# User Plane Management Function: A Solution for Automatic Deployment of UPF on cloud-native 5G Core Network Architecture


Kieu-Ha Phung*, Luong Vu Dung
*Hanoi University of Science and Technology*
Ha Noi, VietNam
ha.phungthikieu@hust.edu.vn
dung.lv190095@sis.hust.edu.vn

Quang Tung Thai, Namseok Ko
*Network Research Division, ETRI,*
Daejeon, Korea
tqtung@etri.re.kr & nsko@etri.re.kr



***Abstract*—5G and beyond mobile networks have emerged to provide QoS stringent services for the huge number of user devices with several technological enablers including edge computing, network slicing, SDF, NFV. The Service Based Architecture 5G facilitates the separation of control plane from data plane. The service establishments are directed by network functions in the control plane, while data session establishment are served by User Plane Functions located at the edge. Various UPFs combining to working network slices, serving as the anchor points for user data traffic plays a pivotal role in ensuring seamless data delivery, quality of service (QoS), edge computing, and security. This article proposes a solution to effectively manage UPF, a new network function called User Plane Management Function integrates between SMF and UPF. The point-to-point packet forwarding control protocol is replaced by the Service Based Interface. By implementing our solution, UPF deployment on each network slice can be automatically configured, easily scalable, and efficiently managed; opening the way to deploy UPFs in the cloud-native infrastructure.**



***Index Terms*—5G and beyond, Cloud Computing, Service Based Architecture**


## I. Introduction

Advanced technologies in 5G enable complex services relating to diverse use cases such as enhanced Mobile Broadband (eMBB), massive Machine Type Communication (mMTC) and ultra-Reliable Low Latency Communication (uRLLC). One of the pioneering technology is the network transformation, from a monolithic in nature lacking flexibility, scalability, agility, and support to diverse services, to a Service Based Architecture (SBA) network, allowing the in corporation of different cutting-edge technologies such as Network Function Virtualization (NFV), Software Defined Networking (SDN), Multi-access Edge Computing (MEC) and Network Slicing.

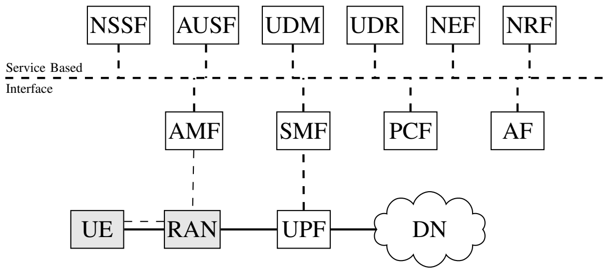
In SBA 5G, all the network elements are defined based on Network Functions (NFs), as shown in Figure 1. A set of Network Functions (NFs) provide services to other authorized NFs. NFs are interfaced via Application Peripheral Interface (APIs) in client-server model. Traditional telecom signalling messages are replaced with API calls on a logically shared service bus, hence, modularity, scalability, reliability, cost-effective operation, easy deployments, and faster innovation are some of the benefits of moving to SBA.

In SBA 5G, the major change also reflects in the transition from a point-to-point protocol to consumer-producer communication paradigm. In the point-to-point protocol, both consumer and producer must establish a direct channel of communication between them, hence, requiring to have prior knowledge of each other's existence and identifiers, so the implementation is very static, difficult for provisioning, scaling. In the consumer-producer or client-server model of SBA 5G, service consumers will discover suitable network services available through service discovery and registration mechanisms to obtain connection information. This mechanism is implemented by the Network Repository Function (NRF) - storage of information about available services and their respective manufacturers. SBA-5G is much more flexible and easy to scale, and enables cloud-based deployment.

In current version of 3GPP documents, user plane and control plane are separated. User Plane Function (UPF) serving as the anchor point for user data traffic in 5G networks is not available in SBA, especially the transmission between UPF and Session Management Function (SMF) uses point-to-point protocol, which poses challenges on scalability, and orchestration. That's why we need an effective UPF management mechanism.

In this paper, we propose a new design architecture for UPF, a new network function is added to monitor and manage various of UPFs to support various working slices for heterogeneous types of service. Our proposed implementation is able to integrate to ETRI architecture[13] for 5G and beyond core network, in supporting multiple working slices as well as finding the optimal paths on a specified working slice according to the UE requirements.

