Cornell University

Software-Defined Networks

"Data networks you want to use"

Zhiyuan Teo
Cornell University

*some slides adapted from my 'A' exam

Administrative announcements

- Friday's office hours moved to 11am.
- HW2 is due 23:59 Friday, 2 Oct.
- Materials for this lecture are examinable.

Topics for today

- Introduction to SDNs.
- What is OpenFlow?
- OpenFlow rules and how they work.
- Practical applications of OpenFlow.
- Research applications.

What is SDN and why the fuss?

- In the past: network switches driven by firmware.
- Now: use software to control network switches.
- This software is known as a controller.
- "Provides separation between control plane and data plane".
- What's the big deal?

"Separate control plane from data plane"

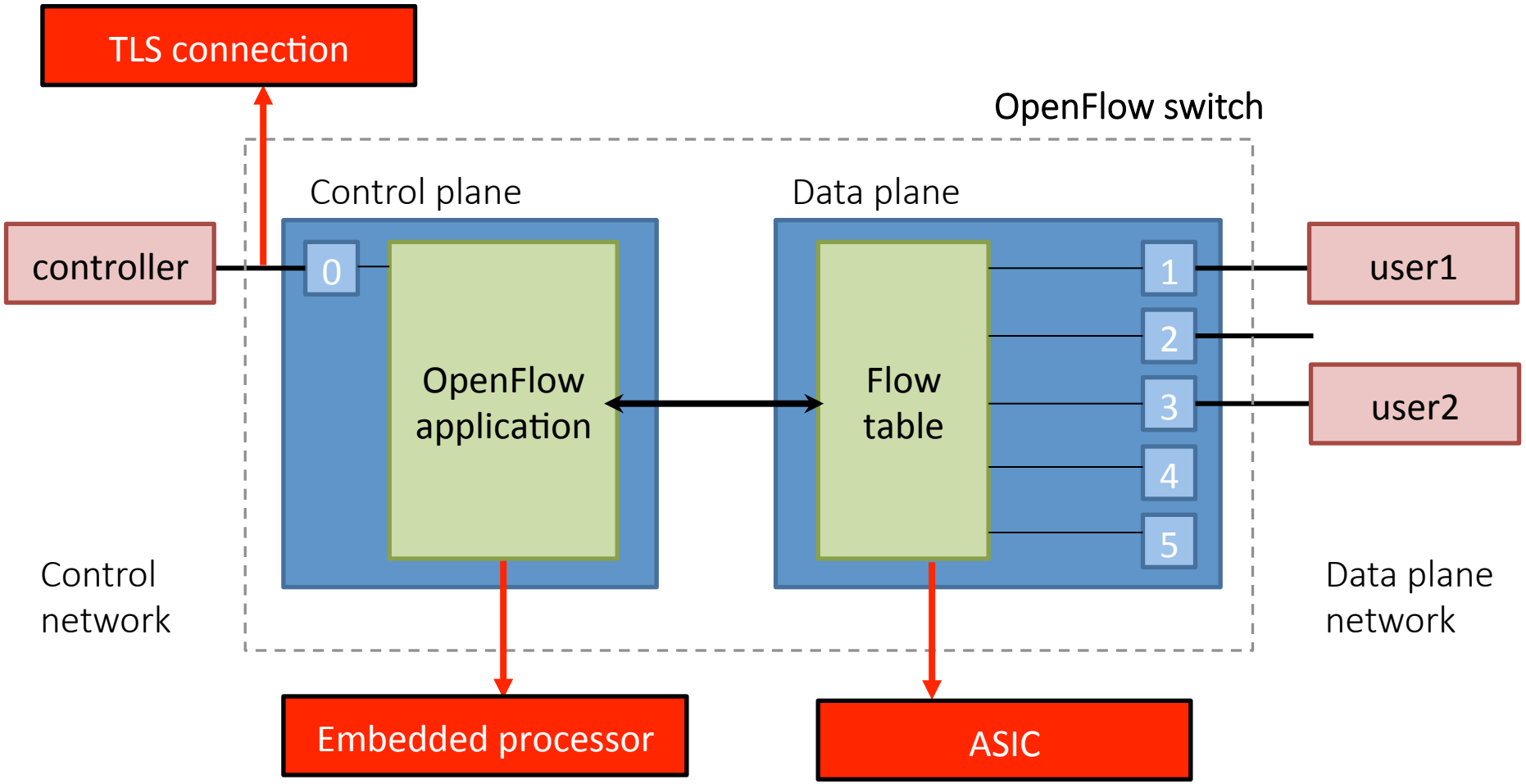
- Decouple switching decisions from the high speed fabric used to move data.
- Controller provides the brain, switch provides the brawn (to execute tasks quickly).

OpenFlow

- A protocol for a controller to talk to a switch.
- Uses TLS/TCP to exchange data.
- Many revisions to the standard. Current version is 1.4.

- OpenFlow is not SDN! It's one way to do SDN.

Control plane and data plane

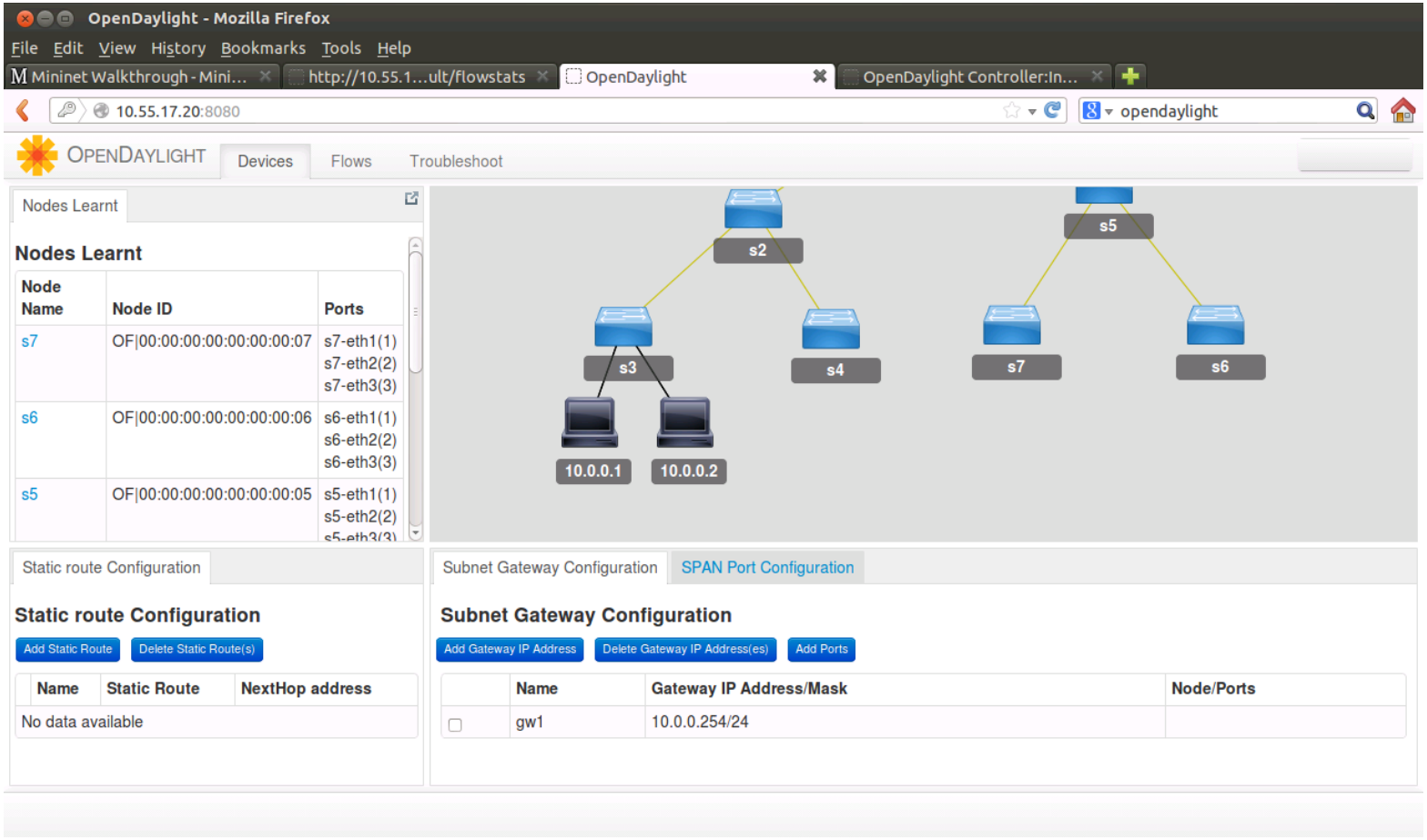


Noteworthy things

- The control network is logically separate from the data plane network. Why?
- OpenFlow application transforms controller requests into ASIC instructions.
- ASIC performs the high speed work.
- Similar to how CPUs instruct GPUs to accelerate graphics in your computer.

OpenFlow Controllers

- Many different vendors available.
- Open source: OpenDaylight, POX, Floodlight, etc.
- Proprietary: NEC PF6800, Big Cloud Fabric, etc.
- Logically centralized. All switches connect to one controller.


OpenDaylight - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Mininet Walkthrough - Mini...
http://10.55.1...ult/flowstats
OpenDaylight
OpenDaylight Controller:In...
10.55.17.20:8080
opendaylight
OPENDAYLIGHT
Devices
Flows
Troubleshoot
Nodes Learnt
Nodes Learnt
Node Name
Node ID
Ports
s7
OF|00:00:00:00:00:00:00:07
s7-eth1(1)
s7-eth2(2)
s7-eth3(3)
s6
OF|00:00:00:00:00:00:00:06
s6-eth1(1)
s6-eth2(2)
s6-eth3(3)
s5
OF|00:00:00:00:00:00:00:05
s5-eth1(1)
s5-eth2(2)
s2
s3
s4
s5
s7
s6
10.0.0.1
10.0.0.2
Static route Configuration
Static route Configuration
Add Static Route
Delete Static Route(s)
Name
Static Route
NextHop address
No data available
Subnet Gateway Configuration
SPAN Port Configuration
Subnet Gateway Configuration
Add Gateway IP Address
Delete Gateway IP Address(es)
Add Ports
Name
Gateway IP Address/Mask
Node/Ports
gw1
10.0.0.254/24

What can you do in OpenFlow?

