Novel Core Network Architecture for 5G Based on Mobile Service Chaining

Dinand Roeland(✉) and Zhang Fu

Ericsson Research, Stockholm, Sweden
{dinand.roeland,zhang.fu}@ericsson.com

Abstract. A key requirement for the next generation mobile networks is flexibility to support multiple use cases with different network requirements. In this article, we focus on this challenge and propose a novel mobile core network architecture for 5G. We advocate an approach based on Software Defined Networking (SDN) that enables a solution without the need for centralized user plane nodes and with a strict division between control plane and user plane. This allows for flexibility to rapidly deploy network service functions in different deployment setups, supporting multiple use cases. The proposed architecture supports traffic aggregation on multiple granularities, not only per-device tunnels. This allows for efficiency and scalability required to support the massive amount of devices in the 5G time frame.

Keywords: Core network · Architecture · Mobility · Service chain · SDN · 5G · De-composition · CP/UP split · Anchorless

1 Introduction

In the last years a number of activities have defined requirements on the next generation mobile network. One effort is the 5G White Paper by the Next Generation Mobile Networks (NGMN) Alliance [1], which lists a very diverse set of use cases, including IoT (Internet of Things), vehicle-to-vehicle communications, controlling industrial robots, high quality media delivery, etc. These use cases define the requirements for the next generation of mobile networks, flexibility being one of the key requirements. For each use case user plane packets should traverse a different sequence of network service functions. A 5G core network architecture should offer an infrastructure to support flexibility of organizing such service chains. The 3rd Generation Partnership Project (3GPP) is currently running a study for such new 5G core network architecture [2, 3].

The current Evolved Packet Core (EPC) [4] architecture is optimized for the mobile broadband use case where traffic for an end user passes a Packet Data Network Gateway (P-GW) acting as mobility anchor point. It is limited in its flexibility to support new use cases due to the time consuming standardization process. E.g., adding network functionality to support an interface between P-GW and Wireless Local Area Network (WLAN) access [5] took more than a year of standardization, even though all new functionality was fully contained within an operator's network and had no impact on the user device. Furthermore, it is envisioned that the 5G core network architecture

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2017
R. Agüero et al. (Eds.): MONAMI 2016, LNICST 191, pp. 44–57, 2017.
DOI: 10.1007/978-3-319-52712-3_4

will have to handle many more devices, which may make EPC inefficient since it maintains at least one tunnel per device.

The need for a new mobile core network architecture has been identified in several articles in literature with a variety of solutions proposed [6–9]. Briefly, most of the proposals leverage the Software-Defined Networking (SDN) paradigm to the architecture of the core network, in order to achieve the aforementioned flexibility by separating the user plane and the control plane. The flexibility can be further enhanced with Network Function Virtualization (NFV).

However, adopting SDN into the design is just the very beginning. Work is already ongoing to further separate control plane from user plane in EPC [10]. However, incorporating SDN in the current EPC architecture this way still uses per-device General Packet Radio Service Tunneling Protocol (GTP) tunnels via a P-GW anchor. In a 5G time frame, we will likely see an increase in use cases like, e.g., device-to-device communication or edge computing [1], where routing via a central P-GW is not preferred. Thus, we should allow an end-to-end SDN-based service deployment from base station to peering point. This goes beyond today's typical service chain deployments with SDN introduced above the P-GW anchor point [8, 11].

In this article we propose a novel mobile core network architecture for the 5G time frame. The architecture is based on mobile service chaining which is the ability of the network to combine service chaining with mobility handling. We advocate an all-SDN approach with a strict division between control and user plane. We abandon the limitation of using only a single mobility anchor point per user connection, allowing for a flexible deployment strategy. The architecture gives a flexibility to support multiple use cases, and rapid deployment of network service functions for various use cases. Regarding the packet forwarding within a service chain, we propose to use tags, allowing to aggregate traffic on different granularities than the GTP per-device tunnels. Furthermore, the approach does not require reconfiguration of user plane switches as a result of mobility events. This allows for an efficiency and scalability required to support the massive amount of devices in the 5G time frame.

The rest of this article is organized as follows. First we give a brief overview of today's EPC architecture in Sect. 2. In Sect. 3, we outline the components of the novel architecture and how service chaining is supported. Section 4 describes how to implement a number of use cases in the proposed architecture, followed by a comparison between the novel architecture and EPC in Sect. 5. We give a brief overview of our prototype in Sect. 6. After that we discuss a number of scalability aspects of the novel architecture in Sect. 7. We briefly review related work in Sect. 8. Finally, we draw our conclusions and point out some further investigations in Sect. 9.

