RouteBricks:
Exploiting Parallelism to Scale Software Routers

Leon, Jack, Jihoon

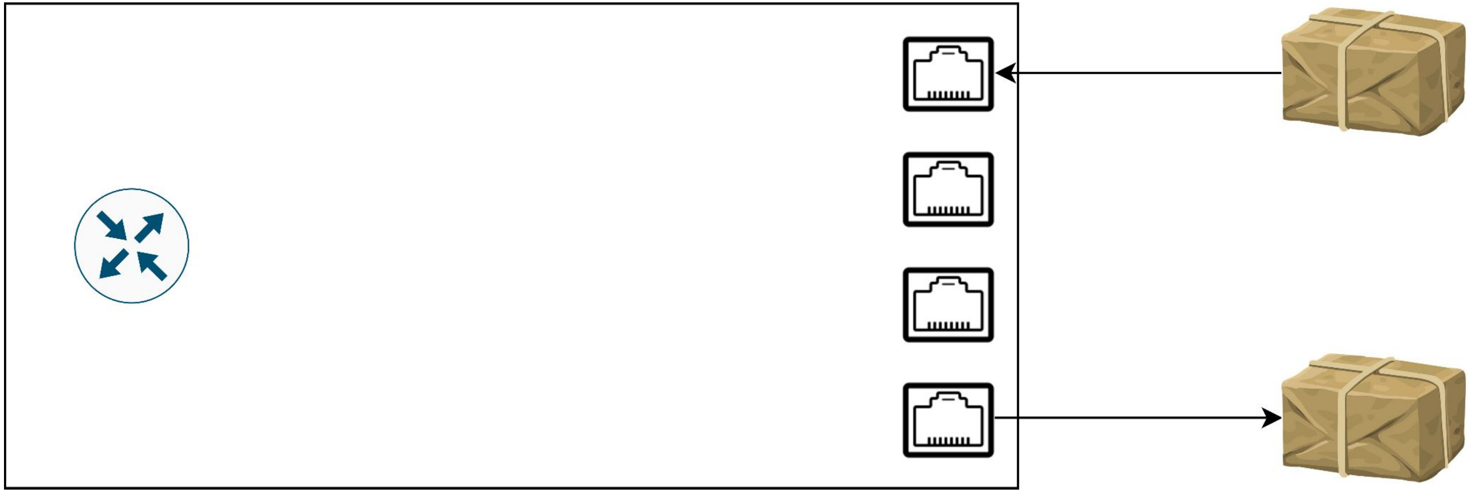
Introduction

- Modern networks / Service providers want to offer services which go beyond capabilities of hardware routers
  - Currently: appliance middleboxes
  - “Often involve modification to the per-packet processing on a router’s high-speed data plane”
  - Application acceleration, measurement and logging, encryption, filtering and intrusion detection, etc.
- Hardware Routers are expensive, inflexible, hard to program (but power and space efficient and really fast)
- Composing one large router out of small pieces - cluster router
- They predicted the rapid increase in demand for bandwidth

