SDX: A Software-Defined Internet Exchange

Arpit Gupta

Laurent Vanbever, Muhammad Shahbaz, Sean Donovan, Brandon Schlinker, Nick Feamster, Jennifer Rexford, Scott Shenker, Russ Clark, Ethan Katz-Bassett

Georgia Tech, Princeton University, UC Berkeley, USC
The Interdomain Ecosystem is Evolving ...

Flatter and densely interconnected Internet*

*Labovitz et al., Internet Inter-Domain Traffic, SIGCOMM 2010
...But BGP is Not

- Routing **only on destination IP prefixes**  
  (No customization of routes by application, sender)

- Can only influence **immediate neighbors**  
  (No ability to affect path selection remotely)

- **Indirect** control over data-plane forwarding  
  (Indirect mechanisms to influence path selection)

How to overcome BGP’s limitations?
SDN for Interdomain Routing

• Forwarding on **multiple header fields** (not just destination IP prefixes)

• Ability to **control entire networks** with a single software program (not just immediate neighbors)

• **Direct control** over data-plane forwarding (not indirect control via control-plane arcana)

How to incrementally deploy SDN for Interdomain Routing?
Deploy SDN at Internet Exchanges

- **Leverage:** SDN deployment even at single IXP can yield benefits for tens to hundreds of ISPs

- **Innovation hotbed:** Incentives to innovate as IXPs on front line of peering disputes

- **Growing in numbers:** ~100 new IXPs established in past three years*

*https://prefix.pch.net/applications/ixpdir/summary/growth/*
Background: Conventional IXPs

- AS A Router
- AS B Router
- AS C Router
- Route Server
- BGP Session
- IXP
- Switching Fabric
SDX = SDN + IXP

AS A Router

BGP Session

AS B Router

AS C Router

SDX Controller

SDN Switch

SDX
SDX Opens Up New Possibilities

• More flexible business relationships
  Make peering decisions based on time of day, volume of traffic & nature of application

• More direct & flexible traffic control
  Define fine-grained traffic engineering policies

• Better security
  – Prefer “more secure” routes
  – Automatically blackhole attack traffic
SDX Enables Innovations at IXPs

• **Dropping of attack traffic**
  – Blocking unwanted traffic in middle of Internet

• **Inbound traffic engineering**
  – Divide traffic by sender or application

• **Application-specific peering**
  – Video traffic via Comcast, non-video via ATT

• **Server load balancing**
  – Select data centers to handle request

• **Redirection through middleboxes**
  – E.g., transcoding, caching, monitoring, etc.
SDX Enables Innovations at IXPs

- **Dropping of attack traffic**  
  - Blocking unwanted traffic in middle of Internet
- Inbound traffic engineering  
  - Divide traffic by sender or application
- Application-specific peering  
  - Video traffic via Comcast, non-video via ATT
- Server load balancing  
  - Select data centers to handle request
- Redirection through middleboxes  
  - E.g., transcoding, caching, monitoring, etc.
Dropping of Attack Traffic
Dropping of Attack Traffic

AS C under attack originating from AS A
Dropping of Attack Traffic

ASC can remotely block attack traffic at SDX(s)
SDX vs. Traditional blackholing

• **Remote influence**
  Physical connectivity to SDX not required

• **More specific**
  Drop rules based on multiple header fields, source address, destination address, port number …

• **Coordinated**
  Drop rules can be coordinated across multiple IXPs
SDX Enables Innovations at IXPs

• Dropping of attack traffic
  – Blocking unwanted traffic in middle of Internet

• Inbound traffic engineering
  – Divide traffic by sender or application

• Application-specific peering
  – Video traffic via Comcast, non-video via ATT

• Server load balancing
  – Select data centers to handle request

• Redirection through middleboxes
  – E.g., transcoding, caching, monitoring, etc.
Inbound Traffic Engineering

AS A Router

SDX Controller

SDX

AS B Router

AS C Routers

10.0.0.0/8
Inbound Traffic Engineering

<table>
<thead>
<tr>
<th>Incoming Traffic</th>
<th>Out Port</th>
<th>Using BGP</th>
<th>Using SDX</th>
</tr>
</thead>
<tbody>
<tr>
<td>dstport = 80</td>
<td>C1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Inbound Traffic Engineering

**AS A Router**

**AS B Router**

Incoming Data

---

**Fine grained policies not possible with BGP**

---

<table>
<thead>
<tr>
<th>Incoming Traffic</th>
<th>Out Port</th>
<th>Using BGP</th>
<th>Using SDX</th>
</tr>
</thead>
<tbody>
<tr>
<td>dstport = 80</td>
<td>C1</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>
Inbound Traffic Engineering

Enables fine-grained traffic engineering policies

<table>
<thead>
<tr>
<th>Incoming Traffic</th>
<th>Out Port</th>
<th>Using BGP</th>
<th>Using SDX</th>
</tr>
</thead>
<tbody>
<tr>
<td>dstport = 80</td>
<td>C1</td>
<td>?</td>
<td>match(dstport =80) → fwd(C1)</td>
</tr>
</tbody>
</table>
Building SDX is Challenging

- Programming **abstractions**
  How networks define SDX policies and how are they combined together?

- **Interoperation** with BGP
  How to provide flexibility w/o breaking global routing?

- **Scalability**
  How to handle policies for hundreds of peers, half million prefixes and matches on multiple header fields?
Building SDX is Challenging

• Programming **abstractions**
  How networks define SDX policies and how are they combined together?

