Software Defined Networking

Dr. Nick Feamster
Professor

In this course, you will learn about software defined networking and how it is changing the way communications networks are managed, maintained, and secured.

Software Defined Networking

- Changing how we manage networks
  - Data centers, backbones, enterprises, …
- But, so far, mostly *inside* these networks
  - Network virtualization, traffic engineering, …
- In this lecture:
  - Fundamentally change *interdomain* traffic delivery
  - Starting with SDN at *boundaries* between domains

BGP is Not Flexible Enough

- Routing only on **destination IP address blocks**
- Can only influence **immediate neighbors**
  (No ability to affect path selection remotely)
- **Indirect** control over packet forwarding
  (Indirect mechanisms to influence path selection)
- Enables only basic packet **forwarding**
  (Difficult to introduce new in-network services)

Valuable Wide-Area Services

- Application-specific peering
  - Route video traffic one way, and non-video another
- Blocking denial-of-service traffic
  - Dropping unwanted traffic further upstream
- Server load balancing
  - Directing client requests to different data centers
- Steering through network functions
  - Transcoders, scrubbers, caches, crypto, …
- Inbound traffic engineering
  - Splitting incoming traffic over multiple peering links

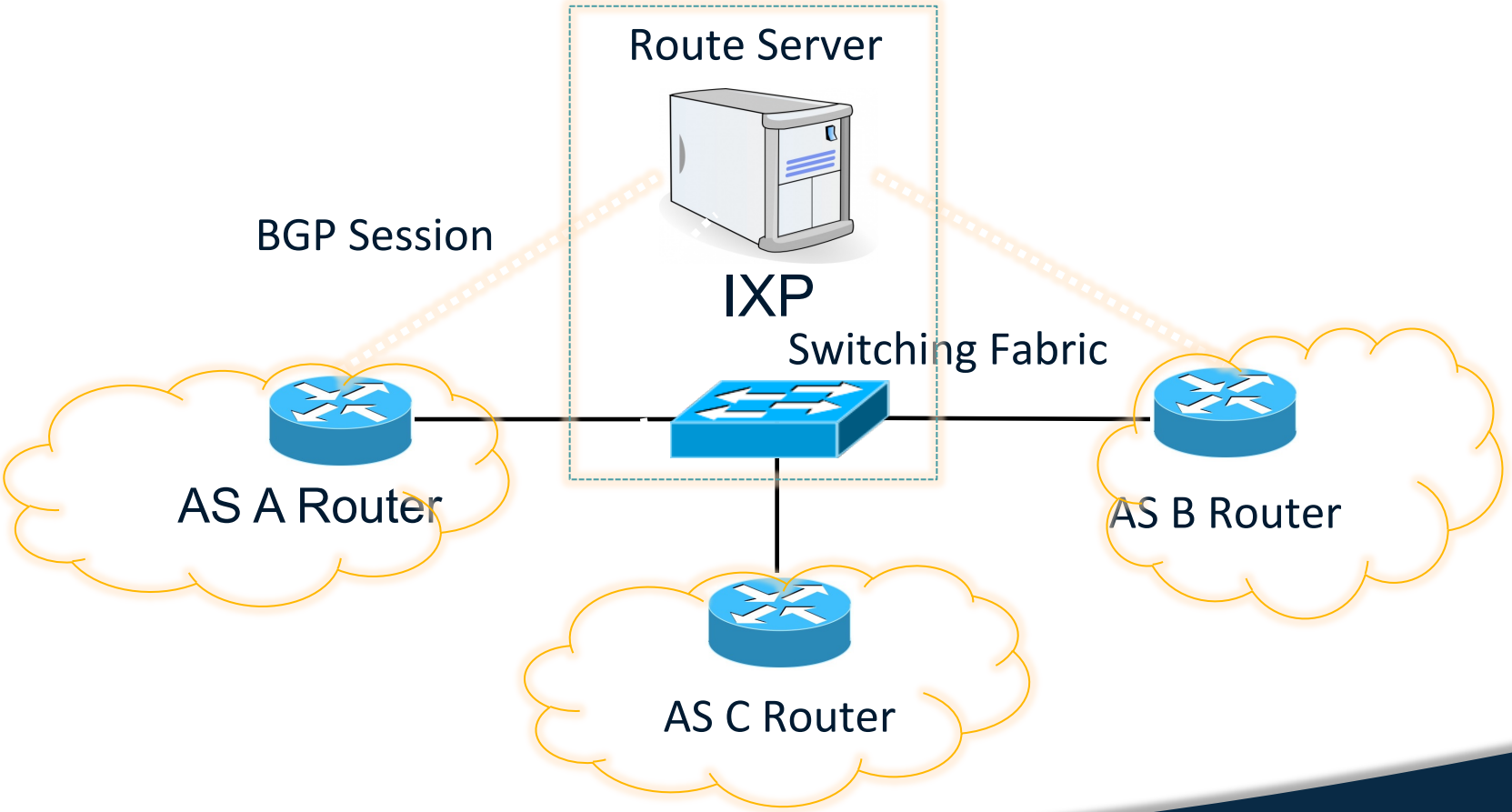
Software-Defined Networking

- Match packets on **multiple header fields**
  (not just destination IP address)
- Control **entire networks** with a single program
  (not just immediate neighbors)
- **Direct control** over packet handling
  (not indirect control via routing protocol arcana)
- Perform many different **actions** on packets
  (beyond basic packet forwarding)

