MOCA: A Lightweight Mobile Cloud Offloading Architecture

Arijit Banerjee
School of Computing
University of Utah
50 S. Central Campus Drive
Salt Lake City, UT - 84112
arijit@cs.utah.edu

Xu Chen
AT&T Labs - Research
180 Park Ave
Florham Park, NJ - 07932
chenxu@research.att.com

Jeffrey Erman
AT&T Labs - Research
180 Park Ave
Florham Park, NJ - 07932
erman@research.att.com

Vijay Gopalakrishnan
AT&T Labs - Research
180 Park Ave
Florham Park, NJ - 07932
gvijay@research.att.com

Seungjoon Lee
AT&T - Labs Research
180 Park Ave
Florham Park, NJ - 07932
slee@research.att.com

Jacobus Van Der Merwe
School of Computing
University of Utah
50 S. Central Campus Drive
Salt Lake City, UT - 84112
kobus@cs.utah.edu

Abstract

We present our work on MOCA, a lightweight Mobile Cloud Offloading Architecture, which uses an in-network cloud platform to provide offloading resources. MOCA integrates with existing mobile network architectures without requiring significant changes, and utilizes software defined networking techniques in the data plane to redirect appropriate traffic to and from the cloud platform. We show the feasibility of MOCA by a prototype implementation using a LTE/EPC mobile testbed.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

Keywords

Mobile architecture; LTE; Cloud; Traffic offloading; SDN

1. INTRODUCTION

New mobile devices are putting tremendous strain on current mobile networks both in terms of the amount of traffic they generate and the manner in which they interact with the network. Particular trends include increased use of rich media content on mobile devices [1] and the use of highly interactive mobile applications, like online gaming [13]. It is widely understood that current mobile network architectures, strongly resembling their voice-only forebears, are poorly suited to these applications [9]. In particular, current mobile network architectures are highly centralized, which result in inefficient routing and high delays [5]. The potential value of various offloading strategies to address these concerns have been demonstrated [6, 12]. However, due in part to the inherent complexity and the resulting difficulty to affect change in these networks, these proposals have had very limited impact despite their potential benefits [16].

From a technology trends perspective, widespread and increased adoption of cloud computing in various domains and the emergence of software defined networking (SDN) as an agent for change in networking, are widely expected to have a long term impact on future mobile networks [11, 4, 10]. We would argue, however, that these initial works fail to capture the true potential of combining these different technologies and again, are hampered by the complexity of existing mobile network architectures.

In this paper, we present our work on taking a pragmatic approach to realize offloading utilizing the power of SDN and cloud technologies. Specifically, we present MOCA, a lightweight Mobile Cloud Offloading Architecture, which enables offloading of mobile traffic to *in-network cloud computing platforms*. A key design principle in our work has been the desire to be able to adopt SDN and cloud technologies into our architecture, without requiring a complete clean slate redesign of the mobile architecture. In MOCA, we realize this principle by having tight interaction between the existing mobile architecture and our offloading infrastructure, but in a manner that requires minimal changes to the existing mobile infrastructure.

The key components of MOCA include: (i) Small extensions to the signaling protocols and functionality of the mobility management entity (MME) in an LTE/EPC network to drive *dynamic offloading* of specific mobile traffic. (ii) The use of an in-network cloud platform to dynamically create *cloud-based mobile network elements* as well as *customer resources* to make use of offloading. (iii) The use of software defined networking technology to facilitate *data plane redirection* mechanisms to allow designated traffic to reach the cloud offloading platform.

Our proposed architecture is grounded in the reality that architectural changes are often driven by business needs and opportunities. This observation informs the MOCA archi-

MobiArch'13, October 4, 2013, Miami, Florida, USA
Copyright 2013 ACM 978-1-4503-2366-6/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505906.2505907.

