Novel Applications for a SDN-enabled Internet Exchange Point

Laurent Vanbever

vanbever@cs.princeton.edu

RIPE 67, Athens

October, 14 2013

Joint work with

Arpit Gupta, Muhammad Shahbaz, Hyojoon Kim,
Russ Clark, Nick Feamster, Jennifer Rexford and Scott Shenker

BGP is notoriously unflexible

and difficult to manage

BGP is notoriously unflexible and difficult to manage

Fwd paradigm

Fwd control

Fwd influence

BGP is notoriously unflexible and difficult to manage

	BGP
Fwd paradigm	destination-based
Fwd control	indirect *protocol configuration*
Fwd influence	local *at the BGP session level*

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

	BGP	SDN
Fwd paradigm	destination-based	any *source addr, ports, VLAN, etc.*
Fwd control	indirect *protocol configuration*	direct *via an open API (e.g., OpenFlow)*
Fwd influence	local *at the BGP session level*	global *via remote controller control*

Internet Exchange Points are perfect places to deploy new interdomain features

Internet Exchange Points (IXPs) ...

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

Internet Exchange Points (IXPs)	AMS-IX (*):
connect a large number of participants	> 600 participants

(*) See https://www.ams-ix.net

Internet Exchange Points are perfect places to deploy new interdomain features

Internet Exchange Points (IXPs)	AMS-IX (*):
■ connect a large number of participants	> 600 participants
■ carry a large amount of traffic	> 2400 Gb/s (peak)

(*) See https://www.ams-ix.net

Internet Exchange Points are perfect places to deploy new interdomain features

Internet Exchange Points (IXPs)	AMS-IX (*):
connect a large number of participants	> 600 participants
carry a large amount of traffic	> 2400 Gb/s (peak)
are a hotbed of innovation	BGP Route Server Mobile peering Open peering ...

(*) See https://www.ams-ix.net

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!

SDX = SDN + IXP

Augment IXP with SDN capabilities

default forwarding and routing behavior is unchanged

Enable fine-grained interdomain policies

simplifying network operations

... with scalability in mind

support the load of a large IXP

What does SDX enable that was
hard or impossible to do before?

SDX enables a wide range of novel applications

security	Prevent/block policy violation
	Prevent participants communication
forwarding optimization	Middlebox traffic steering
	Traffic offloading
	Inbound Traffic Engineering
	Fast convergence
peering	Application-specific peering
remote-control	Upstream blocking of DoS attacks
	Influence BGP path selection
	Wide-area load balancing

Novel Applications for a SDN-enabled Internet Exchange Point

1 **SDX Architecture**
data- and control-plane

2 **App#1: Inbound TE**
easy and deterministic

3 **App#2: Fast convergence**
<1s after peering link failure

Novel Applications for a SDN-enabled Internet Exchange Point

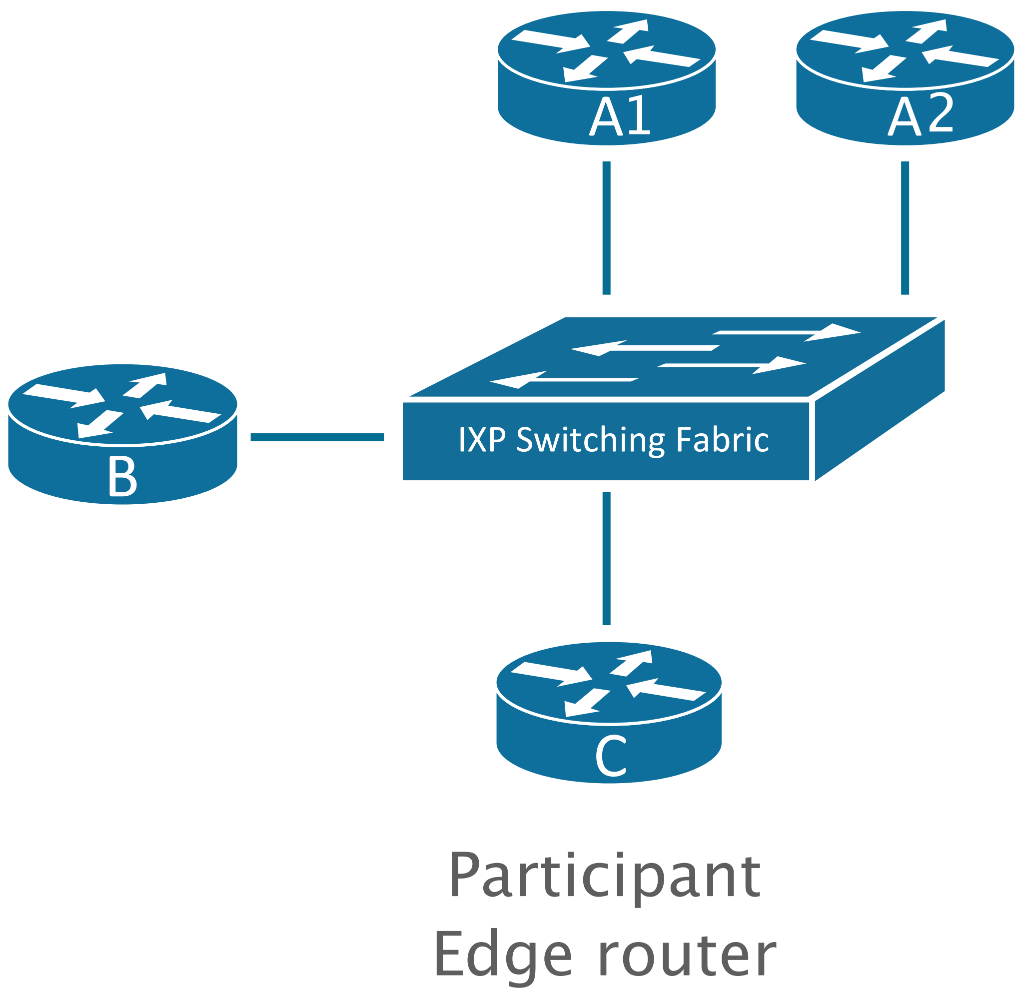
1 SDX Architecture
data- and control-plane

App#1: Inbound TE
easy and deterministic

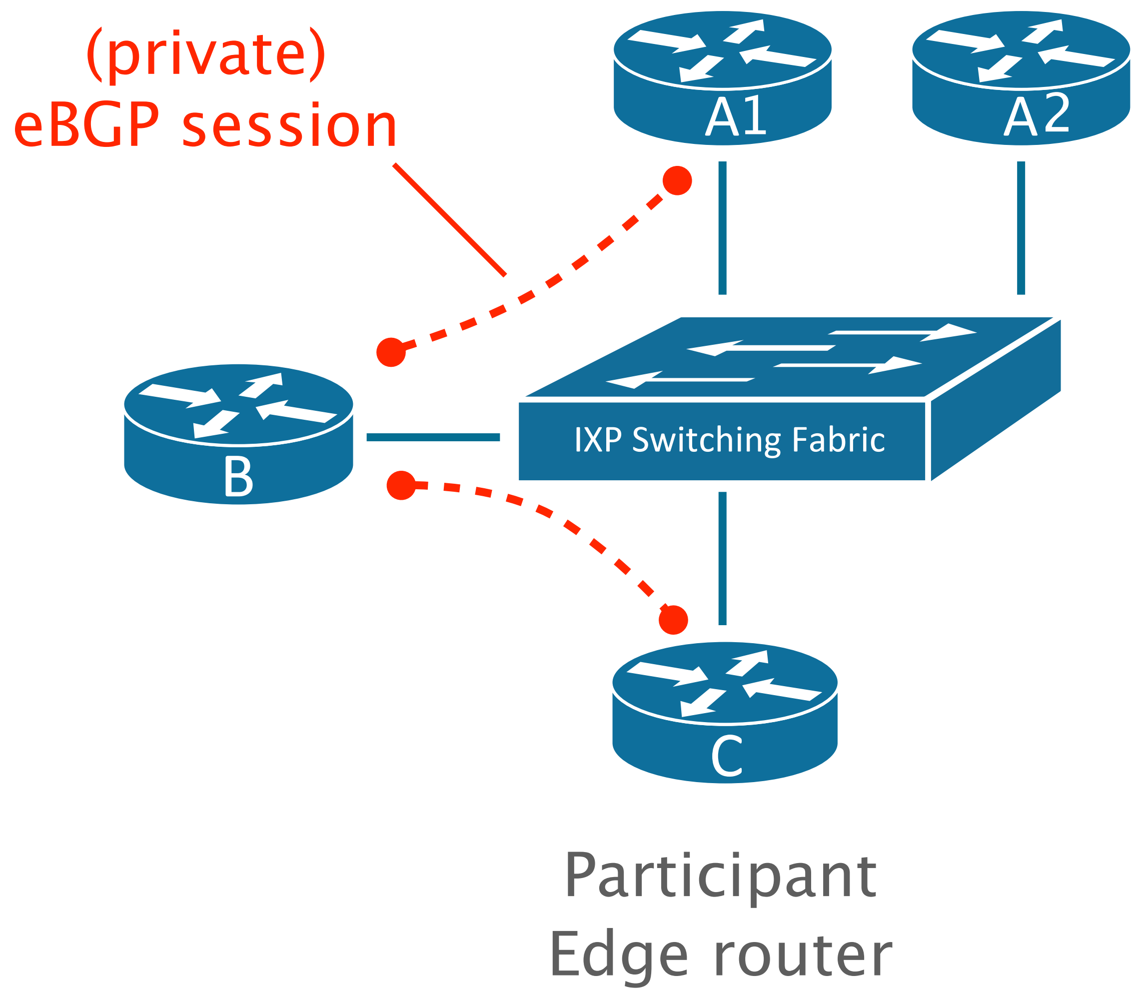
App#2: Fast convergence
<1s after peering link failure

