Demo: An All-In-One Community LTE Network

Spencer Sevilla, Matthew Johnson, Pat Kosakanchit, Jenny Liang, Kurtis Heimerl
{sevilla,matt9j,pathik,jliang9,kheimerl}@cs.washington.edu
Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle, WA

ACM Reference Format:
Spencer Sevilla, Matthew Johnson, Pat Kosakanchit, Jenny Liang, Kurtis Heimerl. 2019. Demo: An All-In-One Community LTE Network. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19), October 21–25, 2019, Los Cabos, Mexico.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3300061.3343371

1 INTRODUCTION

Affordably providing broadband Internet to the long tail of rural, disconnected communities worldwide is an open research challenge. One particularly promising solution is *Community Networking*. Community Networks, largely defined as networks built and operated by local actors in a community-centric and often cooperative fashion, mitigate many of the economic concerns of operating in rural areas. Prior work in community networks has leveraged a wide range of access technologies, including WiFi [1] as well as 2G [2] and 3G [3] cellular networks.

LTE has been recently shown [4] to be a good fit for rural access. This is the case for several reasons: it is wide-area, high-bandwidth, can use IP primitives that remove the need for telecom interconnect, and has recently developed a robust uptake of client devices even in remote areas [5]. LTE is also available in over forty different bands, a number of which are unlicensed or available to small operators. Despite these advantages, LTE is still fundamentally a telecom technology, designed for highly centralized operation wherein the cellular radios (eNodeBs) are managed by a set of specialized network functions kept in a single location under the operator's control. In cell networks these functions are commonly referred to as the "core," and in LTE as the Enhanced Packet Core (EPC).

The introduction of small-scale hardware manufacturers into the LTE market have caused recent reductions in the price of COTS LTE base stations (eNodeBs), along with increased hardware availability in a wider range of bands. These market shifts have made LTE an increasingly attractive solution for bridging the digital divide. However, a license for a commercial EPC remains prohibitively expensive. The OpenAirInterface project [6] provides an open-source EPC, but this project is primarily intended to be a reference implementation used by telecom researchers and engineers working on 5G technology, and was not built for a production or live network context.

To bridge this gap and address this need, we stabilized, extended, and enhanced the OpenAirInterface EPC to create CoLTE, the Community LTE Project. CoLTE is a stable, reliable LTE EPC suitable for use in a live network with paying users. Our key contributions that we wish to demonstrate are threefold: First, we contributed a number of key improvements to the OpenAirInterface codebase necessary for stabilization and use in a production network, and made a number of novel architectural choices to this effect. Second, we built a number of locally-hosted webservices that allow for configuration of OpenAirInterface as well as user-based account management. Finally, we built an IP-based network manager called Haulage that interfaces with OpenAirInterface to provide user accounting and authorization. All of our code is fully free and open-source, and is available at the following github repositories: https://github.com/uw-ictd/colte, https://github.com/uw-ictd/openair-cn, and https://github.com/uw-ictd/haulage. A full discussion of CoLTE design principles, and a six-month evaluation of an initial CoLTE deployment in Papua, are presented as a separate work in the main conference [7].

2 DESIGN AND IMPLEMENTATION

2.1 OpenAirInterface Productization

OpenAirInterface is distributed to the public as a C source code repository paired with a set of fragile and often outdated build scripts. We took this workflow and first streamlined it by removing unnecessary build dependencies and simplifying the set of build tools into a standard Makefile. We then packaged the OpenAirInterface binaries and configuration files into Debian packages for easy installation by non-developers running either Debian 9 (Stretch) or Ubuntu 18.04 (Bionic).

MobiCom '19, October 21–25, 2019, Los Cabos, Mexico

ACM ISBN 978-1-4503-6169-9/19/10.
https://doi.org/10.1145/3300061.3343371

In addition to packaging OpenAirInterface, we also built a tool called `colteconf` to aid and simplify system configuration. This tool takes as input simple configuration variables (e.g. “enb_interface” and “wan_interface”) and updates all other configuration files as necessary, thereby ensuring that all components of the EPC are always synchronized and configured correctly with respect to each other.

2.2 Over-The-Top Telecommunications

A unique characteristic of CoLTE, when compared to other LTE core networks, is that CoLTE does not support the network-native telecommunications typically provided by a cellular network (e.g. voice and SMS). It follows that telecommunications can only be enacted via over-the-top IP-based services such as Skype or WhatsApp.

