Software Defined Networking

Dr. Nick Feamster
Professor

In this course, you will learn about software defined networking and how it is changing the way communications networks are managed, maintained, and secured.

This Module: Programmable Data Plane

- Two Lessons
  - Programming the data plane: Click
  - **Scaling programmable data planes**
    - **Making software faster**
    - Making hardware more programmable
- **Optional** programming assignment (in Click)
- Quiz on Concepts

Motivation

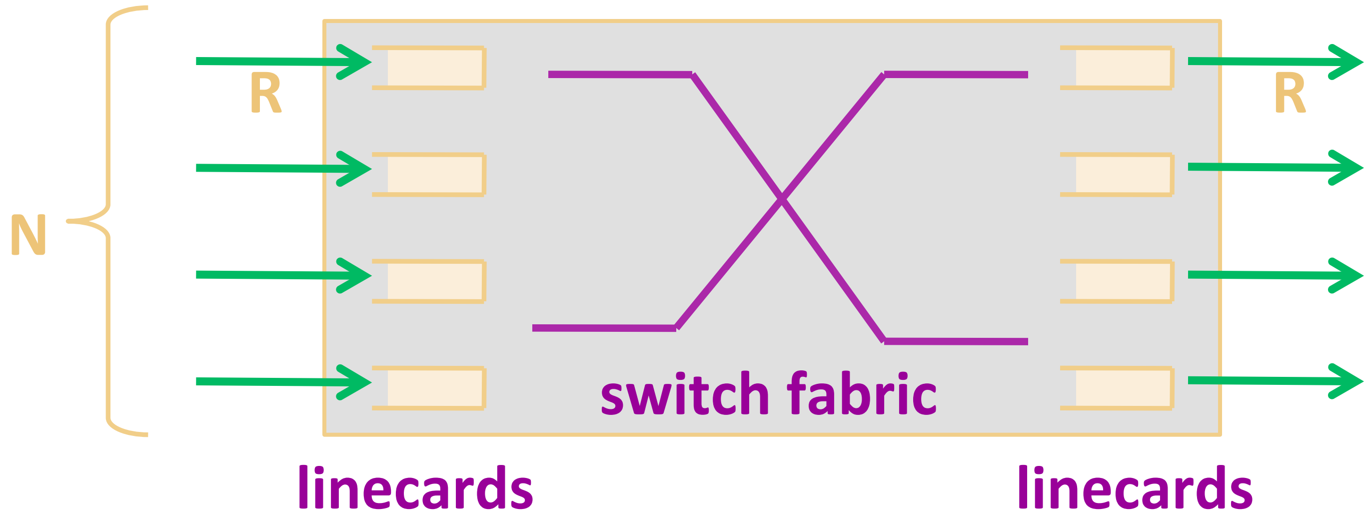
- Many new protocols require data-plane changes.
  - Examples: OpenFlow, Path Splicing, AIP, …
- Protocols must forward packets at acceptable speeds.
- May need to run in parallel with existing protocols

- **Need:** Platform for developing new network protocols that
  - Forwards packets at high speed
  - Runs multiple data-plane protocols in parallel

Existing Approaches

- Develop **custom software**
  - **Advantage:** Flexible, easy to program
  - **Disadvantage:** Slow forwarding speeds
- Develop modules in **custom hardware**
  - **Advantage:** Excellent performance
  - **Disadvantage:** Long development cycles, rigid
- Develop in **programmable hardware**
  - **Advantage:** Flexible and fast
  - **Disadvantage:** Programming is difficult

Hardware Router



- Processing at rate ~R per line card
- Switching at rate N x R by switch fabric

Dobrescu, Mihai, et al. "RouteBricks: exploiting parallelism to scale software routers." *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*. ACM, 2009.

