# Towards extreme network KPIs with programmability in 6G

Gergely Pongrácz, Atilla Mihály, István Gódor
Ericsson Research
Budapest, Hungary
{gergely.pongracz},{attila.mihaly}, {istvan.godor}@ericsson.com

Sándor Laki
Faculty of Information Systems
ELTE Eötvös Loránd University,
Budapest, Hungary
lakis@inf.elte.hu

Anastassios Nanos
NUBIS S.A.
Athens, Greece
ananos@nubis-pc.eu

Chrysa Papagianni
Institute for Informatics
University of Amsterdam
Amsterdam,
The Netherlands
c.papagianni@uva.nl

## ABSTRACT

6G's superpower must be simplicity, which should not be viewed as a constraint, but rather as the organic outcome of using the most advanced technologies at our disposal. Programmability in the data plane, cloud-native features like automatic scaling and failover, transparent acceleration of both network functions and applications and AI-driven optimizations are already present. We only need to integrate these different innovations into a consistent architecture and offer a simple yet powerful solution for the very different applications that would use future mobile networks. The application space is getting more and more heterogeneous, e.g., legacy Internet-based services still using the good old TCP protocol, future media services relying on multipath transport - always utilizing the best available connection, or control applications of robots or drones requiring extreme low and stable latency. The different applications will require very different Key Performance Indicators (**KPI**s) from the network. In this paper, we present a novel architecture called DESIRE6G (**D6G**) architecture that aims to fulfill these requirements by integrating the key technological innovations mentioned above. Besides supporting the diverse KPIs of future applications, the novel architecture should also simplify the mobile network itself by promoting modularity and service-based network function selection which can replace traditional control plane centric solutions, e.g., for handover.

## CCS CONCEPTS

• Networks → *Network architectures; Mobile networks.*

## KEYWORDS

Programmable networks, 6G, manageability, efficiency, resiliency, extreme network KPIs


MobiHoc '23, October 23–26, 2023, Washington, DC, USA

ACM ISBN 978-1-4503-9926-5/23/10…$15.00
https://doi.org/10.1145/3565287.3617610

**ACM Reference format:**

Gergely Pongrácz, Attila Mihály, István Gódor, Sándor Laki, Anastassios Nanos and Chrysa Papagianni. 2023. Towards extreme network KPIs with programmability in 6G. In *ACM MobiHoc '23 6G Programmable Deterministic Networking with AI (6G-PDN) workshop, October 23, 2023, Washington, DC, USA*. *ACM, New York, NY, USA, 6 pages.* https://doi.org/10.1145/3565287.3617610

## 1 Introduction

While 5G networks are currently being rolled out commercially, the research community is already thinking about how to improve the capabilities of networks even further: a sixth-generation or 6G-network to be rolled out in early 2030s. The question is: why? 5G and its evolution to 5G-Advanced [1] already enables a wide range of use cases, but it seems the new use cases take off slower than expected.

Envisaged future challenges beyond 2030 require a more adaptable and dynamic architecture. 6G will naturally push the performance limits even further, but it must do it without creating a more complex system. Accommodating different services (e.g., URLLC and eMBB) to a 5G network requires complex network management and control functionality to allocate proper network functions (**NF**s), employing network slicing to cope with the different service requirements with potentially isolated network resources. Thus, both OPEX and CAPEX of such heterogeneous networks could significantly increase [2]. Fortunately, we have every tool at hand to simplify and optimize it: programmable data planes [3] can make the network not just more flexible and resilient, but also more performant, as it will be possible to do more service-oriented optimizations. Cloud-native technologies can introduce simpler scalability and resiliency methods. Distributed, fast control loops can increase reaction speed towards service assurance .

Naturally, data transmission over the wireless medium is prone to errors hence wireless access links will always be vulnerable. Application developers might need to adapt to this reality rather than pursuing increasingly stringent KPIs. On the other hand, pushing the limits of the network further has a clear advantage: make our imagination the biggest limitation when it comes to shaping new use cases or services. The network-compute fabric should not be the limiting factor - not just technologically, but also financially. This is our ultimate goal; introduce flexibility and easy manageability and at the same time decrease complexity. This is the reason for applying automatic acceleration features in the data plane and autonomic optimizations in the control plane.

