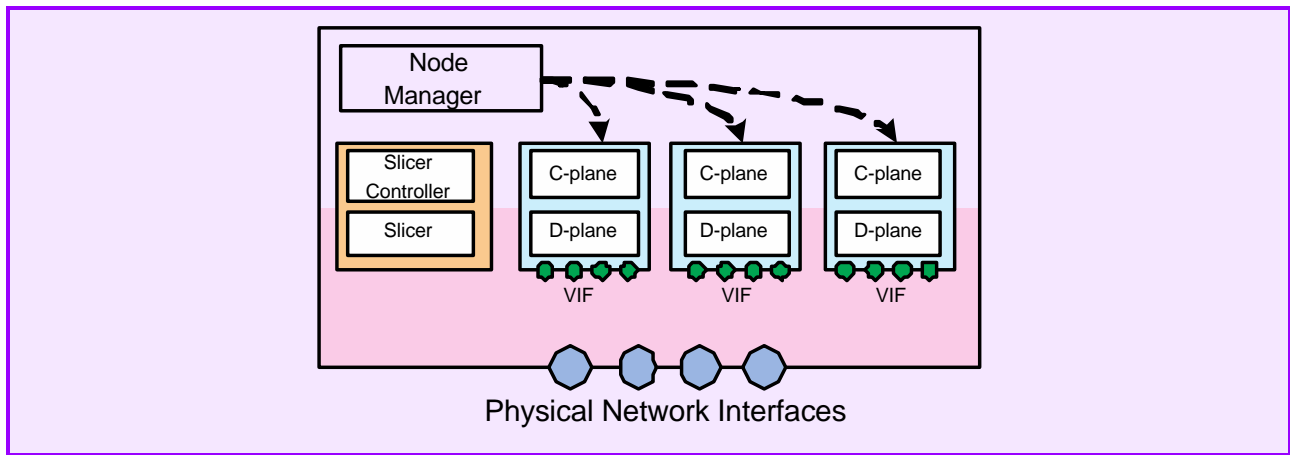


Figure 7.3-1 – Underlying infrastructure of FLARE [Ref.7.3-5].

FLARE is an open deeply programmable node architecture that can run multiple isolated virtual network functions on a physical node simultaneously. As shown in Figure 7.3-1, a FLARE node consists of a hybrid of many-core network processor and x86 processor. With the technology of virtualization, we sliced both network resources (CPUs, memory and link bandwidth) in both many-core processors and x86 processors environment into isolated slivers. For each sliver, control plane runs on x86 processors while data plane runs on many-core processors. Control plane and data plane communicate via Ethernet-over-PCI interface [Ref.7.3-5].

All incoming packets will be scanned and classified by Slicer and then diverted to slivers. Although not shown in the figure, there is a central node called FLARE central that talks to Node Manager to manage the resources of each FLARE nodes. The control module called Node Manager is in charge of adding/removing slivers from a FLARE node. Users can also configure and program their slivers via the interface provided by FLARE central [Ref.7.3-5].

FLARE's architecture helps to fill in the gap between SDN and NFV, because it allows to implement software programs of the both types on a same node while keeping individual logics isolated. The many-core architecture enables this implementation without sacrificing performance. One such example (using 'Lagopus' software switch) will be discussed in Section 7.3.5 and 7.3.6.

7.3.4 Use cases of FLARE architecture

Use case 1: Application-Specific Mobile Edge Computing [Ref.7.3-3 and 7.3-4]

Mobile Edge Computing (MEC) has been recognized as one of the key emerging technologies in the evolution towards 5G cellular wireless networks. In this paper, we present an application-specific MEC architecture that applies the concept of data plane programmability to Mobile Virtual Network Operators (MVNOs) network.

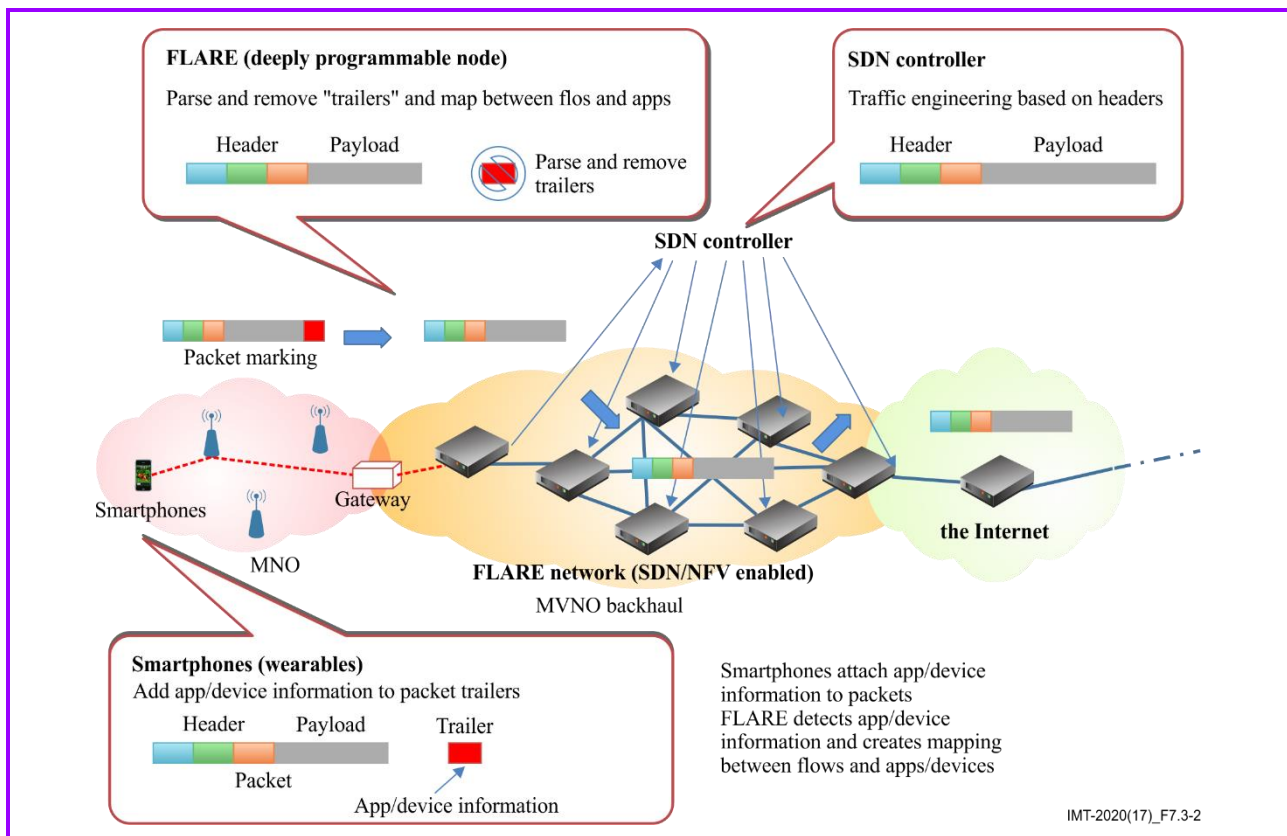


Figure 7.3-2 – Network Architecture of MVNO in the University of Tokyo [Ref.7.3-3]

We set up an MVNO network connecting customized smartphones so that we can identify applications and devices from the given traffic with 100% accuracy using FLARE switch. As shown in Figure 7.3-2, we have designed trailer slicing, where meta information on applications and devices at the end of packets. We install our software on smartphones to capture the very first packets for an application (e.g., TCP SYN packets) and examine the process table and the socket table of the operating system to look for a corresponding application process name that uses the flow space and attach the information as a trailer.

After adjusting of header fields in Layer-3 and Layer-4, we can get packets with trailers through existing network appliances since trailers are treated as Layer-7 data bits. Comparing with storing data identifier in header option fields, trailer identifier can pass all the network appliances while non-standard header option may be removed during transmission. Also, we can attach device information as well as that of application. Note that in our design, the meta-information may include many other kinds of information.

Utilizing FLARE prototypes, we have implemented our prototype system to enable application and/or device specific slicing for MVNO as shown in the overview of our design depicted in Figure 7.3-2. We have developed Android smartphone software to enable trailer slicing, i.e., embedding a slice identifier at the trailer of TCP SYN packets and QoS traffic engineering per slice on our FLARE platform. We have discovered that we can use TCP SYN trailers unless ISPs do not filter unusual TCP SYN in fear of SYN Flooding, which is not performed in most MVNO services of today.

Use case 2: Softwarized LTE in FLARE Network Slices [Ref.7.3-5 and 7.3-6]

We softwarize both eNodeB (eNB) and Evolved Packet Core (EPC) using modified OpenAirInterface (OAI) and implement LTE network in a slice on top of a FLARE node. OpenAirInterface (OAI) is an open experimentation and prototyping platform created by EURECOM. It provides a software implementation of all elements of the 4G LTE/5G architecture including use equipment (UE), eNodeB (eNB), Home Subscriber Server (HSS) and Evolved Packet Core (EPC) components. A compound EPC component composes of Serving Gateway (S-GW), Packet Data Network Gateway (P-GW) as well as the Mobility Management Entity (MME). The eNB and EPC components are responsible for creating channels (namely bearers) with UE and forwarding the user traffic.

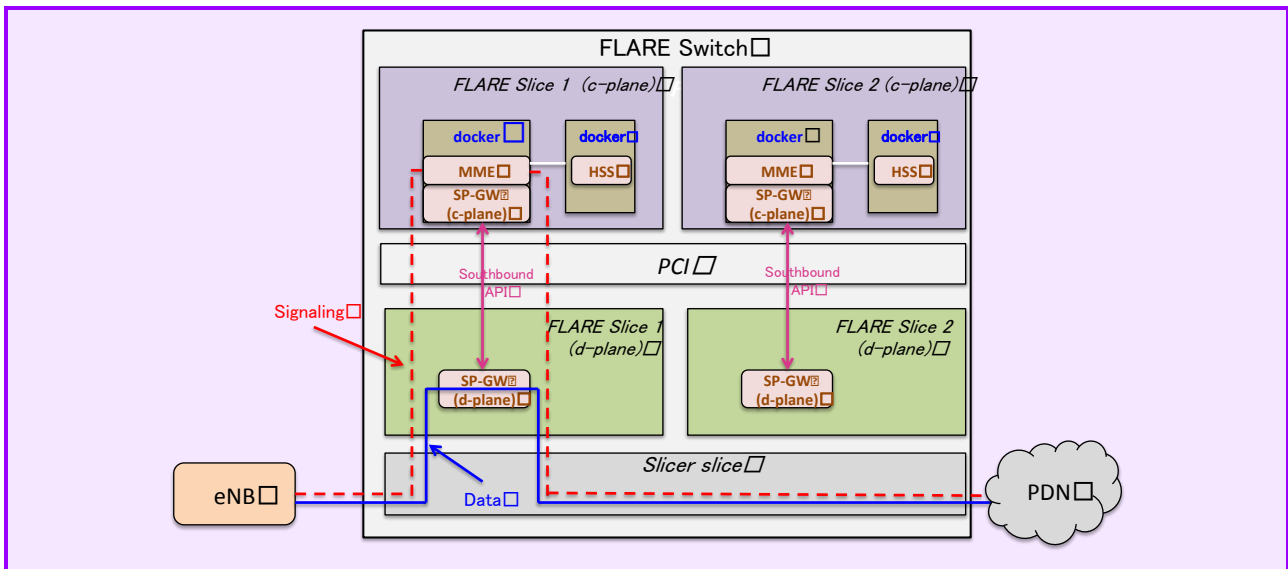


Figure 7.3-3 – Architecture of Softwarized LTE in FLARE Network Slices

Performance is an important issue in programmable 5G networks, which is highly dependent on the underlying hardware infrastructure. Hardware product EPC can achieve high performance but it lacks of flexibility once the logic has been programmed.

In this section, we introduce how to implement an EPC slice in a FLARE slice shown in Figure 7.3-3, where signalling related EPC entities (e.g., MME) will be implemented in a control plane while user data forwarding and processing (e.g., SGW and PGW) will be implemented in a data plane. One benefit of our approach is to reduce the user data processing delay at EPC as well as increasing computing and processing capability via many-core processors.

- 1) **Data-plane:** We offload the GTP-U channel creation and user data processing from control plane to data plane, which is implemented with GTPV1-U kernel module in naïve OAI software. One challenge of EPC implementation is in offering such extensibility while at the same time achieving a good performance. To scale network function, one promising approach is to divide functionality and parallelize packet processing across on-chip multiple processors. Flare enables rapid deployment of new network functions by providing the Click network-programming framework. We abstract the underlying architecture such as I/O engine, inter-core communication and only expose the relevant necessary details to a set of predefined Click elements. We implement SP-GW data-plane with chained Click elements. When a FLARE switch receives packets from eNB, its Slicer slice will classify packets to different slices as well as classifying signalling packets (e.g., GTP-C) from data packets (e.g., GTP-U). The signalling packets will be forwarded to control-plane while the data packets will be processed in data plane with many-core processor.
- 2) **Control-plane:** We run the signalling entities of EPC slice (e.g., MME and the control-plane of SP-GW) in a Docker instance. We can also run HSS entity in another Docker instance within the same FLARE slice. These two Docker instances are isolated and replaceable without interfering others. For example, we can install different version of packages in MME and HSS instances while they may conflict when installed on the same host machine. The interfaces between EPC and HSS entities are implemented with internal Ethernet links. They can communicate with each other via TCP and SCTP protocols.
- 3) **Southbound API:** We need to define the Southbound API between data-plane and control-plane so that an GTP-U tunnel from data-plane to eNB could be established when parsing and processing GTP-C packets in the control-plane. When receiving GTP-C packets, MME will ask SP-GW to establish, update and maintain the GTP-U tunnels in the data plane. It is also responsible for transferring GTP tunnelling parameters including endpoint identifier with the Tunnel End point Identifier (TEID) to eNBs.

In implementing our prototype, we adopt OpenFlow's pattern-match-action convention for programming abstraction and define one's own programming abstraction as API as following

<UEID, TEID><Action><Stat>,

where UEID could be a UE's IP address assigned by MME through signalling channel, Action refers create/update/remove a GTP-U tunnel.

LTE network slice instances are isolated without interfering one another. We demonstrate play back of YouTube movies on a smart phone (Nexus5) connected to an OAI slice on FLARE [Ref. 7.3-6].

7.3.5 Functional Enhancement in Data Plane Using SDN Software Switches

SDN aims primarily at flexible networking enabled by software control. For example, OpenFlow, the most widely used SDN technology, specifies OpenFlow Switch Specifications as SBI, by which an SDN controller can impose packet forwarding rules onto OpenFlow-compliant switches. Unlike conventional Layer-2/3 switches, the rules are not limited to predetermined, proprietary set of packet forwarding criteria, but are programmed using openly defined interfaces. In this sense, programmability of SDN resides largely in control layer.

However, there have been technological developments in which data plane is made programmable while retaining the SDN framework. (That is, the application-control-data structure and NBI and SBI interfaces.) An example is Protocol Oblivious Forwarding (POF), which will be described in detail in the next section (7.4). It should also be mentioned that the SDN architecture discussed at ITU-T SG13 encompasses not only data forwarding functionalities but also data processing functionalities in data plane.

Another approach to realize data plane programmability in SDN is to enhance data plane functionalities by using it in combination with NFV, or more broadly speaking, computational capabilities. As stated in Section 7.3.1, synergy between SDN and NFV has been discussed widely. This is because both the technologies make use of abstraction of hardware and/or its capabilities and are thus in a complementary relationship with each other, enabling their combined use to realize flexible and sophisticated control of packets by software. In fact, FLARE, described above, can incorporate this idea into its architecture.

Software-based SDN switches fit well in this approach. There are, however, issues in doing so, most notable one being performance. As stated in Section 7.3.2, to keep reasonable and predictable performance becomes the key when considering using a software switch that runs on general-purpose CPUs.

7.3.6 Introduction to Lagopus [Ref.7.3-7 and 7.3-8]

There are several SDN software switch products, both commercial and open-sourced. Open-source software switches are especially useful when trying to explore cutting-edge network softwarization moves. At the same time, they are paving way to commercial usage as their functionalities, performance, and reliability continue to improve.

'Lagopus' is an open-source software switch that runs on x86 CPUs and are fully compliant with OpenFlow Switch Specifications. Its development started under O3 Project, aiming at a switch with high performance, functional extensibility, and usability for wide area network uses including telecom carrier networks. (See Section 6.2.1 for more information about O3 Project.) It features supporting multiple WAN networking protocols, management protocols/interfaces, and large-scale flow entries to name a few [Ref.7.3-7].

Regarding performance, Lagopus has a number of characteristics in its software architecture and design. The switch's software is divided into two main components: switch agent and data plane. The switch agent component has a unified data store functionality to configure and manage switch resources and provides interfaces to OpenFlow controllers. The data plane component is responsible for all the processes that packet forwarding involves. It utilizes Intel DPDK libraries to accelerate network I/O performance, which enables to bypass packet processing in Linux OS kernel and to access directly to NIC packet buffers from userspace programs. It also exploits multiple CPU cores to achieve fast, efficient handling of packet flows, using parallel processing technique. Figure 7.3-4 shows the parallel processing architecture from ingress to flow lookup and header modification to egress. By assigning specific CPUs dedicatedly to the I/O receive (RX) and the I/O transmit (TX) threads, overheads in these threads can be well reduced. In addition, flow lookup is accelerated by employing fast flow-table look schemes as well as CPU caches, leading to an overall improvement of packet forwarding performance [Ref.7.3-8].

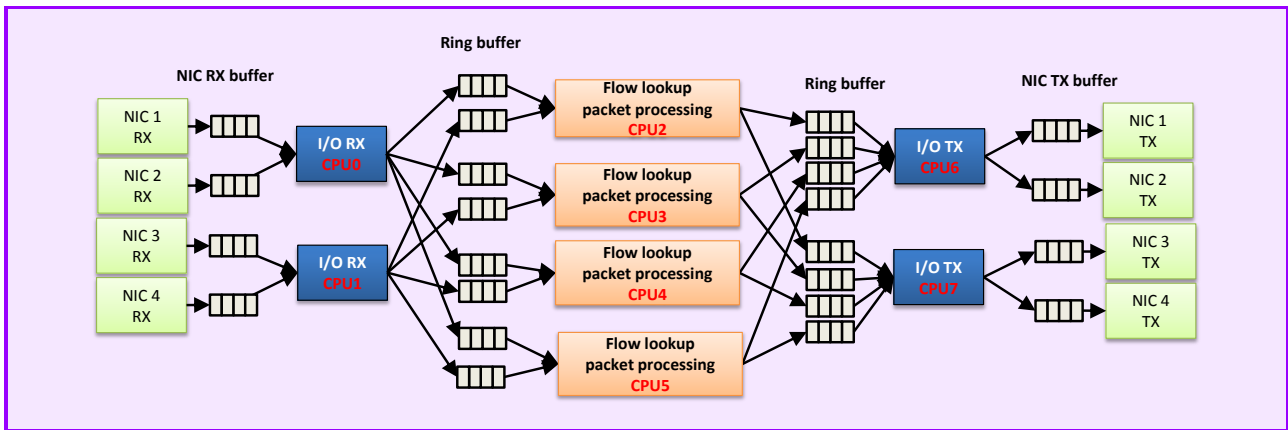


Figure 7.3-4 – Parallel processing of packet flows in Lagopus data plane

The design of both the switch agent and the data plane components is module-based, meaning programs can be updated individually and incrementally. This design enables continuous improvement of not only functionalities but also performance of the switch.

7.3.7 Protocol Oblivious Forwarding (POF)

7.3.7.1 Background

Conventional network devices only provide limited configuration capability. SDN decouples the control plane and the data plane. It allows users to create various applications/services and send the configurations (e.g., flow rules) to network devices via a standard southbound interface. Currently SDN provides programmability only for existing forwarding protocols. It cannot support new protocols automatically. To implement an application/service based on a new protocol, one has to 1) rely on device vendors to upgrade the device code/implementation to support the new protocol, and 2) rely on the related standards organization to update the southbound interface to support the new protocol. Another issue of current SDN is that the programmability is limited to flow rule configuration. It cannot create new flow tables and conduct in-band table modification at runtime. If a user needs to simplify or extend the existing packet forwarding pipeline, she still has to ask device vendors to upgrade the devices. New service deployment is time-consuming and costly, which limit the innovation of network applications.

IMT-2020 network has several use cases that drive the invention and introduction of new protocols, services, and architectures. The current network programmability focuses on control plane primarily. It is also needed to provide full programmability on data plane to support these use cases without changing the SDN southbound interfaces or relying on the device upgrade provided by the device vendors.

IMT-2020 network devices are required to provide full programmability that allows users to create, modify, or delete the packet forwarding and processing functions via SDN southbound interface. In addition, the SDN southbound interface in IMT-2020 system is required to be protocol oblivious, flexible, and has high performance.

7.3.7.2 Introduction of POF

Protocol Oblivious Forwarding (POF) enhances the OpenFlow-based SDN forwarding architecture. POF enables network devices to support any new protocols without modifying any code of the devices. To support new protocols, users only need to download new configurations into the forwarding devices. POF also enables users to create new forwarding tables on device data plane at runtime.

POF can help users to deploy new policies or services based on new protocols conveniently and rapidly. For example, carriers can implement security or QoS policies on any video, voice, or P2P protocols easily using POF.

In POF, flow table search keys are defined as $\{offset, length\}$ tuples, and instructions access packet data using $\{offset, length\}$ tuples. Figure 7.3-5 shows the IPv4 forwarding processing comparison between current protocol-dependent SDN and POF in data plane. In current protocol-dependent SDN forwarding processing, the data plane needs to be preconfigured the exact packet format and the processing procedure. For example, when the data plane packet processor processes TTL, the processor element should be preconfigured to understand what a TTL field means, where to get the TTL field, and how to process with TTL. In POF forwarding process, the data plane does not need to understand the semantic of packet protocols. Using the same TTL processing as an example, the controller only sends instructions equivalent to “find data with packet offset of 22 byte and data length of 1 byte, subtract the data by 1” to data plane. The POF data plane does not need to be aware of what TTL is but still can process TTL as user defined.

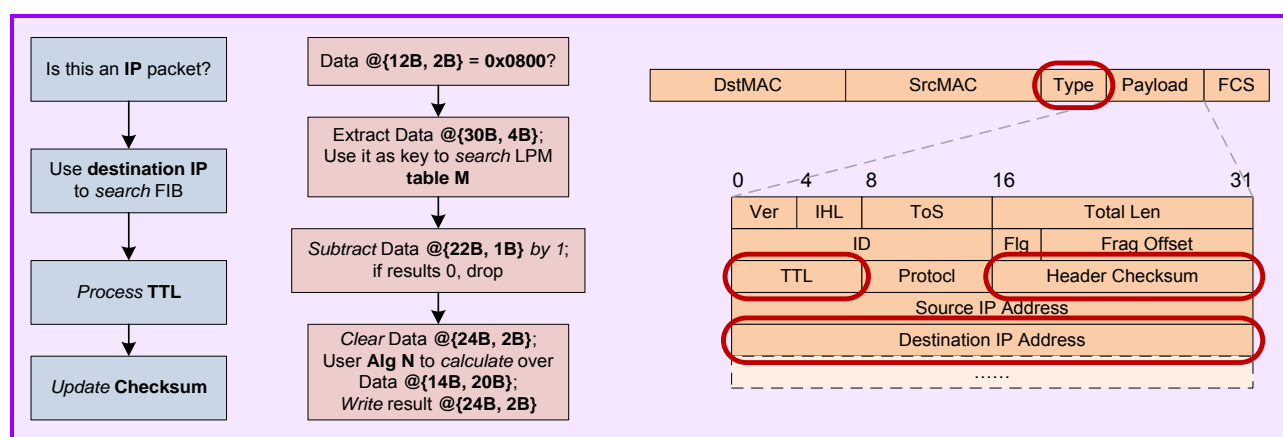


Figure 7.3-5 – IPv4 Forwarding Example

7.3.7.3 Data Plane Programmability of POF

POF supports full programmability on data plane as follows:

- The network data plane device resources are open to users;
 - The resources include table, port, metadata, counter, data memory, instruction memory, etc. Users can dynamically use these resources, which means users can freely create, modify, delete, and invoke these resources as needed anytime (e.g., creating a new forwarding table);
- Users can dynamically create flow tables as needed;
 - The table search key can be defined to be any parts of a packet header. i.e., the flow table search keys are defined as $\{offset, length\}$ tuples. It is not bound to any specific protocols. Therefore, any new protocol can be parsed.
- User can dynamically create forwarding instruction blocks as needed;
 - The data plane supports a set of built-in POF forwarding instructions. It provides the primitive packet processing commands such as goto_table, set_field, calculation, drop, output, etc. The role of the Forwarding Instruction Set (FIS) on network devices is similar to the role of the x86 instruction set in general processors.
- User can dynamically use FIS to build custom packet processing functions as needed at any time. The FIS is not bound to any specific protocol. Therefore, any new protocol can be processed. Users can define custom packet forwarding applications at control plane and apply them to data plane;
 - The packet forwarding application is composed of a set of flow tables, flow entries, and flow instruction blocks;

In IMT-2020 networks, POF can support any kind of forwarding protocols for services such as Internet of Things (IoT) and for content delivery such as Information Centric Networking (ICN) or Content Centric Networking (CCN). In ICN case, POF can implement both stateful and stateless name-based forwarding mechanisms including variable-length naming, data structure and forwarder logics, and POF supports parallel algorithms to create, maintain and lookup the protocol state tables at line rate. In general, POF supports dynamic programmability on data plane to enable processing any new protocol packets with flow tables, flow forwarding instruction block and flow entries.

To support IMT-2020 network slicing, POF provides finer granularity on data plane programming. User can create a packet processing pipeline customized for each slice with the minimum resource. POF allows smart table memory and instruction memory sharing so that each slice has the lowest possible resource consumption without performance loss.

The prototype of POF, built on NE40E platform, has been demonstrated in ONS2013. POF controller code and a software switch are open sourced and available at <http://www.poforwarding.org>. POF south bound interface is under consideration as the foundation of the next generation of OpenFlow in ONF.

References

- [7.3-1] Akihiro Nakao, "Flare: Open deeply programmable network node architecture," http://netseminar.stanford.edu/seminars/10_18_12.pdf.
- [7.3-2] Akihiro Nakao, "Software-defined data plane enhancing SDN and NFV." IEICE Transactions on Communications 98.1 (2015): 12-19.
- [7.3-3] Akihiro Nakao, Ping Du and Iwai Takamitsu, "Application Specific Slicing For MVNO Through Software-Defined Data Plane Enhancing SDN", IEICE Transactions E98-B, vol. 11, pp.2111-2120, 2015.
- [7.3-4] Ping Du and Akihiro Nakao, "Application Specific Mobile Edge Computing through Network Softwarization", IEEE International Conference on Cloud Networking (CloudNet), 2016.
- [7.3-5] Akihiro Nakao, Ping Du and Yoshiaki Kiriha et. al., "End-to-End Network Slicing in 5G Mobile Networks", Journal of Information Processing, IPSJ (to appear).
- [7.3-6] Akihiro Nakao, Ping Du, et.al. "Introduction to FG IMT-2020 network softwarization work and demo of softwarized LTE in FLARE network slices", <http://www.itu.int/en/ITU-T/Workshops-and-Seminars/201612/Pages/Programme.aspx>.
- [7.3-7] Lagopus switch web site, <http://www.lagopus.org/>.
- [7.3-8] Yoshihiro Nakajima, Hirokazu Takahashi, et.al. "Scalable, High-performance, Elastic Software OpenFlow Switch in Userspace for Wide-area Network", <https://www.usenix.org/sites/default/files/ons2014-poster-nakajima.pdf>.

8 Horizontal extension of slicing

8.1 Capability exposure and APIs

Next generation network will accommodate a lot of various types of devices, which belong to different industries. Therefore new diverse use cases will need to be supported by the network. The new uses cases are expected to come with a high variety of requirements on the network. For example, there will be different requirements on functionality such as charging, policy control, security, mobility etc. Some use cases such as Mobile Broadband (MBB) may require e.g. application specific mobility and policy control while other use cases can be handled with simpler mobility or policies. The use cases will also have huge differences in performance requirements.

Capability exposure based on network softwarization enables the operator to create customised network (e.g., a network slice) to provide optimized solutions for different market scenarios which have diverse requirements, e.g. in the areas of functionality and performance.

The potential operational requirements are as follows:

- The IMT-2020 system shall be able to customize the network functions within a slice dynamically based on the variation of the 3rd party (e.g., enterprises, service providers, contents providers, etc.) demand.
- The IMT-2020 system shall also support dynamic utilization of resources (compute, network and storage resources) within a slice as per the 3rd Application requirement, subject to operator policy.

Figure X illustrates the Capability Exposure Architecture for slice management.

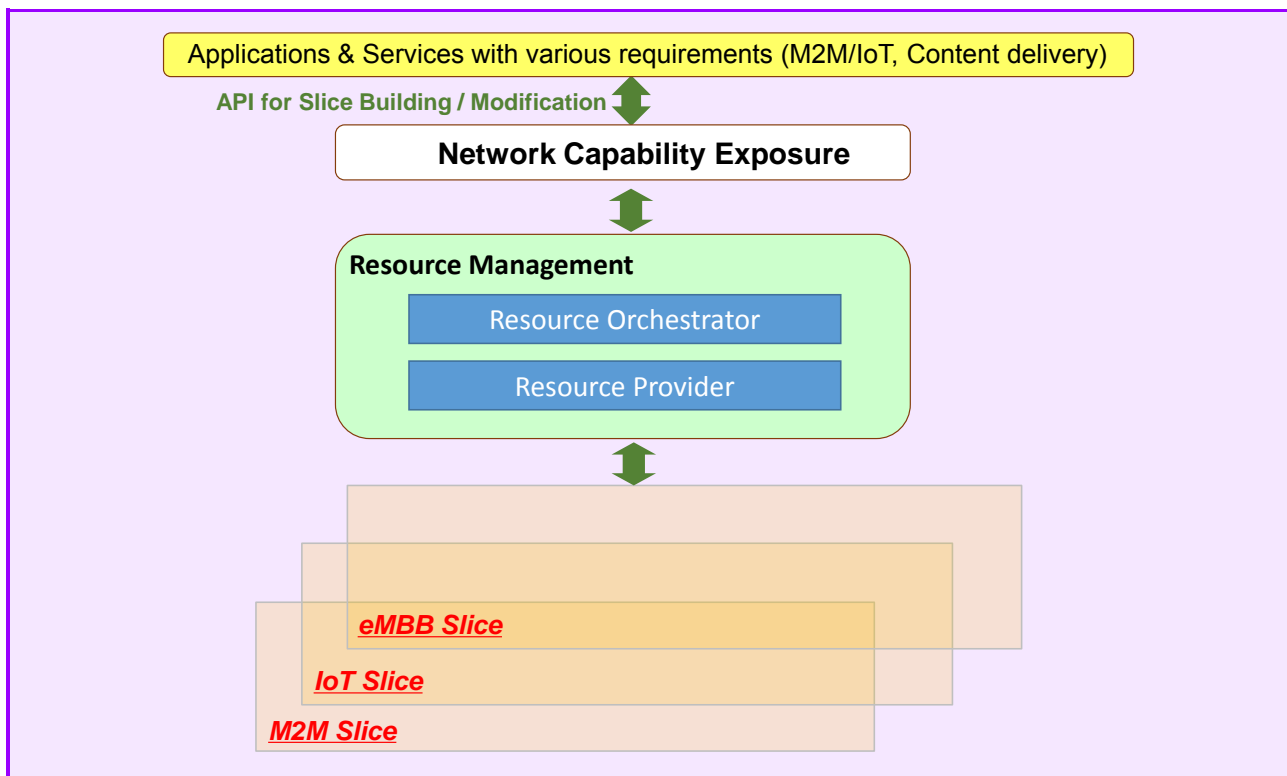


Figure 8.1-1 – Capability Exposure for Slice Building

Use Case 1: The Creation or instantiation of Slice Triggered by the 3rd Party

1. The 3rd party indicates the functionality and performance requirements to create a slicing via Slice Building API. In terms of implementation, a ServiceTemplate Profile may be sent to by the API. And this temple contains the parameters to describe the functionality and performance requirements.
2. The network capability exposure function transfers the above slice building request to the resource management function.
3. The Resource Orchestrator function authorizes functionality and performance requirements based on the agreement between the operator and the 3rd party. If the request is allowed, the Resource Orchestrator function forwards resource requirement to the Resource Provider and accordingly the Resource Provider allocates the required resource (hardware and software) to create or instantiate the dedicated slice.

Use Case 2: The Dynamic Modification of Functionality and Performance Configuration of Slice

1. The 3rd party indicates the modification of functionality or performance for a pre-created slicing via Slice Modification API. The modification may be triggered for the reason of lack of resource or new function needed by the 3rd party in the slice. In terms of implementation, a Service Template Profile may be sent to by the API. And this temple contains the parameters to describe the functionality and performance modification requirement.
2. The network capability exposure function transfers the above slice modification request to the resource management function.
3. The Resource Orchestrator function authorizes functionality and performance modification requirements based on the agreement between the operator and the 3rd party. If the request is allowed, the Resource Orchestrator function forwards resource requirement to the Resource Provider and accordingly the Resource Provider re-allocates the required resource (hardware and software) to modify the dedicated slice.

Resource orchestrator can act as the slice management and orchestration, and it can be realized in hierarchical way. Authorization between application and network capability exposure platform is needed. Resource provider can act as the resource management and orchestration, which realizes the deployment of network slice functions including network connectivity. Transport SDN is an underlying technology of providing the overall bandwidth guarantee on an instantiated network slice.

8.2 End-to-end slice

8.2.1 Overview

The softwarization of the IMT-2020 wireline network requires that nearly all components of the IMT-2020 network be programmable. Many of the wireline components available today are already programmable with various API's and it is expected that these will not be changed substantially but will instead be augmented to support new applications like 5G. As a result the softwarization process requires that multiple existing technologies together with the new technologies be blended in a manner that permits convenient programming of the entire end to end network from a single location. Since SDN is already (or will soon be) a major component of networks it is expected that SDN controllers/orchestrators of various types will be responsible for the different network components and that a layer above those SDN controllers/orchestrators will be required to control and co-ordinate the end to end 5G wireline behaviour.

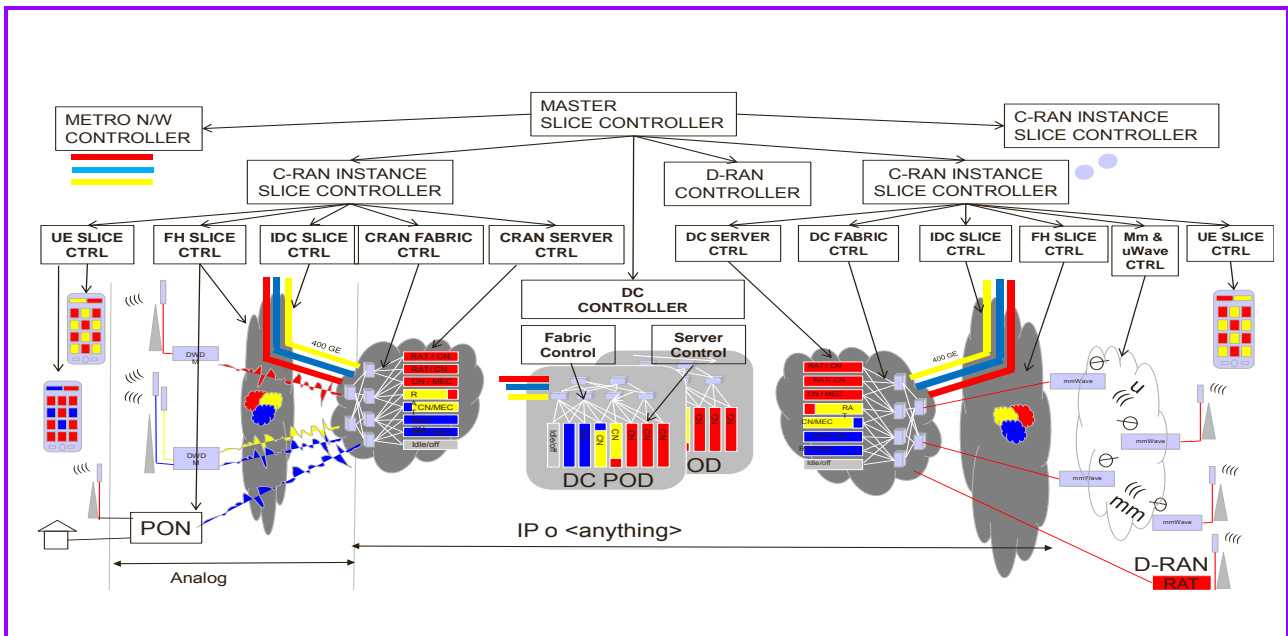


Figure 8.2-1 – Hierarchy of controllers and relationship to wireline orchestration

Figure 8.2-1 gives a likely logical/hierarchical arrangement of controller entities that are required to control the 5G wireline and in particular to implement slicing at the wireline level. Note that the picture depicts different controller entities but they may be combined of course. Note also that a controller does not uniquely imply a direct SDN (OpenFlow) model, indirect (MPLS/GMPLS etc.) distributed control planes with various flavours of PCE are also possible and even likely.

Based on the above figure, this contribution will start to elaborate the sequences of events that would occur as a result of various orchestration events. We do not try to detail the formats or APIs but rather what would happen when a particular orchestration event is requested or triggered. In other words we are discussing the actors and their general conversations not the exact language used.

First we begin with a list of high level descriptions of the various controller functions (actors involved):

Master Slicing Controller: Is responsible for an entire country or large region. It talks to the individual C-RAN and D-RAN slicing controllers and to the Metro & Core Network Controllers that govern the connectivity between them.

C-RAN Instance Slicing Controller: Is responsible for a Cloud Radio Access Network. It talks to the Fronthaul control, C-RAN Fabric Control, C-RAN server control and possibly to UE's (indirectly). If PON is used for fronthaul/backhaul then the slicing controller will talk to the PON controller. If PON is used to provide wireline access services for home/enterprise then the C-RAN Slicing Controller may talk to the PON controller to configure convergence between the fixed network and the wireless network by merging the relevant parts of the packet core processing on both networks. When mobile edge computing is available, some of the resources in the C-RAN can be assigned and interconnected for this purpose. There is no separate mobile edge computing controller; all resources are considered usable for all functions (RAN/CORE/MEC) in the CRAN. This does not preclude an ETSI MEC model, however it means an ETSI MEC model would need to be virtualized so that its resources are not dedicated.

D-RAN Controller: Is responsible to talk to all the Distributed Radio Access Network devices. Essentially this is a miniature C-RAN Instance Slicing Controller and is used when only a very few compute resources are distributed close to the antennas. In this case the fronthaul is likely nailed up and non-configurable and the fabric and processing controls are very limited; however some provision for mobile edge computing is probably provided and configurable but as with CRAN there is no separate mobile edge computing controller; the resources of the DRAN are usable for RAN/CORE and MEC as required. This does not preclude an ETSI MEC model, however it means an ETSI MEC model would need to be virtualized so that its resources are not dedicated.

Metro Network Controller/Core Network Controller: These controllers are pre-existing SDN or TSDN based controllers which control the mesh of network elements that provide DWDM/OTN and Packet connectivity on the scale of a metro or a country. Here we use the term "Core" in its networking not wireless sense.

FH slice control: This controller will ensure that a given Antenna is connected to the C-RAN fabric and ultimately to the proper processing entities responsible for it. Since fronthaul can be analog, digitized analog, mmWave, uWave, or even over packet, there are a lot of possibilities from complex, where there is a mesh network to control, to trivial, where the interconnect is a point to point fibre. It is also possible that different front haul technologies are used in different locations in the network based on fibre availability which may imply different RAT split choices at different locations and possibly even at different times.

IDC slice control: This controller will manage the connectivity C/D-RAN to C/D-RAN and C/D-RAN to DC Pods. This is most likely a transport SDN controller if the connectivity is OTN/DWDM, or an IP SDN controller if the connectivity is IP/MPLS. It is possible there are two controllers if the connectivity is IP/MPLS over OTN/DWDM. In the case where the IDC is implemented over a dark fibre then control is subsumed by the DC or C-RAN controllers.

C-RAN Fabric Control: This controller manages the fabric inside the C-RAN. The C-RAN has very tight timing requirements and therefore a normal DC Fabric Controller must be augmented. Clock distribution is required as are control of pre-emption, scheduled Q's and other very high performance packet behaviours. Possibly bypass tunnel OTN/DWDM may be required within the C-RAN for delay/jitter requirements depending on the architecture.

C-RAN Server Control: This controller manages all the processing entities in the C-RAN. These include general purpose servers, servers with FPGA associated (hybrid), and likely non-standard acceleration hardware required for special purpose digital signal processing. These servers may run VM's, containers, bare metal and may be used for RAT, MEC and CORE as required.

DC Controller: This controller is a pre-existing Data Center controller or hierarchy of controllers that are not only dedicated to wireless. Their job of allocating connectivity between compute resources and of allocating compute resources is simply re-used to allow control of 5G related packet processing entities to reside outside of the C-RAN/D-RAN and to allow chaining of the relevant functions as dedicated by the packet core chains for a given slice. The resources controlled in the DC are unlikely to be able to run RAT due to their distance from the Antennas and the jitter/loss requirements; however they should be able to run MEC and CORE equivalently with no distinction between the two.

8.2.2 Orchestration actions

This section lists the actions that the orchestration system will have to process and gives a high level description of what each action means. Subsequent sections describe each of these actions and then break them down into the timeline of sub-actions that must occur in the layers below. Again this is done without specifying any particular format. For most of these actions it is assumed that a **SliceProfile** exists that describes in a high level language the desired properties of a slice.

A **SliceProfile** at a high level has to describe the radio access technology to be deployed, at what bands, on what antennas, and then must describe the packet processing functions (vCore) that are required to support the RAT. The profile must give an indication of the minimum amount of processing required for each component and any constraints on the various functions as to their delay/placement.

Components such as a CRAN, DC, IDCNetwork, uWave network etc. Have **Capabilities** which enumerate the resources they have available for higher level orchestration. All resources have network wide unique names and where resources in one component connect to another it must be possible to infer the connectivity through a name match database. For example if a router that is part of a metro network has a port that is connected to a switch port in a datacenter then the resources in the metro network need to have the same name for the interconnection as the resources for the datacenter. There are various ways to achieve this such as juxtaposing the sorted pair of the names of the two end points as discovered through well known in band adjacency discovery protocols.

Action_connectToCran: This action will bring a new C-RAN under the orchestration control. Auto discovery is unnecessary and configured secure tunnels is likely sufficient. Capabilities of the CRAN: **CranCapabilities** will be propagated into the orchestrator.

Action_connectToDran: This action will bring a new D-RAN under the orchestration control. Auto discovery is unnecessary and configured secure tunnels is likely sufficient. Capabilities of the DRAN: **DranCapabilities** will be propagated into the orchestrator. Not all resources need to be propagated, and it may be desirable to propagate subsets to different orchestration systems.

Action_connectToIDC: This action will bring a new inter data-center network controller under the orchestration control. Auto discovery is unnecessary and configured secure tunnels is likely sufficient. Capabilities of the inter data center **IDCCapabilities** will be propagated into the orchestrator, the names of interconnection end points must match to allow end to end orchestration. Not all resources need to be propagated, and it may be desirable to propagate subsets to different orchestration systems

Action_connectToDC: This action will bring a new metro DC under the orchestration control. Auto discovery is unnecessary and configured secure tunnels is likely sufficient. Capabilities of the DC: **DCCapabilities** will be propagated into the orchestrator. Not all resources need to be propagated, and it may be desirable to propagate subsets to different orchestration systems.

Action_connectToUWave: This action will bring a new micro or millimetre wave network under the orchestration control. Auto discovery is unnecessary and configured secure tunnels is likely sufficient. Capabilities of the network are expressed as **uWaveCapabilities** and will be propagated into the orchestrator. Not all resources need to be propagated, and it may be desirable to propagate subsets to different orchestration systems.

Action_createSlice: This action requests the creation of a new **Slice** but does not activate it. It will take a **Profile** of what the new slice looks like, produce an internal representation (called a **Slice**) which can be used to work through the allocation of resources and track the state of resources as they are activated by subsequent actions. The contents of the **Profile** are an important aspect of softwarization which needs to be defined.

Action_computeSliceResources: This action given a **Slice** will check to see if the slice can actually be instantiated given the **xxxCapabilities** of the attached components. If not it would report what other actions would need to be taken to allow this slice to be activated. This likely involves an optimization over all resources in all **xxxCapabilities** to see what the best allocation of resources to this slice would be. An interesting question is how resources can be stolen by a higher priority Slice from a lower priority Slice (**bumping** is the traffic engineering term).

Action_activateSliceResources: This action causes a *Slice* to be activated according to the results of a computation of which resources should be allocated. It will have to configure the fronthaul network, decide where the DSP needs to go, allocate the resources both hardware and software for the RAT/DSP create any necessary fabric interconnect in the C-RAN, configure any inter C-RAN connectivity, configure any datacenter fabric, configure the servers to run the core, pick the software for the core and request it be started, configure any IPVPNs and/or L2VPNs, chain together the core elements and of course provide QOS control end to end. These would be done by making requests to the CRAN components, DC components and IDC components controllers respectively. The error handling here is complex, for example, should the slice come up if all resources are not available.

Action_computeSliceResourceDelta: This would take a running Slice and a new Slice definition based on a changed profile and would compute what new resources are needed (or what needs to be released) to make the running Slice equal to the new Slice. It does not take any action but simply tries to optimize what the changes should be and augments the running Slice with the delta results.

Action_activateSliceResourceDelta: This uses the results of the delta computation action to actually allocate/de-allocate resources in the running slice to bring the current slice into agreement with the delta. If a RAT / CORE or MEC function is going to be aborted it should be given time to gracefully shut down and therefore interfaces would be required to communicate this between the orchestration system and these applications.

Action_shutdownSlice: This would force a slice to be shut down. There are a variety of ways to do this from a graceful slow approach to a sudden and abrupt termination and freeing of all resources. The *Profile* could specify shutdown behaviours for various components. The RATs/COREs/MECs functions should be informed that they are going down and given time to gracefully shut down prior to being aborted.

Action_querySlice: This event would return the status of the slice. It would query the independent components that make up the slice and would returned detailed status of each individual component. Various wild carded forms of query should be supported to get subsets of the information.

Action_xyzSlice: A place holder for many more possible actions.

8.2.3 Action timelines

This section lists the rough sequence of events that have to occur for a given action at the orchestration layer and what the sub controllers would need to do in response to that action.

Action_connectToCran: This action will attempt to establish a secure tunnel to the controller for the named C-RAN. A key exchange is done to obtain a secure connection after which a request is sent for the *CranCapabilities*. These capabilities should contain all the antennas, their locations, properties, name of their attachment points (what would switchable fronthaul look like outside the C-RAN?), descriptions of the general purpose compute resources and special purpose hardware resources and the fabric that interconnects them together with different RATs available and for each its requirements in terms of compute / acceleration components and available spectrum options.

Action_connectToDran: This action will attempt to establish a secure tunnel to the controller for the named D-RAN. A key exchange is done to obtain a secure connection after which a request is sent for the *DranCapabilities*. These capabilities should contain all the antennas, their locations, properties, name of their attachment points, descriptions of the general purpose compute resources and special purpose hardware resources and the fabric that interconnects them together with different RATs available and for each its requirements in terms of compute / acceleration components and available spectrum options. This is very similar to a CRAN except much smaller and may have only a few antennas and very limited compute resources.

Action_connectToIDC: This action will attempt to establish a secure tunnel to the controller for the named Inter Data Center network (metro etc.). A key exchange is done to obtain a secure connection after which a request is sent for the *IDCCapabilities*. Since the IDC controller may be flexible and able to allocate new resources on demand, the Capabilities are not hard upper bounds and may have a flexible representation to indicate this. The Capabilities should list the attachment point names such that correlations can be made to the attachment point names in the C-RANs and other DC's or other networks.

Action_connectToDC: This action will attempt to establish a secure tunnel to the controller for the named Data Center. A key exchange is done to obtain a secure connection after which a request is sent for the **DCCapabilities**. Since the DC controller may be flexible and able to allocate new resources on demand, the Capabilities are not hard upper bounds and may have a flexible representation to indicate this. The Capabilities should list the attachment point names such that correlations can be made to the attachment point names in the C-RANs and other DC's or other networks. The Capabilities include server capabilities and the availability of different packet processing functions (vEPC etc.).

Action_connectToUWave: This action will attempt to establish a secure tunnel to the controller for the named U or mm Wave network. A key exchange is done to obtain a secure connection after which a request is sent for the **UWaveCapabilities**.

Action_createSlice: A **Profile** needs to be parsed and given a high level integrity check. So the resources described have to be verified that they in fact exist by name although at this point we are not checking to see if they are free. Essentially this is an internal representation of a description of a slice in a compiled form.

Action_computeSliceResources: The internal form of the slice is subjected to a computation to see which resources are free and best suited for this slice. There are many possible optimization goals here and it's unclear how 'bumping' of resources should be done. This action would need to determine what resources are needed for fronthaul connections, which CPU's / Acceleration hardware is required for the RAT(s), check backhaul capacity both inter and intra C-RAN / DC, check DC/C-RAN resources for CORE/MEC functions. This is likely the output of a global optimizer of some kind with a goal of minimizing total cost but there are many optimization goals and methods possible.

Action_computeSliceResourceDelta: The internal form of the slice is subjected to a computation to see which resources can be added to the slice to increase (or decrease) its capacity while best maintaining some overall network objective function. For example if we needed to temporarily increase the CORE capacity, or decrease the RAT capacity such an action could be invoked with the new slice and old slice requirements. The delta would be computed and then an optimization run to determine how best to address the delta.

An interesting question is how to address the addition of a Virtual Network Operator with its own core to an existing RAT, or how to address the addition of a dedicated CORE for some users of a RAT. Are these changes to an existing slice, or are these considered new slices with a shared RAT? These could actually be managed internally by the slices' OAM itself or it could be done at a higher layer by the 5G wireline orchestration. How is this managed today for MVNO attachments to a common 4G eNodeB?

Action_activeSliceResourcesOrDelta: After computing how best to allocate resources or delta resources to a slice this action would be responsible for taking that action and validating that it has happened correctly. In the event it is unable the previous state of the slice should be restored. Where possible this action should be without impact on end users although this is likely very difficult for some cases. An option should be provided to allow the operation to take place over some time interval so as to minimize impact or wait for lulls in usage to better mitigate impacts. An antenna may be re-assigned, spectrum changed, fronthaul may be reconfigured, RAT processing increased or decreased, C-RAN fabric adjusted, CORE processing increased or decreased, inter/intra C-RAN connectivity increased or decreased, inter C-RAN/DC connectivity increased or decreased, MEC processing capacity can be increased/decreased or moved.

8.2.4 New "user" defined actions/triggers

This section discusses programmability from the perspective of the operator. Previously we discussed orchestration actions that are likely required for normal operations however it's impossible to anticipate all of the likely cases. In such circumstances it is normal to create a programmatic tool kit to allow extensions by the user (in this case the network operator). In almost all circumstances such programming environments involve a set of trigger conditions specified in some Boolean language together with user defined actions. The orchestration system, together with the various controllers, is responsible for efficiently monitoring the trigger conditions and when they are met, to launch the set of actions programmed by the user. For example:

ON: \$time==10:00:00 EDT && slice("slice-A").sector("name").activeUEQuantity() >100
DO: userAction3(slice("slice-A").sector("name").activeUEQuantity());

For example we may want to look at the utilization in a particular set of sectors in the network and if they are above some threshold and growing we may want to look for another sector in a lower priority slice where we can steal some resources, reassign them to the higher priority slice in the sectors in question allocate new CORE CPU, interconnect it to the RATs and then when things calm down, return the resources. In order to do this the orchestration system will need to allow the user to specify the appropriate conditions including which sectors to check, how to determine utilization, look at the trend in utilization, check it against a threshold, then look for low utilization sectors in lower priority slices, then determine which has more resources than necessary (or can be sacrificed), then compute a delta to the current higher priority slice and finally apply the resources.

We can imagine a long list of things that can be queried, of Boolean operations and of a primitive set of actions must be defined in order to implement such a tool kit. Various interfaces for Java/Python/C++ etc. would need to be provided.

In addition to match/action type behaviours there is also a need for various **auditing functions**. It would seem likely that the orchestration system should audit itself and the network for correct functioning via various assertions.

