CellSDN: Software-Defined Cellular Networks

Li Erran Li
Bell Labs
erranlli@research.bell-labs.com

Z. Morley Mao
University of Michigan
zmao@umich.edu

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

ABSTRACT

Existing cellular networks suffer from inflexible and expensive equipment, complex control-plane protocols, and vendor-specific configuration interfaces. In this paper, we argue that software defined networking (SDN) can simplify the design and management of cellular data networks, while enabling new services. However, supporting many subscribers, frequent mobility, fine-grained measurement and control, and real-time adaptation introduces new scalability challenges that future SDN architectures should address. As a first step, we present a software-defined cellular network architecture that (i) allows controller applications to express policies based on the attributes of subscribers, rather than network addresses and locations, (ii) enables real-time, fine-grained control via a local agent on each switch, and (iii) extends switches to support features like deep packet inspection and header compression to meet the needs of cellular data services, (iv) supports flexible "slicing" of network resources based on the attributes of subscribers, rather than the packet header fields, and flexible "slicing" of base stations and radio resources by having the controller to handle radio resource management, admission control and mobility in each slice.

1. INTRODUCTION

The growing popularity of smart phones and tablet computers places an increasing strain on cellular networks. Yet, despite tremendous innovation in mobile applications, the cellular network infrastructure is remarkably brittle. Software defined networking (SDN) can simplify network management, while enabling new services. However, supporting many subscribers, frequent mobility, fine-grained measurement and control, and real-time adaptation introduces scalability challenges that future SDN architectures should address.

1.1 Today's LTE Cellular Data Networks

Long Term Evolution (LTE) cellular networks connect base stations (eNodeB) to the Internet using IP networking equipment, as shown in Figure 1. The user equipment (UE) connects to a base station, which directs traffic through a serving gateway (S-GW) over a GPRS Tunneling Protocol (GTP) tunnel. The S-GW serves as a local mobility anchor that enables seamless communication when the user moves from one base station to another. The S-GW must handle frequent changes in a user's location, and store a large amount of state since users retain their IP addresses when they move. The S-GW tunnels traffic to the packet data network gateway (P-GW). The P-GW enforces quality-of-service policies and monitors traffic to perform billing. The P-GW also connects to the Internet and other cellular data networks, and acts as a firewall that blocks unwanted traffic. The policies at the P-GW can be very fine-grained, based on whether the user is roaming, properties of the user equipment, usage caps in the service contract, parental controls, and so on.

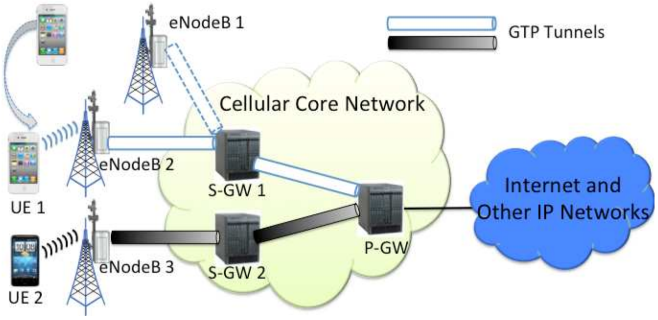

Figure 1: LTE data plane

Besides data-plane functionalities, the base stations, serving gateways, and packet gateways also participate in several control-plane protocols, as illustrated in Figure 2. In coordination with the Mobility Management Entity (MME), they perform hop-by-hop signaling to handle session setup, tear-down, and reconfiguration, as well as mobility e.g., location update, paging, and handoff. For example, in response to a UE's request for dedicated session setup (e.g., for VoIP call), the P-GW sends QoS and other session information (e.g., the TCP/IP 5-tuple) to the S-GW. The S-GW in turn forwards the information to the MME. The MME then asks the base station to allocate radio resources and establish the connection to the UE. During handoff of a UE, the source base station sends the handoff request to the target base station. After receiving an acknowledgement, the source base station transfers the UE state (e.g., buffered packets) to the target base station. The target base station also informs the MME that the UE has changed cells, and the previous base station to re-