2 EPC Architecture

As background, we provide a brief overview of the EPC architecture [4] in Fig. 1.

A mobile device, called UE for User Equipment, connects to a Base Station (BS). In EPC terminology the BS is called evolved Node B (eNB) and resides in the

E-UTRAN (Evolved Universal Terrestrial Radio Access Network). The BS is connected to the Mobility Management Entity (MME) which controls the network upon mobility of the device. In particular, the MME controls the setup and maintenance of GTP user plane tunnels between BS and P-GW via a Serving Gateway (S-GW). The P-GW acts as a global mobility anchor point and includes functions like Quality-of-Service (QoS) handling and charging support. The S-GW acts as a local mobility anchor point and includes functions like idle-mode buffering. Policy and charging is governed from the Policy and Charging Rules Function (PCRF). The Home Subscriber Server (HSS) is the prime database for subscription-related information.

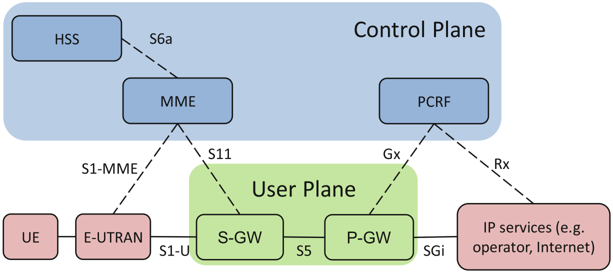

Fig. 1. Current EPC architecture

There is a standardized mechanism for flexible service function chaining above the P-GW on the SGi interface. Traffic steering control towards those service functions is performed from the PCRF [11]. The functions themselves are not standardized, but may include, e.g., a parental control function, a Transmission Control Protocol (TCP) proxy, etc. Note that in this solution the rest of the EPC architecture remains unchanged. In other words, the flexible chaining is not end-to-end down to the BS.

3 Proposed Novel Architecture

Figure 2 shows the proposed novel architecture. Note that this is a functional architecture; the relation to a product implementation is not shown. The functional architecture may run on a platform that may be distributed over multiple sites in the operator's network, like a distributed cloud. In an implementation different components could be virtualized and may be combined.

As EPC, the architecture is divided into a control plane and a user plane. A device communicates with the control and user plane via one or more Access Networks (AN). The concepts described in this article are equally applicable to all accesses. The AN could, e.g., be a novel 5G radio, E-UTRAN or even a fixed access. In this article we assume re-using the S1 interface from EPC.

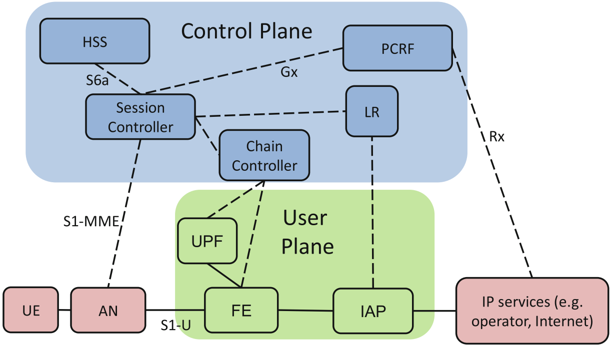

Fig. 2. Proposed generic functional architecture

3.1 Control Plane

The control plane (CP) contains all control plane logic, allowing for a strict separation between control and user plane. The Session Controller includes an access management function for each access. For an LTE access, this would include the access-specific functions of the MME. The Session Controller further deals with user session management and the creation of the service chain for the user. The Chain Controller deals with the setup and maintenance of the service chain in the user plane. Besides this, the CP contains the PCRF and HSS. The LR is explained in the next section. This article does not go further into the specifics of the CP. Instead it focuses on the user plane and the signaling between user plane and control plane.

3.2 User Plane

The user plane contains three types of function nodes: Forwarding Element (FE), User Plane Function (UPF) and Internet Protocol (IP) Advertisement Point (IAP).

An FE forwards each packet to one of its ports based on rules it has received from the CP. An FE may forward a packet through one or more UPFs. An FE is only concerned with the actual forwarding; it does not classify or modify a packet.