We made this decision intentionally for several reasons, including (1) the high costs of securing PSTN interconnect and phone numbers, (2) the carrier-whitelist needed to enable VoLTE on handsets, (3) the inability to acquire SIM cards with the ISIM applet needed for VoLTE, and (4) the ubiquity of WhatsApp adoption in rural areas including our target communities. Additionally, this design helps to simplify our core network architecture by removing (1) the need to stabilize and deplo an IP Multimedia Subsystem (IMS), which consists of four separate logical components, and (2) the need to extend the OpenAirInterface HSS to interconnect with the IMS I-CSCF.

Our team is considering adding VoLTE support as future work. However, we expect that this work will represent a significant systems engineering effort involving (1) modifications to OpenAirInterface, (2) integrating OpenAirInterface against an open-source IMS server, (3) writing an ISIM applet and a tool to side-load it onto a SIM card, and (4) potential modifications to the Android OS.

2.3 Local Web Services

In addition to productizing OpenAirInterface, we also implemented several locally-hosted webservices that run on the EPC. These webservices include a configuration and management tool restricted to administrators as well as a web-based account management tool for users to top-up their account and send currency to other users, illustrated in Figure 1. We also host local copies of user-focused webservices, including OpenStreetMaps, Wikipedia, YouTube, and a media server.

2.4 Splitting Control From Management

Because OpenAirInterface was designed for small-scale research testbeds, it dynamically assigns IP addresses out of a preconfigured IP address subnet (default value `192.168.150.0/24`). Notably, this system dynamically assigns users a different IP address every time they disconnect and reconnect (i.e. after an idle detach-reattach). This presented a challenge for operating in a production/managed network context that requires accurate usage tracking for purposes of accounting and billing.

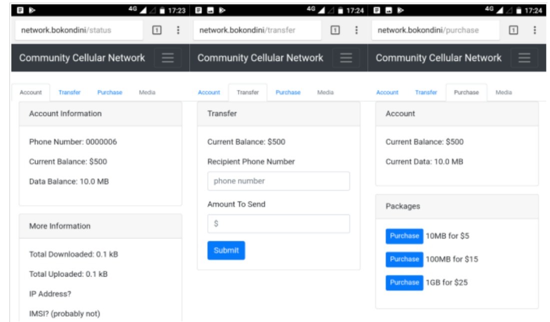

Figure 1: CoLTE Web Interface

We resolved this issue by adding a stub RADIUS client to the OpenAirInterface SPGW. When enabled, this client uses the RADIUS protocol to query a Authentication, Authorization, and Accounting (AAA) server with the user’s IMSI for purposes of IP address assignment. This represents a departure from traditional telecom network architecture, wherein the network management system and tools (i.e. the PCRF and PCEF) are deeply integrated with the network control plane as part of a single cohesive system. In contrast, our system seeks to separate these two tasks (management and control) into two separate systems with a cleanly defined interface (RADIUS) between them.

This split radically eases our software deployment model in two key ways. First and foremost, it enables OpenAirInterface to be deployed with or without an AAA server. If an AAA server does not exist, OpenAirInterface defaults to an unmanaged and unbilled open-access[1] LTE network suitable for many common use-cases. Second, our use of RADIUS, a relatively ubiquitous protocol, enables integration with many different IP-based AAA servers without requiring or enforcing vendor lock-in.

This split has powerful implications for the future of mixed-access networks, by enabling a simple and clear division between the network *control* plane (i.e. LTE-specific content like a user’s IMSI, Ki, and IP address) and the network *management* plane (i.e. AAA operations such as billing). Under this split, a AAA server can map a single user-account across multiple access mediums, such as LTE, WiFi, and Ethernet, using standard IP-based accounting and enforcement mechanisms.

[1]Note that SIM cards with correct keys must still be procured.

2.5 Haulage

Building off of the above section, we built a simple IP-based network management and accounting tool called Haulage. We wrote Haulage to fill a very specific gap, in that all other account-managing services we found were either postpaid or cloud-based.

Haulage sits on the data forwarding plane between the EPC and the Internet, and is responsible for (1) accounting the bytes a user sends and receives, (2) managing these bytes from a user-balance, (3) cutting users off from Internet access when appropriate, (4) maintaining the user-IP binding needed to accomplish tasks 1-3, and (5) exposing the user-IP binding via a RADIUS server. Haulage accomplishes these tasks via high-speed packet processing functions written in go, and interfaces with the kernel forwarding table via `iptables` commands.