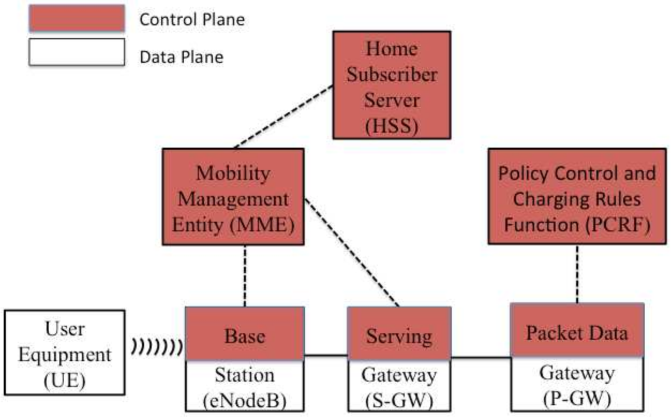

Figure 2: Simplified LTE network architecture

lease resources (e.g., remove the GTP tunnel).

The S-GW and P-GW are also involved in routing, running protocols such as OSPF. The Policy Control and Charging Function (PCRF) manages flow-based charging in the P-GW. The PCRF also provides the QoS authorization (QoS class identifier and bit rates) that decides how to treat each traffic flow, based on the user's subscription profile. QoS policies can be dynamic, e.g. based on time of day. This must be enforced at the P-GW. The Home Subscriber Server (HSS) contains subscription information for each user, such as the QoS profile, any access restrictions for roaming, and the associated MME. In times of cell congestion, a base station reduces the max rate allowed for subscribers according to their profiles, in coordination with the P-GW.

Today's cellular network architectures have several major limitations. Centralizing monitoring, access control, and quality-of-service functionality at the packet gateway introduces scalability challenges. This makes the equipment very expensive (e.g., more than 6 million dollars for a Cisco packet gateway). Centralizing data-plane functions at the cellular-Internet boundary forces all traffic through the P-GW, including traffic between users on the same cellular network, making it difficult to host popular content *inside* the cellular network. In addition, the network equipment has vendor-specific configuration interfaces, and communicate through complex control-plane protocols, with a large and growing number of tunable parameters (e.g., several thousand parameters for base stations). As such, carriers have (at best) indirect control over the operation of their networks, with little ability to create innovative services.

1.2 SDN for Cellular Data Networks

Cellular data networks are ripe for the introduction of Software Defined Networking (SDN), where the network equipment performs basic packet-processing functions at the behest of applications running on a logically-centralized controller. In the next section, we discuss how SDN could give cellular operators greater control over their equipment simplify network management and introduce value-added services. SDN can enable carriers to distribute data-plane rules over multiple, cheaper network switches, reducing the scalability pressure on the packet gateway and enabling flexible handling of traffic that stays within the cellular network.

Supporting real-time updates to many fine-grained packet-handling rules raises significant scalability challenges. Frequent user mobility can require forwarding state at the level of individual subscribers, and the state must change quickly to avoid service disruptions. To detect when subscribers exceed their usage caps, the switches must perform fine-grain monitoring of traffic volumes. The network must adapt quickly to the measurement data to adapt QoS policies, or transcode content to offer good service during times of congestion. To address these challenges, we propose four main extensions to existing SDN architectures:

Flexible policies on subscriber attributes: Many controller applications need to apply policy based on the properties of cellular subscribers, including the network provider, device type, subscriber type, and recent usage. Our architecture automatically translates policies based on subscriber attributes to packet-processing rules based on IP addresses and network locations.

Scalability through local switch agents: The need for fast and frequent updates to a large amount of data-plane state would put tremendous pressure on a central controller. Instead, we believe the switches should run software agents that perform simple local actions (such as polling traffic counters and comparing against thresholds), at the behest of the controller.

Flexible switch patterns and actions: Today's OpenFlow API already supports flexible packet classification and actions. That said, cellular networks would benefit from support for deep-packet inspection, header compression, and message-based control protocols. New APIs customized to cellular base stations could enable flexible adaptation of the air interface, and sharing across multiple carriers through virtualization.

Network virtualization on subscriber attributes: Our architecture supports "slicing" the network based on the attributes of subscribers, rather than the packet header fields. We virtualize the base station resources, and have the controller handle radio resource management, and admission control and mobility in each slice.

2. CELLULAR SDN APPLICATIONS

In this section, we identify several major challenges in cellular data networks, and how Software Defined Networking can enable better solutions.

