Intel ONP - Intel Open Network Platform

Lionel Auroux

2016-07-15

Intel ONP, SDN, NFV, . . .

- A lot of acronym
- A lot of technology layers
- Seems complex

SDN?

Software-Defined Networking

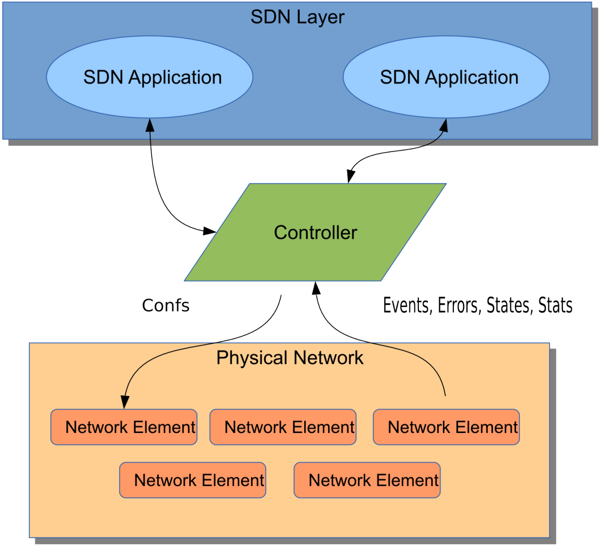
- A Concept
  - Abstraction of lower-level network functionality
- Unify control
  - route packet sequences (flows)
  - logically centralized but physically distributed
- Open Standards
  - protocols, API

Why?

- Network requirement of application
  - non-SDN: Implicit or Indirect (need human processing)
  - SDN: Application could declare networks requirements (Qos)
- Network monitoring and adaptation:
  - Throughput, Delay... insurance for applications

Important informations are exchanged and used for adapting network to requirements...

Overview


SDN Layer
SDN Application
SDN Application
Controller
Confs
Events, Errors, States, Stats
Physical Network
Network Element
Network Element
Network Element
Network Element
Network Element

Which functionnalities?

- Basically Network Elements are:
  - L2: Switch
  - L2.5: MPLS...
  - L3: Routers
  - L4-7: WAN accel, IDS/IPS
- But also more complex elements

Notes

- L2: Data Link (ethernet...)
- L3: Network (IP, ARP, ICMP)
- L4: Transport (TCP, UDP)
- L4-7: Application (HTTP, SSH, DHCP, SNMP...)
- L2.5: Just to say f.ck to OSI

Is that real?

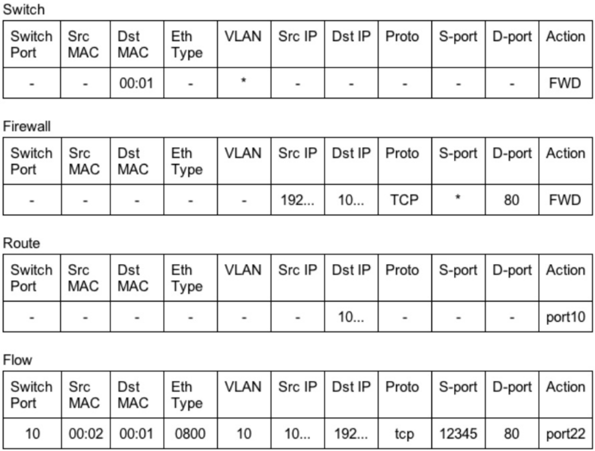
- Check open communities that provide SDN support:
  - OpenDaylight - provides software/standard (like openflow)
- Check support of hardware. . .
  - OpenFlow - Standard protocol for switch/router

OpenDaylight

- Java base code for Network Element
- Implements SDN open standards
- A complex ecosystem of product & vendors. . .
    - OpenDaylight is a part of Intel ONP

Open Flow

- A standard Way to send “forwarding” table to hardware



Switch

Switch Port	Src MAC	Dst MAC	Eth Type	VLAN	Src IP	Dst IP	Proto	S-port	D-port	Action
-	-	00:01	-	*	-	-	-	-	-	FWD

Firewall

Switch Port	Src MAC	Dst MAC	Eth Type	VLAN	Src IP	Dst IP	Proto	S-port	D-port	Action
-	-	-	-	-	192...	10...	TCP	*	80	FWD

Route

Switch Port	Src MAC	Dst MAC	Eth Type	VLAN	Src IP	Dst IP	Proto	S-port	D-port	Action
-	-	-	-	-	-	10...	-	-	-	port10

Flow

Switch Port	Src MAC	Dst MAC	Eth Type	VLAN	Src IP	Dst IP	Proto	S-port	D-port	Action
10	00:02	00:01	0800	10	10...	192...	tcp	12345	80	port22

