iSDX: An Industrial-Scale Software-Defined IXP

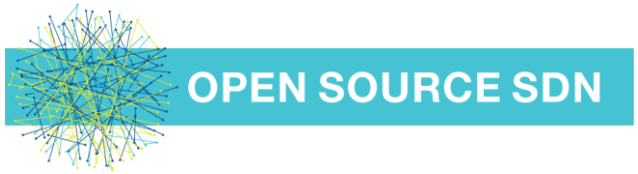
Arpit Gupta, Princeton University

http://sdx.cs.princeton.edu

Robert MacDavid, Rüdiger Birkner, Marco Canini,

Nick Feamster, Jennifer Rexford, Laurent Vanbever

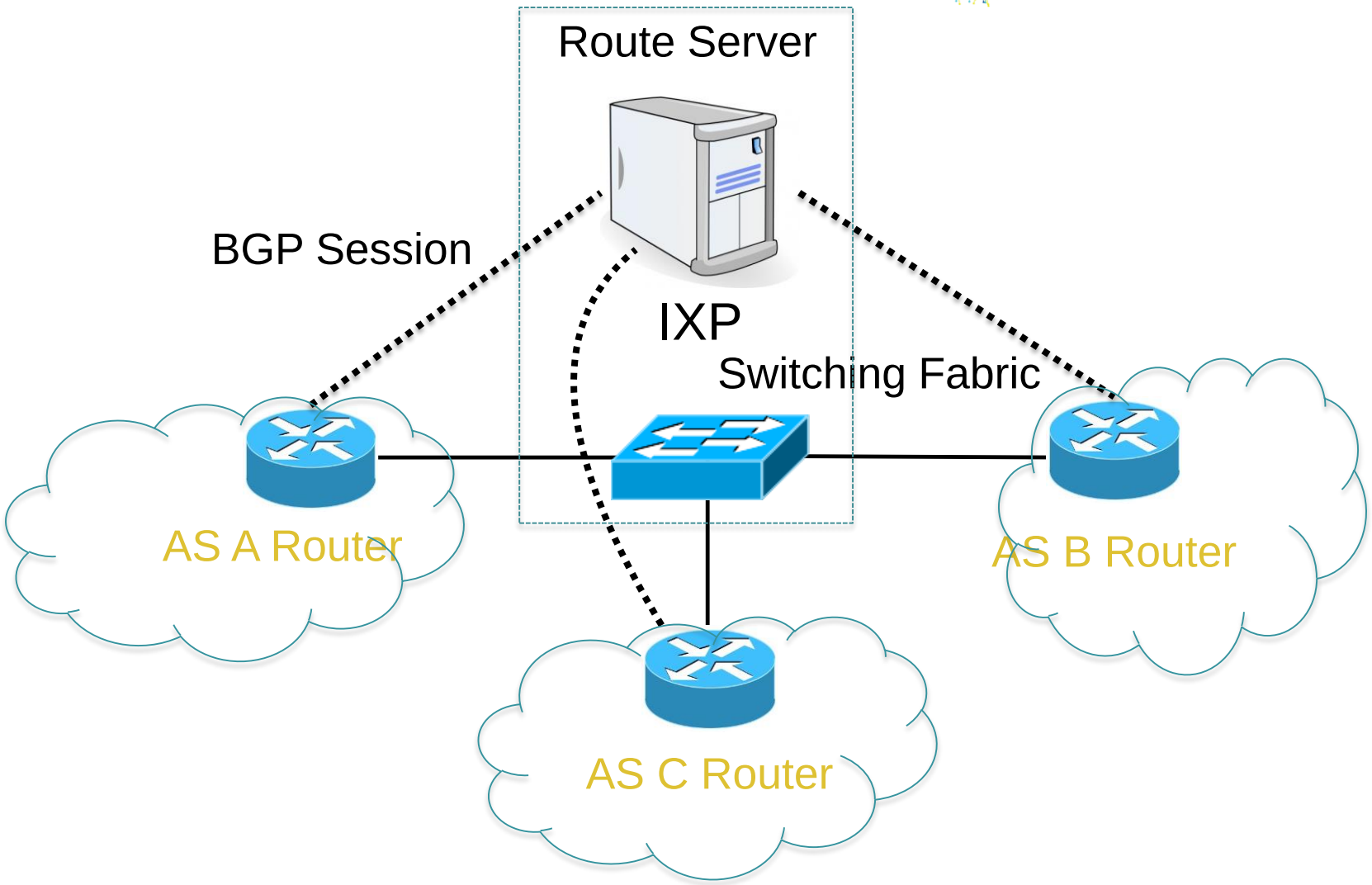
Disclaimer



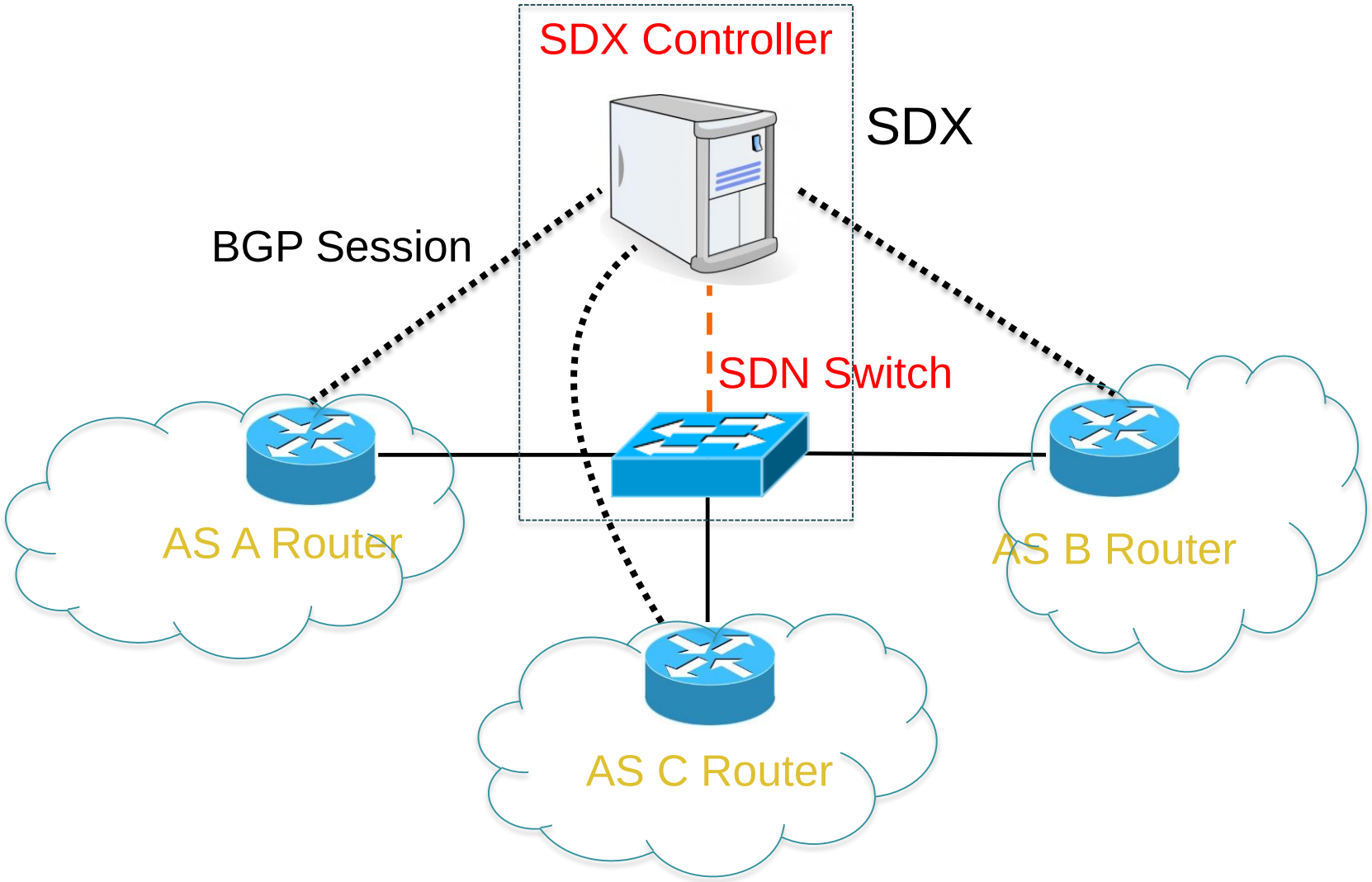
Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, you are granted the Licensed Rights in consideration of your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions. For full details, explanations, and examples of the use of this License, please visit: http://creativecommons.org/licenses/by/4.0/legalcode. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

