Network Function Virtualization & Software Defined Networking

Rencontres Inria Industrie « Télécoms du futur »
13 novembre 2014

Walid Dabbous
Outline

• Network Functions Virtualization

• Software Defined Networks

• NFV & SDN interplay
Traditional network model

- Network functionalities are **based on specific HW&SW**
- One physical node per role
Motivation

Problem Statement

• Complex carrier networks
  – with a large variety of proprietary nodes and hardware appliances.

• Launching new services is difficult and takes too long
  – Space and power to accommodate
  – requires just another variety of box, which needs to be integrated.

• Operation is expensive
  – Rapidly reach end of life
  – due to existing procure-design-integrate-deploy cycle.
Hardware vs. Software

**Telco Cycle**
- Idea !!
- Telco Operators
- Equipment Vendors
- SDOs

- Demand
- Drive
- Critical mass of supporters
- Standardise
- Implement
- Sell
- Deploy

2-6 Years

**Service Providers Cycle**
- Idea !!
- Service Providers

- Develop
- Deploy
- Publish

2-6 Months

Source: Adapted from Diego Lopez Telefonica I+D, NFV
Trends in IT

- Commodity x86
- Convergence
- Virtualization
- Cloud
- Mobility

Telco Challenges

- Huge capital investment
- Disparity between costs and revenues
- Complexity
- Reduced hardware life cycles
- Lack of flexibility and agility
- Launching new services is difficult and takes too long.

Source: Adapted from D. Lopez Telefonica I+D, NFV
The NFV Concept

A means to make the network more flexible and simple by minimising dependence on HW constraints

**Traditional Network Model:**
- **APPLIANCE APPROACH**
- Network Functions are based on specific HW&SW
- One physical node per role

**Virtualised Network Model:**
- **VIRTUAL APPLIANCE APPROACH**
- Network Functions are **SW-based over well-known HW**
- Multiple roles over same HW

Source: Adapted from D. Lopez Telefonica I+D, NFV
• Fragmented non-commodity hardware.
• Physical install per appliance per site.
• Hardware development large barrier to entry for new vendors, constraining innovation & competition.

Source: NFV
Benefits & Promises of NFV

• Network Functions Virtualization is about implementing network functions in software
  – that today run on proprietary hardware
  – leveraging commodity hardware and IT virtualization

• Provides opportunities for pure software players
  – Facilitates innovation
  – Faster time to market

• Supports multi-versioning and multi-tenancy of network functions, which allows use of a single physical platform for different applications, users and tenants

Source: Adapted from D. Lopez Telefonica I+D, NFV
Benefits & Promises of NFV

• Targeted service introduction based on geography or customer sets
  – More service differentiation & customization
• Services can be rapidly scaled up/down as required.
• IT-oriented skillset and talent
• Reduced equipment costs (CAPEX)
  – through consolidating equipment and economies of scale of IT industry.

Source: NFV
Benefits & Promises of NFV

• Improved operational efficiency
  • by taking advantage of the higher uniformity of the physical network platform and its homogeneity to other support platforms.

• Reduced (OPEX) operational costs: reduced power, reduced space, improved network monitoring

Source: Adapted from D. Lopez Telefonica I+D, NFV
Use cases

Complex home environment

- Home environment
- Network environment

Home simplification

- Simplification or even suppression (STB)
- No need for home router replacement as it is updated by configuration
- Fast deployment for new services
- Inexpensive IPv6 migration maintaining legacy home routers

Multiple IP Edges

- An IP Edge for each service (voice, video content, Internet)
- Scattered and not well integrated control functions (e.g. DPI, BRAS, PCRF)

Virtual IP Edge

Virtualisation control

A unified software IP Edge

- SW-based BRAS
- HW pool management
- SW-based CG NAT
Other use cases

- All the network concentrated in the base station

- C-RAN: All the base station functionalities, except for the antennas and power amplifiers, concentrated in a centralized location

- Having the flexibility of moving functionalities between different locations may help to network to adopt the best option in each case

- Leverage on open source routing project as rich and widely tested protocol suite while assuring data plane performance
  - Common routing protocols supported and extended by open source project
  - High-performance line-rate data plane
  - Running in separate process, does not lead to licensing issues

Source: Adapted from D. Lopez Telefonica I+D, NFV
Many others...