RouteBricks: Linecards on Servers

- Processing at rate ~*R* per server
- Switching at rate ~*R* per server

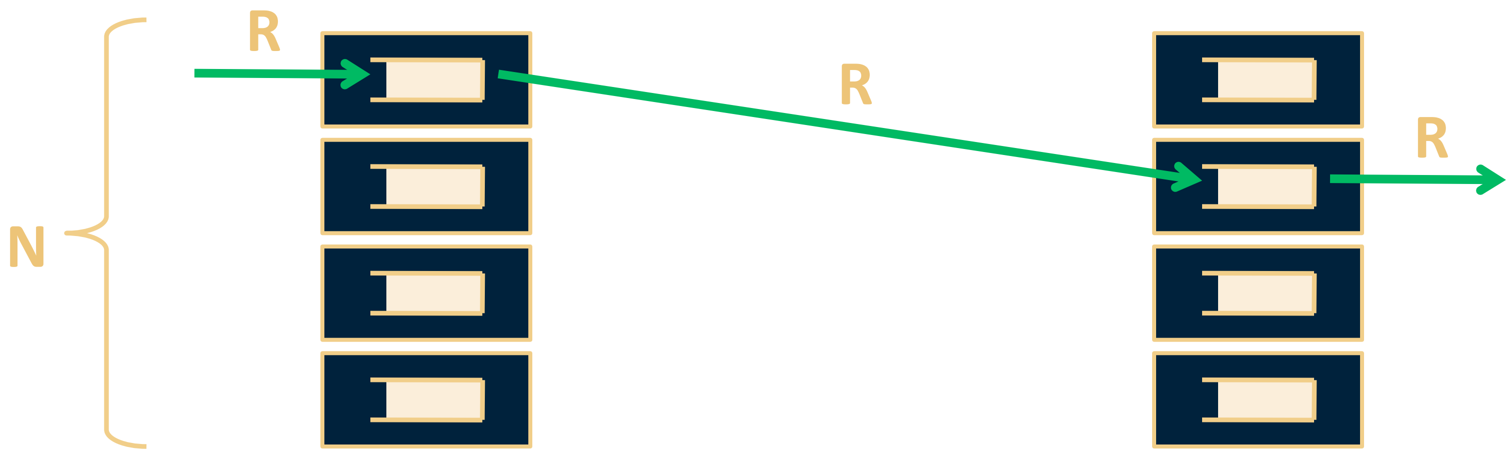
Requirements

- Internal link rates < *R*
- Per-server processing rate: *c* x *R*
- Per-server fanout: constant

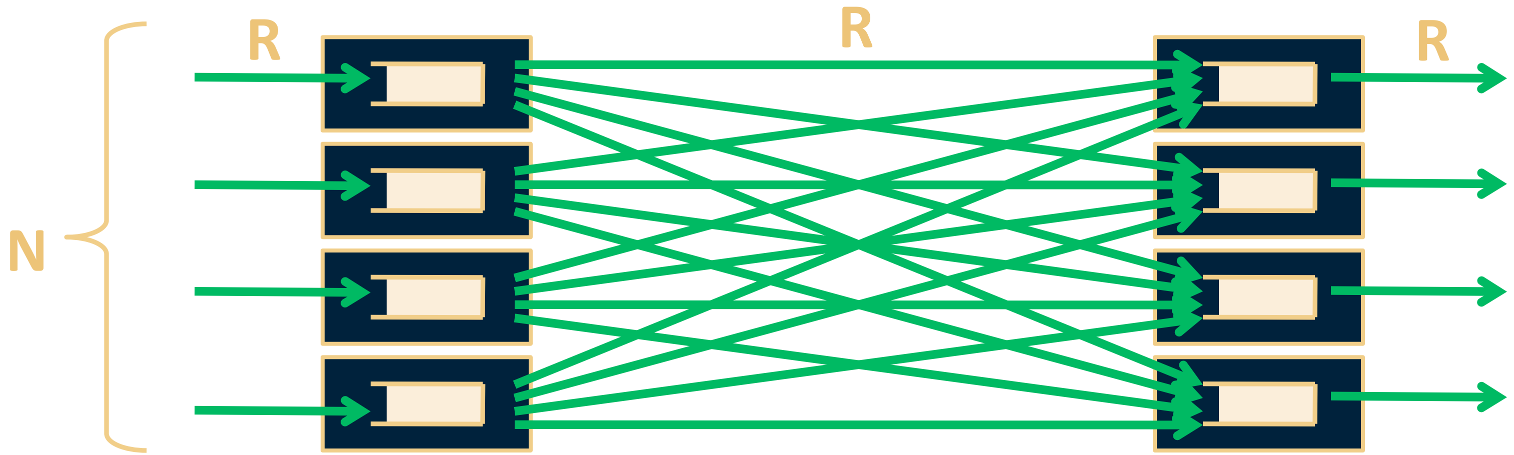
Challenges

- **Limited internal link rates:** Internal links can't exceed external link rates
- **Limited per-node processing rate:** Desire to use commodity hardware
- **Limited per-node fanout:** Due to limited NIC slots/ports