Haulage identifies individual users based on their IMSI (or other unique identifier), and assigns each user a consistent IP address for purposes of accounting. Haulage maintains two balances for a given user, their *account balance* (in currency) and their *data balance* (in megabytes). Users top-up by purchasing credit (i.e. their account balance) from a reseller or network operator, and then use this balance to purchase data packages. When a user's data balance is drawn down to zero, Internet access is restricted to the user's account page, where they can purchase additional data.

3 DEMONSTRATION REQUIREMENTS

For our demo, we will setup and run a nano-scale CoLTE network. We will provide a sample phone as well as multiple SIMs, thereby enabling audience members to either use the sample network or try to connect their own device. If Internet access is available, users will be able to use our network to connect to the Internet; if not, we can still demonstrate network attach and operation, but only locally-based services will be available.

We will bring with us all the equipment needed: two computers (one laptop, one miniPC), a small USB software defined radio (an Ettus B205), a commercial smartphone, and all cables/power adapters required. Figure 2 illustrates our entire testbed.

Our demo will require a single table, and the default setup will be fine. We would appreciate Internet access if possible. Our demo will require approximately ten minutes of setup time, three minutes of presentation time, and can continue as long as there is interest.

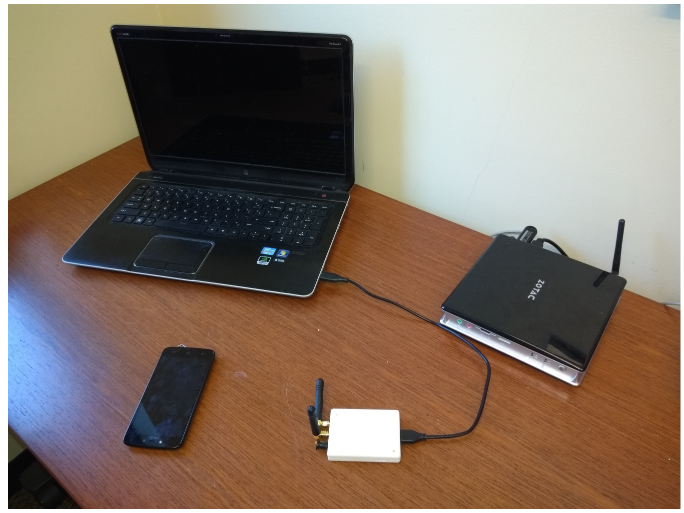

Figure 2: Our Demonstration Testbed

REFERENCES

[1] Roger Baig, Ramon Roca, Felix Freitag, and Leandro Navarro. Guifi. net, a crowdsourced network infrastructure held in common. *Computer Networks*, 90:150–165, 2015.

[2] rhizomatica. https://www.rhizomatica.org.

[3] Andrés Martínez Fernández, José Vidal Manzano, Javier Simó Reigadas, Ignacio Prieto Egido, Adrián Agustín de Dios, Juan Paco, and Álvaro Rendón. The tucan3g project: wireless technologies for isolated rural communities in developing countries based on 3g small-cell deployments. *IEEE communications magazine*, 54(7):36–43, 2016.

[4] Matthew Johnson, Spencer Sevilla, Esther Jang, and Kurtis Heimerl. dlte: Building a more wifi-like cellular network:(instead of the other way around). In *ACM HotNets*, 2018.

[5] Kushal Shah, Philip Martinez, Emre Tepedelenlioglu, Shaddi Hasan, Cedric Festin, Joshua Blumenstock, Josephine Dionisio, and Kurtis Heimerl. An investigation of phone upgrades in remote community cellular networks. In *Proceedings of the Ninth International Conference on Information and Communication Technologies and Development*, page 6. ACM, 2017.

[6] Navid Nikaein, Mahesh K Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. Openairinterface: A flexible platform for 5g research. *ACM SIGCOMM Computer Communication Review*, 44(5):33–38, 2014.

[7] Spencer Sevilla, Matthew Johnson, Pathirat Kosakanchit, Jenny Liang, and Kurtis Heimerl. Design, implementation, and deployment of colte, a community lte solution. In *ACM MobiCom*, 2019.