A practical experience in designing an OpenFlow controller

Roberto Bifulco*
Marcus Brunner*¹
Roberto Canonico**
Peer Hasselmeyer*
Faisal Mir*

¹ Marcus Brunner is no more affiliated with NEC Lab Eu but this work has been done while he was an NEC employee

** Università di Napoli “Federico II”
From Special Purpose to General Purpose

Legacy Networks
- Today’s networks are defined by the boxes that compose them
- A single box includes hardware, operating system, applications
- Boxes are interconnected to provide specific network functions

Software Defined Networks
- Networks are defined by software programs
- Boxes are dummy but programmable
- The network is general purpose, network functions are implemented as applications on top of a Network OS
Programmable network

- Abstractions and Frameworks: Onix, FlowVisor, Trema
- Programming Languages: Frenetic, NetCore
- Operating Systems: NEC ProgrammableFlow, NOX, FloodLight, BEACON
- Machine Languages: OpenFlow
- Legacy interfaces
- DataPlane (Hardware/Software)

Tools:
- NICE
- ndb
- FlowChecker
- HSA
- Mininet
The need for Abstractions

We want to make the things **easier** and **faster**
We are looking for suitable **abstractions**

**Sharing** issues and findings to develop **experience**:
- we need to learn a lot
An old but new world

We are used in providing old network functions in the old way
We are exploring how to provide old network functions in the new way
We are exploring new ways of providing old network functions
We are exploring new network functions enabled by the new way
Our Experience

Implementation of the **Follow-Me Cloud (FMC)** Controller at **NEC Lab Europe**


**Scalability of a Mobile Cloud Management System**

R. Bifulco, M. Brunner, R. Canonico, P. Hasselmeyer, and F. Mir, "**Analysis of the Handover Procedure in Follow-Me Cloud**" to be presented at CloudNet 2012 poster session.
What is FMC
What is FMC

A

IPa

IPb

B

IPa

A

IPa

B

Empowered by Innovation
Observations

Functions coexistence
- FMC and may be implemented together with other functions (e.g., Ethernet switching) on the same switch

Hierarchical network handling
- FMC related functions require a global network visibility
- Ethernet related functions require a local network visibility

Geographical network
- A high number of networks/switches to manage
- Need to support scalability
Design: object oriented data model

Network global data:
- General configurations (IP address range, gateway, etc.)

Switch data:
- Datapath id
- Ports data
- Flow table
- Flow counters
Design: control logic into model objects

Exploiting the **Object Oriented** paradigm, we can provide **extensibility** through **inheritance**, but...

- OFSwitch
  - Control logic:
    - Flow table entries handling
    - Packet in processing
    - Dynamic processing state
    - ...

- LearningSwitch
  - Eth Learning Switch Control logic:
    - ARP handling
    - MAC based FTEs
    - MAC-Port table
    - ...

- FMCSwitch
  - FMC Control logic:
    - IP address “migration” function
    - Locator addresses handling
    - Advanced ARP handling

...there is a paradigm **mismatch** between **OO** programming and **OF** programming

More details on this in the paper.
Functions separation and distribution

Controller

Network A

Learning Switch
Learning Switch
Learning Switch
Learning Switch

Network B

FMC Switch
Learning Switch
Learning Switch
Learning Switch

A

B
Hierarchical control plane

Global Level (e.g., all operator’s networks)

Domain Level (e.g., single network)

Local Level (e.g., single switch)
Hierarchical control plane

North-bound Interface (E.g., for external systems)

Global Level
(e.g., all operator’s networks)

Domain Level
(e.g., single network)

Learning Switch

Local Level

A

B

Network
Hierarchical control plane

North-bound Interface (E.g., for external systems)

Global Level
(e.g., all operator’s networks)

Domain Level
(e.g., single network)

Local Level
(e.g., single switch)
Scalability: distribution of networks
Scalability: distribution of switches
Conclusions

Applying **SDN** (and OpenFlow) in different contexts **uncover** new **issues** and **possibilities**

**Sharing** our **knowledge** is important: the (SDN) world is big!

In FMC, we applied **OO modeling** for an OF **controller design** and our conclusions are:

- It is **good** to organize the network **data**
- It is **good** to **structure** the application for **scalability**
- It is **not so good** in helping the **extension** of available network functions: the process is still tricky