A 50-Gb/s IP Router

~2Gb/s per author

Presenter: Ashutosh Dhekne

PhD student, Computer Science,

University of Illinois at Urbana Champaign

dhekne2@Illinois.edu

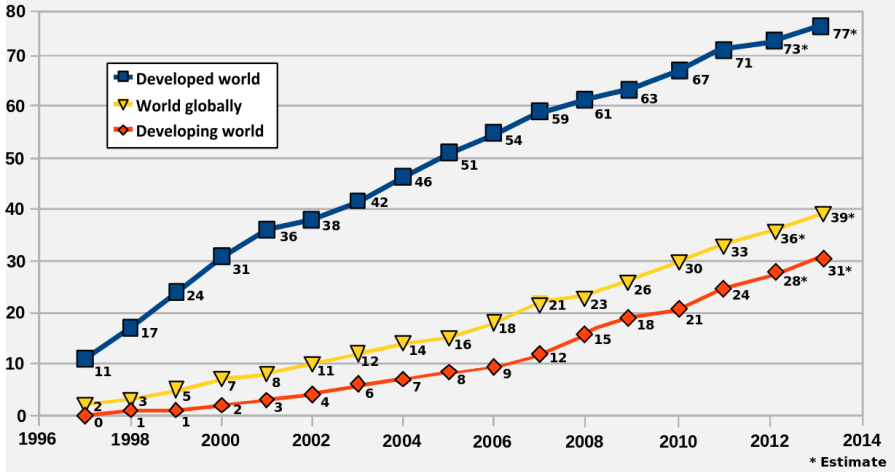
Why so many authors?

- Router developing is hard
- Lot of hardware and assembly level software expertise
- Complete understanding of architectural techniques
- Helps shorten the acknowledgement section
- Okay, let's get a bit more serious!

Growth of the Internet

- Paper claims that number of Internet users are doubling every year.
- The paper is from 1997 so over-estimates the growth a bit.
- Need for 50Gbps router, however, was real.

In 2014

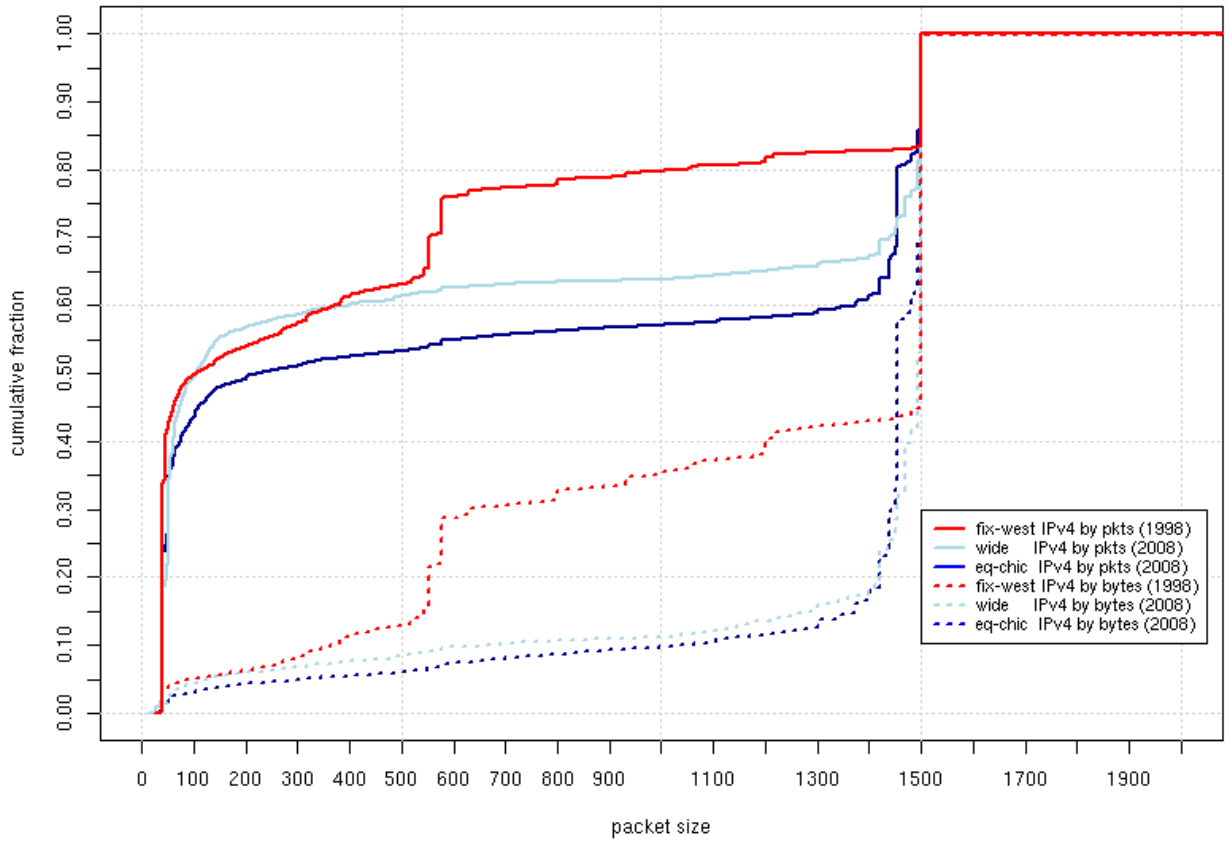


- Number of users and bandwidth requirement did not double every year.
- Cisco 7609 has 256 Gbps bandwidth
- Alcatel Lucent and BT tested 1.4Tbps link speed on 21 Jan 2014

