The CrossPath Attack: Disrupting the SDN Control Channel via Shared Links

Jiahao Cao, Qi Li, Renjie Xie, Kun Sun, Guofei Gu,
Mingwei Xu, and Yuan Yang

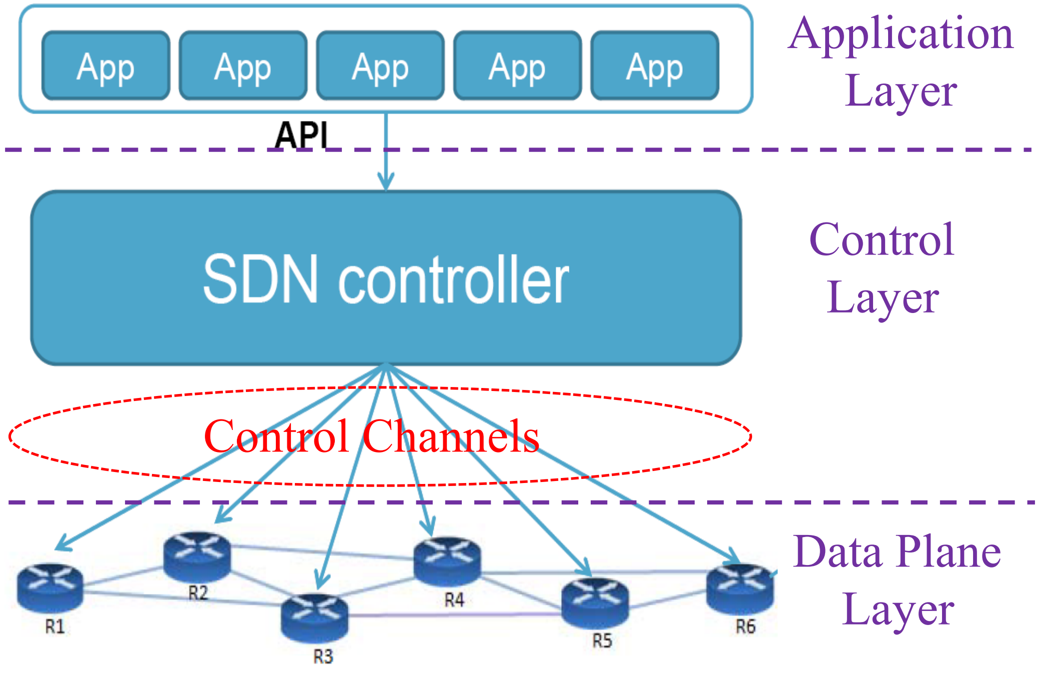
Tsinghua University
GEORGE MASON UNIVERSITY
TEXAS A&M UNIVERSITY

Outline

- Background
- Overview of the CrossPath Attack
- Challenges
- Adversarial Path Reconnaissance
- Evaluation
- Possible Defense
- Conclusion

Software-Defined Networking (SDN)

- Software-Defined Networking
  - separate control and data planes
  - take centralized network control
  - enable network programmability
- SDN Control Channels
  - deliver all control traffic
  - failure results in serious disasters
  - security and reliability are vital



