Productizing and Deploying The OpenAirInterface EPC

Spencer Sevilla, Kurtis Heimerl
This Talk

• Background/Context
• Modifications we made to openair-cn
• User Accounting
• Helper web-apps
• Results from the field
Background

• 4 billion humans are still not online; mostly in rural/developing areas
• We (UW’s ICTD lab) work directly on closing that gap.
• Prior work on 2G (Osmocom, Endaga) led us to LTE
• **Point #1**: We build novel network technologies as needed
• **Point #2**: We work with relatively low-tech-literate users
Context

• In August 2018 we deployed a small-scale LTE network in rural Papua
• Tower has two commercial eNBs
• Single mini-PC running the OAI EPC
• Backhaul a 3 Mbps satellite link
• Currently serving 70+ users
• And growing!
Goals

- A reliable EPC that works with a wide variety of handsets
- Provides the functionality needed to run a commercial network
- Easy and consistent systems to build, deploy, configure
- Focus on downstream on-ramps
A Quick Note On Naming/Versioning

• Started by forking OPENAIRINTERFACE/openair-cn in Spring 2018
• Named our project CoLTE (for Community LTE)
• CoLTE = (base OAI + modifications) + peripheral apps/services
• https://github.com/uw-ictd/openair-cn
• https://github.com/uw-ictd/colte
Modifications to openair-cn

- Build system
- Binary packages
- Systemd integration (not daemon mode)
openair-cn: Build system

• Focus only on what we can test/support: Debian 9 and Ubuntu 18.04
• Turned the build scripts into a Makefile in top directory: “make spgw”, “make all”
• Removed legacy dependencies, reduced downloads needed to build
• Makefile *also* provides all code needed for custom libraries: “make libraries” compiles freeDiameter, asn1c, libgtpnl, liblfds
• Copying code over prevents compatibility/version issues or dead links
openair-cn: Binary packages

• Makefile builds binary packages for Debian 9 and Ubuntu 18.04
• We host these at http://colte.cs.washington.edu
• “apt-get install colte-epc” provides a sane and stable EPC install!
• postinst scripts setup freeDiameter s6a keys
• Package includes sane MySQL database and conf files
• Package also installs systemd .service files...
openair-cn: Systemd Services

- Installs three separate systemd services (hss, mme, spgw)
- Can control all of them (systemd start colte-epc) or individuals (systemd start colte-mme)
- Allows us to enable service on boot-up (after network.target)
- Services stay online and reboot if (when) they crash
- MME also reboots SPGW to ensure system stays in-sync
- Run as a thread (not daemon) allows us to monitor service health
- This also enables smart logging with journalctl
User Accounting

• **Big problem:** No support for user accounting/billing!
• We’re just an ISP, all we have to do is log bytes consumed per user
• SPGW dynamically assigns IP addresses, makes it hard to track users
• **Option #1:** build a billing system into the SPGW
• **Option #2:** Integrate EPC with some other IP-based billing platform
User Accounting: IP Data-Plane
User Accounting: Traditional Telecom Stack

- OFCS
- TDF
- PCRF
- OCS
- BBERF
- SGW
- PCEF
- PGW

UE connected to Internet via S5/S8, Gxx, Gz, Gx, Gy, Sd, Sy
User Accounting: Simpler Is Better

• We are researchers and like to move quickly
• Writing full PCRF integration would take a while
• Wanted to be modular and integrate with existing ISP tooling
• **Solution:** Add a RADIUS client at SPGW
User Accounting: RADIUS Data-Plane
User Accounting: RADIUS Data-Plane

Many Commercial Systems

UE → SPGW → Traffic Operations → AAA Server → Internet
RADIUS Operations At SPGW

• When UE joins network, SPGW asks AAA server for IMSI->IP mapping
• AAA server assigns IP; connection fails if AAA rejects access
• (other system) does auth and accounting based on assigned IP

• Backup: If no AAA server configured, SPGW assigns IP out of pool
Haulage For Network Accounting