Cisco 7609 Alcatel-Lucent BT

Average Packet Size

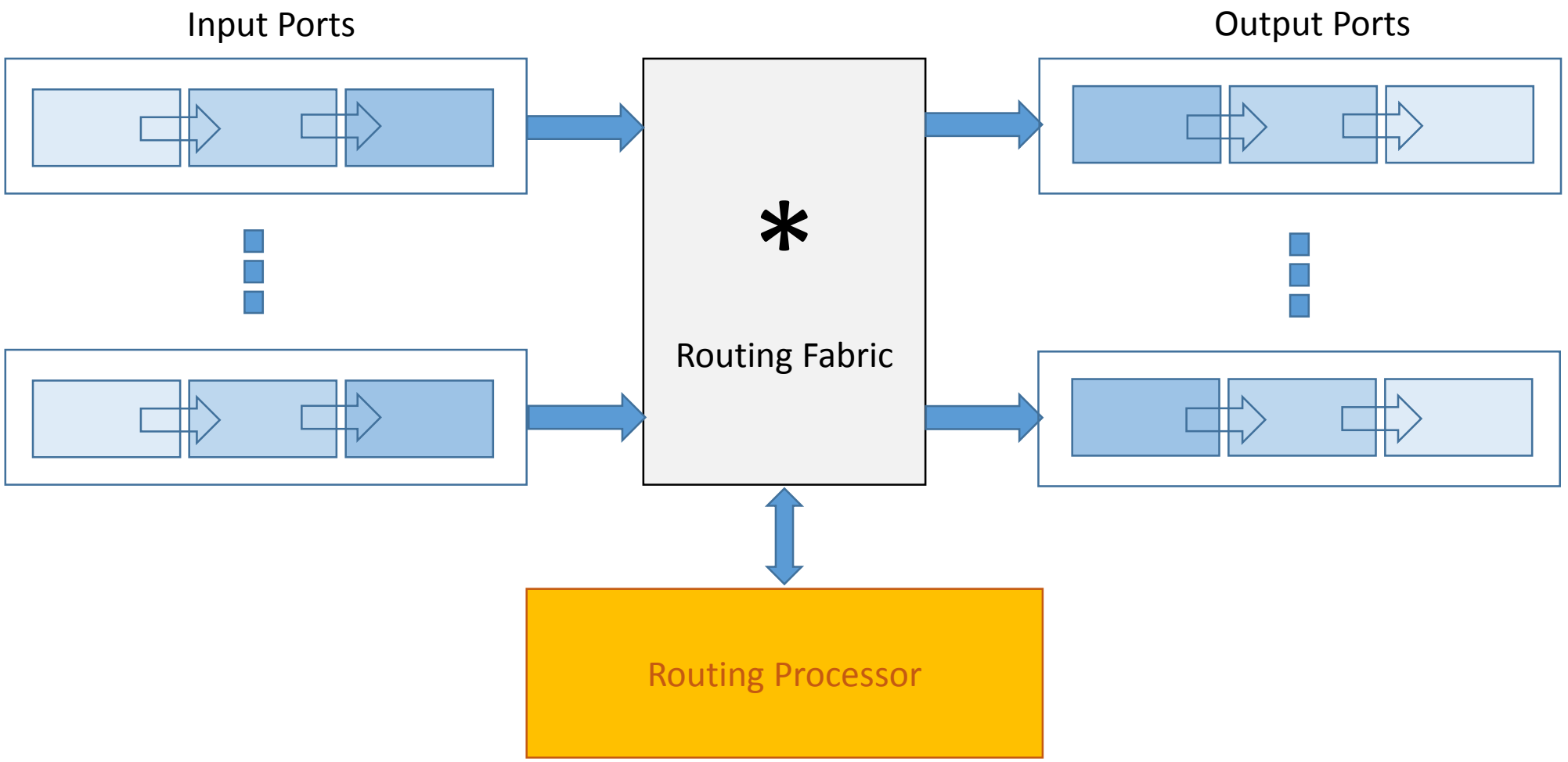
- The paper uses 1000bits as average packet size
- What to optimize?
- Number of packets processed or number of byte processed?



