I Know What Your Packet Did Last Hop: Using Packet Histories to Troubleshoot Networks

Nikhil Handigol

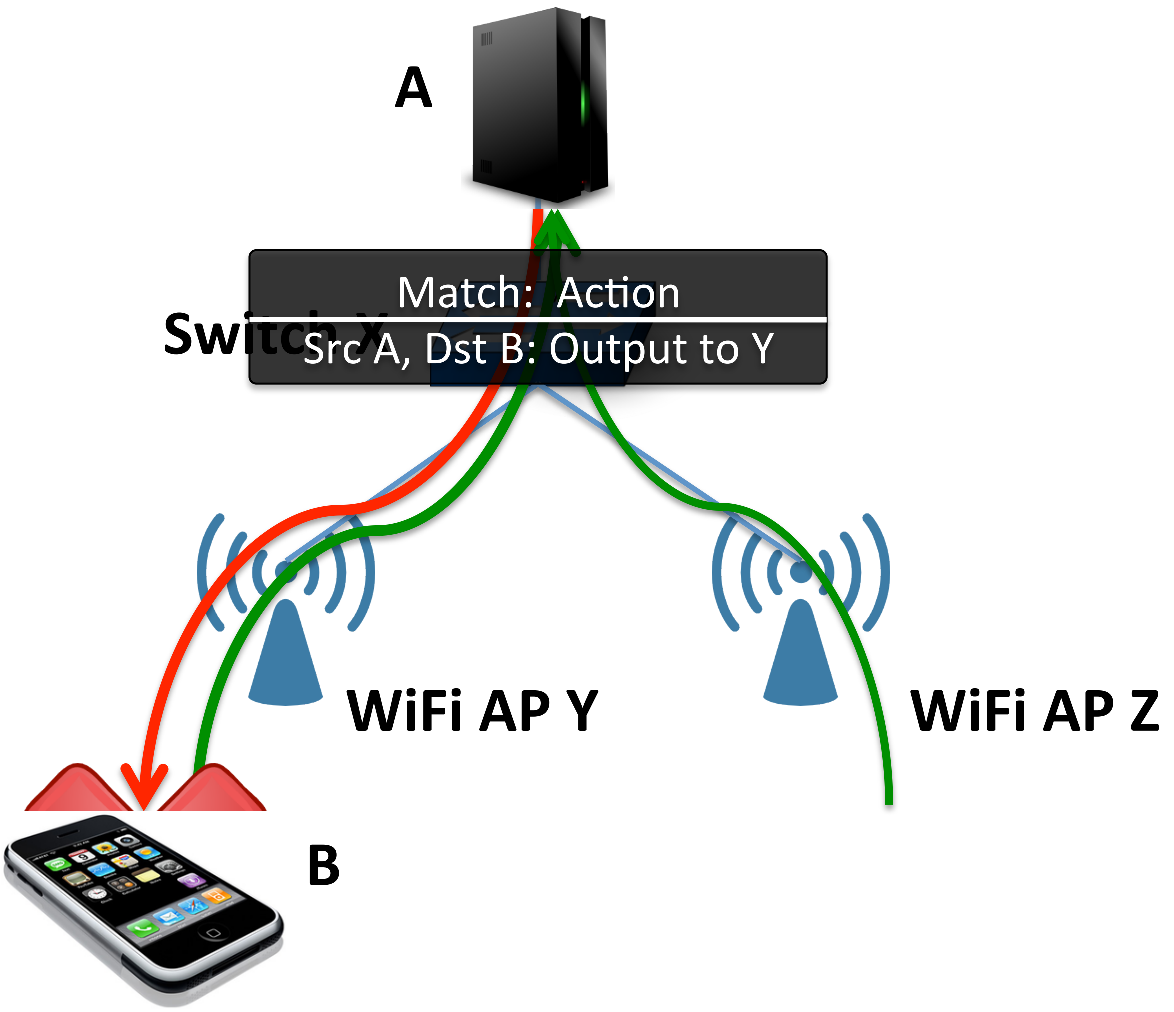
With

Brandon Heller, Vimal Jeyakumar, David Mazières, Nick McKeown

NSDI 2014, Seattle, WA

April 2, 2014

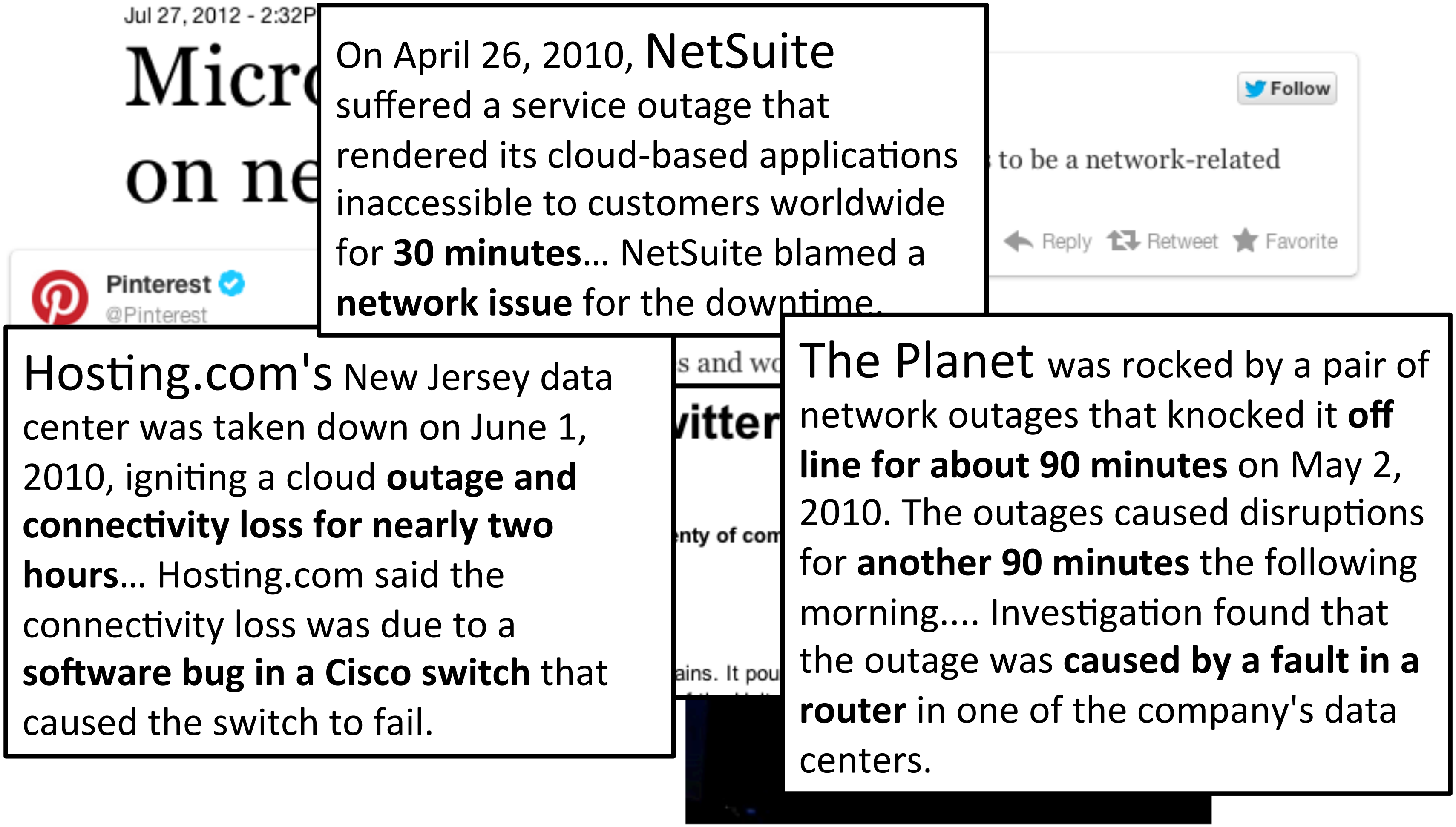
Bug Story: Incomplete Handover



Match:	Action
Src A, Dst B:	Output to Y

Network Outages

make news headlines

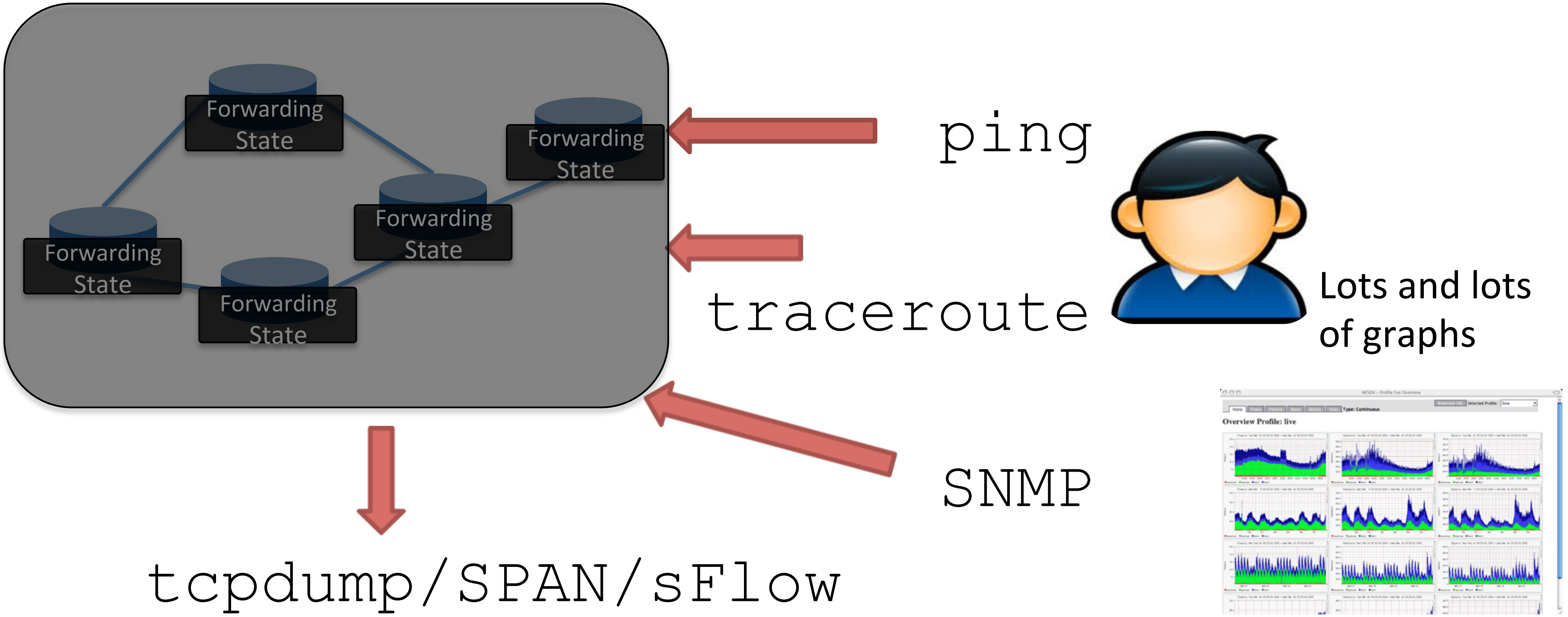


On April 26, 2010, NetSuite suffered a service outage that rendered its cloud-based applications inaccessible to customers worldwide for **30 minutes**… NetSuite blamed a **network issue** for the downtime.

