CS 6480: Project

Junguk Cho , Binh Nguyen, Jacobus Van Der Merwe[1]

[1]School of Computing, University of Utah

November 22, 2013

1 Introduction

As the mobile network and mobile devices (e.g., smart phone, tablets) are developed more and more, most of mobile users enjoy the video streaming service(need citation) and online games on mobile network. These services usually require low delay for users to enjoy them and generate enormous amount of traffic.

However, the current mobile network architecture inherently has several drawbacks for supporting these services effectively. First, the hierarchical routing causes the delay since the small number of S-GWs (Serving Gateway) and P-GWs lead to additional routing delay [1, 2]. Second, the traffic from mobile users always go through the P-GW (Packet Data Network Gateway) which routes the packet to internet as well as operates many functionality like QoS (Quality of Service), charging, and NAT (Network Address Translation). It will be the bottleneck in current mobile network.

- Explain that it is hard to modify current mobile network because of compatibility with other network (e.g, circuit network, 3G, wifi, and Wimax)

To overcome these inherent problems and provide quality of service for users when they use delay critical applications, we propose the traffic offloading service in core network to reduce the delay by hierarchical routing and centralized P-GW without any modifications of the current mobile network.

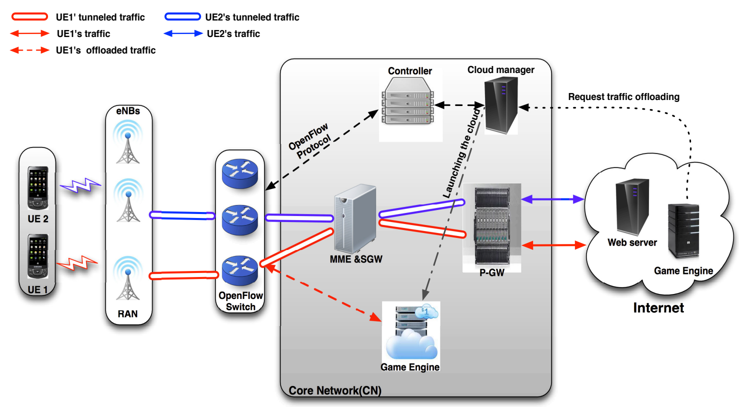

Figure 1: The system architecture

2 Background and Motivation

* explain the current mobile network architecture

* complex, bottleck (p-gw, scalability) , require more services

3 Architecture

3.1 Overview

Our work adds to the current cellular network two main components: a SDN substrate, and a registration and control server. The SDN substrate locates in between the Radio Access Network (RAN) and core network. This substrate consists of Openflow switches (OFSes) that talk to eNodeBs via GTP tunnel. The OFS proactively reroutes offloading traffic at the behest of its controller. The registration server acts as a brain of the system. It stores information of traffic that needs offloading (e.g., IP prefix of a game server) and talks to the SDN substrate to realize the offloading functionality.

A basic workflow of the system is described as follow: users (e.g., CDN providers) register their services on the central registration center. The central register center then translates these information into SDN rules (e.g., match:actions) and deploys the rules on the SDN substrate. Once these rules are deployed, the SDN substrate is able to reroute registered traffic to appropriate offloading servers while leaving other traffic untouched.

3.2 Realization

3.2.1 Registration server

The brain of the system is the registration server. This server provides an interface that allows users to register for their offloading services. Users specify the properties of their offloading traffic and the QoS associated with the traffic. For example, a game provider registers a list of IP prefixes of game engines that it wants to provide offloading functionality. It is also be able to specify the Quality of Service (QoS) of these offloading traffic if needed. The registration server then translates the high-level registration information into low-level actual SDN rules that are deployable in the SDN substrate. These rules should be able to distinguish and redirect the offloading traffic to offloading servers inside the core (since the SDN substrate is a set of OFSes, routing inside the substrate needs more elaboration).

3.2.2 SDN substrate

SDN substrate is a set of OFSes controlled by controller(s). These OFSes forward traffic at the behest of controllers. In contrast with the current architecture, OFSes contact with offloading servers using normal IP links instead of GTP tunnels (as in Firgure 1. Current Openflow does not support packet payload matching so OFSes can not forwarding packets based on GTP tunneling information. We propose to modify the OFS implementation to force it do GTP decapsulating before feeding packets into the normal OFS pipe line (packet matching and forwarding based on flow table).

3.2.3 Packet offloading

When attaches to the network for the first time, UE follows a normal attaching process (e.g. authentication, bearers establishment, and acquiring an IP address from the PGW). After resources are allocated, the UE can talk to the server using server's IP address. Traffic from UE, however, are not fully routed to the Internet as usual but intercepted by the OFS, and traffic destined for the registered server are rerouted to the local offloading server. (dotted arrow in Figure 1).

Packets arrived at the OFS are filtered by OFS's controller. When arrived at the OFS, packets from UE are GTP encapsulated. Since Openflow's data path does not support payload matching, GTP packets are sent to OFS's controller for packet decapsulation and matching. OFS's controller decapsulates the GTP packets and compares their destination IPs with the database of registered offloading server's IPs. If matched, controller forwards decapsulated packets to port that connects to the offloading server. In the mean time, controller installs a rule for packets in the reversed direction. This rule changes the source IP address of the packets in the reversed direction as if these packets come from the server in the Internet. Therefore, the communication is transparent at UE's perspective.

4 Evaluation

* Measure the packet transfer delay between offloaded traffic and non-offloaded traffic with web server and streaming service.

5 Related Work

Our work is strongly inspired by MOCA [1]. In MOCA architecture, although modifications on the current network were intentionally minimized, the MME still played a centralized role and need to be modified. Our work, in the other hand, does not require any change on the MME or any component in the current architecture. The only requirement is traffic between eNodeBs and SGW have to go through SDN switches.

Local IP Access (LIAP) and Selected IP Traffic Offload (SITOP) have been proposed in 3GPP standard for data offloading. While LIAP is more suitable for home environments, SITOP is proposed to route data out of the core network as soon as possible but not to the CDN servers inside the core. LIAP and SITOP also require specific hardware to be deploy while the goal of our work is not to touch the current components.

There are works [3, 4] that address scalability and inflexibility problems of PGW using SDN. These works suggest the possibility of using SDN to enable flexible routing inside the core. MOCA relates to the works in the sense that it leverages SDN to share the pressure of data delivering on the PGW.

6 Conclusion

References

[1] Banerjee, A., Chen, X., Erman, J., Gopalakrishnan, V., Lee, S., and Van Der Merwe, J. Moca: a lightweight mobile cloud offloading architecture. In *Proceedings of the eighth ACM international workshop on Mobility in the evolving internet architecture* (2013), ACM, pp. 11–16.

[2] 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.

[3] Jin, X., Li, L. E., Vanbever, L., and Rexford, J. Softcell: Scalable and flexible cellular core network architecture.

[4] Li, L. E., Mao, Z. M., and Rexford, J. Toward software-defined cellular networks. In *Software Defined Networking (EWSDN), 2012 European Workshop on* (2012), IEEE, pp. 7–12.