BNV: Enabling Scalable Experimentation through Bare-metal Network Virtualization

Pravein Govindan Kannan, Ahmad Soltani, Mun Choon Chan Ee-Chien Chang
No Single Topology is suitable for all use cases

How to validate ideas over multiple topologies?
Mininet, Maxinet: Elegant tools enabling functional testing/learning.

.Container-based virtualization on a single machine or cluster of machines.

Cannot achieve the peak line-rate for performance testing.
Network Testbeds – High Fidelity

DeterLab, CloudLab, GENI, Staging Platforms:

- Can provide high fidelity emulation with Bare-Metal provisioning.
- Can provide only a fixed topology or subset for Software Defined Networks.
- Change in Topology might require a long time (sometimes in order of weeks).
How do we achieve the flexibility of software and fidelity of hardware (bare-metal)?

**Topology Scaling:**
One-to-Many Switch Virtualization
Virtual Topology

Switch1

Network Hypervisor

Tenant 1
OpenDaylight Controller

Tenant 2
ONOS Controller

Tenant 3
POX Controller

Tenant n
Any SDN Controller

FatTree, JellyFish, HyperCube, HyperX, Xpander, Arbitrary topologies... etc

Tenant Mapping

OpenFlow

Switch1

Switch2

Switch3
Virtual Topology

Switch1

Switch2

Switch3

BNV

OpenFlow

Flow Translation

Topology Abstraction

Tenant 2

ONOS Controller

Tenant 3

POX Controller

Tenant 1

OpenDaylight Controller

Tenant n

Any SDN Controller

FatTree, JellyFish, HyperCube, HyperX, Xpander, Arbitrary topologies...etc

Tenant Mapping

Virtual Topology

BNV Mapper

BNV

Tenant 1

Tenant 2

Tenant 3

Tenant n
Flow Translation

Topology Abstraction

Flow Translation
Idea: Loopback links in substrate topology
Idea: Loopback links in substrate topology
Idea: Loopback links in substrate topology
Software-Configurable Loopbacks

ToR SDN Switch 1

Core N/W

ToR SDN Switch N

L2 Switch 1

... 

L2 Switch N

Rack 1

Rack N
Topology + Loopback link

```
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoopLink1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>H5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoopLink2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CLink1 10G  Backbone link
```
Physical Topology

Virtual Topology

---

**Virtual Topology**

**Physical Topology**
Physical Topology

Virtual Topology

V1

1

2

3

vS1

vL1 -&gt; CLink1

vL3

vL2

vS2

vS3

V3

pSwitch1

1

2

3

4

pSwitch2

1

2

3

4

CLink1

H1

H2

LoopLink1

H5

H6

LoopLink2
Physical Topology

Virtual Topology
Final Mapping

Virtual Topology

Physical Topology

Final Mapping

Virtual Topology

Physical Topology
Physical Topology

Virtual Topology
Consistency & Fidelity of mapping.

Virtual Topology

Physical Topology
How to achieve buffer isolation between multiple virtual switches?

A single virtual switch can fill up the entire shared Buffer!!
How to achieve buffer isolation between multiple virtual switches?

Burst Traffic

S1

Metering burst
(burst-rate < Approx Buffer Size)

S2

S3

Shared Buffer
How to achieve link isolation between multiple virtual links?
Virtual Topology

Tenant 1
OpenDaylight Controller

Tenant 2
ONOS Controller

Tenant 3
POX Controller

Tenant n
Any SDN Controller

FatTree, JellyFish, HyperCube, HyperX, Xpander, Arbitrary topologies...etc

BNV Mapper

Topology Abstraction

Flow Translation

BNV

Virtual Topology

Tenant Mapping

OpenFlow

Switch1

Switch2

Switch3
BNV Mapper
BNV Mapper

• A Virtual Topology Consists of:
  • A set of hosts $V$,
  • A set of (virtual) switches $S$, (feature: TCAM capacity, meter capacity, etc)
  • A set of (virtual) links $L$ (feature: Bandwidth)
BNV Mapper

