Bringing SDN to the Internet, one exchange point at the time

Joint work with: Arpit Gupta, Muhammad Shahbaz, Sean P. Donovan, Russ Clark, Brandon Schlinker, E. Katz-Bassett, Nick Feamster, Jennifer Rexford and Scott Shenker

Laurent Vanbever
Princeton University

AT&T/Intel/ONRC SDN retreat
October, 15 2014
BGP is notoriously inflexible and difficult to manage
BGP is notoriously inflexible and difficult to manage

BGP          SDN

Fwd paradigm

Fwd control

Fwd influence
BGP is notoriously inflexible and difficult to manage

Fwd paradigm  destination-based

Fwd control  indirect

Fwd influence  local

BGP session

BGP  SDN
SDN can enable fine-grained, flexible and direct expression of interdomain policies

<table>
<thead>
<tr>
<th></th>
<th>BGP</th>
<th>SDN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fwd paradigm</strong></td>
<td>destination-based</td>
<td>any</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source addr, ports,...</td>
</tr>
<tr>
<td><strong>Fwd control</strong></td>
<td>indirect</td>
<td>direct</td>
</tr>
<tr>
<td></td>
<td>configuration</td>
<td>open API (e.g., OpenFlow)</td>
</tr>
<tr>
<td><strong>Fwd influence</strong></td>
<td>local</td>
<td>global</td>
</tr>
<tr>
<td></td>
<td>BGP session</td>
<td>remote controller control</td>
</tr>
</tbody>
</table>
How do you deploy SDN in a network composed of 50,000 subnetworks?
How do you deploy SDN in a network composed of 50,000 subnetworks?

Well, you don’t …
Instead, you aim at finding locations where deploying SDN can have the most impact.
Instead, you aim at finding locations where deploying SDN can have the most impact

Deploy SDN in locations that

- connect a large number of networks
- carry a large amount of traffic
- are opened to innovation
Internet eXchange Points (IXP) meet all the criteria

Deploy SDN in locations that

- connect a large number of networks
- carry a large amount of traffic
- are opened to innovation

AMS-IX

675 networks
3.2 Tb/s (peak)
BGP Route Server
Mobile peering
Open peering...

https://www.ams-ix.net
A single deployment can have a large impact

Deploy SDN in locations that

- connect a large number of networks
- carry a large amount of traffic
- are opened to innovation

AMS-IX

- 675 networks
- 3.2 Tb/s (peak)
- BGP Route Server
- Mobile peering
- Open peering...

https://www.ams-ix.net
\[ SDX = SDN + IXP \]
SDX = SDN + IXP

Augment the IXP data-plane with SDN capabilities
keeping default forwarding and routing behavior

Enable fine-grained inter-domain policies
bringing new features & simplifying operations
Enable fine-grained inter-domain policies with scalability and correctness in mind supporting large IXP load and resolving conflicts.

Augment the IXP data-plane with SDN capabilities keeping default forwarding and routing behavior bringing new features & simplifying operations.

\[
SDX = SDN + IXP
\]
SDX enables multiple stakeholders to implement policies and apps over a shared infrastructure.
Bringing SDN to the Internet, one exchange point at the time

1. Architecture
   programming model

2. Scalability
   control- & data-plane

3. Applications
   inter domain bonanza
Bringing SDN to the Internet, one exchange point at the time

1 Architecture
programming model

Scalability
control– & data–plane

Applications
inter domain bonanza
An IXP is a large layer-2 domain
An IXP is a large layer-2 domain where participant routers exchange routes using BGP.
To alleviate the need of establishing eBGP sessions, IXP often provides a Route Server (route multiplexer).
IP traffic is exchanged directly between participants
With respect to a traditional IXP,
With respect to a traditional IXP, SDX data-plane relies on SDN-capable devices.
With respect to a traditional IXP, SDX control-plane relies on a SDN controller.
SDX participants express their forwarding policies in a high-level language, built on top of Pyretic (*)

(*) http://frenetic-lang.org/pyretic/
SDX policies are composed of a pattern and some actions

match (Pattern), then (Actions)
Pattern selects packets based on any header fields, while Actions forward or modify the selected packets.