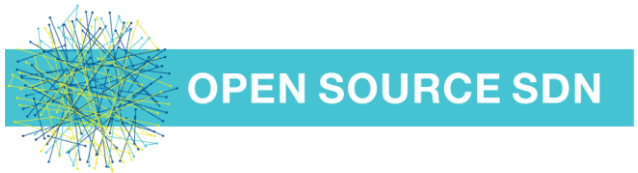
Internet Exchange Points (IXPs)



Software Defined IXPs (SDXs)

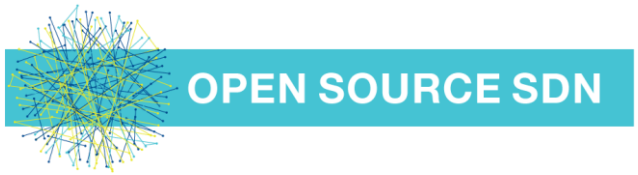


SDX Opens Up New Possibilities



- 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 black hole attack traffic

Deployment Ready SDX is Hard!

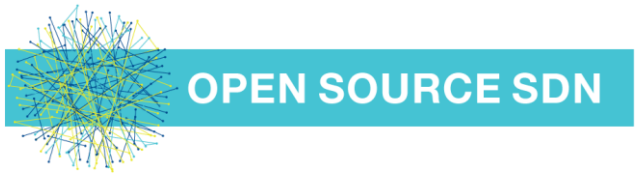


- **Deployment Experience:**
  - Inter-Agency Exchange
  - Large IXP in Europe
  - Smaller IXPs in Asia

- **Challenges:**
  - Scalability
  - …

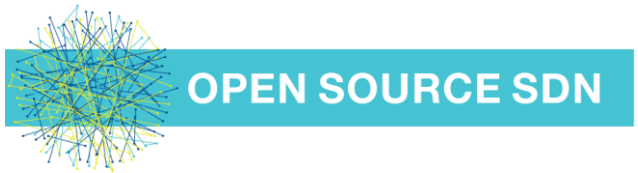
- We will focus on the **Scalability Challenge** today

Scalability Challenge



Devices	Operations	Data Plane Performance	
		State (# entries)	Update Rate (flow-mods/s)
	Match-Action on Multiple Headers	100K	2,500

