Dynamic control of real-time communication (RTC) using SDN: A case study of a 5G end-to-end service

Samuel Jero*, Vijay K. Gurbani†, Ray Miller†, Bruce Cilli†, Charles Payette†, and Sameer Sharma†
*Purdue University
Email: sjero@purdue.edu
†Bell Laboratories, Alcatel-Lucent
Email: {vijay.gurbani,ray.miller,bruce.cilli,chuck.payette,sameer.sharma}@alcatel-lucent.com

Abstract—**The next-generation 5G mobile network architecture will support the rapid deployment of new, dynamic network services that are capable of responding to current network conditions and demands. Software-defined Networking (SDN), virtualization technologies, and real-time analytics are the core components that will enable an adaptive and responsive 5G network. We present a case study of a real-time communications (RTC) video service that highlights the manner in which the core components (SDN, virtualization, analytics) allow a flexible and elastic 5G network. Because an end-to-end 5G network does not exist today, we construct one using artifacts from the current 4G/LTE network to host our dynamic network enabled RTC service. We identify three main insights from executing our service that could prove beneficial to the 5G network evolution: need for efficient horizontal control, need to limit identifier proliferation, and the existence of control-plane network functions in service network-function graphs.**

I. INTRODUCTION

Mobile network connectivity has evolved rapidly in the last 15 years. Today's 4G LTE (4^{th} Generation Long Term Evolution) provides better, faster connectivity than ever before, fueling the demand for ever more bandwidth and connectivity. Global mobile data traffic grew 69% in 2014 to reach a volume of 2.5 exabytes per month [1]; the mean monthly usage of data exchanged on US cellular mobile networks increased from 465 MB to 582 MB while the median monthly usage grew from 102 MB to 118 MB [2]. Even data traffic on metropolitan access and aggregation networks is set to increase by 560% by 2017 [3], driven by services like streaming video (720% increase), real-time video calls, data center-based applications (440% increase), and tactile Internet.

It is in this context that the fifth-generation mobile network (5G) is being developed. There are some notable trends driving the move to 5G, two of which are salient for our work: the notion of mobile performance evolving to include quality of experience (QoE) and the need for network adaptability. Mobile performance considerations in 5G will no longer be limited to peak data rates, coverage area, or spectral efficiency; instead, performance will be characterized by factors such as ease of service connectivity and a more user-centric and context-aware experience driven by real-time data analytics. In short, the QoE will be reflective of the flexibility and adaptability of the network to the needs of the user's application. The second trend is being driven by the realization that 5G should be the "network of us": instead of users adapting to network idiosyncrasies, the network should adapt itself to the expectations of the services running on behalf of the user. To accommodate both of these trends, the key enabling technologies are Software Defined Networking (SDN) and real-time analytics. SDN allows for more informed control of the network, as the controller is in the unique position of retaining an authoritative view of the network gathered from the devices under its control. This information includes real-time analytics reported by a device or service.

The introduction of these technologies enables the network to optimize itself to the needs of services; choosing radio access technologies, processing locations, routing, and Quality-of-Service (QoS) levels to meet the demands of current services. It also greatly increases the ease with which new services may be deployed and enables these services to respond dynamically to changing network conditions and demands from users. For instance, a low latency service may request certain expensive network-optimizations only when network latency exceeds some critical threshold. These characteristics will greatly increase the efficiency of network services and their ease of development. However, there is still much uncertainty regarding the types of services that will be deployed, the types of requests these services will make of the network, and the frequency of these requests.

An end-to-end 5G network does not exist today, and indeed some of the access technologies are still under development. However, it seems reasonable to approximate a 5G network using artifacts from deployed 4G/LTE networks controlled through an SDN controller. In this paper, we construct a precursor to a 5G network to understand the dynamics of the service–network intersection, namely, how can the network fulfill expectations by adapting to demands from its services?