Hosting.com's New Jersey data center was taken down on June 1, 2010, igniting a cloud **outage and connectivity loss for nearly two hours**… Hosting.com said the connectivity loss was due to a **software bug in a Cisco switch** that caused the switch to fail.

The Planet was rocked by a pair of network outages that knocked it **off line for about 90 minutes** on May 2, 2010. The outages caused disruptions for **another 90 minutes** the following morning.... Investigation found that the outage was **caused by a fault in a router** in one of the company's data centers.

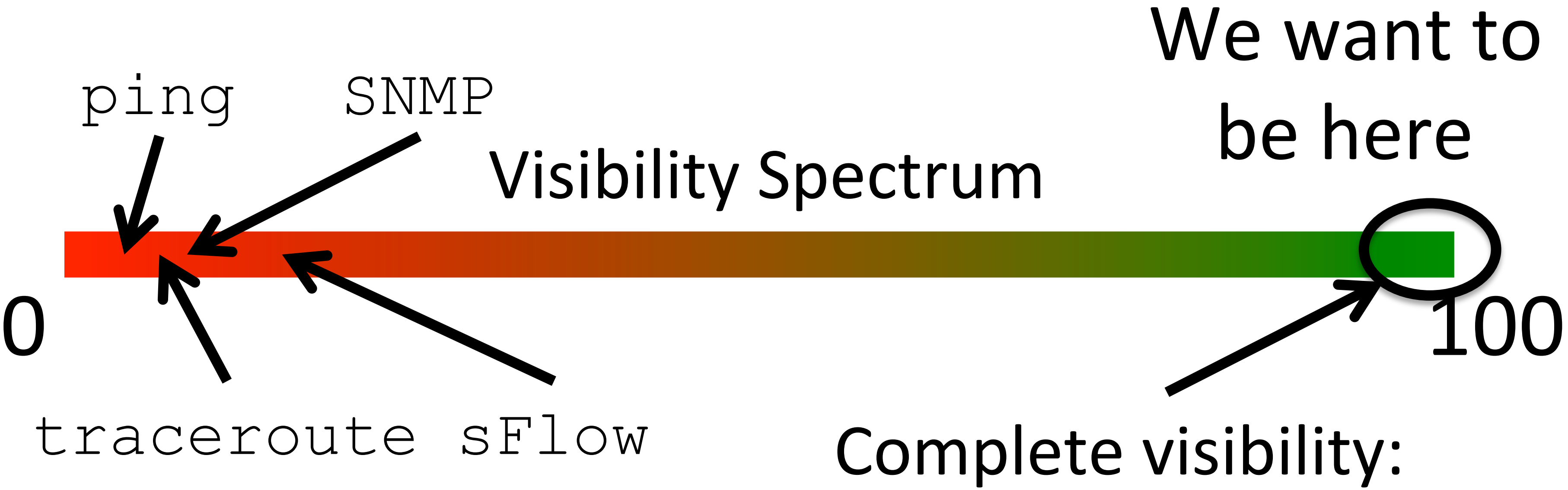
Troubleshooting Networks is Hard Today



- Tedious and ad hoc
- Requires skill and experience
- Not guaranteed to provide helpful answers

(source: NANOG Survey in "Automatic Test Packet Generation", Hongyi Zeng, et. al.)

We want complete network visibility



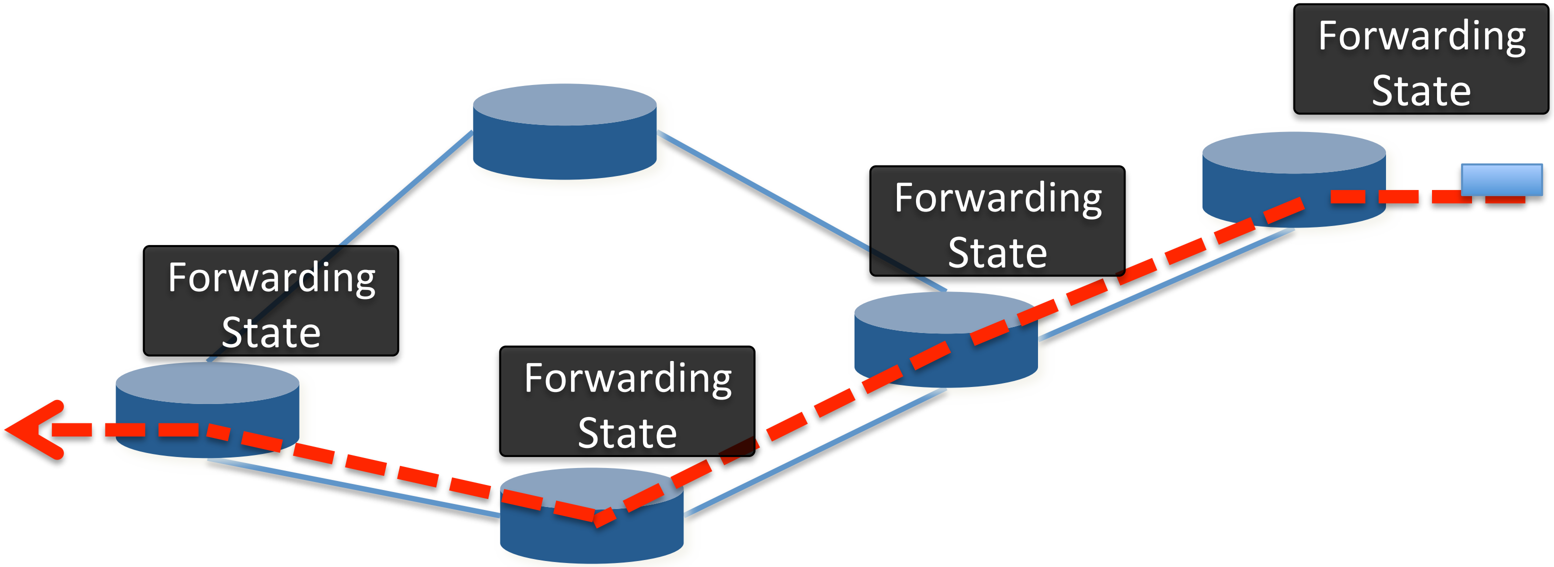
Complete visibility: ***every event*** that ***ever*** happened to ***every packet***