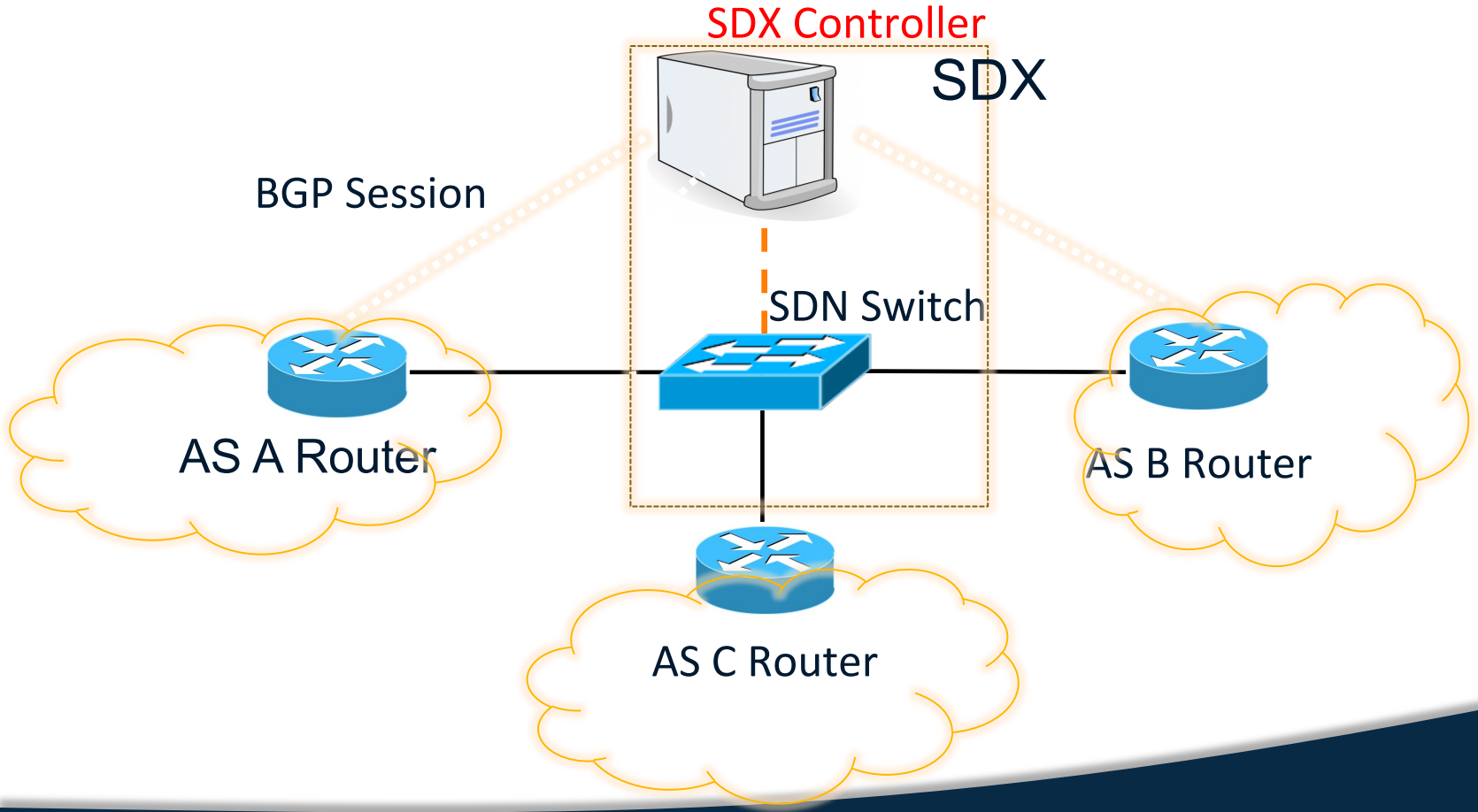
Deploy SDN at Internet Exchanges

- **Leverage:** SDN deployment even at single IXP can benefit tens to hundreds of providers
  - *Without providers deploying new equipment!*
- **Innovation hotbed:** Incentives to innovate, as IXPs on front line of peering disputes
- **Growing in numbers:**
  - 350-400 IXPs
  - ~100 new IXPs established in past few years