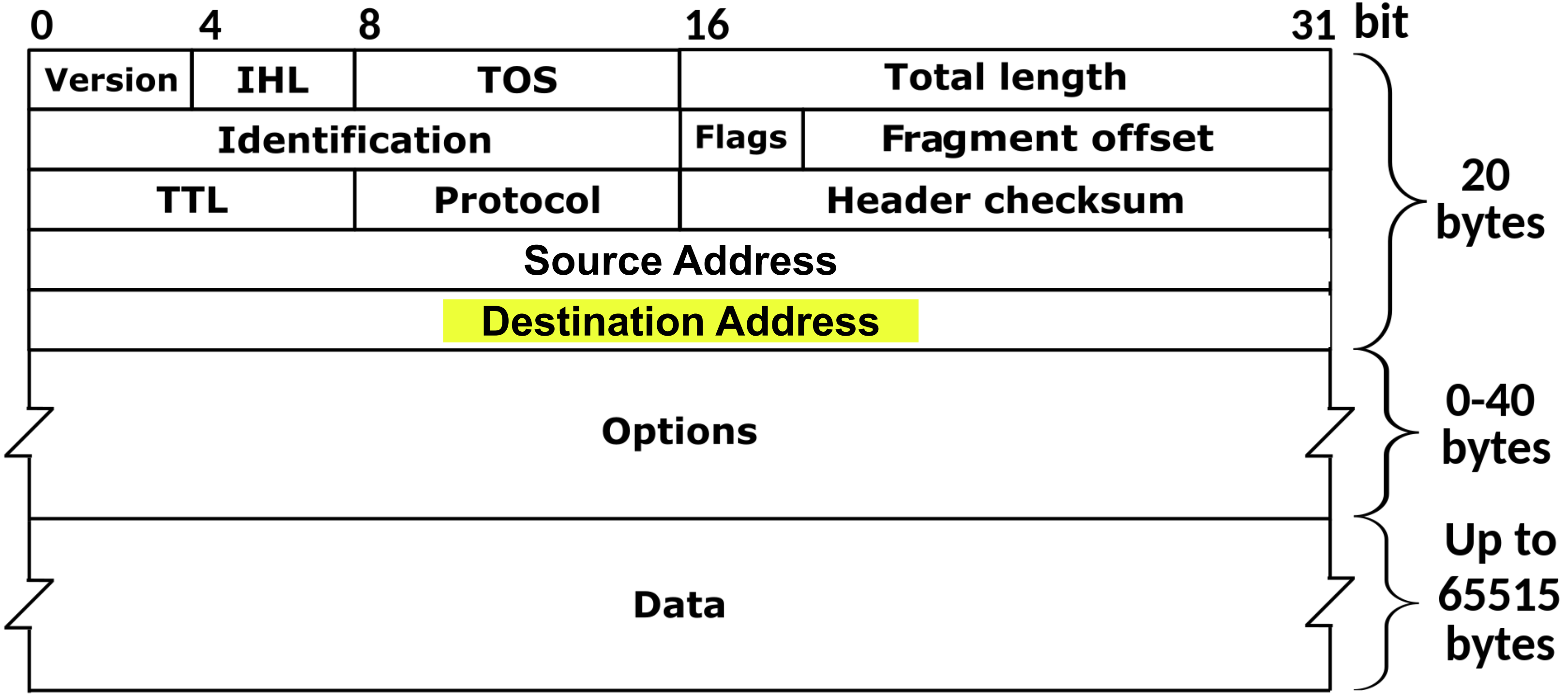
Hardware or Software today?

- The “new services” they talked about in the introduction have become more common, as they predicted
- P4 network programming language created in 2013 for hardware routers, industry standard
- Hardware routers are used more often when bandwidth matters
- Software routers for stateful things like firewalls, stateful address translation service, etc.
- Middleboxes do still exist: network security appliances, VPNs, cache servers

Recap: what's an IP-Router?