- Create/modify/delete flows.
- Set up QoS.
- Inspect/inject packets on/to data plane.
- Look at flow statistics.
- Turn on/off ports.

- … it's pretty comprehensive.

What is a flow?

- Match criteria.
- Action set.
- Priority.
- Timeout settings.

- Acts like a filter. Packets matching the filter are grouped into the same flow.
- Flows are held in flow tables on a switch.

Match criteria

- 12-tuple.
- Ingress port.
- Ethernet source/destination.
- EtherType.
- VLAN ID/priority.
- IP src/dest/ToS.
- TCP/UDP src/dest port.

*IP addresses can be specified with suffix wildcards. Eg. 10.1.2.3/24

Priority rules

- What if multiple rules match the packet?
- "Most specific rule" has highest priority.
- Tiebreaker: priority value of the flow.

```
Criteria: in_port = 1
Action  : drop
```

```
Criteria: in_port  = 1
          eth_dst  = de:ad:be:ef:00:01
Action  : out_port = 2
```

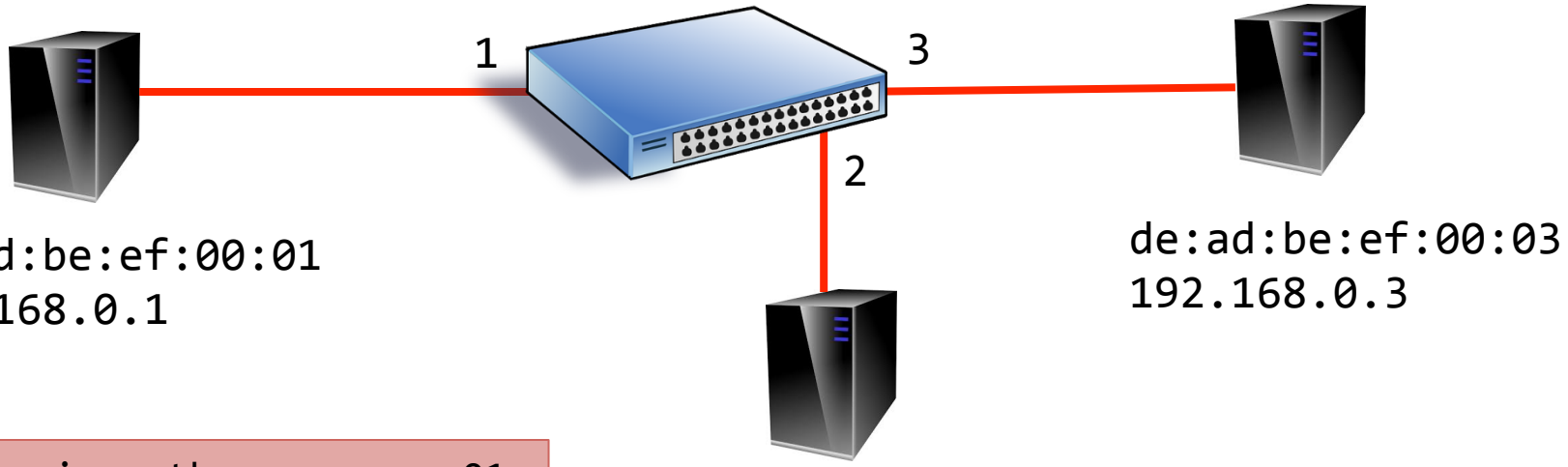
Flow actions

- Drop packet.
- Forward to port or ports. Forward to controller.
- Flood.
- Modify Ethernet src/dest.
- Modify IP src/dest.
- Modify TCP/UDP port.
- … etc.

Reactive controllers

- Switch connects to the controller.
- Packets enter the switch.
- On a flow criteria match, perform flow action.
- Otherwise, this is a flow-miss. Forward to the controller for decision-making.
- Controller can install a flow in response.

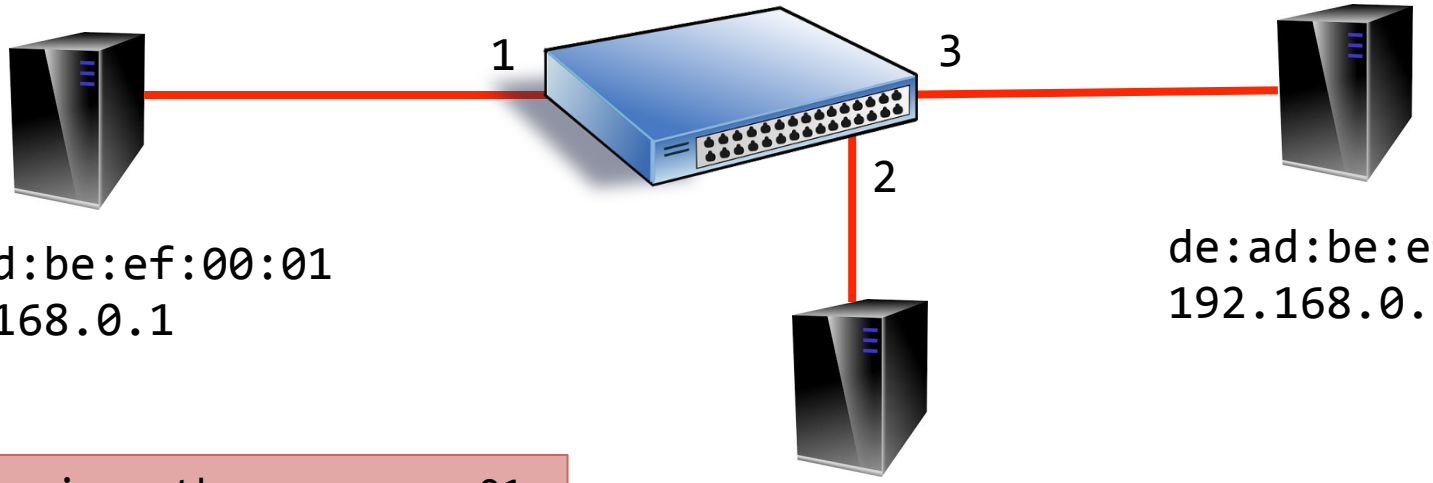
Example 1




```
Criteria: eth_src = ...:01
          eth_dst = ...:03
Action  : out_port = 3
```

src/dest forwarding

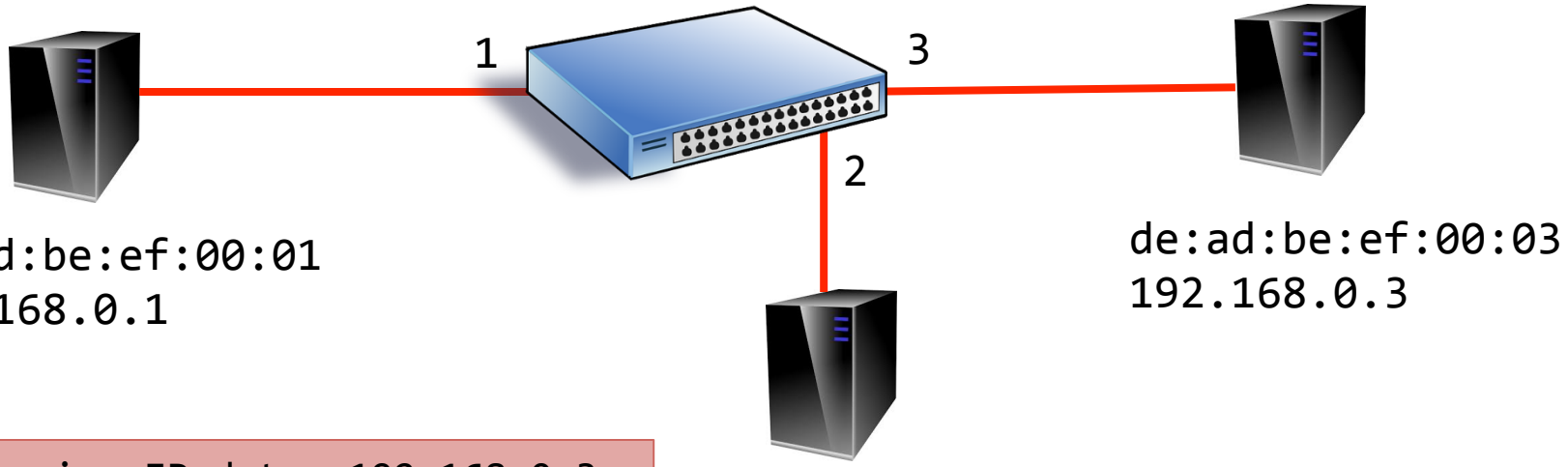
Example 2




```
Criteria: eth_src = ...:01
          eth_dst = ...:03
Action  : out_port = 2
          out_port = 3
```

Forwarding with snooping?

Example 3



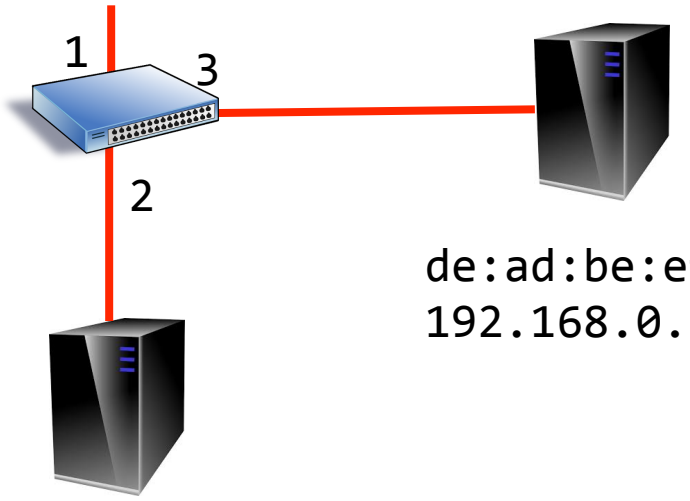

```
Criteria: IP_dst = 192.168.0.3
          TCP_dst_port = 80
Action  : drop
```

Blacklisting HTTP server on specific IP

Example 4

```
Criteria: in_port = 1
          IP_dst = 192.168.0.10
          TCP_dst_port = 80
Action  : set IP_dst = 192.168.0.2
          out_port = 2
```

```
Criteria: in_port = 2
          IP_dst = 192.168.0.10
          TCP_dst_port = 80
Action  : set IP_dst = 192.168.0.3
          out_port = 3
```



HTTP load balancing.

