

In-Network Computing to the rescue of Faulty Links

Hans Giesen

University of Pennsylvania

Lei Shi

University of Pennsylvania

John Sonchack

University of Pennsylvania

Anirudh Chelluri

University of Pennsylvania

Nishanth Prabhu

University of Pennsylvania

~~Nik Sultana~~

University of Pennsylvania

Latha Kant

Perspecta Labs

Anthony J McAuley

Perspecta Labs

Alexander Poylisher

Perspecta Labs

André DeHon

University of Pennsylvania

Boon Thau Loo

University of Pennsylvania

Acknowledgements: Isaac Pedisich (UPenn), Gordon Brebner (Xilinx),
DARPA Contracts No. HR0011-17-C-0047 and HR0011-16-C-0056, and NSF grant CNS-1513679.





Packet loss ->³ Application malfunction



Congestion
Unstable, various mitigations



Congestion



5

Corruption

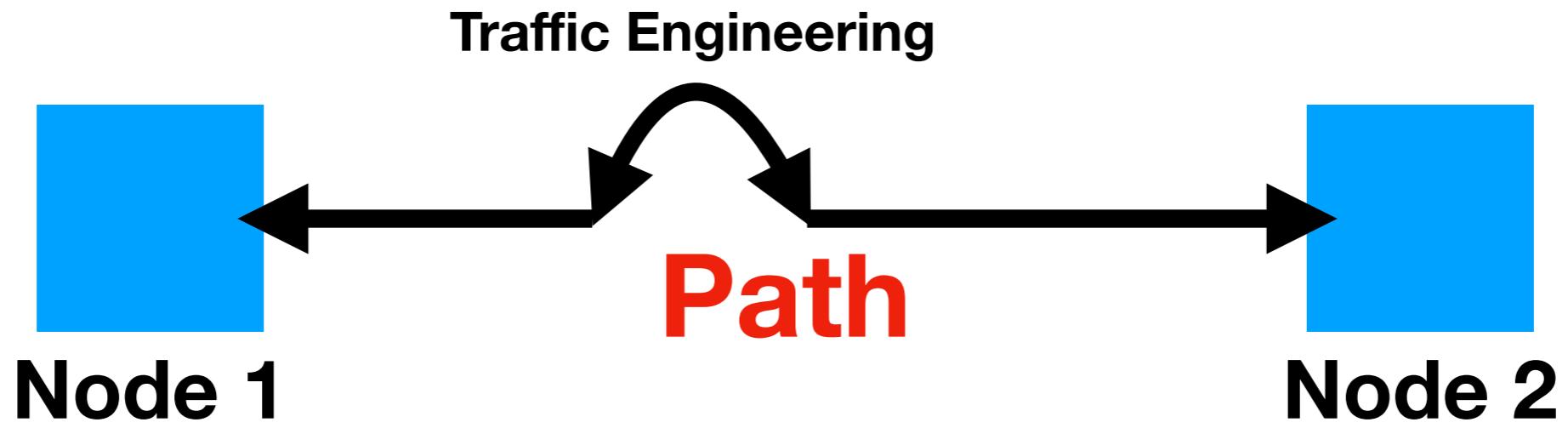
Stable, rerouting mitigation
+ replacement



Congestion



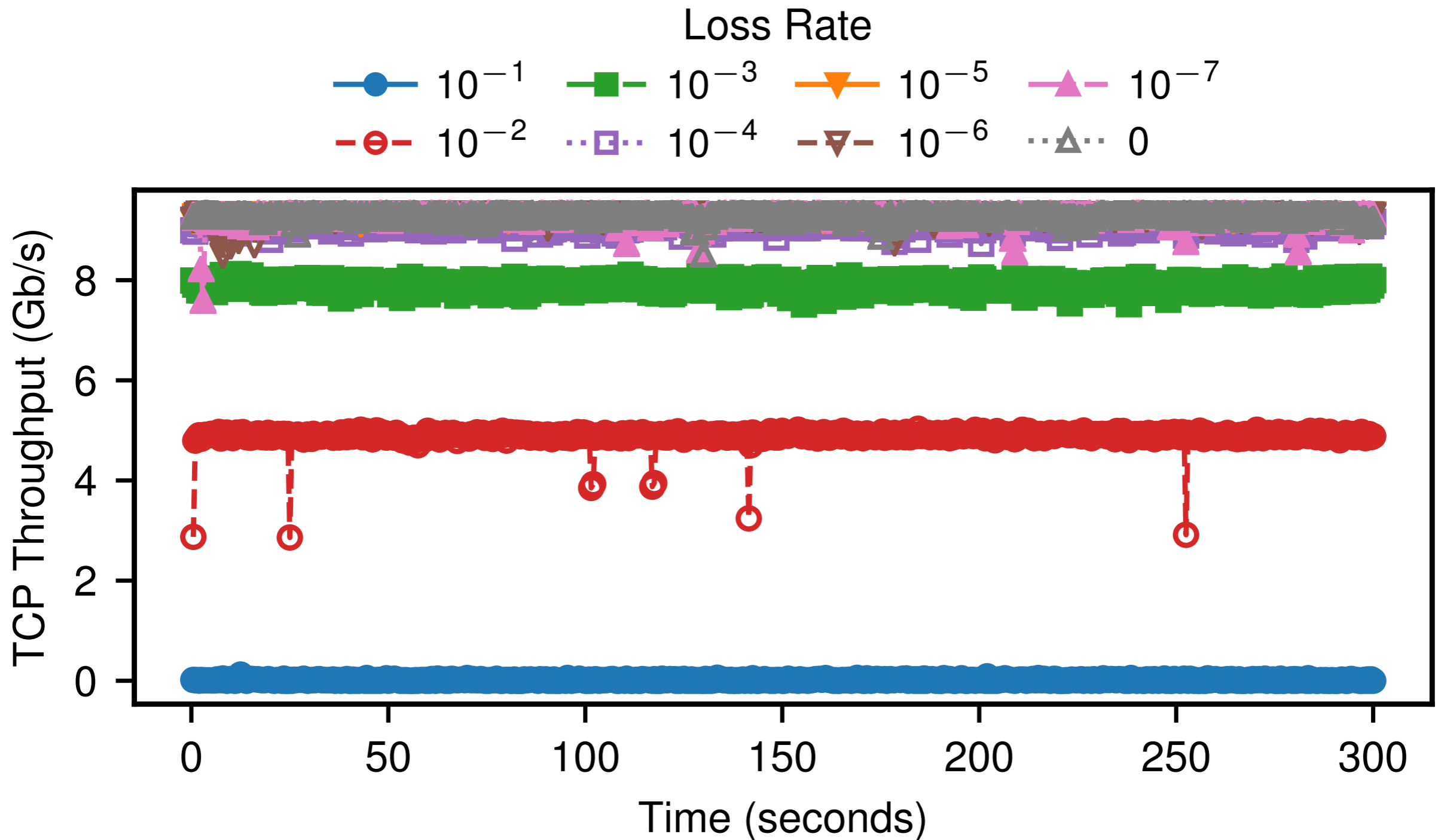
Corruption



Congestion  Corruption

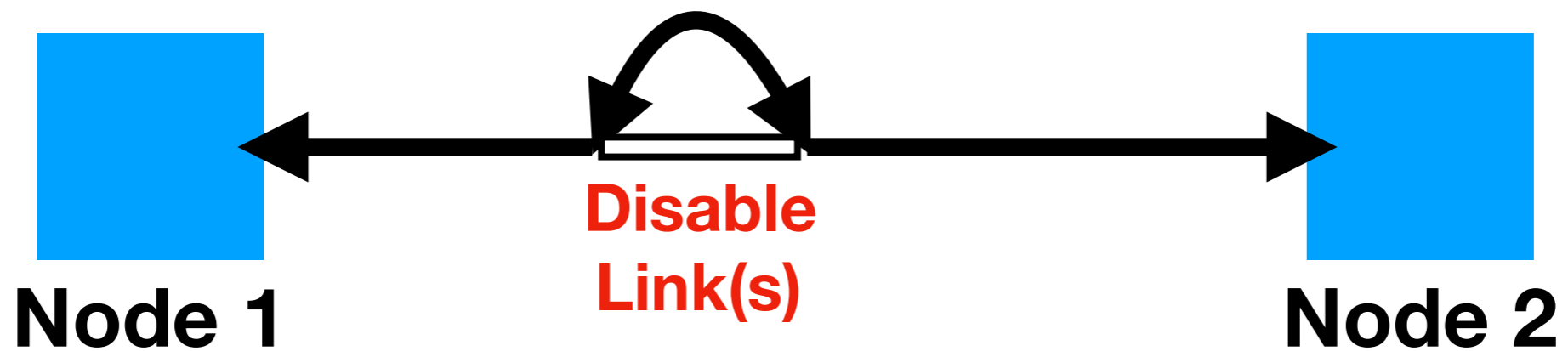
The text 'Congestion' is in grey, 'Corruption' is in orange, and a blue toilet icon is positioned between them.