An IXP is a large L2 domain where participant routers peer using BGP

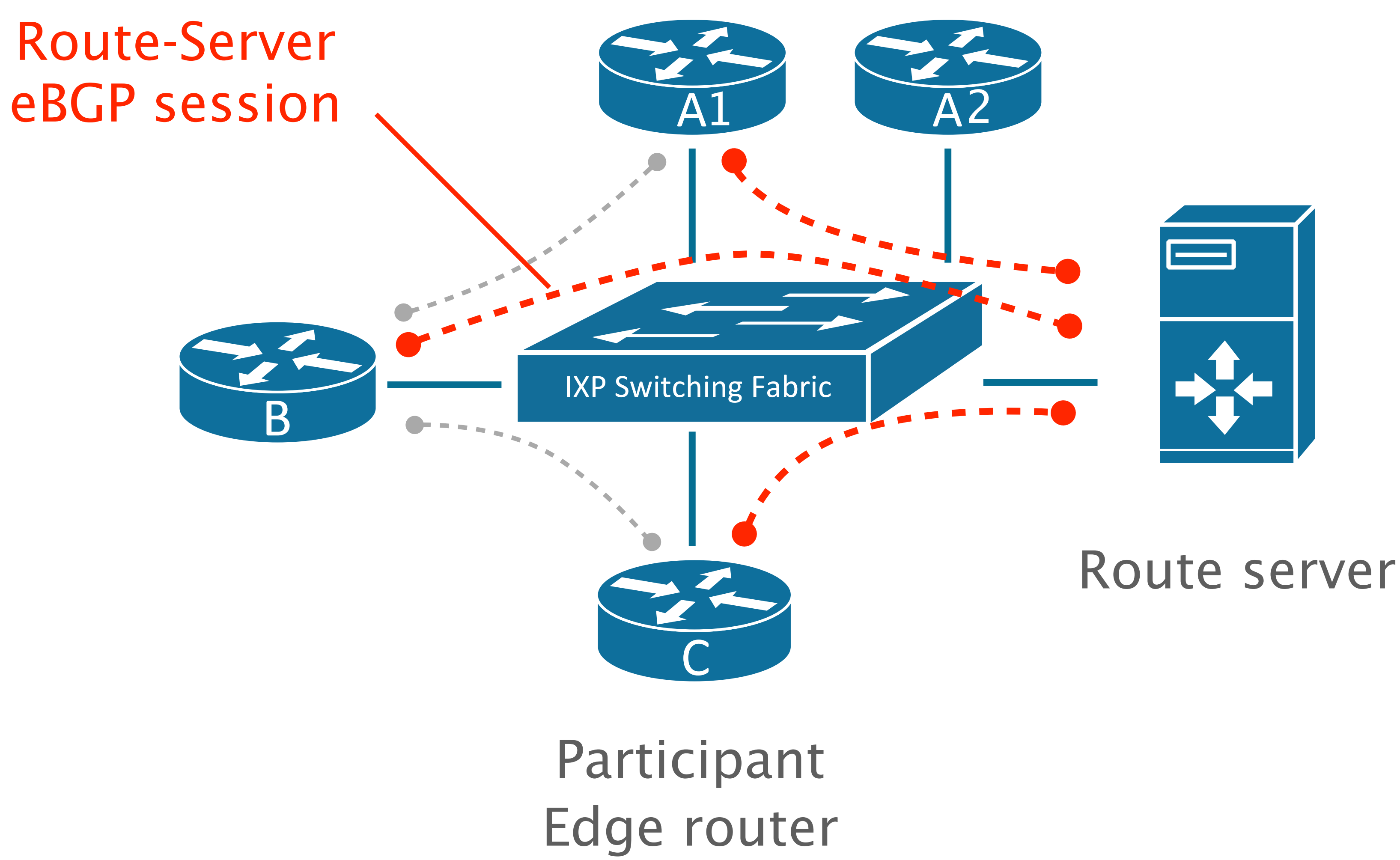
An IXP is a large L2 domain where participant routers peer using BGP



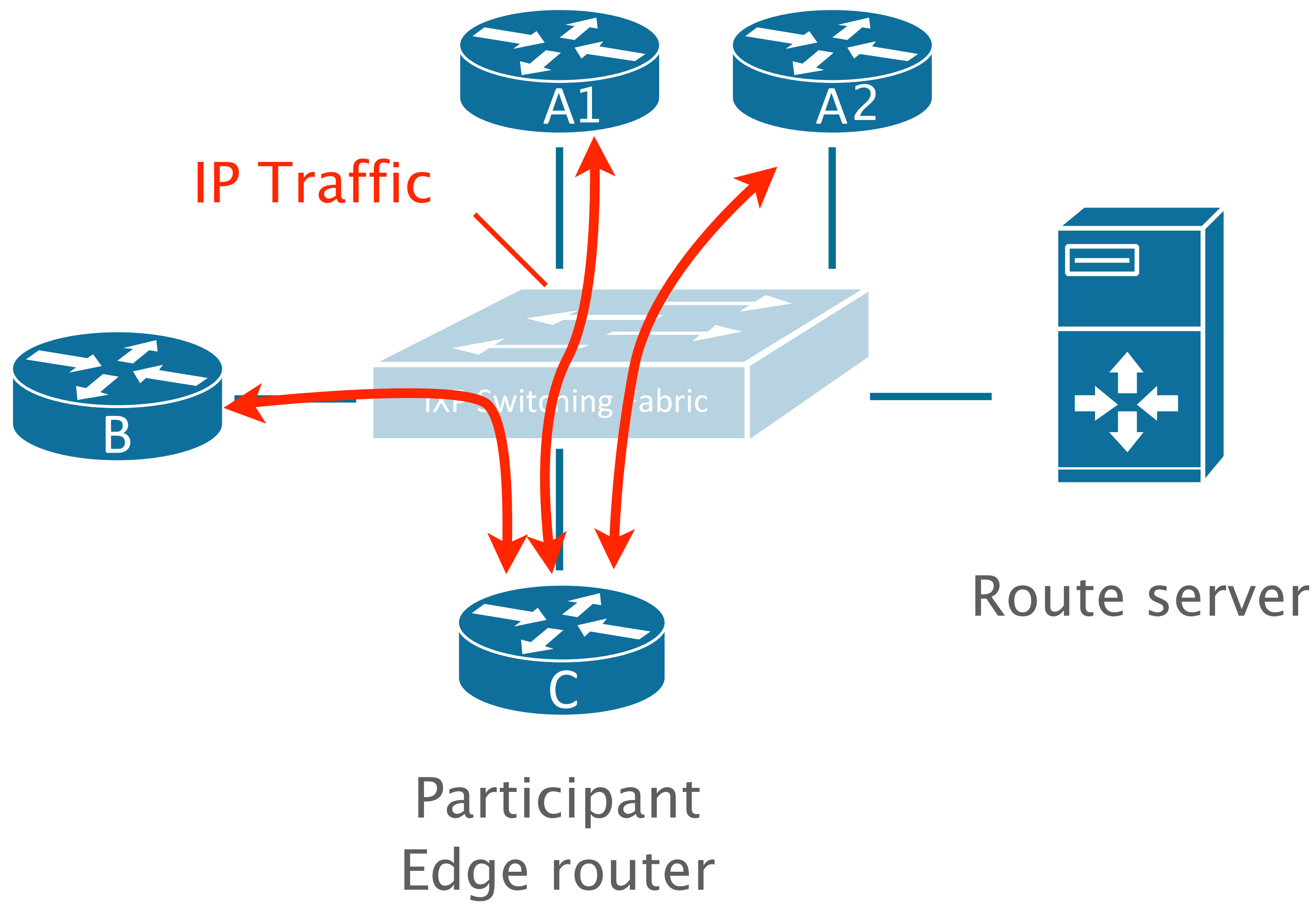
An IXP is a large L2 domain where participant routers peer using BGP



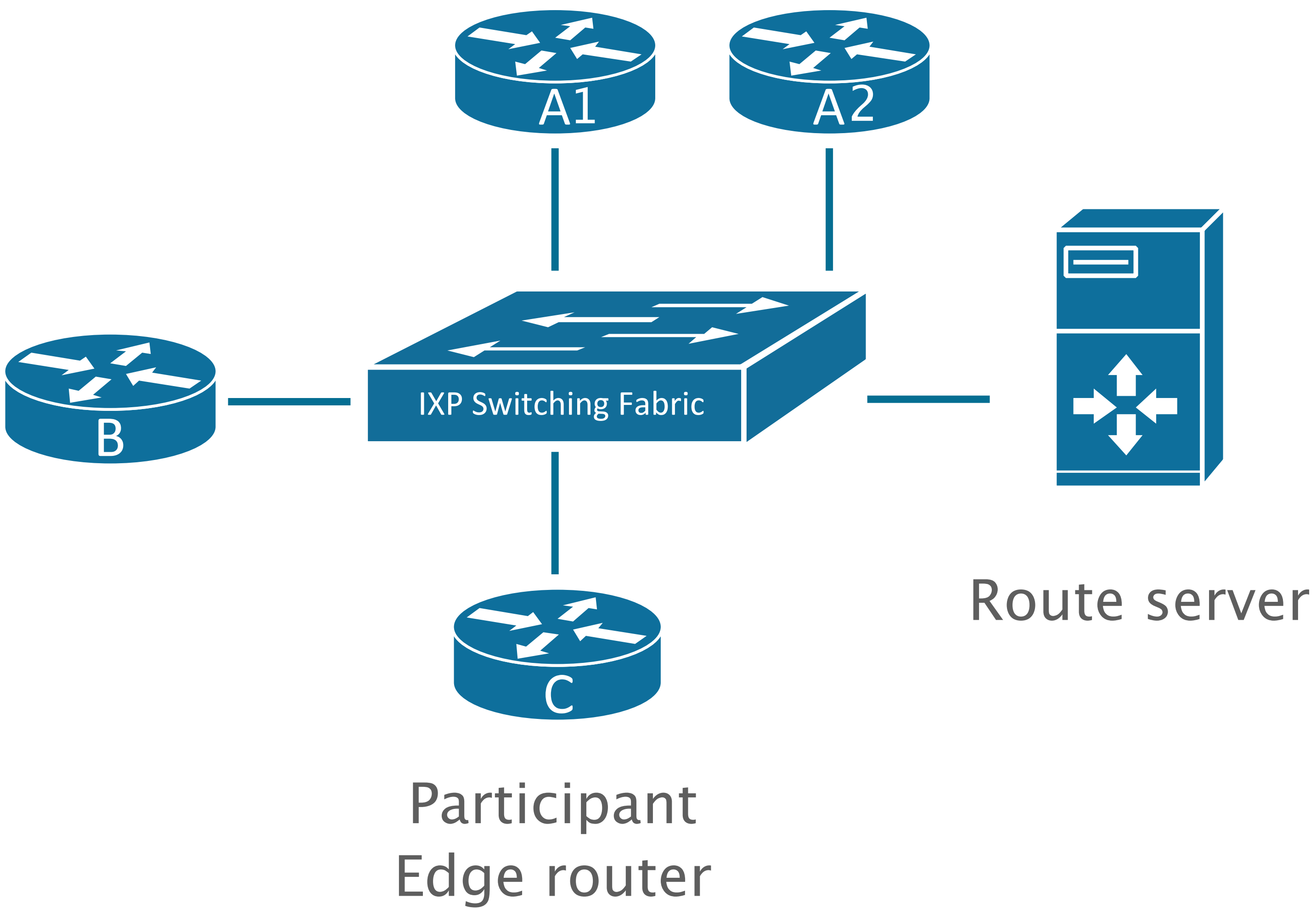
An IXP is a large L2 domain where participant routers peer using BGP