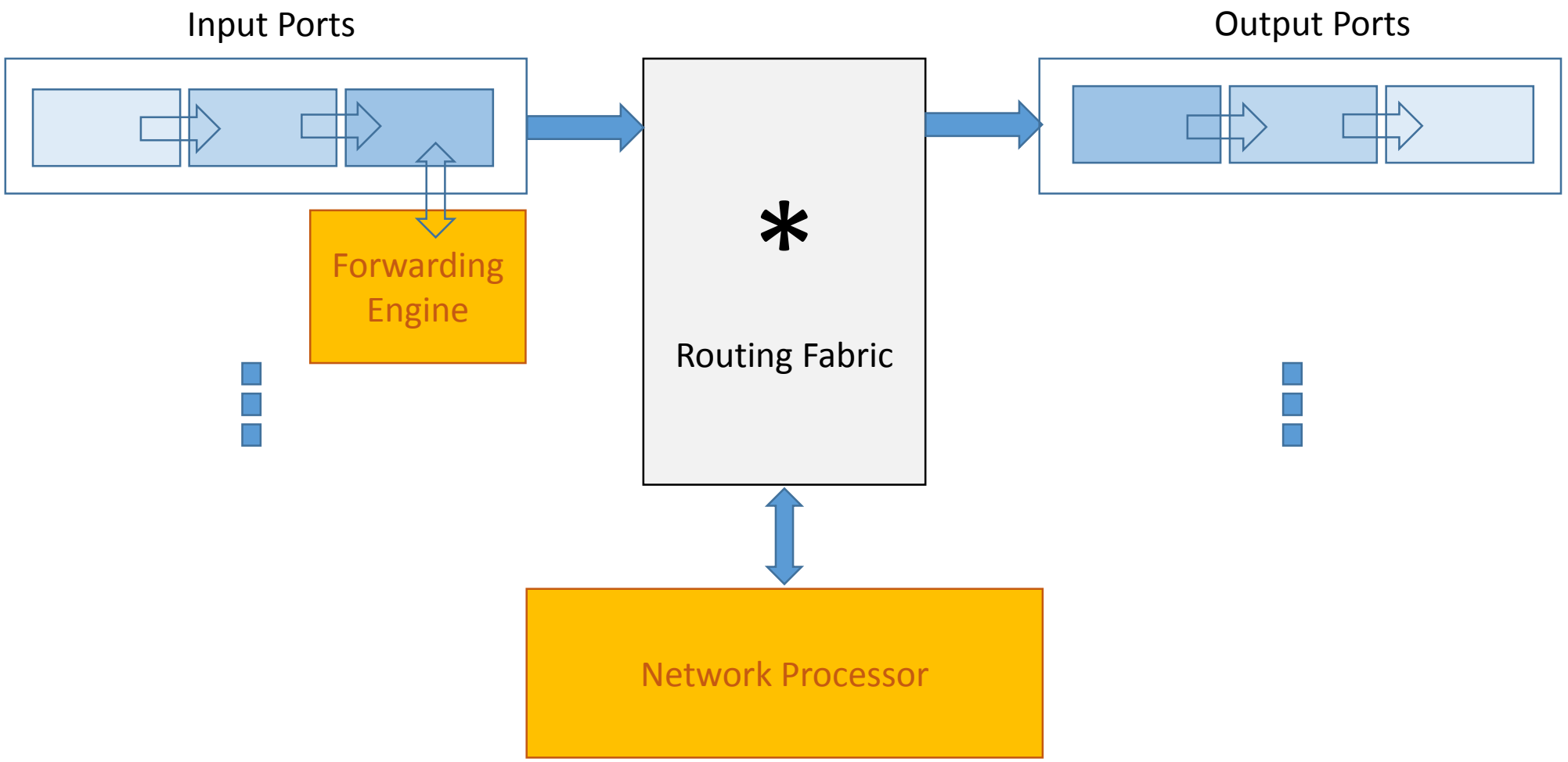
In 2014

- Avg 500-700
- Sharp rise at 1500

Router Components – A simplistic view

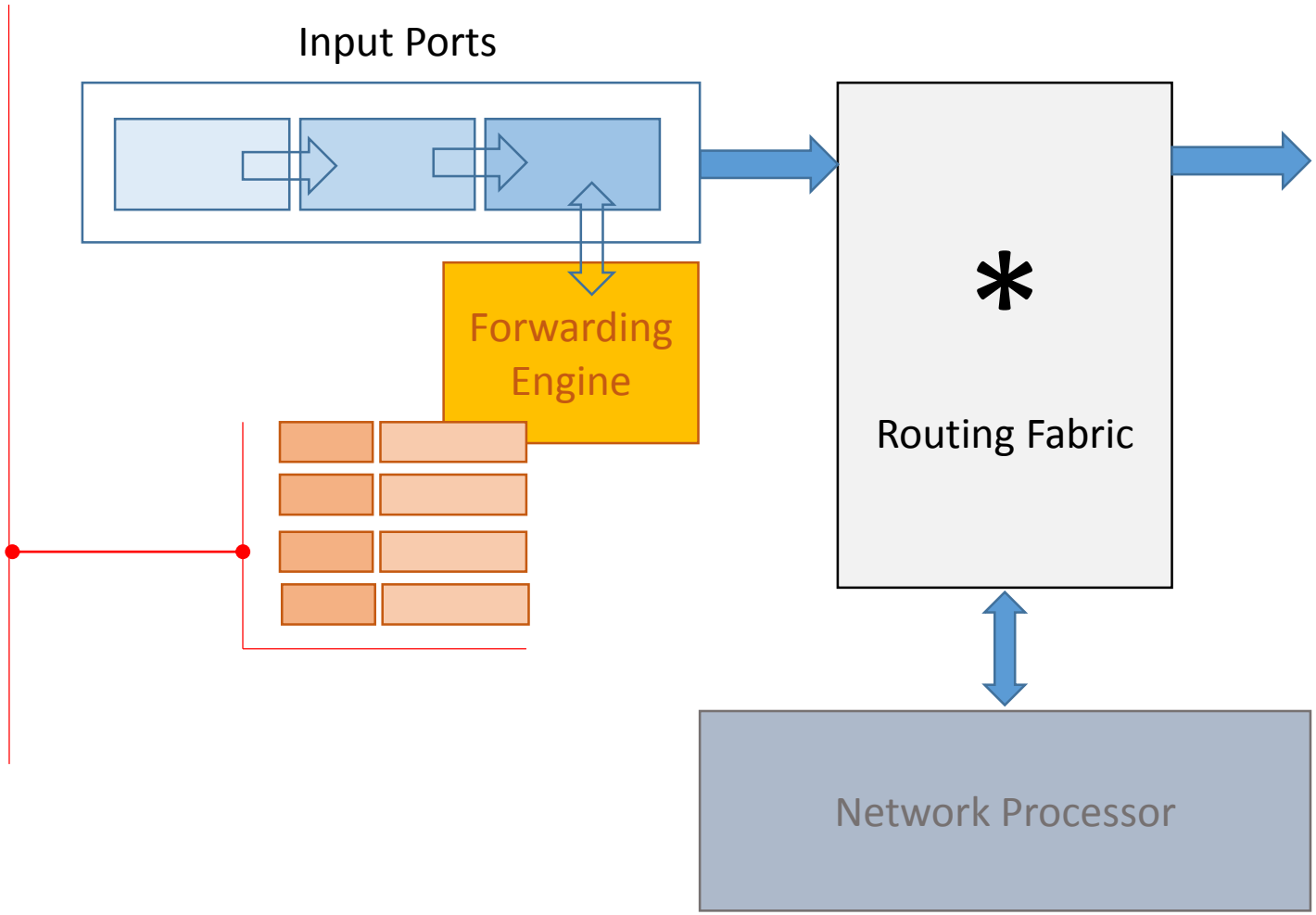


Router Components – Innovations



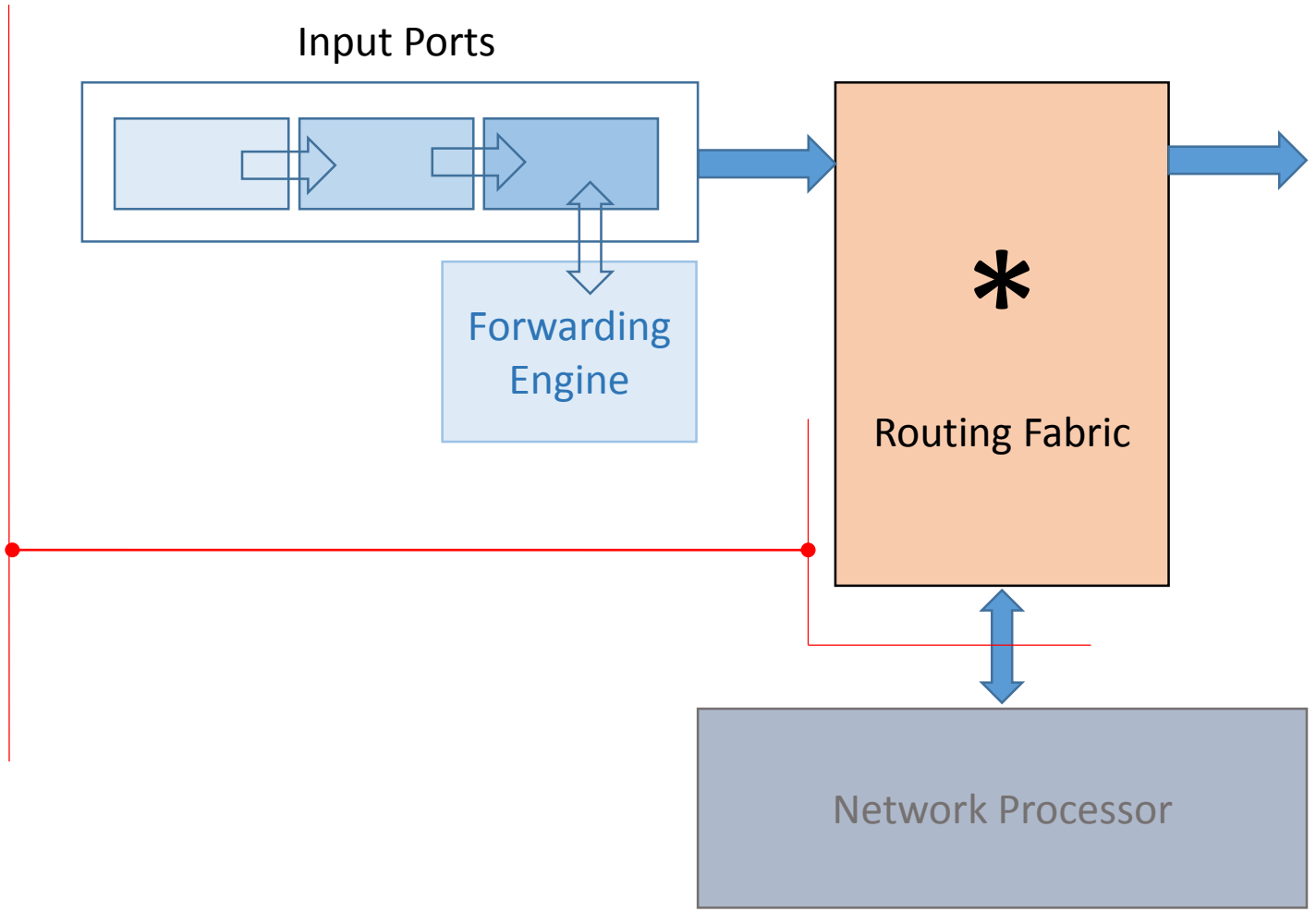
Innovations – Forwarding Tables

- Complete set of routing tables at the forwarding engine
- Reduces probability of cache miss to a great degree
- Next Hop info only