Talk Outline

1. How to achieve complete network visibility
   - An abstraction: Packet History
   - A platform: NetSight

2. Why achieving complete visibility is feasible
   - Data compression
   - MapReduce-style scale-out design

Packet History

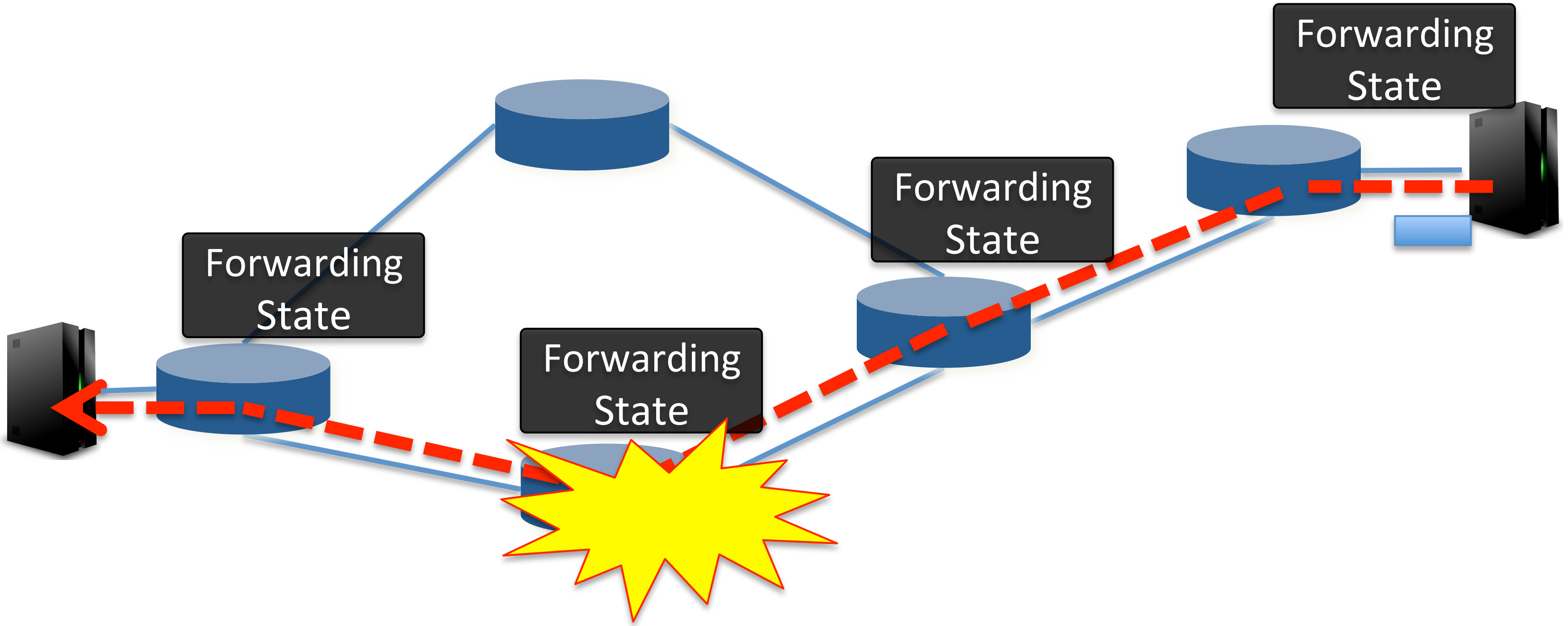


Packet history = Path taken by a packet

+ Header modifications

+ Switch state encountered

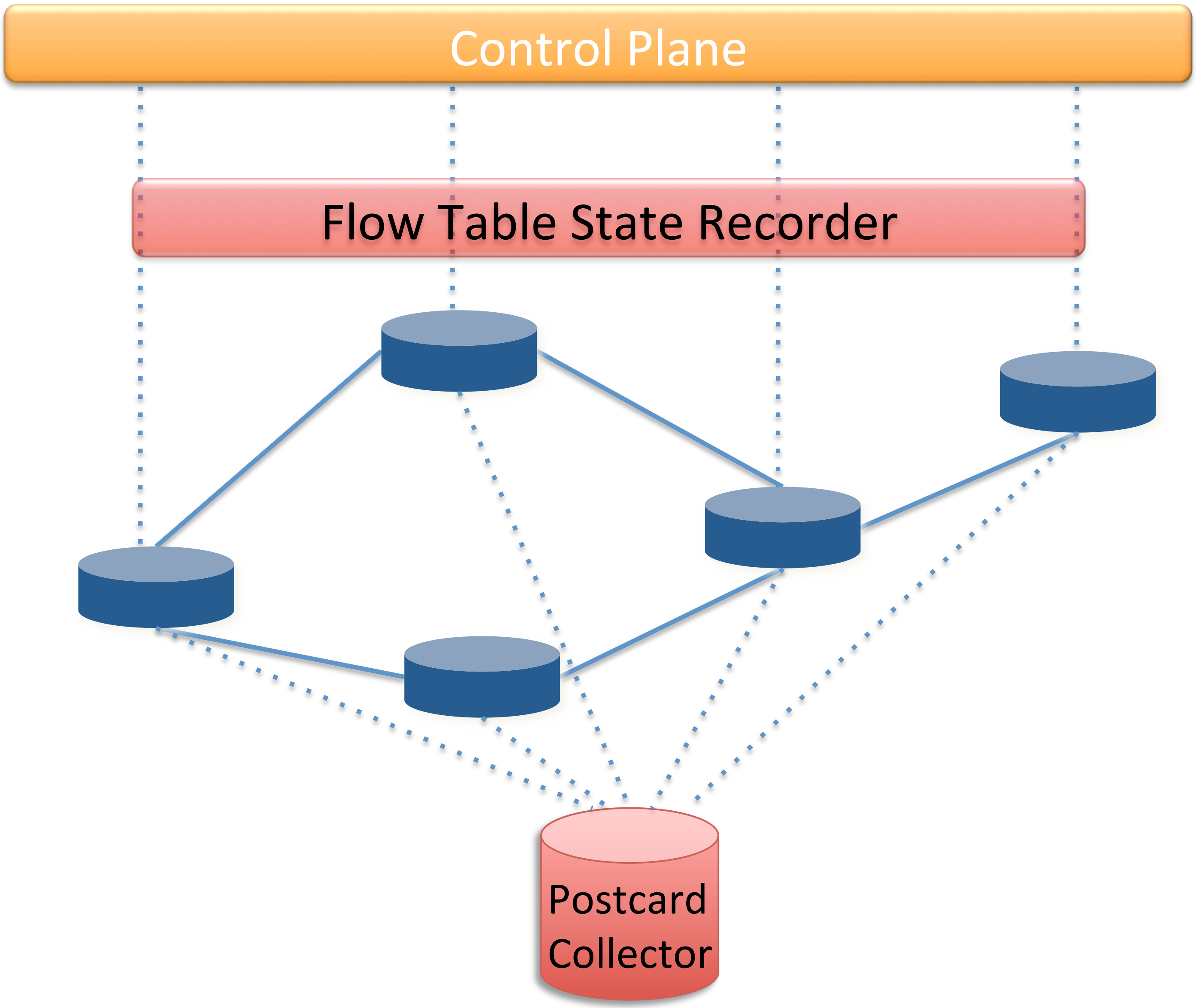
Our Troubleshooting Workflow

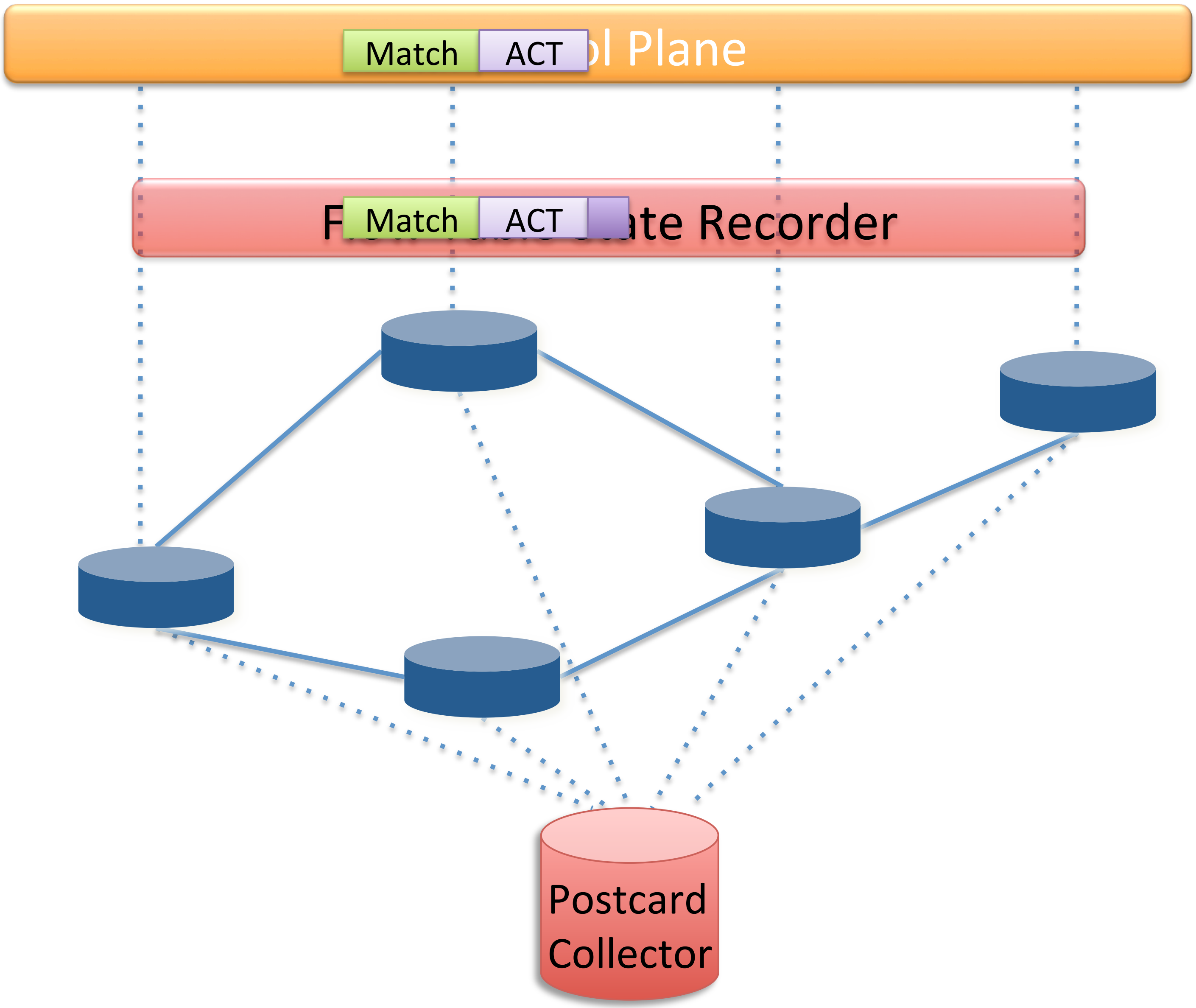


1. Record and store all packet histories
2. Query and use packet histories of errant packets

