Software-Defined Border Router on Campus

Hyojoon Kim (Princeton University)
Chris Tengi (Princeton University)
What is a Software-Defined Border Router?

Traditional BGP border router

Internet

E-BGP

Forwarding table

Campus Network
What is a Software-Defined Border Router?

Software-Defined BGP border router

Data plane

Control plane

Internet

Campus Network

Forwarding table

E-BGP

Software!
Related Work

• Design and Implementation of a Routing Control Platform (a.k.a. RCP)
  • Caesar, Matthew, Donald Caldwell, Nick Feamster, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. *USENIX NSDI 2005*
  • “RCP is a logically-centralized platform, separate from the IP forwarding plane, that performs route selection on behalf of routers and communicates selected routes to the routers…”

• RouteFlow
  • https://sites.google.com/site/routeflow/

• Atrium project from ON.Lab
  • https://wiki.onosproject.org/display/ONOS/Peering+Router+-+ONF's+Project+Atrium
Goal of this talk

• Share our deployment experience

• Ignite discussion about doing things differently

• Talk about possible applications and future plans
Outline

• Science DMZ at Princeton University
• Software-Defined border router deployment
• Extensions and applications
• Q & A
Software-Defined Border Router in Rack

Internet2

To campus (ScienceDMZ)

To controller VM
Forwarding Pipeline in the Switch

Packet arrives at switch

- adds default output action based on physical topology

“Wires” to SW Router

Table 0

Forward BGP packets to Internet2

BGP Data

MAC based filter

Table 1

Forward BGP packets to Quagga VM

L3/L4 ACL filter

Table 2

Adjacency Table

Table 3

Group 0

Group 1

Output port to ScienceDMZ

17,583 prefixes

Table 0

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dst=1.2.4.0/24</td>
<td>Group 1</td>
</tr>
<tr>
<td>Dst=1.8.18.0/24</td>
<td>Group 1</td>
</tr>
<tr>
<td>Dst=128.112.229.0/24</td>
<td>Group 0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 1

- Replaces default output - Go to group table

Table 2

Table 3

- Replaces default output - Go to group table

Output port to Internet2

Adjacency Table
Evaluation

- Forwarding Performance
- Control traffic load
- Reliability
Forwarding Performance

Throughput
Before: 10 Gb/s
After: 10 Gb/s

Packet Loss
Before
After

Latency
Before
After
Throughput

Packet Loss

Latency

10 Gb/s
Data Plane Reliability – bulk transfers

Activity

- ESnet Read-Only Test DTN at CERN to Princeton Test DTN
  - transfer completed a minute ago

Overview

- Task ID: 229de0c0-aeb1-11e7-b018-22000a92523b
- Owner: Hyojoon Kim (hyojoonk@princeton.edu)
- Source: ESnet Read-Only Test DTN at CERN
- Destination: Princeton Test DTN
  - owner: esnet@globusid.org
- Condition: SUCCEEDED
- Requested: 2017-10-11 02:22 pm
- Completed: 2017-10-11 02:32 pm
- Transfer Settings: verify file integrity after transfer, transfer is not encrypted, overwriting all files on destination

Event Log

- Files: 1
- Directories: 0
- Bytes Transferred: 100 GB
- Effective Speed: 165.44 MB/s

Pending: 0
Succeeded: 1
Cancelled: 0
Expired: 0
Failed: 0
Retrying: 0
Skipped: 0

view debug data
Control Traffic Load

- BGP traffic between switch and controller
Control Traffic Load

- OpenFlow traffic between switch and controller
Control Plane Reliability

• What if the controller dies?

  • **Data forwarding**: not disrupted *until* there is a route update
  • **BGP session**: disrupted until recovery
  • **Recovery time**: depends on VM migration time
Why are we doing this?

• Separation of control plane and data plane

• Downsides
  • Adds complexity
  • More failure points

• Upsides
  • Easier to add customization
  • Easier to innovate
  • Less vendor lock-in
  • Less hardware dependent

Because we think these outweigh the downsides!
Applications
Network Hardware Virtualization

- Able to create multiple “instances” (Virtual Forwarding Context)
  - Can assign hardware resources to each VFC
  - Each VFC can connect to a separate control plane
Application 1: iptables-based Firewall

- OpenFlow Switch as A Low-impact Firewall
  - UCSD. Dr. Philip Papadopoulos
  - [https://youtu.be/ugH86z8QDJk](https://youtu.be/ugH86z8QDJk)

```
# Allow icmp
iptables -A INPUT -p icmp -j ACCEPT
#
# Globus DTN Infrastructure Servers to Science DMZ hosts
iptables -A INPUT -s 184.73.189.163 -d sciencedmz -p tcp -dport 2811 -j ACCEPT
iptables -A INPUT -s 174.129.226.69 -d sciencedmz -p tcp -dport 2811 -j ACCEPT
iptables -A INPUT -p tcp -d sciencedmz -dport 50000:51000 -j ACCEPT
iptables -A INPUT -p tcp -s sciencedmz -dport 50000:51000 -j ACCEPT
```
Application 2: Dynamic Bypass

- Automatically bypass Globus data transfer traffic, and remove bypass when transfer is done
Application 3: BGPSRx

- **BGP-SRx**: [https://bgpsrx.antd.nist.gov](https://bgpsrx.antd.nist.gov)
- **Work by Georgia Tech**
  - Brian Lebiednik (USMA West Point). ([brian.lebiednik@usma.edu](mailto:brian.lebiednik@usma.edu))
  - Sean Donovan ([sdonovan@gatech.edu](mailto:sdonovan@gatech.edu))
  - Russ Clark ([russ.clark@gatech.edu](mailto:russ.clark@gatech.edu))

![Diagram of BGP network with VM, L3 BGP, Internet2, Science DMZ, and Quagga with BGPSec and RPKI Route Origin Validation]
Application 4: BGP Anomaly Detection

• **Nemecis**. Georgos Siganos et. al. INFOCOM 2004
• **Prefix Hijack Alert System (PHAS)**. Mohit Lad et. al. USENIX Security 2006
• **Pretty Good BGP (PGBGP)**. Josh Karlin et. al. ICNP 2006.
  • PGBGP + Quagga ([http://www.cs.unm.edu/~karlinjf/pgbgp/](http://www.cs.unm.edu/~karlinjf/pgbgp/))
**Application 5: Route Optimization**

- **Application-specific traffic engineering**

  - **ISP A**
  - **Internet2**
  - **ISP B**

  - Video traffic
  - Bulk data transfer
  - Latency-sensitive traffic

  - L3 BGP
  - Route optimizer
  - Update based on outcome

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dst=1.2.4.0/24, Src, Src_port, Dst_port</td>
<td>Group 0</td>
</tr>
<tr>
<td>Dst=1.8.18.0/24, Src, Src_port, Dst_port</td>
<td>Group 1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Summary

• Princeton deployed a **Software-defined border router** in semi-production

• Been running **stable** for the **several months**
  • perfSONAR node (v4.0.0.2)
  • A Data Transfer Node (DTN)

• Running additional custom applications
  • Iptables-based firewall application
  • Automatic bypass for bulk data transfers (Globus)
Future Plans

• Continue to run and maintain

• Add more hosts to our ScienceDMZ network

• Implement and apply aforementioned applications
  • Starting with BGP-SRx (for path and origin validation)
Lessons Learned

• Running an SDN solution is not hard!

• Start with what you are familiar with
  • E.g., BGP configuration w/ Quagga.

• Start deployment on less mission-critical services

• Prove its stability, and focus on its benefits

• Get commercial support if possible
Q&A

joonk@princeton.edu