Forwarding architectures

Brighten Godfrey
cs598pbg Sept 16 2010

slides ©2010 by Brighten Godfrey unless otherwise noted

Announcements

- Presentation matching by tomorrow
- Comments on project proposals early next week

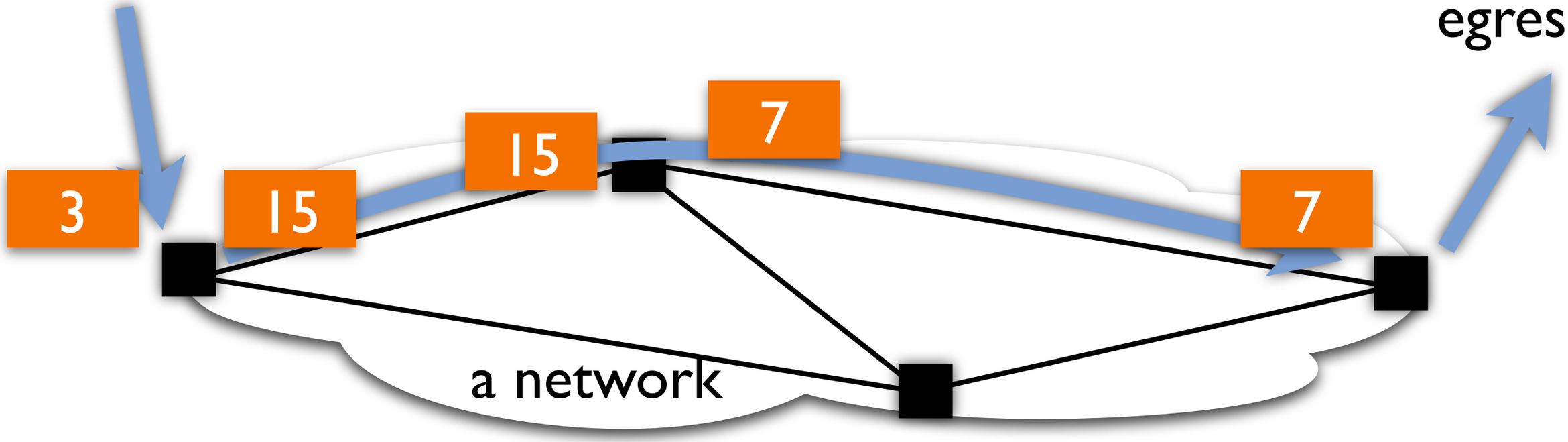
A shared concern

Can IP scale to high bandwidth?

- MPLS answer: simpler data plane is more efficient (but note this was not the only motivation for MPLS)
- "50-Gb/s IP Router" answer: let's make IP scale with careful design

MPLS design

Ingress:
Traffic classification, label packets ("forwarding equivalence class")



- Control plane constructs label-switched paths and coordinates labels
- Can also stack labels, concatenating