Three-Layer SDN Architecture

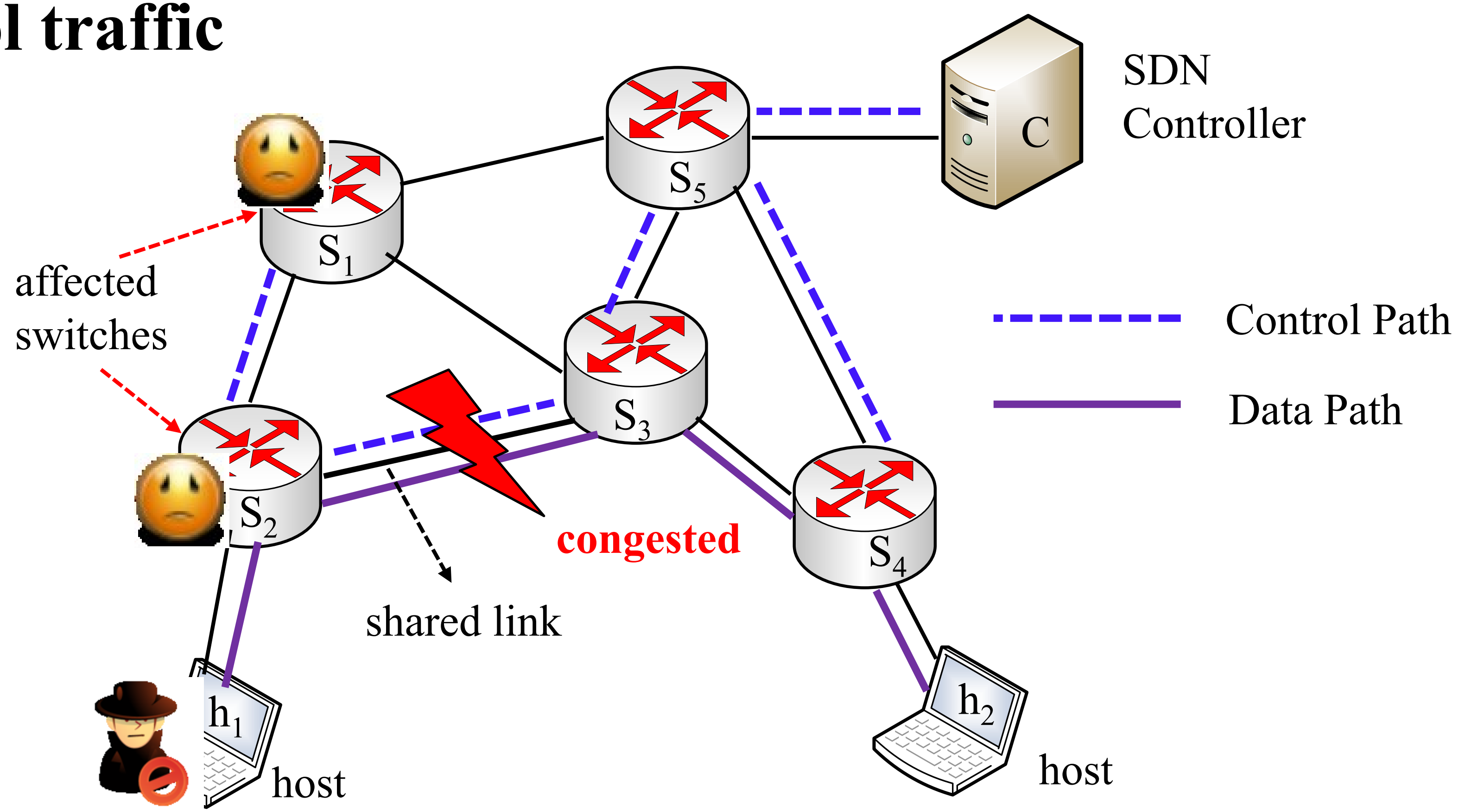
CrossPath Attack

- We uncover a new attack to disrupt SDN control channels
  - leverage **shared links** between paths of control and data traffic
  - allow **data traffic** to disrupt **control traffic**
  - disrupt **a wide range of** SDN functionalities

- Threat Model
  - an attacker compromises a host inside the target SDN
  - the target SDN applies **in-band** control

A Toy Example

- A malicious host sends **data traffic** to congest **shared links** delivering **control traffic**



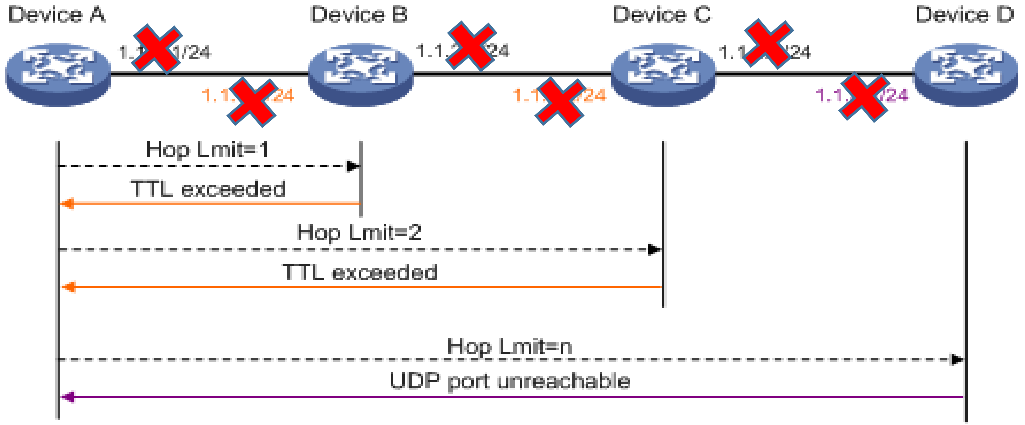
Challenges

- How to find a data path that contains shared links?

- Randomly choose a data path to attack?
  - low success ratio due to only a few shared links

Assume m switches in total,
- $O(m^2)$ total links
- $O(m)$ shared links connecting them with the controller

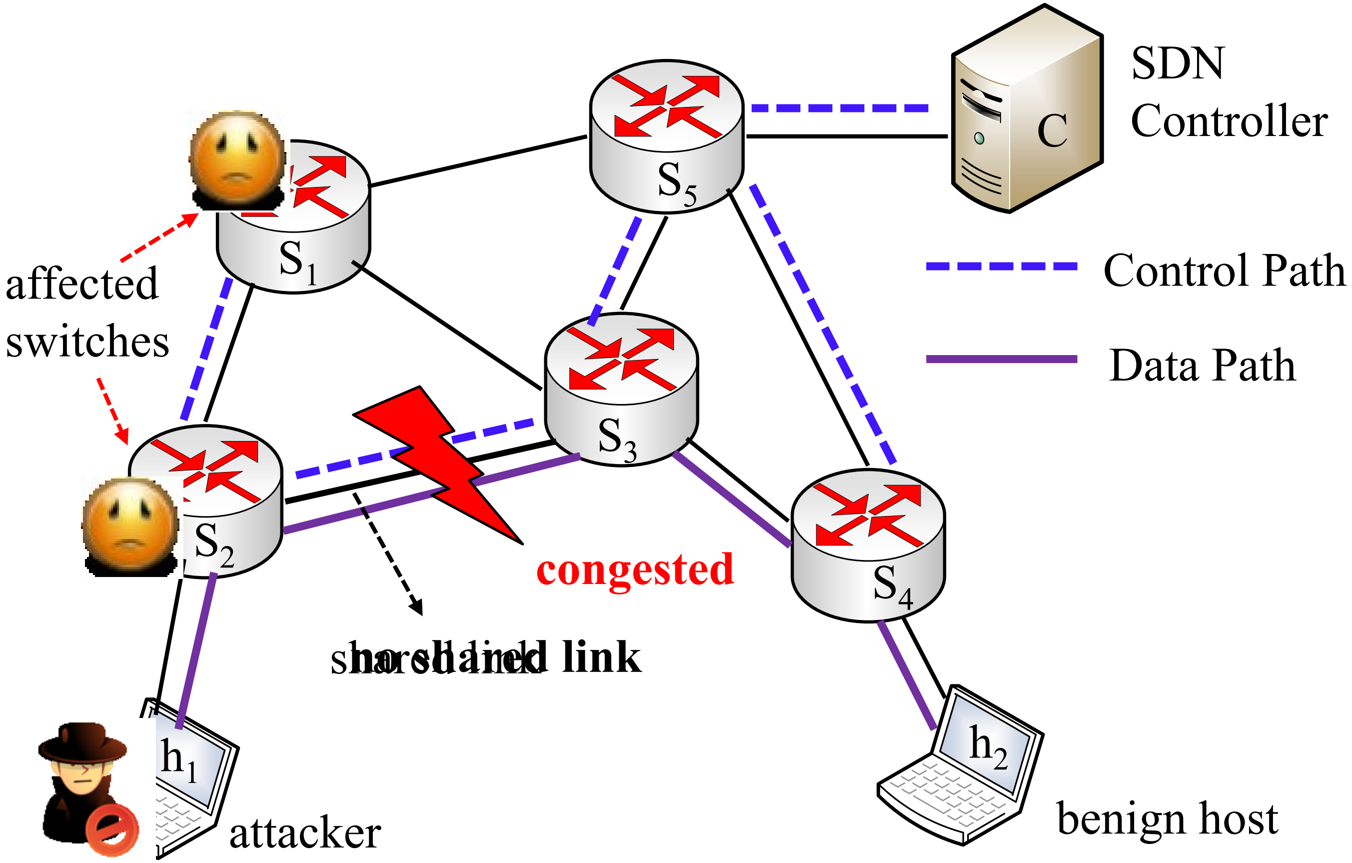
- Apply existing scanning tools to find such a data path?
  - ineffectiveness due to unique SDN data plane