A UPF is a service function that processes user plane packets. Processing may include altering the packet's payload and/or header. UPFs are not expected to know topological information regarding the chain, including which other UPFs are in the chain and how to reach them. A UPF may serve multiple users, and may or may not keep user-specific data. We call such data for context in this article.

The IAP is a key component to achieve an anchorless network; i.e. a network without a mobility anchor point. Just like a plain IP router, an IAP advertises a range of IP addresses/prefixes towards an outer IP network. This may be Internet or an operator-internal network. A single IP address/prefix may be advertised by multiple IAPs. If the IP address of a specific device is advertised by multiple IAPs, then packets for that device can enter the network via any of those IAPs. Similarly, an anchored approach can be achieved by allowing only a single IAP to advertise the IP address for that device. Each IAP is thus pre-configured with one or more address ranges. It is the CP that assigns an IP address/prefix to the device upon attachment.

The CP contains a Location Registry (LR). This is a table of entries, where each entry is a mapping from device IP address/prefix to current device location. The location is the address of the first UPF in the chain, plus optionally additional location information, e.g., a BS Identifier (ID). When a device moves from one BS to another, the CP ensures that the LR is updated with the new location. An IAP is only used for downlink packets heading towards the device. For each downlink packet, the IAP performs the following steps: (1) Query the LR based on the destination IP address of the packet in order to retrieve the location; (2) Tag the packet with the location; (3) Forward the packet via an FE to the first UPF in the service chain as indicated in the LR reply. The concept of tagging is further explained in the next section. Note that the LR can be implemented in an optimized fashion. E.g., the IAP query may be performed towards an IAP-internal cache. Only if no entry is found in that cache, the LR is queried. For non-mobile devices, implementing the query is simplified as the entry in the LR for that device will not change.

3.3 Service Chaining

FEs forward packets to different UPFs and BSs according to which service chain the packets need to traverse and where the corresponding devices are located. Such information is added to the packet as tags by the classifiers.

A classifier (CL) is a UPF that determines which service chain a packet takes based on the packet header and rules it has received from the CP. A CL may change the packet's header, e.g., adding a tag to indicate which service chain the packet traverses. A CL may contact the CP when a packet cannot be classified, or it may drop such packet. The classifier can be configured by the CP with rules at several occasions:

- Before a device attaches; for generic rules that apply to multiple devices.
- When a device attaches; for rules that apply to the specific user that attaches.
- After the device has attached. These updates might originate from user-specific real-time events that are reported to the CP from, e.g., a UPF performing Deep Packet Inspection (DPI), or an external application that requests the CP for a specific QoS treatment.

We assume that there is at least one uplink and one downlink CL in the network which classify the traffic from the devices and to the devices, respectively. Classifiers could be placed early in the chain; e.g., uplink CL co-located with BS and downlink CL co-located with IAP, or could be placed at every branch point.

FEs forward packets according to tags in the packets. Tags are logically expressed with a name/value pair. A packet may have one or more tags. There are multiple ways to carry tags in packets. E.g., Q-in-Q [12] where tags are encoded as Virtual Local Area Networks (VLANs), or a tunneling protocol where a variable number of tags can be carried as metadata [13], or even an evolution of GTP. In certain cases, an existing protocol element can act as tag value; e.g., an IP address can act as device ID. Regarding the implementation, the FEs may be implemented as OpenFlow switches, given that OpenFlow [14] supports multiple tags and also multiple flow tables. We have followed an approach were the FEs and UPFs are be implemented as virtualized network entities running on general-purpose hardware; see Sect. 6.

A UPF handles a collection of flows. The definition of flows is kept flexible and can be configured by the CP depending on the use case. Examples of flows include: packets with the same IP 5-tuple, all packets to/from a specific BS.

Putting it all together, an uplink packet would traverse the BS and one or more FEs. Each FE may forward the packet via one or more UPFs. Similarly, a downlink packet would traverse the IAP, one or more FEs and a BS. In both uplink and downlink, at least one UPF acts as CL.

The next sections describe how to implement a number of use cases in the proposed architecture.