• Solved using ILP
• Variables:

  Core Link: \[ X_{i,v} = \begin{cases} 1 & \text{if } i \text{ is mapped to } v \\ 0 & \text{otherwise} \end{cases} \]

  Host link: \[ Y_{j,w} = \begin{cases} 1 & \text{if } j \text{ is mapped to } w \\ 0 & \text{otherwise} \end{cases} \]
BNV Mapper

Objective is to Minimize the usage of Core-links:

a) Maximizes isolation to the tenant, keeping links dedicated.

b) Provides scope for dynamic expansions. (Adding/upgrading tenants)

\[
\min : \sum_{i \in L, v \in Y} x_{iv} b_i + \sum_{m \in S, \{p, q\} \in R} M_{pq}^m
\]

More details in paper!
Implementation

Implemented on top of OpenVirteX

Integrated with a Bare-metal provisioning service (Testbed).

Deployed on National Cybersecurity Lab (NCL) Testbed.

Support OpenFlow 1.0/1.3
Implementation

Dynamic Loopback Wiring using L2 intermediate Switch

Static loopback wiring 24-ports of SDN Switch

Dynamic Loopback Wiring using L2 intermediate Switch
Evaluation

Can BNV:

1) faithfully replicate behaviour of topologies?

2) Provide control/data plane isolation to multiple tenants?

3) Scale of complex topologies embedded?
Evaluation: loopbacks & Big Switch Abstraction
Evaluation: loopbacks & Big Switch Abstraction

[Diagram showing network configurations with loopbacks and Big Switch abstraction]
Evaluation: loopbacks & Big Switch Abstraction
Mapping Complex Topologies
Evaluation: Fidelity of Complex Topologies

Single-Rooted Tree

Fat Tree

Jellyfish

Star
Evaluation: Fidelity of Complex Topologies
Evaluation: Fidelity of Complex Topologies

- Single-Rooted Tree
- Fat Tree
- Jellyfish
- Star

Graph showing the comparison of average shuffle read times for different topologies.
Evaluation: Fidelity of Complex Topologies

- Single-Rooted Tree
- Fat Tree
- Jellyfish
- Star

![Graph showing comparison of average shuffle read time for different topologies: Tree, Fat Tree, JellyFish, and Star. The graph indicates that the Tree topology has the highest average read time, followed by Fat Tree, JellyFish, and Star.]
Evaluation: Fidelity of Complex Topologies

Low Intra-pod locality
Heavy cross rack traffic!!
Evaluation: Fidelity of Complex Topologies

High Intra-pod locality
Low cross rack traffic!!

This observation is backed by Tale of Two Topologies [SIGCOMM ’17]

Take-away: BNV emulates the intrinsic characteristics of the topologies, otherwise observed with heavy wiring & building costs
Evaluation: Isolation of multiple tenants

Take-away: BNV can support multiple arbitrary tenants providing isolated & repeatable environments.
Evaluation : Scale of Topologies
Evaluation: Scale of Topologies

The graph shows the percentage core bandwidth utilization as a function of the number of virtual switches for different topologies. The x-axis represents the number of virtual switches (n), and the y-axis represents the percentage core bandwidth utilization. The graph includes four different topologies:

- n.sqrt(n) Links (no loops)
- n.sqrt(n) Links (loops)
- 2n Links (no loops)
- 2n Links (loops)

The graph illustrates how the bandwidth utilization changes with the number of virtual switches for each topology. The slopes and trends of the lines indicate the scalability of each topology.
Evaluation: Scale of Topologies

**Take-away:** Addition of loopback links increases the scalability significantly. Can emulate topologies with 130 switches and 300 links with just 5 ToR switches.
BNV provides High-fidelity network experimentation/cloud platform.

BNV can scale a network to support arbitrary topologies using the existing network.

Can emulate networks of more than 100 network devices with just 5 ToR Switches.

Deployed in NCL testbed [https://ncl.sg](https://ncl.sg)

Code Available: [https://github.com/praveingk/bnvirt](https://github.com/praveingk/bnvirt)