SoftMoW: A Dynamic and Scalable Software Defined Architecture for Cellular WANs

Mehrdad Moradi† Li Erran Li⋆ Z. Morley Mao†
Bell Labs, Alcatel-Lucent⋆ University of Michigan†

1. LIMITATION OF TODAY'S MOBILE WIDE AREA NETWORKS

The current 4G LTE network architecture is organized into very large and rigid regions, each with an access edge consisting of only base stations and an Internet edge comprised of centralized packet gateways (PGWs) that enforce almost all network policies. In this architecture, there is minimal interaction among regions other than interference management at the edge, and all users' outgoing traffic must traverse a PGW and possibly go through the Internet, even if the other endpoint is served by a close base station in the neighboring regions.

The centralized policy enforcement, rigid organization of large regions, and a lack of inter-region interaction make the current cellular network architecture incredibly inflexible and inefficient. First, a recent study [1] shows that the lack of a sufficiently close PGW is a major cause of path inflation, suboptimal routing, and QoS degradation in large carriers. Second, there is no support or simple solutions for IP-based mobility between regions ("inter-PGWs"). Thus, users crossing regions experience service interruption [2]. Third, the sheer amount of traffic and centralized policy enforcement at PGWs, and the inability to directly route traffic between regions take a heavy toll on the scalability and reliability of PGWs and the cellular architecture as a whole. Fourth, with the exponential growth of mobile data and rapidly changing traffic patterns, the current architecture is ill-suited to adapt to the rise of new applications such as machine-to-machine (e.g., connected vehicles, telehealth) and bandwidth-intensive applications.

Rather than organizing mobile wide area networks as rigid regions with no direct traffic transit, we argue that the cellular networks should have a fully connected core topology, small logical regions, and more egress points. In addition, operators should leverage software defined networking to manage the entire network with a logically-centralized controller. The controller directs traffic through efficient network paths that might cross region boundaries, supports and optimizes inter-region handoffs, and dynamically adapts to traffic patterns with efficient inter-region traffic engineering.

Such an architecture raises unique scalability challenges in comparison with data-center and enterprise networks due to the geographically distributed nature of mobile WANs. Indeed, a logically-centralized controller in one point-of-presence with a flat architecture quickly becomes infeasible, if the mobile WAN spans a large region. This is due to the high latency between the controller and the data plane switches, the amount of signaling load from mobile users, and the very high number cellular handoffs.

2. SoftMoW ARCHITECTURE

In response to the above challenges, SoftMoW presents a software-driven architecture for a wide area mobile network that can support tens of thousands of base stations and hundreds of millions users. To achieve such scalability, along with ensuring simplicity and manageability, it *hierarchically* builds up a network-wide control plane for the core and radio access networks. In each level, it abstracts both control and data planes and exposes a set of *dynamic and logical* programmable data plane components to the control plane of the level above.

For the rest of the paper, we explain our architecture with two hierarchical levels. In the first level, SoftMoW uses distributed policy enforcement by replacing expensive and inflexible hardware devices (i.e., PGWs and SGWs) with programmable switches and middleboxes. Switches are distributed over a large geographical area (e.g., a country) in a connected topology. Base stations are attached to switches at different locations to provide coverage. Network policies are enforced by directing traffic through middlebox instances that can be flexibly placed. SoftMoW partitions the wide area network into a set of *independent and dynamically defined* logical regions (Fig. 1). A child controller is responsible for managing data plane components of each region. Logical regions can be defined based on processing capabilities and geographical locations of child controllers, but they are subject to change through the control plane of the second level for optimization purposes (e.g., reducing the number of inter-controller handoffs).

In our architecture, child controllers have four responsibilities: 1) enforce network policies within their region by directing incoming traffic through partially ordered sets of middleboxes (also known as posets) and exiting from predetermined egress-points, dictated by the control plane of the second level. 2) manage radio access networks and handle intra-region hand-offs within their region. 3) expose dynamic and logical programmable components to the control plane of the second level. 4) interact with each other to seamlessly exchange the logical associations of programmable switches and base stations upon the command from the control plane of the second level.