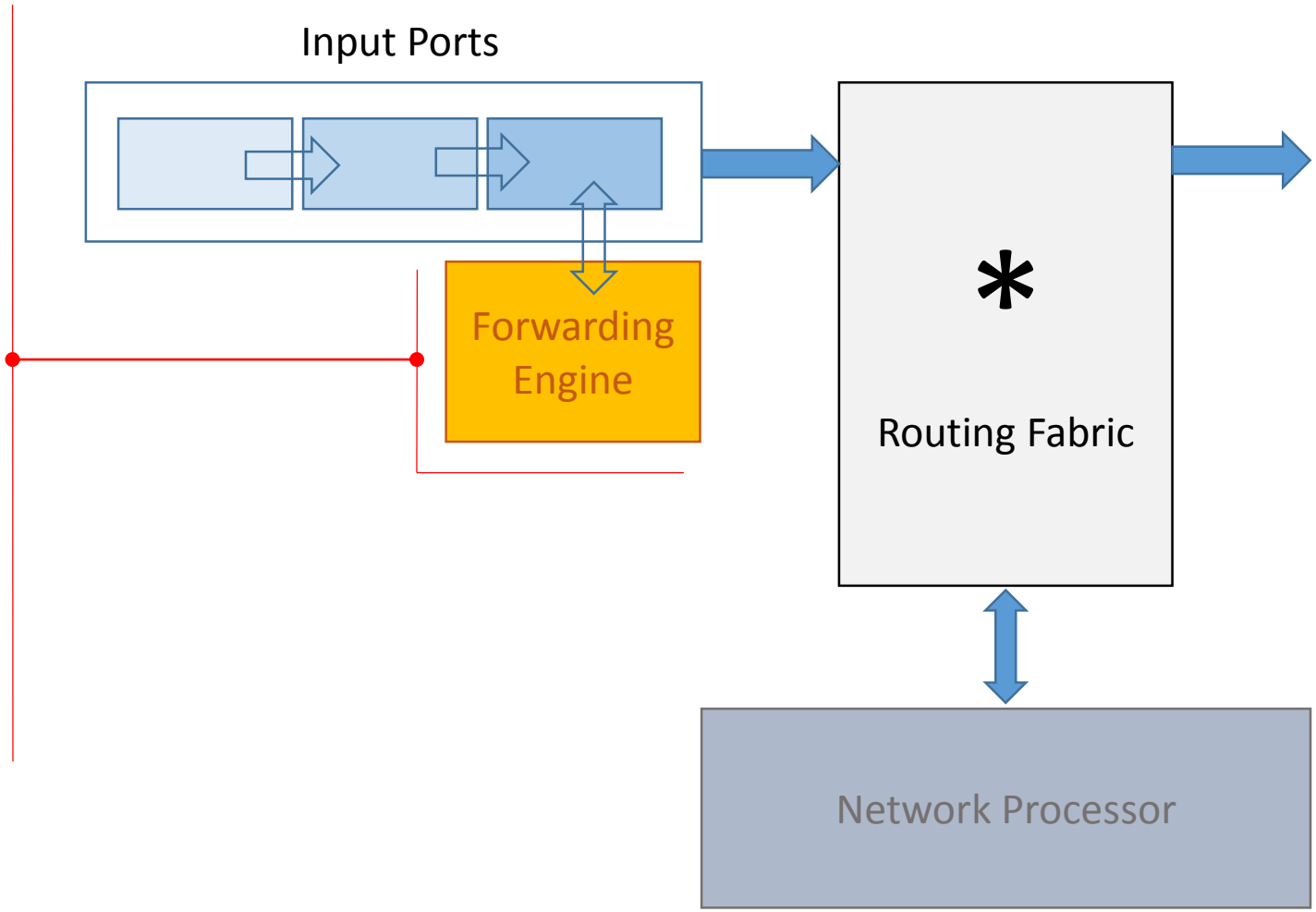
Innovations – Switched Backplane

- Switched backplane instead of shared bus
- Improves speed



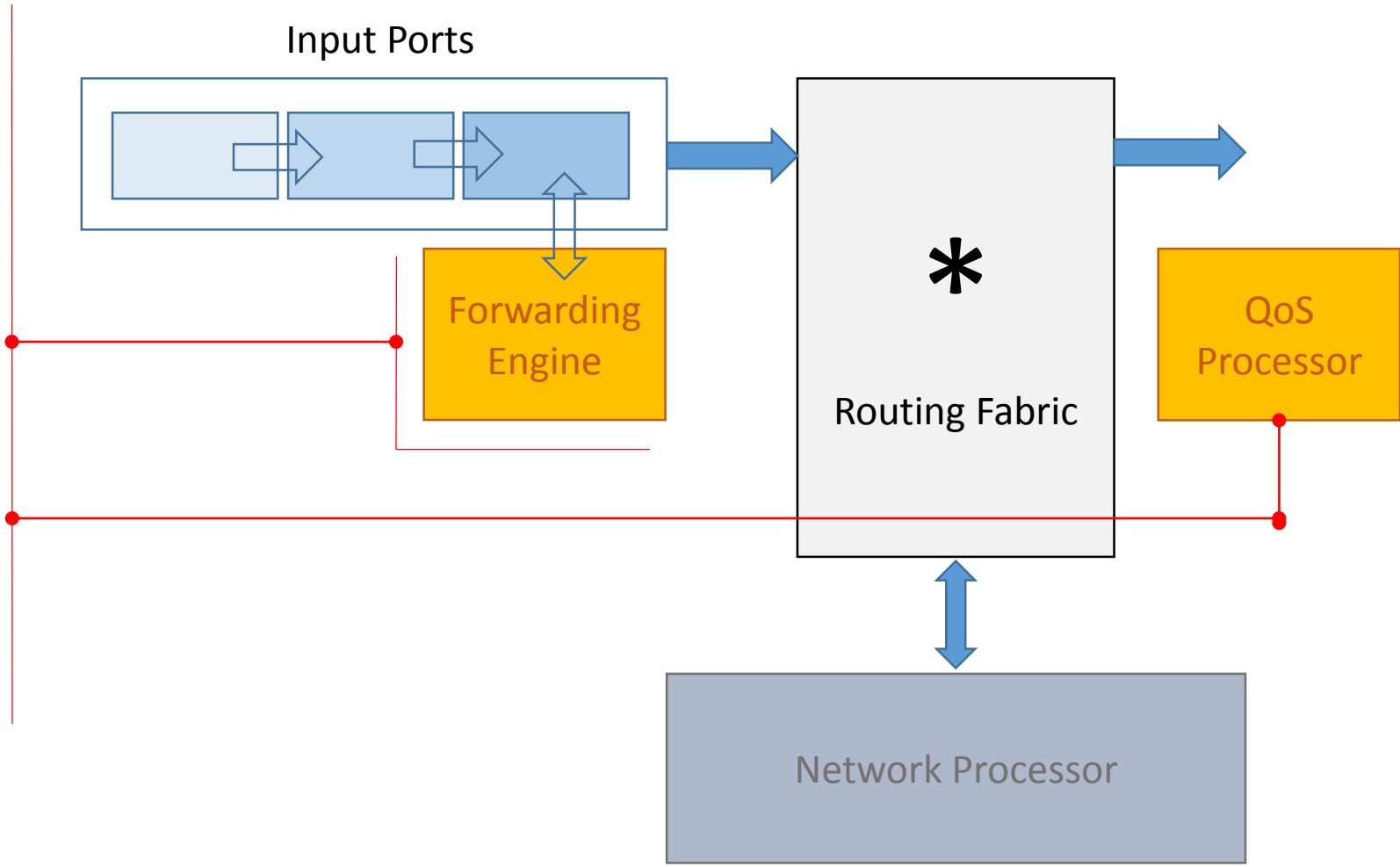
Innovations – separate board forward engine