• High-speed packet processing app
• Runs on the EPC, which is also our Internet gateway
• Simple architecture: uses libpcap, written in go
• Currently supports two billing plans: per-MB and per-month
• Prepaid plans are standard in developing areas
• [https://github.com/uw-ictd/haulage](https://github.com/uw-ictd/haulage)
Haulage
Haulage

• Haulage splits nw management from nw control and link access
• Install OAI to run an unmetered LTE network (lab bench systems)
• Install OAI + Haulage to run a commercial LTE network
• Haulage can be deployed at a gateway and used to monitor/bill any IP-based network: LTE, WiFi, Ethernet, etc.
• Future work: integration with DHCP, Mikrotik, etc.
Long-Term Research Agenda: Mixed-Access

• **Option #1:** The LTE management core also regulates WiFi access (this is the standard telecom model: PCRF, Non-3GPP Trusted Access)

• **Option #2:** LTE is just one access technology of many
  - IP-based management plane is more flexible and extensible
  - Split the link-access system from the network-access system
  - Modular deployment: EPC on its own is unmetered, add a manager if you like
New Apps

• Colteconf
• WebAdmin Tool
• User WebGui
New Apps – Colteconf

• OAI EPC has many, many variables/options
• Updating just one of them can lead to out-of-sync problems
• We wrote a tool (called colteconf) that keeps the EPC in sync
• Powered by ansible: updates .confs and restarts services as needed
• Variable names are relatively sane and easy-to-understand
• Usage: Edit /usr/local/etc/colte/conf.yml, then call “colteconf update”
New Apps – WebAdmin Tool

• Allows web-based configuration and administration of EPC
• Essentially a colteconf front-end
• Password-protected; also supports IP-based restrictions
• Optionally accessible from RAN for specific users (default = False!)
• Hooks into Haulage to provide account management as well
New Apps – WebAdmin Tool
New Apps – WebAdmin Tool
New Apps – User WebGUI

- Phones can access a user portal at http://network.{network_name}
- Shows their phone number, account balance, data usage, and more
- Allows users to top up, buy data, and send money to other users
- Access restricted to the LTE interface (gtp0)
New Apps – User WebGUI
Field Deployment: Technical Details

• Two BaiCells Nova-233 eNodeBs: 850 MHz, 1-Watt
• 120° sectors, 2x2 panel antennas
• 15-foot pole mount on schoolhouse
• Power provided by solar micro-grid
• EPC housed inside
• Currently 70+ active users
Field Deployment: Coverage
Field Deployment: Coverage
Field Deployment: Quirks

• No telecom services, just IP connectivity
• (Everyone in the community uses WhatsApp)
• Solar power: Network shuts off every evening 9pm-6am
• Backhaul a 3Mbps satellite link
• (Yes, it’s congested)
## Field Deployment: Costs

<table>
<thead>
<tr>
<th>Capex</th>
<th>Cost (USD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPC + eNB + SIM + Antennas</td>
<td>8,349</td>
</tr>
<tr>
<td>Local Integration (Tower + Power)</td>
<td>175</td>
</tr>
<tr>
<td>Import + Shipping Fees</td>
<td>810</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>9334</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opex</th>
<th>Cost (USD/Month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backhaul (3 Mbps VSAT)</td>
<td>300</td>
</tr>
<tr>
<td>Maintenance</td>
<td>91</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>391</strong></td>
</tr>
</tbody>
</table>
Field Deployment: Revenue

• Data sold at a flat rate of 7.02 USD per Gigabyte
• Mean usage of 110 Mb/day/user
• Total usage of 4.4 GB/day -> revenue of over 1500 USD/month
• **Bottom line:** not just sustainable but profitable as well!

• Profits currently go to local school and scholarship programs
• Tradeoff: Reducing rates vs congestion
• Future research: Intersection of economics and network consumption
Conclusions

• We’re running OAI in production!
• Minor modifications to stabilize and systemize
• Added some peripheral apps as needed
• This network works well, hoping to build more soon
• Thanks a ton for all your work and help