* Are these rules enough?

Firewall

- Flow rules are essentially packet filters.
- Can you write a firewall with OpenFlow rules?
- Some examples to think about:

```
Block all access to 192.168.0.1
```

```
Disallow all HTTP traffic across the switch
```

```
Permit SSH access to 192.168.0.1 only if you
are in the 192.168.0.1/24 subnet
```

```
Redirect all HTTP traffic to 192.168.0.1
```

Limitations of OpenFlow

- OpenFlow uses packet header matching.
- Modification is confined to packet headers.

- Can you use OpenFlow to inspect packet payloads?

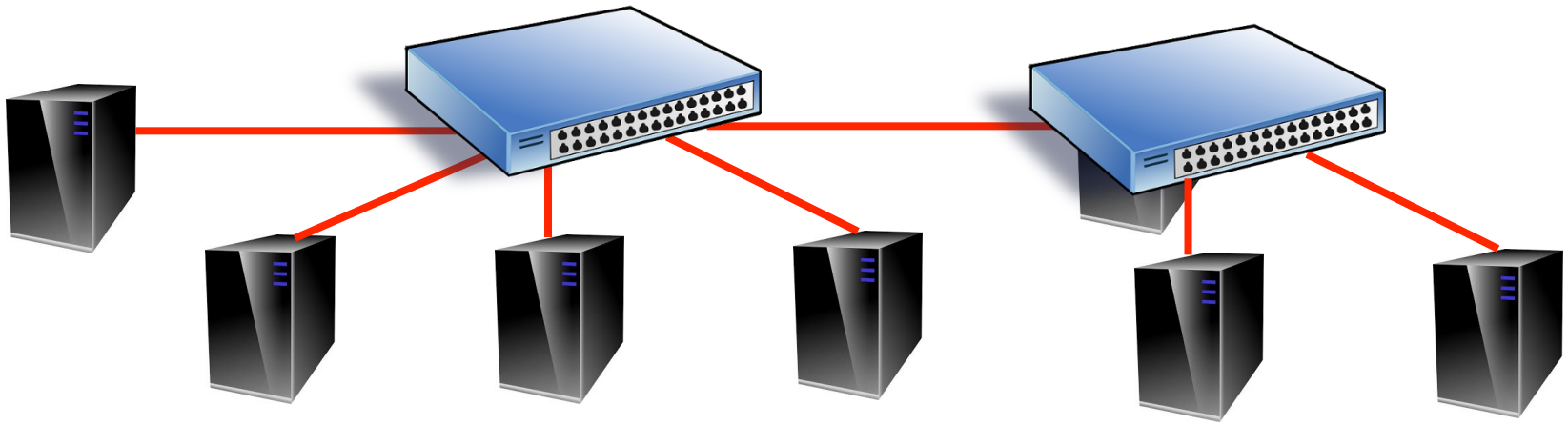
Research applications

Pet peeves with today's networks

- Slow.
- Unreliable.
- Not secure.

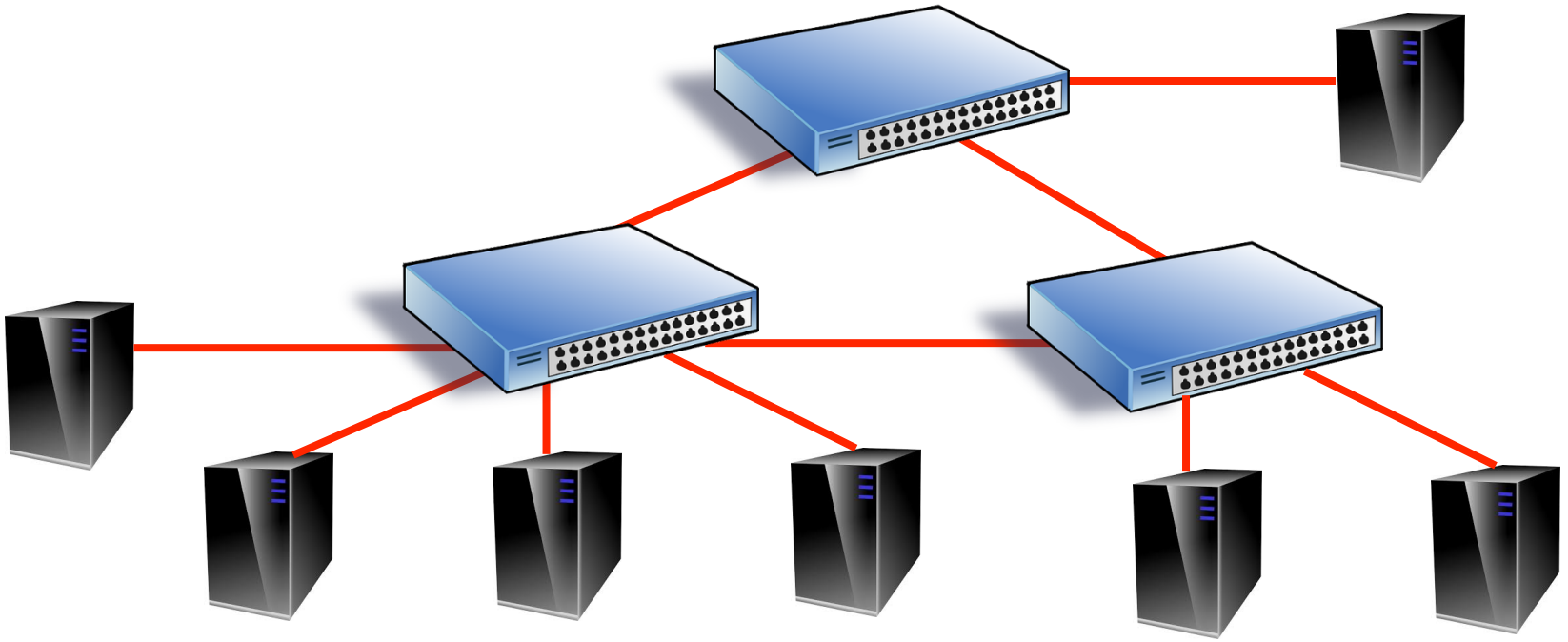
Background

- Let's consider topology in Ethernet networks.
- To expand network segment, add more switches.



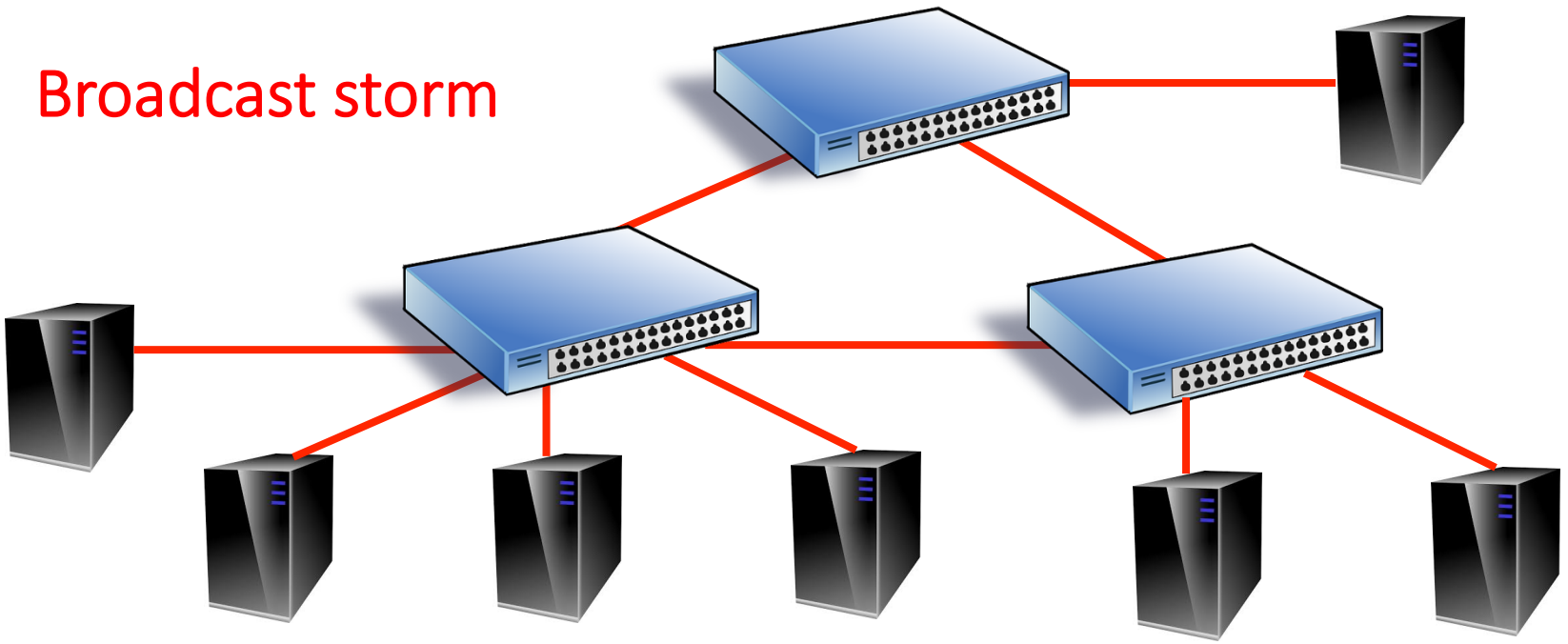
Ethernet topology

- Can we do this arbitrarily?



