Treating software-defined networks like disk arrays

Zhiyuan Teo
Cornell University

Joint work with Noah Apthorpe, Vasily Kuksenkov, Ken Birman and Robbert van Renesse

Problems with today’s Ethernet

- Slow.
- Unreliable.
- Not secure.

Focus of this paper

Work in progress

How did we get ourselves into this terrible state?

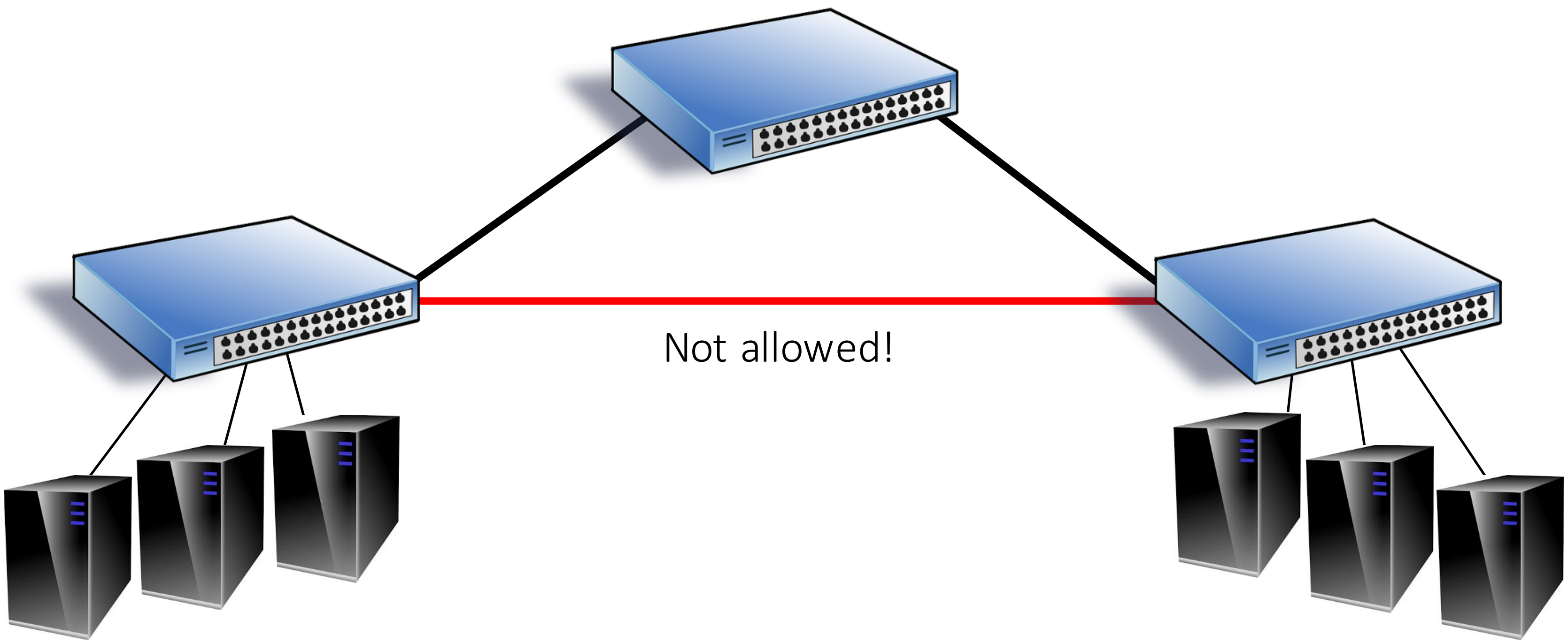
Spanning Tree Protocol.

* How popular is Ethernet?

> 85% , according to Cisco.

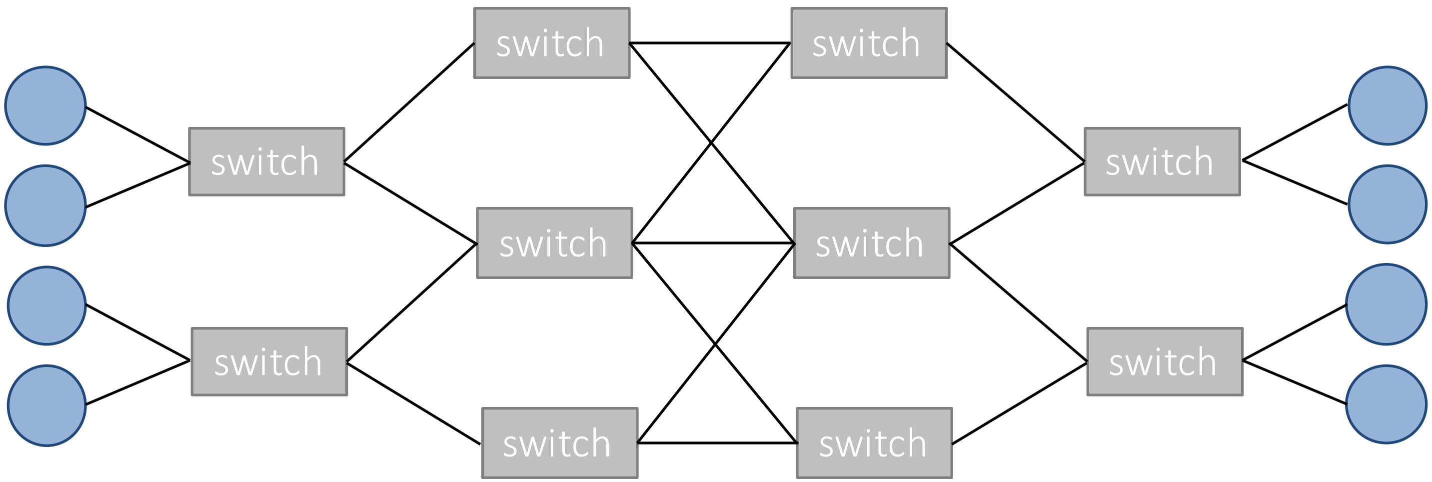
http://www.cisco.com/c/en/us/tech/lan-switching/ethernet/index.html

What is spanning tree protocol and why should I care?

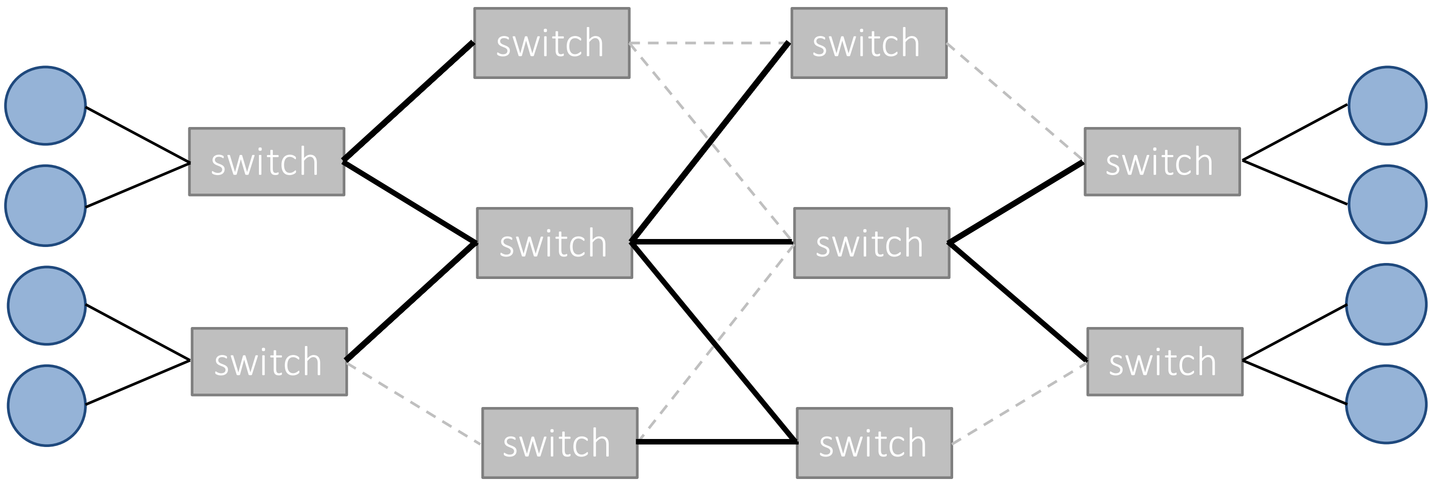


Ethernet standards from 1990! [IEEE 802.1D]

A more complicated example



A more complicated example



STP will disable some bridge links to prevent loops.

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

What does multipath forwarding really mean?

1. You can’t change standards. (must use STP)
2. But you can employ some tricks to give the illusion of multiple paths in forwarding .

Proposed multipath techniques

1. Equal cost multiple paths (ECMP) [1]
2. Multiple Spanning Tree (MSTP) [10]
3. Link Aggregation (IEEE 802.3) [6]
4. Multipath TCP (MPTCP) [7]
5. Multiple Topologies for IP-only protection against network failures [11]
6. STAR routing [21]
7. SPAIN [20]

…and more.