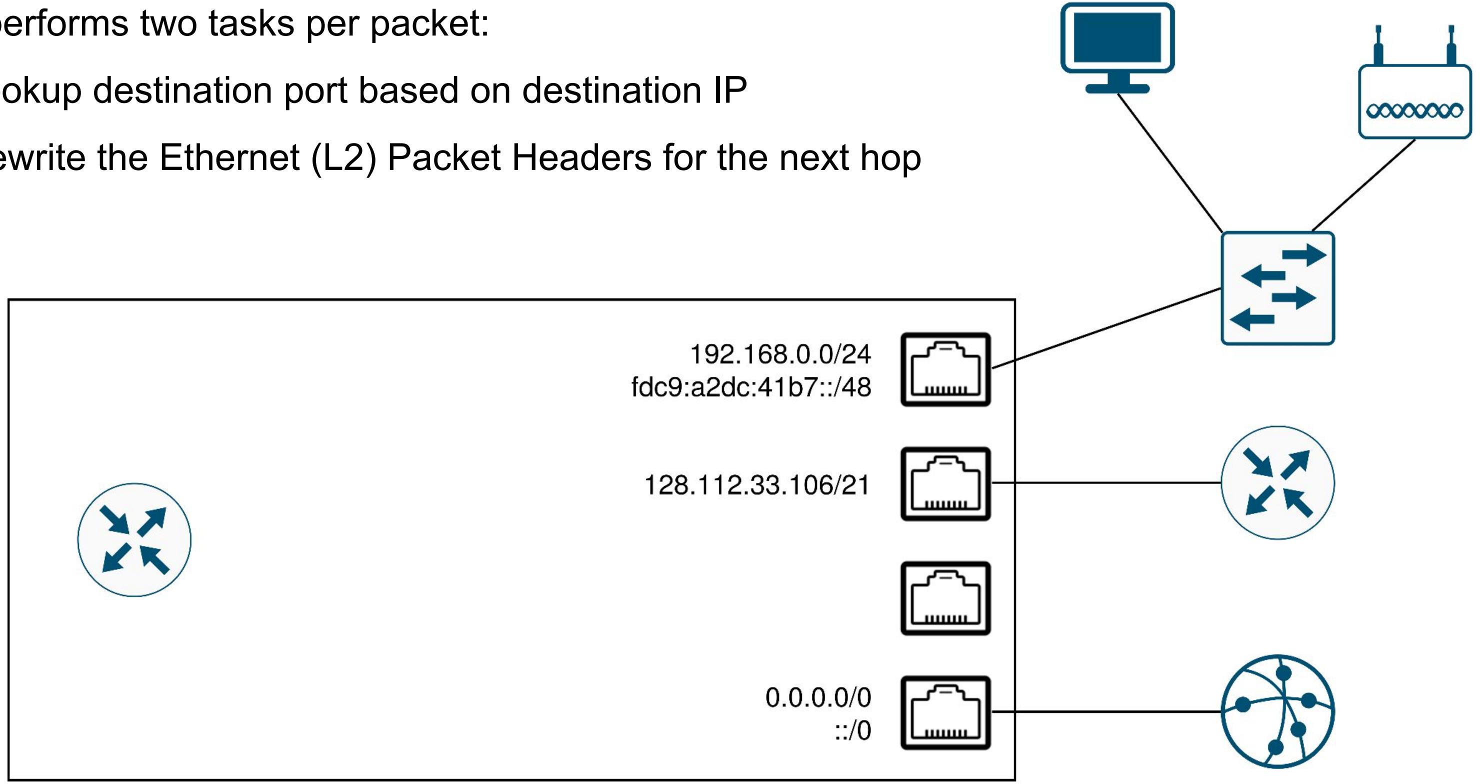
Recap: what’s an IP-Router?



Recap: what’s an IP-Router?

Mostly performs two tasks per packet:

1. Lookup destination port based on destination IP
2. Rewrite the Ethernet (L2) Packet Headers for the next hop

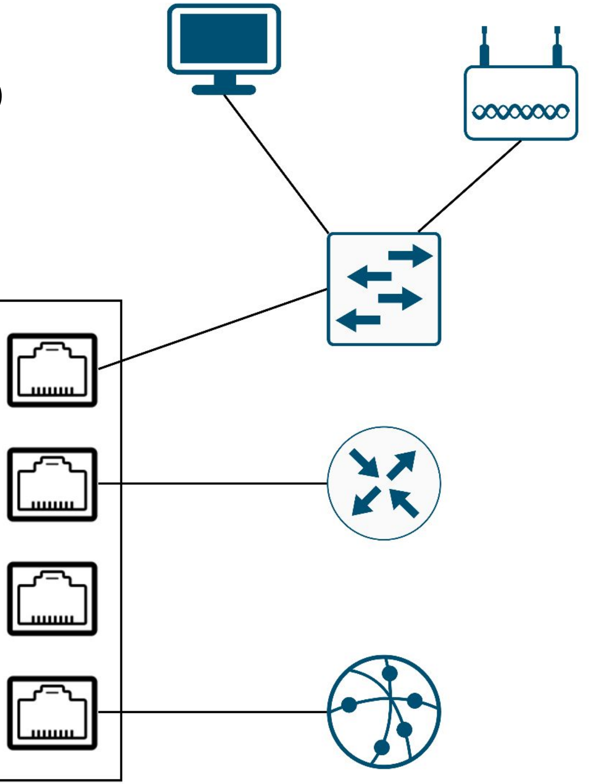


Recap: what’s an IP-Router?

