SDN-enabled Internet Exchange Point

Muhammad Shahbaz
Georgia Tech

Internet2 Innovation Award

Joint collaboration with:
Arpit Gupta, Laurent Vanbever, Hyojoon Kim, Sean Donovan, Russ Clark, Nick Feamster, Jennifer Rexford and Scott Shenker
BGP is notoriously inflexible and difficult to manage

Operating BGP has at least three limitations
BGP is notoriously inflexible and difficult to manage

Operating BGP has at least three limitations

- assume destination IP based routing
BGP is notoriously inflexible and difficult to manage

Operating BGP has at least three limitations

- assume destination IP based routing
- what people really want
- customized routing decisions
BGP is notoriously inflexible and difficult to manage

Operating BGP has at least three limitations

- assume destination IP based routing
- policies are applied to direct neighbors

\textit{what people really want}
BGP is notoriously inflexible and difficult to manage.

Operating BGP has at least three limitations:

- Assume destination IP based routing
- Policies are applied to direct neighbors
- Effect end-to-end paths

What people really want:
- Customized routing decisions
BGP is notoriously inflexible and difficult to manage

Operating BGP has at least three limitations:

- assume destination IP based routing
- policies are applied to direct neighbors
- indirectly influence forwarding paths

What people really want:
- customized routing decisions
- affect end-to-end paths
BGP is notoriously inflexible and difficult to manage

Operating BGP has at least three limitations

- assume destination IP based routing
- policies are applied to direct neighbors
- indirectly influence forwarding paths

"what people really want"

- customized routing decisions
- affect end-to-end paths
- directing traffic on specific paths
SDN can enable fine-grained, flexible and direct expression of interdomain policies

SDN devices forward based on any packet-header fields at line rate, enabling flexible forwarding.

SDN controller can be controlled by remote parties on a bilateral basis, without any global standards.

SDN controller exerts direct control on the data plane using a standardized API such as OpenFlow.
Internet Exchange Points are perfect places to deploy new interdomain features
Internet Exchange Points are perfect places to deploy new interdomain features
Internet Exchange Points are perfect places to deploy new interdomain features

Internet Exchange Points (IXPs)

- connect a large number of participants
Internet Exchange Points are perfect places to deploy new interdomain features

- connect a large number of participants
  600 participants

Internet Exchange Points (IXPs)

AMS-IX:
Internet Exchange Points are perfect places to deploy new interdomain features

<table>
<thead>
<tr>
<th>Internet Exchange Points (IXPs)</th>
<th>AMS-IX</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect a large number of participants</td>
<td>600 participants</td>
</tr>
<tr>
<td>carry a large amount of traffic</td>
<td>&gt; 2250 Gb/s (peak)</td>
</tr>
</tbody>
</table>
Internet Exchange Points are perfect places to deploy new interdomain features

- Internet Exchange Points (IXPs)
  - connect a large number of participants
  - carry a large amount of traffic
  - are a hotbed of innovation

AMS-IX:
- 600 participants
- > 2250 Gb/s (peak)

BGP Route Server
Mobile peering
Open peering
...

Internet Exchange Points are perfect places to deploy new interdomain features

Internet Exchange Points (IXPs)

- connect a large number of participants
- carry a large amount of traffic
- are a hotbed of innovation

Even a single deployment can have a large impact!
An IXP is a large L2 domain where participants routers peer using BGP
An IXP is a large L2 domain where participants routers peer using BGP
An IXP is a large L2 domain where participants routers peer using BGP.
An IXP is a large L2 domain where participants routers peer using BGP.
An IXP is a large L2 domain where participants routers peer using BGP.
With respect to IXPs, SDN-enabled IXPs (SDX) ...
With respect to IXPs, SDN-enabled IXPs (SDX) *data plane* relies on SDN-capable devices.
With respect to IXPs, SDN-enabled IXPs (SDX) control plane relies on a SDX controller.
SDX participants write policies using a high-level language on top of a virtual topology.
SDX participants write policies using a high-level language on top of a virtual topology

match(dstip=ipA) >> fwd(outA)

match(dstip=ipC) >> fwd(C) +
macth(dstip=ipA) >> fwd(A) +
macth(dstip=ipB) >> fwd(outB)

match(dstip=ipC) >> fwd(outC)
The SDX controller composes policies together ensuring *isolation* and *correctness*

```plaintext
match(dstip=ipC) >> fwd(C) + 
mismatch(dstip=ipA) >> fwd(A) + 
mismatch(dstip=ipB) >> fwd(outB)
```

```plaintext
match(dstip=ipA) >> fwd(outA)
```

```plaintext
match(dstip=ipC) >> fwd(outC)
```

SDX controller
The SDX controller composes policies together ensuring *isolation* and *correctness*.