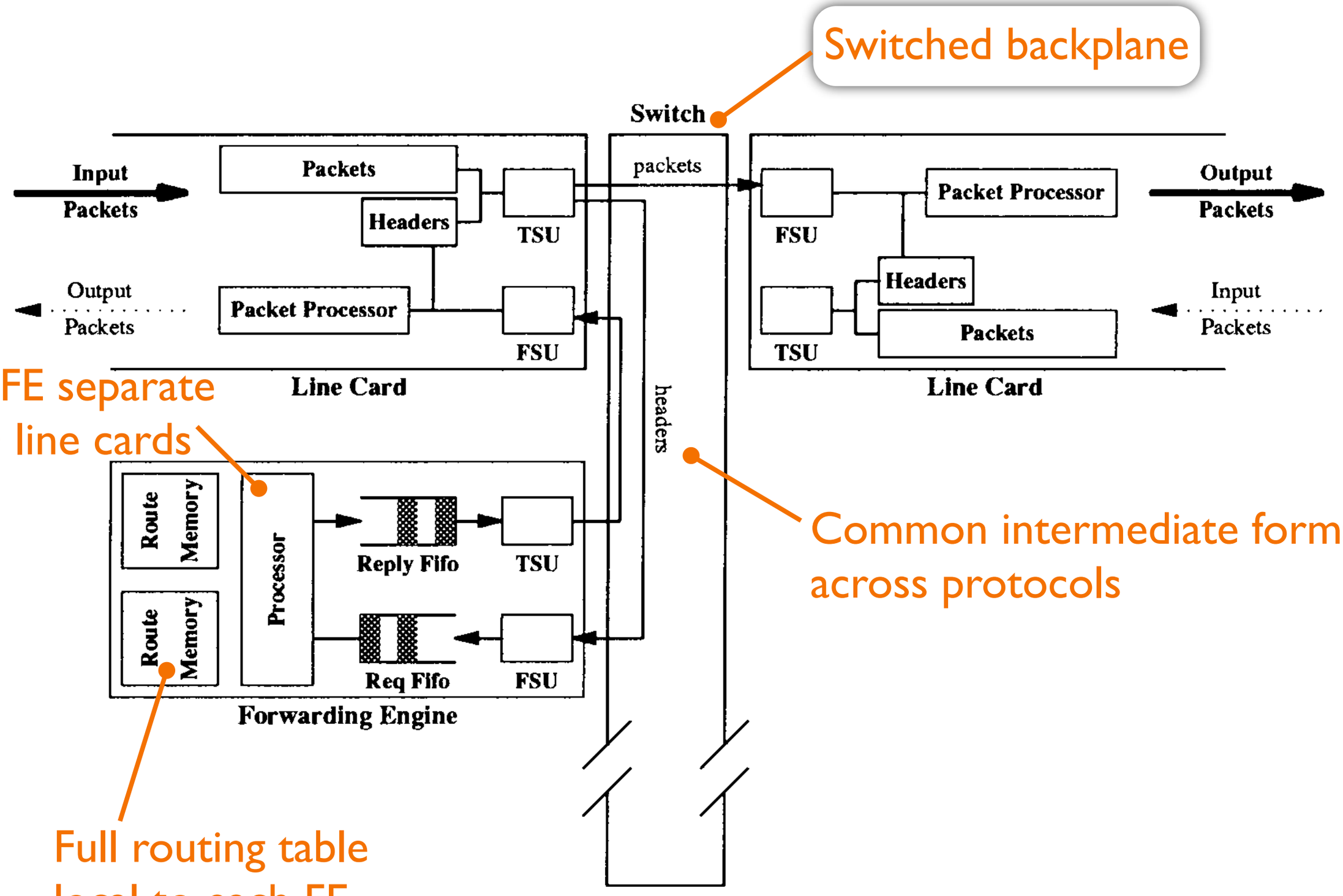
MPLS motivation

- In the design doc
  - High performance forwarding
  - Minimal forwarding requirements, so can interface well with many types of media such as ATM
  - Flexible control of traffic routing
- What matters today? Flexibility
  - Virtual Private Network (VPN) service along dedicated paths between enterprise sites
  - Traffic engineering on per-"flow" granularity
  - Control backup paths with MPLS Fast ReRoute

Partridge: 50 Gb/sec router

- A fast IP router
- Good exhibition of the guts of a router and problems to be solved in router architecture

Inside the router



[Partridge et al '98]

Switching fabric

- Operates in epochs: 128 bytes sent by each line card to the next-hop line card
  - Each line card can send to only one other card, and can receive from only one other card

... to outputs

Inputs ready to send...

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

- Maximum matching problem solved by an allocator and dictated to the line cards in each epoch