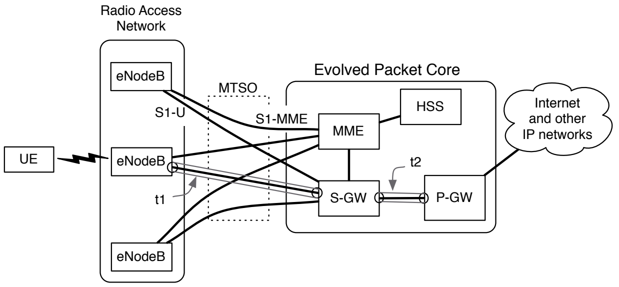

Figure 1: Standard LTE/EPC Architecture

tecture in that we assume offloading to be a special case that might be associated with business arrangements between the mobile network provider and the entities receiving a benefit from offloading. This approach mimics current business arrangements in content distribution and cloud computing. From a technical perspective, focusing on the specific needs of in-network cloud offloading allows us to explore the benefits of SDN and cloud computing in mobile networking, without having to deal with the complexities involved with a complete re-architecting of the mobile network.

While narrow in scope, we believe that our approach presents the first step towards a more comprehensive and general evolution of mobile network architectures through the use of cloud and SDN technologies. Likewise, because of the reality of resource constraints and dynamic workloads in any network, we expect the mechanisms associated with dynamic offloading and redirection to have general applicability in future mobile network architectures.

To summarize, we make the following contributions:

- Design and implement an in-network cloud offloading architecture.
- Extend current LTE/EPC signaling protocols to realize dynamic offloading, while requiring minimal changes to the existing mobile network architecture.
- Use SDN and cloud technologies to realize the offloading architecture in an efficient manner.
- Implement a proof-of-concept realization of our proposed architecture on an LTE/EPC testbed.

2. BACKGROUND

We provide a brief description of the long term evolution (LTE) and evolved packet core (EPC) architecture as representative of mobile networks currently being deployed. As depicted in Figure 1, LTE/EPC packet system consists of three components: the User Equipment (UE) (i.e., cell phone or other mobile device), the Radio Access Network (RAN), and the Core Network. LTE RAN consists of the eNodeB (enhanced NodeB), which communicates with mobile devices (i.e., UEs) via the radio link and then forwards user packets to an S-GW (Serving Gateway) via the S1-U interface. The eNodeB also performs radio resource control and cooperates with the MME (Mobility Management Entity) for mobility management (e.g., location update, handover) via the S1-MME interface. The EPC packet core consists of the MME, the S-GW, and the P-GW (Packet Data Network Gateway). The MME is a control plane only device responsible for mobility management and user authentication via the HSS (Home Subscriber Server). It also interacts with S-GW for data session establishment/release. The S-GW and the P-GW are on the data path, and their main function is packet routing/forwarding, traffic management and accounting, and policy enforcement. While the S-GW serves as the mobility anchor point for roaming users, the P-GW acts as a gateway to the external network (e.g., the Internet). To provide large geographic footprint and high quality service, a typical cellular service provider deploys tens of thousands of eNodeBs. There are far fewer mobile core components (MMEs, S-GWs, P-GWs), e.g., low hundreds, and these are typically deployed in a few centralized locations [16].

Like its predecessors, LTE/EPC employs hierarchical routing where packets are routed using GTP (GPRS Tunneling Protocol) tunnels among core components, before they are sent out in the Internet. The data path from an UE to the Internet consists of two GTP tunnels - one from the eNodeB to the SGW (t1 in Figure 1) and, the other from the SGW to the PGW (t2 in Figure 1). A data (IP) packet from the UE is sent to the eNodeB via the RAN, where it is encapsulated in a GTP header and sent to the SGW using t1. The SGW, in turn, extracts the original IP packet and encapsulates it again in another GTP header to send it to the PGW using t2. The PGW decapsulates the original IP packet and sends it to the public Internet. The reverse traffic from the Internet follows the same path back to the UE via the PGW, SGW and, the eNodeB. GTP tunnels map to the logical concept of "bearers" which determine the treatment (priority, QoS etc.) packet flows from a UE is subjected to. When a UE first attaches to the network, a "default bearer" with no associated guaranteed QoS (best effort) is created for it.

Figure 1 also shows the so called Mobile Telephone Switching Office (MTSO) in relation to the LTE/EPC infrastructure. Mobile operators typically aggregate traffic from a large number of eNodeBs covering a large geographical area (comprising of several metropolitan areas) in a regional MTSO before they are sent to the operator core network. Note that, all the LTE/EPC elements are oblivious to the presence of MTSOs which are part of the underlying operator network infrastructure. We include a description of MTSO here as this is relevant for the placement of the data-plane redirection mechanisms of MOCA.