Loss and TCP th'put



Loss disproportionate to corruption

Current solution

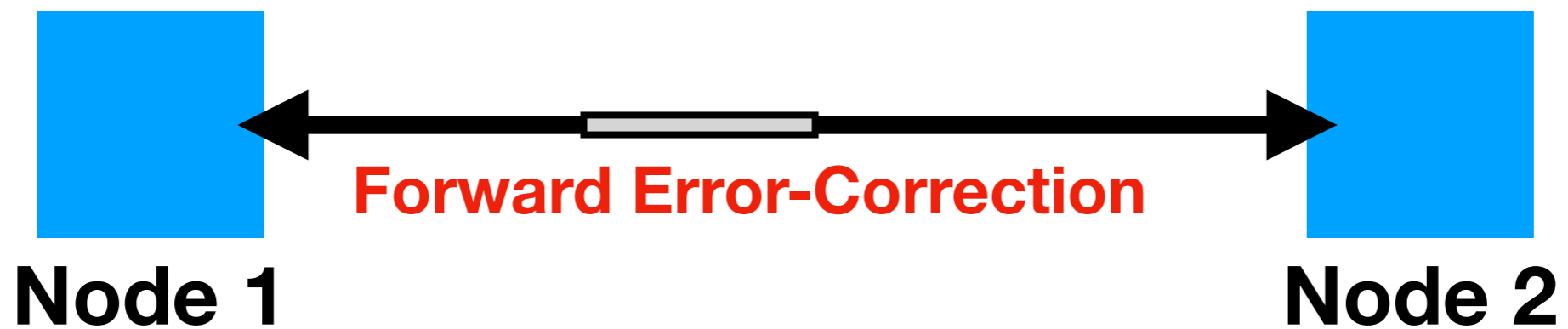


Congestion



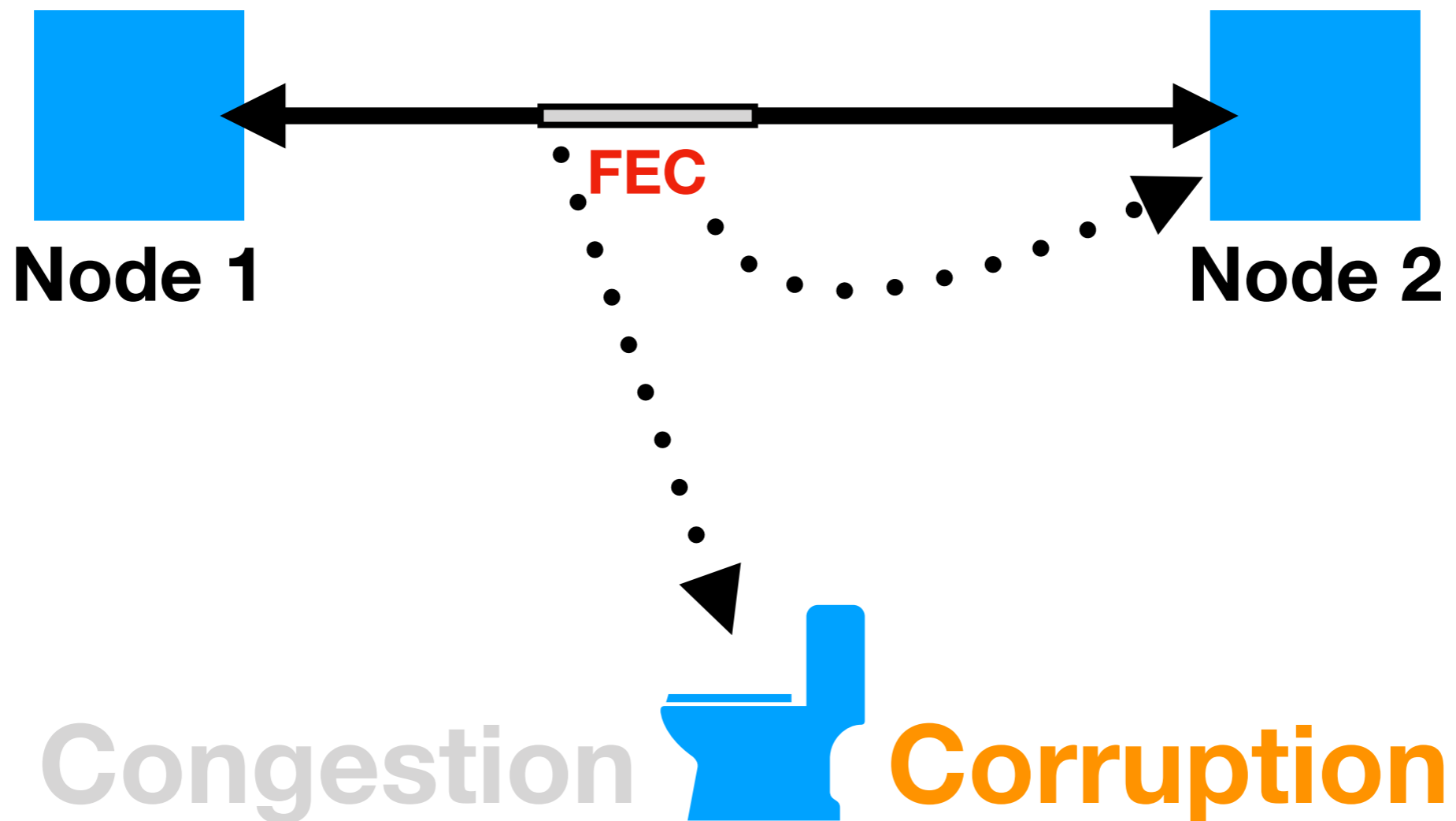
Corruption

This talk

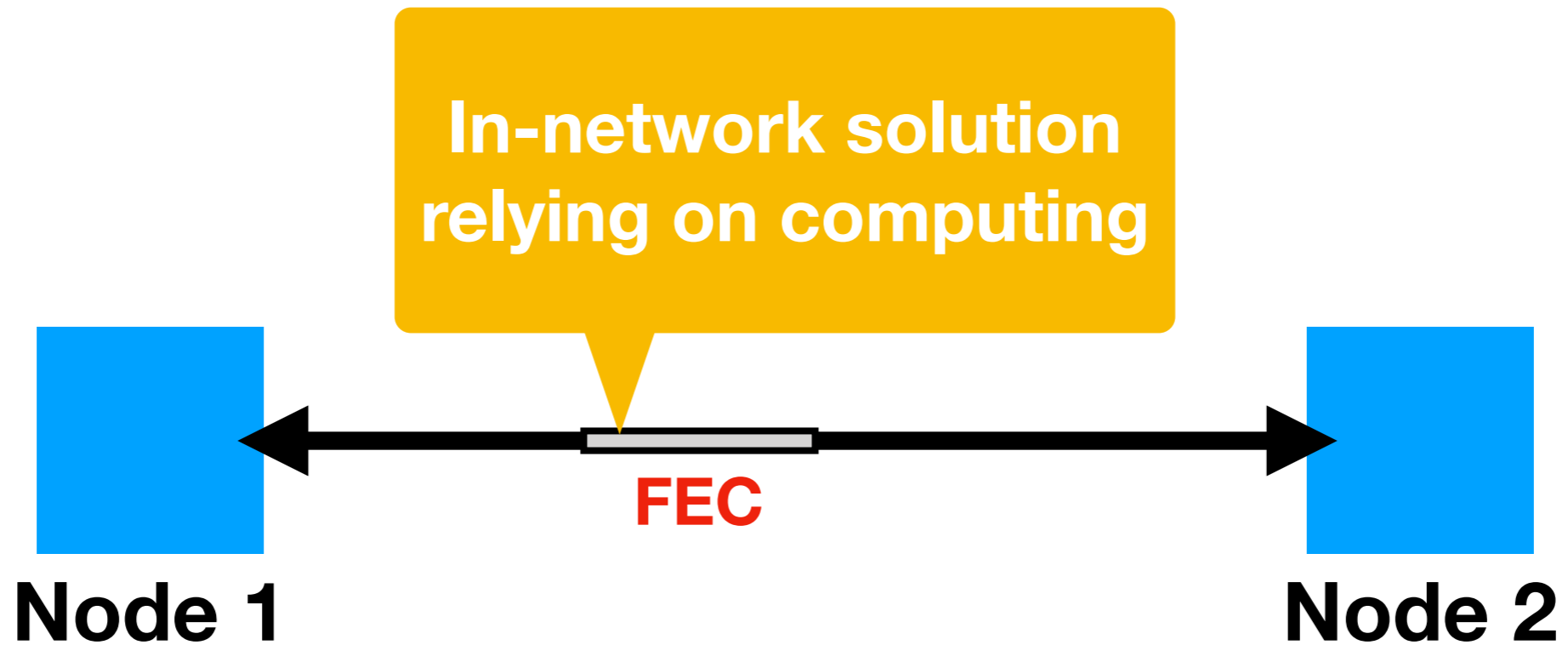


Congestion  Corruption

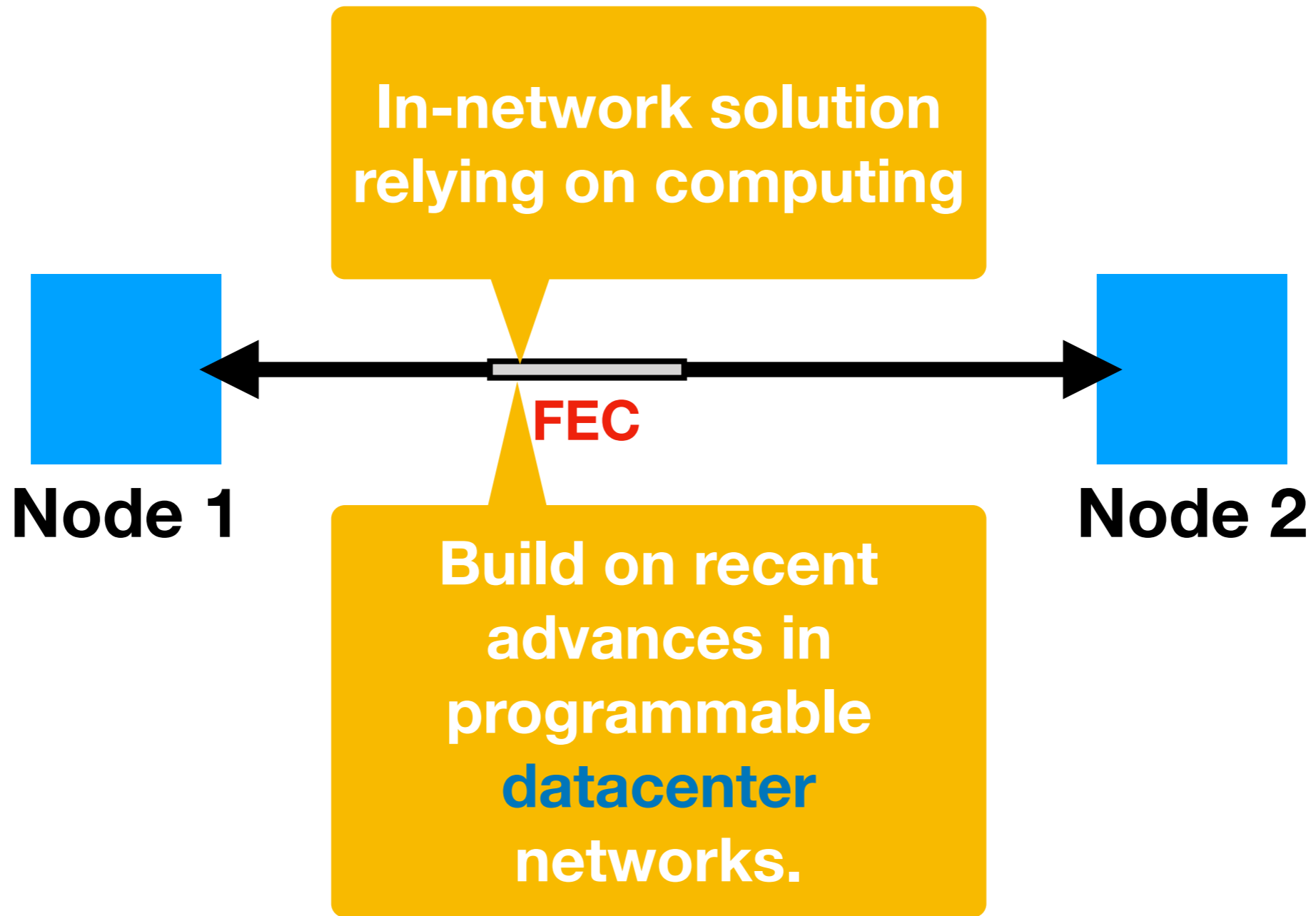
This talk



This talk



This talk



Design goals

- **Transparent** to rest of the network
- Low **overhead** (beyond the FEC overhead)
- Low complexity **activation**: between adjacent elements about whether to activate FEC
- Support for different **traffic classes** (affecting **latency** and **redundancy**)

Where to decide?



Central

(Single element decides
for other elements' links)



Distributed

(Each element sees to its own links.
Faster reaction time)

What to do?



Repeat

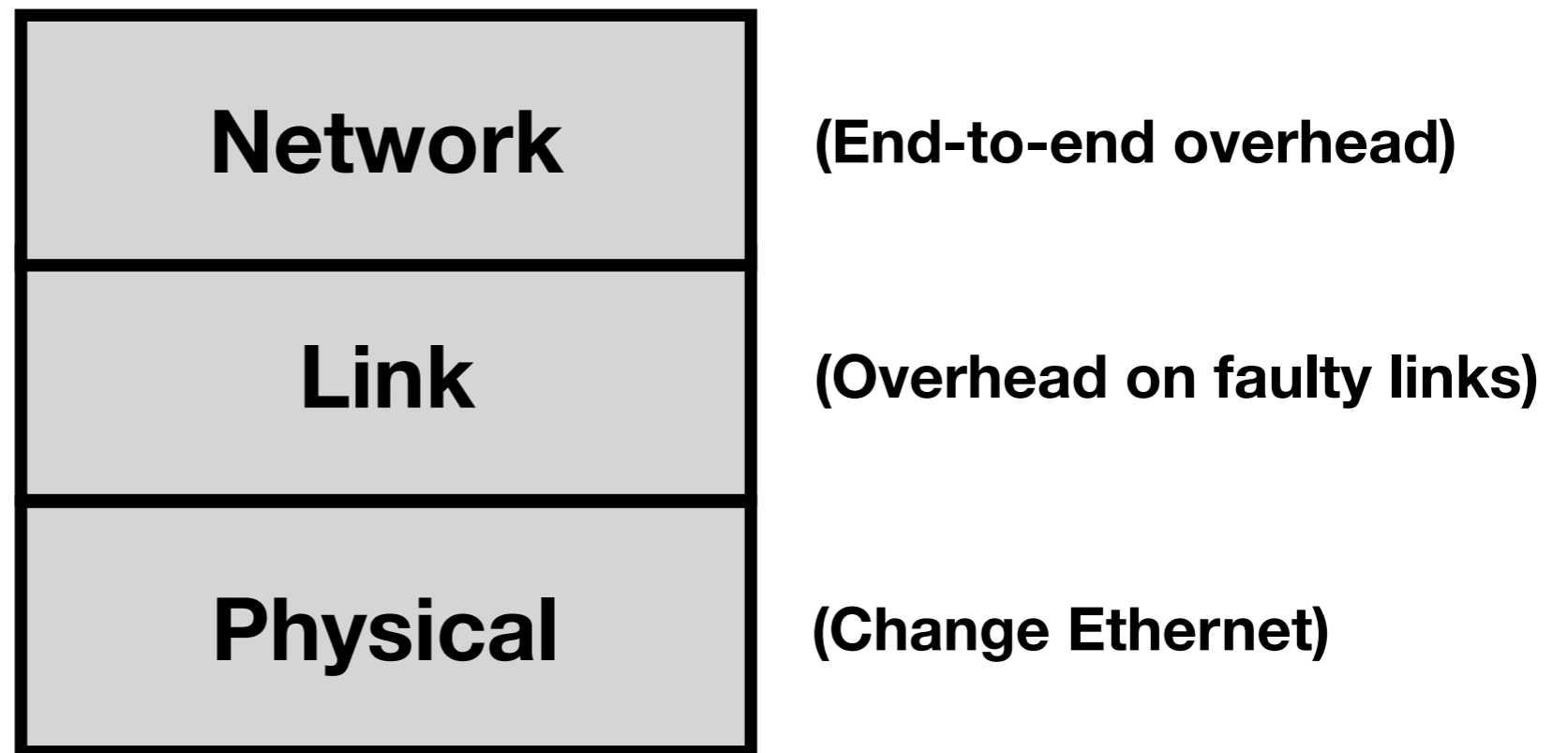
(Resend information)