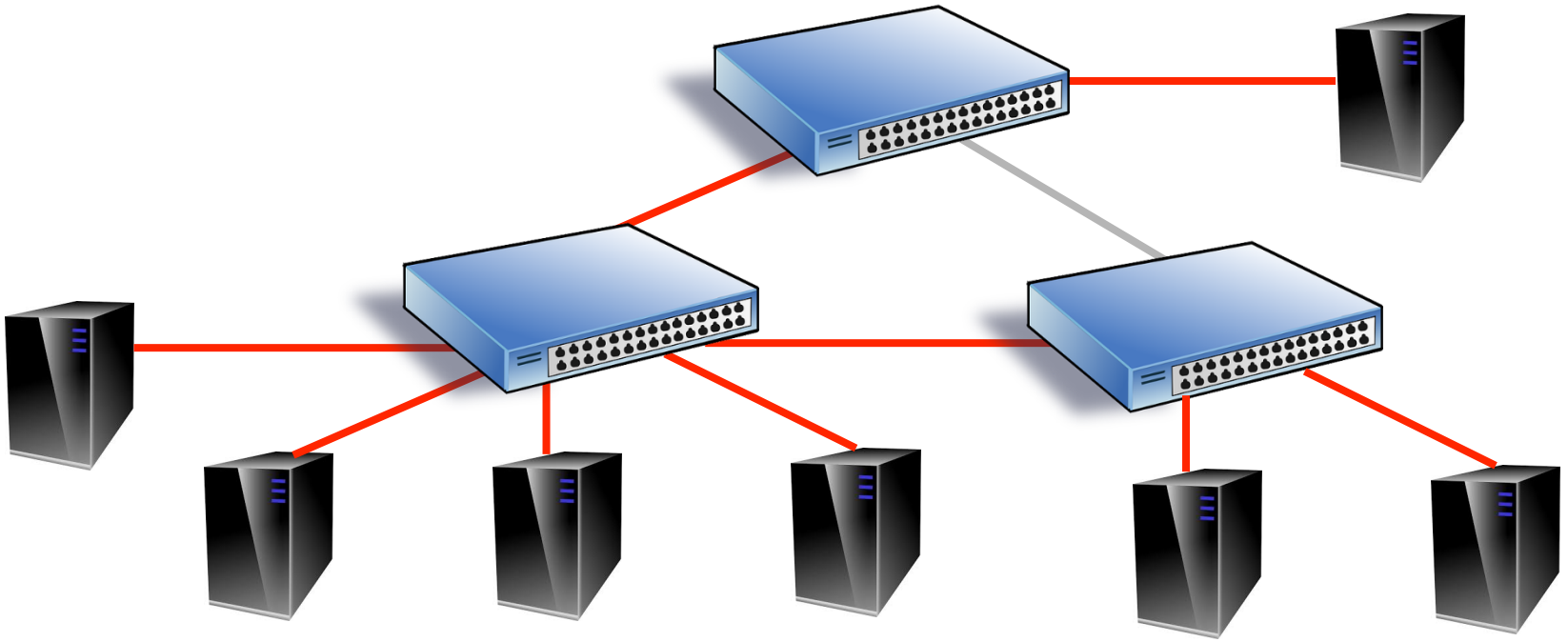
Network loop

- Consider what happens when a broadcast packet enters the loop.



Spanning Tree Protocol

- Switches figure out the topology and shut off links that create cycles.



Implications of spanning tree

1. Spanning tree links are potential bottlenecks.
2. Single source-destination path.
3. Long recovery times on tree breakage.
4. Data travels over predictable paths.

affects performance

affects reliability

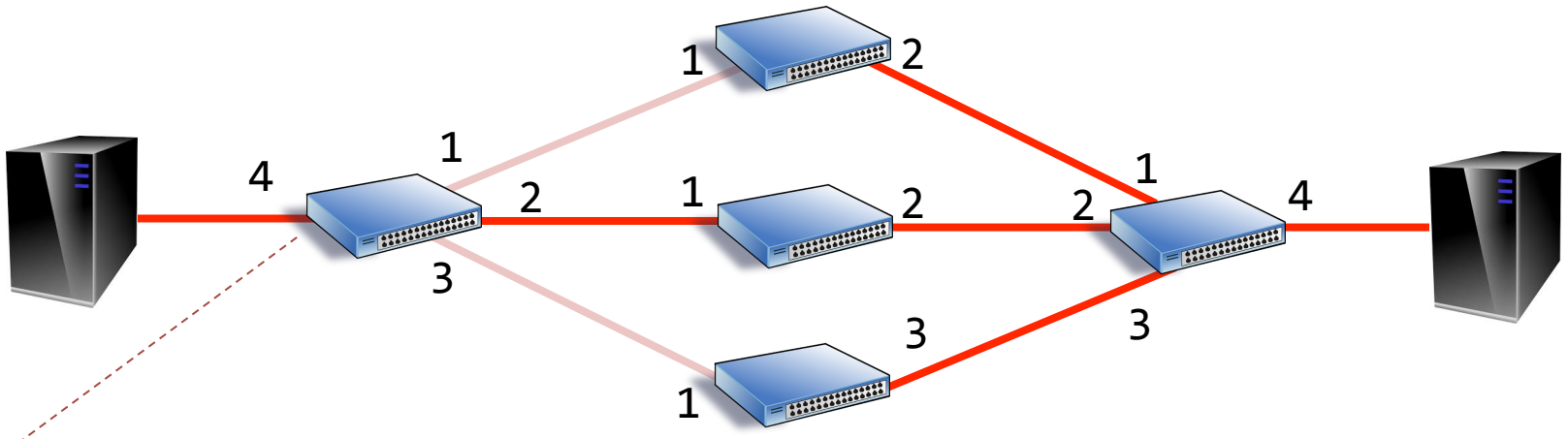
affects security

Use multipath forwarding

How would you do that with OpenFlow?

1. Use spanning tree for ordinary operations.
2. Use flow rules to create disjoint paths for source-destination pairs.

Multipath forwarding




```
Criteria: eth_dest = aa:aa:aa:aa:aa:aa
Action  : out_port = 1
```

```
Criteria: eth_dest = bb:bb:bb:bb:bb:bb
Action  : out_port = 2
```

```
Criteria: eth_dest = cc:cc:cc:cc:cc:cc
Action  : out_port = 3
```

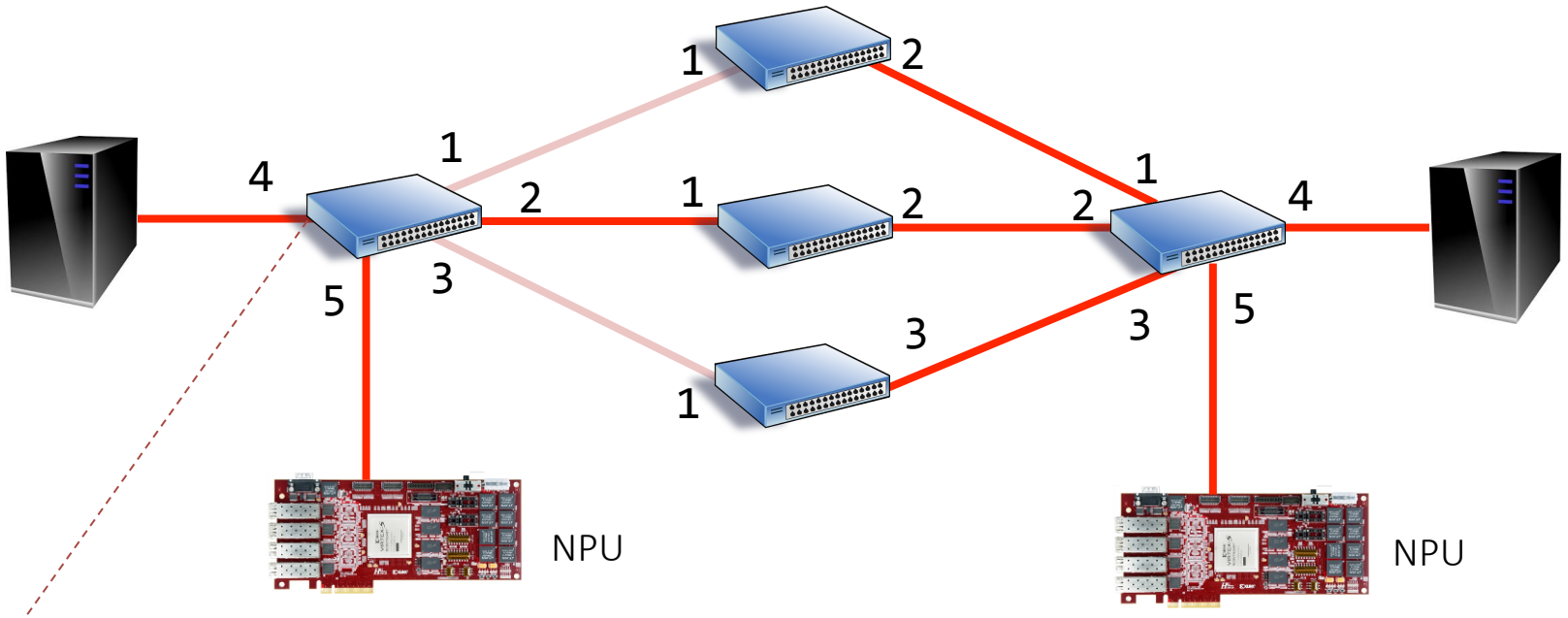
How to make a single flow take multiple paths?

Glossary: Network Processing Units

Xilinx NetFPGA 10G cards. 4x 10Gbps ports.

- Custom FPGA-based packet processor.
- Modifies packets at line rate.