- **match(dstip=ipC) >> fwd(C)**
- **match(dstip=ipA) >> fwd(A)**
- **match(dstip=ipB) >> fwd(outB)**

**OpenFlow rules**

**SDX controller**
To ensure compatibility and scalability, SDX supports MAC-based forwarding by default.

```
match(dstmac=MACA) >> fwd(outA)
match(dstmac=MACB) >> fwd(outB)
match(dstmac=MACC) >> fwd(outC)
```

*Participants’ policies subsume default forwarding behavior*
What does SDX enable that was **hard** or **impossible** to do before?
SDX enables a wide range of novel interdomain applications

<table>
<thead>
<tr>
<th>Category</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>security</td>
<td>Prevent/block policy violation</td>
</tr>
<tr>
<td></td>
<td>Prevent participants communication</td>
</tr>
<tr>
<td>forwarding optimization</td>
<td>Middlebox traffic steering</td>
</tr>
<tr>
<td></td>
<td>Traffic offloading</td>
</tr>
<tr>
<td></td>
<td>Inbound Traffic Engineering</td>
</tr>
<tr>
<td>peering</td>
<td>Application-specific peering</td>
</tr>
<tr>
<td>remote-control</td>
<td>Wide-area load balancing</td>
</tr>
<tr>
<td></td>
<td>Influence BGP path selection</td>
</tr>
<tr>
<td></td>
<td>Upstream blocking of DoS attacks</td>
</tr>
</tbody>
</table>
SDX enables a wide range of novel interdomain applications

security
  Prevent/block policy violation
  Prevent participants communication

**forwarding optimization**
  Middlebox traffic steering
  Traffic offloading
  **Inbound Traffic Engineering**

peering
  Application-specific peering

remote-control
  Wide-area load balancing
  Influence BGP path selection
  Upstream blocking of DoS attacks
SDX can improve inbound traffic engineering
SDX can improve inbound traffic engineering

Given an IXP Physical Topology

- eBGP session
SDX can improve inbound traffic engineering

Given an IXP Physical Topology and a BGP topology

192.0.1.0/24
192.0.2.0/24
192.0.3.0/24

192.0.{1,2,3}.0/24
SDX can improve inbound traffic engineering

Implements B’s inbound policy

<table>
<thead>
<tr>
<th>to</th>
<th>from</th>
<th>receive on</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.0.1.0/24</td>
<td>A</td>
<td>B1</td>
</tr>
<tr>
<td>192.0.2.0/24</td>
<td>B</td>
<td>B2</td>
</tr>
<tr>
<td>192.0.2.0/24</td>
<td>ATT_IP</td>
<td>B2</td>
</tr>
<tr>
<td>192.0.2.0/24</td>
<td>*</td>
<td>B1</td>
</tr>
<tr>
<td>192.0.3.0/24</td>
<td>*</td>
<td>B2</td>
</tr>
</tbody>
</table>
How do you do that with BGP?

Implements B’s inbound policy

<table>
<thead>
<tr>
<th>to</th>
<th>from</th>
<th>receive on</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.0.1.0/24</td>
<td>A</td>
<td>B1</td>
</tr>
<tr>
<td>192.0.2.0/24</td>
<td>B</td>
<td>B2</td>
</tr>
<tr>
<td>192.0.2.0/24</td>
<td>ATT_IP</td>
<td>B2</td>
</tr>
<tr>
<td>192.0.2.0/24</td>
<td>*</td>
<td>B1</td>
</tr>
<tr>
<td>192.0.3.0/24</td>
<td>*</td>
<td>B2</td>
</tr>
</tbody>
</table>
It is at least hard... BGP provides few knobs to influence remote decisions.

Implementing such a policy is configuration-intensive using AS-Path prepend, MED, community tagging, etc.
and even impossible for some requirements...

BGP policies **cannot** influence remote parties' decisions based on source addresses

<table>
<thead>
<tr>
<th>to</th>
<th>from</th>
<th>receive on</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.0.2.0/24</td>
<td>ATT_IP</td>
<td>B2</td>
</tr>
</tbody>
</table>
In any case, the outcome is unpredictable

Implementing such a policy is configuration-intensive using AS-Path prepend, MED, community tagging, etc.

Absolutely no guarantee that the remote party will comply one can only “influence” remote decisions

Networks engineers have no choice but to “try and see” which makes it difficult to adapt to traffic pattern
With a SDX, implementing B’s inbound policy is easy

SDX policies give B direct control on its forwarding paths

<table>
<thead>
<tr>
<th>to</th>
<th>from</th>
<th>fwd</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.0.1.0/24</td>
<td>A</td>
<td>B1</td>
</tr>
<tr>
<td>192.0.2.0/24</td>
<td>B</td>
<td>B2</td>
</tr>
<tr>
<td>192.0.2.0/24</td>
<td>ATT_IP</td>
<td>B2</td>
</tr>
<tr>
<td>192.0.2.0/24</td>
<td>*</td>
<td>B1</td>
</tr>
<tr>
<td>192.0.3.0/24</td>
<td>*</td>
<td>B2</td>
</tr>
</tbody>
</table>

B’s SDX Policy

- `match(dstip=192.0.1.0/24, srcmac=A) >> fwd(B1)`
- `match(dstip=192.0.2.0/24, srcmac=B) >> fwd(B2)`
- `match(dstip=192.0.2.0/24, srcip=ATT) >> fwd(B2)`
- `match(dstip=192.0.2.0/24) >> fwd(B1)`
- `match(dstip=192.0.3.0/24) >> fwd(B2)`
Several challenges remain

We need authentication mechanisms to validate policies
e.g., using Resource Public Key Infrastructure (RPKI)

We need “access-control” to constrain the policies
e.g., limiting the capabilities available to each participant

We need to make the platform scalable
as SDN devices currently support a relatively small # of rules
SDN-enabled Internet Exchange Point

Muhammad Shahbaz
www.cc.gatech.edu/~mshahbaz

Internet2 Innovation Award
April, 9 2014