4 Implementing Use Cases

The most important use case for many mobile network operators today is the mobile broadband offering. Given that a large portion of the mobile broadband traffic can typically be cached [15], let us assume that the operator has deployed a number of Content Delivery Networks (CDNs) to reduce peering cost. In a typical EPC deployment, such CDNs would reside in a central site together with P-GWs and other EPC components. With our novel architecture, EPC's P-GW and S-GW would be de-composed into multiple smaller UPFs. In this specific use case, there may be one chain of functions for traffic towards the CDNs and one chain for traffic towards Internet. Both chains may have segments in common. The functions may be deployed across the network topology. Figure 3 shows an example of a simplified deployment. The Internet chain consists of F1–F2. F1 could, e.g., be a bandwidth limiter, and F2 could, e.g., be a DPI, or a complex charging function. The CDN chain consists of F1–F3, where F3 may perform simple charging. The peer in the CDN chain is the actual CDN and is deployed in the IP services network of the central site. Note that the chains in the figure are simplified with regards to uplink and downlink symmetry. The chaining concept itself allows certain UPFs only to be traversed in one direction; e.g., the uplink classifier only in the uplink.

In some use cases it may be beneficial to perform processing in a local site or base station site instead of in a central site which is far away from the device. E.g., the CDN from the use case above may be placed in a local site in order to save bandwidth between local and central site. Or, a base station site may host a specific application that requires very low latency; e.g., an industry application where the device is a factory robot. In Fig. 3 such use case is shown as a third chain F1–F4. The peer in the IP services network of the local site acts as industry application.

Note that the use cases above are just examples. They can also easily be combined; e.g., a device may access Internet but at the same time access the peer in the local site. The use cases above are used in the following sections as a guiding example to explain data exchange and mobility handling.

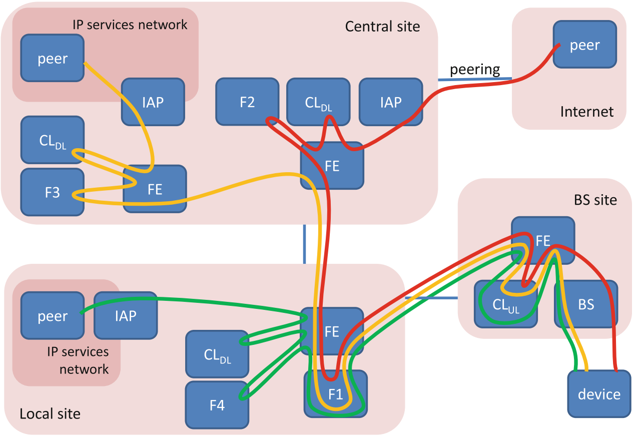

Fig. 3. Example of a deployment and three chains: (1) device-BS-CL$_{UL}$-F1-F2-CL$_{DL}$-IAP-peer, shown in red; (2) device-BS-CL$_{UL}$-F1-F3-CL$_{DL}$-IAP-peer, shown in yellow; (3) device-BS-CL$_{UL}$-F1-F4-CL$_{DL}$-IAP-peer, shown in green. (Color figure online)

4.1 Data Exchange

Figure 4 illustrates a packet exchange between a mobile device connected to a BS and the CDN in the central site. This is the yellow CDN chain from Fig. 3. FEs are not shown in this call flow. The text below each arrow lists a subset of the packet's header fields. The uplink classifier CL$_{UL}$ classifies the packet (step 2) based on rules it has received from the CP. In this example, the CL$_{UL}$ has a rule "if destination IP address is x then set Tag$_C$ = y", where x is the address of the CDN and Tag$_C$ indicates that this is a CDN chain. There may be multiple types of CDN chains, and the value y indicates the type. Eventually the packet reaches the CDN (step 4). Note that the CDN is on its own IP network and may be a third party product not aware of service chaining. In this example, the IAP on that IP network may announce the complete IP address range used for the operator's mobile devices. The IAP performs the lookup to the LR and tags the packet with the BS ID (steps 6–7). The IAP then forwards, via an FE, to the first UPF in the chain based on information received from the LR (steps 7–8). The first UPF is here the downlink classifier CL$_{DL}$, which marks the packet with Tag$_C$ = y to denote that this is a specific type of a CDN chain (step 9). In the downlink towards the device, the FEs use Tag$_C$ to forward the packets through the right UPFs and Tag$_{BS}$ to forward the packet to the current location of the device (steps 10–11).

Note that the call flow assumes that the CLs and FEs have been provisioned with rules before the packet exchange starts. These rules may have been provisioned when the device attached to the network. At that point in time, certain UPFs may also need to be provisioned with context for the particular device. E.g., the bandwidth limiter F1 may be informed about the maximum bandwidth for this specific subscriber. Many rules may also be common to a group of devices. In this example, forwarding rules for Tag$_{BS}$ and Tag$_C$ can all be provisioned to the FEs once for many devices.