ASSERT: slice("slice-A").sector(*).frequencyRange().notOverlaps(2.60Ghz, 2.90Ghz) ;

It's very likely that two independent orchestration systems will be required. One which can perform the actions and another which simply audits and ensures that high level rules of behaviour are in fact being honoured and issuing warnings or taking punitive actions if they are not. One example would be to monitor spectrum use to ensure that no license rules are being violated. Since the various RATs are highly programmable it would be relatively easy to accidentally transmit on the wrong band in the wrong location or wrong power level. An auditing system would allow for the specification of rules which must at all time be met and make it less likely the system would operate for too long in an unsafe or non-compliant manner. Auditing could be done either at the time of specification or at the time the change is actually implemented, or post change.

8.2.5 Cookies

"Cookies" is a term used to describe data which is stored by a second software entity on behalf of a first software entity without really knowing what it is. Cookies are useful because they allow state to be maintained even when the first software entity goes away completely.

The various components that the orchestration system talks to should store cookies on behalf of the controller sub-systems associated with the various objects under their control. This will allow an orchestration system to resynchronize with manual or previously automated slices and greatly reduce the burden of migration and auditing. For example, resources associated with a particular slice could be associated with a cookie whose key is the name of the slice, this includes all the end to end resources such as antennas, fronthaul, C-RAN fabric, C-RAN compute, etc. and all could carry such tags. This is also a very useful debugging tool and would allow lower level query as to what resources were being used for without talking directly to the higher layer orchestrator.

8.2.6 Control plane enhancement for 5G transport network

8.2.6.1 Control plane for low latency, jitter and packet loss

It is expected that many critical services which require ultra low latency, jitter and packet loss will be carried by the 5G network. For example, an E2E latency requirement of 1ms is being considered for applications like tactile communication, AR and mission-critical IoT.

This stringent QoS requirement is a big challenge for existing transport network, and some innovative data plane technologies are being considered, e.g. IEEE802.1 TSN, FlexE and Detnet. By employing techniques such as zero congestion loss, pinned path, packet replication and duplication, Detnet aims to create layer 3 forwarding path with determined QoS, while IEEE 802.1 TSN and FlexE and technologies for deterministic layer 2 transport.

Although there could be some gains to use these technologies in some portion of the network, it is important that corresponding control plane mechanisms are necessary to efficiently integrate and coordinate this kind of technologies in the whole transport network.

8.2.6.2 Control plane for guaranteed bandwidth

5G network will provide diversified services on the same physical infrastructure, some of these services require strict bandwidth guarantee, while some others have flexible bandwidth requirement and can adapt via statistical multiplexing. It will be a challenge for existing transport networks to meet the bandwidth requirements for both kinds of services simultaneously.

Some technologies such as FlexE can provide guaranteed bandwidth on an Ethernet data link for a specific service. However in order to achieve the bandwidth guarantee on the end-to-end transport network, some additional control plane mechanisms are needed.

8.2.6.3 Control plane for very large scale network

Widespread deployment of small-size cells is required to support high-speed and large-capacity mobile communications. For instance, assuming that a macro cell of 2km radius is replaced with small cells of 200m radius, the number of cells calculated based on the surface area would increase 100 times. As a result the scale of the transport networks need to increase accordingly. The control planes must also be sufficiently scalable for such large scale transport network. It is anticipated that neither totally centralized control nor completely distributed control could meet such requirement, thus a hybrid control mode should be considered. Enhancements to both the existing distributed protocols and the centralized SDN mechanisms would be needed, and the control plane would make use of the advantages of both approaches.

8.2.6.4 Control plane for flexible connectivity and topology

With the evolution of distributed mobile Core, and the introduction of C-RAN and MEC, 5G services will have diversified connectivity requirements. The connections will not only be between the RF sites and the centralized mobile core, but would also include various connectivity among the RF sites, C-RAN, MEC, the distributed Core (or edge located Core) and other DCs. This requires the transport network to provide flexible connectivity among all these entities. In addition, since network slicing will be used to provide dedicated slices for different services, in each slice it is necessary to provide specific service with the customized network topology. Potential control plane mechanisms need to be investigated to establish the customized network topologies and set up the required connections in a flexible manner.

8.2.6.5 Control plane for flexible resources assignment and sharing

With the network slicing mechanism, when a virtual or physical network slice is created, the control plane should be used to dynamically assign the required resources for the slice on the physical network. Since different market segments and verticals such as mobile broadband, MVNO and various IoT services will use different network slices, the control plane needs to support flexible resource assignment to meet diverse resource requirements. Typically the network resources assigned to network slices can belong to two categories: dedicated or shared. Dedicated resources can only be consumed by the assigned slice, and should be isolated with other slices, while the shared resources can be shared among several slices. Control plane should provide mechanisms to support both resource isolation and sharing.

In addition, when a dedicated resource has been allocated to a slice, a mechanism must exist so that the actual resource usage can be monitored and can grow or shrink as required. In this manner the control plane could provide a slow adaptive method to allow resources to move from slice to slice without having to use statistical multiplexing. Ideally this should be done in a make-before-break manner.

8.2.7 Control plane considerations for transport network slicing

8.2.7.1 Hybrid control plane for transport network slicing

In order to create an End-to-End network slice for 5G services, the control plane needs to obtain the network topology information, and be able to compute paths which meet both the policies and service performance requirements of the slice. The control plane also needs OAM mechanisms to monitor the real-time network status and performance of the paths it has created for the given slices. This performance information is then used as an important input for reoptimization and protection computations.

When an End-to-End network slice is initiated, different parts of the End-to-End network would respectively use the appropriate technology to realize the mapping from network slice to the physical infrastructure and resources. In addition, the stringent performance requirement of some 5G services requires that many innovative technologies be used in the transport, most notably FlexE, Detnet, etc.

When a service is provisioned in a specific network slice, it is necessary to monitor the performance of the service to ensure that network SLA is always guaranteed (this is especially true of statistically multiplexed connections), and trigger optimization, protection or restoration of the network slice when needed.

A logically centralized controller can be used to create E2E network slices according to specific service requirements. In real network deployments, this logically centralized controller is usually realized with the combination of a distributed control plane and some centralized control functions (in SDN this is referred to as an in-direct model). This is not only because distributed control planes already exist and can't be replaced immediately, but also due to the fact that the distributed control plane has lots of advantages, such as scalability, short response time and robustness against network failure and insensitivity to a central controller failure. Therefore, a hybrid control plane (in-direct model) should be considered for 5G network slicing especially with respect to the transport and packet portions of the end to end slice

8.2.7.2 Layered architecture for transport network slicing

The E2E communication infrastructure consists of UE, RAN, transport network, core network, mobile edge computing network, etc. Each segment will choose suitable slicing technologies to achieve several slices, and the slices in each segment are combined together to achieve E2E slicing. For example, one UE slice can match one application or some applications which have similar requirements in the terminal. RAN slicing refers to slice software and hardware resources to meet different service requirements. Meanwhile, RAN slicing could coordinate with different core network slice to provide differentiated services. Transport network slicing can obtain separated transport network slices in the same physical network, and each network slice provides the service with the required characteristics, such as delay, bandwidth, packet loss rate, etc. The core network slices that deal with authentication and mobility etc., and mobile edge computing network slices that offer users cloud-computing capabilities and various applications at the edge of the mobile network can be provided with combination of network functions which run on partitioned computing, storage and network resources to meet the requirements of particular services.

This section describes the proposed control plane architecture for transport network slicing. Essentially it is a layered architecture:

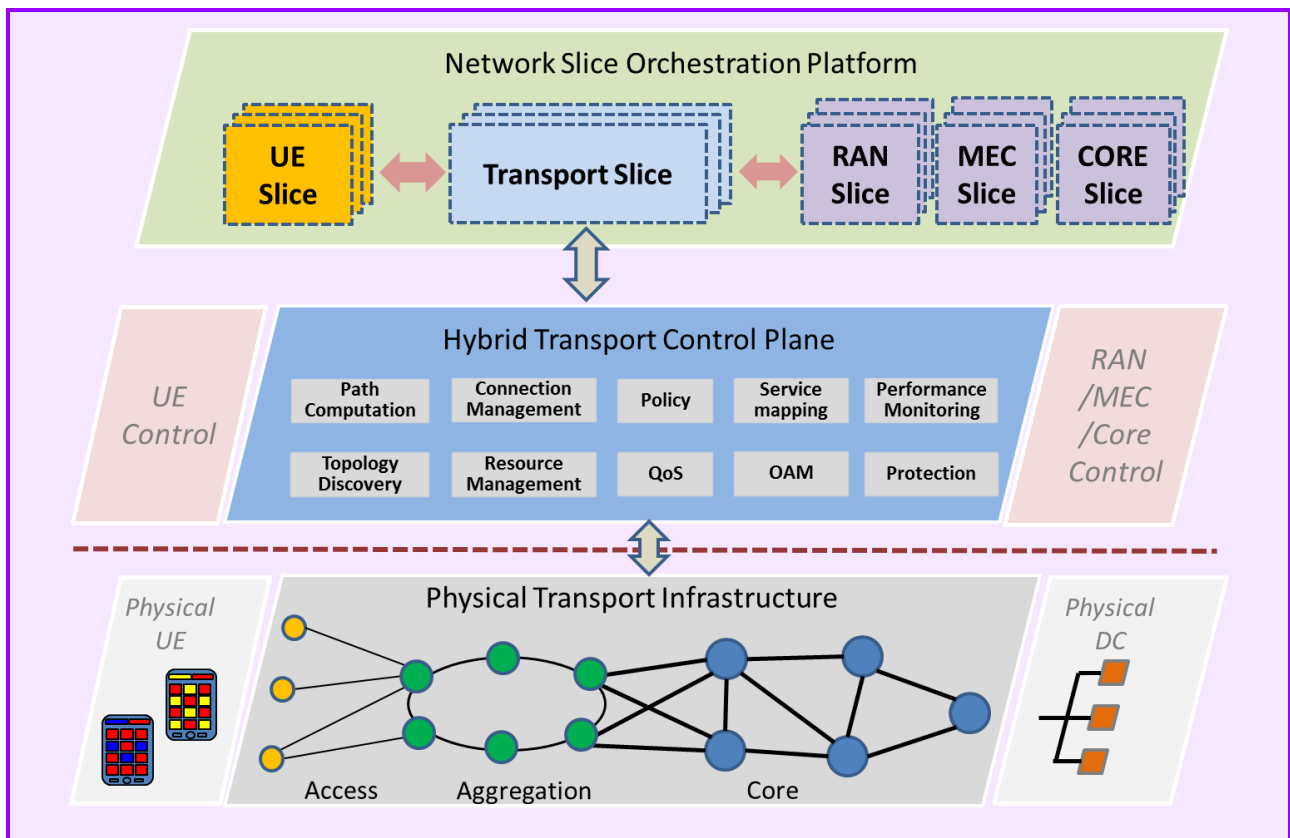


Figure 8.2-2 – Control plane architecture for network slicing

In the upper layer, the virtualized network components are managed by the network slice orchestration platform, which orchestrates the virtualized components and functions to create the End-to-End network slice according to the service requirements. This layer does not need to be aware of the different technologies used in the physical infrastructure.

In the lower layer, the End-to-End virtual network slice is instantiated in different parts of the physical network infrastructure, which means this layer is responsible for the mapping from virtual network slice to physical infrastructure.

For the transport network, the lower layer is the hybrid transport control plane, which provides the adaptation to different transport technologies and the interaction with physical devices. It utilizes the capabilities of the physical transport network to provide the required performance and characteristics for the network slice. To achieve this, the lower layer includes a set of necessary network functions, such as topology discovery, path computation, connection management, service mapping, OAM and protection, etc. Some of these functions can be provided by a centralized controller, while some others may be better implemented with a distributed control plane. For those centralized control functions, standard southbound interfaces to the network elements need to be considered.

9 Application scenario of network softwarization

9.1 Scenario 1: Edge computing

The application of MEC&SFC in IMT-2020

Mobile Edge Computing is a natural development in the convergence of IT and telecommunication networking. Based on a virtualized platform, MEC is recognized by the European 5G PPP (5G Infrastructure Public Private Partnership) research body as one of the key emerging technologies for 5G networks (together with Network Functions Virtualization (NFV) and Software-Defined Networking (SDN)), and can satisfy the demanding requirements for ultra-low latency and stimulating innovation in IMT-2020 network.

By application of MEC and SFC in mobile network, operator can extend the network softwarization scope into service application domain, to support programmability on applications platform at edge network.

MEC system can be deployed at multiple locations at edge of mobile network, such as at the access network aggregation site, or at the edge of the core network. When there have multiple MEC application servers installed in one MEC system, MEC system can be deployed with combination of SFC architecture, figure 4 illustrate an example of MEC deployment scenario in IMT-2020 network.

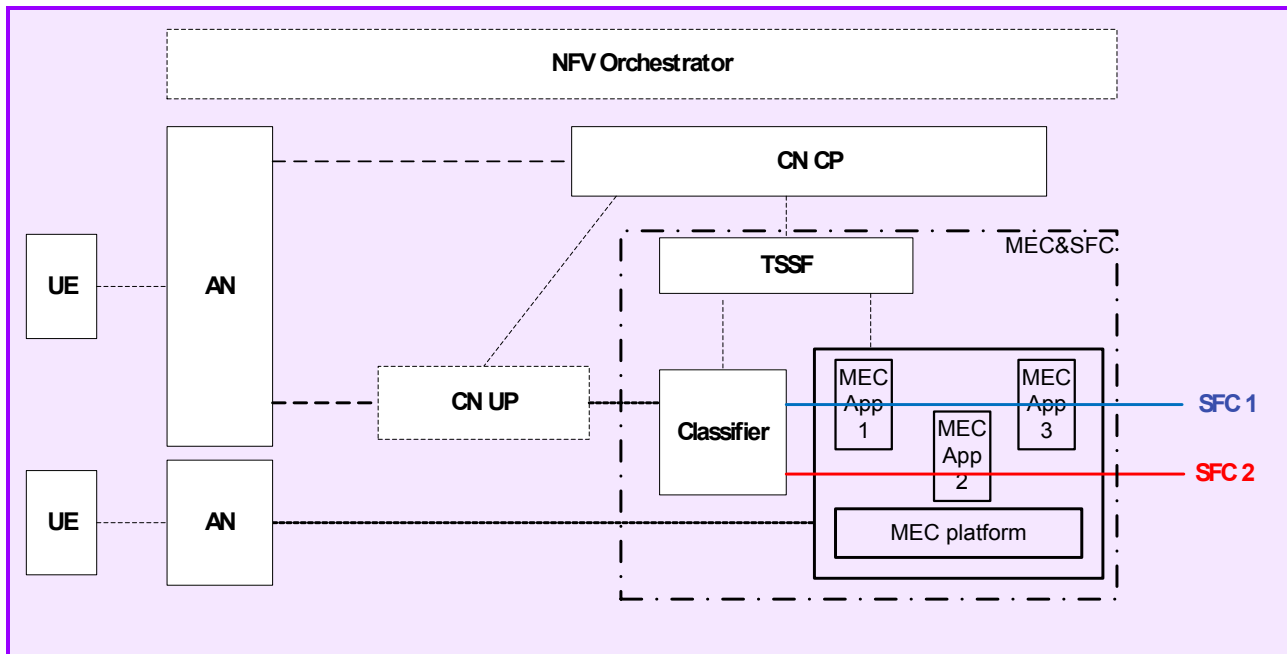


Figure 9.1-1 – An example of MEC deployment scenario in IMT-2020 network

9.2 Scenario 2: ICN on a Slice

9.2.1 ICN value to 5G network

One of the benefits of Network softwarization is the capability of providing the merging network architecture as a slice. Especially, owing to the data plane programmability, emerging network architectures using new packet forwarding schemes such as ICN can be realized as a slice.

ICN is one of the emerging network architecture which is widely studied. It is expected to become a solution to cope with problems in 5G era such as increasing video contents traffics, shorter response time requirement, resilient to disasters, and IoT accommodation. Because of these possibilities, ITU-T SG13 question 15 FNDAN (Future Network Data Aware Networking), which is basically the same as ICN, is discussing about the standardization of this new architecture.

Basic feature and merits that ICN will provide are as follows:

- **Server location independent access by contents name:** In ICN, contents are accessed by its name instead of the server location information that current IP based content retrieval is using.
- **Traffic and possible response time reduction by in-network caching:** ICN network nodes are equipped with a content store which caches content going through a node by an autonomous selection of content to cache based on the needs of the users accessing the node. Content will generally move towards the network edge node where the specific named content is frequently requested. Once the popular content is cached at the network edge node, subsequent requests of the same content will be served at the network edge node, resulting in a total reduction of the network traffic and lessening the overall server load. This operation mode also reduces the response time because of the shorter travelling path length.

- **Easy provisioning of in-network data processing:** In-network data processing is a concept that network nodes will perform network wide data processing and provide application services with the aim of reducing the network congestion as well as shortening response time. In-network processing can be considered generally as an expanded form of edge computing, where data processing and service provisioning will be provided dynamically at any place on a network that is appropriate. In this case the requested name is the data processing and service instead of contents. Studies toward this direction are going on as a sub category of ICN called as NFN (Named Function Networking). Due to the dynamic nature of service and data processing points, ICN's basic mechanism of accessing by name rather than location is especially suitable to provide in-network data processing. This works well for IoT use cases, such as assembling a large number of small granularity data, applying first stage processing, and providing on-path data processing on a transmission path which is frequently used in big data processing.
- **Contents security:** In some ICN architecture such as CCN and NDN, content security is provided as a basic function. Since security is a key concern in several systems like content delivery and IoT, having a built-in security mechanism is very attractive point of ICN.
- **Robustness to network failures by multi path routing:** To enable the content access by name, ICN routing/forwarding is capable of multi-path routing, because the contents once cached in certain node will not be available at the next chance. In ICN multi-path routing, when the response does not come back from the direction the interest is sent out, the node will automatically issue the same request to another direction. This mechanism is very helpful when the part of the network failed down such as the disaster case, and makes the network robust to the failure.

9.2.2 Network softwarization benefits to ICN

Even ICN can provide many attractive features, it will take long time to make a new network operated in the real field. In addition, ICN is still in the research stage, and the very frequent revision will be expected. To enjoy the possible merit of ICN in early days, network softwarization can contribute in the following aspects.

- Network softwarization can easily provide with an emerging architecture network on the basic network platform as a slice when the node is realized by software. Since current ICN node is realized by software, ICN can enjoy this benefit, and can offer some attractive features from the early days.
- Version change is easy on network softwarization platform. This is also a big benefit for the evolving network architecture such as ICN.
- ICN employs new data forwarding scheme different from the current systems. Then the data plane programmability provided by network softwarization is an essential tool to realize the ICN slice.
- There remain some functions to be developed for current ICN. The pragmatic approach to provide ICN application services in early days is the ICN overlay approach on popular network architecture such as IP. It is expected that basic functions of popular network architecture are prepared and provided as slice elements in network softwarization. This situation will ease the ICN overlay realization. This also works to support the migration scenario of ICN.
- Each network architectures has strong side and weak side. From the application service view point, it is attractive to use different network architectures during the provision of an application service using the strong side of each network architectures (hybrid approach). Since the network softwarization will provide different network architecture slices, the hybrid approach is feasible. The current ICN needs more functions to be developed, and the hybrid approach is beneficial to provide the attractive features of ICN by supplementing the yet lacked features by other network architecture slices.

9.2.3 IoT on ICN use case

The use case of ICN tailored to IoT described here focuses on the IoT aspect of ICN such as name based smart routing, In-network processing, and the content security aspect.

Figure 9.2-1 shows the concept of real time IoT information search using ICN smart routing scheme and In-network data processing. In this case, the user wants to find the location of the lost child or elder person. He submits an Interest packet requesting the person search processing. The interest packet contains the following information.

- Person search processing request by processing name
- Targeted geometrical search area name
- Information of target person (Still picture, etc.) as an extended field of name

By receiving this Interest packet, the smart routing function of each node checks the face of each node to forward based on the keyword such as geometrical search area and lost people search processing, and if necessary send the interest to plural faces at a time. When the Interest approaches to the targeted area, it is automatically spread to relevant nodes that cover the area, and finally the monitor cameras. Monitor cameras received the Interest send back their images as Data packet to the network. Then the Data packet receiving node checks whether the person search processing program is available on its node. If not available, it forward the Data packet back to the face that the Interest packet came. On the other hand, when the person search processing is available, the node analyses the data from a monitor camera with the targeted person's information. When the image fits, the node send back the "find" message with extracted image to the requester. When the image does not fit, the node terminates the Data packet. By this mechanism, the user can receive the service by sending just one Interest to the network. In the current IP, user himself must find the locations of the monitor cameras in the targeted area, establishes many sessions with these cameras, and apply the person search program on the collected data. Compared to this, new service scenario described makes the operation much easier for the user to receive a service, and also considerable bandwidth saving for image data transmission can be achieved.

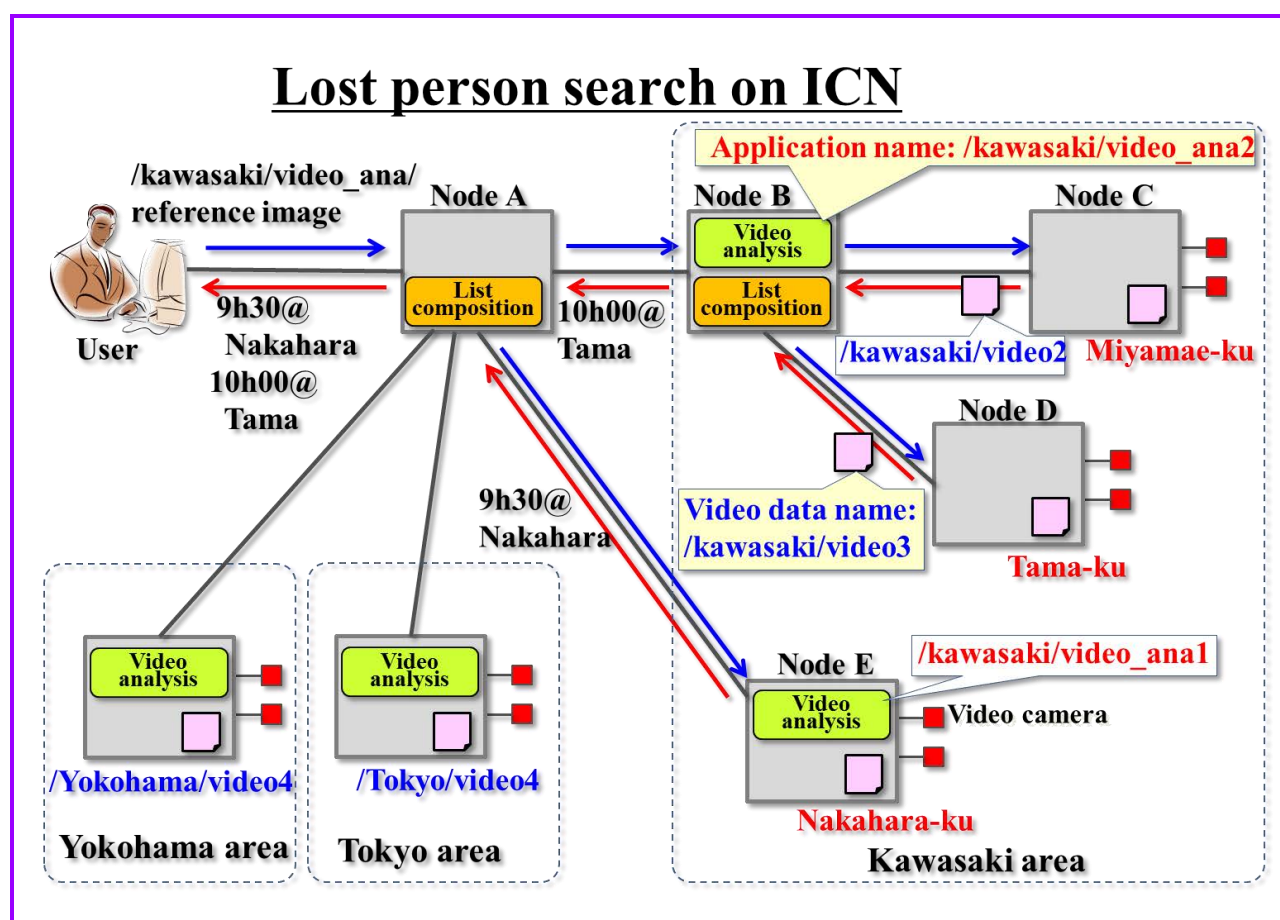


Figure 9.2-1 – Lost person search example on ICN

The key component of this application is a smart routing technology that enables an efficient interest forwarding based on a keyword table. Network node does compose/decompose of basic message of ICN (Interest, Data). There are four patterns depending on whether request should be decomposed or not, and whether response should be composed or not. Figure 9.2-2 shows these patterns. Which pattern should be used is either decided by node policy or automatically decided by what is requested in request message. As for decompose of request (B1, B2), a request message (Interest) is split into different messages depending on what is requested in request message and location of each content. In case of no decompose of request (A1, A2), a request message is just copied as it is and multiple copies are sent to multiple producers. For both cases a requester only has to send one request message, but for the former case (B1, B2), content volume of response messages is expected to be reduced. As for compose of response (A2, B2), a response messages (Data) are merged into one message, where multiple contents are packed in it. In case of no compose of response (A1, B1), each response message is sent to a requester separately. For the former case (A2, B2), the number of response messages are expected to be reduced. In addition, if there is duplication between content in each response, an intermediate node can remove it and reduce content volume.

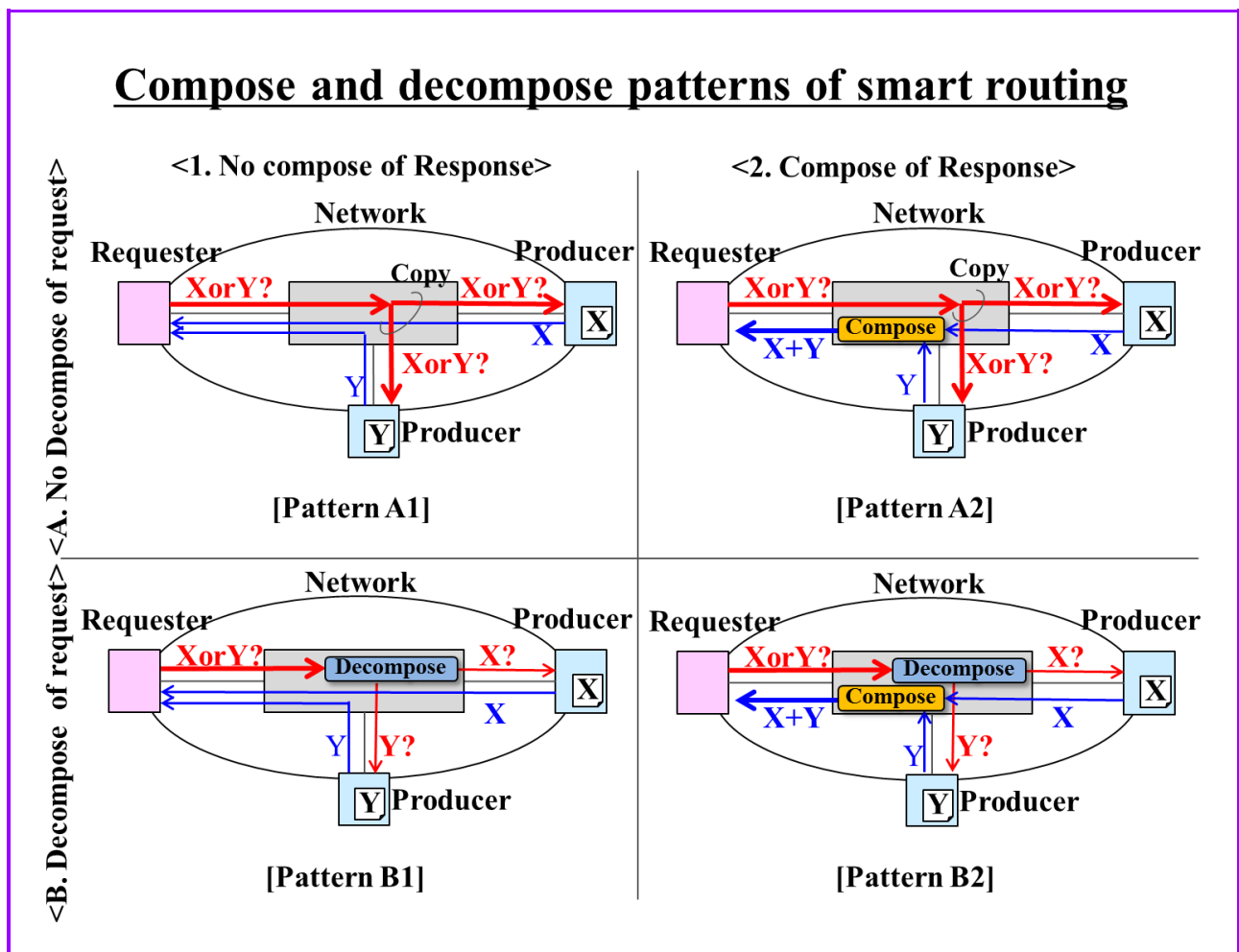


Figure 9.2-2 – Compose /decompose patterns of the smart routing

Abbreviations and Acronyms

- ICN(Information Centric Network)
- IoT(Internet of Things)

References

- [9.2-1] 5G!Pagoda: <https://5g-pagoda.aalto.fi/>

9.3 Scenario 3: LTE in a slice

The softwarization of the 5G wireline network must of course permit multiple simultaneous RATs with multiple simultaneous COREs and multiple simultaneous MEC functions. The 5G wireline must provide the required isolation / connectivity and computational resources to guarantee correct QOS/QOE to each of these 5G slices.

While slicing is intended to support the very different air interface requirements of different use cases in 5G, there are a number of practical considerations with respect to migration from 4G to 4.5G to 5G and also the requirement to support 4.xG flavours long after the deployment of 5G.

We expect therefore that a single 5G wireline infrastructure will be adopted well before 5G RATs/COREs are actually deployed and that there will be initially one or two LTE slices running in this 5G wireline infrastructure prior to additional 5G slices being enabled; The alternative which would require parallel 4G and 5G sets of antennas, fronthaul, CRAN, DRAN, DC, backhaul and IDC interconnect resources is a vastly more expensive and therefore undesirable option. While a common infrastructure will save CAPEX we also need to save OPEX during any migration. The infrastructure's orchestration software should therefore permit resources to slowly be moved from the LTE slices to the 5G slices and then of course between the 5G slices as required. If we break down the infrastructure network into its components (Figure 9.3-1) we can begin to analyze each for their respective softwarization requirements implied by simultaneous support of both 4G and 5G.

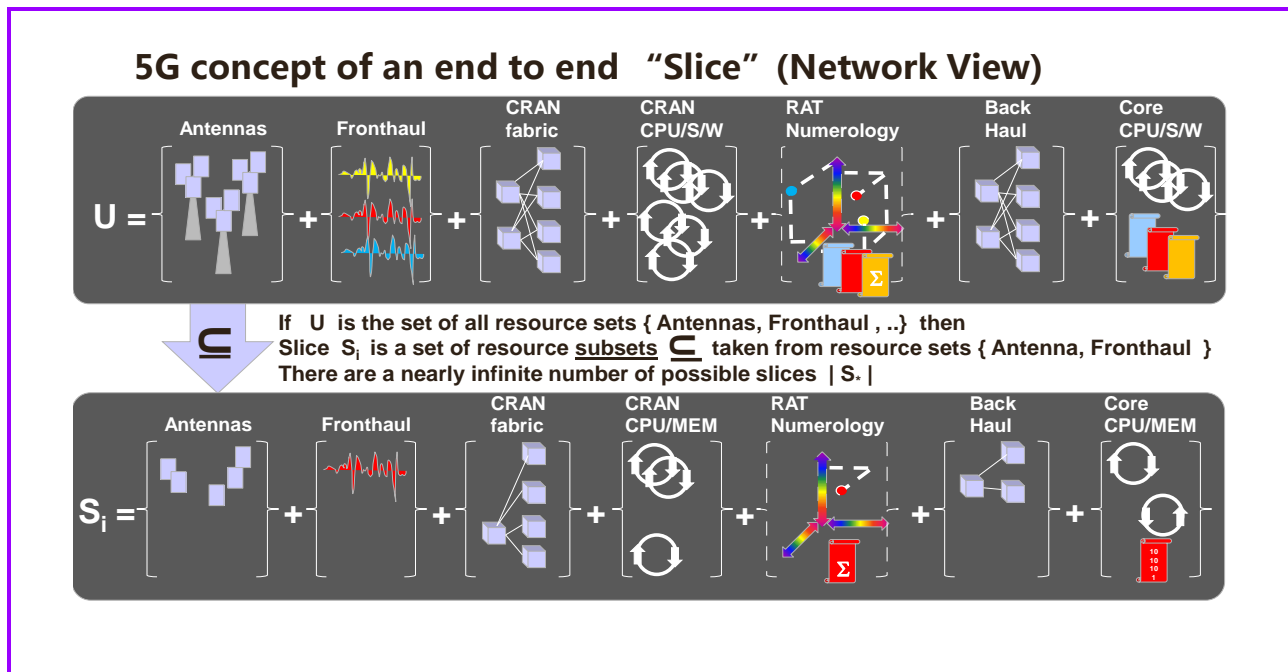


Figure 9.3-1 – End to end view of a slice through the 5G wireline infrastructure.

Antennas – ideally the antenna should support 4G and 5G. This means that software must be able to query the capability of the antenna. If the antenna is MIMO then the question of if some segments can be used for 4G while others are used for 5G would lead to additional requirements on capability query. If an antenna can be physically aimed or aligned then software would need to be able to control this hardware especially in the case aim is different from 4G to 5G. Minimally the control system would need to know the position of the antenna in some convenient R^3 and also its current aim etc.

Fronthaul – if the fronthaul is not switchable there is not much for software to control. Some form of testing and software reporting would be required to know if the fronthaul used for 4G can be re-used for 5G. If the fronthaul is switchable then software must be able to connect a previously 4G antenna to its 5G RAT. If 5G adopts a packet fronthaul then support for digitized analog over that packet fronthaul would be required to support 4G and control of any intermediate switches for QOS guarantees to minimize jitter (buffering/retiming) requirements would also be required.

C-RAN – The CRAN architecture would need to be able to run both dedicated 4G hardware together with 5G hardware which is capable of running any RAT including 4G (this is currently possible with 3G and 4G so we can expect similar options to be available for 4G and 5G). To effect the migration the older hardware in the CRAN would need to be phased out by connecting it to 4/5G RAT capable hardware. Ideally this would involve a software configurable fronthaul switch of some kind. In the case where the C-RAN is running CORE/MEC functions, the hardware must support the virtualization that is currently being used for 4G (likely NFV and vEPC in a VM). Ideally hardware used for the RATs can also be repurposed for CORE/MEC depending on load requirements. Software in the CRAN would need to make the special purpose hardware look like general purpose VM/Container capable devices/OS to allow any third party hardware/OS to populate this special purpose hardware in the C-RAN without requiring the deployment of additional general purpose hardware (which cannot easily support RAT at large scales).

Backhaul/IDC – QOS from RAT/Core server/MEC server must be configurable and likewise IPVPN's (both V4 and V6) must be supported so that different LTE flavours can run overlapping GTP tunnels/addressing while 5G can run whatever new overlapping tunnels (or not) it requires. Likely this means simultaneous existence of IPVPN's and SDN. It is likely that a few 10's of IPVPNs would be required to span the infrastructure which is not particularly stressful to IPVPN technology (via distributed or central control) although the potential number of IPV4/V6 addresses and FIB table sizes is of concern if the UE addressing gets exposed throughout the infrastructure.

DC – The DC hardware must support the virtualization that is currently being used for 4G (likely NFV and vEPC in a VM) together with whatever is chosen for 5G (Containers etc.). Since the DC already runs LTE COREs (and MVNO COREs) the software changes required would be mostly to support 5G than to support 4G.

MEC – Where the C-RAN is running MEC functions, the hardware must support the virtualization that is currently being used for 4G (likely NFV and vEPC in a VM). Ideally hardware used for RAT can also be repurposed for MEC depending on load requirements. This means it must support both bare metal (for 4G/5G RAT) as well as VM and Container technology for MEC. Software in the CRANs would need to make the special purpose hardware appear like general purpose VM/Container capable devices/OS and should not require special purpose libraries / compilers or OS's.

Control Hierarchy/Orchestration – Since the 5G control/orchestration is unlikely to be complete or fully functional or fully trusted while 4G is being moved to a 5G type infrastructure it will be necessary for a manual OSS approach to operations to co-exist with the early stages of a 5G wireline orchestration system. This means that the orchestration system will need to learn what the current state of the infrastructure is without impacting it in any manner. The orchestration and control system would need initially to query the state of all the components, determine what slices they are assigned to and then at a minimum be able to display the status of the slices prior to being given any control over the attributes of the slices themselves. In order to learn (or resynchronize a controller) the concept of a cookie or some form of tags that can be attached to resources and which can be queried by any OSS or orchestration system would be required. At some point the orchestration system would need to bring up 5G slices and at that point resources would need to be taken from the 4G components (antennas through to core processing elements). Initially this would be in the form of a test done during low traffic times and in low traffic vicinities after which resources would be returned to 4G, analysis would be performed on the results of the tests and new tests scheduled. Eventually the 5G network would be turned up permanently and the resources taken from 4G and not returned. The more quickly and automatically this iterative testing could be done the more quickly and inexpensively 5G could be deployed.

It is also possible that resources could be moved by an active 5G orchestration system between various 4G flavours , for example to/from NB-LTE and LTE (prior to any 5G deployments).

9.4 Scenario 4: Network Slicing

In IMT-2020 networks, it is necessary to consider end-to-end application quality and enablement through network softwarization platform. Therefore, the current SDN and NFV technology should be utilized to transform the infrastructure to realize end-to-end slice management and orchestration. Especially network slice can be deployed and managed across the E2E network including Fronthaul/Backhaul Network, IP Transport Network, Access DC, Regional DC and Core DC.

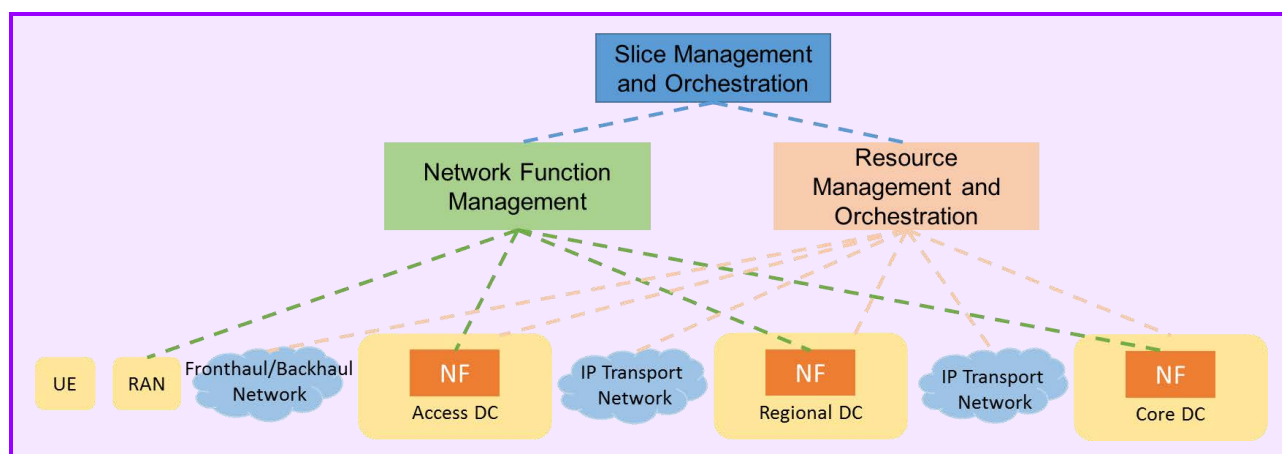


Figure 9.4-1 – Framework of network slice management and orchestration

Based on the existing network management (Network Function Management) and resource management (Resource Management and Orchestration) in NFV/SDN, slice Management and Orchestration will be introduced in network slicing. There are three scenarios that Slice Management and Orchestration should be considered.

Network function management is responsible for FCAPS management of IMT-2020 functions, which can include OSS, EMS and VNFM.

Resource management and orchestration is responsible for global and multi-domain resource management and orchestration. Multi domains include Fronthaul/Backhaul network domain, IP network domain, and data centers domain. It can combine multi-domain (i.e. Fronthaul/Backhaul network, IP network and DCs) resources to meet the resource requirement of an E2E network slice instance, and assign necessary resource to the slice instance, global resource monitoring, and global resource management and coordination. Management and orchestration of Fronthaul/Backhaul network, IP network resources can be achieved by SDN technology (i.e., introducing SDN controllers). And management and orchestration of resources in DCs can be realized by NFV technology (introducing NFV MANO, VIM, VNFM). Resource management and orchestration can communicate with orchestrators (e.g., NFVO) of access DC, regional DC and core DC to initiate or deploy different network functions in different DCs according to the service requirement of a specific network slice instance, and communicate with controllers of other domains (IP and transport domain) to meet the instantiation and SLA requirement of the slice instance.

Prioritization mechanism on multiple slices can be supported by slice management and orchestration based on the SLA of the requested slice and level of customers, e.g., the prioritization of deploying/scaling the slice especially when the resources is not enough.

Scenario 1: Network slice design

Slice Management and Orchestration can support to select network functions and network topology to design one network slice template with different network slice functions to cater for different types of scenarios, e.g. eMBB/MTC/CC. Basic slice template with basic functions can be identified.

After that, some detail parameters such as performance (e.g., bandwidth/latency), capacity, preferred deployment location of NF, life cycle management policy and reliability policy can be configured based on basic template. Different kind of complete templates can be created based on different network slice configurations, e.g. different eMBB network slice instances with various latency and numbers of subscribers.

A Network slice template is composed of the following information elements:

- **Topology:** It describes a set of network functions consisting of a network slice, network connections between these network functions and these functions' resource requirements (e.g., computing resource, storage resource and network resource).
- **Configuration:** It includes the configuration information of resources and functions which compose a network slice instance.
 - **Resources configuration:** It describes the configuration information of computing resources, storage resources and network resources of a specific network slice instance, e.g. IP connections between network functions, service-level and resource-level lifecycle management rules.
 - **Functions configuration:** It describes some function-specific configurations that consisting of a network slice, e.g., IP address pool for UEs.
- **Work Flows:** It describes the process of instantiating the network functions in a network slice, e.g., the deployment sequence of network functions or dependence between them.

NOTE 1 – [Ref.9.4-1] can be a candidate implementation on configuration.

NOTE 2 – Slice template model can be recursive, as described in Section 8.2. It can be used to support flexible and efficient slice template management.

Scenario 2: Network slice deployment

Slice Management and Orchestration can select the generated template, instantiate network function and establish network connection according to the requirement of network slice complete template. Based on the template, resource states, and operator policy, it can also allocate and reserve appropriate resources for the slice.

Scenario 3: Network slice monitor

Slice Management and Orchestration can monitor the operating states of one network slice such as number of connected subscribers and service traffic of the whole network slice.

References

[9.4-1] GENI configuration model (RSPEC), <http://groups.geni.net/geni/wiki/GENIExperimenter/RSpecs>.

9.5 Scenario 5: Satellite integration in the 5G Ecosystem

The application of NFV and “Cloud RAN” aspects to the satellite component paves the way towards the full virtualisation of satellite head-ends, gateways/hubs and even satellite terminals, thus entirely transforming the satellite infrastructure, enabling novel services and optimising resource usage. In this context, several enhancements/adaptations of current SDN/NFV technologies (e.g., extensions of the OpenFlow protocol) are envisaged, in order to be fully applicable to the satellite component domain and exploit satellite-specific capabilities.

Satellite network architecture can be designed to enable virtualisation and SDN-based control of the network components supporting advanced service delivery, including through hybrid satellite / terrestrial infrastructure.

NFV will facilitate the management and deployment of virtualised functions of the satellite network, and SDN-based control can be achieved through programmable interfaces for satellite resources.

NFV / SDN integration in satellite networks will allow for the smoother integration of satellite infrastructure into the 5G ecosystem, the provision of innovative services, and ease the evolution of services where deployed.

Specific application examples benefiting from SDN / NFV include:

Scenario 1: flexible satellite bandwidth on demand through enhanced customisation and flexibility in the provision of satellite network services.

Scenario 2: satellite backhauling of terrestrial network, enhancing management of satellite backhaul capacity, and extending multi-operator sharing.

Scenario 3: Satellite/ terrestrial hybrid network services facilitating flexible traffic control and content distribution between satellite and terrestrial access networks.

Scenario 4: Flexible programmable satellite payloads.

Examples of ongoing projects include:

Satellite CDN Overlay project: SES and Rutgers University's WINLAB are building a proof-of-concept demonstrator of a content delivery overlay network designed to alleviate congestion in terrestrial networks (wireless/telco/cable, etc.) by combining satellite multicasting with intelligent traffic routing, edge caching, and data-driven cache management (popularity metrics) to optimize overall network performance and cost effectiveness. The demonstrator is built up from large scale wireless and terrestrial networking test beds (ORBIT and GENI) used to support research in these areas. Actual satellite capacity and Very Small Aperture Terminals (VSAT) will be integrated to create a hybrid networking infrastructure upon which new investigations and demonstrations of network optimization and content delivery can be performed. The demonstrator is under construction and will become operational in Q1 2017.

Appendix I

LTE in Slice for Softwarization

I.1 Overview

As part of the terms of reference for the focus group we are looking at various open source collaboration efforts to help understand and advance the 5G wireline work being done by the focus group. This is a request for a short time slot in Beijing to show a little bit of progress with the OpenAirInterface open source collaboration. This contribution outlines the main points of what will be described in this timeslot.

As part of our proof of concept/open source work we are hoping to use OpenAirInterface together with an open orchestration system to show some of the concepts of end to end orchestration of a slice where OpenAirInterface and off the shelf smart phones are examples of a real RAT within a slice.

Obviously we would like to use 5G RAT's and UE's/Cores (but they don't exist yet) however it is reasonable to expect that any architecture for 5G wireline must be able to support any RAT/CORE/MEC including 4.*G LTE flavours so this is a useful starting point. We have tried two different open 4G implementations so far (OpenLTE and OpenAirInterface) and the latter is much better suited and more complete.

I.2 Discussion

Using openAirInterface.org software combined with a software defined radio board (ETTUS B200 accepts I/Q over USB-3) we have been able to bring up a complete end to end smartphone -> eNB -> vEPC -> Internet connection including web browsing, software downloads and other normal IP functions end to end. We have so far had most success with an unlocked brand new Huawei Mate 7 phone. We have also had some success with an ACER Z630 (still debugging EPC compatibility issues). We had no luck with a used unlocked Samsung Galaxy S4 due to low level operator locks (Boot Rom) or various Apple products which seem not to want to connect to any test networks for unknown reasons. The OpenAirInterface wiki lists other devices they have had success with.

The OpenAirInterface eNB and vEPC are currently run on an i5 quad core with Ubuntu server 14.04 and a Linux 3.19 low latency kernel and with a USB-3 fronthaul interface to the ETTUS B200. Splitting up the vEPC is supported (but not yet tried) and an Ethernet fronthaul is apparently possible (but not tried yet). We expect to try both of these shortly with the goal of being able to script or orchestrate where each piece runs and how it interconnects in our lab DC / transport network environment.

We have found the OpenAirInterface software quite impressive but it is non-trivial to get everything working, mostly because the phones themselves have many different permutations and combinations of options all of which together with custom SIM cards have to be configured perfectly for everything to work. Debugging, especially the phones is challenging and Android seems to offer the best success so far although other teams indicate success with IOS based devices.

The goal of this work is to use LTE as an example within a slice and now that we are able to use OpenAirInterface and various smart phones the next step is to work on multiple instances in different slices and to explore the requirements of end to end orchestration.

I.3 Components

Following are links to the software and hardware that we have found worked for us. Note that OpenAirInterface has a wiki and mailing lists etc. for support and much more detail. Below we describe what we found worked for us. We also got timely support from other teams using this software via the mailing lists.

The trickiest components are the smart phones and it is recommended to get unlocked brand new phones that have never seen a SIM card or been used for LTE previously on other networks. Introduction of carrier SIMs can change the configurations permanently by limiting operating bands, accepted network IDs by geographic region etc. These problems can make future testing with these phones extremely difficult and frustrating.

You can program the SIMs yourself however we just bought 10 pre-programmed cards from www.sismocom.de. The programming options are shown below:

Here are links to the relevant components in our lab:

RADIO: <https://www.ettus.com/product/details/UB200-KIT>

S/W: <http://www.openairinterface.org/>

SUPPORT: <https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirSoftwareSupport>

PHONE1: <http://consumer.huawei.com/minisite/worldwide/Ascend-Mate7/>

PHONE2: <http://www.acer.com/ac/en/IN/content/model/HM.HT6SI.001>

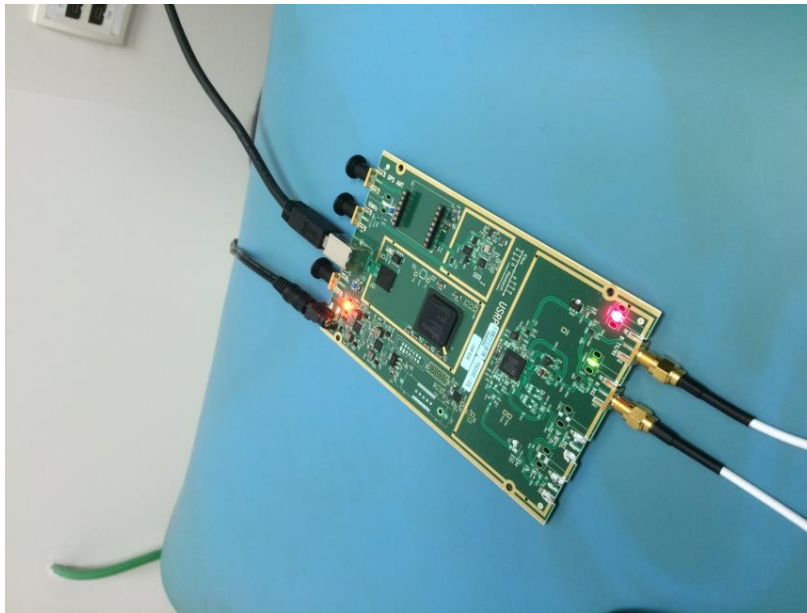
SIMs: <http://www.sismocom.de/>

I.4 SIM card programming

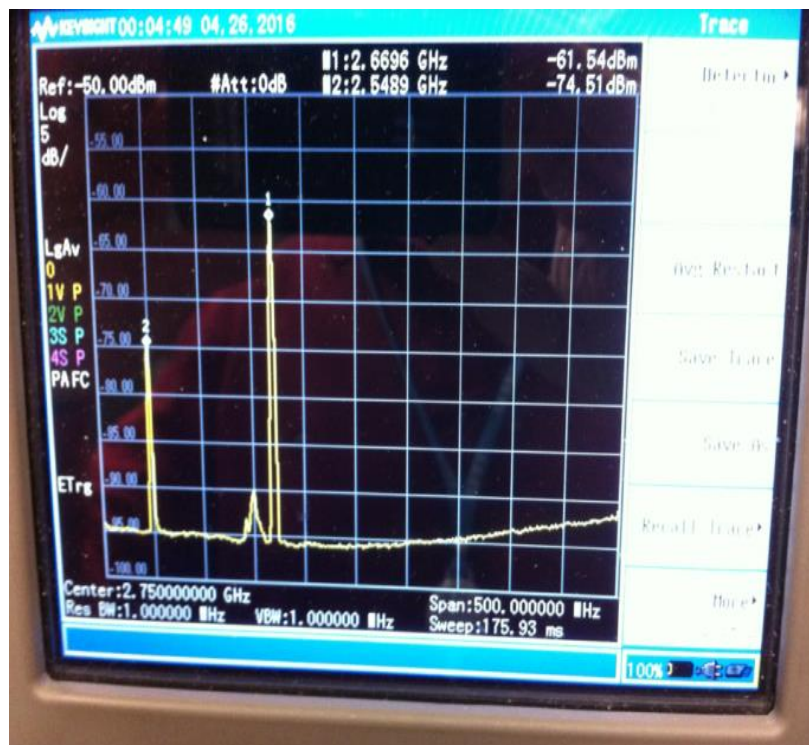
This is how we programmed our 10 SIM cards as purchased from [sismocom.com](http://www.sismocom.com). However we recommend getting a programmer and doing it yourself since you may need to vary the network id to work past restrictions with a phone's ROM/region locks.