NetSight

A platform to capture and filter packet histories of interest

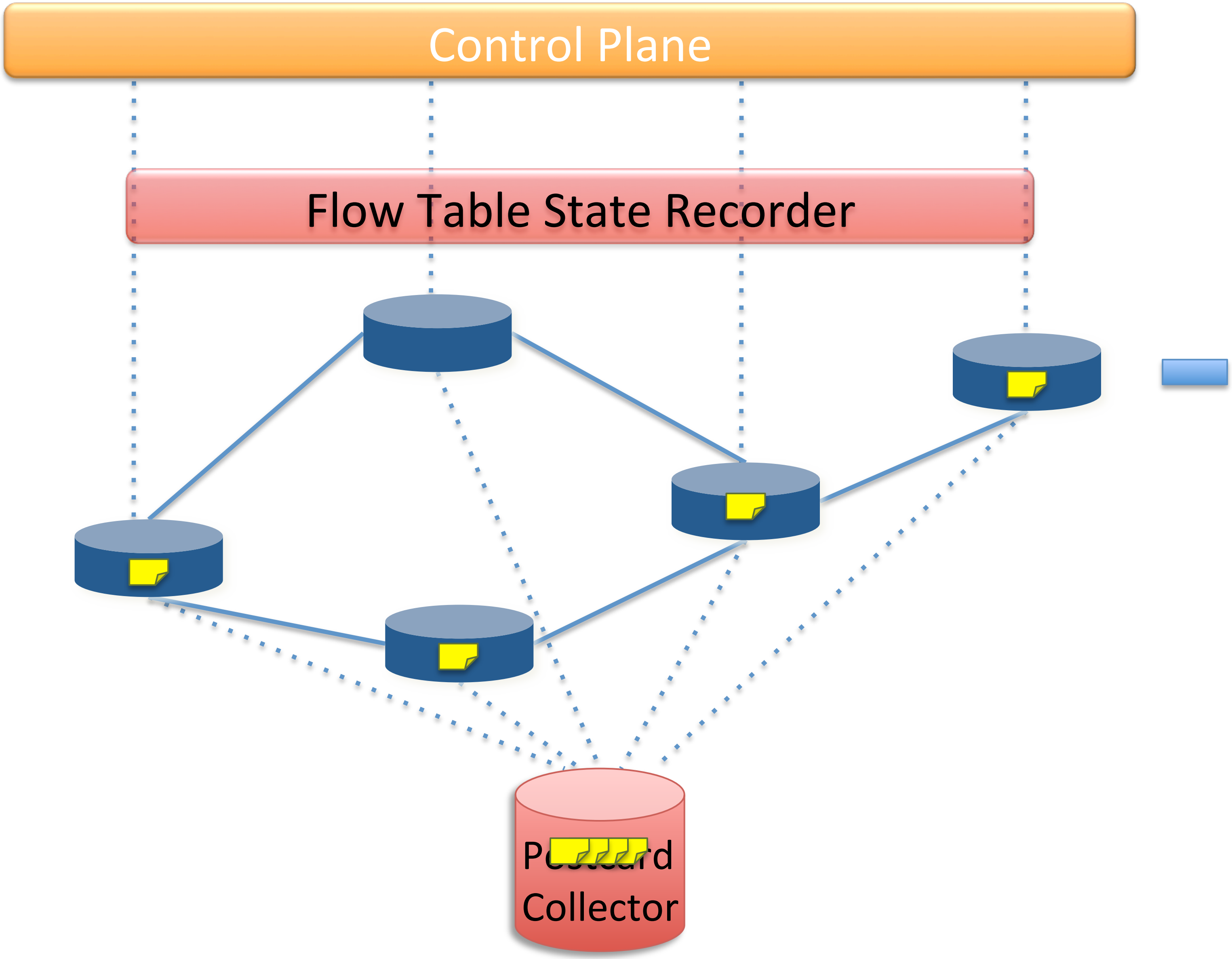

Control Plane
Flow Table State Recorder
Postcard Collector

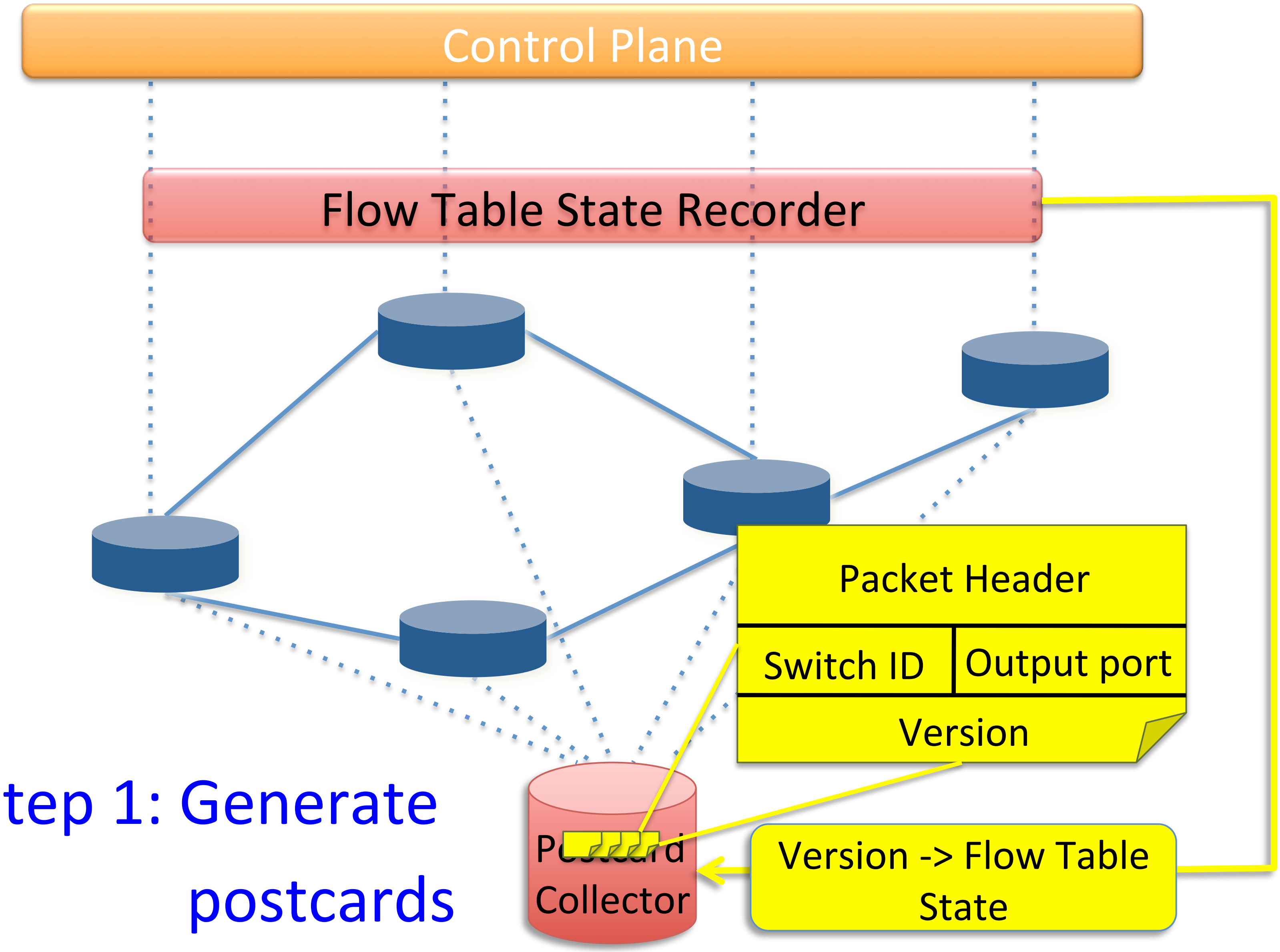


Match ACT ol Plane

F Match ACT ate Recorder

Postcard Collector

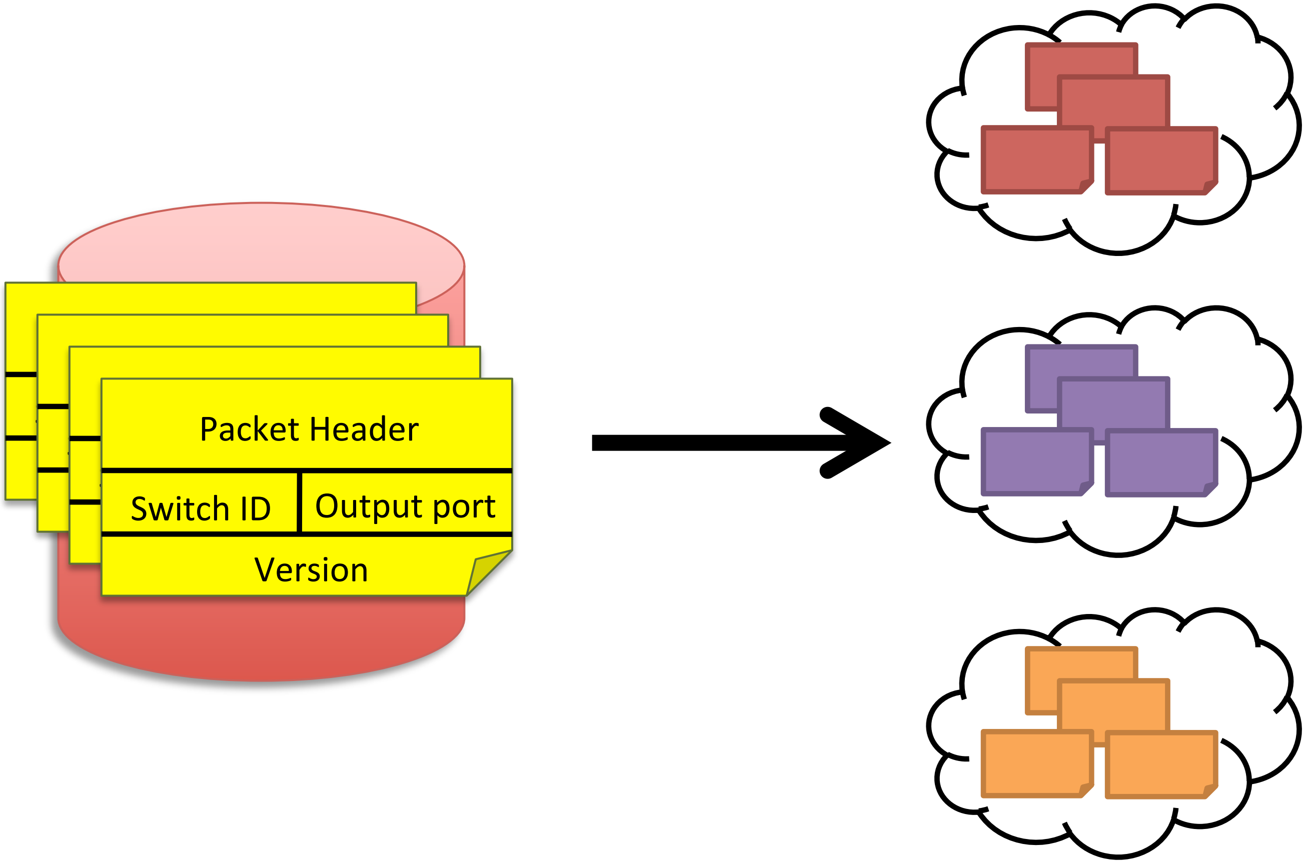

Control Plane
Flow Table State Recorder
Postcard
Collector



Step 1: Generate postcards

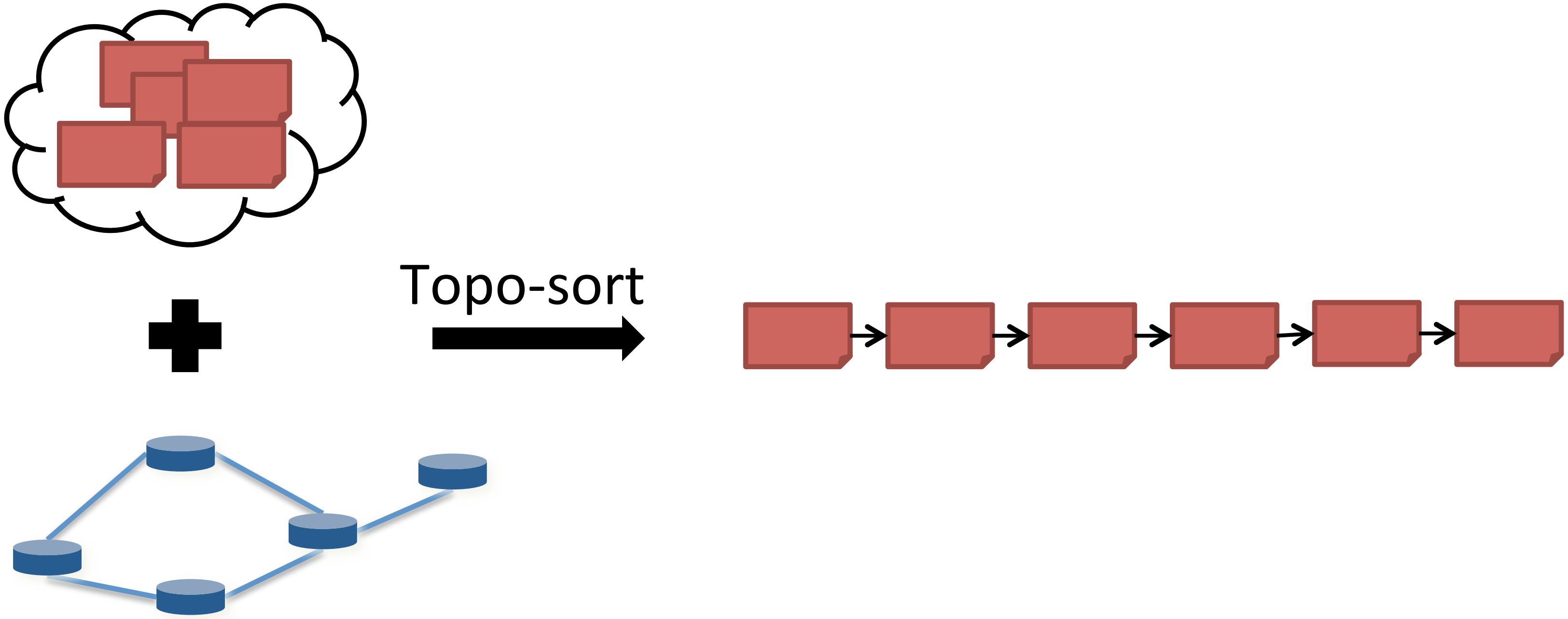
Reconstructing Packet Histories

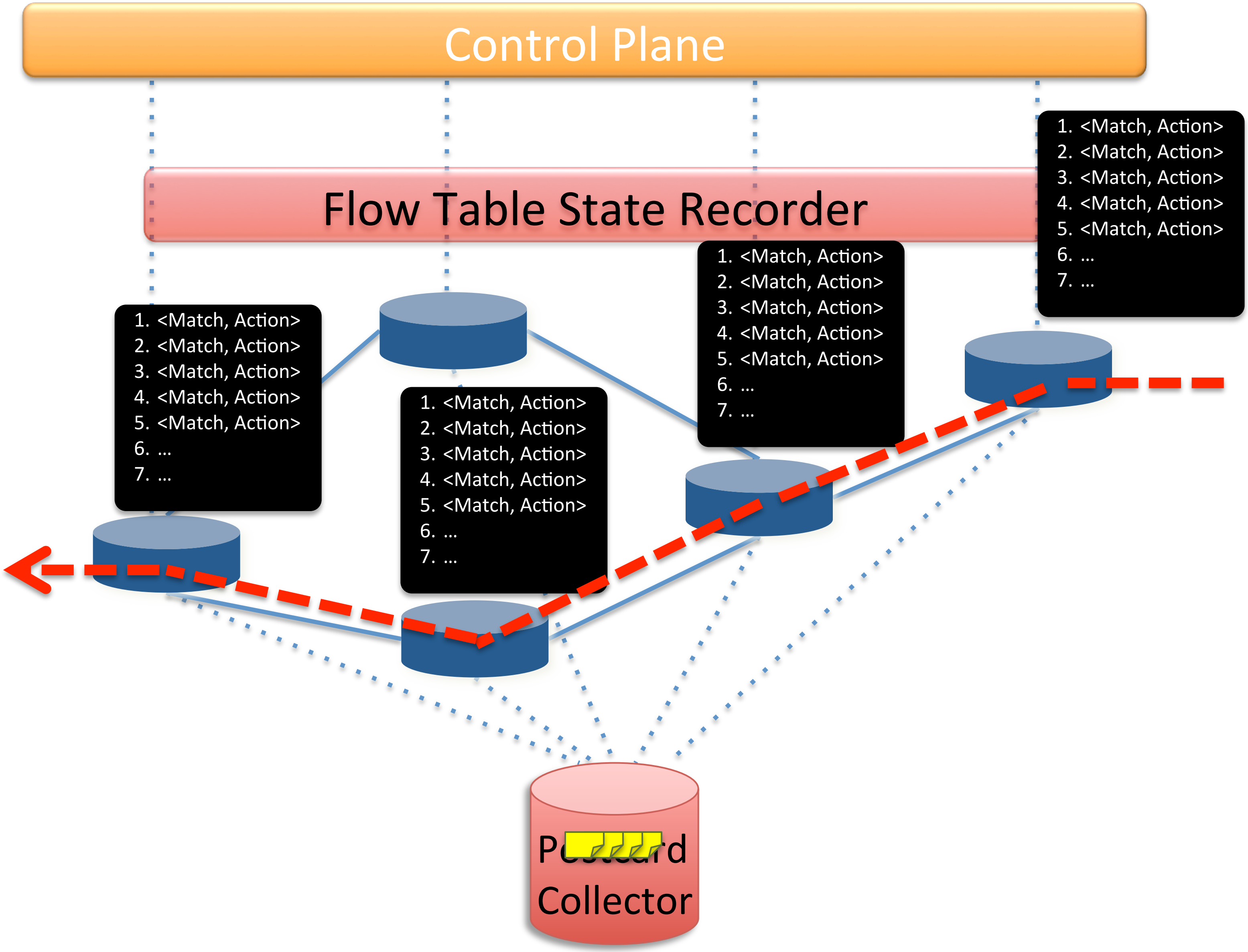
Step 2: Group postcards by generating packet