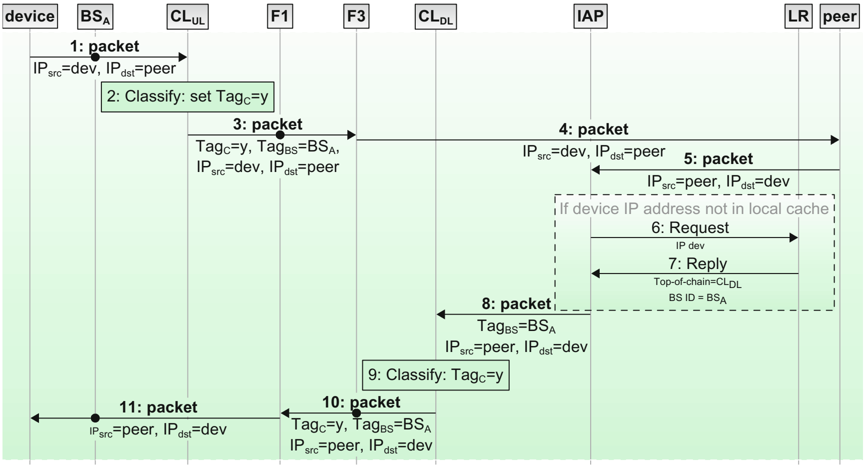

Fig. 4. Exchange of a packet between device and peer over the CDN chain

For the Internet chain and the local processing chain, the call flow is the same except that a different tag will be set at classification. This causes a different set of UPFs to be traversed. Note that the IAPs for these chains may all announce the IP address of all the operator's mobile devices. The device uses only a single IP address and is not aware of the chaining infrastructure and where its packets gets routed in the operator's network.

4.2 Dealing with Mobility

Handling mobility is remarkably simple in the proposed architecture. Figure 5 gives a call flow for a handover from BS_A to BS_B. BS_A is connected to a first local site and BS_B to a second local site. In this example we assume that both local sites are connected to the same central site.

After the actual handover (step 4) the target BS sends a "path switch request" to the CP (step 5) as in EPC's eNB handover procedure [4]. After this BS handover, parts of the chain may also need to be moved. In this particular example, CL_{UL} and F1 need to move from the first local site A to the second local site B. The CP provisions the target UPF instances $CL_{UL,B}$ and $F1_B$ (steps 6 and 8). This step may involve copying device-specific context from source to target UPF instance (steps 7 and 9), similar to copying, e.g., keying context in EPC's eNB handover procedure. The CP stores the new location of the device in the LR (step 10). It informs any IAP that recently has queried the location of the device (step 11), such that the IAP-internal cache has the up-to-date information. After this the handover can be acknowledged (step 12). Finally, resources that are no longer needed are released (steps 13–15). After this user packets can be exchanged again (steps 17–18). Such packet exchange is as in Fig. 4, with the difference that a new value for Tag_{BS} will be set.

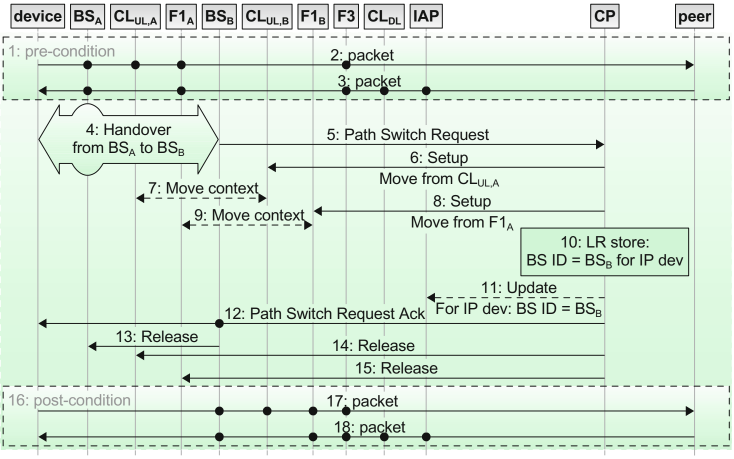

Fig. 5. Handover between base stations

Note that it is possible to change forwarding rules in the FEs as part of the procedure. However, in the example this is not required as all FEs can be pre-provisioned with rules for Tag_{BS} and the chain tags. Also note that the IAP in the target site will simply perform the query to the LR to retrieve the current location of the device. There is no change of IP address needed due to the handover. This also means that the UE is not impacted.