Primary Challenges:

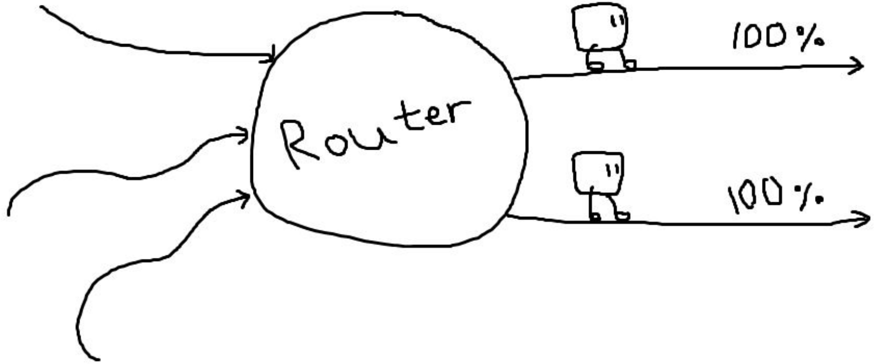
- High per-port bandwidth (up to 100GBit/s⟶ >200Mpps)
- Huge routing tables (~1Mio announced prefixes)

Prefix	Port
192.168.0.0/24	0
10.42.0.0/16	0
128.112.33.106/21	1
141.70.120.0/21	3
45.90.132.0/22	3