Existing multipath techniques are flawed

- ‘Multipath’ as an aggregate statement.
- Pre-computed solutions for failures.
- Reliance on extensive hardware/software support.
- Fixing the problem after the fact.

Let’s take a step back

- Questions about the network should be answered by the network itself.
- The answers should be dynamic, current and intelligent, not precomputed.
- Multipath should really mean simultaneous use of multiple paths!

Our approach

- Use SDN to provide baseline “regular” network access.
- For special flows, use multiple disjoint paths simultaneously.
- Select a data scheme for each flow to favor performance/reliability.

Completely backward compatible: does not require change or awareness from network clients.

How is this relevant to IoT?

- IoT devices require data networking access.
- Specific applications may require more bandwidth, lower latency, etc.
- Many IoT devices are sealed; cannot upgrade easily.

How we build multipath networking

- Regular network access.
- Access via special flows.

Regular forwarding

- On cold start, controller computes topology.
- Build a default spanning tree.
- Regular flows use spanning tree.
- Controller emulates learning switch algorithm.
- Network operates as normal by default.

Special flows

- For performance and reliability, use disjoint paths in the network.
- Key insight: model after RAID.

Redundant Array of Independent Disks (RAID)

Redundant Array of Independent Links (RAILS)

RAID schemes

- Encoding applied on a predetermined granularity (usually disk block).

- RAID 0 = combine all independent disks.
- RAID 1 = replicate over all independent disks.
- RAID 2-6 = parity protected striping.

- RAID controller performs actual write.

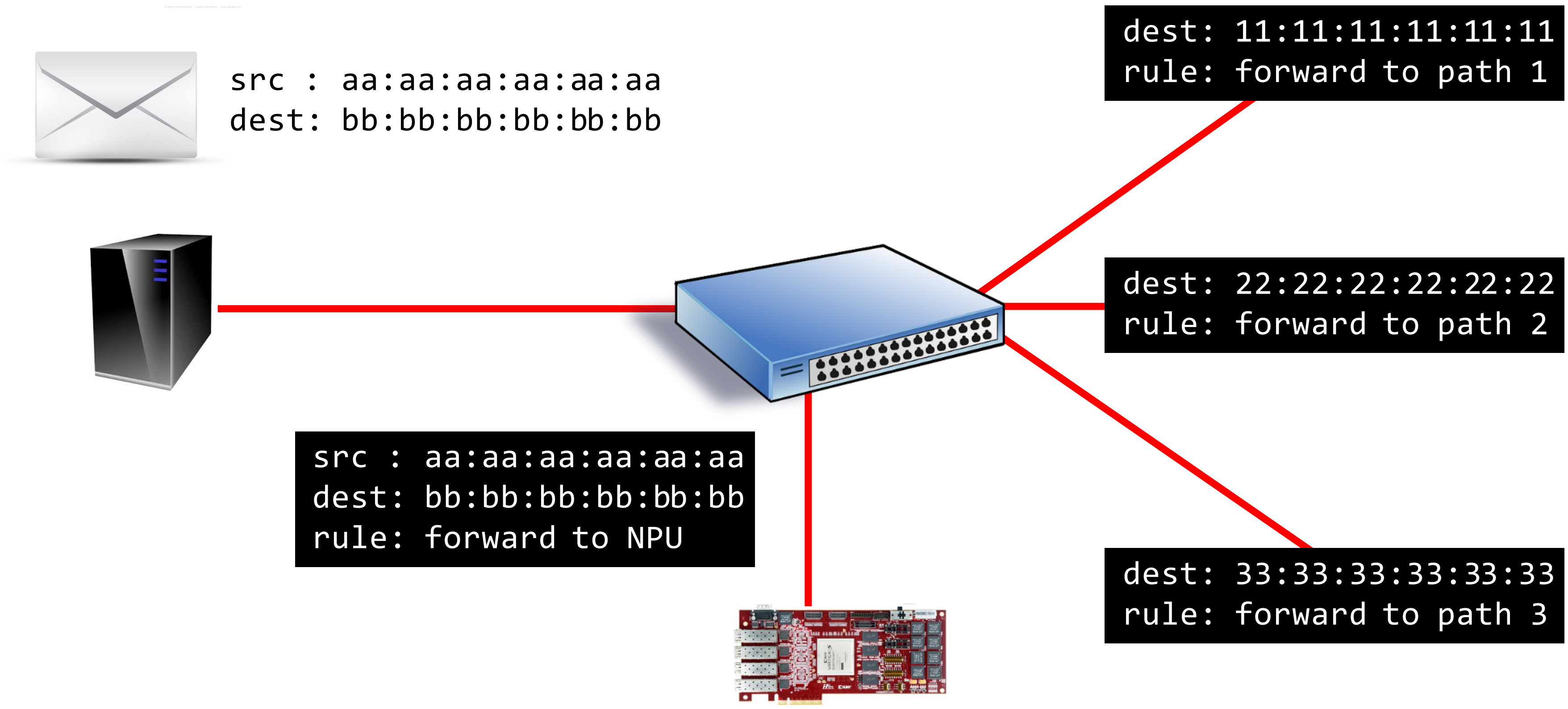
RAIL schemes

- Apply RAID encoding on the granularity of a packet.

- RAIL 0 = round robin packets over paths.
- RAIL 1 = replicate packets across paths.
- RAIL 4 = one parity packet per n-1 paths.

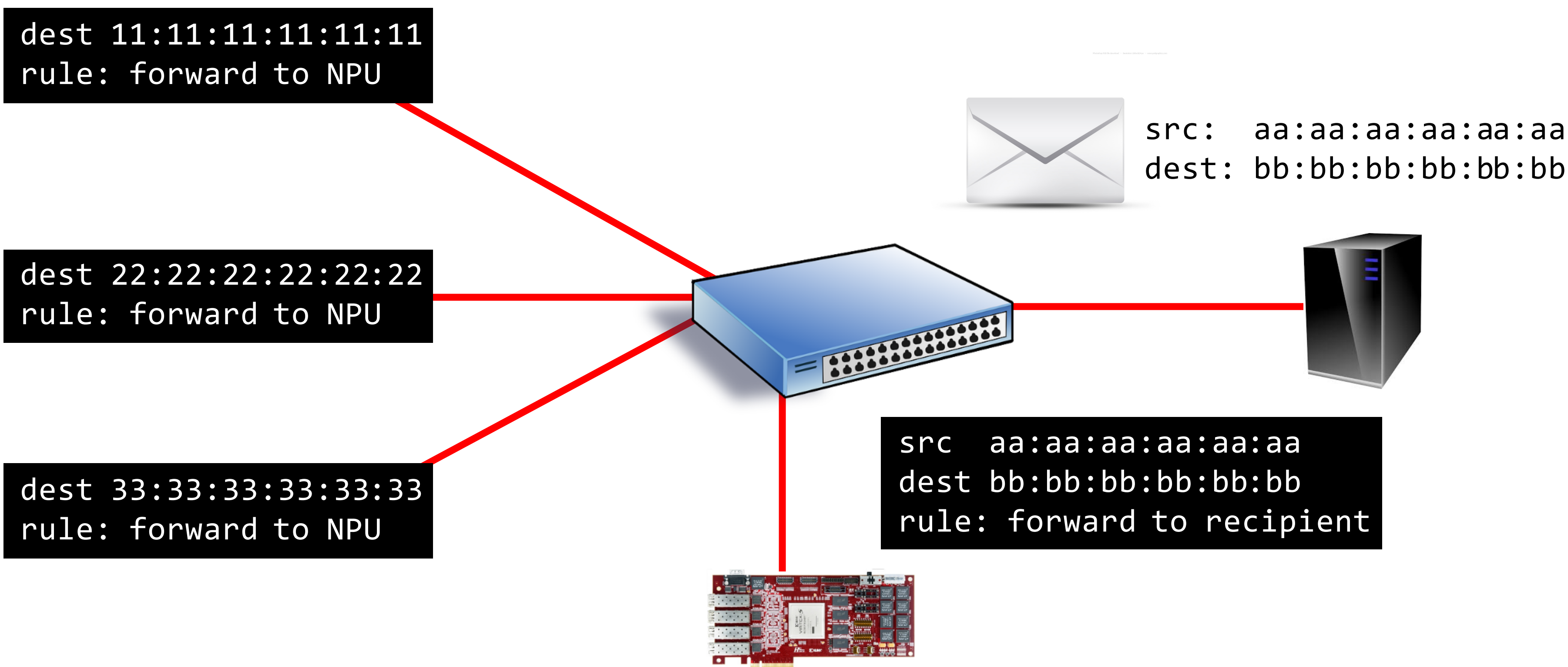
- Packets written by Network Processing Unit.