5 Comparison to EPC

The proposed architecture imposes a number of advantages compared to EPC. These advantages are in particular important to fulfill the expectations on a 5G core network.

The architecture allows for a de-composition of functions and a flexible composition of these functions. As shown in the example, not all UPFs need to be involved in all service chains. This is difficult to achieve in EPC's P-GW and S-GW where all functions are specified by standards.

The strict separation of control plane and user plane in the proposed architecture allows them to scale independently. This is only partly possible in EPC where the P-GW consists mainly of user plane functionality but also contains control plane functionality, limiting independent scaling of those planes. E.g., the P-GW deals with GTP tunnel setup handling based on commands received from the PCRF over the Gx interface.

The above mentioned advantages also imply that it is easier to distribute the user plane to, e.g., local sites. If needed, the control plane can remain in a central site. In EPC it would be possible to place the P-GW in a local site. However, that causes inefficient routing after a handover, since the P-GW anchor function requires the traffic to be routed back to the source site. In our proposed architecture routing can be kept optimal due to the anchorless concept. An example of this was provided in Fig. 5 where the packets after handover go via the target local site without passing the source local site.

The de-composition and strict separation of user plane also allows us to place different UPFs on different processing platforms. In the example of Fig. 3, the UPFs in the local site are fairly simple functions and could together with the FE be implemented on a high-performance packet processing environment. Service functions such as DPI and parental control that require complicated processing may require a general-purpose processing environment like a complete virtual machine. The next section gives a brief overview of how we have implemented FEs and UPFs in a prototype.

6 Prototyping

A preliminary prototype has been implemented by our colleagues in order to investigate the performance [16]. The user plane is implemented in servers with moderate configuration (3.5 GHz 6-core Intel Xeon E5-1650 CPUs and 32 GB RAM). In the prototype, the FEs and the UPFs are realized as Click elements in a Click modular router [17]. There are three types of UPFs in the prototype: counter, traffic-rate limiter and header compressor. All of the UPFs require packets modifications, e.g. adding counter values in the payload, adding markers in the packet headers and compressing packets. Three types of UPFs were added incrementally into the chain, and the resulted throughput varied from around 5 Mpps to around 2 Mpps. Note that the UPFs are also elastic, meaning that there could be multiple instances of the same type of UPF, thus user plane packets can be processed in parallel resulting higher aggregated throughput.

7 Scalability Discussion

In the 5G time frame, data rates, data volumes and number of devices are expected to increase drastically [1]. This places high demands on the scalability of a 5G core network.

In the proposed architecture, each IAP may announce the complete IP address range of the operator's devices. In the uplink, once the last UPF of the chain has been traversed, the packet can leave the mobile core network at any place. This offers the opportunity for traffic to use an optimal route.

As explained in Fig. 4, when an IAP receives packets for a specific device it performs a query to the LR and stores an entry for the current location of the device in its local cache. We define active time of a device as the time the device keeps sending or receiving packets. The IAP may remove the entry when it does not receive any packets for that device for a predefined time period. So, after the active time plus this timeout, the entry in the local cache is removed. Assume we model the devices becoming active as a Poisson process with arriving rate λ. Further assume that the active time of the devices follows an exponential distribution with the mean τ. We can then compute the probability that there are n entries in the IAP using the formula for $M/M/\infty$ queue [19], that is $P_n = \rho^n/n! \cdot e^{-\rho}$, where $\rho = \lambda(\tau + \text{timeout})$. Figure 6 shows

P_n with different λ values. We can see that at most of the time the number of entries in the IAP is around λ(τ + timeout) which is typically much less than the number of IP addresses that IAP announces.

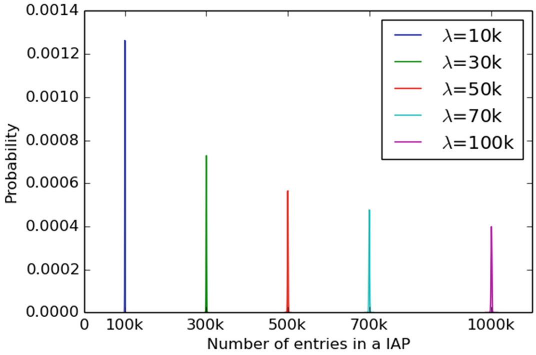

Fig. 6. We choose λ as [10k..100k], so on average [10k..100k] devices become active every second; τ as 5, so on average each device is active 5 s; and timeout as 5 s.