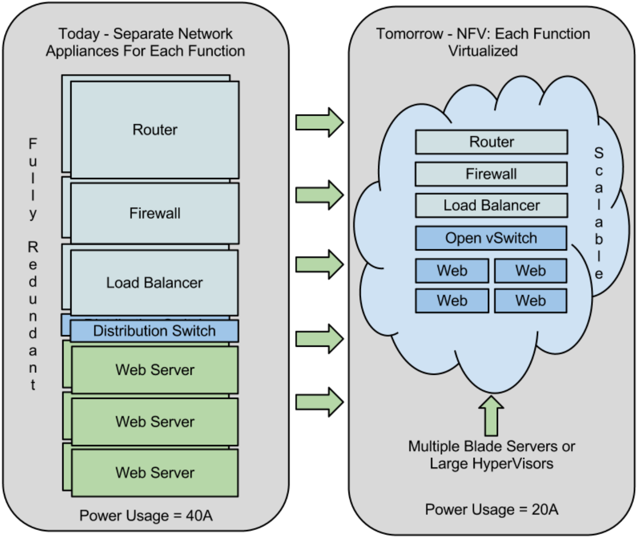
NFV?

Network Function Virtualization

Motto

Any service able to be delivered on proprietary, application specific hardware should be able to be done on virtual machine.

overview


Today - Separate Network Appliances For Each Function
F u l l y R e d u n d a n t
Router
Firewall
Load Balancer
Distribution Switch
Web Server
Web Server
Web Server
Power Usage = 40A
Tomorrow - NFV: Each Function Virtualized
Router
Firewall
Load Balancer
Open vSwitch
Web
Web
Web
Web
S c a l a b l e
Multiple Blade Servers or Large HyperVisors
Power Usage = 20A

Why?

- Virtualization Hardware/System support -> Performance
- Cost:
  - Fit consumption needs
  - Economy of scale

Need to address a lot of problem

- Implement/provide open network functionnalities (OpenvSwitch)
- Hardware handling into VM

IO Passthrough

- Target system need to handle directly the hardware thru hypervisor (VT-d)
- concept of IO passthrough

Example: IO MMU / VFIO

- IO MMU allow to map IO MEM (base address of device registers) directly into VM memory.
- The linux kernel module VFIO handle and provide a clear API to do it.

SDN != NFV

- Attention, SDN and NFV are two different concepts.
- But they are complementary to achieve a full Open Network.
  - Full abstraction between hardware/software of all network component
  - Scalable, Flexible. . .
  - The better of two world

The Intel initiative

The Intel initiative

- Advances open source community work
- Extends product development
- Enables an open NFV/SDN ecosystem
- POC and trials collaboration

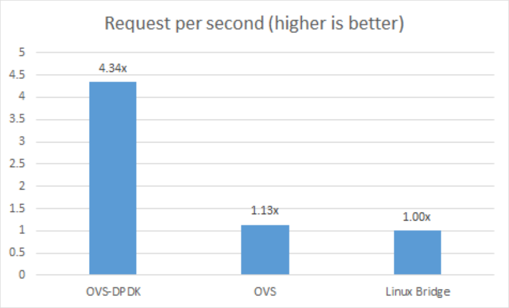
Intel ONP Server Stack

- OpenDaylight
- Open vSwitch
- OpenStack: For Cloud computing (Iaas)
- Linux/KVM: For Hypervisor
- DPDK: The little black sheep

DPDK

overview DPDK

How to reach performance pick with 10gb, 20gb, Xgb?



Using KVM kernel, e1000 driver for OpenvSwitch-with-DPDK vs OpenvSwitch vs Linux bridge

A virtual appliance with DPDK

A data plane is the part of network that carries user traffic.

Data-plane Development Kit?

A set of library tools to develop software for fast network treatment?

Is it SDN? Is it NFV? is both. . .

DPDK

We can implement network functionalities in Userland with best performance boost.

motto

DPDK uses IO passthru(to userland) & Big memory pool to boost user traffic carrier.

DPDK Requirements

- Use Huge chunk of memory

```
mkdir -p /mnt/huge
mount -t hugetlbfs nodev /mnt/huge
d1="/sys/devices/system"
d2="/node/node0/hugepages"
d3="/hugepages-2048kB/nr_hugepages"
echo 64 > $d1$d2$d3
```

DPDK Requirements

- Need specific driver for NICs (but provided)
- Some hacking skills and patience: compiling “stable” 16.04 with T=x86_64-native-linuxapp-gcc
  2 errors about brace in...
  build/build/lib/librte_eal/linuxapp/kni/ixgbe_82599.c
  build/build/lib/librte_eal/linuxapp/kni/e1000_phy.c

DPDK Programming

A clean API:

- Environment Abstraction Layer
- Lot of helpers libs: ring, mempool, mbuf, poll, crypto, timer, packet distributor, kernel NIC, Qos, ip fragmenting&reassembly, packet ACL

DPDK Conclusion

A lot of samples... But need concrete project: MPLS, BGP, OSPF...

It's just the beginning of the journey...

Questions?