Link-Level Access Cloud Architecture Design Based on SDN for 5G Networks[1]

P. Ameigeiras*, J.J. Ramos-Munoz*, L. Schumacher**, J. Prados-Garzon*, J. Navarro-Ortiz*, J.M. Lopez-Soler*

**CITIC, University of Granada*
***Department of Computer Science, University of Namur*

Abstract— The exponential growth of data traffic and connected devices, and the reduction of latency and costs are considered major challenges for future mobile communication networks. The satisfaction of these challenges motivates revisiting the architecture of these networks. We promote an SDN-based design of a hierarchical architecture for the 5G packet core. In this paper we focus on the design of its *Access Cloud* with the goal of providing low latency and scalable Ethernet-like support to terminals and MTC devices including mobility management. We examine and address its challenges in terms of network scalability and support for link-level mobility. We propose a link level architecture that forwards frames from and to edge network elements (access points (AP) and routers) with a label which identifies the APs through which the terminal is reachable. An SDN local controller tracks and updates the users' location information at the edge network elements. Additionally, we propose to delegate in SDN local controllers the handling of non-scalable operations, such as broadcast and multicast messages, and network management procedures.

Keywords – 5G, Packet Core, SDN, Openflow, Mobility.

1. Introduction

The data traffic forecast [1] of x1000 increase in wireless networks by 2020 due to the success of smartphones and Machine-Type Communication (MTC) is considered a major challenge for future networks. Analysis of this traffic increase reveals new types of applications with extremely challenging requirements, such as vehicular

[1]Sections of this paper were submitted to CloudNet 2014

communications or medical applications. The challenges that 5G systems must address can be summarized as [2][3]:

- System capacity and data rate: high capacity, up to 1000 times mostly supporting streaming applications.
- Massive number of connections: the amount of MTC devices simultaneously connected to the network will increase exponentially.
- End-to-end latency: the network must support fast mobility, enabling vehicular communications, low ”zero-perceived” latency and high reliability for applications requiring real time feedback or critical applications, including public safety.
- Cost reduction: reducing costs associated to the deployment, maintenance and operation of the network infrastructure is principal. Network and mobile devices should reduce energy consumption.
- Quality of Experience (QoE). The users' perceived quality can be improved with techniques to optimize the traffic regarding the type of application.

Simultaneously, the Information and Communication Technologies (ICT) industry is witnessing a radical paradigm shift with the commoditization of hardware resources and the adoption of cloud computing. In the computer networking field, Software Defined Networking (SDN) and Network Function Virtualization (NFV) are achieving significant success, and they are expected to play a relevant role in 5G networks.

Leveraging on these technologies, we proposed in [4] a cloudified Packet Core architecture aiming at satisfying the demands of 5G. The architecture presents three hierarchical levels, the *Access Cloud*, the *Regional Distributed Cloud* and the *National Centralized Cloud*. It aims at a radical latency reduction by moving latency-critical functionalities and services to the network edge, while achieving a major cost reduction by means of SDN, NFV and commoditized hardware. Additionally, it is especially designed to natively support small cell deployments, and it avoids the inefficiencies of the Evolved Packet Core (EPC) of 4G systems, such as non-scalable centralized devices (e.g. the *Packet Data Network Gateway*) or the GPRS Tunneling Protocol (GTP) protocol.

In this paper we focus on the design of the *Access Cloud* (AC) architecture. Our goal is that the AC provides low latency and scalable Ethernet-like support to terminals and MTC devices including mobility management. We concentrate on two main link-level

challenges to achieve this goal: mobility support and network scalability. To solve them, we propose an AC architecture and procedures to support terminal mobility with attainable switch table sizes. Additionally, we design link-level procedures to reduce network-wide flooding. The proposal exploits the SDN paradigm.

The paper is structured as follows: section 2 summarizes relevant literature. Section 3 provides an overview of the Packet Core architecture proposed in [4]. Section 4 addresses the architecture and procedures of the *Access Cloud*. Section 5 presents a link-level mobility scheme, and section 6 draws the main conclusions.

2. Related Work

Recently, providers started using Ethernet in aggregation networks and in the mobile backhaul to reduce capital and operational expenditures. The cost reduction of Ethernet technology can also be exploited for 5G networks. However, the deployment of large Ethernet networks arises scalability issues. Moreover, Ethernet has not been designed to natively support mobility. Next subsections provide a bibliographic review of solutions for large Ethernet networks, and link-level mobility schemes. We commence the section with a review of SDN fundamentals and SDN-based cellular networks.

2.1. Software Defined Networks and SDN-based cellular networks

SDN is a new paradigm in network architecture design where control plane consists of a logically centralized controller that controls a set of inexpensive and simple network devices that make up the data plane. The most extended SDN control protocol is OpenFlow [5], an open protocol that allows the controller to configure the forwarding tables and to monitor packet statistics of switches.

Several works propose SDN-based backhaul and core mobile networks to exploit the benefits of costs reduction and flexibility offered by the SDN paradigm.