Pattern

\[
\text{match (eth_type, vlan_id, srcmac, dstmac, protocol, dstip, tos, srcip, srcport, dstport, ...) }, \text{ then (Actions)}
\]
Pattern selects packets based on any header fields, while actions forward or modify the selected packets.

Actions
- drop
- forward
- rewrite

match (Pattern), then ( )
Each SDX participant writes her policies independently

Participant #2 policy

\textbf{match}(dstport=80), \textbf{fwd}(#3)

\textbf{match}(dstport=22), \textbf{fwd}(#1)
Each SDX participant writes her policies independently

Participant #2 policy

\[
\text{match}(\text{dstport}=80), \ fwd(\#3) \\
\text{match}(\text{dstport}=22), \ fwd(\#1)
\]

Participant #3 policy

\[
\text{match}(\text{srcip}=0*), \ fwd(\text{left}) \\
\text{match}(\text{srcip}=1*), \ fwd(\text{right})
\]
... and transmit them to the SDX controller

Participant #2 policy

\textit{match}(\text{dstport}=80), \textit{fwd}(\#3) \\
\textit{match}(\text{dstport}=22), \textit{fwd}(\#1)

Participant #3 policy

\textit{match}(\text{srcip}=0^*), \textit{fwd}(\text{left}) \\
\textit{match}(\text{srcip}=1^*), \textit{fwd}(\text{right})
The controller compiles all the policies into SDN forwarding rules.

**Participant #2 policy**

match(dstport=80), fwd(#3)
match(dstport=22), fwd(#1)

**Participant #3 policy**

match(srcip=0*), fwd(left)
match(srcip=1*), fwd(right)

SDX controller

SDN

Forwarding rules
SDX compilation stage implements each participant policy in the data-plane

Ensuring isolation

Resolving conflict

Considering BGP
SDX compilation stage implements *each* participant policy in the data-plane

 Ensuring isolation

 Resolving conflict

 Considering BGP

 Each participant controls one “virtual” switch connected to participants it can communicate with
SDX compilation stage implements *each* participant policy in the data-plane
SDX compilation stage implements *each* participant policy in the data-plane

Ensuring isolation

Resolving conflict

Considering BGP

Policies are augmented with BGP information
guarantee correctness
and reachability
Bringing SDN to the Internet, one exchange point at the time

Architecture
programming model

2

Scalability
control- & data-plane

Applications
inter domain bonanza
The SDX platform faces scalability challenges in both the data– and in the control–plane.
512k prefixes, 500+ participants, potentially $10^9$ of forwarding rules

forwarding rules must be updated dynamically according to BGP
To scale, the SDX platform leverages existing infrastructure & domain-specific knowledge

Data-plane: 
- Space
  - Aggregate rules, on existing routers

Control-plane: 
- Time
  - Leverage policy structure
aggregate rules, on existing routers
SDX groups IP prefixes according to their behavior through the fabric

- policies are prefix-based
  just the way the Internet works

- forwarding actions are shared for a lot of prefixes
  e.g., all prefixes advertised by X
SDX groups IP prefixes according to their behavior through the fabric

- policies are prefix-based
  just the way the Internet works
- forwarding actions are shared for a lot of prefixes
  e.g., all prefixes advertised by X
- group prefixes by equivalence class
SDX leverages edge routers to map packets to their equivalence class.
SDX considers edge routers’ FIB as the first stage of a multi-stage FIB.
Routers FIB match on the destination prefix and set a tag accordingly.