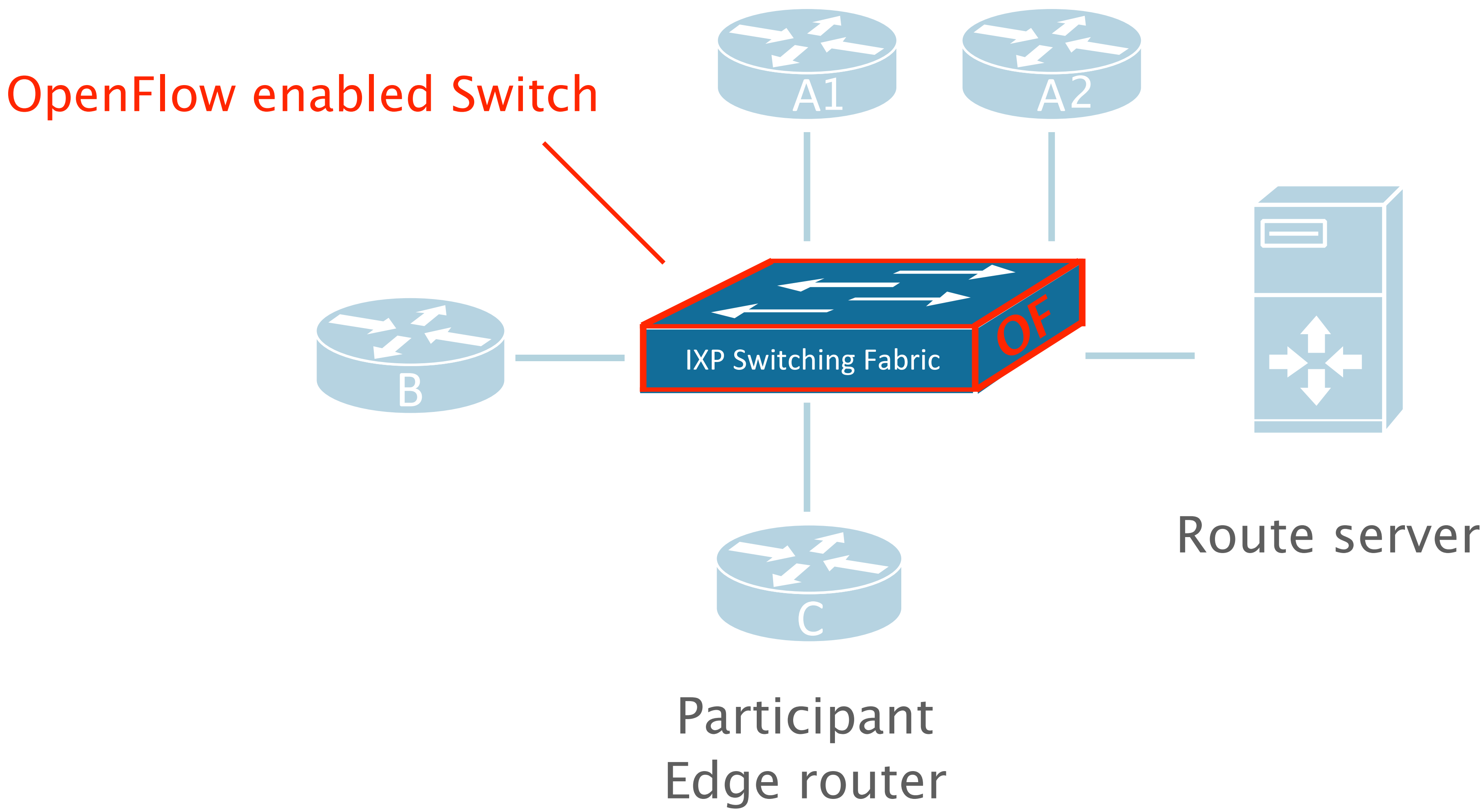
An IXP is a large L2 domain where participant routers peer using BGP



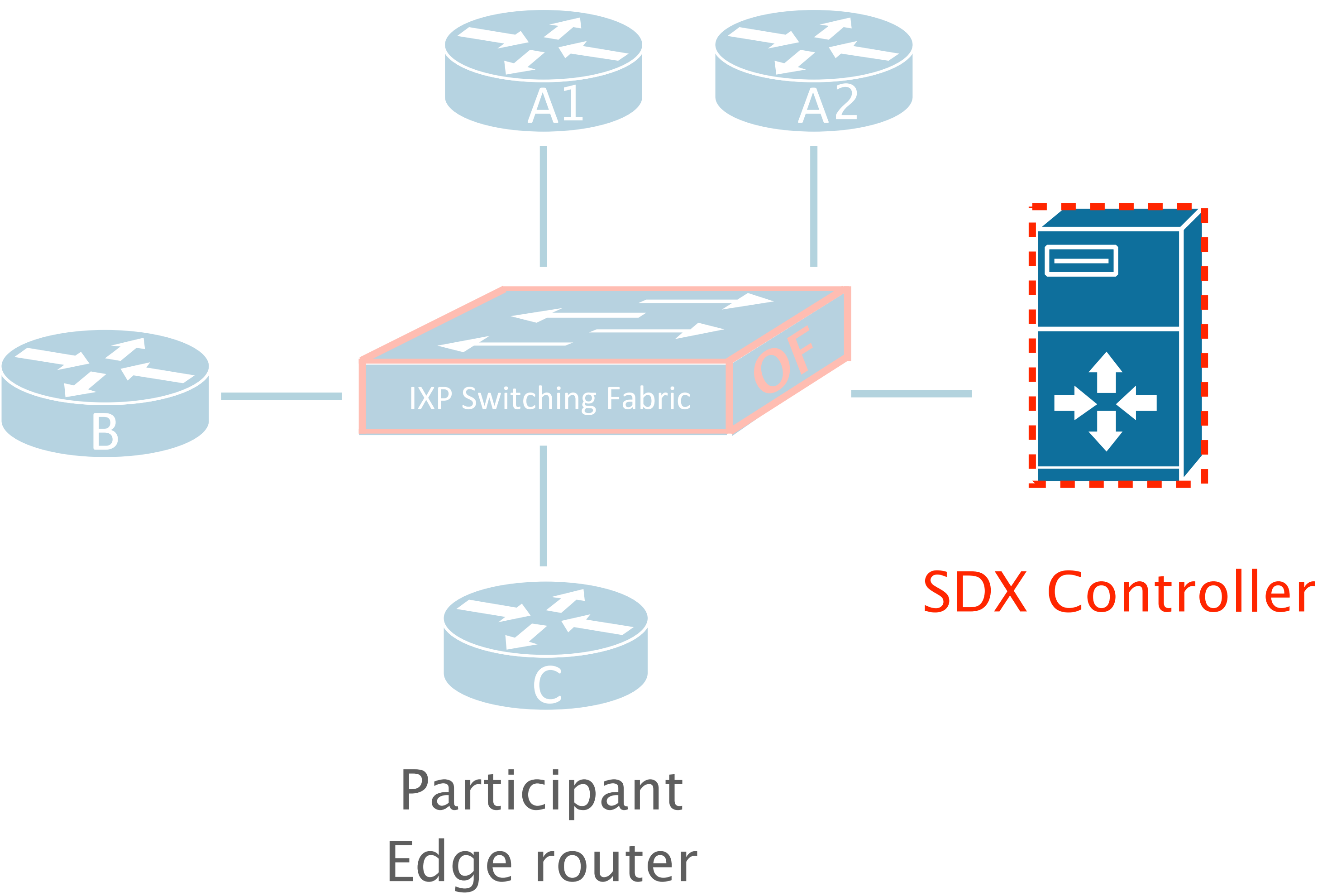
With respect to IXPs, SDN-enabled IXPs (SDX) ...



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 SDN controller



SDX participants write their inter domain policies using a high-level language built on top of Pyretic (*)

(*) See http://frenetic-lang.org/pyretic/

SDX policies are composed of
a *pattern* and some *actions*

```
match (      Pattern      ), then (  Actions  )
```

(*) See http://frenetic-lang.org/pyretic/

Pattern selects packets based on any header fields, while *Actions* forward or modify the selected packets

```
         Pattern                              Action
         eth_type
         vlan_id                                drop
         srcmac                                 forward
match (  dstmac    , &&, ||  ), then (          rewrite   )
         protocol
         dstip
         tos
         srcip
         srcport
         dstport
```

(*) See http://frenetic-lang.org/pyretic/

Each SDX participant writes her policies independently

Participant A's policy:

```
match(dstip=ipA.1), fwd(A1)
match(dstip=ipA.2), fwd(A2)
```

Participant B's policy:

```
match(dstip=ipC), fwd(C)
match(dstip=ipA), fwd(A)
match(dstip=ipB), fwd(B)
```



```
match(dstip=ipC), fwd(C)
```

Participant C's policy:

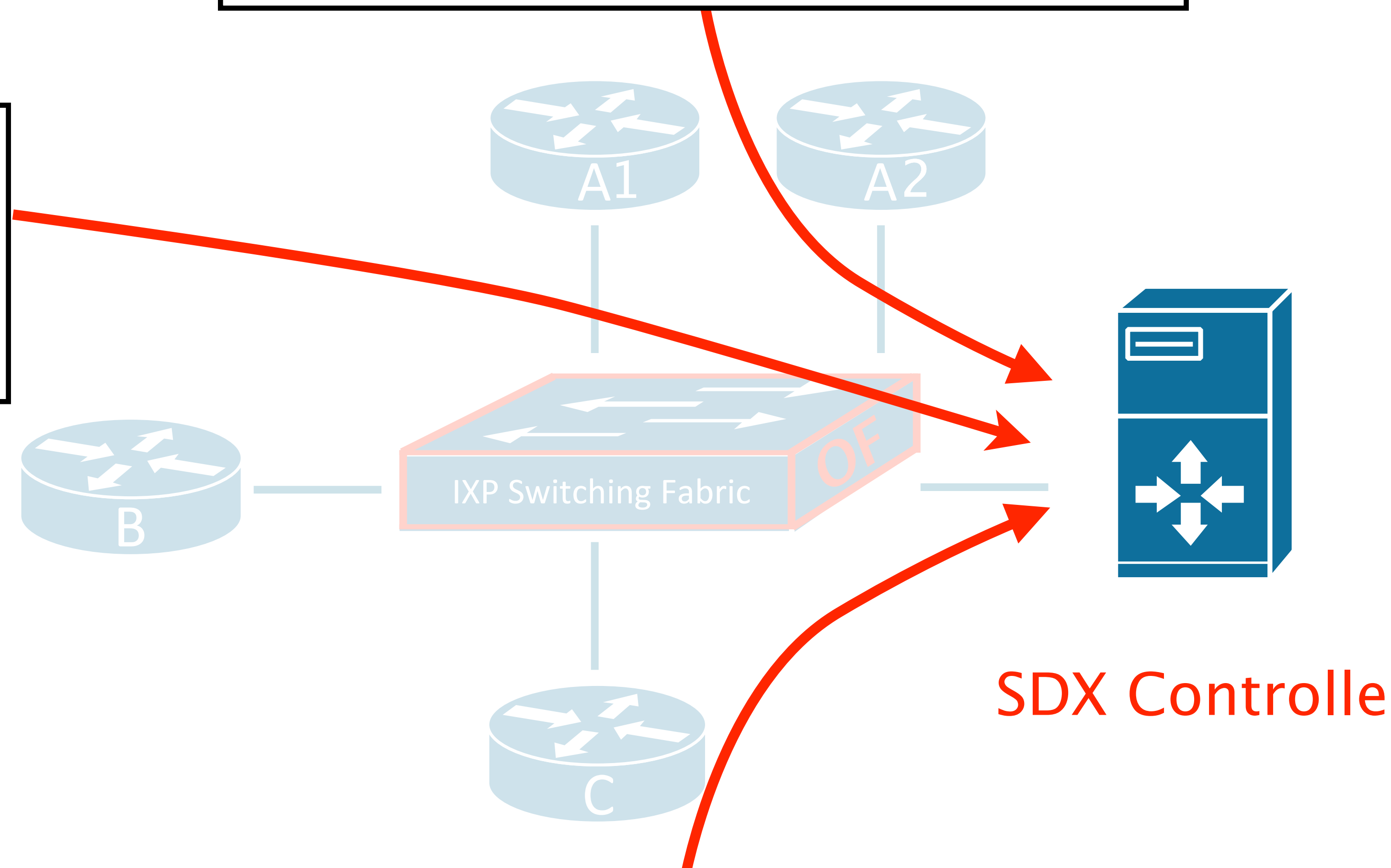
The SDX controller composes these policies together ensuring *isolation* and *correctness*

Participant A's policy:

```
match(dstip=ipA.1), fwd(A1)
match(dstip=ipA.2), fwd(A2)
```

Participant B's policy:

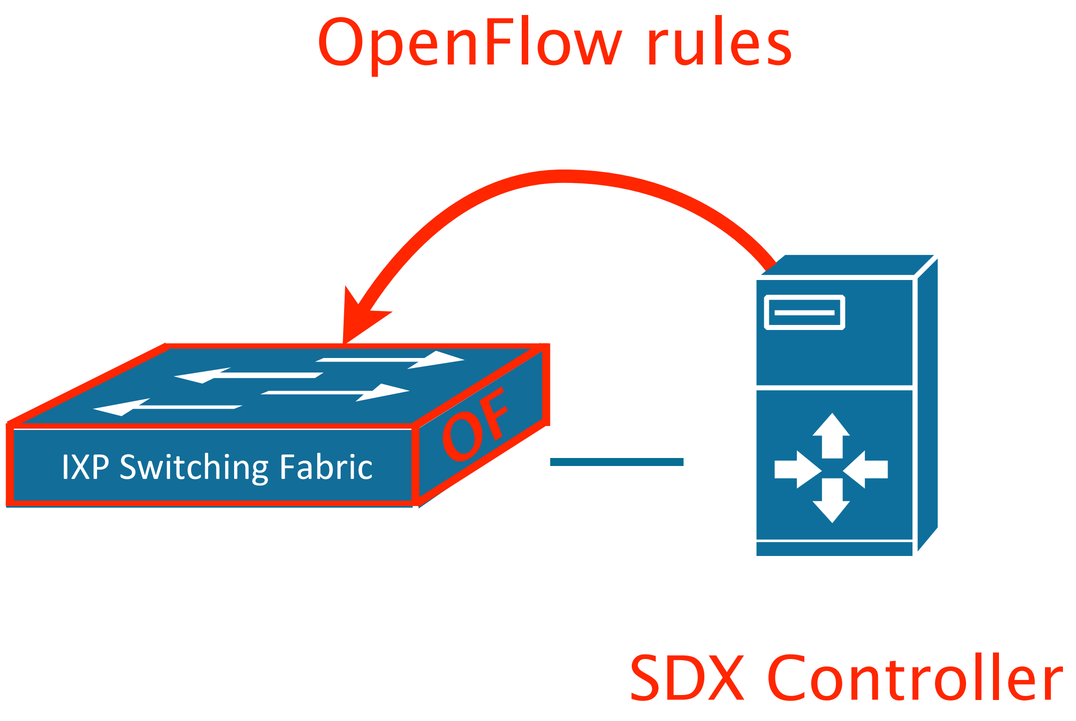
```
match(dstip=ipC), fwd(C)
match(dstip=ipA), fwd(A)
match(dstip=ipB), fwd(B)
```




```
match(dstip=ipC), fwd(C)
```

Participant C's policy:

After compiling the policies, the SDX controller provisions the IXP data plane using OpenFlow



Building a SDX platform is challenging, but possible

Challenge #1: Authentication

How do we?

Check that it is legitimate for remote participants to provision a policy *P*?

Challenge #1: Authentication

How do we?

Check that it is legitimate for remote participants to provision a policy *P*?

We…

Use the RPKI system to authenticate policies scope

only the prefix owner can act on the traffic remotely

Challenge #2: Access control

How do we? Prevent participants from performing unwanted actions (*e.g.*, rewrite the source mac)?

Challenge #2: Access control

How do we? Prevent participants from performing unwanted actions (*e.g.*, rewrite the source mac)?

We… Use access-lists to limit the actions available to each participant

Challenge #3: Isolation

How do we? Avoid clashes between participant policies acting on the same traffic?

Challenge #3: Isolation

How do we?

Avoid clashes between participant policies acting on the same traffic?

We...

Use virtual topologies to limit participants' visibility

each participant can only talk with its own neighbors

Challenge #4: Scalability

How do we?

Manage millions of forwarding entries with hardware supporting only hundred thousands of them?

Challenge #4: Scalability

How do we?

Manage millions of forwarding entries with hardware supporting only hundred thousands of them?

We…

Leverage routers' routing tables

tailored for IP prefixes matching

Novel Applications for a SDN-enabled Internet Exchange Point

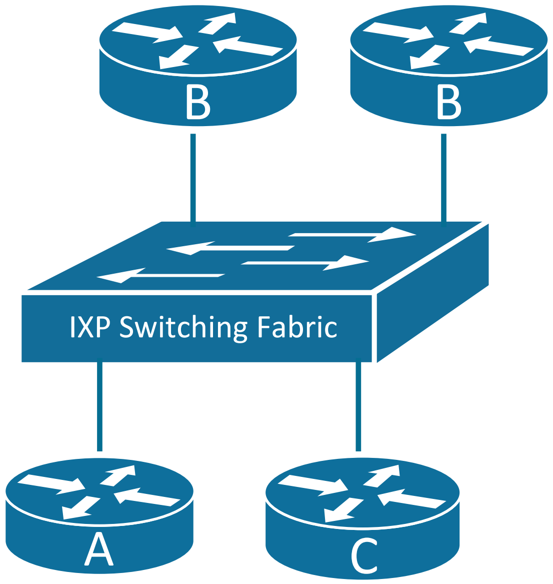
SDX Architecture
data- and control-plane

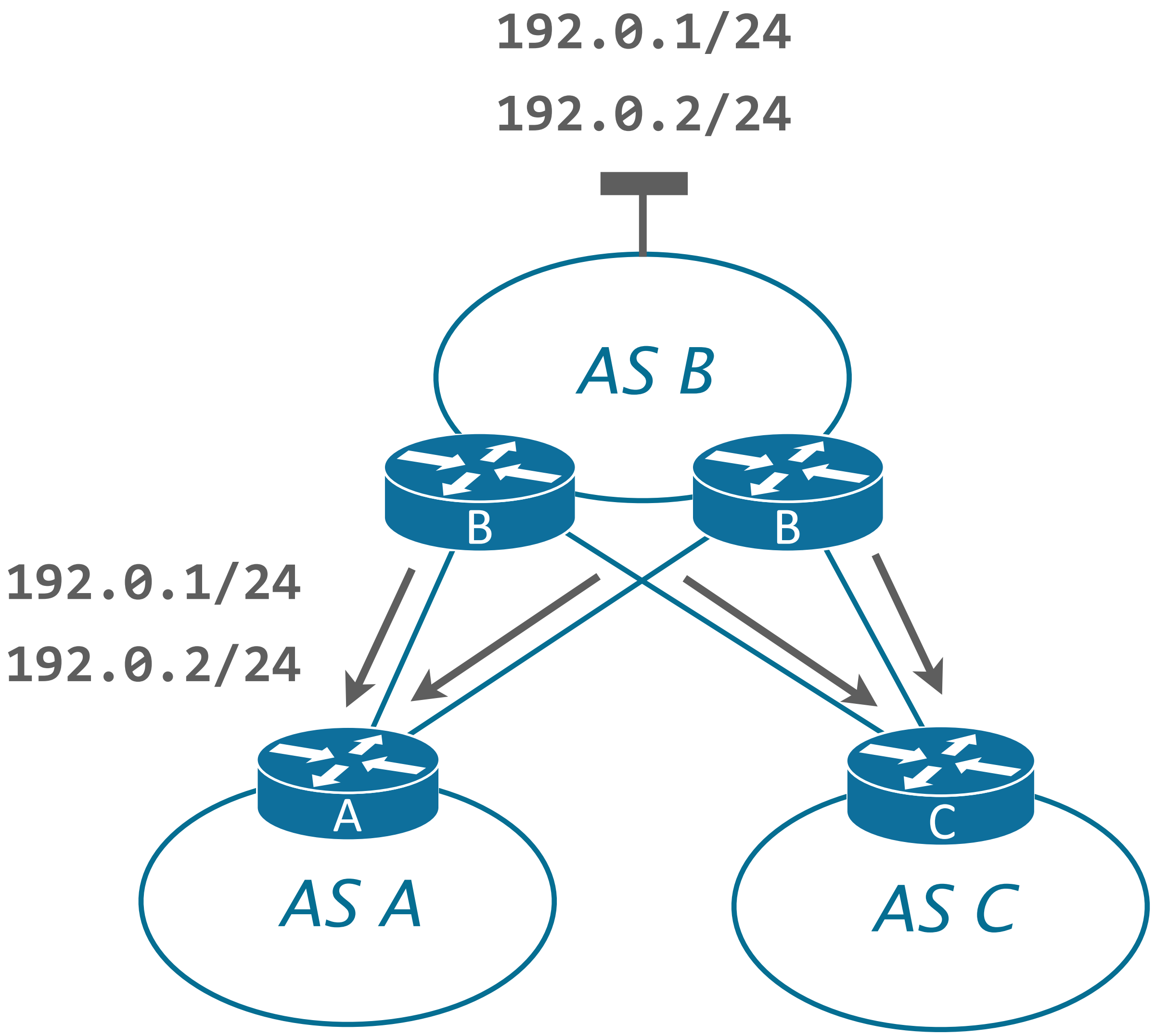
2 App#1: Inbound TE
easy and deterministic

App#2: Fast convergence
<1s after peering link failure

SDX can improve inbound traffic engineering

Given an IXP Physical Topology and a BGP topology,

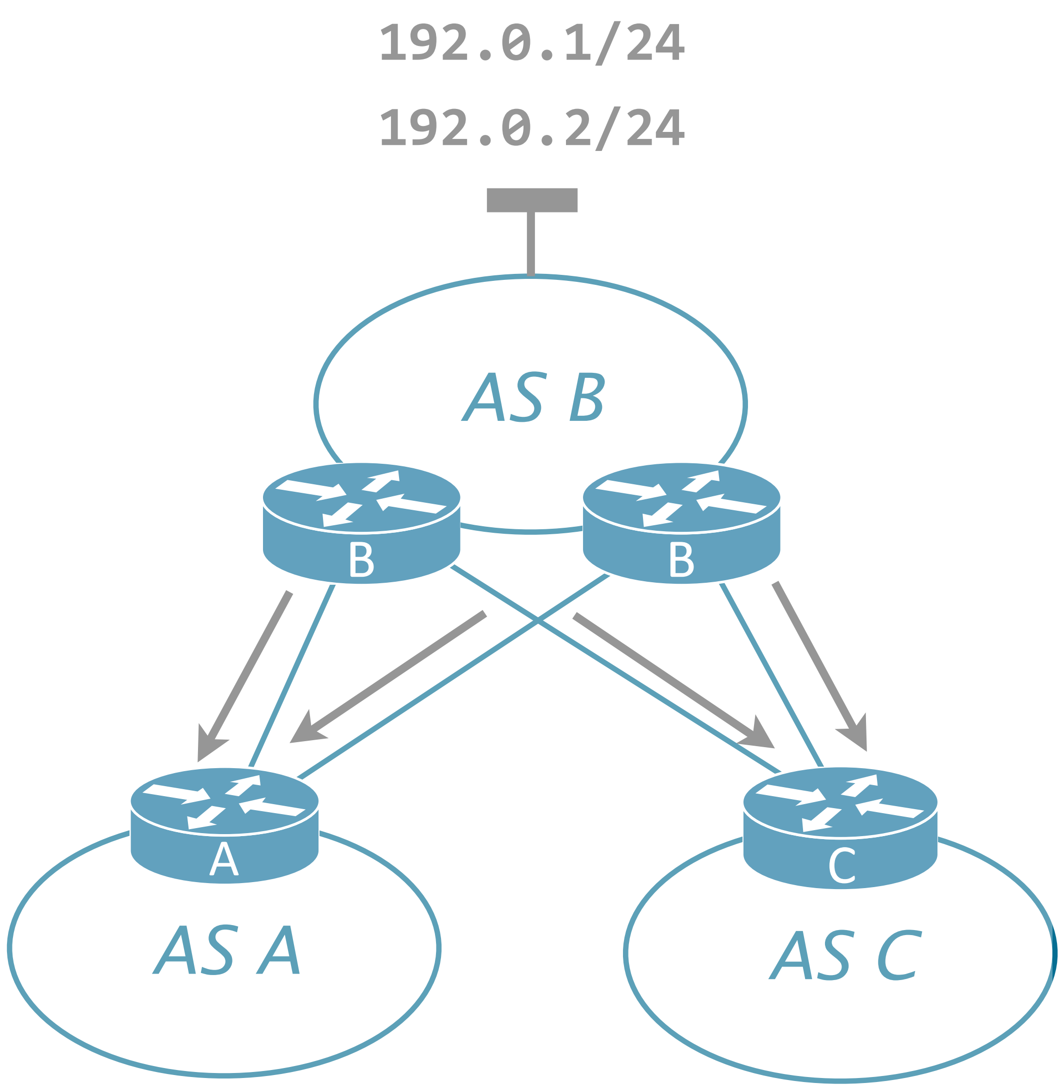




Given an IXP Physical Topology and a BGP topology, Implement B's inbound policies