3. MOCA ARCHITECTURE

Mobile networks, as described in Section 2, employ hierarchical routing. This causes packets originating from (and destined to) geographically dispersed mobile devices to be tunneled to a few S/P-GW locations, traversing extra distance which results in significant additional delay [5]. This sub-optimal routing is especially problematic for delay sensitive applications and services like online gaming [13] and content distribution [16].

To address these concerns, we present a practical lightweight architecture for mobile offloading. We assume the existence of in-network cloud platforms distributed throughout the mobile provider core network. This cloud infrastructure presents general computing resources for applications and services that could benefit from offloading. For example, and without loss of generality, we assume in the remainder of the paper that a gaming service provider is the entity that wants

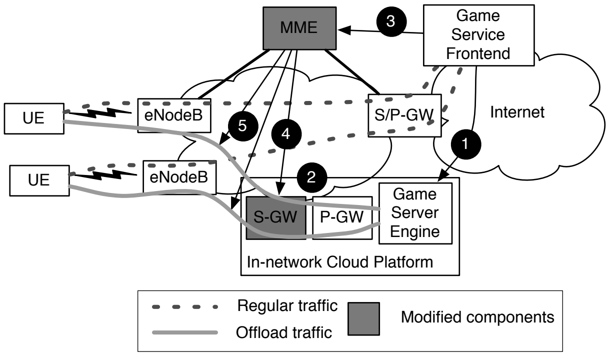

Figure 2: MOCA Dynamic Offloading

to make use of the cloud platform and the offloading mechanisms. As outlined below, the cloud platform resources are also used to dynamically instantiate mobile networking elements to realize the offloading architecture.

3.1 Overview

The main architectural components of our proposed solution is depicted in Figure 2. We assume the existence of a cloud platform *inside* the mobile provider network. We further assume that, like regular cloud instances, such cloud resources are made available to customers on a commercial basis, e.g., in this case for the gaming service provider. The gaming service provider instantiates a game server engine on the in-network cloud platform – (1) in the figure. The purpose of the in-network cloud platform is for off-loading traffic from mobile devices. As such the cloud platform is also an integral part of the mobile network, and the mobile network utilizes the cloud resources to instantiate virtual machine instances of mobile network elements. For example, as shown by (2) in Figure 2, the network operator can instantiate a software instance of an S/P-GW on the cloud platform and associate it with the game server engine. We assume that the game service provider still maintains a game service front-end on the Internet. However, it has the ability to request the mobile network to send game specific traffic to the in-network game server engine. This is depicted by (3) in Figure 2. We note that in general this “trigger” to request the instantiation of cloud resources and activating offloading to use it is application/service specific. Based on this request the mobile network proceeds to take two traffic manipulations actions. First, as shown with (4) in Figure 2, the network signals to the appropriate cloud based S/P-GW to prepare it for game specific traffic from the mobile device. Second, as shown with (5) in Figure 2, the network needs to divert game specific traffic (using software defined network policies) towards the (in-network) cloud-based S/P GW and game server engine.

3.2 Realization

As outlined below, our solution requires relatively modest modifications to the MME component in the mobile network. All other existing mobile network components (UEs, eNodeBs, S/P-GWs) continue to function as before. Further, the new cloud based S-GW is a slightly modified version of a standard S-GW, while the cloud-based P-GW is unchanged. Note that the use of an SDN substrate to direct appropriate flows to the new S-GW is crucial to meet our driving objective of realizing MOCA without requiring significant changes to the existing mobile architecture. Specifically, this eliminates the need to modify the eNodeBs to use the new functionality. (Given the large number of eNodeBs in a typical mobile network, requiring any such modifications would in practice be infeasible.)