- Forwarding engines on distinct boards
- Allows flexibility
- Allows various link layer protocols on the link card



Innovations – QoS

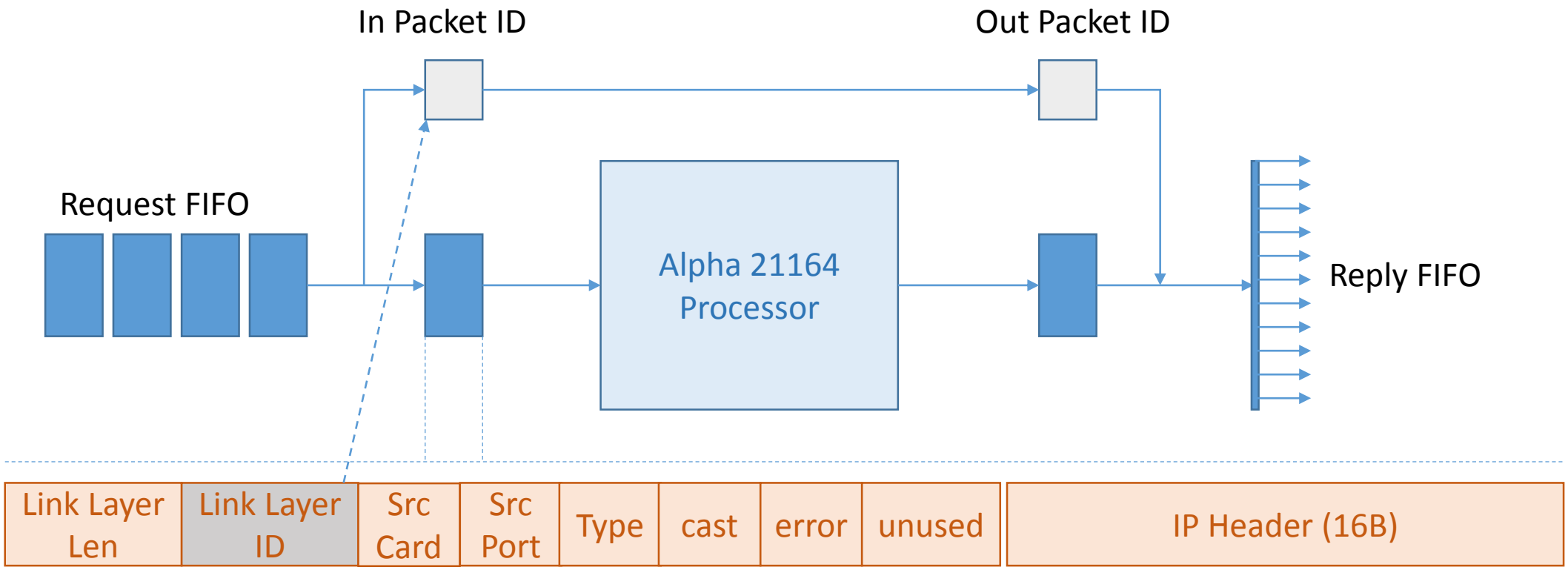
- QoS function is added
- Forwarding engine classifies by assigning it to a flow
- QoS processor on the outbound card scedules the packet



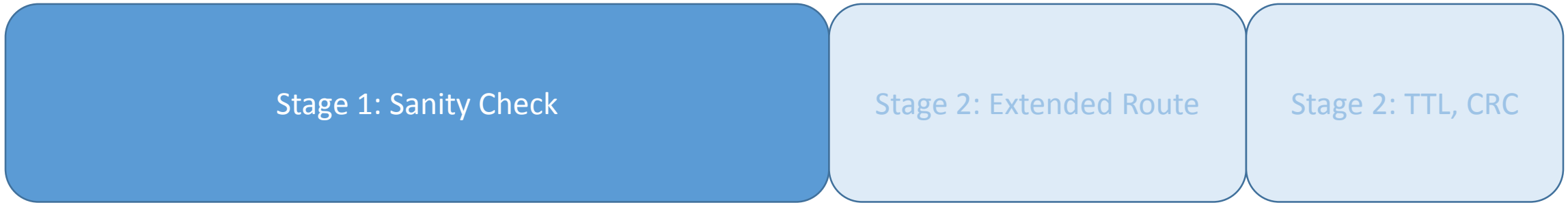
The Forwarding Engines – Hardware Overview

- Alpha 21164 Processor
  - 415MHz, 64 bit, 32 regs
  - 2 int, 2 float exec units
  - 4 instruction groups
  - 3 internal, 1 ext cache
  - Very high data speeds
  - Large instruction cache
  - Large secondary cache
  - Control over r/w sequencing
- Interesting tweaks
  - Place two pairs of int instructions together
  - Or, place 1 pair of int and 1 pair of float instruction
  - Fit entire fwd code in 8kB Icache
  - Fit 12000 routes in Scache [95%]
  - Fit all routes in Bcache
  - Allow the network processor to update this cache