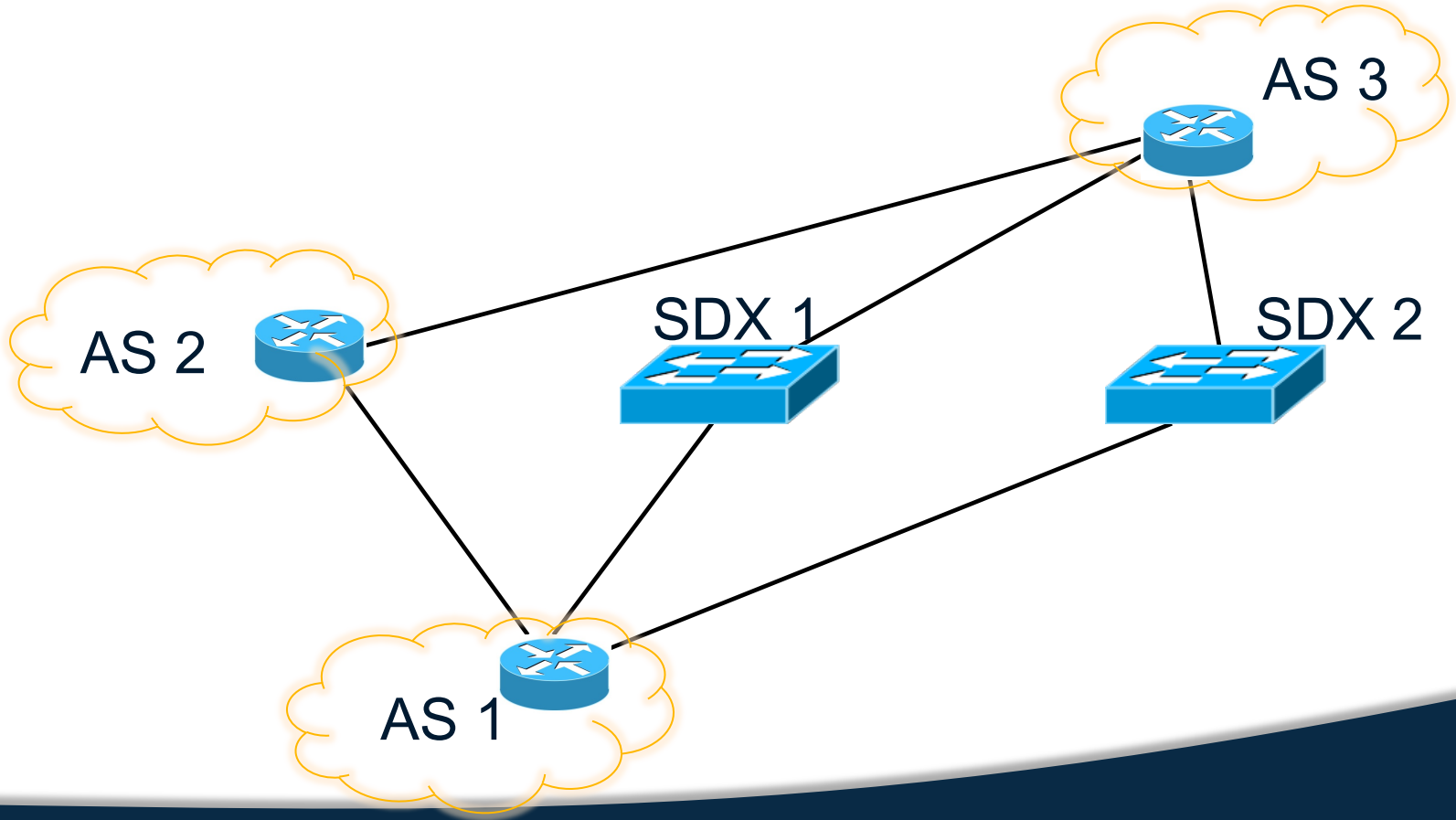
Conventional IXPs



SDX = SDN + IXP

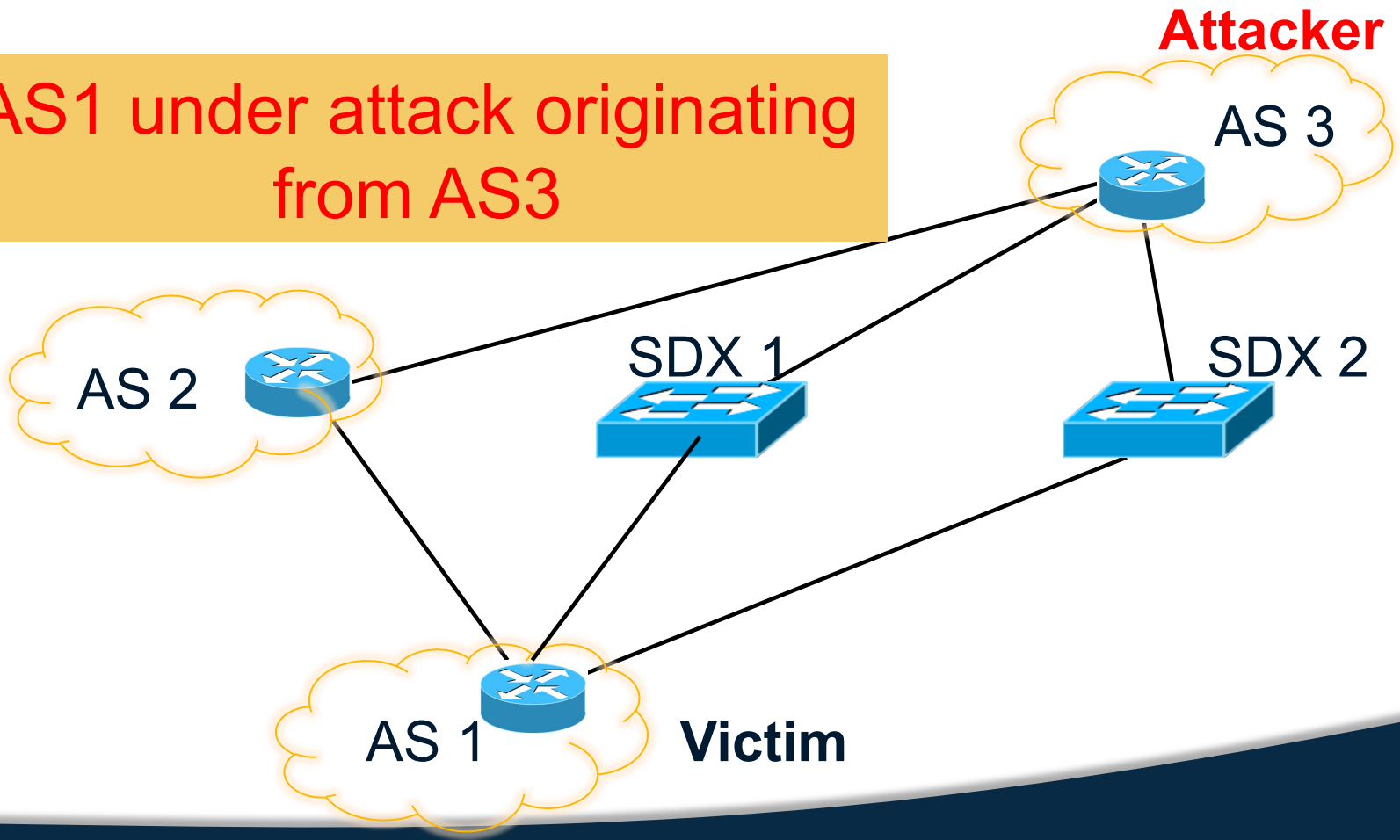

SDX Controller
SDX
BGP Session
SDN Switch
AS A Router
AS B Router
AS C Router

Prevent DDoS Attacks