SDN
- **No IP** addresses in switch ports
- **No TTL decrease** for packets passing SDN switches

Adversarial Path Reconnaissance

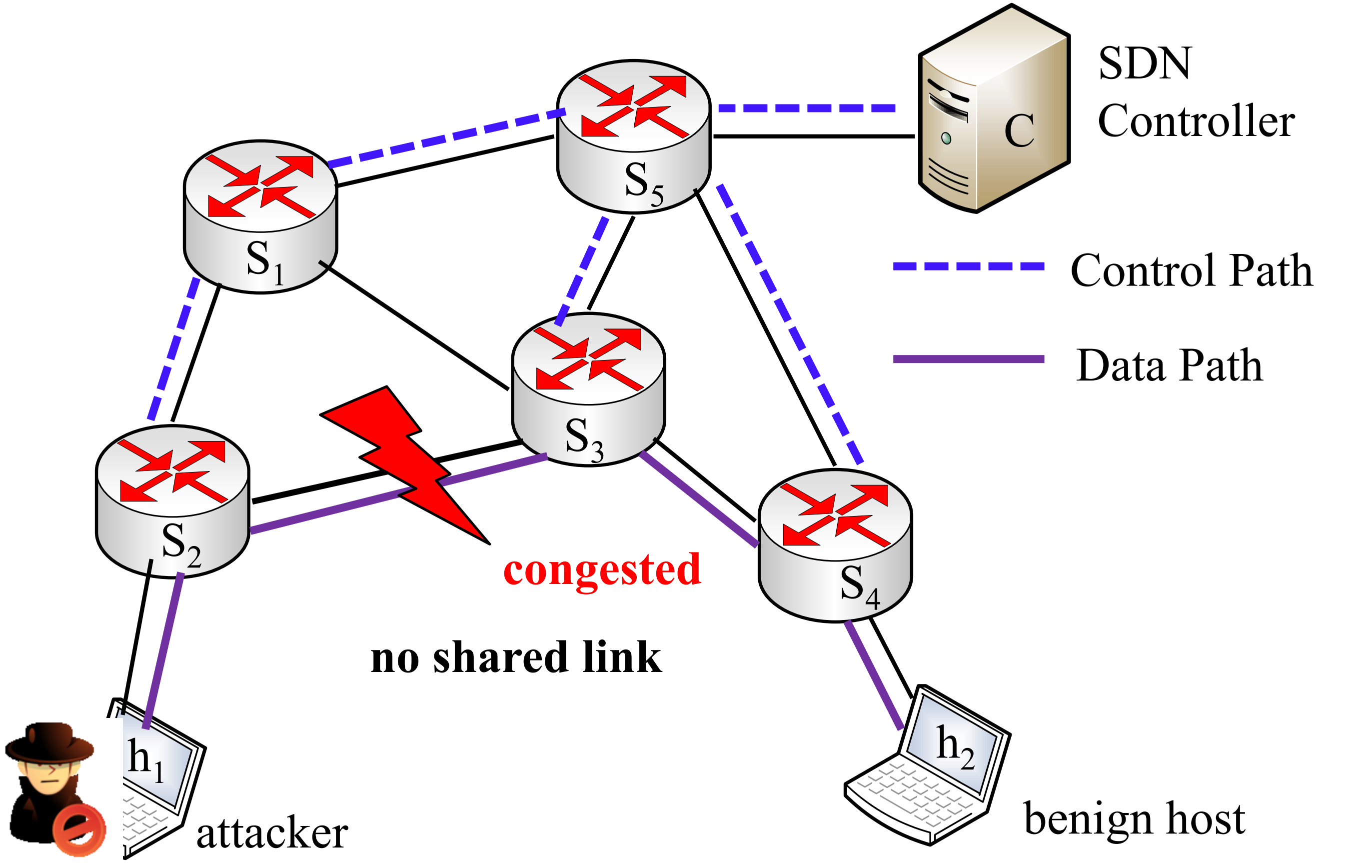
- Key Observation: **control path delays** can be an **indicator** on whether a data path contains shared links



- Control Path Delay between S_2 and C: $T_{S_2,C}$
- Case 1: a data path contains shared links
  - $T_{S_2,C}$ = 100 *ms* due to congestion