Efficiency vs. extensibility

Hardware routers

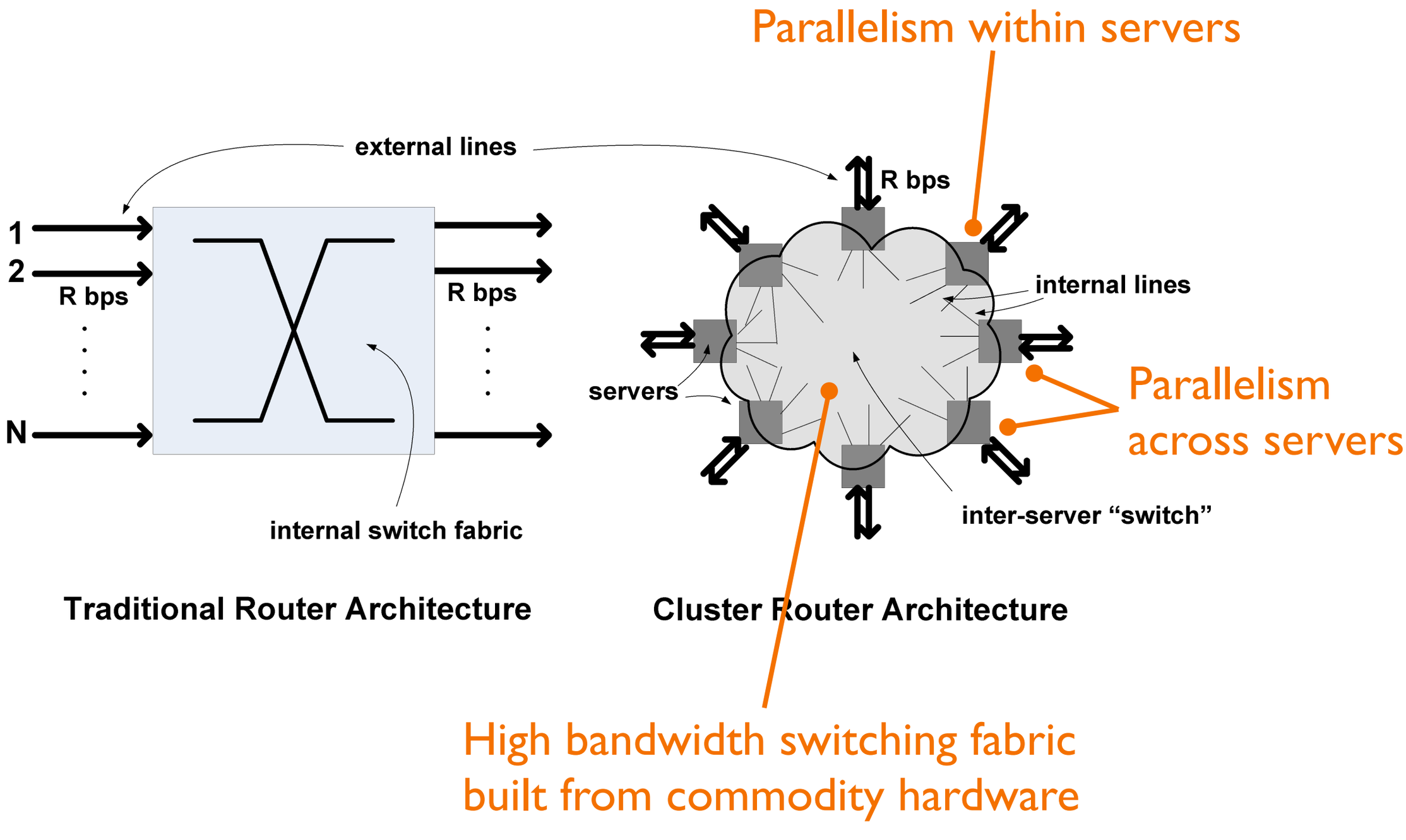
- Fast
- Specific functionality
- Result: many physical devices (routers, firewalls, intrusion detection, ...)

Software routers

- Slow
- Extensible

Can we get the best of both worlds?