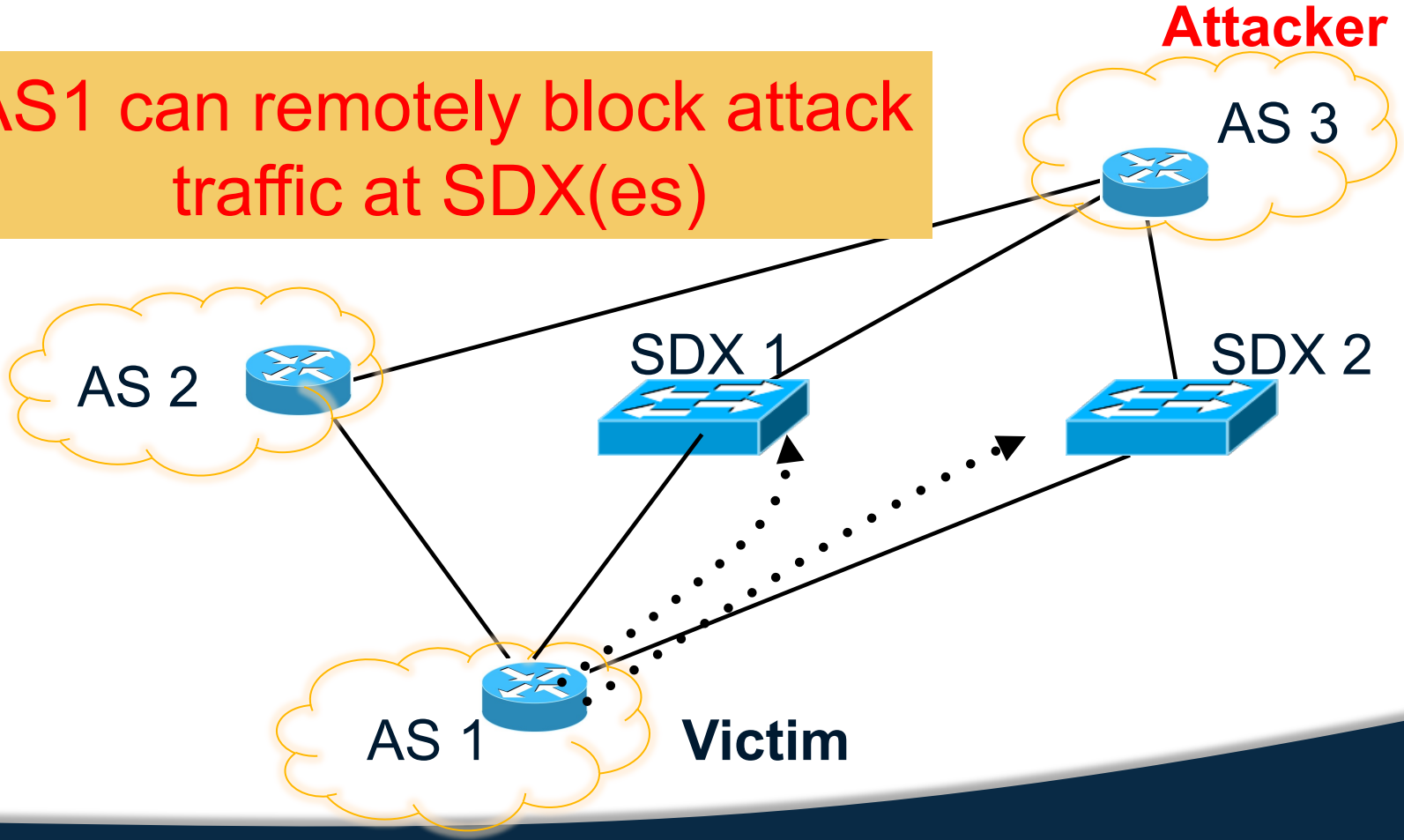
Prevent DDoS Attacks

AS1 under attack originating from AS3



Use Case: Prevent DDoS Attacks

AS1 can remotely block attack traffic at SDX(es)



SDX-based DDoS protection vs. Conventional Defenses/Blackholing