- **Switching elements**: BNG, CG-NAT, routers.
- **Mobile network nodes**: HLR/HSS, MME, SGSN, GGSN/PDN-GW.
- **Home networks**: Functions contained in home routers and set top boxes to create virtualised home environments.
- **Tunnelling gateway elements**: IPSec/SSL VPN gateways.
- **Traffic analysis**: DPI, QoE measurement.
- **Service Assurance**: SLA monitoring, Test and Diagnostics.
- **NGN signalling**: SBCs, IMS.
- **Converged and network-wide functions**: AAA servers, policy control and charging platforms.
- **Application-level optimisation**: CDNs, Cache Servers, Load Balancers, Application Accelerators.
- **Security functions**: Firewalls, virus scanners, intrusion detection systems, spam protection.

Source: NFV
Rethinking relayering

- applications
  - operating systems
  - hypervisors
  - compute infrastructure
  - network infrastructure
  - switching infrastructure
  - rack, cable, power, cooling

- applications
  - network functions
  - operating systems
  - hypervisors
  - compute infrastructure
  - switching infrastructure
  - rack, cable, power, cooling
Traditional Computer Networks

Data plane:
Packet streaming

Forward, filter, buffer, mark, rate-limit, and measure packets
Traditional Computer Networks

Control plane:
Distributed algorithms

Track topology changes, compute routes, install forwarding rules
Traditional Computer Networks

Management plane:
Human time scale

Collect measurements and configure the equipment
Software Defined Networking (SDN)

Decoupled control

API to the data plane (e.g., OpenFlow)

Switches
Software Defined Networking (SDN)

Decoupled control

Smart, slow

API to the data plane (e.g., OpenFlow)

Switches
Software Defined Networking (SDN)

Decoupled control

API to the data plane (e.g., OpenFlow)

Smart, slow

Dumb, fast

Switches
Software Defined Network (SDN)

Abstract Network View

Network Virtualization

Global Network View

Network OS

Packet Forwarding

Control Programs

f(View)

Packet Forwarding
Software Defined Network (SDN)

```c
firewall.c
...
if (pkt->tcp->dport == 22)
    dropPacket(pkt);
...
```
Software Defined Network (SDN)

\[ f(\text{View}) \]

Control Programs

Network Virtualization

Global Network View

Network OS

Packet Forwarding

firewall.c

...  
if( pkt->tcp->dport == 22)  
dropPacket(pkt);  
...
Software Defined Network (SDN)

f(View)

Control Programs

Network Virtualization

Global Network View

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

Packet Forwarding

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. ...
7. ...

firewall.c

... 

if (pkt->tcp->dport == 22) 
  dropPacket(pkt);
...

NFV and SDN

- NFV: re-definition of network equipment architecture
- NFV was born to meet Service Provider (SP) needs:
  - Lower CAPEX by reducing/eliminating proprietary hardware
  - Consolidate multiple network functions onto industry standard platforms
- SDN: re-definition of network architecture
- SDN comes from the IT world:
  - Separate the data and control layers, while centralizing the control
  - Deliver the ability to program network behavior using well-defined interfaces
Manage router service today
Managed router using NFV
Managed router using NFV & SDN
Service Fonctions Chaining (SFC)

• Set of network services, such as firewalls or application delivery controllers interconnected through the network to support an application
• In the past
  – Required specialized hardware & (re)configuration
  – Unnecessary bandwidth and CPU consumption
• NFV/SDN enables to quickly and inexpensively create, modify and remove service chains.
  – entire service chains can be provisioned and constantly reconfigured from the controller
  – No hardware purchase; NF execute as elastic VM
Service Chaining for NW Function Selection in Carrier Networks

**Current IP routing**
Current IP routing cannot easily control and configure packet forwarding per user.

**Proposed use case**
Controller adapted to NTT’s service chaining method

- **vDPI**: CSR 1000v (Cisco Systems)
- **vCPE**: VSR1000 (Hewlett-Packard)
- **vFW**: FireFly (Juniper Networks)
- VIM (NW Controller): Service Chaining Function (prototype) + Ryu (NTT)

Source: ETSI Ongoing PoC
NFV challenges

• Achieving high performance virtualised network appliance
  – portable between different HW vendors, and with different hypervisors

• Co-existence with bespoke HW based network platforms
  – enabling efficient migration paths to fully virtualised network platforms

• Management and orchestration of virtual network appliances
  – ensuring security from attack and misconfiguration