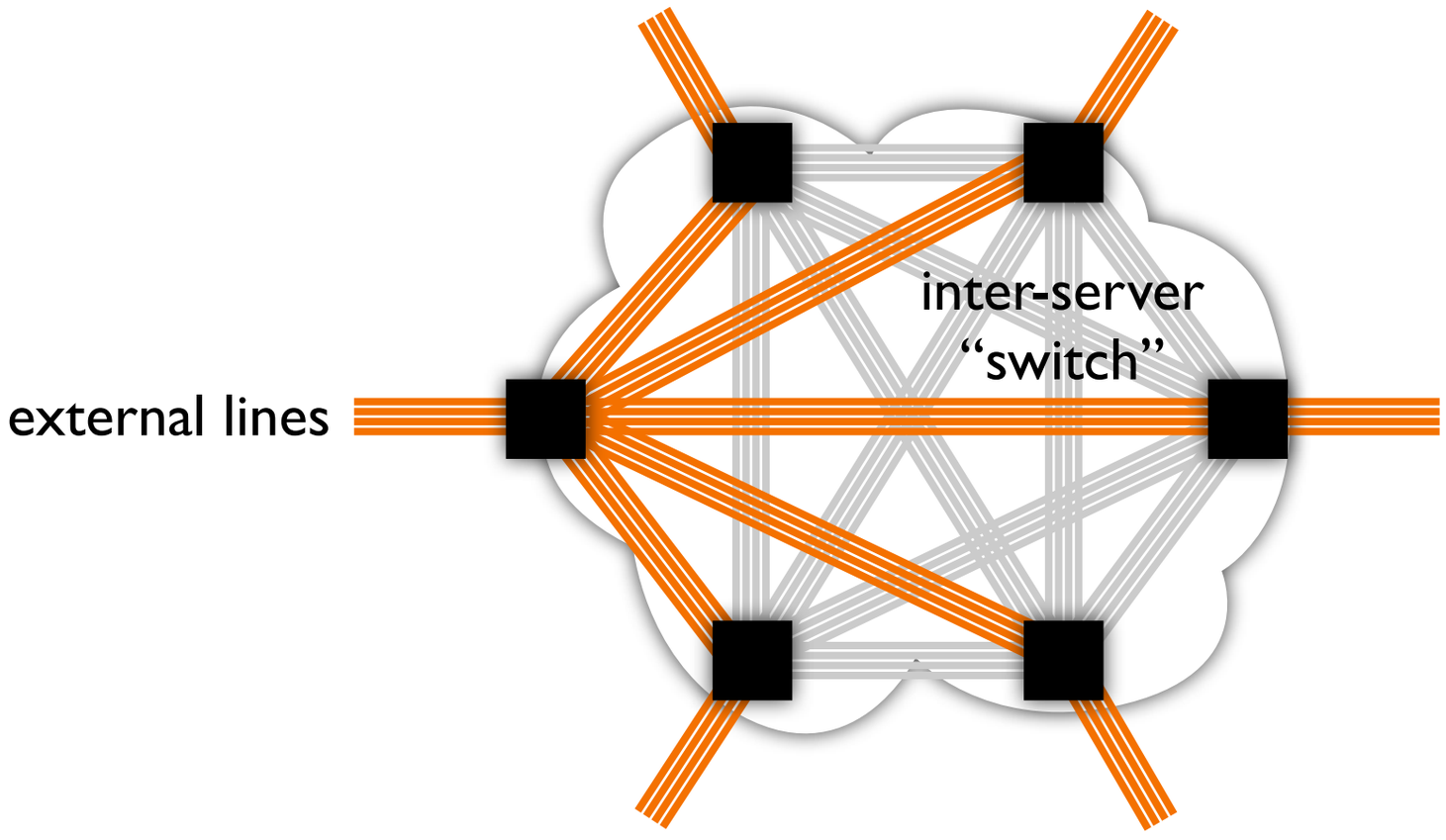
RouteBricks approach



[Dobrescu et al, NSDI 2009]