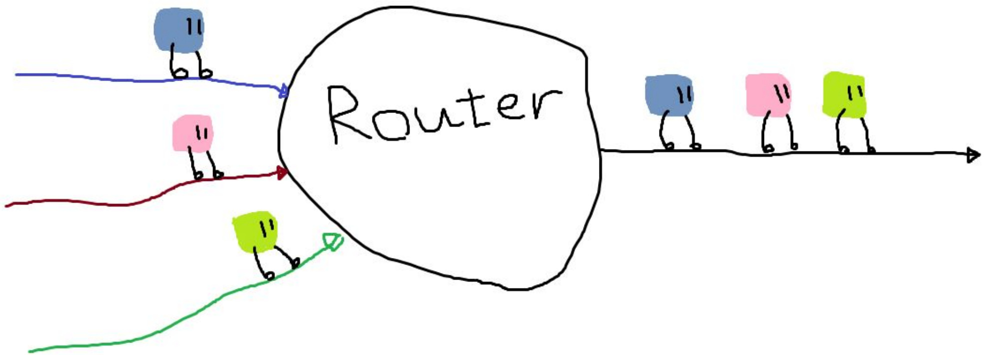
Problem of Parallelizing Across Servers

1. 100% throughput



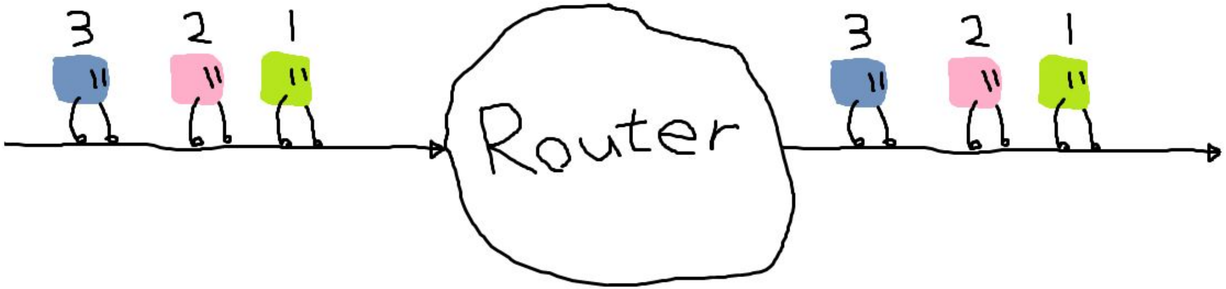
Problem of Parallelizing Across Servers

2. Fairness

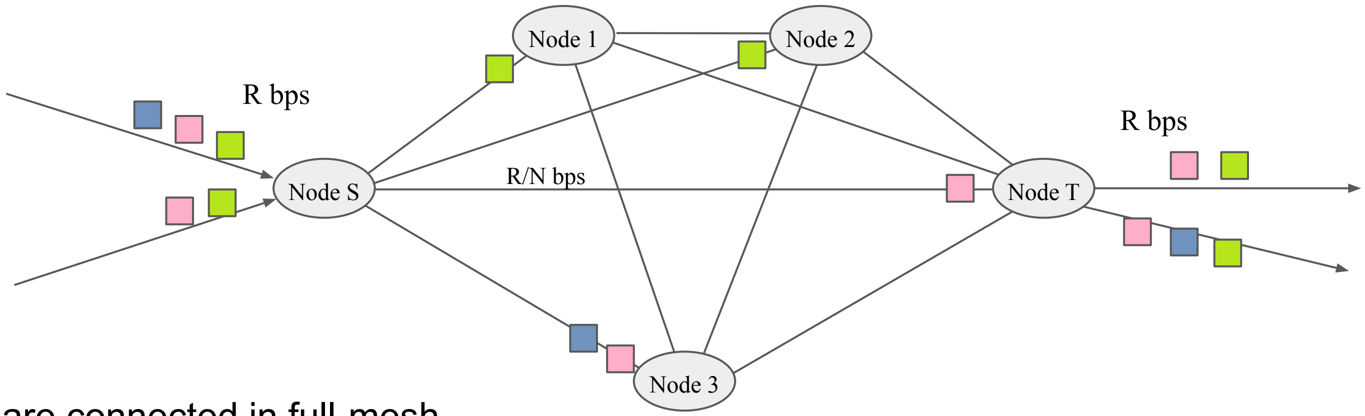


Problem of Parallelizing Across Servers

3. Avoid Reordering



Valiant Load Balancing (VLB)

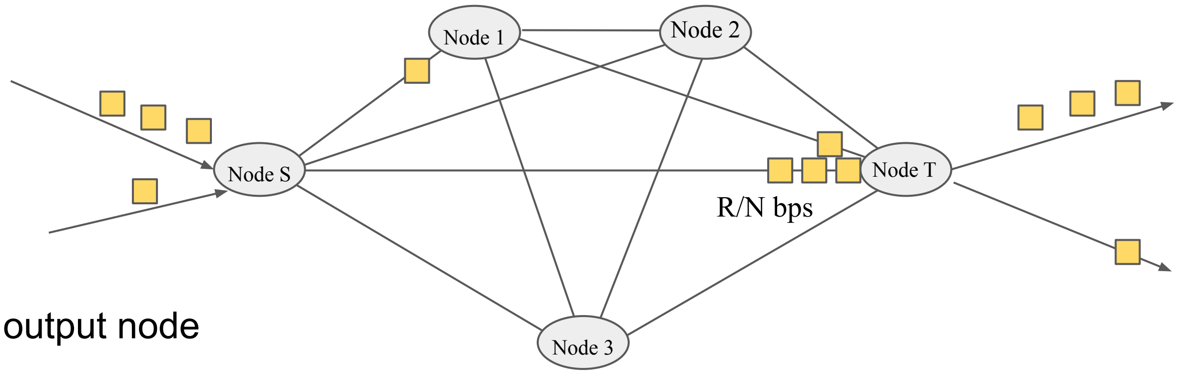


- Assume that all nodes are connected in full-mesh
- Packets are randomly sent to intermediate nodes (Phase 1)
- Packets are sent to output node (Phase 2)

