SDN optimized caching in LTE mobile networks

Jose Costa-Requena, Maël Kimmerlin, Jukka Manner, Raimo Kantola
Communications and Networking Department
AALTO University
Espoo, FINLAND
{jose.costa, mael.kimmerlin,jukka.manner,raimo.kantola}@aalto.fi

Abstract — **This paper provides an overview of the current LTE architecture and the proposed solutions to integrate Software Defined Network (SDN) technology. The integration of SDN into mobile networks become Software Defined Mobile Network (SDMN), which provides new benefits such as dynamic and efficient caching. Firstly, the paper proposes the integration of SDN in LTE networks on disruptive approach that replaces current mobile transport with SDN based network. Secondly, the benefits of SDN integration to provide optimized content caching is presented. This paper presents a concrete advantage of SDMN for content delivery. Thus, the paper describes the benefits from operator and end user points of view based on a simple scenario of content delivery using dynamic caching reallocation.**

Keywords—caching; LTE; SDN; SDMN

I. Introduction

Major challenge of future mobile networks is providing the needed throughput for the increased traffic demand with acceptable cost. This paper proposes a vision of how this can be done by applying the concept of Software Defined Networks (SDN) to mobile networks. Moreover, we propose a disruptive transport based on the benefits of SDN technology. This new transport mobile networks allows to design and deploy an optimized content delivery system based on dynamic caching. While we have verified some key parts of the vision with experiments, we realize that the effectiveness of the proposed approach depends on the adoption of SDN technology for other purposes so that mass production of SDN switches leads to significant economies of scale. The paper shows how we can model mobile networks using SDN concepts and migrate the 3GPP [1] mobile architecture to SDN, thus setting the basis for 5G networks. The resulting SDN based mobile networks with the disruptive transport provides the basis of a caching system enhanced with dynamic reallocation. The data plane consists of simplified access points and SDN switches. Our experiments are based on using OpenFlow as the interface between the control and data planes.

By the end of 2013, the number of mobile devices has surpassed the total population of the world, and by 2017 there will be nearly 1.4 mobile devices per user [2]. Furthermore, it is anticipated that a significant portion of the generated traffic will be due to video-related services. NSN predicts [3] that from 2010 till 2020 the traffic in mobile networks will grow 1000 times. To meet this demand mobile networks will have to improve spectral efficiency 10-fold. Another factor of 10 comes from additional spectrum being made available for mobile use. The last factor of 10 will need to come from increased number of small base stations. It is natural that operators will invest last to increase the number of base-stations because this has the highest cost. At the same time users are unlikely to want to pay more than they are paying today for the service. For the mobile operators and vendors this sets a serious challenge: how to improve the technology 1000 times without increasing the cost including both CAPEX and OPEX.

In particular, operational expenditure tends to grow significantly as a function of the number of network nodes or sites. To lower the growth curve for the increased number of (small) base stations, it makes sense to design the physical base-station as simple as possible; it will mainly consist of an antenna and an Ethernet card for the backhaul connection. The control software of a number of small base-stations will be run remotely at a location that is relatively close to the physical base stations for keeping the time delays for the communication low. At the same time we believe that it is wise to avoid replacement investments. Therefore the architecture must provide interworking for unchanged legacy base-stations. In this paper we take LTE as the starting point for migration. We show how to structure the LTE control functions into a chain of SDN applications, so that the data plane of the mobile network can be built using standard OpenFlow [4] switches. A technical challenge is that the 3GPP architecture heavily relies on mobile network specific tunneling methods in order to hide mobility from the core of the Internet and OpenFlow does not directly support these tunneling methods. We propose a disruptive data plane where the tunneling methods are removed. Instead we rely on L2 or L3 transport and SDN to perform all the required QoS, mobility management and security, thus eradicating the need of mobile specific tunneling techniques. As a result of using SDN we can remove the GTP tunnel from the current LTE networks. This results in the possibility of placing caches besides any switch in the network. Previously, the fact of having GTP tunnel between eNodeB abd the Packet GW (PDN) allowed placing the caching either in the eNodeB before data enters the GTP tunnel or after the PDN once the data leaves the GTP tunnel.

The paper is organized as follows: Section II gives a background on mobile networks. Section III lays down our vision about data plane without mobile specific tunneling. Having replaced mobile transport with SDN based data plane we show the benefits to optimize the content delivering with dynamically allocated caching in section IV.

