CellSDN: Software-Defined Cellular Core Networks

Xin Jin† Li Erran Li⋆ Laurent Vanbever† Jennifer Rexford†
Bell Labs⋆ Princeton University†

1. PROBLEMS OF CELLULAR CORE NETWORKS

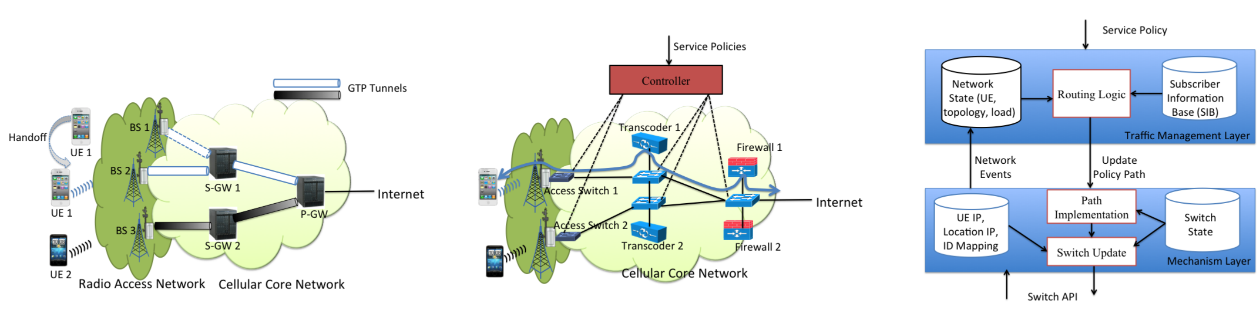
Cellular core networks carry traffic between base stations and the Internet on behalf of user equipment (UE) (see Figure 1(a)). The network relies on specialized equipment such as serving gateways (S-GWs) that provide seamless mobility when UEs move between base stations, and packet gateways (P-GWs) that perform a wide variety of functions like traffic monitoring and billing, access control, and parental controls.

The current architecture of cellular core networks are remarkably complex and inflexible [1, 2], an unfortunate legacy of their circuit-switched origins. Centralizing critical data-plane functionality at the boundary with the Internet forces all traffic to flow through the packet gateway—including device-to-device traffic. With so much functionality in one box, packet gateways are complex and expensive, and force carriers to buy functionality they do not need. Indeed, carriers cannot "mix and match" capabilities from different vendors (e.g., a firewall from one vendor, and a transcoder from another), or "scale up" the resources devoted to a specific function [2, 3]. Since the packet gateways are hard to change, carriers are forced to replace them to deploy new functionality, even when the existing equipment suffices for most purposes.

To make matters worse, growing link speeds and more diverse network policies will put even greater strain on packet gateways. Cellular networks can apply customized policies based on a wide variety of *subscriber attributes* as well as on the *application* [2]. For example, the carrier may direct traffic for older cell phones through an echo-cancellation gateway, video traffic through a transcoder during times of congestion, and all traffic through a firewall, while applying different monitoring policies depending on the billing plan, usage cap, roaming status, and the application.

Rather than perform all these functions at the Internet boundary, we argue that cellular providers should leverage software defined networking to adopt a network design more akin to modern data centers. The network should consist of a fabric of simple core switches, with most functionality moved to low-bandwidth access switches (at the base stations) and a distributed set of middleboxes that the carrier can expand as needed to meet the demands (see Figure 1(b)). These middleboxes could be dedicated appliances, virtual machines running on commodity servers [3], or simply packet-processing rules installed in the switches [4, 5]. A logically-centralized controller can direct traffic through the appropriate middleboxes, via efficient network paths, to realize a high-level service policy (e.g., directing a UE's video traffic through a transcoder and a firewall).

Such architecture raises unique scalability challenges compared to data-center and enterprise networks. Indeed, fine-grained policies can easily lead to an explosion in the data-plane state needed to direct traffic through the right middleboxes. The switches need to forward packets differently based on multiple factors (e.g., the UE and the application), which typically requires expensive TCAM (Ternary Content Addressable Memory) for packet



(a) LTE Network Architecture (b) CellSDN Network Architecture (c) CellSDN Controller Design

Figure 1: LTE Network Architecture vs CellSDN Network Architecture

classification. However, the merchant silicon chipsets used in commodity switches have just a few thousand to tens of thousands of TCAM entries. (See Table 2 in [6].) Supporting much larger packet classifiers would significantly increase the cost of the core switches.

