The Router (Packet Switch) architectures

Wing C. Lau
IERG5090 Spring 2017

Acknowledgements

- The Slides used in this lecture are mostly adapted (with permission) from a series of talks by:
  - Prof. Nick Mckeown and his collaborators/ graduate students from Stanford University.
  - Prof. Jim Kurose (UMass) and Prof. Keith Ross (Polytechnic of NY)
  - Prof. Isaac Keslassy of Technion
  - Information about netflow from Cisco and Juniper web site

Copyrights of these materials belong to the original copyright holders and their contributions are hereby acknowledged.

Outline

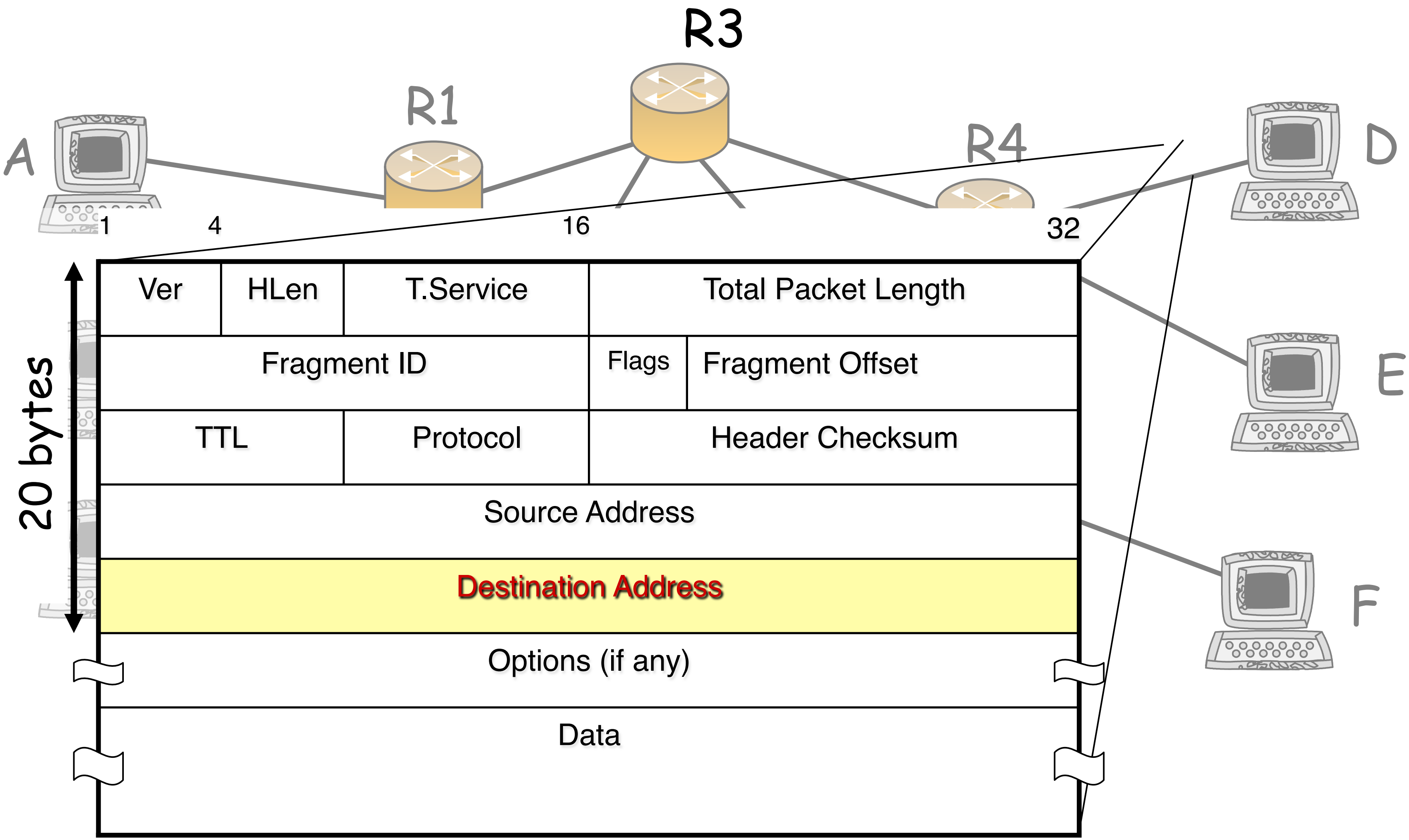
Background

- What is a router?
- Why do we need faster routers?
- Why are they hard to build?

Architectures and techniques

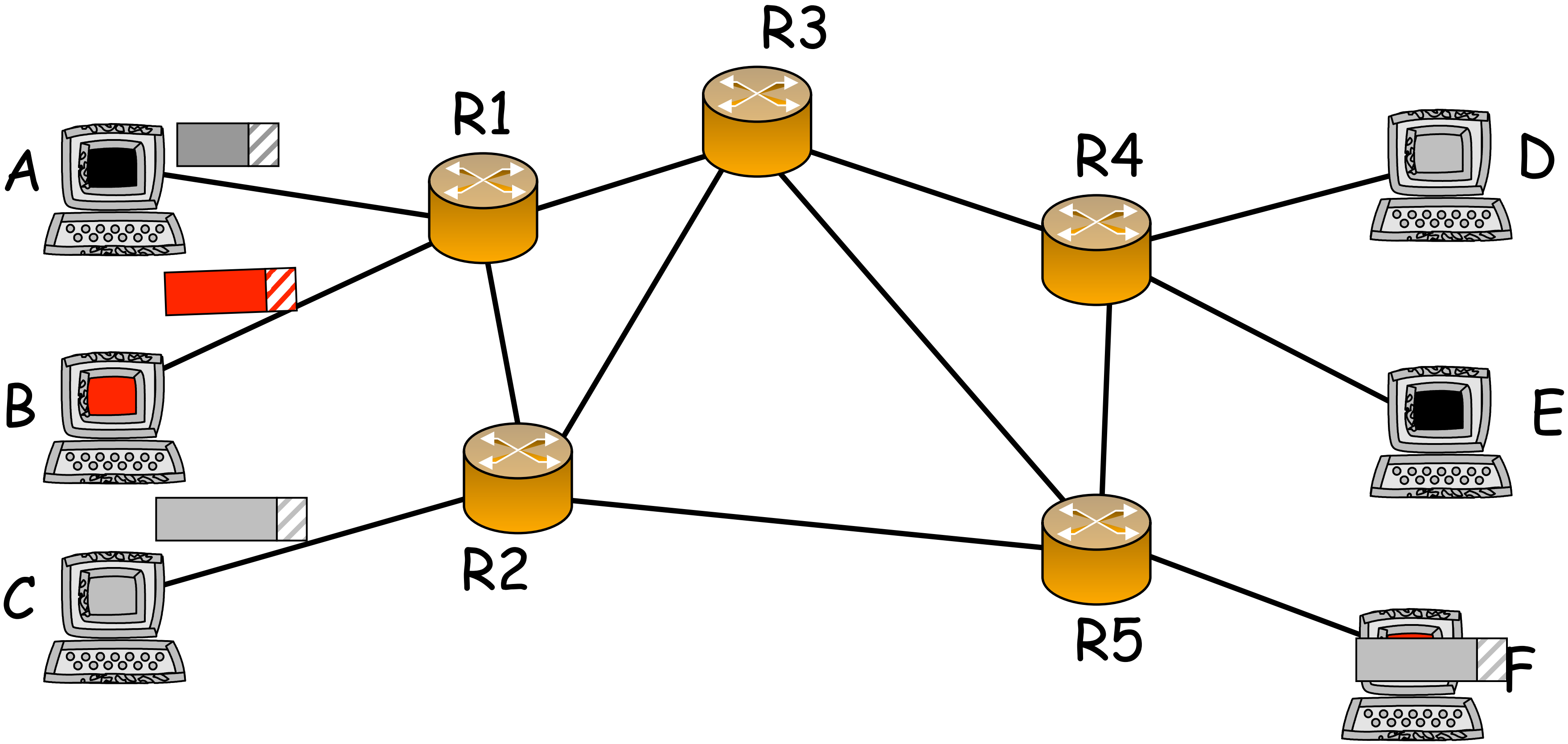
- The evolution of router architecture.
- Packet Classification
- IP address lookup
- Packet buffering.
- Switching.
- Example Switching Architectures in practice and in theory
- Future Directions

What is Routing?

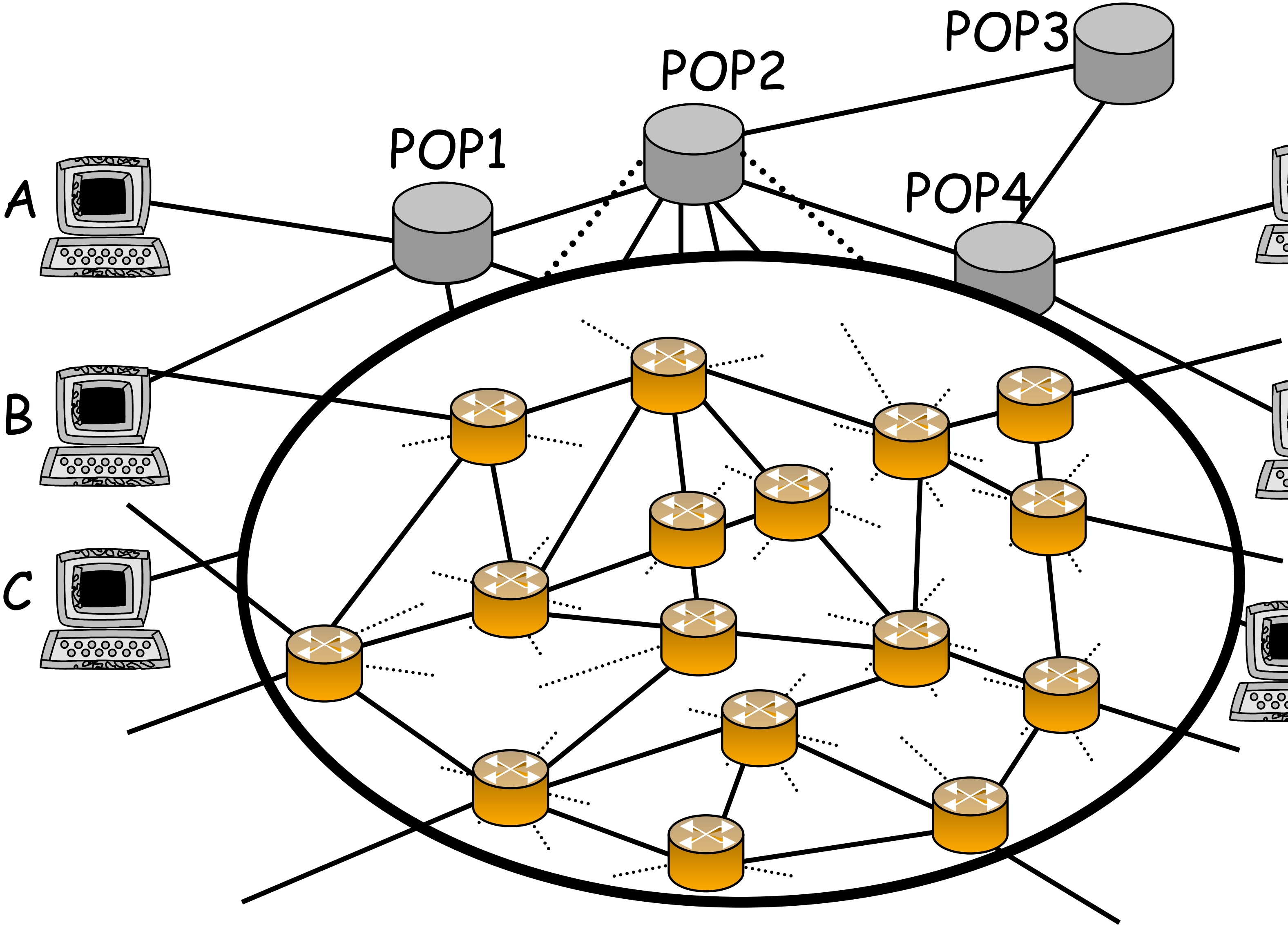


1	4		16		32
Ver	HLen	T.Service	Total Packet Length		
Fragment ID			Flags	Fragment Offset	
TTL		Protocol	Header Checksum		
Source Address					
Destination Address					
Options (if any)					
Data					

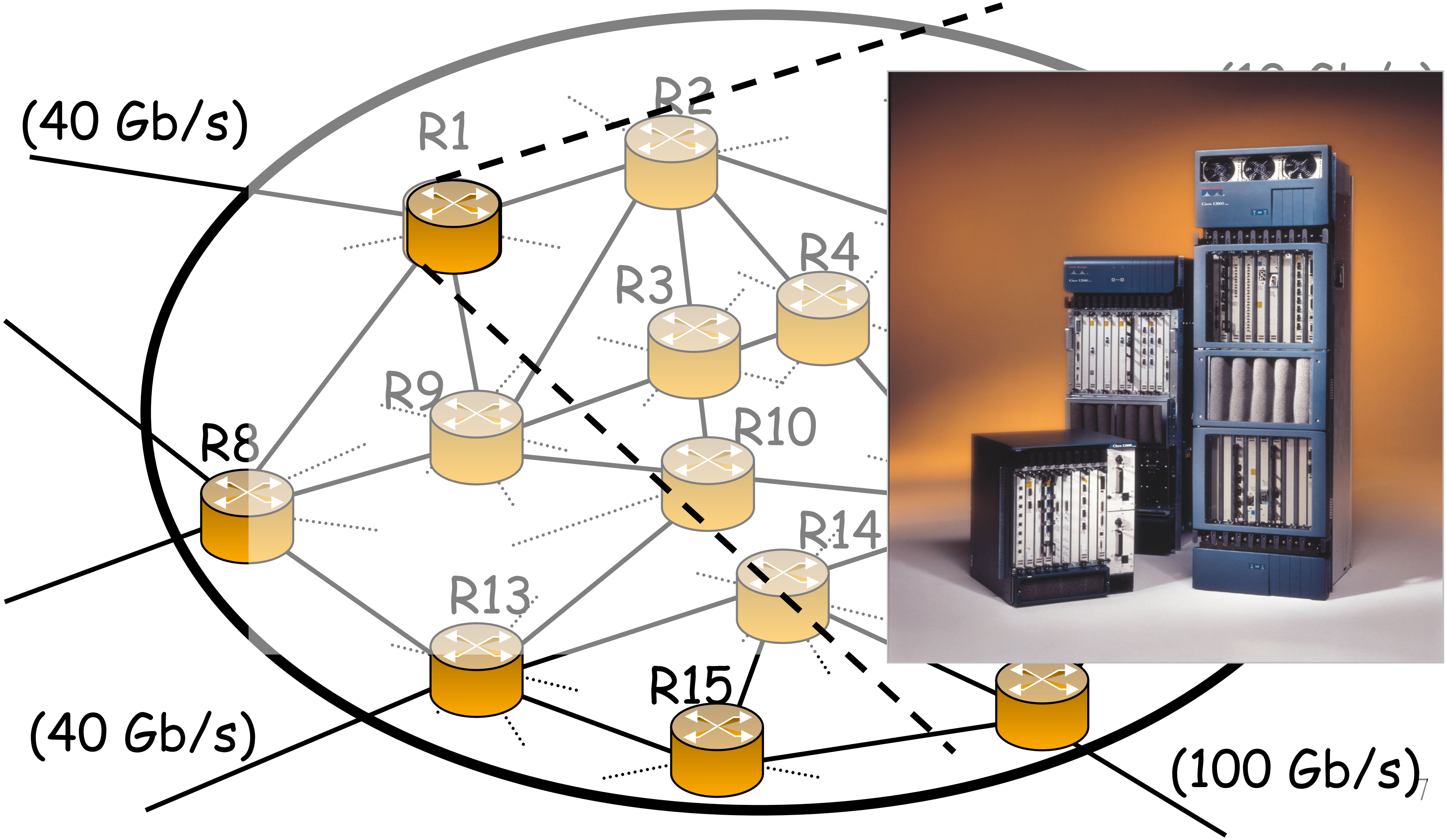
What is Routing?