978-1-4799-6786-5/14/$31.00 ©2014 IEEE

II. LTE Mobile Networks

Mobile networks consist of physical and logical layers. The physical layer is made of network switches (L2), routers (L3) and physical links with different technologies and topologies as shown in Fig 1. The logical layer consists of network elements (e.g. eNodeB, MME, S/P-GW, HSS, etc) that perform the attachment of user devices, mobility and transport of data from mobile devices across the mobile network. The physical layer (L2 and L3) provides the connectivity and transport functionality to the logical layers that implement the mobile specific control functions. The access network consists mainly of the eNodeBs that provide the radio access to the User Equipment (UE). The backhaul consists of all the network switches for aggregating the traffic from the access network and provide the connectivity towards the core network. Finally, all the connection services, mobility services and billing functionality are implemented by the network elements (i.e. MME, S/P-GW, PCRF, HSS) located in the core network.

Mobility is a critical functionality in mobile networks and a new technology has to deliver a reliable and low latency handover. Mobility in LTE networks is implemented through different methods depending on whether the new target eNodeB is under a different Tracking Area ID (TAI) associated to a different Mobility Management Element (MME) or the new TAI is managed by the same MEE. In the former scenario mobility management uses the S1-MME interface between the eNodeB and the MME. The latter scenario mobility management uses the X2 interface between eNodeBs. The handover process is based on the S1-MME interface between logical elements (e.g. eNodeB, MME and S-GW) when the change of MME is required.

A fundamental problem in IP protocol from the mobility point of view is that the IP address identifies the node and fixes its location to a certain IP subnet. The common solution in mobile networks consists of using tunneling UE IP packets in GTP tunnels that are established between the eNodeB and the S/P Gateway. A GTP tunnel uniquely identifies traffic flows that receive a common QoS treatment between a UE and a P-GW. The traffic Flow Templates (TFT) are used for mapping traffic to an EPS bearer.

A. Logical elements and mobility control process

The GTP tunnel endpoint identifier (TEID) unambiguously identifies the tunnel endpoint of a user data packet, separates (identifies) the users and also separates the bearers of a certain user as depicted in Fig 1.

When an UE moves to a new eNodeB, the GTP tunnel has to be recreated between the new eNodeB and the S/P Gateway while the inner data flow keeps using the original UE IP address.

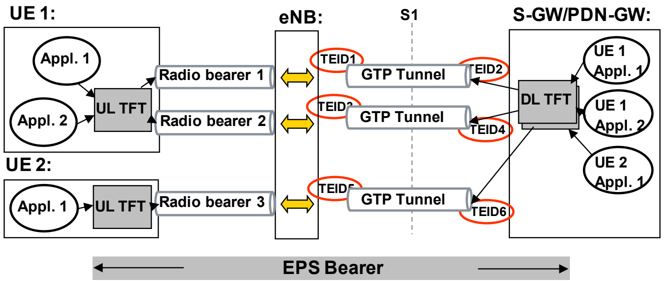

Fig. 1. Tunnelling of user traffic over mobile networks

The handover process is initiated and managed through the S1 interface as shown in Fig 2. MME is aware of the mobility process and communicates with the S/P-GW to recreate the GTP tunnel between the new eNodeB and the S/P-GW as shown in Fig. 3.

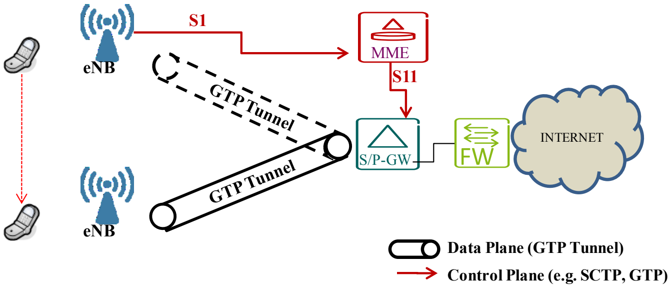

Fig. 2. Handover process controlled from MME through S1 and communicated to S/P-GW to recreate a GTP tunnel

III. SDN Mobile Data Plane

Integrating SDN controller functionality with MME provides a smooth integration in the long term as well as disruptive solution in mobile networks. The current SDN have to enable mobile specific requirements where the data plane is optimized for high speed and flow-level processing (using OpenFlow). In SDMN the control plane is moved out of the basic networking elements into centralized servers – these servers resemble classical *anchor points* used on many mobility protocols. Therefore, the proposal is to move controller and current S/P-GW functionality in the same network element together with the MME functionality. Therefore, the current S/P-GW functionality disappears and instead a SDN based switched packet network is used. This approach will add flexibility and value to networking with different increments and support the gradual introduction of high network throughputs, optimal flow management and traffic engineering possibilities. Fig 3 shows the integration of mobility with SDN controller and included as part of MME network element. In the proposed integration we remove GTP transport entirely and replace it with Ethernet VLANs or MPLS. Mobility is a major requirement in mobile networks which needs to be handled properly with minimum delay and reduced signaling. In current mobile networks use GTP for maintaining the sessions during handover process but SDN will replace GTP mobility functionality.

