

SURFnet programmable network

ESnet SDN Workshop, July 14, 2015

Hans Trompert
SURFnet



From monolithic

- Hardware + NMS + supporting applications often from one vendor
- Reimplementing much of the same functionality going from one to the next generation network:
 - IP, point to point and multipoint services
 - Monitoring, capacity management and reporting
- In many cases only could reuse the customer portals
- Did develop multi domain BoD service (but slowly)
- Network functions only as add-on
- Little room for innovation

To layered

- Separate functions into well defined layers
- Innovate each layer independently from other layers
- Different and multiple vendors for each layer
- Open Standards APIs
- No more big-bang new network but continuously evolve network:
 - Only change small parts
 - Reuse the rest
- Leaves more time to actually innovate

So many choices ...

Network hardware:

- Traditional forwarding with switches and routers
- White label OpenFlow switches
- SDN 2.0 with traditional forwarding core and service aware (SDN) edge
- Programmable data plane (flexpipe, NetFPGA, ..)
- ...

So many choices ...

Network Operating System:

- Traditional NMS (usually ships with hardware vendor)
- Device independent scripting (e.g. NCS from Tail-F, ...)
- SDN controller (ODL, ONOS, ...)
- Combination of the above?
- ...

So many choices ...

Orchestration and integration:

- NFVI and NFV-MANO (OPNFV, ...)
- Cloud integration (OpenStack, ...)
- Multi domain services (NSI CS 2.0, CoCo, MD-VPN ...)
- ...

Software

- Most functionality is moving up into software
- Enables rapid innovation
- Enables easier automation of all kind of functions
- Writing good software is difficult and resource intensive
- Probably only be able to code 5% ourselves
- Need large open source communities around software solutions
- Distributed state full databases (because there are still functions that need it)

Network abstractions

What do end users want from the network?

- It is always available
- Have plenty of available bandwidth all the time
- It just is invisible

So develop new network abstractions for:

- non-network service providers
 - Compute/VM, storage, VOIP, ...
- Software developers
 - workflow and other applications
- Operating Systems
- Multi domain network services

Next steps

- Identify and describe the layers needed
- Develop a strategy to migrate to a layered setup
- Demand Open Standard APIs from vendors
- Automate as much of the network functionality as possible (especially on lower layers)
- Come up with new and better higher level network abstractions
- Integrate NFV in the complete stack
- Aim the (intent based) APIs mostly at non-network service providers and software developers
- Find and contribute to open source communities



`hans.trompert[at]surfnet.nl`



hanstrompert



`www.surfnet.nl`



Creative Commons “Attribution” license:
<http://creativecommons.org/licenses/by/3.0/>