Points of Presence (POPs)



Where High Performance Routers are Used



Current Generation of Cisco's Carrier-class Routing System (available since 2H2013)

Cisco CRS-X

Capacity:

400 Gbps per slot ;
16 slots per Chassis
=> 6.4Tbps switching capacity per Chassis
Power Consumption in the order of 10kw per Chassis

Architecture can interconnect upto 72 chassis together, i.e. 1152 slots in total
=> 460Tbps aggregated Switching capacity

Competing Product (current Gen.) from Juniper

The Juniper T4000 Core Router

Capacity:

3.84 Tbps switching capacity, and
2.4 billion packets per sec (pps)
Per HALF-RACK Chassis

240 Gbps per slot

Each T4000 supports upto

- 208 ports of 10GbE or
- 16 ports of 40GbE Interfaces
- 16 ports of 100GbE Interfaces ;

Competing Product (current Gen.) from Juniper

T4000

	Ports
208	10 Gbps
16	40 Gbps
16	100 Gbps

T1600

	Ports
80	10 Gbps
16	40 Gbps
8	100 Gbps

T640

	Ports
40	10 Gbps
8	40 Gbps

TX Matrix Plus

	Ports
832	10 Gbps
64	40 Gbps
64	100 Gbps

TX Matrix

	Ports
160	10 Gbps
32	40 Gbps

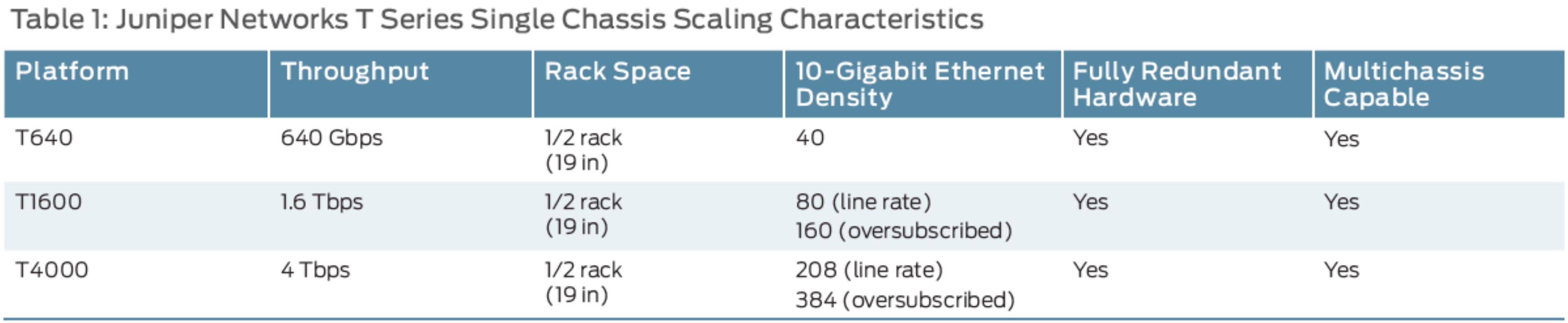
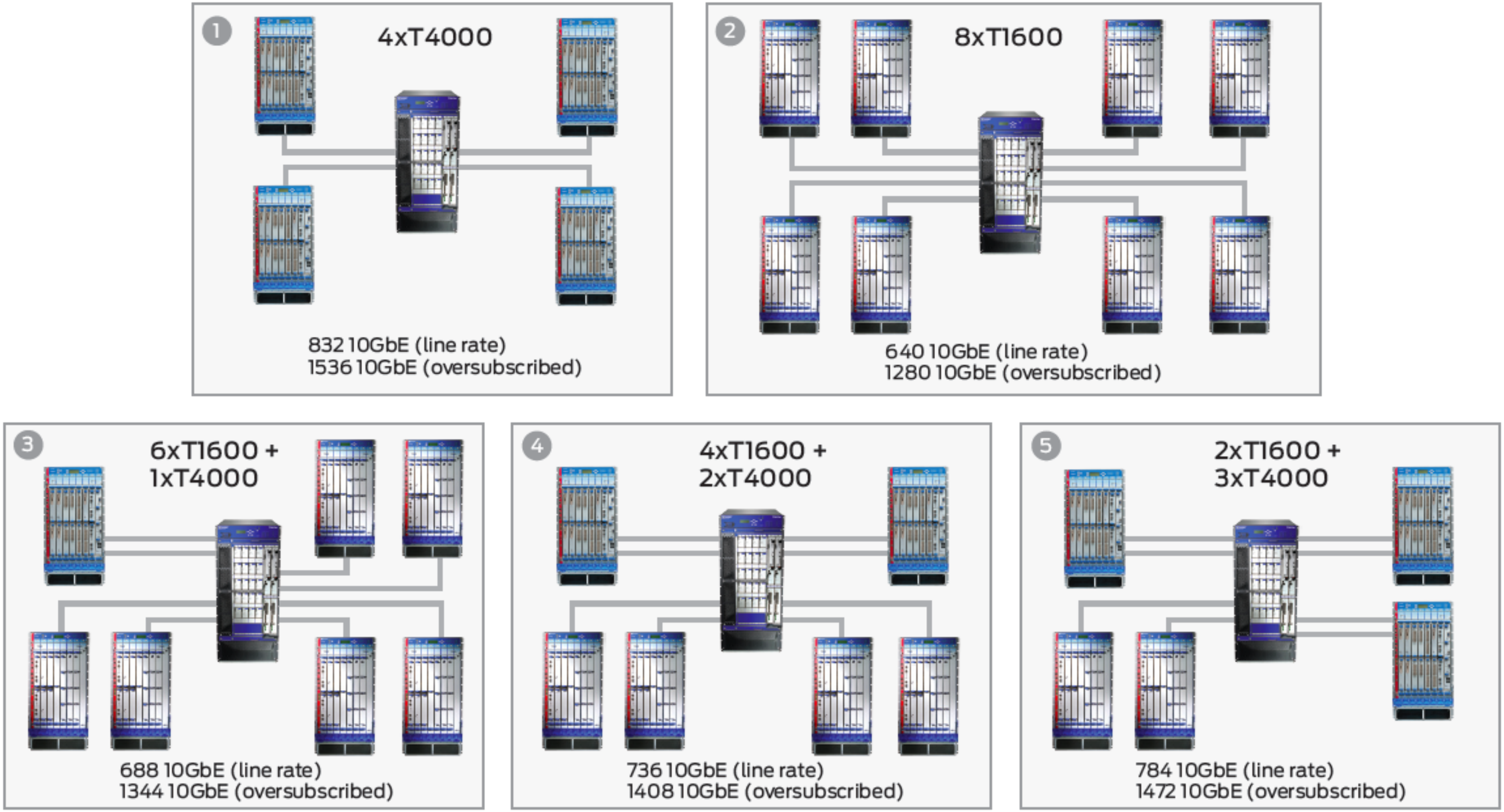


Table 1: Juniper Networks T Series Single Chassis Scaling Characteristics

Platform	Throughput	Rack Space	10-Gigabit Ethernet Density	Fully Redundant Hardware	Multichassis Capable
T640	640 Gbps	1/2 rack (19 in)	40	Yes	Yes
T1600	1.6 Tbps	1/2 rack (19 in)	80 (line rate) 160 (oversubscribed)	Yes	Yes
T4000	4 Tbps	1/2 rack (19 in)	208 (line rate) 384 (oversubscribed)	Yes	Yes

Competing Product (current Gen.) from Juniper



1
4xT4000

832 10GbE (line rate)
1536 10GbE (oversubscribed)

2
8xT1600

640 10GbE (line rate)
1280 10GbE (oversubscribed)

3
6xT1600 +
1xT4000

688 10GbE (line rate)
1344 10GbE (oversubscribed)

4
4xT1600 +
2xT4000

736 10GbE (line rate)
1408 10GbE (oversubscribed)

5
2xT1600 +
3xT4000

784 10GbE (line rate)
1472 10GbE (oversubscribed)

Competing Product (current Gen.) from Juniper

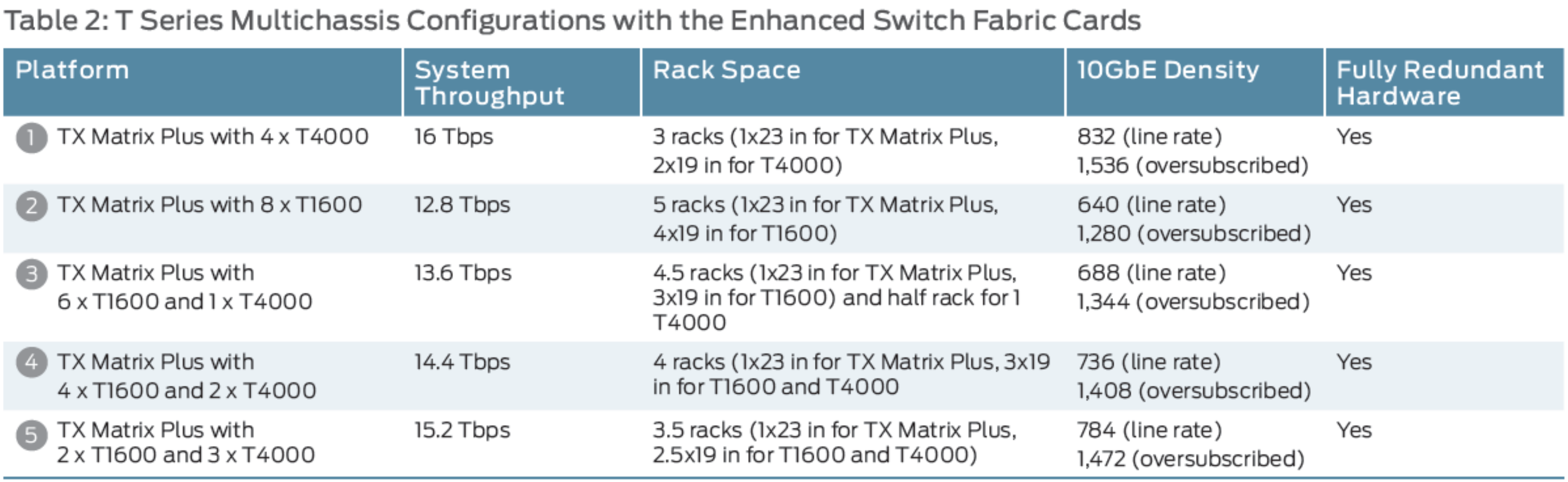
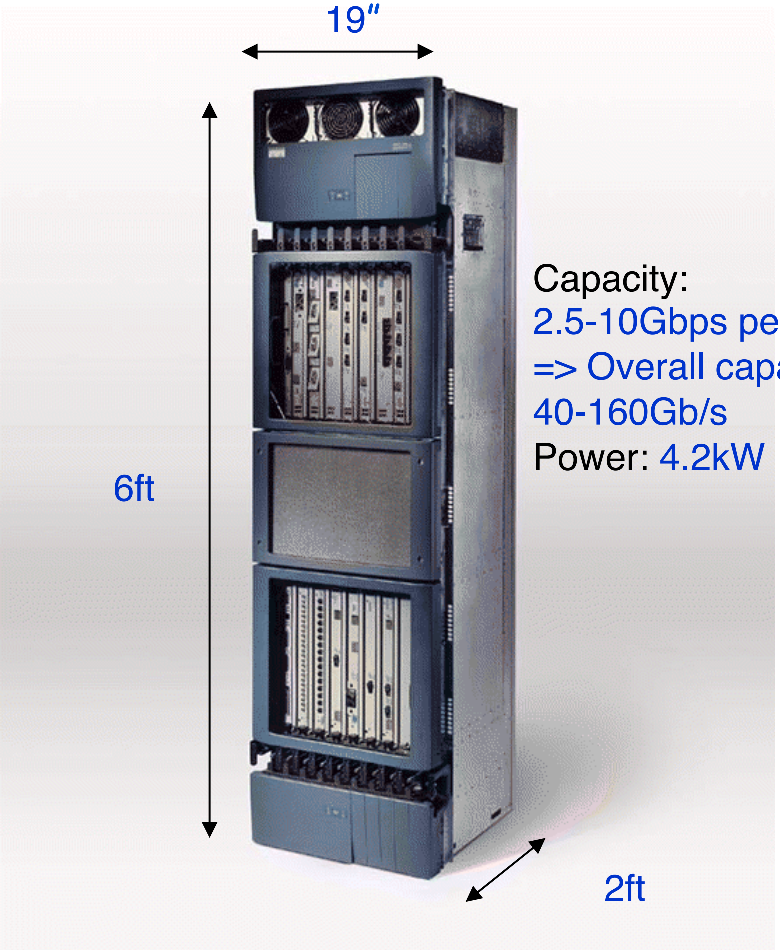