The paper is organized as follows: Section 2 describes the high-level architecture of the D6G system [4][5], Section 3 discusses the aspects of QoS and service definition, while Section 4 shows the details of some example network functionalities that can be solved in a novel way.

Between the D6G sites there can be non-programmable elements. These can be controlled by Software-Defined Networking Controller(s) (**SDNC**) or by traditional routing protocols. The minimum requirement is that the D6G infrastructure has to understand the reachability of the other sites from each given site (e.g., IP addresses of the other site's gateway).

The main components and their roles are described in the following subsections. Service Management and Orchestration (**SMO**) in 2.1, Multi-Agent-based network intelligence System (**MAS**) and telemetry in Section 2.2, and the Programmable Data Plane (**PDP**) layer including the Infrastructure Management Layer (**IML**) and the HW/SW specific implementations of the NFs (Infra/Custom NF-DPs in the figure) in Section 2.3.

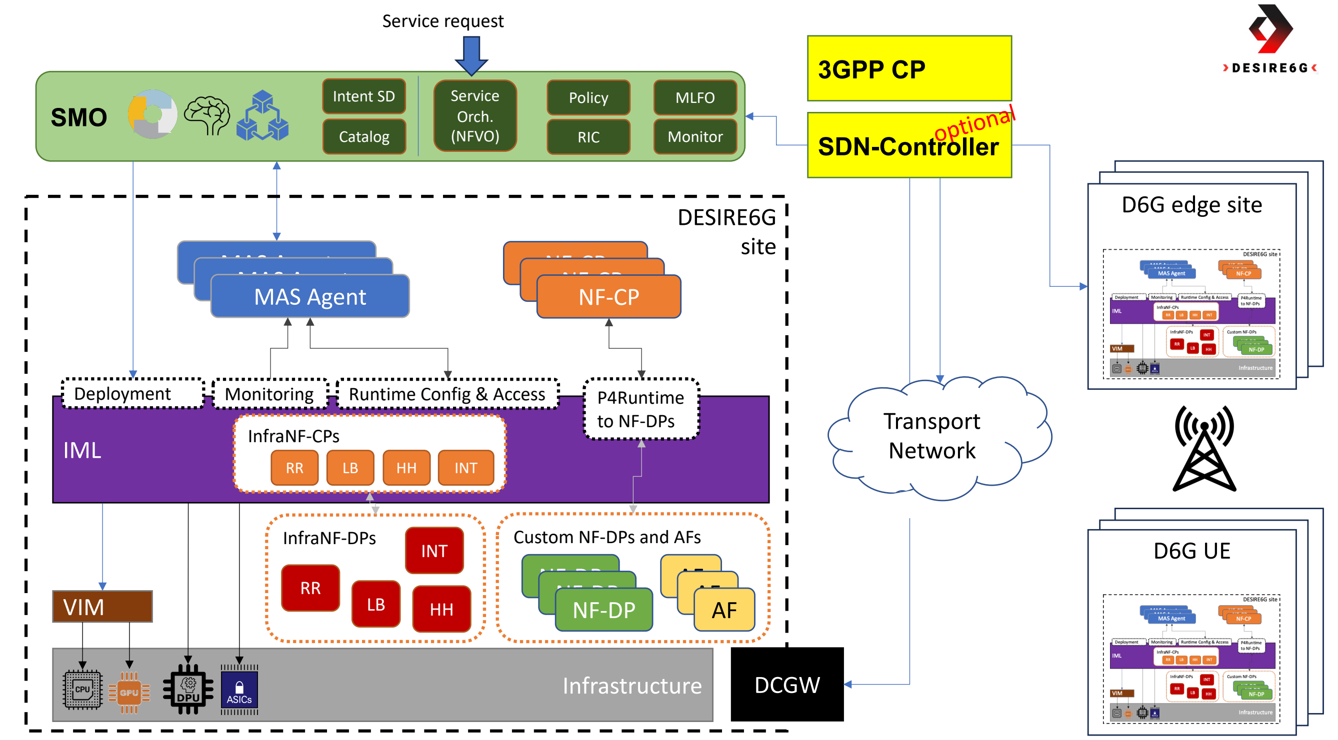


**Figure 1: Desire6G high-level architecture**

## 2 High-level D6G architecture

Figure 1 shows the components of the D6G system. The network consists of topologically separated D6G sites, including local HW and compute resources running NFs necessary for the execution of the network services. Each D6G site has almost identical setup with slight variances based on necessity, e.g., the sites running Radio Access Network (**RAN**) functionality are equipped with sufficient hardware entities and/or non-programmable (static) functions. All of which are still considered NFs by the system. Also note that the User Equipment (**UE**) might also be able to run a stripped-down version of the D6G stack using its own hardware and software stacks. This is a big difference compared to existing UE approaches and it makes end-to-end service control possible. Of course, legacy UEs can still use the system as today attaching to some "edge-to-edge" service, e.g., normal Internet access.

### 2.1 Service Management and Orchestration

Similar to existing specifications (i.e., NFV orchestrator for ETSI MANO), the **SMO** is responsible for end-to-end service life-cycle management, including service provisioning and deployment, network slice management and network optimizations. It contains several modules responsible for these different tasks and it is the main interface towards the external world. See the SMO modules in Figure 1 for illustration. The main functions are the following:

- *Service catalog*: service descriptions are stored in the catalog. This is a rather static subsystem: service introduction is done with setting the service graphs for the required service along with other parameters, such as QoS handling preferences. The interface towards the SMO uses entries in the catalog most probably with user-friendly description (e.g., name).

- The catalog entries (service descriptions) can be defined manually but can also be set via an *intent-based service definition* framework [6].
- *Service orchestrator* (and high-level network function orchestrator - NFVO) is the key entity at service deployment. Orchestration might take place statically before UEs start using the service, but there is also an option to make (partial) deployment when a user is attaching to the given service – we will call this the "dynamic" case (see Section 3). The orchestrator has full understanding of the operator's D6G sites including site resources and inter-site connections. It instructs the IML (see Section 2.3) that plays the role of the VNF Manager in the ETSI MANO terminology while also wrapping VIM functionality, to deploy NF-sub-graphs to selected sites. It is also involved in full or partial service re-deployments when requested by the underlying MAS.
- *Policy framework* and *non-real-time RIC*: these are entities already available in the O-RAN [7] ecosystem. They are responsible for policies and generic optimizations on medium to long timescales. Thus, the D6G SMO integrates O-RAN SMO functionality.
- *MLFO*: the machine learning function Orchestrator that manages and monitors all the elements and resources that build an AI pipeline.
- The *Monitoring* service including a cloud data lake, that is the repository where data (processed telemetry information, events etc.) is stored.

## 2.2 Multi-agent-based Network Intelligence

The D6G MAS [8] implements distributed network intelligence closer to the physical infrastructure, as it is responsible for receiving service-specific monitoring information and fine-tuning the network and compute resources. It configures and uses a pervasive telemetry system to receive service specific KPIs, e.g., by monitoring end-to-end latency for latency-sensitive or latency-critical services. The components of the telemetry system are illustrated in Figure 2:

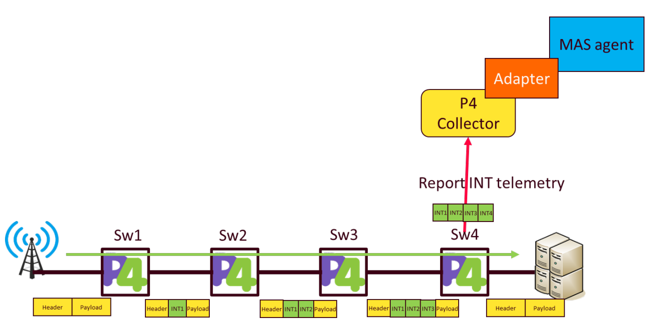


**Figure 2: in-band telemetry collection for MAS**

With the help of the PDP layer telemetry can take place in the packet path itself (see the green In-band Telemetry [9] – INT headers in Figure 2), which allows very accurate measurements with the possibility to correlate the end-to-end performance with the per-node behaviour, without the need for specially crafted probe packets. Telemetry info can be put to the D6G header as metadata and carried over to the other end of the service graph where a report can be generated and sent via an adapter logic to the MAS function.