The remainder of the paper is organized as follows. In Section II, we discuss UPF and the challenges that need to be addressed. Section III proposes our solution architecture and implementation design. Section IV illustrates Proof of concept setup, and conclusion and future work is given in Section V.




| | |
|---|---|
| AUSF | Authentication Server Function |
| AMF | Access and Mobility Management Function |
| AF | Application Function |
| DN | Data Networks |
| SMF | Session Management Function |
| NEF | Network Exposure Function |
| NSSF | Network Slice Selection Function |
| NRF | Network Repository Function |
| PCF | Policy Control Function |
| UDM | Unified Data Management |
| UPF | User Plane Function |
| (R)AN | (Radio) Access Network |

Fig. 1. Overview 5G Service-Based Architecture[3]

## II. User Plane Function in 5G core architecture

UPF provides the interconnect point between the mobile infrastructure and Data networks. Its role is to create, maintain a tunnel from an UE to a data network to transport UE's application data flows. It plays a crucial role in the 5G network in facilitating low latency and high throughput required service applications. It serves as an essential component that evolves the data plane aspect of the Control and User Plane Separation (CUPS) strategy which aim to separate control plane functions from user plane functions, enabling packet processing and traffic aggregation to occur either in the core network or closer to the network's edge.

The User Plane Function has four distinct reference points[3]:

- N3: Interface between the RAN (gNB) and the (initial) UPF;
- N4: Interface between SMF and the UPF.
- N6: Interface between DN and the UPF;
- N9: Interface between two UPFs (i.e the Intermediate UPF, I-UPF, and the Session Anchor UPF);

The PDU session establishment procedure occurs after a UE has already established a connection, registered, and authenticated with the network (Figure 2). The UE initiates the procedure by sending a Pdu session establishment request to AMF. The AMF then relays the essential information to the SMF. In response, the SMF sends a message to the AMF, acknowledging or rejecting the session establishment. Notably, the SMF may query the UDM for subscription data if it is not available locally.

The following steps, calling on a PCF to find the optimal paths across UPFs are optional according to the standard but are crucial for edge computing supporting enhanced mobile broadband or mission-critical applications. Initially, the SMF must determine which PCF to use for the UE. In the simplest scenario, only one PCF is available. Alternatively, the SMF may base its decision on local configuration or information obtained from NRF. Subsequently, the SMF sends a request to the PCF, aiming to create a Session Management Policy Association. This request includes context information such as the subscriber identity (Subscription Permanent Identifier, SUPI), Data Network Name (DNN), and Network Slice Selection Assistance Information (NSSAI). PCF then queries its local configuration or the subscriber profile stored in UDR to find a matching policy for the SUPI.

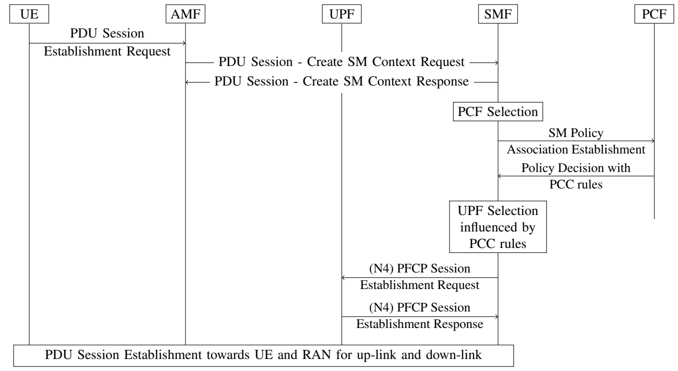



Fig. 2. A simplified PDU session establishment request with Policy Association [12], adapted from 3GPP TS 23.502 [2].

Once SMF receives the policy, it utilizes this information to make the selection of one or multiple User Plane Functions (UPFs). This step is vital for enabling edge computing since all user plane traffic passes through the UPFs. To enable edge computing, the SMF must choose a UPF that is in close proximity or co-located with an edge data center. Additionally, the user's location (e.g., cell ID or Tracking Area Code, TAC) should be considered when selecting the appropriate UPFs. Finally, at the N4 reference point, the SMF configures the selected UPF(s) accordingly and instructs the UPF how to route traffic between the UE and the Data Network.