- **Remote influence**

  Physical connectivity to SDX not required

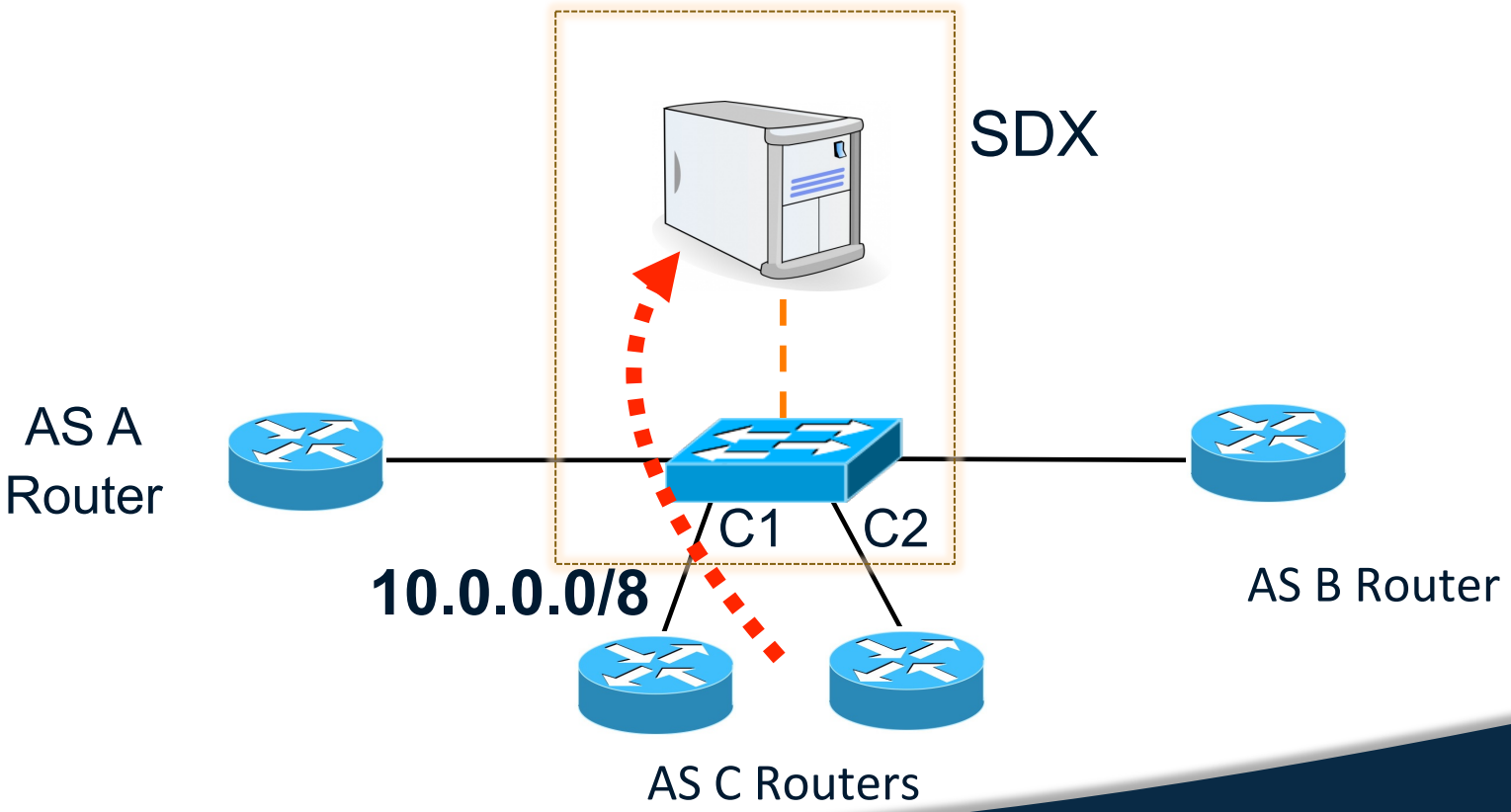
- **More specific**

  Drop rules based on multiple header fields, source address, destination address, port number …