SoftCell is a SDN-based architecture for cellular backhaul/core networks aiming at enforcing fine-grained service policies [6]. It steers traffic through chains of middleboxes. To achieve scalability, *SoftCell* aggregates traffic to reduce the forwarding tables' size, and performs the packet classification at the access switches. To provide basic mobility, it enforces flows to traverse the same sequence of middleboxes.

MobileFlow is another SDN-based mobile network core composed by forwarding engines (MFFEs) and a controller (MFC) [7]. MFFEs are carrier-grade switches with

advanced functionality, configured by the MFC to forward user traffic to different middleboxes (e.g., video caching) through the network.

Although these interesting solutions apply SDN to 4G networks, some of its ideas are also applicable to 5G networks.

2.2. Large Ethernet Enterprise Networks

Ethernet networks' scalability is limited principally by the network-wide floodings caused by bootstrapping protocols used by end hosts (e.g. Address Resolution Protocol ARP, Dynamic Host Configuration DHCP, etc).

To overcome those limitations, new Ethernet-based architectures, such as SEATTLE [8] and Ethane [9], try to reduce the floodings of broadcast frames while offering the services expected for Ethernet.

SEATTLE aims at providing a configuration-free protocol which scales to large networks. To do that, it implements a network-level directory with *Distributed Hash Tables* (DHT) at SEATTLE switches to maintain the location of each host's media access control (MAC) address. Additionally, it uses caching and a unicast-based link-state advertisement protocol to avoid broadcasts for updating the forwarding tables. Broadcast ARP or DHCP messages are converted into unicast queries to the directory service. Although SEATTLE seems a promising solution for scaling Ethernet networks, it requires costly non-standard switches.

Ethane [9] proposes other architecture with an OpenFlow-like philosophy. Although Ethane aims at defining and enforcing network-wide policies, it addresses the broadcasts issue by delegating in the controller the handling of bootstrapping protocols such as ARP, reducing thus the broadcasts traffic and the switches' MAC table sizes. We will leverage on a similar SDN approach to overcome the scalability problem.

2.3. Link-level mobility

The authors in [10] propose a link-level mobility approach for 4G networks. They identify the disadvantages of the GTP protocol of 3G Partnership Project (3GPP) mobile networks, and propose Ethernet and Carrier Ethernet as transport backhaul. They propose a DHT location service to avoid flooding mechanisms for frames with an unknown destination address. In [10], the handovers within the same network segment

are unnoticed by the core network by storing the user id (e.g., IP prefix) and user location (e.g., base station MAC address) mapping at the customer edge switch.
Although this proposal focuses on 4G networks, some of its ideas can also provide benefits in 5G networks.

3. Envisioned Architecture for the 5G Core

The architecture for the core network proposed in [4] defines three hierarchical levels, listed bottom-up as the *Access Cloud*, the *Regional Distributed Cloud* and the *National Centralized Cloud* (see Fig. 1).
Each terminal has a 64-bits Extended Unique Identifier (EUI64) MAC address which is a unique identifier of the network interface of the terminal in the 5G network. The *Access Cloud* (AC) provides link-layer Ethernet communication to the attached terminals. Additionally, device to device Ethernet communications are supported between devices attached to Access Points (APs) in the same or different AC. We propose to implement the AC with APs and OpenFlow commodity switches, rather than with Ethernet switches (see section 4). The size of an AC is mainly determined by the scalability limits of the link-level technology. Yet, the AC can be subdivided if this limit is reached.
We foresee local breakouts within the AC for offloading the Core Network from best effort Internet traffic or accessing local services (such as content caching) with low delay.
The traffic from various ACs aggregates at the *Regional Cloud* (RC). The RC is composed of datacenters. It contains network-layer (L3) routing devices that act as gateway towards a high speed core and execute L3 mobility functions. They handle terminal mobility between ACs through Distributed Mobility Management (DMM) at L3 (see [4]). Additionally, the RC can have a regional breakout to provide a reduced latency Internet access. At L3, we use IPv6 to handle the large number of expected devices in 5G systems.
Finally, a high speed core network interconnects the different RCs between them, and these in turn to the *National Cloud* (NC). The NC keeps on logically centralized datacenters functionalities such as subscription information, charging, and interconnection to legacy Radio Access Networks.

The proposed architecture includes a SDN plane in charge of managing and orchestrating the networking control plane functions. The controller is composed of coordinated *Policy Decision Points* (PDPs) distributed through the architecture as close as possible to the devices they control. In the AC, APs and OpenFlow switches are *Policy Enforcement Points* (PEPs) managed by PDPs which logically form a *Local-SDN* (L-SDN) controller. In the RC, L3 routers are managed by a *Regional-SDN* (R-SDN) controller, which also operates in coordination with the *L-SDN* controller. Similarly, a *Centralized-SDN* (C-SDN) controller is kept in the NC. The objective is to partly or completely execute the control plane functions locally for improving delay and scalability performances.

4. Access Cloud Architecture and Procedures

In this section we address the design of the link-level architecture, and the procedures required to support Ethernet services for IPv6, starting with the principles of our design.