MAS can apply AI algorithms for optimization purposes and can have several roles based on its actual purpose: it can be domain-specific like the transport or the RAN optimizer or service specific with end-to-end knowledge about the given service and the used domains. The different agents share information with each other and this way they can solve complex, network-wide optimization or fault handling functionality.

## 2.3 End-to-end programmable data plane (PDP)

The E2E Programmable Data Plane is employing NFs to carry out the logic of the selected service. NFs contain a Control Plane component (**NF-CP**) and the packet processing Data Plane logic (**NF-DP**). Note that at this level the main role of NF-CP is to configure and update the objects (e.g., tables, registers, etc.) in the corresponding NF-DP. These are separated by the IML responsible for transparent scaling, acceleration, and deployment, i.e., it is the bridge between the logical and the physical network function. The NF decomposition is visible in Figure 3. IML hides the implementation and deployment details of NF-DP from NF-CP by providing a simple unified view of the data plane. For example, even if NF-DP is vertically or horizontally disaggregated into multiple data plane programs running on different HW/SW targets, IML ensures that NF-CP only sees a single NF-DP instance and IML handles the complexity of managing the disaggregated packet processing components.

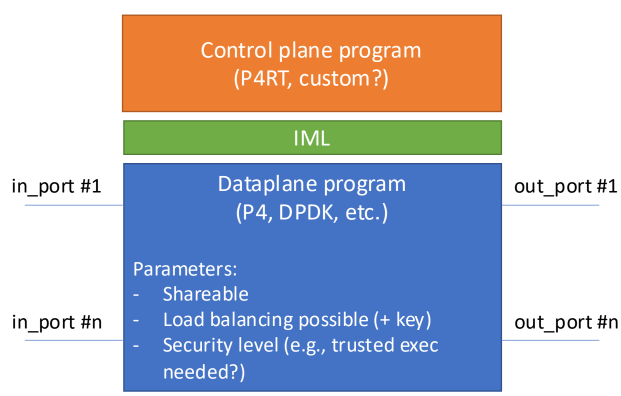


**Figure 3: Internal structure of a network function**

The IML uses the Virtualized Infrastructure Manager (**VIM**, e.g., Kubernetes) to deploy physical instances of the PDP components, and it also handles the deployment of Application Functions (**AF**s) by proxying the request to the VIM. Its main features include:

- *HW-offload in CP-agnostic way*: the NF-DP can run on various platforms and in various physical instances, yet the NF-CP will only see one logical entity.
- *NF aggregation with CP separation*: multiple NF-DPs must run on the same HW to support multitenancy. In this case it is mandatory to aggregate the respective NF-DPs and expose an isolated view to the NF-CP entities.
- *NF disaggregation* is the opposite: having an NF-DP and splitting it to two (or more) smaller programs yet still showing a single point of control for the respective NF-CP. *Load balancing* is an example in which case the NF-DP is automatically combined with a load balancer NF.
- *Adaptation to non-PDP domains*: between D6G sites there might be non-PDP transport domains in which case the proper encapsulation and site addressing is necessary. This is independent of the logic of the service and should be done automatically.
- *Automatic heavy-hitter offload*: in a multi-hardware setup, typically the highest performing targets have the most limited memory, or they have other constraints like length of the logical pipeline (stages). On the other hand, CPU targets are not limited by such constraints, but have lower performance. In such setups users with the heaviest traffic load should be assigned to the hardware unit(s). This is typically achieved by the business logic itself, which leads to several different implementations. IML can bridge this gap and do automatic heavy hitter offload for all NFs that can run on multiple targets.
- *Optimal routing between NF-DPs, including collocation optimization*: traditional service chaining suffered heavily from I/O overhead. To mitigate this, IML will optimize NF-DP paths with collocation (NF-DP aggregation), optimized or HW-accelerated I/O.

After discussing the main components, Section 3 will talk about the main elements and procedures.

# 3 QoS, slicing and services

D6G's main concept revolves around two main logical elements: slices and services. This section tries to shed light on how they are defined and what their relation is.

## 3.1 QoS and slicing concept

A simple slicing model similar to the ETSI ZSM was established as can be seen on Figure 4. During service definition, the queuing parameters of the QoS class can already be established. The latency related configuration cannot be set at this point (routing, site selection), that is done during user attachment – in which case the "endpoint" of the service for that particular user becomes available.