Dynamic offloading. The message flow for our dynamic offload implementation is depicted in Figure 3. We clone the GTP tunnels associated with the default bearer in the new S/P-GWs and install routing rules in the core network to direct traffic destined to the game server to the new S/P-GWs. The MME records the following parameters during the initial attach (which also triggers the default bearer creation) of the UE: IP address assigned to the UE by the PGW (UE-IP), SGW User Plane Tunnel End-Point Identifier (SGW-TEID), and eNodeB IP address (ENB-IP) and User Plane Tunnel End-Point Identifier (ENB-TEID). The first two are available in Create Session Response control message from the SGW, and the third is available in Initial Context Setup Response control message from the eNodeB.

Once the initial attach procedure is complete, the MME identifies a new cloud-based SGW (learned from step (3) in Figure 2) and triggers offloading procedure by sending the new SGW an overloaded Create Session Request control message, which contains UE-IP and SGW-TEID to be used for offloaded traffic.

The new cloud-based SGW creates uplink S1-U GTP data tunnel using the MME supplied SGW-TEID, so that the new SGW has exactly the same bearer state for uplink traffic as its original counterpart, and its GTP layer accepts any offloaded traffic belonging to the default bearer. Similarly, the new SGW uses ENB-IP and ENB-TEID to clone S1-U GTP data tunnel state for downlink traffic. Then, the new SGW passes UE-IP to the new PGW using the PAA Information Element [2] in a Create Session Request control message. The new PGW is able to establish necessary tunnels and routing rules using UE-IP as per standard procedure.

Data plane redirection. In conjunction with the establishment of tunnels for offloaded traffic, we must establish policies in the SDN substrate to offload the game server specific traffic to the cloud-based S/P-GWs. This requires inspection of the destination address of the content being requested by the UE. The address is the destination address of the original IP packet sent by the UE which in turn gets encapsulated in a GTP header by the eNodeB before it is sent to the SGW. Current Openflow specifications (www.opennetworking.org) do not support such matching of GTP encapsulated “inner” packet header using basic openflow match fields; hence it is not possible to realize the solution using existing openflow-compliant switches. Specialized openflow-extended switches with support for GTP-encapsulated packets have been proposed [10], and will enable an Openflow SDN-substrate solution.

In view of lack of required hardware support in the current standard, we propose to use an “SDN middlebox” as a practical solution. Specifically, we set up a middlebox in the core network and routing rules in the eNodeB and the SGWs, such that all traffic between the eNodeB and the SGW passes through the middlebox, which realizes the offloading policies. Our approach conceptually results in a

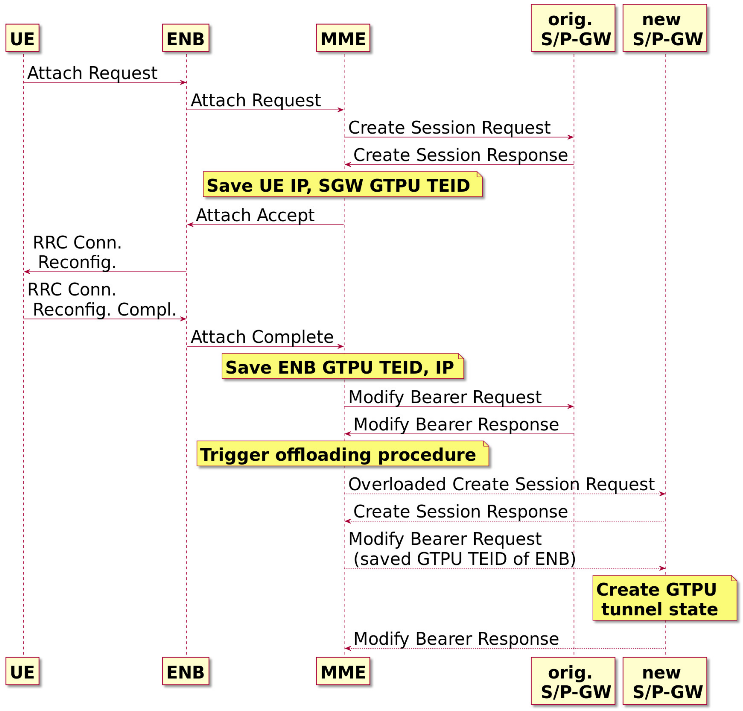

Figure 3: Dynamic offload message flow

transparent "split" of the tunnel between the eNodeB and the original S/P-GW so that regular traffic is still carried to these original gateways, while off-loaded traffic is carried to the new S/P-GW. Thus, creation of data tunnels with proper state in cloud-based S/P-GWs and establishment of offloading policies in the network using the middlebox complete our MOCA realization.