Accordingly, the contributions of this paper are:

- A working proof-of-concept of a dynamic, adaptive network that can fulfill the expectations of services. An SDN controller is introduced into the 4G/LTE network; it uses analytics and central control of components to adapt the network to the expectations of a service;
- A number of important lessons learned from this proof-of-concept that we believe need to be applied to the design of the 5G network going forward;

978-1-5090-0223-8/16/$31.00 © 2016 IEEE

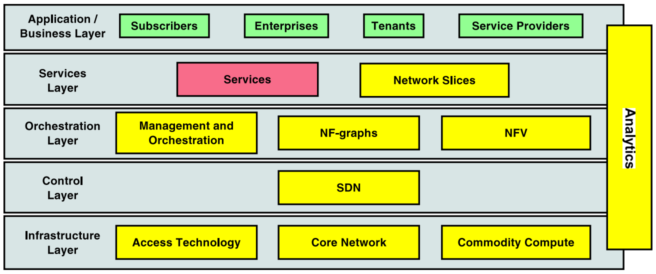


Fig. 1. An expanded view of the 5G architecture

- A new understanding of network function graphs in Network Function Virtualization (NFV) that includes control-plane elements that do not process packets.

The rest of this paper is organized as follows. Section II looks at an abstract layered architecture of 5G, Section III introduces the real-time communication service and describes the evolution of the service from an over-the-top (OTT) service to a dynamic 5G network-enabled service. Section III provides the details of our implementation, including morphing a 4G/LTE network into a precursor of an SDN-controlled 5G network. Section V discusses the lessons we learned during implementation; we put our work in context of existing literature in Section VI and conclude in Section VII.

II. 5G CONCEPTUAL ARCHITECTURE

The 5G system must cater to an increased range of requirements when compared to the current 4G/LTE systems. Data rates will vary from very low (sensor data) to extremely high (4K and beyond video), latencies will range from sub-millisecond latency to applications where it is not a concern, and packet sizes will vary from *tinygrams* to *jumbograms*. All this will occur over multiple radio access technologies. The network will be complex. A conceptual layered architecture for 5G has been specified by the NGMN Alliance [4]; the architecture makes use of SDN and NFV to virtualize functions and capabilities. Our expanded view of the NGMN Alliance architecture is illustrated in Figure 1.

In this view, the NGMN Business Enablement Layer is expanded into a Services Layer, an Orchestration Layer, and a Control Layer. Users and applications at the Business Layer request services. Services are mapped to network slices which are in turn mapped to graphs of virtualized network functions. These virtualized network functions are abstractions of the functions and capabilities provided by the infrastructure layer, orchestrated/managed by the Orchestration Layer, and controlled by the Control Layer. The Analytics Layer collects and processes data from the system at all layers to allow for improved performance and personalization. Within such an architecture, the real-time service described in Section III is an end-to-end service that uses the functions and capabilities of the network discussed here.

III. THE REAL-TIME COMMUNICATION (RTC) SERVICE

To study the dynamics of an end-to-end service with the 5G network, we chose a real-time communication service in web browsers, colloquially known as webRTC. This service is being standardized both by the Internet Engineering Task Force (IETF [5]) and the World Wide Web Consortium (W3C [6]). WebRTC enables real-time communications such as audio, video, and text between users using the browser as a communication medium without the aid of any plug-ins or other components that need to be downloaded. IETF and W3C are jointly defining and standardizing the JavaScript application programming interfaces (API), HTML5 tags, and the communication protocols to setup and manage a reliable communication stream between browsers.

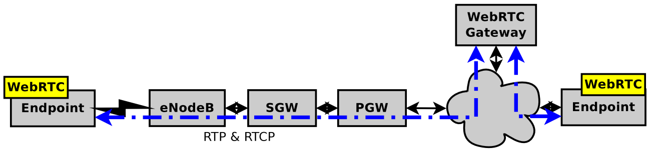


Fig. 2. OTT webRTC over 4G/LTE

We discuss the RTC service from its pure OTT form, where the network provides best-effort delivery of packets, to its evolution as a dynamic network enabled RTC (DNE-RTC) service running in our proof-of-concept 5G network. We demonstrate how the 5G network provides improved and more consistent video quality for video sessions in webRTC.