Ingress switch setup



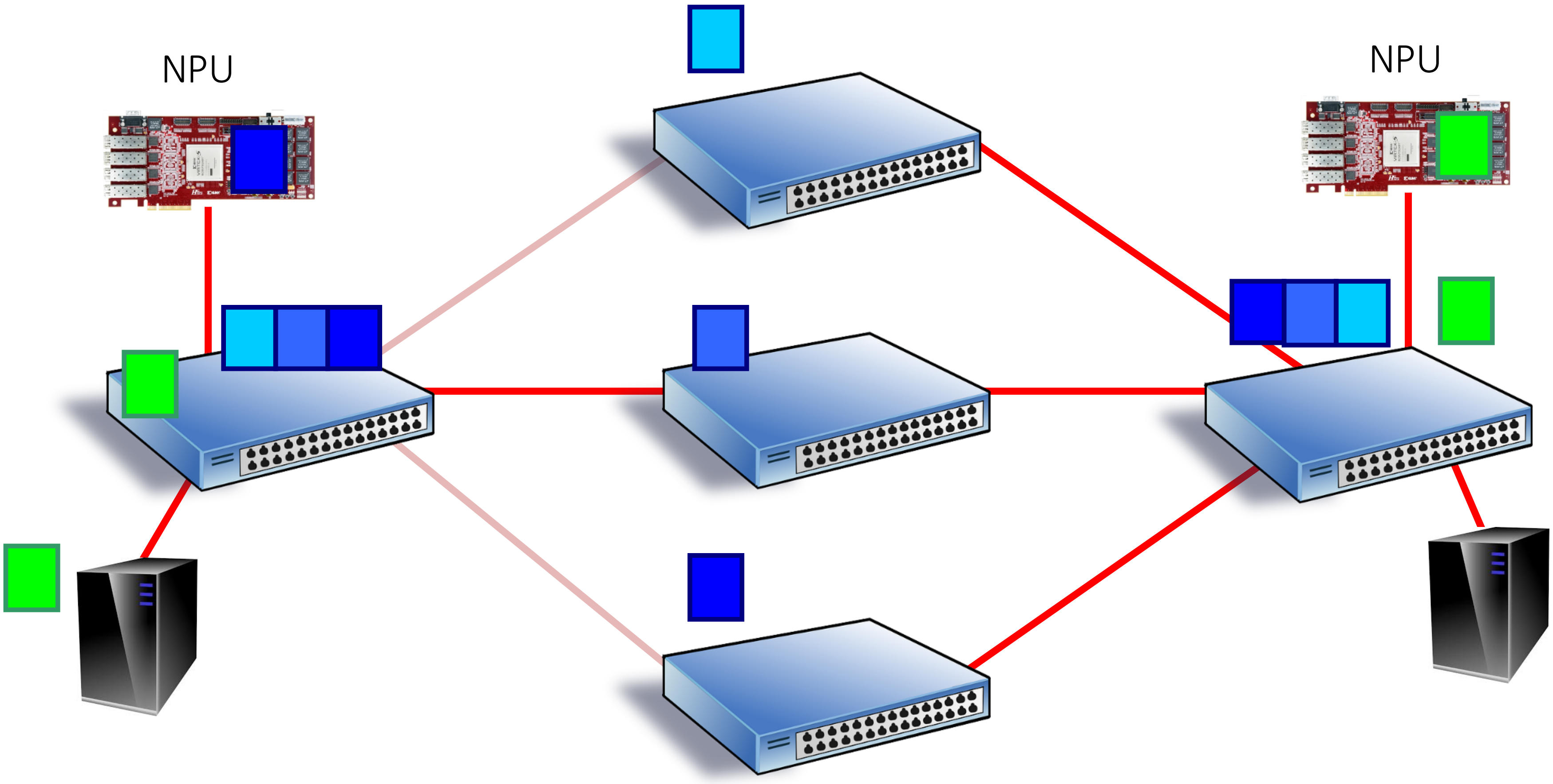
NPU rewrites packets and transform dest MAC to path addresses

Egress switch setup



NPU rewrites packets and transforms path addresses to original dest MAC

High level idea

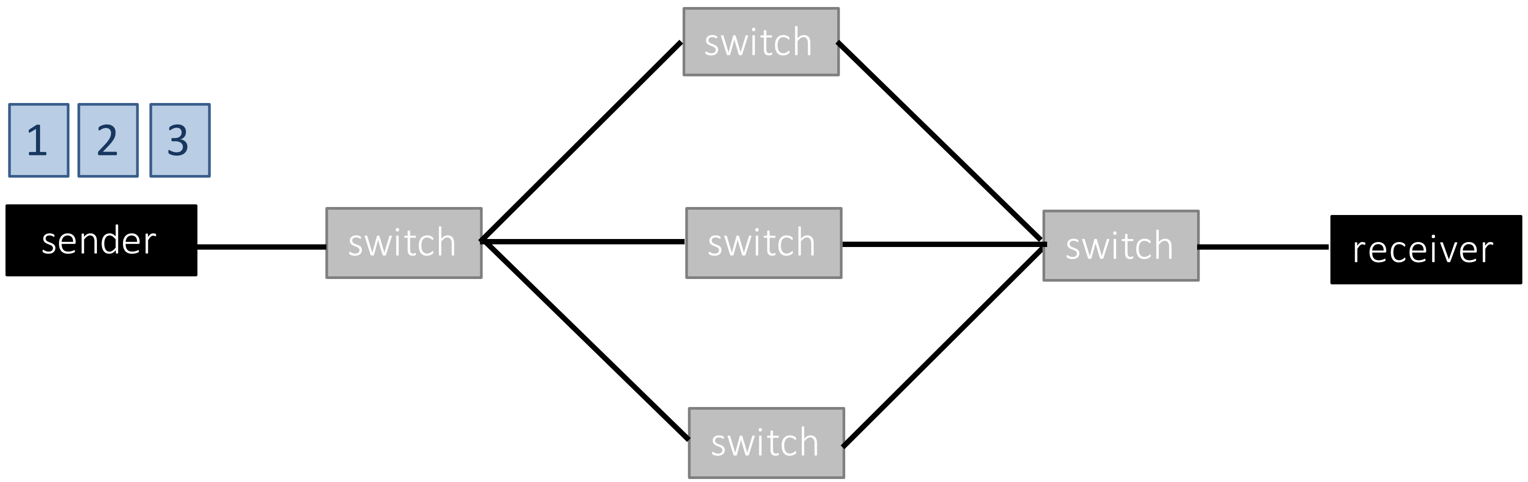


Improving performance

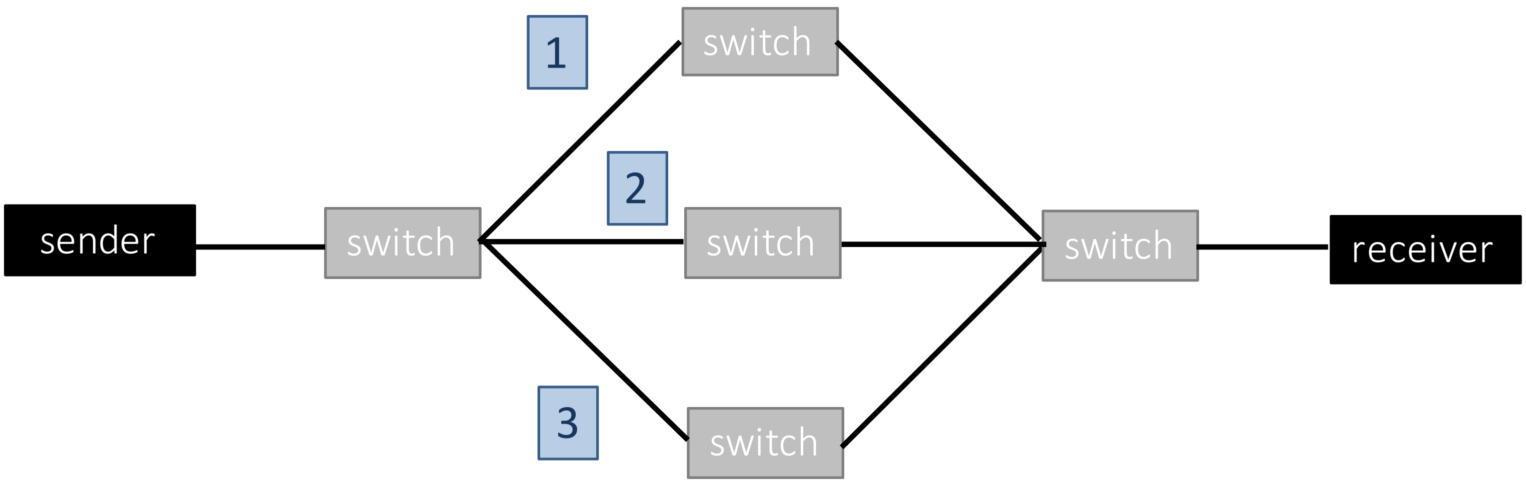
- Similar to RAID0.
- Send disjoint sets of packets down each path.
- Buffer and reorder packets on egress.
- Can adjust per-path load weightage on the fly.

Disadvantage: high latency. Need to wait for packets from slowest link.