IMSI	KI	OPc
101020000000001	ABCD1234ABCD1234ABCD1234ABCD1231	11111111111111111111111111111111
101020000000002	ABCD1234ABCD1234ABCD1234ABCD1232	11111111111111111111111111111111
.....		
.....		
101020000000009	ABCD1234ABCD1234ABCD1234ABCD1239	11111111111111111111111111111111
11	ABCD1234ABCD1234ABCD1234ABCD123A	11111111111111111111111111111111

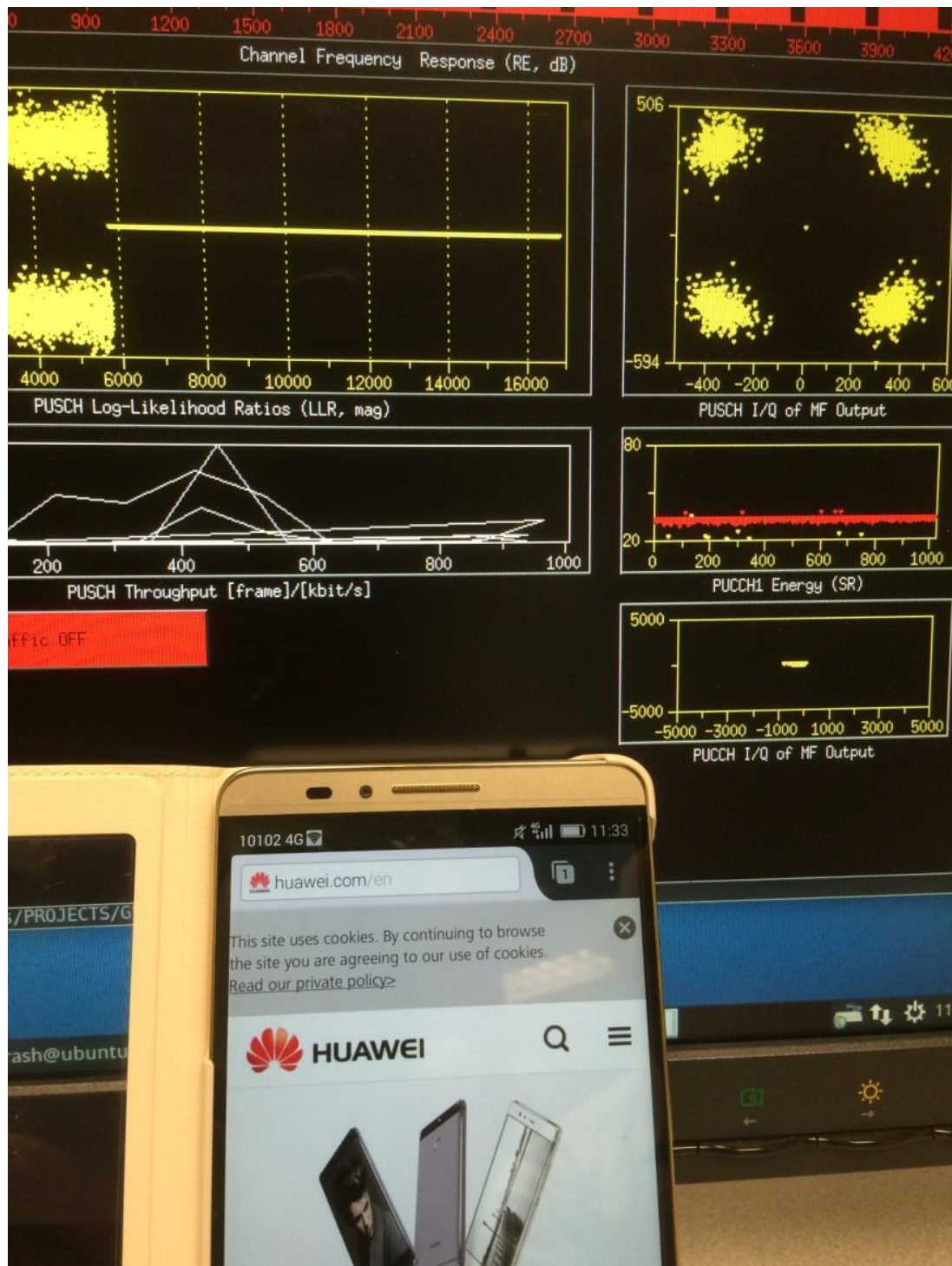
I.5 Lab pictures



Ettus B200 SDR board (USB3 carries simple I/Q) – you need USB3 on server. Other SDR models may work, we have not tried them. We found you need to separate the TX/RX antennas by a few meters with coax extenders for best performance.



Scope – not really required but since you want to ensure low power and proper band it's helpful. Here is the 2.5GHz uplink and 2.6GHz downlink shown on band 7 with 120Mhz separation. Band 7 block was graciously loaned to us by Canadian operator Telus. This is licensed spectrum so either you need a Faraday cage, permission or a test licence.



Huawei Mate 7 - shamelessly browsing Huawei.com but other pages work as do APP downloads, APPs and normal IP behaviour including tethering. (no voice/volte supported this time). Note the network ID "10102 4G" shown on the phone. Acer Z630 not shown but we expect it to work with a very minor configuration patch.

Appendix II

FlexE

II.1 IP/Ethernet based Mobile Backhaul

Mobile backhaul networks these days are based on, or in the process to migrate to IP and MPLS based technology, where MPLS LSP or Ethernet circuit are usually configured per user, site or/and service with specific SLA associated, as illustrated in Figure Appendix-FlexE-1. MPLS LSP or Ethernet circuit uses software based resource partition lacking assured service to users.

FlexE based end-to-end connections (sometimes referred to as channels) could play a big role here to replace MPLS LSP and Ethernet circuit with the following benefits:

- Hardware based resource partition to assure service with dedicated performance and QoS, low and predicted latency, and guaranteed bandwidth per SLA.
- Dedicated and individual channel guarantee strict traffic isolation for data security and privacy.
- Data channel can be configured based on demand, and the associated bandwidth can be resized (refer to Section 6.1.7.4) with much agility and programmability.

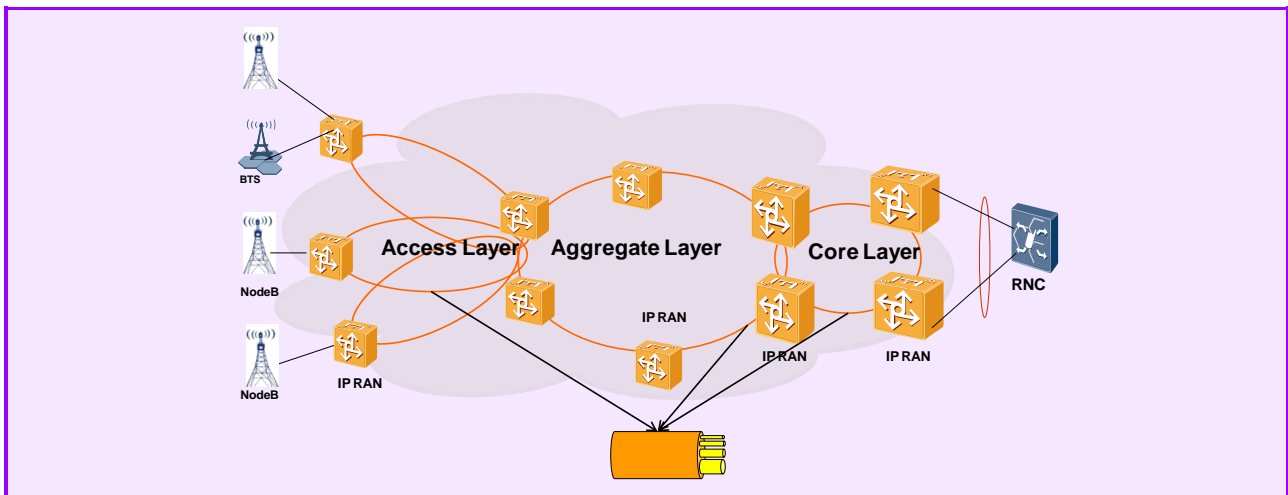


Figure II-1 – IP/Ethernet based Mobile Backhaul

II.2 OAM Functions for FlexE

OAM (operations, administration and management) is a mandatory component when deploy FlexE technology in networks. FlexE connections are essentially logical data pipes on the top of Ethernet, and so OAM functions and mechanisms defined for Ethernet ([Ref. Appendix II-1]) must be implemented and deployed in FlexE-enabled networks.

A FlexE-enabled network must support the following performance monitoring functions on their Ethernet interfaces according to Section 8 of ITU-T G.8013/Y.1731 [Ref. Appendix II-1]:

- Frame loss ratio
- Frame delay
- Frame delay variation
- Throughput

A FlexE-enabled network must support the following performance measurements on their Ethernet interfaces according to Section 8 of ITU-T G.8013/Y.1731 [Ref. Appendix II-1]:

- Frame loss measurement.
- Frame delay measurement.
- Frame delay throughput measurement.

A FlexE-enabled network must support the following fault management according to Section 7 of ITU-T G.8013/Y.1731 [Ref. Appendix II-1]:

- Ethernet continuity check
- Ethernet loopback
- Ethernet link trace
- Ethernet alarm indication signal
- Ethernet remote defect indication
- Ethernet locked signal
- Ethernet test signal

Because FlexE connections are contained logically within Ethernet PHYs, the Ethernet OAM functions can be used to assist with SLA assurance, capacity planning, performance monitoring, and diagnostic analysis for FlexE connections as well. In addition, some Ethernet OAM events (e.g., alarm signals) must be propagated to the relevant FlexE shim layer so proper actions can be taken.

Additional OAM functions are required in FlexE-enabled networks, as elaborated in the following sections.

II.2.1 FlexE Neighbor Discovery

There requires an automatic discovery mechanism between two devices (routers or switches) that are interconnected. The information that needs to be made known to the two interconnected devices for FlexE to operate include FlexE group ID, PHY number, capability of shim entity, mux and demux functions, etc. Manual configuration is recommended in as a general approach. For better scaling in operation and management, an automatic discovery and connectivity verification mechanism is more promising in proposals including those in [Ref. Appendix II-2] and [Ref. Appendix II-3].

II.2.2 FlexE End-to-End Connection Connectivity Verification

For an end-to-end FlexE Connection, there requires an OAM tool for fault detection and diagnostic mechanisms that can be used for end-to-end fault detection and diagnostics. Using this tool, the two ends of a FlexE connection can exchange ping-alike messages to assure the connection is operational with healthy status and integrity. Note this communication must be in-band with the FlexE connection.

II.3 FlexE-3 Resizing of FlexE Connection

A FlexE connection (or channel for that matter) is associated with a fixed MAC rate. Due to dynamic nature of business environment, user experiences, evolvement of applications, etc., the bandwidth of a FlexE client actually in use may vary, sometimes is smaller but other time require bigger than the MAC rate of the relevant FlexE connection. For the former case, a separate FlexE connection with smaller MAC rate is desirable so that the unused bandwidth out of the associated Ethernet PHYs can be assigned to other applications and users. For the latter case, a separate FlexE connection with larger MAC rate is required. The operation that moves FlexE client from an existing FlexE connection to another that has either smaller or larger MAC rate is called resizing of FlexE connection. Figure Appendix-FlexE-2 illustrates an example where a FlexE connection of 50G MAC rate can be either resized to another one of 75G MAC rate (upsizing) or 25G MAC rate (downsizing).

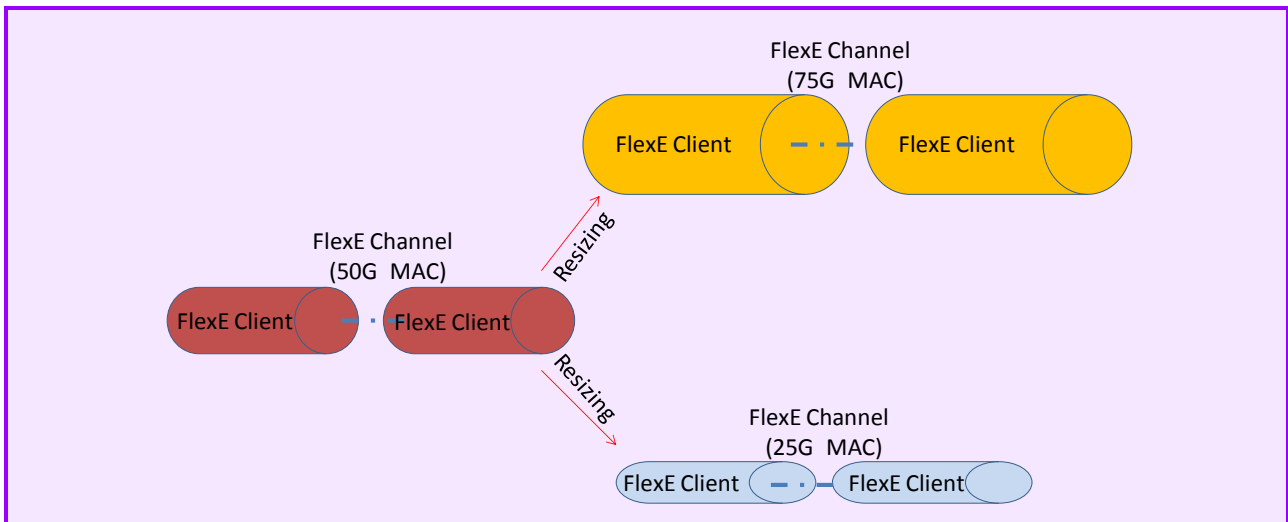


Figure II-2 – Resizing FlexE Connection

II.3.1 Hitless Resizing Operation

A mandatory requirement when performing resizing operation on a FlexE connection is that there **MUST** not be any degradation on the FlexE client's live traffic, including packet loss, orders out of sequence, unexpected transmit latency, etc., and a non-disruptive operation is commonly known as a *hitless* operation, and in this context, we say the resizing of FlexE connection **MUST** be a hitless resizing.

Resizing a FlexE connection can be accomplished by switching between the two FlexE calendar configurations "A" and "B" (refer to Section 6.1.7.4). This operation must be done hop-by-hop between two directly connected FlexE-capable devices (routers or switches) along an existing end-to-end FlexE connection.

As an example illustrated in Figure Appendix-FlexE-3, there are two FlexE connections with bandwidth size as 10G and 25G, respectively, between router R1 and router R2 interconnected by a group of 100G Ethernet PHYs (not shown), and the associated FlexE calendar configuration is "A" per provisioning. At a later time, the 10G connection and 25G connections need to be resized to 35G (upsizing) and 20G (downsizing), respectively, and the FlexE calendar configuration "B" is provisioned accordingly. Through action in control plane or management plane, a *switch* operation takes place, such that the client traffic originally carried on the 10G and 25G connections are moved to the 35G and 20G connections, respectively.

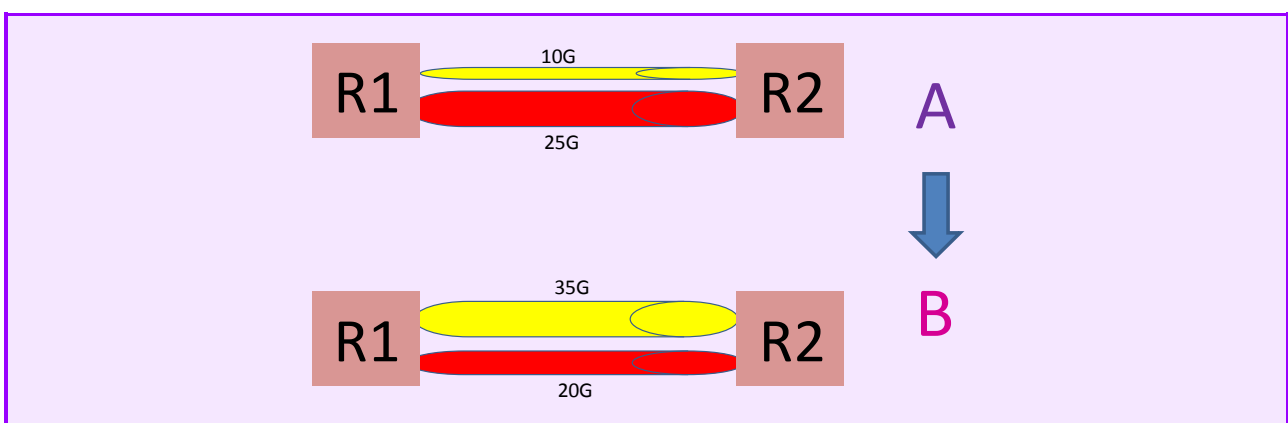


Figure II-3 – Resizing two FlexE Connections between two Routers

Via *switch* operation between FlexE calendar configuration "A" and "B" to resize an end-to-end FlexE connection, calendar configuration and the *switch* operation must be accomplished on every hop in a synchronized manner along the connection. Note that the routing path associated with the devices (routers or switches) for the client traffic remains the same, before and after the *switch* operation.

References

- [Appendix II-1] ITU-T G.8013/Y.1731, “OAM functions and mechanisms for Ethernet based networks”, ITU-T, 2013.
- [Appendix II-2] Oif2016.211.02, “Preview of Project Start Proposal for FlexE Neighbor Discovery”, OIF, 2016-10-26.
- [Appendix II-3] Oif2016.247.00, “FlexE Connectivity Verification”, OIF, 2016

Contributors (in Alphabetical Order)

This is the list of all contributors who submitted any written form of comments or contributions.

- Akihiro Nakao, The University of Tokyo
- Alain Mourad, InterDigital Communications
- Alex Galis, University College London
- Andrea Di Giglio, Telecom Italia
- Antonio Manzalini, TIM
- Dean Cheng, Huawei
- Donna Bethea-Murphy, Inmarsat
- Edward Ehrlich, InterDigital Communications
- Fabio Cavaliere, Ericsson
- Fang Li, CAICT
- Ghani Abbas, Ericsson
- G.Q Wang, Futurewei Technologies
- Hiroaki Harai, NICT
- Hiroshi Ou, NTT
- Homare Murakami, NICT
- Hui Cai, China Mobile
- Hui Ding, CAICT
- Kenichi Fukuda, Fujitsu Laboratories
- Kentaro Ishizu, NICT
- Jaehyun Ahn, InterDigital Communications
- Jeongyun Kim, ETRI
- Jian Song, Huawei
- Jie Dong, Huawei
- Josep Mangues, CTTC
- Lijun Dong, Futurewei Technologies
- Ling Xu, Huawei
- Luca Cominardi, InterDigital Communications
- Luca Pesando, TIM
- Luis M. Contreras, Telefónica
- Mach Chen, Huawei

- Miquel Payaró, CTTC
- Mònica Navarro, CTTC
- Namseok Ko, ETRI
- Nikola Vucic, Huawei
- Nobuo Suzuki, ATR
- Nurit Sprecher, Nokia Networks
- Pekka Kuure, Nokia Networks
- Peter Ashwood-Smith, Huawei
- Ping Du, The University of Tokyo
- Qing Wei, Huawei
- Raul Muñoz, CTTC
- Raymond Knopp, EURECOM
- Roya Rezagah, ATR
- Stanislav Filin, NICT
- Stanley Russo, SES
- Sunhwan Lim, ETRI
- Takashi Nishitani, Mitsubishi Electric
- Takashi Shimizu, NTT
- Takeshi Kinoshita, NTT
- Tarik Taleb, Aalto University
- Thomas Deiss, Nokia Networks
- Toshiaki Suzuki, Hitachi
- Toshitaka Tsuda, Waseda University
- Yoshiaki Kiriha, NEC*
- Yoshiaki Kiriha, The University of Tokyo*
- Yoshihiro Nakahira, Oki Electric Industry
- Wei Chen, China Mobile
- Wei Lu, Nokia Networks
- Weixing Wang, Nokia Networks
- Xavier Costa Perez, NEC
- Xiaowen Sun, China Mobile
- Xiayu Li, CAICT
- Xinyuan Wang, Huawei
- Yachen Wang, China Mobile
- Zongpeng Du, Huawei

(*same person)

Acknowledgements

This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (R7117-16-0127, Development of Standard for Service Adaptive Dynamic Network Slicing)

This work was partially supported by the MIC (The Japanese Ministry of Internal Affairs and Communications) Project "Research and Development on Virtualized Network Technologies".

This work was partially supported by the MIC (The Japanese Ministry of Internal Affairs and Communications) Project "Research and Development on Secure Management Technologies for Autonomous Mobility System".

This work was partially supported by the European Commission through the H2020-ICT-2014-2 project Flex5Gware (Contract Number 671563), FANTASTIC-5G (Contract number 671660) and 5G-Crosshaul (Contract Number 671598), and the Spanish Ministry of Economy and Competitiveness (MINECO) through the projects DESTELLO (TEC2015-69256-R), AETHer (TEC2014-58341-C4-4-R), CellFive (TEC2014-60130-P), 5GNORM (TEC2014-60491-R), and ELISA(TEC2014-59255-C3-1-R).

This work includes results of the research conducted under R&Ds for Network System and ICT Testbed Operation by NICT, Japan.

This work includes results of the research that has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 671551 (5GXHaul).

This work was partially supported the EU H2020 5G PPP projects: 5GEX ("5G Multi-Domain Exchange"; <https://www.5gex.eu>) and SONATA ("Service Programing and Orchestration for Virtualized Software Networks"; <http://sonata-nfv.eu/>)

This work includes results of the research conducted under a contract of R&D for Expansion of Radio Wave Resources, organized by the Ministry of Internal Affairs and Communications, Japan.

This work was partially supported by the research contract of "Research and Development on control schemes for utilizations of multiple mobile communication networks", for the Ministry of Internal Affairs and Communications, Japan.

This work was partially supported by the EU-JAPAN initiative by the EC Frame-work Programme (Horizon2020/2014-2020) (5G core network) and Japan's Ministry of Internal Affairs and Communications (MIC) under Contract by the project, "Research and Development on 5G Core network", the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN, and by MIC's Project for Promotion of Advanced Communication Applications Development. This is partially the output of 5G! Pagoda project.



CAP



IMT-2020 Network Management Requirements

EX

Summary

This document describes network management requirements based on the implementation scenario of IMT-2020. IMT-2020 network management should involve a combination of existing and evolving systems, like LTE-Advanced, Wi-Fi and Fixed Network, coupled with new, revolutionary technologies designed to meet new requirements, such as low latency and massive connectivity. To meet those new requirements, the cost of deployment and operation of such system will increase massively. Network operators need to optimize CAPEX/OPEX by strategically interacting with multiple technology ecosystems especially for different radio access technologies. Therefore, network management in IMT-2020 should include both existing network management requirements as well as evolving network management requirements.

Table of Contents

1	Scope
2	References
3	Definitions
4	Abbreviations and acronyms
5	Conventions
6	IMT-2020 deployment scenarios
7	eNMS requirements for IMT-2020
7.1	High Level Network Management Requirements for IMT-2020 based on ITU-T
7.2	Functional Network Management Requirement for IMT-2020
7.3	Application and Service Management Requirements for IMT-2020
Appendix I – IMT-2020 Networks, Services and Resources Orchestration Functional Requirements	
I	Functionality Applicable to all Networks, Services and Resources Orchestration
I.1	System Primitives
I.2	Service Information Enablers
II	Service Development Functionality
II.1	SDK Primitives
II.2	SDK Tools
III	Service Platform Functionality
III.1	Service Platform Primitives
III.2	Service Platform Tools
IV	Service Orchestration Functionality
IV.1	Service Orchestration Primitives
IV.2	Resource Orchestration Primitives
Contributors (in Alphabetical Order)	



1 Scope

This Recommendation introduces IMT-2020 deployment scenarios and eNMS requirements for IMT-2020.

2 References

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. All users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- [1] ITU-T Recommendation M.3010 (2000), *Principles for a telecommunications management network*.

3 Definitions

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

For the purpose of this Recommendation, the following abbreviations are used:

API	Application Programming Interface
BSS	Business Support Systems
BW	Bandwidth
DDoS	Distributed Denial of Service
eMBB	Enhanced Mobile Broadband
eNMS	Enhanced Network Management System
EPC	Evolved Packet Core
HSS	Home Subscriber Server
ISP	Internet Service Provider
IWF	Interworking Functions
M&C	Management Control Protocol
mMTC	Massive Machine Type Communications
MVNO	Mobile Virtual Network Operator
NE	Network Equipment
NFV	Network Function Virtualization
NFVI	Network Functions Virtualization Infrastructure
NFVI-PoPs	Network Functions Virtualization Infrastructure Point of Presence
NFVO	Network Function Virtualization Orchestrator
NS	Network Service
OAM	Operations and Management
OPNFV	Open Platform for NFV Project
OSS	Operations Support Systems

QoS	Quality of Service
RAT	Radio Access Technology
SC	Service Components
SDK	Software Development Kit
SDN	Software Defined Network
SGW	Serving
SLA	Service Level Agreement
SP	Service Provider
TMN	Telecommunications Management Network
UE	User Equipment
USIM	Universal Subscriber Identity Module
vEPC	Virtual Evolved Packet Core
VIM	Virtualized Infrastructure Manager
VNF	Virtualized Network Function
VM	VNF Manager

5 Conventions

None.

6 IMT-2020 deployment scenarios

As shown in the figure below, a mobile operator can deploy IMT-2020 with two different options. At first, a mobile operator can deploy an independent IMT-2020 system separate from the existing mobile network. Secondly, a mobile operator can deploy enhanced network management system (eNMS) covering existing mobile network and IMT-2020 network supporting different networks (3G, 4G and IMT-2020) as shown on the right side of the figure below.

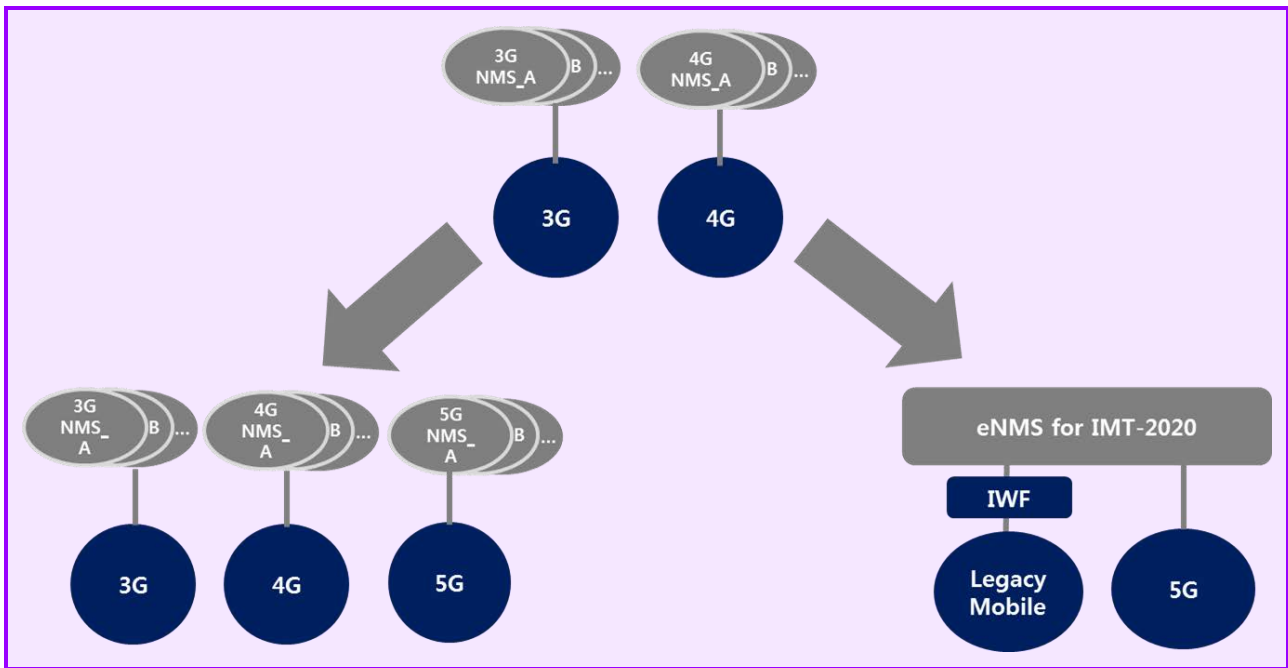


Figure 8 – A simple scenario of mobile operator's network migration towards IMT-2020

On the other hand, many operators provide both fixed and mobile services. For such operators, it is essential to introduce eNMS for IMT-2020 accommodating both fixed and mobile network as shown in the figure below.

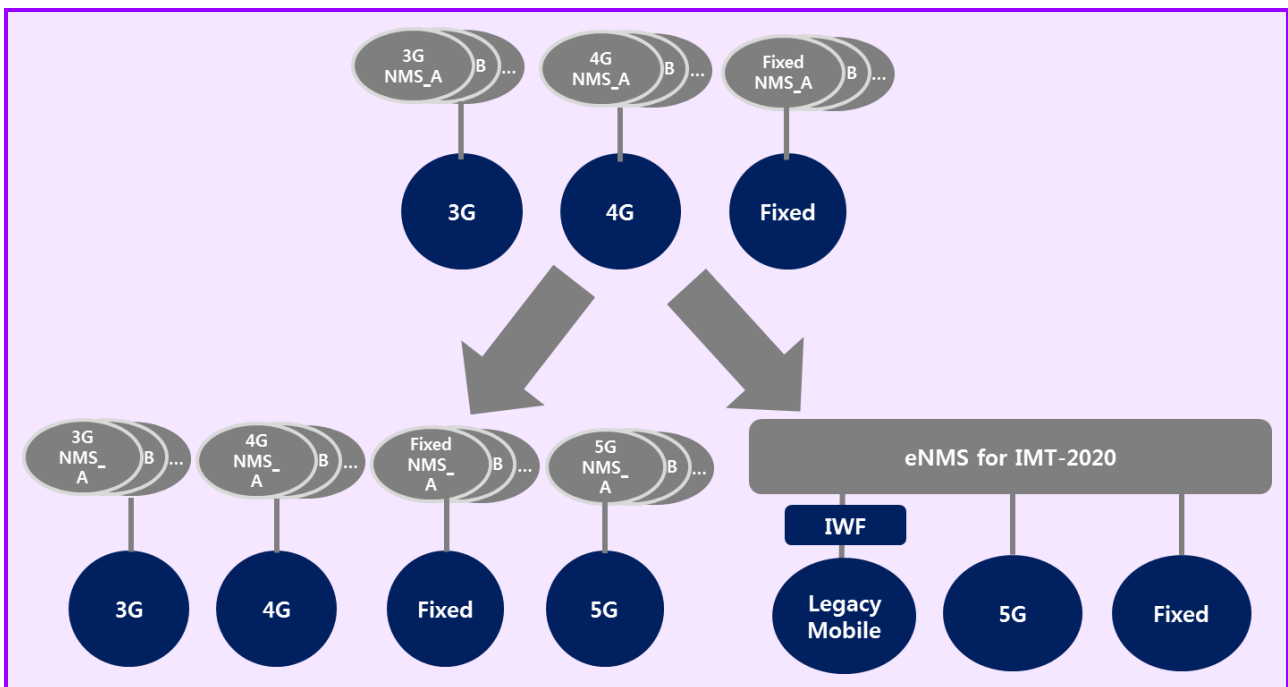


Figure 9 – A simple scenario of both fixed and mobile operator's network migration towards IMT-2020

There are many other network migration options available for operators depending on their focusing network service. In this document, IMT-2020 network management system includes developing a eNMS covering both fixed and mobile network to provide a seamless service regardless of network technologies.

As identified in Phase 1 of FG IMT-2020, standardized protocol is recommended to host all components of the network including different RATS and the fixed network. Operators can have common management viewpoint with eNMS regardless of network technologies as shown in the figure below.

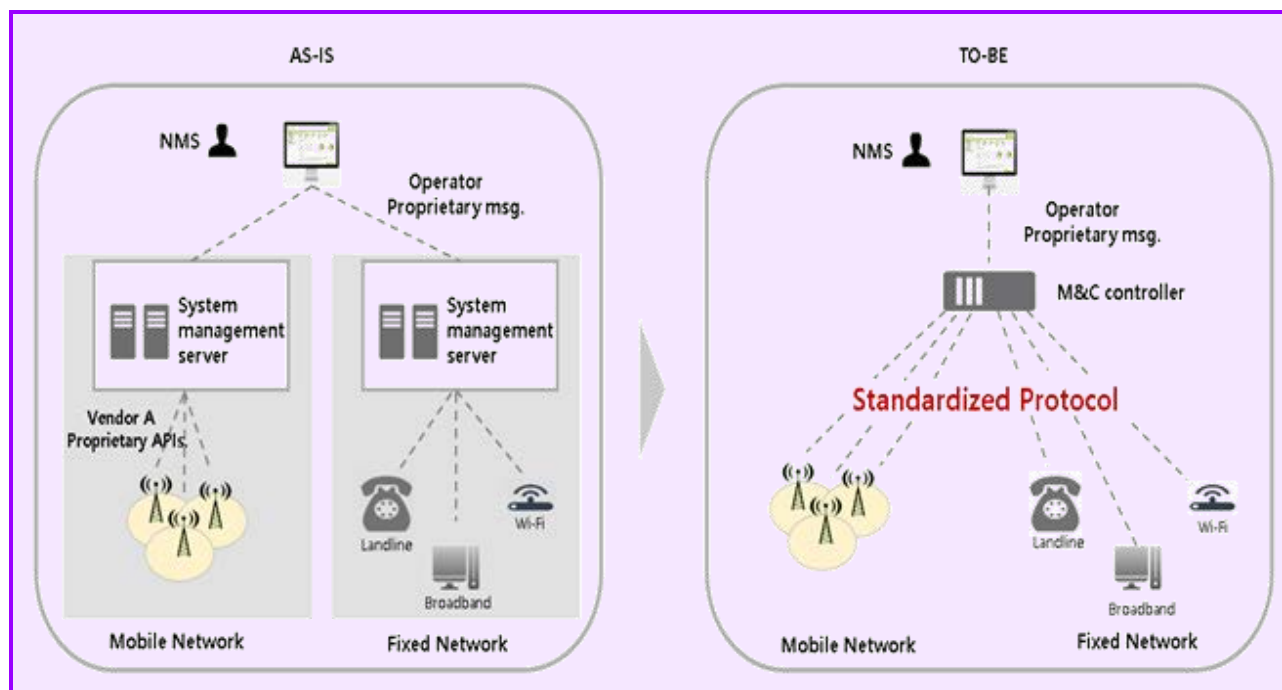


Figure 3 – Standardized network management system

By developing standardized protocol, network operator is fully aware of all network equipment status such that it can optimize network functions without management signalling burden. In order to develop such protocol, requirements for IMT-2020 network management system are discussed in detail.

7 eNMS requirements for IMT-2020

The objective for CMS is to provide converged network management systems for IMT-2020. By introducing the concept of generic network models for management, it is possible to perform general management of diverse equipment, network and services using generic information models and standard interfaces.

eNMS is intended to support a wide variety of management areas which cover the planning installation, operations, administration, maintenance and provisioning of telecommunications networks and services.

ITU-T Recommendation M.3010 describes the scope of management through the following two main concepts: Telecommunication Managed Areas and TMN management Services. In addition, ITU-T categorized management info into five board management functional areas. These functional areas support the management scope described by M.3010. They provide a framework through which the appropriate Management Services support the public telecommunication operator's business processes. Five management functional areas identified to date are as follows:

- Performance management;
- Fault management;
- Configuration management;
- Accounting management;
- Security management;

In the following sections, eNMS requirements are described regarding existing requirements (ex., ITU-T M.3010) and requirements of IMT-2020 from network perspective.

7.1 High Level Network Management Requirements for IMT-2020 based on ITU-T

Regarding to Recommendation principles for a telecommunications management network M.3010 [1], eNMS provides;

- To exchange management information across the boundary between the eNMS environment;
- To convert management information from one format to another so that management information flowing within the eNMS environment has a consistent nature;
- To transfer management information between locations within the eNMS environment
- To analyse and react appropriately to management information;
- To manipulate management information into a form is useful and/or meaningful to the management information user;
- To deliver management information to the management information user and to present it with the appropriate representation;
- To ensure secure access to management information by authorized management information user;
- To achieve technology independence based on requirements and to be extendable to include prominent and available management technologies in its implementations, as appropriate;

CMS for IMT-2020 means network management system unifying fixed and mobile network. eNMS for IMT-2020 provides:

- An architecture made of operations system and network elements and the interface between them.
 - To solve the traffic explosion and latency requirement of application, the enhanced mobility architecture is required
 - More flexible and optimized multi-RAT interworking architecture should be studied, which may also affect the design of new architecture.
 - Standard OAM protocols are required because parts of IMT networks such as fronthaul network aren't standardized yet.
- The methodology to define interface.
 - To make more flexible and resilient, upgrades of session or bearer management and network slice are included.
 - To improve the robustness of the network and throughput, multi-connectivity is required.
- Other architecture tools such as logical layered architecture that help to further refine and define the management architecture of a given management area.
 - To support a true fixed and mobile convergence ensuring a seamless user experience within the fixed and mobile domains, it is required to interwork with other radio access networks in fixed broadband access network.
- A number of generic and/or common management functions to be specialized/applied to various and specific ITU-T TMN interfaces.
 - Existing signaling procedure is heavy and not optimized for emerging new services, so the mobility management is required.
- To optimize softwarization in mobile network, functions in SDN solution are required.

7.2 Functional Network Management Requirement for IMT-2020

NOTE – For further study.

7.3 Application and Service Management Requirements for IMT-2020

In order to prepare IMT-2020 network management scheme, IMT-2020 services defined by the most major standard bodies need to be looked into. User equipment is included in this additional network management requirement for each service. Three major types of IMT-2020 achievement are defined as the following:

- ultra reliable and low latency communication
- enhanced mobile broadband
- massive machine type communication

For each key major type achievement, service management requirements are suggested.

7.3.1 Connected car service management requirement for IMT-2020

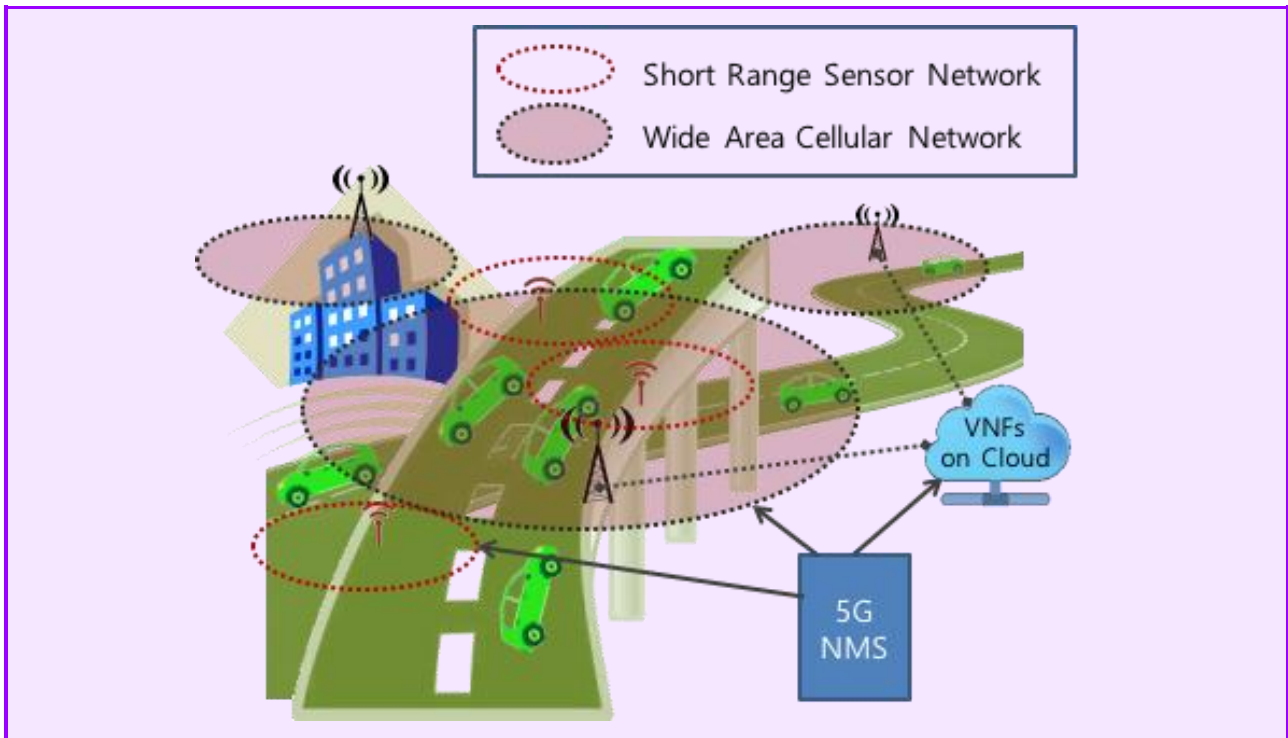


Figure 4 – A simplified connected car service useful for revealing network management requirements

In this service, the number of cars passing through a specific network domain at a specific time will vary widely. Instead of allocating enough networking resource all the time, the operators would want to manage their expensive resource in a scalable way.

Multiple radio access technologies may affect network management as well. In figure 4, there exist differences between the sensor communication and the LTE communication technologies such as sizes of the cells and hand over times. However, all of these technologies should be managed together since the communication devices installed in a car move and stop together.

Based on these insights (but not limited to), following requirements are suggested for the IMT-2020 network management:

- There should be principles, policies and methods to manage non-USIM devices.
 - The principles may include the way network accept and authenticate the non-USIM type of devices.
 - The policies may determine whether the non-USIM type communication should be charged at a certain rate or monitored during the operation time.

- Most of the devices communicating in the sensor cells may not be equipped with the formal methods to be connected to the network.
- Mixed communication among USIM-based and non-USIM-based devices should be also taken into consideration.
- Network management system should be able to handle different features of multi RATs in a very sensitive way.
 - Sensor communication should be monitored focusing on the latency, while LTE communication focuses on the bandwidth.
 - In spite of these different aspects, all communication technologies should provide operators with a similar (or fundamentally the same) way of monitoring and provisioning.
- Network management system should be scalable reflecting availabilities of the network resources proportionally.
 - When the size of the network resources increases, appropriate network management resources should be allocated.
- A brand-new “group-type” management entity should be defined to manipulate the multiple communications in a single car.
 - Since all the communication devices in a car moves together geographically, the most proper key attribute for this new “group type” data entities should be GPS position.
- A single and unified architecture should be prepared for fixed and mobile converged network, registration or free access networks, and various sliced networks.
 - The unified architecture should provide operators with a single view of the network status, and a holistic way of operation when they resolve the problems such as device faults.

7.3.2 Personal broadcasting service management requirement for IMT-2020

IMT-2020 network management requirements are discussed based on “personal broadcasting” services, which is regarded as one of the most feasible use cases for eMBB (enhanced Mobile Broadband) type of services.

Figure 5 shows a simplified version of personal broadcasting service. In this service, a person can produce the content and send it to the streaming server to distribute it to its watchers over the network.

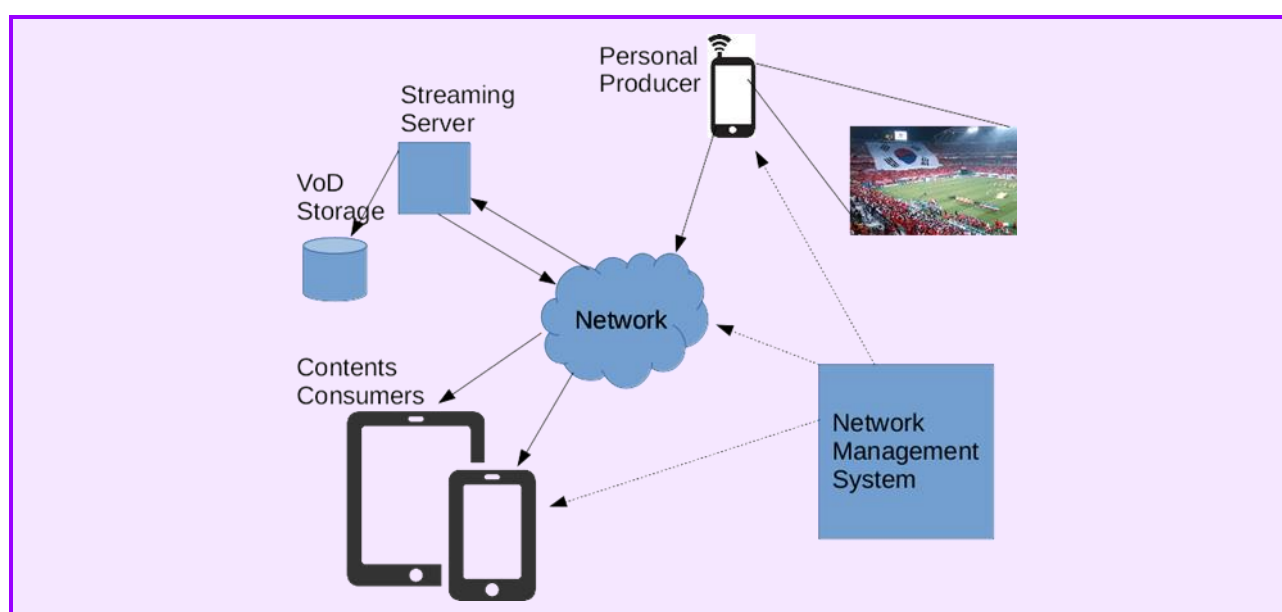


Figure 5 – A simplified personal broadcasting service

Capturing live event on the field and sending it to the watchers may require network slice technology in IMT-2020 environment. Typically, under 10ms latency and over 1Gbps bandwidth requirements are underline features of the IMT-2020 network to support this kind of eMBB services.

Due to the multicasting characteristics of the broadcasting service, the domain-wise slicing might be more reasonable than the end-to-end slicing passing through the whole domains. Creating slices for each one-to-one connections between each producer-consumer combinations may not be feasible in broadcasting service.

At least two slices will be stitched together to serve a channel, which include the first slice between the producer and the streaming server, and the second between the streaming server and the consumer.

Based on these insights (but not limited to), following requirements can be suggested for the IMT-2020 network management:

- In this service, the network management functions should monitor and manage the network equipment, especially the ones installed before and after the stitching point, in this case the stream server.
 - The stitching point can be the spot where the virtual and physical equipment met, and the network management functions should be able to be dealing with the subtlety of this point.
- Since the slices can be created and deleted flexibly, the network management functions should be characterized with proper agility accordingly.
- Since various encoding protocols can be used depending on UE's contents players, the streaming bit rates will be diverse. In this case, the bit rates for the streams should be managed and controlled under the total bandwidth allocated on that slice. This situation addresses the need of priority management for the slices and for the streams in a single slice.
- One other issue is that a single UE can connect to multiple channels. When a networking problem occurs on a specific channel, the operating staff should be able to describe and resolve the problem with the help of network management system. That means the network management functions should be able to handle each slices separately, so that it can recover each slice without rebooting the physical resources beneath them.

7.3.3 IoT service management requirement for IMT-2020

IMT-2020 network management requirements are discussed based on the traditional sensor networking and Home-IoT service which are relatively feasible in mMTC service category. After spending a long struggling period, the IoT services started to be deployed in the real market recently. The IMT-2020 the network management architecture should consider the IoT network features, such as, the big number of devices, the configuration flexibility due to the battery-operated devices, delegated management scheme via the sensor controller, and so forth.

IoT services can be grouped into 2 categories – 1) A public IoT service type, started from the beginning of IoT technology, in which numerous IoT devices are spread over the specific area human cannot access easily, to monitor the area and collect the information. 2) A convenience service type, where IoT devices are installed at home, office, or factory to provide convenience to human.

Networking features appeared in those two service categories are investigated and derive several requirements for IMT-2020 network management.

Figure 6 shows a traditional sensor network deployment, where a large number of sensor node monitor the specific area.

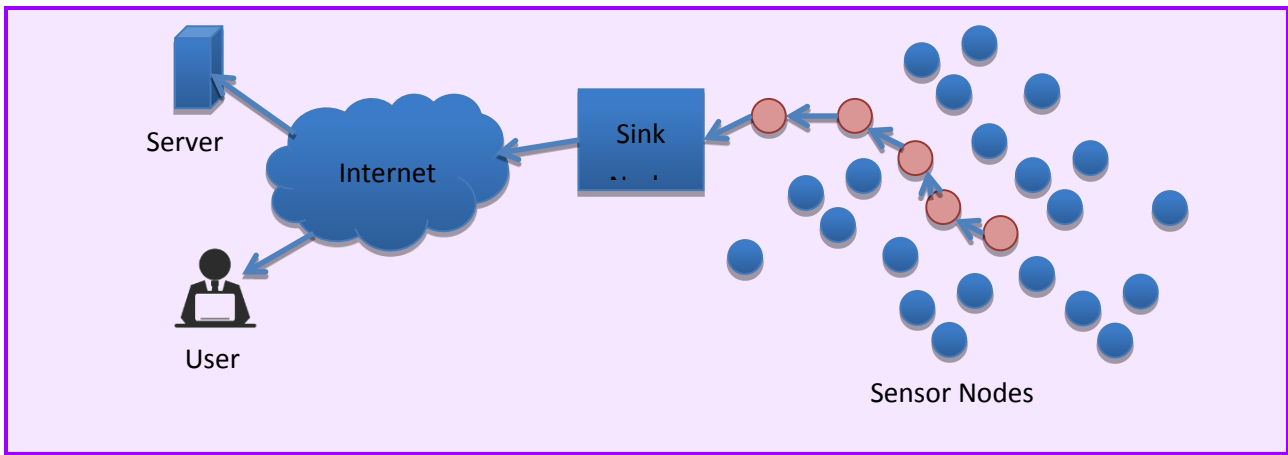


Figure 6 – A traditional sensor network service for revealing networking features

This service shows the following networking features.

- There are a large number of network elements and their types can be varied.
- The network hopping can be very lengthy.
- Network elements (sensor nodes) can be discarded once its battery consumed out.
- The capability of sensor node is weak, so it relies on the sink node to communicate with outside.

Figure 7 shows a simplified version of convenience type IoT service, where the IoT devices can help the human life by monitoring everyday life environments and actuating proper management and control actions.

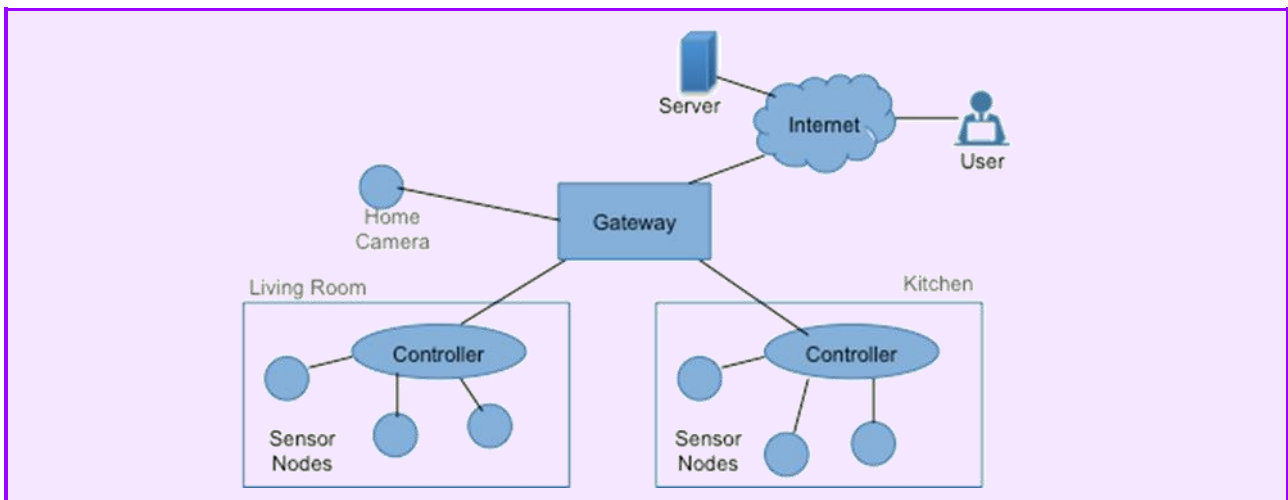


Figure 7 – A simplified version of convenience type IoT service for revealing networking features

This service shows the following networking features.

- The number of network elements is smaller than the sensor network system, but still larger than the legacy IT systems and their types are varied.
- Some types of devices can connect to the Internet gateway directly, but most sensors pass through the sensor controller to reach to the users and servers.
- Sensor controller is to be equipped with enough computing and communicating capability.