Adversarial Path Reconnaissance

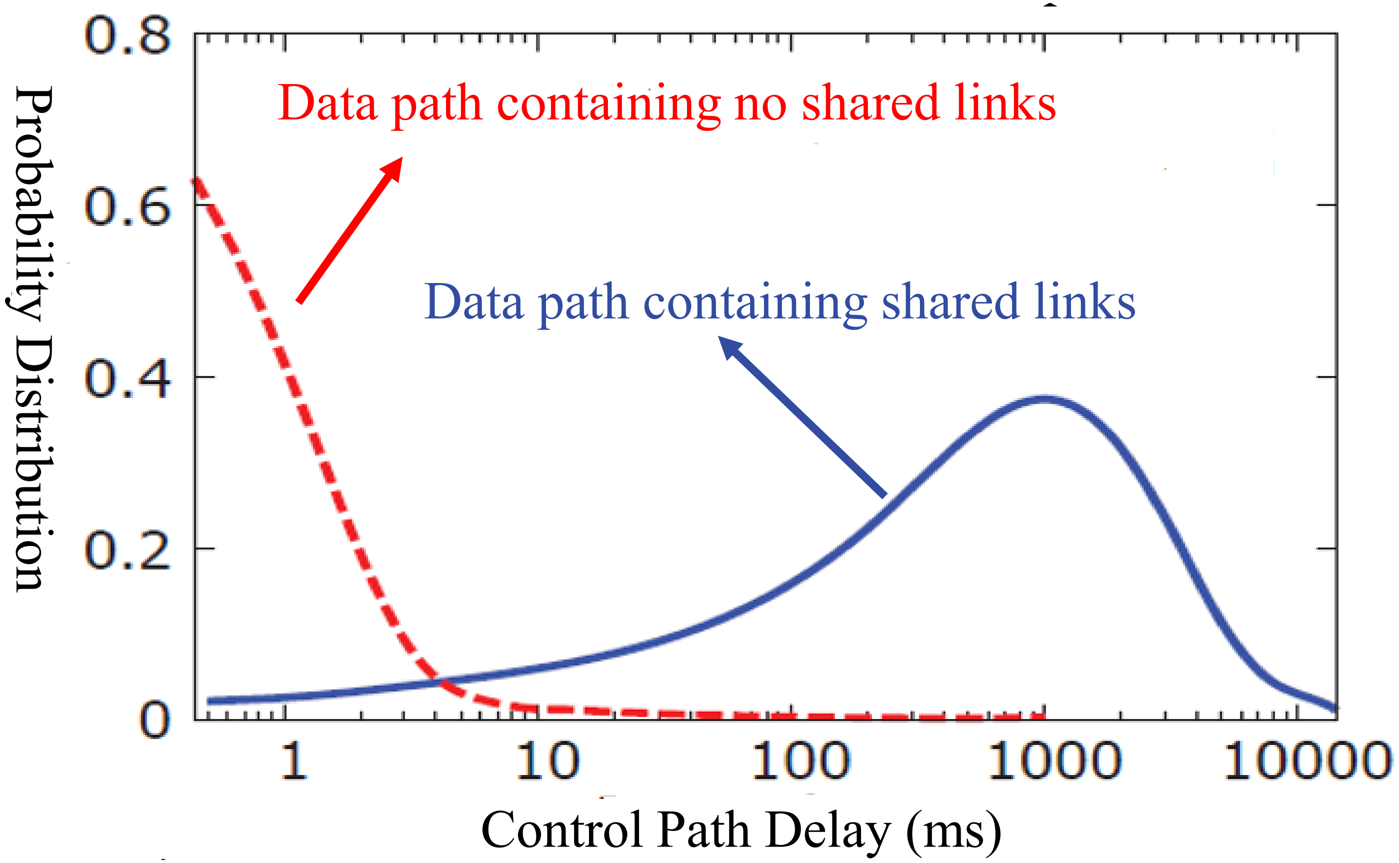
- Key Observation: **control path delays** can be an **indicator** on whether a data path contains shared links



- Control Path Delay between S_2 and C: $T_{S_2,C}$
- Case 1: a data path contains shared links
  - $T_{S_2,C} = 100\ ms$ due to congestion
- Case 2: a data path contains no shared links
  - $T_{S_2,C} = 10\ ms$

Adversarial Path Reconnaissance

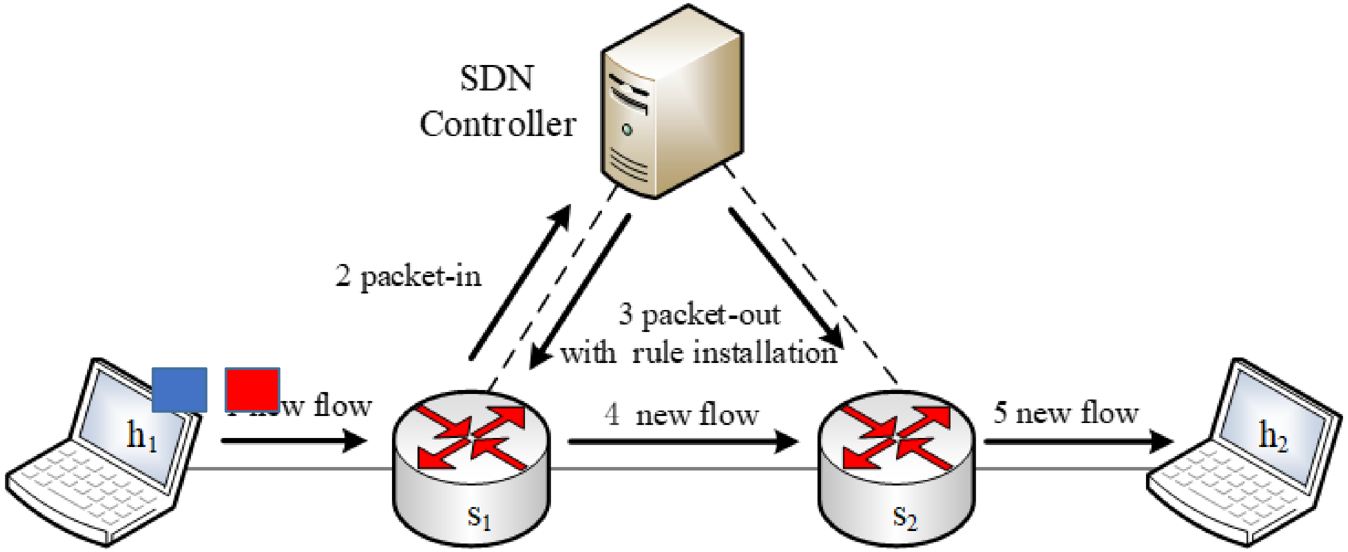
- Key Observation: **control path delays** can be an **indicator** on whether a data path contains shared links