- Instead of 2R per-server processing rate, it becomes 3R

Universal Schemes for Parallel Communication, L. Valiant and G. Brebner., STOC 1981

Direct VLB



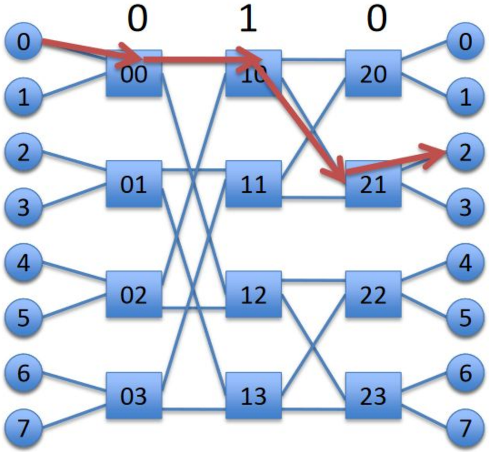
- Up to R/N traffic goes directly to output node
- Load-balance the rest.

- Ideally, when network is uniformly random
  This can lead to 2R server process rate

On Direct Routing in the Valiant Load-Balancing Architecture, R. Zhang-Shen and N. McKeown, GLOBECOM 2005

Topology

- However full-mesh might not be possible
- k-ary n-fly butterfly topology
  - n stage
  - k output/input ports per node
  - k^n terminals
  - nk^{n-1} internal nodes



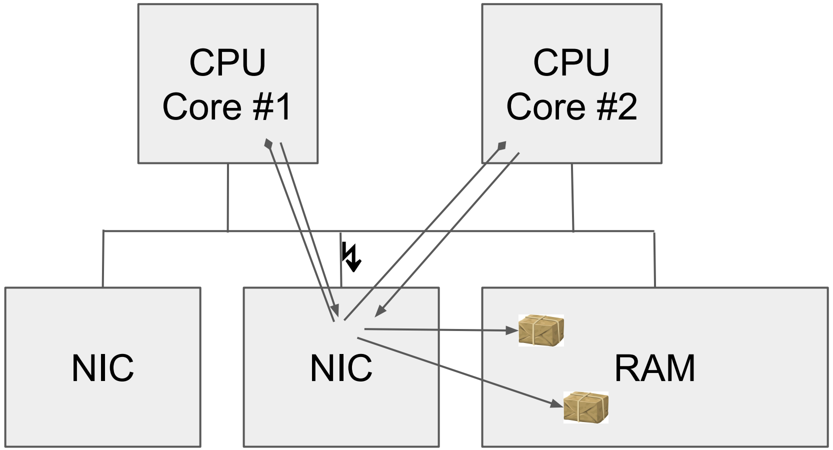
2-ary 3-fly

Image from: https://studylib.net/doc/5613680/flattened-butterfly--a-cost-efficient-topology-for-high

Problem of Parallelizing Within Servers

Paper presents two rules:

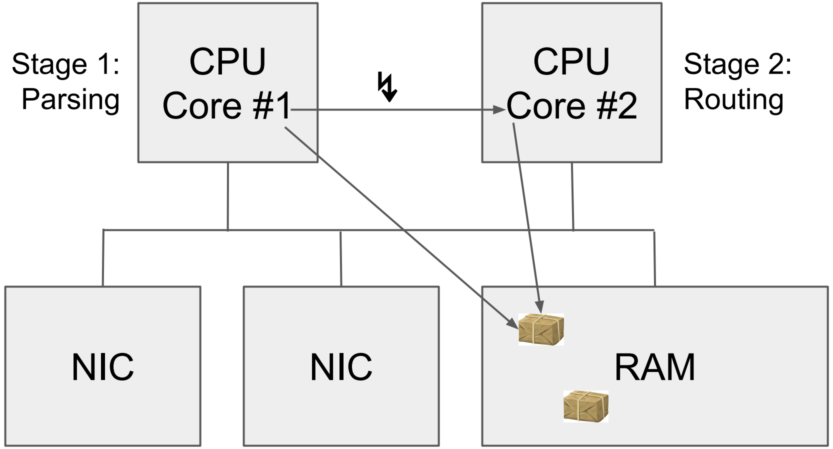
1. Each network queue should be accessed by a single core.
2. Each packet should be handled by a single core.