4.1. Design Principles

To satisfy 5G demands we have revisited the architecture of the mobile core network. Its currently centralized design is not adequate to fulfill the latency requirement, traffic volume increase and foreseen growth of connections. The mobility signaling explosion introduced by small cells and the massive connectivity of devices trigger requests and updates in the central nodes, be it Mobility Management Entity (MME) or gateways, causing a significant burden in the core infrastructure. Additionally, EPC still heavily relies on GTP to achieve terminal mobility. However, GTP tunnel management is cumbersome when handovers frequently occur [10][11]. Additionally, tunnels impair time-sensitive applications and introduce header overhead. Therefore, solutions more mobile friendly than GTP are needed.

Henceforth, to satisfy 5G requirements [2], we promote an architecture based on the following principles:

- Network Function Virtualization: our architecture calls for full virtualization of the EPC functionalities. This is a requirement to enable the distribution of the features on the vanilla switches of the SDN architecture, cheaper than dedicated gear to acquire and easier to evolve, reconfigure and maintain [7][12].

- Hierarchical distribution: the distribution of features across the network respects a hierarchical split between *Access*, *Regional* and *National* levels. In that hierarchy, the latency constraints guide the role distribution among the nodes, thereby enabling content caching at the network edge and offloading through breakouts towards Internet at the earliest possible stage. For instance, the L-SDN controllers address immediate Radio Access Technology related issues whereas the R-SDN controllers run at a slower pace to push optimal routes for a given traffic considering its Quality of Service (QoS) and the congestion status of the networks they monitor. The distribution of the roles hence reflects a trade-off between latency and reduction of costs and signaling traffic.
- Subsidiarity: we aim at a subsidiarity approach of the SDN paradigm, meaning that decisions should always be taken at the lowest possible level or closest to where they will have their effect. In this approach, the PDP are distributed close to the relevant PEP to achieve stringent delay constraints, from a few seconds at end-user/device session level down to fractions of milliseconds for procedures related to the physical layer at the radio interface.

To provide mobility, we aim at removing GTP while offering mobility at link-and network-level thanks to Ethernet and IPv6. The adoption of an Ethernet-based link-level approach at the *Access Cloud* allows reducing costs, since the related hardware is cheaper, and the plug-&-play nature of Ethernet eases its management. Additionally, terminal movements within the AC do not affect the devices' IP addresses since only the AC is involved.

4.2. Access Cloud Architecture

We propose to implement the 5G *Access Cloud* through a SDN-based link-level architecture. In the data plane, the AC comprises the APs and the access router that interconnects the AC with the rest of the 5G core. These network elements are located at the edge of the AC, and they will be hereafter referred to as *Edge Network Elements* (ENEs). Additionally, in the data plane, the AC comprises interconnected OpenFlow switches which communicate the ENEs of the AC between them. In the control plane, the AC comprises an L-SDN controller, which controls all the network elements in the data plane.

In our architecture, the controller establishes optimal routes between the ENEs and between each AP and the access router by signaling OpenFlow table entries to the switches. This enables to communicate each AP and its attached terminals to the rest of the 5G core. To enable device-to-device within the AC communications, routes between pairs of APs can be established.

For every incoming frame, the ingress ENE pushes a *Multiprotocol Label Switching* (MPLS) header with a *Label* field. At every switch, the frame is processed and forwarded according to the OpenFlow table entry for its label. The egress ENE simply removes the MPLS header and delivers the frame. If the egress ENE is an AP, the frame is passed to the radio link protocols for downlink transmission towards the destination terminal. If the egress ENE is the access router, the IP packet is passed to the IP layer. Although other tags could be used, we opt for MPLS because its header is lightweight and its 20 bits *Label* field enables a large number of routes.

The controller also stores and updates device location information by defining a *Device Location Table* (DLT) that has an entry for each terminal device attached to the AC. Each entry includes the EUI64 of a device and an associated label that specifies the AP to which the device is attached (see Fig.2). An entry with the access router's link-layer address is included. That is, the label identifies the egress ENE to which the frame should be forwarded to reach the destination link-layer address.

Each ENE keeps a modest cache of the *DLT*. The cache only keeps entries for devices to which recent packets have been destined. If a terminal sends a packet to an address not available in the cache, the ENE queries the controller for the corresponding entry, and the controller replies with the associated label.

Memory requirements for the different tables in the elements of our architecture are summarized in Table 1, as well as the processing costs in the different switches due to packet forwarding.

4.3. Link-Level Procedures

Our proposal aims at designing a scalable link-level access network which provides Ethernet service to IPv6 capable terminals. Since broadcast floods limit the network scalability, we filter them at ENEs and forward them directly to the L-SDN controller which replies with the proper messages. This strategy saves bandwidth and network resources.

In next subsections we identify the main operations which the link-level should support and how our design deals with them in a scalable way. Specifically, we describe the link-level multicasting, IPv6 autoconfiguration and IPv6 to MAC resolution.

4.3.1. Multicasting

IPv6 heavily relies on multicasting to accomplish some of its functionalities, exploiting the multicasting support of Ethernet.