- Control Path Delay between S_2 and C: $T_{S_2,C}$
- Case 1: a data path contains shared links
  - $T_{S_2,C} = 100\ ms$ due to congestion
- Case 2: a data path contains no shared links
  - $T_{S_2,C} = 10\ ms$

Control Path Delay Measurement

- How to measure control path delays with an end host?
  - Leverage **side effects** of dynamic flow rule installation to measure them



$RTT_1 = t_{data} + t_{control}$

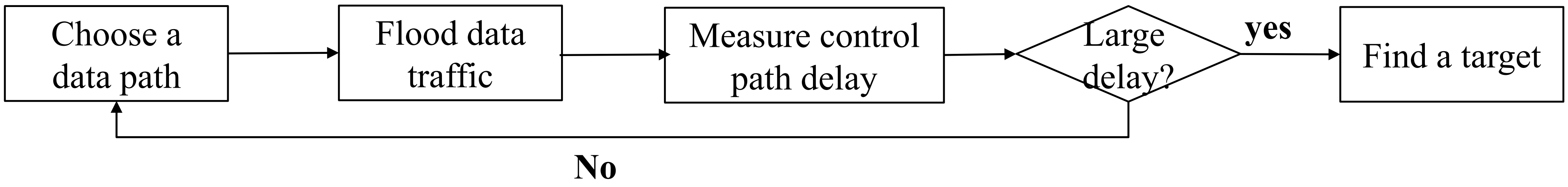
$RTT_2 = t_{data}$

$t_{control} = RTT_1 - RTT_2$

Control path delays can be calculated based on the first two packets of a new flow

Reconnaissance Algorithm

- Algorithm

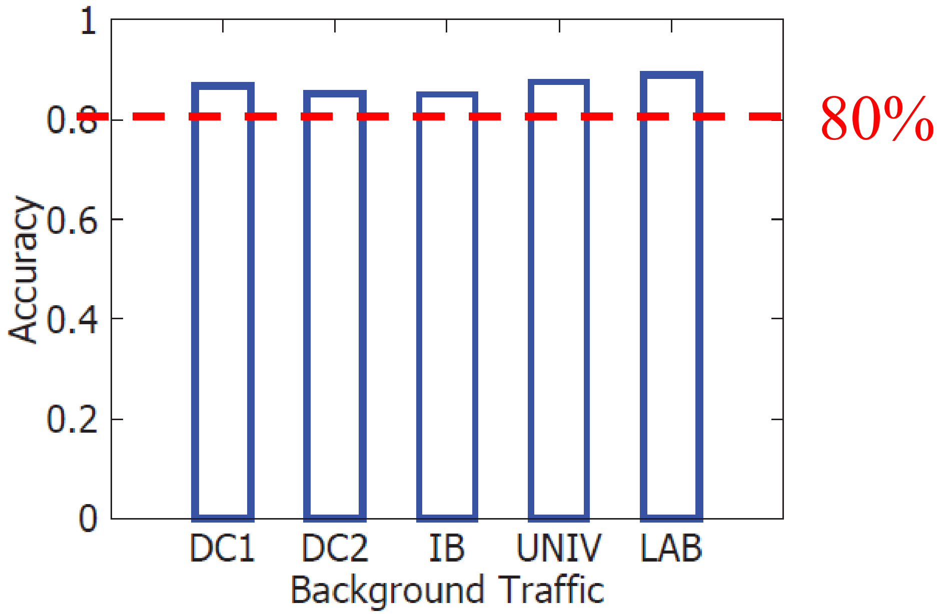


- Optimization
  - Improve the accuracy of reconnaissance
    - e.g., reduce the impacts of network jitters
  - Improve the efficiency of reconnaissance
    - e.g., enable concurrent reconnaissance

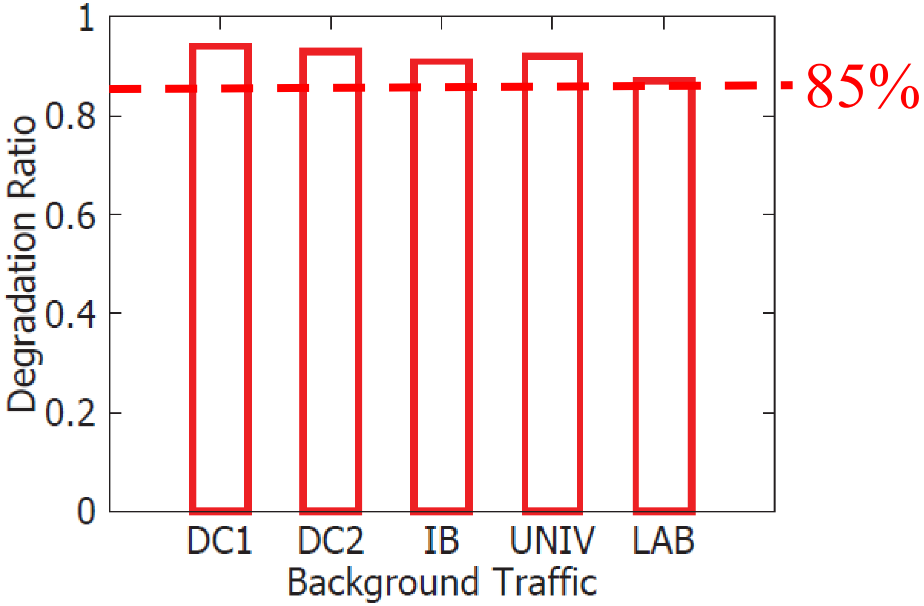
Experiment Setup

- A real SDN testbed consists of
  - commercial hardware SDN switches
  - an open source controller, Floodlight
  - physical hosts connecting to switches
- We replay five types of real traffic trace
  - traffic of two data centers
  - traffic of one university
  - traffic of one internet backbone
  - traffic of one computer lab
- We evaluate
  - the accuracy of adversarial path reconnaissance
  - the degradation ratio of control traffic