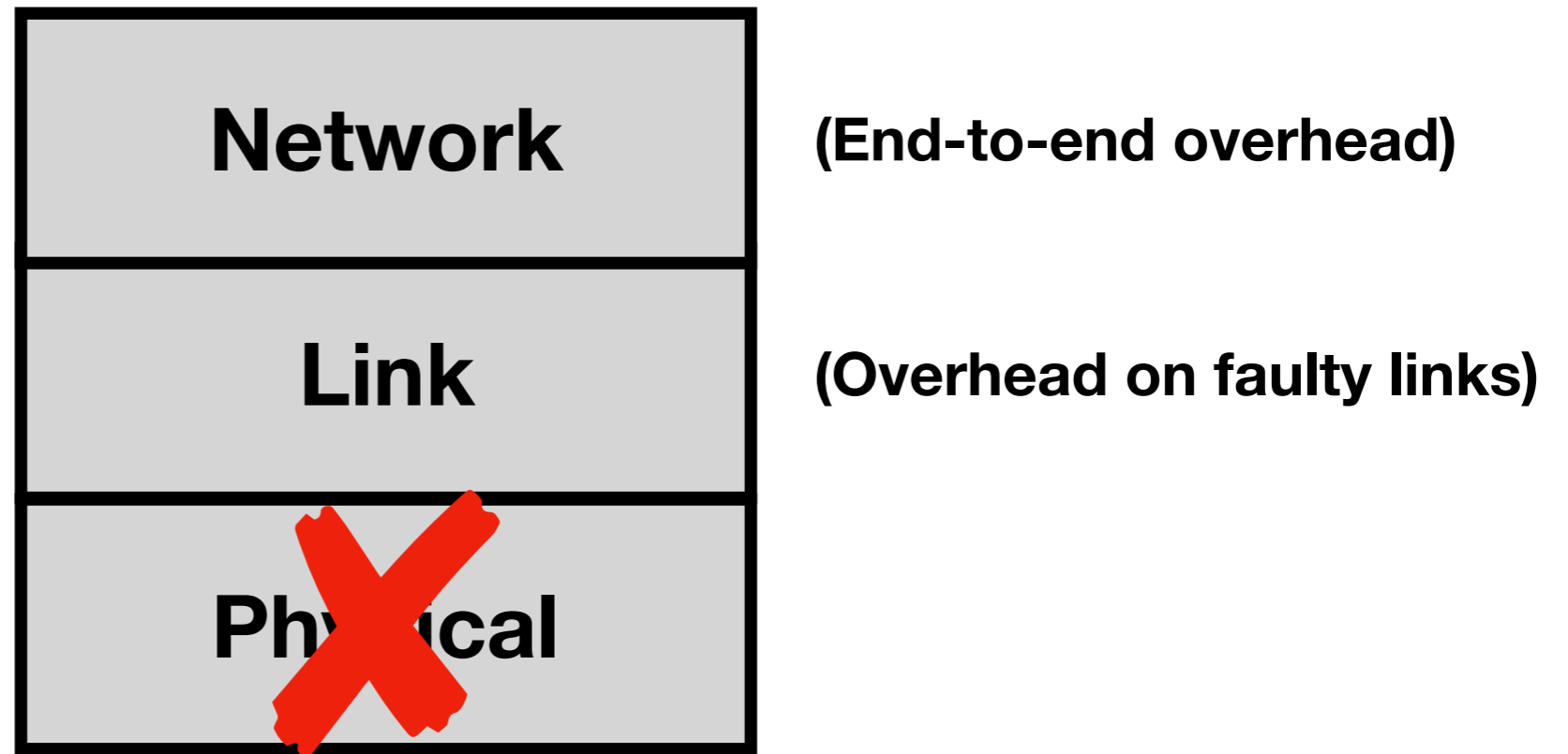
Redund

(In the hope more info gets through)

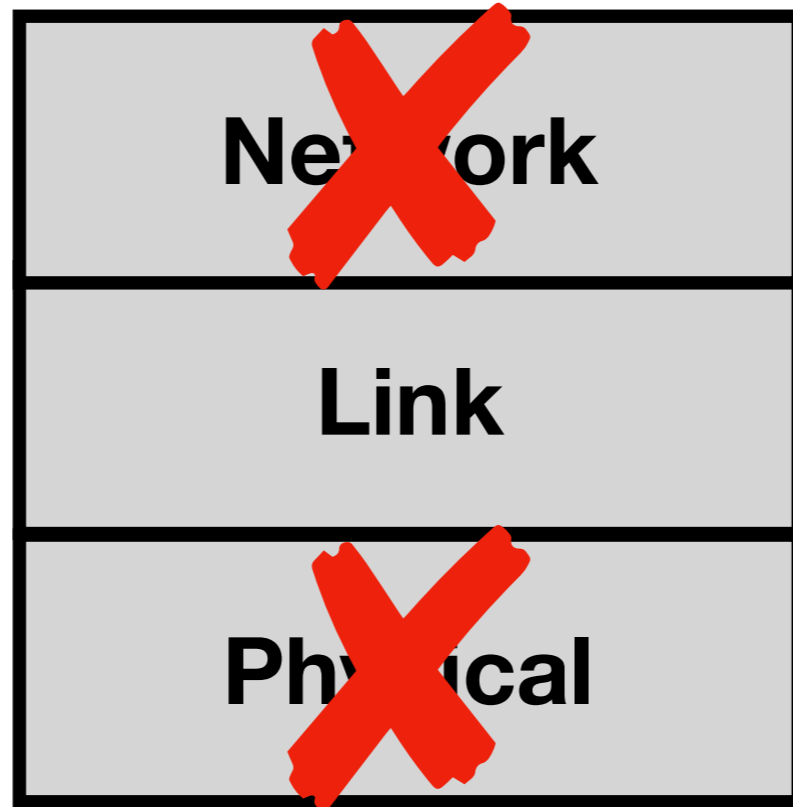
What layer FEC?



What layer FEC?

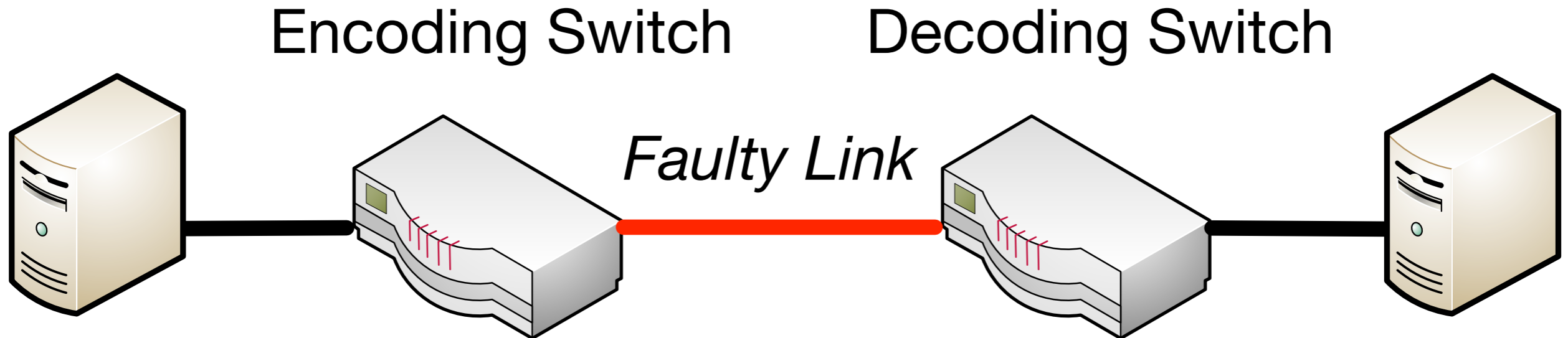


What layer FEC?



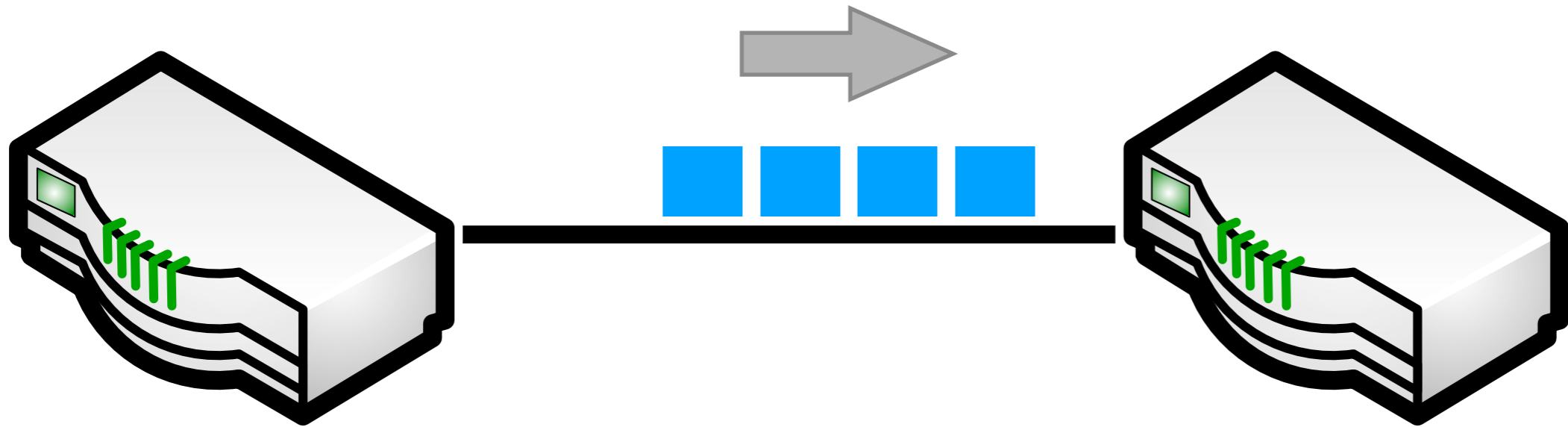
(Overhead on faulty links)