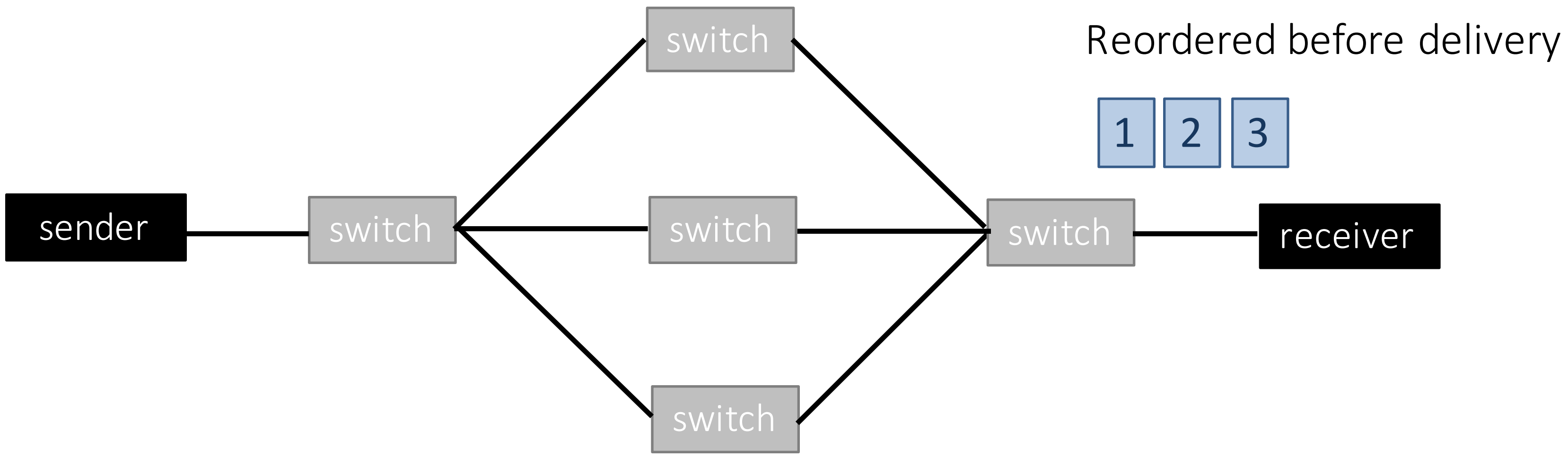
RAIL 0



RAIL 0



RAIL 0

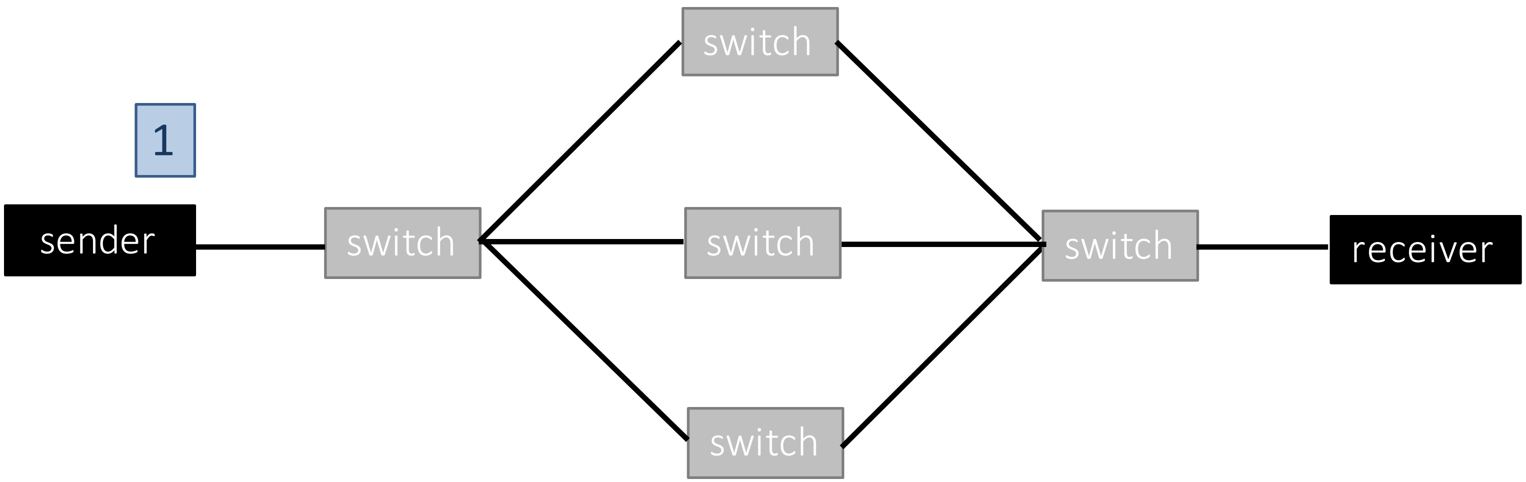

Reordered before delivery
1
2
3
sender
switch
switch
switch
switch
switch
receiver

Improving reliability

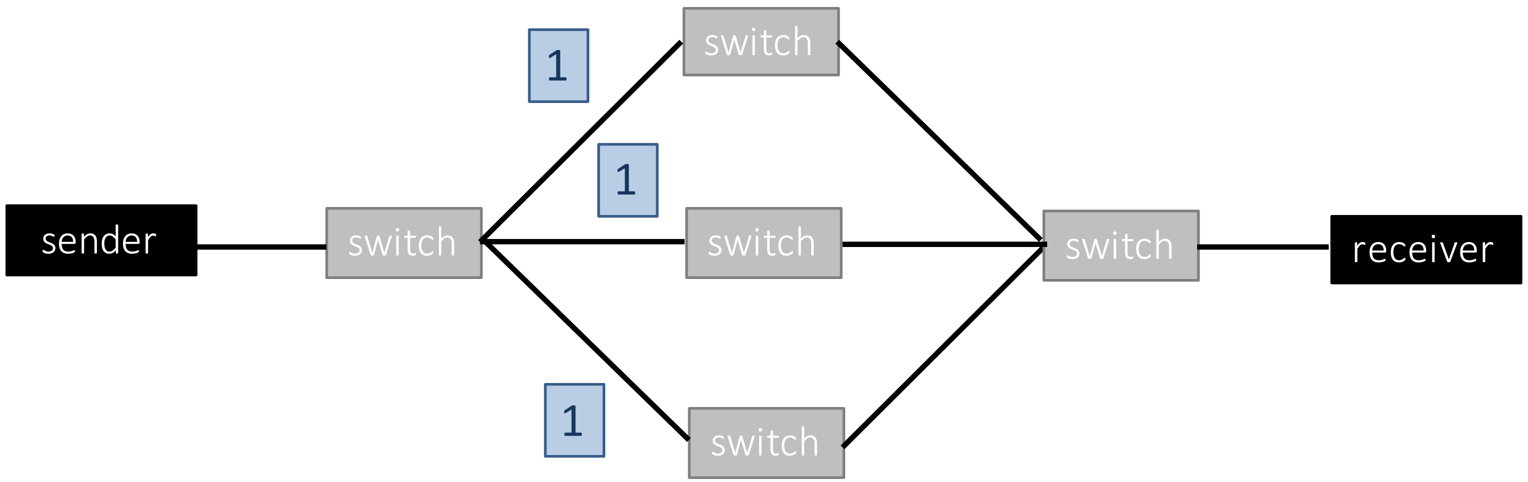
- Similar to RAID1.
- Replicate packets on each path.
- Reorder packets and discard duplicates on egress.

Disadvantage: bandwidth wastage from redundant copies.

