Proteus: A Network Service Control Platform for Service Evolution in a Mobile Software Defined Infrastructure

Aisha Syed and Kobus Van der Merwe

This work was supported by the National Science Foundation under grant numbers 1305384 and 1302688.
Motivation

- Mobile network service deployment and evolution very slow
Motivation

- Monolithic service abstraction: mobile broadband

Human-to-human  Human-to-machine
• Future mobile networks
• Future mobile networks

Monolithic service abstraction won’t work
• Future mobile networks

Monolithic service abstraction won’t work

Need varied service abstractions and offerings
• Future mobile networks

Monolithic service abstraction won’t work

Need varied service abstractions and offerings

Need more evolvable infrastructure
Motivation

• With SDN and in-network clouds enabling NFV, we realize trend towards a mobile software-defined infrastructure (SDI):
With SDN and in-network clouds enabling NFV, we realize trend towards a mobile software-defined infrastructure (SDI):
Motivation

• With SDN and in-network clouds enabling NFV, we realize trend towards a mobile software-defined infrastructure (SDI):
Motivation

• With SDN and in-network clouds enabling NFV, we realize trend towards a mobile software-defined infrastructure (SDI):
Motivation

• With SDN and in-network clouds enabling NFV, we realize trend towards a mobile software-defined infrastructure (SDI):

Success of such an SDI will depend on the control platform
Proteus

- A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI
Proteus

• A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI

Component Implementation & Specification
Proteus

• A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI
Proteus

• A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI

<table>
<thead>
<tr>
<th>Component Implementation &amp; Specification</th>
<th>Service Abstraction Composition &amp; Specification</th>
<th>Proteus runtime</th>
</tr>
</thead>
</table>
Proteus

- A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI
Challenges and Design Principles

• Clean separation between infrastructure and service abstractions
Challenges and Design Principles

- Clean separation between infrastructure and service abstractions
Challenges and Design Principles

- Clean separation between infrastructure and service abstractions
Challenges and Design Principles

• Clean separation between infrastructure and service abstractions
Challenges and Design Principles

- Diversity in implementation and requirements
Challenges and Design Principles

- Diversity in implementation and requirements
Challenges and Design Principles

• Diversity in implementation and requirements
  • Polymorphic templates
Challenges and Design Principles

• Diversity in implementation and requirements
  • Polymorphic templates
Challenges and Design Principles

- Diversity in implementation and requirements
  - Polymorphic templates

Diagram:
- Physical eNodeB
- Virtual eNodeB
- eNodeB Template
- Specialized eNodeB Templates
- PGW
- SGW
- Physical mobile resources
- Compute
Challenges and Design Principles

• Mobility specific requirements
Challenges and Design Principles

• Mobility specific requirements
  • Templates capture logical service topology and dependencies
Challenges and Design Principles

- Mobility specific requirements
  - Templates capture logical service topology and dependencies
  - Templates and platform itself inherently data centric
Challenges and Design Principles

• Mobility specific requirements
  • Templates capture logical service topology and dependencies
  • Templates and platform itself inherently data centric
Challenges and Design Principles

- Mobility specific requirements
  - Templates capture logical service topology and dependencies
  - Templates and platform itself inherently data centric
  - Service agnostic resource placement
Challenges and Design Principles

• Mobility specific requirements
  • Templates capture logical service topology and dependencies
  • Templates and platform itself inherently data centric
  • Service agnostic resource placement
Challenges and Design Principles

• Safe service evolution and hosted multiplicity
Challenges and Design Principles

• Safe service evolution and hosted multiplicity
  • Service realization plan
Challenges and Design Principles

• Safe service evolution and hosted multiplicity
  • Service realization plan
Challenges and Design Principles

• Safe service evolution and hosted multiplicity
  • Service realization plan

Service Template

Init  
Stop  
Start

Service

Template

setupOS

configureNetworking

installPackages

setupDatabases

setupConfigFiles(args)

startProcess

OS_OEPC

ifconfig ..
route ..
DNS resolver ..

apt-get ..
wget myPkg ..

mysql PGW_bindings_db ..

myPkg/PGW_config.sh args

myPkg/PGW_start.sh
Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing
Challenges and Design Principles

• Safe service evolution and hosted multiplicity
  • Service realization plan
  • Component sharing
Challenges and Design Principles