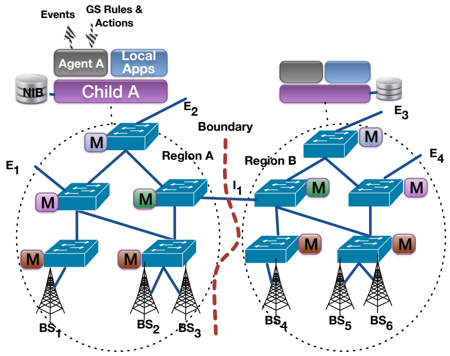

Figure 1: First level

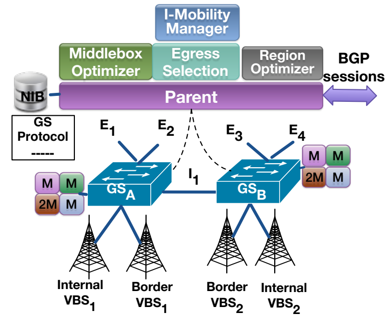

Figure 2: Second level

To build data plane components for the next level (Fig. 2), a module running on each child controller, called an *agent*, exposes *a G-switch, some virtual base stations, and some virtual middleboxes* corresponding to its logical region. A G-switch is a dynamic abstraction over all distributed physical switches of a logical region. Each port of a G-switch is connected to an ISP, another G-switch, or a virtual base station. A G-switch also runs a group of virtual middleboxes, each representing all distributed middlebox instances of the same type in the corresponding region. In addition, base stations deployed in a region are grouped based on their physical connectivity and handoff patterns, and then abstracted as a set of virtual base stations. Virtual base stations are further classified into two types depending on their geographical location: *border* and *internal*. Each logical region has a number of border virtual base stations and a few internal virtual base station. Border virtual base stations are on the border between two regions, and can be logically reassociated with another region during the change of region boundaries. An internal virtual base station represents a group of internal base stations and cannot be logically reassociated to a new region before moving border virtual base stations.

At the second level, a parent controller controls all the exposed logical components that correspond to the union of regions of child controllers. It uses a protocol to exchange control messages (i.e., events and commands) with G-switches through a publish-subscribe mechanism. In our application, an event can be a change in capacity and utilization of a virtual middlebox, throughput and coverage of a virtual base station, bandwidth and latency of a port of G-switch, etc. In response to these events, the parent may install pairs of rule and action into the logical flow table of G-switches. Each rule and action pair specifies **egress-point policies** and **a virtual middlebox poset** for incoming traffic. In addition, the parent has several responsibilities: 1) participates in the inter-domain routing protocol on behalf of all egress points. 2) runs a global link discovery protocol in cooperation with agents to learn the logical topology in addition to inter-region links not detected by any of child controllers through existing link discovery protocols (i.e., BDDP and LLDP). 3) supports seamless mobility across logical regions, and optimizes performance by updating logical region boundaries. 4) runs network wide applications, builds valid virtual middlebox posets and egress-point policies, and dictates to G-switches.

We prototype SoftMoW architecture by extending Floodlight to support multi-controller deployments, and make the following contributions:

- **Scalable and simple control plane:** SoftMoW decouples control and data planes at multiple levels to bring both simplicity and scalability to the control plane of large cellular networks by defining logical radio and non-radio data plane components, and offering a hierarchical control logic. It also improves failure localization and recovery as each child controller can perform failure detection and troubleshooting locally. Moreover, our design can be extended to more levels to accommodate very large operators.
- **Dynamically defined control plane:** SoftMoW starts with initial logical regions, and optimizes them over time due to installation of new base stations, switches, and middlebox instances. In particular, the parent controller minimizes the number of inter-region handoffs to avoid transferring control and data planes states among G-switches. This is done by exposing virtual base stations and changing their associated logical regions over time.
- **Application based traffic engineering and load balancing:** SoftMoW makes the deployment and design of network-wide applications feasible. In contrast to the existing packet core architecture with only a few egress points performing application-agnostic traffic engineering, SoftMoW offers network-wide application-based traffic engineering and middlebox load balancing. It also supports seamless mobility across regions.

3. REFERENCES

[1] K. Zarifis, T. Flach, S. Nori, D. Choffnes, R. Govindan, E. Katz-Bassett, Z. M. Mao, and M. Welsh, "Diagnosing Path Inflation of Mobile Client Traffic," in *PAM '14*, March 2014.

[2] J. C. Zuniga, C. J. Bernardos, A. de la Oliva, T. Melia, R. Costa, and A. Reznik, "Distributed Mobility Management: A Standards Landscape," *Communications Magazine, IEEE*, vol. 51, no. 3, pp. 80–87, 2013.