RAIL 1

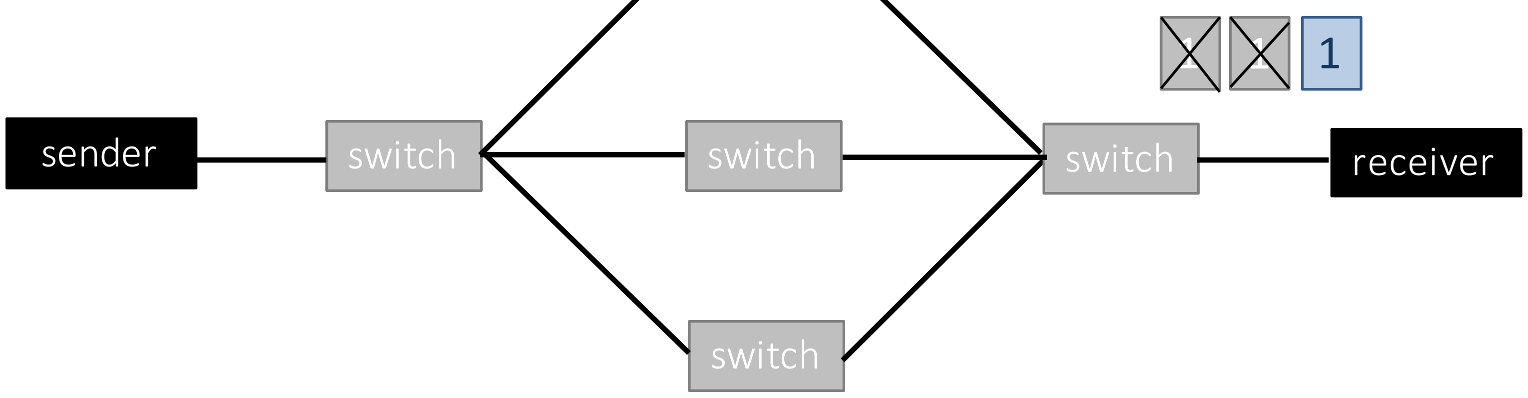


RAIL 1



RAIL 1

Duplicates are removed before delivery

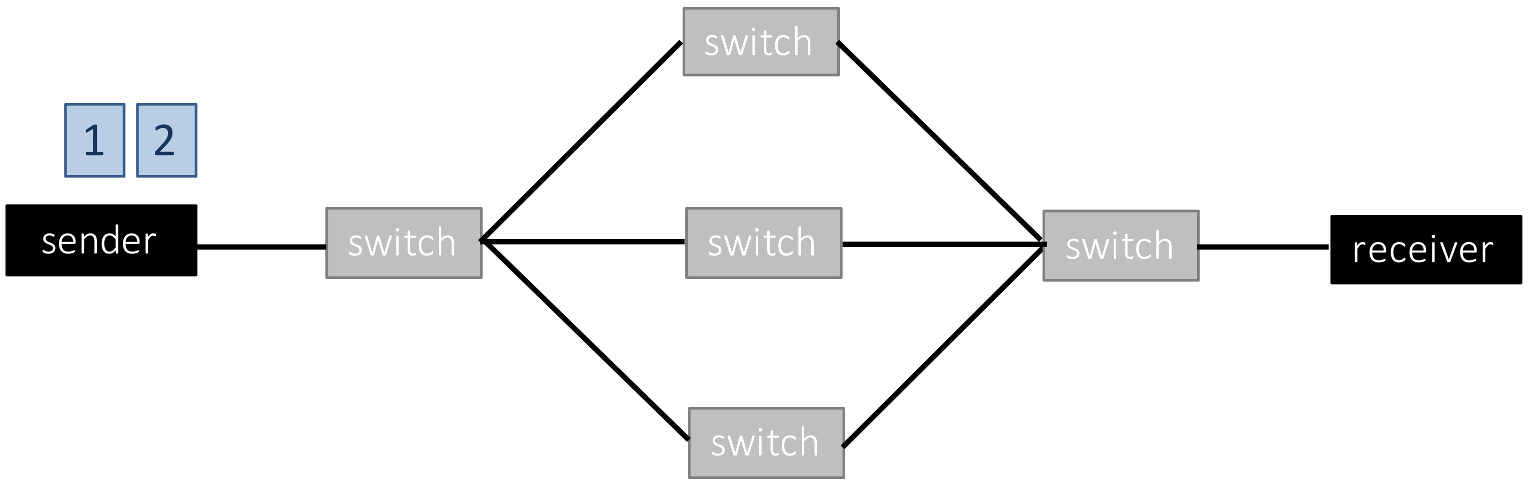


Improved performance & reliability

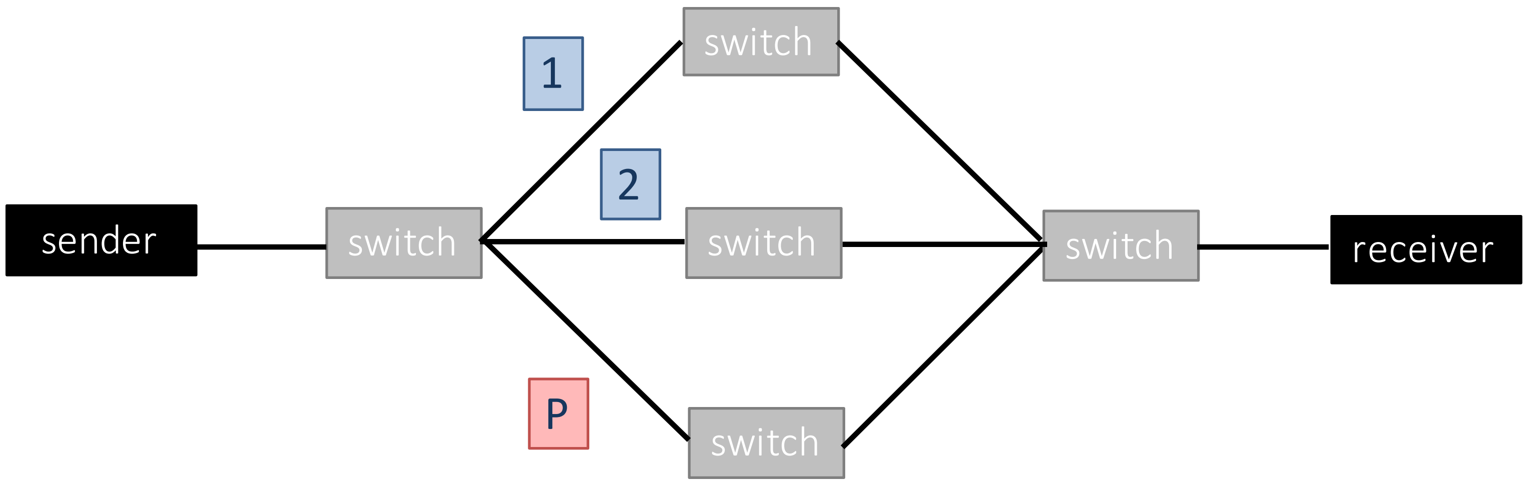
- Tolerance for one link failure: use RAIL4.
- For each n-1 packets, compute a parity packet.
- Reorder and reassemble packets on egress.

Disadvantage: high computational cost.