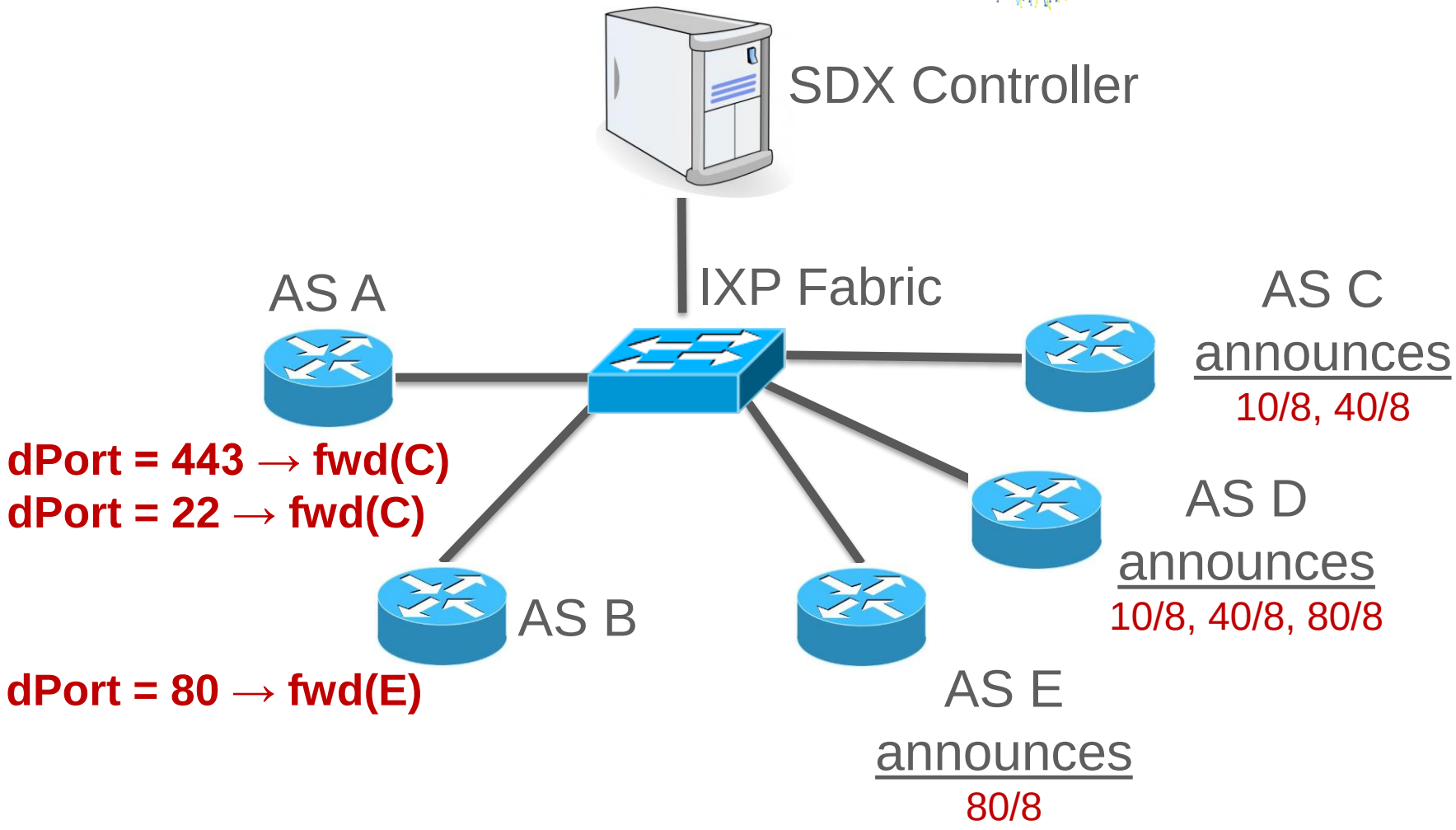
Scalability Challenge



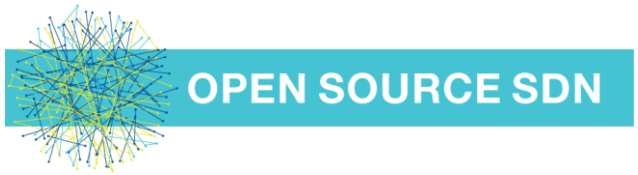
Devices	Operations	Data Plane Performance	
		State (# entries)	Update Rate (flow-mods/s)
	Match-Action on Multiple Headers	100K	2,500
	Matches on IP Prefixes only	~1M	N/A

Problem: Optimize the usage of available devices

Simple Example



Forwarding Table Entries at SDX



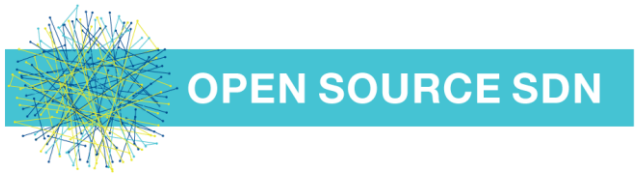
SDN Policies	# Forwarding Table Entries
dPort = 443 → fwd(C)	1
dPort = 22 → fwd(C)	1
dPort = 80 → fwd(E)	1

AS A

AS B

Number of forwarding table entries for A & B's Outbound SDN Policies

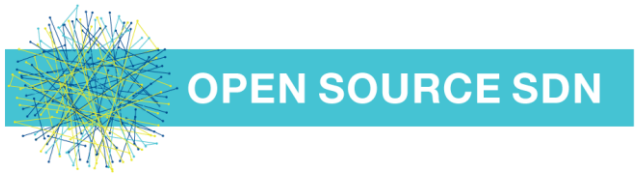
Goal Tracker



	Simple Example
Baseline	**3**

- **Large IXP Dataset:**
  - BGP RIBs & Updates from large IXP
  - 511 IXP participants
  - 96 million peering routes for 300K IP prefixes
  - 25K BGP updates for 2-hour duration