Table #1

Edge router

Table #2

SDX switch

set a TAG based on IP prefix
SDX FIB matches on the tag

Edge router

Table #1

set a TAG based on IP prefix

Table #2

match TAG

SDX switch
SDX uses BGP NH as a provisioning interface and MAC addresses as tag in the data-plane.
SDX accommodates policies for 100+ participants, with less than 30k rules.
data-plane
  \textit{space}

control-plane
  \textit{time}

leverage
  policy structure
SDX policies share key characteristics

<table>
<thead>
<tr>
<th>Type</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>disjointness</td>
</tr>
<tr>
<td>Dynamic</td>
<td>locality</td>
</tr>
<tr>
<td></td>
<td>burstiness</td>
</tr>
</tbody>
</table>
SDX policies share key characteristics

- **Static**
  - **disjointness**

  disjoint policies don't need to be composed
  significant gain as composition is costly

- **Dynamic**
  - locality
  - burstiness
SDX policies share key characteristics

Static disjointness

Dynamic locality

burstiness

policy updates usually impact few prefixes

75% of the updates affect no more than 3 prefixes
SDX policies share key characteristics

Static

- disjointness

Dynamic

- locality

**burstiness**

Policy updates are separated by large periods of inactivity

In 75% of the case, updates are separated by 10s or more
These characteristics enable an efficient, 2-stage compilation algorithm:

**Stage 1**  
*Fast*, non-optimal algorithm upon updates  
can install more forwarding rules than required

**Stage 2**  
*Slow*, but optimal algorithm in background  
regroup rules according to forwarding behavior
These characteristics enable an efficient, 2-stage compilation algorithm

- **Fast**, non-optimal algorithm upon updates
  - can install more forwarding rules than required

- **Slow**, but optimal algorithm in background
  - regroup rules according to forwarding behavior

- Time vs Space trade-off
In most cases, the SDX takes <100 ms to recompute the entire policy.
Bringing SDN to the Internet, one exchange point at the time
SDX enables a wide range of novel 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></td>
<td>Upstream blocking of DoS attacks</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></td>
<td>Fast convergence</td>
</tr>
<tr>
<td>peering</td>
<td>Application–specific peering</td>
</tr>
<tr>
<td>remote–control</td>
<td>Influence BGP path selection</td>
</tr>
<tr>
<td></td>
<td>Wide–area load balancing</td>
</tr>
</tbody>
</table>
SDX enables a wide range of novel 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></td>
<td>Upstream blocking of DoS attacks</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><strong>Inbound Traffic Engineering</strong></td>
</tr>
<tr>
<td></td>
<td>Fast convergence</td>
</tr>
<tr>
<td>peering</td>
<td>Application–specific peering</td>
</tr>
<tr>
<td>remote–control</td>
<td>Influence BGP path selection</td>
</tr>
<tr>
<td></td>
<td>Wide–area load balancing</td>
</tr>
</tbody>
</table>
SDX can improve inbound traffic engineering
Given an IXP Physical Topology and a BGP topology,
Given an IXP Physical Topology and a BGP topology, Implement B’s inbound policies

B’s inbound policies

<table>
<thead>
<tr>
<th>to</th>
<th>from</th>
<th>receive on</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.0.1/24</td>
<td>A</td>
<td>left</td>
</tr>
<tr>
<td>192.0.2/24</td>
<td>C</td>
<td>right</td>
</tr>
<tr>
<td>192.0.2/24</td>
<td>ATT_IP</td>
<td>right</td>
</tr>
<tr>
<td>192.0.1/24</td>
<td>*</td>
<td>right</td>
</tr>
<tr>
<td>192.0.2/24</td>
<td>*</td>
<td>left</td>
</tr>
</tbody>
</table>
How do you that with BGP?

B’s inbound policies

<table>
<thead>
<tr>
<th>to</th>
<th>from</th>
<th>receive on</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.0.1/24</td>
<td>A</td>
<td>left</td>
</tr>
<tr>
<td>192.0.2/24</td>
<td>C</td>
<td>right</td>
</tr>
<tr>
<td>192.0.2/24</td>
<td>ATT_IP</td>
<td>right</td>
</tr>
<tr>
<td>192.0.1/24</td>
<td>*</td>
<td>right</td>
</tr>
<tr>
<td>192.0.2/24</td>
<td>*</td>
<td>left</td>
</tr>
</tbody>
</table>
It is hard.
BGP provides few knobs to influence remote decisions.