Accuracy and Effectiveness



reconnaissance accuracy



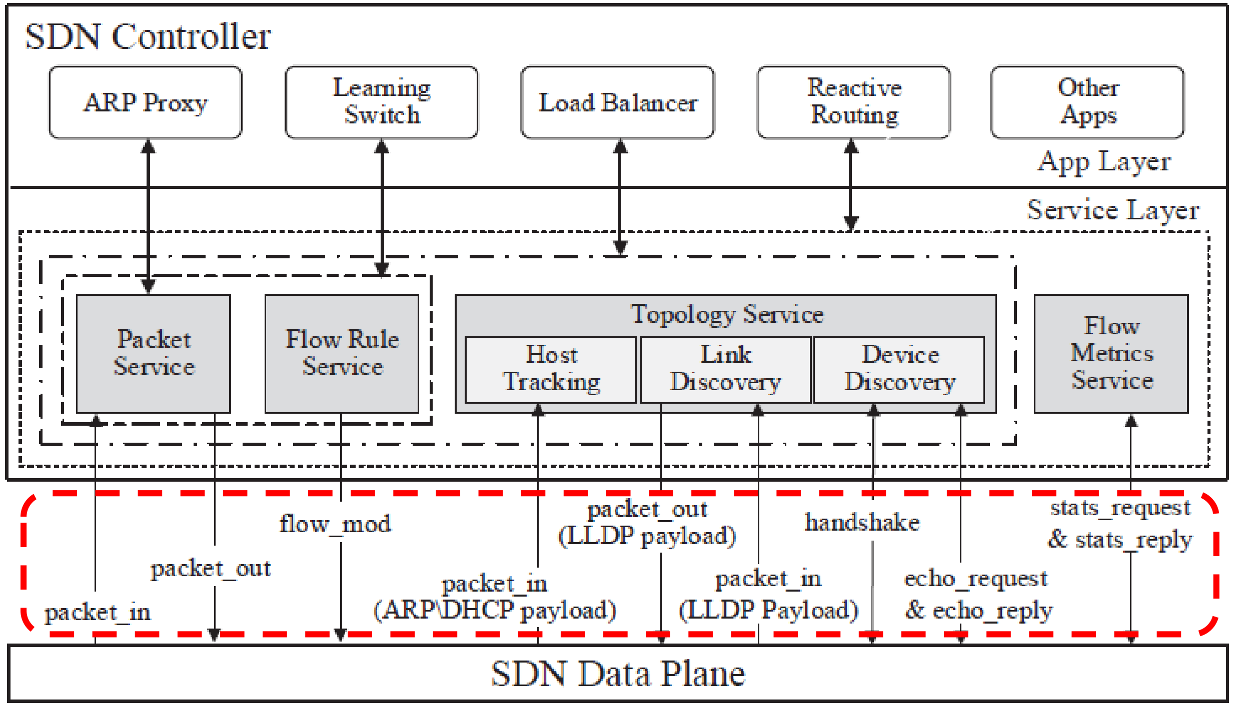
control traffic degradation

DC: datacenter traffic, **IB**: internet backbone traffic, **UNIV**: university traffic, **LAB**: our computer laboratory traffic

Attack Impacts on Network Functionalities

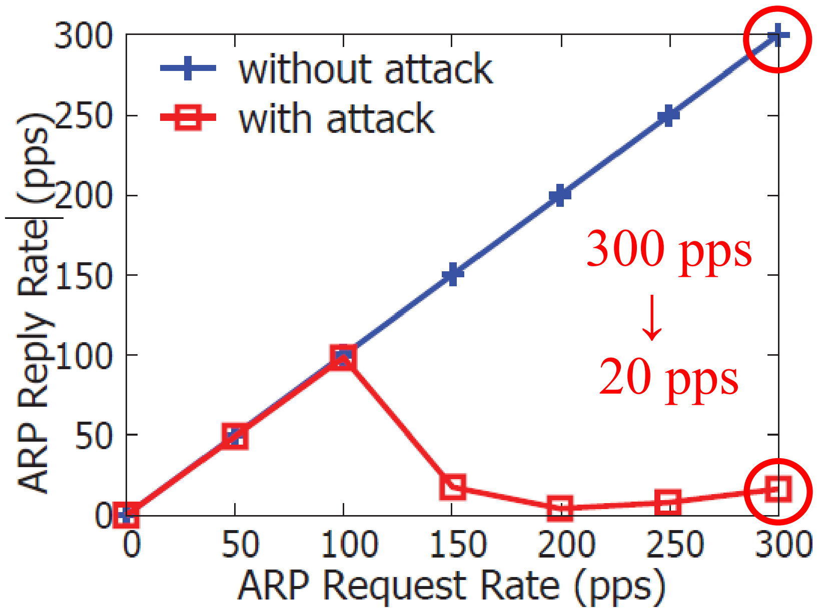
- Almost all SDN applications depend on control messages delivered in control channels to enable network functionalities

- We measure the impacts on three popular SDN APPs
  - ARP Proxy
  - Reactive Routing
  - Load Balancer