LL FEC

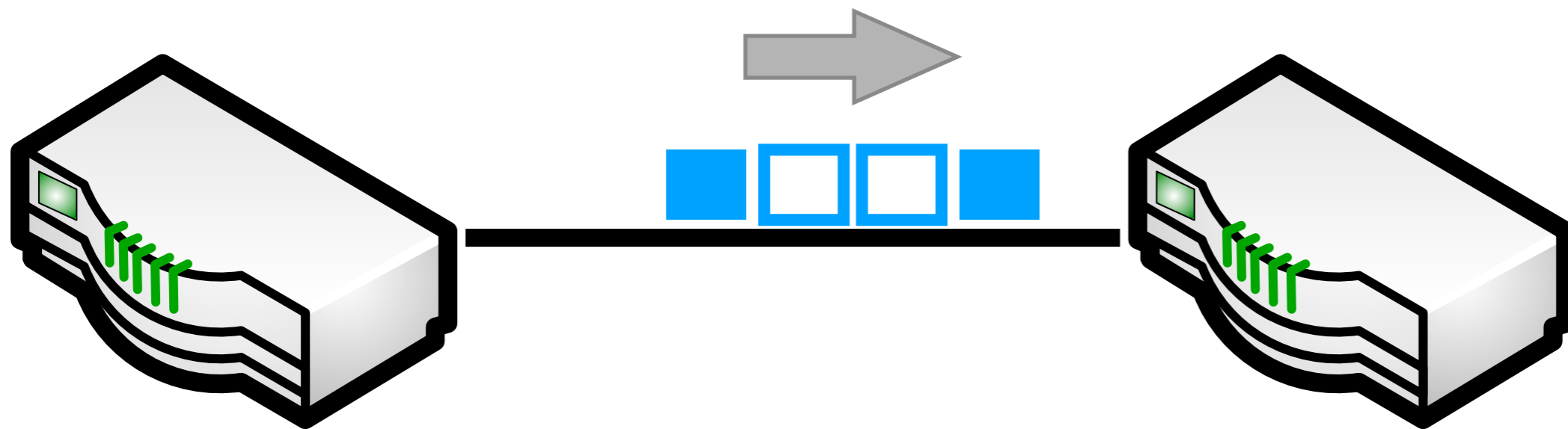


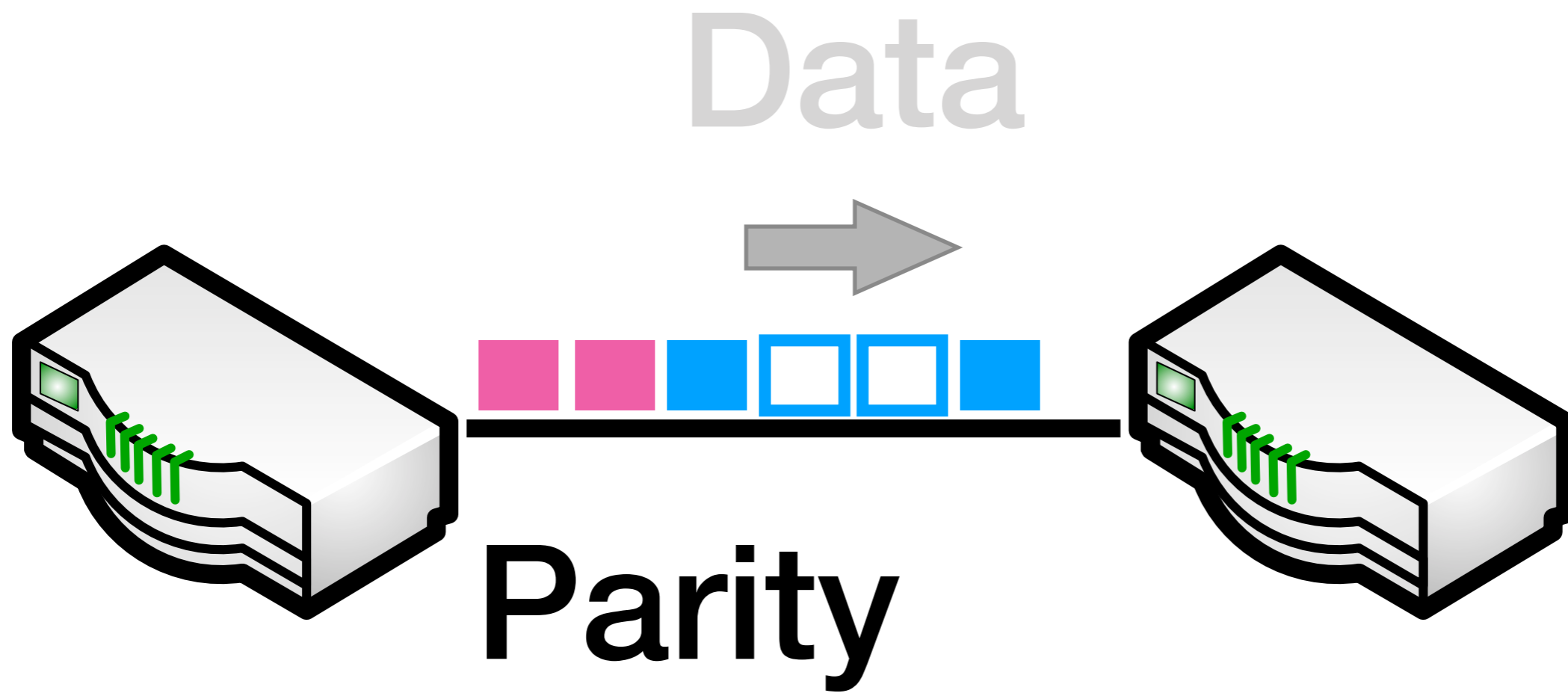
(But design could work on non-switch-to-switch links)

Data

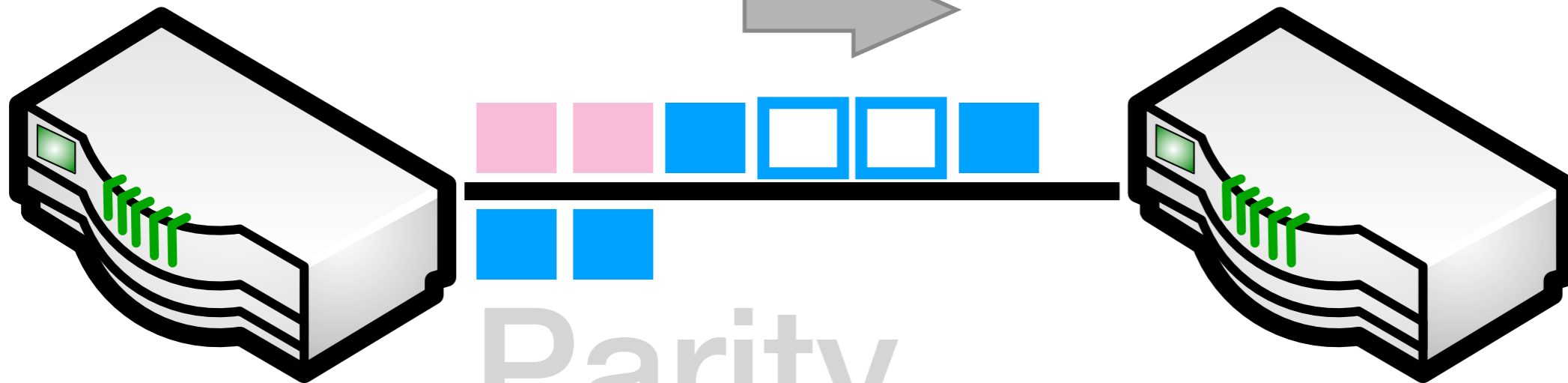
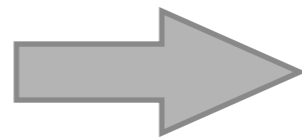


