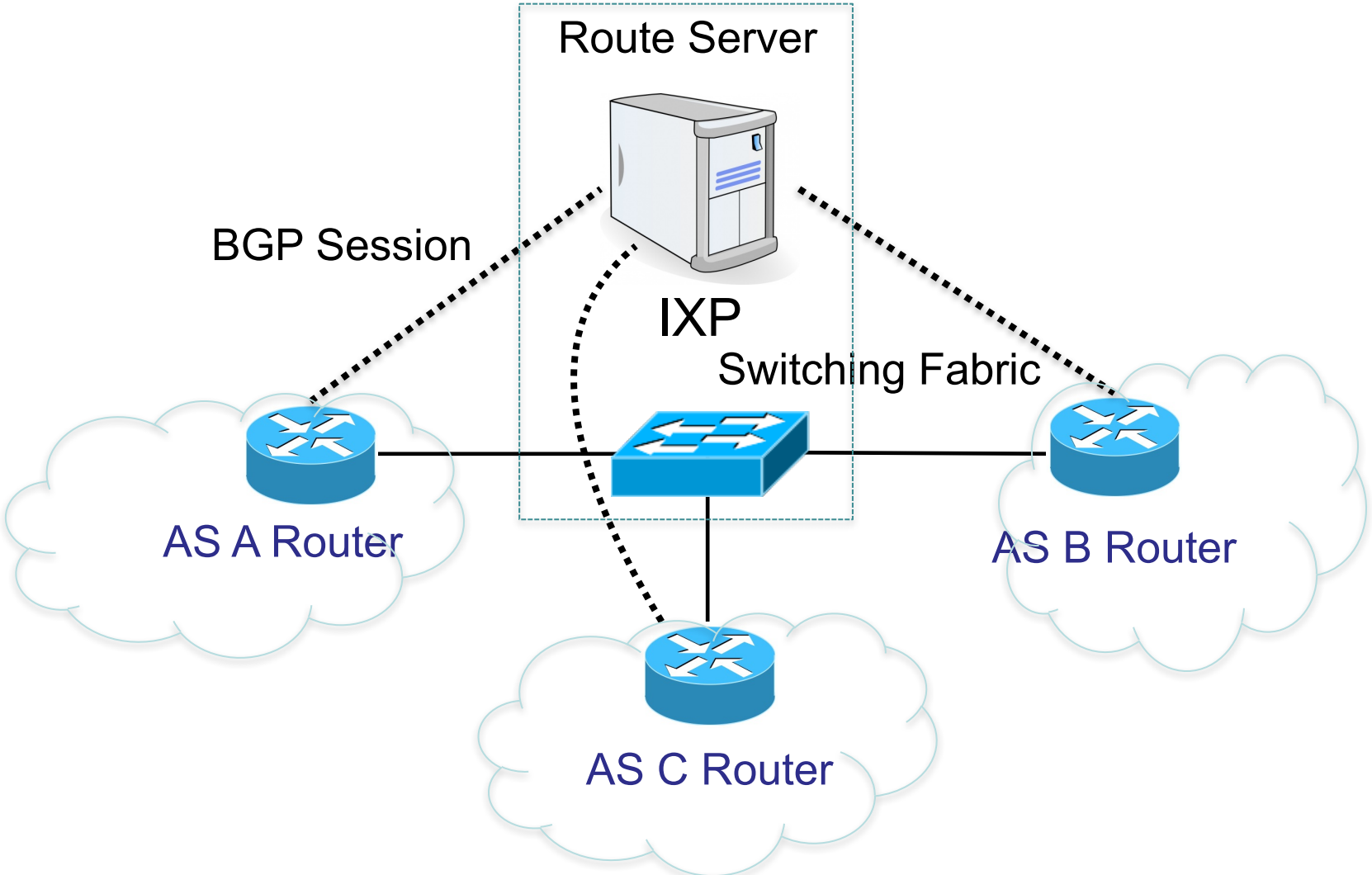
Authorizing Network Control at Software Defined Exchange Points

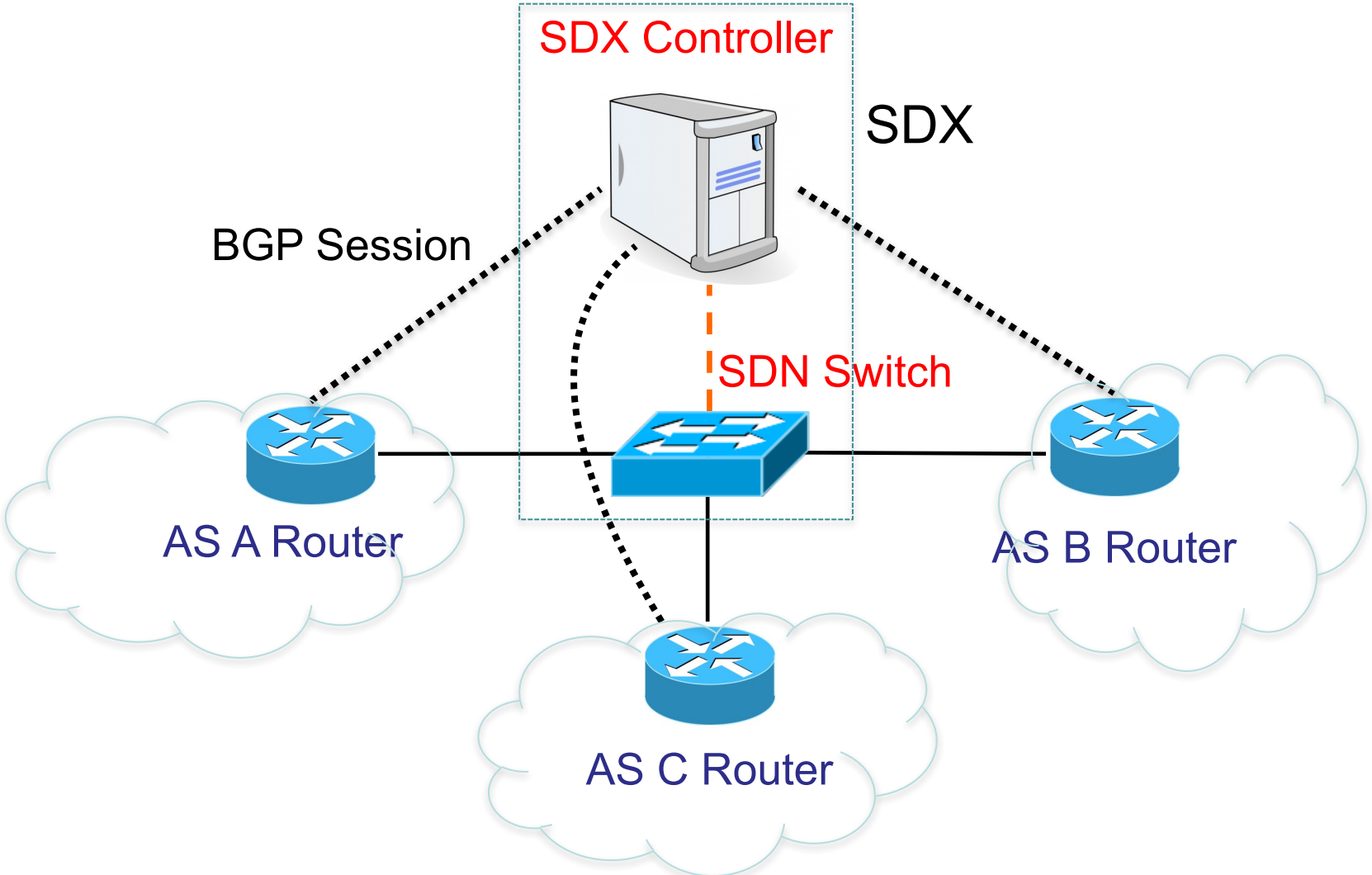
Arpit Gupta
Princeton University
http://sdx.cs.princeton.edu