2. CELLSDN DESIGN

To address these challenges, we present CellSDN, a scalable software-defined cellular core network architecture for supporting fine-grained policies for mobile devices. The CellSDN controller realizes high-level service polices by directing traffic through a sequence of middleboxes, optimized to the network conditions and UE locations. To ensure data-plane scalability, the core switches forward traffic on *hierarchical addresses* (grouped by base station) and *policy tags* (identifying middlebox paths). CellSDN pushes fine-grained packet classification to the access switches, which can be implemented easily in software (e.g., using Open vSwitch). These access switches apply fine-grained rules, specified by the controller, to map UE traffic to the policy tags and hierarchical addresses. To ensure control-plane scalability, a local agent at the base station caches the service policy for each attached UE.

The CellSDN controller guarantees that packets in the same connection traverse the same sequence of middleboxes (*policy consistency*), and that bidirectional traffic traverses the same middleboxes in both directions (*policy symmetry*), even in the presence of mobility. CellSDN has an *asymmetric* edge architecture that does *not* require sophisticated packet classification of return traffic arriving at the gateway switches. CellSDN either *embeds* the policy tags in the UE IP address and port number (essentially "piggybacking" the information in the packets sent to the Internet), or *caches* them at the gateway (in a simple Network Address Translation table). This ensures return traffic flows through the right middleboxes, without requiring any support from the rest of the Internet. CellSDN also does not require any changes to UEs or the radio access network hardware, and can run on commodity switches and middleboxes.

The CellSDN controller design cleanly separates traffic management from the low-level mechanisms for installing rules and minimizing data-plane state, as shown in Figure 1(c). The *traffic-management* layer determines the service attributes for a UE from the Subscriber Information Base (SIB), and consults the service policy to compute policy paths that traverse the appropriate middleboxes and optimize traffic-management objectives. The *mechanism* layer realizes the policy paths by installing rules in the underlying switches. This layer hides all the details of location-dependent addresses, the encoding of policy tags, the path implementation algorithm, and assuring path consistency during mobility. This modular controller design allows each layer to evolve independently, to adopt new innovations in how to manage traffic and data-plane state, respectively.

In designing, prototyping, and evaluating CellSDN, we make the following contributions:

- **Fine-grained service polices:** CellSDN supports fine-grained traffic steering based on applications and subscriber attributes, as well as flexible traffic engineering in selecting the network and middlebox paths.
- **Asymmetric edge design:** CellSDN places most functionality at the many, low-bandwidth access switches, allowing the core network to use commodity hardware for the Internet gateway and other core switches.
- **Scalable data plane:** CellSDN minimizes data-plane state in the core switches through multi-dimensional aggregation by policy tags, base station IDs, and UE IDs, and an algorithm for selecting policy tags.
- **Scalable control plane:** To ensure control-plane scalability, access switches run local agents that cache service policies for the attached UEs, and the controller isolates the access switches from core topology changes.
- **Policy consistency and symmetry:** CellSDN ensures that all traffic in the same TCP or UDP connection traverses the same sequence of middleboxes in both directions, even in the presence of mobility.
- **Realistic performance evaluation:** We evaluate the scalability our architecture based on traces ($\sim$1TB) from a large LTE deployment, micro-benchmarks on a prototype controller (built on top of Floodlight [7] OpenFlow controller), and large-scale simulation experiments.

3. REFERENCES

[1] B.-J. Kim and P. S. Henry, "Directions for future cellular mobile network architecture," Tech. Rep. TD:101033, AT&T Labs–Research, December 2012. http://www.research.att.com/techdocs/TD_101033.pdf.
[2] S. Elby, "Carrier vision of SDN and future applications to achieve a more agile mobile business," October 2012. Keynote address at the SDN & OpenFlow World Congress, http://www.layer123.com/sdn-live.
[3] "Network functions virtualization: Introductory white paper," October 2012. http://www.tid.es/es/Documents/NFV_White_PaperV2.pdf.
[4] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. Gude, N. McKeown, and S. Shenker, "Rethinking enterprise network control," *IEEE/ACM Trans. Networking*, vol. 17, August 2009.
[5] R. Wang, D. Butnariu, and J. Rexford, "OpenFlow-based server load balancing gone wild," in *Hot-ICE Workshop*, March 2011.
[6] B. Stephens, A. Cox, W. Felter, C. Dixon, and J. Carter, "PAST: Scalable Ethernet for data centers," in *ACM SIGCOMM CoNext Conference*, December 2012.
[7] "Floodlight OpenFlow Controller." http://floodlight.openflowhub.org/.