Strawman Approach

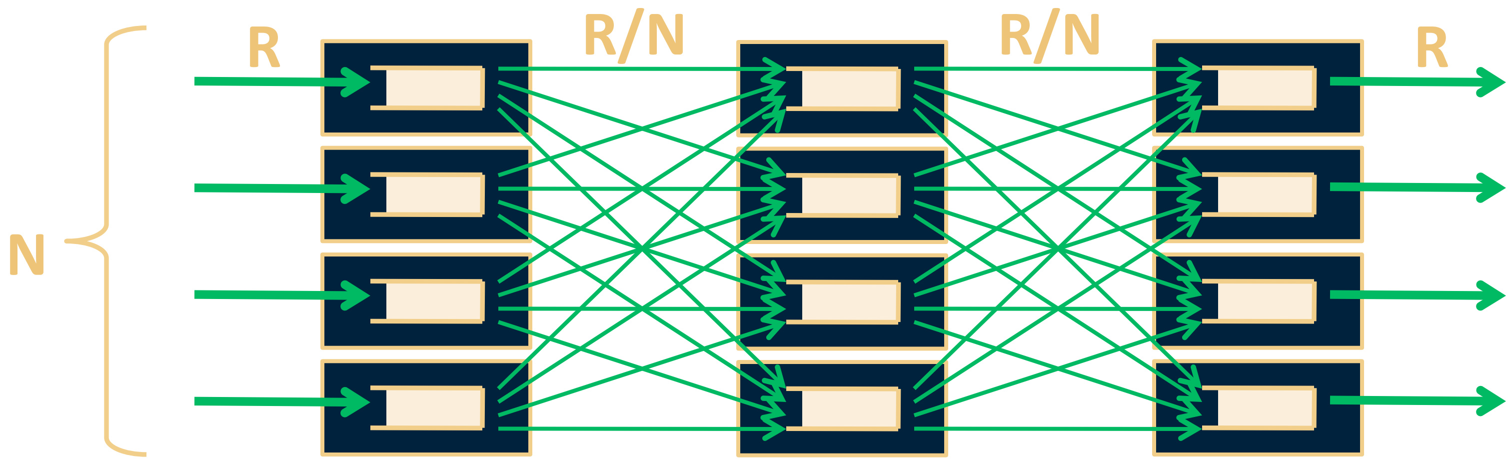


Strawman Approach



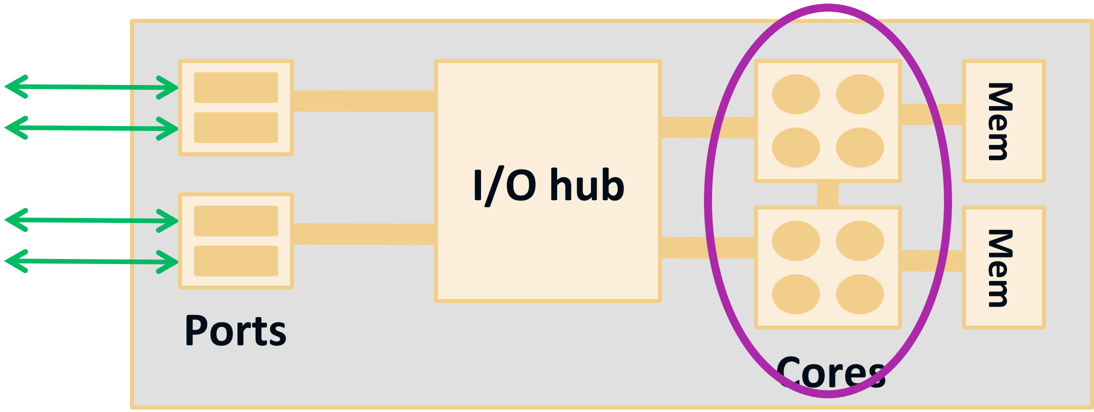
- *N* external links of capacity *R*
- N^2 internal links of capacity *R*

Valiant Load Balancing



- Per-server processing rate: 3*R*
- With uniform traffic (avoiding first phase): 2*R*