• **Interoperation** with BGP
  How to provide flexibility w/o breaking global routing?

• **Scalability**
  How to handle policies for hundreds of peers, half million prefixes and matches on multiple header fields?
Directly Program the SDX Switch

AS A & C directly program the SDX Switch

Switching Fabric

match(dstport=80) → drop

match(dstport=80) → fwd(C1)
Conflicting Policies

Switching Fabric

A1 → drop? C1? → B1

match(dstport=80) → drop
match(dstport=80) → fwd(C1)

How to restrict participant’s policy to traffic it sends or receives?
Virtual Switch Abstraction

Each AS writes policies for its own virtual switch
Combining Participant’s Policies

Policy\( (p) = Pol_{A} \rightarrow Pol_{C} \)
Building SDX is Challenging

• Programming abstractions
  How networks define SDX policies and how are they combined together?

• Interoperation with BGP
  How to provide flexibility w/o breaking global routing?

• Scalability
  How to handle policies for hundreds of peers, half million prefixes and matches on multiple header fields?
Requirement: Forwarding Only Along BGP Advertised Routes

match(dstport=80) → fwd(C)
Ensure ‘p’ is not forwarded to C

\[
\text{match}(\text{dstport}=80) \rightarrow \text{fwd}(C)
\]

dstip = 20.0.0.1
dstport = 80
Solution: Policy Augmentation

\[(\text{match}(\text{dstport}=80) \land \text{match}(\text{dstip} = 10/8)) \rightarrow \text{fwd}(C)\]
Building SDX is Challenging

• **Programming abstractions**
  How networks define SDX policies and how are they combined together?

• **Interoperation with BGP**
  How to provide flexibility w/o breaking global routing?

• **Scalability**
  How to handle policies for hundreds of peers, half million prefixes and matches on multiple header fields?
Scalability Challenges

• **Reducing Data-Plane State:** Support for all forwarding rules in (limited) switch memory

• **Reducing Control-Plane Computation:** Faster policy compilation
Scalability Challenges

- **Reducing Data-Plane State**: Support for all forwarding rules in (limited) switch memory; millions of flow rules possible.

- **Reducing Control-Plane Computation**: Faster policy compilation; policy compilation could take hours.
Reducing Data-Plane State: Observations

• Internet routing policies defined for groups of prefixes.*

• Edge routers can handle matches on hundreds of thousands of IP prefixes.

*Feamster et al., Guidelines for Interdomain TE, CCR 2003
Reducing Data-Plane State: Solution

Group prefixes with similar forwarding behavior

SDX Controller
Reducing Data-Plane State: Solution

Advertise one BGP next hop for each such prefix group

Edge router
Reducing Data-Plane State: Solution

Flow rules at SDX match on BGP next hops

[Diagram showing flow rules and BGP next hops]
Reducing Data-Plane State: Solution

For hundreds of participants’ policies, few millions $\Rightarrow < 35K$
flow rules
Reducing Control-Plane Computation

- **Initial policy compilation time**
  - Leveraged domain-specific knowledge of policies
  - Hundreds of participants requires < 15 minutes

- **Policy recompilation time**
  - Leveraged bursty nature of BGP updates
  - Most recompilation after a BGP update < 100 ms
SDX Platform

• Running code with full BGP-integration

• SDX Testbeds:
  – Uses Transit Portal
  – Emulates edge routers (Mininet)

Github repo: https://github.com/sdn-ixp/sdx/
SDX Enables Innovations at IXPs

- Dropping of attack traffic
  - Blocking unwanted traffic in middle of Internet
- Inbound traffic engineering
  - Divide traffic by sender or application
- **Application-specific peering**
  - Video traffic via Comcast, non-video via ATT
- Server load balancing
  - Select data centers to handle request
- Redirection through middleboxes
  - E.g., transcoding, caching, monitoring, etc.
Use Case: Application-specific Peering

Transit Portal brings real traffic to SDX Fabric
Use Case:
Application-specific Peering

Policy = match(dstport = 80) → fwd(B)
Use Case: Application-specific Peering

- AS A
- AS B
- SDX Fabric
- AS C

204.57.0.64

Traffic Rate (Mbps)

Time (seconds)

Application-specific peering policy

Route withdrawal
SDX Deployment

• Research & Education Networks
  Internet2, GENI, SOX, ESnet, NSA-LTS

• Commercial Networks
  Regional IXPs in US, Europe & Africa

• NSF program to encourage SDX deployments
Next Steps

Building SDX-mediated Internet
SDX currently considers a single deployment
Step 1: Interconnecting SDX platforms
Step 2: Completely replacing BGP with SDX-mediated Internet
SDX-Mediated Internet: Advantages

• **New endpoint peering paradigm**
  More flexible, tailored to the traffic exchanged

• **Simple, scalable, and policy neutral “Spine”**
  SDX-to-SDX only, just carry bits

• **In-sync with current Internet Ecosystem**
  Content consumers vs providers vs transit providers
SDX-Mediated Internet: New Research Challenges

• New endpoint peering paradigm
  Policy Analysis ?

• Simple, scalable, and policy neutral “Spine”
  Routing Mechanisms ?

• In-sync with current Internet Ecosystem
  New players ?
Summary

• **SDN-based exchange (SDX)** is promising for fixing Internet routing

• Solved various challenges in building a real deployable SDX

• Many open research problems, both for building and using SDX

**Contact:** agupta80@gatech.edu

**Github repo:** [https://github.com/sdn-ixp/sdx/](https://github.com/sdn-ixp/sdx/)