2.1 Directing Traffic Through Middleboxes

Cellular network operators offer many fine-grained services implemented in network appliances, or middleboxes. In a dynamic environment, cellular provides need to adapt video quality in real time based cell tower congestion, device type, and the subscriber's service plan [1]. To improve security for enterprise customers, providers direct traffic through intrusion prevention systems (IPS). Certain legacy devices need in-network support for echo cancellation for VoIP calls. However, today's cellular carriers do not have fine-grained control over routing, forcing them to either direct excess traffic through unnecessary middleboxes or manage an unwieldy set of tunnels.

SDN provides fine-grained packet classifier and flexible routing, which can easily direct a chosen subset of traffic through a set of middleboxes. As a result, middleboxes will handle much less traffic, and be made much cheaper. In addition, with support for deep-packet inspection, SDN switches could support some middlebox functionality directly, reducing the number of extra devices in the network.

2.2 Monitoring for Network Control & Billing

Due to frequent user mobility, and changing channel conditions, cellular networks are subject to rapid changes in traffic demands, and frequent signaling messages to migrate connections. Real-time traffic monitoring is crucial for triggering fast adaptation. This includes load balancing data traffic from base stations to different S-GW, and from different S-GW to different P-GW, load balancing control traffic from base station and S-GW to different MME. Real time monitoring also enables rapid per application content adaptation (e.g., video conferencing, or streaming from Netflix) to meet per subscriber QoS. Existing traffic-monitoring solutions [2] require additional equipment that captures every packet at every interface of a S-GW, and provides a summary to a backend SQL server every few minutes. These measurements provide no real-time visibility into cellular network devices such as the eNodeB, S-GW, and MME.

The packet-handling rules in SDN switches include byte and packet counters. By adjusting these rules over time, the cellular provider can efficiently monitor traffic at different levels of granularity to drive real-time control loops on the SDN controller. In addition, associating packet classifiers with traffic counters is useful to drive billing decisions and determine whether a subscriber has reached a usage cap. The recent interest in allowing content providers to cover usage charges for mobile users will put even more pressure on cellular providers to collect fine-grained measurements.

2.3 Seamless Subscriber Mobility

Cellular networks must respond quickly to subscriber mobility to avoid disruptions in service. Yet, today's cellular provider do not have direct control over routing, or common protocols for controlling forwarding across different cellular technologies (e.g., 3G, LTE, WiMax, and WiFI. As a result, handoff across technologies involves complex procedures that lead to longer delays and higher packet loss rates.

SDN would provide a common control protocol (e.g., OpenFlow) that works across different cellular technologies, making mobility management much easier. In addition, rather than performing hop-by-hop signaling to create a new session, the controller can push new forwarding rules to multiple switches at the same time for a lower set-up delay.

2.4 QoS and Access Control Policies

In today's networks, the packet gateway is the central point for fine-grained policy enforcement and charging based on the subscriber profile, application, and usage. The P-GW classifies packets based on the 5-tuple of the TCP/IP header and either drops packets (if they violate a firewall policy) or map them into QoS classes. These QoS classes further map into the Diff-Serve Code Points (DSCP) when the packets traverses IP networks en route to the base station. At the base station, only simple policies such as a maximum rate are enforced. This leads to scalability problems at the P-GW, and missed opportunities to optimize the use of bandwidth inside the cellular network.

SDN would enable the distributed enforcement of QoS and firewall policies based on a network-wide view. Distributed enforcement is especially important for handling any traffic that stays within the cellular network. An application running on the controller can spread access-control rules over multiple switches, and manage the scheduling of traffic by QoS classes across multiple hops in the network.

2.5 Network Virtualization

Today's cellular networks have relatively limited support for virtualization. LTE can isolate different enterprise customers' traffic into virtual private networks using traditional BGP/MPLS VPN technologies. However, LTE does not allow different carriers to share the infrastructure to offer a complete virtual LTE network to their customers. Virtual operators may want to innovate in mobility management, and policy and charging, without investing the substantial resources necessary to build and manage a wireless network. For example, content providers like Akamai could leverage a virtual infrastructure to better deliver content to mobile users.

Virtualization would also be useful to provide isolation and separate control for different classes of traffic. For example, a carrier way want to carry traffic for roaming subscribers on different virtual network from its own customer, for security reasons.

SDN makes it relatively easy to support network virtualization by partitioning the "flow space" of packet headers. Different controller applications can manage rules acting on each portion of flow space, enabling customized control while ensuring isolation [3]. Virtualizing the cellular network requires virtualizing the base stations [4, 5], by slicing resources at the physical layer (physical channels), link layer (scheduling), or network layer (traffic shaping).