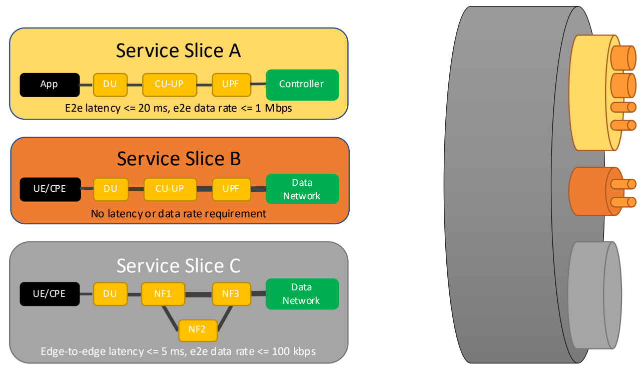


**Figure 4: Slices and services**

Between different slices, the usual resource sharing method is weighted fair queuing, e.g., set the bandwidth ratio between the slices to X:Y. Other methods can also exist, e.g., strict priority for a given slice – typically used for "mission critical" services. The example on Figure 4 shows a simple system with 3 different service slices. Note that inside a service it is possible to treat attached users differently, e.g., setting a different weight based on user classes.

## 3.2 Service definition

Service in the D6G context equals to the end-to-end behaviour that the system provides. The service is described by a service graph (or NF-graph), as shown on Figure 5.

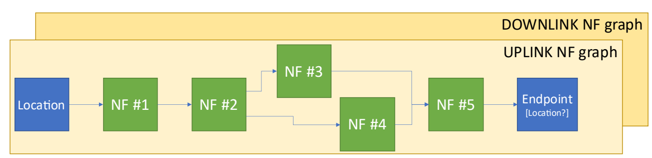


**Figure 5: Service description as NF graph (or service graph)**

The service graph is described both in uplink and downlink direction, as there could be different behaviour based on the direction.

In D6G there are two types of services:

- *Edge-to-edge*: this means that the app and the server are not part of the service graph. This is the "legacy" service type, e.g., Internet access, where only the access part is controlled by the operator. Slice B and C on Figure 4 are examples for this.
- *End-to-end*: the service graph includes applications (e.g., UE app – edge app), meaning that the entire end-to-end path is controlled by the D6G system. This is preferable for latency critical applications. Slice A on Figure 4 is an example for end-to-end slices.

There is also a possibility to describe hybrid services, e.g., end-to-end up to an application frontend, but behind the frontend traditional cloud applications or microservices can run and offer the real service to the UE through the frontend.

A service can be selected by the UE at any time after the UE is connected to the network. A UE may be active in multiple services at the same time. The simplest mapping inside the UE is via using different OS level interfaces, but other types of mapping are also possible, especially if the UE can run its own D6G stack.

From the QoS point of view a service belongs to a given QoS class. For simplicity, we assume that each service runs in a different service slice. Note that a complex application might have different service flows that would be best mapped to different QoS classes.. In theory it would be possible to set different QoS classes inside one service slice, but the service itself will also have parameters that will be used to describe QoS treatment (like end-to-end latency).

## 3.3 Logical service to physical deployment

During deployment the logical view of a service with its service graph is transformed to an intermediate view with multiple infrastructure functions such as load balancers or transport adapters. These are needed for the actual physical deployment. During this step it is possible to instantiate the service-specific MAS instances and required telemetry functions. It can be part of the service template, but the MAS can also request telemetry functions during the attachment of a user. This way it will be possible to execute user-specific end-to-end telemetry collection used e.g., for root cause analysis.

Note that if the "dynamic" approach is used, meaning that the necessary NFs and Application Functions (AFs) are deployed on a per-request basis (i.e., only upon request from a UE), then whenever a new UE attaches to the service the SMO has to check whether the selected D6G sites already have running NFs (and possibly AFs) that are required by the service sub-graph on the given site. Before this step (UE attach) the endpoint locations are unknown, so end-to-end latency related configuration (e.g., site selection, routing, latency budget distribution) is not possible.

An example enriched service graph used for physical deployment can be seen on Figure 6.