Forwarding Engine – Hardware Operation

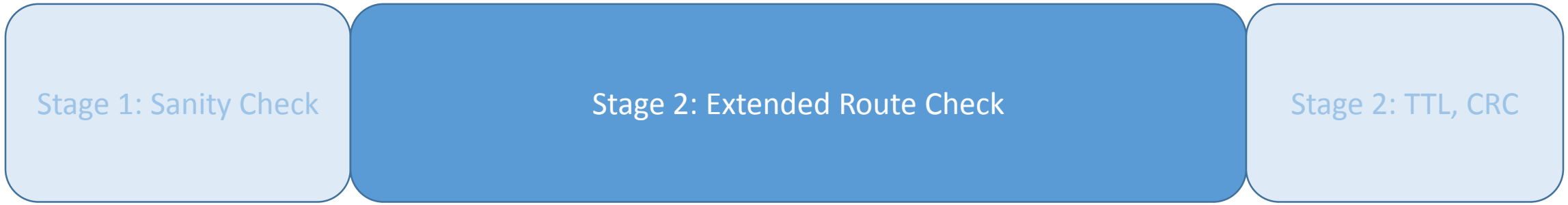


Forwarding Engine – Software



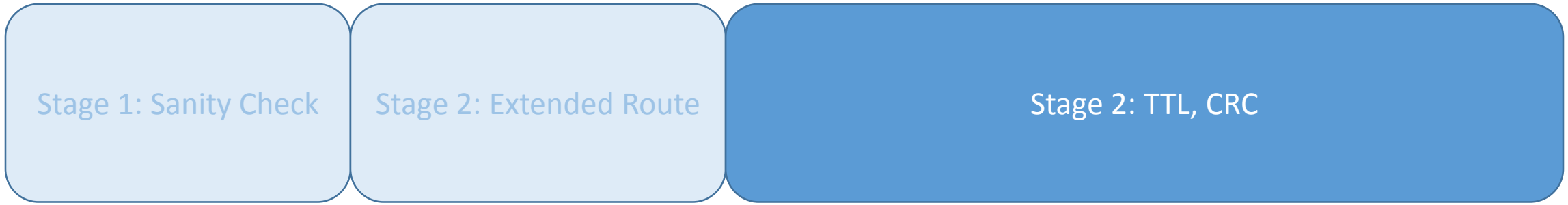
- Confirm that packet header is from IPv4 datagram
- Confirm packet length and header length is reasonable
- Confirm no IP options are used
- Compute hash offset into route cache and load route
- Start loading the next header

Forwarding Engine – Software



- Does cached route match the destination?
- If not, consult the Bcache
- Is packet destined for this router? Then don’t update TTL
- Update TTL otherwise

Forwarding Engine – Software



- Updated TTL is written to IP header
- New checksum is written
- Routing info is extracted
- New link layer info computed
- Entire packet is written out

Instructions in Fast Path

- Some hardware specific tweaks are employed
- `nop` and `fnop` used to pad instructions – eq to efficiency gains in x86
- Many bit operations required to extract data

Instructions	Count	%	Exec Unit
And, bic, bis, ornot, xor	24	28	E0/E1
ext*, ins*, sll, srl, zap	23	27	E0
add*, sub*, s*add	8	9	E0/E1
branches	8	9	E1
ld*	6	7	E0/E1
addt, cmpt*, fcmov*	6	7	FA
st*	4	5	E0
fnop	4	5	FM
wmb	1	1	E0
nop	1	1	E0/E1

Tricks and Exceptions

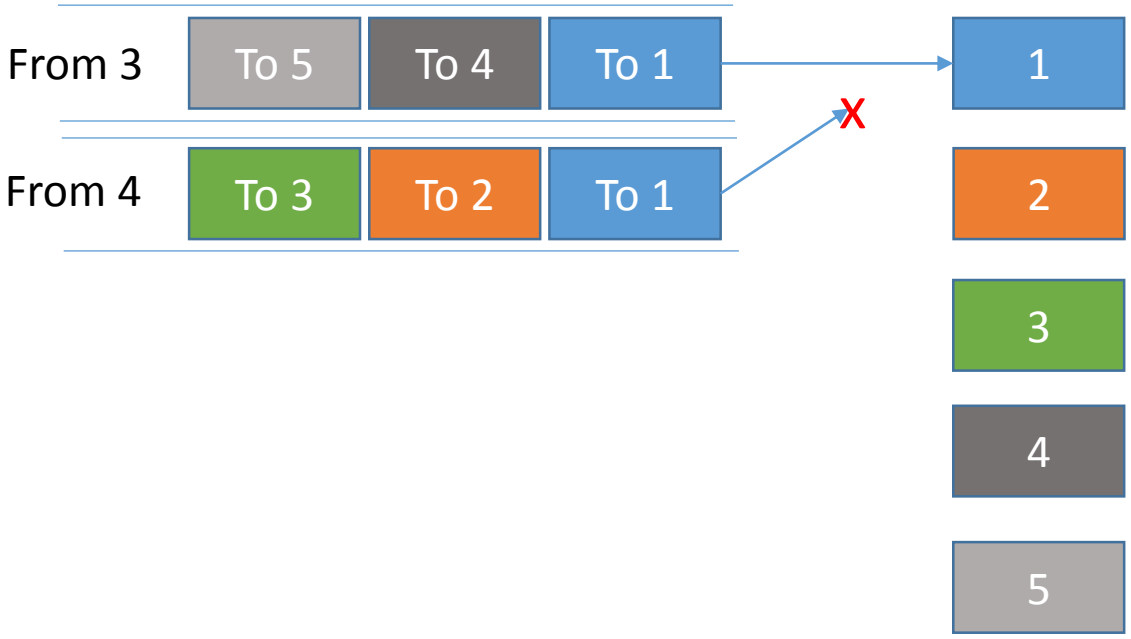
- IP header checksum is not checked
  - Saves 9 cycles – about 21%
  - Errors are rare. They are corrected by TCP anyways
  - IPv6 does not implement error checking
  - What do modern routers do?
- What is not handled by fast code?
  - Destinations that miss the route cache
  - Headers with errors
  - Headers with IP options
  - Datagrams that require fragmentation
  - Multicast datagrams

Switched Bus

- 15 port crossbar type switch is used
- Connects one source port to one destination port
- Multicast requires lowering of throughput and possible inherent fairness issues at any moment
- During a transfer 15 simultaneous transfers can happen
- Remember, only one output packet per port

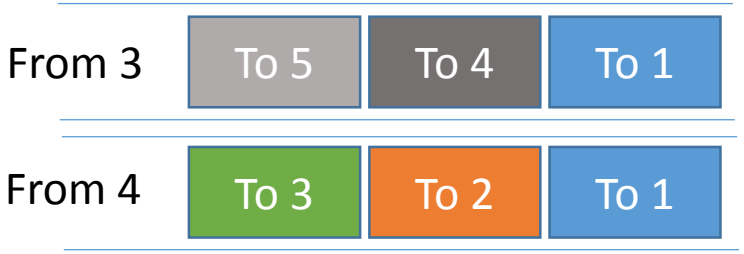
Head of line Blocking

- Head of line blocking occurs when the first packet in a queue is blocked because its destination interface is busy
- This causes all subsequent packets to be blocked even if their dest interfaces are free
- Avoided in a scheme where an interface gets to disclose all interfaces it is interested in



Matrix of Interests

- Disclose all interfaces for which your queue has at least one packet
- Allows transfer of a packet even from behind the queue