Suppose the users are active for a long time; e.g., they are watching a live video, which may cause the number of entries in the IAP to accumulate to a high number. In such case, we believe that the performance of the IAP will not be a problem, since much work has already been done on high performance switches, e.g., CUCKOOSWITCH [18], which combines high performance lookup table with highly-optimized I/O engine (e.g., Intel DPDK). The IAPs can be implemented as such switches and can achieve tens of millions pps throughput, even if there are a billion of entries.

For scalability reasons, it is of course possible to duplicate the number of IAPs where each pair announces half of the address range thus handling less traffic then before. Furthermore, network-wide mobility would likely only be needed for very few devices; e.g., a device that acts as hotspot in long-distance trains or a device in an entertainment system of a car. The majority of devices will be less mobile. Their addresses only need to be announced by a subset of the IAPs, accepting a less optimal routing if they move. Or, alternatively, accepting an IP address change when they move, e.g., to a new central site.

Regarding the signals exchanged between the IAPs and the LR, we argue that it is mainly affected by three factors: the active time of each device, the handover frequency of the devices, and the timeout for location entries in the IAP's cache. In the active time, the IAP will perform an LR query when the first downlink packet for the device arrives. Upon every handover, the IAP will receive an update as long as the location entry is active. Once the entry times out, no more updates will be sent.

The proposed architecture does not mandate any specific tagging scheme. Instead, we give the control plane the freedom of deciding the scheme, e.g., how many tags are used, what their meaning is, and how to encode them in the packet header. By using the proper tagging scheme, we can reduce the signals from the LR to the IAPs. As an example, assume a deployment as in Fig. 3 with a strict hierarchical setup; many BS sites to one local site, many local sites to one central site, and vice versa. In such setup, the IAP of the Internet chain in the central site would be updated upon every mobility event happening for one of the devices underneath that site (step 11 in Fig. 5). In order to reduce that update rate, it is fully possible to use a different tagging scheme. E.g., the LR may store not only a BS ID, but also a local site ID. The IAP in the central site would only query the local site ID from the LR. Consequently, it would only get updates (step 11 in Fig. 5) for devices moving between local sites. Forwarding to the local site would be based on a local site tag. Once in the local site, a second LR query would be performed to find the BS ID.

The tagging scheme can also be used for load balancing purposes. Take, e.g., Fig. 3 where F2 is a complex UPF like a DPI engine. To accommodate scaling, F2 may come in multiple instances at that site, where the FEs select an instance based on, e.g., Tag_{BS} or the device IP address. Similarly, the FEs can be scaled this way. Taking the example of Fig. 3 again, there may be multiple FEs in the central site connecting to local sites. Each such FE would only handle a subset of the local sites.

An important scalability aspect is the aggregation of devices. In EPC, every device has at least one GTP tunnel. This tunnel needs to be setup at attachment time and changes at every mobility event. In our proposed concept this is possible but not required. As shown in the example of Fig. 4, FEs can be pre-provisioned with aggregate rules for multiple devices. These rules do not need to change upon mobility events.

8 Related Work

Solutions for mobile service chaining were proposed in a number of articles.

MobileFlow [6] proposes an architecture based on SDN technology. However, the solution is still based on per-device GTP tunnels and re-use of EPC reference points. The article opens up for novel mobility handling that does not rely on GTP tunneling, but no comprehensive solution is described.

SoftCell [7] focuses on a scalable architecture that does not require centralized nodes like P-GW. It proposes an access switch in each BS that performs packet classification. The switch acts as a Network Address Translator (NAT) for uplink traffic, where the new source IP address and port denote not only the location of the switch but also the required network service for that packet. Such approach may induce a number of problems. For example, encoding the required service in IP address and port fields only works for flows that originate from the mobile device. Because the peer is not aware of any encoding scheme, the approach does not work for flows originating from the peer. This implies that no mobile device can act as server. Furthermore, after a mobile device has moved to a new BS, existing flows are still routed via the switch associated with the old BS, which introduces sub-optimal routing. This is in particular a disadvantage for long-lasting flows.