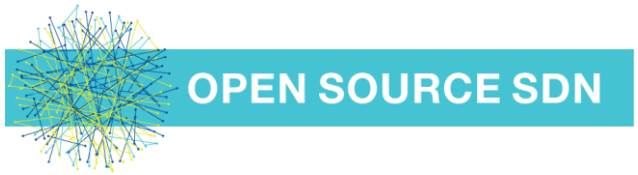
Goal Tracker



	Simple Example	Large IXP
Baseline	**3**	**62K (0)**

- **Large IXP Dataset:**
  - BGP RIBs & Updates from large IXP
  - 511 IXP participants
  - 96 million peering routes for 300K IP prefixes
  - 25K BGP updates for 2-hour duration

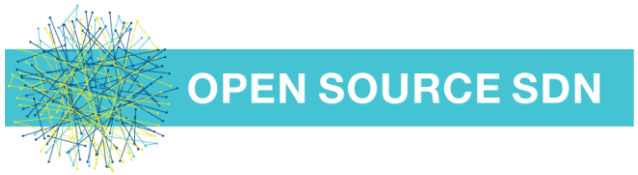
Goal Tracker



	Simple Example	Large IXP
Baseline	**3**	**62K (0)**

Satisfies design goals, but …

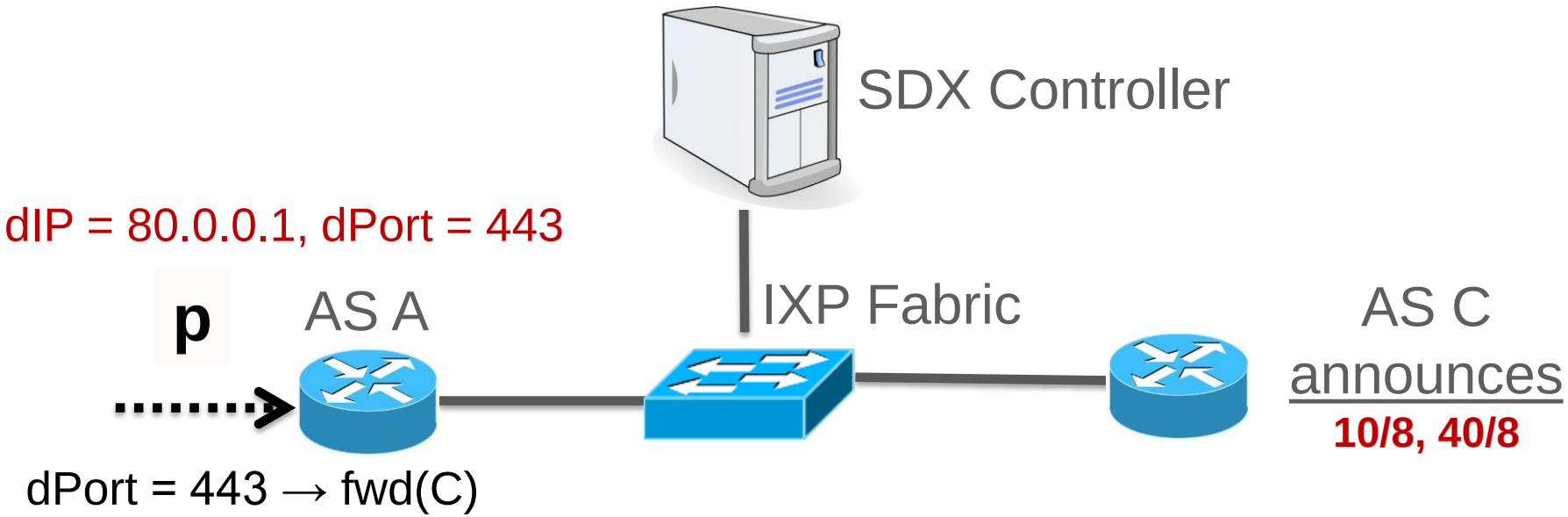
Goal Tracker



	Simple Example	Large IXP
Baseline	**3**	**62K (0)**

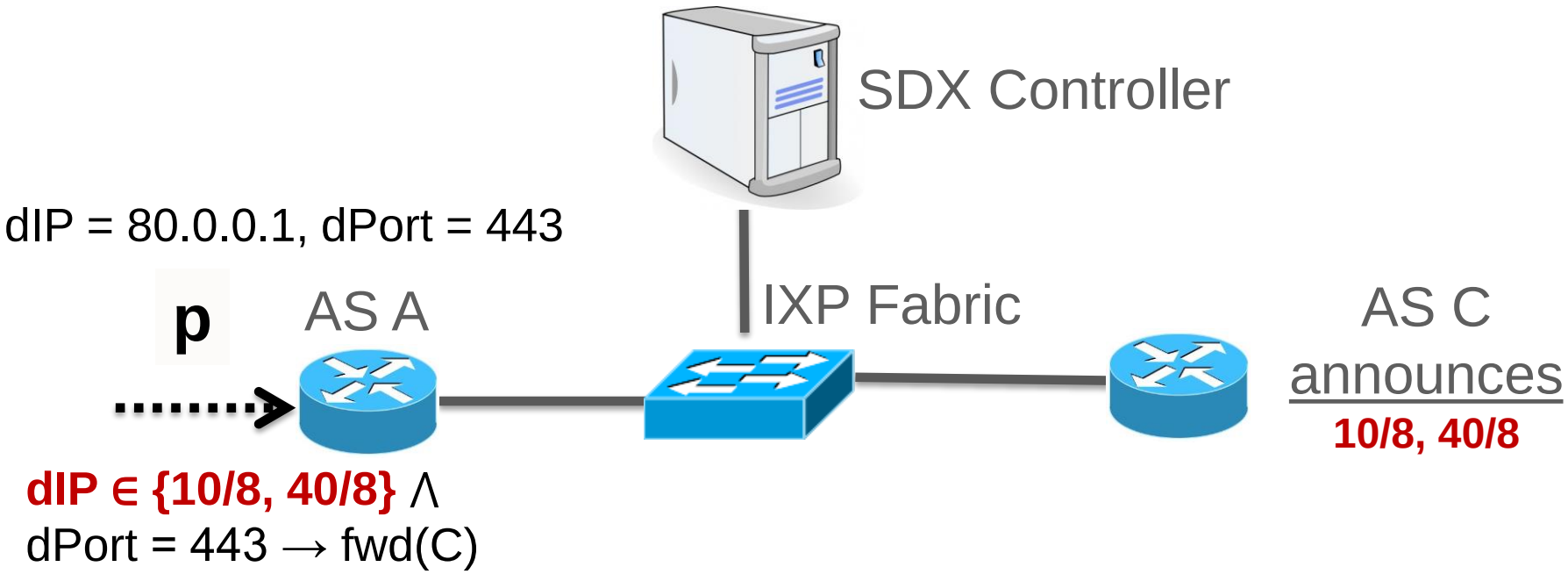
… not congruent with BGP!