Basically, the Ethernet address for an IPv6 multicast address is composed by the Ethernet multicast prefix 33-33- followed by the last 32 bits of the IPv6 address. Every Ethernet network interface (NI) keeps a list with the addresses which it is willing to receive (its own MAC address as well as the multicast addresses to which it is subscribed). Although some modern switches may selectively forward multicast frames to subscribers only, simple Ethernet switches just broadcast them.

In our design, the L-SDN controller keeps track of which terminals joined a multicast group, and configures the multicast tree to avoid unnecessary broadcasts. There exist IPv6 multicast addresses reserved for special uses. For example, a message sent to a solicited-node multicast address (SNMA, prefix FF02::1:FF00:0/104) [13] will be delivered to all the NIs which IPv6 addresses match the last 24 SNMA bits.

Frames with well-known prefix multicast addresses will be forwarded to the L-SDN controller, which will process them as explained in following subsections.

4.3.2. Stateless Address Autoconfiguration

An IPv6 host can configure its own address [13]. This autoconfiguration process includes generating a link-local address, generating global addresses via stateless address autoconfiguration, and verifying the uniqueness of those addresses by using the Duplicate Address Detection (DAD) procedure. Fig. 3 summarizes how our network implements this process.

The first step in the autoconfiguration is generating a link-local unicast address to communicate with neighboring nodes of the same link. A NI generates its link-local address by concatenating the prefix FE80::/64 to its interface ID (the EUI64 in our design).

After that, it checks if this address is already in use. To this end, it starts the DAD procedure by sending a *Neighbor Solicitation* message to its link-local address' SNMA.

If another NI in the network has that IPv6 address, it will send a *Neighbor Advertisement* message to the all-nodes multicast address (FF02::1). Otherwise, after a period, the address is assigned to the NI.

Once the link-local address is set, the NI generates a global IPv6 address by concatenating the network prefix and the EUI64. To discover the network prefix, it issues a *Router Solicitation* message to the all-routers multicast address (FF02::2). The ENEs will forward this frame to the L-SDN controller, which will reply with a *Router Advertisement* message, containing the router Prefix and other configuration parameters. After generating the global IPv6 address, the DAD procedure is repeated.

In our proposal, the burden of the multicast IPv6 messages is mitigated, since the ENEs forward to the L-SDN each frame with a multicast Ethernet address. The controller can identify the type of IPv6 message received, replying with the appropriate message. As the controller knows the assigned IP addresses, during the DAD procedure it can check its uniqueness. Since it knows the whole network configuration, it can reply with the *Router Advertisement* information when a *Router Solicitation* is intercepted. This strategy eliminates thus the multicasting of these IPv6 messages through the network.

Computational costs of the autoconfiguration procedure in the different elements are summarized in Table 1.

4.3.3.Address resolution

In IPv6, each network interface keeps a *Neighbor Cache* which stores IPv6 to MAC addresses bindings. To send a message to an IPv6 address which is not in its *Neighbor Cache*, the address resolution procedure is triggered [13]. It involves sending an IPv6 *Neighbor Solicitation* message to the address' SNMA to obtain the corresponding MAC address.

Since IPv6 addresses in our network include the EUI64, we can extract the link-level address from the IPv6 one without issuing messages.

5. OpenFlow-Based Link-level Mobility

Let us describe the link-level procedure performed when a terminal moves from the coverage area of a source AP (SAP) to the one of a target AP (TAP) (see Fig. 4). We will concentrate on the path switching within the AC, and we will keep the handover procedures associated with the radio interface protocols as in 3GPP-based systems.

When the signal level from a TAP overcomes a threshold, the terminal sends a *Handover Request* message to the SAP. Then, the SAP sends a *Handover Request* message to the TAP. Next, the TAP executes admission control. If it has available resources, it acknowledges the handover to the SAP, which, in turn, acknowledges it to the terminal (see Fig. 4).

After receiving the *Handover Acknowledgement* message, the SAP establishes a redirection procedure whereby it forwards frames to the TAP. It first forwards the frames it has buffered for transmission/reception to/from the terminal. Next, it forwards any new downlink frame received at the SAP from ingress ENEs. To differentiate frames forwarded by the SAP from frames sent by ingress ENEs, the SAP can use the *EXP* field of the MPLS header. This allows the TAP to prioritize the transmission of these frames in the air interface.

This redirection procedure allows supporting both seamless and lossless handover types [11]. In the case of lossless handover, the SAP must provide the sequence numbers of the forwarded frames. For this, the SAP can send a *Handover Context Information* message to the target which includes, for each frame, its sequence number at the radio interface, if it is uplink or downlink, and a frame identifier (e.g. with packet hashing).

When the terminal synchronizes with the TAP, it sends a *Handover Confirmation* message to the TAP. At this stage, the TAP transmits to the terminal all downlink frames forwarded from the SAP. Additionally, the TAP can directly send uplink frames to the destination ENE.

Next, the TAP informs the controller with another *Handover Confirmation* message. Then, the controller updates its *DLT* and it informs the ENEs with a signaling message. The ENEs with an entry for that terminal update their *DLT* caches. Additionally, each of these ENEs sends an *End of Path* (EOP) message to the SAP.