2.6 Remote Control of the Base Station

In LTE, base stations are involved in distributed control protocols to manage handoff, interference, etc. Base stations also handle admission control, radio resource allocation. Radio resource is an inherently shared resource among base stations. The lack of central control makes it difficult to optimize radio access related tasks. We decouple the control plane from the radio hardware. Radio hardware exposes a well-defined API which can be controlled by the control plane. Radio Resource Management Module (RRM) on top of a cellular operating system is much more flexible and easier to innovate in admission control, radio resource allocation, interference management. For example, RRM can redirect a UE to a nearby lightly loaded base station or increase the transmission power of a congested base station. If the base station has multiple antennas, the control plane can decide whether the antennas should be used for boosting signals (diversity combining to boost signals for delay sensitive applications) or for spatial multiplexing (multiple parallel transmissions). Although in 3G, base stations are controlled by a central entity, the radio network controller (RNC), RNC couples control plane and data plane, and handles much fine-grained tasks like packet scheduling. In contrast, we decouple control plane from data plane. In addition, our RRM module does not handle packet scheduling.

3. CELLULAR SDN ARCHITECTURE

Cellular networks need an SDN architecture that offers fine-grain, real-time control without sacrificing scalability. We propose five main extensions to SDN, leading to the architecture in Figure 3. First, controller applications should be able to express policy in terms of subscriber attributes, rather than IP addresses or physical locations, as captured in a *subscriber information base*. Second, to improve scalability, each switch should run a local *cell agent* that performs simple actions (such as polling traffic counters and comparing against a threshold), at the behest of the controller. Third, switches should support more flexible packet classification based on deep packet inspection, and additional actions such as header compression. Fourth, we enable semantic space slicing of the network resources (a slice in the semantic space is the set of packets whose subscriber attributes satisfy the same predicates). We also enable flexible slicing of base station resources by taking the control out of base stations.

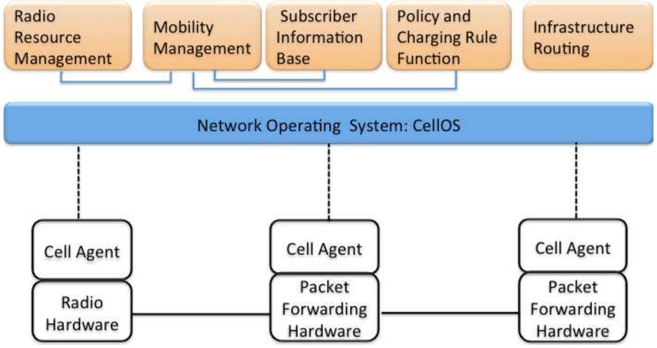

Figure 3: CellSDN architecture

3.1 Flexible Policies on Subscriber Attributes

The SDN controller consists of a Network Operating System (NOS) running a collection of application modules, such as as radio resource management, mobility management, and routing, as shown in Figure 3. The handling of individual packets often depends on multiple modules. For example, the flow of traffic through the network depends on the subscriber's location (determined by the mobility manager) and the paths between pairs of network elements (determined by infrastructure routing), and traffic monitoring and packet scheduling depends on the policy and charging rule function. As such, the NOS should support *composition* to combine the results of multiple modules into a single set of packet-handling rules in each switch, as in the Frenetic language and run-time system [6].

Many of the controller application modules need to apply policy based on the properties of cellular subscribers, including the cellular network provider (e.g., whether the user is roaming or not), device type (e.g., whether the user has a legacy phone that requires echo cancellation), subscriber type (e.g., usage cap, parental controls, etc.), and recent usage (e.g., whether the user is near exceeding the usage cap). Yet, the switches match packets and perform actions based on packet header fields, based on ephemeral identifiers such as a subscriber's current IP address and location. The controller should offer effective ways to bridge the gap.

Our architecture includes a Subscriber Information Base (SIB) that stores and maintains subscriber information, including relatively static subscriber attributes as well as dynamic data like the user's IP address, location, and total traffic consumption. Policies expressed in terms of subscriber attributes are translated into switch rules that match on packet headers. Similarly, the SIB translates network measurements (such as traffic counters) to the appropriate (sets of) subscribers, to allow the application modules to focus on subscribers and their attributes rather than ephemeral network identifiers. This is similar to the way the Ethane system [7] supports access-control policies based on named principals, rather than IP addresses and network locations, though cellular networks must support a much wider range of subscriber attributes and control actions.