RAIL 4

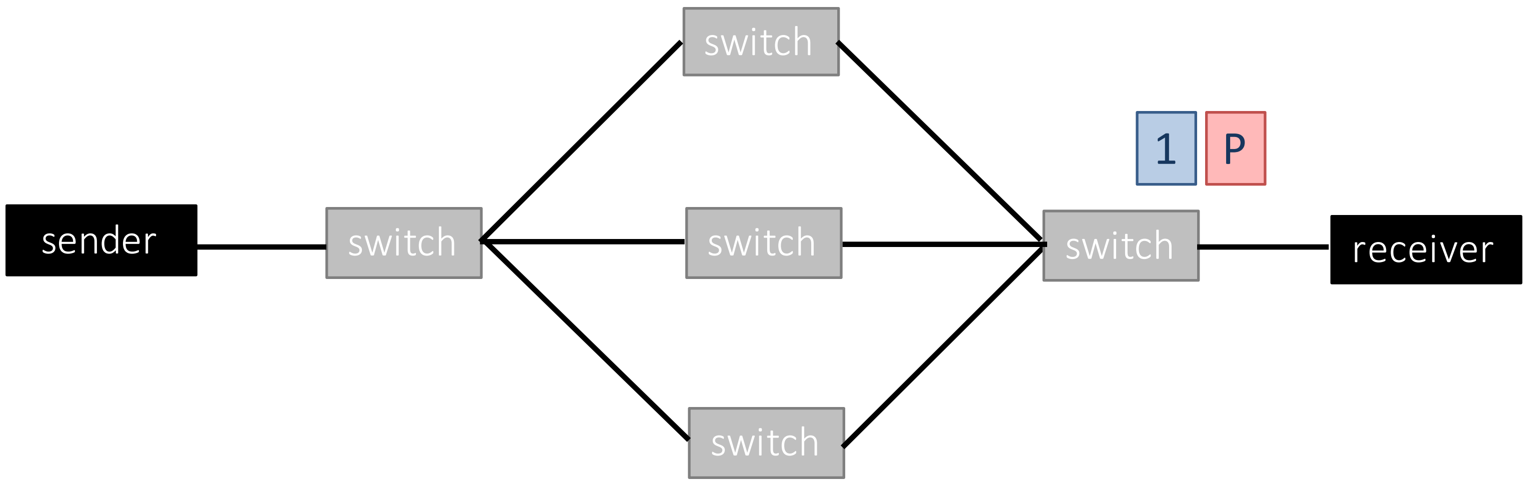


RAIL 4

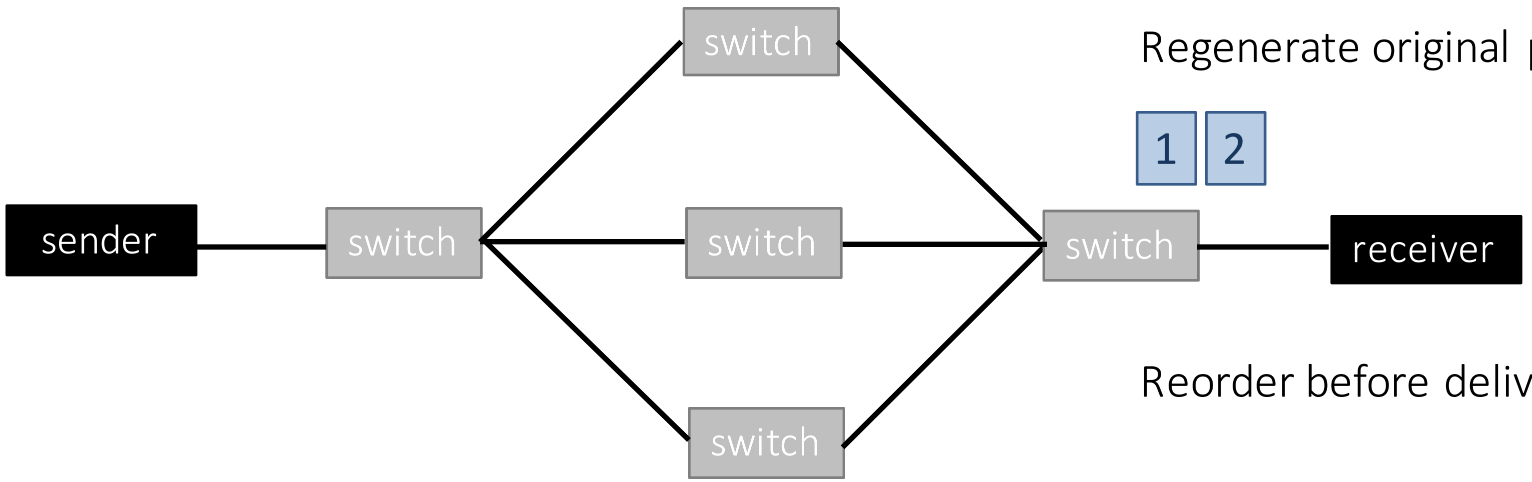


P = 1 ⊕ 2

RAIL 4



RAIL 4



Regenerate original packet

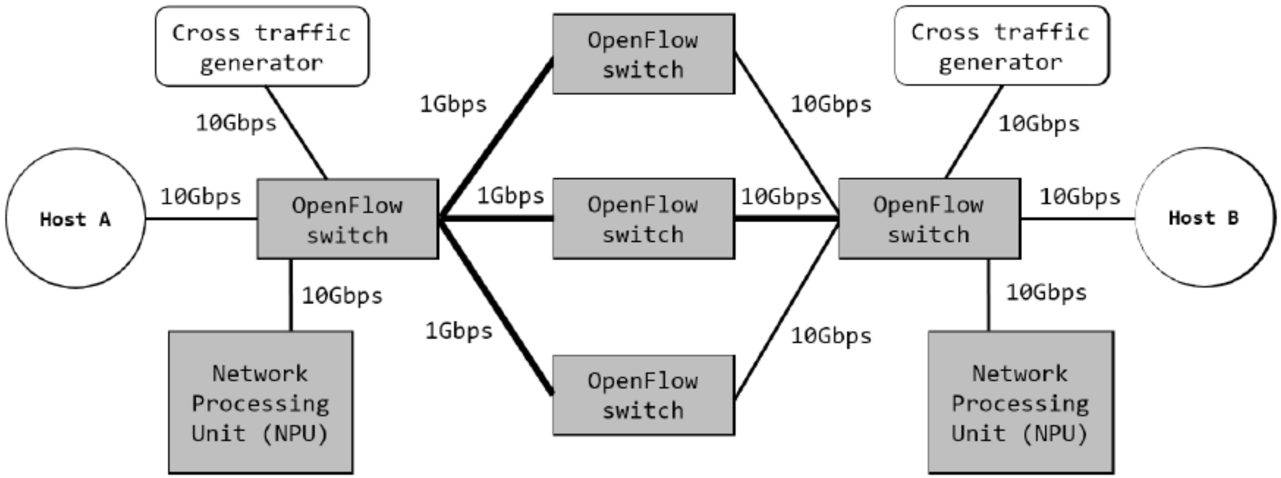
Reorder before delivery.

Generalized k-of-n paths

- Tolerates up to k failures.
- Maintain a counter c. For each packet, replicate k+1 times.
- Send each replica down the c mod n path.
- Reorder and discard duplicates on egress.

Disadvantage: not the most efficient representation.

Results: quiescent network



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.

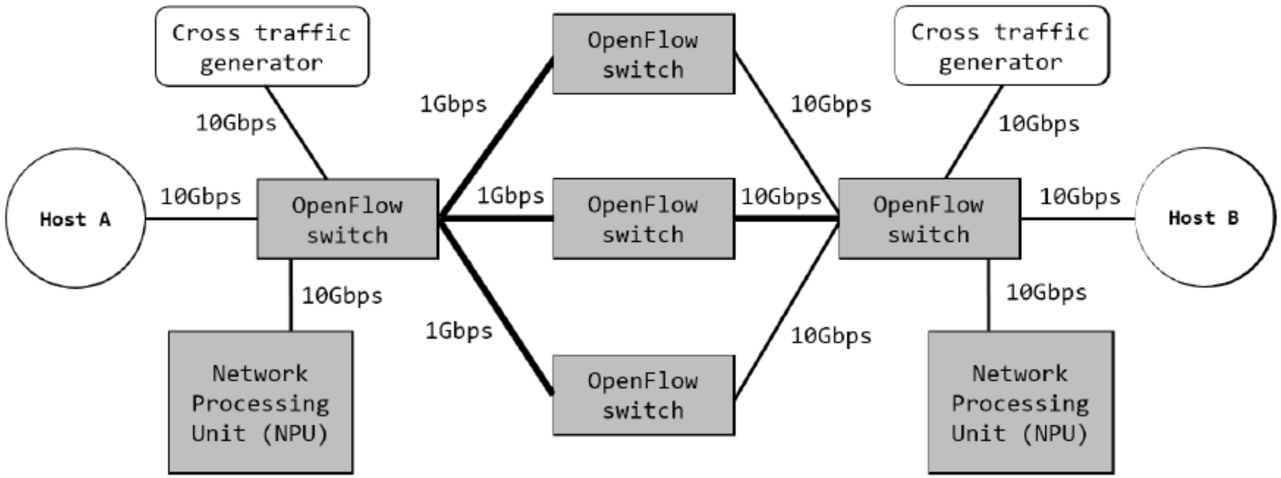
Bandwidth / no load

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

FAQ

- Can everybody use this at the same time?
- What if OpenFlow virtual paths tunnel over same physical links?
- Are these the most efficient representations?

Related work

- [1] IEEE 802.1Qbp. Equal Cost Multiple Paths, IEEE 2014.
- [2] Reitblatt, Mark, et al. "FatTire: declarative fault tolerance for software-defined networks." Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013.
- [3] Floodlight OpenFlow controller. http://www.projectfloodlight.org/floodlight/
- [4] Al-Fares, Mohammad, et al. "Hedera: Dynamic Flow Scheduling for Data Center Networks." NSDI. Vol. 10. 2010.
- [5] http://standards.ieee.org/develop/regauth/ethertype/eth.txt
- [6] IEEE 802.1-AX 2008. Link Aggregation, IEEE 2008.
- [7] A. Ford, C. Raichu, M. Handley, O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", IETF, RFC 6824, Jan. 2013. [Online]. Available: https://tools.ietf.org/html/rfc6824
- [8] Kostopoulos, Alexandros, et al. "Towards multipath TCP adoption: challenges and opportunities." Next Generation Internet (NGI), 2010 6th EURO-NF Conference on. IEEE, 2010.
- [9] R. Winter, M. Faath, A. Ripke, "Multipath TCP Support for Single homed End-Systems", IETF, Internet-Draft draft-wr-mptcp-singlehomed-05, Jul. 2013. [Online]. Available: https://tools.ietf.org/html/draftwr-mptcp-single-homed-05
- [10] IEEE 802.1Q-2011. VLAN Bridges, IEEE 2011.
- [11] Apostolopoulos, George. "Using multiple topologies for IP-only protection against network failures: A routing performance perspective." ICSFORTH, Greece, Tech. Rep (2006).
- [12] Marian, Tudor, Ki Suh Lee, and Hakim Weatherspoon. "NetSlices: scalable multi-core packet processing in user-space." Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems. ACM, 2012.
- [13] OpenFlow Switch Consortium. "OpenFlow Switch Specification Version 1.0.0." (2009).
- [14] Open vSwitch. http://openvswitch.org/
- [15] Motiwala, Murtaza, et al., Path splicing. ACM SIGCOMM Computer Communication Review. Vol. 38. No. 4. ACM, 2008.
- [16] POX. http://www.noxrepo.org/pox/about-pox/
- [17] Patterson, David A., Garth Gibson, and Randy H. Katz., A case for redundant arrays of inexpensive disks (RAID). Vol. 17. No. 3. ACM,
- 1988.
- [18] IEEE 802.1D-2004. Media Access Control (MAC) Bridges, IEEE 2004.