Table 2: T Series Multichassis Configurations with the Enhanced Switch Fabric Cards

Platform	System Throughput	Rack Space	10GbE Density	Fully Redundant Hardware
1 TX Matrix Plus with 4 x T4000	16 Tbps	3 racks (1x23 in for TX Matrix Plus, 2x19 in for T4000)	832 (line rate) 1,536 (oversubscribed)	Yes
2 TX Matrix Plus with 8 x T1600	12.8 Tbps	5 racks (1x23 in for TX Matrix Plus, 4x19 in for T1600)	640 (line rate) 1,280 (oversubscribed)	Yes
3 TX Matrix Plus with 6 x T1600 and 1 x T4000	13.6 Tbps	4.5 racks (1x23 in for TX Matrix Plus, 3x19 in for T1600) and half rack for 1 T4000	688 (line rate) 1,344 (oversubscribed)	Yes
4 TX Matrix Plus with 4 x T1600 and 2 x T4000	14.4 Tbps	4 racks (1x23 in for TX Matrix Plus, 3x19 in for T1600 and T4000	736 (line rate) 1,408 (oversubscribed)	Yes
5 TX Matrix Plus with 2 x T1600 and 3 x T4000	15.2 Tbps	3.5 racks (1x23 in for TX Matrix Plus, 2.5x19 in for T1600 and T4000)	784 (line rate) 1,472 (oversubscribed)	Yes

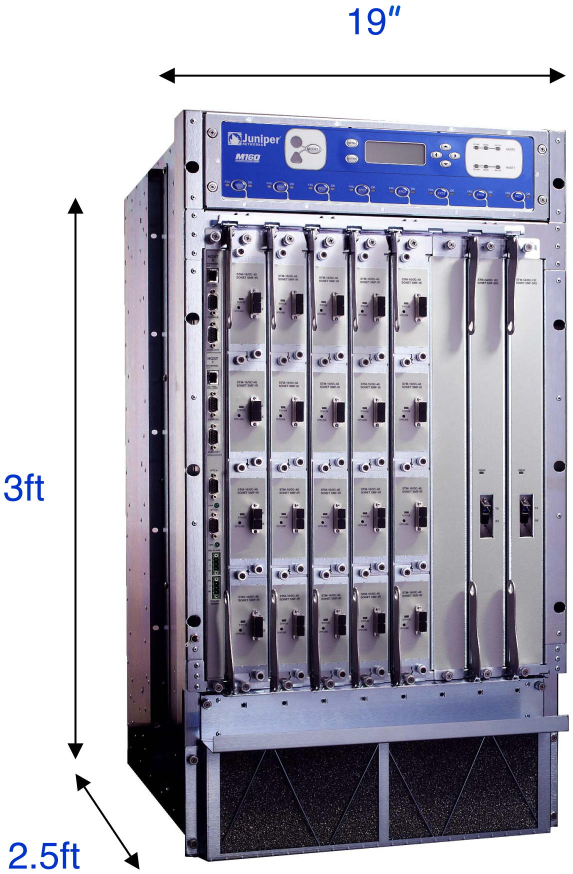
Top-of-the-line (~100 Gigabit) Routers around Year 2000-2002

Cisco GSR 12416



Capacity:
2.5-10Gbps per slot
=> Overall capacity:
40-160Gb/s
Power: 4.2kW

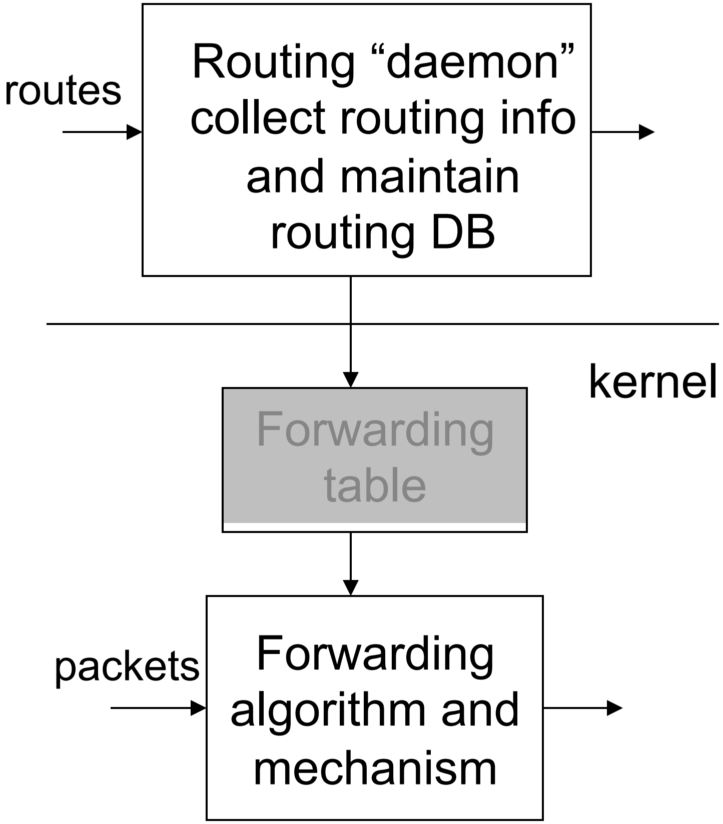
Juniper M160



Total Capacity: 80Gb/s
Power: 2.6kW

Two components of a Router

- Control Component
  - Decides where the packets will go
  - Use a set of routing protocols (e.g. OSPF, BGP) to collect information and produce a "forwarding table"
  - "Control plane"
- Forwarding component
  - Moving packets from input to output ports according to forwarding table and packet header
  - "Forwarding plane" to carry out per-packet processing


routes
Routing "daemon" collect routing info and maintain routing DB
kernel
Forwarding table
packets
Forwarding algorithm and mechanism

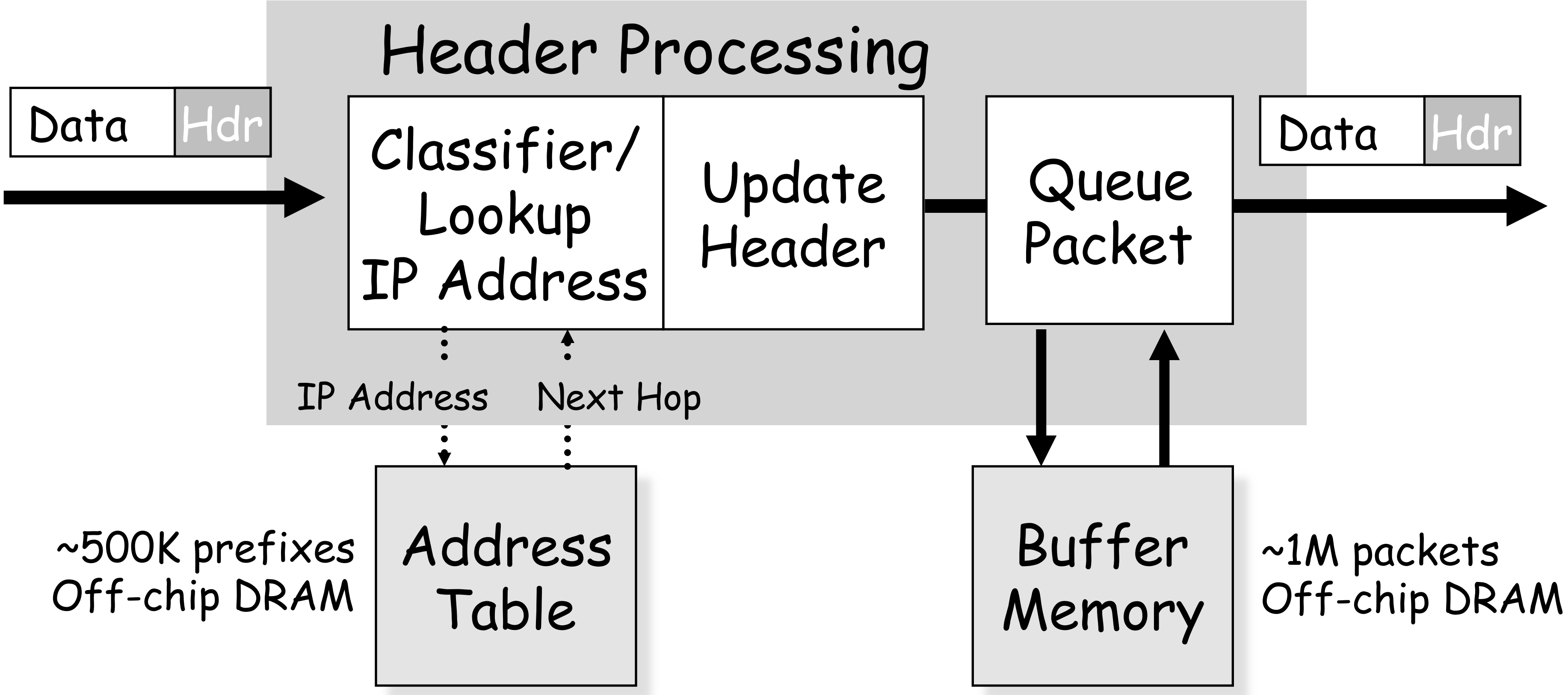
Per-packet processing in an IP Router

1. Accept packet arriving on an incoming link.
2. Packet Classification

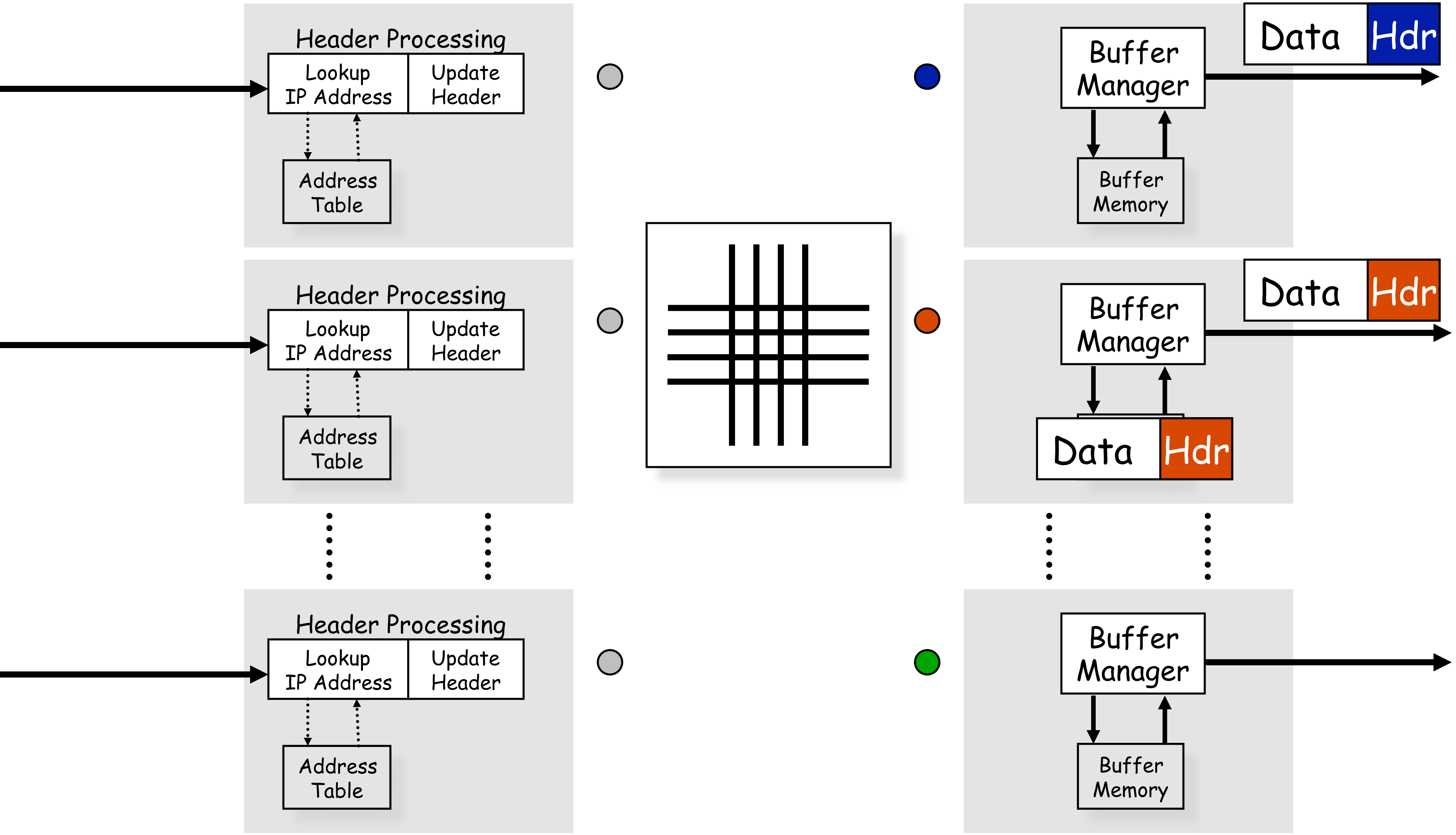
   e.g. to enable different QoS/priority treatment of different type of packets ; packet-filtering firewall

3. Lookup packet destination address in the forwarding table, to identify outgoing port(s).
4. Manipulate packet header: e.g., decrement TTL, update header checksum.
5. Send packet to the outgoing port(s).
6. Buffer packet in the queue.
7. Transmit packet onto outgoing link.