Based on the above insights (but not limited to), following requirements can be suggested for the eNMS:

- Since the number of nodes is increasing and the moments of installing and discarding the device are very frequent, the network management functions should be very scalable.
- The polling tasks for the devices to collect the real-time management information can give great burden to them, shortening the survival time of the node and even harming the service experience.
- Most of the small IoT devices do not provide any network management protocols.
- The network management function should properly communicate with the sensor controller to collect the information for each sensor under the controller, and sometimes even to send the control command to the controller to actuate the sensors.
- The sensor controller should provide general management protocols, and be able to delegate the management tasks for the sensors under its domain.

Appendix I

IMT-2020 Networks, Services and Resources Orchestration Functional Requirements

NOTE – The following requirements are the discussion/agreement from the sixth meeting of FG on IMT-2020. It requires the further review in order to be inserted into main text.

I Functionality Applicable to all Networks, Services and Resources Orchestration

I.1 System Primitives

- Function Name: *Slice Orchestration*: - Slice life cycle management (i.e. including concatenation of slices in each segment of the infrastructure and vertical slicing of the Data plane + Control plane + Service plane); slice elasticity, placement of VMs in slices. It takes over the control of all the virtualized network functions and network programmability functions assigned to the slice, and (re-)configure them as appropriate to provide the end-to-end service. [Slice is the collection of virtual network functions connected by links to create an end-to-end networked system. Slices are composed of multiple virtual resources which is isolated from other slices. Slicing allows logically isolated network partitions with a slice being considered as a unit of programmable resources such as network, computation and storage. Considering the wide variety of application domains to be supported by IMT-2020 network, it is necessary to extend the concept of slicing targeted by the current SDN/NFV technologies.
- Function Name: *Coordination*: It coordinates the infrastructure components with the view of protecting it from instabilities and side effects due to the presence of many service components running in parallel. It ensures the proper triggering sequence of SC and their stable operation. It defines conditions/constraints under which SC will be activated, taking into account operator service and network operation requirements (inclusive of optimize the use of the available network & compute resources and avoid situations that can lead to sub-par performance and even unstable and oscillatory behaviors).
- Function Name: *Service Platform Hosting* – IMT-2020 Node hosting functions for Service Platform full or partial functionality.
- Function Name: *Recursiveness Enablers*: Virtualization, slicing and orchestration are recursive and involve far more than simply subdividing, aggregating or combining resources. A domain orchestrator sees a set of resources for its exclusive use in satisfying the service request. Recursively within each subordinate/sub domain, the local orchestrator likewise sees and coordinates resources for its own use. Recognizing the generic and recursive nature of virtual resources, the Service Platform may instantiate a VNF of its choice on some available lower-layer container that it knows about in satisfying the service request.

I.2 Service Information Enablers

- Function Name: *Service Platform Information Coordination* - information / knowledge collection, aggregation, storage/registry, knowledge production, distribution and use across all service platform & SDK & infrastructure functions. The importance of the use of uniform information model cannot be overstated as the risk of semantic mismatch is exacerbated if different functions have incompatible information models. This allows purpose-specific APIs to be produced, while enforcing common semantics for the resources of concern.

- Function Name: *No Unnecessary Information* - No unnecessary information duplication when NS/VNFs and the NFVI are operated by different SPs: Information held by an instance of an IMT-2020 system must not create unnecessary duplication or other dependencies. In particular, a IMT-2020 system offering NFV Infrastructure as a service must not hold specific information about the type of VNFs or network services into which the VNFs are composed. Similarly, an IMT-2020 system of a client SP composing network services hosted on NFV Infrastructure controlled by another service provider must not hold implementation details of the infrastructure.

II Service Development Functionality

- Main Function Name: *SDK - Service Development Kit*: It supports the development of software running inside the actual infrastructure (i.e., both simple virtual network functions and, more importantly, composed services) or running inside the service platform (e.g., decision algorithms about placement or scaling of a service). It also supports the packaging and analysis of services.

II.1 SDK Primitives

- Function Name: *SDK*: Service Development Kit (SDK) for fast service specification based on a common and uniform resource abstractions model for connectivity, computing and storage.
- Function Name: *Invariant specification methods*: invariant specification methods for developers and corresponding verification and debugging tools for invariant checking and reporting during testing and operations.
- Function Name: *Profiling tools*: profiling tools for monitoring and reporting of service performance and scaling behaviour at run time as feedback to developers.
- Function Name: *SDK Packaging*: service package that can be handed over to the service platform for execution. This package needs to describe constituting components (individual NFVs, how the composition looks like, scaling properties, certificates for authorization, etc.). We define a package format that describes and encapsulates all such components and that can be processed by the gatekeeper. Support the developer in two ways. One is a set of editors for the various description formats (e.g., for service compositions, for placement optimizations logics). The second is the packaging function, which takes input from the editors, collects all the necessary software artefacts (e.g., required NFV virtual machine images), and produces a package.
- Function Name: *SDK Catalogue Access*: support the developer in reusing existing NFVs, services, or decision logics. It can interface with, query, or browse existing catalogues, e.g., a private one of the developer, those of the target service platform, or even public marketplaces (like developed by T-Nova or Murano), possibly handling licensing and payment issues. This is akin to dynamically installing libraries in scripting languages from public repositories.
- Function Name: *VNF Catalogue*: The SDK must provide a location to store the different IoT related VNFs. It should be possible for the VNF developer to add update or delete VNFs from that location. Using this list the IMT-2020 service developer can compose a complex service. The system shall offer a VNF catalogue from which the customer can select the desired VNFs.
- Function Name: *VNF Scaling metadata*: The SDK must allow definition of SLA levels for selected VNFs, other metadata should be possibly be specified as well, such as when and how the IMT-2020 operator should scale the VNF, as well as the scaling strategy (up/down, in/out). This information can be used by the IMT-2020 platform to automatically scale the IoT gateways using appropriate methodologies. The developer should describe in the VNF Descriptor recipes for scaling VNF; VNF composing the network service have to be able to scale up or down depending of the users' demand. IMT-2020 SDK should have the capability to specify different parameters (VM load, BW, latency) to be used by IMT-2020 Orchestrator for scaling (up or down). The developer should describe in the VNF Descriptor recipes for scaling his/her VNF.

II.2 SDK Tools

- Function Name: *VNF SLA Monitor*: IMT-2020 must provide an interface to monitor VNFs SLAs and resource usage. It must highlight VNFs with high and low usage that may need scaling or other kind of manual intervention. The system shall expose service and VNF metrics to the network application.
- Function Name: *VNF Resource Report*: IMT-2020 must provide an interface to list all resources allocated to a specific service. This service allows the developer or administrator to get an overview of how the service is evolving and what datacenter resources are committed to each service.
- Function Name: *Authorization*: IMT-2020 service must limit operations based on access levels and provide means to create and manage access profiles. This will be used to define the different access levels each user will have to the system, as an example, a VNF developer should be able to deploy a VNF on the catalogue, but should not have the permission to trigger its deployment to a production network.
- Function Name: *VNF Deployment*: IMT-2020 must support placement instructions that express proximity to other entities, e.g., (i) set where the service gateways will be placed on the operator network, (ii) deploy a VNF as near as possible to a specific location, (iii) select where the VNF will be deployed .
- Function Name: *VNF Status Monitor*: IMT-2020 should provide a high level state for each VNF, e.g., (i) deployment, (ii) operating, (iii) error.
- Function Name: *IoT traffic simulator*: Given that there is not yet the amount of IoT traffic this use case is designed to address, there must be a way to simulate IoT sensor traffic with functions like increase or decrease traffic levels per sensor and number of sensors in order to simulate a real IoT sensor environment.
- Function Name: *VNF integration with service*: IMT-2020 must allow new VNFs to be integrated in existing services. It must allow network flow reconfiguration in order to integrate a newly deployed VNF in an existing service graph with minimum or no downtime at all.
- Function Name: *SDK VNF customization*: The SDK must allow the development of custom VNFs with specific algorithms to manipulate IoT traffic, like processing and batching.
- Function Name: *Multiple IoT sensor vendors*: Framework must support traffic from different IoT sensor vendors. Traffic from each sensor should be routed through the appropriate gateway.
- Function Name: *Multiple IoT tenants*: Framework must support multi tenancy, i.e., The infrastructure must support multiple IoT services operating in parallel without any data meant to one operator being routed to another operator's service.
- Function Name: *Support for Service Templates*: The programming model must support service templates. In other words, it must support the inclusion of types of nodes, or at least the notion of cardinalities in inter-node relationships, e.g. in order to define an unspecified number of nodes. Support for corresponding annotations (or primitives) in the service programming model/language.
- Function Name: *Inter-VNF QoS constraints*: The programming model must support end-to-end QoS properties for inter-VNF communication, such as delay, jitter, reliability (which could be mapped to multi-path transmission by the orchestrator, the developer does not care necessarily), oversubscription.
- Function Name: *Placement constraints for VNFs*: The programming model must support to specify placement constraints for VNFs, e.g. disjoint placement of active and standby VNF on physically separate machines, pinning a VNF to a specific node or node type (e.g., turbine control must run on a turbine node), hosting nodes must offer certain real-time capabilities or security isolation features, satisfaction of rules of compliance, etc.

III Service Platform Functionality

- Main Function Name: *Service Platform*: service platform realizes the management functionality to deploy, provision, manage, scale, and place services on the actual infrastructure. It does not execute the services themselves; it rather triggers execution on the actual infrastructure by requiring start-up, migration, shutdown, etc of (virtual) network functions on the actual infrastructure.

III.1 Service Platform Primitives

- Function Name: *Infrastructure Abstraction* – assuming no uniform control interface has appeared, a shim layer functionality to hide idiosyncrasies of an infrastructure. This should be lightweight function and hopefully will disappear once standards emerge.
- Function Name: *Conflict Resolution*: Since service-specific logics are likely selfish, conflicts over resource usage will arise.
- Function Name: *Service Platform Scalability*: The service platform must be scalable to support a very large number of devices (e.g., sensors), a high traffic load, high dynamics etc. depending on the use case.
- Function Name: *Service Platform Customizability*: The service platform must be customizable to support large-scale, complex deployments (such as carrier networks) as well as smaller, lightweight deployments (such as enterprises or industrial networks).
- Function Name: *Capability Discovery in Service Platform*: The service platform, notably the infrastructure abstraction layer, must support the discovery of capabilities of the physical infrastructure, e.g. support for hardware-acceleration for certain functions such as encryption or the availability of a Zigbee interface. That way, it will become possible to optimize function placement and maybe even tune applications that have access to the capability-enriched network model via the service platform.
- Function Name: *Isolation constraints for VNFs*: The programming model must support isolation constraints for VNFs. This is in terms of performance, e.g. in order to guarantee min. capacity without being pre-empted by concurrent services. But it is also in terms of security, e.g. in order to restrict visibility of (virtual or real) infrastructure to a particular service, or to constrain a service to specific time windows (e.g., only between 10am and 11am, or expiry one hour after first use).
- Function Name: *Multi-tenancy*: Some components can be dedicated to a tenant (hard isolation) and some other can be shared (soft isolation).
- Function Name: *Security VNF availability*: Security virtual network functions require specific capabilities that are not so common in generic VNF, like Anti DDoS or signature detection of IDS. These functionalities must be present to allow creating a valid use case. IMT-2020 VNF Catalogue must include some Security VNFs.
- Function Name: *Personalized VNF*: VNF catalogue and management framework in IMT-2020 must support the concept of “personal” in the sense that VNFs are assigned as a non-shareable resource with other users in the platform. Also Users identities in IMT-2020 framework must allow a direct mapping between user and his VNFs case.

III.2 Service Platform Tools

- Function Name: *Catalogues and repositories*: a service (along with its constituting parts: logics, NFVs) is placed into corresponding catalogues inside the service platform, from where both kernel and actual infrastructure can access them. The catalogues hold known entities; they are complemented by repositories, which hold information about running entities as well as other frequently updating data, e.g., monitoring data.
- Function Name: *GateKeeper*: This service platform function will check whether a service can be executed by the service platform before accepting it. It will check, e.g., authorization of a developer submitting a service, completeness of the service description, or the availability of all NFVs composing the service.

- Function Name: *Service Monitoring and Monitoring Analysis*: The service monitoring and the monitoring analysis working closely together, will collect and analyse service-level (not just network) performance indicators.
- Function Name: *NFVI Northbound API*: The IMT-2020 system must be able to support an API which exposes the NFV Infrastructure as a service.
- Function Name: *Southbound Plugin to use NFVI API*: The IMT-2020 system must be able to support a southbound interface which can request elements of NFV Infrastructure as a service.
- Function Name: *Timely alarms for SLA violation*: The monitoring system must supply alarms for SLA violations (or malfunctioning components) in a timely manner depending on the SLA and type of problem. This means that the failure detection, but also the service platform message bus and notification system must have real-time capabilities. E.g. VNF unavailability for real-time traffic must be signaled as fast as possible, while in the case of best-effort traffic alarm signaling can happen with modest delays. Likewise, urgency of alarms is higher for VNFs with 1000s of users compared to single-user VNFs in the general case.
- Function Name: *Manage update of components*: Sequence/ strategy of update using DevOps. Sequence for validation and migration.
- Function Name: Support different modes of management/control: EPC can be fully managed by the operator, i.e. EPC fully deployed and managed by the customer (e.g., a MVNO). Or Hybrid where components are managed by the operator (e.g., SGW) and others by the customer (e.g., HSS). Or fully managed by the customer.
- Function Name: *Support Services with high SLA*: vEPC is a service that operates under a 5 nines SLA, it can not allow service degradation when scaling / healing / updating / migrating.
- Function Name: *Support "state-full" services*: vEPC is a "state-full service" - all its components are state-full, they can not lose their state when scaling / healing / updating / migrating.
- Function Name: Integration with OSS: vEPC service operation involves integration with OSS system, IMT-2020 should expose relevant APIs.
- Function Name: *Distributed NFVI*: ISP and Network Operators architecture requires a geographical distribution of PoP (Point of Presence) where instantiate multiples VNFs as close as possible to user or based on the service demand. One example is when a security attack happens it is preferred to react as close as possible to the source of the attack. As a consequence IMT-2020 orchestration layer should support multiples NFVI and VIM in distributed networks case .
- Function Name: *Open interfaces towards NFV*: NFV components like VIM, NFVI or NFVO could be deployed with multiples providers. Indeed the number of NFV solutions is growing day by day. IMT-2020 orchestration framework must support open or standards interfaces (southbound towards NFVI) to ensure the smooth integration of different NFV providers.
- Function Name: *VNF Real-time Monitoring*: In order to detect and react to security incidents, VNFs will generate in real time information useful for Monitoring and response. IMT-2020 framework must be able to collect, store, process and report in valid time windows to be useful to the ISP.
- Function Name: *VNF reporting to BSS/OSS and subscriber*: In order to detect and react to security incidents, VNFs will generate in real time information useful for Monitoring and response. IMT-2020 framework must be able to collect, store, process and report in valid time windows to be useful to the ISP.
- Function Name: Legacy support: Any ISP or Network Operators or corporations has today deployed security networks solutions in virtualized or bare metal appliances. The most relevant example is a Firewall device. If IMT-2020 has the aim to offer complex solutions and integrate with existing network environment, then IMT-2020 need to interact and manage with not only VNFs.
- Function Name: *Quality of service monitoring*: One of the key method to detect security problems are the deterioration in the QoS. Metrics generation and degradation detection of network traffic, i.e. caused by a overloaded NFVI node or a attack, should be supported and reported case.

- Function Name: *VNF and topology validation*: Based on the principle of providing security service, IMT-2020 service framework by itself, or using third parties, must offer a validation capacity of VNFs when it is deployed in the NFVI. This validation should cover the integrity of the VNF, user attestation and data paths case.

IV Service Orchestration Functionality

- Main Function Name: *Orchestrator*: service orchestrator that maps services to connectivity, computing, and storage resources and that
- processes the output of the IMT-2020 SDK to generate a resource mapping and composition of a new service from virtualized building blocks,
- manages the service lifecycle (deployment, operation, modification, termination),
- supports isolation and policing between different virtual services, and virtual service providers.
- Uses abstract interfaces for interoperability with different underlying technologies such as OpenStack, Apache Cloudstack, OpenVim, OPNFV, etc.

IV.1 Service Orchestration Primitives

- Function Name: *VNF Placement*: The programmability framework shall allow the customer to deploy VNFs at arbitrary points into the network and set where the components/ gateways will be placed on the operator network. For example, deploy a VNF as near as possible to a specific location or select where the VNF will be deployed.
- Function Name: *Manual Service Function Placement*: It should be possible to manually decide and configure where a service is to be placed. This can be very important for services where the service developer knows that a Service Function has to run in a certain location / on a certain node, but is either unable to or not confident with defining placement constraints in a way that placement can be done by the orchestrator. This may be particularly the case in non-carrier verticals where experience with services may be lacking, deployment are simple, and ease of use is the primary objective.
- Function Name: *SFC: Service chaining* - the programmability framework shall allow the customer to interconnect VNFs in an arbitrary graph.
- Function Name: *Service Chaining Support Across Wide Area Networks*: The Service Platform must support service function chains that include service functions separated by a wide area network, e.g. across different data centres, or between the core data centre and an edge cloud.

IV.2 Resource Orchestration Primitives

- Function Name: *Abstract Programming Model*: abstract programming models for networked services, which enable a high level of automation in service development and deployment processes. Such models refer to the installation of executable code as virtual machines representing application-specific services components into the hosting elements in order to create the new functionality at run time - realizing application-specific service logic, or performing dynamic service provision on demand.
- Function Name: *Multi NFVI orchestration*: IMT-2020 Orchestrator should be able to orchestrate multiple VNF execution environments (NFVI-PoPs) located in arbitrary places in the operator network topology. The NFVI-PoPs are considered to be controlled and managed by VIMs.
- Function Name: *Lifecycle Management*: The lifecycle management plugin deal with triggering (groups of) VM start-up, shutdown, ... actions in the actual infrastructure; the service contextualization executive plugin contextualizes VNFs/services via actions provided by the service description.
- Function Name: *Placement and Scaling*: The placement and scaling logic executive executes algorithms that place and scale a running service, both at start-up and continuously (e.g., when load goes up). While we will provide a fall-back algorithm (based on-going projects), our contribution is the ability to execute service-specific logics, provided in a service's description. Challenges here are e.g. security concerns, which we intend to address by sandboxing approaches.

- Function Name: *Multi NFVI orchestration*: IMT-2020 Orchestrator should be able to orchestrate multiple VNF execution environments (NFVI-PoPs) located in arbitrary places in the operator network topology. The NFVI-PoPs are considered to be controlled and managed by VIMs.
- Function Name: *Integration with existing VNFs or components*: The programming model and service platform must allow components or VNFs of a new service to be integrated with existing services, VNFs or system components (such as sensors or actuators).

Contributors (in Alphabetical Order)

(This appendix does not form an integral part of this Recommendation)

This is the list of all contributors who submitted any written form of comments or contributions.

- Alex Galis, University College London, U.K.
- Hyungsoo Kim, KT
- Jongpil Lee, KT
- Olivia Heeyun Choi, KT
- Sangwoo Kang, KT
- Seongbok Baik, KT

Acknowledgement

(This appendix does not form an integral part of this Recommendation)

This work was partially supported the EU H2020 5G PPP projects: 5GEX (“5G Multi-Domain Exchange”; <https://www.5gex.eu>) and SONATA (“Service Programing and Orchestration for Virtualized Software Networks”; <http://sonata-nfv.eu/>)



An aerial night view of a city skyline, likely Tokyo, featuring numerous illuminated skyscrapers and a complex highway interchange. A semi-transparent white rectangular box is overlaid on the upper right portion of the image, containing the title text. The background image is vibrant with city lights and a clear sky.

Network Management Framework for IMT-2020

Summary

This is the output document introducing: Network Management Framework for IMT-2020.

This document describes general aspect of Network Management Framework for IMT-2020. The objective of this document is to provide network management architecture and functional components for design, deployment, operation to implement IMT-2020 network covering fixed and mobile networks. More specifically, specifying the functionality for interactions with the IMT-2020 network management systems is in the scope to support end-to-end management but the management of mobile radio access network itself is out of scope of this document.

Table of Contents

1	Scope
2	References
3	Definitions
3.1	Terms defined elsewhere
3.2	Terms defined in this Recommendation
4	Abbreviations and acronyms
5	Conventions
6	Overview of IMT-2020 end-to-end network management
6.1	Motivation
6.2	Underlying Technologies
6.3	Clarification of Management and Orchestration for IMT-2020
7	IMT-2020 Network Management General Architecture
8	IMT-2020 Slice Life Cycle Management Functional Architecture
8.1	Physical Data Plane & Resource Management Functional Component
8.2	Virtual Data Plane & Resource Management Functional Component
8.3	Control Plane Management Functional Component
8.4	Application and Service Management Functional Component
8.5	Multi-Plane Management Orchestration and Slice Lifecycle Management Support Functional Component
8.6	External Relationship Management Functional Component
9	IMT-2020 Management Procedure and Implementation Scenarios
9.1	IMT-2020 Management Procedure
9.2	IMT-2020 Management Implementation Scenarios
9.3	Virtual Network Management
9.4	Integrated Network Management
	Appendix I – Contributors (in Alphabetical Order)
	Appendix II – Acknowledgement



1 Scope

This Recommendation specifies the overview of IMT-2020 end-to-end network management and a general architecture of IMT-2020 Network Management, a slice life-cycle management functional architecture, management procedures and implementation scenarios.

2 References

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. All users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- [1] ITU-T Recommendation M.2083-0 (2015), *Framework and overall objectives of the future development of IMT for 2020 and beyond*.
- [2] ITU-T Recommendation Y.3001 (2011), *Future networks: Objectives and design goals*.
- [3] ITU-T Focus Group on IMT-2020 (2015), *Report on Standards Gap Analysis*.
- [4] ETSI GS NFV-MAN 001 V1.1.1 (2014), *ETSI NFV Management and Orchestration - An Overview*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 IMT-2020 [b-ITU-R M-2083-0]: systems, system components and related aspects that support to provide far more enhanced capabilities than those described in Recommendation ITU-R M.1645.

3.1.2 software-defined networking [b-ITU-T Y.3030]: A set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner.

3.1.3 domain [b-ETSI NFV MANO]: Administrative domain is a collection of systems and networks operated by a single organization or administrative authority. Infrastructure domain is an administrative domain that provides virtualized infrastructure resources such as compute, network, and storage or a composition of those resources via a service abstraction to another administrative domain, and is responsible for the management and orchestration of these resources.

3.1.4 network softwarization [b-ITU-T O-016]: Network softwarization is an overall transformation trend for designing, implementing, deploying, managing and maintaining network equipment and network components by software programming, exploiting characteristics of software such as flexibility and rapidity of design, development and deployment throughout the lifecycle of network equipment and components, for creating conditions that enable the re-design of network and services architectures; allow optimization of costs and processes; and enable self-management.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 IMT-2020 Planes: A plane is a subdivision of the specification of a complete IMT-2020 system, established to bring together those particular pieces of information relevant to some particular area of concern during the analysis or design of the system. Although separately specified, the planes are not completely independent; key items in each are identified as related to items in the other planes. Each plane substantially uses foundational concepts. However, the planes are sufficiently independent to simplify reasoning about the complete system specification.

3.2.2 Multi-Service Management Plane: The functions and interfaces in this plane are used to set up and manage groups of network instances and/or nodes. More specifically, the setup consists of creating/installing/arranging NFs and interfaces according to the available physical and virtual resources. It also comprises the set of functions associated with the network operations, such as fault management, performance management and configuration management. It further includes the lifecycle management of individual network functions and mobile network instances as a whole. In current mobile networks, this role is often performed by the Operations Support System (OSS). The idea is to enable the creation, operation, and control of multiple dedicated communication service networks running on top of an IMT-2020 E2E infrastructure.

3.2.3 Integrated Network Management & Operations Plane: Enables the creation, operation, and control of dedicated management functions operating on top of an IMT-2020 E2E infrastructure. The collection of resources responsible for managing the overall operation of individual network devices.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

For the purpose of this Recommendation, the following abbreviations are used:

BSS	Business Support System
CAPEX	Capital Expenditure
CMS	Common Management System
E2E	End to End
eNMS	Enhanced Network Management System
NE	Network Equipment
NFV	Network functions virtualization
KPI	Key Performance Index
LTE	Long Term Evolution
MANO	Management and Orchestration
NAT	Network Address Translation
OAM	Operations and Management
OCH	Orchestration
OPEX	Operational Expenditure
OSS	Operations Support System
RAN	Radio Access Network
SDN	Software-defined Network
VIM	Virtualized Infrastructure Manager
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager
WAN	Wide Area Network

5 Conventions

None.

6 Overview of IMT-2020 end-to-end network management

Since the current network architecture may not be appropriate to support various IMT-2020 network requirements, enhancement of the network architecture has been studied. Among the enhanced capabilities including distributed function deployment, network slicing, and resource allocation, IMT-2020 also requires a study on new aspects of end-to-end network management whose results are described in this Recommendation.

6.1 Motivation

Based on the understanding of the new trends of networking technology as mentioned above, Phase 1 of FG on IMT-2020 identified the following three gap analysis items related to the end-to-end network management:

- Multiple network management protocols in different network domains make it difficult to support unified network operations over multiple network domains. A unified end-to-end network management should be considered to ensure compatibility and flexibility for the operation and management of an IMT-2020 network.
- OAM protocols are not standardized in some parts of IMT networks such as the front haul network. Standard OAM protocols should be studied for fault management and performance management between network equipment that may be commonly used across the IMT-2020 network.
- There are two aspects to consider for the network management and orchestration for the network softwarization. The first aspect is how to manage and orchestrate the softwarized network components. The second is how to softwarize network management and orchestration functionality. The current technology gaps to be filled in are provided.

There are additional items to be studied further respect to the end-to-end network management such as network slicing, integrated access, open-source platform and etc. Therefore, this Recommendation specifies an end-to-end network management framework for IMT-2020 in a systematic approach including aforementioned gap analysis items as well as those additional items.

6.2 Underlying Technologies

Together with SDN, NFV and cloud computing technologies, network softwarization is established for rapid service creation especially a new service. With seamless service assurance and management across global and local network, virtual and physical resources are managed with real-time operation systems and processes. Software-defined networking dynamically connects distributed and diverse workloads, networks and devices. White box switches are blank standard hardware that supports a set of basic networking features customized to meet any specific business and networking needs. End-to-end virtual network paths or slices are created continuously with dynamic reconfiguration. By leveraging standard virtualisation technology, network equipment such as switch and storage is consolidated onto industry standard high volume servers located in data centers, network nodes and end user premises. The embedding of the cloud in the network plays a key role to optimize network performance. It is crucial to include end-to-end network management for softwarized infrastructure.

6.2.1 Software-defined networking (SDN)

Software-defined networking (SDN) is an umbrella term encompassing several kinds of network technology aimed at making the network as agile and flexible as the virtualized server and storage infrastructure of the modern data center. The goal of SDN is to allow network engineers and administrators to respond quickly to changing business requirements. In a software-defined network, a network administrator can shape traffic from a centralized control console without having to touch individual switches, and can deliver services to wherever they are needed in the network, regardless of what specific devices a server or other device is connected to. The key technologies are functional separation, network virtualization and automation through programmability. Further study is needed for SDN.

6.2.2 Network functions virtualization (NFV)

NFV (also known as virtual network function (VNF)) offers a new way to design, deploy and manage networking services. NFV decouples the network functions from proprietary hardware appliances such as network address translation (NAT), firewalling, intrusion detection, domain name service (DNS), and caching. All network functions can run in software.

It's designed to consolidate and deliver the networking components needed to support a fully virtualized infrastructure – including virtual servers, storage, and even other networks. It is applicable to any data plane processing or control plane function in both wired and wireless network infrastructure. Further study is needed for NFV.

6.2.3 Cloud computing

Cloud computing technologies in telecommunication infrastructure brings new challenges in management perspective. One important challenge for telecommunication operators is efficient management of cloud computing taking into account the legacy management system framework and assuring the customer's satisfaction including the end-to-end quality of service.

Cloud computing is different from traditional telecommunication networks since it does not expose individual elements to the telecommunication management system. Moreover, cloud computing does not distinguish between management operations carried out on behalf of customer and network operator.

6.3 Clarification of Management and Orchestration for IMT-2020

It is essential to clarify management and orchestration for IMT-2020 before going further. Network architecture view of IMT-2020 combined network management and orchestration in a single plane.

In order to clarify the difference of orchestration (OCH) and management (MNG), the following implementation scenarios are able to be taken into account.

- 1) OCH and NMS will exist independently, with close collaborating relationship.

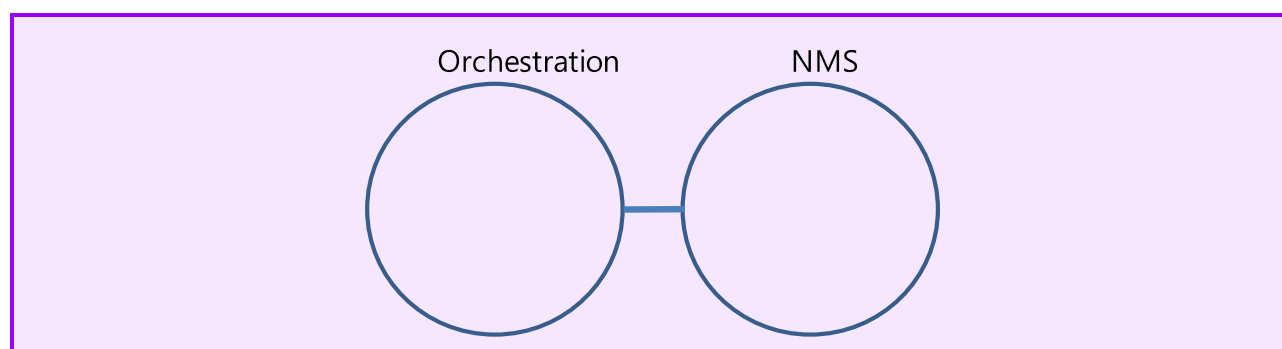


Figure 1 – Independent OCH and NMS

- 2) Among OCH and NMS, one of them will include the other.

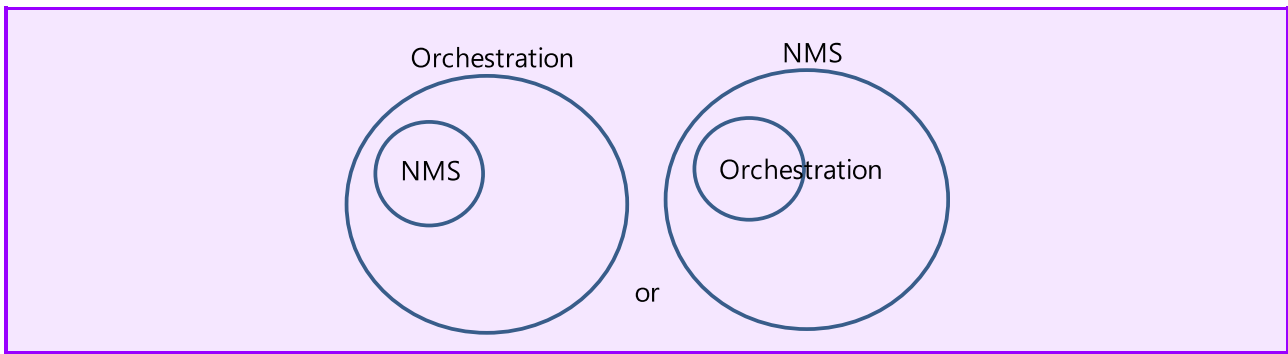


Figure 2 – OCH/NMS includes the other

But, it might be reasonable to see the current status as an intersection type of relationship between them.

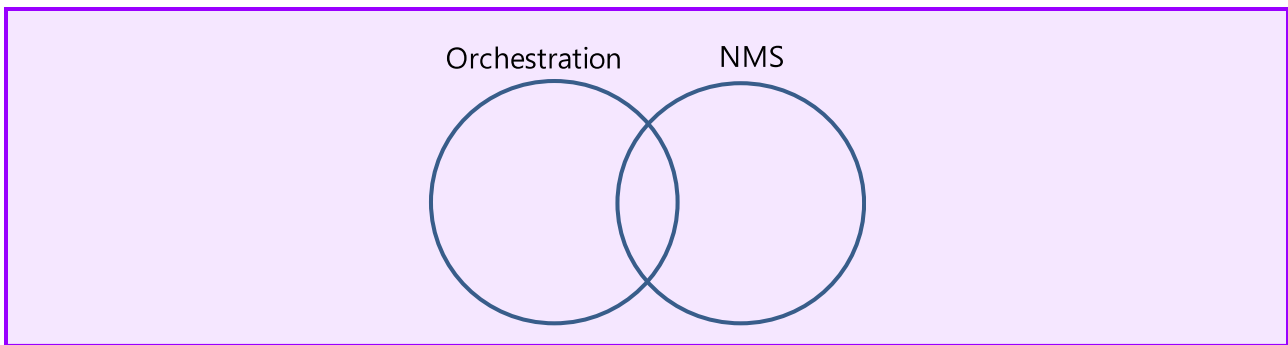


Figure 3 – OCH and NMS with an intersection

Following table shows particular contents to differentiate the functions of OCH and NMG.

Table 1 – Difference between OCH and NMS

	Orchestration	Management
Monitoring purpose	Availability	Healthiness
Action purpose	Provisioning	Maintaining
Representative Actions	Control / Configuration Create / Destroy / Move	Monitor / Alarm for Event Detection / Isolation / Resolve for Fault
Target Resources	Dissimilar Devices	Similar Devices

Based on the above classification, the definitions of orchestration and management are specified.

Management: Management is managing the network functions in both physical and virtual infrastructure including compute, storage, and network resources within one operator's infrastructure sub-domain. Overall coordination and adaptation for configuration and event reporting are achieved between network function infrastructure and network management systems. It includes the collection and forwarding of performance measurements and events. Network function lifecycle management is included with network function instance management. Network management system is authorized to exercise control over and/or collect management information from another system. It is tightly connected with BSS/OSS such that the most efficient and effective way to access, control, deploy, schedule and bind resources is chosen as requested by customers.

Orchestration: Orchestration is the automated arrangement, coordination, and management of complex network systems including middleware for both physical and virtual infrastructure. It is often discussed as having an inherent intelligence or even implicitly autonomic control. Orchestration results in automation with control network systems. Orchestration is the key function in management plane. Orchestrator manages network service lifecycle and coordinates the management of network service life cycle, network function lifecycle and network function infrastructure resources to ensure optimized allocation of the necessary resources and connectivity. It is also tightly connected to OSS/BSS for service management.

7 IMT-2020 Network Management General Architecture

IMT-2020 network shall provide network services demanding diverse requirements, by using network functions instantiated at right place. IMT-2020 infrastructure will provide required infrastructure resources to instantiate the network functions. Network operators can provision and operate many different network slices according to their business strategies.

Network slicing enables the operator to create logically partitioned networks customized to provide optimized solutions for different market scenarios which demand diverse requirements in terms of service characteristics, required functionality, performance and isolation issues.

The functional architecture of IMT-2020 network shall provide a complete set of network functions required to support all IMT-2020 services. A network slice is comprised of only necessary network functions. They are collected from a complete set of network functions in the IMT-2020 network functional architecture, and orchestrated for the particular service and purpose.

The general framework of IMT-2020 can be represented by two separate architecture levels, i.e. 'slice orchestration and management' level and 'network slice instances' level as shown in Figure 4. Functions for creating and managing network slice instances the functions instantiated in the network slice instance are mapped to respective architecture level.

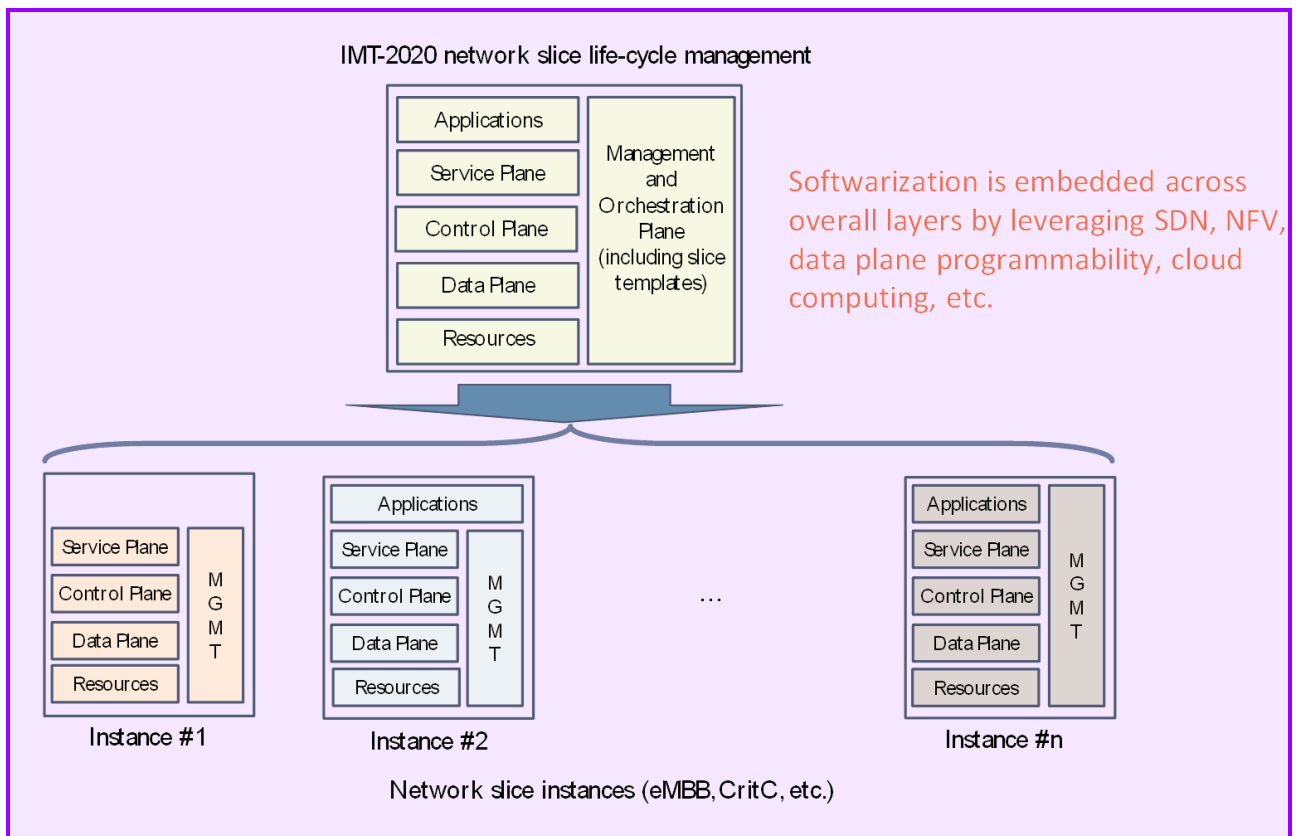


Figure 4 – Conceptual IMT-2020 non-radio network architecture

As the general framework of IMT-2020 consists of two level, the associated management architecture is also required to deal with two level: management in IMT-2020 network slice life-cycle management and management in each network slice instances. This Recommendation specifies both management functionality in separate clauses, Clause 8 and 9 respectively.

8 IMT-2020 Slice Life Cycle Management Functional Architecture

This clause describes detailed management functionality in the Slice Lifecycle Management (SLCM) functional component. Figure 5 shows its functional elements.



Figure 5 – Slice Lifecycle Management functional components

IMT-2020 Slice Lifecycle Management Customer Care Support

The IMT-2020 Slice Lifecycle Management Support (SLM-S) functional element provides a standard interface to the Slice Lifecycle Management functionality to its customers and applications, it supports requesting and receiving management operations and associated information in SLM.

Slice Capacity Planning & Optimization

The Slice Capacity Planning and Optimization (SCPO) functional element is responsible for planning of necessary resources for the requested slice provisioning and optimizing usage of resources for creating and maintaining slices. It provides capabilities as follows:

- making planning decisions based on the available resources discovered by the Slice Resource Monitoring & Analytics functional element and customers' requests.
- trying always to find optimal available resource matches against the customer's requests.
- monitoring and ensuring quality of the provisioned slices and take necessary actions (including re-provisioning or modification of the existing slice resource) if resource re-optimization is needed.

Slice Provisioning

The Slice Provisioning (SP) functional element is responsible for provisioning requested slices by the customers and provides capabilities for:

- provisioning the requested slices by the customers.
- mapping and translating customer's high-level slice provisioning profile into technology-aware slice provisioning policies
- managing provisioning policy lifecycle.

NOTE – Slice provisioning can involve interactions with the MANO NFV orchestrator, other NFV management systems, and/or SDN controllers depending on administration boundaries of the underlying slice resources. If the SP functional element provides full provisioning capabilities, the slice provisioning can be done internally by itself. If interactions with external provisioning functional entities are needed, it can be done through IMT-2020MPS functional element which is the interface to the external IMT-2020 management systems.

Inter-Slice Orchestration

The Inter-Slice Orchestration (ISO) functional element is responsible for orchestration of inter-slice matters and provides capabilities for;

- orchestrating multiple slices provisioning
- resolving inter-slice quality, fault, anomaly, and charging issues

Slice Fault Management

The Slice Fault Management (SFM) functional element is responsible for fault management of the provisioned slices and provides capabilities for:

- detecting anomalous events which cause failure of the provisioned slice resources.
- analyzing a root cause of the failure of the provisioned slice resources
- generating failure resolving policies and interact with SCPO functional element for the actual healing actions.

Slice Security Management

The Slice Security Management (SSM) functional element is responsible for security management of the provisioned slices and provides capabilities for

- providing authentication and authorization capabilities of the provisioned slices
- detecting and avoiding anomalous attacks of the provisioned slices

Slice Charging Management

The Slice Charging Management (SCM) functional element is responsible for accounting management of the provisioned slices resource usage and provides capabilities for metering and reporting slice resource usage data for charging. Resource usage data can be metered per slice or per end-user/customer.

Slice Resource Monitoring & Analytics

The Slice Resource Monitoring and Analytics (SRMA) functional element is responsible for collecting the status and events of the provisioned slice resources and analyzing them for the purpose of fault, quality, and security management and provides capabilities for:

- monitoring the activities, status, anomalous events of the application resources in the provisioned slices
- analyzing the monitored data and providing reports on the behavior of the resources, which can take the form of alerts for behavior which has a time-sensitive aspect (e.g., the occurrence of a fault, the completion of a task), or it can take the form of aggregated forms of historical data (e.g., resource usage data);
- storing and retrieving monitored data and analysis reports as logging records in the Slice Resource Repository.

Slice Resource Repository

The Slice Resource Repository (SRR) functional element is responsible for storing the contents discovered by the SFM, SSM, SCM, and SRMA and customers request profiles, and managing the lifecycle of the contents in the repository and provides capabilities for:

- storing and providing APIs for query the contents in the repository.

- lifecycle management of the contents in the repository (e.g., creation by storing, modification, deletion, etc.)

External Management Entity Support

The External Management Entity Support (EMES) functional element provides an interface to the external management system including IMT-2020 Multi-Plane Management functional component for requesting and receiving management operations and associated information for slice management specific to a particular slice instance and to the external IMT-2020 management systems for provisioning and provide capabilities for:

- requesting and receiving slice instance specific operational status
- requesting and receiving slice instance specific performance, fault, security related statistics and events
- requesting and receiving slice provisioning requests and responses to/from the external IMT-2020 management systems.

IMT-2020 Slice Instance Management Functional Architecture

This clause specifies network slice instance management functional architecture and components. Figure 6 shows a general network slice instance management plane architecture and relationship, interfaces with other IMT-2020 network slice instance functional entities.

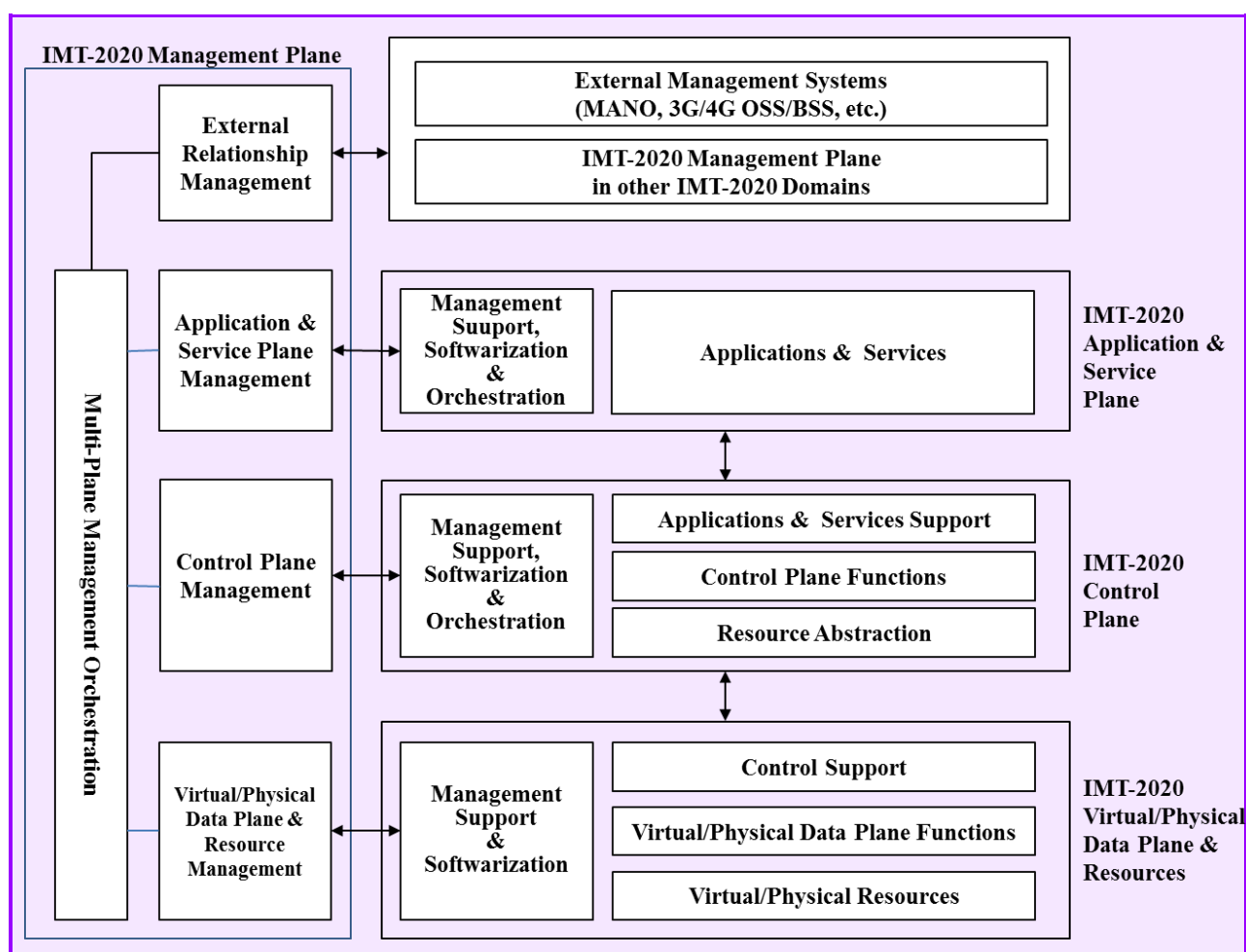


Figure 6 – IMT-2020 slice instance management functional components architecture

The remaining part of this clause specifies the details of each management functional components.

8.1 Physical Data Plane & Resource Management Functional Component

Physical Data Plane & Resource Management Functional Component manages physical data plane and resources including compute, storage, and configuration. It provides complete visibility into the physical state of the data plane and resource at any given time. Handling many devices is achieved providing both device-centered perspective and network-wide perspective. Connectivity is established over the physical data plane and resource through its management. All device accesses pass through physical data plane and resource management such that access and catalogue real-time status information about physical layer data plane and resource is available at all times. The state of every physical Data Plane & Resource connection is reported whether or not it is connected, how much bandwidth it can carry and the type with corresponding mapping and alarming. It manages the entire physical layer, mapping routes, issuing work orders, reserving available ports, reporting on flow/path information and other physical functions.

This sub-clause describes detailed management functionality in the Physical Data Plane & Resource Management (PDPRM) functional components. Figure 7 shows its functional components.

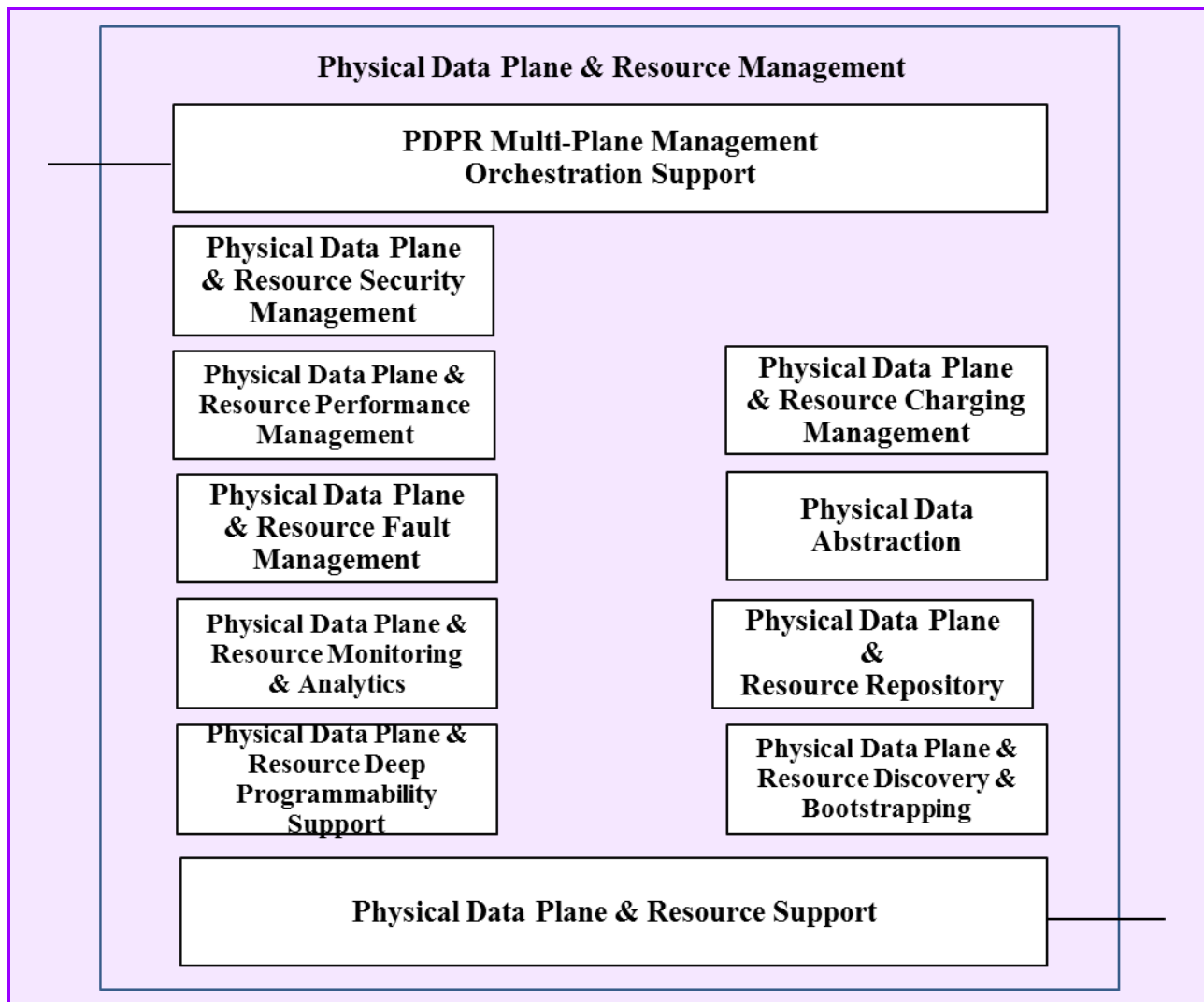


Figure 7 – Physical Data Plane & Resource Management functional components

Physical Data Plane & Resource Support

The Physical Data Plane & Resource Support (PDPR-S) functional element provides a standard interface to the PDPR Management Support and Softwarization in the Resource Plane for requesting and receiving management operations and associated information in ASP.