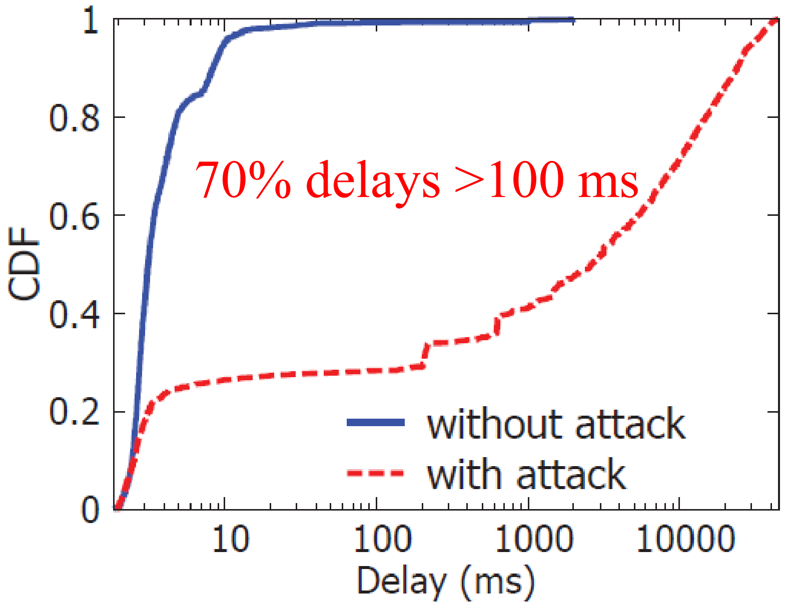


ARP Proxy

- The performance of ARP Proxy significantly degrades



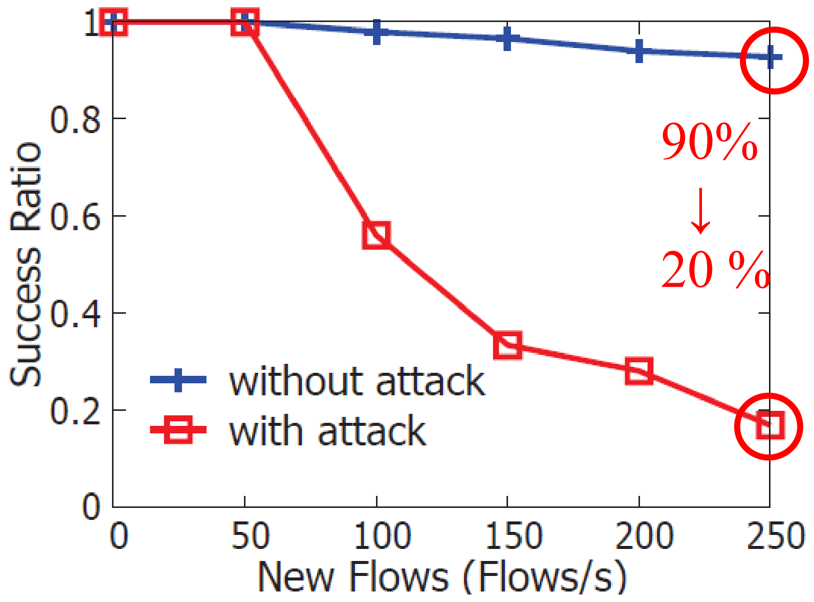
ARP throughput



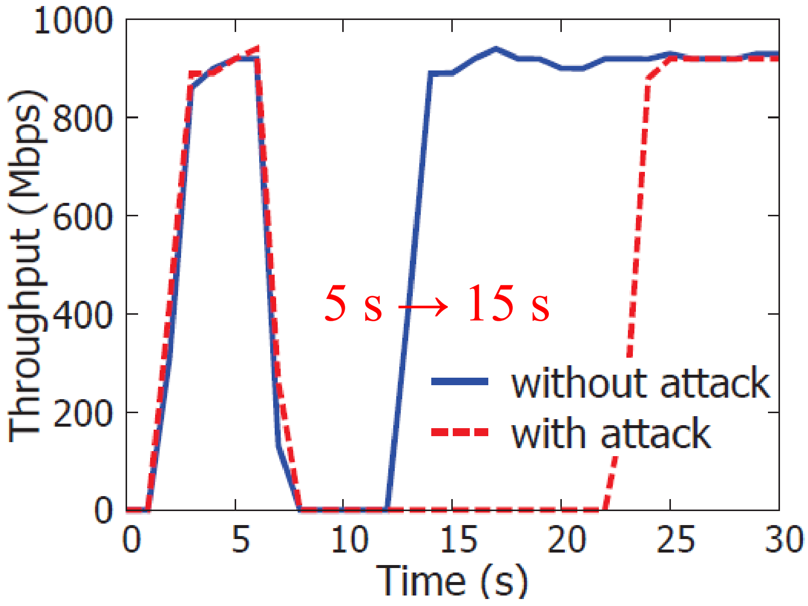
ARP query delay

Reactive Routing

- Reactive Routing generates various anomalies



success ratio of rule installation



host migration time

Reactive Routing

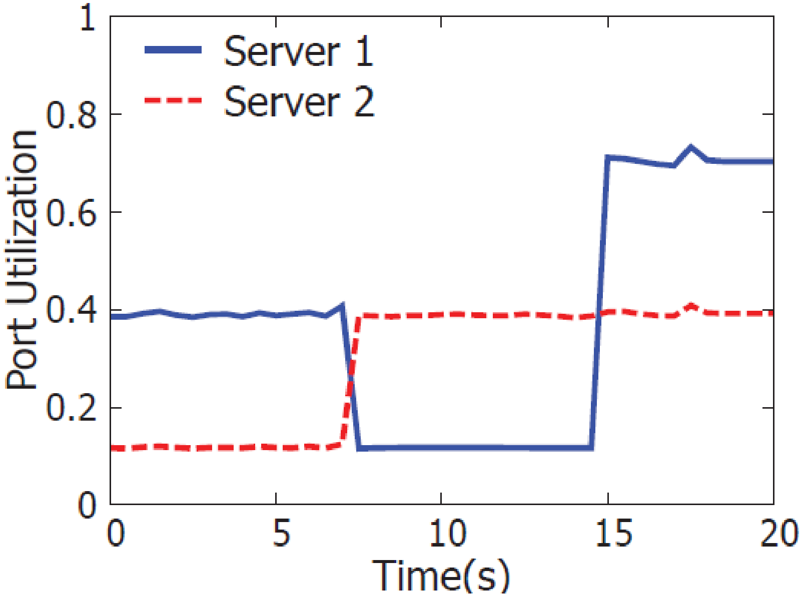
- Reactive Routing generates various anomalies

```
17:37:46.344 INFO  [n.f.t.TopologyInstance] Route [id=RouteId [src=1c:48:cc:37:ab:a0:a8:41
dst=9d:54:cc:37:ab:a0:a8:41], switchPorts=[[id=1c:48:cc:37:ab:a0:a8:41, port=37],
[id=9d:54:cc:37:ab:a0:a8:41, port=31]]]
17:38:01.62 INFO  [n.f.l.i.LinkDiscoveryManager] Inter-switch link removed: Link
[src=a4:e7:cc:37:ab:a0:a8:41 outPort=38, dst=9d:54:cc:37:ab:a0:a8:41, inPort=42, latency=6]
17:38:01.95 INFO  [n.f.t.TopologyManager] Recomputing topology due to: link-discovery-
updates
17:38:01.345 INFO  [n.f.t.TopologyInstance] Route [id=RouteId [src=1c:48:cc:37:ab:a0:a8:41
dst=9d:54:cc:37:ab:a0:a8:41], switchPorts=[[id=1c:48:cc:37:ab:a0:a8:41, port=32],
[id=a4:e7:cc:37:ab:a0:a8:41, port=36], [id=a4:e7:cc:37:ab:a0:a8:41, port=38],
[id=9d:54:cc:37:ab:a0:a8:41, port=42]]]
```

