Virtualization techniques for network functions

Fabrice Guillemin, Orange Labs, OLN/CNC/NCA
September 14, 2015
Introduction

- Network functions are today hosted by dedicated hardware, typically high performance servers
  - Evolved Packet Core function (HSS, MME, etc.) for mobile networks
  - Broadband access servers for fixed networks
- Advantages:
  - high performance and reliability
  - centralized functions easier to monitor
  - the equipment provider for assistance
- Drawbacks
  - vulnerable points in the architecture (cf. HLR breakdown in Orange mobile network)
  - lack of flexibility (a few functions per server)
  - dependence on the equipment provider for upgrade (extra costs)
- Virtualization: a possible evolution to make the architecture more flexible and less expensive (?)
Virtualization today and further possibilities for operators

- Many initiatives for network function virtualization (ETSI NFV, commercial solutions by many equipment/software providers:
  - virtual Evolved Packet Core (vEPC)
  - virtualized IMS
  - RAN sharing
  - virtualized monitoring
  - …

- Target applications:
  - network function packages “hosted in the cloud”: an IMS core or a vEPC dedicated to a client (e.g., a company)
  - RAN sharing between several (virtual) mobile operators
  - …

- Solutions are provided by “usual” equipment manufacturers
  - software testing and bug reporting … 😞
  - … but virtuous iteration loops

- A more challenging approach: open source solutions
An example: The Convergent Gateway (under development by BCOM)

Backhaul all access technologies (fixed, WiFi, cellular) through the same gateway all-IP design

- IP traffic, no GI functions (e.g., DPI) services in OTT mode
- pure IP, IP/MPLS, L2 or L1 possible colocation with NGPoPs
- need for a connection between addresses and AP

same address pool for the APs connected to the CGW
The functional blocks of the CGW

All the functions are hosted by virtual machines or containers (NFV) and are based on open source software.

- DHCP Server: address allocation
- AAA: authentication
- MME (LISP PxTR): mobility management
- L-ANDSF\(^1\): choice of the best AP
- Monitoring: supervision of the AP
- vEPC: termination of GTP tunnels
- WiFi controller: Control of the WiFi AP

\(^1\)ANDSF (Access Network Detection and Selection function): standardized by 3GPP (centralized version), assists the terminal to select the best AN depending upon user’s subscription (WiFi offloading)
Some critical issues in the design of virtualization

- vEPC used by BCOM: OpenAir Interface
  - bugs have to be progressively fixed (to be compliant with 3GPP spec)
  - traffic goes through VM – because of GTP tunnels 😞
- Next steps (proposed for 5G): separate control and data planes in cellular networks
  - GTP-U should be handled by the data plane only
  - GTP-C should be hosted by separate VMs
  - need for a convergence function in the data plane and possibly for an interface to configure the data plane
- Same problems with WiFi controllers (openCAPWAP)
- Extend to BBU (and BBU hostelling) when CGW are collocated with NGPoP (hosting OLT and BBU functions)

Need for neatly delineating control plane (GTP-C termination, access control, etc.) and data plane (switching, packet dropping, etc.) functions
Ideal design/cloud

which API? (OF is too poor)

cloud (centralized data centers)

more than 200 NGPoP in Orange/France

NGPoP

WiFi Hotspot (public or private)

eNodeB

eNodeB
Ideal design/fog computing

NGPoP enriched with CDN and other services (TURN servers, etc.) in the fog
Global view of the network

Data centers in fog computing facilities are smaller than those in clouds.

IP collect Network

- WiFi Hotspot (public or private)
- eNodeB
- NGPoP

Fog computing facility
Back to the CGW: “on the fly” instantiation

- CGWs should be instantiated on the fly on data centers (OpenStack) distributed at network edges (fog computing)
  - there is hence a need for a tool capable of configuring such elements (typically OpenStack)
  - But CGWs should be interconnected, sometimes with bandwidth constraints
  - OpenStack is not sufficient by itself, there is a need for a tool able to configure the network in order to interconnect CGW (e.g., OpenDaylight), typically when used to backhaul an enterprise network

- OpenDayLight and Openstack have been developed for given purposes, there is a need for a tool with a global view of the network in terms of storage, computing and bandwidth

➡ GlobalOS under study at OrangeLabs
Configuration of a CGW (OpenStack – centralized view)

Openstack can control and configure a CGW from the edge (nothing in the network)
Configuration of a CGW (OpenStack + ODL)
Configuration of CGW (orchestration)
Global OS: first view

- Services
- Security
- ...

Orchestration by using network abstraction (topology, resources, etc.), scheduling, etc.

Openstack

OpenDayLight

API

(network programming language)

Data center

Configuration and monitoring (openflow, netconf, etc.)
Basic questions to be solved (for network valorization)

Resource allocation

- the network (Global OS) receives requests either from services or for internal usage (e.g., on the fly deployment of convergent gateways)
- requests require: bandwidth, storage and computing resources
- request can be decomposed as a set of elementary functions which can be instantiated on several VM (or containers) – correlated allocation of resources, chaining
- large number of data centers with limited capacities (when fog computing)
- requests are of different types:
  - assignment of tasks
  - some tasks can be scheduled (e.g., book ahead reservations)

optimization vs. stochastic analysis (complexity of resource demand, large number of facilities)
Network programming language

- Very active domain of research
- Many languages have been proposed so far
  - static approach
    - NetKAT (Kleen Algebra with Test): express network procedures into an equational system (for formally proving properties of procedures)
    - NICE
    - MERLIN
  - dynamic approach
    - Kinetic
    - VeryFlow
- Most languages are packet based and do not include resource allocation aspects (except Merlin)

scaling aspect: more than 1000 – 2000 nodes to configure
Thank you!