Qualifying SDN/OpenFlow Enabled Networks

Dean Lee
Senior Director, Product Management
Ixia
Agenda

- SDN/NFV – a new paradigm shift and challenges
- Benchmarking SDN enabled Network
- Test Results
SDN (Software Defined Networking) Defined

- SDN allows a controller (server based software) to define how packets are forwarded by networking elements

- 3 distinct SDN features
  - Separation of the control plane from the data plane
  - A centralized controller and view of the network
  - Programmability of the network by external applications

SDN realization with OpenFlow
OpenFlow switch forwards packets. OpenFlow controller configures flows table in OF switch
SDN Implemented with Overlay Network

- Connectivity established across existing L2/L3 networks
- Examples include GRE, VxLAN, VPNs, PWs, LISP etc
- Divides physical infrastructure into multiple logical networks to support multi-tenants
SDN Impact: Network Transformation

- Application Layer: Customized network for better user experience
- Orchestration Layer: Inter-domain orchestration, open interfaces to support NaaS
- Network Control Layer: Global view, central control, network virtualization
- Data Transport Layer: Service/protocol-agnostic HW, open, standard interface

SDN Allows Applications to Intelligently Utilize the Network
What SDN Promises to Carriers?

- **Customization (Value add):** custom services, collaboration between applications and the network
- **Simple:** operation and management
- **Instant:** fast service provisioning
- **Elastic:** flexible evolution of infrastructure

Recent Service Provider SDN/NFV Survey Yielded (July 2013 – 21 operators):

1. By mid 2015 SDN/NFV will move to testing and development stage
2. Top 5 areas that will see NFV realization – Mobile Core/EPC, Virtual gateways, virtual routers, CDNs and IMS

Source: Infonetics Research
<table>
<thead>
<tr>
<th></th>
<th>ONOS (Open-Source SDN Stack)</th>
<th>Open-source controller: OpenDaylight</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBI</td>
<td>Controller SBI mainly uses OpenFlow</td>
<td>Controller SBI supports various types through SAL, including Openflow、BGP、PCEP、I2RS、SNMP、Netconf, etc.</td>
</tr>
<tr>
<td>Target Audience</td>
<td>Mainly for operators</td>
<td>Applicable to both data centers and service provider – Hydrogen Base/Virtualization/Service Provider versions</td>
</tr>
</tbody>
</table>
Carrier SDN Initiatives: SDO Efforts

Initiated NFV Industry Standards Group
- Network-operator-driven - Started by 13 Operators (VZ, DT, ATT..)
- Virtualization of Network Functions
- Complimentary to SDN and Open Innovation

Several Efforts related to SDN and smooth migration using existing NEs
- Programmatic Interfaces
- SDN Agents and Controllers
- Infrastructure Virtualization

User driven standardization organization
- Promote open standards for SDN
- Gatekeeper of ONF Protocol Specification
- Different working groups focusing on different aspects of SDN

Open Source framework for SDN platform under the umbrella of Linux Foundation
- To foster the development of open source applications
- Vendors have committed large dollar amount and developers to this effort
SDN Evolution Challenges

- Smooth Migration from current networks toward SDN
  - Significant installed base of existing carrier networks
  - Coexistence during migration
  - Evolution versus revolution

- Reliability and scalability of centralized controllers
  - Expose much higher risk than distributed control planes
  - Fast recovery from data path failures
  - Supporting very large carrier networks

- Flexibility versus Performance
  - Software flexibility and performance rely on hardware capability
  - Finding the correct hardware trade-offs

- Lack of robust testing methodologies for validating various SDN implementations
Why Benchmarking SDN is Important

**Deployment**
- Benchmark against legacy
- Interworking with legacy
- Easy provisioning

**Monitoring**
- End to End Performance
- Analytics
- New SDN infrastructure

**Operational**
- Configuration accuracy and performance
- SDN orchestration scale
- Integration/Migration from current OSS/BSS