OTT-RTC: OTT-RTC is provided as a best effort service. The architecture for this service over an 4G/LTE network is shown in Figure 2. Two end devices are used, the left endpoint is a mobile smartphone that has a browser capable of running webRTC. The right endpoint is on the wired network and is similarly configured with a browser that supports webRTC. We use Janus [7] as our webRTC gateway. Janus hosts the JavaScript document that allows the endpoints to make and receive video calls and relays the media (RTP [8]) and media control (RTCP [8]) packets between the endpoints. The leftmost endpoint is configured to be on a cellular network; all communication between it and the core network is facilitated and controlled by a base station (eNodeB). A pair of network elements (a signaling gateway and a packet gateway, respectively) connect the radio network, which includes the left endpoint and eNodeB, to the core network. The left endpoint (a smartphone), the radio access network between it and the eNodeB, and the two intermediary network elements are components of the 4G/LTE network. There are other 4G/LTE network elements which are not shown in Figure 2; they will be introduced later.

A webRTC video session is established by one of the endpoints downloading a JavaScript document from Janus and executing it in a browser. The resulting video communication session is set up between the endpoints as shown in Figure 2. Packets destined to the left endpoint are routed by the core network to eNodeB and sent to the endpoint over the radio access network. Similarly, packets from the left endpoint are transmitted over the radio access network to the eNodeB and then routed towards the destination endpoint.

The biggest disadvantage of OTT-RTC is that QoE will suffer during periods of high traffic in the radio access network. The proportional fair scheduling algorithm of an eNodeB provides equal radio resources to all active best-effort flows; thus an eNodeB with a large number of best-effort clients will amortize the available bandwidth among those clients, leading to poor QoE for latency-sensitive flows (e.g., a real-time video session). We observed, not surprisingly, that webRTC video calls exhibit significant latency in a congested network, a fact confirmed independently by others [9], [10]. Additionally, we observed that webRTC video streams frequently do not compete fairly with each other, with individual stream rates varying by more than a factor of two under similar network and wireless signal conditions; this is especially pronounced when cellular load changes.

Dynamic network enabled RTC (DNE-RTC): DNE-RTC improves upon Network Enabled RTC (NE-RTC), an existing technology developed by Bell Laboratories. Some background in NE-RTC helps to understand DNE-RTC.

NE-RTC is a network service that provides improved and consistent quality of service for webRTC or other real time video calls. NE-RTC addresses the disadvantages of OTT-RTC described in the previous section by leveraging an eNodeB technology called Adaptive Guaranteed Bit Rate (AGBR) [11]. Specifically, NE-RTC utilizes an advanced version of the eNodeB scheduling algorithm from AGBR to calculate the resources allocated to each NE-RTC user. This algorithm is designed to protect the throughput of other users in the cell while providing a more stable throughput for NE-RTC users. It accomplishes this by splitting the radio resources into pools for NE-RTC and normal users. These pools are sized in real time by the algorithm based on the number of NE-RTC users and their signal-to-interference-plus-noise ratios (SINRs) and the number of normal users in the cell and their bit rates. The existing implementation of NE-RTC requires the use of Guaranteed BitRate (GBR) *bearers* for the video flows (a *bearer* is an end-to-end association of a radio endpoint and the packet gateway). These GBR bearers enable the base station to protect the video streams from adverse fluctuations in wireless channel conditions and result in the the video flows being prioritized over other traffic in the network, protecting them from any network congestion.

We refer to NE-RTCs traffic allocation scheme as the Target Bitrate Calculator Function (TBRCF) because it calculates the maximum throughput a flow should use in the cellular network. The TBRCF calculates a throughput called the Target BitRate (TBR), and for each flow the TBR is exported from the eNodeB to an external application running on the Janus webRTC gateway. This application combines the TBR with other data, like endpoint type or maximum bitrate, to establish an optimum bitrate that is ultimately used to set a bitrate cap on the video encoder via the appropriate RTCP messages. Each device receiving these RTCP messages adjusts its video codec-related parameters, if necessary, to send the highest quality video possible without exceeding its available bandwidth.