Placement of data plane redirection. We propose that MTSOs (refer Section 2) are suitable places to install mechanisms to realize data plane redirection. First, since all the eNodeB traffic is routed through the MTSOs in cellular networks, having the middlebox in the MTSO will result in minimal overhead. Next, when a user hands over from one eNodeB to another, we need to ensure that the middlebox is in the path of the new eNodeB also. Given that a single MTSO covers a large metropolitan area, and given the typical user mobility pattern [17, 8], we expect the new eNodeB to also route via the same MTSO. In the rare scenario where the traffic shifts from one MTSO to another, the MME needs to trigger installation of the offloading rules in the new MTSO and delete the rules from the old MTSO. This will require the MME to be aware of the mappings between the eNodeBs and the MTSOs. We nominally assume that these mappings are relatively static and that the cellular operators maintain these mapping in some sort of database. We envision that the MME can consult this operator specific database to get these mappings.

4. IMPLEMENTATION

In this section, we first describe the mobile testbed, called AlterNet, we use to realize our offloading solution and then present a proof-of-concept implementation of our traffic offloading architecture.

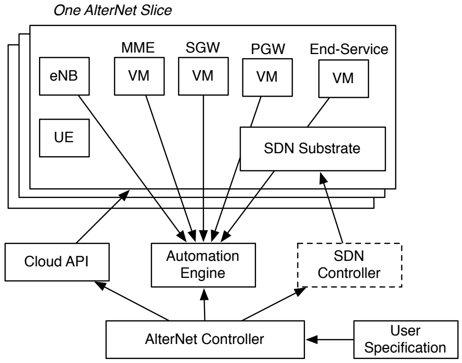

Figure 4: AlterNet Testbed

AlterNet Testbed. Figure 4 depicts the AlterNet testbed we use to realize our prototype offloading solution. AlterNet includes EPC and eNodeB software, both in binary and source code format, acquired from Radisys (www.radisys.com), a company that builds production grade LTE/EPC software. AlterNet uses an LTE femto-cell development board acquired from MindSpeed, operating in a commercial radio band, to realize the RAN (www.mindspeed.com). The femto-cell works with off-the-shelf LTE mobile devices, allowing users to perform end-to-end LTE/EPC experiments. Each cell can support up to 32 concurrent UEs. We use laptops equipped with Huawei 4G dongles to realize our UEs.

AlterNet leverages an Openstack (www.openstack.org) cloud platform to dynamically allocate and deallocate a pool of virtual machines that may emulate different distribution approaches (e.g., wide area vs local), depending on the user specifications. These VMs can be used to run EPC components, like MME/S-GW/P-GW, or any custom code the user wants. AlterNet uses Chef (www.opscode.com/chef) automation engine to manage the configuration of the VMs according to user specifications. A SDN substrate composed of Arista 7150S switches equipped with the latest Intel/Fulcrum Alta chipset provides the network connectivity among the VMs. The SDN controller is responsible for managing network connectivity among the VMs. (Our current realization of AlterNet does not contain a SDN controller. We are investigating the use of SDN controllers to enable AlterNet users to "plug in" customizations of their VMs' network substrate.)