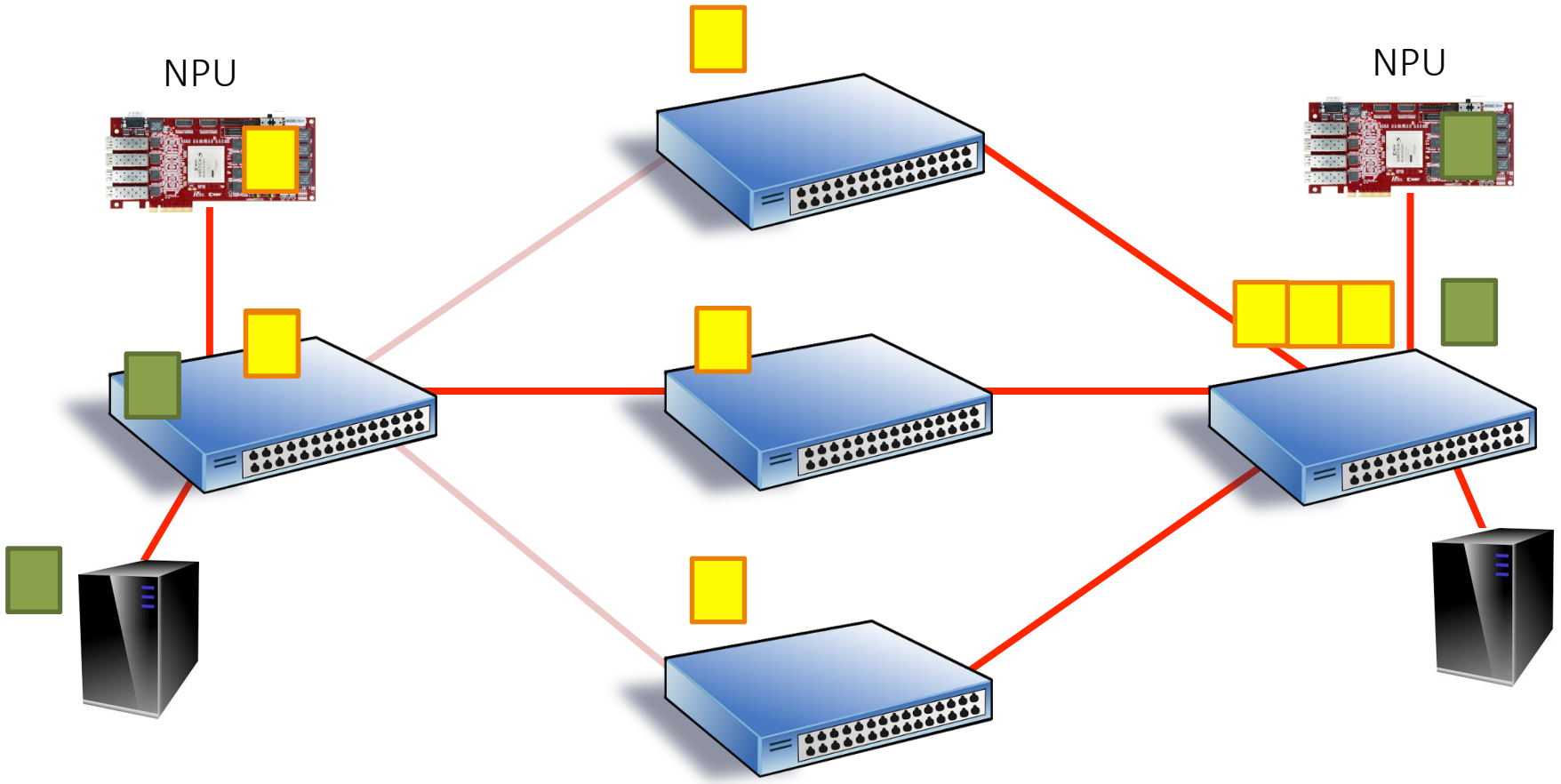
Multipath forwarding




```
Criteria: in_port = 4
Action  : out_port = 5
```

NPU rewrites Ethernet dest of packet to select a path.

High level idea

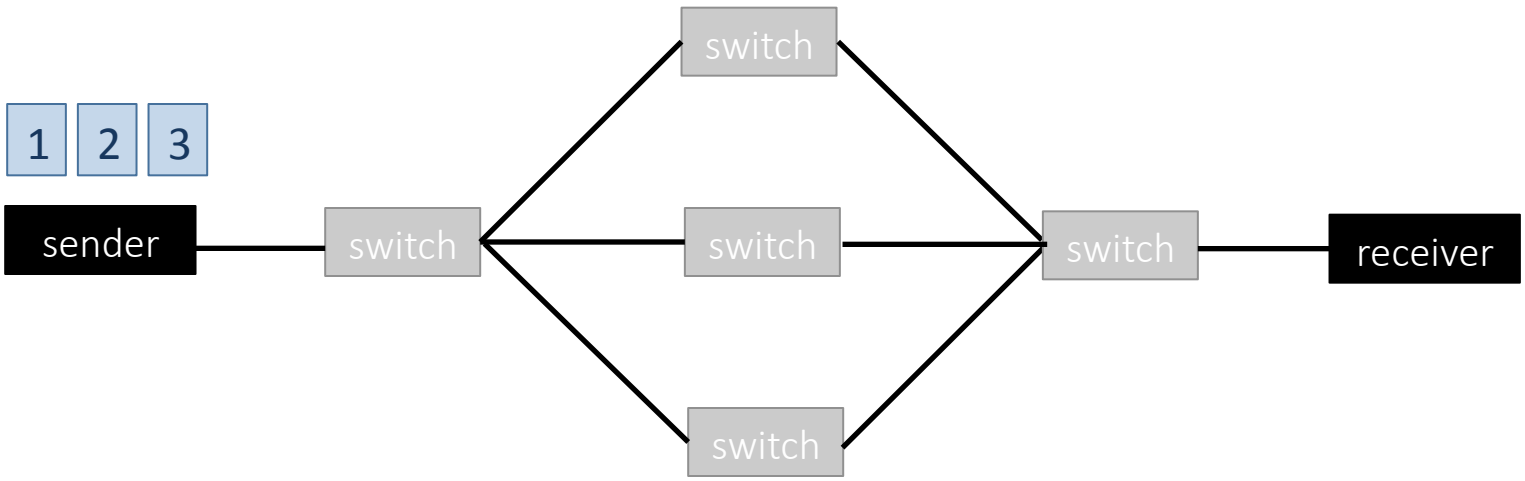


What can you do with this?

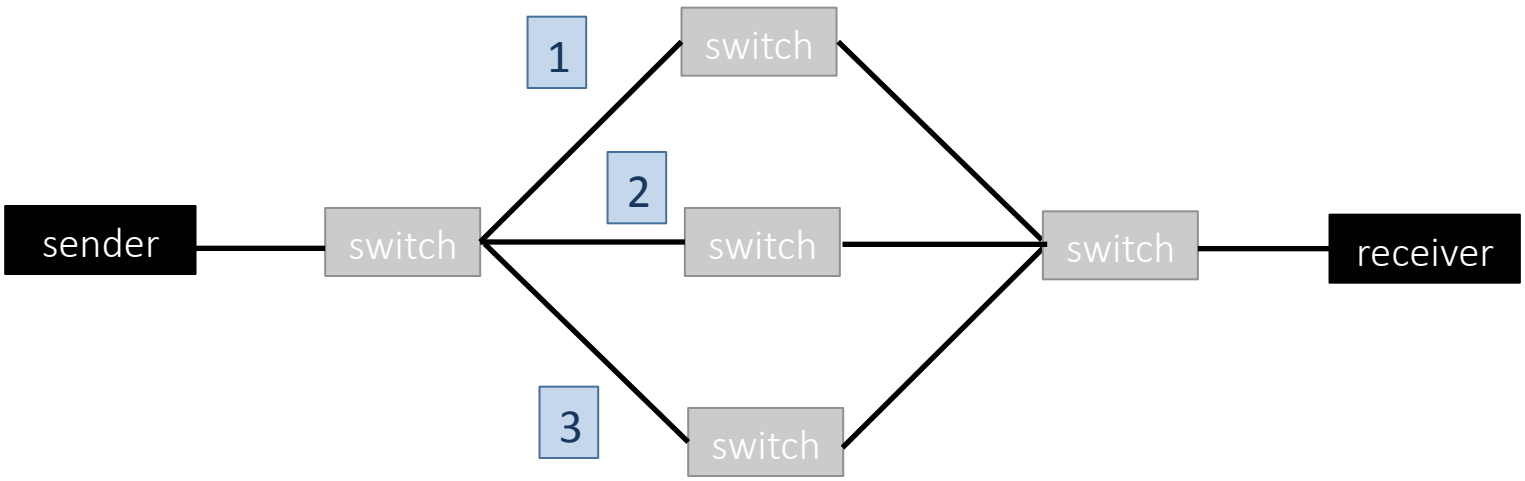
- Run it like disk RAID system to get performance and reliability.
- RAID 0
- RAID 1
- RAID 4

Redundant Array of Independent Links (RAIL)