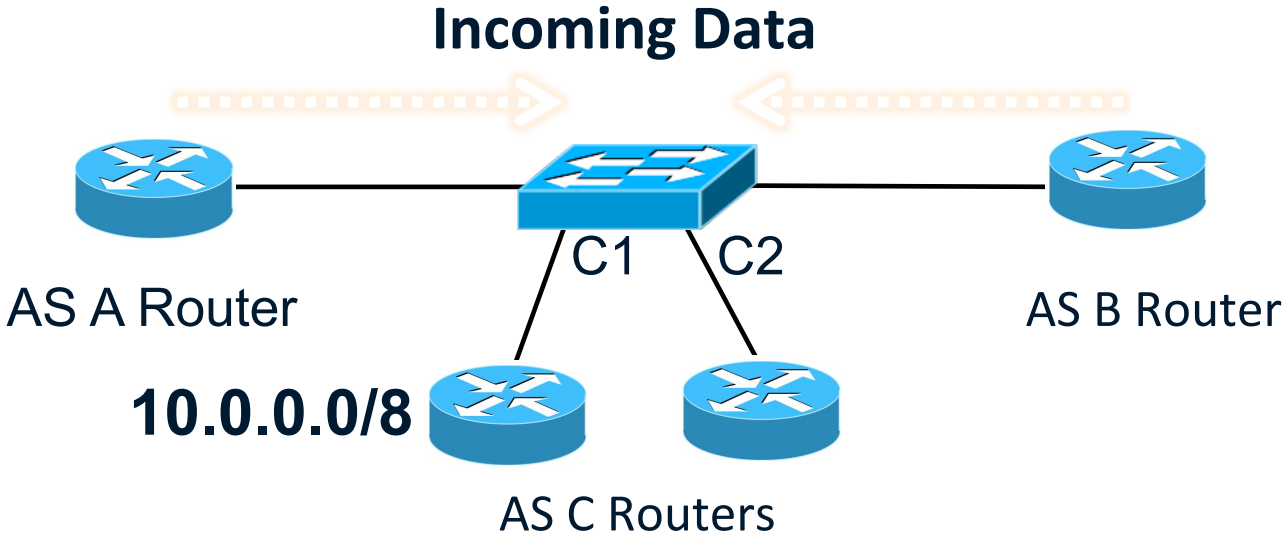
- **Coordinated**

  Drop rules can be coordinated across multiple IXPs

Inbound Traffic Engineering

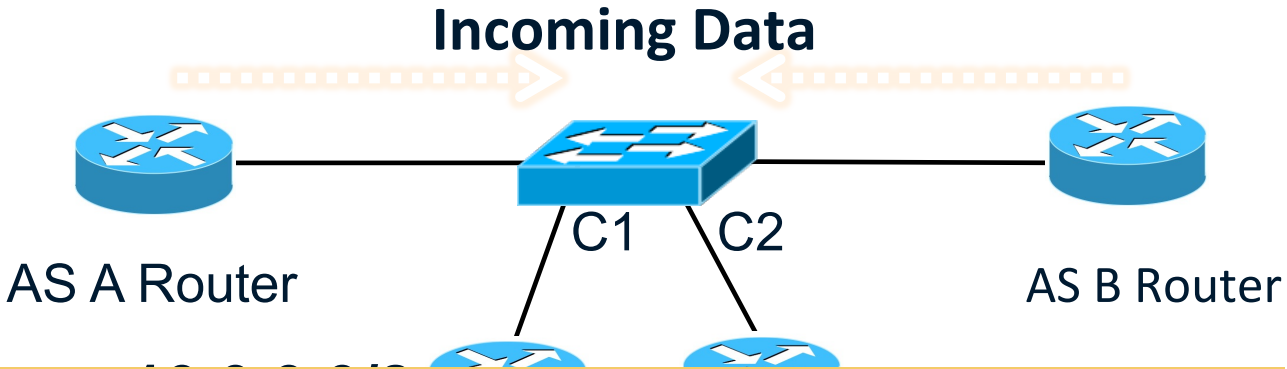


Inbound Traffic Engineering



Incoming Traffic	Out Port	Using BGP	Using SDX
dstport = 80	C1		

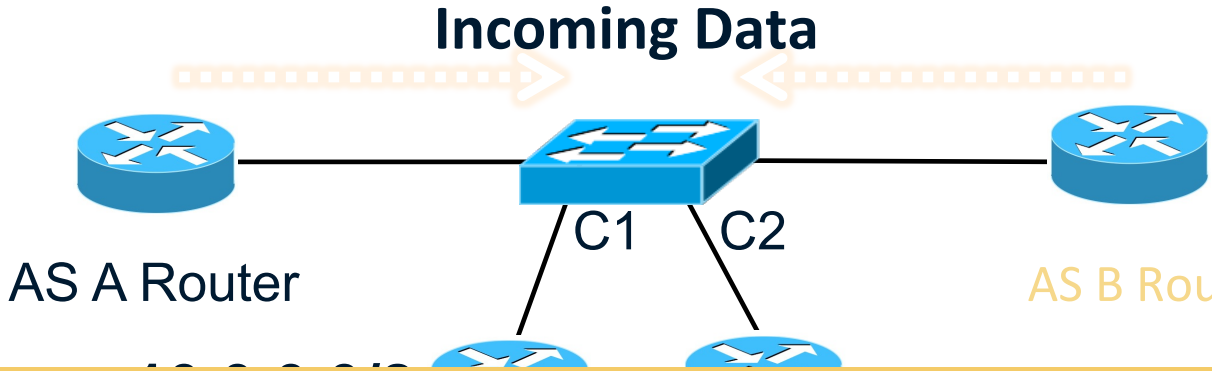
Inbound Traffic Engineering



Fine grained policies not possible with BGP

Incoming Traffic	Out Port	Using BGP	Using SDX
dstport = 80	C1	?	