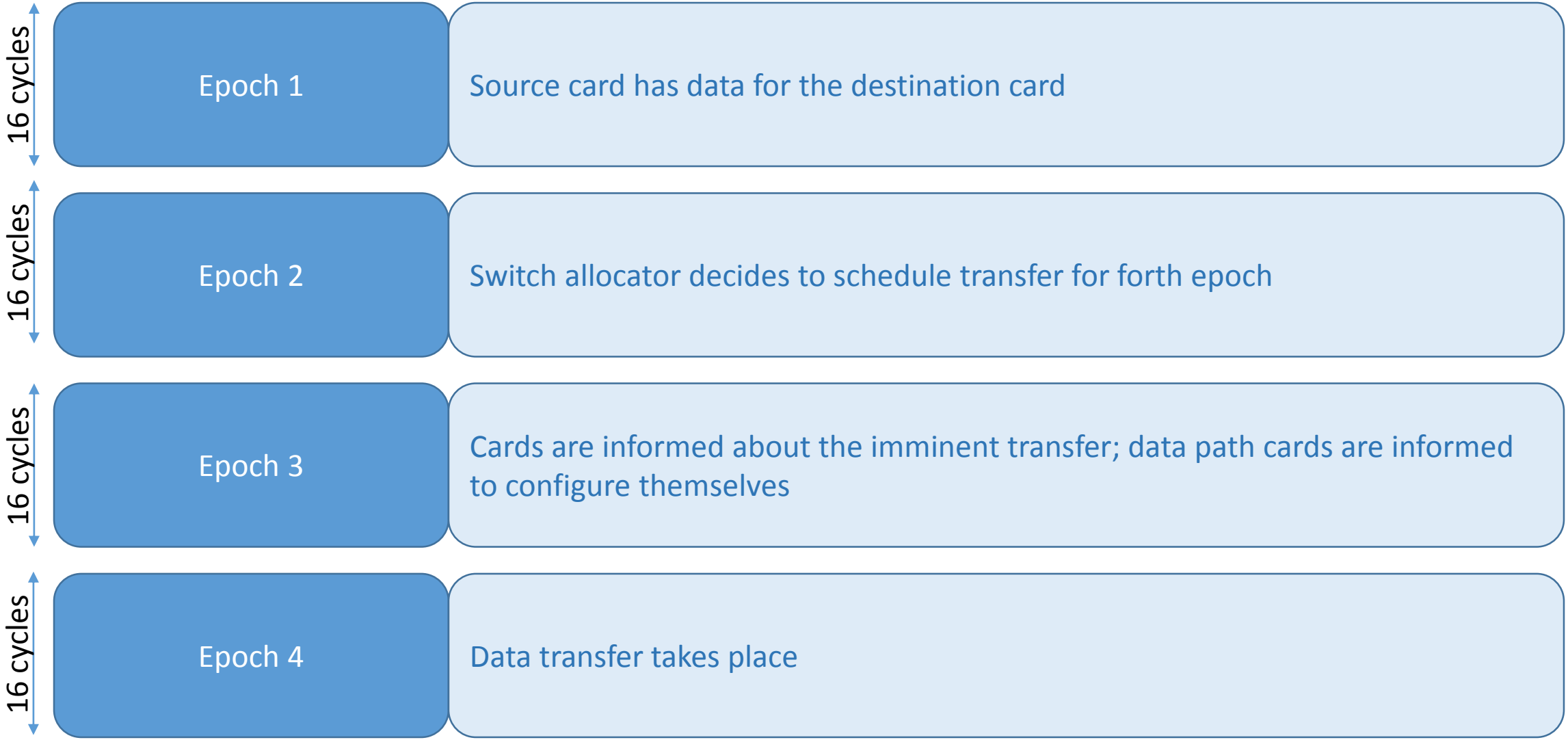
x	1	2	3	4	5
1			x	x	
2				x	
3				x	
4			x		
5			x		

Wavefront Method

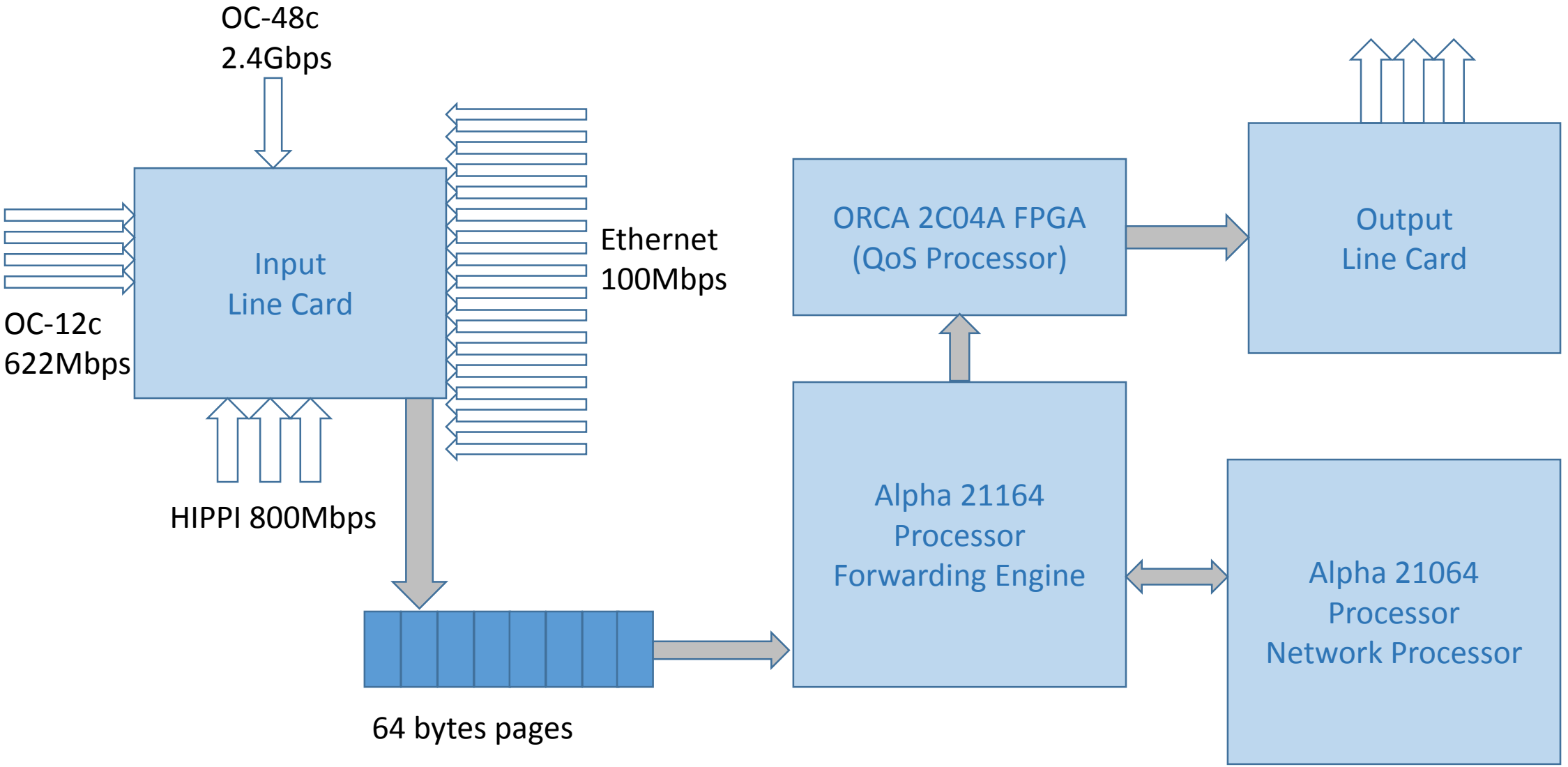
- By allowing a particular transfer to go through, the algorithm disallows few other possible transfers
- Therefore, instead of rasterizing through all entries look at only the allowed entries.
- This speeds up the algo by reducing the number of comparisons
- 3 is allowed to send to 1. No need to check 4→1 and 5→1

x	1	2	3	4	5
1			x	x	
2				x	
3				x	
4			x		
5			x		

Scheduling of switch



Line Card Design, Inbound, Outbound Packets



Conclusions

- Provides a huge impetus to the router industry to start working towards faster core routers.
- It shows that it is possible to examine every packet header and make decisions fast enough for multi Gbps routers
- It is unique in discussing the actual assembly level tricks to achieve the required throughput
- The paper creates an appreciation of the issues faced by the core routers on the Internet

Routers can keep up!