Generic Router Architecture



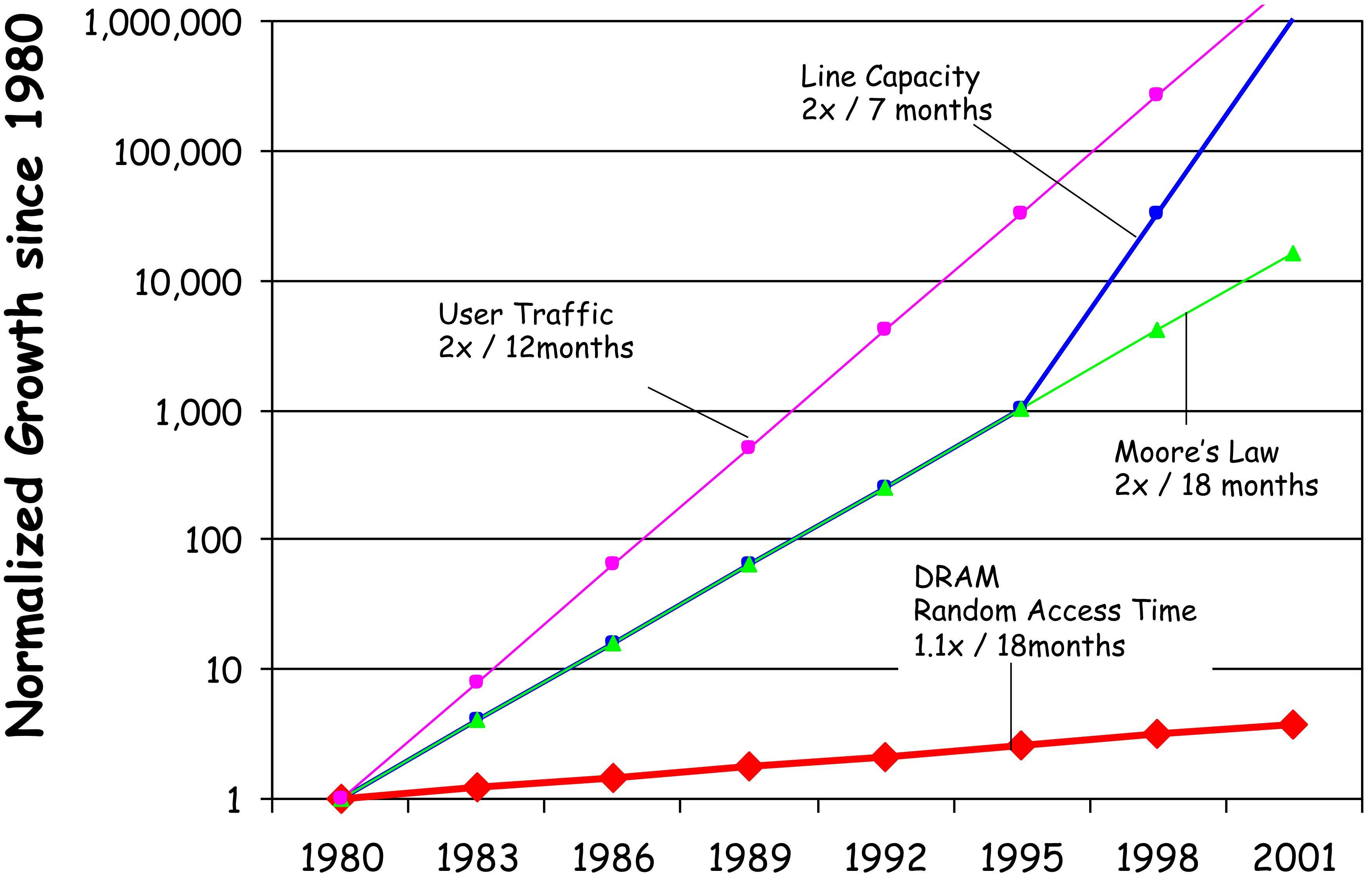
Generic Router Architecture



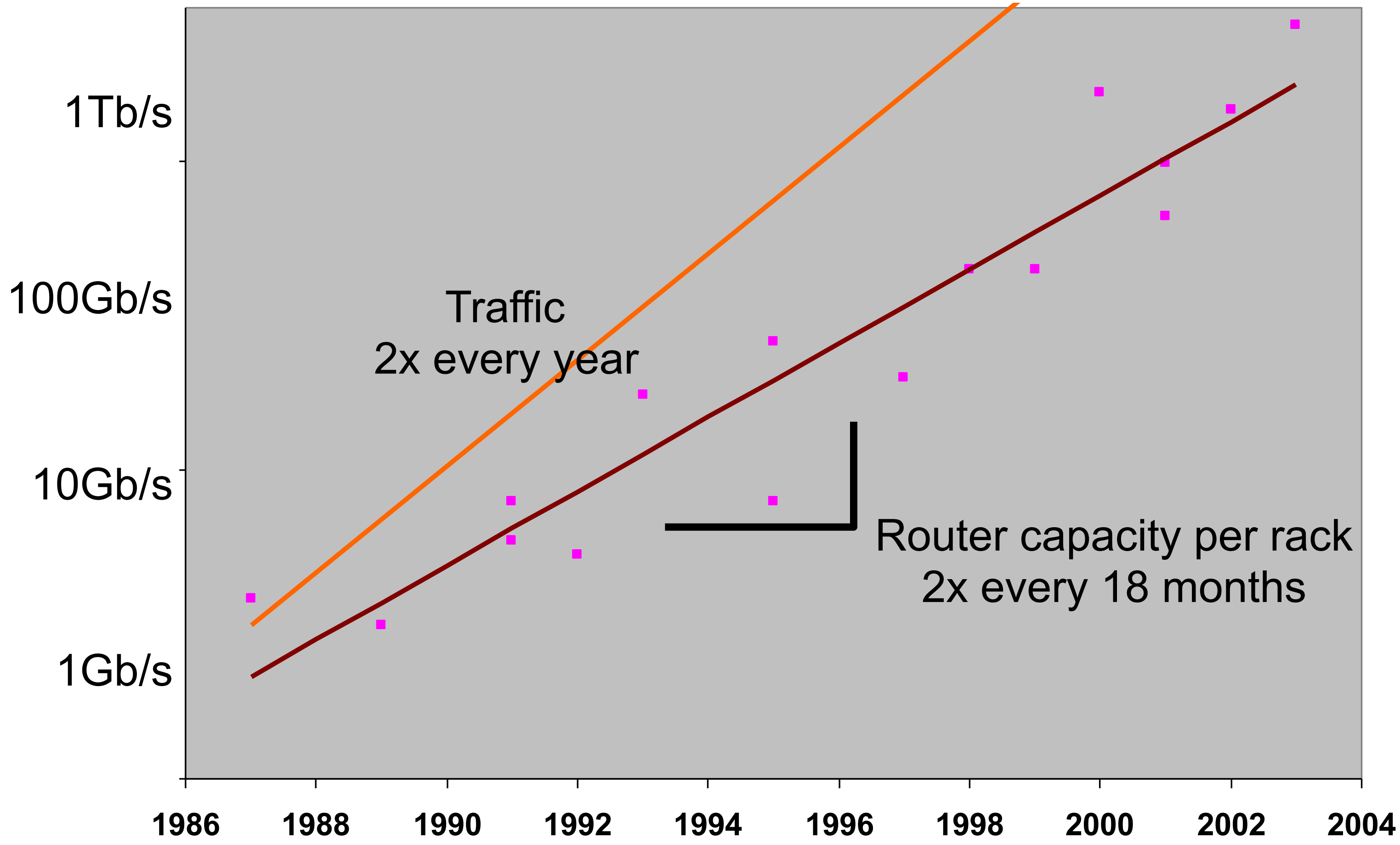
Why do we Need Faster Routers?

1. To prevent routers becoming the bottleneck in the Internet.
2. To increase POP capacity, and to reduce cost, size and power.

Earlier trends



Earlier Backbone router capacity



Current Prediction:
Cisco’s own Visual Networking Index (VNI) predicts
Global IP traffic to grow three-fold (300%) from 2012 to 2017
=> Still a respectable 25% growth per year

Router Performance Growth

Growth in capacity of commercial routers (1 full-rack space):

- Capacity 1992 ~ 2Gb/s
- Capacity 1995 ~ 10Gb/s
- Capacity 1998 ~ 40Gb/s
- Capacity 2001 ~ 160Gb/s
- Capacity 2003 ~ 640Gb/s

…..

- Capacity 2013 ~ 4 to 6.4Tb/s

Average growth rate:

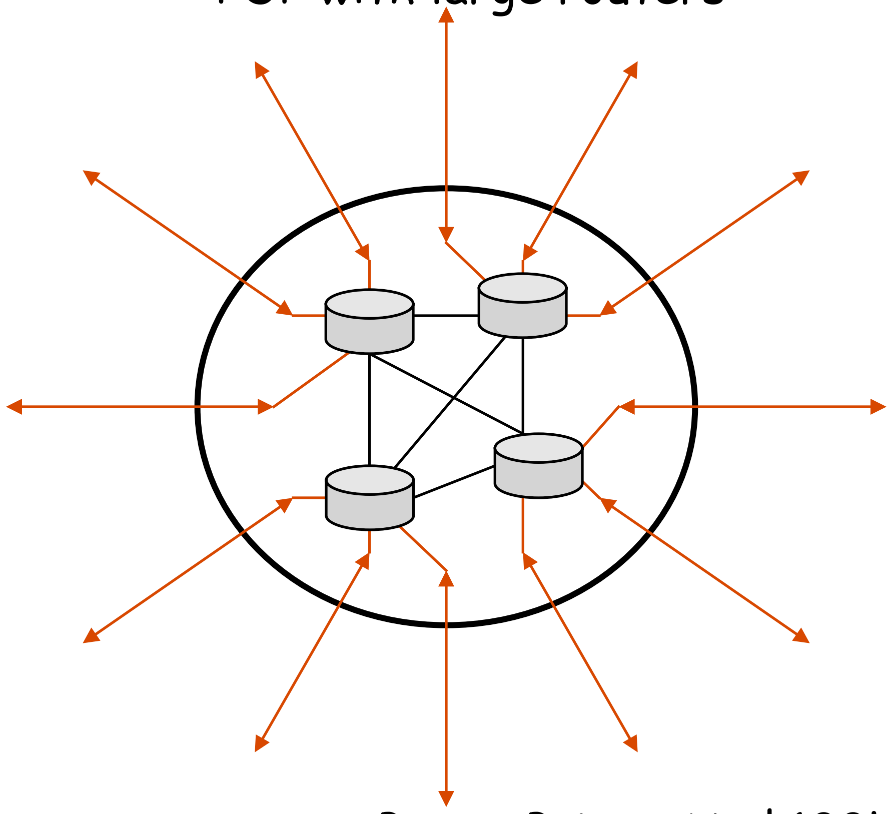
~ 2x / 18 months from 90's to 2002 ;

"only" 10x in the past 1.5 decade (from 2003 – 2013).