Physical Data Plane & Resource Plane Resource Discovery and Bootstrapping

The Physical Data Plane & Resource Discovery and Bootstrapping (PDPR-DB) functional element is responsible for discovering and bootstrapping physical Data Plane & resources and provides capabilities for:

- discovering technology specific physical data plane & resources. The discovered resources are stored in the PDPR resource repository. Note that control layer is responsible for abstract resource discovery which is common across any underlying heterogeneous technology specific physical data plane & resources.
- bootstrapping of physical data plane & resources to make them ready for operation based on the bootstrapping policies.

Physical Data Plane & Resource Repository

The Physical Data Plane & Resource Repository (PDPR-R) functional element is responsible for storing the contents discovered by the PDPR-DB and managing the lifecycle of the contents in the repository and provides capabilities for:

- storing and providing APIs for query the contents discovered by the PDPR-DB
- storing and providing APIs for query the contents generated by the PDPR-RMA
- Lifecycle management of the contents in the repository (e.g., creation by storing, modification, deletion, etc.)

Physical Data Abstraction

The Physical Data Abstraction (PDA) is responsible for generating abstractions of technology specific physical resources into technology independent common information and provides capabilities for:

- converting device dependent resource data into independent abstracted information
- storing abstracted information in PDPR-R and providing APIs to other functional components which need abstraction information

Physical Data Plane & Resource Monitoring and Analytics

The Physical Data Plane & Resource Monitoring and Analytics (PDPR-MA) functional element is responsible for collecting the status and events of physical Data Plane & resources and analyzing them for the purpose of FCAPS and provides capabilities for:

- monitoring the activities, status, anomalous events of the physical data plane & resources in the underlying IMT-2020 networks
- analyzing the monitored data and providing reports on the behavior of the data plane & resources, which can take the form of alerts for behavior which has a time-sensitive aspect (e.g., the occurrence of a fault, the completion of a task), or it can take the form of aggregated forms of historical data (e.g., resource usage data)
- storing and retrieving monitored data and analysis reports as logging records in the Physical Data Plane & Resource Repository functional element.

Physical Data Plane & Resource Fault Management

The Physical Data Plane & Resource Fault Management (PDPR-FM) functional element is responsible for fault management of the PDPR and provides capabilities for:

- detecting anomalous events which cause failure of the underlying physical data plane & resources.
- analyzing a root cause of the failure including the correlated event among physical data plane & resources
- generating failure resolving policies and interact with control and provisioning functional components for the actual healing actions.

Physical Data Plane & Resource Performance Management

The Physical Data Plane & Resource Performance Management (PDPR-PM) functional element is responsible for ensuring performance of the physical data plane & resources including energy-aware data plane & resource management and provides capabilities for:

- monitoring and ensuring performance of the physical data plane & resources based on the given KPIs
- estimating total energy consumption costs of underlying physical data plane & resources (physical nodes and links) with the monitored data plane & resource status information
- Calculating energy efficient optimal data plane & resource mapping based on the current estimated total energy consumption costs and the requested KPI

Physical Data Plane & Resource Security Management functional element

The Physical Data Plane & Resource Security Management (PDPR-SM) functional element is an optional functional element responsible for security management of physical data plane & resources and provides authentication and authorization capabilities and detecting and avoiding anomalous attacks of physical data plane & resources.

Physical Data Plane & Resource Charging Management

The Physical Data Plane & Resource Charging Management (PDPR-CM) functional element is responsible for accounting management of physical data plane & resources and provides capabilities for:

- metering and reporting data plane & resource usage data for charging. Data Plane & Resource usage data can be metered per flow or aggregated flows of physical links

Physical Data Plane & Resource Multi-Plane Management Orchestration Support

The Physical Data Plane & Resource Multi-Plane Management Orchestration Support (PDPR-MMOS) functional element provides an internal interface to the Multi-Plane Orchestration support functional element in the MMOS functional component for requesting and receiving management operations and associated information for multi-layer orchestration specific to resource layer management.

8.2 Virtual Data Plane & Resource Management Functional Component

Virtual Data Plane & Resource Management (VDPRM) functional component manages virtual infrastructure data plane and resources. Management of virtual data plane and resource function packages supports subsequent instantiation of a virtual data plane and resource function and validation. Instantiation of VDPRM is managed based on present mandatory virtual infrastructure information. Validation and authorization of virtual infrastructure resource is handled in response to network service request. Virtual infrastructure resource is managed across operator's infrastructure domain including distribution, reservation, and allocation for network service instances and VDPRM instances. It also collects usage information of virtual infrastructure resources and forward required information for update.

This sub-clause describes detailed management functionality in the VDPRM. Figure 8 shows its functional elements.

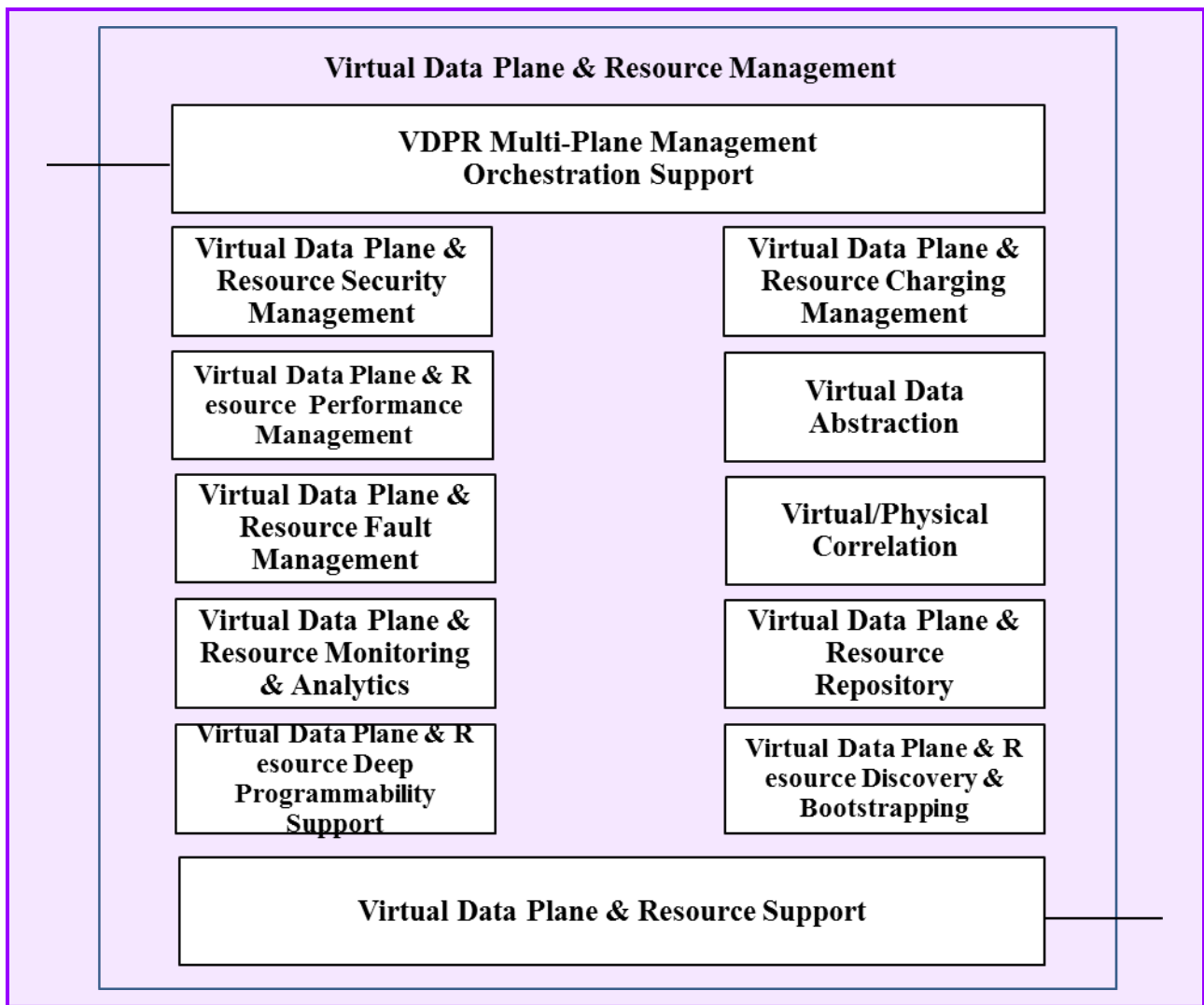


Figure 8 – Virtual Data Plane & Resource Management functional components

Virtual Data Plane & Resource Support

The IMT-2020 Virtual Data Plane & Resource Support (VDPR-S) functional element provides a standard interface to the VDPR Management Support and Softwarization in the Data Plane & Resource Plane for requesting and receiving management operations and associated information in ASP.

Virtual Data Plane & Resource Plane Resource Discovery and Bootstrapping

The Virtual Data Plane & Resource Discovery and Bootstrapping (VDPR-DB) functional element is responsible for discovering and bootstrapping virtual data plane & resources and provides capabilities for:

- discovering technology specific virtual data plane & resources. The discovered data plane & resources are stored in the PDPR resource repository. Note that control layer is responsible for abstract data plane & resource discovery which is common across any underlying heterogeneous technology specific virtual data plane & resources.
- bootstrapping of virtual data plane & resources to make them ready for operation based on the bootstrapping policies.

Virtual Data Plane & Resource Repository

The Virtual Data Plane & Resource Repository (VDPR-R) functional element is responsible for storing the contents discovered by the PDPR-DB and managing the lifecycle of the contents in the repository and provides capabilities for:

- storing and providing APIs for query the contents discovered by the VDPR-DB

- storing and providing APIs for query the contents generated by the VDPR-RMA
- lifecycle management of the contents in the repository (e.g., creation by storing, modification, deletion, etc.)

Virtual Data Abstraction

The Virtual Data Abstraction (VDA) is responsible for generating abstractions of technology specific virtual resources into technology independent common information and provides capabilities for:

- converting device dependent resource data into independent abstracted information
- storing abstracted information in VR-R and providing APIs to other functional components which need abstraction information

Virtual/Physical Correlation

The Virtual/Physical Correlation (VPC) functional element is responsible for correlating the relationship between virtual and physical resources in RP and provides the following capabilities for:

- identifying correlation information among virtual and physical resources in the underlying IMT-2020 networks for efficient provisioning, performance monitoring and fault detection and root-cause analysis
- identifying correlation information between virtual and physical flows for charging purpose
- storing correlation information in a common resource information repository and providing programming interfaces to other functional components which need correlation information

Virtual Data Plane & Resource Monitoring and Analytics

The Virtual Data Plane & Resource Monitoring and Analytics (VDPR-MA) functional element is responsible for collecting the status and events of virtual data plane & resources and analyzing them for the purpose of FCAPS and provides capabilities for:

- monitoring the activities, status, anomalous events of the virtual data plane & resources in the underlying IMT-2020 networks
- analyzing the monitored data and providing reports on the behavior of the resources, which can take the form of alerts for behavior which has a time-sensitive aspect (e.g., the occurrence of a fault, the completion of a task), or it can take the form of aggregated forms of historical data (e.g., resource usage data)
- storing and retrieving monitored data and analysis reports as logging records in the Virtual Data Plane & Resource Repository functional element.

Virtual Data Plane & Resource fault management

The Virtual Data Plane & Resource Fault Management (VDPR-FM) functional element is responsible for fault management of the VDPR and provides capabilities for:

- detecting anomalous events which cause failure of the underlying virtual data plane & resources.
- analyzing a root cause of the failure including the correlated event among virtual data plane & resources
- generating failure resolving policies and interact with control and provisioning functional components for the actual healing actions.

Virtual Data Plane & Resource Performance Management

The Virtual Data Plane & Resource Performance Management (VDPR-PM) functional element is responsible for ensuring performance of the virtual data plane & resources including energy-aware resource management and provides capabilities for:

- monitoring and ensuring performance of the virtual data plane & resources based on the given KPIs

- estimating total energy consumption costs of underlying virtual data plane & resources (virtual nodes and links) with the monitored data plane & resource status information
- Calculating energy efficient optimal data plane & resource mapping based on the current estimated total energy consumption costs and the requested KPI

Virtual Data Plane & Resource Security Management functional element

The Virtual Data Plane & Resource Security Management (VDPR-SM) functional element is an optional functional element responsible for security management of virtual data plane & resources and provides authentication and authorization capabilities and detecting and avoiding anomalous attacks of virtual data plane & resources

Virtual Data Plane & Resource Charging Management

The Virtual Data Plane & Resource Charging Management (VDPR-CM) functional element is responsible for accounting management of virtual data plane & resources and provides capabilities for;

- metering and reporting data plane & resource usage data for charging. Data Plane & Resource usage data can be metered per flow or aggregated flows of virtual links

Virtual Data Plane & Resource Multi-Plane Management Orchestration Support

The Virtual Data Plane & Resource Multi-Plane Management Orchestration Support (VDPR-MMOS) functional element provides an internal interface to the Multi-Plane Orchestration support functional element in the MMOS functional component for requesting and receiving management operations and associated information for multi-layer orchestration specific to data plane & resource layer management.

8.3 Control Plane Management Functional Component

This following sub-clause describes detailed management functionality in the Control Plane Management (CPM) functional component. Figure 9 shows its functional elements.

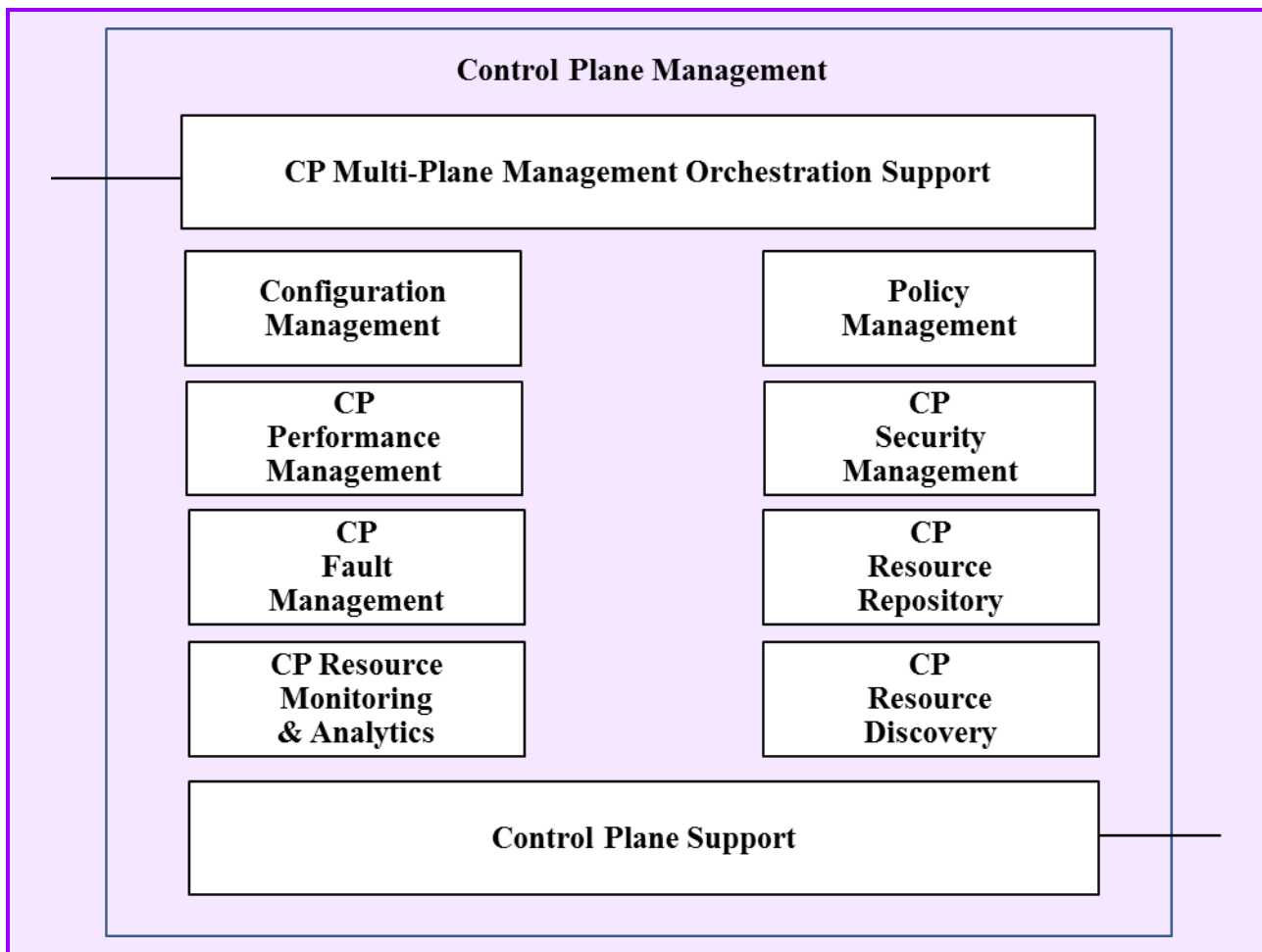


Figure 9 – Control Layer Management functional components

Control Plane Support

The Control Plane Support (CP-S) functional element provides a standard interface to the CP Management support, Softwarization & Orchestration for requesting and receiving management operations and associated information in CP.

Control Plane Resource Discovery

The Control Plane Resource Discovery (CP-RD) functional element is responsible for discovering control resources in the CP and provides capabilities for:

- discovering control resources in the CP in the managed IMT-2020 networks. The discovered resources are stored in the CP-Resource Repository.

Control Plane Resource Repository

The Control Plane Resource Repository (CP-RR) functional element is responsible for storing the contents discovered by the Data Plane Discovery and Bootstrapping functional element in the DP and managing the lifecycle of the contents in the repository and provides capabilities for:

- storing and providing APIs for query the contents discovered by the CP Resource Discovery functional element
- storing and providing APIs for query the contents generated by the CP Resource Monitoring and Analytics functional element
- lifecycle management of the contents in the repository (e.g., creation by storing, modification, deletion, etc.)

Control Plane Resource Monitoring and Analytics

The Control Plane Resource Monitoring and Analytics (CP-RMA) functional element is responsible for collecting the status and events of CP resources and analyzing them for the purpose of performance, fault, and security management and provides capabilities for:

- monitoring the activities, status, anomalous events of the control resources in the CP of the underlying IMT-2020 networks
- analyzing the monitored data and providing reports on the behavior of the resources, which can take the form of alerts for behavior which has a time-sensitive aspect (e.g., the occurrence of a fault, the completion of a task), or it can take the form of aggregated forms of historical data (e.g., resource usage data);
- storing and retrieving monitored data and analysis reports as logging records in the CP Resource Repository

Control Plane Configuration Management

The Control Plane Configuration Management (CP-CM) functional element is responsible for configuration management of the CP and provides capabilities for:

- provisioning control resources in the CP
- scaling in/out of control resources based on the demand and availability

Control Plane Fault Management

The Control Plane Fault Management (CP-FM) functional element is responsible for fault management of the CP and provides capabilities for:

- detecting anomalous events which cause failure of the CP resources
- analyzing a root cause of the failure of the CP resources
- generating failure resolving policies and interact with control and provisioning functional components for the actual healing actions.

Control Plane Performance Management

The Control Plane Performance Management (CP-PM) functional element is responsible for ensuring performance of the CP resources in the CP and provides capabilities for monitoring and ensuring performance of the CP resources based on the given KPIs.

Control Plane Security Management

The Control Plane Security Management (CP-SM) functional element is an optional functional element is responsible for security management of CP and provides capabilities for:

- providing authentication and authorization capabilities of CP
- detecting and avoiding anomalous attacks towards CP
- storing and providing APIs for query the contents discovered by the CP-RD
- storing and providing APIs for query the contents generated by the CP-RMA
- lifecycle management of the contents in the repository (e.g., creation by storing, modification, deletion, etc.)

Control Plane Policy Management

The Control Plane Policy Management (CP-PoM) functional component provides capabilities to define, store and retrieve policies that apply to CP-services. Policies can include business, technical, security, privacy and certification policies that apply to CP-services and their usage by IMT-2020 applications.

Some policies can be general and apply to a CP-service irrespective of the IMT-2020 application concerned. Other policies can be specific to a particular IMT-2020 application.

Control Plane Multi-Plane Management Orchestration Support

The Control Plane Multi-Plane Management Orchestration Support (CP-MMOS) functional element provides an internal interface to the Multi-Plane Orchestration support functional element in the MMOS functional component for requesting and receiving management operations and associated information for multi-layer orchestration specific to control layer management.

8.4 Application and Service Management Functional Component

Service management monitors end-to-end network service utilization aspects such as bandwidth, latency etc. in response to customers request. Service management is established by BSS/OSS in response to customer request. OSS assembles services and assures network performance to customers. BSS manages accounts and payment with customer support and service modification. Customer KPI is established and managed through service management. It enables the creation, operation, and control of multiple dedicated communication service networks running on top of an IMT-2020 infrastructure. A unified end-to-end service management ensures compatibility and flexibility of the operation and management for IMT-2020.

Network management only needs to forward the current status report of network management systems to BSS/OSS in order to meet customers' needs. Service management shall also be included in management. This principle shall remain the same with the ongoing softwarization trend. There is definitely a need for an interface between BSS and OSS in order to exchange the information within the management plane. Therefore, service management should be included in management with interface to BSS/OSS for network service management.

The following sub-clause describes detailed management functionality in the Application and Service Plane Management (ASPM) functional component. Figure 10 shows its functional elements.

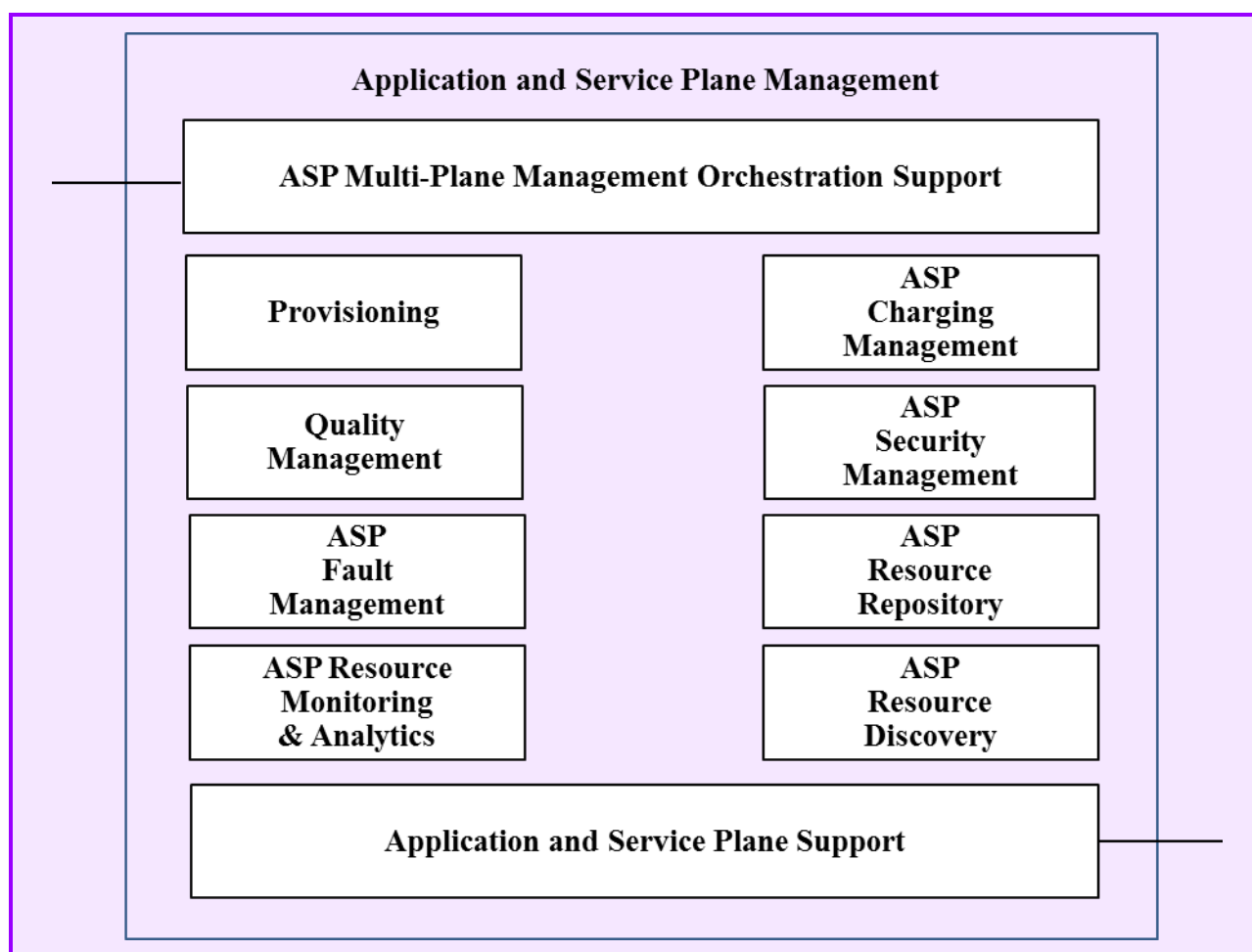


Figure 10 – Application and Service Plane Management functional component

Application and Service Plane Support

The IMT-2020 Application and Service Plane Support (ASP-S) functional element provides a standard interface to the ASP Management Support, Softwarization & Orchestration for requesting and receiving management operations and associated information in ASP.

Application and Service Plane Resource Discovery

The Application and Service Plane Resource Discovery (ASP-RD) functional element is responsible for discovering application and service resources in the ASP and provides capabilities for discovering application and service resources in the ASP of the managed IMT-2020 networks. The discovered resources are stored in the ASP Resource Repository.

Application and Service Plane Resource Monitoring and Analytics

The Application and Service Plane Resource Monitoring and Analytics (ASP-RMA) functional element is responsible for collecting the status and events of ASP resources and analyzing them for the purpose of fault, quality, and security management and provides capabilities for:

- monitoring the activities, status, anomalous events of the application resources in the ASP of the underlying IMT-2020 networks
- analyzing the monitored data and providing reports on the behavior of the resources, which can take the form of alerts for behavior which has a time-sensitive aspect (e.g., the occurrence of a fault, the completion of a task), or it can take the form of aggregated forms of historical data (e.g., resource usage data)

- storing and retrieving monitored data and analysis reports as logging records in the ASP Resource Repository.

Application and Service Plane Resource Repository

The Application and Service Plane Resource Repository (ASP-RR) functional element is responsible for storing the contents discovered by the ASP Resource Discovery and managing the lifecycle of the contents in the repository and provides capabilities for:

- storing and providing APIs for query the contents discovered by the ASP Resource Discovery.
- storing and providing APIs for query the contents generated by the ASP Resource Monitoring and Analytics.
- lifecycle management of the contents in the repository (e.g. creation by storing, modification, deletion, etc.)

Application and Service Plane Provisioning

The Application and Service Plane Provisioning (ASP-P) functional element is responsible for provisioning applications and services in the ASP and provides capabilities for:

- provisioning application and service in the ASP. The application and service provisioning will trigger the ASP orchestration operation which will further trigger the CP orchestration and eventually allocating requested resources in the DP.
- mapping and translating customer's high-level application/service provisioning profile into technology-aware provisioning policies
- managing provision policy lifecycle.

Application and Service Plane Fault Management

The Application and Service Plane Fault Management (ASP-FM) functional element is responsible for fault management of the ASP and provides capabilities for:

- detecting anomalous events which cause failure of the ASP resources.
- analyzing a root cause of the failure of the ASP resources
- generating failure resolving policies and interact with control and provisioning functional components for the actual healing actions.

Application and Service Plane Quality Management

The Application and Service Plane Quality Management (ASP-QM) functional element is responsible for ensuring performance of the ASP resources in the ASP and provides capabilities for:

- monitoring and ensuring quality of the ASP application and service resources based on the given KPIs

Application and Service Plane Security Management

The Application and Service Plane Security Management (ASP-SM) functional element is responsible for security management of ASP and provides capabilities for

- providing authentication and authorization capabilities of ASP
- detecting and avoiding anomalous attacks of ASP

Application and Service Plane Charging Management

The Application and Service Plane Charging Management (ASP-CM) functional element is responsible for accounting management of ASP and provides capabilities for metering and reporting application and service resource usage data for charging. Resource usage data can be metered per application/service or per end-user/customer.

Application and Service Plane Multi-Plane Management Orchestration Support

The Application and Service Plane Multi-Plane Management Orchestration Support (ASP-MMOS) functional element provides an internal interface to the Multi-Plane Management Orchestration Support functional element in the IMT-2020-MP for requesting and receiving management operations and associated information for multi-plane management orchestration specific to application layer management.

8.5 Multi-Plane Management Orchestration and Slice Lifecycle Management Support Functional Component

Orchestrator manages automated arrangement, coordination, and management of both physical and virtual network, control and service resources. Physical and virtual network resources are orchestrated in a manner best suited to match the network constraints specified by other management plane components. Dynamic resource selection is determined based on the current physical and virtual network management status information. It ensures optimized allocation of the necessary resources and connectivity for the best suited service information from service management. Based on resource availability and load, it coordinates related resources to assure the network management direction. Therefore, it plays the key role in management plane for IMT-2020.

This sub-clause describes detailed management functionality in the Multi-Plane Management Orchestration and Slice Lifecycle Management (MMO-SLM) functional component. Figure 11 shows its functional elements.

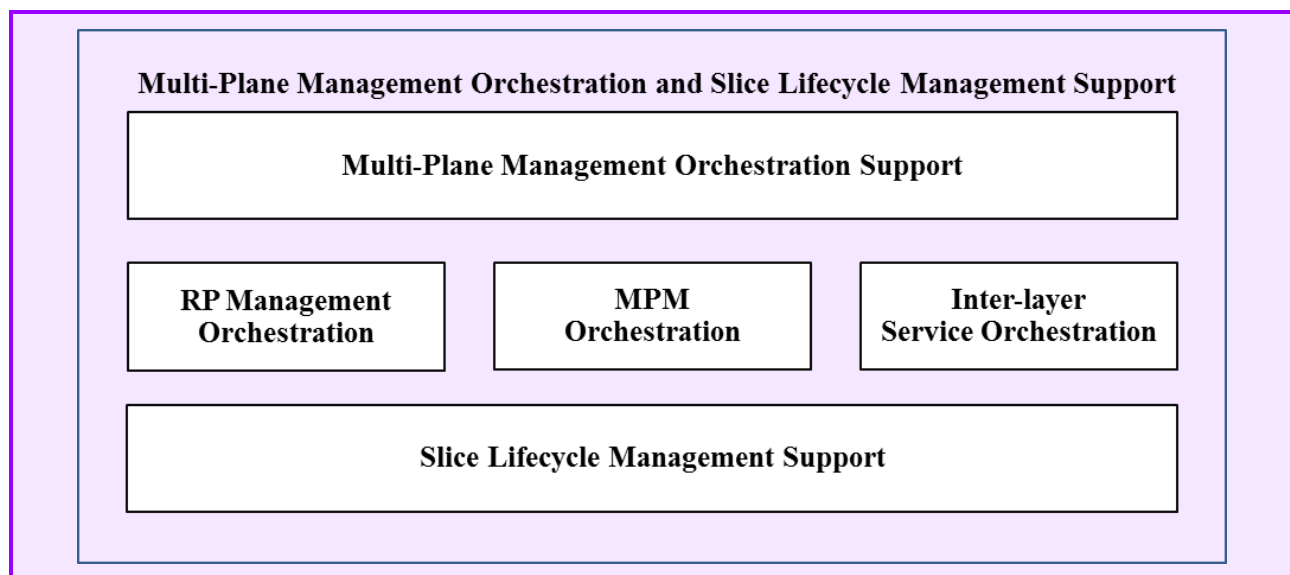


Figure 11 – Functional elements of Multi-Plane Management Orchestration

Multi-Plane Management Orchestration Support

The Multi-Plane Management Orchestration Support (MMOS) functional element provides an internal interface to the Multi-Plane Orchestration Support functional element in the ASPM, CPM, RPM functional components for requesting and receiving management operations and associated information for multi-layer orchestration.

Virtual/Physical Resource Management Orchestration

The Virtual/Physical Resource Management Orchestration (VPR-MO) functional element provides orchestration of the coordinated virtual and physical resources provisioning and configuration.

Multi-Plane Management Functions Orchestration

The Multi-Plane Management Functions Orchestration (MMFO) functional element provides functionality for supporting the lifecycle management of IMT-2020 application/network services across the entire IMT-2020

operator's domain (e.g., multiple IMT-2020 access and core networks, Data Centers interconnected by a WAN transport network, etc.).

Inter-layer Service Orchestration

The Inter-layer Service Orchestration (ISO) functional element provides orchestration of multi-layer resource management. It coordinates management operations among application, control, and resource layers, especially relationship among virtualized and physical resource across multi-layer scope. Some examples can be: orchestration for a multi-layer virtual to physical resource fault correlation, orchestration for scale-in and scale-out of control element (e.g., controller instances) depending on the traffic demand changes in the underlying resource layer, and orchestration for an application layer service provisioning request to resource layer relevant resources.

Slice Lifecycle Management Support

The Slice Lifecycle Management Support (SLMS) functional element provides support functionality of the slice lifecycle management functional component (SLMFC). It communicates with SLMFC to receive requests from SLMFC on specific management requests and respond with results of the requested management operation. Examples of such operations are the current slice resource status, performance statistics, any fault or security related events, and etc.

8.6 External Relationship Management Functional Component

This sub-clause describes detailed management functionality in the External Relationship Management (ERM) functional component. Figure 12 shows its functional elements.

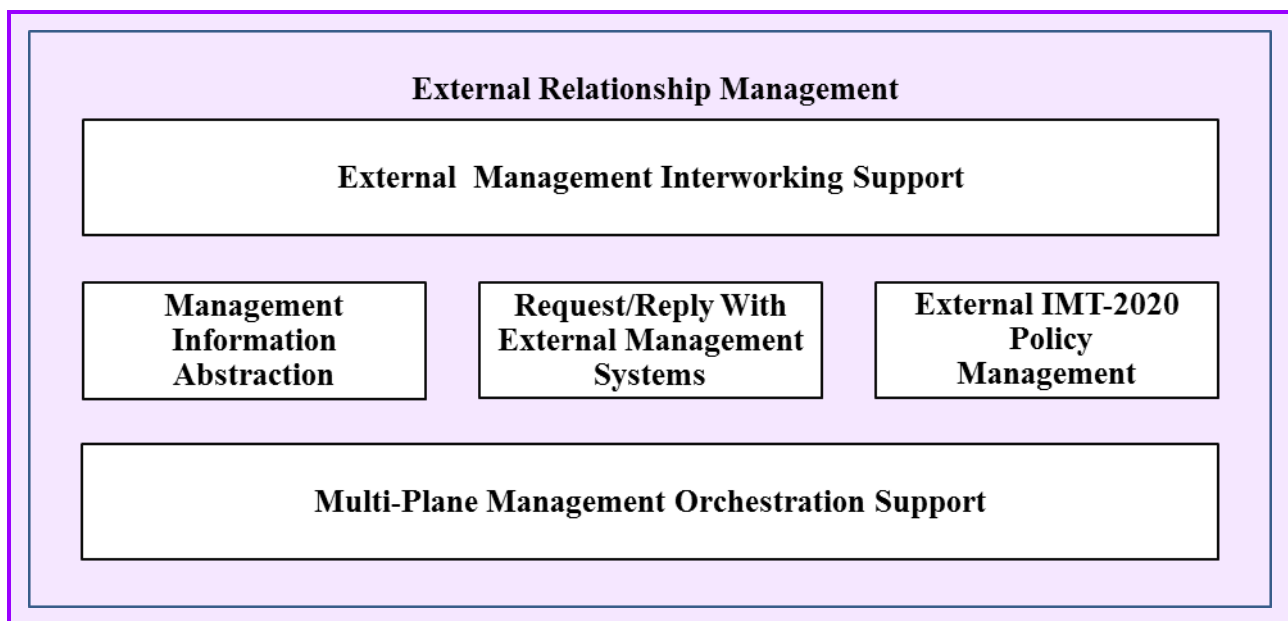


Figure 12 – External Relationship Management functional components

The External Relationship Management provides management functionality to interwork with external management entities. External management entities can be a 2G/3G/LTE OSS/BSS, MANO, cloud management entities, or other management functionality which can be defined in the future. The ERM plays a role of the representative interface of IMT-2020 management toward the external management entities and IMT-2020 management planes in other IMT-2020 domains. Its main functionality includes abstraction of IMT-2020 management information for the exchange and request/reply of management operations with external management entities. It can be used for external IMT-2020 policy management, data analytics, charging, etc.

ERM is also responsible for MMF to interact with an external DevOps system to enable efficient development of IMT-2020 functionality by providing developer environment setup processes, build and test management, and deployment.

External Management Interworking Support

The External Management Interworking Support (EMIS) functional element provides a standard interface to an external OSS/BSS or a IMT-2020-MP in other IMT-2020 domain for requesting and receiving management operations and associated information.

Management Information Abstraction

The Management Information Abstraction (MIA) functional element provides abstraction of IMT-2020 management information for the exchange with external management entities or IMT-2020-MP in other IMT-2020 domains for inter-domain management information hiding purpose.

Request/Reply with External Management

The Request/Reply with External Management (RREM) functional element provides functionality associated with request/reply management operations with external management entities.

External IMT-2020 Policy Management

The External IMT-2020 Policy Management (E5PM) functional element provides external IMT-2020 policy exchanges involved between IMT-2020-MP and external management entities, data analytics, charging, and interaction with an external DevOps system to enable efficient development of IMT-2020 functionality by providing developer environment setup processes, build and test management, and deployment.

Multi-Plane Management Orchestration Support

The Multi-Plane Management Orchestration Support (ERM-MMOS) functional element provides an internal interface to the Multi-Plane Orchestration Support functional element in the Multi-Plane Management Orchestration functional component for the purpose of inter-domain orchestration between IMT-2020-MP and external OSS/BSS and/or IMT-2020-MP in other IMT-2020 domains.

9 IMT-2020 Management Procedure and Implementation Scenarios

As stated in section 6 Overview, end-to-end network management principles are discussed in conventional aspect and IMT-2020 specific aspect.

9.1 IMT-2020 Management Procedure

This sub-clause describes a Slice Lifecycle Management procedure which is illustrated in Figure 13.

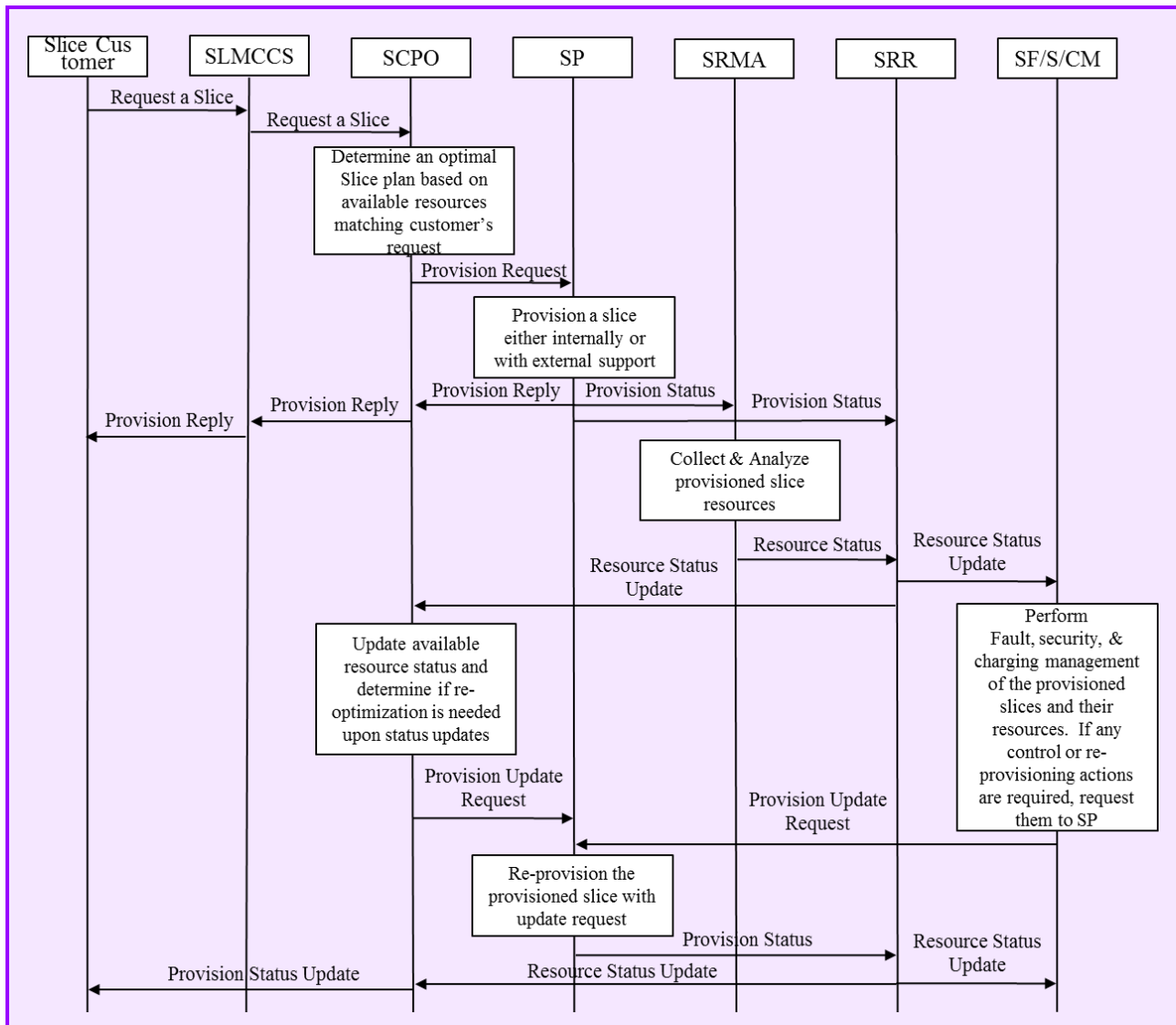


Figure 13 – Slice Lifecycle Management Procedure

- The IMT-2020 Customer requests a slice to be provisioned with its requirements specified. The template for the slice provision is provided by the IMT-2020 service provider as a form of IMT-2020 slice service catalogue through its service portal.
- IMT-2020 Slice Lifecycle Customer Care Support (SLMCCS) functional element receives the customer's request and carries it to the Slice Capacity Planning & Optimization functional element (SCPO). SCPO then determines an optimal slice plan based on available resources which matches customer's request. The detailed optimization algorithms is out of scope of this document.
- Once the provisioning policy is determined, SCPO requests provisioning to Slice Provisioning (SP) functional element. SP then performs the requested slice provisioning task. It involves various sub-tasks. If the provisioning functionality is fully supported by SLM functional component, the whole tasks are performed internally. If not, SP then interacts with external slice provisioning related functional entities such as MANO Orchestrator, SDN controller, etc. When SP interacts with external management entities, it uses External Management Entity Support (EMES) functional element. Upon completion of the provisioning process, SP sends provision reply message to the customer via SLMCCS. At the same time, it sends provision status Slice Resource Monitoring and Analytics (SRMA) functional element to initiate collection and monitoring of the provisioned resources. It also sends the status update to Slice Resource Repository (SRR) to store the provisioned resource information.

- SRMA performs collection, monitoring, and analysis tasks of the provisioned slice resources. Data and information collected and analyzed are then store in SRR for further processing by other functional elements.
- When SRR receives any resource status updates, it stores in the repository and, at the same time, it emits notification to all functional elements who are listening to the status updates. In this case, it sends its update notification to Slice Fault Management (SFM), Slice Security Management (SSM), Slice Charging Management (SCM), and SCPO.
- When SCPO receives the notification, it updates available resource status and determine if re-optimization is needed upon status updates. Also SF/S/CM receive the notification, they perform fault, security, & charging management of the provisioned slices and their resources and determines any control or re-provisioning actions are required. If so, they request to SP for provisioning update processes.
- SP, by receiving the provisioning update requests, performs re-provisioning tasks for the provisioned slices. When re-provisioning tasks are done, SP generates provision status to SRR and SRR further conveys the notification to SF/S/CM, SCPO and IMT-2020 slice customer for resource status updates.

9.2 IMT-2020 Management Implementation Scenarios

9.2.1 eNMS Architecture for IMT-2020 network management

Technologies for implementing new network functions and deploying new networks using them are the fundamental building blocks for IMT-2020, but brand-new network management technologies should be supported with the same importance to make the new networking technology working properly. In order to manage the highly flexible and scalable IMT-2020 networks, traditional network management architecture needs to be enhanced fundamentally.

IMT-2020 heavily relies on the virtualization technologies to acquire the flexibility and scalability. It will use a lot of virtual resources along with the mixed connections among virtual and physical resources. Efficient management of virtual resources will become one of the most challenging goals for IMT-2020 NMS.

Because of the special type of services, mostly requiring low latency, many network functions are moving down to the edge networks. The network functions at the edge need to be managed locally to meet the requirements.

We have investigated IMT-2020 service categories to derive the fundamental requirements for IMT-2020 network management and the results of the investigations have been published on ITU-T library. Based on the insights from the above investigations, a conceptual IMT-2020 network management architecture that is scalable and flexible is considered.

The IMT-2020 service scenarios have been investigated to obtain the requirements for IMT-2020 network management. The service scenarios are categories into three groups according to most IMT-2020 related documents, including ITU-T. We have derived three fundamental requirements for the three service categories:

- eMBB (enhanced Mobile BroadBand)
- Multiple high-speed connections maintained in a flexible way → Scalability
- mMTC (massive Machine Type Communication)
- Numerous devices, connections and conjunction points → Recursive
- URLL (Ultra Reliable and Low Latency Communications)

Network functions moved down near to user → Decentralized

This Recommendation describes following Architecture for IMT-2020 network management, called eNMS (enhanced NMS), according to the above requirements.

Figure 14 shows the basic eNMS architecture frame. eNMS consists of traditional network management modules, orchestration type of modules, and the interfacing modules. These modules are to be run on the virtual machines in order to make the architecture scalable.

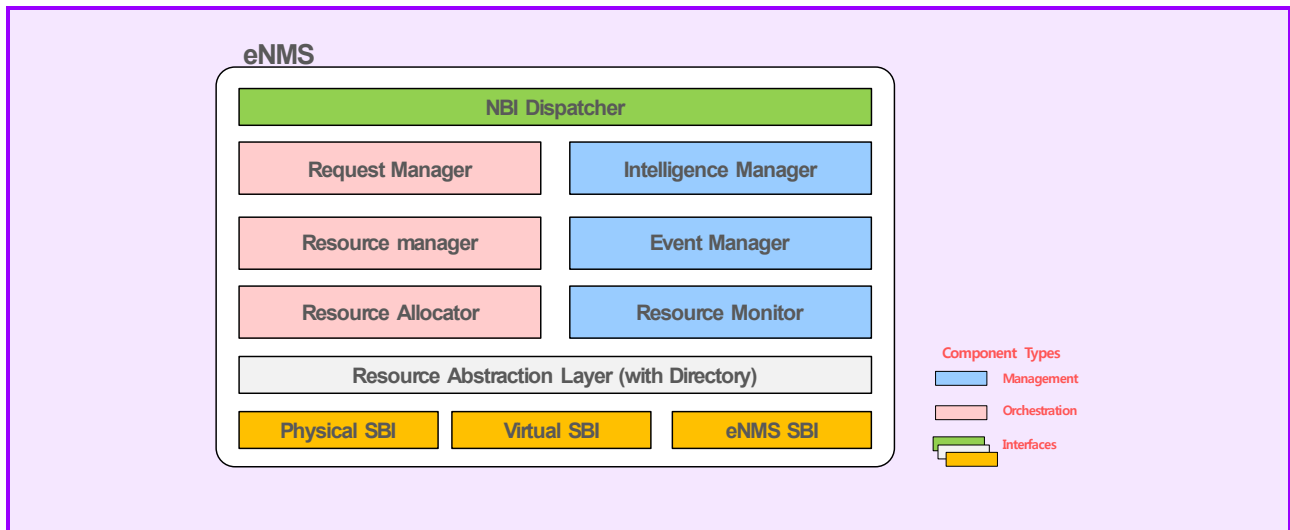


Figure 14 – eNMS Architecture Frame

eNMSs can be deployed many places in the network as it needed and their cooperation is possible by the “eNMS SBI” in Figure 14.

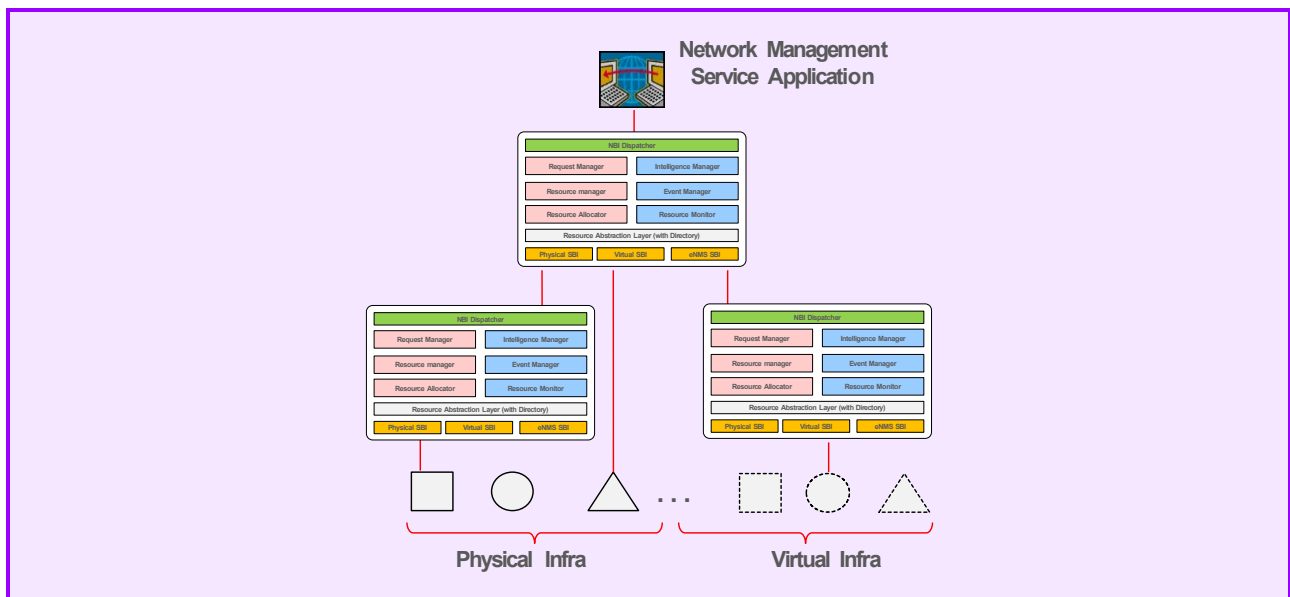


Figure 15 – eNMS recursive deployment

Figure 15 shows an example of eNMS deployment which is recursive. The eNMS SBI of one eNMS instance will be interfacing to the NBI dispatcher of the other instance taking care of different domain.

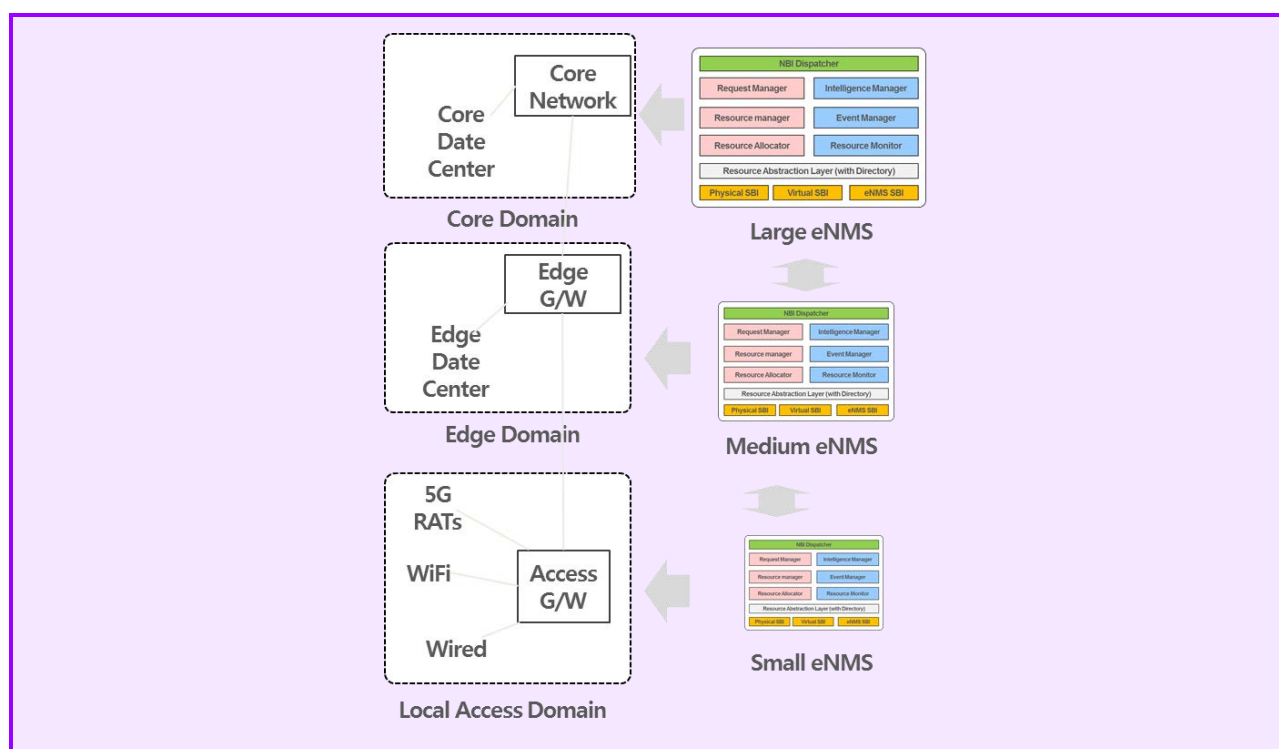


Figure 16 – Multiple eNMSs for different domains

Figure 16 explains the way eNMSs can manage the various domains. The size of the eNMS will be set up based on the size of the domain, but their structures are basically same. They will cooperate each other to provide one view to the other systems and operators working out of eNMS.