Data

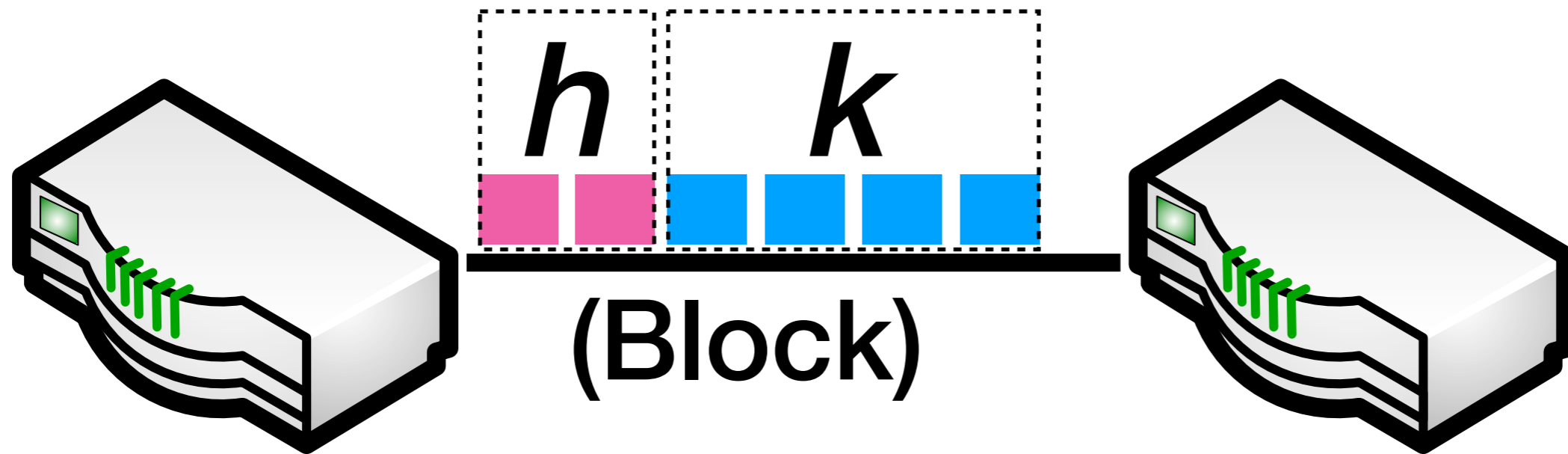


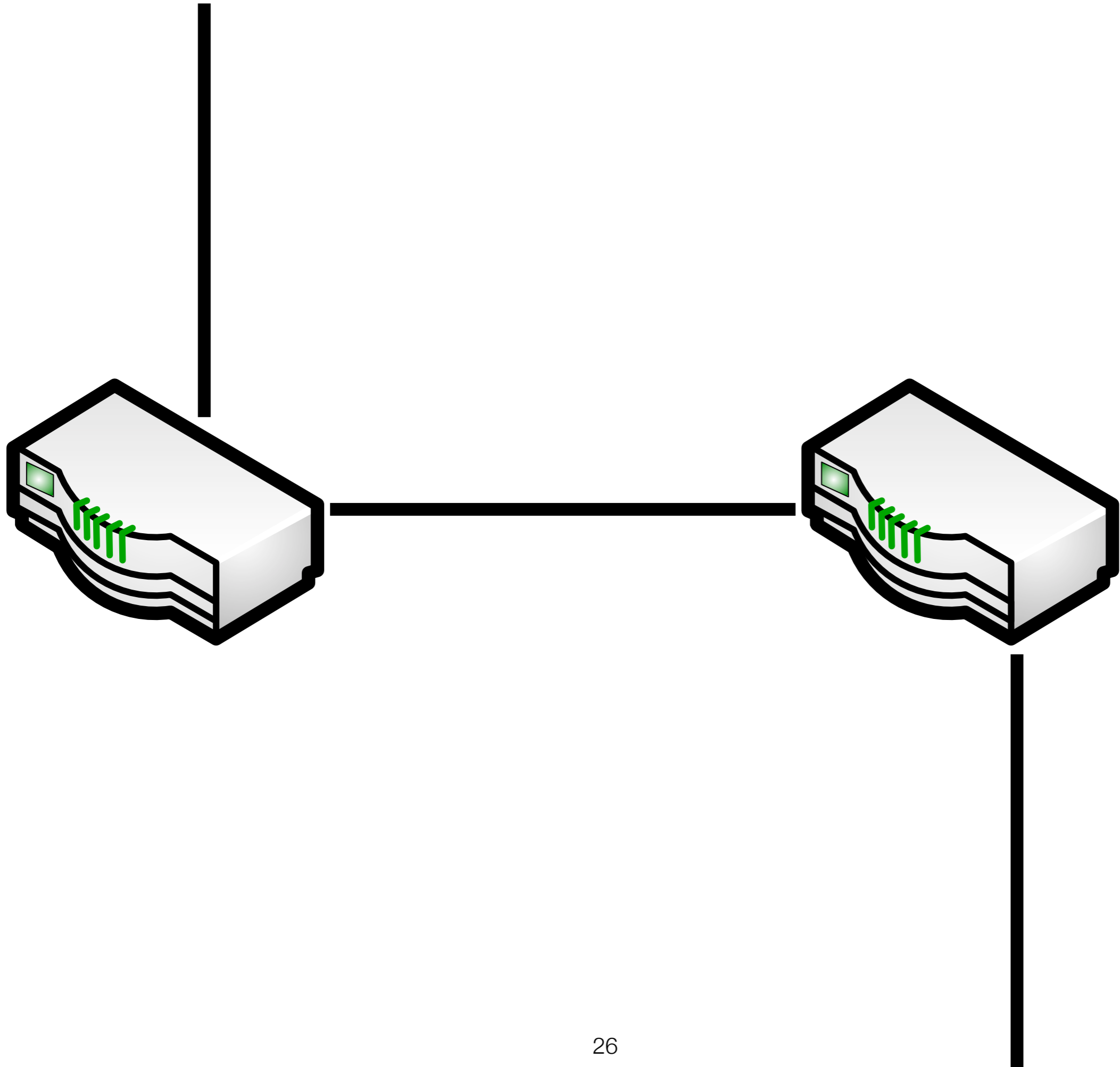


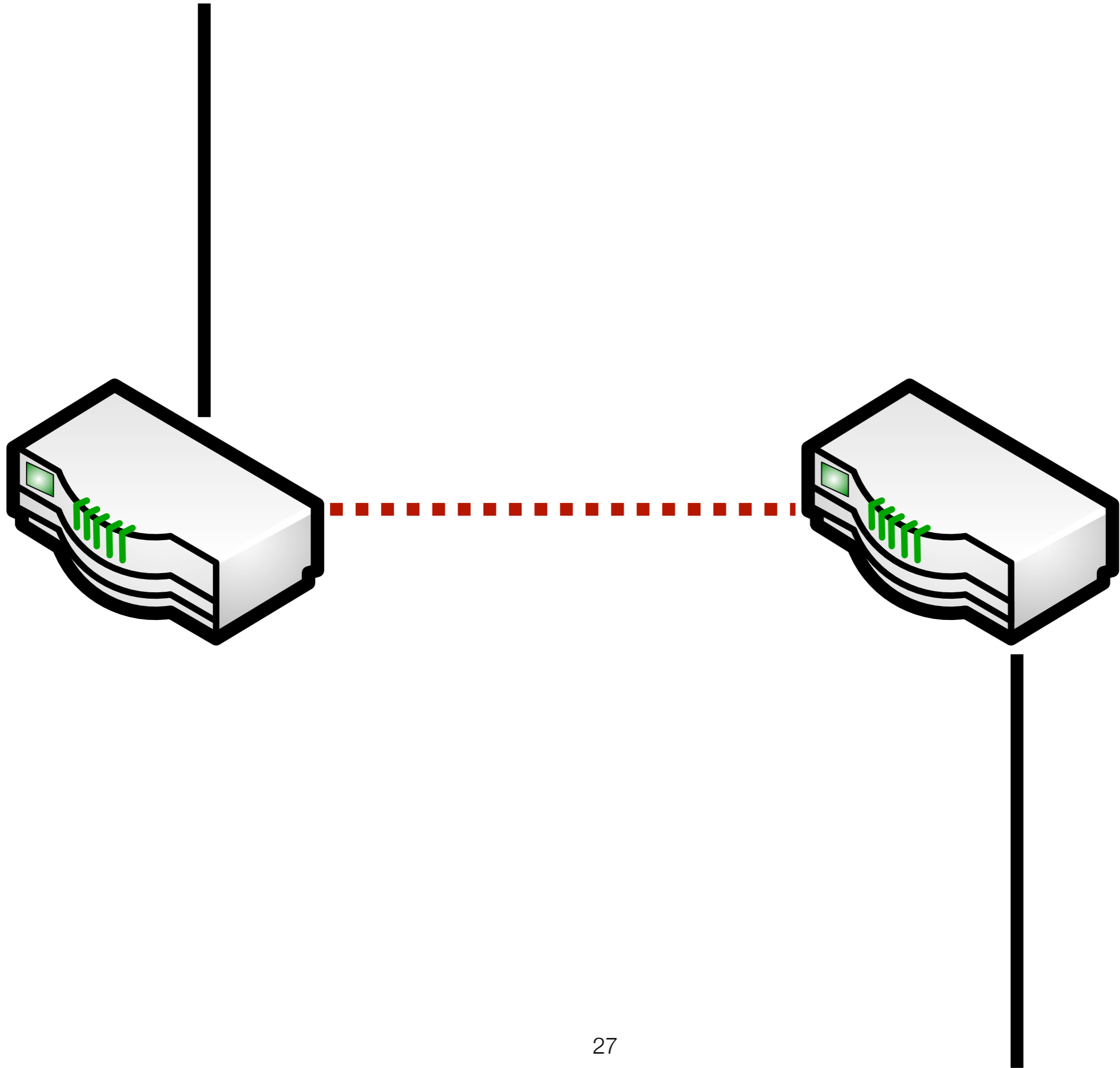
Data

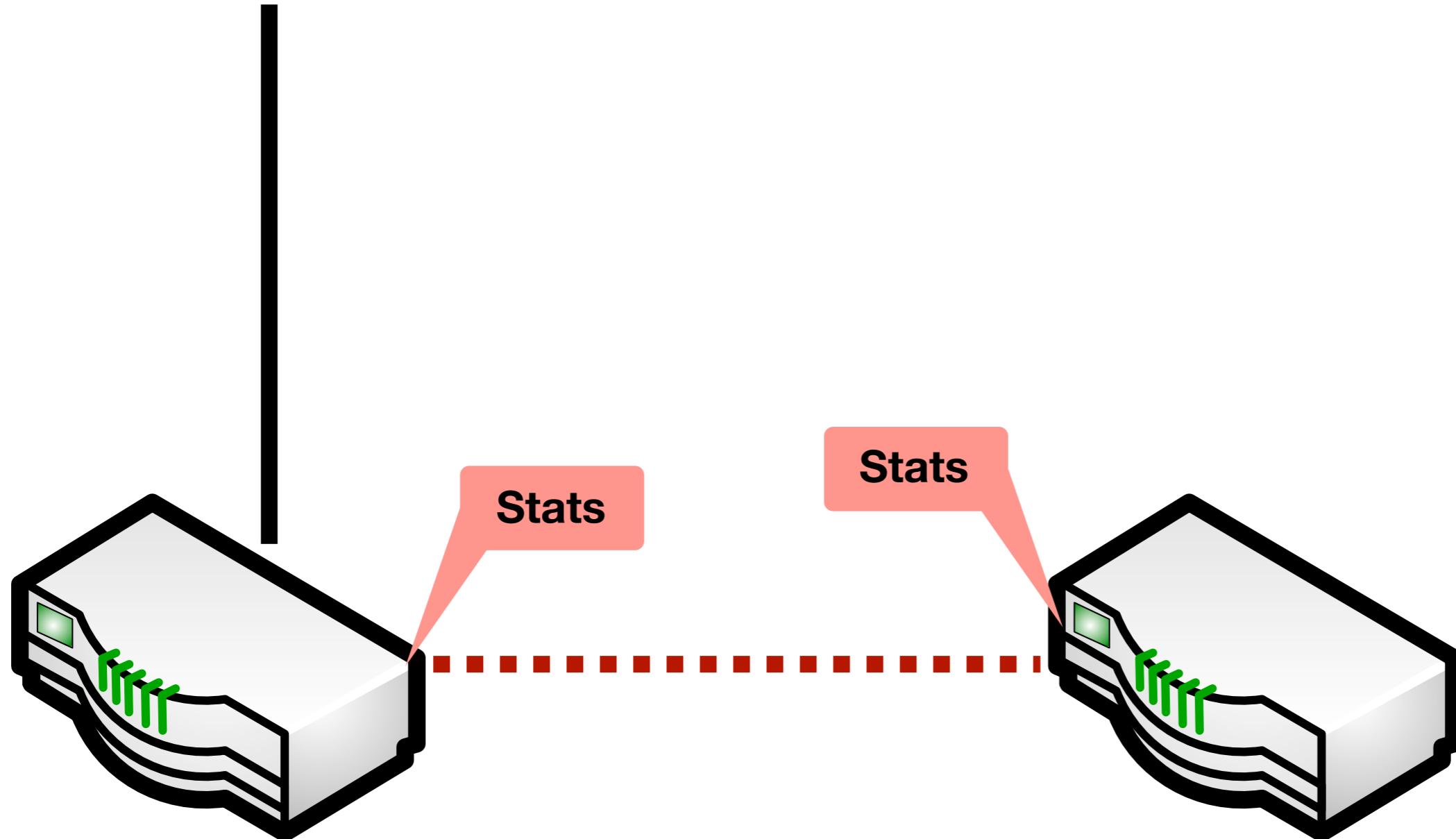