• Safe service evolution and hosted multiplicity
  • Service realization plan
  • Component sharing
  • Component migration and traffic redirection primitives
Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing
  - Component migration and traffic redirection primitives

Location: SF
Challenges and Design Principles

• Safe service evolution and hosted multiplicity
  • Service realization plan
  • Component sharing
  • Component migration and traffic redirection primitives

Location: SF
Challenges and Design Principles

• Safe service evolution and hosted multiplicity
  • Service realization plan
  • Component sharing
  • Component migration and traffic redirection primitives
Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing
  - Component migration and traffic redirection primitives

Location: NYC
Use Cases

LTE/EPC: Standard mobile broadband service
Use Cases

LTE/EPC: Standard mobile broadband service
Use Cases

LTE/EPC: Standard mobile broadband service
Use Case: Hosted Multiplicity

LTE/EPC

Multiple parallel instances

Radio Access Network

Core Mobile Network

IP Substrate

eNodeB

UE

eNodeB

MME

SGW

PGW

UE

eNodeB

UE

UE

MLB

MME-PE

MME-PE

MMP

IP Substrate'

SDN

SDN

SDN

IP Substrate

Low-latency App

Edge Cloud

Offloaded traffic

Regular traffic

IoT group

communication

IOT

Device

(a)

(b)

(c)

(d)
Use Case: SCALE

LTE/EPC

Radio Access Network

IP Substrate

Core Mobile Network

UE

eNodeB

MME

SGW

PGW
Use Case: SCALE

LTE/EPC

Radio Access Network

Core Mobile Network

UE

eNodeB

eNodeB

IP Substrate

MME

SGW

PGW

MLB

MME-PE

MMP

IP Substrate'

MLB

MME'

SGW

PGW

IoT

Device

IoT

Device

IoT

Device

Evolution

SCALE [Banerjee et al., CoNEXT 2015]
Use Case: SCALE

SCALE [Banerjee et al., CoNEXT 2015]
Use Case: SMORE

LTE/EPC

Radio Access Network

- UE
- eNodeB
- eNodeB

IP Substrate

Core Mobile Network

- MME
- SGW
- PGW

LTE/EPC

- UE
- eNodeB
- MME
- SGW
- PGW

IoT Device

- IoT
- IoT
- IoT

Low-latency App

Edge Cloud

Offloaded traffic

Regular traffic

IoT group

Communication

Offloaded traffic

Regular traffic
Use Case: SMORE

LTE/EPC

Radio Access Network

Core Mobile Network

IP Substrate

UE

eNodeB

SGW

PGW

MME

MME-PE

MMP

SGW

PGW

MME

SDN

SDN

SDN

UE

eNodeB

IP Substrate'

Low-latency App

Edge Cloud at MTSO

Low-latency App

Edge Cloud

SMORE [Cho et al. ATC 2014]
Use Case: SMORE

LTE/EPC

Evolution

Radio Access Network

UE → eNodeB

Core Mobile Network

MME

SGW

PGW

IP Substrate

Low-latency App

Edge Cloud at MTSO

SMORE [Cho et al. ATC 2014]

Regular traffic
Use Case: SMORE

LTE/EPC

Radio Access Network

Core Mobile Network

IP Substrate

UE

eNodeB

MME

SGW

PGW

Low-latency App

Edge Cloud at MTSO

SDN

Offloaded traffic

UE

eNodeB

SMORE [Cho et al. ATC 2014]
Use Case: MobiScud

LTE/EPC

Radio Access Network

Core Mobile Network

UE

eNodeB

IP Substrate

MME

SGW

PGW

LTE/EPC
Use Case: MobiScud

MobiScud [Wang et al. ATC 2015]
Use Case: MobiScud

Evolution

Radio Access Network

Core Mobile Network

Edge cloud

UE

eNodeB

SDN

SDN

UE

eNodeB

UE

eNodeB

UE

eNodeB

Edge cloud

PVM

MME

SGW

PGW

MobiScud [Wang et al. ATC 2015]
Use Case: MobiScud

Radio Access Network

Core Mobile Network

Edge cloud

Offloaded traffic

Regular traffic

MobiScud [Wang et al. ATC 2015]
Use Case: MobiScud

Evolution

Radio Access Network

UE ➔ eNodeB ➔ SDN ➔ Core Mobile Network

UE ➔ eNodeB ➔ SDN ➔ Core Mobile Network

MobiScud [Wang et al. ATC 2015]
Use Case: MobiScud

Evolution

Radio Access Network

Core Mobile Network

MobiScud [Wang et al. ATC 2015]
Use Case: MobiScud

Evolution

Radio Access Network

Core Mobile Network

Edge cloud

MobiScud [Wang et al. ATC 2015]
Use Case: Specialized EPC for IoT
Use Case: Specialized EPC for IoT