Related work

- [19] Weatherspoon, Hakim, et al., Smoke and Mirrors: Reflecting Files at a Geographically Remote Location Without Loss of Performance. FAST. 2009.
- [20] Mudigonda, Jayaram, et al., SPAIN: COTS Data-Center Ethernet for Multipathing over Arbitrary Topologies. NSDI. 2010.
- [21] Lui, King-Shan, Whay Chiou Lee, and Klara Nahrstedt. ”STAR: a transparent spanning tree bridge protocol with alternate routing.” ACM SIGCOMM Computer Communication Review 32.3 (2002): 33-46.
- [22] Narayanan, Rajesh, et al., A framework to rapidly test SDN use cases and accelerate middlebox applications. Local Computer Networks (LCN), 2013 IEEE 38th Conference on. IEEE, 2013.
- [23] Narayanan, Rajesh, et al., Macroflows and microflows: Enabling rapid network innovation through a split SDN data plane. Software Defined
- Networking (EWSDN), 2012 European Workshop on. IEEE, 2012.

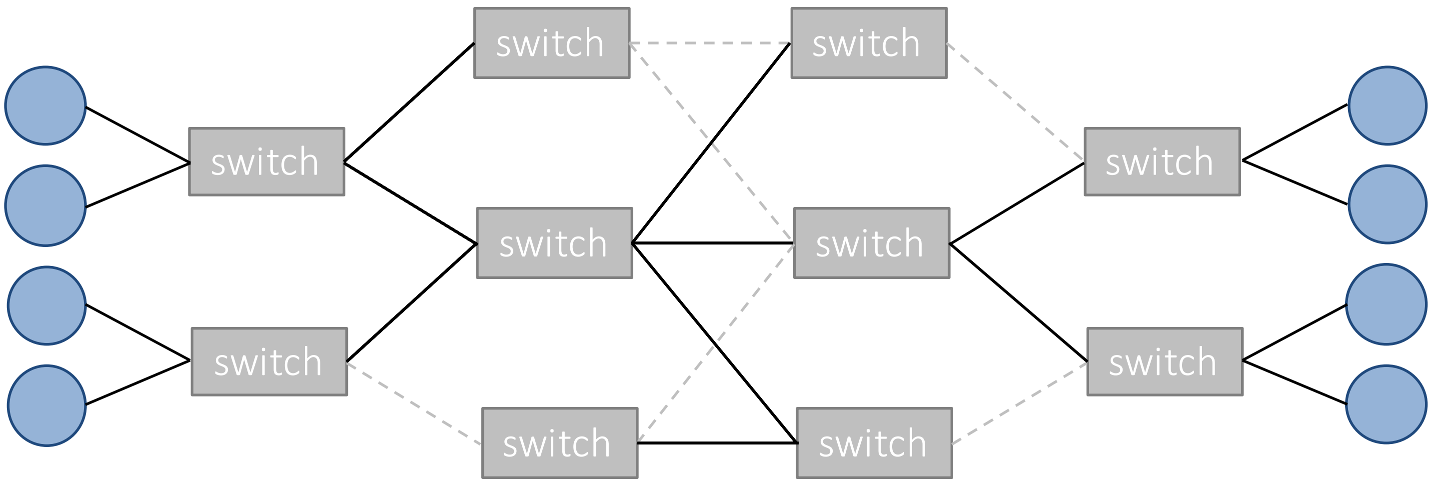
Q&A

Thank you

Backup slides

- Existing multipath techniques.

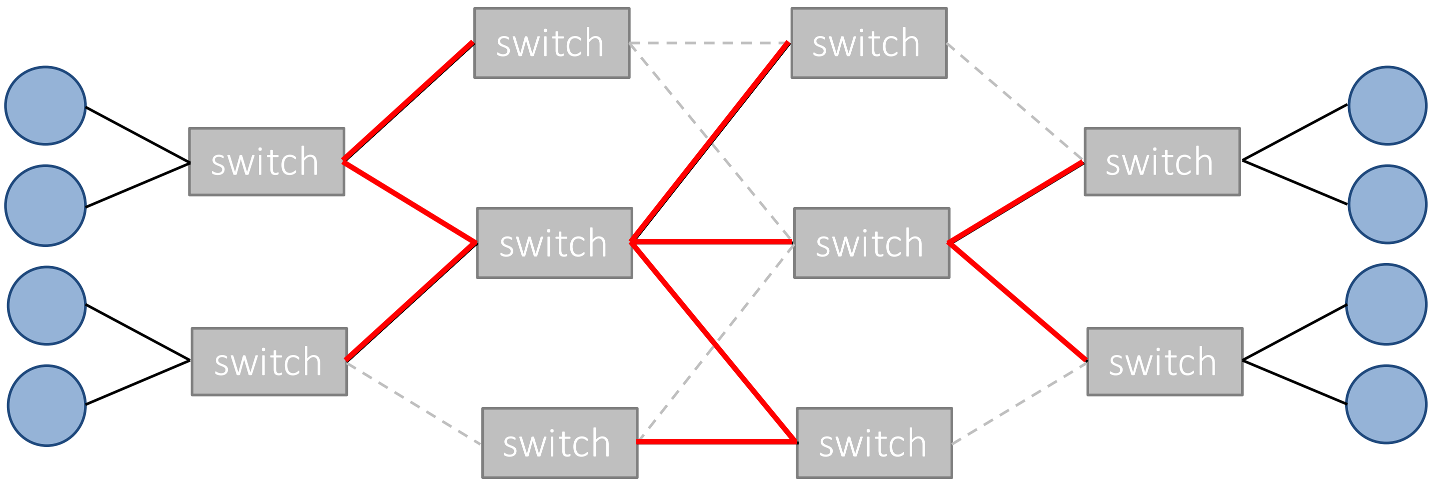
ECMP



Hash flows across multiple paths.

Use of "multiple paths" is an aggregate statement.

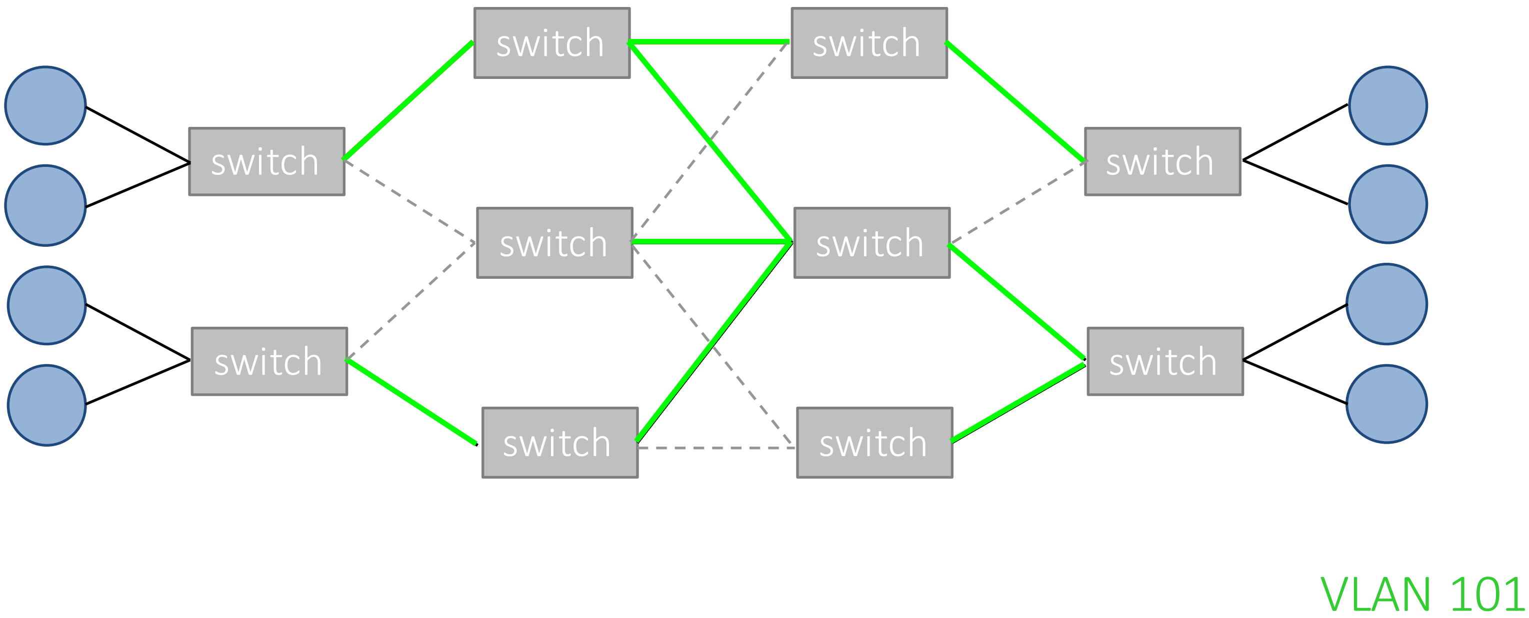
SPAIN [Jayaram et al, NSDI '2010]



VLAN 100

Provision several VLANs with different spanning trees.
Client switches VLANs when failure is suspected.

SPAIN [Jayaram et al, NSDI '2010]



Provision several VLANs with different spanning trees.
Client switches VLANs when failure is suspected.

SPAIN [Jayaram et al, NSDI '2010]

6.6 Handling failures

Failure detection, for a SPAIN end host, consists of detecting a VLAN failure and selecting a new VLAN for the affected flows; we have already described VLAN selection (Algorithm 3).