NE-RTC is still work in progress; however, initial testing indicates that it is highly effective at providing improved video call performance by providing much more even, predictable throughput for video flows. While NE-RTC is effective at improving the QoE of an interactive video session, it is not sufficiently responsive to network conditions. In particular, NE-RTC requires GBR bearers in the core network and the computation of the TBR for each video flow regardless of current network and video call conditions. Thus, even video flows that could be handled by best-effort delivery are assigned GBR bearers, a limited resource to begin with. Further, continuously running TBRCF adds additional computational load to the eNodeB, irrespective of current network conditions. Finally, there is the additional overhead of sending the TBR messages to the video encoder on the endpoint every few seconds. This non-trivial message overhead could be avoided if the TBR is higher than the default webRTC encoding bitrate.

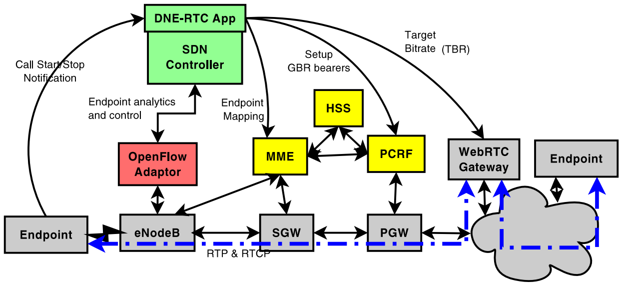


Fig. 3. Proof-of-Concept DNE-RTC implementation

For the reasons above, it is desirable to enable NE-RTC only when it will actually result in improved quality for the video calls currently in progress. This dynamic component is important; it is what characterizes a network that is adaptable [4], [12], [13]. We designed DNE-RTC as a service that works in close concert with the network, a service where the network controls the aspects of the service that impact the workings of the network itself. This level of control is made possible by an SDN controller, which will be a key element of a 5G network. It will responsible for controlling other network elements in the core and the radio network, including the eNodeB (or its 5G equivalent). The SDN controller will host a number of applications, one of which will be DNE-RTC. The DNE-RTC application will examine metrics about each of the wireless devices in the network to decide whether to enable NE-RTC. NE-RTC will only be enabled if it is likely to provide noticeably improved QoE.

IV. IMPLEMENTATION OF DNE-RTC ON A PROOF-OF-CONCEPT 5G NETWORK

While telecommunication equipment vendors and service providers have built selected components of the 5G network in private laboratories, to the best of our knowledge, there are no public operational 5G testbeds yet. Nationally funded research initiatives like the U.S. Global Environment for Networking Innovation (GENI) and the E.U.'s Future Internet Research and Experimentation (FIRE) are gaining prevalence and scale, but do not address 5G yet. Therefore, to study the dynamics of DNE-RTC on a programmable and adaptive network, we first introduce the concept of centralized control by adding an SDN controller to an existing 4G/LTE network as shown in Figure 3. We used ONOS (http://onosproject.org/) because of its modularity, high level northbound API, and popularity in industry. This proof-of-concept gave us a valuable insight into the dynamic control mechanism that we believe will become an intrinsic part of the eventual 5G network.

The 4G/LTE packet network consists of a number of components, each a dedicated hardware device. The eNodeB, or radio base station, is responsible for connecting to the mobile endpoints and facilitating radio resources so that the endpoints can communicate with the rest of the Internet. The Serving Gateway (SGW) establishes bearers based on messages from the Mobility Management Entity (MME); it also routes packets between the radio network, the core network, and the Internet by forwarding the packets to/from the Packet Gateway (PGW). The MME is the key control node for the 4G/LTE packet network. It pages (finds) mobile endpoints and is involved in the bearer activation and deactivation process. The PGW provides connectivity for the mobile endpoint to the Internet by acting as the entry and exit point for traffic for the endpoint. Working together with the Policy and Charging Function (PCRF), the PGW also performs policy enforcement and packet filtering for each user. The Home Subscriber Server (HSS) is a central database that contains information on the services to which the user has a subscription.