9.3 Virtual Network Management

9.3.1 Implementation scenario of multi operators

9.3.1.1 Multi Service Control and Management

One of the main design objective is to increase the flexibility and programmability of IMT-2020 networks with a novel Service Development Kit, a novel, modular Service Platform and Service Orchestrator, and a novel Integrated Infrastructure Management. It will bridge the gap between telecom business needs and operational management systems. The expected key functionality and systems are represented by the Service Development Kit, the Management System and the Service Platform including: a customizable Service Orchestrator, a Resource Orchestrator, a Service Information Base along with various Enablers as represented in Figure 17. This figure covers the Multi-service Control layer, the Integrated Management and Operation layer and the Application and Business Services Layer. This figure also shows the heterogeneity of the physical resources underlying the IMT-2020 infrastructures and related IMT-2020 network segments: radio networks, access networks, aggregation networks, core networks, software networks, data center networks and mobile edge computing clouds. A Multi-Service Control layer is responsible for the creation, operation, and control of multiple dedicated communication network services running on top of a common infrastructure. Functionality for this layer includes: infrastructure abstraction; infrastructure capability discovery; catalogues and repositories; a large number of service and resource orchestration functions such as plugins; information management functionality; and enablers for automatic re-configuration of running services (i.e. part of the integrated management layer). It interworks with a Business Function Layer that maintains IMT-2020 application-related functions, organized in Repositories, and DevOps tools necessary for the creation and deployment of services. Functionality for this layer includes DevOps functionality: Catalogues, Monitoring data analysis tools, testing tools, Packaging tools, Editors and primitives for Application & Service programmability. Figure 17 depicts the way in which IMT-2020 manages various underlying systems.

In conclusion, one of the main design objectives in IMT-2020 Networking and IMT-2020 Multi-Service Control & Management is efficient integration of service programmability, domain orchestration functionality and DevOp functionality. This will maximize the predictability, efficiency, security, and maintainability of operational processes.

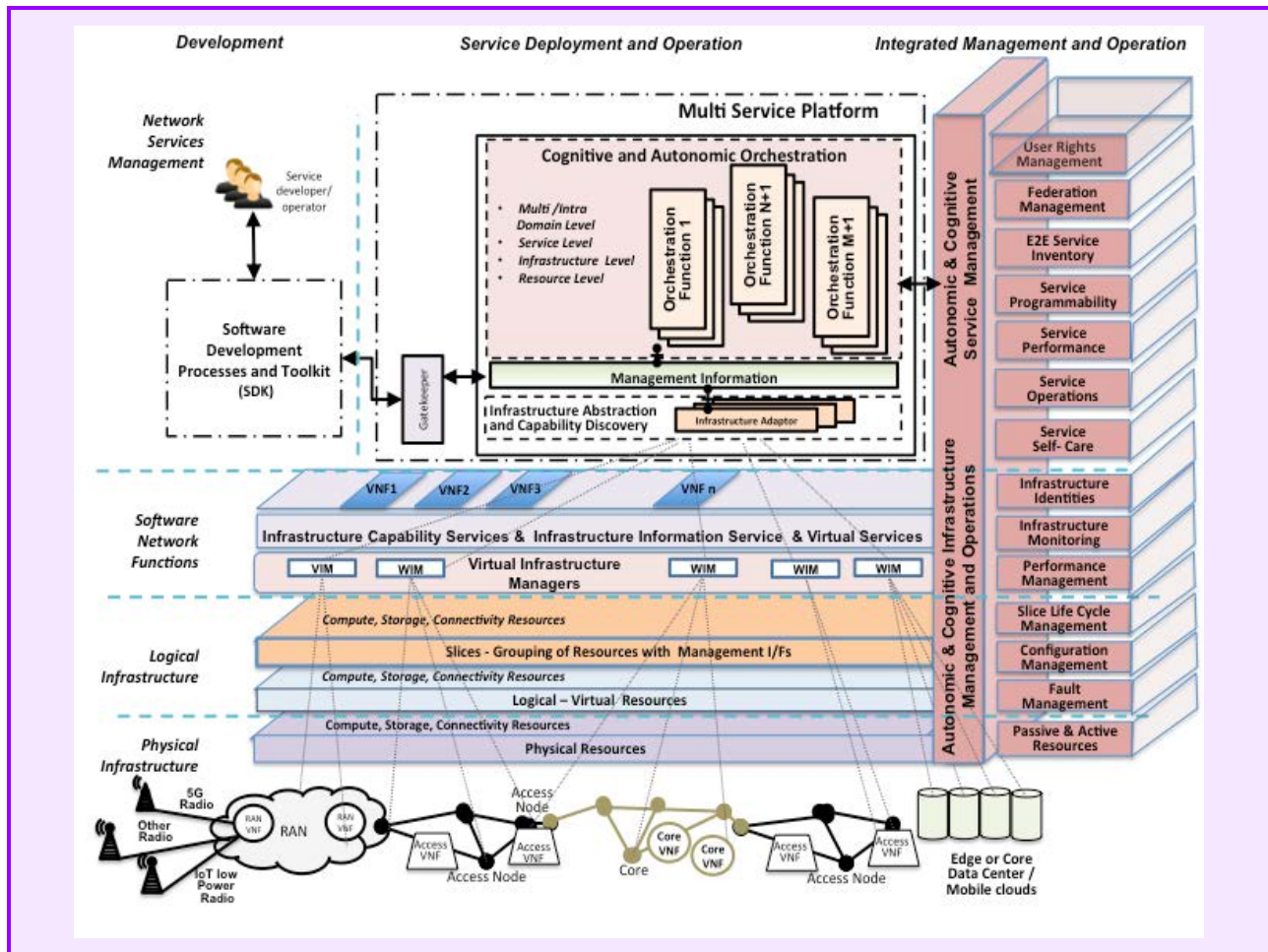


Figure 17 – IMT-2020 Service & Network Management and Orchestration Architecture

9.3.1.2 Multi-Domain Management and Orchestration Architecture

Multi-domain orchestration refers to the automated management of services and resources in multi-technology (multiple domains involving different cloud and networking technology) and multi-operator (multiple administrative domains) environments. The scope of the end-to-end multi-domain management and orchestration plane involves diverse concepts summarized in Figure 18. Figure 18 represents the reference architectural framework for organizing the components and interworking interfaces involved in end-to-end management and orchestration in multi-domain environments. At the lower layer there are resource domains, exposing resource abstraction on interface I5. Domain orchestrators perform resource orchestration and/or service orchestration exploiting the abstractions exposed on I5 by resource domains.

A Multi-domain Orchestrator (MdO) coordinates resource and/or service orchestration at multi-domain level, where multi-domain may refer to multi-technology (orchestrating resources and/or services using multiple domain orchestrators) or multi-operator (orchestrating resources and/or services using domain orchestrators belonging to multiple administrative domains). The Resource MdO belonging to an infrastructure operator, for instance operator A, interacts with domain orchestrators via interface I3 APIs to orchestrate resources within the same administrative domains. The MdO interacts with other MdOs via interface I2-R APIs (business-to-business, B2B) to request and orchestrate resources across administrative domains. Resources are exposed at service orchestration level on interface SI-Or to Service MdOs. Interface I2-S (B2B) is used by Service MdOs to orchestrate services across administrative domains. Finally the Service

MdOs expose on interface I1 service specification APIs (Customer-to-Business, C2B) that allow business customers to specify their requirements for a service. The framework also considers MdO service providers, such as D in Figure 18, which do not own resource domains but operate a multi-domain orchestrator level to trade resources and services.

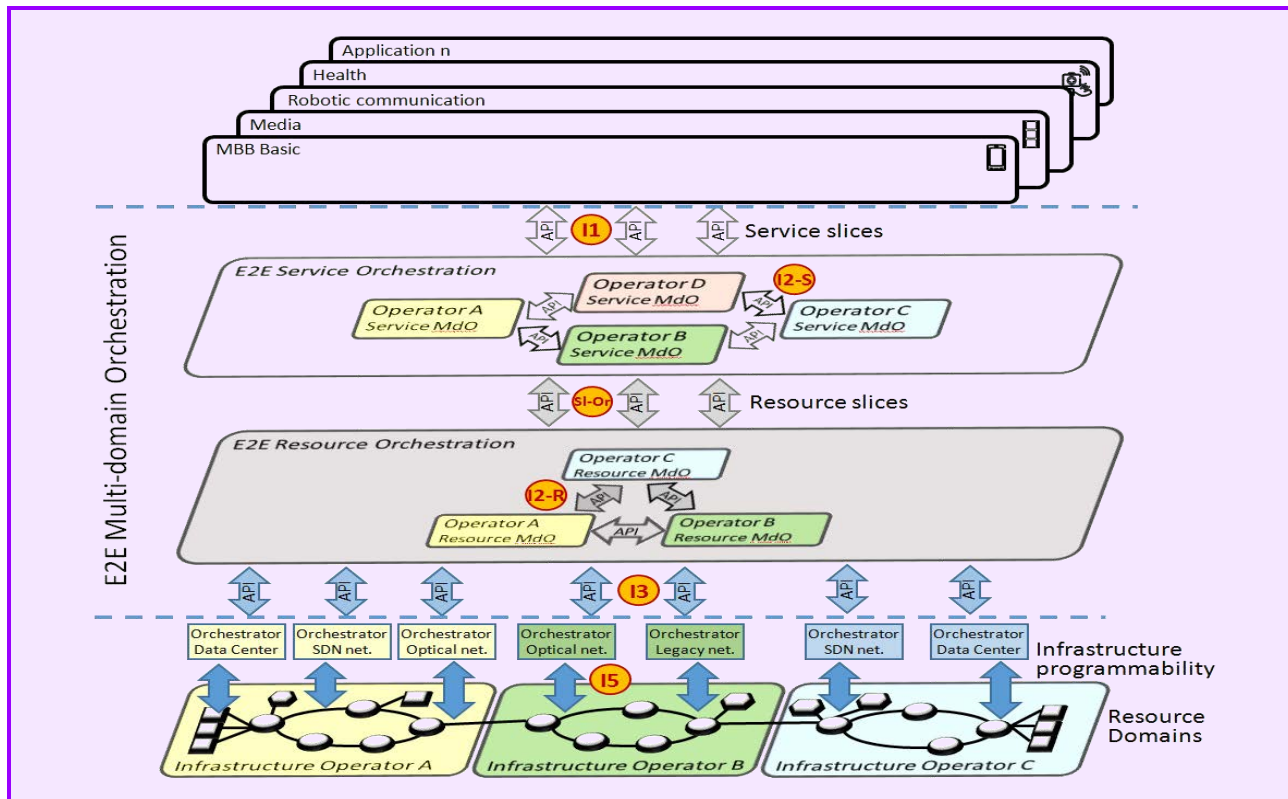


Figure 18 – E2E – Multi-Domain Orchestration of different infrastructure domains belonging to different operators

9.4 Integrated Network Management

Currently, there are two management network systems for managing and control the telecommunication management. First, ITU-T TMN for fixed network, the second one is 3gpp telecommunication management for mobile network.

TMN has been developed in many ways over the past 20 years for fixed network. ITU-T developed TMN architecture to define a framework for the management of telecommunications networks and services in order to control of the network operations as well as to reduce the OPEX and CAPEX.

Recent growth in technologies of the TMN has created complex and heterogeneous network management environments. To manage network devices, elements and services, standard management platforms have been developed using several management protocols such as SNMP and CMIP. However, ITU-T TMN does not define a framework for managing mobile network.

3rd generation partnership project (3GPP) defined requirements of telecommunication management for LTE, 3G and 2G by re-using existing relevant standards such as ITU-T and TMF etc.

With current TMN standard for fixed network and mobile network management technology, it is difficult to build up unified network management system. Based on this understanding it is proposed a new network management framework for IMT-2020.

A standardized network management system for IMT-2020 should operate as an open network management protocol for fixed and mobile network, as shown in Figure 19.

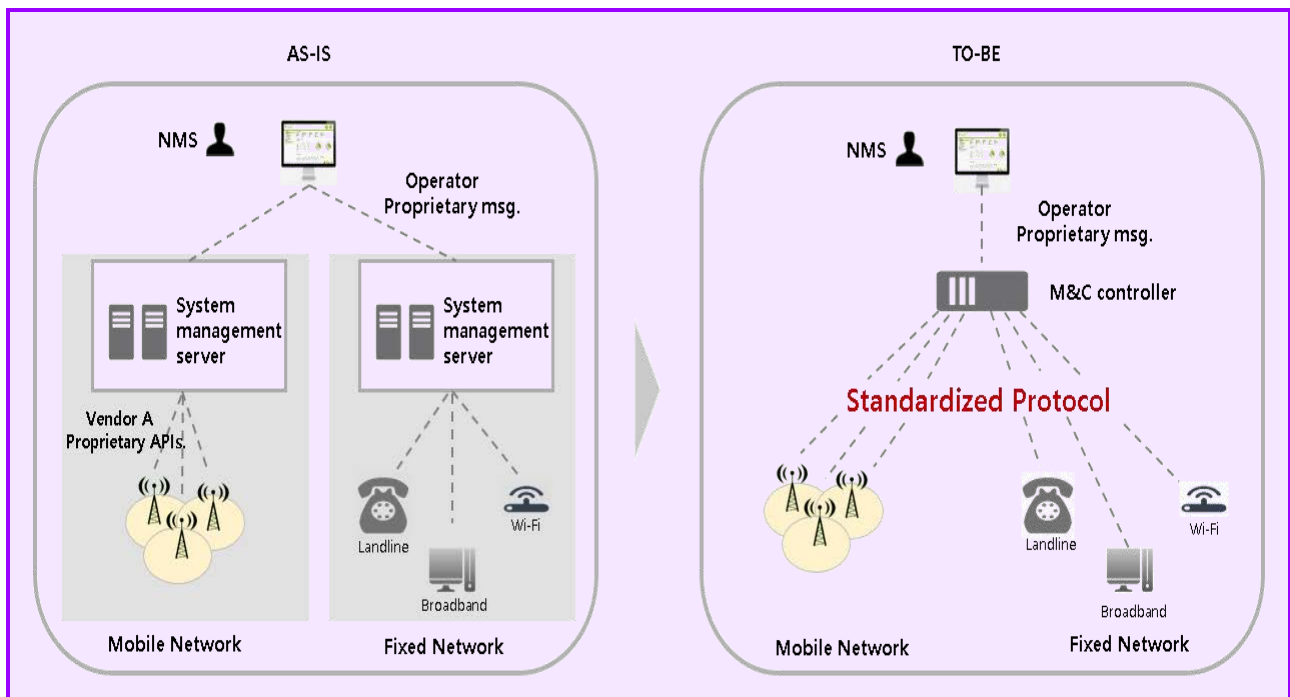


Figure 19 – Standardized network management system

In order to build up a standardized network management system, it is necessary to introduce eNMS accommodating both fixed and mobile network, as shown in Figure 20. Support of legacy network equipment using IWF (Interworking Functions) should be developed with relevant management object. Fault reporting from lower layer network element to higher layer should be supported as well.

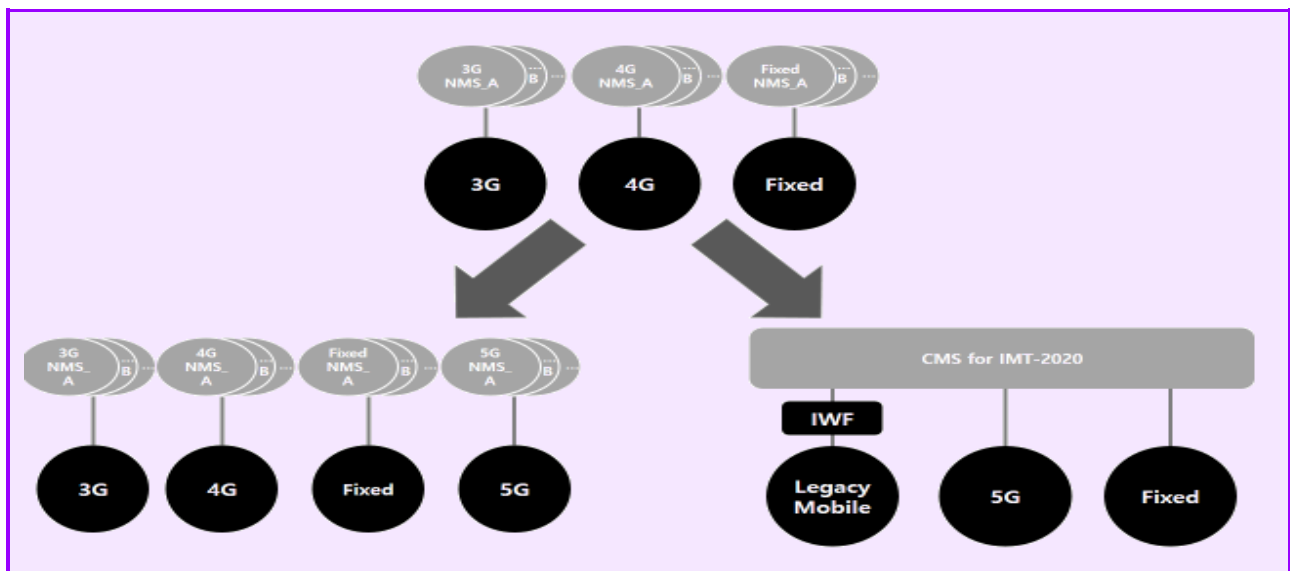


Figure 20 – Integrated Network Management Scenario

Instead of creating a separate network management system just for IMT-2020, eNMS can fully host both legacy network equipment as well as new IMT-2020 network functions. All components of the network are supported by eNMS including different RATs and the fixed network. Operators can have common management viewpoint regardless of connection type. By deploying eNMS, network operation complexity is simplified. eNMS is fully aware of all network equipment status such that it can optimize network functions without management signaling burden. This results in cost saving. Therefore, eNS can save OPEX/CAPEX of network operators.

The eNMS management architecture is based on ITU-T TMN, and will reuse some of TMN functions, method and interfaces that are already defined and suitable for eNMS.

The technical contents of this output document has reached more than 70% majority except the following aspects:

- Defining reference points in slice lifecycle management and slice instance management functional architecture;
- Defining additional IMT-2020 management procedures, if any;
- Defining additional IMT-2020 management scenarios; if any.

Thus, the Focus Group recommends to complete the outstanding issues as soon as possible and request for consent as an ITU-T Recommendation.

Appendix I

Contributors (in Alphabetical Order)

(This appendix does not form an integral part of this Recommendation.)

This is the list of all contributors who submitted any written form of comments or contributions.

- Alex Galis, University College London, U.K.
- Hyungsoo Kim, KT
- Jongpil Lee, KT
- Olivia Heeyun Choi, KT
- Sangwoo Kang, KT
- Seongbok Baik, KT

Appendix II

Acknowledgement

(This appendix does not form an integral part of this Recommendation.)

- This work was partially supported the EU H2020 5G PPP projects: 5GEX (“5G Multi-Domain Exchange”; <https://www.5gex.eu/>) and SONATA (“Service Programing and Orchestration for Virtualized Software Networks”; <http://sonata-nfv.eu/>)
- This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0115-16-0001, 5GCHAMPION – 5G Communication with a Heterogeneous, Agile Mobile network in the PyeongChang wInter Olympic competition).





4.

Information centric networking (ICN)

Application of information centric networking to IMT-2020

Summary

This document describes the five Proof Of Concept (PoC) performed for the ITU-T Focus Group on IMT-2020 by several of the participating organizations: Cisco, Fujitsu Labs of America, Huawei, InterDigital, and KDDI. During the 2015, the Focus Group identified 15 ICN related standardization gaps. These PoCs addressed 6 of the 15 gaps. Two additional studies in 2016 addressed 3 additional gaps, so we now have some input on 9 of the 15 gaps. The Introduction discusses the gaps and gap coverage in more detail.

Table of Contents

1	Scope
2	Abbreviations and acronyms
3	Introduction
3.1	ICN Background
3.2	Overview of the five Proof Of Concept (PoC) performed for the ITU-T Focus Group on IMT-2020
4	ICN Enhanced Mobile Video at the Network Edge (Cisco)
4.1	Application to IMT-2020
5	Functional Chaining System in ICN (Fujitsu Labs of America)
5.1	Abstract
5.2	Application to IMT-2020
5.3	Implications for Standardization
6	End-to-end ICN Service Orchestration with Mobility for IMT 2020 (Huawei)
6.1	Abstract
6.2	Towards ICN Standardization
7	IP Services over ICN (InterDigital)
7.1	InterDigital ICN solution within the IMT-2020 network
7.2	InterDigital Solution: IP over ICN
7.3	Benefits of InterDigital ICN solution
7.4	Integration with IMT-2020 NWs
7.5	Further study and possible standardization
8	ICN transport for mmWave Networks (KDDI)
8.1	Background
8.2	Technical Details
9	Bibliography
10	Annex A: Proof-of-Concept Technical Background
10.1	End-to-end ICN Service Orchestration with Mobility for IMT 2020
10.2	IP Services over ICN
10.3	ICN transport for mmWave Networks (KDDI)

ρ roof

θ f

c oncept



1 Scope

This document describes the five Proof Of Concept (PoC) performed for the ITU-T Focus Group on IMT-2020 by several of the participating organizations: Cisco, Fujitsu Labs of America, Huawei, InterDigital, and KDDI.

2 Abbreviations and acronyms

This document uses the following abbreviations and acronyms:

BRAS	Broadband Remote Access Server
CCN	Content Centric Networking
CloudRAN	Cloud Radio Access Network
CDN	Content Delivery Network
COTS	Commercial Off-The-Shelf
CS	Content Store
DASH	Dynamic Adaptive Stream over Http
DDD	Directional Division Duplex
DNS	Domain Name System
eNodeB	Evolved Node B
EUH2020	European Union Horizon 2020
FDD	Frequency Division Duplex
FIB	Forwarding Information Base
FID	Forwarding Identifier
FQDN	Fully Qualified Domain Name
Gbps	Giga bits per second
GHz	Giga Hertz
GW	Gateway
HTTP	Hyper Text Transfer Protocol
ICN	Information Centric Networking
IoT	Internet of Things
MAC	Medium Access Control
MLDR	Mobility Loss Detection and Recovery
mmWave	Millimeter wave
NAP	Network Access Point
NDN	Named Data Networking
NFV	Network Function Virtualization
ONOS	Open Network Operating System
PCE	Path Computation Element
PIT	Pending Interest Table
PoC	Proof of Concept

POINT	iP Over IcN – the better IP
RIFE	architecture for an Internet For Everybody
SDN	Software Defined Networking
SDO	Standard Development Organization
Snap	Source Network Attachment Point
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URI	Universal Resource Identifier
VLAN	Virtual LAN
VM	Virtual Machine
VSER	Virtual Service Edge Router
WLDR	Wireless Loss Detection and Recovery

3 Introduction

3.1 ICN Background

Information Centric Networking (ICN) is a different approach to addressing and framing data than today's Internet Protocol (IP) semantics. In IP, one uses a source and destination address to identify the two endpoints of a packet. The destination is almost always a unicast address and in a small number of cases an anycast address; the use of IP multicast is very limited. Inside the network, the payload of an IP packet is usually an arbitrarily framed byte stream (TCP) or datagrams (UDP). TCP/IP assigns an ephemeral name to each packet: source IP, source port, destination IP, destination port, byte offset, byte length. These names are not reusable, nor cacheable beyond use for retransmission of lost packets. ICN's approach is to assign a re-usable name to each packet or small group of packets. This allows object re-use and peer-to-peer messaging via name without needing to resolve endpoint identifiers beforehand. ICN also bundles object authenticity with the network packets, such as via Merkel signing of a group of packets in a manifest, so provenance stays with the objects even if cached.

There are several ICN architectures in active use today. The most widely known is Content Centric Networking (CCNx) and its offshoot Named Data Networking (NDN). NDN forked from CCNx around 2012. While there are several important protocol differences between NDN and CCNx, they are close enough in function that we will only describe CCNx.

Because ICN does not require resolving endpoint identifiers before using a name, it opens new possibilities in machine-to-machine and IoT applications. Today, IP-based applications must use specialized rendezvous mechanisms, such as link broadcast, multicast, dynamic DNS, multicast DNS, or SIP. This is because they must resolve an IP address for a desired name. ICN technologies remove the IP abstraction so the network can operate at the name level. This can make the network more responsive to application demands with less infrastructure.

Within a IMT-2020 5G RAN, ICN could serve as the object transport for intra-RAN data. For example, the state of a Slice could be stored and transported as ICN objects, so as services move between enodeB sites its state follows in the named ICN objects.

In the ICN technology CCNx, the name combines both a locator and identifier in to one routable hierarchical structure. One could think of it as routing on URIs, where each name segment can be arbitrary binary data not restricted to the URI syntax. At one end of the spectrum are pre-generated content names, such as for a movie. A movie service could name content with a prefix like /movie_service/superman/h264/768kbps/32kbps/English to indicate a codec and encoding rate. Names can identify things beyond static content. A simple example would be a dynamic web service, such as

/book_store/home/<encrypted_account_identifier>, where the <encrypted_account_identifier> is a blob that the book store server can understand and use to generate a custom home page. Names could also indicate a type of calculation, for example /calc/4/2/times could return a content object with the value “8”. In all these examples, we used ASCII names, but in practice name segments can be binary values not necessarily human-readable.

3.1.1 Elements of ICN

An Information Centric Network is usually made up of content producers, content publishers, content replicas, and content consumers. A producer generates a piece of content, such as a document, photo, movie, or web page. It may have its own digital rights management (DRM) attached by the producer. A publisher packages a piece of content for use in the network. This may include pre-encoding the content to certain formats and names and signing them with a network identity. A replica distributes content from a publisher. A consumer fetches content via network names from replicas. The download process at a consumer understands the inherent security offered by the ICN, which usually allows authenticating every packet via direct signature or implicit hash chain from the publisher. This is different than today’s security model, where authenticity derives from a secure connection to a replica. In the simplest configuration, one entity is a producer, a publisher, and a replica for its content.

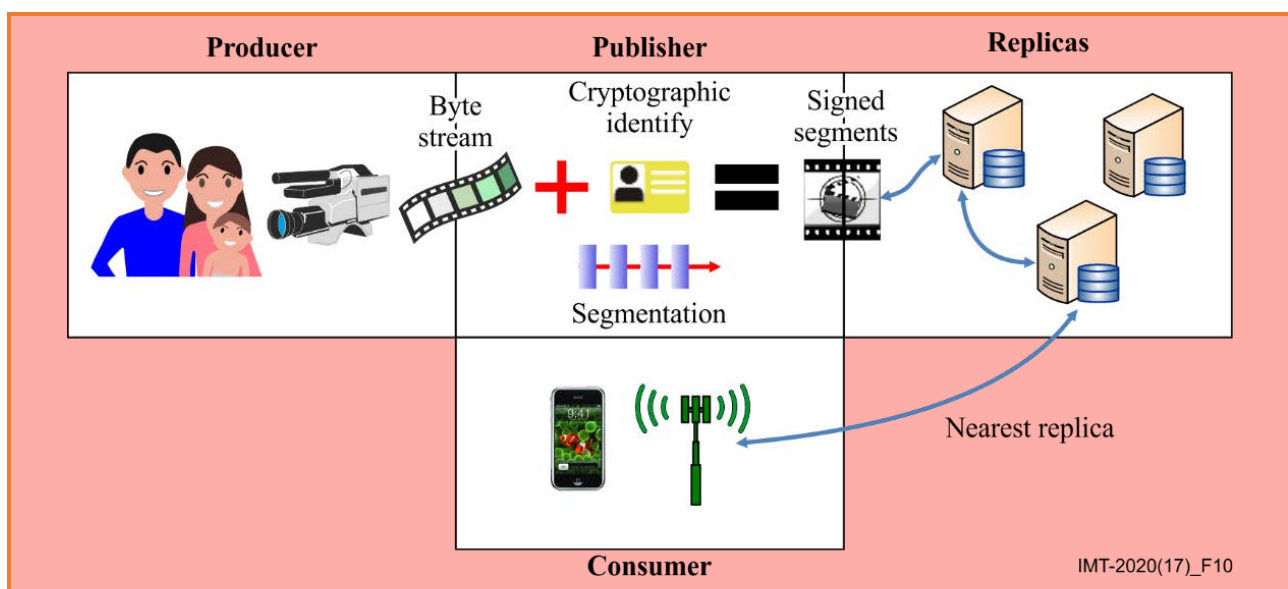


Figure 10 – Typical ICN architecture

Figure 10 illustrates the typical ICN architecture, which we will make concrete by describing how an actual instance of CCNx would handle these activities. In this example, a family videos an activity at home, such as their baby. The video camera produces a structured MP4 byte stream. The publisher function – which may reside on the camera, home gateway, or other device – segments the byte stream to CCNx Content Objects. For a live stream like this, the publisher would segment it to a certain number of video frames in some number of network packets (Content Objects). A CCNx Manifest tree incorporates those Content Objects by hash in to a single signed manifest representing the whole video segment. The video segment is stored on a first replica, such as a home gateway. The publisher updates the movie catalogue to include the new segment, then repeats for the next segment. A consumer queries its nearest replica for the movie catalogue and segments. If the nearest replica does not have it, the request is forwarded towards the publisher until satisfied. The content travels to the consumer, optionally cached at intermediate replicas.

As illustrated in Figure 10, a consumer may fetch the CCNx Content Objects from any replica and still be assured that it is the correct data. This is because the Manifest tree is signed by the publisher and then securely hash-linked to each data Content Object. The consumer and replica may also opportunistically encrypt their session for privacy. The consumer may choose to only trust replicas, for example, that are enumerated by the original content publisher or are provided by the a trusted party, such as the user's carrier or cloud service.

In a second example, a cell phone producing a video could be producer, publisher, and replica all in one. Because one would not want a large number of consumers on the Internet fetching data directly from one's cell phone, it could be configured to only allow the user's home media server to fetch content and then act as the authoritative replica for the Internet. The user could choose to use a carrier service (i.e. cloud-based media server) to act as the authoritative replica.

3.2 Overview of the five Proof Of Concept (PoC) performed for the ITU-T Focus Group on IMT-2020

POC #1: ICN Enhanced Mobile Video at the Network Edge (Cisco)

ICN provides a unified network and transport layer addressing content by name rather than by location. By disrupting traditional connection-oriented communication model, ICN simplifies data delivery, mobility management and secure transmission over a heterogeneous network access. In the demo, we select DASH video delivery as use case and show the benefits of ICN mobility management, in-network control (rate/loss) and network-assisted bitrate adaptation for a multi-homed user device.

PoC #2: Functional Chaining System in ICN (Fujitsu Labs of America)

Information-Centric Networking (ICN) is an emerging Internet architecture in which content is accessed by its name rather than the IP address of the host that stores the content. By separating content from location, ICN is expected to improve network efficiency and reduce the communication cost of accessing popular content. ICN principles can be applied to functions as well as to content. Named functions can then be linked to form service chains that provide optimized service delivery. This demonstration illustrates such a functional chaining system to deliver real-time processed video content to a consumer.

PoC #3: End-to-end ICN Service Orchestration with Mobility for IMT 2020 (Huawei)

The PoC demonstrates one of the important benefits of ICN of offering seamless mobility as part of the network architecture, avoiding any specific gateway functions or tunneling present in current 4G systems. This demo takes advantage of name based routing, more specifically ID/Locator name space split that ICN naturally supports to offer flexibility to the mobile entities to move between administrative domains and also handling in-session mobility when they roam in a single domain.

PoC #4: IP Services over ICN (InterDigital)

The PoC highlights two quantitative benefits of our solution for delivering IP services over ICN. The first one is that of introducing the capability to delivery HTTP responses via multicast to a number of clients. Our solution specifically supports changing multicast groups by forming multicast groups in an ad-hoc manner solely at the source network attachment point. The second aspect is that of the possibility to reduce service latency through the exposure of surrogate service endpoints in a fast and flexible manner. This is enabled by the exposure of HTTP-based resources through the FQDN of their providing servers. Examples for such surrogate functionality is that of choosing alternative HTTP-level streaming servers, localizing video playout to the regions where these playout point serve clients rather than needing to retrieve the content from a central server.

PoC #5: ICN transport for mmWave Networks (KDDI)

To resolve problems with intermittent connectivity in a mmWave network, Tokyo Tech, Sony, JRC and KDDI Labs jointly developed a new wireless access network that combined 40 GHz operation for outdoor networks with 60 GHz operation for mobiles to enable large data size content delivery on the gigabyte scale,

Using a future architecture technology called content centric networking (CCN)¹, KDDI Labs developed a method that operates together with the mmWave small zone (60 GHz band) and large zone long-term evolution (LTE) schemes in HetNets (KDDI Labs 2015). We could therefore realize high-speed file transfer in the mmWave band without the user being aware of switching of bands when passing through the GATE system.

Table 1 – Mapping 2015 Gaps to Recent Work

Gap	POC #1	POC #2	POC #3	POC #4	POC #5	Ref 1	Ref 2
E.1 ICN in IMT2020	✓	✓	✓	✓	✓		✓
E.2 ROHC						✓	
E.3 ICN S-GW							✓
E.4 ICN MME							
E.5 ICN P-GW							✓
E.6 ICN Slice	✓						
E.7 Lawful Intercept							
E.8 Mobility & Routing	✓	✓	✓	✓	✓		✓
E.9 UE Provision	✓						✓
E.10 ICN mgmt SON							
E.11 OAM							
E.12 SDN & Openflow			✓	✓			
E.13 Auth & Encrypt		✓		✓			
E.14 Encrypt							
E.15 QoS							

- Ref 1: (Suthar 2016)
- Ref 2: (Mosko 2015)

Table 1 lists the 2015 Gaps (ITU 2015) and matches them to this year's PoCs and other related publications. All the PoCs addressed the topic of using ICN in IMT-2020 to delivery user services. They also all addressed mobility and routing, as those are areas where ICN could delivery significant improvements compared to the current anchored mobility in LTE. Only one publication (Suthar 2016) addressed the technical challenges of adapting 3GPP signalling to ICN. Some other topics, E.4, E.7, E.10, E.11, E.14 and E.15 did not receive any attention. Overall, the PoCs and two publications addressed 9 of the 15 gaps.

¹ CCN is a future protocol that is currently being discussed by the Internet Research Task Force (IRTF) as a replacement for the Internet Protocol (IP).

4 ICN Enhanced Mobile Video at the Network Edge (Cisco)

ICN provides a unified network and transport layer addressing content by name rather than by location. By disrupting traditional connection-oriented communication model, ICN simplifies data delivery, mobility management and secure transmission over a heterogeneous network access. In the demo, we select DASH video delivery as use case and show the benefits of ICN mobility management, in-network control (rate/loss) and network-assisted bitrate adaptation for a multi-homed user device. We also illustrate how ICN can effectively reduce transport cost via native edge caching and multi-point/multi-source communications over the backhaul. To that aim, we orchestrate an ICN-enhanced virtualized network backhaul and show its utilization over time. An overview of the demo is described in the following figure:

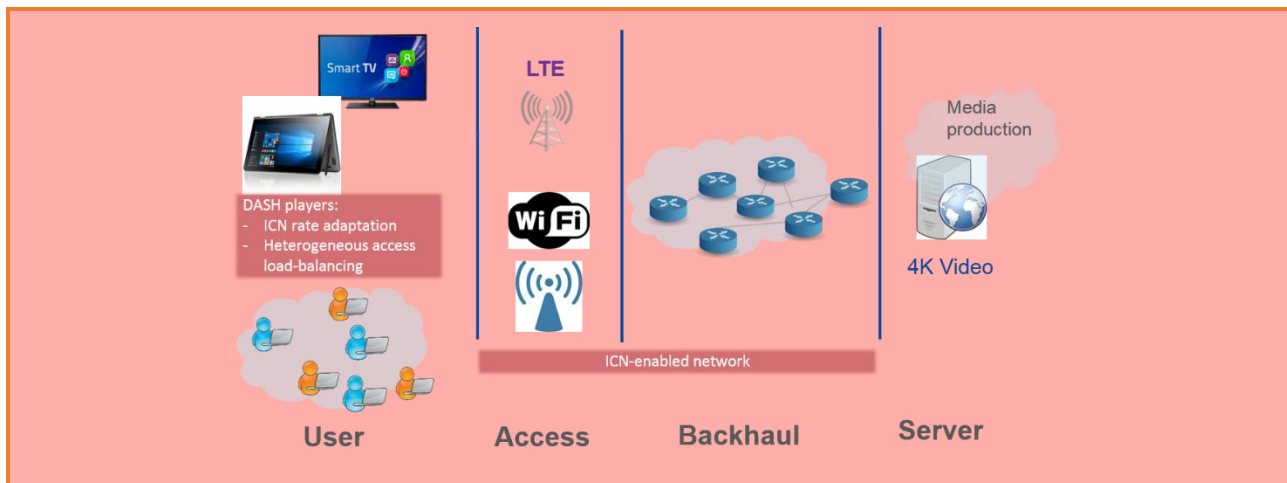


Figure 11 – PoC Architecture

4.1 Application to IMT-2020

The demo shows a number of benefits in adopting ICN with respect to traditional IP networks. In the follow we discuss in more detail these advantages

1) Gap E.1 Considering ICN as a protocol for IMT-2020

ICN provides a connectionless communication model which gives us the opportunity to re-think and implement new ways to handle user mobility, secure content transmission and exploit heterogeneous access technologies.

2) Gap E.8 ICN mobility and routing

Mobility is one of the most important aspects in IMT-2020 networks. IP networks fail to handle mobility in a simple and flexible way. The connectionless nature of ICN gives us the opportunity to define new solution to tackle the mobility problem. By default, ICN supports naturally consumer mobility. To address the micro mobility of content producers we define an anchor-less protocol called MAP-Me (Augé, et al. 2015). MAP-Me is designed to take advantage from the name-based ICN data plane in order to promptly update routes, without waiting for updates from the routing protocol.

Traffic generated from mobile users is prone to losses due to connection through unreliable wireless channels, such as WIFI, or to mobility events. The ICN model allows the deployment of a distributed in-network control that can be used to detect and recover such losses. In this perspective we developed WLDR and MLDR (Carofiglio, et al. 2016), two protocols designed to detect losses in-network and, when possible, recover them. In the PoC we show how these protocols improve the performance of our ICN transport protocol (Carofiglio, Gallo, et al. 2013) and, as a consequence, the quality of experience of the user.

3) *Gap E.9 ICN UE provisioning*

In IMT-2020 networks a user is expected to utilize heterogeneous access technologies, such as WIFI and LTE, at the same time. ICN gives us the opportunity to do this in a natural way. A user can request different pieces of content over different medium, since there is no direct connection between the user and the producer on a particular path. In the PoC we show how we can exploit different access technologies and switch among them, according to the client preferences. The usage of multiple connections increases the bandwidth available at the client and improves the user video experience: the DASH client asks for videos with higher quality, reducing, at the same time, the number of rebuffering events.

4) *Gap E.6 ICN Protocol Execution*

In the demo we deploy a virtualized network backhaul, as well as multiple clients that run a DASH video player application. We use an orchestrator to deploy and modify over the time the topology setting.

5 **Functional Chaining System in ICN (Fujitsu Labs of America)**

This POC demonstrates features and capabilities of ICN to dynamically construct functional chains that deliver on numerous IMT-2020 goals. Specifically, the functional and performance advantages are highlighted and standardization gaps are identified. The POC includes a live demonstration, technical details of which are described in Annex B.2.

5.1 **Abstract**

Information-Centric Networking (ICN) is an emerging Internet architecture in which content is accessed by its name rather than the IP address of the host that stores the content. By separating content from location, ICN is expected to improve network efficiency and reduce the communication cost of accessing popular content. While there are several representative ICN designs (Ahlgren, et al. 2012) (Xylomenos, et al. 2014), this demonstration makes use of the Named Data Networking (NDN) (Named Data Networking 2016) architecture (but its concepts can be applied equally to other ICN architectures).

ICN principles can be applied to functions as well as to content. Named functions can then be linked to form service chains that provide optimized service delivery. This demonstration illustrates such a functional chaining system to deliver real-time processed video content to a consumer.

There are multiple video sources (1, 2, 3) and multiple video processing functions (video combiner, video compression) linked by NDN routers (A, B, C, D) as shown in Figure 12.

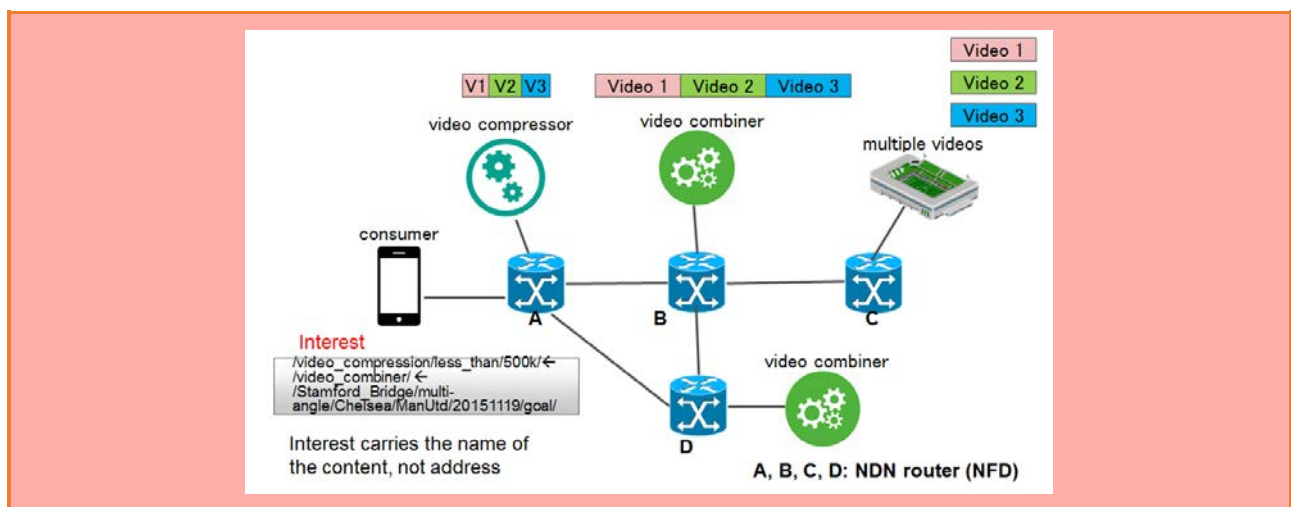


Figure 12 – Functional Chaining System demo setup

5.2 Application to IMT-2020

Many ICN gaps were identified in the FG IMT-2020 Phase-1 study (ITU 2015). This POC addresses the following gaps and illustrates corresponding advantages:

1) Gap E.1 Considering ICN as a protocol for IMT-2020

ICN/NDN transport provides the foundation for creating service chains of named functions (Sifalakis, et al. 2014) and named content (identified in the Interest request).

2) Gap E.8 ICN mobility and routing

Routing optimization is shown where functions may have multiple copies in the network, and each router is capable of selecting the next function node which is closer to the remaining functions / content in the request as a result of name-based function routing and the knowledge of all required functions / content for the whole chain. As shown in Figure 13, router A forwards the Interest to router B rather than router D for processing because the video combiner at router B is closer to the video content (Liu, et al. 2016).

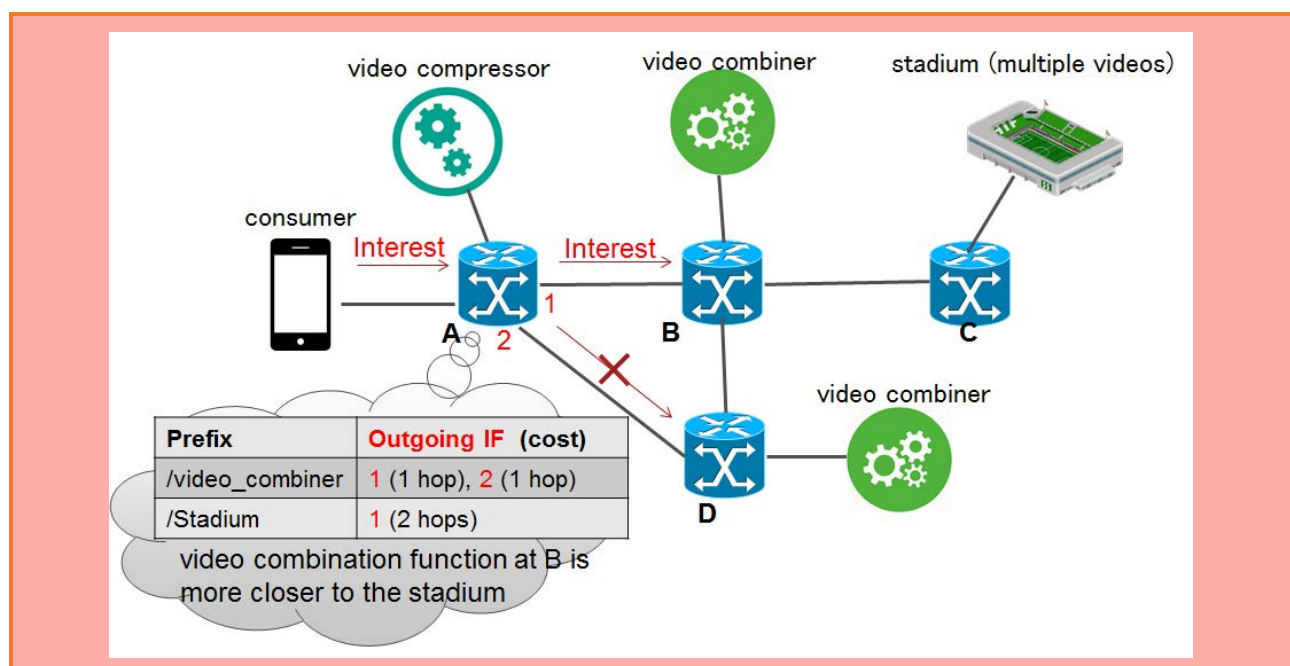


Figure 13 – Routing Optimization

Once the fully combined and compressed video has been served to the original requester via router A, a subsequent Interest request received at router A for the same combined video can be handled entirely by router A due to in-network caching as shown in Figure 14.

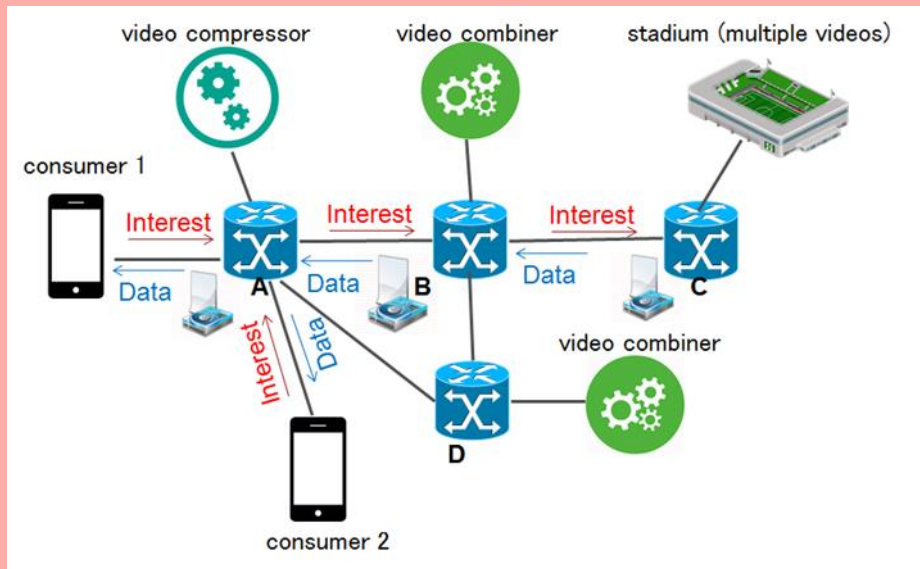


Figure 14 – In-network Caching

3) *Gap E.13 ICN Security (authentication and encryption)*

Whole-chain authentication allows the data consumer to verify every node in the service chain (NOTE – packet-level authentication mechanisms are not sufficient for the functional chain). The proposed whole-chain authentication prepends fixed-length hashed content to a message stack, as well as an unmodified signature (for the hashed content) to a signature stack, for each node along the chain. As shown in Figure 15, the data consumer is capable of identifying a malicious node (which is not immediately adjacent to the data consumer) (Bahrami, et al. 2017).

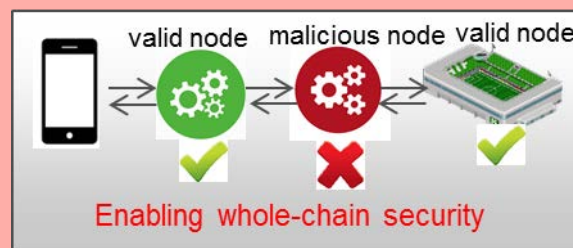


Figure 15 – Whole-chain authentication

5.3 Implications for Standardization

The POC addresses the above gaps, which are all candidates for standardization in IMT-2020. Notably, ICN topics are being actively pursued in IRTF/IETF. While the demonstration is focused on video service, it can be applied to any other processed data delivery scenario.

6 End-to-end ICN Service Orchestration with Mobility for IMT 2020 (Huawei)

6.1 Abstract

This proposal is targeted to Phase-2 of IMT-2020 working group and proposes a Proof-of- Concept (POC) for the Information Centric Networking (ICN) working group. The PoC demonstrates one of the important benefits of ICN of offering seamless mobility as part of the network architecture, avoiding any specific gateway functions or tunneling present in current 4G systems. This demo takes advantage of name based routing, more specifically ID/Locator name space split that ICN naturally supports to offer flexibility to the mobile entities to move between administrative domains and also handling in-session mobility when they

roam in a single domain. Here ID binds to applications and Locators binds to the ICN network entities. In addition, ID/Locator name space split in ICN also enables features such as multi-homing of not only end devices but also the ability to host content and services anywhere in the ICN network to meet application requirements. In addition to the mobility benefit, this demo also addresses the question of enabling ICN in a 5G environment as a slice over a generic infrastructure pool and the ability to orchestrate ICN services over well known compute and network virtualization platforms, i.e. OpenStack and ONOS.

6.2 Towards ICN Standardization

We try to address the following standardization gaps identified in Phase-1 of the focus group output document (ITU 2015) towards the use of ICN in IMT-2020 networks:

6.2.1 Gap E.1 ICN as a protocol for IMT-2020 Network:

This gap deals with the deployment of ICN. This PoC shows the feasibility of an overlay deployment over IP over a generic infrastructure. We demonstrate this by realizing a video conferencing service over a generic infrastructure over which other ICN services can also be realized. The ICN transport is based on Virtual Service Edge Router (VSER) platform (Chakraborti and al. 2015) (Ravindran, et al. 2013) running over COTS hardware. It runs CCN as host processes, managed in a centralized manner using application-driven ICN Service and Network Controller. Towards network slicing, and dynamic realization of network functions, these host processes can also be virtualized using Container or Virtual Machines. From a deployment perspective, VSER is an ideal platform for edge deployment such as for the central office that can take advantage of most of the features offered by ICN, which include mobility, multi-homing, multicasting and in-network computing depending on the points of ICN enablement in the service delivery chain. Feasibility of handling mobility over ICN is also considering the use of edge cloud resources to realize eNodeB functions, collocated with the CloudRAN implementation. This allows ICN to interface with heterogeneous RAT stacks such as LTE or Wifi.