Reconstructing Packet Histories

Step 3: Sort postcards using topology





Control Plane

Flow Table State Recorder

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. …
7. …

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. …
7. …

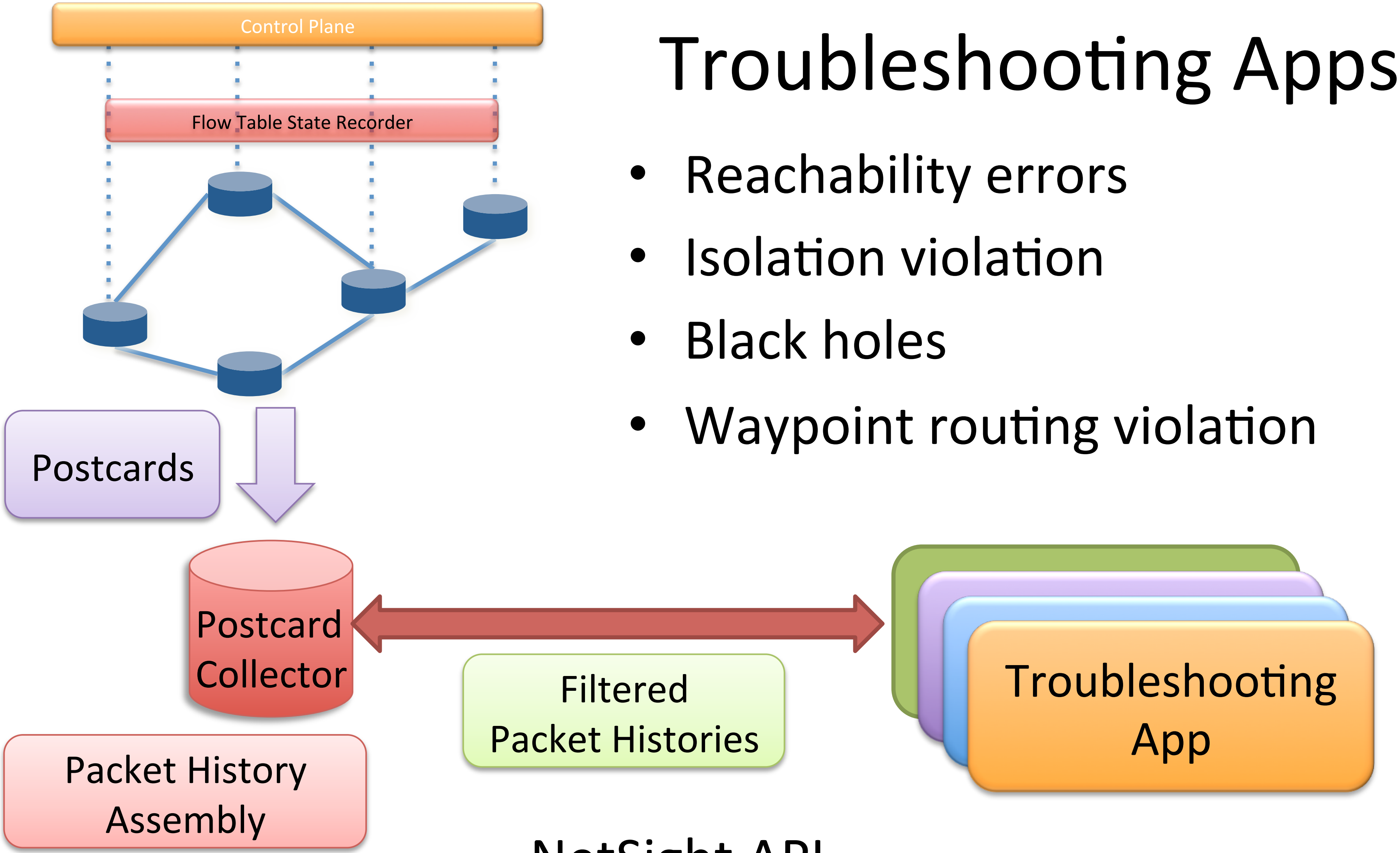
1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. …
7. …

1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. …
7. …

Postcard Collector

Troubleshooting Apps

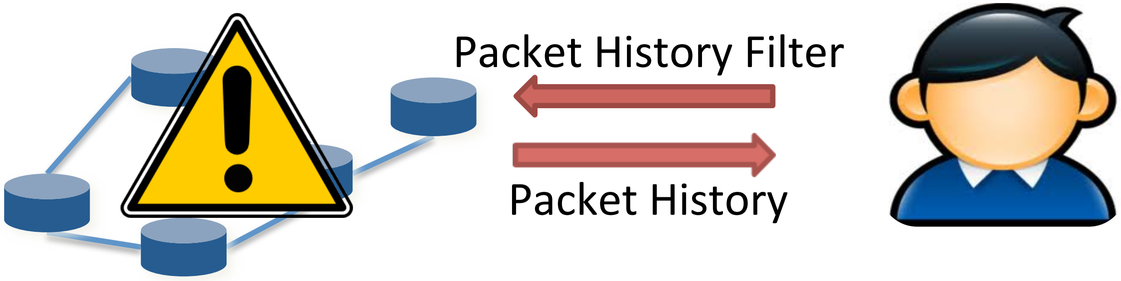
- Reachability errors
- Isolation violation
- Black holes
- Waypoint routing violation