To this mix, we introduce the ONOS SDN controller, an eNodeB OpenFlow adaptor, and a DNE-RTC SDN application, which runs as a module within ONOS and is responsible for enabling or disabling NE-RTC for each endpoint based on the application analytics arriving at the DNE-RTC application from the eNodeB. Note that our proof-of-concept implementation uses only a single SDN controller for the entire network. We wrote an eNodeB OpenFlow adaptor that serves as a canonical protocol translator between OpenFlow and the custom command-line interface (CLI) exposed by the eNodeB. With this adaptor in place, the eNodeB simply becomes a device under the control of the SDN controller.

When an endpoint starts a webRTC session, it notifies the DNE-RTC application through a RESTful API that we designed. The session initially begins with best-effort delivery. The DNE-RTC application begins to track the throughput of the endpoint using an exponentially weighted moving average, with the smoothing factor, $\alpha = 0.7$. We started aggressively with $\alpha = 0.8$ as this value is used by TCP's RTT estimator [14], which faces the similar issue of adapting quickly network changes while ignoring transients. We experimentally found that a value of $\alpha = 0.7$ worked better, but did not investigate α extensively. While tracking the throughput, if the DNE-RTC application determines that the average throughput has dropped below a low threshold (500 kbps), it will enable the NE-RTC service. Once enabled, a second, higher threshold (2 Mbps) is used to disable NE-RTC. This use of hysteresis prevents undue oscillation. Like α, the low and high thresholds were determined experimentally. We also found it necessary to prevent NE-RTC from being enabled during the first 15 seconds of a webRTC session, as bitrate gradually ramps up.

The DNE-RTC application receives analytics related to the throughput of the mobile endpoint via the eNodeB OpenFlow adaptor. The adaptor connects to the eNodeB via *ssh(1)* and enables the exporting of wireless endpoint throughput information. The eNodeB exports this information (about 1 Kbyte) every second over a UDP socket. The adaptor receives, parses, and exposes this throughput information as a statistic on a virtual OpenFlow "port", which is queried by the SDN controller every 2 seconds. An OpenFlow `SET_CONFIG` message is used to configure NE-RTC's base station TBRCF component by triggering CLI commands, as described next.

Enabling and disabling NE-RTC is a fairly involved process. First, GBR bearers are added to the video flow that needs to be improved. This is accomplished via a request to the PCRF through a RESTful API. Next, the TBRCF is enabled on the eNodeB by issuing OpenFlow configuration commands that are translated to the appropriate CLI commands by the eNodeB OpenFlow adaptor. At this point, NE-RTC is enabled and the TBRCF is calculating the target bitrate for the video flow. This TBR is collected by the eNodeB OpenFlow adaptor and exposed to the SDN controller as an additional statistic. The DNE-RTC SDN application sends this TBR to the webRTC gateway, which in turn propagates it (over RTCP) to the endpoint encoding the video. To disable NE-RTC, the GBR bearer is removed from the video flow via a request to the PCRF, and then the TBRCF is disabled on the eNodeB.

V. LESSONS LEARNED

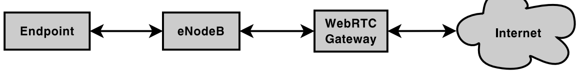
Our work on the DNE-RTC network service in a 5G proof-of-concept testbed has identified a number of important lessons that we believe should be applied to the design of 5G in the future. In this section we examine each of these lessons.

1. Horizontal SDN controller interactions are common: Our experience with the DNE-RTC network service suggests that horizontal interactions between SDN controllers for different network domains will be common and, indeed, frequent. Today's network is separated into a number of domains both geographically and logically: the radio access network, the core network, and the application data center [15].

We believe that geographic and administrative constraints will continue to dictate separate domains controlled by separate SDN controllers in the 5G network. In spite of this, our experience with the DNE-RTC service suggests a need to communicate across these domains: first, there is a need to communicate with the eNodeB (in the radio access network) to receive metrics and send commands; second, communications with the core network is required to add or remove GBR bearers; and finally, there is incentive to communicate with servers residing in data center networks (in our case, the webRTC gateway to control the target bitrate). When the SDN controller begins to handle routing and QoS, as in the 5G network, this cross-domain communication will only increase.