Parity

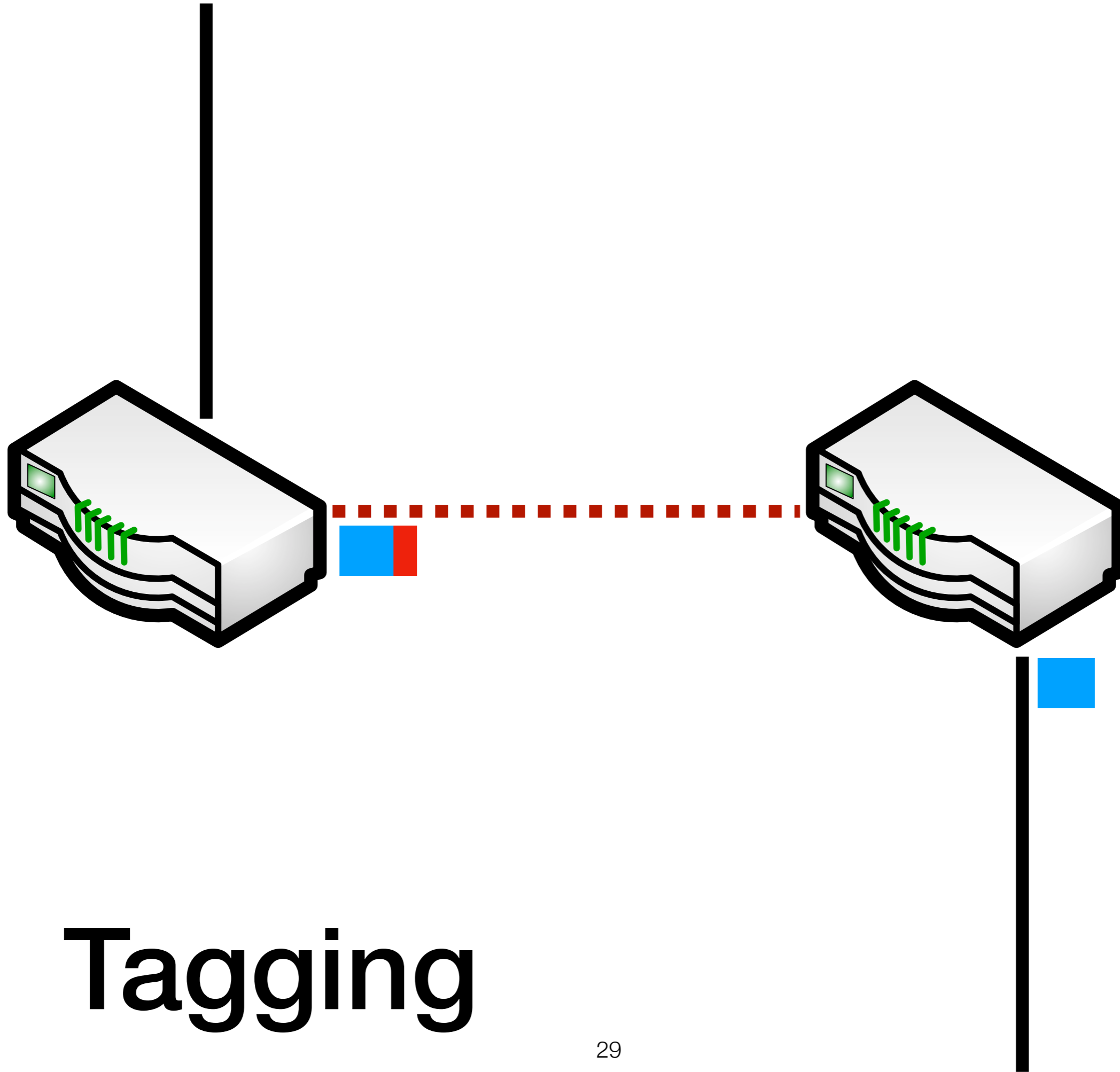




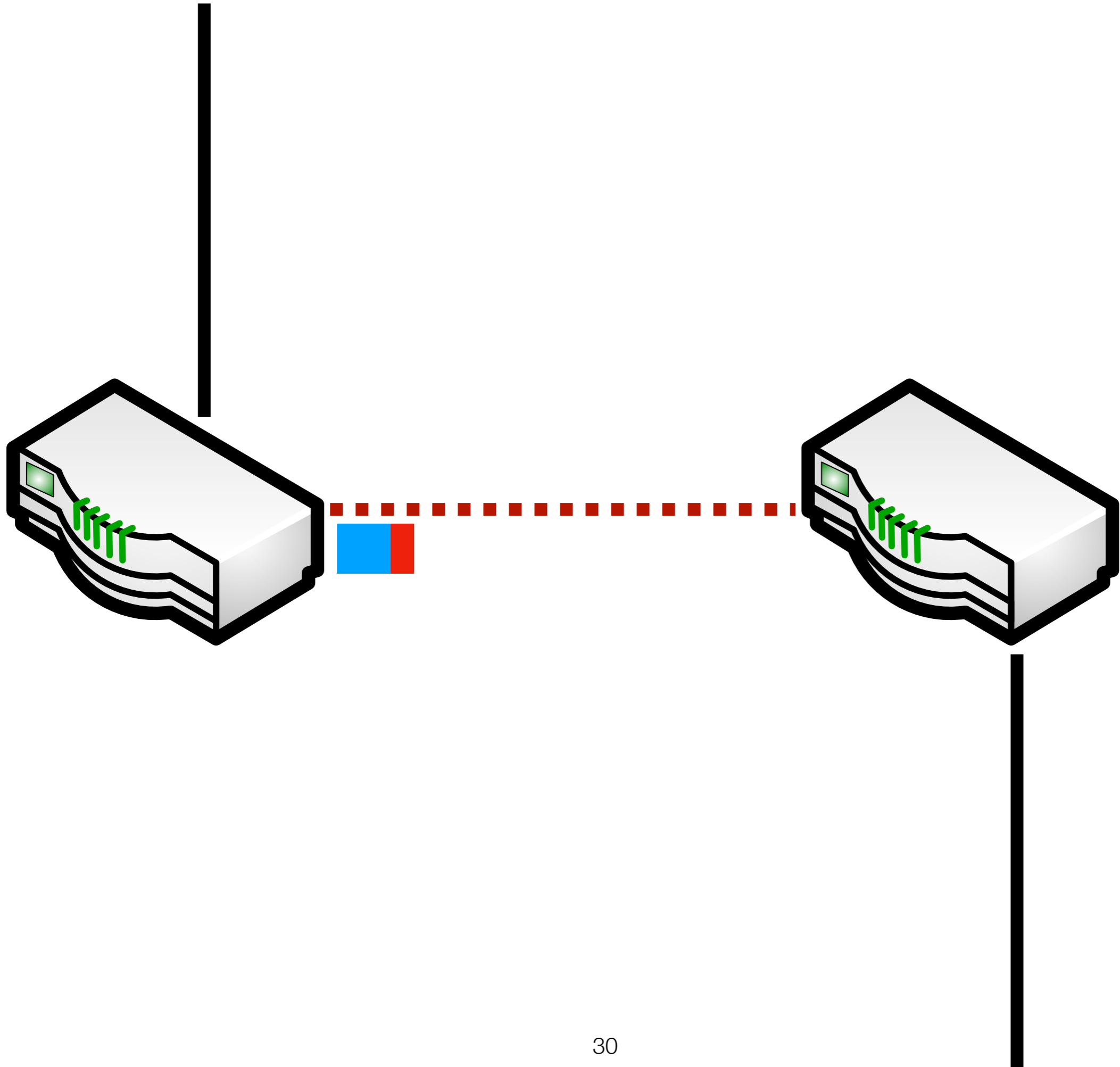


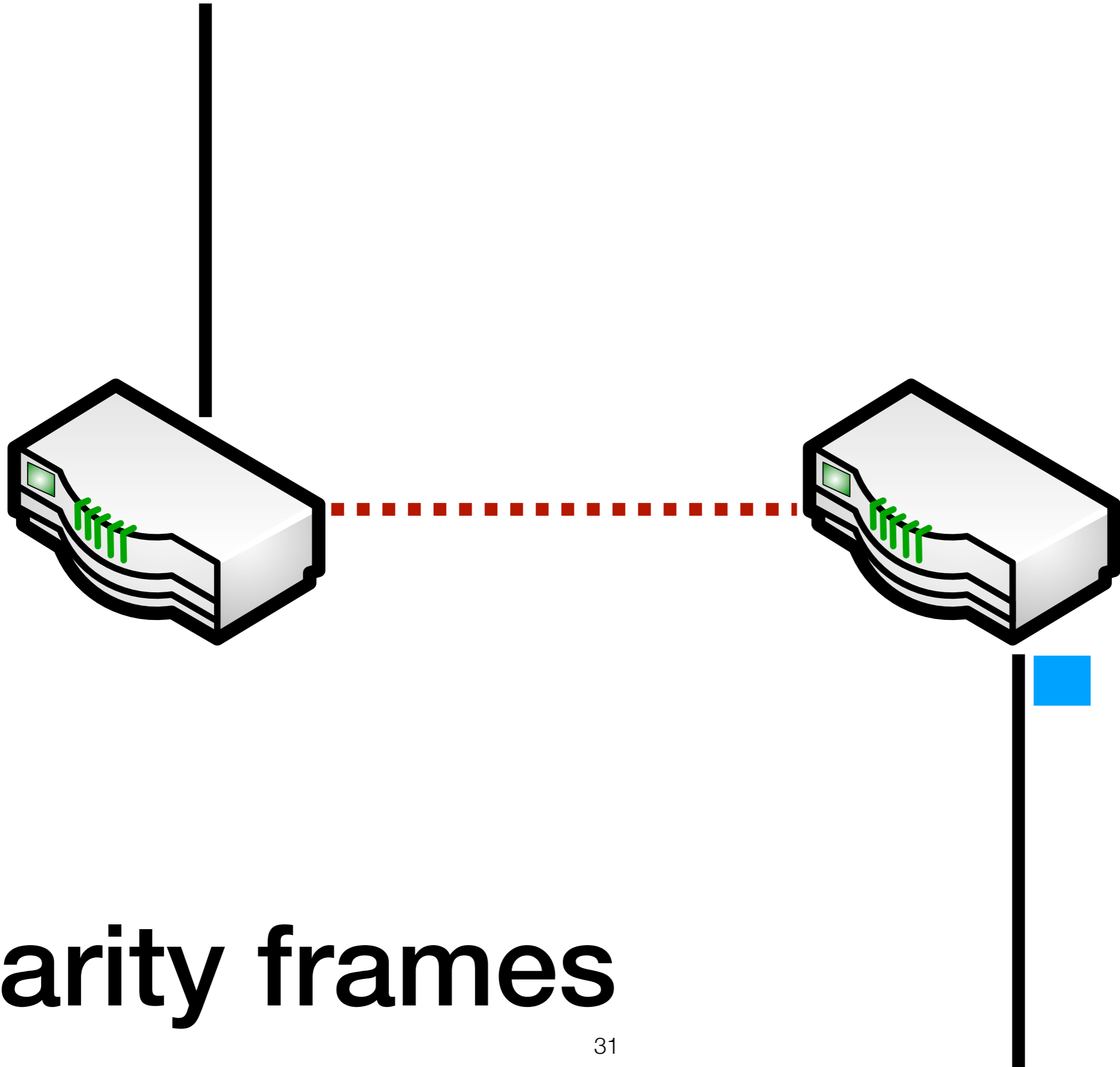


(see paper)