Nick Feamster, Laurent Vanbever

Internet Exchange Points (IXPs)



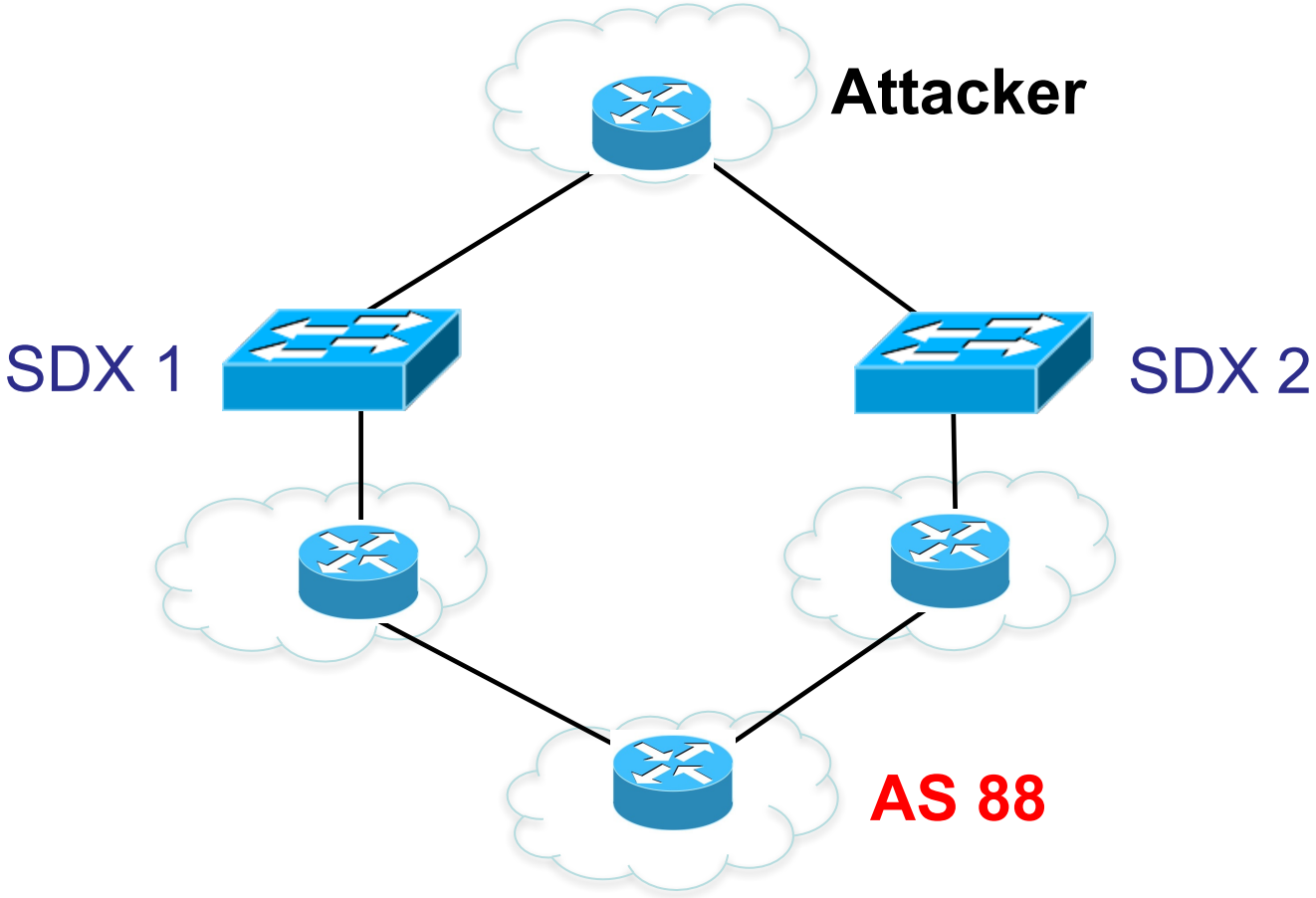
Software Defined IXPs (SDXs)