In our proof-of-concept, a single SDN controller communicated directly with components in different network domains. However, a cleaner, more practical, design calls for horizontal

(a) Best Effort Network Function Graph:



(b) Dynamic NE-RTC Network Function Graph:

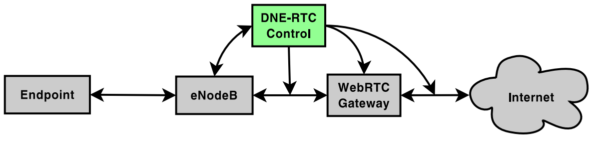


Fig. 4. Network Function Graphs

communication between separate controllers residing in, and managing components in, different domains. Any controller coordination method chosen will need to be efficient at horizontally communicating between controllers. This tends to suggest that a federated controller model [16], [17], where the SDN controllers directly communicate with each other, may be more desirable than a hierarchical model [18]. Further work is needed to determine under what situations each of these models is optimal.

2. Identifier proliferation complicates service development: A second lesson we quickly learned was that identifier proliferation in the current 4G/LTE network dramatically complicates service deployment. In particular, our implementation had to understand and map among no less than four different identifiers for a wireless device (Table I). This suggests a need for a central naming service in 5G that translates between the appropriate network-level identifiers.

SDN will help significantly here because the centralization of network control means that all of these identifiers will be available in a single logical entity for easy mapping. However, careful thought still needs to be given to the design of service and communication APIs so that the number of identifiers that services need to handle is minimized and those that are exposed can be easily mapped.

3. Network Function Graphs can contain control-plane elements: Network-function graphs — also known as forwarding chains, forwarding graphs, and service chains [19], [20] — define network services by indicating the interconnections between the different network elements used to process network flows according to the requirements of the service. These network-function graphs operate at a higher level of abstraction than SDN, providing the specialized, service-specific processing unique to a particular service (e.g. WebRTC video). SDN would usually be used to construct these network-function graphs, but tunneling or VLAN technologies could also be used. The European Telecommunication Standards Institute (ETSI) and Internet Research Task Force (IRTF), among other organizations, are in the process of defining a network-function graph representation of network services for use with network function virtualization [19], [20].

TABLE I
IDENTIFIERS AND COMPONENTS USING THEM

Identifier	Components
Endpoint S1AP ID	eNodeB, MME
IP address	MME, PCRF, Endpoints, webRTC gateway
IMSI	MME, PCRF
SIP URI	Endpoints, webRTC gateway

Traditionally, the network elements in network-function graphs have been data-plane devices like firewalls, network address translators, deep packet inspection devices, and load balancers which are configured once on service instantiation. Through our work with DNE-RTC, we realized that these data-plane elements may need to be configured dynamically by the service. This requires expanding our concept of network-function graphs to include the service-specific control-plane elements responsible for this configuration. Consider Figure 4(a), which represents the network-function graph for a generic, best-effort real-time video communication service (OTT-RTC), and Figure 4(b), showing the equivalent DNE-RTC service network-function graph. In both of these figures, the data-plane elements remain the same; the interesting contribution from our work is the new service-specific control-plane element added by DNE-RTC that dynamically configures the different components in the eNodeB (radio access network), the core network, and the webRTC gateway (data center network). Unlike the SDN control-plane, which controls generic packet processing for the movement of data across the network, this DNE-RTC control-plane dynamically controls configuration narrowly specific to the service being provided, like the enablement of the TBRCF and sending the TBR to the video encoder via the WebRTC gateway.

We believe that such control-plane elements will be a key component of other services as well and, therefore, argue that the concept of a network-function graph must be expanded to include control-plane elements that coordinate the data-plane but do not actually process packets. We are not aware of other work in the literature that points out this limitation of the current network-function graph model.