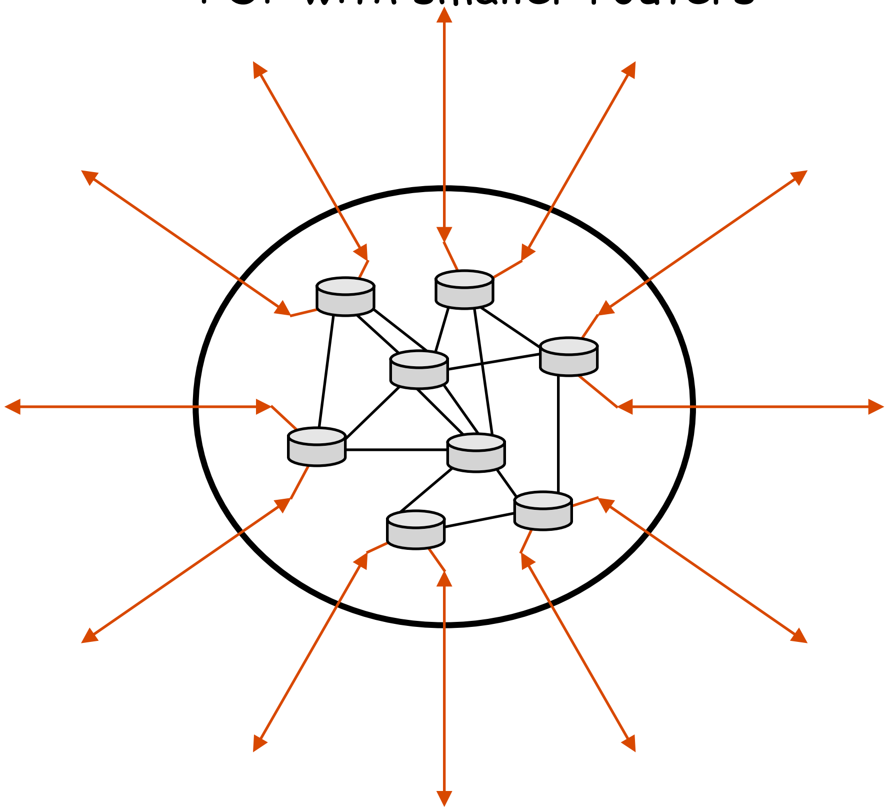
Why we Need Faster Routers

2: To reduce cost, power & complexity of POPs

POP with large routers



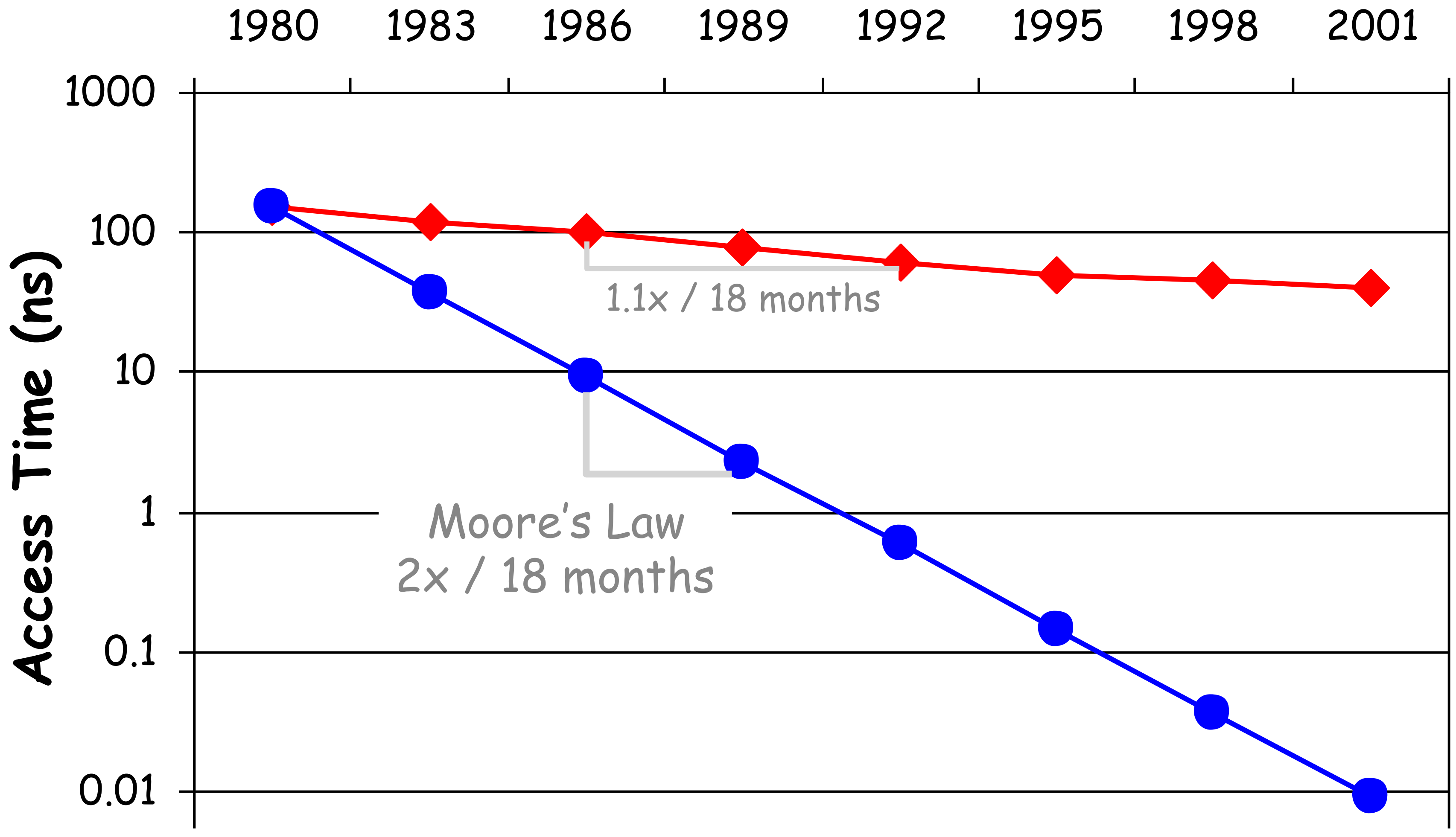
POP with smaller routers



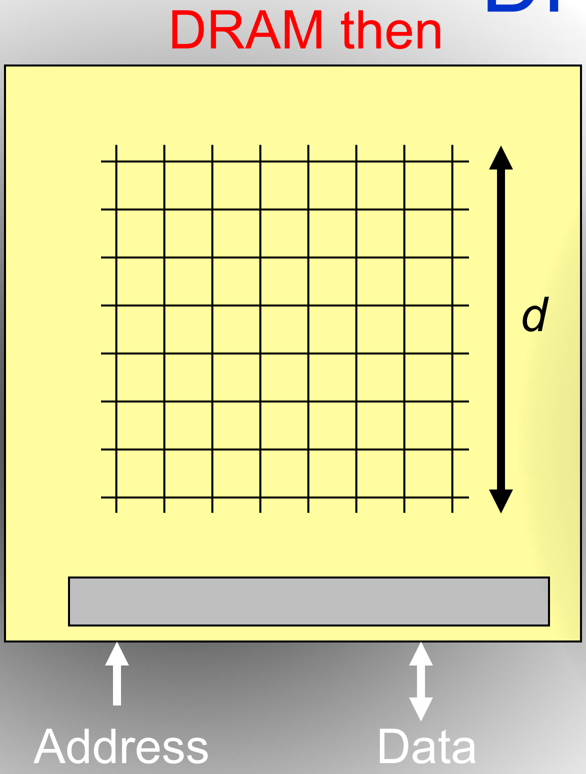
- Ports: Price > US$100k, Power > 400W.
- It is common for 50-60% of ports to be for interconnection.

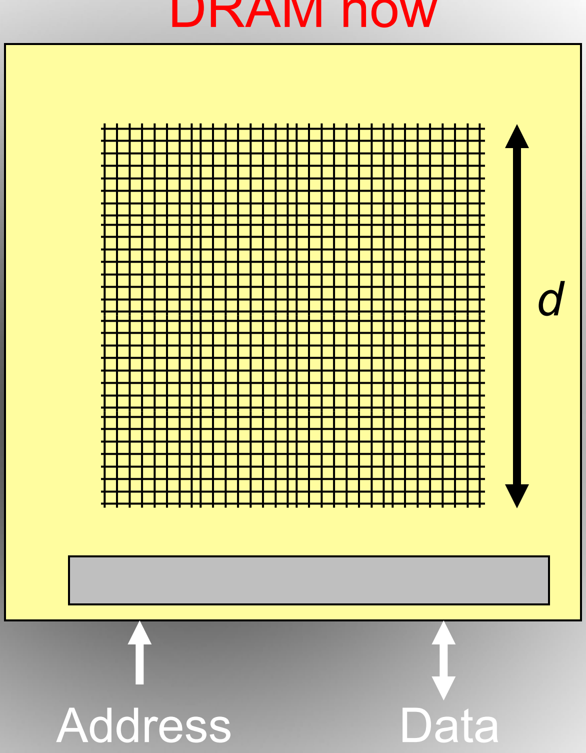
Why are Fast Routers Difficult to Make?

Speed of Commercial DRAM



DRAM as bottleneck

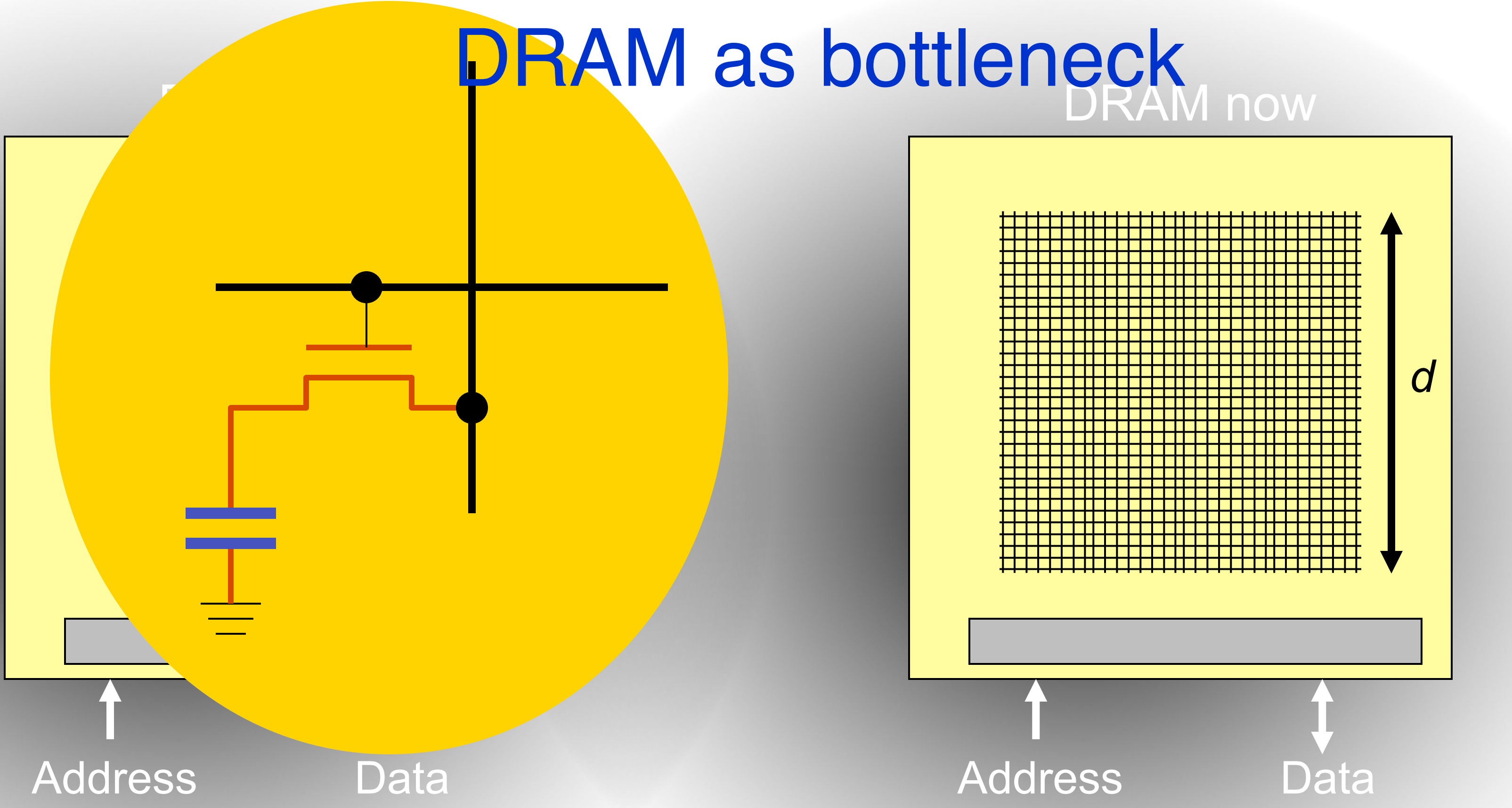




- DRAMs designed to maximize number of bytes
- *d* has stayed constant (yield)
- *d* determines access time (capacitance)

⇨ Access time (“speed”) has stayed constant

DRAM as bottleneck



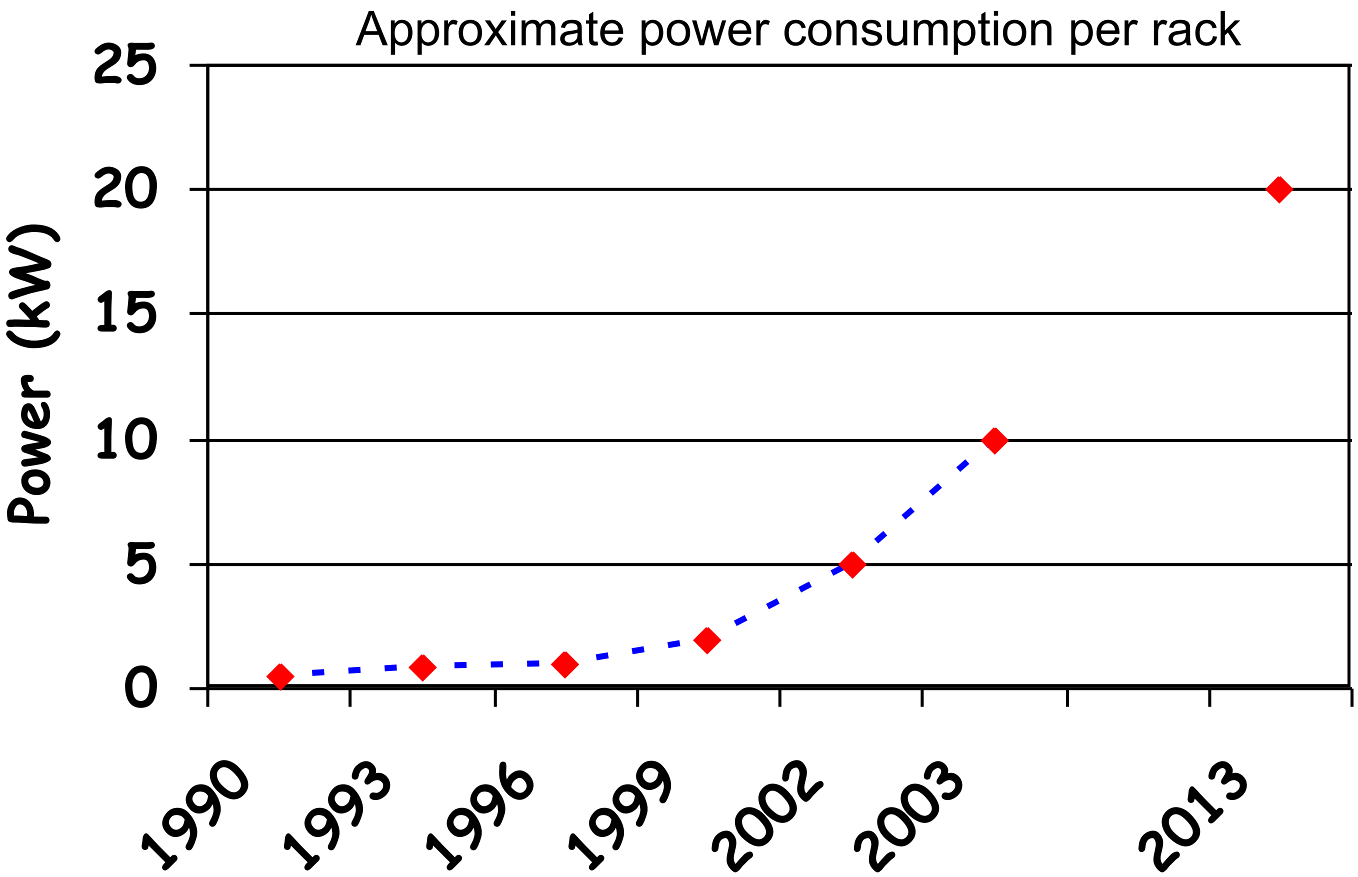
- ❖ DRAMs designed to maximize number of bytes
- ❖ *d* has stayed constant (yield)
- ❖ *d* determines access time (capacitance)

⇨ Access time has stayed constant

Why are Fast Routers Difficult to Make?

- It’s hard to keep up with Moore’s Law:
    - The bottleneck is memory speed.
  - Memory speed is not keeping up with Moore’s Law.
- Moore’s Law is too slow:
  - Routers need to improve *faster* than Moore’s Law in order to keep up with Traffic growth demand.
    - Packet buffers need to operate above 100Gb/s
- Extra processing on the datapath
- Switch architecture with throughput guarantees

What limits router capacity?