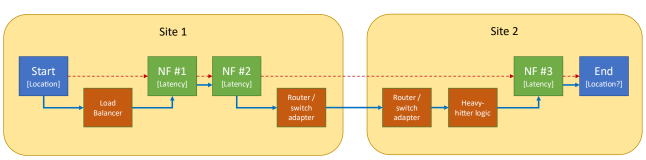


Figure 6: The enriched service graph

By separating the infra-NFs from the business logic NFs we can achieve similar advantages as using cloud for running applications: automatic scaling and acceleration, independence from the underlaying network infrastructure and reliability. After this intermediate step, the IML can do the actual physical deployment at each site (see Figure 7 for illustration). IML can put together the combined NF-DP code wherever it is possible and can ask the VIM to start workers.

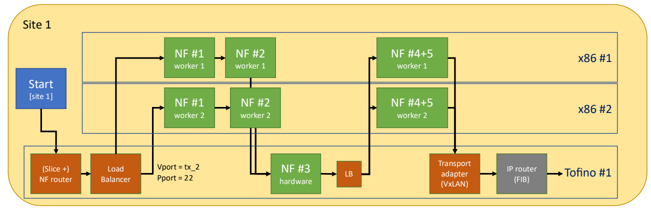


Figure 7: Physical mapping

After the deployment the service components are running on the given site, so the UEs can start attaching to the corresponding service.

## 3.4 Physical deployment to UE attach

After the required components are running on all selected sites, the NFs and AFs need to know that a new UE is about to attach to the given service. Note that the system needs to inform the respective NF-CPs only. Once it is done, the proper NF-DPs will be configured via the IML. For some NFs or AFs, it is essential to know the user as the function is user specific. In D6G it is possible to carry a generic "user ID" field as metadata this way making the life of user-specific functions easier.

There are multiple ways to convey the user ID to the PDP layer.

- The legacy way is to use the 3GPP control plane entities. They can be directly connected to the respective NF-CPs.
- Another possible approach is to use the SMO to proxy this info to the NF-CPs. This assumes that the SMO is in the loop when a user wants to attach to a specific service (the "dynamic" case) and so it will understand site selection for that given user.
- A further option is to use MAS being able to trigger user-specific monitoring (or monitoring a group of users, e.g., 5% of the population).

# 4 Runtime procedures

Once the user is attached to a certain service, the network behavior is described by the service graph (cf. Figure 5). In the following it is described how the system works with many users and many services, where most of the NFs are shared between the different users and a specific user employs a specific NF path on the graph.

## 4.1 NF routing

NF routing is a key component of the PDP layer. It has two distinct tasks: when a packet enters the D6G system it needs to identify which service it belongs to, and when a D6G-internal packet is received it needs to send it to the next NF (or AF) in the graph.

The "service ID" can be put into the D6G header helping the NF Routing (**NFR**) function. But this alone is not enough. Many users can share the same service and their site mappings are partially different. In uplink functions have many to 1, or at least many to less mapping, while in downlink direction to find the actual edge and/or RAN site the user is active at the NFR must make a lookup on the “user ID” too. Note that this is only valid when the NFR notices that the rest of the graph is on a different site. Until that step simple “service ID” based routing is available between the NFs.

To make the NFR more scalable it is possible to set a “location” parameter together with the “service ID” when the packet first enters the D6G system, similarly to SRv6-based NF-chaining approaches [10]. This way the inter-site lookups can be simplified: instead of a “user ID” → “next site” mapping the NFR can use the “location” parameter, which itself is a RAN site, but can also be used as a lookup input for intermediate (edge) sites – here we assume that the proper intermediate site can be selected if the user location is known. If that is not the case, as e.g., there are multiple intermediate sites equally equipped to serve a given location then the selection can only be made with the help of some load balancing / equal cost routing function. Since this is a not very likely scenario here the details are not discussed.

## 4.2 Handover

The legacy way to do handovers is to do it in the mobile system, e.g., use path switch so the GTP endpoint will change and will point to another CU-UP (another RAN site). However in D6G, due to the use of the NFR this functionality is redundant. In D6G mobility will be simply a change in the service graph since the UE “location” in the graph changes. In the UL graph that does not really make a difference, but in DL the UPF (or similar NF) must send the packets according to the new position. Note that this method can also be used between RAN entities, such as CU-UP and DU or even below DU and in the proposed form it is transparent to the NFs.