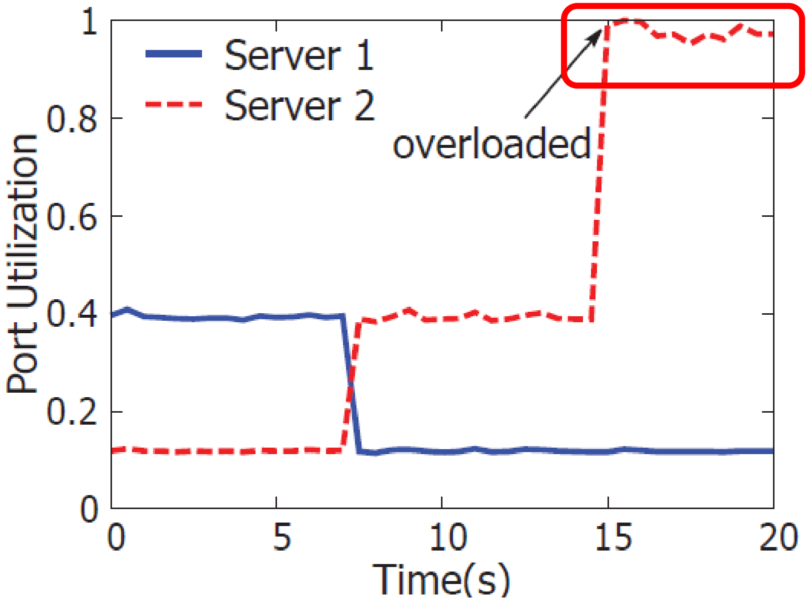
A routing path is evicted due to a deactivated link

Load Balancer

- Load balancer incorrectly balances traffic among servers



without the attack



with the attack

Possible Defense

- Deliver control traffic with a high priority
  - implementation with priority queue or weighted round robin queue

- Proactively reserve bandwidth for control traffic
  - implementation with meter tables

Defense Strategy	Rule	Match	Actions
Control traffic delivery with high priority [1]	#1	control flows	OutPort(x), . . . , *SetQueue(ID=highPriQueue)*
	#2	data flows	OutPort(x), . . . , *SetQueue(ID=lowPriQueue)*
Proactive bandwidth reservation for control traffic [2]	#1	data flows	OutPort(x), . . . , *SetMeter(ID=RateLimit)*

[1] It requires SDN switches to support PQ or WRR queuing mechanism.
[2] It is used when SDN switches fail to enable PQ or WRR mechanism.

Conclusion

- Data traffic passing shared links can congest control traffic to disrupt SDN control channels
- A data path containing shared links can be found by measuring control path delays and leveraging side effects of dynamic rule installation
- Network administrators should enable priority queue or reserve bandwidth for SDN control traffic to protect control channels

Thank you!

Jiahao Cao
caojh15@mails.tsinghua.edu.cn

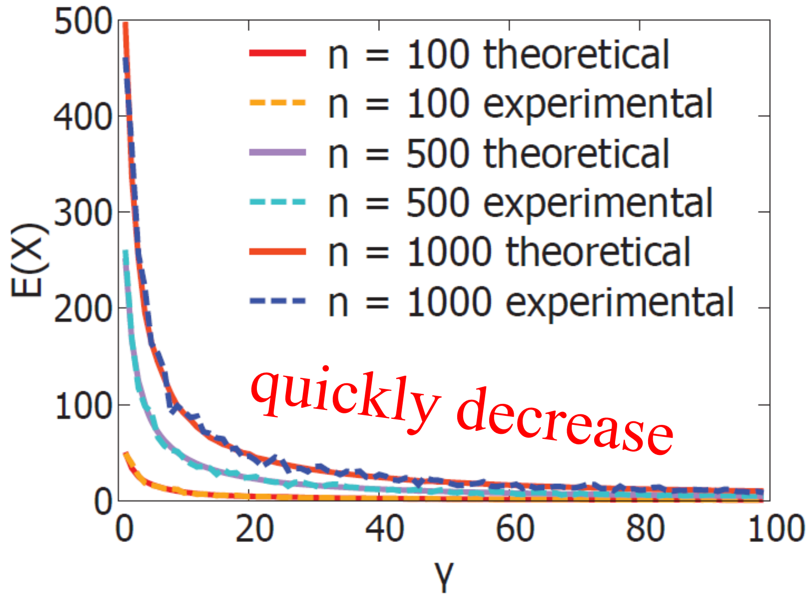
Backup: Theoretical Analysis

- The number of explored data paths to find a target data path containing a shared link

$$E(X) = \sum_{k=1}^{n-\gamma} \frac{k\gamma}{n-k} \prod_{j=0}^{k-2} \left(1 - \frac{\gamma}{n-1-j}\right)$$

n: The total number of hosts in SDN

γ**:** The total number of data paths containing shared links, depending on the topology and the routing decision



Backup: Coverage

- Simulation among 261 real network topologies
- Connections between the controller and switches
  - shortest path (SP)
  - minimum spanning tree (MST)
  - random (RS)