Challenge: Congruence with BGP



Ensure **p** is not forwarded to C

Solution: SDN Policy Augmentation



Match on prefixes advertised by C

Data Plane State Explosion!

OPEN SOURCE SDN

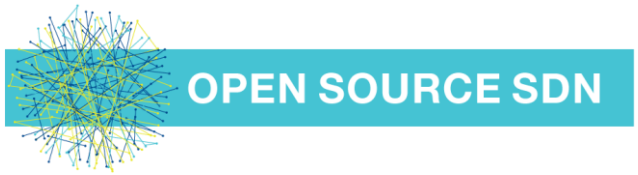
SDN Policies	# Forwarding Table Entries		
	10/8	40/8	80/8
dPort = 443 → fwd(C)	1	1	0
dPort = 22 → fwd(C)	1	1	0
dPort = 443 → fwd(D)	1	1	1

4

3

SDN Policy Augmentation increases forwarding table entries

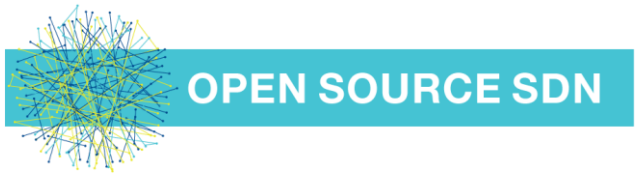
Goal Tracker



	Simple Example	Large IXP
Baseline	3	62K (0)
Policy Augmentation	**7**	**68M (16K)**

Not possible to support these many forwarding table entries and update rate!

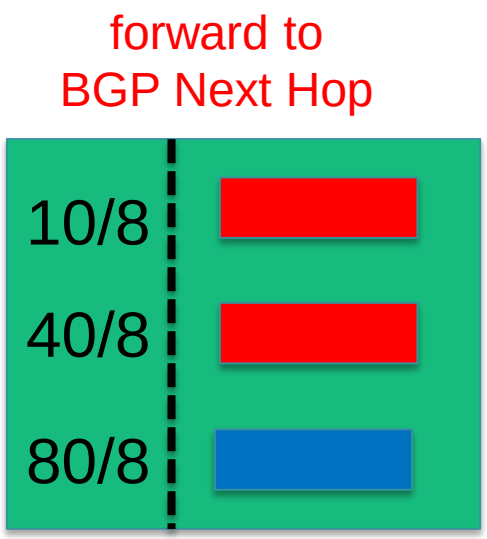
Forwarding Equivalence Classes

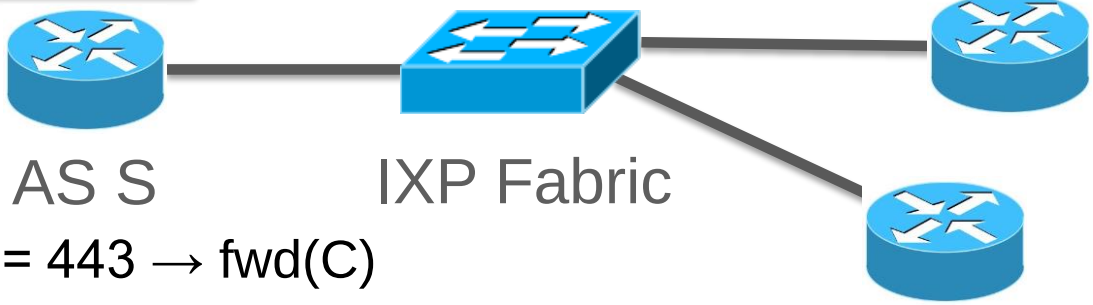


SDN Policies	# Forwarding Table Entries		
	10/8	40/8	80/8
dPort = 443 → fwd(C)	1	1	0
dPort = 22 → fwd(C)	1	1	0
dPort = 443 → fwd(D)	1	1	1