While we do not have a formal proof, we believe that SPAIN can almost always detect that a VLAN has failed with respect to an edge switch *es*, because most failures result in observable symptoms, such as a lack of incoming packets (including chirp responses) from *es*, or from severe losses on TCP flows to hosts on *es*.

Rely on symptoms to guess network failure. Fix the problem after it occurs.

MPTCP [IETF rfc 6824 '13]

1.1. Design Assumptions

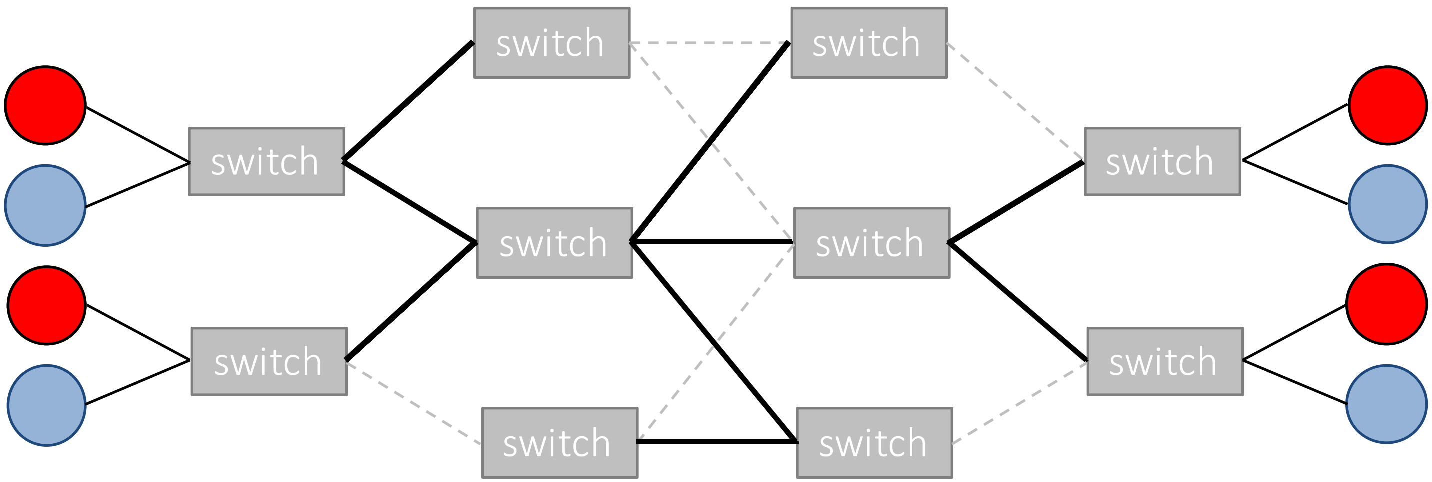
In order to limit the potentially huge design space, the working group imposed two key constraints on the multipath TCP design presented in this document:

- It must be backwards-compatible with current, regular TCP, to increase its chances of deployment
- It can be assumed that one or both hosts are multihomed and multiaddressed

To simplify the design we assume that the presence of multiple addresses at a host is sufficient to indicate the existence of multiple paths. These paths need not be entirely disjoint: they may share one or many routers between them. Even in such a situation making use of multiple paths is beneficial, improving resource utilisation and resilience to a subset of node failures. The congestion control algorithms defined in [5] ensure this does not act detrimentally. Furthermore, there may be some scenarios where different TCP ports on a single host can provide disjoint paths (such

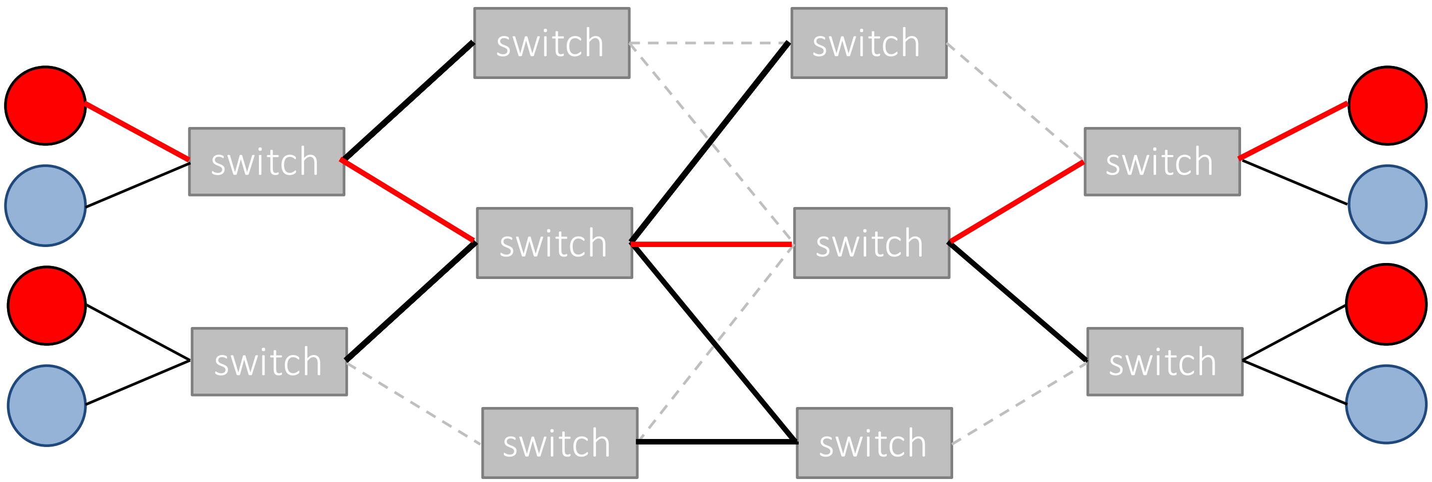
Access the network through multiple interfaces. Hope for path diversity.

MPTCP [IETF rfc 6824 '13]



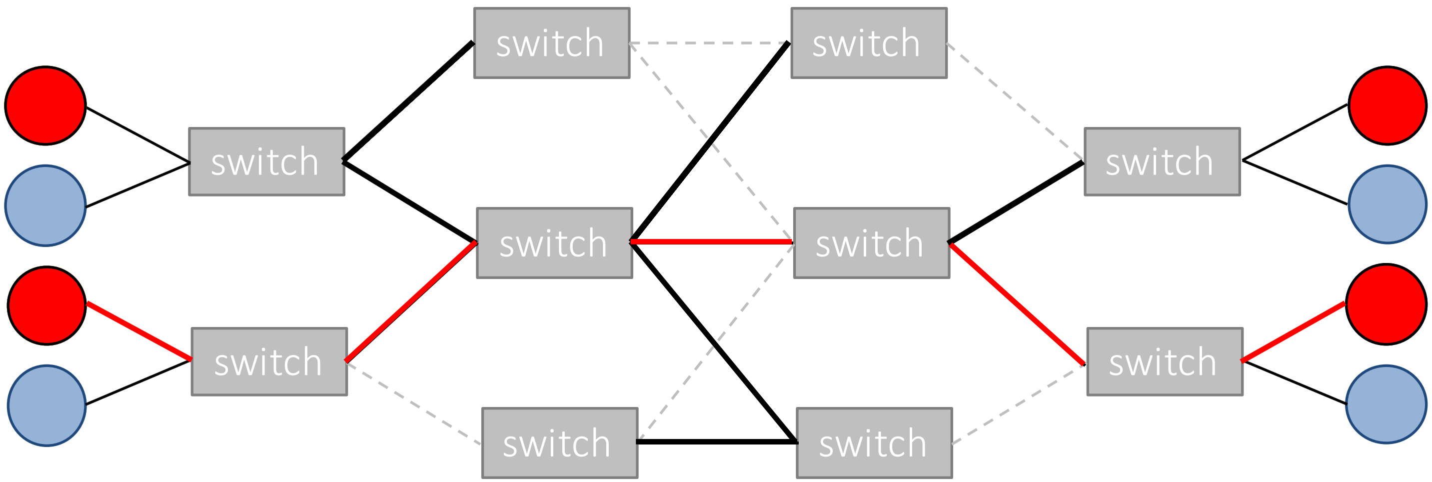
Assumptions valid?

MPTCP [IETF rfc 6824 '13]



Assumptions valid?

MPTCP [IETF rfc 6824 ’13]



Assumptions valid?

Single-homed MPTCP [IETF draft '14]

2.1. Exposing Multiple Paths Through End-host Auto-configuration

Multipath TCP distinguishes paths by their source and destination IP addresses. Assuming a certain level of path diversity in the Internet, using different source and destination IP addresses for a given subflow of a multipath TCP connection will, with a certain probability, result in different paths taken by packets of different subflows. Even in case subflows share a common bottleneck, the proposed multipath congestion control algorithm [RFC6356] will make sure that multipath TCP will play nicely with regular TCP flows.

Issue a network interface multiple addresses. Assume configuration will result in multiple paths.