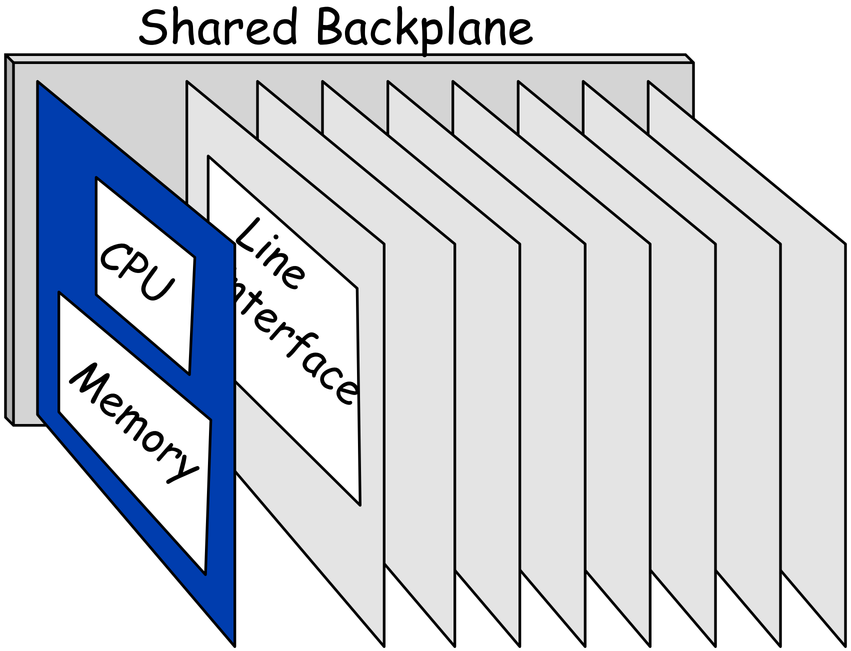


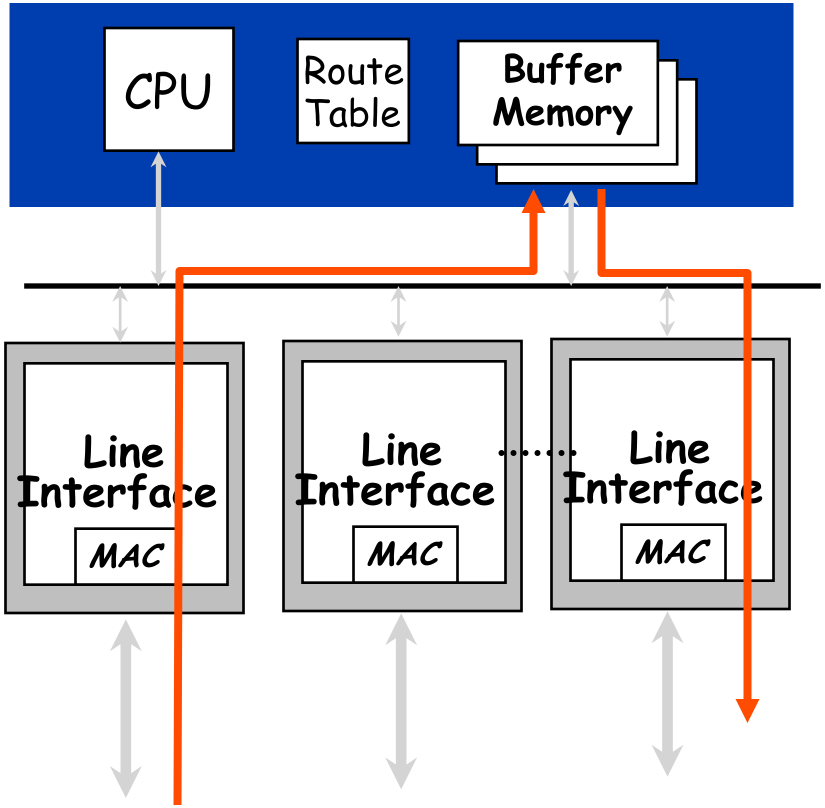
Power density is still a critical limiting factor

Outline

- Background
  - What is a router?
  - Why do we need faster routers?
  - Why are they hard to build?
- Architectures and techniques
  - The evolution of router architecture.
  - Packet Classification
  - IP address lookup
  - Packet buffering.
  - Switching.
  - Example Switching Architectures in practice and in theory
  - Future Directions

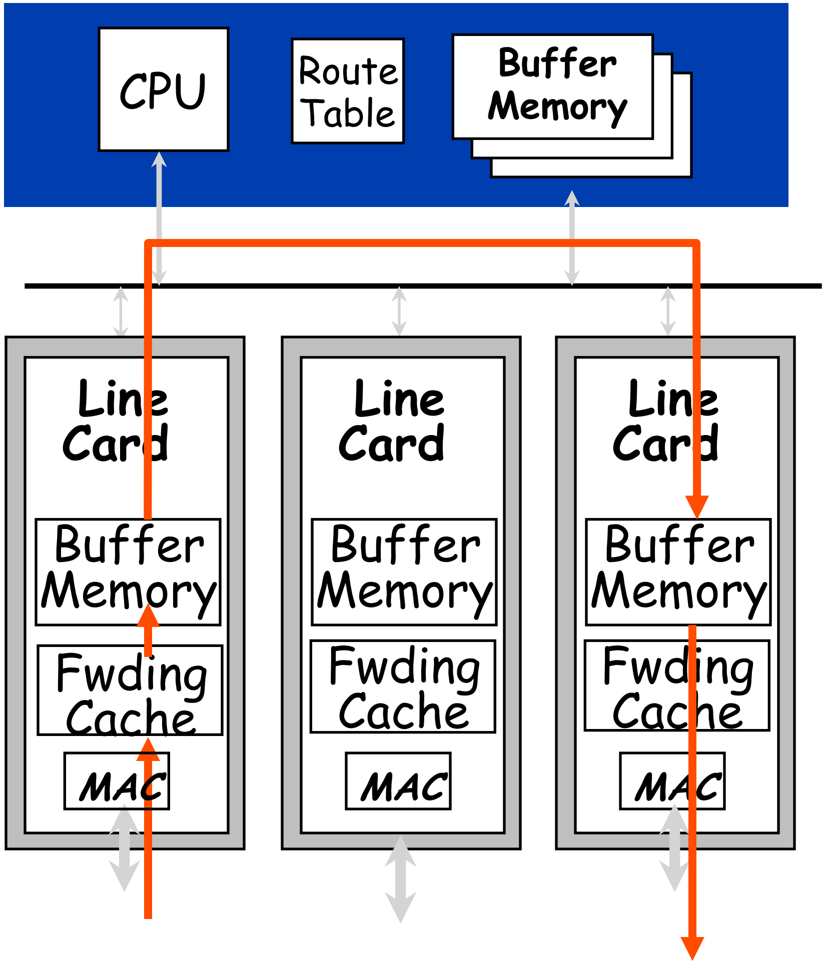
First Generation Routers





Typically <0.5Gb/s aggregate capacity

Second Generation Routers

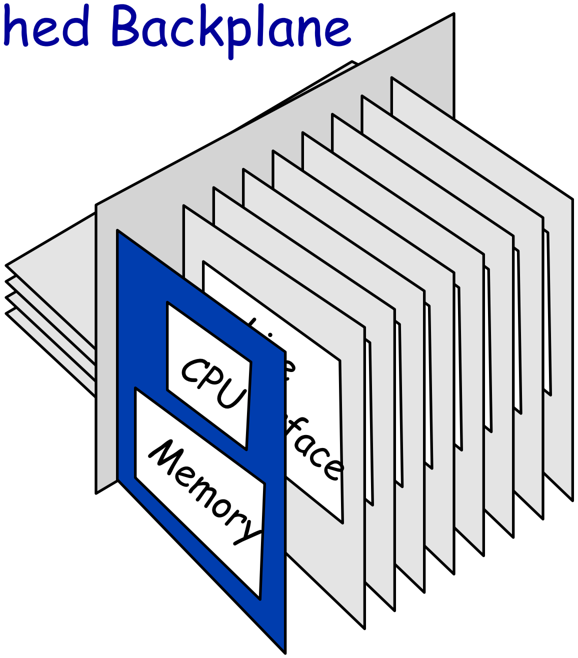


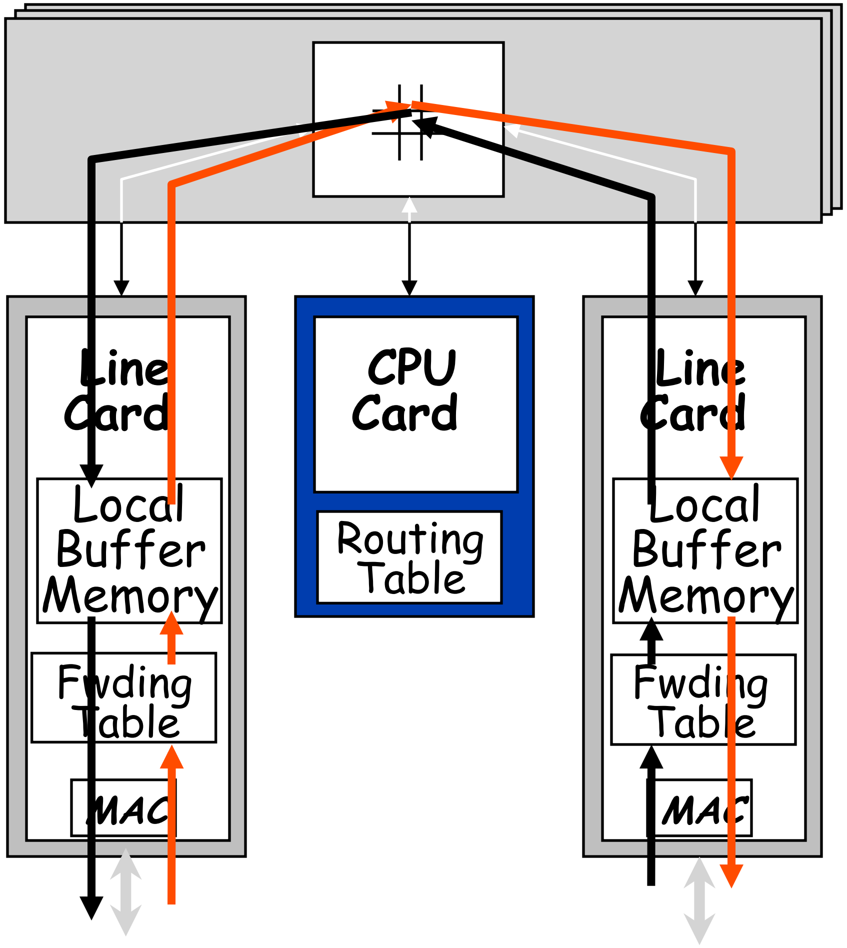
Direct Memory Access (DMA) between the pair of Ingress and Egress Line Cards ;
but still bottlenecked by the Single Shared BUS

Typically <5Gb/s aggregate capacity

Third Generation Routers

Switched Backplane



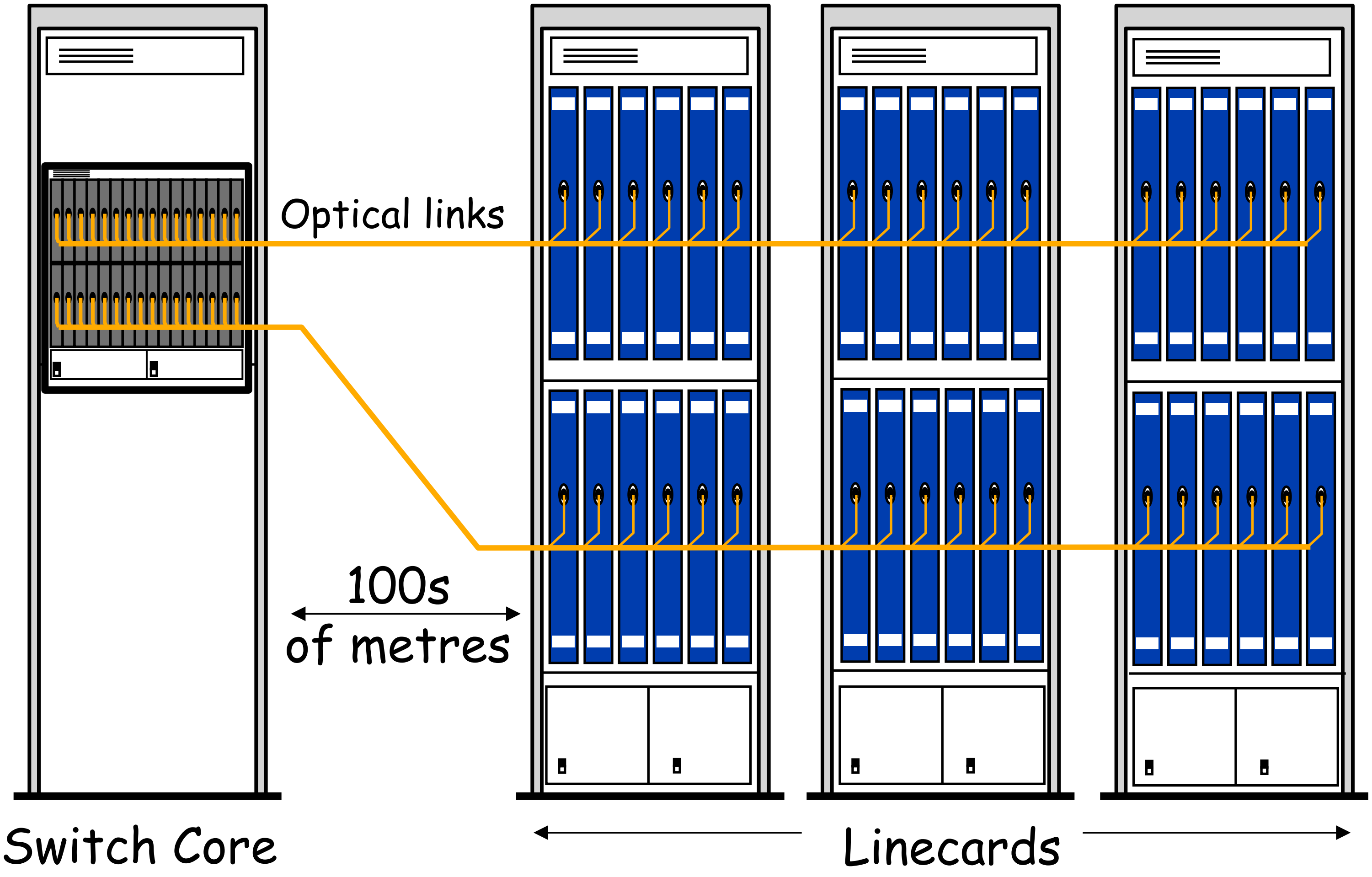


Built-in Switch Fabric inside a Router to provide PARALLEL packet transfer between different pairs of Ingress/Egress Line Cards