B's inbound policies

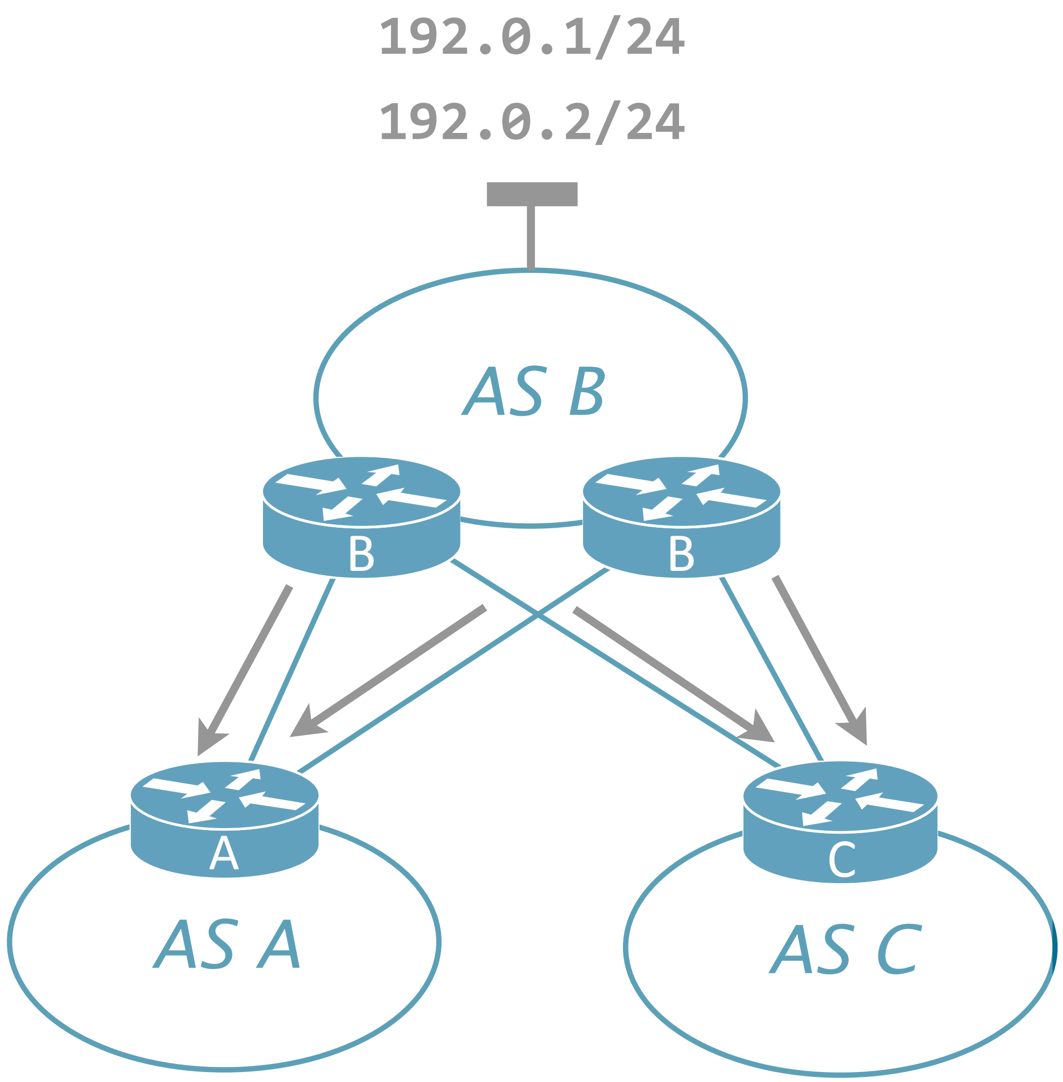
to	from	*receive on*
192.0.1/24	A	left
192.0.2/24	C	right
192.0.2/24	ATT_IP	right
192.0.1/24	*	right
192.0.2/24	*	left



How do you that with BGP?

B's inbound policies

to	from	*receive on*
192.0.1/24	A	left
192.0.2/24	C	right
192.0.2/24	ATT_IP	right
192.0.1/24	*	right
192.0.2/24	*	left



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.

... and 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

Implementing such a policy is configuration-intensive

using AS-Path prepend, MED, community tagging, etc.

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

to	from	fwd
192.0.1/24	A	left
192.0.2/24	B	right
192.0.2/24	ATT_IP	right
192.0.1/24	*	right
192.0.2/24	*	left

B's SDX Policy

```
match(dstip=192.0.1/24, srcmac=A), fwd(L)
match(dstip=192.0.2/24, srcmac=B), fwd(R)
match(dstip=192.0.2/24, srcip=ATT), fwd(R)
match(dstip=192.0.1/24), fwd(R)
match(dstip=192.0.2/24), fwd(L)
```