MOCA Prototype Implementation. We developed a proof-of-concept implementation of our mobile content offloading architecture using the AlterNet testbed. Our implementation allows a UE to access content from a specific content provider server through a new virtual S/P-GW that clones the original data tunnel corresponding to the default bearer. We instrument the MME source code to record the required parameters, as described in Section 3.2, during default bearer creation and, send the same to the cloud based SGW using the Overloaded Create Session Request and, the corresponding Modify Bearer Request (Figure 3), respectively. We modify the code of the cloud based SGW so that it can receive the Overloaded Create Session Request from the MME and clone the default bearer state using the passed parameters (Section 3.2). Our cloud based PGW uses the vendor supplied source code in unmodified form. In our implementation, we pro-actively instantiate the cloud based new S/P-GWs along with the other EPC components which run as processes in Ubuntu-based virtual machines in the AlterNet testbed. Because it is not possible to implement our offloading policy using Openflow SDN-capable switches, we use a Linux machine to realize the SDN middlebox approach (Section 3.2) and statically configure routing policies in the network so that all the traffic from the eNodeB and the SGWs (original and cloud-based) are routed through the middlebox. The offloading policy is implemented using Iptables rules, wherein we check for the destination IP address requested by the UE in the "inner" IP datagram within GTP encapsulation. If the inner destination IP address matches that of the cloud-based content server, the middlebox changes the outer destination IP address of the GTP packet from the IP address of the original SGW to that of the new SGW and forwards the packet to the interface that lead to the new SGW. Otherwise, no address translation on the packet is done, and the packet is routed to the original SGW. Similarly, for the return traffic from the new SGW to the eNodeB, we change the outer source IP address of the GTP packet from the IP address of the new SGW to that of the original SGW. Without this address translation, the eNodeB GTP layer would discard downlink packets from the new SGW, since the eNodeB has GTP tunnel states only for the original SGW.

5. DISCUSSION

While we present a proof-of-concept realization of the MOCA architecture, there are many other aspects we need to consider in practice including how to handle user mobility. Several works in literature have shown that daily mobility patterns of users in metropolitan areas are very restricted. Zang et al. [17] showed that 96% of the users in Manhattan and Brisbane visit fewer than 40 cells in a month. A separate study [8] reported that the daily commute distance of users in New York and Los Angeles areas are less than 10 miles in most cases, while 98% of users never commute beyond 36 miles in these areas. Given this restricted mobility pattern of urban users and our offloading use cases, we argue that it is sufficient to extend our solution to only support eNodeB handovers to satisfy the mobility requirement of the user. Mobility that requires handover between core EPC nodes is left for future work. In LTE, all handover procedure eventually involves the MME which inform the SGW about the eNodeB change using a "Modify Bearer Request" so that the SGW can update the corresponding tunnel state. In our case, the MME also needs to a) send a "Modify Bearer Request" to the cloud-based new SGW and, b) install new routing rules in SDN substrate to offload the game server specific traffic from the new eNodeB to the cloud, after which the old offloading rules can be deleted. These changes in routing rules can be performed in a transparent manner as far as the involved eNodeBs and the original SGW are concerned and, requires minor modifications in the MME logic. The cloud based service and the new virtual S/P-GWs will typically cover areas of metropolitan scale, hence they need not be changed as the user moves from one cell tower to another.

Another related aspect is compatibility with existing capabilities. One such example is location management including paging, where the network needs to locate the UE for a new incoming message. In our design, the "default" SGW responsible for regular traffic handles paging, and a cloud-based S/P-GWs exist only for the duration of application-specific services for a given UE.

6. RELATED WORK

Our solution is inspired in part by the cloudlet work, which suggested the use of rapidly deployed virtual machine instances in close proximity to mobile users in a WiFi network, to enable low latency applications to utilize cloud based resources [14].

Current hierarchical routing approaches used by mobile network operators provides suboptimal performance for mobile content providers [5, 16]. In addition, to prevent overburdening of the core network resources due to huge surge of data traffic emerging from new applications, different offloading mechanisms have been proposed in the literature [3, 6, 7]. Research efforts like [3] showed how WiFi can be used to offload significant percentage of cellular traffic locally. Han et al. [7] suggested opportunistic communications using delay tolerant network paradigm to deliver mobile content. However, these mechanisms are limited to delay tolerant traffic only and assume the presence of other forms of network connectivity like WiFi. Our offloading solution does not require any additional network connectivity and works equally well for real time traffic with strict delay requirements (e.g., gaming traffic).

The standards body 3GPP has proposed two mechanisms - Local IP Access (LIPA) and Selected IP Traffic Offload (SIPTO) - for offloading mobile traffic to local network using Home-eNodeBs and to the Internet via local gateways close to the user location respectively. Ma et al. proposed a optimized LIPA-SIPTO solution that perform bearer specific offloading of mobile traffic [12]. However, the proposed solutions require significant fundamental changes in the standard protocols (including changes in the UEs and eNodeBs) and/or use of additional expensive offloading infrastructure like Home-eNodeBs, local GWs etc. In contrast, we propose a conservative offloading solution that requires a few minor changes in existing standard (no change required in the UEs and the eNodeBs) and minimizes infrastructural overhead.