Switching fabric challenges

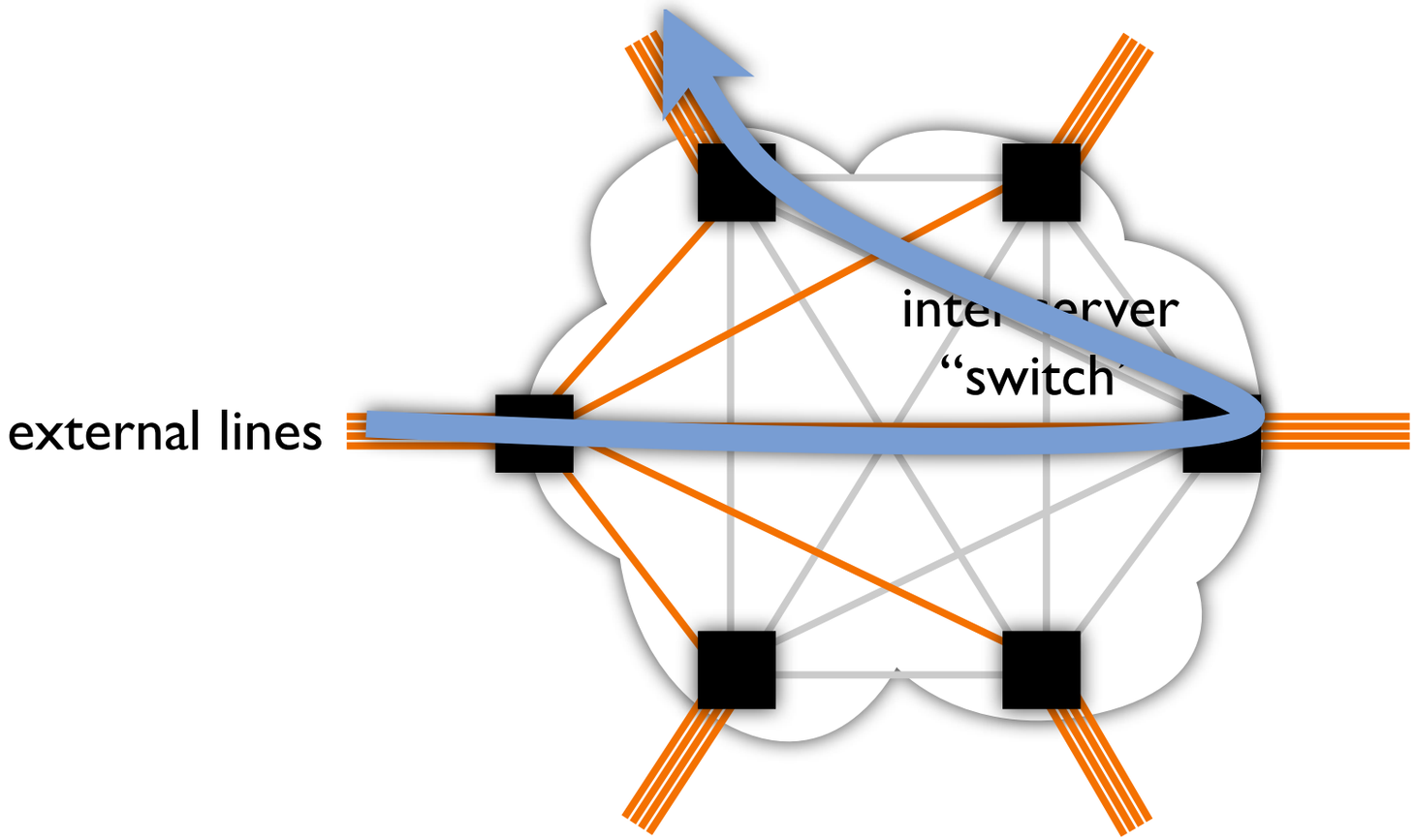
- Handle any traffic pattern: for example, all input traffic at a server might go to any one output server
- Low degree: we're using commodity hardware
- Naïve approach:



Useless: might as well just use one server!

Low degree solution

- Just one link out for each link in
- Total out b/w enough, but doesn't go where we need
- Solution (Valiant load balancing): send packet to random intermediate node, then on to destination



VLB guarantees & questions

- Guaranteed to nearly full throughput for any traffic demands!
  - "nearly" = 2x. Why?
  - So, switch fabric needs to be 2x as fast as external links to provide guarantees
- Why does sending to a random intermediate node work?
- Still using one port per server. What if # servers > # ports available?

PacketShader

- [Han, Jang, Park, Moon, SIGCOMM 2010]
- Substantially improved performance for a single-node software router
  - Would resort to RouteBricks approach to scale beyond one server
- Key steps
  - GPU for massively parallel packet processing
  - Careful optimization

Key questions

- At what point, if ever, will software routers approach production quality & efficiency?
  - Companies exist already
  - Software routers in production use for certain tasks

- Will software-defined networking (next class) gain more traction?