SDX Opens Up New Possibilities

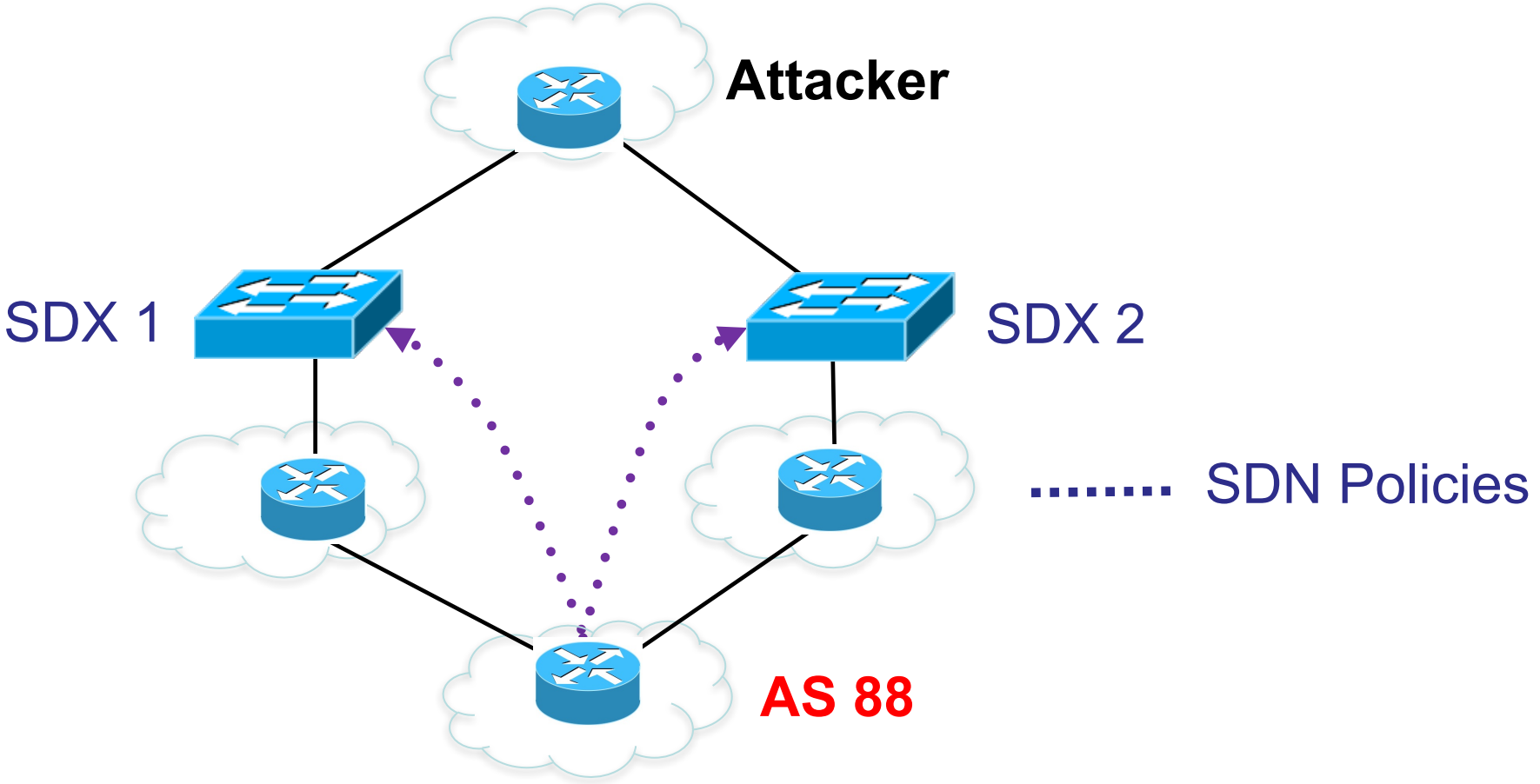
- More flexible **business relationships**
  - Make peering decisions based on time of day, volume of traffic & nature of application
- More direct & flexible **traffic control**
  - Define fine-grained traffic engineering policies
- Better **security**
  - Block or redirect attack traffic at finer level of granularity

SDX for DDoS Attack Mitigation



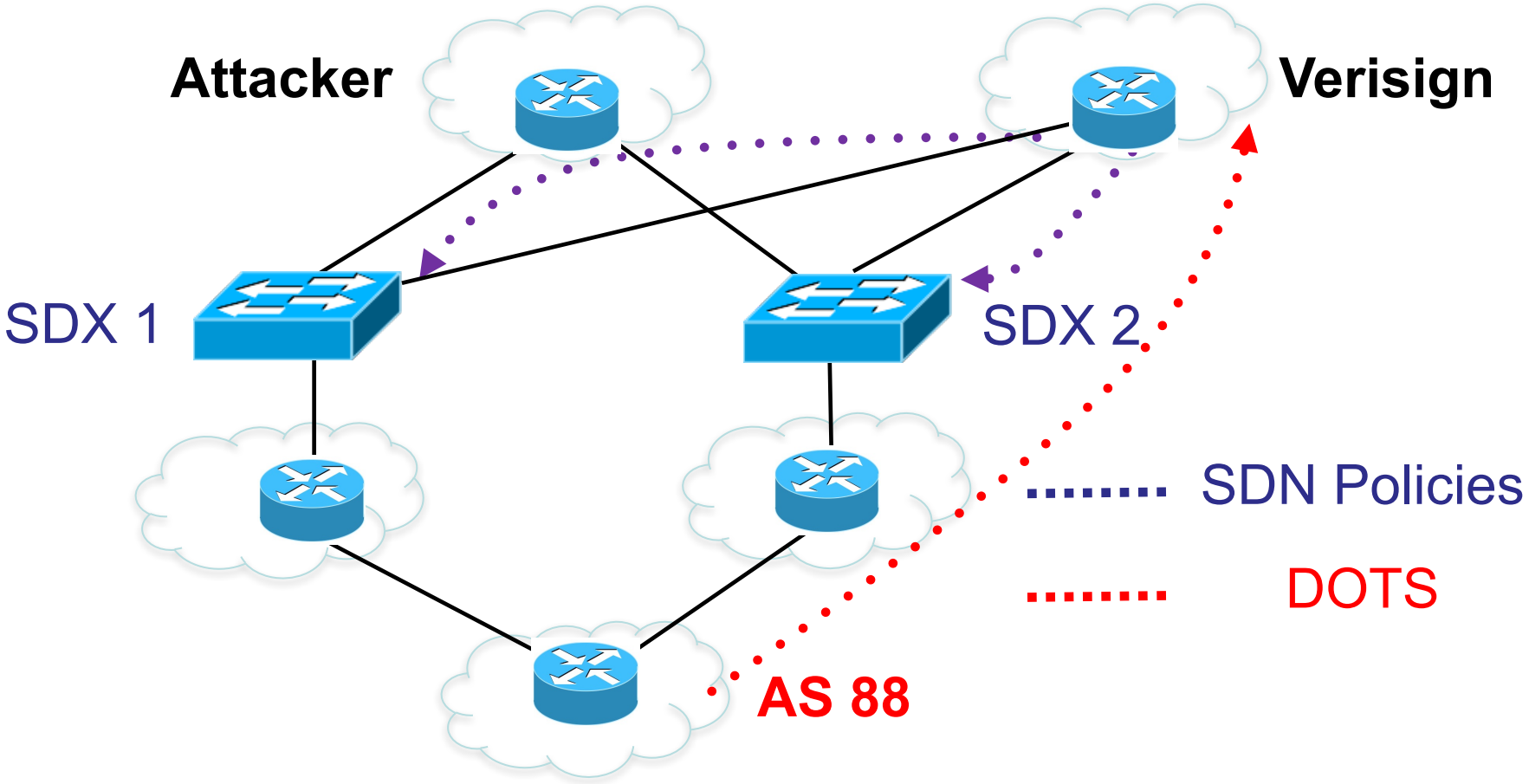
Attack traffic traverses two different SDXs