More specifically, SMF convey crucial instructions to UPF, information regarding network connectivity, including the entities to connect to and the UE IP address. It also informs UPF on how to route traffic between the UE and the Data Network and instructs UPF on the treatment of different data flows:

- First, SMF inform UPF about the network entities it needs to connect to. This involves identifying any intermediate UPFs that could be involved in the data routing process as well as the gNB to which UPF should establish a connection. By providing this information, SMF guarantees that the appropriate network elements are aware of each other and can facilitate seamless communication.
- Second, SMF provide UPF the UE IP address. Hence, UPF can establish a direct connection with the UE, allowing for the exchange of data packets between the UE and the data network. By providing the IP address, SMF guarantees that UPF knows the specific endpoint to which it needs to route the traffic intended for the UE.
- Third, SMF instructs UPF how to route traffic between

the UE and the Data Network. SMF transfer information of the desired routing path, as well as, the QoS parameters, to allow UPFs applying appropriate policies and rules to join the intended route and handle data packets.

Currently defined in 3GPP technical specification 29.244[4], the N4 interface SMF-UPF employs the Packet Forwarding Control Protocol (PFCP) to support the Control and User Plane Separation (CUPS). PFCP is on top on UDP/IP and functions only in the control plane. SMF uses PFCP to associate with one or more UPFs and subsequently configure PDU sessions for the user plane.

However, the current architecture of UPF and the N4 interface to SMF leads to several limitations as follows:

- Hard deployment: In UDP-based PFCP, either one end-point needs to knows the IP address of the partner in order to create a connection. This is problematic in deployment. In a cloud native platform, life cycles of SMFs, UPFs are managed in a dynamical way, instances of an application can be spawn or killed by the orchestration layer. Their physical address (IP) is ephemeral, thus manual configuration is not an option.
- Poor scalability: when the numbers of UPFs and providing network slices increase, the management and coordination of various UPFs across different network slices becomes more challenging. It demands an official solutions for the automatic and efficient management of various UPFs simultaneously. Currently, in the two popular open-source platforms, such as Free5gc[7] and Open5Gs[1], every change in the topology of UPFs will make SMF to manually re-configure.
- High complexity: Multiple SMFs are to maintain UPFs topology, e.g. multiple SMFs in different network slices participate in the management of several common UPFs, hence there can be an inconsistency issue. SMFs may have different versions of UPFs topology due to configuration mistakes and it requires the SMFs to synchronize in order to have a single UPFs topology. Besides, integrating UPF management functions into SMF makes complexity to the overall system.

Addressing these limitations is crucial to ensure full cloud-native B5G architecture, enabling the deployment with cloud orchestration system like Kubernetes, in support of SMF-UPF establishment automation and orchestration.

## III. Proposed Architecture and Implementation

UPMF serve as a registry for SMFs managing UPFs in the network and computing appropriate paths/routes of UE traffic through network of UPFs. Through this deployment approach, the UPF is involved in the SBI communication of the SBA system, and SMF eliminates the step of statically configuring UPF profiles. UPMF is designed in accordance to the service based architecture (SBA) and shown in Figure 3 and 4.

UPMF contains the two main services, to store the knowledge of UPF topological information including all UPFs of working network slices (with corresponding UPFs' IP

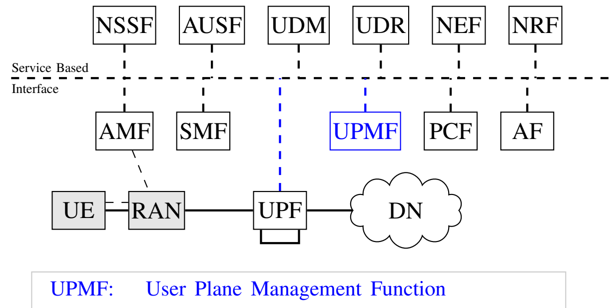



Fig. 3. User Plane Management Function (UPMF) in Service Based Architecture of B5G networks