In [9] an all-SDN network architecture is proposed. The article focuses on the control plane aspects and proposes a hierarchical set of controllers instead of a single one. An underlying assumption is made that mobility handling always leads to a reconfiguration of the user plane switches, which is not necessary in our architecture.

OpenBox [20] proposes highly de-composed UPFs and a means to re-compose them in an efficient way. Besides this, there are other efforts on implementing a service chain in an efficient way; for example, FD.io [21]. However, neither [20] nor [21] address mobility.

9 Conclusions and Future Work

In this article, we presented a novel core network architecture based on mobile service chaining. The architecture provides flexibility to support multiple use cases. The flexibility is end-to-end, from BS to peering point. De-composing into user plane functions allows for a more flexible composing depending on the specific use case. The architecture provides a strict separation between user plane and control plane. These merits allow for easier introduction of new functions supporting faster time-to-market. It also allows for greater flexibility in the choice of execution platform and the deployment of the network; e.g., a distributed deployment. The architecture is efficient as no central nodes like P-GW are required. Instead, an anchorless approach is used allowing for shortest path routing. Furthermore, tags are used for maximum traffic aggregation given the assumption that many devices follow the same service chain. Service chains on a per-device granularity, or even per-device-and-flow granularity, can still be supported. Furthermore, the architecture is scalable as forwarding tables in FEs and classifier rules in CLs can be pre-configured for most of the traffic. There is no need for re-configuration due to mobility events. Tags can be organized in any hierarchy, e.g., depending on the size and topology of the network. This allows limiting the table sizes and the table update rates. Lastly, the architecture works with both anchorless and anchored setups.

Regarding future work, we are currently extending our data plane prototype [16] with control plane aspects. The flexible architecture we propose also requires a large flexibility in the setup and management of the network. We are currently integrating our prototype with orchestration and life-time management handling.

References

1. NGMN 5G White Paper. https://www.ngmn.org
2. 3GPP TR 22.891: Feasibility Study on New Services and Markets Technology Enablers
3. 3GPP TR 23.799: Study on Architecture for Next Generation System
4. 3GPP TS 23.401: General Packet Radio Service (GPRS) Enhancement for Evolved Universal Terrestrial Radio Access Network (E-UTRAN)
5. Roeland, D., Rommer, S.: Advanced WLAN integration with the 3GPP evolved packet core. IEEE Commun. Mag. **52**(12), 22–27 (2014)
6. Pentikousis, K., Wang, Y., Hu, W.: Mobileflow: toward software-defined mobile networks. IEEE Commun. Mag. **51**(7), 44–53 (2013)
7. Jin, X., Li, L.E., Vanbever, L., Rexford, J.: Softcell: scalable and flexible cellular core network architecture. In: Proceedings of CoNEXT (2013)

8. Zhang, Y., et al.: StEERING: a software-defined networking for inline service chaining. In: ICNP (2013)
9. Kozat, U.C., Sunay, M.O.: A new control plane for 5G network architecture with a case study on unified handoff, mobility, and routing management. IEEE Commun. Mag. **52**(11), 76–85 (2014)
10. 3GPP TS 23.714: Study on Control and User Plane Separation of EPC nodes
11. 3GPP TS 23.203: Policy and charging control architecture
12. IEEE Std 802.1Q-2005: IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks
13. IETF: Network Working Group, Internet-Draft, draft-ietf-sfc-nsh
14. OpenFlow Switch Specification. https://www.opennetworking.org
15. Ramanan, B., et al.: Cacheability analysis of HTTP traffic in an operational LTE network. In: Wireless Telecommunications Symposium (WTS), 2013. IEEE (2013)
16. Pinczel, B., et al.: Towards high performance packet processing for 5G. In: IEEE Conference on Network Function Virtualization and Software Defined Networks, pp. 67–73 (2015)
17. Kohler, E., Morris, R., Chen, B., et al.: The click modular router. ACM Trans. Comput. Syst. **18**(3), 253–297 (2000)
18. Zhou, D., Fan, B., Lim, H., et al.: Scalable, high performance ethernet forwarding with CUCKOOSWITCH. In: Proceedings of CoNEXT (2013)
19. Adan, I., Resing, J.: Queuing systems (2002)
20. Bremler-Barr, A., Harchol, Y., Hay, D.: OpenBox: a software-defined framework for developing, deploying, and managing network functions. In: SIGCOMM, pp. 511–524 (2016)
21. FD.io. https://fd.io/technology