Problem of Parallelizing Within Servers

Paper presents two rules:

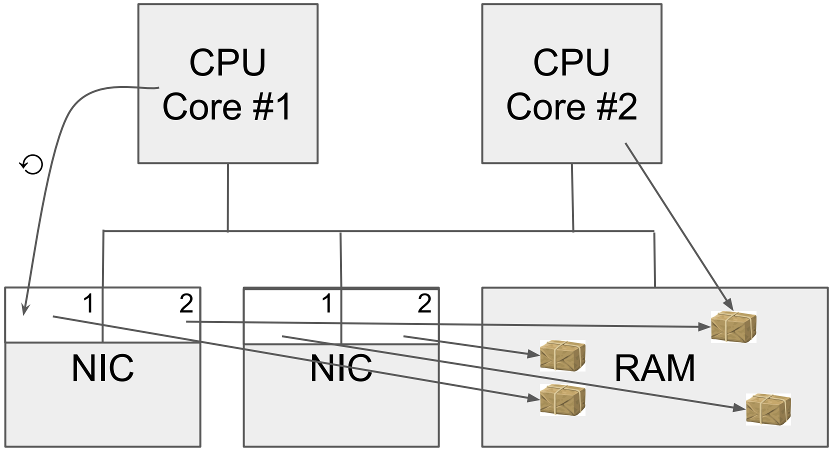
1. Each network queue should be accessed by a single core.
2. Each packet should be handled by a single core.



Problem of Parallelizing Within Servers

Paper presents two rules:

1. Each network queue should be accessed by a single core.
2. Each packet should be handled by a single core.



Evaluation

This paper makes significant contributions to the state of the art at the time of publication, providing one of the earliest examples of a parallelized software cluster router.

This design was evaluated on a few workloads across different packet sizes and the Abilene trace.

This problem remains an open area of research today.

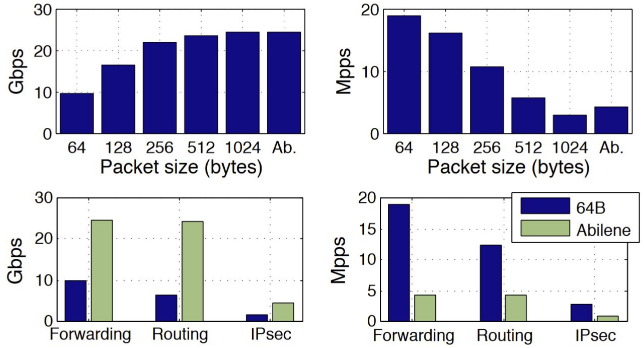


Figure 8: Forwarding rate for different workloads. Top: as a function of different packet-size distributions, when the server performs minimal forwarding. Bottom: as a function of different packet-processing applications, for 64B packets and the Abilene trace. "Ab." refers to the Abilene trace.

Discussion Points

- ➔ What constitutes a router? Is RouteBricks a *network of routers* or a *single* router?
- ➔ Is the Abilene trace a standard measurement?
- ➔ 0.15% packet reordering – any good? Consequences?
- ➔ Why is development for hardware (ASIC) switches & routers so difficult?
  Can this be solved through advances in e.g., programming languages? What about FPGAs?
- ➔ Are the evaluated workloads representative of “middlebox” workloads?
- ➔ How do typical network systems and architectures look in 2023?
  Is there a justification for software routers today?
- ➔ What about fault tolerance?

Composing a single router of multiple