VI. RELATED WORK

A number of works have proposed architectures for an SDN-based 5G network [21], [22], [23], [24], [25]. These works suggest separating the control plane and data plane of the existing 4G/LTE network and moving the control plane to an SDN Controller. This brings a number of benefits, from simplifying mobility [21], [23] to improved resiliency [22], and the ability to create virtual networks [24]. Unfortunately, these works provide no implementations or simulations of their proposals. To ease the testing of such proposals, a few 5G testbeds are under development. The Fraunhofer FOKUS center for Next Generation Network Infrastructures is developing a 5G simulator called Open5Gcore under a propriety license [26] while Surrey University, in collaboration with Huawei and the 5GIC project, will soon have a 5G testbed [27]. The EU's FIRE initiative is also developing such a testbed [28].

A few works have looked at how to adapt SDN protocols for mobile networks. MobileFlow [29] and MobileVisor [24] both propose OpenFlow variations that are extended with support for GTP tunnels to better support mobile networks while SDMN [23] proposes removing GTP tunnels entirely and leveraging the per-flow routing ability of the SDN controller (GTP, or GPRS Tunneling Protocol is a group of IP-based protocols that facilitate packet-based cellular communications). Lai et al. [30] proposed an algorithm for SDN-assisted HTTP adaptive streaming system for 5G. This system leverages the SDN controller's global network view to obtain predictions of future bandwidth to make better decisions about the video quality to request. Simple simulations are used examine the proposed algorithm and show that it is effective at adapting to changes in available bandwidth. In contrast, our work develops a prototype implementation of a service to improve real-time video quality under radio network congestion and identifies several important takeaways for 5G from the experience.

VII. CONCLUSION

The goal of the 5G network is not only to support even more 4G/LTE-like broadband services, but also enable next generation applications and use cases. End users want personalized service delivery with higher capacity and better QoE, network operators want to effortlessly transform their end-to-end infrastructure to meet current needs and offer novel value-added services, application and content providers seek insights into the end-to-end network's capabilities and real-time condition to adapt and provide better user experiences. All of this creates a large diversity of service needs and complexities and a need for personalization in the network. To overcome these obstacles, the next generation network must be flexible, elastic, and dynamically adaptive to individual needs.

We have developed a proof-of-concept implementation of a 5G network by introducing key components — SDN and analytics — to study the dynamics of a service (DNE-RTC) in this new environment. Our work has resulted in a number of important lessons for 5G that provide insight and guidance as 5G is further standardized, architected, and implemented.

ACKNOWLEDGMENT

The authors wish to thank Cristina Nita-Rotaru, Samuel Jero's advisor, for her support of this project, and Peter Andrews for his help with the webRTC gateway.

REFERENCES

[1] Cisco White Paper, "Cisco visual networking index: Global mobile data traffic forecast update, 2014-2019," 2015. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html

[2] Sandvine White Paper, "Global internet phenomena report 2H 2014," 2015. [Online]. Available: https://www.sandvine.com/downloads/general/global-internet-phenomena/2014/2h-2014-global-internet-phenomena-report.pdf

[3] Bell Laboratories, Alcatel-Lucent White Paper, "Metro network traffic growth: An architecture impact study," 2013. [Online]. Available: http://resources.alcatel-lucent.com/asset/171568

[4] NGMN Alliance, "5G white paper," NGMN Alliance, Tech. Rep., 2015.

[5] C. Holmberg, S. Hakansson, and G. Eriksson, "Web real-time communication use cases and requirements," RFC 7478 (Informational), Internet Engineering Task Force, Mar. 2015.

[6] A. Bergkvist, D. Burnett, C. Jennings, and A. Narayanan, "WebRTC 1.0: Real-time communication between browsers," 2015. [Online]. Available: http://www.w3.org/TR/2015/WD-webrtc-20150210/

[7] A. Amirante, T. Castaldi, L. Miniero, and S. P. Romano, "Janus: A general purpose webrtc gateway," in *Proceedings of the Conference on Principles, Systems and Applications of IP Telecommunications*, ser. IPTComm '14. ACM, 2014, pp. 7:1–7:8.

[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 3550 (Standard), Internet Engineering Task Force, Jul. 2003.