The use of SDN [11] in cellular networks has been suggested to provide flexibility or even redesign of the architecture with mobility as the foundation [15, 4]. In our work, we show how the use of SDN substrate to realize (offloading) policy based routing and existence of in-network cloud infrastructure to host virtual S/P-GWs and services like game server engine provide much needed flexibility to the cellular network architecture.

7. CONCLUSION

In this paper, we have presented MOCA- a lightweight architecture for offloading mobile content that can be realized in a real operator network using cloud infrastructure and SDN capabilities. We also presented a proof-of-concept implementation of our proposed architecture using a mobility testbed. We believe that, our approach presents the first step towards a more comprehensive and general evolution of the mobile network architectures through the use of cloud infrastructure and SDN.

8. REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic. `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf`.

[2] GTPv2-C. `http://www.3gpp.org/ftp/Specs/html-info/29274.htm`.

[3] Balasubramanian, A., Mahajan, R., and Venkataramani, A. Augmenting mobile 3G using WiFi. In *MobiSys* (2010).

[4] Baliga, A., Chen, X., Coskun, B., de los Reyes, G., Lee, S., Mathur, S., and Van der Merwe, J. VPMN: virtual private mobile network towards mobility-as-a-service. In *Proceedings of the second international workshop on Mobile cloud computing and services* (2011).

[5] Dong, W., Ge, Z., and Lee, S. 3G meets the internet: understanding the performance of hierarchical routing in 3G networks. In *Proceedings of the 23rd International Teletraffic Congress* (2011), ITCP, pp. 15–22.

[6] Han, B., Hui, P., Kumar, V. A., Marathe, M. V., Shao, J., and Srinivasan, A. Mobile data offloading through opportunistic communications and social participation. *Mobile Computing, IEEE Transactions on 11*, 5 (2012), 821–834.

[7] Han, B., Hui, P., and Srinivasan, A. Mobile data offloading in metropolitan area networks. *ACM SIGMOBILE Mobile Computing and Communications Review 14*, 4 (2011), 28–30.

[8] Isaacman, S., Becker, R., Cáceres, R., Kobourov, S., Martonosi, M., Rowland, J., and Varshavsky, A. Ranges of human mobility in los angeles and new york. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on* (2011), IEEE, pp. 88–93.

[9] jo Kim, B., and Henry, P. Directions for Future Cellular Mobile Network Architecture. FirstMonday.org, December 2012.

[10] Kempf, J., Johansson, B., Pettersson, S., Luning, H., and Nilsson, T. Moving the mobile evolved packet core to the cloud. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on.*

[11] Li, L. E., Mao, Z. M., and Rexford, J. Toward software-defined cellular networks. Proc. European Workshop on Software Defined Networking, October 2012.

[12] Ma, L., and Li, W. Traffic offload mechanism in epc based on bearer type. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on* (2011), pp. 1–4.

[13] Manweiler, J., Agarwal, S., Zhang, M., Roy Choudhury, R., and Bahl, P. Switchboard: a matchmaking system for multiplayer mobile games. In *Proceedings of the 9th international conference on Mobile systems, applications, and services* (2011), MobiSys '11.

[14] Satyanarayanan, M., Bahl, P., Caceres, R., and Davies, N. The Case for VM-based Cloudlets in Mobile Computing. In *IEEE Pervasive Computing* (November 2009).

[15] Seskar, I., Nagaraja, K., Nelson, S., and Raychaudhuri, D. Mobilityfirst future internet architecture project. In *AINTEC* (2011).

[16] Xu, Q., Huang, J., Wang, Z., Qian, F., Gerber, A., and Mao, Z. M. Cellular data network infrastructure characterization and implication on mobile content placement. In *SIGMETRICS* (2011).

[17] Zang, H., and Bolot, J. C. Mining call and mobility data to improve paging efficiency in cellular networks. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking* (2007), ACM, pp. 123–134.