Novel Applications for a SDN-enabled Internet Exchange Point

SDX Architecture

data- and control-plane

App#1: Inbound TE

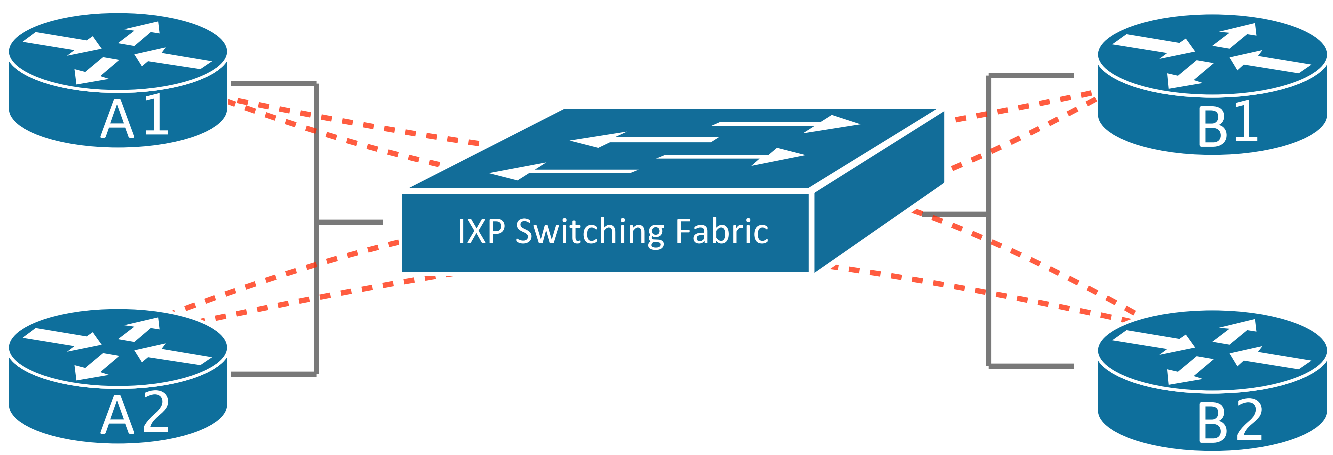
easy and deterministic

3 App#2: Fast convergence

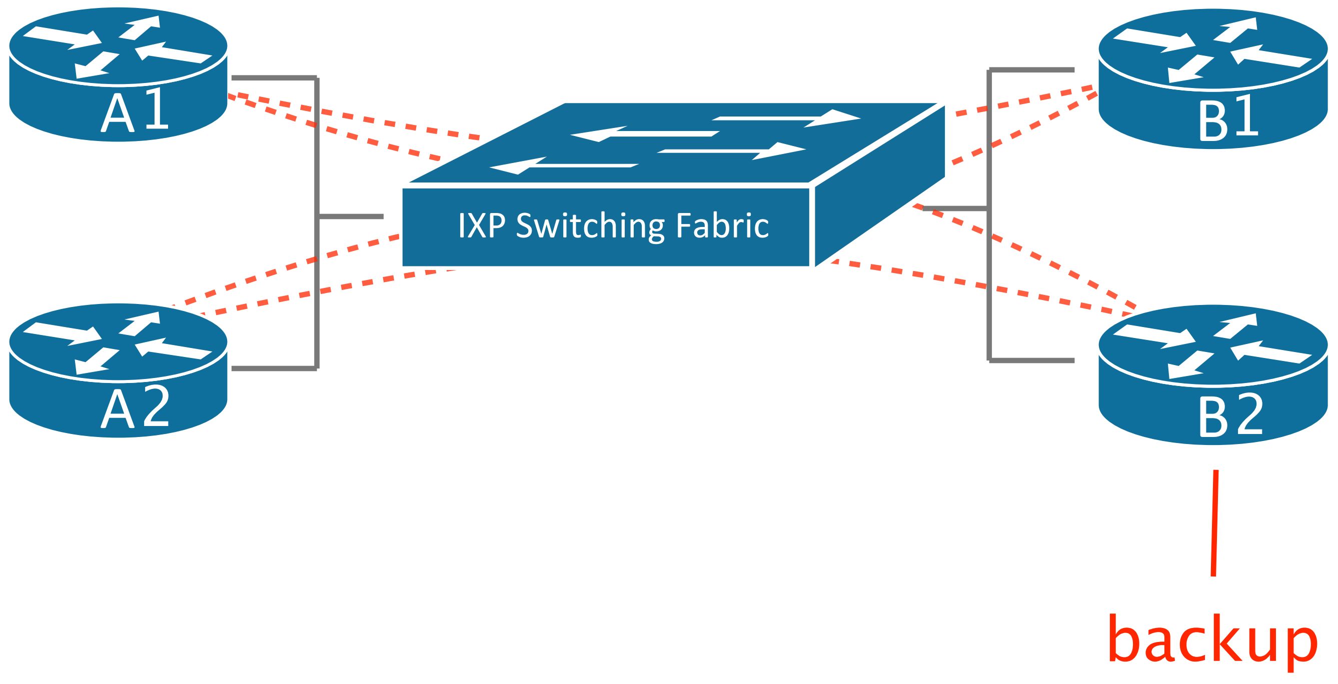
<1s after peering link failure

BGP is pretty slow to converge upon peering failure

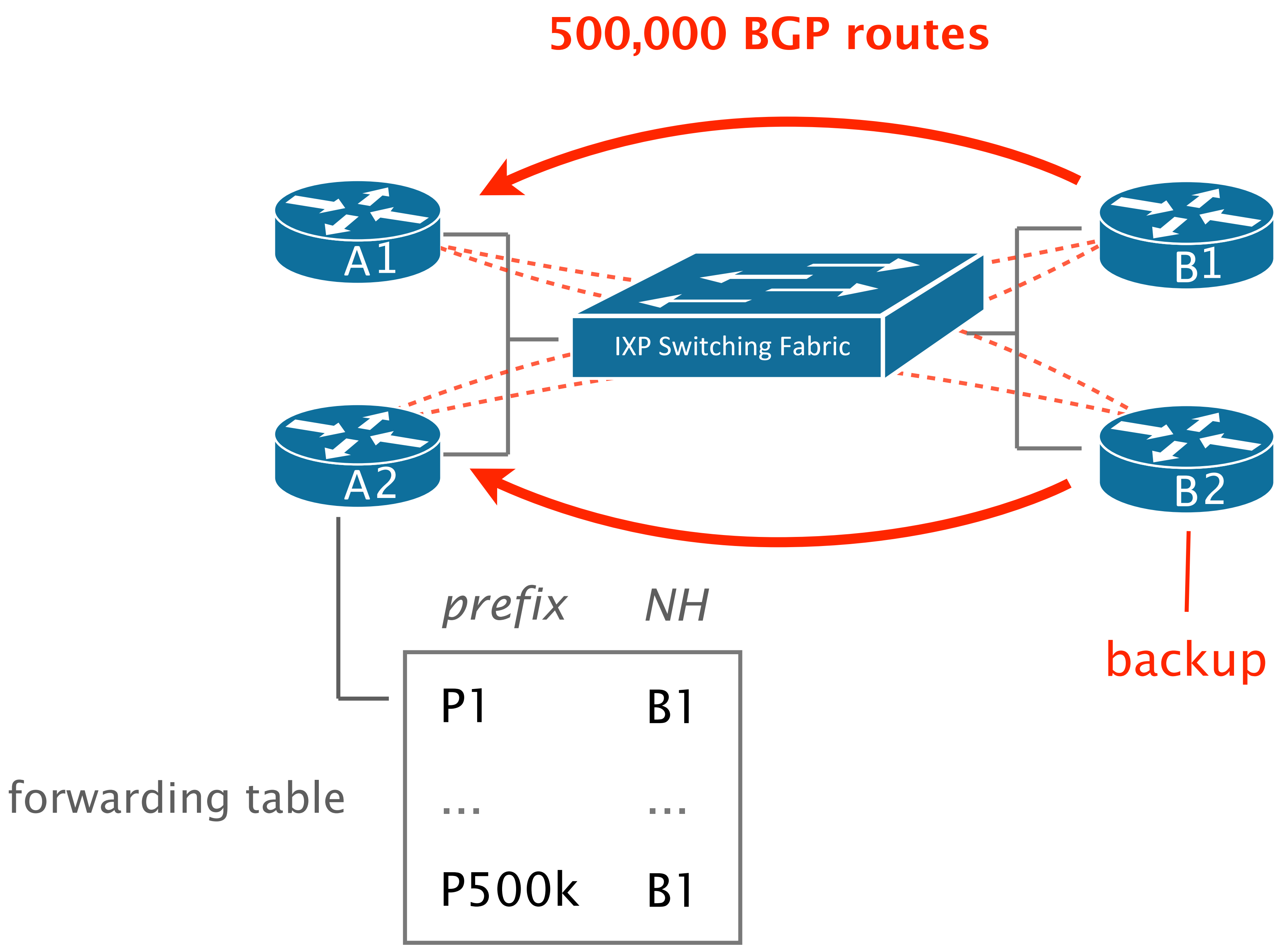
Let's consider a simple example with 2 networks, A and B, with B being the provider of A



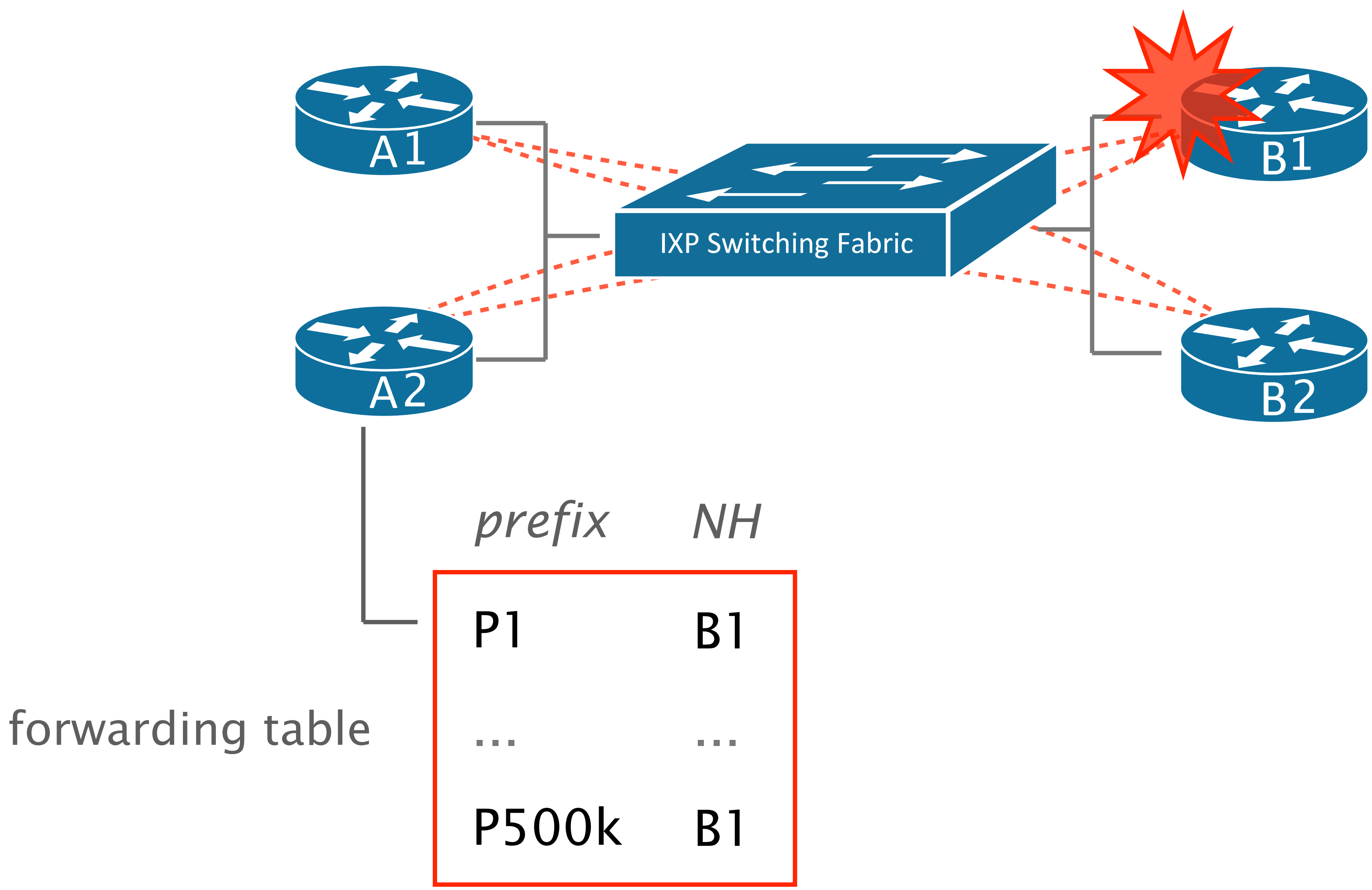
Router B2 is a backup router,
it can be used only upon B1's failure



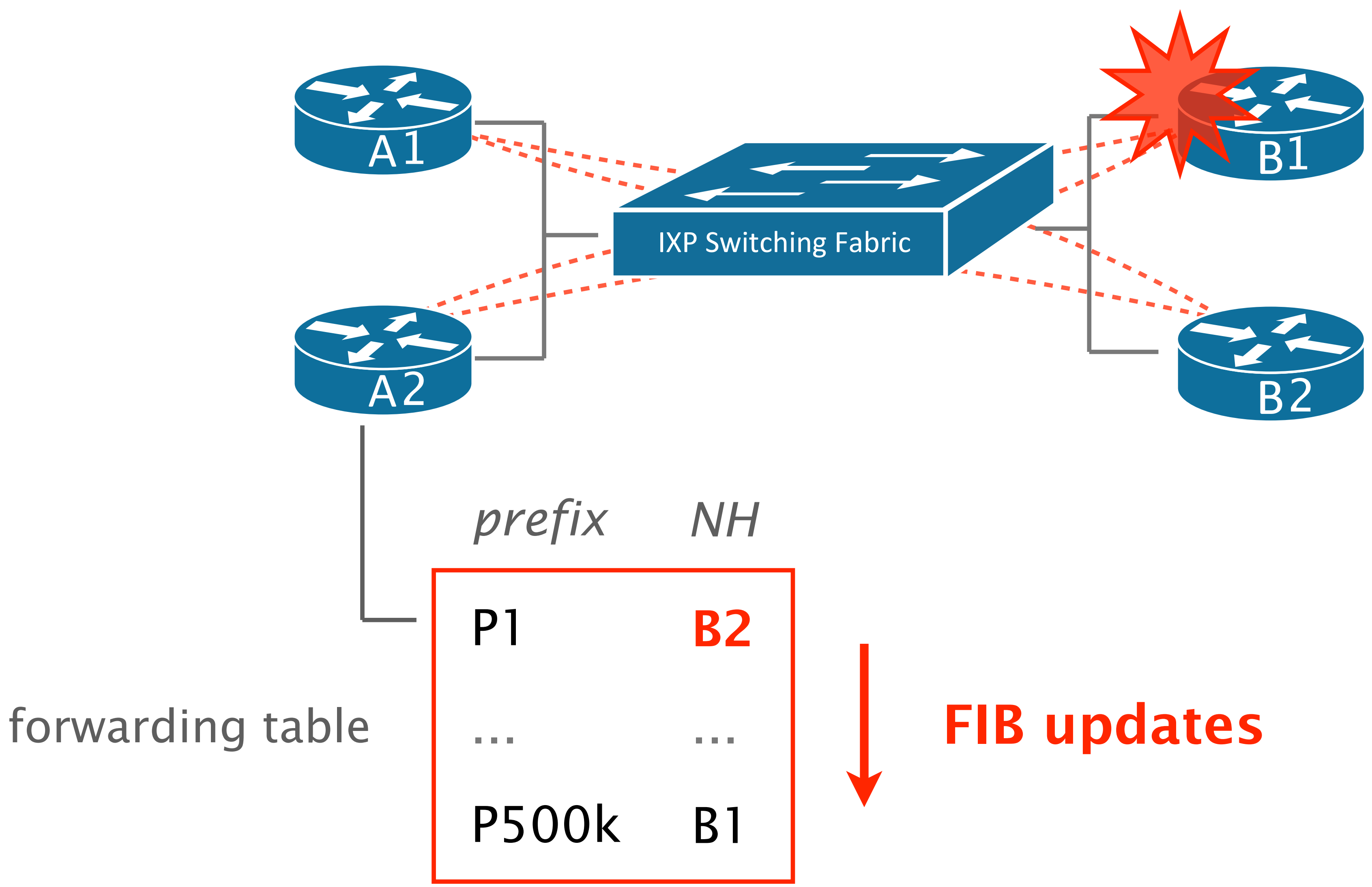
Both A1 and A2 prefer the routes received from B1 and install them in their FIB