• NFV will only scale if all of the functions can be automated

• Appropriate level of resilience to HW and SW failures

• Integrating multiple virtual appliances from different vendors
  – Network operators need to be able to “mix & match” HW
  – hypervisors and virtual appliances from different vendors
  – without incurring significant integration costs and avoiding lock-in
SDN challenges

- Heterogeneous Switches
- Controller Delay and Overhead
- Distributed Controller
- Testing and Debugging
- Programming Abstractions
DISCO: Distributed SDN Controllers for rich and elastic services

- ANR INFRA 2013
- Industrial project
- Starting date: 1st January 2014
- Duration: 30 months

Partners:
- Thales Communications & Security (coordinator)
- INRIA Sophia Antipolis (DIANA research team)
- ENS Lyon (DANTE research team)
- 6WIND
DISCO’s use case and Next generation SDN

- **Use cases**
  - Telco offers Video on Demand (VoD), medical data sharing system...

- **Deployment context**
  
  
  Heterogeneous and Time-varying resources (resources / cost / failures)

  Enterprise WAN
  Mission Critical WAN
  Private Operated WAN

  Paris Datacenter
  London Datacenter
  New York Datacenter
  Micro Datacenter
DISCO’s Challenges
DISCO’s Challenges

Next generation data plane
• New scalable appliances for DPI and caching
• Dynamic allocation of cores between applications and network services
• Optimization of the dynamic placement of network appliances in SDN-based infrastructures
DISCO’s Challenges

Next generation data plane
- New scalable appliances for DPI and caching
- Dynamic allocation of cores between applications and network services
- Optimization of the dynamic placement of network appliances in SDN-based infrastructures

SDN Control Plane
- Resilient control plane for more robustness, scalability and flexibility
- Network-wide distributed and multiclass admission control
- Extensions of the controlling capabilities both for monitoring and configuration of the data plane
DISCO’s Challenges

Next generation data plane
- New scalable appliances for DPI and caching
- Dynamic allocation of cores between applications and network services
- Optimization of the dynamic placement of network appliances in SDN-based infrastructures

Network as a Service
- Efficient content distribution mechanism leveraging dynamic resource allocation
- A NaaS API to allow network applications to benefit from dynamic core and network appliance allocation and extended monitoring capabilities

SDN Control Plane
- Resilient control plane for more robustness, scalability and flexibility
- Network-wide distributed and multiclass admission control
- Extensions of the controlling capabilities both for monitoring and configuration of the data plane

DISCO’s Challenges

Network as a Service
- Efficient content distribution mechanism leveraging dynamic resource allocation
- A NaaS API to allow network applications to benefit from dynamic core and network appliance allocation and extended monitoring capabilities

SDN Control Plane
- Resilient control plane for more robustness, scalability and flexibility
- Network-wide distributed and multiclass admission control
- Extensions of the controlling capabilities both for monitoring and configuration of the data plane

Next generation data plane
- New scalable appliances for DPI and caching
- Dynamic allocation of cores between applications and network services
- Optimization of the dynamic placement of network appliances in SDN-based infrastructures

DISCO’s Challenges

Network as a Service
- Efficient content distribution mechanism leveraging dynamic resource allocation
- A NaaS API to allow network applications to benefit from dynamic core and network appliance allocation and extended monitoring capabilities

SDN Control Plane
- Resilient control plane for more robustness, scalability and flexibility
- Network-wide distributed and multiclass admission control
- Extensions of the controlling capabilities both for monitoring and configuration of the data plane

Next generation data plane
- New scalable appliances for DPI and caching
- Dynamic allocation of cores between applications and network services
- Optimization of the dynamic placement of network appliances in SDN-based infrastructures
REFLEXION REsilient and FLEXible Infrastructure for Open Networking

- ANR INFRA 2014
- Industrial project
- Starting date: 2015
- Duration: 24 months

- Partners:
  - Thales Communications & Security (coordinator)
  - INRIA Sophia Antipolis (DIANA research team)
  - ENS Lyon (DANTE research team)
  - 6WIND
  - Orange
  - UPMC
  - TPT
Fault and disruption management for virtualized services

Robust and scalable control plane for next generation SDN

Dynamic performance management of low level resources in SDN/NFV environments

Distribution and optimization of virtual network functions in SDN environments
Take home messages

• SDN is about network programmability

• NFV is about software network functions

• Current vendors/operators should adapt to software development cycles
Questions?