When the SAP has received all *EOP* messages from the ingress ENEs, it notifies the TAP, which can start the transmission of the downlink frames received from the ENEs through the new paths.

The estimated computational costs due to handovers are summarized in Table 1.

6. Conclusions and Future Work

In this paper we propose the design of the *Access Cloud* architecture to provide low latency and scalable Ethernet-like support to mobile terminals and MTC devices including mobility management.

The proposed design is based on the principles of NFV, hierarchical architecture, and subsidiarity. This brings the benefits of cost reduction, scalability in the data and control planes, and latency reduction as detailed in section 4.1. Additionally, the proposed architecture requires attainable forwarding tables in the AC switches. We achieve it by forwarding frames from and to edge network elements (ENEs) with an MPLS label which identifies the ENEs through which the terminal is reachable. Additionally, it requires updating the *Device Location Table* and *Caches* to track the mobility of the terminal.

Another advantage of the proposed architecture is that it addresses non-scalable Ethernet operations, such as broadcasting. This is achieved by letting the SDN controller filter these messages while still supporting the Ethernet procedures. Regarding bottlenecks of the AC architecture, we identify the SDN controller as a potential candidate as it processes the majority of the control plane signaling. Additionally, the *Device Location Cache* at the access router has to keep an entry for each active terminal.

Regarding the future work, several challenges lie ahead. One of the main challenges is the optimum distribution of control plane functionalities within the core network architecture taking into account the trade-off between latency, reduction of costs and signaling traffic. Another challenge is the implementation of the SDN controller. The alternative of distributing coordinated PDPs is considered in the architecture presented in [4], although mechanisms such as federating PDPs or maintaining PDP hierarchies are an open issue. Further challenges ahead are the impact on the core network architecture of procedures for session management (i.e. session establishment, QoS, and authentication & security) with support for MTC communications, interaction of link-level and network-level mobility procedures, and mobility management for inactive terminals.

Acknowledgements

We thank Andrea Fabio Cattoni and professor Preben Mogensen for their helpful discussions and insights. This work is partially supported by the Spanish Ministry of Economy and Competitiveness (project TIN2013-46223-P), and the Granada Excellence Network of Innovation Laboratories (projects GENIL-PYR-2014-20 and GENIL-PYR-2014-18).

References

[1] Cisco, "Visual networking index: Global mobile data traffic forecast update, 20132018," Tech.Rep., Cisco, 2013.

[2] P.K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, A. Benjebbour, "Design considerations for a 5G network architecture," *Communications Magazine, IEEE* , vol.52, no.11, pp.65-75, Nov.2014

[3] RAS Cluster, "5G Radio Network Architecture," Tech.Rep., Radio Access and Spectrum – FP7 Future Internet Cluster, 2014.

[4] A.F. Cattoni, P.E. Mogensen, S. Vesterinen, M. Laitila, L. Schumacher, P. Ameigeiras, and J.J. Ramos-Munoz, "Ethernet-based mobility architecture for 5G,"in Proceedings of the *IEEE 3rd International Conference on Cloud Networking (CloudNet)*, 2014.

[5] Open Networking Foundation, "OpenFlow Switch Specification, version 1.3.0," June 2012. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf Accessed:Dec.2014.

[6] Xin Jin, Li Erran Li, Laurent Vanbever, and Jennifer Rexford, "SoftCell: Scalable and Flexible Cellular Core Network Architecture," in *Proceedings of the ninth ACM conference on Emerging Networking Experiments and Technologies*, pp.163-174, New York, USA. 2013. DOI:10.1145/2535372.2535377

[7] Kostas Pentikousis, Yan Wang, and Weihua Hu, "MobileFlow: Toward Software-Defined Mobile Networks," *Communications Magazine, IEEE*, vol.51, issue 7, pp.44-53, Jul.2013. DOI:10.1109/MCOM.2013.6553677

[8] Changhoon Kim, Matthew Caesar, and Jennifer Rexford, "SEATTLE: A Scalable Ethernet Architecture for Large Enterprises," *ACM Transactions on Computer Systems(TOCS)*, vol. 29 issue 1, N.Y., USA. Feb.2011. DOI:10.1145/1925109.1925110.

[9] Martin Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker, "Ethane: taking control of the enterprise," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, vol. 37 issue 4, pp.1-12, N.Y., USA. Oct.2007. DOI:10.1145/1282427.1282382.