---

**Key SDN Barriers**
- Incomplete or non-existent standards: 62%
- Finding SDN knowledgeable engineers: 62%
- Unclear cost-benefits tradeoffs: 62%
- Immature technologies: 57%
- Deploying SDNs in existing networks: 38%
- How to prioritize SDN projects: 38%
- SDNs across multiple vendors’ equipment: 33%
- SDNs across multiple layers: 33%
- No integration of OpenFlow and MPLS: 29%
- SDN software controller can slow high capacity traffic: 29%
- SDNs access and aggregation networks: 24%
- Testing of SDN services in existing network: 24%
- Hybrid SDN: 19%

*Source: Infonetics Research*
SDN Benchmarking Technologies
Benchmarking SDN

Target Areas

- Sizing Controller performance
- Network scale, reaction to network events, comparing various SBI protocols for optimum customization for a routed versus switched network
- Forwarding Plane performance and scale
- SDN enabled network convergence
- Service provisioning latency of new services
- Effective comparison based on multiple SBI protocols
- Impact of unplanned network events in forwarding and control plane, reachability of north and south bound targets
Forwarding Plane Validation - OpenFlow

Application & Orchestration

OpenFlow Controller

Forwarding throughput
Forwarding latency
Flow tables capacity
FIB convergence time

Openflow、BGP、PCEP、I2RS、SNMP、Netconf,
Network Validation - Overlay

Forwarding throughput
Forwarding latency
Tunnels capacity

VXLAN, NVGRE, MPLS
Controller Validation

Application & Orchestration

Controller

Physical Network

Virtual Infrastructure

NB API

SB API
Protocols conformance
Interoperability
Scalability to handle switches
Flows table capacity
Failover & high availability
Virtualized Infrastructure Simulation

Emulates
- Hypervisor/Host including VM Manager
- ESX, KVM
- Virtual Machines
- IPv4, IPv6 with traffic integration
- VM Events
- Start, Stop, Migrate

Validates
- Measure VM migration convergence time
- Performance of SDN in dealing with massive virtualized infrastructure
Infrastructure Validation – Benchmark Performance

RFC benchmark tests
Service interruption time

Application & Orchestration

Controller

Physical Network

Virtual Infrastructure

NB API

SB API
Sample Test Results
**Objective** - This test measures the convergence time taken by an implementation to service the change of flows table.

- **Test steps** –
  - OF controller installs two flows reaching site B. Flow 1 has better attribute than flow 2.
  - CE1 sends traffic toward customer site B. Traffic is forwarded via flow 1 (CE2).
  - OF controller withdraw flow 1 at “Flow-Withdraw-Event-Time”.
  - CE3 receives traffic at “Flow-Switch-Time”.
Test Results

- Repeatable test methodology for black box characterization
- Similar test was done with BGP router with over 200ms convergence time

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Flows Withdraw Conv (s)</th>
<th>%Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.384</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1.369</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1.337</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1.356</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1.325</td>
<td>0</td>
</tr>
<tr>
<td>Avg.</td>
<td>1.354</td>
<td></td>
</tr>
<tr>
<td>Dev.</td>
<td>0.019</td>
<td></td>
</tr>
</tbody>
</table>

DUT is a first generation OF 1.0 switch
1000 IP flows

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Flows Withdraw Conv (ms)</th>
<th>%Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>553</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>569</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>477</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>614</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>608</td>
<td>0</td>
</tr>
<tr>
<td>Avg.</td>
<td>564</td>
<td></td>
</tr>
<tr>
<td>Dev.</td>
<td>38</td>
<td></td>
</tr>
</tbody>
</table>

DUT is an OF 1.3 switch
1000 IP flows
Flows Capacity Test

Objective - This test measures the total flow table capacity of OF switch

Test steps –
- OF controller installs n flows in DUT
- OF controller validates the flows by retrieving flow status from DUT
- Validates the reachability of installed flows by sending test packets
- Repeat the test with additional amount of n flows until the validation fails, record flow #
Test Result

Flows status retrieved from DUT

Flows added to DUT
Thank You