10/8, 40/8 exhibit similar forwarding behavior

Leveraging Forwarding Equivalence





AS S

IXP Fabric

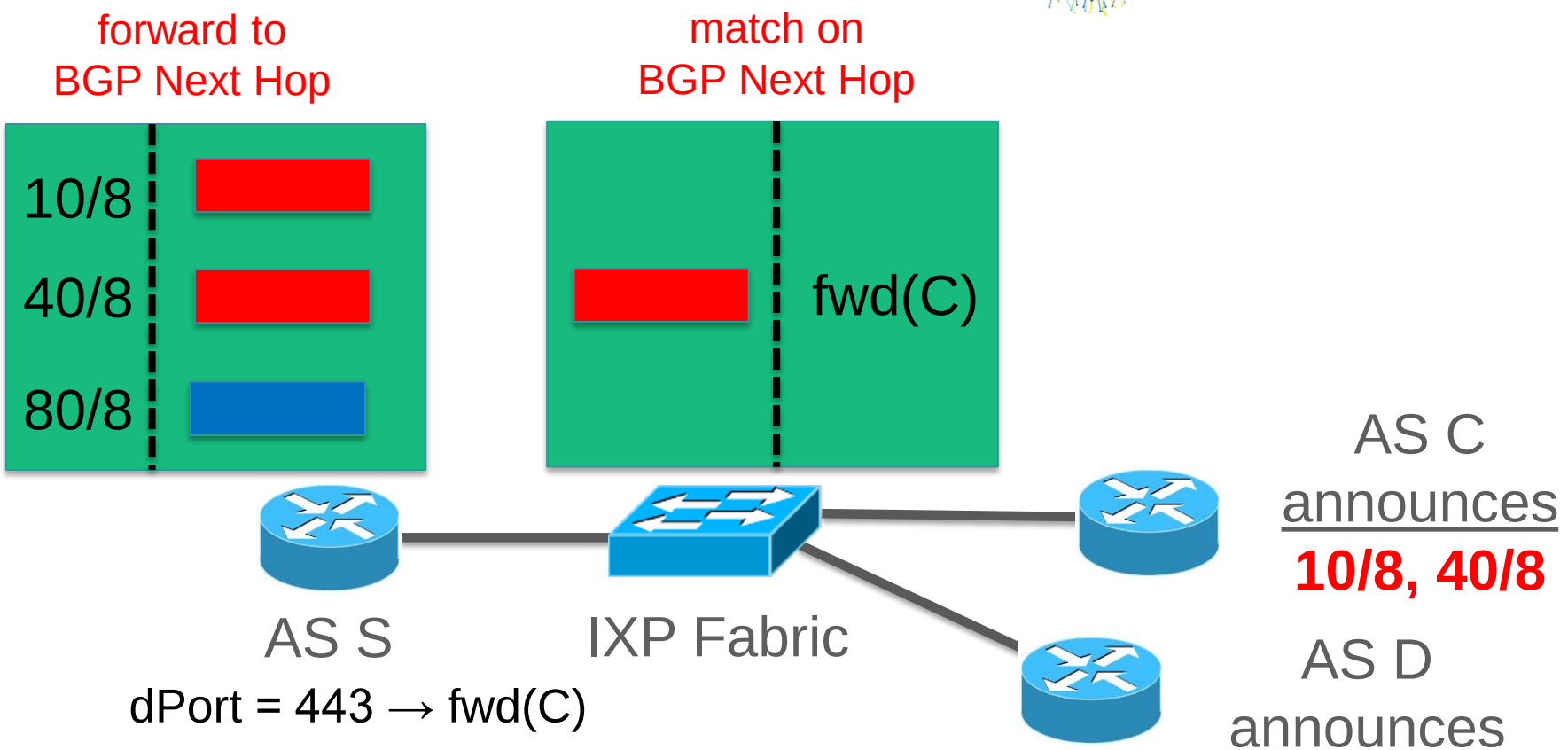
AS C
announces
10/8, 40/8

AS D
announces
10/8, 40/8, 80/8

dPort = 443 → fwd(C)

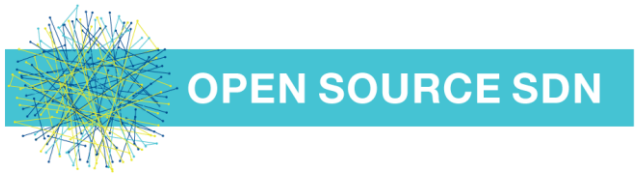
Single BGP Next Hop for 10/8, 40/8

Leveraging Forwarding Equivalence



Flow Rules at SDX match on BGP Next Hops

Goal Tracker

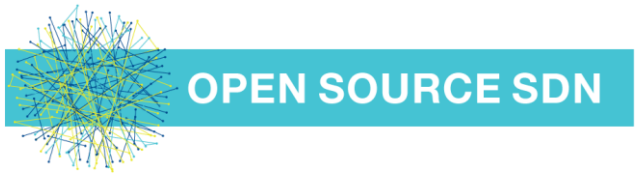


	Simple Example	Large IXP
Baseline	3	62K (0)
Policy Augmentation	7	68M (16K)
***FEC Computation**	**4**	**21M (35K)**

[*Gupta et al., SIGCOMM'14]

Still not possible to support these many forwarding table entries and update rate!

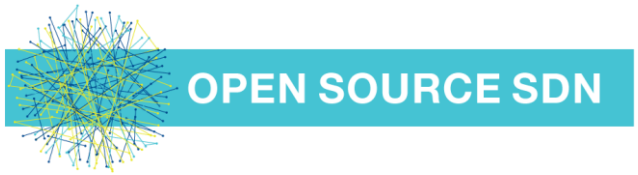
More Efficient FEC Computation



SDN Policies	# Forwarding Table Entries	
	{10/8, 40/8}	80/8
dPort = 443 → fwd(C)	1	0
dPort = 22 → fwd(C)	1	0
dPort = 443 → fwd(D)	1	1

Independent FEC Computation can be more efficient

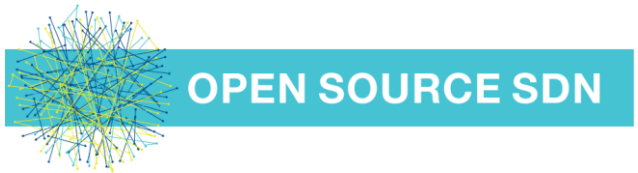
Partitioning FEC Computation



- Large number of SDX participants
  - Many different policies on groups of prefixes
  - Leads to a large number of small FECs of prefixes

- Compute FECs independently
  - Separate computation per participant
  - Leads to small number of large FECs, and less frequent recomputation
  - Enables “scale out” of the FEC computation

FEC Computation Partitioning in Action



SDN Policies	# Forwarding Table Entries	
	{10/8, 40/8}	80/8
dPort = 443 → fwd(C)	1	0
dPort = 22 → fwd(C)	1	0