[9] N. Hermanns, L. Hamm, and Z. Sarker, "A framework and evaluation of rate adaptive video telephony in 4G LTE," in *World Telecommunications Congress (WTC)*, 2014.

[10] F. Fund, C. Wang, Y. Liu, T. Korakis, M. Zink, and S. S. Panwar, "Performance of DASH and WebRTC video services for mobile users," *20th International Packet Video Workshop (PV)*, 2013.

[11] D. De Vleeschauwer, H. Viswanathan, A. Beck, S. Benno, G. Li, and R. Miller, "Optimization of HTTP adaptive streaming over mobile cellular networks," in *IEEE INFOCOM*, April 2013, pp. 898–997.

[12] P. Kwadwo, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Communications Magazine*, no. November, pp. 65–75, 2014.

[13] A. Hakiri and P. Berthou, "Leveraging SDN for the 5G networks: Trends, prospects and challenges," *Computing Research Repository (CoRR)*, vol. abs/1506.02876, 2015.

[14] J. Postel, "Transmission control protocol," RFC 793 (Standard), 1981.

[15] M. Olsson, S. Rommer, C. Mulligan, S. Sultana, and L. Frid, *SAE and the Evolved Packet Core: Driving the mobile broadband revolution*. Academic Press, 2009.

[16] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in *Proceedings of the internet network management conference on research on enterprise networking*, 2010.

[17] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed Multi-domain SDN Controllers," in *IEEE Network Operations and Management Symposium (NOMS)*, 2014.

[18] S. H. Yeganeh and Y. Ganjali, "Kandoo: a framework for efficient and scalable offloading of control applications," in *HotSDN*, 2012, pp. 19–24.

[19] ETSI, "Network functions virtualisation (NFV); virtual network functions architecture," ETSI, Tech. Rep. GS NFV-SWA 001 - V1.1.1, 2014.

[20] S. Lee, S. Pack, M.-K. Shin, E. Paik, and R. Browne, "Resource management in service chaining," Internet-Draft: draft-irtf-nfgrg-resource-management-service-chain-01, 2015.

[21] T. Mahmoodi and S. Seetharaman, "On using a SDN-based control plane in 5G mobile networks," in *Wireless World Research Forum*, vol. 32, 2014.

[22] S. B. H. Said, M. R. Sama, K. Guillouard, L. Suciu, G. Simon, X. Lagrange, and J. M. Bonnin, "New control plane in 3GPP LTE/EPC architecture for on-demand connectivity service," in *IEEE International Conference on Cloud Networking (CloudNet)*, 2013, pp. 205–209.

[23] J. Costa-Requena, "SDN integration in LTE mobile backhaul networks," *The International Conference on Information Networking (ICOIN)*, pp. 264–269, 2014.

[24] V. G. Nguyen and Y. H. Kim, "Slicing the next mobile packet core network," *11th International Symposium on Wireless Communications Systems (ISWCS)*, pp. 901–904, 2014.

[25] R. Guerzoni, R. Trivisonno, and D. Soldani, "SDN-based architecture and procedures for 5G networks," in *International Conference on 5G for Ubiquitous Connectivity (5GU)*, 2014, pp. 209–214.

[26] Fraunhofer FOKUS, "Open5Gcore - the next mobile core network testing platform." [Online]. Available: http://www.open5gcore.net/Open5GCore/index.html

[27] K. Dyer, "5GIC will have first 5G test bed live by april 2015," 2015. [Online]. Available: http://the-mobile-network.com/2014/11/5gic-will-have-test-bed-by-april-2015/

[28] H. Schaffers, "FIRE portfolio: Current status and development towards the future," in *Workshop on 5G testbeds and hands-on experimental research*, July 2015.

[29] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 44–53, 2013.

[30] C.-f. Lai, R.-h. Hwang, H.-c. Chao, M. M. Hassan, and A. Alamri, "A buffer-aware HTTP live streaming approach for SDN-enabled 5G wireless networks," *IEEE Network*, no. February, pp. 49–55, 2015.