3.2 Scalability Through Local Switch Agents

Cellular data networks face significant scalability challenges, in terms of the number of subscribers, frequent changes in user location, fine-grain access-control and quality-of-service policies, and real-time adaptation to network conditions. For example, the switches may need to direct a video stream through a transcoding proxy if the network becomes congestion, or give certain traffic lower priority if the user exceeds his usage cap. These measurement and control functions could easily overwhelm a logically-centralized controller. In addition, the controller may not be able to respond as quickly to local events as the underlying switches themselves. This argues for having some control-plane functionality on the underlying switches, though arguably not the same complex software that runs on the switches in today's cellular networks.

In our architecture, each switch runs an agent that performs simple local actions, under the command of the controller. For example, the controller could offload simple measurement tasks to the local agents, such as periodically polling the traffic counters and notifying the controller if a count exceeds a threshold. The local agent could also perform simple control operations, such as automatically changing the weight or priority of a queue when traffic counts exceed a threshold, or pushing a tag on a packet to direct the traffic through an intermediate middlebox. Performing these operations on the local agents would reduce the load on the controller, and enable faster responses to critical events.

Supporting local agents on the switches raises many interesting research problems. Partitioning functionality between the controller and the agents requires ways for application modules to expose the inherent parallelism across agents and the necessary aggregation of information at the controller. In addition, the partitioning of functionality must work in the presence of multiple modules that form a single application. We plan to explore the design of the local agent and techniques for partitioning functionality in our future work.

3.3 Flexible Switch Patterns and Actions

Today's OpenFlow [8] switches already support many of the features needed in cellular data networks. Flexible packet classification based on Ethernet, IP and TCP/UDP header fields enables fine-grain quality-of-service, access control, and monitoring. The forwarding actions in today's OpenFlow switches would enable carriers to direct selective traffic through middleboxes, change the paths to and from a mobile user, and mark and schedule traffic according to QoS policies. The byte and packet counters associated with each rule would support traffic measurement, real-time adaptation based on congestion or exceeding a subscriber's usage cap, and usage-based billing. Still, these switches may need larger rule tables, or more stages of tables, than today's commodity switch implementations to efficiently support multiple levels of fine-grained policies.

That said, software-defined cellular networks would benefit from new switch capabilities. TCP/UDP port numbers are no longer a sufficiently reliable way to identify applications. Instead, support for deep packet inspection (DPI) would enable much packet classifications based on the application, such as Web, peer-to-peer, video, and VoIP traffic. This is important to divide traffic into separate traffic classes for different packet-scheduling and routing policies, as commonly done in today's cellular networks [9]. DPI would also help support intrusion detection and prevention systems that much analyze packet contents to identify malicious traffic. To contain cost, we do not envision that *every* switch supports DPI functionality, requiring additional support on the controller to place these packet-classification rules in the appropriate locations in the underlying network.

In addition to DPI-based packet classification, switches could also support techniques like header compression and decompression to reduce the overhead for applications with small packet payloads. For example, VoIP packets are typically small, making the headers a relatively high fraction of the traffic. Compressing these packets before transmission on low-bandwidth links substantially lowers the overhead. VoIP packet sizes range from 20 to 150 bytes, and the combined overhead of the RTP, UDP, and IP headers is 40 bytes. Robust header compression (ROHC) reduces the 40-byte overhead to 1 byte. As with DPI, this functionality may not be available on every switch, but instead mainly on switches with links to and from low-bandwidth regions of the network.

To enable real-time control, the switch could use a fast, message-oriented control protocol rather than the reliable byte-stream TCP protocol used in OpenFlow. Communication between the cell agent and the data plane can use a local system-call and upcall interface to send and receive information. Communication between the local agents and the controller can use the Streaming Control Transport Protocol (SCTP) [10] commonly used for real-time signaling in cellular networks. Compared to TCP, SCTP avoids head-of-line blocking and allows multi-homed nodes (e.g., switches with multiple interfaces) to shift control messages from one attachment point to another for faster failover.