Radio Access Network

LTE/EPC

Evolution

Core Mobile Network

IP Substrate

SDN

Hypothetical future architecture

UE

eNodeB

MME

SGW

PGW

MME-PE

MMP

IP Substrate'

LTE/EPC Evolution

UE

eNodeB

Iot app

Edge Cloud

IoT group

communication

SDN

UE

eNodeB

MME

SGW

PGW

UE

eNodeB

IoT

Device

IoT

Device

IoT

Device

(a)

(b)

(c)

(d)
Use Case: Specialized EPC for IoT

Radio Access Network

- UE
- eNodeB

IP Substrate

Core Mobile Network

- MME
- SGW
- PGW

Hypothetical future architecture

- UE
- eNodeB
- SDN

IoT app

Edge Cloud

IoT group communication

Evolution
Use Cases

- 4G LTE/EPC
- Multiple parallel instances
- SCALE (MME replacement with a more elastic version)
- SMORE (selective low-latency offloading)
- MobiScud (selective offloading with PVMs)
- IoT-specific mobile architecture
Use Cases

- 4G LTE/EPC
- Multiple parallel instances
- SCALE (MME replacement with a more elastic version)
- SMORE (selective low-latency offloading)
- MobiScud (selective offloading with PVMs)
- IoT-specific mobile architecture
Proteus

• Proteus enables
Proteus

- Proteus enables
  - safe, rapid, dynamic service evolution
Proteus

- Proteus enables
  - safe, rapid, dynamic service evolution
  - multiple mobile service instances in parallel
Proteus

- Proteus enables
  - safe, rapid, dynamic service evolution
  - multiple mobile service instances in parallel
  - cloud-like mobile network abstractions
Proteus

Proteus enables
- safe, rapid, dynamic service evolution
- multiple mobile service instances in parallel
- cloud-like mobile network abstractions
- single mobile platform to be opened up to third-party providers
Evaluations

Topology in PhantomNet mobility testbed
Evaluations
Evaluations

• Service evolution and parallel instances
  • LTE/EPC
  • EPC evolved with SMORE
  • EPC evolved with MobiScud

• Data-centric service management
  • Handling bursts of UE attachments
  • Migrating to a new location

• Scalability
  • End-to-end orchestration request completion time
  • Orchestration time for individual components
  • End-to-end parallel orchestration request completion time
Evaluations

• Service evolution and parallel instances
  • LTE/EPC
  • EPC evolved with SMORE
  • EPC evolved with MobiScud

• Data-centric service management
  • Handling bursts of UE attachments
  • Migrating to a new location

• Scalability
  • End-to-end orchestration request completion time
  • Orchestration time for individual components
  • End-to-end parallel orchestration request completion time
Evaluations

Dynamic PGW state migration
- T0: PGW migration starts
- T1: PGW migration ends
Evaluations

Dynamic PGW state migration
- T0: PGW migration starts
- T1: PGW migration ends

![Chart showing RTT (ms) over Time (s) from T0 to T1]

- RTT values remain constant from T0 to T1.
Evaluations

Dynamic PGW state migration
- T0: PGW migration starts
- T1: PGW migration ends
Source Location: SF

Target Location: NY

RTT (ms)

Time (s)

T0

T1
Source Location: SF

Target Location: NY

RTT (ms)

Time (s)

Source Location: SF

Target Location: NY

X

T0

T1

RRT (ms)

Time (s)
Source Location: SF

Target Location: NY

RTT (ms)

Time (s)

Source

Location:

Target

Location:

T0

T1

R(TT (ms)

20

30

40

0  10  20  30

Time (s)
Conclusion

• Problem
  • Mobile network service deployment and evolution very slow
  • Future mobile networks will be unable to meet app demands without being more evolvable

• Our solution
  • Trend towards mobile software-defined infrastructure (SDI)
  • Proteus, a control platform that allows safe and rapid service creation and evolution in a mobile SDI
Questions?