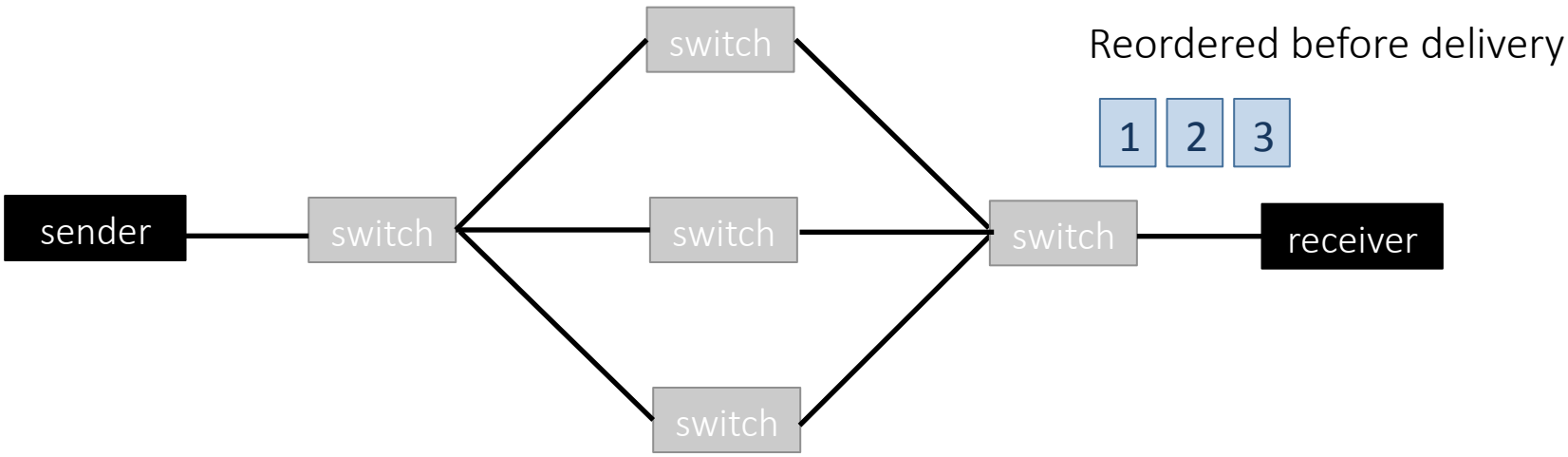
Performance: RAIL 0



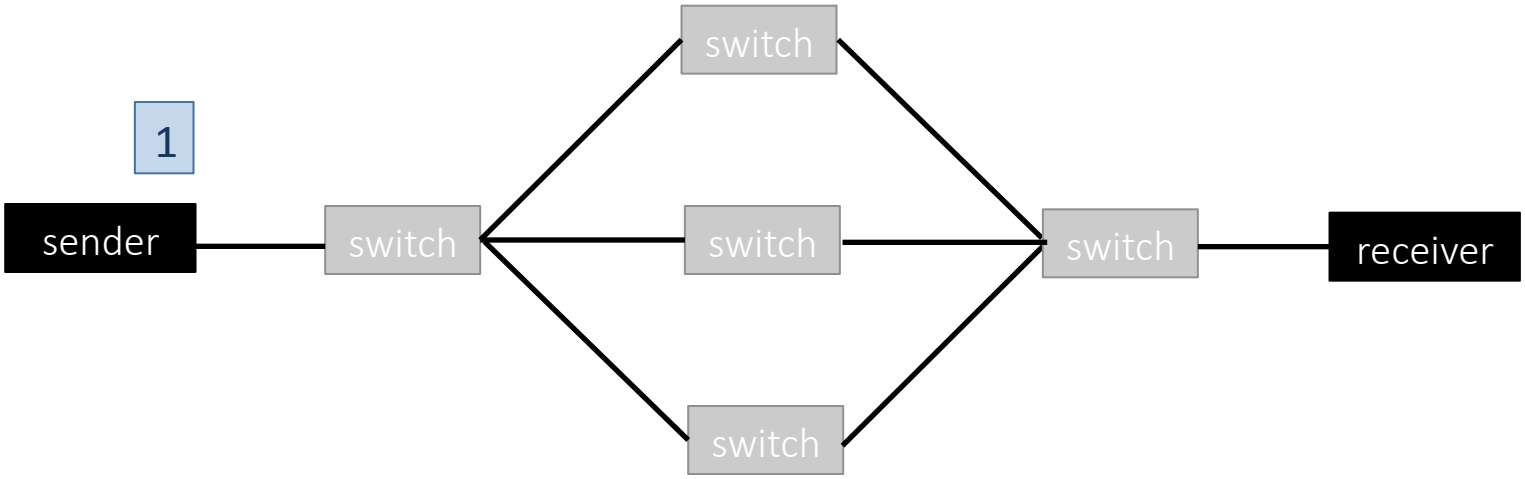
Performance: RAIL 0



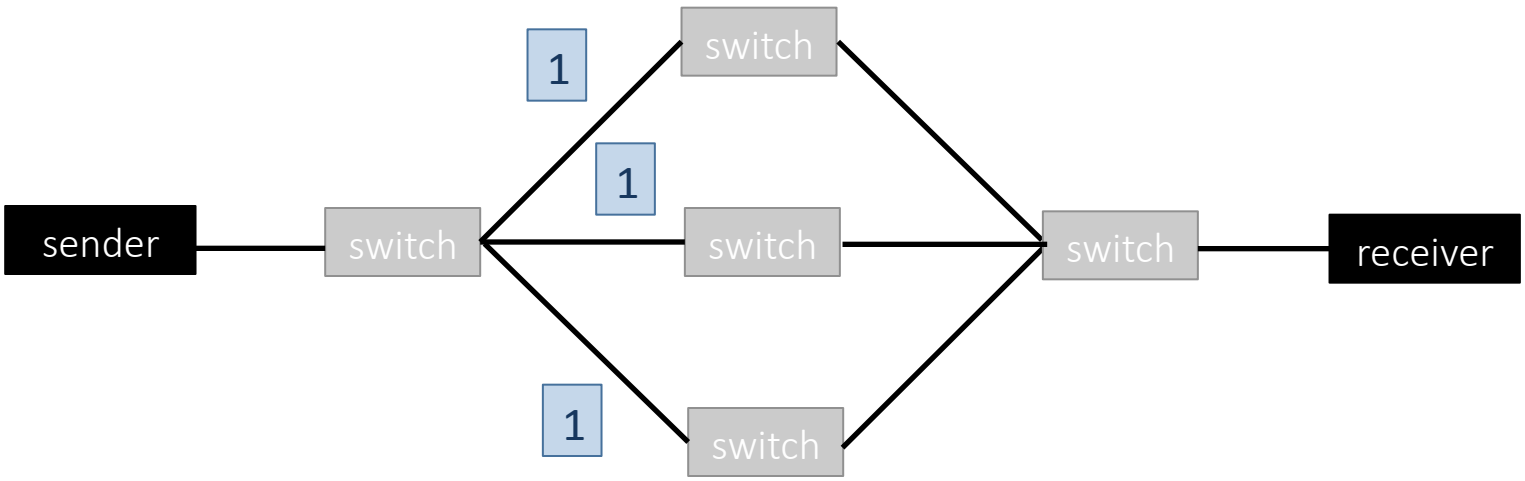
Performance: RAIL 0



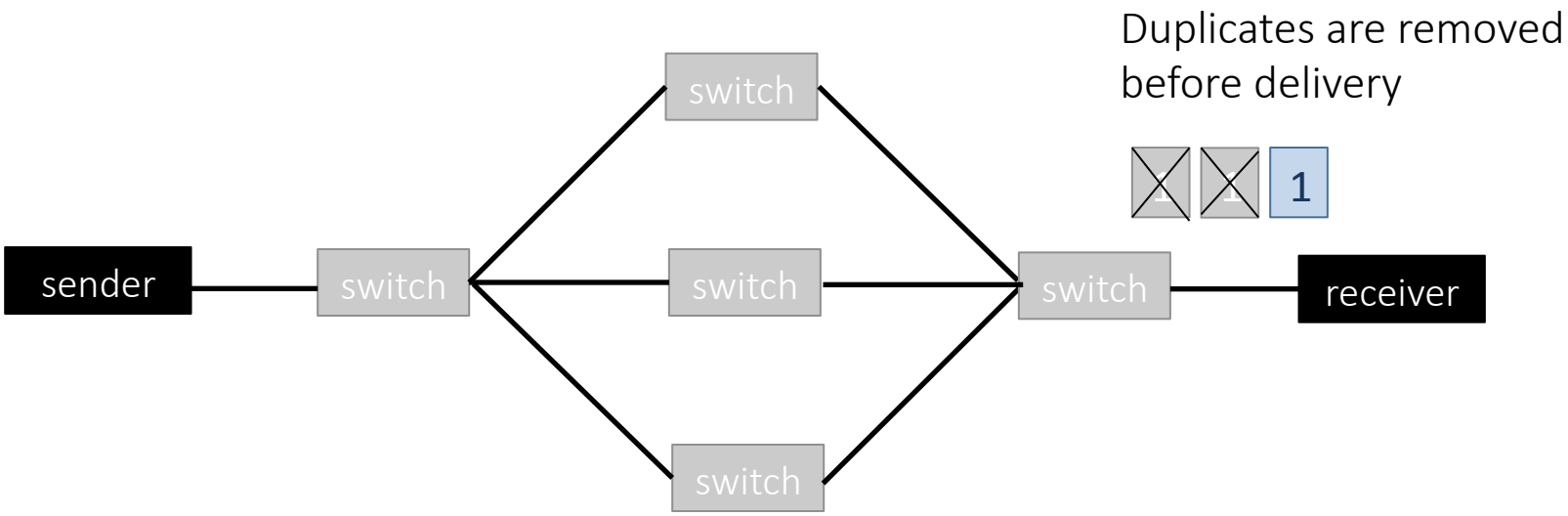
Reliability: RAIL 1



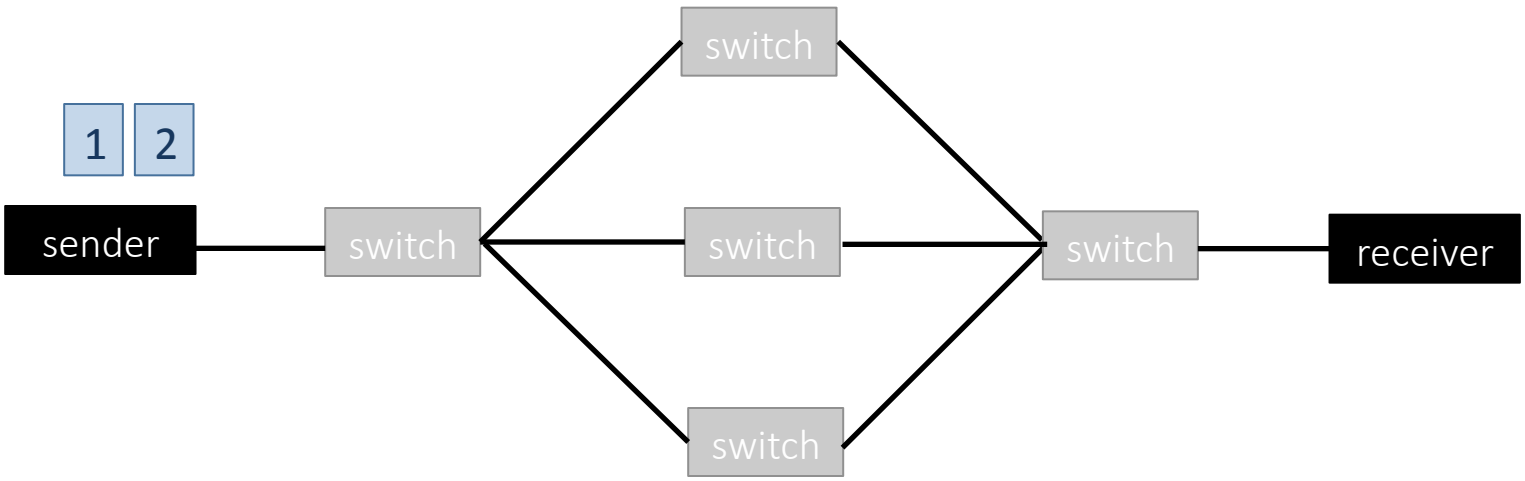
Reliability: RAIL 1



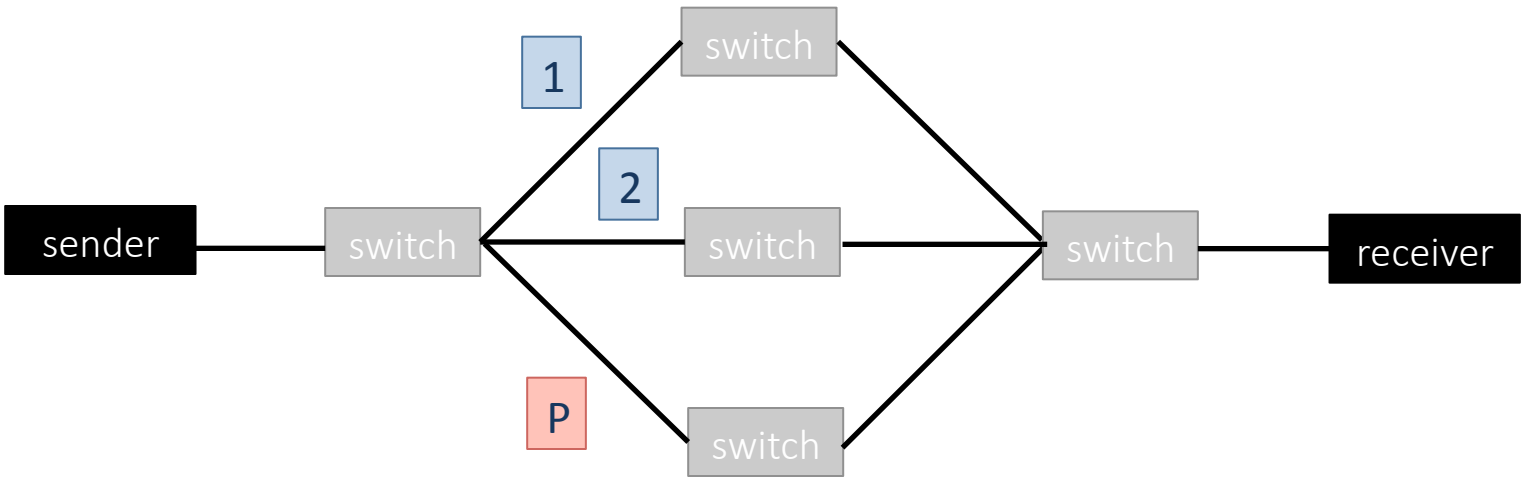
Reliability: RAIL 1



Performance + Reliability: RAIL 4

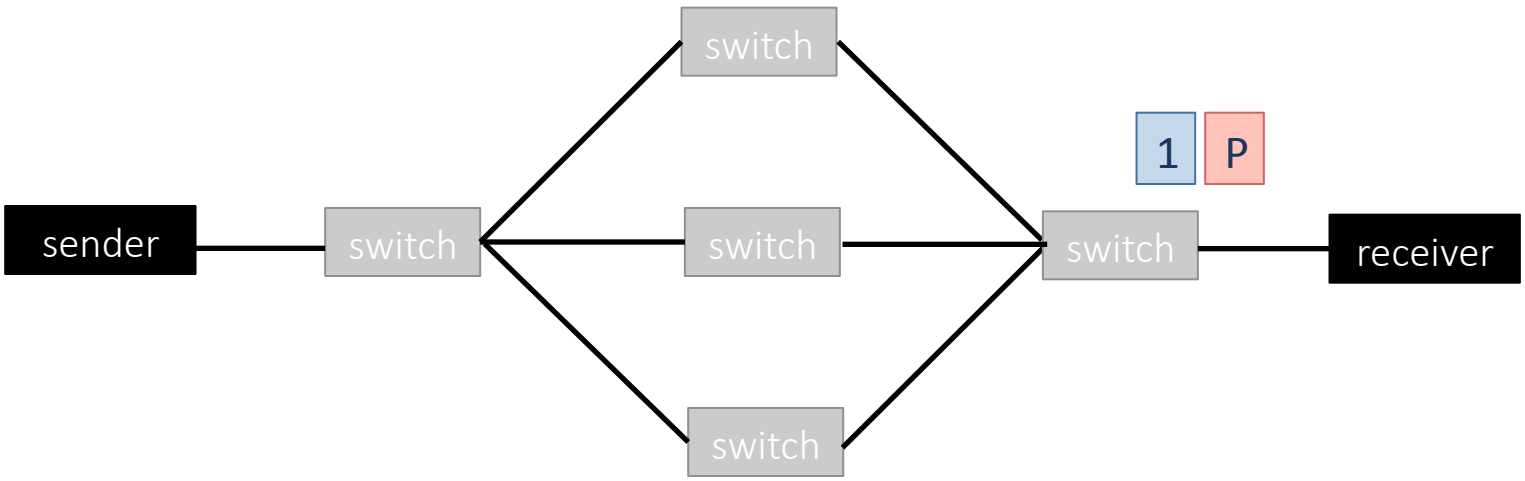


Performance + Reliability: RAIL 4

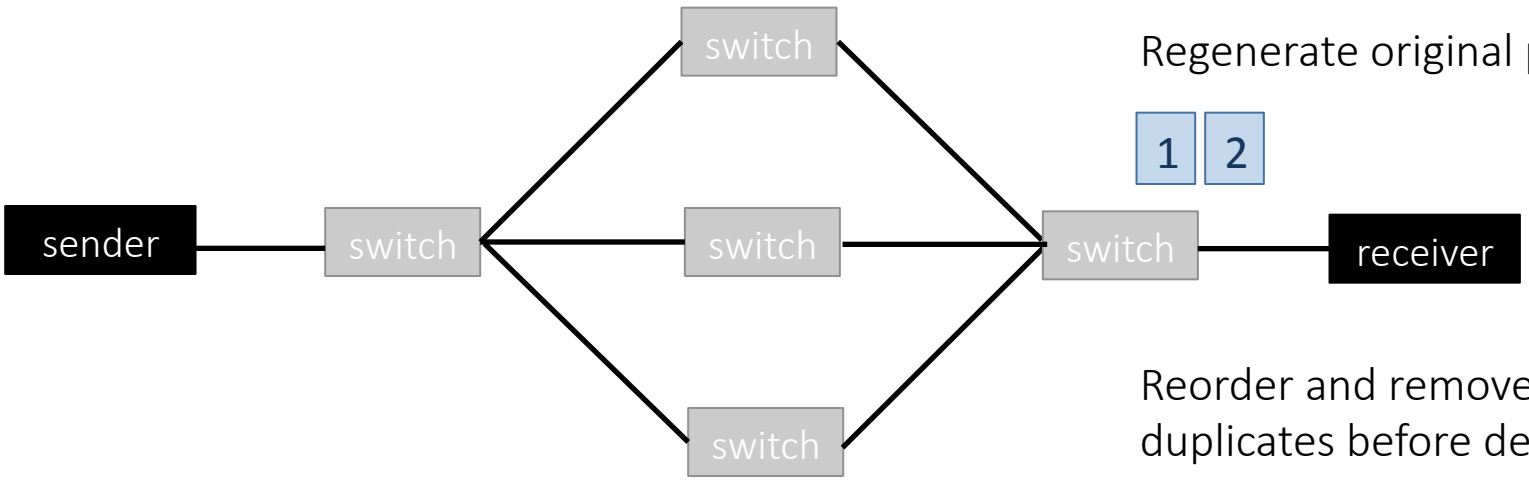


$P = 1 \oplus 2$

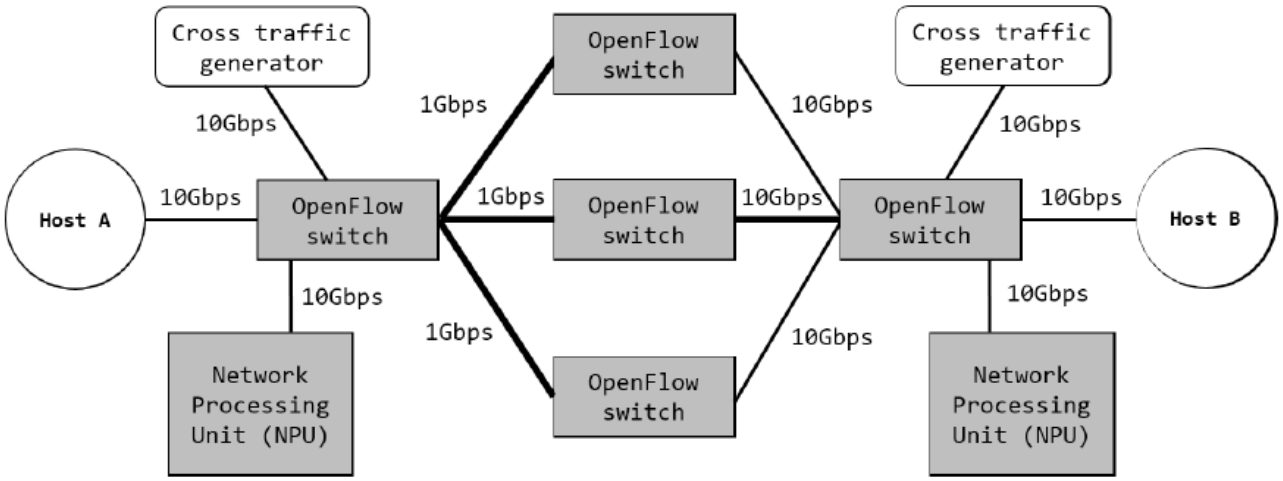
Performance + Reliability: RAIL 4



Performance+ Reliability: RAIL 4



Results: quiescent network



A. Microbenchmark results

	Ethernet STP	RAIL 0	RAIL 1	RAIL 4
latency[1]	0.122ms	0.126ms	0.125ms	0.125ms
min/avg/max	0.152ms	0.166ms	0.160ms	0.158ms
	0.185ms	0.196ms	0.210ms	0.184ms
bandwidth[1]	0.85Gbps	2.55Gbps	0.85Gbps	1.52Gbps
latency[2]	4.017ms	0.126ms	0.125ms	0.126ms
min/avg/max	11.911ms	3.244ms	0.161ms	0.175ms
	17.506ms	13.157ms	0.200ms	0.215ms
bandwidth[2]	0.51Gbps	2.02Gbps	0.85Gbps	1.52Gbps
link failures tolerated	0	0	2	1

[1] Without cross traffic. [2] With cross traffic.