Inbound Traffic Engineering



Enables fine-grained traffic engineering policies

Incoming Traffic	Out Port	Using BGP	Using SDX
dstport = 80	C1	?	match(dstport =80)→ fwd(C1)

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 address blocks, and matches on multiple header fields?

Building SDX is Challenging

- Programming **abstractions**

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

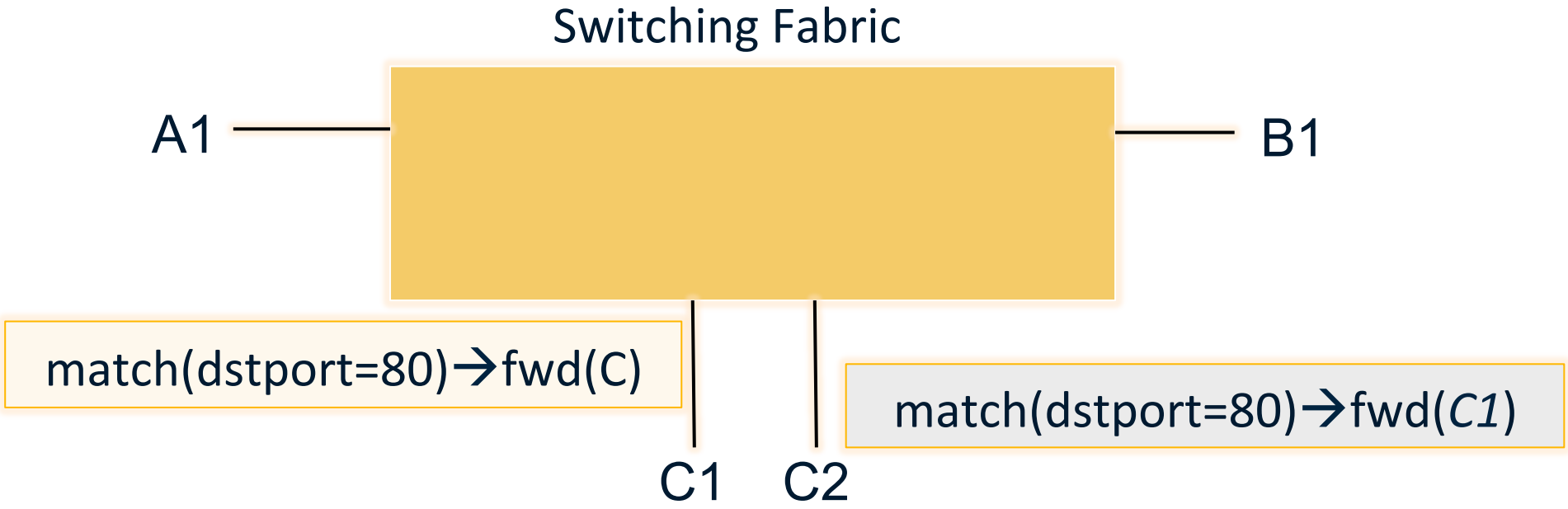
- **Interoperation** with BGP

  How to provide flexibility w/o breaking global routing?

- **Scalability**

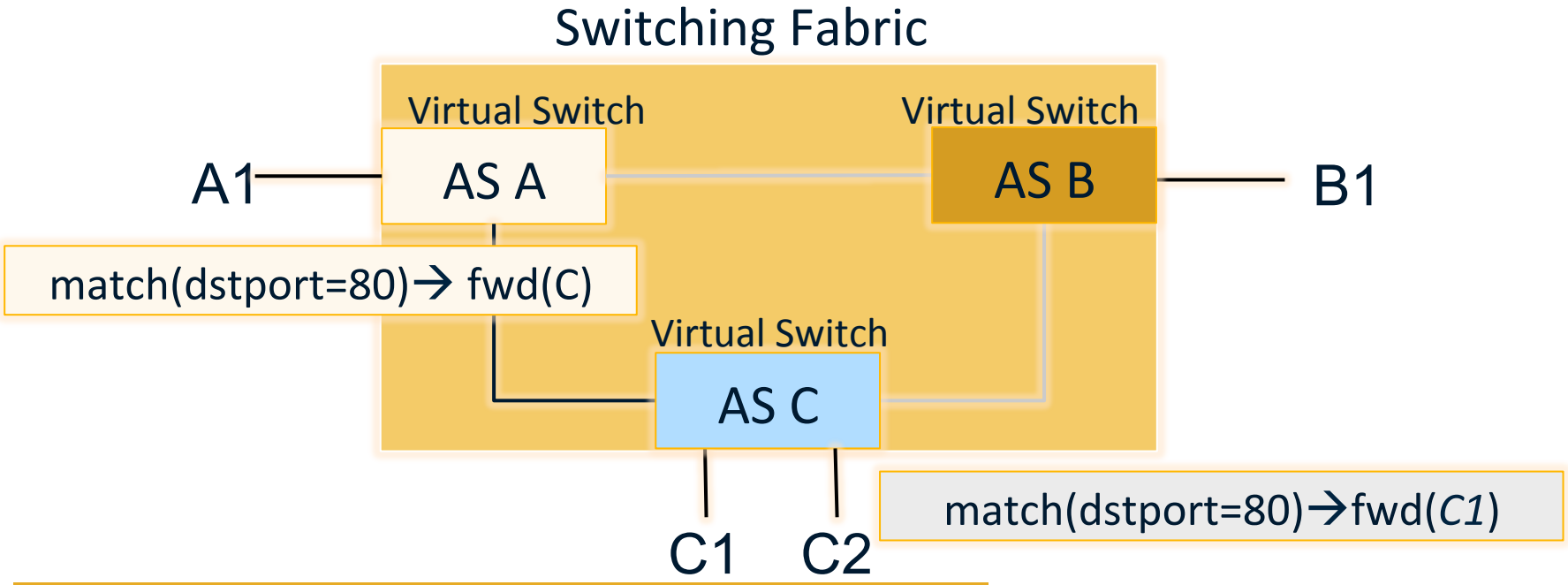
  How to handle policies for hundreds of peers,
  half million prefixes and matches
  on multiple header fields?