Each Server Must Also Be Fast

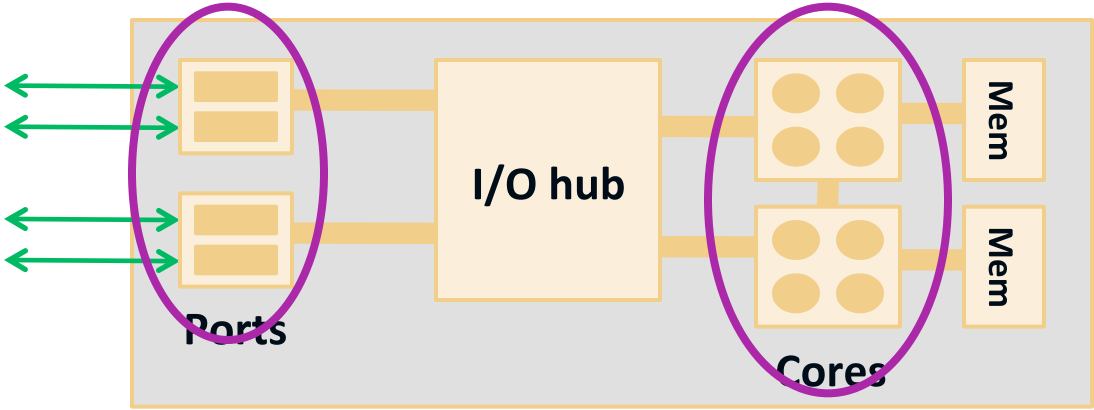


- First try: **1.3 Gbps**

Problem #1: Bookkeeping

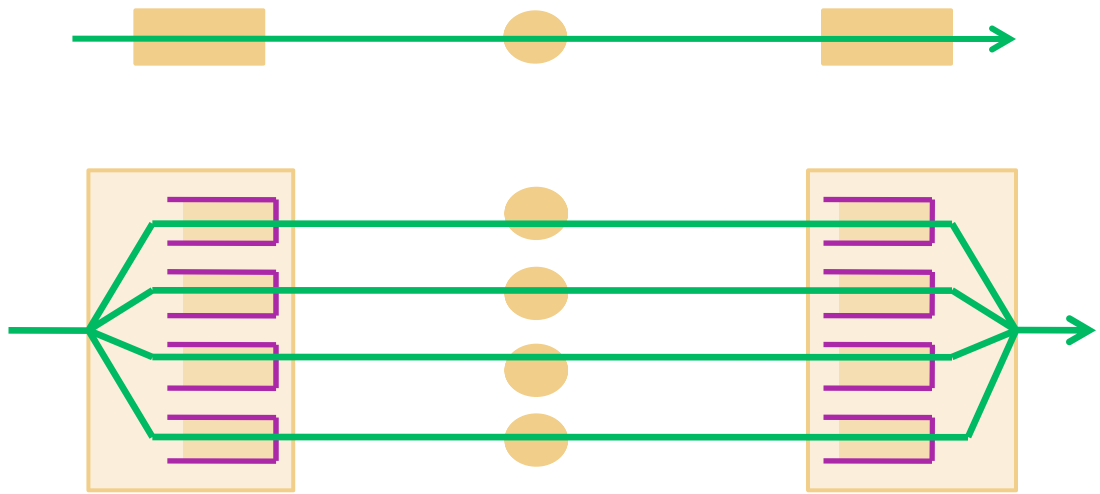
- **Managing packet descriptors**
  - moving between NIC and memory
  - updating descriptor rings
- **Solution: batch packet operations**
  - NIC batches multiple packet descriptors
  - CPU polls for multiple packets
  - Cost: increased latency

Single-Server Performance



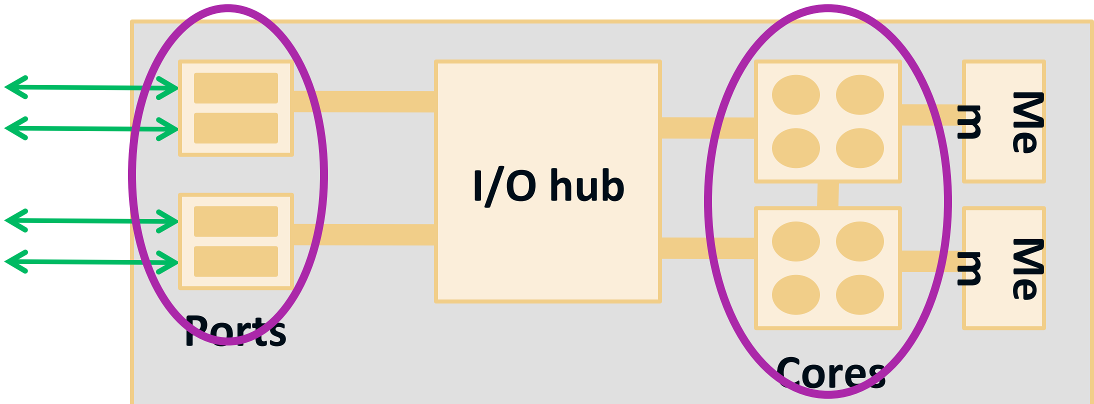
- First try: 1.3 Gbps
- With batching: 3 Gbps

Problem #2: Queue Access



- Rule #1: 1 core per queue (avoids locking)
- Rule #2: 1 core per packet (faster)

Single-Server Performance



- **First try:** 1.3 Gbps
- **With batching:** 3 Gbps
- With multiple queues: **9.7 Gbps**

Fast Software Forwarding: Other Tricks

- Large packet buffers to hold multiple packets
- Batch processing
- Ethernet GRE (to avoid complicated lookup)
- Avoiding lookups on bridge between virtual interfaces and physical interfaces

Han, Sangjin, et al. "PacketShader: a GPU-accelerated software router." *ACM SIGCOMM Computer Communication Review* 40.4 (2010): 195-206.
Bhatia, Sapan, et al. "Trellis: A platform for building flexible, fast virtual networks on commodity hardware." *Proceedings of the 2008 ACM CoNEXT Conference*. ACM, 2008.

Summary

- **Scalability: Make the software faster**
  - **Software routers can be fast!**
- General purpose infrastructure is capable of fast forwarding performance
  - The low-level details, optimizations matter
- Other efforts underway
  - Intel DPDK