Typically <50Gb/s aggregate capacity

Fourth (Current) Generation (Multi-stage) Routers/ Switches

Optics inside a router for the first time

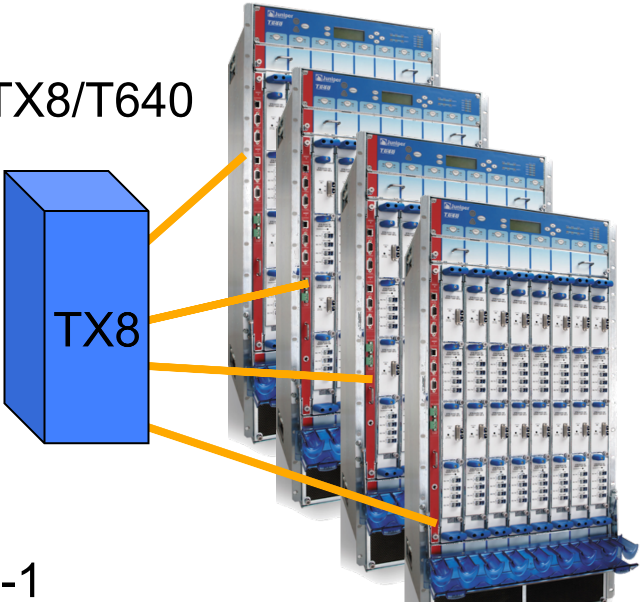


0.3 - 100+Tb/s routers in the market

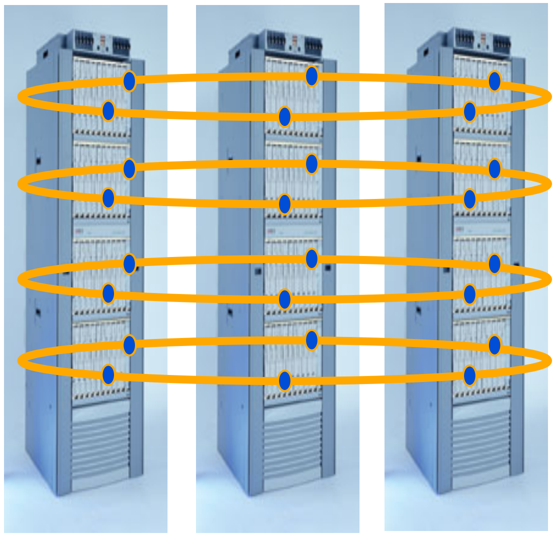
Some (early) 4th Generation Commercial Routers

Alcatel 7670 RSP

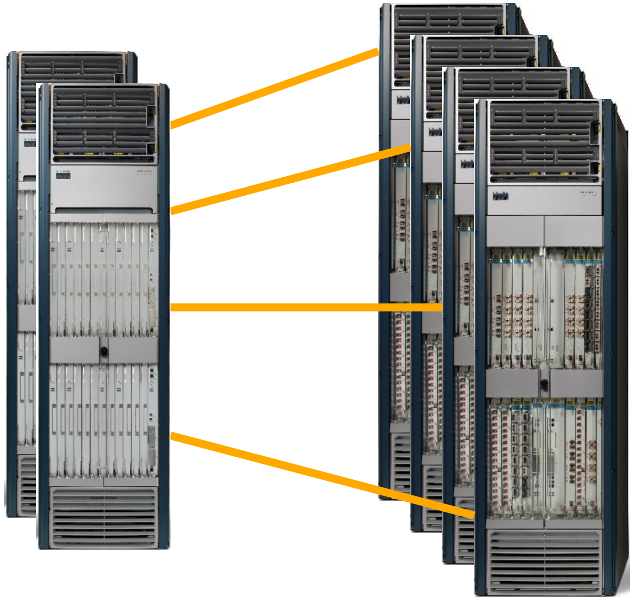
Juniper TX8/T640



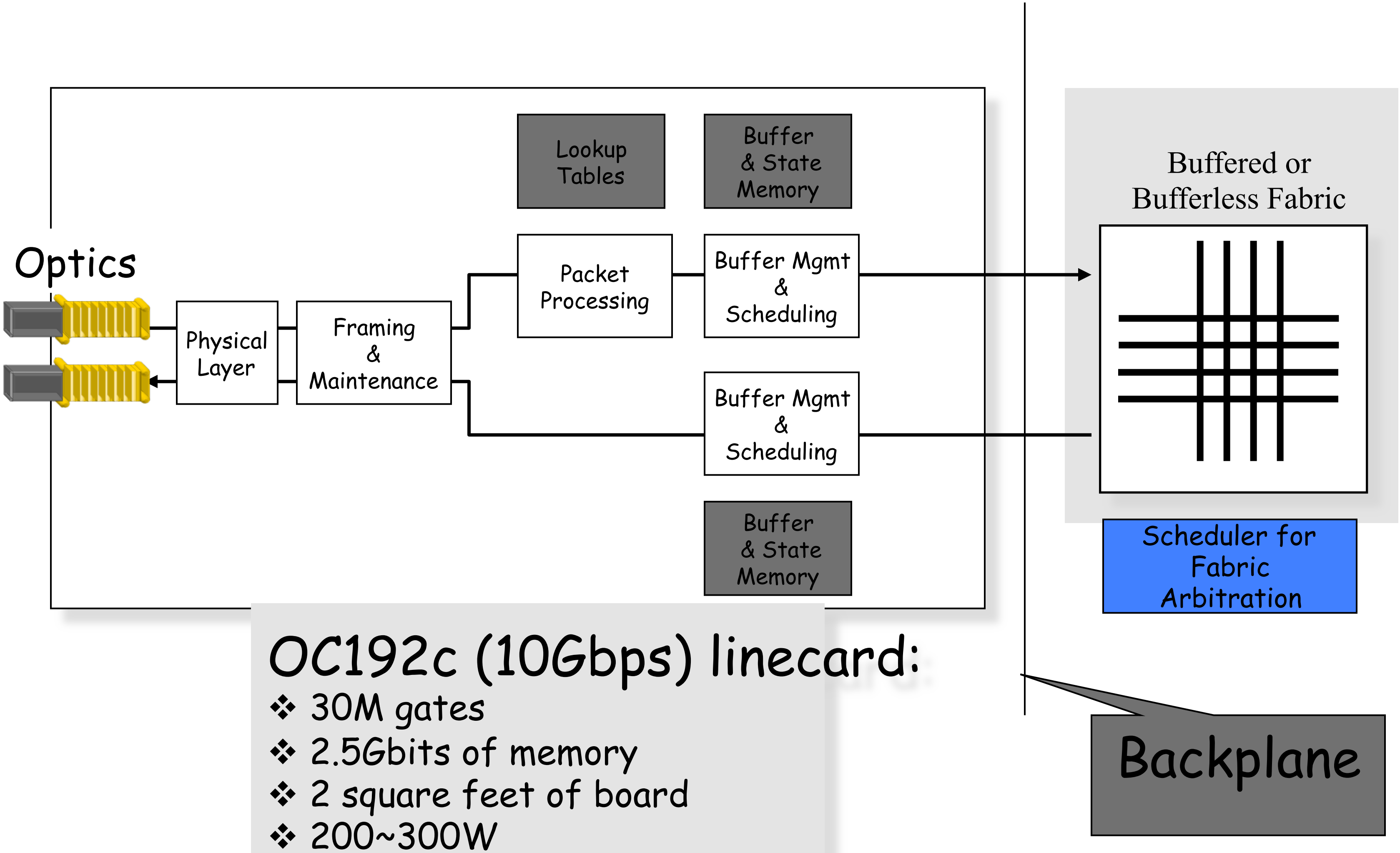
Avici TSR



Cisco CRS-1



A Typical IP Router Linecard



OC192c (10Gbps) linecard:

- 30M gates
- 2.5Gbits of memory
- 2 square feet of board
- 200~300W
- US$20k cost, US$100K price

40-55% of power in chip-to-chip serial links

Outline

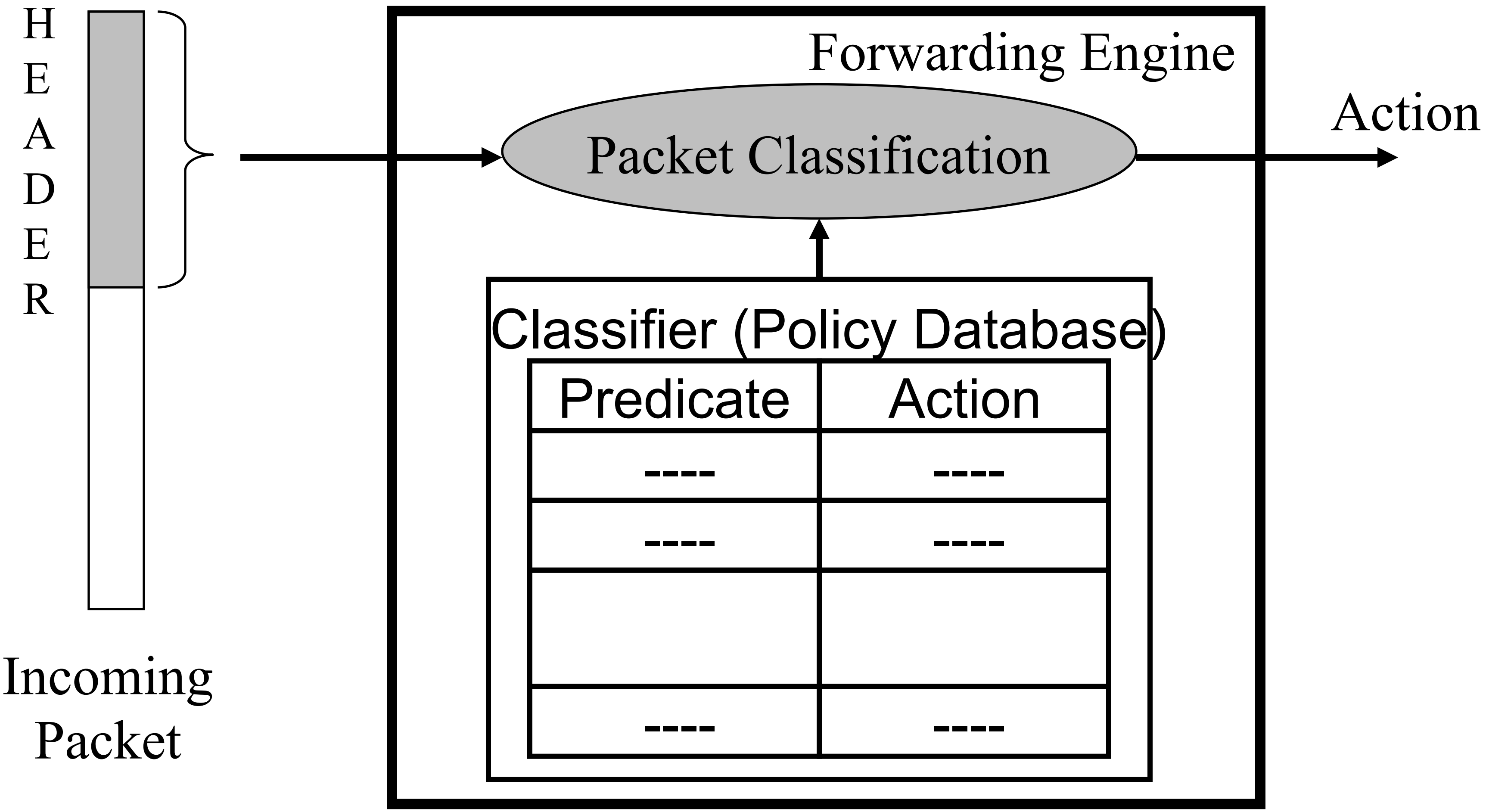
Background

- What is a router?
- Why do we need faster routers?
- Why are they hard to build?

Architectures and techniques

- The evolution of router architecture.
- Packet Classification
- IP address lookup
- Packet buffering.
- Switching.
- Example Switching Architectures in practice and in theory
- Future Directions

Packet Classification



HEADER

Incoming Packet

Forwarding Engine

Packet Classification

Action

Classifier (Policy Database)

Predicate	Action
----	----
----	----
----	----

Yesterday' s Packet Classification Systems

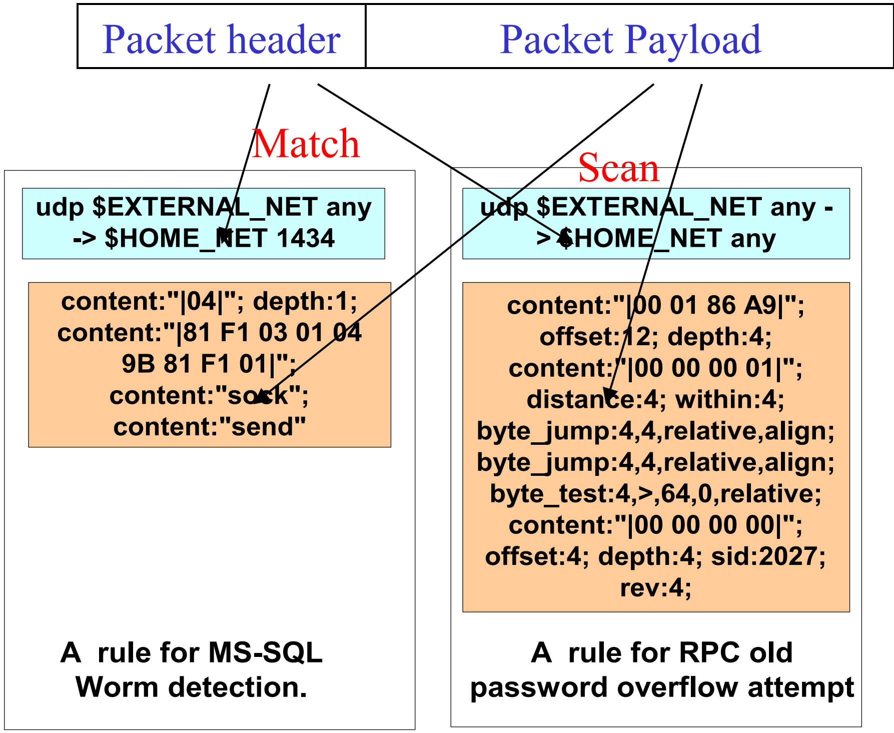
- A classifier consists of *N* rules, each with *F* fields
  - e.g. Next hop routing using destination IP (*F=1*)
  - e.g. Filters for typical L3/L4 firewall *(F=5)*

Source IP	**Destination IP**	**Source Port**	**Destination Port**	**Protocol**	Action	Priority
128.59.67.100	128.*	*	15	TCP	drop	2
128.*	128.2.3.1	*	25	TCP	allow	1

- **Single-Match Classification:**
  - Assumption: all the rules are associated with priorities
  - Only the highest priority match matters
  - E.g., longest prefix match

New Applications

- Intrusion Detection Systems (e.g., SNORT)
  - Rule header: a 5-field classification rule for the ***packet header***
  - Rule options: specify intrusion patterns for the ***entire packet*** scanning.



```
udp $EXTERNAL_NET any
-> $HOME_NET 1434
```

```
content:"|04|"; depth:1;
content:"|81 F1 03 01 04
9B 81 F1 01|";
content:"sock";
content:"send"
```

A rule for MS-SQL Worm detection.

```
udp $EXTERNAL_NET any -
> $HOME_NET any
```

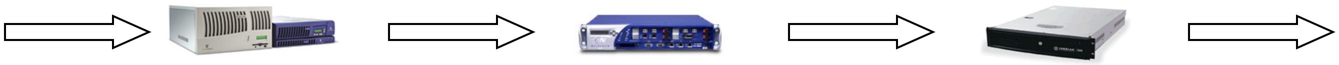
```
content:"|00 01 86 A9|";
offset:12; depth:4;
content:"|00 00 00 01|";
distance:4; within:4;
byte_jump:4,4,relative,align;
byte_jump:4,4,relative,align;
byte_test:4,>,64,0,relative;
content:"|00 00 00 00|";
offset:4; depth:4; sid:2027;
rev:4;
```

A rule for RPC old password overflow attempt

- **Multi-Match Classification:** Identify all the matching rule headers
  - No priority among filters
  - Identify all the related rules
  - Also required by accounting applications

New Applications (cont.)