NetSight API

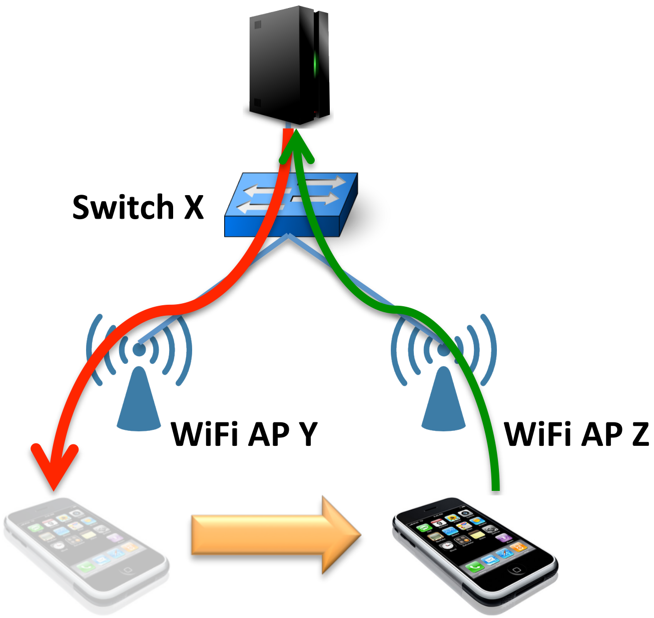
Packet History Filter: A regular-expression-like language to specify packet histories of interest

Bug Story: Incomplete Handover



Packet History Filter

Packet History

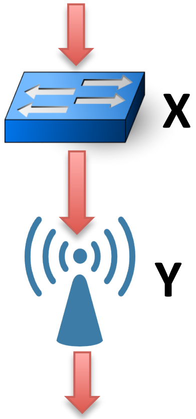


Packet History Filter

```
"Pkts from server not reaching
the client"
```

Packet History

```
Switch X:
  inport: p0,
  outports: [p1]
  mods: [...]
  state version: 3
Switch Y:
  inport p1,
  outports: [p3]
  mods: ...
  …
```



Troubleshooting Apps

ndb:
Interactive network debugger

nprof:
Hierarchical network profiler

netwatch:
Live network invariant monitor

netshark:
Network-wide wireshark

But will it scale?

Why generating postcards for every packet at every hop is crazy!

Network Overhead

- 64 byte-postcard/pkt/hop
- Stanford Network: 5 hops avg, 1031 byte avg pkt
- 31% extra traffic!

Processing Overhead

- Packet history assembly and filtering

Storage Overhead

Why generating postcards for every packet at every hop is ^not crazy!

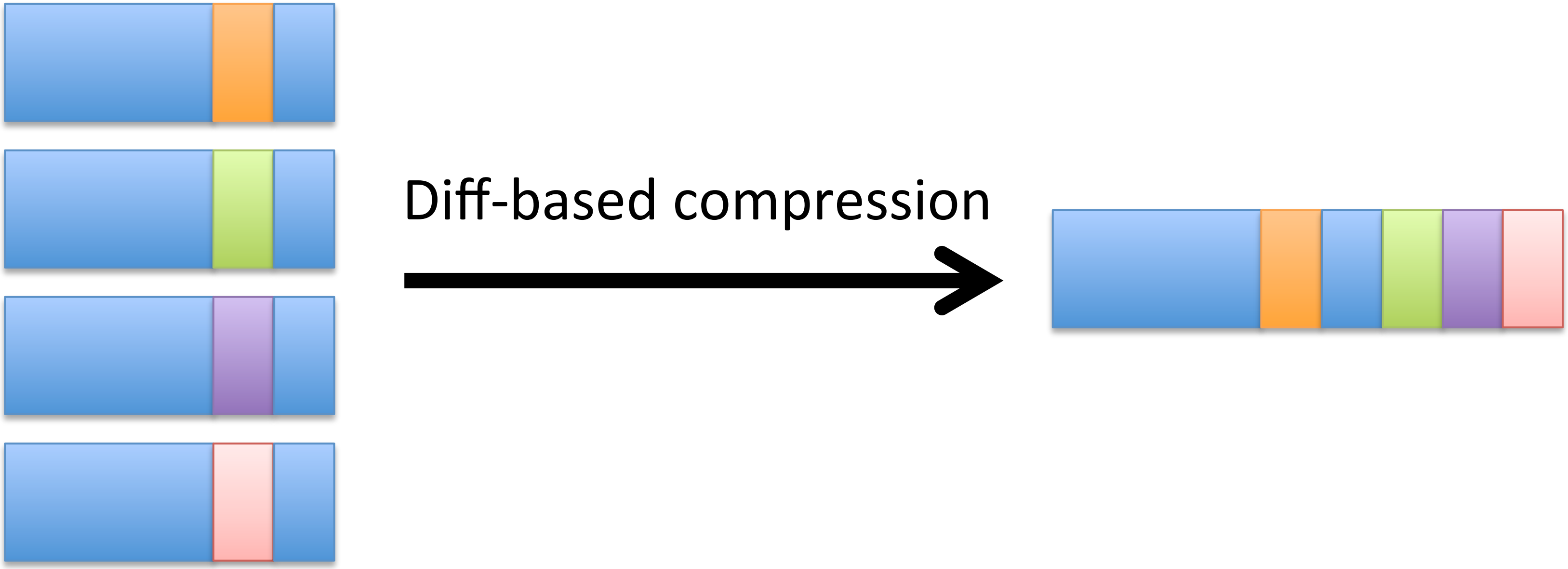
Cost is OK for low-utilization networks

- E.g., test networks, “bring-up phase” networks
- Single server can handle entire Stanford traffic

Why generating postcards for every packet at every hop is not crazy!

Huge redundancy in packet header fields

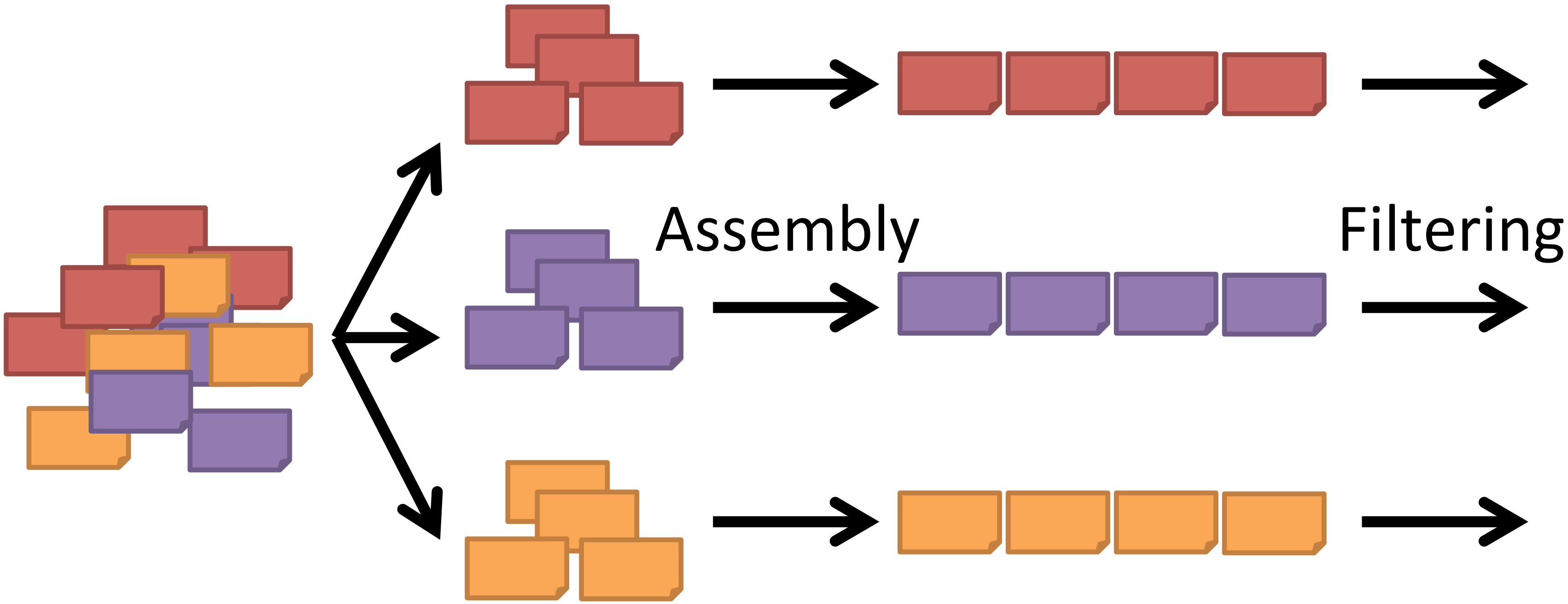
- Only a few fields change – IP ID, TCP seq. no.
- Postcards can be compressed to 10-20 bytes/pkt



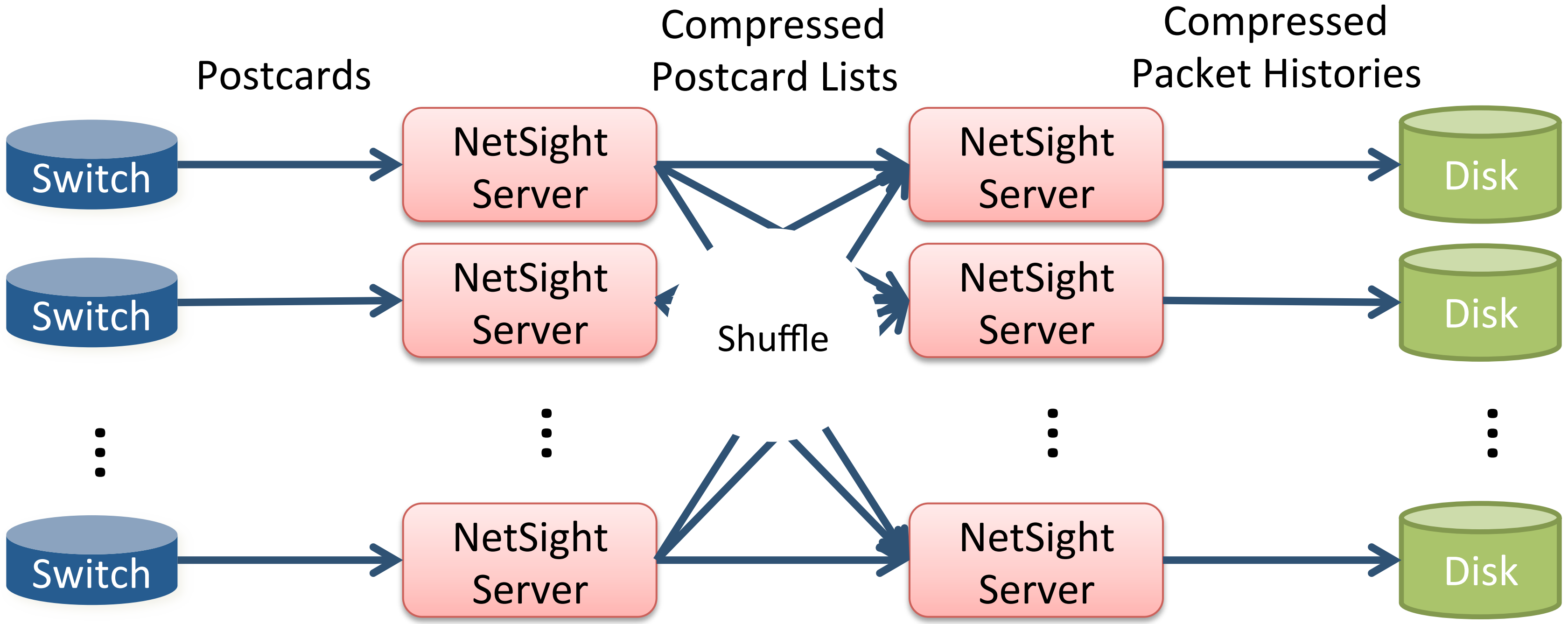
Why generating postcards for every packet at every hop is not crazy!