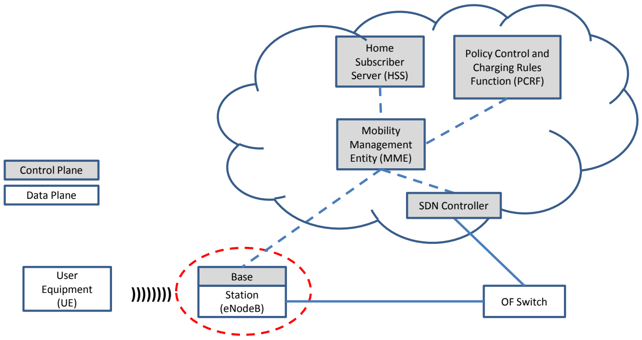
Home Subscriber Server (HSS)
Policy Control and Charging Rules Function (PCRF)
Mobility Management Entity (MME)
Control Plane
Data Plane
SDN Controller
User Equipment (UE)
Base Station (eNodeB)
OF Switch

Fig. 3. Disruptive integration of SDN with MME.

Having tight linkage between the MME and the SDN controllers allows that the time-constrained functions of mobility are handled efficiently from the SDN controller. This integration provides efficient handover management in SDMN. The OpenFlow controller adds and removes the flow entries from the flow table as soon as handover event is received from the MME. An entry in the flow table has three fields: a packet header to define the flow, an action that defines packet processing, and finally statistics. Besides the integration of the MME and the SDN controller we propose using 802.1ad to allow double tagging in Ethernet switches.

This double tagging allows having up to $2^{\wedge 12}$ service tags as the outer tunnel and $2^{\wedge 12}$ customer tags for inner tunnels (i.e. total of 2096 inner and outer tunnels). These outer VLANS can be used for establishing tunnels between the eNodeBs and IP router located in the same Ethernet segment to provide access to public Internet. The outer VLANS established from the eNodeBs can be assigned to different Mobile Virtual Network Operators (MVNO). The $2^{\wedge 12}$ inner tunnels can serve 10 MVNO in an area of 400 eNodeBs.

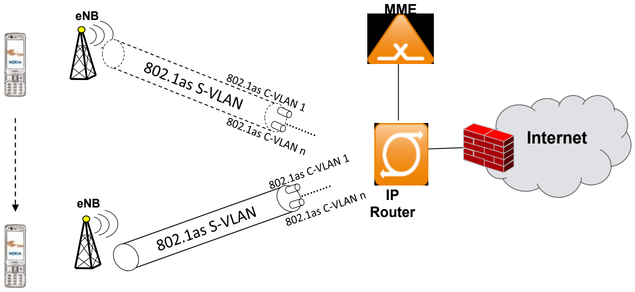

Fig. 4. VLAN tunneling between eNodeB and IP router.

The integration of SDN in LTE networks provides benefits in terms of CAPEX and OPEX since control functionality is deployed as cloud services, thus benefit from commodity computing facilities. Another benefit resides in the fact that the transport network is simplified by removing the GTP tunnel for the data plane. This allows using commodity off the shelf OpenFlow switches which will provide the required data forwarding features controlled from the cloud.

From the different options of integrating SDN with LTE the proposal of including together the MME and the SDN controller brings several benefits. The controller needs to have the necessary information about the location of the UE and the associated mobile operator as well as the necessary attachment and handover events. Therefore, the controller should be integrated with the MME and S/P-GW to receive those events and perform the required MAC in MAC and Q in Q mapping. Moreover, this integration results in the next disruption where data plane is managed from a single MME/Controller element. The evolution towards this architecture can be done progressively where the MME will keep current interfaces for receiving the signaling through S1-MME interface. The MME maintains the current standard process and establish GTP tunnels between legacy eNodeB and S/P-GW. Simultaneously the MME can include the new SDN functionality establish communications between them new model of eNodeB and IP router directly at layer 2 without GTP tunneling. In this scenario the same MME when receiving the signaling from the SDN based eNodeB through the S1-MME interface it will establish the connection with the termination SDN switch over L2 using TUN interfaces.