Remotely Block Attack Traffic



Victim remotely pushes block rules to SDX

Subscribe to Third Party Services



Victim Subscribes to Verisign for DDoS Protection

SDX vs. Traditional DDoS Defense

- **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

Spider-Man Dilemma

With Great Power Comes Great Responsibility

- **Authorize Remote Requests**
  - Is AS 88 owner of flow space under attack?
- **Authorize Third Party Requests**
  - Is Verisign authorized by AS 88 to block or redirect attack traffic?
  - Is AS 88 owner of flow space under attack?

Authorization Logic

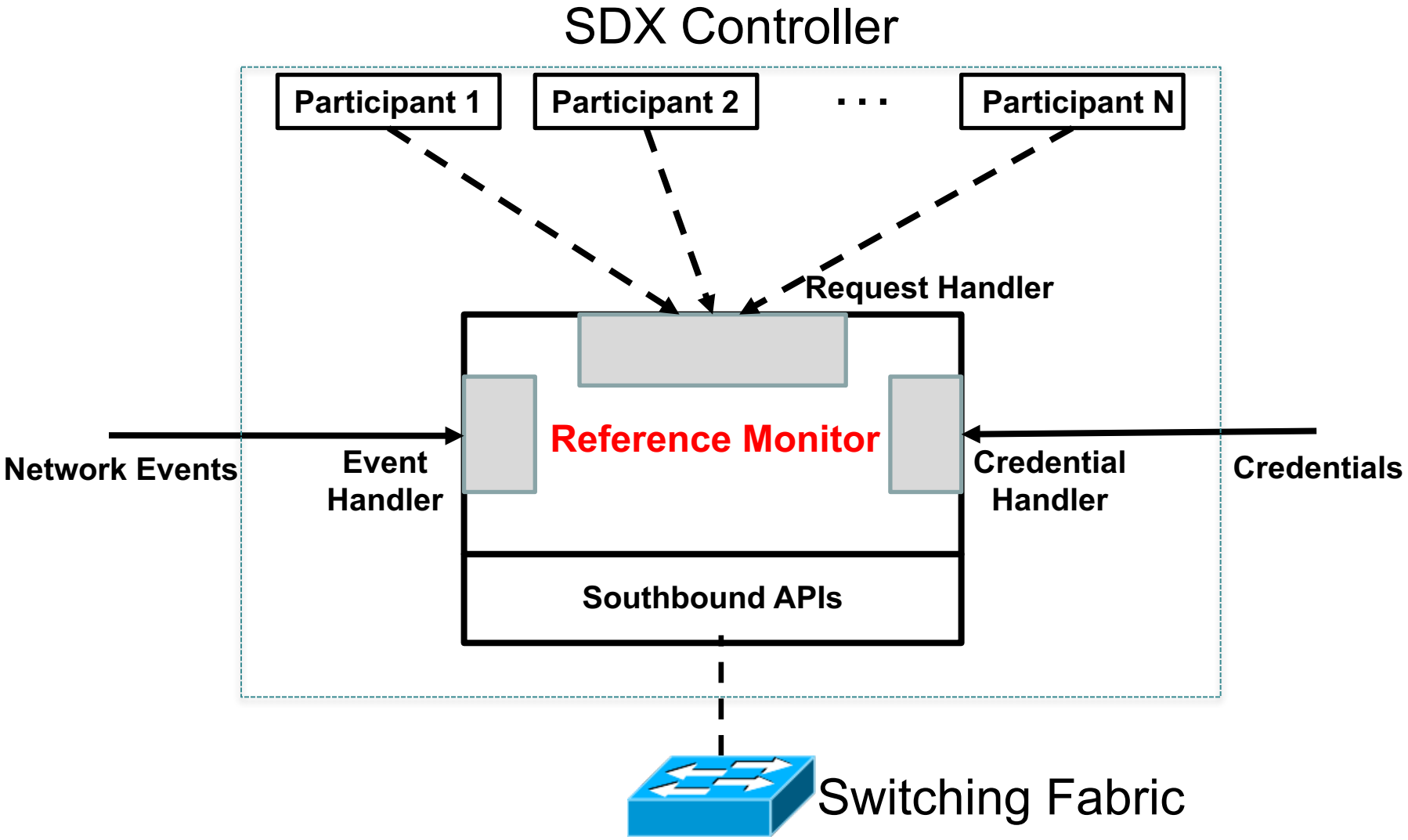
- **Conventional Authorization Logic**
  - Applied over discrete resources
  - Limited allowable actions (read/write etc.)