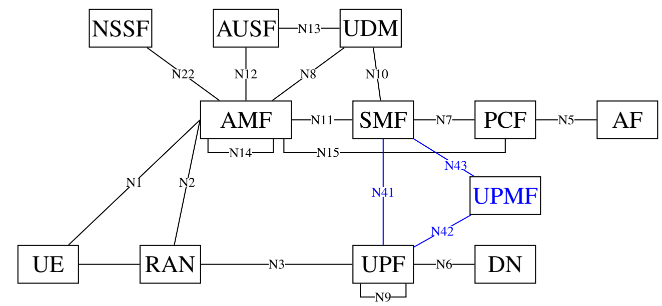



Fig. 4. Reference point architecture with the introduction of User Plane Management Function (UPMF)

addresses) and the links connecting among UPFs); and to implement the path computation function which is currently embedded in UPF.

Figure 4 shows the three reference points related to the UPMF, N41, N42, and N43, to replace the N4 SMF-UPF using PFCP with a SBI (Service Based Interface) using HTTP/HTTPS. With the involvement of a new NF, we have deployed additional interfaces between the UPMF and other NFs. Figure 5 shows the flexible SBI interface better off the manual configuration because of using a point-to-point protocol at the interface of SMF-UPF.

We will elaborate essential service functions of UPMF, including UPF management, and UPF path query.

**UPF management**: The UPMF is a network function provided to all network slices, and its address is known by all deployed UPFs. When an UPF starts, it registers its information with the UPMF. The necessary configuration information should consist of the UPF identifier, the network slice that the UPF belongs to and the interfaces (N3, N6, and N9). When there is any changes in UPFs, UPFs also updates the latest information to UPMF. With this information, the UPMF can construct a topological map of all UPFs within each working slices. Note that all UPFs and gnBs in the network are assigned unique identifiers in the deployment and such information is also submitted to and stored at NFR (Network Function Repository).

**Path Query**: UPMF receives a query message from SMF, query message will provide Network Slice Selection Assistance Information (NSSAI), Source Network Interface (at-

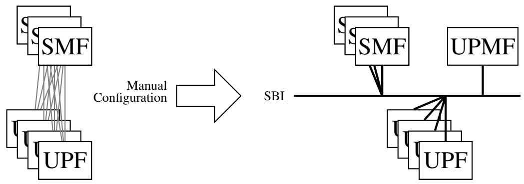



Fig. 5. Packet Forwarding Control Protocol to Service Base Interface

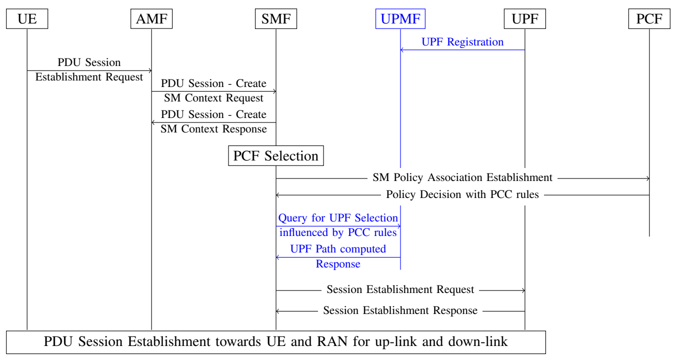



Fig. 6. The call flow for PDU session establishment involving UPMF

tached gNB) and Destination NIC (anchoring UPF). UPMF should implement the data structure for the management of UPF topological slices corresponding to each NSSAI. Every time a query message is received by UPMF, UPMF using an routing algorithm, e.g. Dijkstra algorithm[6] in the current implementation, to find all the paths that satisfy the query's requirements. A shortest path is chosen and sent in the response message back to SMF.

These functionalities address the two essential issues of system deployment scalability.

- eliminate the manual configuration of UPF topological information which is now auto-configured by UPMF (Zero-touch configuration). It allows fast, effective scaling of a large number of UPFs in a cloud-native environment (in the support of Kubernetes).

- select optimal path for each request with specific QoS requirements with the updated information of UPFs' status, traffic flows, etc... The separation of the path computation from SMF helps to easily update routing algorithms.

Figure 6, has shown the service functions described above in the new call procedure flow. At the last stage of the call procedure, SMF chooses the shortest UPFs path from the response of UPMF, it sends Session Establishment Request message to all selected UPFs. UPF performs network resource allocation and QoS enforcement for the PDU session.