Right selection of hardware, placement and software is crucial.

Open Questions

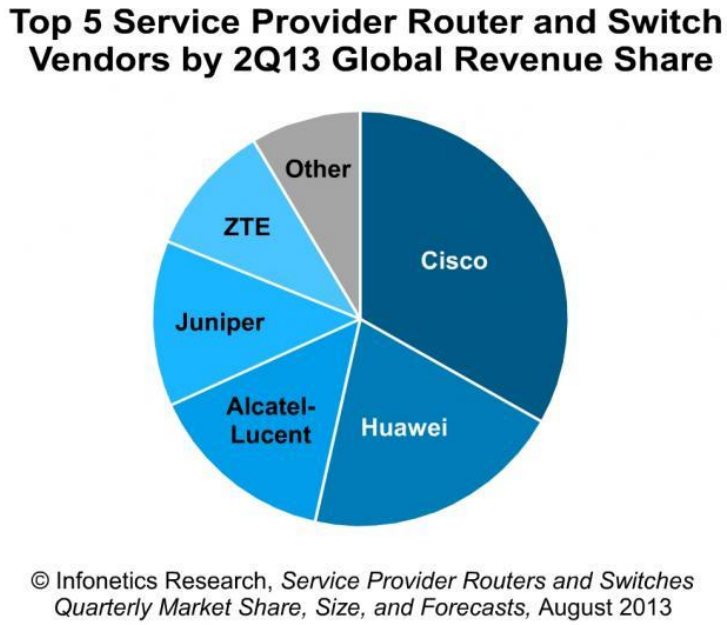
- The network processor ARPs all possible addresses at a low frequency. Is doing such a thing acceptable on the Internet backbone?
- When the Network Processor writes the newly discovered routes to the B-cache, how does it synchronize with possible reads from the forwarding engine?
- How often do the S-cache flush because of a change in B-cache?
- How expensive was it to build this router?
- No security aspects such as mitigation of DoS attacks not discussed
- No audit trails

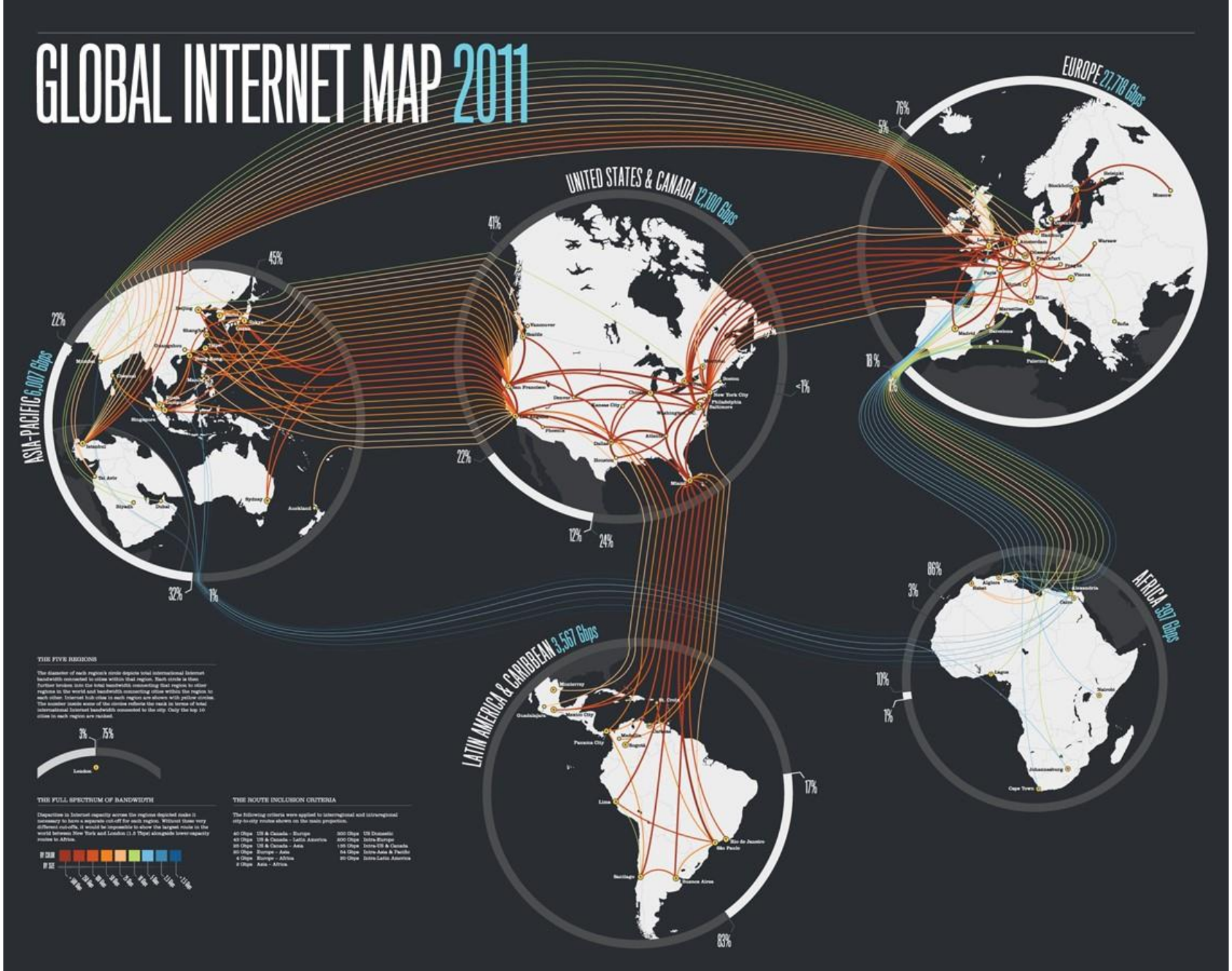
Open Questions

- The network processor ARPs all possible addresses at a low frequency. Is doing such a thing acceptable on the Internet backbone?
- When the Network Processor writes the newly discovered routes to the B-cache, how does it synchronize with possible reads from the forwarding engine?
- How often do the S-cache flush because of a change in B-cache?
- How expensive was it to build this router?
- No security aspects such as mitigation of DoS attacks not discussed
- No audit trails

Today's State of the Art

- Routing using GPU [link to paper]
- Cisco nPower X1, with 336 multi-threaded processor cores [link]
- The Alcatel-Lucent 7950 Extensible Routing System has 16Tbps bandwidth [link]
- Cisco's Carrier Routing System routes at 400Gbps per link [link]





Asia-Pacific
6007Gbps

US-Canada
12100Gbps

Europe
27718Gbps

Africa
397Gbps

Latin-America
3567Gbps

[reference]

Thank You!