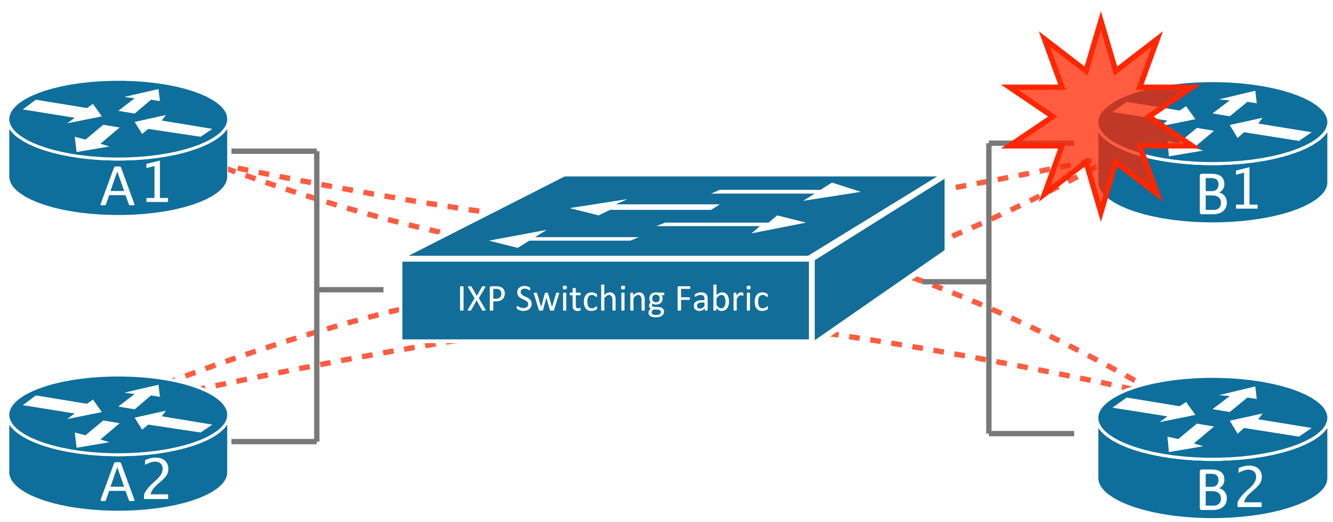
Upon B1's failure, A1 and A2 must update every single entry in their FIB (~500k entries)



Upon B1's failure, A1 and A2 must update every single entry in their FIB (~500k entries)



Upon B1's failure, A1 and A2 must update every single entry in their FIB (~500k entries)



forwarding table

prefix	*NH*
P1	**B2**
…	…
P500k	**B2**

FIB updates

On most routers, FIB updates are performed linearly, entry-by-entry, leading to *slow* BGP convergence

convergence time $\quad$ 500k entries * 150 $\frac{\text{usecs}}{\text{entry}}$

average time
to update one entry

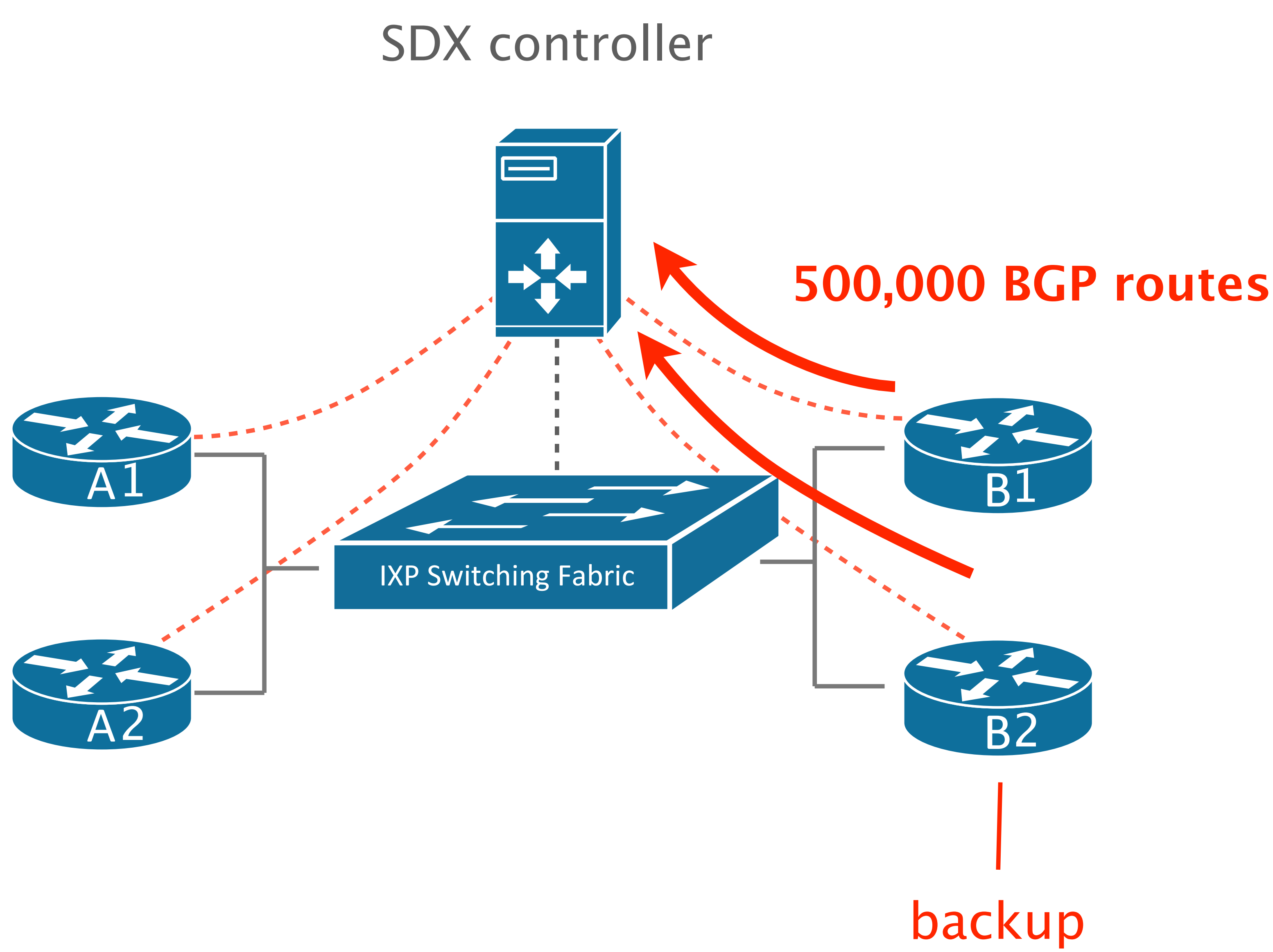
On most routers, FIB updates are performed linearly, entry-by-entry, leading to *slow* BGP convergence

convergence time $\quad$ 500k entries * 150 $\frac{\text{usecs}}{\text{entry}}$ = **O(75) seconds**

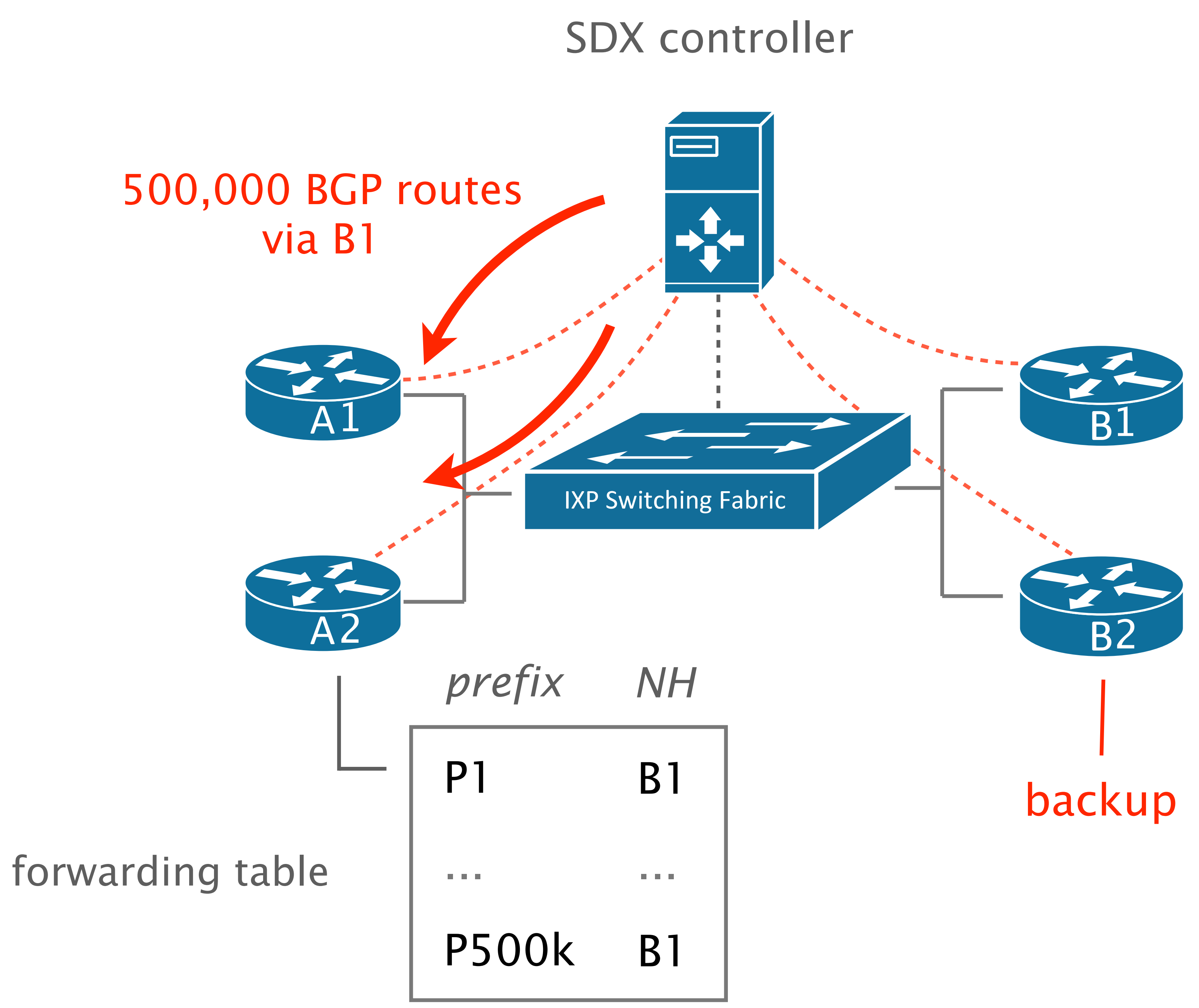
average time
to update one entry

With SDX, sub-second peering convergence can be achieved with any router

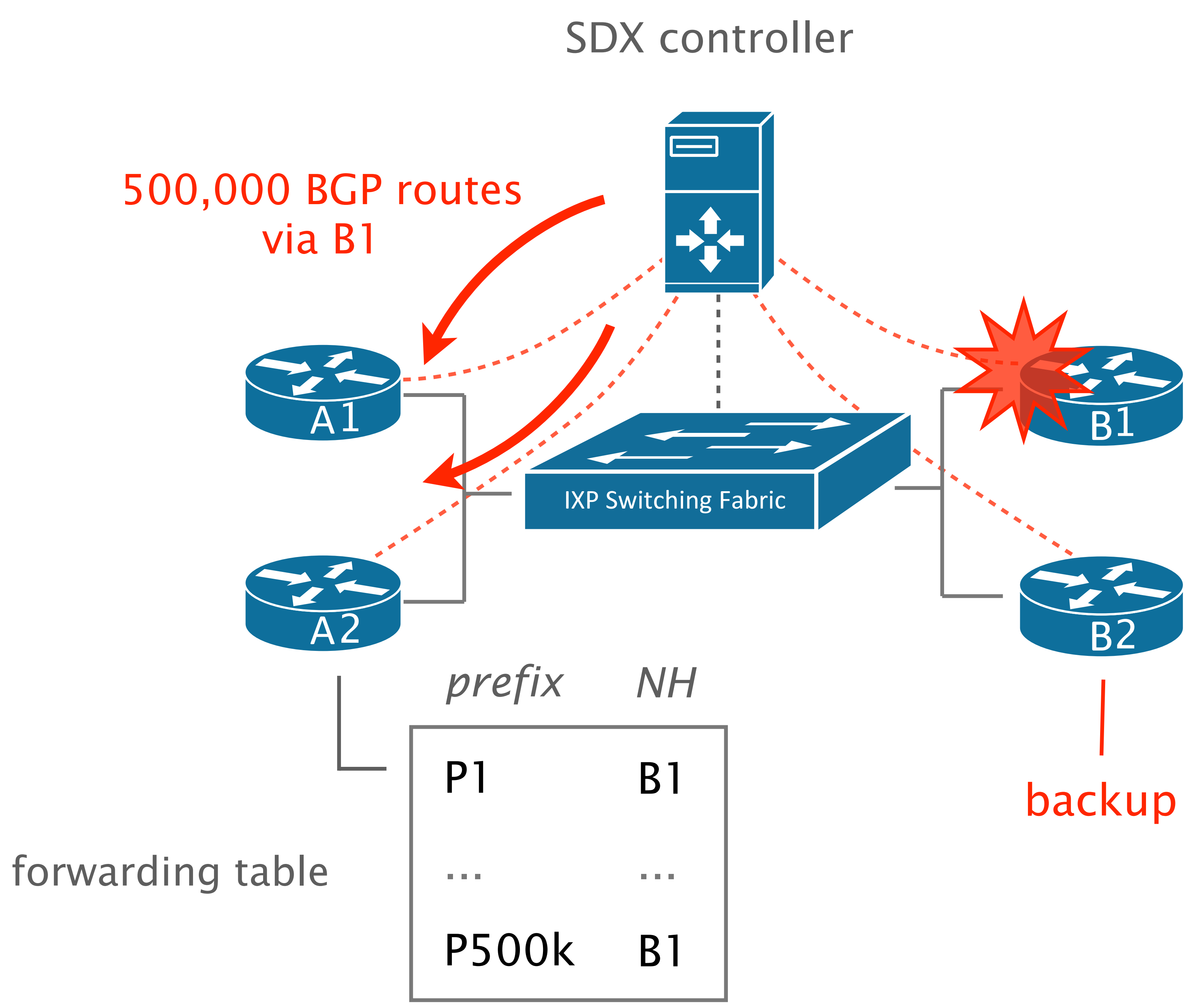
When receiving multiple routes, the SDX controller pre-computes a backup NH for each prefix