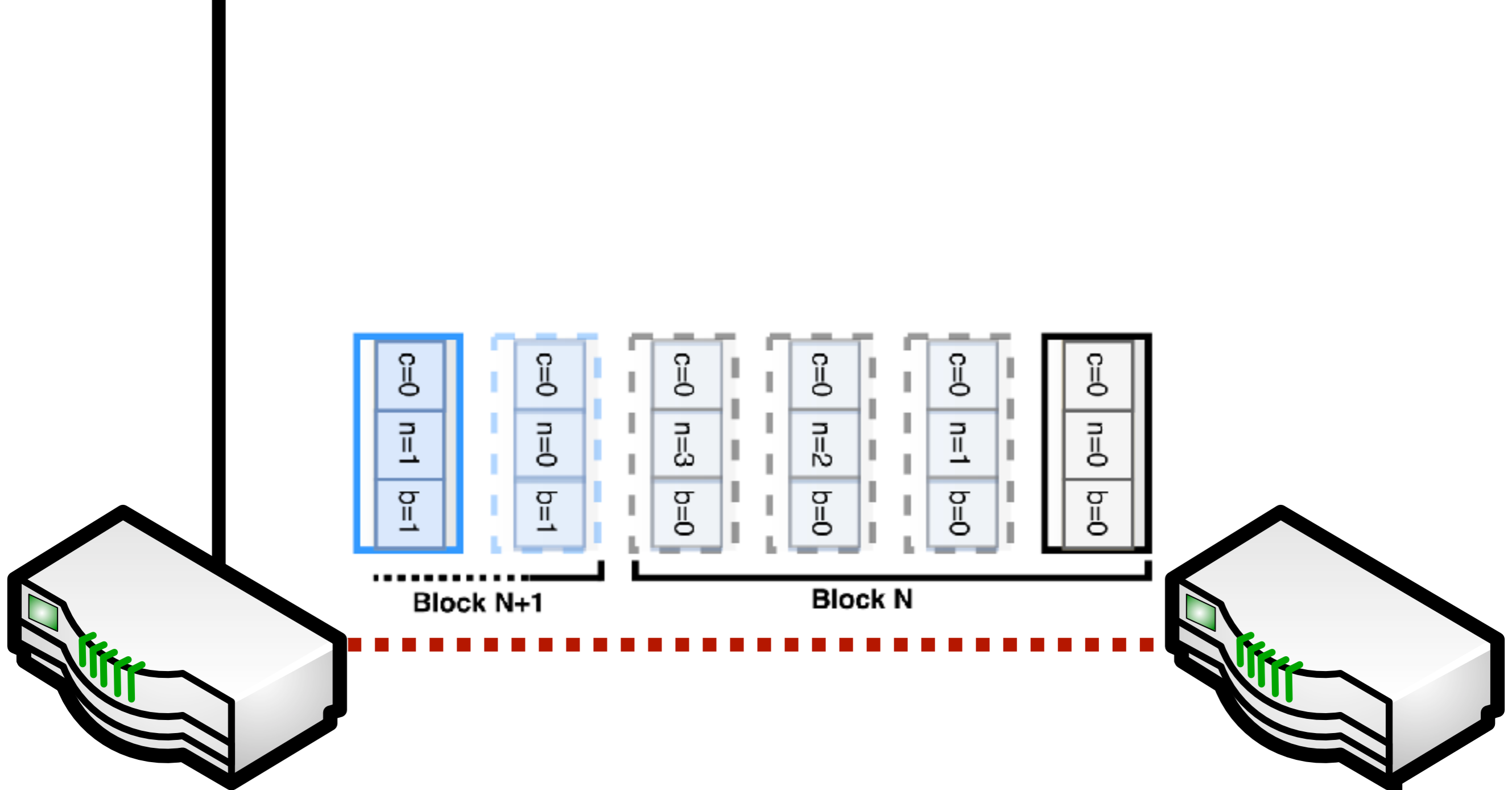


Tagging

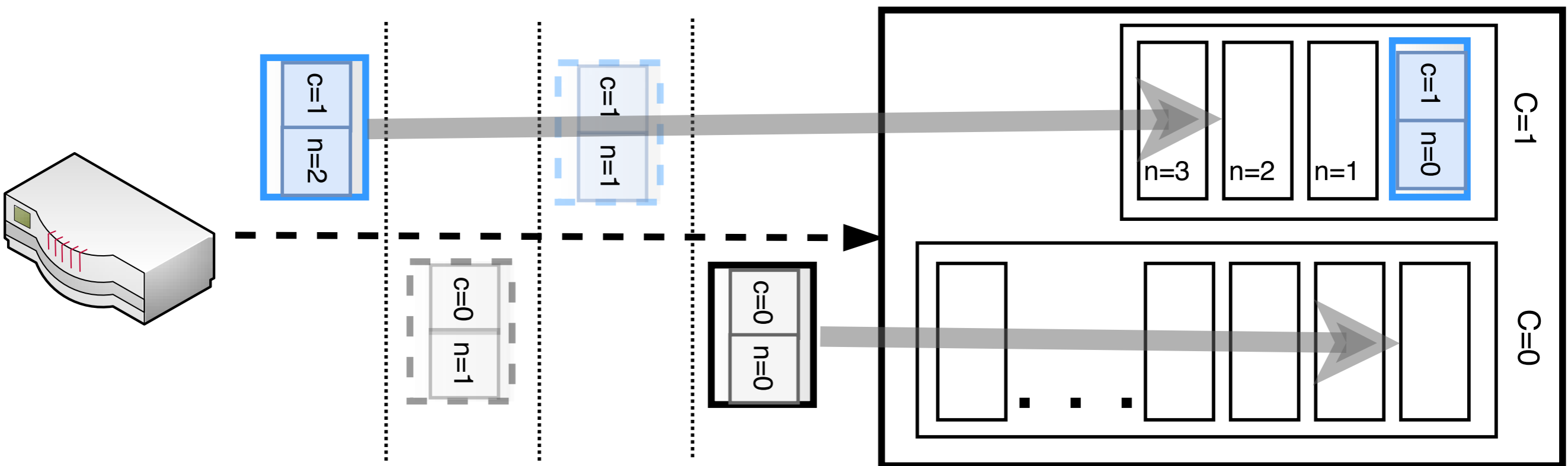




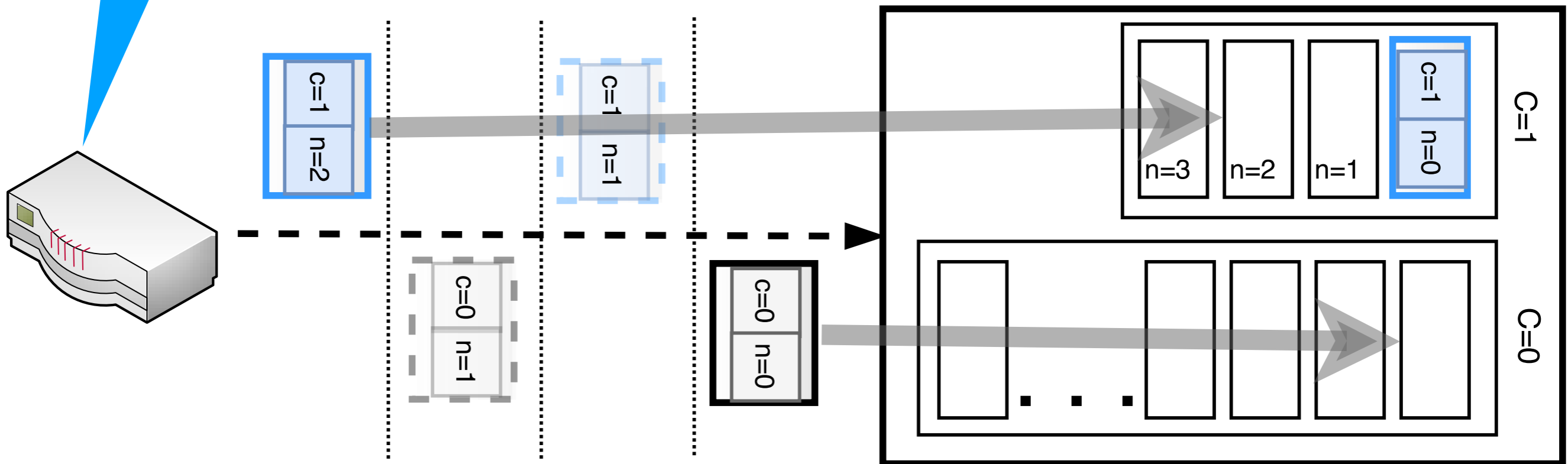
Parity frames



**1 block = k data frames
+ h parity frames**

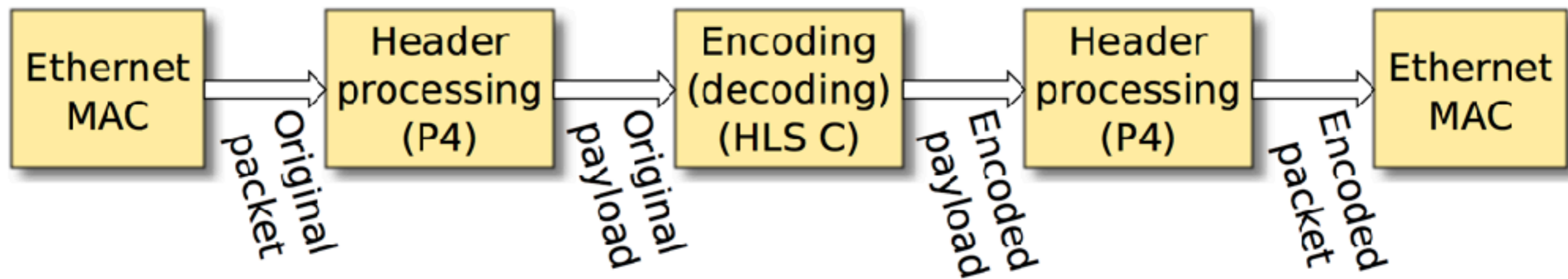


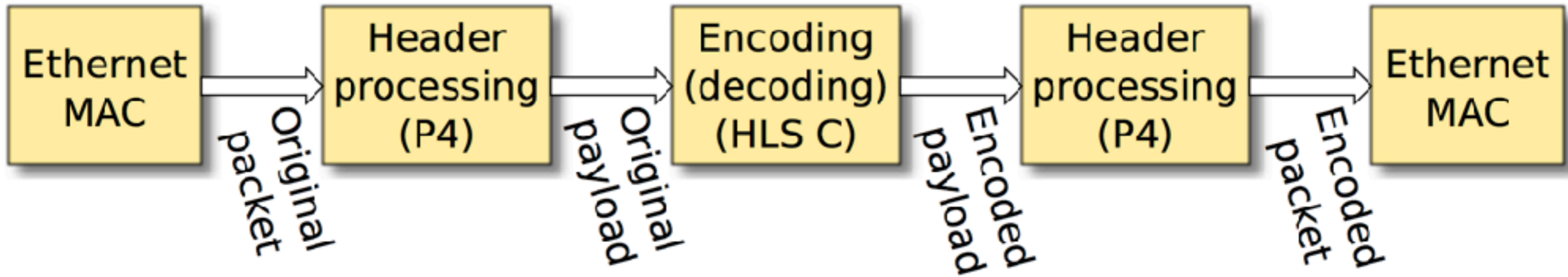
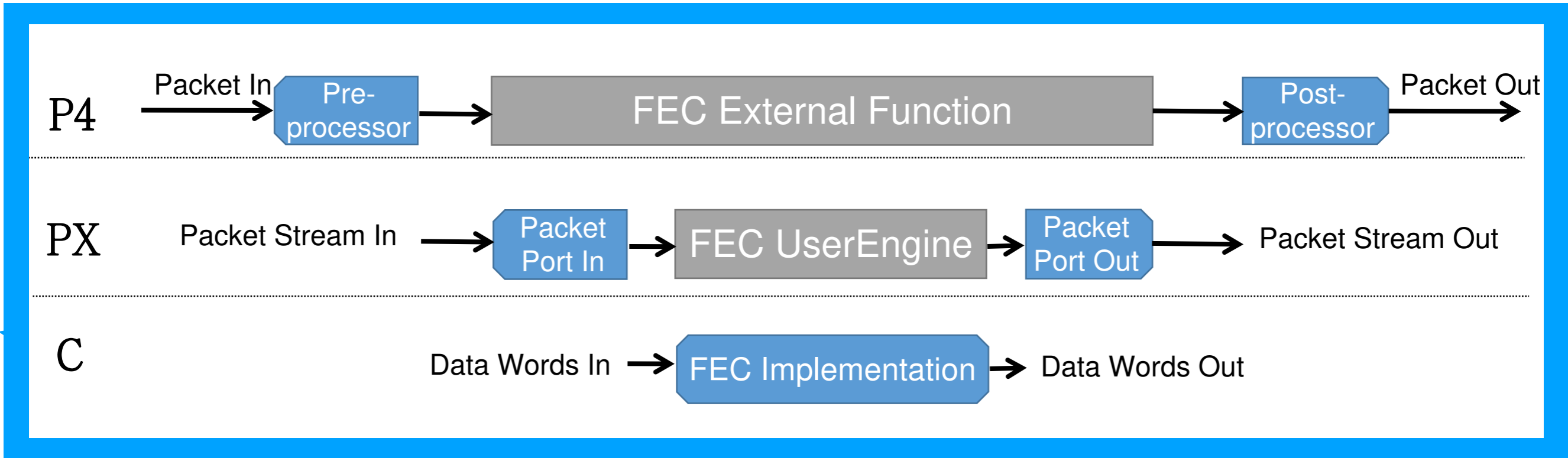
Traffic classification: protocol+port
(Configured by network controller)



Implementation

- High-level logic in P4 (e.g., traffic classification)
 - Two toolchains: Xilinx's SDNet and P4's p4c-BMv2
- External logic in C, targeting both FPGA board (Xilinx ZCU102) and CPU (x86)
- Work-in-progress: stats gathering, hardware decoding.