The networking stack currently used for the user plane is depicted in Fig 5. The radio layers are terminated in the eNodeB from where GTP is used up to the S-GW and the P-GW that provides the bridge to the public Internet.

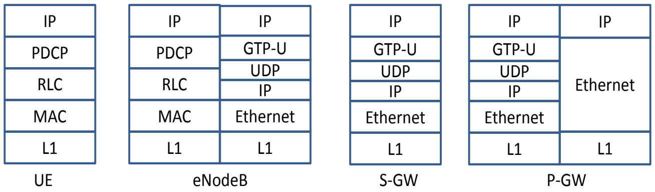

Fig. 5.a. Current LTE user plane networking stack.

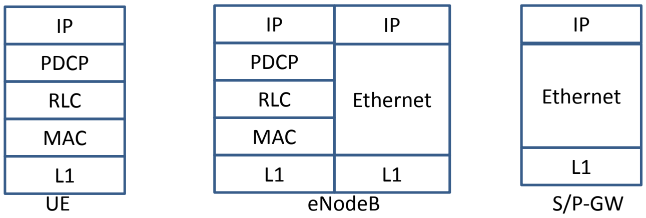

Fig. 5. b. SDN based user plane networking stack.

The usage of 802.1ad in the backhaul and integration of MME with the SDN controller allows the removal of GTP. This will result in the simplification of the stack in the eNodeB that terminates the radio layers and includes Ethernet switch towards the rest of the network in the backhaul as shown in Fig 5.b. Moreover, the S/P-GW is simplified after removing the GTP and consists of simple Ethernet switch and IP router towards the public Internet. In this architecture the mobility is performed by the SDN controller.

This architecture leads to an optimized transport network as well as scalable control plane that converges into single network element now MME with embedded SDN controller functions. This MME would run in either dedicated HW or as cloud service to allow launching multiple instances as needed to overcome scalability of having all functionality into single network element.

IV. SDN Optimized Content Delivery

The always-increasing role of content delivery networks (CDN) in the Internet traffic shows a clear shift to content consumption. CDNs leverage the power-law nature of content popularity distribution where many users request popular contents within short period of time. Therefore, storing a copy of popular contents in caches placed at the proximity of end-users reduces server load, decreases network congestion, and lowers delay. In the context of 5G mobile networks, placing caches directly at the edge would be of great benefit for the network. Still, placing caches exclusively at the base stations is not the ultimate solution as the number of users that would use the cache would be too limited to really benefit from demand patterns. Instead, we advocate the usage of multi-stage caches with a rather small general purpose Last Recently Used (LRU) cache collocated in every base station to absorb retransmission events high temporal locality demands (e.g., live streaming), and large caches spread over the different points of presence and large cache spread over the different points of presence and able to aggregate traffic of a large portion of users, reducing so the traffic in the core. By looking at the reasonable backplane speeds in our 100M 5G network, we deduce that it is economical to connect large caching servers behind the CGE switches each serving close to 1 million users. Cache hits would then reduce the number of required high cost eOFS switches as well as reduce Internet connection charges of the mobile operator.

The usage of SDN with the proposed integration facilitates the dynamic relocation of the cache based on the number of users. We deployed pilot to demonstrate the effect on the network when moving the cache. We use HTTP live streaming for video file and the tests have been performed with the architecture presented in Fig 6.

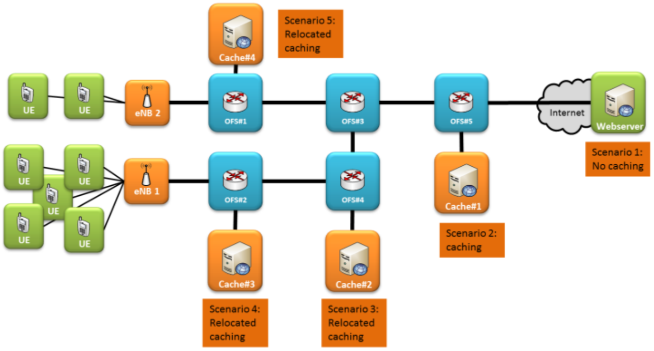

Fig. 6. Architecture of the testbed

In the first scenario, all the requests are forwarded to the gateway to fetch the web server. In the testbed, we used the gateway as the webserver so that there are no external limitations coming from fetching a web server on internet.