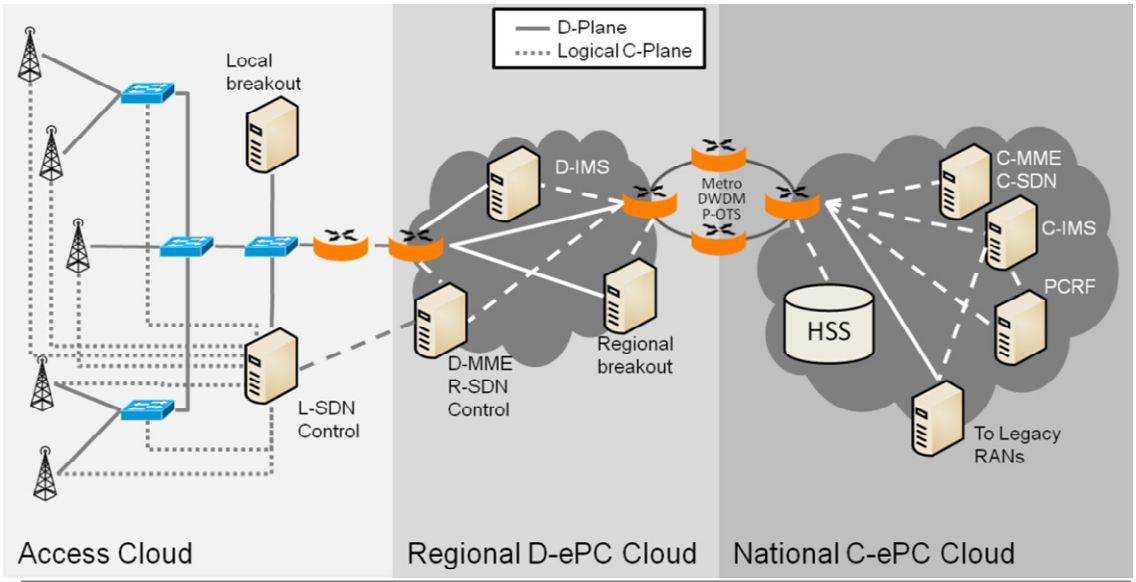
[10] Nuutti Varis, Jukka Manner, and Johanna Heinonen, “A Layer-2 Approach for Mobility and Transport in the Mobile Backhaul,” in *Proceedings of International Conference on ITS Telecommunications*, pp.268-273, Aug.2011. DOI:10.1109/ITST.2011.6060066

[11] S. Sesia, I. Toufik y M. Baker, "LTE: The UMTS Long Term Evolution from theory to practice," ETSI, France: John Wiley&Sons, 2009.

[12] Jain, R.; Paul, S., "Network virtualization and software defined networking for cloud computing: a survey," *Communications Magazine, IEEE*, vol.51, no.11, pp.24-31, Nov.2013. DOI:10.1109/MCOM.2013.6658648

[13] Rick Graziani, “IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6,” Cisco Press, Indianapolis, 2013. ISBN-13:978-1-58714-313-7.

[14] METIS (Mobile and wireless communications Enablers for the Twenty-twenty Information Society) project, deliverable D6.1, “Simulation Guidelines”, Oct.2013. Available https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D6.1_v1.pdf Accessed:Dec.2014.



C/D-IMS: Centralized/Distributed IP Multimedia Subsystem	L/R/C-SDN Control: Local/Regional/Centralized SDN Controller
C/D-MME: Centralized/Distributed Mobility Management Entity	PCRF: Policy and Charging Rules Function
HSS: Home Subscriber Server	RAN: Radio Access Network

Figure 1. Overall 5G Core Architecture.

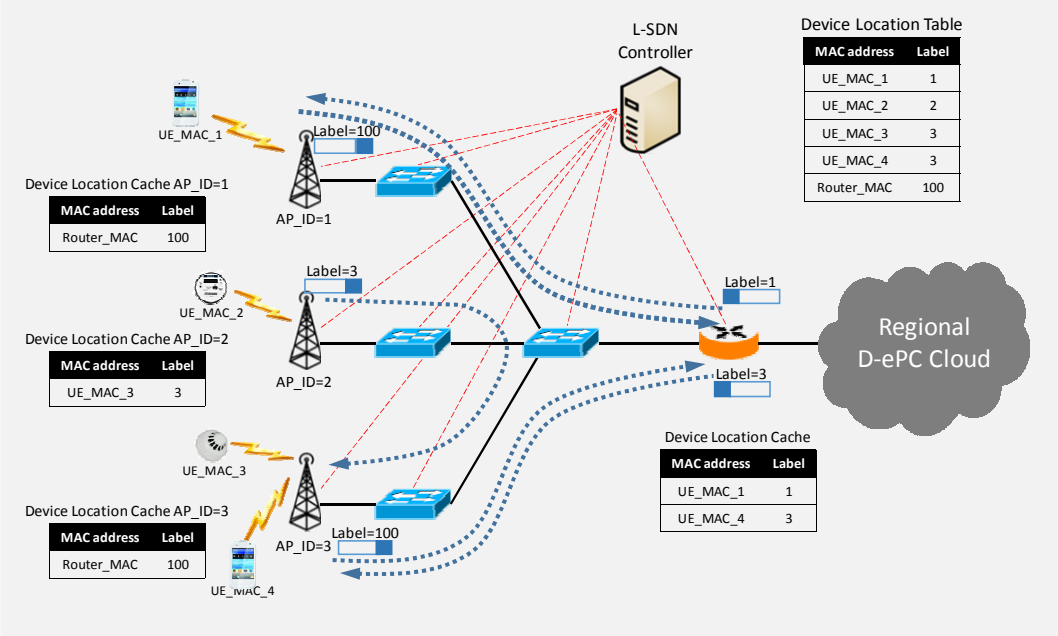


Figure 2. 5G Access Cloud Architecture.