3.4 Network virtualization on subscriber attributes

Today's SDN FlowVisor [3] allows the same hardware forwarding plane be shared among multiple logical networks. Much as a hypervisor resides between software and hardware on a PC, the FlowVisor uses OpenFlow as a hardware abstraction layer to sit logically between control and forwarding paths on a network device. FlowVisor is transparent to both the network hardware and the controller managing the virtual networks. FlowVisor defines a slice as a set of flows running on a topology of switches. The virtualization layer enforces strong isolation between slices. Resources that can be sliced are bandwidth, topology, traffic, device CPU, and forwarding tables.

We propose *CellVisor* which extends FlowVisor to slice cellular network resources.

- **Flexible slicing of base station resources:** CellVisor adds support to flexibly slice base station resources. CellVisor supports slice creation and slice deletion. During slice creation, CellVisor takes a slice configuration file. The slice configuration file will have configuration parameters for a virtual base station. For example, a slice can ask for base stations with a specific MAC. If the protocol stack is not supported at the software defined base station. The slice configuration file will point to a kernel module so that they can be downloaded into the base stations.

- **Slicing the semantic space:** CellVisor supports high level semantic space definition. A slice of the semantic space is the set of packets whose subscriber attributes satisfy the same predicates. For example, traffic of all roaming subscribers, all roaming subscribers from cellular provider A, all subscribers of a certain monthly plan, all UEs with a certain capability. This allows the cellular provider to isolate roaming traffic, to provide specific support for certain roaming traffic, isolate dumb phone traffic or dumb phone traffic using legacy protocols.

  To enable scalable slicing of semantic space, we can make use of MPLS tags or VLAN tags. For example, at ingress points (base stations or packet gateways), the packet is marked with the tag representing the slice. All data plane entities can apply the same "meta policies" (i.e. policy for all traffic of the slice) on packets of the slice. This does not preclude per-subscriber or per application policies. These policies can still be applied at ingress points. For scalability reasons, other data plane entities in the cellular networks should not keep per-subscriber policy states.

4. RELATED WORK

Our work is heavily inspired and follows the high-level vision of OpenRoads (or OpenFlow Wireless) which is a platform for innovation and realistic deployment of services for wireless networks. OpenRoads is the first software-defined wireless network. It is mainly based on WiFi and offers no special support for cellular networks. In contrast, CellSDN addresses specific cellular network requirements such as real-time session management which runs on top of SCTP instead of TCP, paging, UE (User Equipment) state tracking, policy enforcement, charging and radio resource management.

5. CONCLUSION AND FUTURE WORK

In this paper, we argue that software defined network principles make cellular networks much simpler and easier to manage, introduce new services, and inter-operate with other wireless network technologies and other operator networks. We sketch out the design of our CellSDN software defined cellular networks. We are in the process of prototyping CellSDN using open source LTE implementation.

6. REFERENCES

[1] "Skyfire launches its rocket 2.0 platform to help mobile operators deal with a growing data tsunami." `http://skyfire.com/images/press_releases/skyfirerocketoptimizerpress.pdf`.

[2] "Alcatel-Lucent 9900 wireless network guardian." `http://www.alcatel-lucent.com/wps/portal/products/detail?LMSG_CABINET=Solution_Product_Catalog&LMSG_CONTENT_FILE=Products/Product_Detail_000590.xml#tabAnchor1`.

[3] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Can the production network be the testbed?," in *OSDI*, Oct 2010.

[4] G. Bhanage, I. Seskar, R. Mahindra, and D. Raychaudhuri, "Virtual basestation: Architecture for an open shared WiMAX framework," in *Proc. ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, pp. 1–8, ACM, 2010.

[5] K.-K. Yap, R. Sherwood, M. Kobayashi, T.-Y. Huang, M. Chan, N. Handigol, N. McKeown, and G. Parulkar, "Blueprint for introducing innovation into wireless mobile networks," in *Proc. ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, pp. 25–32, ACM, 2010.

[6] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker, "Frenetic: A network programming language," in *ICFP*, Sep 2011.

[7] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. Gude, N. McKeown, and S. Shenker, "Rethinking enterprise network control," *Trans. on Networking*, vol. 17, Aug 2009.

[8] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM CCR*, vol. 38, no. 2, pp. 69–74, 2008.

[9] "Mobile application assurance on the Alcatel-Lucent 7750 service router mobile gateway: Optimize network resources, enrich and personalize user experiences, and monetize the services," 2011. Application Note.

[10] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol," Oct 2000. RFC 2960.