6.2.2 Gap E.8: ICN Mobility and Routing

Another challenge raised was the support for ICN mobility and routing scalability. This is also important considering that IMT-2020 mobility requirements (NGMN 2015) has more demanding requirements for mobility such as to support for in-session user experience even at 1000km/h. Current mobility support in architectures like LTE is supported over tunnels terminating at specialized gateway functions in the network. This increases network cost, complexity and flexibility of network deployment and its management. ICN addresses the basic architectural issue in IP by allowing applications to bind to names, where ICN manages its resolution to a location in the network either through in-network routing or through the use of a name resolution system. With ICN, IP is a transport over which ICN PDUs are exchanged. Towards mobility, this prototype takes advantage of ID/Locator split that ICN naturally supports to offer flexibility to the mobile entities to move between administrative domains and also handling in-session mobility when they roam in a single domain. This removes the need for gateways or anchor nodes in the network. The demo is developed in the context of ICN/CCN. The demo shows seamless mobility of an end user generating live video being consumed by one or more participants with session interruption of ~100ms after each handover. Centralized orchestration applying SDN/NFV principles in ICN provides resource and topology abstraction to applications to inter-connect service resources to the user requests in an efficient manner exploiting compute, storage and connectivity resources of an ICN transport. Specifically, scalability of centralized routing within a domain is addressed by conducting routing in the domain using locators and limiting name to locator binding only to the edges of the network, departing from the standard SDN method of setting per-flow rules at every hop. Further the caching of the name to locator binding in the network edge reduces the need to resolve popular Interest flow in a centralized manner.

6.2.3 Gap E. 12: Operations and management (SDN/NFV)

SDN and NFV based ICN transport is very powerful as it achieves the objectives of realizing application driven networking. SDN virtualizes ICN resources i.e. ICN router's connectivity, compute, cache resources to applications to request, expand or shrink resources dedicated to it on the ICN forwarders. NFV allows the management of the service functions in the forwarders, while interacting with the SDN controller on its status. Through this prototype, we demonstrate the use of centralized programmability of the CCN transport

using OpenStack (OS) and ONOS. Both these frameworks have been adapted to an ICN/CCN context. OS is used to manage ICN Service Function (VMs executing specific in-network service logic), provisioning them on demand by respective service controller over any VSERs. ONOS is extended to execute multiple ICN controllers to manage foundational functions such as managing the virtual topology as viewed by applications, proxy the signalling from the ICN forwarders and de-multiplexing to appropriate service controllers for further actions and provisioning the FIB rules as required by application controllers.

7 IP Services over ICN (InterDigital)

In this PoC, we aim at a solution for delivering IP-based services over an ICN-based routing solution within a single operator or across collaborating operators—at higher efficiency and lower latency than possible in today's solutions. With this PoC, we provide a possible migration approach for the introduction of ICN, enabling full backward compatibility of IP-based services, applications and user equipment, while also offering the qualitative and quantitative performance advantages of ICN. Specifically, our solution enables the multicast delivery of HTTP request responses in scenarios such as those of personalized viewing of video content. Furthermore, we also enable the reduction of experienced latency through the flexible placement as well as quick activation of surrogate HTTP servers within the network and closer to the end user. Our solution does not rely on DNS-based methods, overcoming the inherent limitations of DNS-based indirection in terms of scalability, dynamicity and operational assurance, while piggybacking on the proliferation and deployment of SDN-based transport networks.

In our specific PoC example, we will showcase the easy integration of ICN networks with existing HTTP based applications and demonstrate multicast gain in a personalized video scenario, which includes on-site human users as well as emulated users in a remote data center.

The work was done in conjunction with the EU H2020 projects POINT and RIFE.

This PoC specifically addresses the following gaps identified by the Focus Group:

- *Gap E.1 Considering ICN as a protocol for IMT-2020:* Our solution is based on a native ICN solution that directly and efficiently integrates with SDN without any extensions needed to current OpenFlow specifications (1.2+). As such, it provides the capability for future native ICN applications. In addition, however, our solution provides a strong migration path for IP-based services, enabling any IP-based service and application to run on our network as well as connecting standard IP-based user equipment to the attachment points of our solution, while utilising the multicast and information routing capabilities of ICN to quantitatively improve on network utilization and latency.
- *Gap E.8 ICN mobility and routing:* Our solution provides an answer to mobility and flexible routing in that direct path routing is provided, based on handover triggers delivered to the PCE, and path/server resilience
- *Gap E.12 SDN & Openflow:* Our solution uses SDN to control the ICN underlay to transport IP services. As described above, we do not require any modifications to OpenFlow 1.2 or later.
- *Gap E.13 ICN Security:* Our solution provides the ability to securely route HTTP service requests to authorized servers, therefore preventing data leakage often occurring in CDN-based redirections. Furthermore, our solution supports secure HTTP through providing an HTTPS interception proxy at the ingress and egress NAP (or GW), requiring a certificate sharing agreement between operator and content providers. Note however that this security improvement is complementary to the main idea of the PoC which is to use HTTP on top of ICN.

7.1 InterDigital ICN solution within the IMT-2020 network

7.1.1 ICN over SDN

Within (Trossen and Parisi 2012) individual information items are identified by names or statistically unique fixed size labels which in themselves hold no meaning. Multiple information items can be placed in a scope which is also named in the same manner and can be nested within other scopes thus creating graphs of information which allow computations leading to specific information elements.

Information elements are published and subscribed to. The semantics are realized by the Rendezvous (RV), Topology Management (TM) and Forwarding (FN) functions. RV matches requests and information elements. The TM then creates a communication path to the subscriber and the FN forwards the information along path. The path is identified by a forwarding identifier (FID) that represents the path information as a bitfield information in which each bit signifies a specific link in the overall network. With that, a simple AND and COMPARE operation at each forwarding node (e.g., SDN switch) is performed to test the membership of the node's output port (identified through a specific bit position in the bit field).

Recent developments (Reed, et al. 2016) have shown that the aforementioned ICN forwarding can be directly implemented in OpenFlow controlled SDN switches. This realization relies upon the fact that SDN switches, from OpenFlow v1.2 (ONF 2011), can implement flow-rule matching using an arbitrary bit-mask across a number of header fields, including IPv6 addresses and Ethernet MAC addresses. Such arbitrary bit-mask matching is directly equivalent to the necessary AND and COMPARE operation when considering a bit-mask with only the specific output port's bitposition set. Consequently, it is possible to implement the ICN forwarding using SDN switches if the FID is inserted into arbitrary match capable fields such as the IPv6 addresses and/or the Ethernet MAC addresses. This allows a FID length of 256-bits if the IPv6 header is used or 352-bits if using both IPv6 and MAC headers, while solutions have been developed to overcome this size limitation in larger networks. Through the use of a unique Ethernet VLAN ID it is possible to separate the ICN encoded traffic from conventional Ethernet/IPv6 traffic and thus integration with existing deployments is possible.

7.2 InterDigital Solution: IP over ICN

The intention of our system is to preserve the perception of an IP-based autonomous system towards any connected peering network through standard IP-based protocols, while exposing an IP-based interface to attached user devices as well as service providers. With this, our architecture does not impose any changes to existing user and server (as well as data centre) equipment, while enabling the full application base of today's Internet. Nonetheless, the provided network attachment points can expose native ICN interfaces that would enable native ICN applications for future ICN use cases.

The translation of IP-based communication, either directly at the IP or the HTTP level, is realized at the Network Attachment POINT (NAP) placed towards the user or server equipment (both denoted as UE), while the ICN Gateway provides a translation towards peering IP networks, if required. Annex A provides more detail as regards to the operations for translating HTTP exchanges into an ICN-compliant message exchange with lowest delay possible. In general, the request-response protocol of HTTP is translated into a publication of the encapsulated HTTP request at the client-facing NAP towards the fully qualified domain name (FQDN) of the HTTP server, while the client subscribes to the response URL in full. The server-facing NAP, in turn, will have subscribed to all FQDNs exposed at its local IP interfaces, therefore receiving any request as intended. As a consequence, it will receive any such publication and forward the HTTP request to the locally connected server. Upon receiving a response from the server, it is published by the server-facing NAP, in turn being received by the client-facing NAP. The optimizations realized in our solutions allow for such operations akin to HTTP request exchanges, i.e., with initial (domain-local) DNS-like resolution followed by efficient direct path exchanges between client and server, while enabling the possibility to form multicast responses for requests for the same resource and arriving at the quasi-same time. More information can be found in the mentioned Annex.

7.3 Benefits of InterDigital ICN solution

The PoC highlights two quantitative benefits of our solution. The first one is that of introducing the capability to delivery HTTP responses via multicast to a number of clients. We appreciate that the nature of such multicast delivery is likely to change from request to request due to the unsynchronized nature of the HTTP requests – nonetheless, our solution specifically supports this aspect by allowing to form multicast groups in an ad-hoc manner solely at the sNAP and based on the path information of the individual clients only. No specific signalling is required for the multicast support since a mere binary OR operation over all member of the multicast group suffices – such operation can easily be done for another set of multicast responses in the case of another request, again at no additional costs for signalling.

The second aspect is that of the possibility to reduce service latency through the exposure of surrogate service endpoints in a fast and flexible manner. This is enabled by the exposure of HTTP-based resources through the FQDN of their providing servers. Through an authoritative registration interface to the ICN routing solution, our PoC can enable such surrogate endpoints within the network at speeds of less than 1s, therefore enabling the service completion from a possibly closer endpoint than the one originally being chosen. Examples for such surrogate functionality is that of choosing alternative HTTP-level streaming servers, localizing video playout to the regions where these playout point serve clients rather than needing to retrieve the content from a central server.

With respect to mobility, the path management of the PoC allows for recalculation of path information in the case of mobility, e.g., triggered by a handover event. Through replication of the PCE (path computation element) of our solution, the path computation can be regionalized, further reducing the delay for recalculation. Nonetheless, in typical mobility scenarios, we observe similar signalling delays as for anchor point approaches. However, our recalculation ensures direct path data transfer, leading to a reduction of path stretch compared to anchor point approaches.

On the device-local link, there are no direct benefits of our solution, similar to many MEC solutions which mainly focus on the access network rather than the access link per se.

7.4 Integration with IMT-2020 NWs

From the above it is clear that once SDN is integrated into IMT-2020, integrating our ICN solution is a simple matter of interfacing the TM of our solution with a standard SDN controller, while relying on OpenFlow-compliant proactive rule insertion as outlined above and in the annex. Other than an SDN capable network, the network only requires NAP close to the clients and servers. While a 'close' deployment could be in customer premise equipment, other suitable locations are natural aggregation points, such as a serving gateway or a local GW (such as BRAS in fixed line networks) for the client and with a PDN gateway for the server.

SDN is already under study for inclusion in IMT-2020 by many SDOs (incl. ITU-T SG-13).

7.5 Further study and possible standardization

The following areas will benefit from standardization:

- ICN is integrated over SDN through native operations of OpenFlow (1.3) switches over certain e.g. IPV6 and or Ethernet MAC header fields. The specific fields and their location / significance will have to be standardized.
- For the network access points:
 - The hashing function performed over the (URL) domain names will need to be defined
 - The trigger events which cause the sNAP to register a service
 - Control signalling necessary to implement RV, TM & FN functions
- Namespaces or both HTTP and IP mapping
- Signalling to support path management with topology changes
- Signalling to support network mobility
- (optional) handling of HTTPS: agree of key management

8 ICN transport for mmWave Networks (KDDI)

Tokyo Institute of Technology (Tokyo Tech), Sony Corporation (Sony), Japan Radio Co. Ltd (JRC) and KDDI R&D Laboratories Inc. (KDDI Labs) announced they have jointly developed and successfully implemented a 40 GHz² and 60 GHz³ wave-based high-throughput wireless access network for large-scale data content distribution. This system provides a way to introduce a high-throughput communication service to next-generation networks using millimeter wave (mmWave)⁴-based wireless systems. The system also enables efficient use of the mmWave communication band, which is much less crowded than the wavebands below 6 GHz. The development partners will demonstrate their achievements through open experiments at the Mobile Communication Workshop sponsored by The Institute of Electronics, Information and Communication Engineers (IEICE), held at Tokyo Tech on March 2-4, 2016. This development was conducted as a part of the “R&D for Expansion of Radio Wave Resources” program sponsored by the Japanese Ministry of Internal Affairs and Communications (MIC).

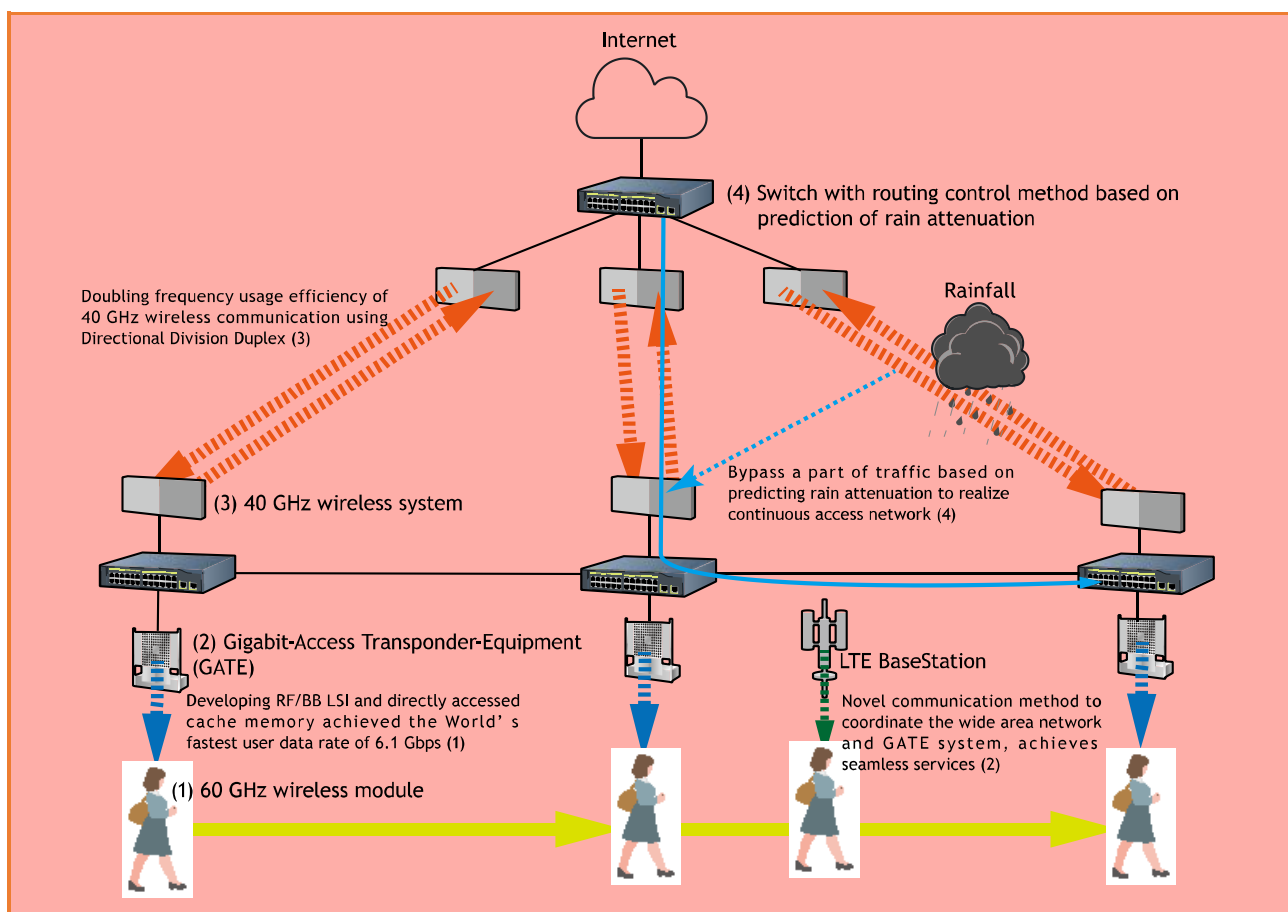


Figure 16 – Schematic overview of proposed wireless network.

² The 40 GHz wave band is a licensed band that is available for high-density applications in the fixed service defined in ITU WRC-2000 (World Radio Communication Conference).

³ The 60 GHz wave band is an unlicensed band that can be used worldwide. In Japan, a band between 57 and 66 GHz is available.

⁴ Millimeter waves are usually understood to be waves with operating frequencies above 30 GHz and wavelengths that are measured in mm.

8.1 Background

The ever-increasing levels of wireless-communication traffic in recent years have consequently led to increasing demand for more communication frequencies. Utilization of the millimeter wave (mmWave) band represents a key technology for the development of the heterogeneous networks (HetNets)⁵ that will be used for 5th generation wireless cellular networks (5G). However, the application of mmWaves to mobile communications is generally considered to be difficult because of the short communication range associated with these waves as a result of the high attenuation of radio power in the mmWave band. For outdoor applications of mmWaves in particular, one major difficulty is how to avoid the effects of rain, which can dramatically reduce the transmitted radio-wave power. For mobile applications of mmWaves, the significance of this problem is that network operators must strive to avoid the effects of low data throughput in commercial mobile devices with maximum data rates of several hundred Mbps, which are much lower than the multi-Gbps data rate of a typical mmWave-based wireless device, while also increasing frequency usage efficiency using multilevel modulation in these wireless devices.

8.2 Technical Details

To resolve the above problems, Tokyo Tech, Sony, JRC and KDDI Labs jointly developed a new wireless access network that combined 40 GHz operation for outdoor networks with 60 GHz operation for mobiles to enable large data size content delivery on the gigabyte scale,

Using a future architecture technology called content centric networking (CCN)⁶, KDDI Labs developed a method that operates together with the mmWave small zone (60 GHz band) and large zone long-term evolution (LTE) schemes in HetNets (KDDI Labs 2015). We could therefore realize high-speed file transfer in the mmWave band without the user being aware of switching of bands when passing through the GATE system.

9 Bibliography

- Ahlgren, B, C. Dannewitz, Imbrenda, C., and et al. "A survey of information-centric networking." *IEEE Communications Magazine* 50, no. 7 (July 2012): 22-36.
- Augé, J., G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng. "Anchor-less Producer Mobility in ICN." *ACM ICN*. San Francisco, USA, 2015.
- Azgin, Aytac, Ravi Ravindran, and G.Q. Wang. "Scalable Mobility-Centric Architecture for Named data Networking." *SCENE Workshop, IEEE ICCCN*. 2014.
- . "Seamless Mobility as a Service in Information Centric Networks." *5G/ICN Workshop, ACM ICN*. 2016.
- Bahrami, M., L. Xie, L. Liu, and et al. "Secure function chaining enabled by information-centric networking." *IEEE ICNC*. Silicon Valley, USA, 2017.
- Carofiglio, G., L. Muscariello, M. Papalini, N. Rozhnova, and X. Zeng. "Leveraging ICN In-network Control for Loss Detection and Recovery in Wireless Mobile Networks." *ACM ICN*. Kyoto, Japan, 2016.
- Carofiglio, G., M. Gallo, L. Muscariello, M. Papalini, and S. Wang. "Optimal multipath congestion control and request forwarding in Information-Centric Networks." *IEEE ICNP*. Gottingen, Germany, 2013.
- Chakraborti, Asit, and et al. "ICN based Scalable Audio-Video Conferencing on Virtualized Service Edge Router (VSER) Platform." *ACM ICN*. 2015.

⁵ A HetNet is a network that is used to connect computers and other devices with different operating systems and/or protocols. An example of the application of millimeter wave technology to small-cell networks can be found in: http://search.ieice.org/bin/pdf_link.php?category=B&lang=E&year=2015&fname=e98-b_3_388&abst=

⁶ CCN is a future protocol that is currently being discussed by the Internet Research Task Force (IRTF) as a replacement for the Internet Protocol (IP).

- Handigol, N., B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown. "Reproducible network experiments using container-based emulation." *8th Conference on Emerging networking experiments and technologies*. ACM, 2012. 253-264.
- ITU. "FG IMT-2020: Report on Standards Gap Analysis." TD 208 (PLEN/13), SG-13, 2015.
- Jangam, Anil, Ravi Ravindran, and et al. "Realtime Multi-Party Video Conferencing Service over Information-Centric Network." *Workshop on Multimedia Streaming in ICN (MuSIC)*. 2015.
- KDDI Labs. "Press Release." 2015. <http://www.kddilabs.jp/press/2015/0525.html>.
- Liu, L., L. Xie, M. Bahrami, and et al. "Demonstration of a functional chaining system enabled by named-data networking." *ACM ICN*. Kyoto, Japan, 2016.
- Mosko, Marc. *Header Compression for TLV-based Packets*. 5 Nov 2015. <https://www.ietf.org/proceedings/94/slides/slides-94-icnrg-0.pdf> (accessed Dec 20, 2016).
- Named Data Networking. *NDN: Named Data Networking*. 2016. <http://named-data.net> (accessed 2016).
- NGMN. "NMGM 5G White Paper." 17 Feb 2015. https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf.
- ONF. "OpenFlow Specification v1.2." 2011. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.2.pdf>.
- Ravindran, Ravi, Asit Chakraborti, and Aytac Azgin. "Forwarding Label Support in CCN Protocol." *IETF/ICNrg*. 21 March 2016. <https://tools.ietf.org/html/draft-ravi-ccn-forwarding-label-02>.
- Ravindran, Ravi, Xuan Liu, Asit Chakraborti, and G.Q. Wang. "Towards Software-Defined ICN Based Edge Cloud Services." *IEEE CloudNet*. 2013.
- Reed, M.J., M. Al-Naday, D. Trossen, G. Petropoulos, and S. Spirou. "Stateless multicast switching in software defined networks." *IEEE ICC*. 2016.
- Sifalakis, M., B. Kohler, C. Scherb, and C. Tschudin. "An information centric network for computing the distribution of computations." *ACM ICN*. Paris, France, 2014.
- Sony and Tokyo Tech. "Sony and Tokyo Tech Jointly Develop Low-power LSIs for Wideband Millimeter-wave Wireless Communications that Achieves the World's Fastest Data Transfer Rate of 6.3 Gb/s - Low power design specifically for use on mobile devices." 20 Feb 2012. <http://www.sony.net/SonyInfo/News/Press/201202/12-0220E>.
- Suthar, Prakesh. "Deploying IN in LTE network and Options for 5G." 2016.
- Trossen, D., and G. Parisis. "Designing and Realizing an Information-Centric Internet." *IEEE Communications Magazine* 50, no. 7 (2012): 60-67.
- Xylomenos, G., C. Ververidis, V. Siris, and et al. "A survey of information-centric networking research." *IEEE Communication Surveys Tutorials* 16, no. 2 (May 2014): 1024-1049.

10 Annex A: Proof-of-Concept Technical Background

10.1 End-to-end ICN Service Orchestration with Mobility for IMT 2020

10.1.1 System Architecture

Fig. 1 is the system level view of the PoC. ICN UEs connect to the ICN virtual service edge routers (VSEs) over a single hop IP link (which can also be replaced with a WiFi or LTE access stack implementation). VSE runs on COTS server and implements a CCN forwarding daemon along with the feature of enabling dynamic interaction with any service function orchestrated by the service controllers on these nodes. Three kinds of service functions are employed in our prototype: 1) to support basic network services like discovery and

naming, e.g. the *service access point* (SAP) on the VSERs aids with that; 2) control plane functions to aid network services like mobility; 3) to aid application services, such as in this case the A/V conferencing service.

The VSER nodes are overlaid over IP, whose state is orchestrated by OpenStack and ONOS. ONOS implements multiple controllers towards this demo: 1) the *ICN network controller* abstracts the VSER nodes to applications, to enable connectivity between service functions, UE and the content resources; 2) the *mobility controller* implements the domain level dynamic name resolution function, through which the mapping of a identifier to one or more locator names are managed; 3) the *video conference controller* manages the control and Interest forwarding logic to interconnect the audio/video flows from multiple participants. OpenStack is used to manage the operation of provisioning and deleting Virtual Machines executing specific control plane of application logic in the VSERs; this is conducted through APIs exposed by OpenStack by a custom orchestrator called the *Service Manager*.

ICN services are orchestrated by the Service Manager. It implements the video conferencing service controller through which a user requests to provision a conference instance. The A/V controller in the service manager requests OpenStack to provision the relevant service function in the context of the given service. Once these services are active, they update the application controllers in ONOS to initiate appropriate routing rules to interconnect the services with one another and conduct dynamic routing when events such as participants join and leave the conference.

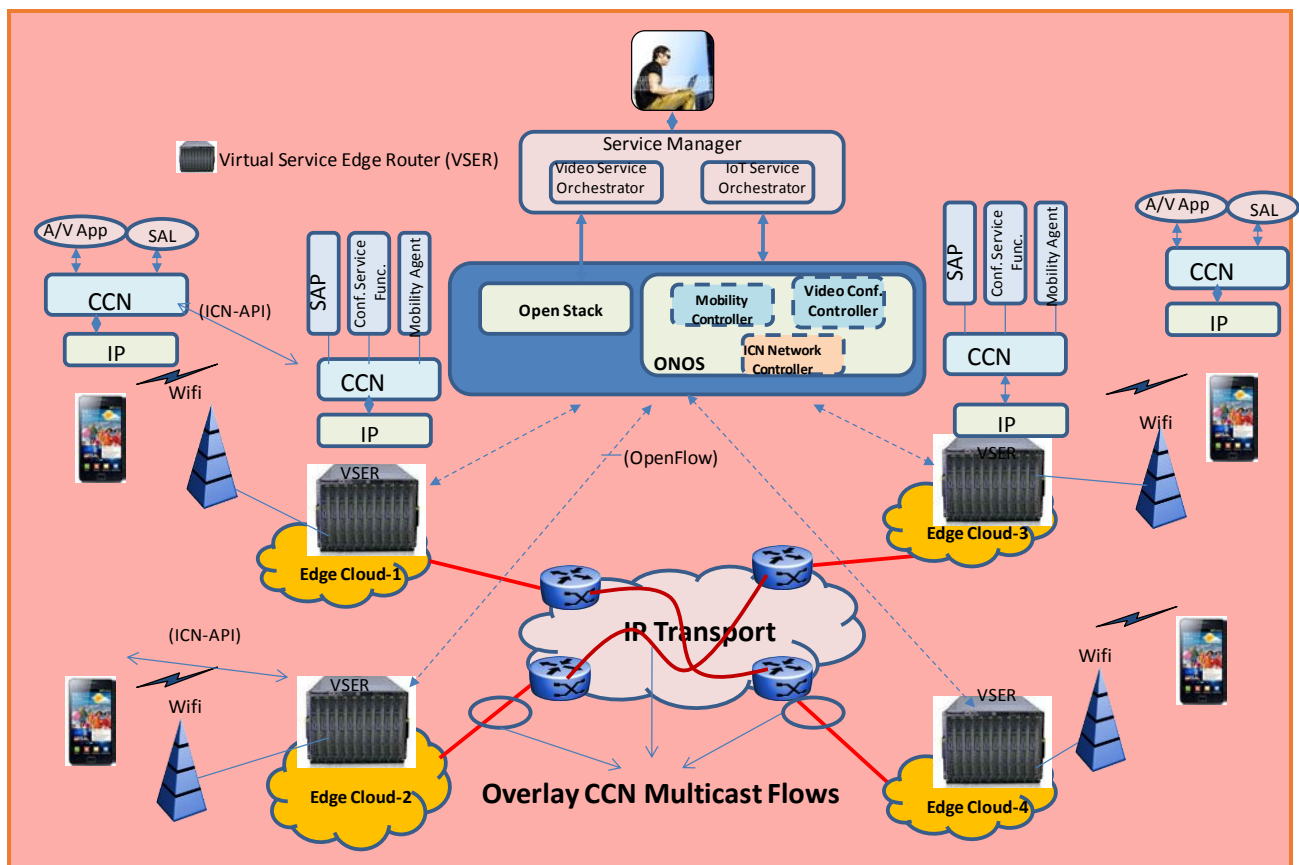


Figure 17 – VSER System Architecture

10.1.2 Mobility Solution

The mobility solution implemented to aid seamless producer mobility in ICN/CCN for this demo is described in (Azgin, Ravindran and Wang 2016). Though the mobility handled by the VSERs are handled in an IP overlay manner, with appropriate control plane support, an ICN application's producer mobility can be handled in the ICN layer, without underlay mobility support. The efficiency of such overlay mobility depends on the specific approach applied to re-connect UE to the new point-of-attachment (PoA) after handoff, and the cross layer communication efficiency between the ICN and L2/L3 when a handoff is triggered and specific ICN layer mobility strategy. We apply a "make-before-break" network-based mobility approach where the IP network binding of the new PoA is provided to the UE before the handoff. This is based on the current practice of UE providing the candidate list of base stations based on the signal quality perceived from its current location.

For seamless mobility, late binding approach (Azgin, Ravindran and Wang, Scalable Mobility-Centric Architecture for Named data Networking 2014) is applied using forwarding label insertion (Ravindran, Chakraborti and Azgin, Forwarding Label Support in CCN Protocol 2016) in the CCN Interest as a result of name resolution applied at the ingress PoA which swapped at the producer end PoA with a new forwarding label, if required, thus achieving seamless mobility. Through this PoC we also demonstrate the feasibility of a mobility-as-a-service realization (Azgin, Ravindran and Wang, Seamless Mobility as a Service in Information Centric Networks 2016) where any application can request the ICN mobility control plane to handle mobility for a name prefix. As a result, all the flows under that name will be provided seamless mobility support. Further mobility service controller itself is service aware, by managing multiple service profiles and managing the service names for which mobility has been requested for each profile.

The prototype is developed to show the feature of realizing mobility as a service. Here any application can request mobility to flows under its name prefix by requesting an agent function in the UE to register it for mobility service offered by the network. The network then creates appropriate mobility state in the VSER nodes and the mobility controller to handle the Interest flows with appropriate mobility support.

10.1.3 A/V Application Design

The architecture of the A/V conferencing service is described in (Ravindran, Liu, et al. 2013) (Jangam, Ravindran and al. 2015). At a high level, the applications implements a producer capturing real-time audio and video from the video cam and the audio source, encoding and then publishing them as content objects in the application cache. Considering stringent end-to-end audio and video latency requirements of 100-150 ms and 200-300 ms respectively, the consumers express pre-fetched Interests stored in the application cache of the producer application. These Interests are satisfied as soon as the content is generated, and when the content arrives at the consumer, it expresses more Interest for the preset pre-fetch duration. In addition, the service functions aiding the conferencing application helps with random participant join and leave, by actively pushing notifications of the producer state in a periodic manner.

10.1.4 POC Workflow and Initial Results

- Figure 18, shows the demo setup. For the demo we use 3 VSER, 1 CCN relay node and 3 participants, along with the OpenStack/ONOS controllers and Web GUI for Service provisioning and Management.
- Service Manager's Browser GUI is used to provision the video conferencing service
 - The requirements are provided in the form of number of participants, sites etc.
 - This will result in provisioning the Conf. Service VMs on the VSERs.
 - It will also program the CCN FIBs for service level connectivity between the VMs.
- The mobility control plane is also provisioned through the OpenStack. This provisions the mobility service agents in the VSER to aid with name resolution.
- The participants then discover the provisioned conferences in the network.

- They choose to join of the conference, this results in discovering all the active participants.
 - As a result the UE reachability is programmed in the CCN FIBs for the new participants Audio/Video flows
- The participant then enables his own Audio/Video stream, which any remote participants will be able to request and receive. In the demo, we show two consumers requesting a mobile producer's content.
 - We will demonstrate an all party conferencing scenario.
- A participant can join and leave the conference at will.
- Mobility-as-a-service will be demonstrated by, first having the conferencing application in the UE explicitly requiring mobility support, which triggers appropriate control and data plane state in created in the network. Then mobility is demonstrated by moving participants between subnets without interruption during their video conferencing without affecting the user experience.
- The demo will be over Wifi, considering the fact that any other licensed transport can also be accommodated with appropriate L3/L2 adaptation.
- We show seamless handover of the mobile A/V producer node by switching its Wifi interface from one AP to another. The consumer side performance during this handover is shown in Figure 19. We note for the demo system, the consumer's session disruption is around 100ms equivalent of video frame loss, ensuring seamless experience.

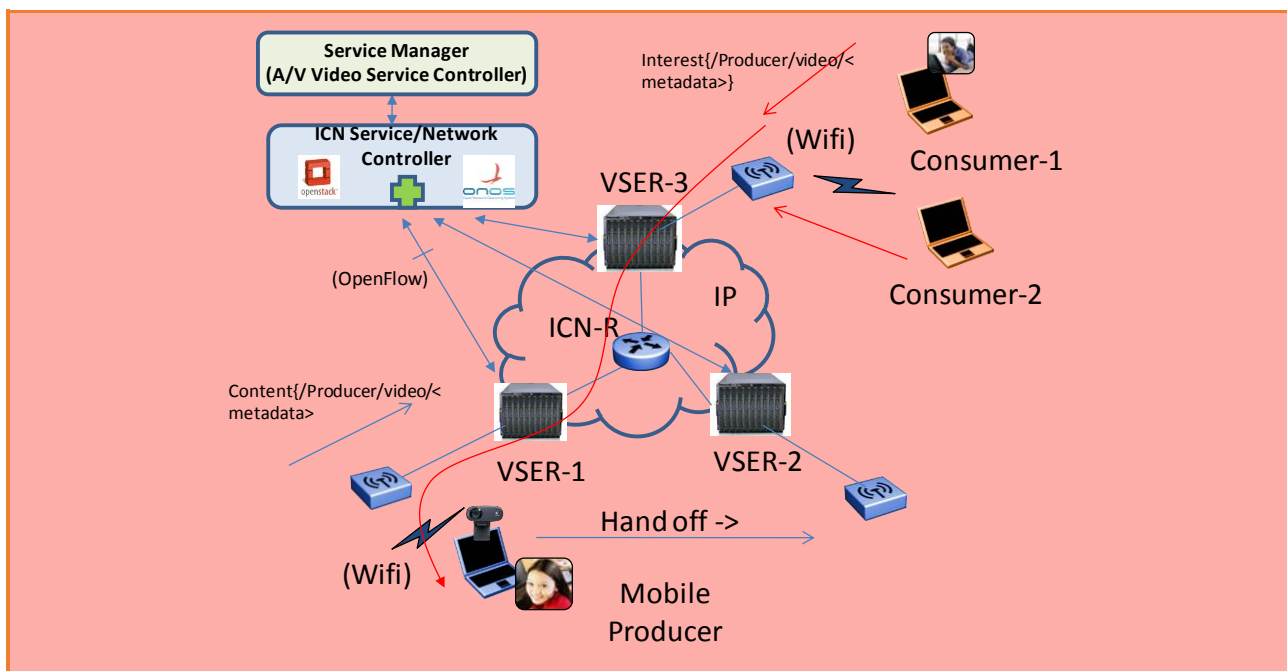


Figure 18 – VSER Demo Setup

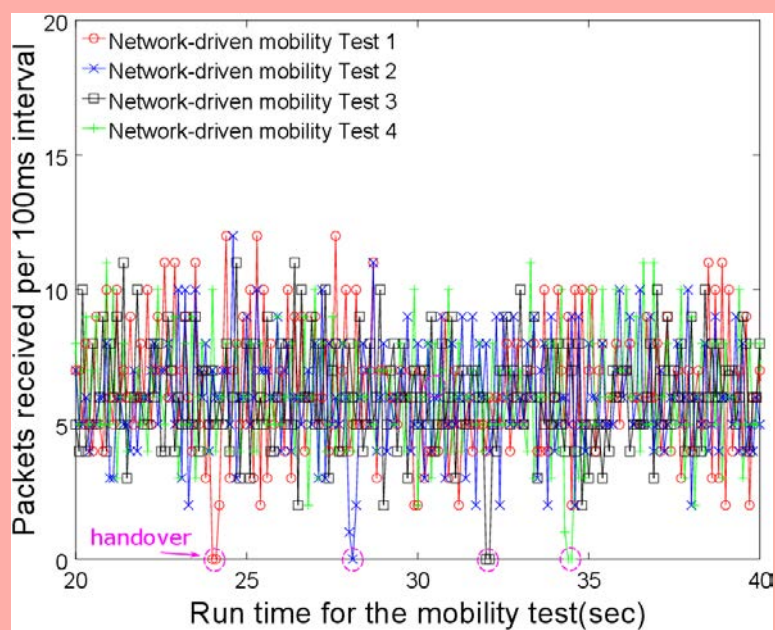


Figure 19 – VSER Consumer Performance

10.2 IP Services over ICN

10.2.1 Network architecture

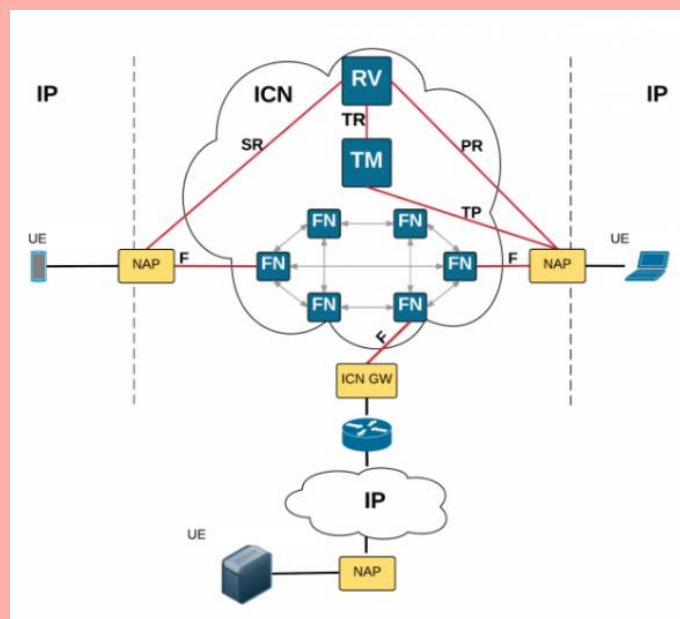


Figure 20 – IP over ICN Network Architecture

Inside the ICN network, the core network functions as outlined above are realized, namely the rendezvous (RV), topology management (TM) and forwarding (FN) function. Although implementable over general L2 networks, we assume in most deployments an SDN-based forwarding solution, based on the BF-based forwarding as presented in Section 2.2 and is the basis for our demo. The combined RV and TM functions fulfill the more traditional path computation element (PCE) role found in SDN environments. Figure 20 presents the various elements of our architecture.

In the case of services across multiple domains, the establishment of a single domain IP-based autonomous system in Figure 20 reduces this case to that of a standard inter-domain IP-based operation. In other words, a service request stemming from a cNAP in one domain will be sent via the ICN GW of the originating domain, governed via standard BGP (Border Gateway Protocol) mechanisms, to the receiving domain. If this domain is another IP-over-ICN domain, the request will enter through the ICN GW of the receiving domain and will be forwarded accordingly within the network, as defined by the operations in our PoC.

10.2.2 Showcase Proof-of-Concept

The demonstration to run IP services over an ICN-based infrastructure and leveraging the novel coincidental multicast concepts for HTTP traffic requires a deployment which allows to showcase exactly this. Figure 21 illustrates the topology of the test-bed to demonstrate the benefits of the proposed solution. All IP endpoints (both clients and servers) are depicted with grey squares and their NAPs with aqua circles; each NAP serves exactly one IP endpoint. The Mininet platform (Handigol, et al. 2012) was used in this topology to construct a cluster of 10 IP endpoints acting as clients requesting content from a server which located at the centre bottom in Figure 21. This IP endpoint is labelled as “Apache Server” and serves *http://video.point* to the Mininet clients. Another IP endpoint and its NAP is depicted in the bottom left, next to the RV / TM, which acts as a trigger client to start the experiment. All yellow circles are pure ICN forwarding nodes with no other functionality than connecting the neighbouring nodes.

The content offered by the server is an MPEG DASH video which allows to stream a video via HTTP. To start the experiment the Gstreamer client issues the initial HTTP request to *http://video.point/stream.mpd* which triggers a dedicated software, running on the NAP serving video.point, to send an ‘out-of-band’ control message to all emulated clients notifying them to start requesting the stream.mpd file too. This mimics a group of clients that happen to watch the same content at roughly the same time.

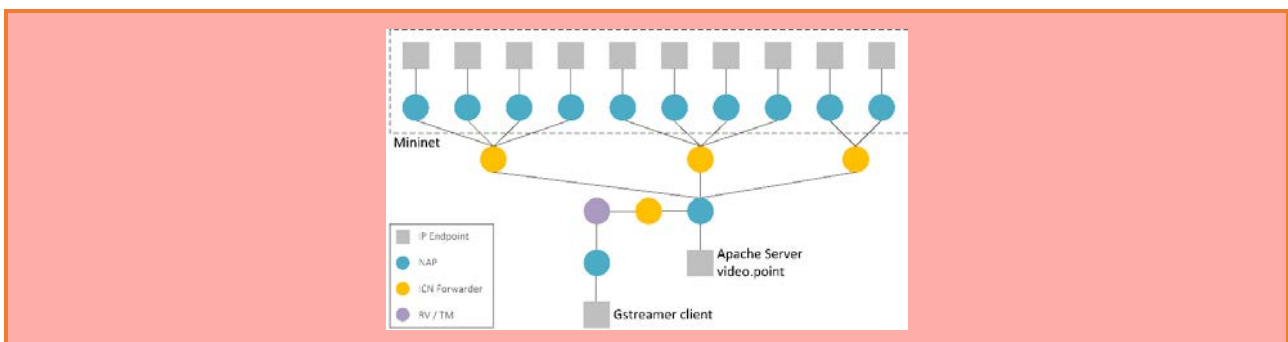


Figure 21 – Test-bed topology with 10 emulated clients

In terms of hard- and software, the deployment is based on COTS x86 machines with Debian 8.3 installed. An unmodified Gstreamer client Version 1.4.4 was used together with an unmodified Apache web server. The MPEG DASH video was encoded with publicly available encoding and packaging tools.

10.2.3 Further details: HTTP Mapping Operations

10.2.3.1 Naming Conventions

In order to realize the desired HTTP-based request & response semantics, we map HTTP onto an ICN namespace as illustrated in Figure 22. The request is named through a hash over the fully qualified domain name (FQDN) of the server, while the response is named through a hash over the full URL of the request. A unique root identifier is chosen in our system and is subject to future standardization.

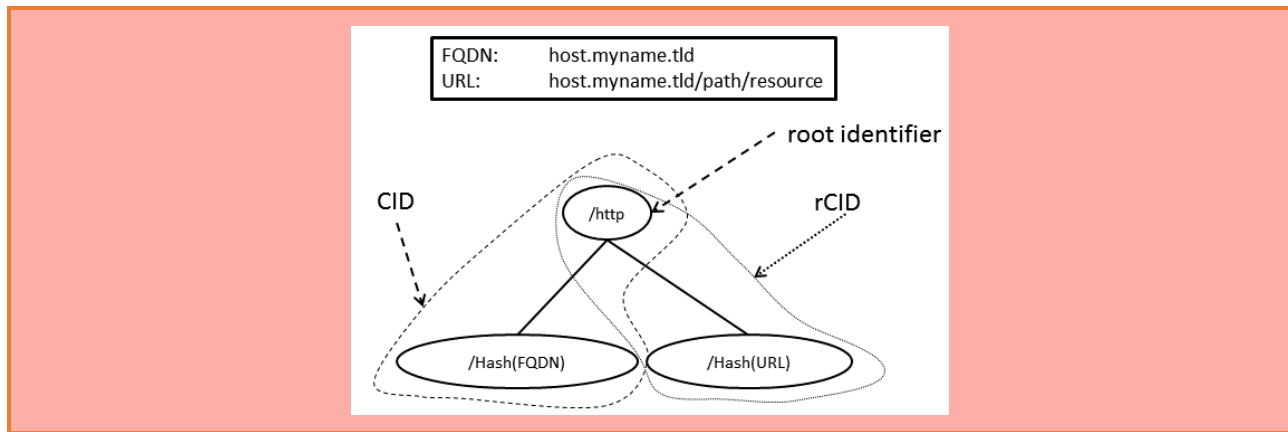


Figure 22 – HTTP-over-ICN Namespace

As in any hashing operation, the appropriate choice for the hashing the FQDN is crucial to avoid conflicts. The underlying ICN ID space is large enough to avoid conflicts when hashing an FQDN. For the response hash (over URLs), the implicit subscription model that is being utilized for delay improvements (see annex) provides a simple resolution due to the request/response association (with the original request being already provided in the request to which the implicit subscription will be associated).

10.2.3.2 Registering HTTP Services

For any HTTP-based service to be exposed in our network, the ICN NAP (see Figure 20) – to which the HTTP-based server is attached – registers the FQDN of the service by subscribing to `/http/hash(FQDN)` in our namespace. We provide several registration triggers through a NAP-internal registration interface. For instance, DNS registrations of the server can be used to trigger such a registration, while others might use static configurations or a proprietary management console of the network operator for registered services provided in the network. Furthermore, dynamic DNS could be realized through a dynDNS-compliant client. Alternatively, we also envision the registration to be triggered by a *surrogate service endpoint management* system. Here, the envisioned management system would ‘activate’ a server with the registration and therefore change the server's state from purely ‘booted’ to ‘connected’.

After the registration subscription, the NAP publishes the current list of local FQDN registrations under a well-known ICN root scope `/DNSlocal`. All ICN NAPs, as well as the ICN BGW, within the local ICN network subscribe to this name, hence they will receive any updated FQDN registrations. Upon receiving such an updated list, each ICN NAP searches its internal FQDN DB for a row with a matching FQDN column. If such a row is found, the receiving NAP/GW will initiate an update of any forwarding information related to the FQDN by removing any existing path information to the FQDN. This will lead to a new RV/TM interaction when a new request will be sent to the FQDN, which in turn will lead to an updated FId that reflects the new availability of the FQDN. As a consequence, this procedure ensures that new surrogate servers are utilized in future requests, leading to the possibility to implement surrogate service endpoints with lower service latency.

It is clear that storing the FQDN DB at each NAP places a burden on the NAP in terms of storage requirements but enables the utilization of newer, possibly better choices for HTTP service endpoints.

10.2.3.3 Handling HTTP Requests

At the client side, the ICN NAP acts as a web proxy towards the client UE, i.e., it terminates the HTTP session at the HTTP level. The extracted HTTP request at the proxy level is then used to create an appropriate ICN name for the request, named *CID*, as */http/hash(FQDN)* and for the response, named *rCID*, as */http/hash(URL)*, where the URL and FQDN are extracted from the HTTP request⁷. Then, it encapsulates the request and publishes the ICN information item towards the local ICN network under the *CID* name. Furthermore, the NAP will subscribe to the information item named *rCID*, i.e., the response to the request, towards the local ICN network.

In the case of a network-local server, the NAP at which the server is registered will receive the publication (this is assured by the procedures described in the previous section). It will then de-capsulate the HTTP request from the ICN packet and forward the HTTP request to the locally attached HTTP server via its own local web proxy.

In the case of a network-external server, i.e., a server located beyond the ICN GW within a peering network, the ICN rendezvous function will realize a *default matching procedure*, in which any request without matching subscriber(s) will yield in matching the publication to a pre-defined wildcard name under the */http* root scope, with the designated default GW of the ICN network having subscribed to this wildcard. In addition, any other possible GW in the network can subscribe to specific FQDNs, e.g., for realizing content-level peering arrangement with large-scale content providers.

For network-external incoming HTTP requests, i.e., sent from network-external users to any network-internal FQDNs, the ICN GWs are interpreted as a NAP to the peering network, realizing the same web proxy termination and ICN publication of the request (with subscription to the response) as any other NAP in the system.

10.2.3.4 Handling HTTP Responses

In the case of responses sent from servers in the local ICN network, the local ICN NAP will receive these responses. It will then determine the appropriate ICN name for the response, i.e., *rCID*, as */http/hash(URL)*, with the URL being extracted from the HTTP request. It will then publish the encapsulated response to *rCID*. The corresponding ICN NAP (or GW in the case of a request sent from a network-external client), having subscribed to this *rCID* in response to sending the original request *CID*, will then receive the response, decapsulate the response and forward the HTTP response to the local client.

Any HTTP response from a server in the Internet will arrive at the corresponding ICN BGW with the same NAP procedures being implemented at the GW.

10.2.4 Addressing Delay Challenges

Figure 23 shows the resulting messaging in the system of Figure 20 with requests being sent to the client-side NAP, which in turn indicates the request availability to the RV function, which in turn triggers the path computation towards the TM, finally providing the path information to the cNAP. After sending the data via the path and subscribing to the response, the messaging on the return path is reversed for the response with name *rCID*.

It is clear that such signalling is hardly conducive for low latency and high request throughput. We have therefore introduced optimizations that will improve the overall request throughput and reduce any lookup latency to the initial request and responses between client and server, similar to the initial DNS local lookup in an IP system.

The first optimization is already supported by the underlying ICN system by providing a node-local caching table that maps previously resolved CIDs to forwarding identifiers (FIDs). This is similar to node-local DNS caches and therefore reduces the request latency to this initial lookup, while any further request to the server will re-use the previously provided FID, since the *CID* remains unchanged as a hash over the FQDN.

⁷ Our current proposal for HTTPS envisions the establishment of an interception proxy at the ingress and egress NAP (or GW), requiring a certificate sharing agreement between operator and content providers.

With that in mind, our main focus is the reduction of the delay incurred in the response path. The problem here lies in the fact that the URL is likely different for every request (assuming some form of meaningful service interaction between client and server). Hence, the channel semantic cannot be applied here, since the corresponding *rCID* different for each publication from the sNAP to the cNAP. In order to remove the delay, we will have to realize the RV and TM function in the sNAP, at least after an initial lookup. For this, we propose to include the result of the RV function in the *CID* publication message sent by the cNAP, i.e., the node information of the client to which the response needs to be published. With this, the sNAP can create a relation between the incoming publication and the response publication without needing to contact the domain-local RV. Furthermore, this node information can then be used at the sNAP to initiate the path computation via the TM by utilizing its own NId (as the publisher of the response) and the NId of the client (which it has received in the client's request).

If the sNAP caches the response of this path computation, i.e., the mapping between client NId and FId, any future response can be directly sent to the client without consulting the TM, therefore allowing for direct client/server exchange after these initial lookups for the request and response path.

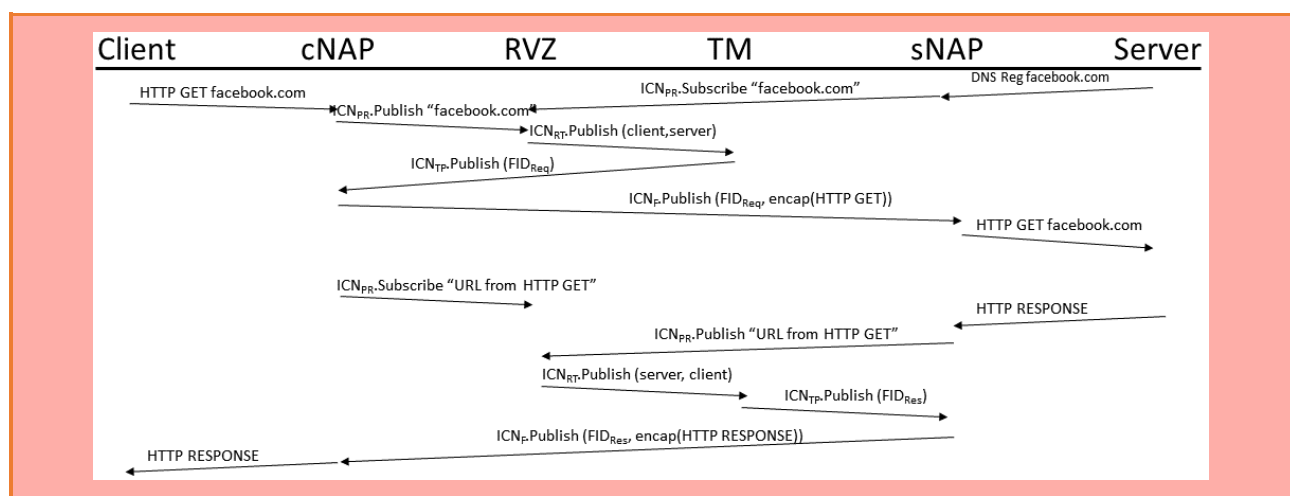


Figure 23 – Messaging for HTTP over ICN

10.2.5 Enabling HTTP Multicast Responses

A significant benefit of introducing the proposed latency improvements in the previous section is that of *spontaneous multicast group formation* for the support of HTTP multicast responses. For this, we assume situations where requests for the same HTTP resource (as identified in the URL) arrive at roughly the same time at the sNAP. With the native multicast capability of the underlying ICN system, it is desirable that the responses to such request could be sent to all requesting clients in an efficient multicast delivery instead of the inefficient HTTP unicast delivery in an IP-routed system.