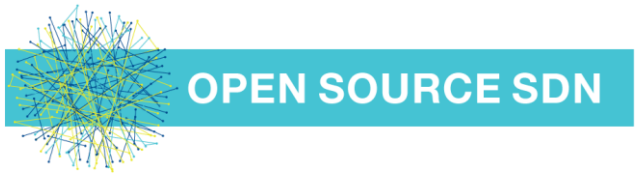
2

dPort = 443 → fwd(D)	1

1

A & B independently compute FECs

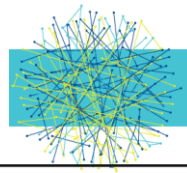
Goal Tracker



	Simple Example	Large IXP
Baseline	3	62K (0)
Policy Augmentation	7	68M (16K)
FEC Computation	4	21M (35K)
Independent FEC Computation	**3**	**763K (15K)**

Still not possible to support these many forwarding table entries and update rate!

Undesired BGP & SDN Coupling



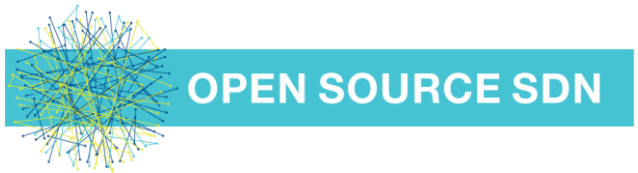
SDN Policies	# Forwarding Table Entries		
	10/8	40/8	80/8
dPort = 443 → fwd(C)	1	1	0
dPort = 22 → fwd(C)	1	1	0

dPort = 443 → fwd(D)	1 → 0	1	1

Incoming BGP Update:

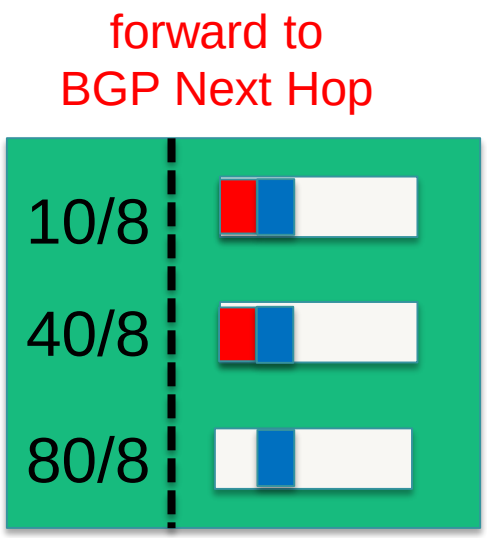
{AS D withdraws route for prefix 10/8}

Decoupling BGP from SDN



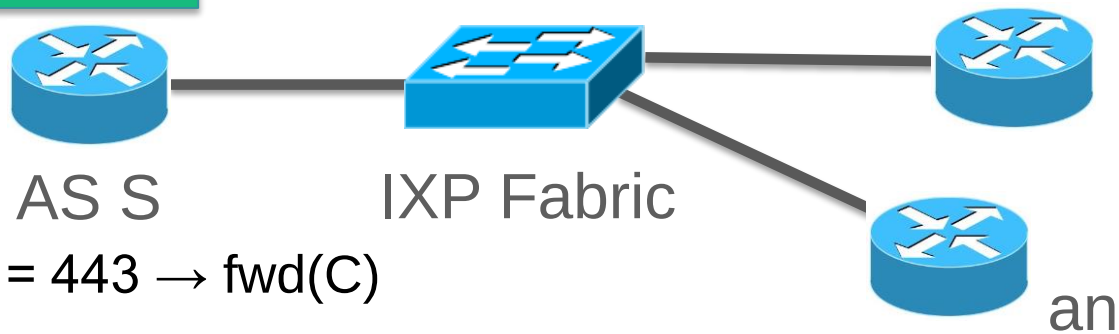
- Leverage advances in commodity hw switches
  - Support for Bitmask Matching (OF 1.3)
- Extend BGP “next hop” encoding
  - So far: encode FECs (single field)
  - New idea: encode **reachability bitmask** (multi field)
- Changing only the BGP announcements
  - No need to update the SDX data plane!