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

BGP policies cannot influence remote decisions based on source addresses

to from receive on

192.0.2.0/24 ATT_IP right
In any case, the outcome is unpredictable

There is no guarantee that remote parties will comply
one can only “influence” remote decisions

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

SDX policies give any participant **direct** control on its forwarding paths

<table>
<thead>
<tr>
<th>to</th>
<th>from</th>
<th>fwd</th>
<th>B’s SDX Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.0.1/24</td>
<td>A</td>
<td>left</td>
<td><strong>match</strong>(dstip=192.0.1/24, srcmac=A), <strong>fwd</strong>(L)</td>
</tr>
<tr>
<td>192.0.2/24</td>
<td>B</td>
<td>right</td>
<td><strong>match</strong>(dstip=192.0.2/24, srcmac=B), <strong>fwd</strong>(R)</td>
</tr>
<tr>
<td>192.0.2/24</td>
<td>ATT_IP</td>
<td>right</td>
<td><strong>match</strong>(dstip=192.0.2/24, srcip=ATT), <strong>fwd</strong>(R)</td>
</tr>
<tr>
<td>192.0.1/24</td>
<td>*</td>
<td>right</td>
<td><strong>match</strong>(dstip=192.0.1/24), <strong>fwd</strong>(R)</td>
</tr>
<tr>
<td>192.0.2/24</td>
<td>*</td>
<td>left</td>
<td><strong>match</strong>(dstip=192.0.2/24), <strong>fwd</strong>(L)</td>
</tr>
</tbody>
</table>
**SDX enables a wide range of novel 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></td>
<td>Upstream blocking of DoS attacks</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></td>
<td>Fast convergence</td>
</tr>
<tr>
<td>peering</td>
<td>Application–specific peering</td>
</tr>
<tr>
<td>remote-control</td>
<td>Influence BGP path selection</td>
</tr>
<tr>
<td></td>
<td>Wide–area load balancing</td>
</tr>
</tbody>
</table>
SDX enables a wide range of novel applications

| security | Prevent/block policy violation |
| security | Prevent participants communication |
| security | **Upstream blocking of DoS attacks** |

| forwarding optimization | Middlebox traffic steering |
| forwarding optimization | Traffic offloading |
| forwarding optimization | Inbound Traffic Engineering |
| forwarding optimization | Fast convergence |

| peering | Application–specific peering |

| remote–control | Influence BGP path selection |
| remote–control | Wide–area load balancing |
SDX can help mitigating DDoS attacks, closer to the source
AS1 is victim of a DDoS attack targeting its web server.
AS1 remotely installs drop policies in all SDXes
AS1 remotely installs
drop policies in all SDXes

AS1 policy

```
match(srcip=*, dstip=10.0.01/32, dstport=80) >> drop()
```
SDX policies are targeted, hence other services stay reachable

AS1 policy

\[
\text{match}(\text{srcip}=*, \text{dstip}=10.0.01/32, \text{dstport}=80) >> \text{drop()}
\]

\( single \text{ IP} \quad single \text{ service} \)
Bringing SDN to the Internet, one exchange point at the time

Architecture
programming model

Scalability
control- & data-plane

Applications
inter domain bonanza
What’s next?
SDX currently consider a single deployment
What about interconnecting SDX platforms?
What about replacing BGP completely with a SDX-mediated Internet?
Let's take over the world
Not only for eBGP. Multi-site SDX can be used within an AS to implement iBGP policies.
SDX is a promising first step towards fixing Internet routing

Enable declarative, fine-grained inter-domain policies many of which are not possible Today

Scale to hundreds of participants both in the control- and in the data-plane

Running code (*) and deployment under way important potential for impact

(*) https://github.com/sdn-ixp/sdx-platform
Bringing SDN to the Internet, one exchange point at the time

Laurent Vanbever
www.vanbever.eu

Internet ♥ SDN
October, 15 2014