- **Authorization Logic for Network Control**
  - Resources → Set of packets within some flow space
  - Actions → Transformations on the packet's metadata

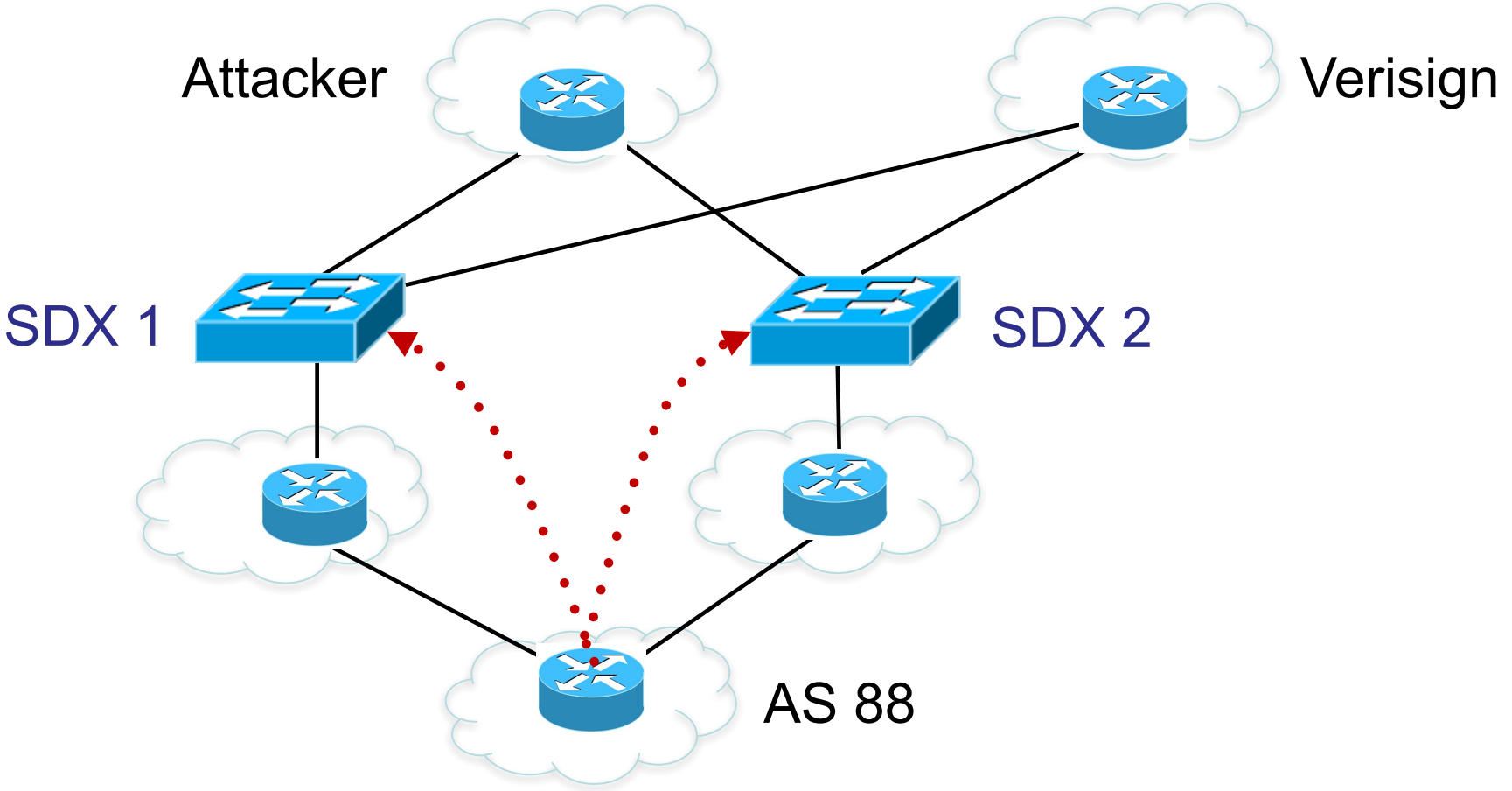
FLANC Authorization Logic

- **Resource Ownership**
  - Principals that own the resource under consideration
- **Allowed Actions**
  - Set of allowed transformations for resource owners,
    T:{sIP, sPort, dIP, dPort, phyPort} → {sIP, sPort, dIP, dPort, phyPort}
  - e.g. Drop Telnet traffic from 10.0.0.1 and 20.0.0.1
    T*:{{10.0.0.1,20.0.0.1}, *,*, {23},*} → {*,*,*,*,{}}*
- **Delegations**
  - Mechanisms by which one principal gives other permission to operate on their resources