Reachability Bitmask in Action



Dedicate one bit per participant

Reachable via AS C



AS C announces **10/8, 40/8**

AS S

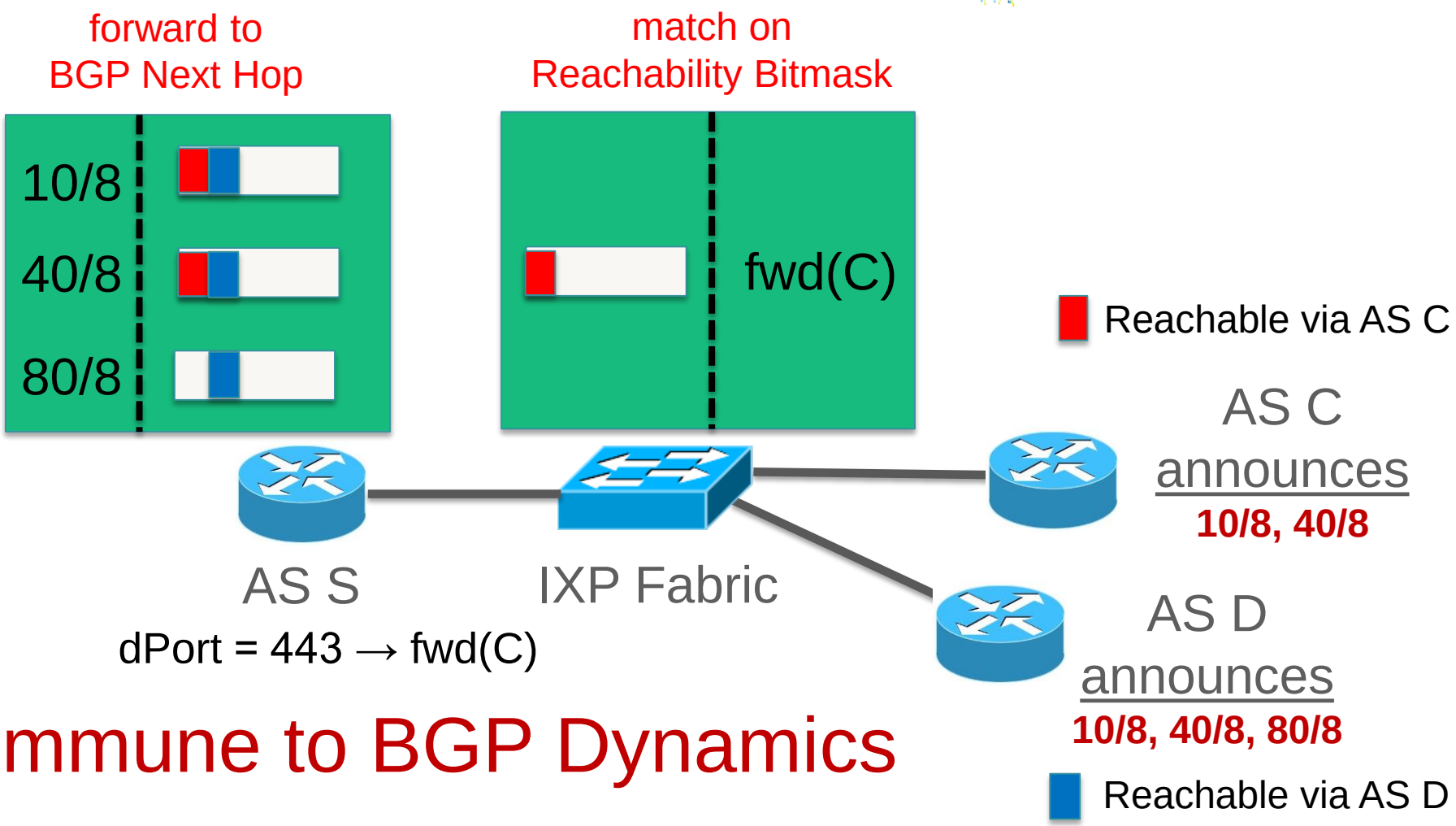
IXP Fabric

AS D announces **10/8, 40/8, 80/8**

dPort = 443 → fwd(C)

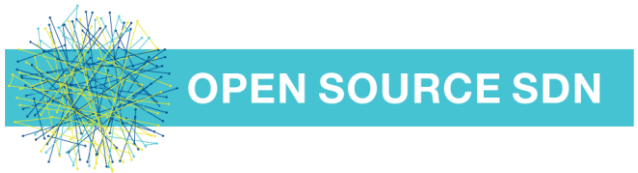
Reachable via AS D

Reachability Bitmask in Action



Immune to BGP Dynamics

Reachability Bitmask in Action



SDN Policies	# Forwarding Table Entries
	C
dPort = 443 → fwd(C)	1
dPort = 22 → fwd(C)	1

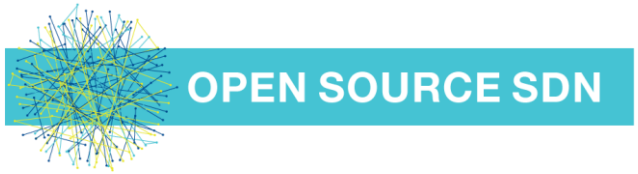
2

dPort = 443 → fwd(D)	1

1

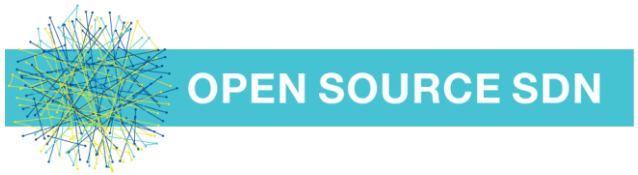
Reduces Data Plane State

Goal Tracker



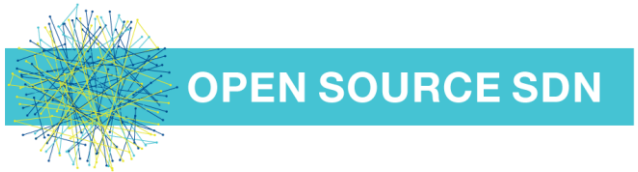
	Simple Example	Large IXP
Baseline	3	62K (0)
Policy Augmentation	7	68M (16K)
FEC Computation	4	21M (35K)
Independent FEC Computation	3	763K (15K)
Reachability Encoding	**3**	**65K (0)**

iSDX Evaluation Summary



- **Data Plane State:**
  - Requires **65K < 100K** forwarding table entries
- **Data Plane Update Rate:**
  - Requires **0** < **2500** updates/second
- **Other Goals:**
  - Processes BGP update bursts in real time **(50 ms)**
  - Requires only **360 BGP Next Hops** compared to 25K from previous solutions

You Can Run iSDX Today!



http://sdx.cs.princeton.edu

- Running code
  - Vagrant & Docker based setup
  - Instructions to run with **Hardware Switches**
- Ongoing efforts
  - Hosted by **Open Networking Foundation**
  - Community Link: https://community.opensourcesdn.org/wg/iSDX/dashboard
  - Mailing List (general info, anyone in the world can register): isdx@community.OpenSourceSDN.org
  - More info and project landing page: https://www.OpenSourceSDN.org
- Deployment
  - Inter-agency exchange
  - IXPs in Europe & Asia