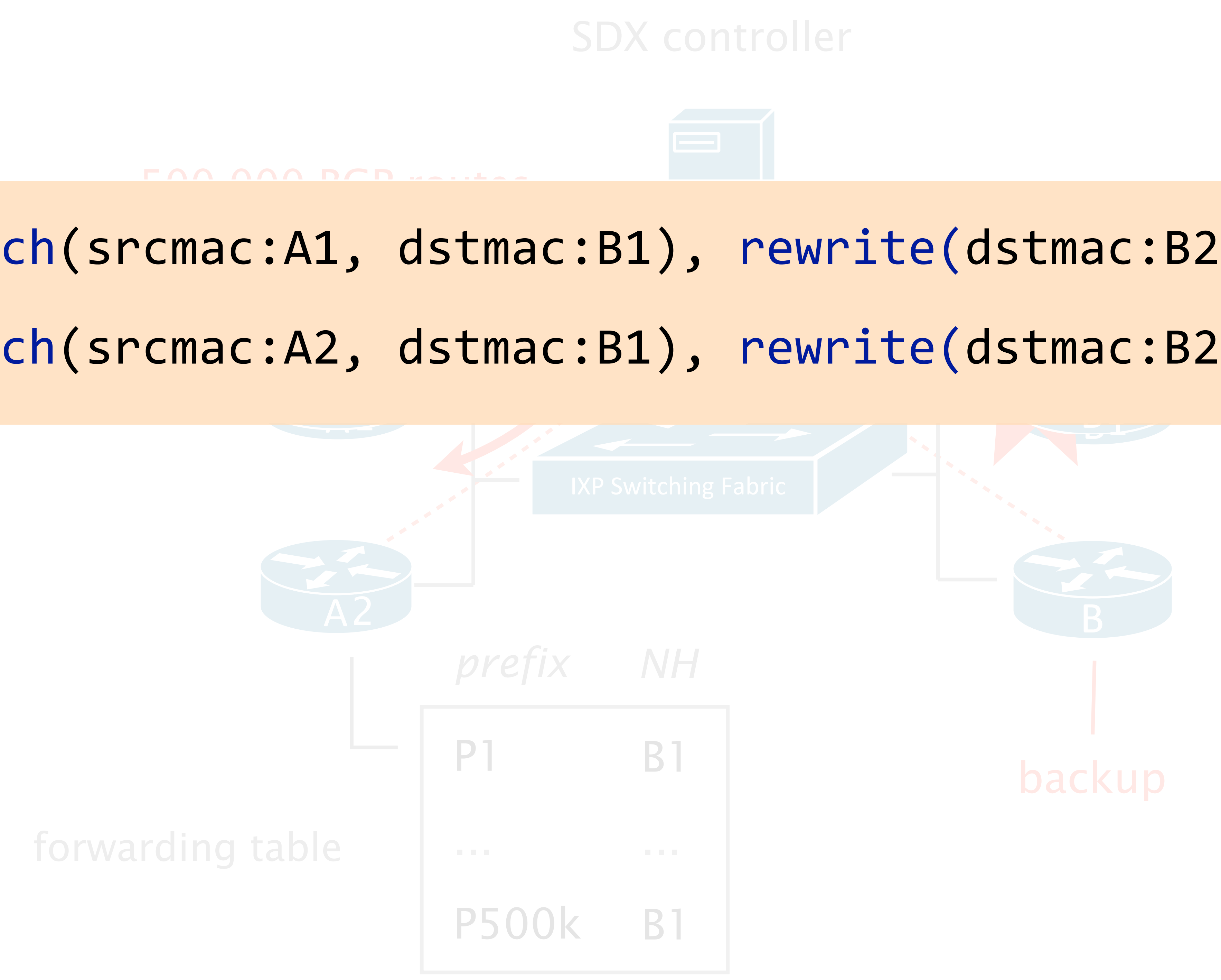
When receiving multiple routes, the SDX controller pre-computes a backup NH for each prefix



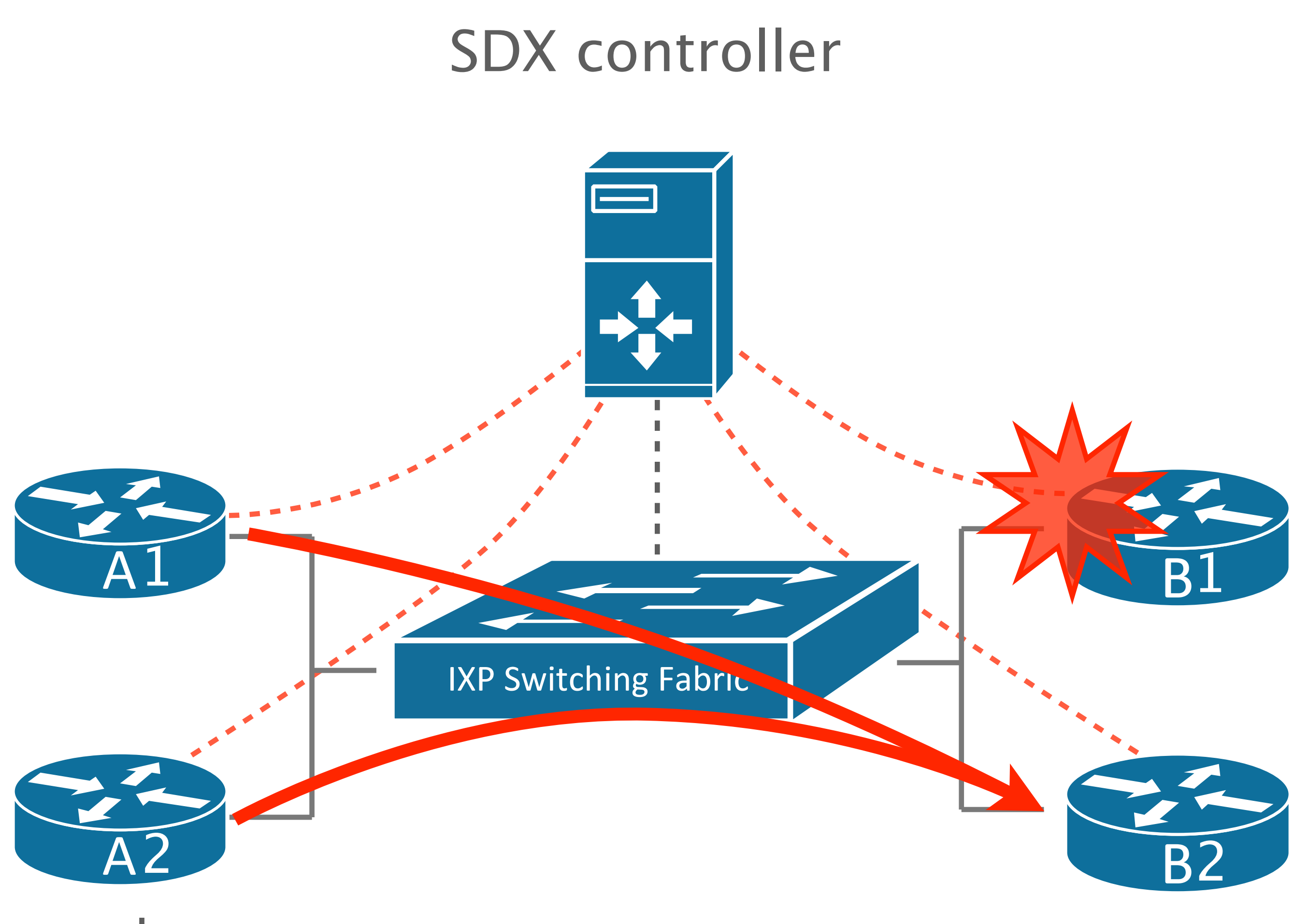
Upon a peer failure, the SDX controller directly pushes next-hop rewrite rules



```
match(srcmac:A1, dstmac:B1), rewrite(dstmac:B2), fwd(B2)
match(srcmac:A2, dstmac:B1), rewrite(dstmac:B2), fwd(B2)
```



All BGP traffic immediately moves from B1 to B2, independently of the number of FIB updates



P1	B1
...	...
P500k	B1

SDX data-plane can enable sub-second, prefix-independent BGP convergence

controller
communication time

convergence time # edge entries * 150 $\frac{\text{usecs}}{\text{entry}}$ + 30~50 ms

average update time per entry

SDX data-plane can enable sub-second, prefix-independent BGP convergence

$$\text{convergence time} \quad \#\,\text{edge entries} \; * \; 150 \; \frac{\text{usecs}}{\text{entry}} \; + \; 30\sim50 \text{ ms}$$

$$= \quad O(30\sim50) \text{ ms}$$

SDX data-plane can enable sub-second, prefix-independent BGP convergence

Most peering links can be protected

since most participants have at least two interfaces

It does not interfere with participant policies

totally transparent to the routing system

It does not require any hardware changes

works on any router, even older ones

Novel Applications for a SDN-enabled Internet Exchange Point

SDX Architecture

data- and control-plane

App#1: Inbound TE

easy and deterministic

App#2: Fast convergence

<1s after peering link failure

SDN can also solve some of the challenges faced by IXP operators

Capture broadcast traffic & unwanted traffic

deal with it at the controller level (*e.g.*, ARP, STP BPDUs)

Enable fine-grained Traffic Engineering, Load-balancing

think traffic steering, monitoring, etc.

Simplify infrastructure management

get rid of STP, perform isolation without VLANs, etc.

We have running code as well as a first deployment site

We have a first SDX controller prototype

which supports policies composition and isolation

We have partnered with a large regional IXP in Atlanta

which hosts many large content providers such as Akamai

We are open for peering request

ping me if you are interested

Participate in our **survey**

Novel Applications for a SDN-enabled Internet Exchange Point

Laurent Vanbever

http://vanbever.eu

RIPE 67, Athens

October, 14 2013