FLANC Authorization Logic at SDX

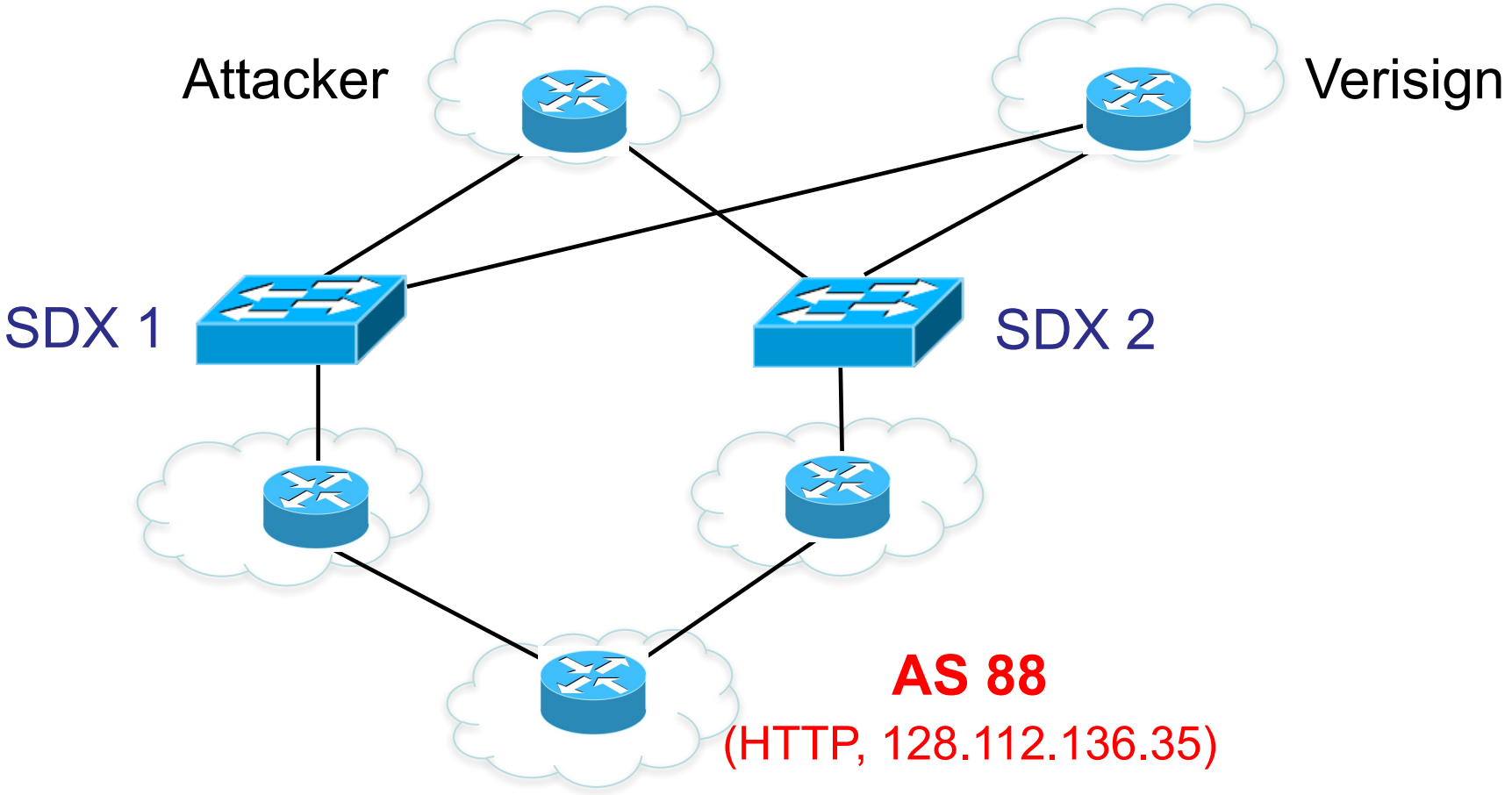


AS 88 sends Delegation Credentials

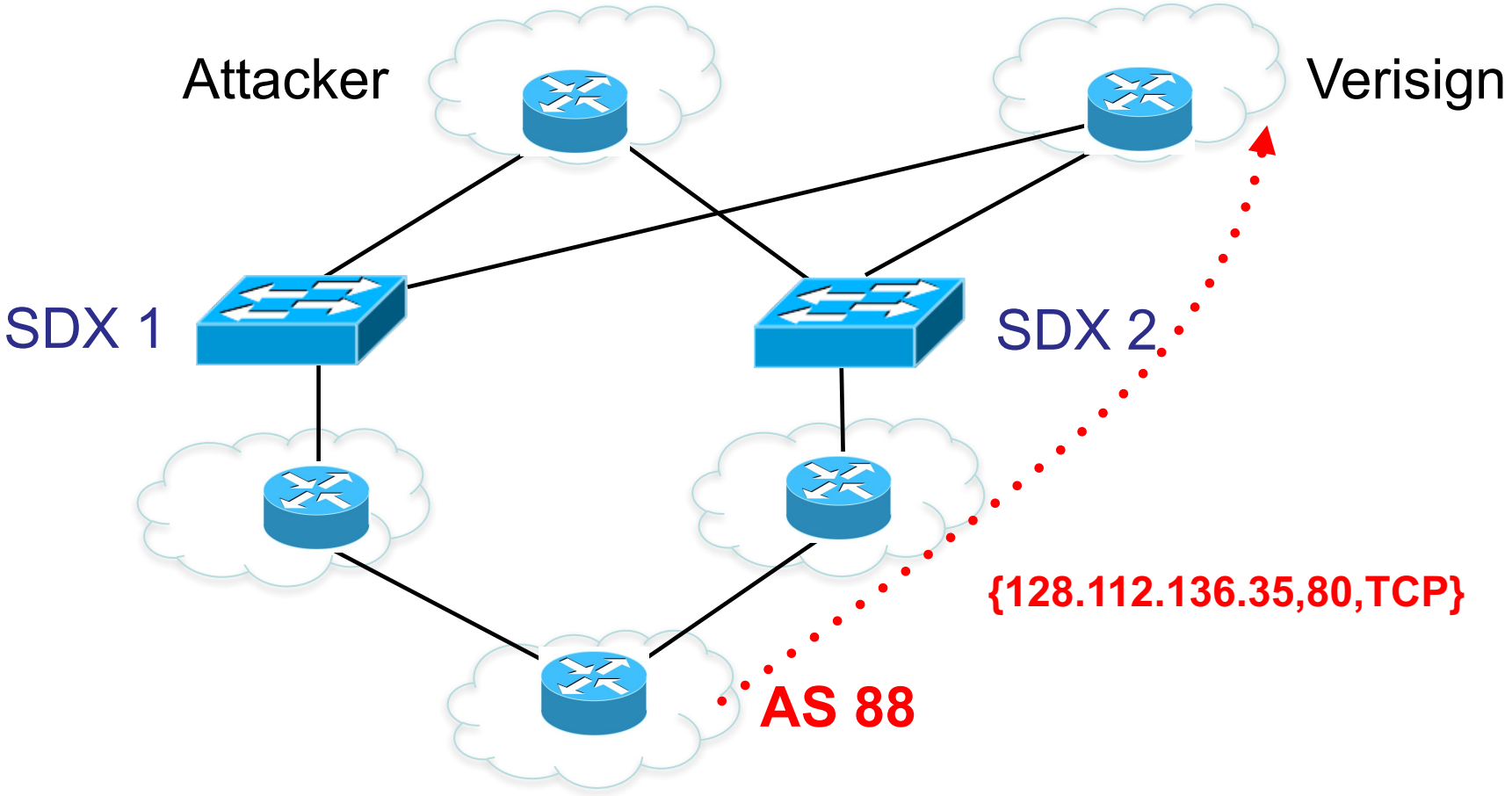


AS 88 says, Verisign speaks for AS 88 for T,
where T:{*,*,{128.112.0.0/16},{80,443},*} → {*,*,*,*,*}