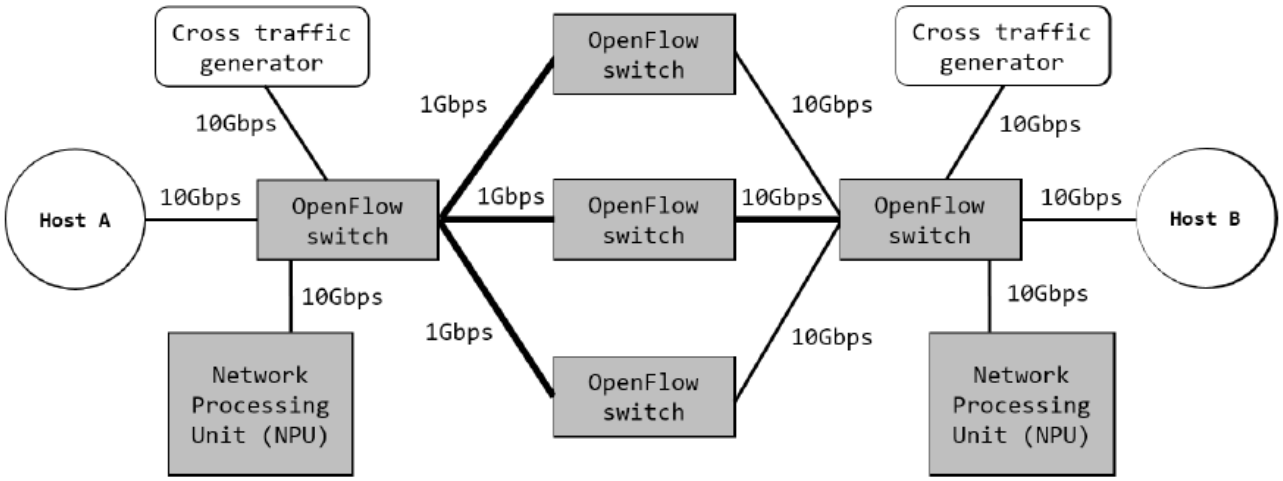
Bandwidth / no load

RAIL0: 3.0x improvement
RAIL1: 1.0x
RAIL4: 1.5x improvement

Latency / no load

RAIL0: unaffected
RAIL1: unaffected
RAIL4: unaffected

Results: with cross traffic



A. Microbenchmark results

	Ethernet STP	RAIL 0	RAIL 1	RAIL 4
latency[1] min/avg/max	0.122ms 0.152ms 0.185ms	0.126ms 0.166ms 0.196ms	0.125ms 0.160ms 0.210ms	0.125ms 0.158ms 0.184ms
bandwidth[1]	0.85Gbps	2.55Gbps	0.85Gbps	1.52Gbps
latency[2] min/avg/max	4.017ms 11.911ms 17.506ms	0.126ms 3.244ms 13.157ms	0.125ms 0.161ms 0.200ms	0.126ms 0.175ms 0.215ms
bandwidth[2]	0.51Gbps	2.02Gbps	0.85Gbps	1.52Gbps
link failures tolerated	0	0	2	1

[1] Without cross traffic. [2] With cross traffic.

Bandwidth / saturated tree

RAIL0: 4.0x improvement
RAIL1: 1.7x improvement
RAIL4: 3.0x improvement

Latency / saturated tree

RAIL0: improved (on avg)
RAIL1: unaffected by traffic
RAIL4: unaffected by traffic

What about security?

We can reuse the same disjoint paths and NPU infrastructure.

- Confidentiality
- Anonymity

Information Slicing [Katti, Sachin, Cohen and D. Katabi].

Information slicing

plaintext M

t	h	e		C	I	A		s	e	e	s

pick d' = 4 paths, d = 3 fragments required for complete assembly.

t	h	e	
C	I	A	
s	e	e	s

d by m matrix

Generate a full rank random matrix A.

1	3	9
4	5	0
2	2	6
7	5	8

d' by d matrix

Information slicing

Compute the product R = AM

1	3	9
4	5	0
2	2	6
7	5	8

x

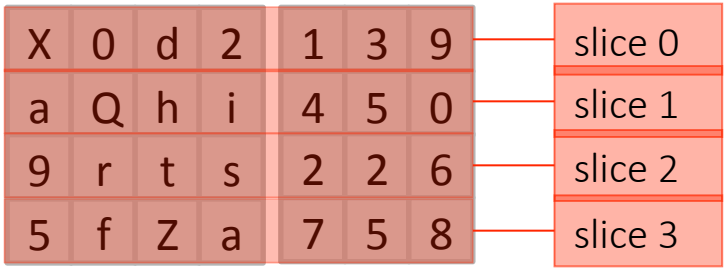
t	h	e	
C	I	A	
s	e	e	s

=

X	0	d	2
a	Q	h	i
9	r	t	s
5	f	Z	a

d’ by d matrix A d by m matrix M d by d matrix R

The information slices are:



Send one slice down each disjoint path.

Joining the slices

Recipient needs d = 3 fragments to compute an inverse A’

X	0	d	2
a	Q	h	i
5	f	Z	a

1	3	9
4	5	0
7	5	8

3 slices received by NPU

0	1	7
9	3	2
8	4	6

Inverse matrix A’

Compute M = A’ R

R

A

0	1	7
9	3	2
8	4	6

x

X	0	d	2
a	Q	h	i
5	f	Z	a

=

t	h	e	
C	I	A	
s	e	e	s

What does information slicing give you?

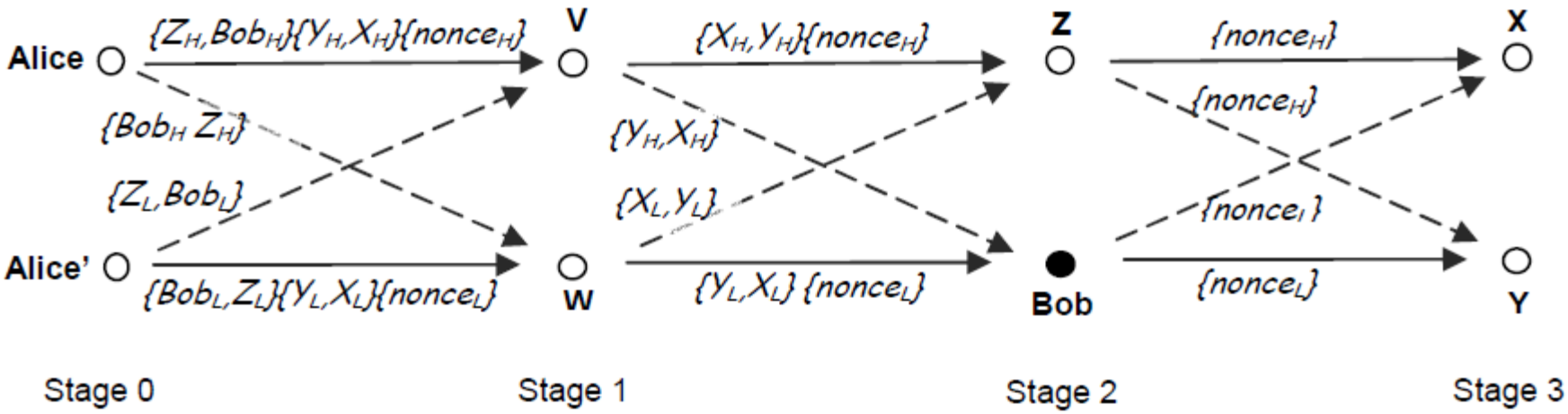
- Isn't this the same as encryption?
  No!
- Settable threshold. Can be changed on the fly.
- No need PKI!
- Can be adapted to provide anonymity as well.

What about anonymity?

- "Anonymity can be built out of confidentiality." [Katti, Sachin, Cohen and D. Katabi]

- Basic idea: hide point-to-point communications by engaging a large group of forwarding peers.
- Let each forwarding peer know its next hop.
- Hops use information slicing to forward data confidentially.
- Destination hop may still forward data!

Anonymity


Alice
Alice'
V
W
Z
Bob
X
Y
{Z_H,Bob_H}{Y_H,X_H}{nonce_H}
{Bob_H Z_H}
{Z_L,Bob_L}
{Bob_L,Z_L}{Y_L,X_L}{nonce_L}
{X_H,Y_H}{nonce_H}
{Y_H,X_H}
{X_L,Y_L}
{Y_L,X_L} {nonce_L}
{nonce_H}
{nonce_H}
{nonce_L}
{nonce_L}
Stage 0
Stage 1
Stage 2
Stage 3

Q&A