Directly Program the SDX Switch



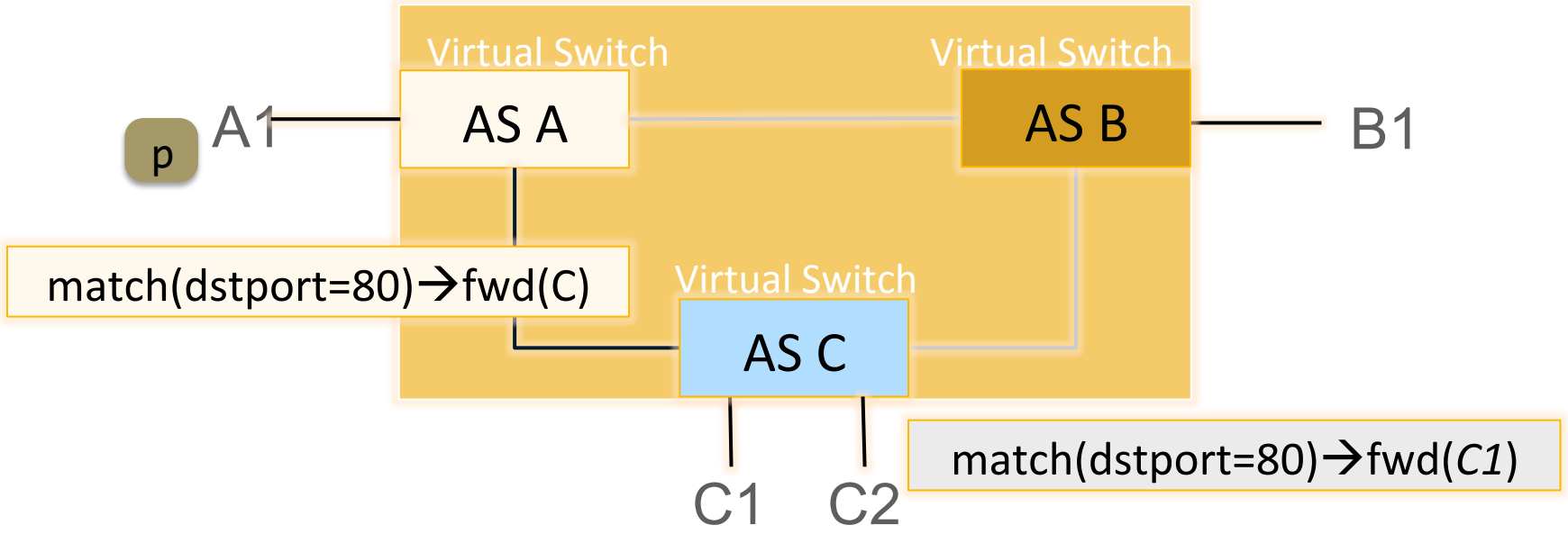
AS A & C directly program the SDX Switch

Virtual Switch Abstraction



Each AS writes policies for its own virtual switch

Combining Participant's Policies



Synthesize: match(inport=A1 & dstport=80) → fwd(C1)

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 address blocks, and matches on multiple header fields?

SDX Platform

- Running code with full BGP integration
  - Github available from http://sdx.cs.princeton.edu
- SDX testbeds:
  - Transit Portal for “in the wild” experiments
  - Mininet for controller experiments
- Ongoing deployment opportunities
  - Princeton, DOD/IC, GENI, SOX, Internet2, ESnet
  - Regional IXPs in US, Europe, and Africa

Conclusion

- The Internet is changing
  - New challenges for content delivery
  - Increasing importance of IXPs
- SDN can let providers innovate
  - New capabilities and abstractions
- Next steps
  - Operational deployments
  - Additional SDX applications
  - Distributed exchange points