But how is this achieved? In the previous subsection it was already discussed that NFR needs to know the user location to do proper inter-site NF routing. It was also stated that either the NFR is user-aware when it comes to intra-site transport, or the graph selector can help by setting a “location” parameter for the given user.

Figure 8 shows a simple handover: the user moves from RAN site #1 to RAN site #2. At the target site it needs to establish connection to a base station. There are two main options there:

- Option #1: After allowing the UE to join the target base station can simply start using the service mapping it learned from the source base station. This way the NFR of the more central site(s) will see a change in the service graph and can learn the new position of the user. Note: this requires that the inter-site transport function of the NFR is user-aware.
- Option #2: The target base station (or the NFR) can report the location change to the SMO, which will do a partial redeployment of the user’s service: it will check the NFs at the target site and configure the IML and this way the NFR with the new location.

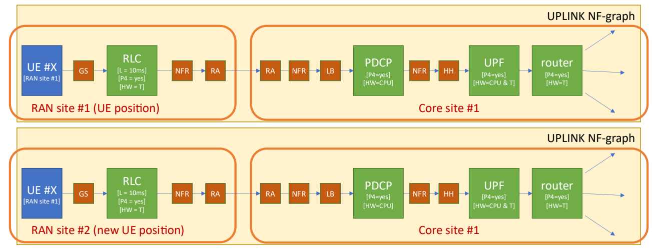


**Figure 8: handling mobility with partial graph change (top: UE is at source base station – RAN site #1) bottom: UE is at target base station – RAN site #2)**

## 5 Summary and future work

The proposed D6G architecture exploits the latest advances of programmable hardware accelerators to enable the customization of end-to-end packet processing and forwarding. It also introduces a new level for runtime management/control based on a distributed multi-agent system to enable fast reactions to changing network/service conditions. The implementation of the D6G architecture has just started. We expect that it will further evolve during the realization of different components and the preparation for being interoperable with existing O-RAN and 3GPP components.

## ACKNOWLEDGMENTS

DESIRE6G has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union’s Horizon Europe research and innovation programme under Grant Agreement No 101096466.

## REFERENCES

[1] Lin X. An overview of 5G advanced evolution in 3GPP release 18. IEEE Communications Standards Magazine. 2022 Sep;6(3):77-83.
[2] Khan BS, Jangsher S, Ahmed A, Al-Dweik A. URLLC and eMBB in 5G industrial IoT: A survey. IEEE Open Journal of the Communications Society. 2022 Jul 12;3:1134-63.
[3] Michel O, Bifulco R, Retvari G, Schmid S. The programmable data plane: Abstractions, architectures, algorithms, and applications. ACM Computing Surveys (CSUR). 2021 May 3;54(4):1-36.
[4] DESIRE6G project. Link: https://desire6g.eu
[5] C. Papagianni and G. Pongracz.. DESIRE6G: Deep Programmability & Secure Distributed Intelligence for Real-Time E2E 6G Network. SNS Journal (2023): 50-52,[Online]https://smart-networks.europa.eu/wp-content/uploads/2023/05/sns-journal2023-web-1.pdf
[6] Leivadeas A, Falkner M. A survey on intent based networking. IEEE Communications Surveys & Tutorials. 2022 Oct 20.
[7] Polese M, Bonati L, D’oro S, Basagni S, Melodia T. Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. IEEE Communications Surveys & Tutorials. 2023 Jan 23.
[8] L. Velasco, M. Ruiz, F. Paolucci, A. Sgambelluri, L. Valcarenghi, and C. Papagianni. 2023. Pervasive Monitoring and Distributed Intelligence for 6G near Real-Time Operation. In European Conference on Networks and Communications (EuCNC)/
[9] Scano D, Paolucci F, Kondepu K, Sgambelluri A, Valcarenghi L, Cugini F. Extending P4 in-band telemetry to user equipment for latency-and localization-aware autonomous networking with AI forecasting. Journal of Optical Communications and Networking. 2021 Sep 1;13(9):D103-14.
[10] Matos G, Verdi FL, Contreras LM, de Almeida LC. When srv6 meets 5g core: Implementation and deployment of a network service chaining function in smartnics. arXiv preprint arXiv:2107.11966. 2021 Jul 26.