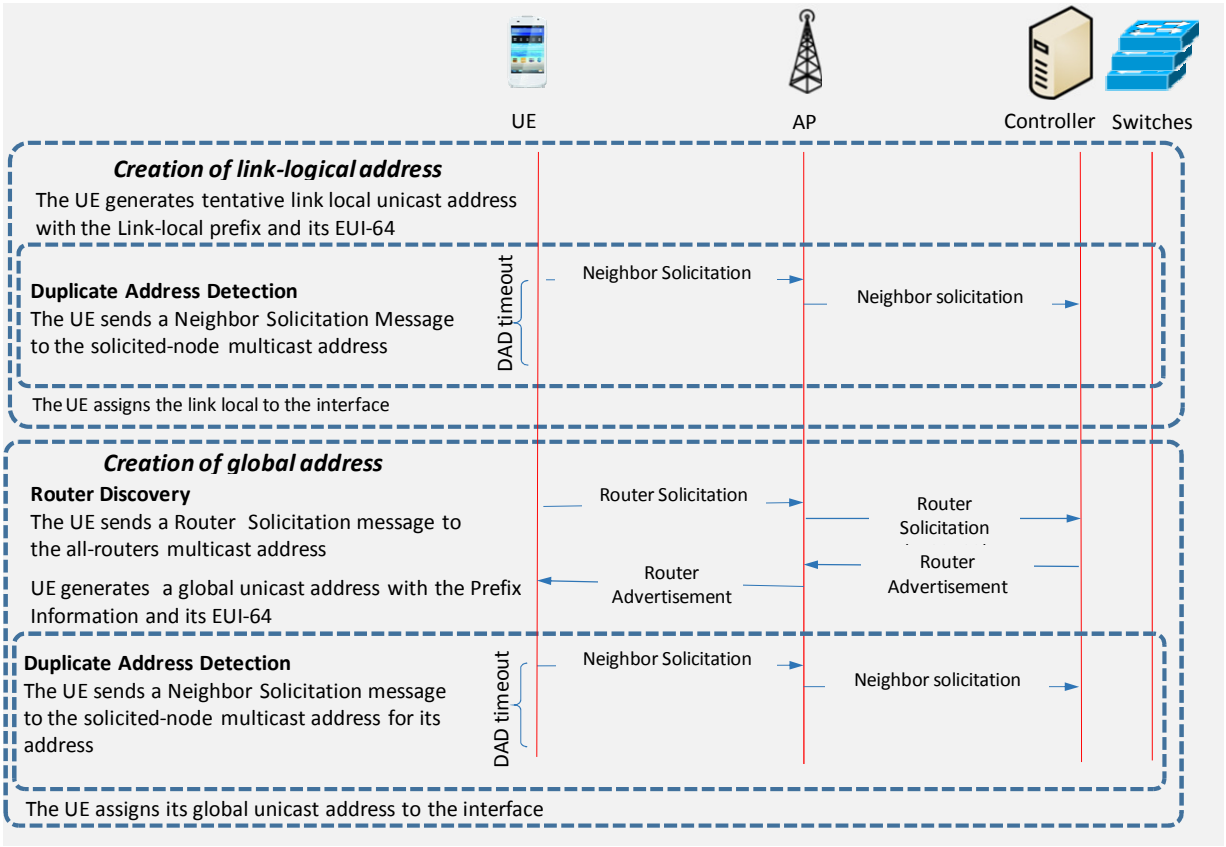


Figure 3. Terminal Device Autoconfiguration Process.