The tests were performed with increasing number of users. They have been performed with up to 16 users on a base station. But we saw an increasing difference in the bandwidth consumption with the expected one, when there were more than 8 users. So the results are only presented with up to 8 users on each base station

This might be because of the way we performed those experiments. Due to limitations in the implementation of the testbed, we simulated a user streaming a video by an http download of the segments with a bandwidth limitation to 2Mb/s, and multiplying the bandwidth limitation by the number of users simulated. Since we are using HLS, there are some breaks between the segment downloads; and with high number of users, the breaks between the files becomes more frequent so they may have an influence on the calculation of the peak bandwidth consumption by the network measurement tool that is calculating it. Fig 8 presents the network load for the different scenarios for a 2mb/s stream, with 5 users on enodeB 1 and 2 on enodeB 2.

The load is calculated as the sum of the loads, related to the download, of each link.

$$L = \sum_{j=1}^{N}\sum_{i=1}^{n} a_i b_{ij} \qquad (1)$$

With N the number of users, n the number of links, a_x the video stream throughput, b_{xy} = 1 if the traffic of the user x is going through the link y or 0 if not.

From Fig 7 we can see that when the content is closer to the base stations, the overall load induced by users fetching this content is reduced. This is what we want to achieve by using cache relocation.

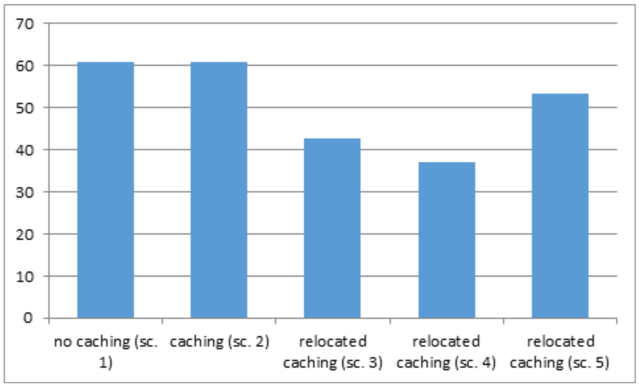

Fig. 7. Load for different type of caching

Then, for the same amount of users per base station (5 on the first one and 2 on the second), Fig 8 presents the impact of the video throughput on the network load.

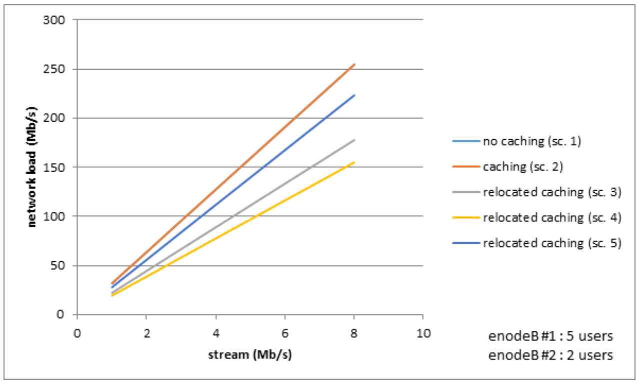

Fig. 8. Network load depending on the video throughput

As expected, the network load is proportional to the throughput of the stream. So for the other tests, we will set it to 2Mb/s. Those results proves that, as expected, a relocated cache is more efficient to reduce the load of the network but only if it is close relocated close enough to the majority of the users requesting the content. If it is not the case, it may perform worse than the original cache, depending on the number of hops to the users. Finally, with a good relocation caching, consequent bandwidth can be saved.

V. Conclusions

The separation of control and data planes has the potential to provide cost savings from capacity sharing and provide economies of scale from the virtualization of network elements in the cloud. The usage of SDN will bring down the costs of acquiring and maintaining standard switches. The separation of control from data plane will lead to the usage of general-purpose switches without mobile dedicated solutions. One of the important benefits of the proposed SDN integration and data plane without mobile tunneling is to allow dynamic relocation of the cache. The results shown might be applicable to any caching but the usage of SDN to remove the GTP allows that caching can be performed besides any switch of the network.

Acknowledgment

The results included in this paper are the result of the work at CELTIC project named SIGMONA funded by Finnish Research Agency TEKES and EIT ICT SDN project.

References

[1] LTE architecture (http://www.3gpp.org/LTE)

[2] Cisco White Paper. Cisco Visual Networking Index: Forecast and Methodology, 2009-2014, June 2010.

[3] NSN White Paper, “Technology Vision 2020, Technology Vision for the Gigabit Experience, June 2013.

[4] OpenFlow (http://www.openflow.org/)