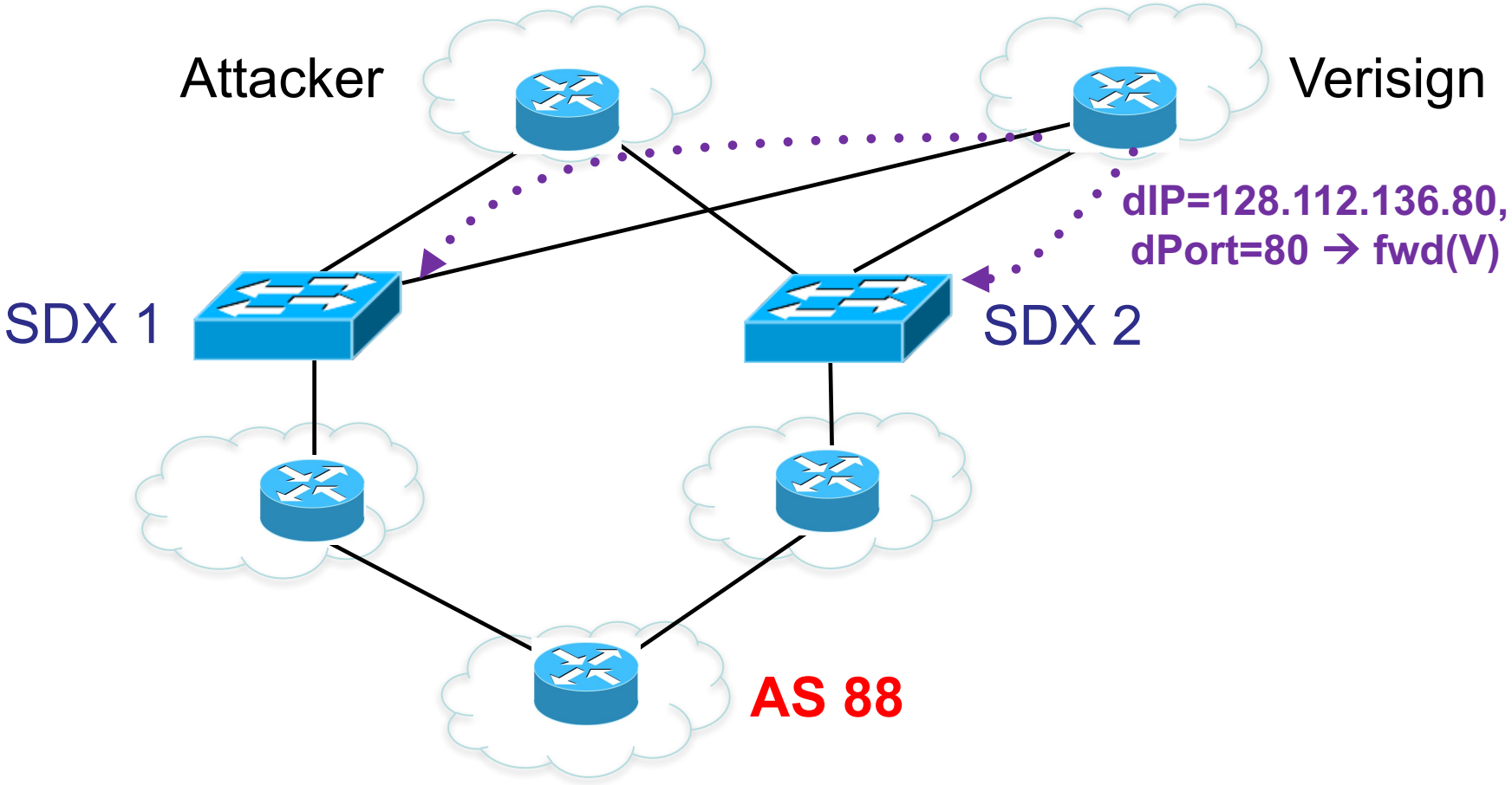
AS 88's HTTP Server under Attack



AS 88 sends DOTS Message



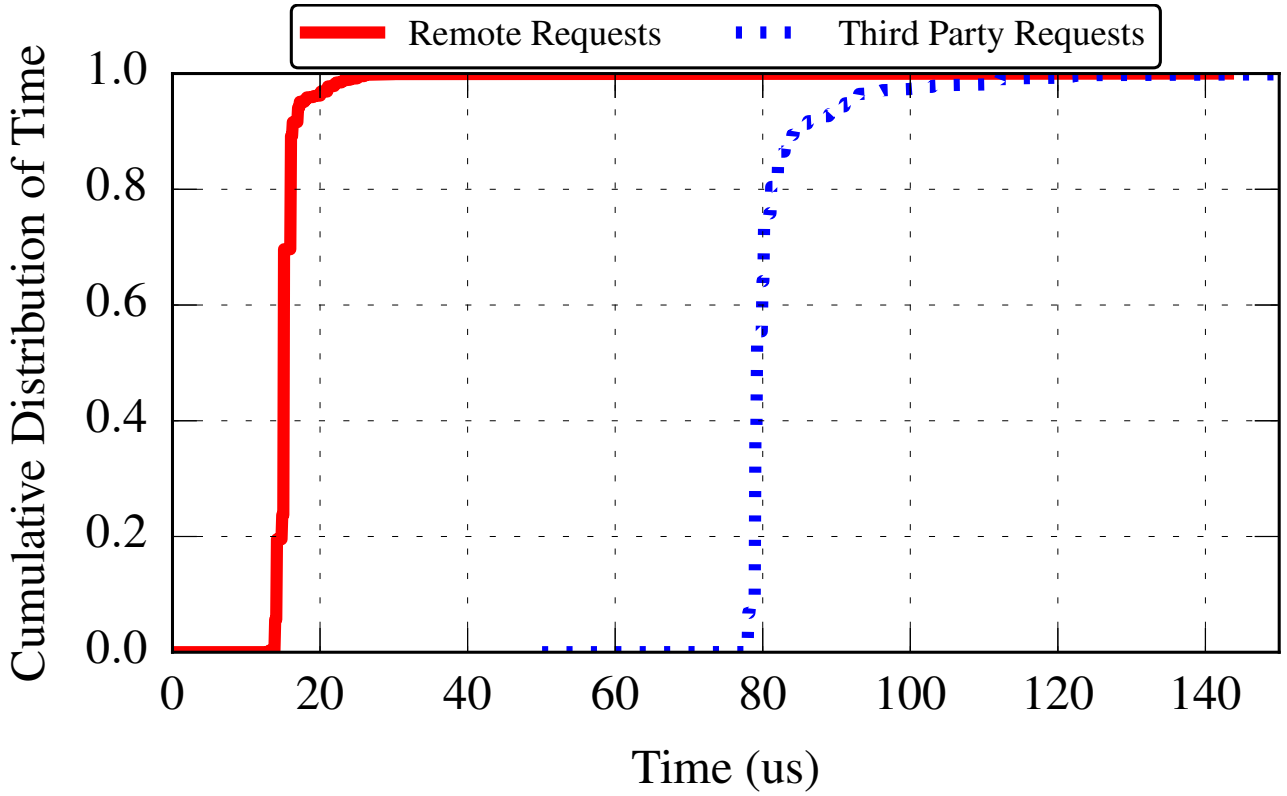
Verisign sends SDN Policies



Checking Authorization at SDX

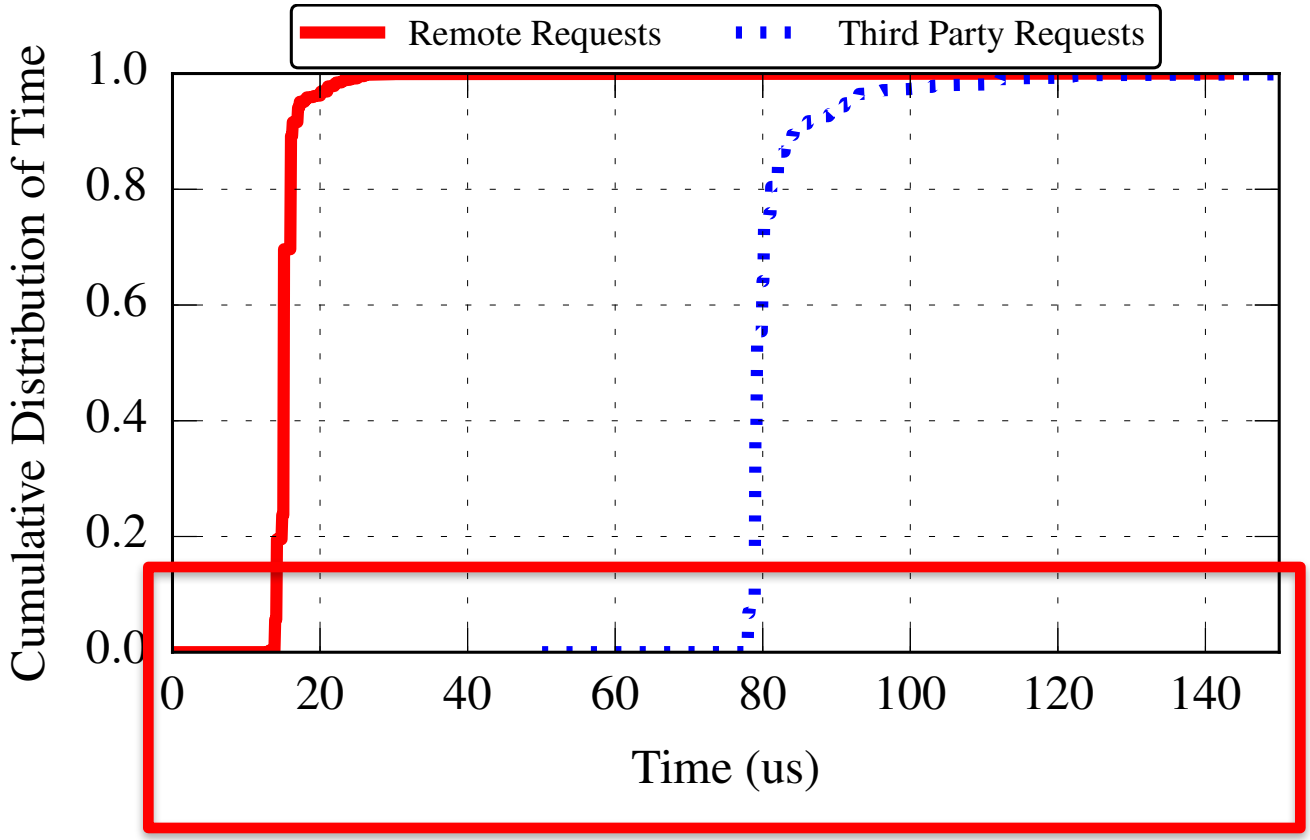
- **Request Handler**
  - Associate request with the principal (Verisign)
  - Extract request transformation
    - T_{req}:{*, *, 128.112.136.80, 80, *} → {*,*,*,*,V}
- **Credential Handler**
  - CA says, "*AS 88 owns {*, *, 128.112.0.0/16, *, * }*"
  - Delegation credentials from AS 88
- **Reference Monitor**
  - Generate a proof, "*Verisign can say T_{req}*"

Evaluation Results



Dataset: AS 88 IPS logs for 1 week, 550K alert events

Evaluation Results



FLANC incurs minimal performance overhead

Takeaways

- Authorizing Network Control at SDX is critical
- *FLANC* is the first step
  - Associates requests with principal
  - Considers flow space abstraction
  - Considers conditional delegations
- *FLANC's* scope is broader than SDX
  - Campus Network
  - Mitigating Route Hijacks