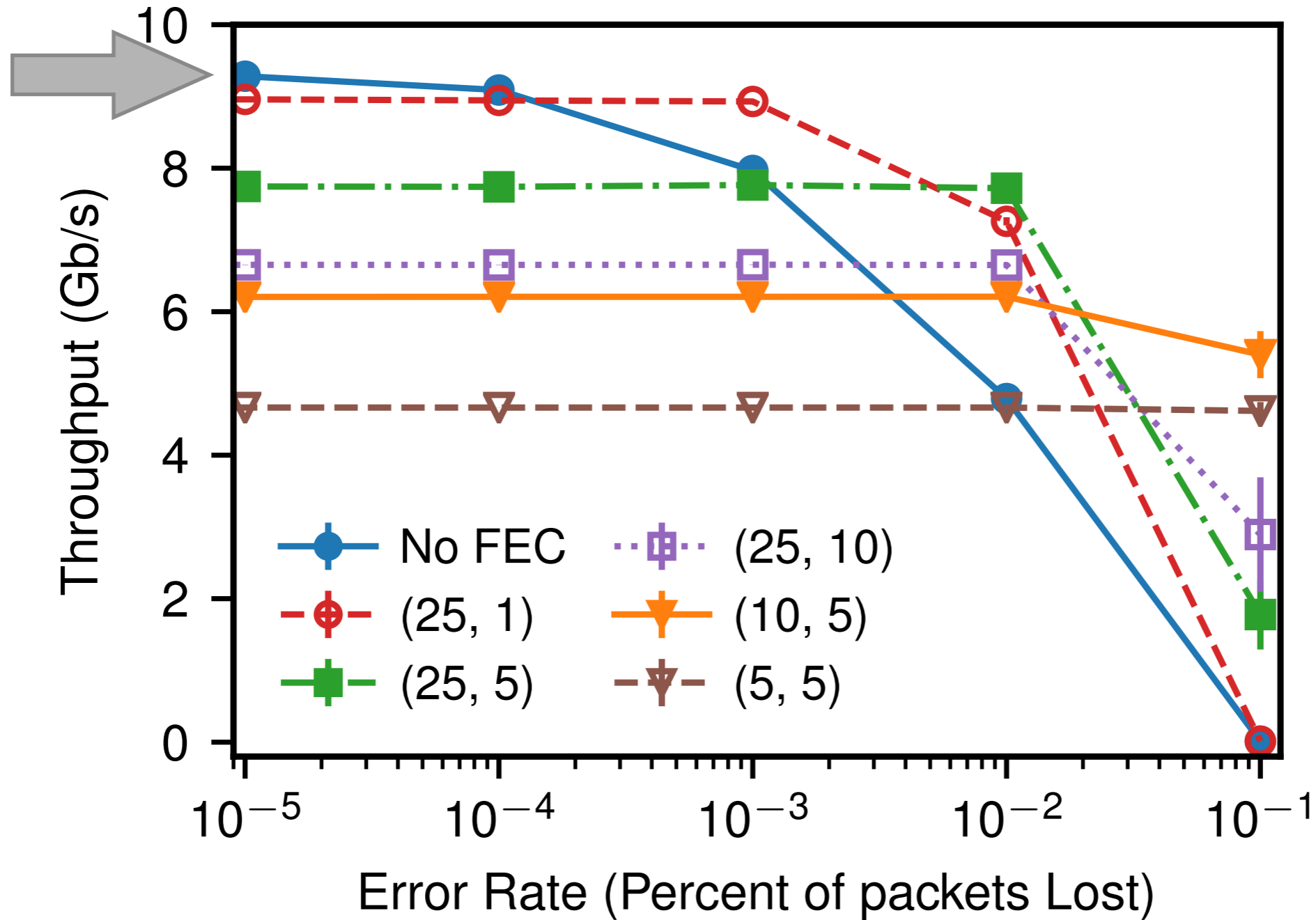




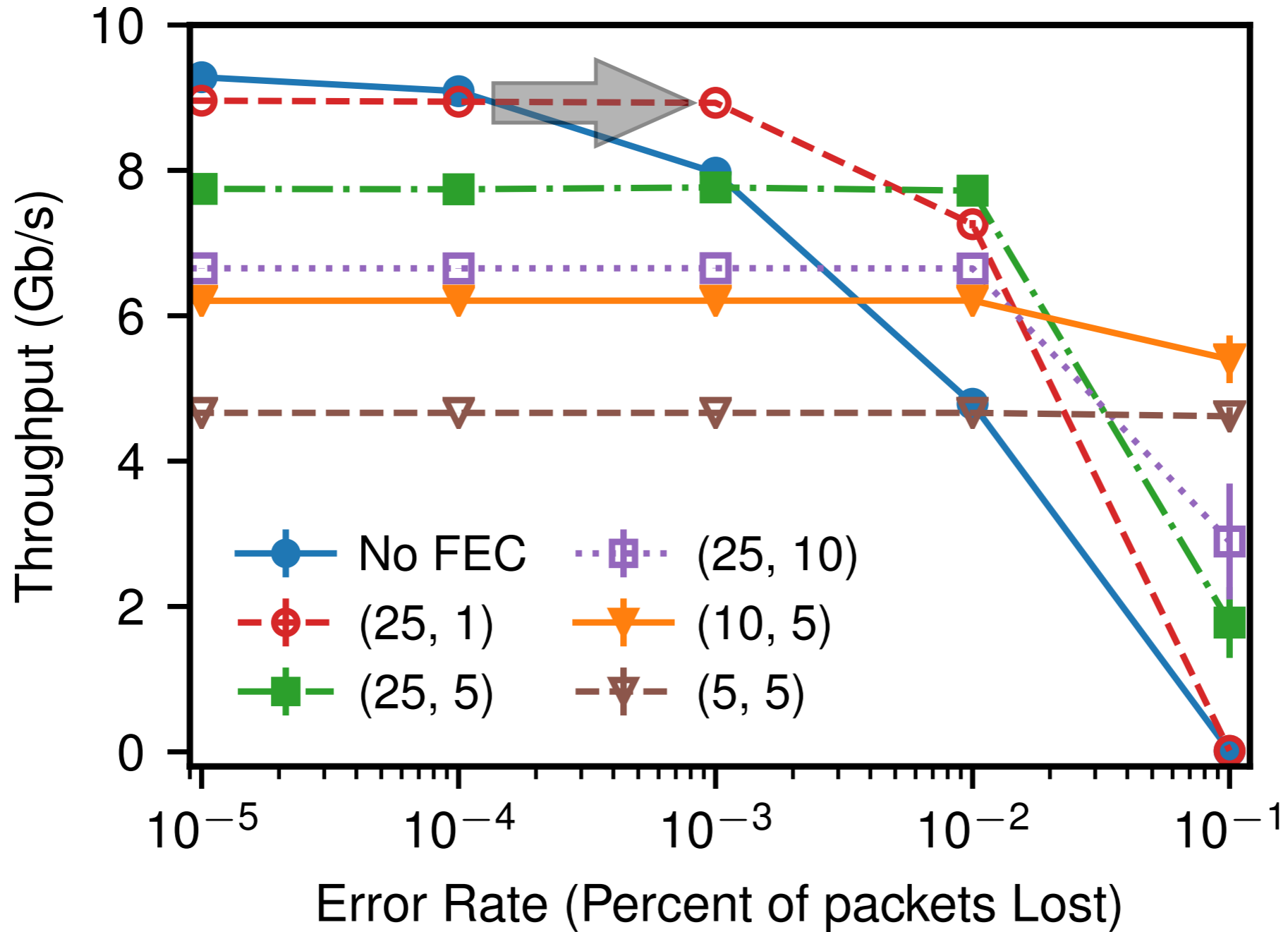
Evaluation

- Unmodified host stacks and applications.
- Raw throughput.
DPDK vs FPGA/CPU implementation of Encoder
FPGA: 9.3Gbps
CPU: 1.4Gbps (8 physical cores)
- Goodput vs Error-rate
iperf vs model.

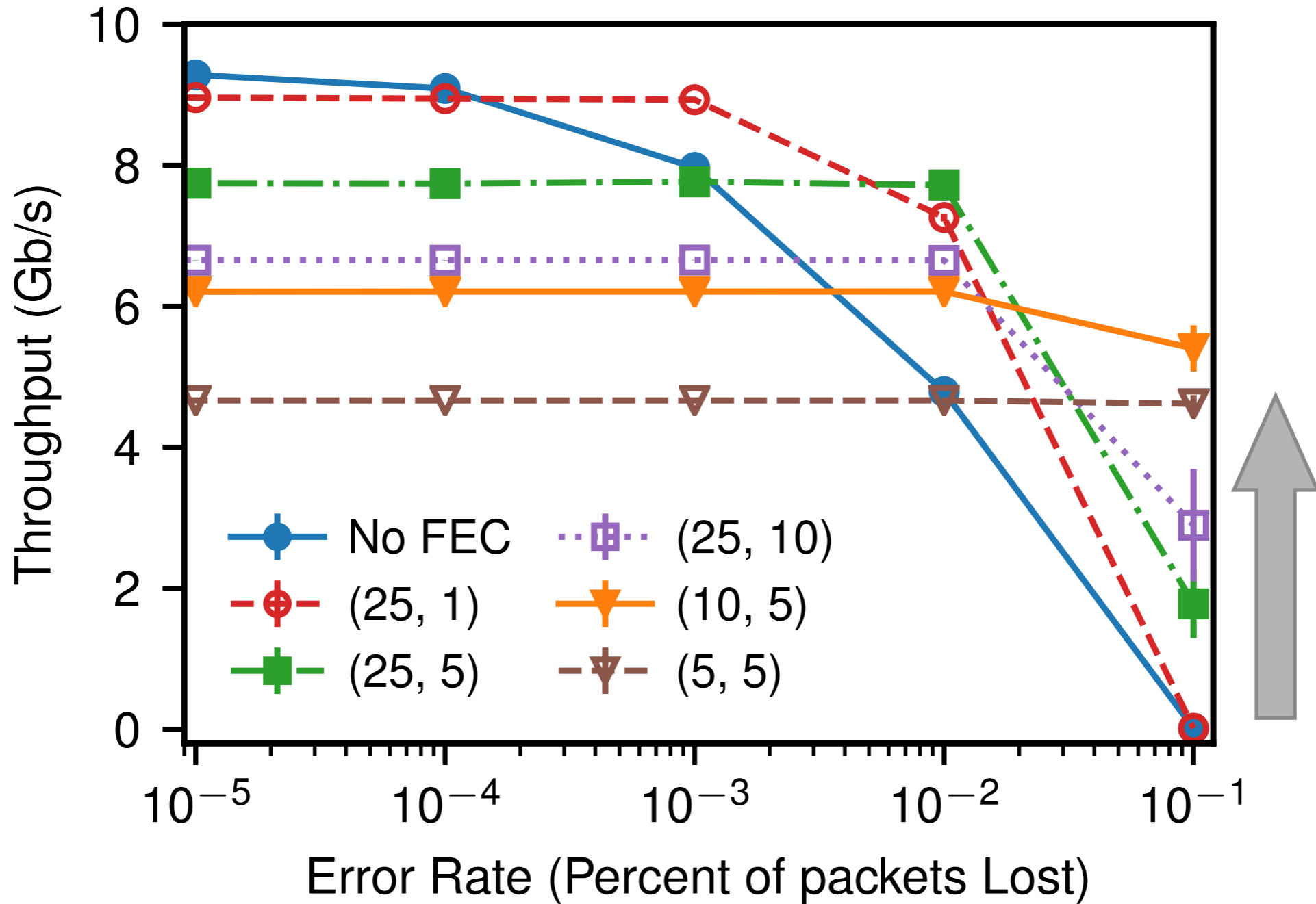
Evaluation

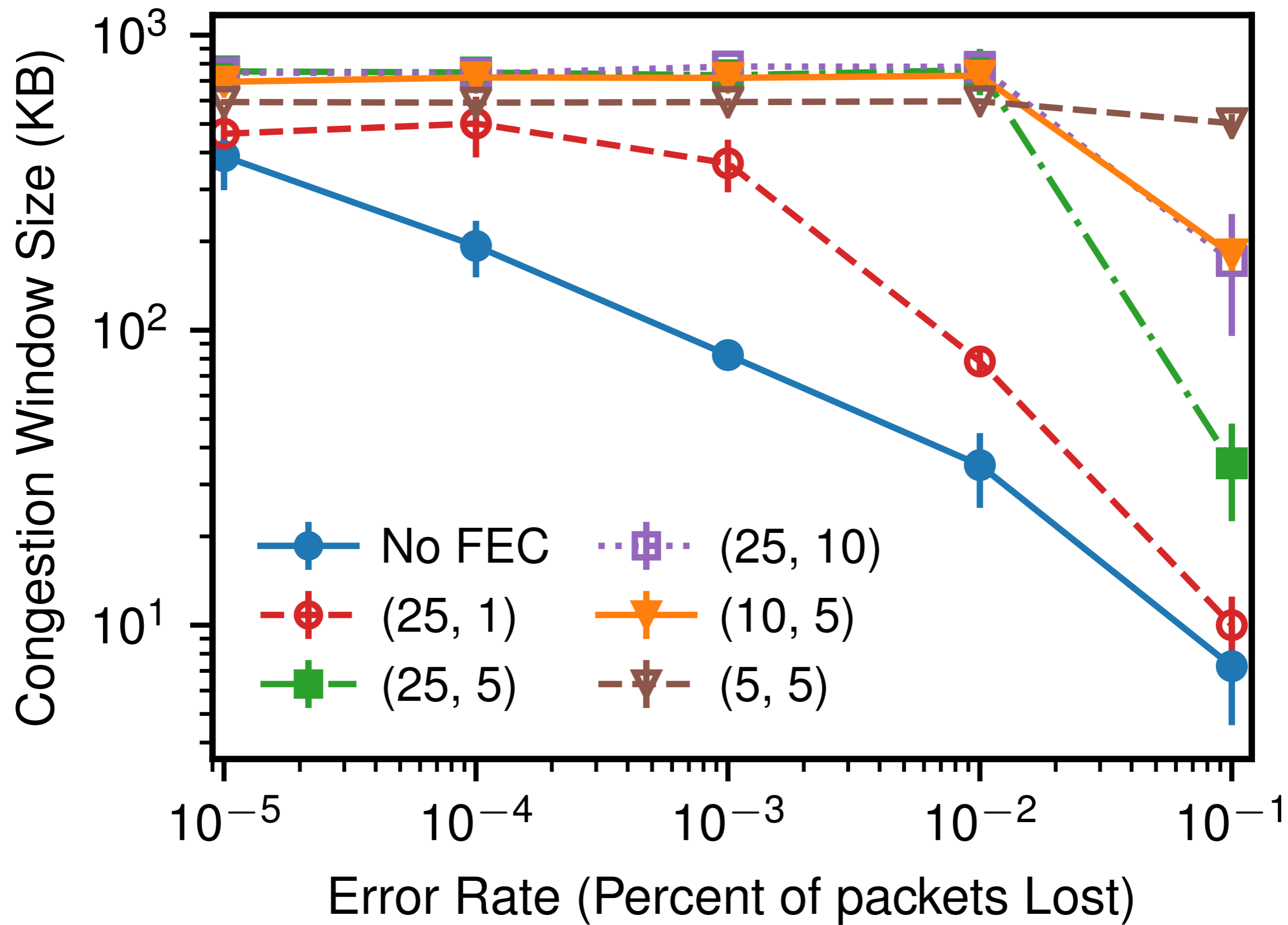


Evaluation



Evaluation





Conclusions

- Design for in-network lossy-link mitigation
Components: FEC + management logic
- Goals: network transparency, quick reaction, configurable classes, low non-FEC overhead.
- Compatible with existing/centralised approaches, to alert technicians/SREs.
- Ongoing work: completing implementation, integrating new “externs” on heterogeneous host/network

In-Network Computing to the rescue of Faulty Links

Hans Giesen

University of Pennsylvania

Lei Shi

University of Pennsylvania

John Sonchack

University of Pennsylvania

Anirudh Chelluri

University of Pennsylvania

Nishanth Prabhu

University of Pennsylvania

Nik Sultana

University of Pennsylvania

Latha Kant

Perspecta Labs

Anthony J McAuley

Perspecta Labs

Alexander Poylisher

Perspecta Labs

André DeHon

University of Pennsylvania

Boon Thau Loo

University of Pennsylvania

Acknowledgements: Isaac Pedisich (UPenn), Gordon Brebner (Xilinx),
DARPA Contracts No. HR0011-17-C-0047 and HR0011-16-C-0056, and NSF grant CNS-1513679.