## IV. PoC System

The proposed UPMF has been implemented as part of ETRI's new B5G core architecture, which is designed to address the limitations of the current 3GPP architecture [13]. In this section, we provide a brief overview of the architecture, highlighting its fundamental changes and explaining

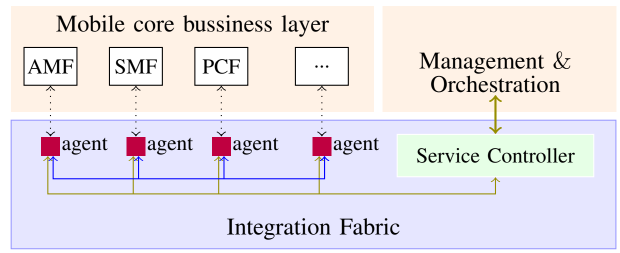



Fig. 7. ETRI's b5gc architecture

how UPMF is integrated into the system for an example deployment.

### A. *ETRI's B5gc core architecure*

The current 3GPP architecture faces challenges due to its attempt to address both UE management and network function management in a single framework. This lack of separation of concerns has led to a complex system design, making implementation cumbersome. Additionally, the current specifications offer multiple deployment models without clear guidance on model selection and compatibility.

ETRI's B5gc architecture, Figure 7, represents a departure from the current 3GPP architecture. It moves all service registration, discovery and routing logic from SBA network functions into a midleware layer so called *integration fabric*, thereby reducing complexity. The fabric consists of an agent library and a service controller. Network functions directly utilize the agent within their executables for service routing, while the controller acts as a service registry, collecting information about agent locations and runtime environment parameters. Furthermore, the routing capabilities of the agents can be customized and configured through interactions with the management and orchestration layer, facilitated by the service controller.

The integration fabric provides a unified programming API through the agent library, enabling seamless interaction with the business layer consisting of core network functions. This unification is achieved by assigning globally unique names to network functions (examples are provided in section IV-B). As a result, the core network's business logic can focus solely on the assigned names without the need to handle service discovery and routing concerns during deployment. Moreover, the fabric, being isolated from the upper layer, allows for different design and implementation approaches tailored to specific needs.

### B. *PoC deployment with UPMF*

In this section, we present a deployment of a core system serving two network slices: 010203-1 for sessions to an edge data network for MEC applications , and 111213-2 for Internet connections (here, slice identity is presented in the sst-sd format). The example is illustrated in Figure 8. For clarity, common network functions such as UDM, UDR, AUSF, and NSSF, shared by all slices, are omitted from the figure.