Postcard processing is embarrassingly parallel

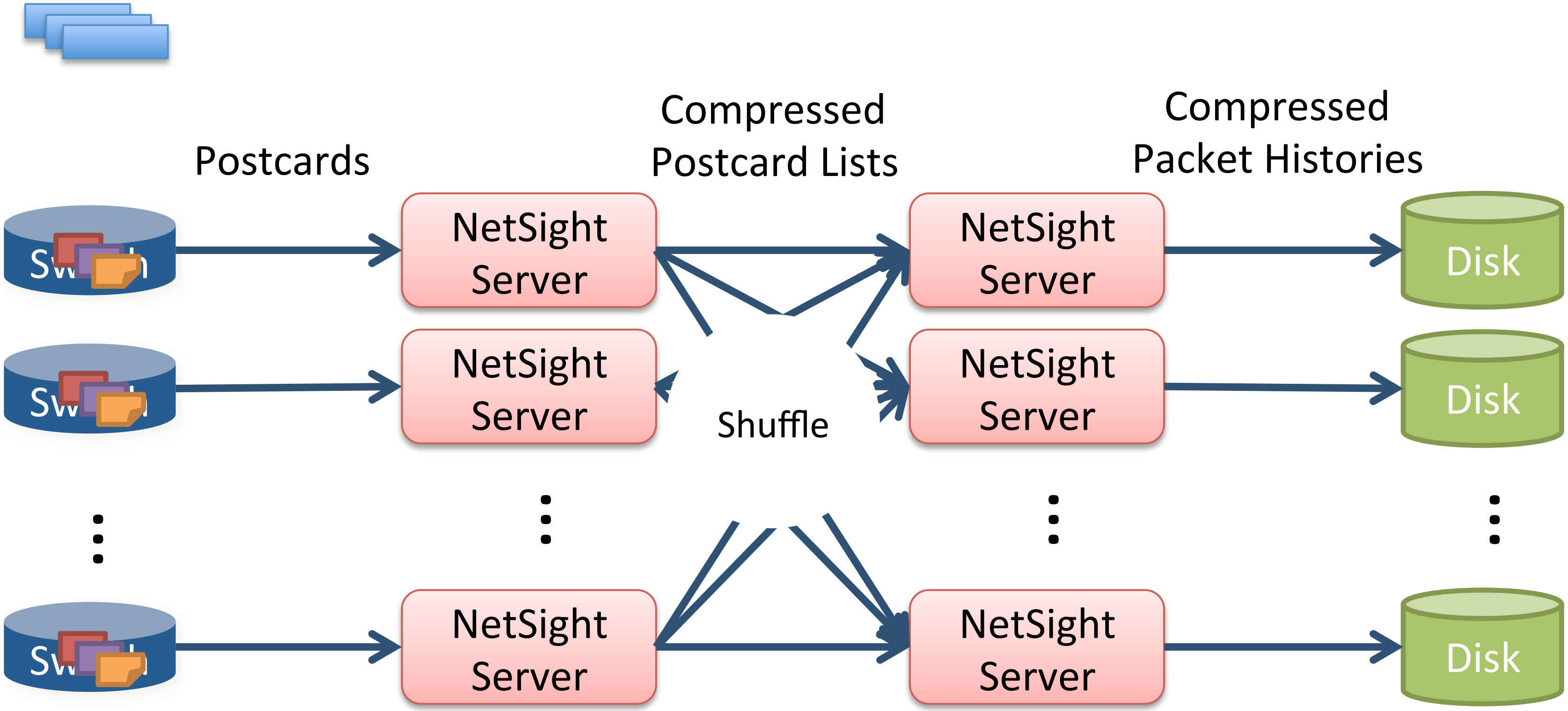
- Each packet history can be processed independent of other packet histories