For this, it is crucial that we are able to spontaneously create multicast groups that are formed from the clients that have quasi-synchronously requested the same response. Such spontaneous group formation is enabled through an interesting characteristic of the underlying BF-based forwarding solution. BF identifiers, FIds, that describe a path in the network (here from the sNAP to a specific cNAP) can be used to create joined paths by simply ORing the individual BF identifiers into a new FId. This new, joined FId, now defines the multicast path to clients previously described through the individual client paths. Hence, when ‘catching’ such possible responses, the sNAP will simply look up all FIds to the NIds that have outstanding subscriptions to the same URL, then perform a binary OR over all FIds, and use this new BF in the response instead. Given the simplicity of this FId joining, it can easily be performed at runtime and at individual response level. We call this capability of our system *co-incidental multicast*.

10.2.6 Flow Control

Since our solution directly transfers HTTP requests and responses over the ICN, it implements its own lightweight transport protocol in order to support possible multicast delivery of responses in a flow controlled manner. Due to the possibility to run TCP services that are not HTTP applications over the solution – these services will then be transmitted via an IP-over-ICN solution that works similar to the described HTTP one – our lightweight HTTP transport protocol is TCP.

10.3 ICN transport for mmWave Networks (KDDI)

Sony and Tokyo Tech had previously developed experimental 60 GHz wireless complementary metal–oxide–semiconductor (CMOS) large-scale integrated circuits (LSIs) that operated with a data rate of 6.3 Gbps in the physical (PHY) layer in 2012 (Sony and Tokyo Tech 2012). Now, they have developed a 60 GHz wireless module with high frequency usage efficiency, *i.e.*, a data rate of 6.57 Gbps in the PHY layer that uses a 2.16 GHz bandwidth, based on the use of a 6 dBi slab-waveguide antenna (developed by Ando-Hirokawa Labs at Tokyo Tech), a 65 nm CMOS 60 GHz direct-conversion radio-frequency (RF) LSI and analog circuit with 40 nm CMOS process that includes a 2.3 GSample/s 7-bit analog-to-digital converter (developed by Matsuzawa-Okada Labs at Tokyo Tech), and a 40 nm CMOS baseband (BB) LSI that incorporates a media-access control (MAC) layer and PHY layer that uses the above analog circuit and rate-compatible low-density parity-check (LDPC) codes⁸ with code rates of 14/15 and 11/15 (developed by Sony). The design of this 60 GHz wireless module is based on the first draft of the IEEE802.15.3e⁹ standard. They also developed a file transfer system with a high cache memory capacity that can be accessed directly from the wireless module with very high throughput. A 60 GHz wireless transfer system using the developed wireless module and file transfer system demonstrated the world's fastest user data rate of 6.1 Gbps (which can transfer a 1 GB file in 1.3 s). The system enables users to receive large quantities of data in moments, without the low data throughput limitations of current commercial mobile devices.



Figure 24 – Photographs of a 60 GHz 6.1 Gbps wireless module (left) and of experimental setup for wireless transfer of files to a smartphone (right)

We have established an actual system that allows multiple wireless systems (hereafter called the GATE systems) installed adjacently each other to be operated independently without interference to demonstrate the high throughput and spatial isolation abilities of the 60 GHz wave-based wireless devices, *e.g.*, a ticket gate at a train station.

⁸ Rate-compatible LDPC codes are LDPC codes that were designed to enable decoding using a single decoder.

⁹ IEEE 802.15.3e is the next-generation 60 GHz wave-based wireless communication standard with a maximum PHY data rate of 100 Gbps and a maximum link-set-up time of less than 2 ms, and is currently being discussed.

A high-gain slot-array antenna (using approximately 1000 elements in experiments) that enabled spatial isolation was developed by Tokyo Tech (at Ando-Hirokawa Labs). In addition, the radio waves do not spread out and are confined for more than 10 m in a cylindrical service area.

In the scenario where users pass through the communication area within a short period of time, Sony has also implemented a MAC protocol on the RF-BB LSIs, enabling reduced link-setup times that allow users to start communications within 2 ms or less. JRC has integrated these technologies to form the GATE system.



Figure 25 – Photographs of the 60 GHz GATE wireless system

To locate the service area for the 60 GHz GATE system, which is a small portable access point, in arbitrary positions in the large zone as quickly as possible, easy installation-type radio link systems to accommodate GATEs are advantageous.

We performed a successful field demonstration of an example configuration that allowed a combined operation of the 60 GHz band GATE system and the 40 GHz band wireless access system with the maximum link length of 1 km or more with 1 Gbps-class speed.

In the 40 GHz band wireless access system used here, the directional division duplex (DDD) system was adopted to perform simultaneous two-way communication on the same frequency and the same polarized wave, rather than the conventional frequency division duplex (FDD) or time division duplex (TDD) methods; DDD doubled the frequency utilization efficiency in principle.

The realization of DDD was only enabled by full use of high-isolation between transmitting and receiving antennas arranged in parallel and cancellation technology of transmitted signal leaked in the circuit.



Figure 26 – Photographs of the 40 GHz wireless system using DDD

Localized torrential rainfall can lead to the disconnection of mmWave links, because mmWaves are attenuated by water. A routing control method that is based on the prediction of rain attenuation avoids potential drops in the communication capacity of mmWave access networks caused by rainfall. When the area of rain is advancing towards the mmWave access network, the routing control method predicts the mmWave links that will be affected by the rainfall, and then selects alternative mmWave links to replace them. A proportion of the network traffic is then passed to the selected links proactively to reduce the drops of and ensure the capacity of the access network.

A man in a white shirt is pointing his right index finger towards the text 'WLAN'. The background is a textured grey wall. The text 'Connect' is on the left, 'Internet' is in the middle, and 'WLAN' is on the right, all in white bold font on semi-transparent rectangular backgrounds.

Connect

Internet

WLAN

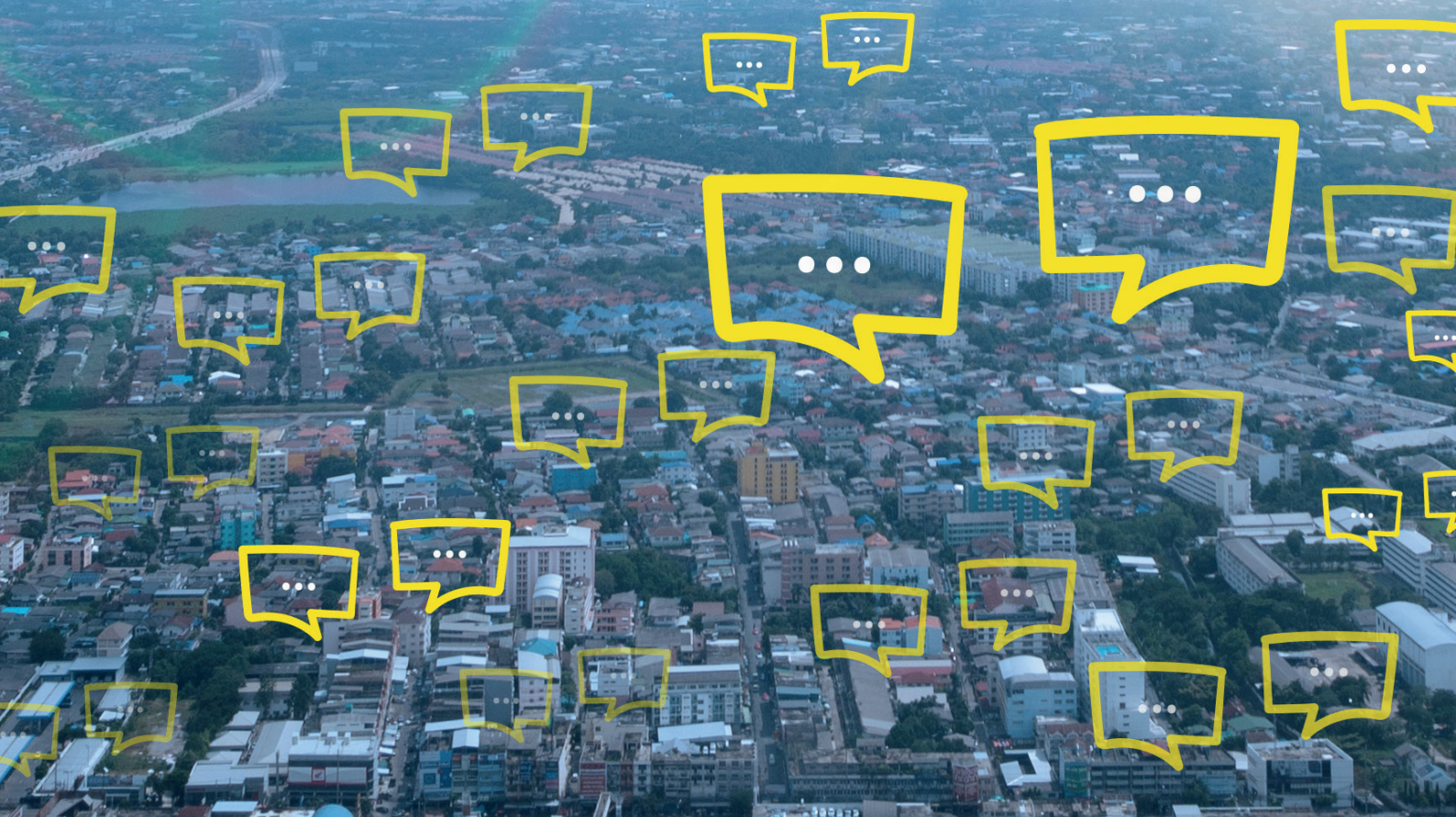
5.

Fixed and Mobile Convergence



Hotspot

5G



Requirements of IMT-2020 Fixed Mobile Convergence



Summary

This document aims to define the requirements on IMT-2020 fixed and mobile convergence over cellular access, fixed access and WLAN access networks. This document will address, but not limited to, consistent users experience, access independence, unified user plane, connection management, interworking, and charging and accounting for fixed and mobile convergence in IMT-2020. This document is intended to be the input of the FG IMT-2020 to its parent group SG13 for further development of this document.

Table of Contents

1	Scope
2	References
3	Definitions
3.1	Terms defined elsewhere
3.2	Terms defined in this document
4	Abbreviations and acronyms
5	Conventions
6	Overview of IMT-2020 FMC
6.1	General objectives of IMT-2020 FMC
7	IMT-2020 FMC service requirements
7.1	Access service support
8	IMT-2020 FMC capability requirements
8.1	Customer-oriented ubiquitous broadband access and service environment
8.2	Unified User Plane
8.3	Access independence
8.4	Quality of service
8.5	Mobility management
8.6	Connection management
8.7	Interworking
8.8	Accounting and charging
8.9	Security



1 Scope

The document describes the requirements and capabilities to support fixed and mobile convergence in IMT-2020 networks.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this document are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

The reference to a document within this document does not give it, as a stand-alone document, the status of a Recommendation.

3 Definitions

3.1 Terms defined elsewhere

This document uses the following terms defined elsewhere:

3.1.1 Fixed mobile convergence [Q.1762/Y.2802]: In a given network configuration, the capabilities that provide services and application to the end user defined in [ITU-T Y.2091] regardless of the fixed or mobile access technologies being used and independent of the user's location.

3.2 Terms defined in this document

None.

4 Abbreviations and acronyms

This document uses the following abbreviations and acronyms:

FMC	Fixed and Mobile Convergence
WLAN	Wireless Local Area Network
FTTx	Fibre To The X
QoS	Quality of service
BRAS	Broadband Remote Access Server

5 Conventions

In this document:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this document is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option, and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with this document.

6 Overview of IMT-2020 FMC

6.1 General objectives of IMT-2020 FMC

The following points provide the general objectives for IMT-2020 FMC:

- Seamless service operation from the end user perspective across heterogeneous fixed networks and mobile networks, subject to any limitations imposed by the characteristics of the particular access technology being used.
- Seamless service provisioning from the service provider's perspective across heterogeneous fixed and mobile networks, subject to any limitations imposed by the characteristics of the particular access technology being used.
- Generalized mobility is supported in FMC (i.e., terminal device mobility, user mobility and session mobility). For a given scenario, different levels of mobility may be needed.
- Ubiquity of service availability where the end users can enjoy virtually any application, from any location, on any terminal device subject to any limitations imposed by the characteristics of the particular access technologies and terminal devices being used, given that the service has been subscribed.

7 IMT-2020 FMC service requirements

The IMT-2020 FMC service should be ubiquitously available. The end users can enjoy any application, from any location, on any terminal device subject to any limitations imposed by the characteristics of the particular access technologies and terminal devices being used.

The following sub-clauses identify the service requirements for IMT-2020 FMC.

7.1 Access service support

IMT2020 FMC is required to support access-independent service for users.

IMT2020 FMC is required to allow the user to select a suitable access connection to obtain service.

8 IMT-2020 FMC capability requirements

8.1 Customer-oriented ubiquitous broadband access and service environment

8.1.1 Description

With the blooming of mobile Internet, cloud computing, big data, and Internet of Things (IoT), as well as industry Internet, the service objects of broadband networks will be further extended in the coming five to ten years, from 5 billion people to 50 billion IoT terminals, and the broadband access service scenarios are going ubiquitous, from ultra-broadband backbone, ultra-broadband access to ubiquitous broadband IoT.

The broadband access technical, such as IMT broadband network, WLAN broadband, and fixed broadband are developing harmoniously to build a ubiquitous top-notch broadband network, where the advantages of mobile broadband (convenience and wide coverage) and wired broadband (high bandwidth and reliability) complement each other. FTTx and WLAN are used indoors, and IMT outdoors for GE fixed access and FE mobile access. Ubiquitous broadband network together with plentiful applications such as smart home, smart city, can offer users ubiquitous, seamless and perfect broadband experience.

8.1.2 Requirements

IMT-2020 FMC is required to provide the capabilities of customer-oriented ubiquitous broadband access environment, i.e., access application servers or service platforms in a wide range including mobile, WLAN and fixed access network.

8.2 Unified User Plane

8.2.1 Description

Control plane and user plane is a key design principle of IMT-2020. IMT-2020 core network user plane is simplified to support basic function like DPI, forwarding, QoS related traffic control and so on. It is similar with access gateway of fixed network (e.g., BRAS). Maybe, these core user plane functions can be unified to realize the unified access control, QoS, mobility and so on.

8.2.2 Requirements

IMT-2020 FMC is required to provide the capability of unified network user plane including IMT-2020 core network user plane function and access user plane of fixed network.

8.3 Access independence

8.3.1 Descriptions

The IMT-2020 network is envisioned to be an access network-agnostic architecture whose core network will be a common unified core network for emerging new radio access technologies for IMT-2020 as well as existing fixed and wireless networks (e.g., WLAN). The access technology-agnostic unified core network should be accompanied by common control mechanisms which are decoupled from access technologies.

8.3.2 Requirements

Services and features offered via IMT-2020 FMC are required to be access-independent so that services are offered to users regardless of how they gain access.

8.4 Quality of service

8.4.1 Descriptions

The IMT-2020 network is expected to be able to provide the required QoS for a variety of different devices, different services, and different traffic characteristics. This requirement enables service level agreements to support user and service requirements.

8.4.2 Requirements

IMT-2020 FMC is required to provide unified QoS mechanisms.

8.5 Mobility management

NOTE – To be added.

8.6 Connection management

8.6.1 Descriptions

The connection management capability is used by a multi-connection UE and a multi-connection capable network to establish, release, and modify connections.

8.6.2 Requirements

IMT-2020 FMC is required to support the management of multi-connections. And this capability is required to provide unified control all connections used by multi-connection UE.

8.7 Interworking

8.7.1 Descriptions

Different types of mobile communication schemes have different characteristics and coverage, in the migration phase to the IMT2020 network, network deployment is not always fully consistent with existing network coverage. When UE leaves out of IMT-2020 coverage to existing network coverage, IMT-2020 core network is required to support interworking with existing LTE network to guarantee services.

8.7.2 Requirements

IMT-2020 FMC is required to support more flexible and optimized multi-RAT interworking.

IMT-2020 FMC is required to support multi-RAT, WLAN and fixed broadband interworking.

8.8 Accounting and charging

8.8.1 Descriptions

IMT-2020 FMC is required to support the ability to collect accounting and charging related information from different access networks, which are used by the charging/billing system to gather together relevant usage data to initiate an unified bill to the specific user for multiple kinds of services with different terminal devices.

8.8.2 Requirements

IMT-2020 FMC is required to provide unified charging and accounting capabilities for multiple access networks including mobile, WLAN, and fixed access networks.

8.9 Security

8.9.1 Descriptions

Security requirements such as access control, authentication, non-repudiation, data confidentiality, communication security, data integrity, availability, and privacy are required to support services in FMC network.

8.9.2 Requirements

IMT-2020 FMC is required to provide unified security mechanisms to meet service requirements.

Suggestions to SG13

The following issues should be studied further, but not limited to:

1. It is better to separate the FMC requirements from service perspective and ones from the network evolution.
2. Mobility management related requirements needs to be addressed further.
3. FMC requirements should be aligned with high level requirements of IMT-2020 in SG13.

Contributors (in Alphabetical Order)

This is the list of all contributors who submitted valuable comments or contributions.

- Aipeng GUO China Unicom
- Namseok KO ETRI
- Shin-Gak KANG ETRI
- Wei CHEN China Mobile
- Yachen WANG China Mobile





**Unified Network
Integrated Cloud
for Fixed Mobile
Convergence**

Summary

This document proposes a technical report on a unified cloud based Fixed and Mobile Convergence architecture in IMT-2020 which includes the motivation of the architecture and the related key technologies. This document will also address FMC service requirements in IMT-2020 and how the whole IMT-2020 network architecture including fixed and mobile network should evolve to be more efficient, cost-effective, and flexible to accommodate new services.

This document is the initial study on IMT-2020 FMC architecture and there are many other issues to be studied further based on this baseline document in the next period of SG13.

Keywords

FMC, Control and User Plane Separation, Service Chain, SDN, NFV, Cloud

Table of Contents

1	Scope
2	References
3	Abbreviations and acronyms
4	Overview
5	Architecture
5.1	Functional Architecture Model
5.2	Deployment Model
5.3	Functional Components
5.4	Reference points
6	Key technologies
6.1	Control and User Plane Separation
6.2	Reconstruction of control plane
6.3	Service Chain
6.4	User Data Convergence
6.5	Network Function Virtualization and Software Define Network
Appendix A – Architecture of the existing networks	
A.1	Fixed broadband
A.2	WLAN
A.3	2G/3G
A.4	4G
A.5	5G

Contributors (in Alphabetical Order)



1 Scope

This document proposes a cloud-based unified fixed and mobile convergence network architecture in IMT-2020 network (called Unified Network Integrated Cloud) with the definition of related functions and key technologies.

2 References

- [1] Recommendation ITU-T Y.2320 (2015), *Requirements for Virtualization of Control Network entities in Next Generation Network evolution*.
- [2] ETSI GS NFV 001 V1.1.1 (Oct. 2013), *Network Functions Virtualisation (NFV); Use Cases*.
- [3] Recommendation ITU-T Y.3300 (2014), *Framework of software-defined networking*.

3 Abbreviations and acronyms

FMC	Fixed and Mobile Convergence
CUPS	Control and User Plane Separation
SDN	Software Define Network
NFV	Network Function Virtualization
UNIC	Unified Network Integrated Cloud

4 Overview

Traditionally, telecom networks for different access technologies, i.e., cellular, fixed broadband and WLAN, have been developing separately and become verticals (see Appendix). Some significant drawbacks have been identified:

- Each vertical network has its own user identity, charging and billing system. Subscribers have to use specific identity for each network and deal with bills for each network. There is lack of service consistency and continuity across the vertical networks. All these decrease the user experience.
- Each vertical network is maintained separately and deep coordination between these networks is hard, while in fact some of equipment can be the same or shared across networks from the maintenance point of view, which will increase the OPEX and CAPEX.

To resolve the issues above, this report provides a cloud based FMC architecture for next generation network, called Unified Network Integrated Cloud (UNIC), in which different kinds of access technologies are aggregated. The following questions are suggested to be taken into account:

- Can control plane functions be reused among different access networks, e.g., identity, charging, user data?
- Can control plane functions be merged or coordinated to realize unified control among different access networks?
- Can UP functions be merged or shared?
- Can Value-add functions be shared among different access networks?

UNIC is facilitated by technologies such as Control and User Plane Separation, Service Chain, NFV [1][2], SDN [3], cloud and other key technologies. Since UNIC is a unified FMC architecture across different access technologies, it has the following advantages:

- Similar functionalities can be reused, which can save CAPEX;
- Single and isolated network can be provided and maintained which can save the OPEX;

- Unique user identity and billing, service consistency and continuity across different access technologies, which improves the user experience.

5 Architecture

5.1 Functional Architecture Model

The UNIC architecture is shown in Figure 5-1:

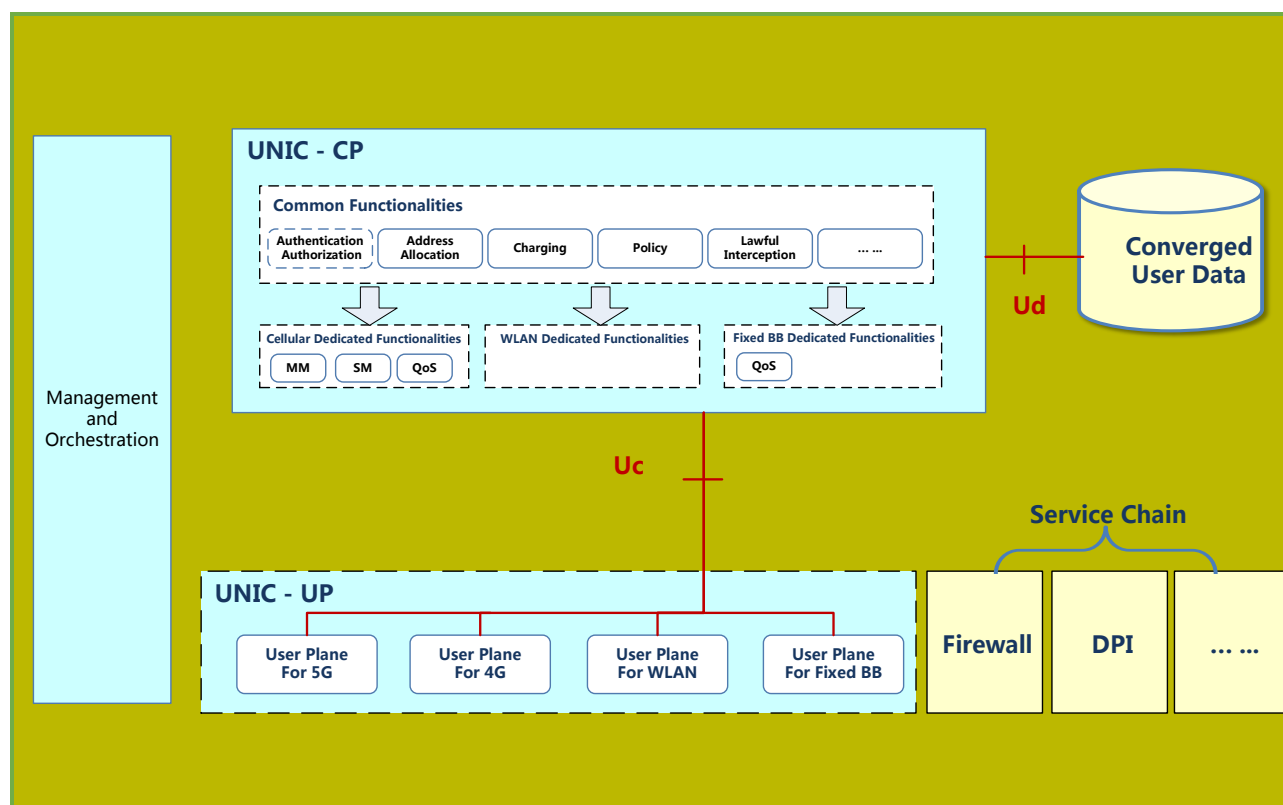


Figure 5-1 – Functional Architecture of UNIC

UNIC is composed of five main components:

- UNIC-CP
- UNIC-UP
- Converged User Data
- Service Chain
- Management and Orchestration

The core function of the architecture is divided into control plane and user plane functions, which are UNIC-CP and UNIC-UP in this architecture. UNIC-CP takes most of the control logics, while UNIC-UP mainly provides packets switching under instruction of UNIC-CP. The Converged User Data is the central data repository in this architecture, in which most of the permanent and temporary data is stored. The Service Chain is a supplementary to the UNIC-UP which provides user plane enhancement features.

5.2 Deployment Model

As shown in Figure 5-2, UNIC functional components can be deployed on the Telecom Integrated Cloud (TIC), NFV/SDN based common infrastructure with management and orchestration. However, some UP functions may be implemented based on dedicated hardware to reach better performance.

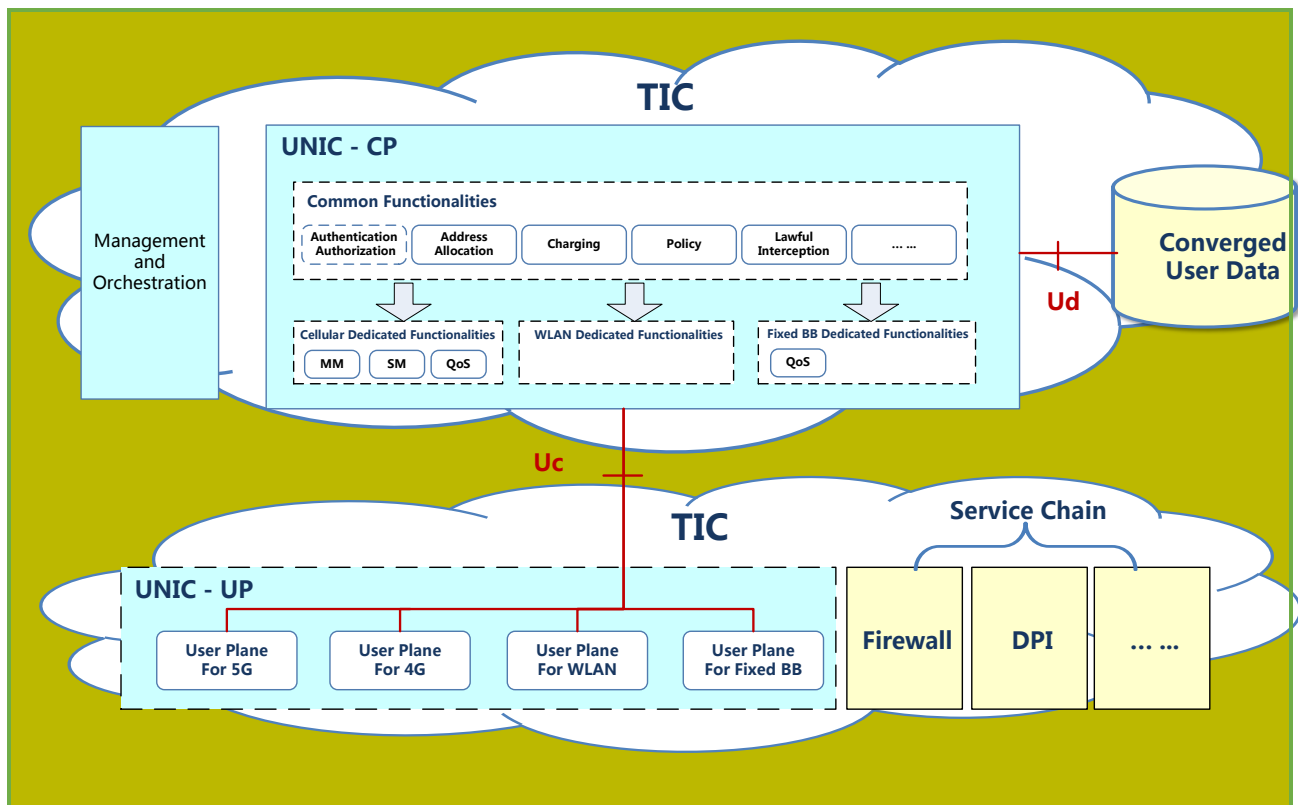


Figure 5-2 – Deployment model of UNIC

5.3 Functional Components

5.3.1 UNIC-CP

The UNIC-CP component is the core component of UNIC which provides the control plane function. The functionalities of UNIC-CP are categorized into common functionalities and dedicated functionalities.

The common functionalities are those required more than one access technologies and the logics are similar or the same. The UNIC-CP has the following functionalities:

- authentication and authorization
- address allocation
- charging
- policy
- lawful interception
- etc.

During runtime, different access technologies which share the common functionalities can use a single common functionality instance.

The dedicated functionalities are those required only by a particular access technology, so there can be couples of sets of dedicated functionalities and each of those will fulfil the requirement of a specific access technology. Mobility management and Session Management required by cellular networks are examples of the dedicated functionalities.

5.3.2 UNIC-UP

The UNIC-UP component provides the user plane function, and works under control of UNIC-CP. The main task of UNIC-UP is packets switching along with a minimum set of necessary logics. The UNIC-UP also reports user plane events to the UNIC-CP on demand of the later.

There can be significant differences across access technologies in the following aspects:

- due to different designs of networks, user plane entities needs to be deployed more centralized for some access technologies, while needs to be more distributed for others.
- functional requirements are different across access technologies. (NOTE – 5G UP is simplified more like IP routing after separating DPI, charging out if possible)
- performance requirements are different across access technologies.

The UNIC-UP contains the following user plane modules:

- user plane for 5G
- user plane for 4G
- user plane for fixed broadband
- user plane for WLAN

Each of the user plane modules is aiming to serve its corresponding access technology. But it is not precluded that different access technologies use the unified user plane module in the future.

5.3.3 Converged User Data

The Converged User Data component is the central data repository of UNIC. All the subscription data e.g. user identity and runtime state data e.g. restoration data are stored in Converged User Data component. See section 8.4 for more details.

5.3.4 Service Chain

The Service Chain related functional component is an enhancement to UNIC-UP. It provides functionalities such as firewall, DPI, etc. The Service Chain functionalities are shared across access technologies and are invoked on demand. Service chain can be an efficient mechanism to realize flexible add-in and removal of value-added services. See section 8.3 for more details.

5.3.5 Telecom Integrated Cloud

The telecom integrated cloud is characterized by NFV and SDN, which is the underlying infrastructure on which the 5G functions are deployed. The Cloud provides “general purpose” hardware, software platform and related management/orchestration function to fulfil the function deployment requirement and resource management requirement. There can be different kinds of infrastructure requirements for different UNIC function components, such as infrastructure requirement aiming for control logics serves UNIC-CP, infrastructure requirement aiming for packets switching serves UNIC-UP, infrastructure requirement aiming for data storage serves Converged User Data.

5.4 Reference points

5.4.1 Uc

The Uc reference point is between UNIC-CP and UNIC-UP. Uc reference point provides the following functionalities:

- deliver traffic steering rules
- policy installation
- event reporting (e.g., reporting user traffic volume)

The protocol used Uc reference point can be GTP, OpenFlow, H.248, etc.

NOTE – Whether interface between different types of UP and CP can be unified is FOR FURTHER STUDY.

5.4.2 Ud

The Ud reference point is between UNIC-CP and Converged User Data. Ud reference point provides the following functionalities:

- authentication

- user data downloading
- user data update
- information query
- restoration

6 Key technologies

6.1 Control and User Plane Separation

Control and User Plane Separation is to decouple the control plane function and the user plane function in the network entity, and each function is implemented in separate equipment. The control plane function can then be deployed in a more centralized way while the user plane function can be deployed more distributed according to the service requirement. The upgrading, scaling can be applied separately on control plane and user plane functions. Control and User Plane Separation can bring the advantages such as more flexibility of network upgrading, faster new service deployment, etc.

Control and User Plane Separation makes it possible to integrate different access technologies under a single network architecture due to the following reasons:

- the composed network entities are too complicated to be converged
- the different user plane functions needs to be deployed centralized or distributed depends on the access technology it serves.

With Control and User Plane Separation, different user plane functions can be deployed on demand while under control of the unified control plane function.

6.2 Reconstruction of control plane

Network slicing is a concept introduced in 5G, by which operators can deploy different network slices to serve different service requirements. A network slice is an end-to-end network, including RAN and CN. The functions in network slice can be virtualized or dedicated. Network slicing, together with NFV/SDN, can enable a flexible, programmable, and extendable network architecture.

Function modularization, customization and composition are the basis of network slicing with NFV/SDN. When redesigning the functions of control plane functions, the following issues should be taken into account:

1. Control plane functions of fixed network should be included
2. Mobile network control plane function should be included
3. WLAN control plane function should be included
4. Which functions can be common?
5. Which functions should be specific?
6. Which functions should support unified control to these different networks?

Besides, orchestration of control plane is also important for network slicing. That can enable creating a network slice which includes corresponding control functions required by service requirement.

6.3 Service Chain

In order to adapt to the rapid development of data communications services, fixed broadband and mobile operators will develop a variety of value-added services such as video optimization, Web cache, HTTP header enhancement and network acceleration based on different user groups and market demand. With the development of value-added services, the value-added service network is gradually evolving into the status of serial connection or hairpin connection, which will lead to data traffic roundabout, management and maintenance difficulties, low business deployment flexibility and time consuming problems.

Service chain can be introduced in the mobile or fixed broadband communication network to break the traditional serial or hairpin mode of value-added services. It can provide customized value-added services for

different users and applications based on time, location, network status and other dimensions. Moreover, the open interface can be designed for third parties to achieve lower business access threshold, rapid on-line and off-line business, unified management and reduced maintenance costs.

At the same time, with the popularization of mobile Internet and home broadband, access modes for fixed broadband and mobile access and the needs of value-added services are constantly converging. Service chain architecture facilitates the implementation of unified traffic management strategy of fixed and mobile convergence, improves resource utilization, and further reduces deployment costs and expenses.

6.4 User Data Convergence

In 4G networks, user data is stored and managed by HSS. And in fixed networks, the user data equipment is AAA Server. Data separation results in key drawbacks, such as data redundancy and service development difficulties. In user data management respect of 5G networks, it is important to focus on user data convergence.

User data convergence means not only to merge user data in HSS and AAA together that belong to the same user, but also to provide unified user data management. Besides, It is expected to store the status data of a large number of users, e.g., registered, unregistered. With the help of NFV, a large scale data-base could provide this service of bulk storage. Due to data convergence of one user and convergence of lots of users' data, it brings advantages of less maintenance and facilities e.g. service development and big data analysis.

From the view of service layer and control plane, user data convergence shall provide multiple kinds of access protocols. And at the same time, to improve the efficiency of a large scale data-base, it is important to define unified access protocols as few as possible.

6.5 Network Function Virtualization and Software Define Network

Network Function Virtualization (NFV) and Software Defined Network (SDN) are the foundation of future network.

Network function virtualization transforms our network from dedicated hardware to common hardware. Network function can be implemented by software. Management and orchestration are introduced to realize unified resource management and allocation, to support auto-scaling, flexible function deployment and quick business on-line.

SDN realizes the separation of control plane and user plane, supports unified management of network connection, and realizes intelligent and optimized traffic scheduling.

Appendix A

Architecture of the existing networks

A.1 Fixed broadband

The architecture of fixed broadband is shown in Figure A-1.

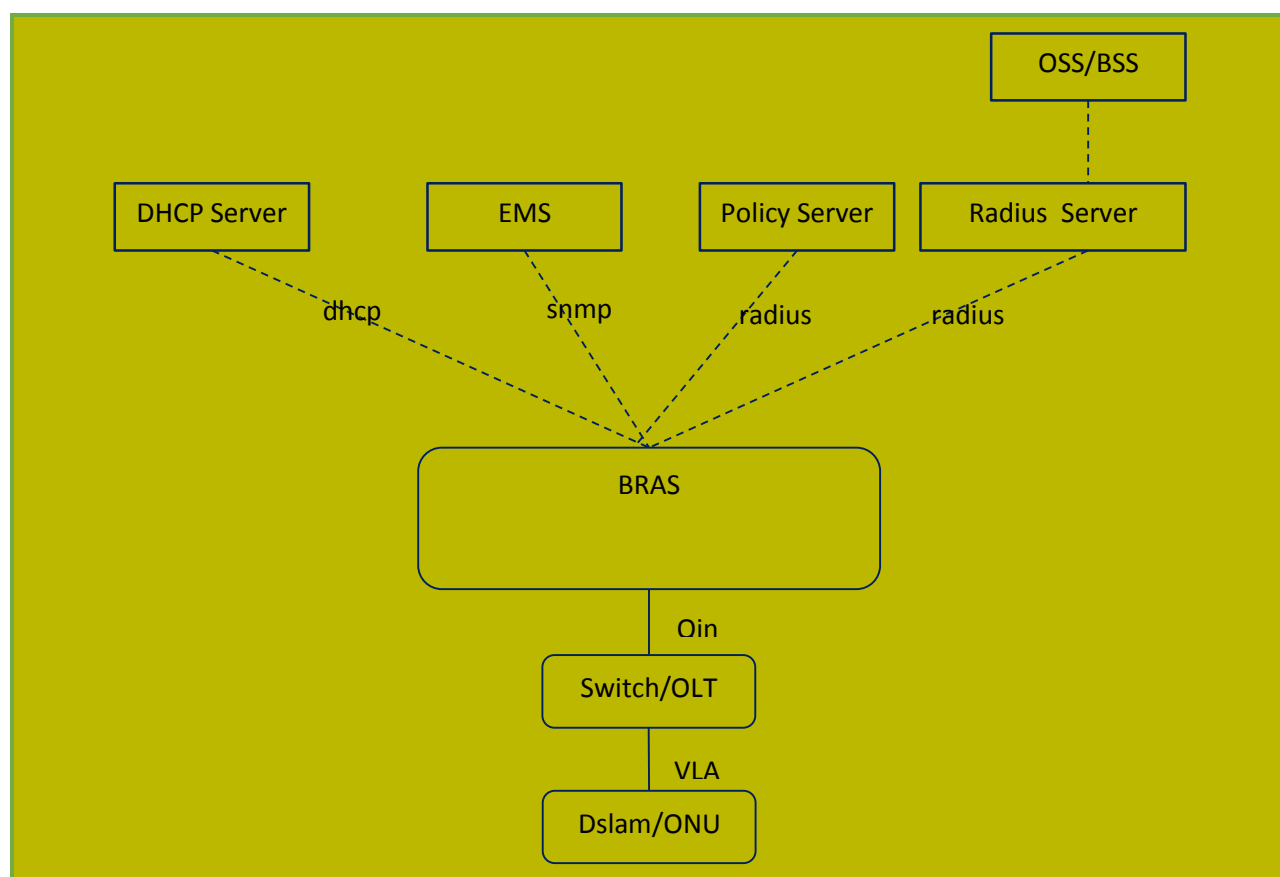


Figure A-1 – Architecture of fixed broadband

- **BRAS** (Broadband remote access server): BRAS is the aggregation point for the subscriber traffic. It is a specialized server based at the edge of an Internet service provider (ISP) network that facilitates the convergence of multiple Internet traffic sources and user sessions. These sources include cable, DSL, Ethernet or broadband wireless. Beyond aggregation, it is also the injection point for policy management and IP QoS in the Regional/Access Networks.
- **Radius** (Remote Authentication Dial-In User Service) **Server**: After users dialing into the ISP, BRAS parses the username and password, and passes them to RADIUS server, which checks whether the information is correct and authorizes access to the ISP system, then send packets to BRAS. BRAS will allocate IP address to users, and send packets to Radius Server which can trigger charging.
- **Policy Server**: One server which can adjust the user's QoS information based on Radius COA message. It can speed up intelligently, but it is also not mandatory to set up such server.
- **DHCP Server**: User's IP address will be allocated by external DHCP server if BRAS is DHCP Relay.
- **EMS**: BRAS and EMS can exchange configuration and get information by SNMP/NETCONF protocols.

A.2 WLAN

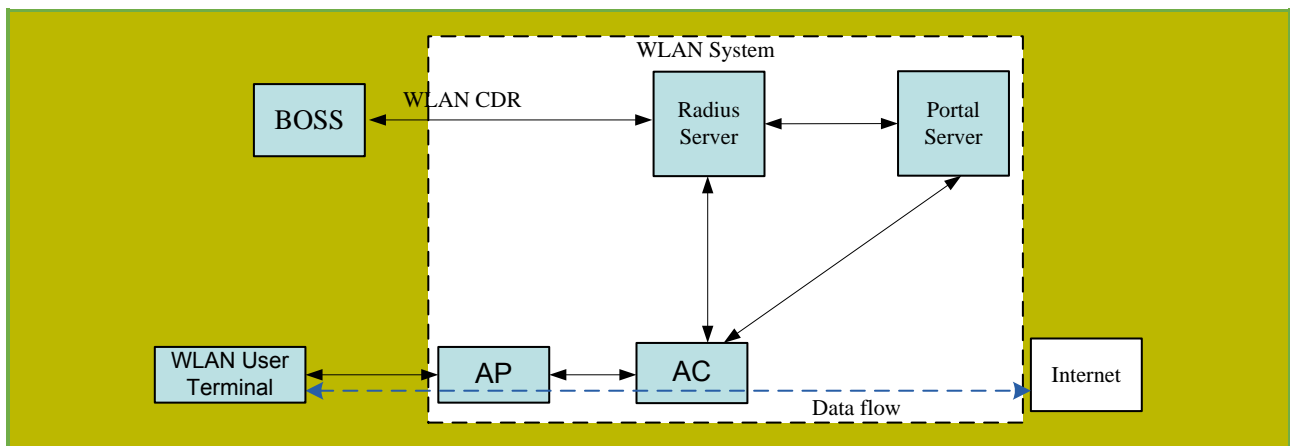


Figure A-2 – WALN Logical Architecture

As shown in Figure A-2, the WLAN logical system is composed of AP (Access Point), AC (Access Controller), portal server and radius server.

WLAN access system includes AP and AC mainly used for user access control, charging data collection and service management. The AP must support 802.11a/b/g/n protocols and ensure compatibility with different WLAN terminals. The AC is composed of user access control functionality and radio control functionality logically.

The functionalities of Radius server can be categorized into user data repository, service authentication, password management, charging and roaming support. The charging information including consumed data volume, duration and user identifier collected by AC can be transferred periodically to Radius server by Radius messages, and finally the WLAN CDRs formed by AC are transferred to BOSS (Business and Operation Support System) for accounting.

The main function of Portal server includes authentication web page push and off line notification.

A.3 2G/3G

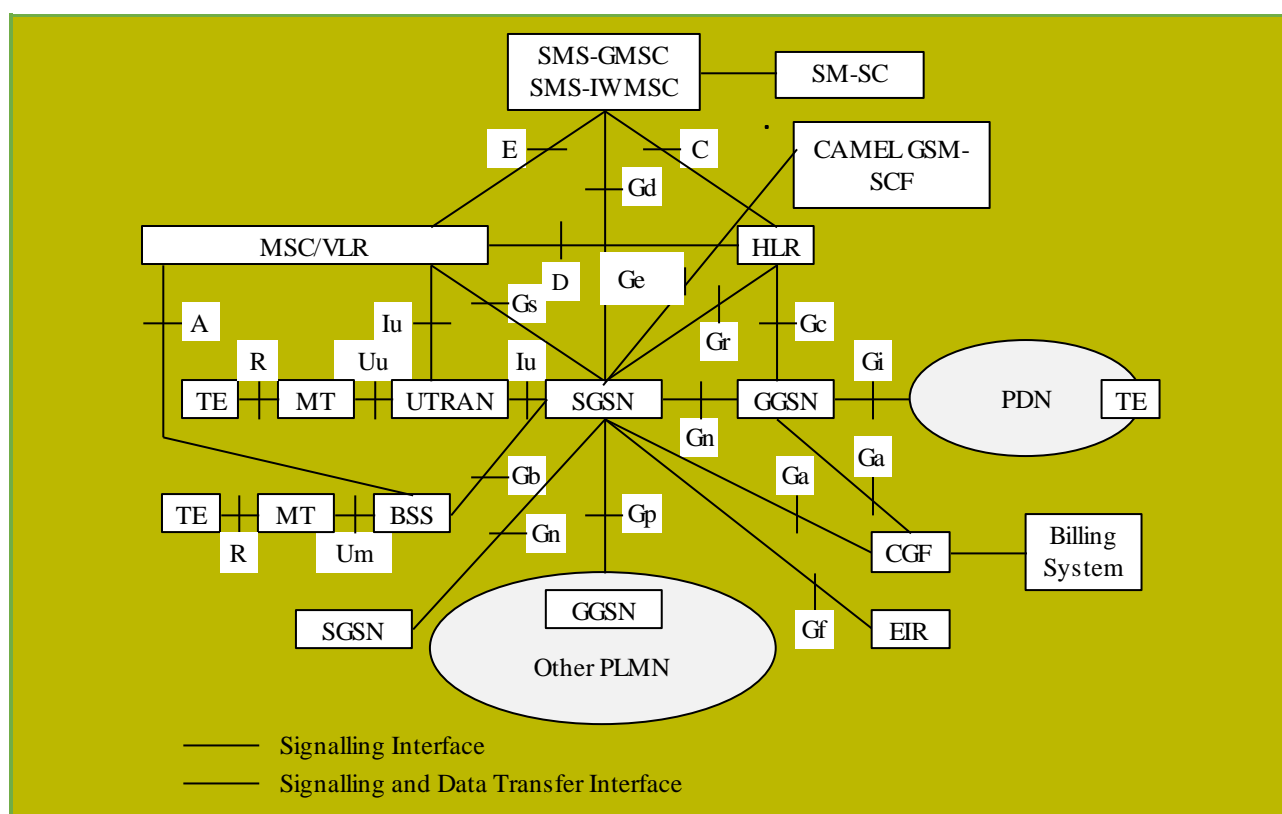


Figure A-3 – 2G/3G Logical Architecture

The architecture of 2G/3G Logical Architecture is described in Figure A-3. The 2G/3G Packet Switched Network functionality is logically implemented on two network nodes, the SGSN (Serving GPRS Support Node) and the GGSN (Gateway GPRS Support Node).

A GPRS Support Node (GSN) contains functionality required to support GPRS functionality for GERAN and/or UTRAN. The GGSN is the node that is accessed by the packet data network. It contains routing information for PS-attached users. The routing information is used to tunnel N PDUs to the MS's current point of attachment, i.e. the Serving GPRS Support Node. The GGSN is the first point of PDN interconnection with a PLMN supporting GPRS. GGSN functionality is common for all types of RANs.

The SGSN is the node that is serving the MS. The SGSN supports GPRS for A/Gb mode and/or Iu-mode. At PS attach, the SGSN establishes a mobility management context containing information pertaining to e.g. mobility and security for the MS. At PDP Context Activation, the SGSN establishes a PDP context, to be used for routing purposes, with the GGSN that the subscriber will be using.

When the SGSN and the GGSN are in different PLMNs, they are interconnected via the Gp interface. The Gp interface provides the functionality of the Gn interface, plus security functionality required for inter-PLMN communication. The security functionality is based on mutual agreements between operators.

A.4 4G

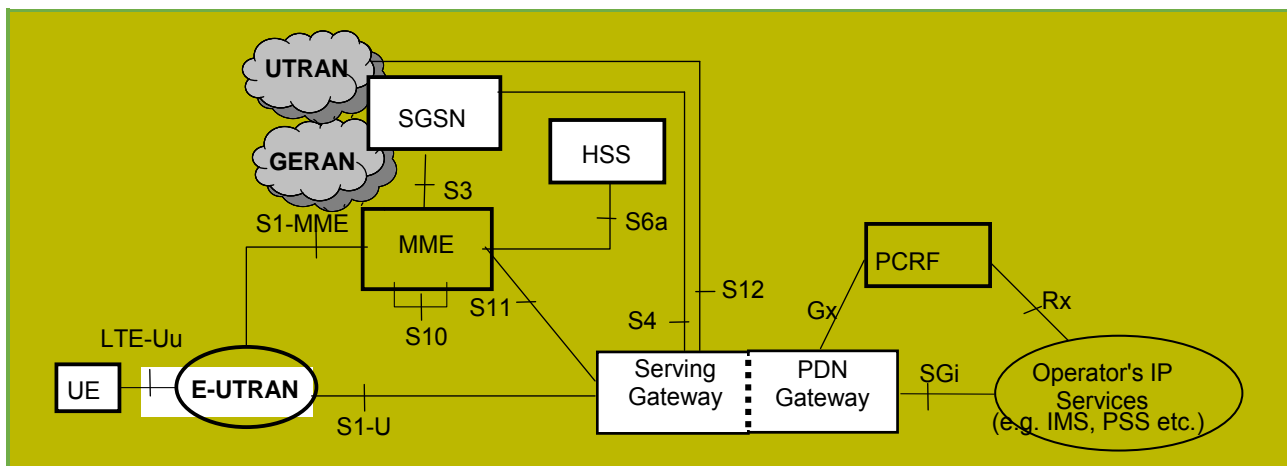


Figure A-4 – Non-roaming EPS architecture for 3GPP accesses

The EPS (Evolved Packet System) provides IP connectivity using the E-UTRAN (Evolved Universal Terrestrial Radio Access Network).

The EPC (Evolved Packet Core) network is the core network of EPS, and only Packet Switched services no Circuit Switched service could be accessed by EPC network, and Mobility Management Entity (MME), S-GW (Serving GW), P-GW (PDN GW), HSS (Home Subscriber Server) are the main network elements, and the non-roaming EPS architecture for 3GPP accesses is shown as Figure A-4. The multi-mode LTE-capable terminals could support interworking and service continuity between 2G/3G and LTE networks by S3/S4/S12 interfaces.

According to 3GPP TS 23.401, MME is the pure control plane element in charge of mobility management, NAS signalling, P-GW & S-GW selection, Authentication, Idle mode UE Tracking and Reachability. Two logical Gateways, S-GW and P-GW exist in EPC network. The PDN GW and the Serving GW may be implemented in one physical node or separated physical nodes. S-GW is responsible for the Local Mobility Anchor point for inter-eNB handover, Mobility anchoring for inter-3GPP mobility and Packet routing & forwarding, and the P-GW is the logical element for Policy Enforcement, Per-user based packet filtering, Charging, UE IP address allocation and so on.

In some operators' commercial networks, S-GW and P-GW may be combined with GGSN, and MME may be combined with SGSN, so the 2G/3G and LTE packet switched networks are converged for supporting multiple accesses, such as GERAN, UTRAN, E-UTRAN, even non-3GPP access like WLAN. The converged 2G/3G/LTE network supports interworking and service continuity between 2G/3G and LTE, multi-mode LTE-capable terminals may connect over 2G/3G when no LTE present.

A.5 5G

NOTE – To be filled in after IMT-2020 architecture is elaborated.

Suggestions to SG13

The following issues should be studied further, but not limited to:

1. The scope of FMC architecture
 - a) Clarification is needed on whether the FMC architecture is a part of IMT-2020 network architecture or an interworking architecture between fixed networks and IMT-2020 mobile networks?
 - b) Clarification is needed on whether the interfaces or protocols between different types of UP and CP can be unified for wireless and fixed networks?
2. The application of network slicing on FMC should be addressed.
3. How to balance between the flexibility and latency in service chaining scenario to introduce new services?
4. Detailed functional components should be further studied including common functions.
5. Which option is more appropriate to realize unified control between the introduction of new functions and the enhancement of existing functions?

Contributors (in Alphabetical Order)

This is the list of all contributors who submitted valuable comments or contributions.

–	Aipeng GUO	China Unicom
–	Bin WEI	China Mobile
–	Bing WANG	China Mobile
–	Byung Jun AHN	ETRI
–	Kefeng ZHANG	ERICSSON
–	Namseok KO	ETRI
–	Peter ASHWOOD-SMITH	Huawei Technologies
–	Shin-Gak KANG	ETRI
–	Shujun HU	China Mobile
–	Wei CHEN	China Mobile
–	Yachen WANG	China Mobile
–	Yue SONG	China Mobile



International
Telecommunication
Union

Place des Nations
CH-1211 Geneva 20
Switzerland

ISBN 978-92-61-25111-6



Published in Switzerland
Geneva, 2017

Photo credits: Shutterstock