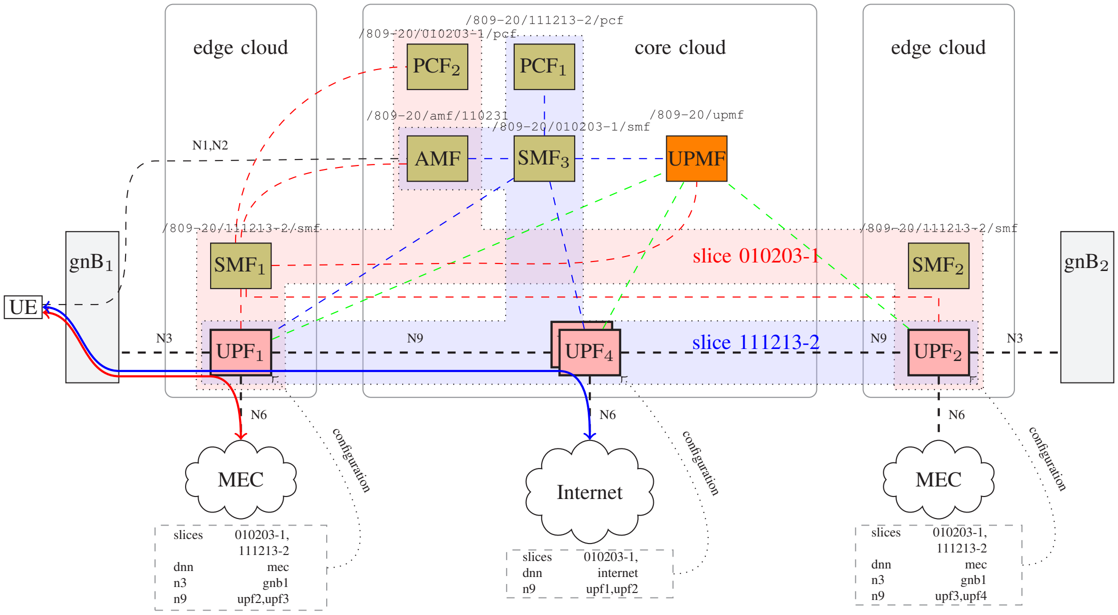



Fig. 8. A PoC deployment of UPMF in Etri's b5gc system

The network functions belonging to a specific slice are marked within regions of the same shade (light red for 010203-1 and light blue for 111213-2, respectively). Each network function on the control plane is assigned a unique name. For example, the name of an SMF in a slice can be a combination of the PLMN identity, slice identity, and the network function type (smf). Multiple instances of the same network function can be executed for scaling purposes. In the MEC slice 010203-1, there are two SMF instances, each on an edge cloud close to a gnB. The AMF is shared by both slices and its name is composed of the PLMN identity (809-20) and the AMF identity (110231), which consists of the AMF region, AMF set, and AMF pointer.

The system includes four UPFs: $UPF_1$ and $UPF_2$ installed at the edge cloud with N3 interfaces to gnB and N6 interfaces to MEC data networks, and $UPF_3$ and $UPF_4$ installed at the core cloud with N6 interfaces to the Internet and N9 interfaces to intermediate UPFs. Such topological information of UPFs are registered at the UPMF.

The figure illustrates the active communication paths among network functions during the PDU session establishment for two requests from the UE. Paths of the same slice are represented by the same color, the red session is to connect to "mec" data network for a MEC application, which belongs to slide 010203-1, and the blue session is to connect to the Internet, which belongs to slice 111213-2.

Upon receiving session creation requests, the AMF selects (through the fabric layer) the SMF that is close to the gnB to which the UE is attached, in order to create the PDU session context. $SMF_1$ is selected for the "mec" application and $SMF_3$ will handle the PDU session of the Internet connection. The both SMFs will query the UPMF for the selection of UPFs path with the directions including the network name, slice number and gnB identity of each session.

The UPMF should remove any UPF from the full topological map that is not relevant to the directions. For the "mec" session, the filtered topological map consists of two unconnected nodes: $UPF_1$ and $UPF_2$. From this map, only $UPF_1$ matches the "mec" query, as it has an N3 interface to the gnB where the UE is connected, and an N6 interface to the "mec" data network. In the meanwhile, for the Internet connection, the filtered topological map still includes all four UPFs, as they are all members of slice 111213-2. From this map, the UPMF may select the path consisting of $UPF_1$ and $UPF_4$, as $UPF_1$ has an N3 interface to $gnB_1$ and $UPF_4$ has an N6 interface to the Internet. $UPF_1$ acts as an I-UPF in the slice.

## V. Related Works

There have been several articles presenting solutions for efficient monitoring and management of UPFs. The closest is "Edge Computing in OAI 5G Core Network" slice [11], the challenge of managing and scaling UPFs dynamically and seamlessly on a Cloud-native infrastructure is addressed by integrating UPF topology management functions into SMF. However, this integration adds complexity to SMF, and one can imagine the potential issues of asynchronous UPF management when multiple SMFs are involved across different network slices. Additionally, the disadvantage of point-to-point PFCP for a cloud-native environment is not resolved.

Several other work focuses on the issue of dynamic management mechanisms for UPF in 5G environment to improve the selection process based on network conditions and service requirements, but above issue has not been taken into consideration. Network load, latency, bandwidth availability, and quality of service are taken into account during the selection process [5]; Efficient management of UPF placement in dynamic network environments, where traffic demands and network conditions fluctuate over time, through intelligent algorithms, optimization techniques, and UPF reconfiguration [8]; Optimization of Edge Service Infrastructure (ESI) and UPF placement in 5G networks, considering factors such as latency, service availability, and resource capacity [9]; Utilizing machine learning and predictive modeling techniques to analyze network conditions, user behavior, and service demands, enabling proactive session management and UPF decision-making [10].