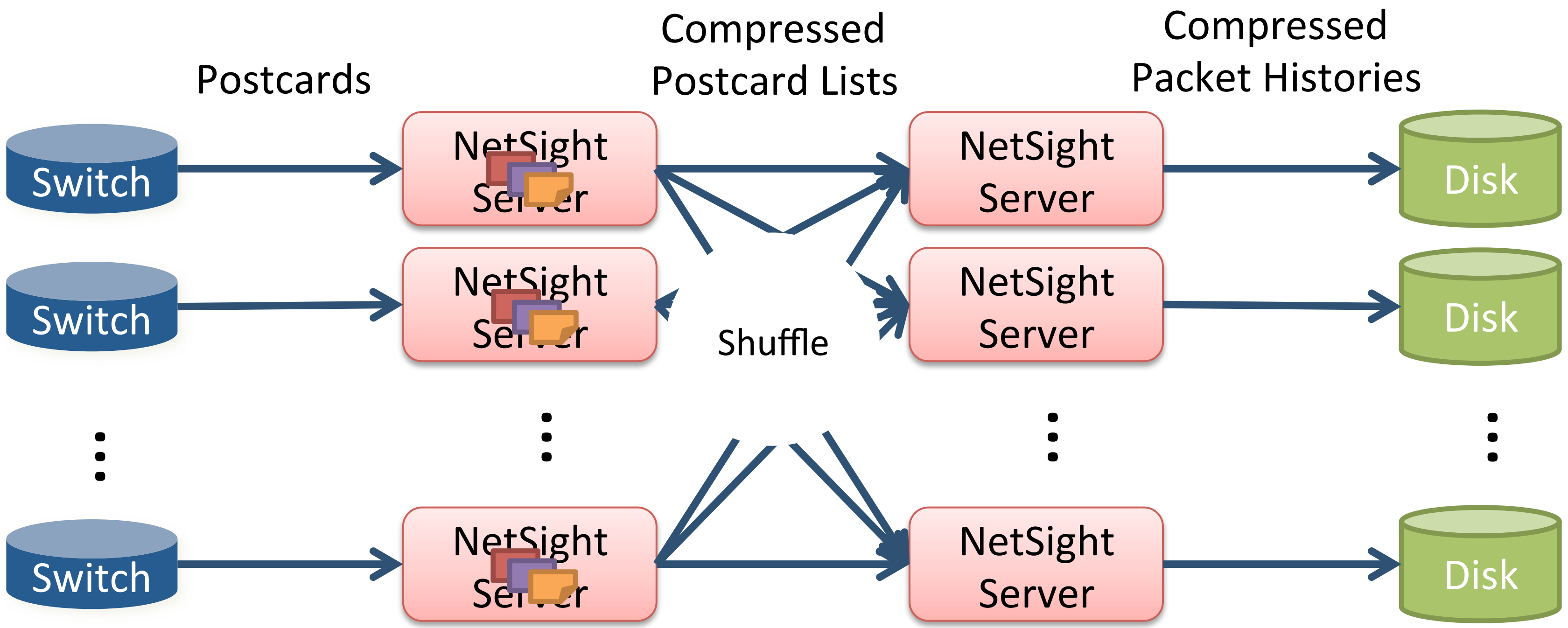
Scaling NetSight Performance



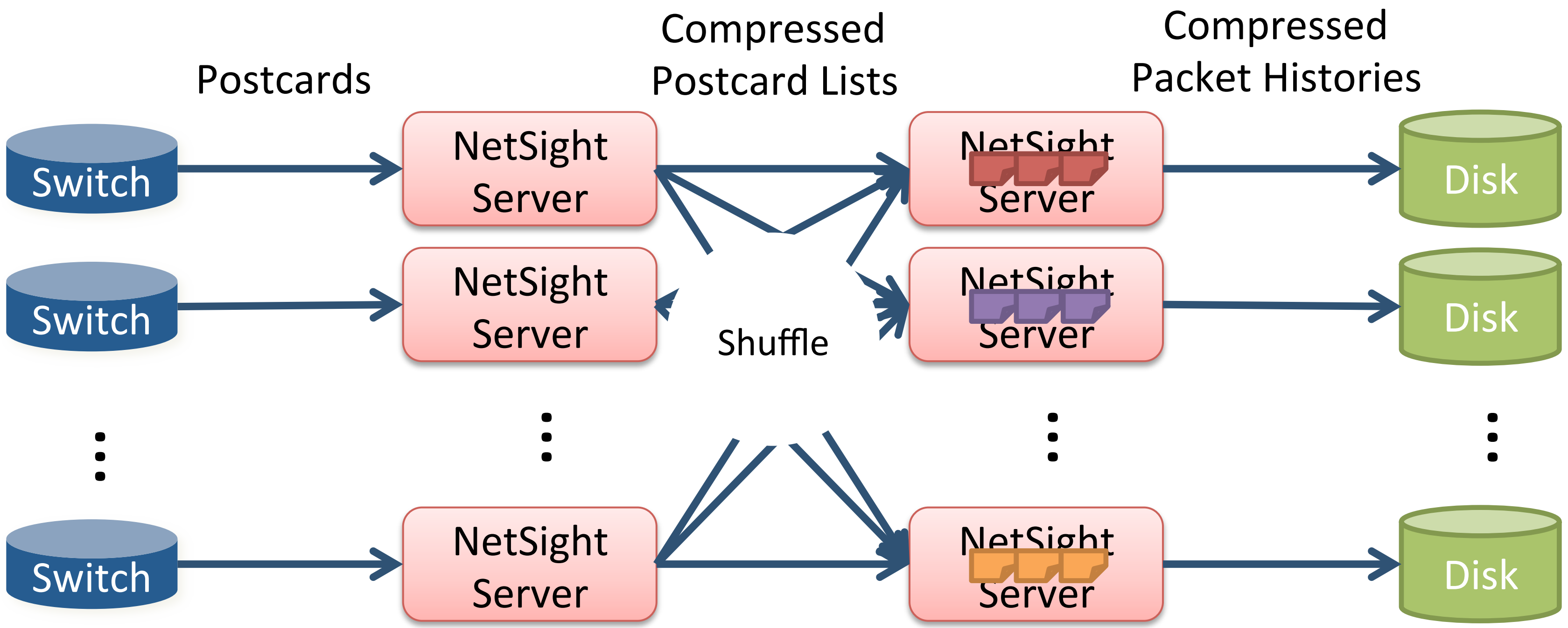
Scaling NetSight Performance



Scaling NetSight Performance



Scaling NetSight Performance



Postcards

Compressed Postcard Lists

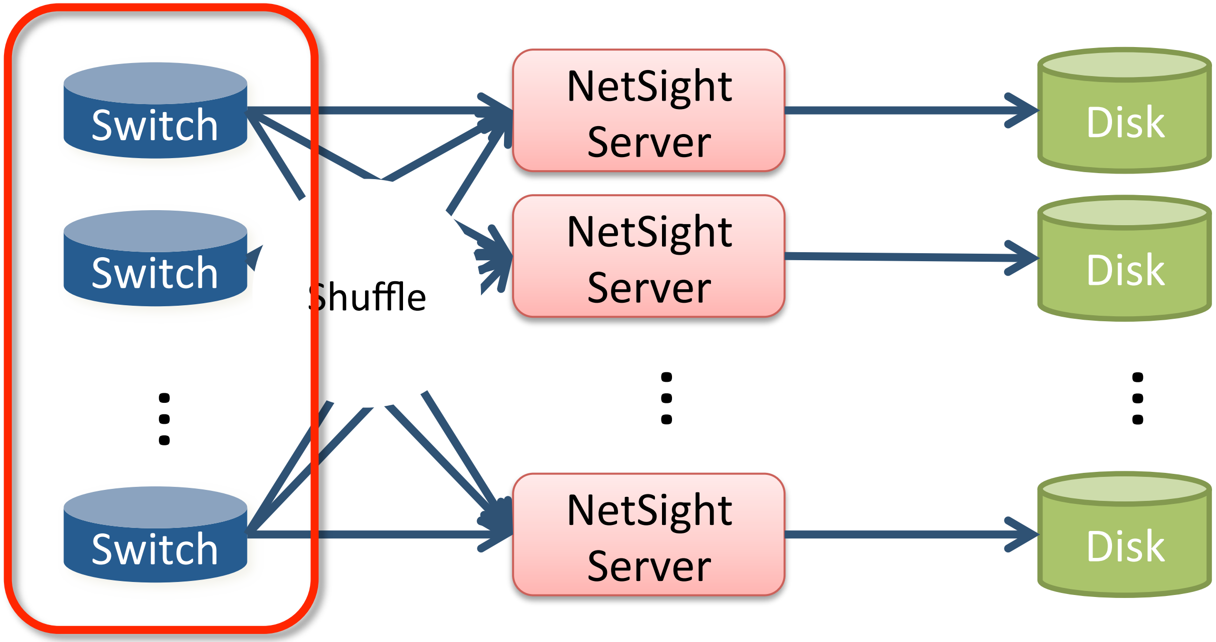
Compressed Packet Histories

Shuffle

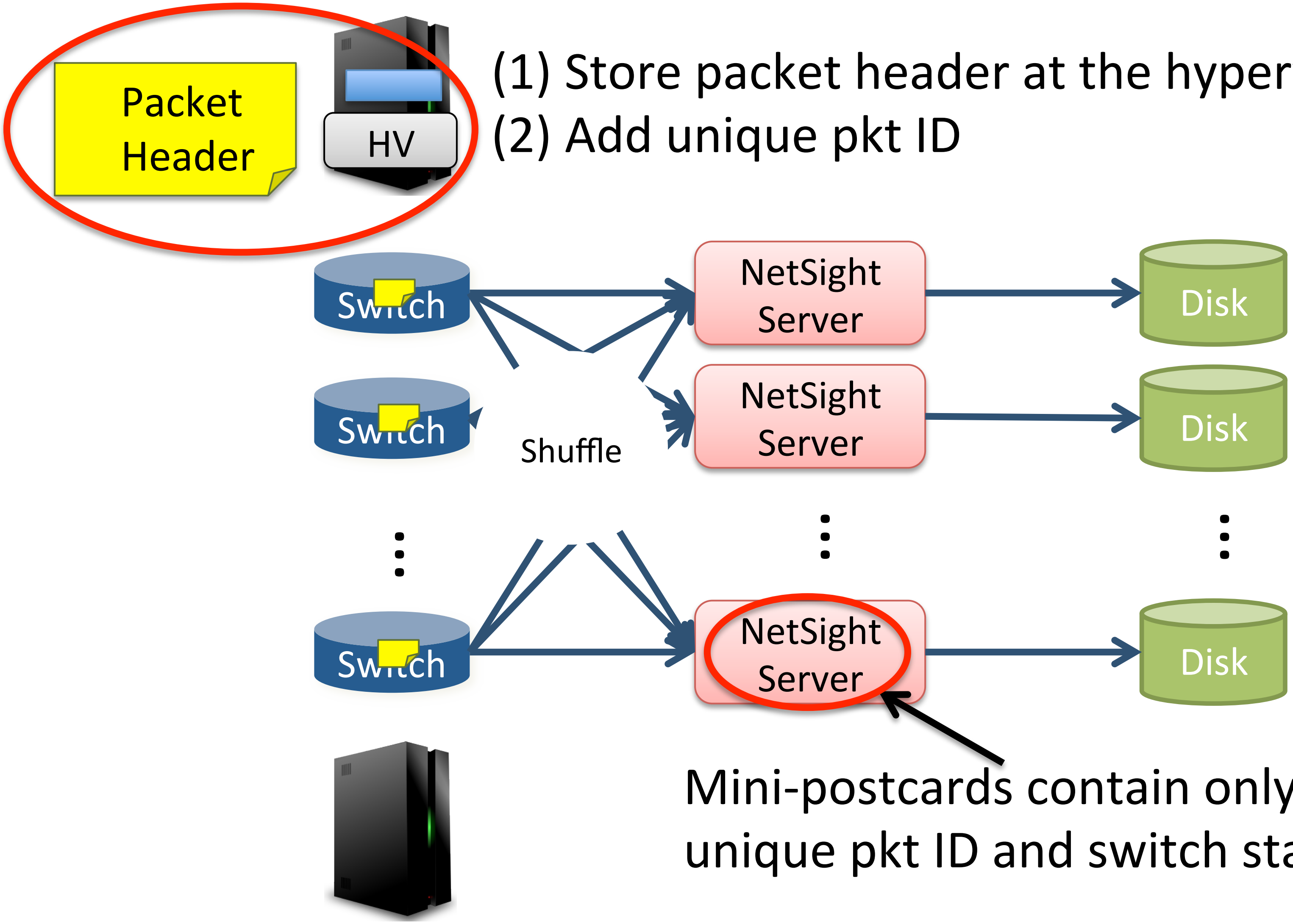
NetSight Variants

NetSight-SwitchAssist moves postcard compression to switches

Move postcard compression to switches with simple hardware mechanisms



NetSight-HostAssist exploits visibility from the hypervisor



(1) Store packet header at the hypervisor
(2) Add unique pkt ID

Mini-postcards contain only unique pkt ID and switch state version

Overhead Reduction in NetSight

Basic (naïve) NetSight : 31% extra traffic in Stanford backbone network

NetSight Switch-Assist: 7%

NetSight Host-Assist: 3%

Takeaways

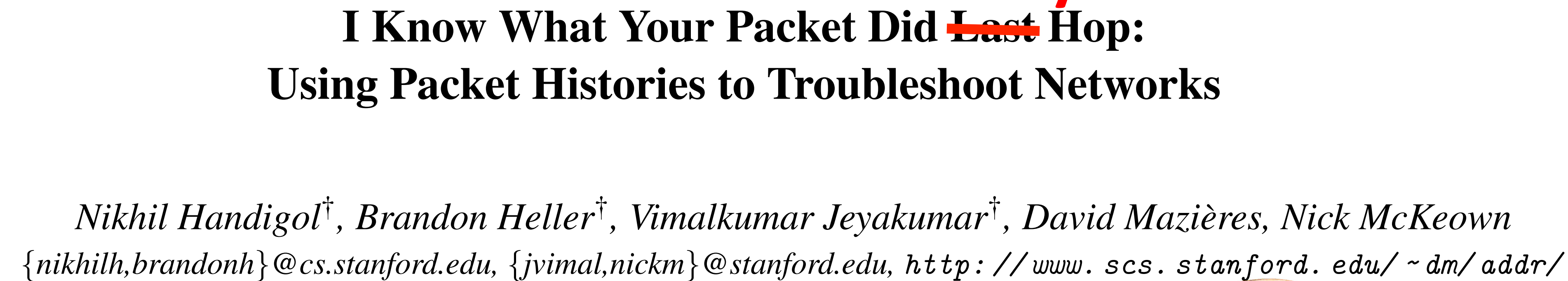
Complete network visibility is possible

- **Packet History**: a powerful troubleshooting abstraction that gives complete visibility
- **NetSight**: a platform to capture and filter packet histories of interest

Complete network visibility is feasible

- It is possible to collect and filter packet histories at scale

Every



I Know What Your Packet Did ~~Last~~ Hop:
Using Packet Histories to Troubleshoot Networks

Nikhil Handigol[†], *Brandon Heller*[†], *Vimalkumar Jeyakumar*[†], *David Mazières, Nick McKeown*
{*nikhilh,brandonh*}@*cs.stanford.edu,* {*jvimal,nickm*}@*stanford.edu,* `http://www.scs.stanford.edu/~dm/addr/`

NetSight API

Abstract

The complexity of networks has outpaced our tools to debug them; today, administrators use manual tools to diagnose problems. In this paper, we show how *packet histories*—the full stories of every

1. “Host A cannot talk to … here packets from A intend … ng with any header modi…

2. “I don’t want forward … vork, even transient ones. Sl … et that passes the same switch

http://yuba.stanford.edu/netsight