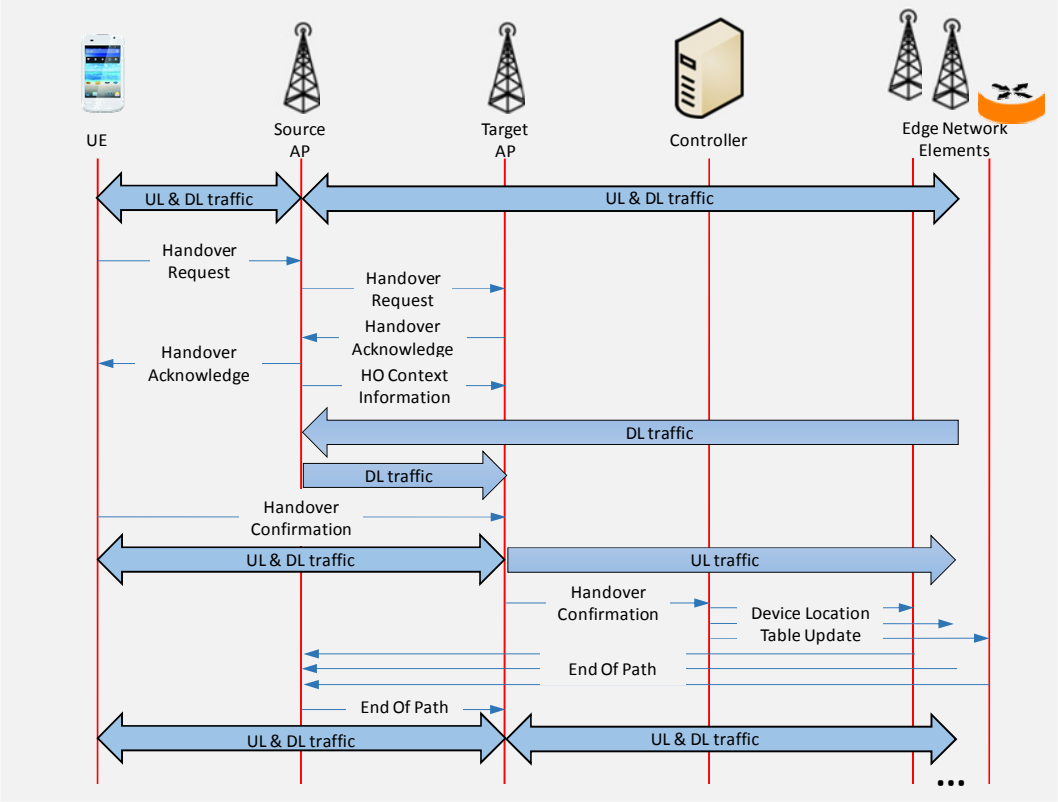


Figure 4. OpenFlow-Based Handover Procedure.

MEMORY CONSUMPTION		
ELEMENT	REQUIREMENT	SAMPLE SCENARIO
OpenFlow Switches	Flow tables: 1 entry per AP + 1 entry for Regional Router (worst case 297 bytes/entry[5])	30 KB (297bytes x (100 APs + 1 Regional Router)
L-SDN Controller	Device Location Table: 1 entry per user and sensor (84 bits = 64 bits (EUI64) + 20 bits (MPLS tag))	3.5 MB ((84 bits / 8 bits/byte) x (50,000 users + 300,000 sensors))
	Duplicate Address Detection table: 1 entry per user and sensor + 1 entry per Regional Router (2 IPv6 addresses per entry).	10.7 MB (2 IPv6 addresses x 16 bytes/address * (50,000 users + 300,000 sensors + 1 Regional Router)
ENE	Device Location Cache in AP: 1 entry assuming that traffic comes/goes to external network (e.g. Internet or MTC devices to a centralized server)	84 bits
	Device Location Cache in Regional Router: 1 entry per active user	5.1 KB ((84 bits / 8 bits/byte) x 500 active users)
PROCESSING COST		
ELEMENT	REQUIREMENT	SAMPLE SCENARIO
Regional Router	Control and data planes: for each packet… - DL: lookup in the Device Location Cache, push MPLS header, packet forwarding to the Access Cloud - UL: pop MPLS header, packet forwarding to core network	- DL: 1.06M lookups, 1.06M MPLS headers pushes, 1.06M packet forwardings - UL: 187.5K MPLS header pops, packet forwardings (1.25 Mpackets/s, i.e. 150 Gbps; 1.06 Mpackets/s (85%) for DL and 0.19 Mpackets/s (15%) for UL)
L-SDN controller	- Terminal device autoconfiguration: per UE attachment, reception of 3 control messages, lookup in DAD table, generation of 1 control message - Handover: reception of 1 control message, updates one entry in the Device Location Table, generates 1 control message	- Autoconfiguration:15 message receptions/s, 5 lookups/s, 5 messages generations/s (5 attachments/s) - Handover: 50 messages receptions/s, 50 entry updates/s, 50 messages generations/s (50 HOs/s)
Access Points	Control and data planes: for each packet… - DL: pop MPLS header, packet forwarding to the radio interface - UL: lookup in the Device Location Cache, push MPLS header, packet forwarding to the Access Cloud	- DL: 106K MPLS headers pops, 106K packet forwardings - UL: 19K lookups, 19K MPLS header pushes, 19K packet forwardings (125 Kpackets/s, i.e. 1.5 Gbps; 106 Kpackets/s (85%) for DL and 19Kpackets/s (15%) for UL)
OF switches	Data plane: for each packet… - Flow matching using its flow table, and output action	Type-1 switch: 2.5Mflow matchings, 2.5M output actions Type-2 switch: 12.5M flow matchings, 12.5M output actions (2.5 Mpackets/s per type-1 switch, i.e. 30 Gbps, 12.5 Mpackets/s per type-2 switch, i.e. 150 Gbps)
SAMPLE SCENARIO DESCRIPTION		
Based on [14] (*dense urban information society* and *massive deployment of sensors and actuators* scenarios) test cases) and [2] (small cells deployment) we have assumed the following sample scenario: Area 0.215 km^2, 100 APs, 50,000 users and 300,000 sensors, traffic load of 150 Gbps (85% DL, 15% UL), 500 active users, 1500 bytes/packet, negligible traffic from sensors, 0.1 handovers/active UE/s, 0.0001 attachments/UE/s, 5 type-1 switches (connected to APs), 1 type-2 switch (connected to type-1 switches and to Regional Router), 1 Regional Router		

Table 1. Computational costs for link-level mobility, handovers and stateless address autoconfiguration procedures.