## VI. Conclusion and Future works

In this study, we highlight the limitations of the 5G Service-Based Architecture (SBA) in managing User Plane Functions (UPFs) and discuss the challenges in effectively managing UPFs in a cloud-native environment to advance beyond 5G. To address these challenges, we propose enhancing the 5G SBA by introducing a new network function called the User Plane Management Function (UPMF). Additionally, our proposal resolves the limitations of the Packet Forwarding Control Protocol (PFCP) by the implementation of the Service-Based Interface (SBI) for the communication between SMF and UPF.

It is important to note that the UPMF has similar functions with the Network Repository Function (NRF), such as managing other network function profiles, i.e. UPF profiles. However, the UPMF also plays a crucial role in managing UPF topology information and includes a path computing component for selecting the UPF paths for PDU sessions. While the current proposed UPMF is a reasonable approach, there may be a more elegant solution worth exploring. One possible alternative is to limit the UPMF's role to path computing and delegate profile management to the NRF, making it the sole point of contact for all network functions in a deployment. Under this approach, UPFs would still register their profiles with the NRF like other network functions, while the UPMF could subscribe to the NRF for notifications of UPF events, such as registration and updates, to maintain the UPF topological information.

Our proposed solution has been implemented and integrated into a new mobile core architecture developed by ETRI [13]. By replacing the PFCP protocol with our SBI-based solution, the configuration and management of UPFs are greatly simplified. However, this enhancement can be extended further to enhance the intelligence of the mobile core control plane. Leveraging the SBI-based architecture enables discoverable UPFs to share their internal status information with other network functions, paving the way for the development of AI models to optimize path selection. We consider this as part of our future work and an area for further exploration and refinement.

## VII. Acknowledgement

This work was partially supported by the ICT R&D program of MSICT/IITP. [2021-0-02094, International collaborative research on 6G network architecture and core technologies]

## References

[1] Open5gs: An open-source 5g core network. https://github.com/open5gs/open5gs. Accessed on 2023-4-25.

[2] 3GPP. 5g; procedures for the 5g system (5gs). Technical Report TS 23.502, 3rd Generation Partnership Project (3GPP), 2020.

[3] 3GPP. 5g; system architecture for the 5g system (5gs). Technical Report TS 23.501, 3rd Generation Partnership Project (3GPP), 2020.

[4] 3GPP. Lte; 5g; interface between the control plane and the user plane nodes. Technical Report TS 29.244, 3rd Generation Partnership Project (3GPP), 2020.

[5] Viktoria-Maria Alevizaki, Markos Anastasopoulos, Anna Tzanakaki, Dimitra Simeonidou, and UK Bristiol. Dynamic selection of user plane function in 5g environments. *Proc. of Optical Network Design and Modelling*, 2021.

[6] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[7] NYCU COMMUNICATION SERVICE/SOFTWARE LABORATORY (CS LAB). Free5gc: An open-source 5g core network. https://github.com/free5gc/free5gc. Free and open-source, Accessed on 2023-4-25.

[8] Irian Leyva-Pupo, Cristina Cervelló-Pastor, Christos Anagnostopoulos, and Dimitrios P Pezaros. Dynamic scheduling and optimal reconfiguration of upf placement in 5g networks. In *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 103–111, 2020.

[9] Irian Leyva-Pupo, Alejandro Santoyo-González, and Cristina Cervelló-Pastor. A framework for the joint placement of edge service infrastructure and user plane functions for 5g. *Sensors*, 19(18):3975, 2019.

[10] Sebastian Peters and Manzoor Ahmed Khan. Anticipatory session management and user plane function placement for ai-driven beyond 5g networks. *Procedia Computer Science*, 160:214–223, 2019.

[11] Stefan Spettel. Edge computing in oai 5g core network, 07 2022.

[12] Stefan Spettel. *Robotics and Edge Computing in 5G: A Prototype for the OpenAirInterface 5G System.* PhD thesis, Wien, 2023. Image adapted from Figure 4.9.

[13] Quang Tung Thai and Namseok Ko. Towards realizing a cloud-native b5g mobile core architecture. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1018–1023, 2022.