- In some edge networks



  - Each box introduces extra delay
  - Common functions like classification are repeatedly applied
  - Highly inefficient!
- Programmable Network Element
  - Support multiple functions in one device
  - Each packet may related to different set of functions
    - E.g., HTTP packets related to firewall and HTTP load balancer
    - E.g., VPN packets related to encryption / decryption
  - **Multi- Match Classification:** identify the all the relevant functions

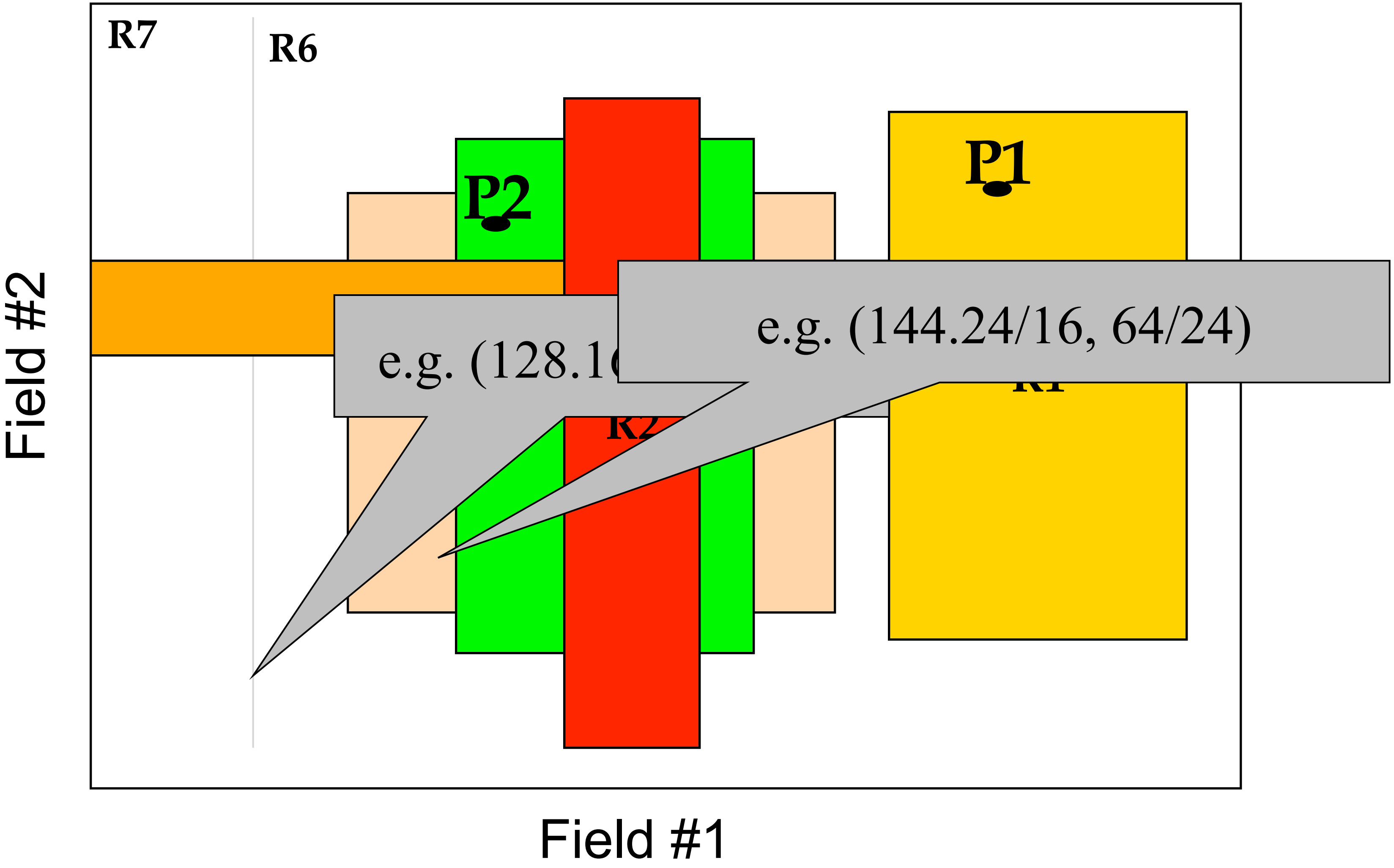
Multi-field Packet Classification with Range support: Single-Match

	Field 1	*Field 2*	…	*Field k*	*Action*
Rule 1	152.163.190.69/21	152.163.80.11/32	…	UDP	A1
Rule 2	152.168.3.0/24	152.163.0.0/16	…	TCP	A2
…	…	…	…	…	…
Rule N	152.168.0.0/16	152.0.0.0/8	…	ANY	An

Given a classifier with N rules, find the action associated with the highest priority rule matching an incoming packet.

Geometric Interpretation of Range-support in 2D

Field #1	Field #2	Data

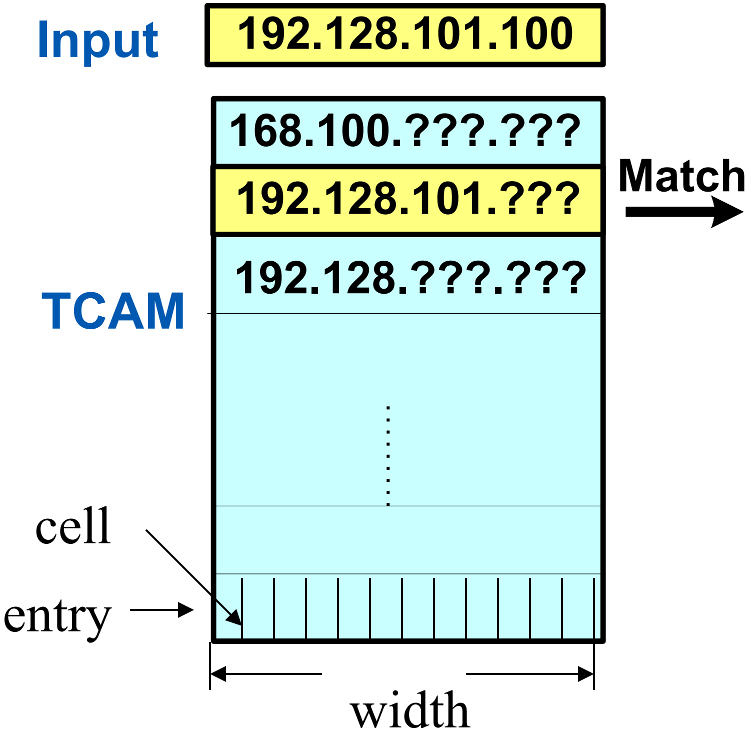


e.g. (128.16

e.g. (144.24/16, 64/24)

Single-Match with Ternary-CAMs (TCAM)

- Fully associative memory compare input string with all the entries in parallel
  - If multiple matches, report the index of the **first** match
- Each cell takes one of three logic states
  - ‘0’, ‘1’, and ‘?’(don’t care)
- Current commercial TCAM technology
  - Fast Match Time: < **3 nsec**
  - Size: as large as 2.5MBytes (as of 2012)
  - Width configurable, e.g. a 1MB T-CAM
    - 1024 entries *1024 bytes width OR
    - 2048 entries *512 bytes width
  - priced at $200-$300
  - Can be used to realize longest-prefix match easily !! (for small forwarding table only)
  - Can be Power-Hungry
  - Not scalable for Large rule sets or very high-speed links.



Input 192.128.101.100

TCAM

168.100.???.???

192.128.101.???

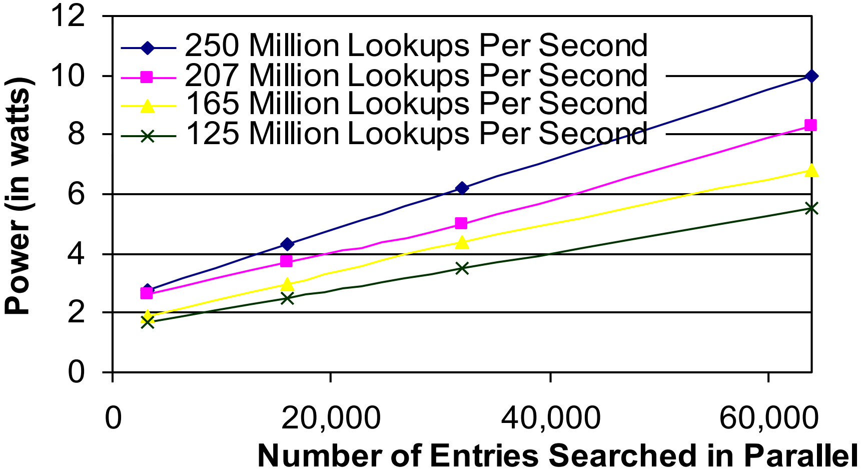
192.128.???.???

Match

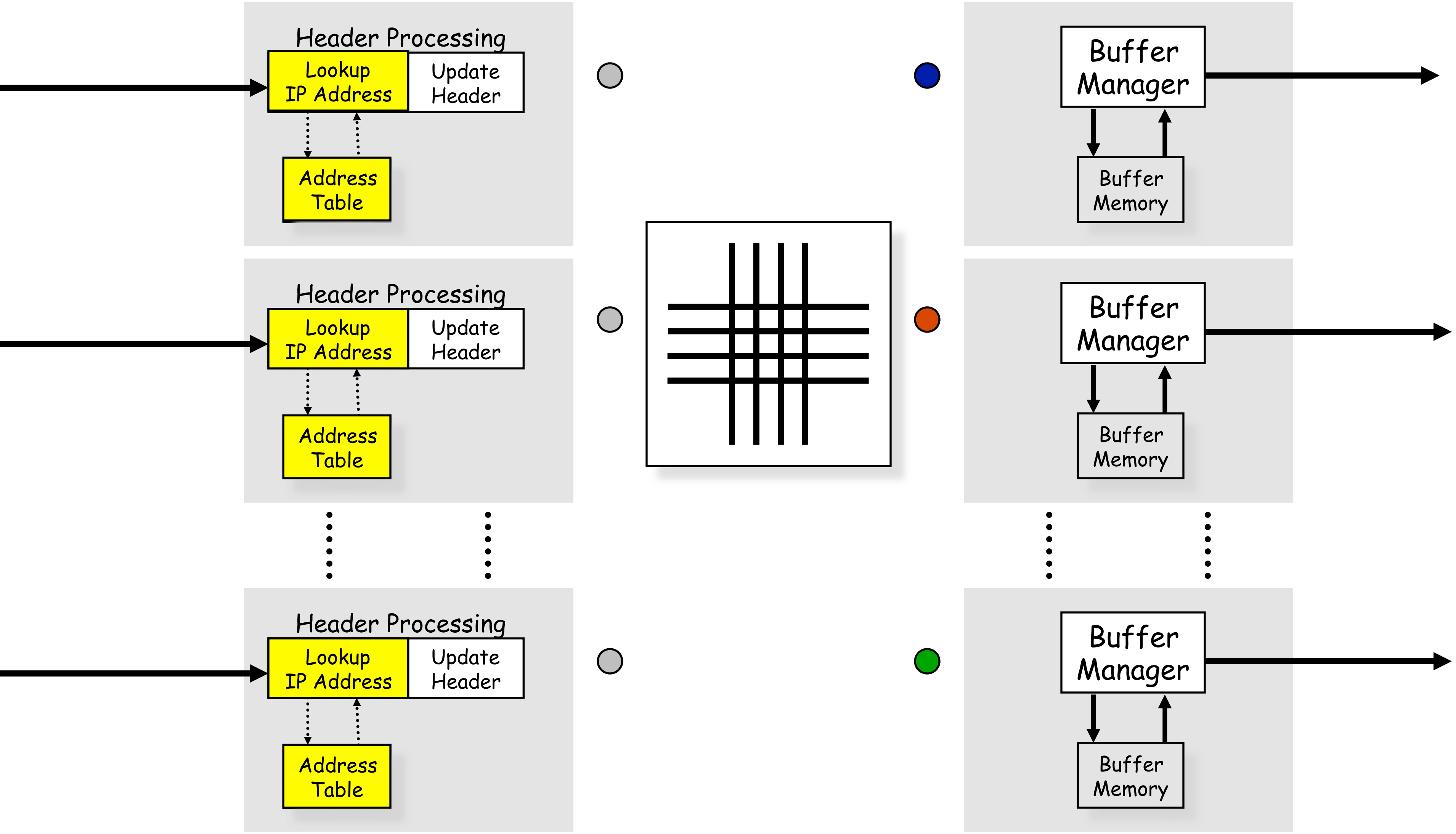
cell

entry

width

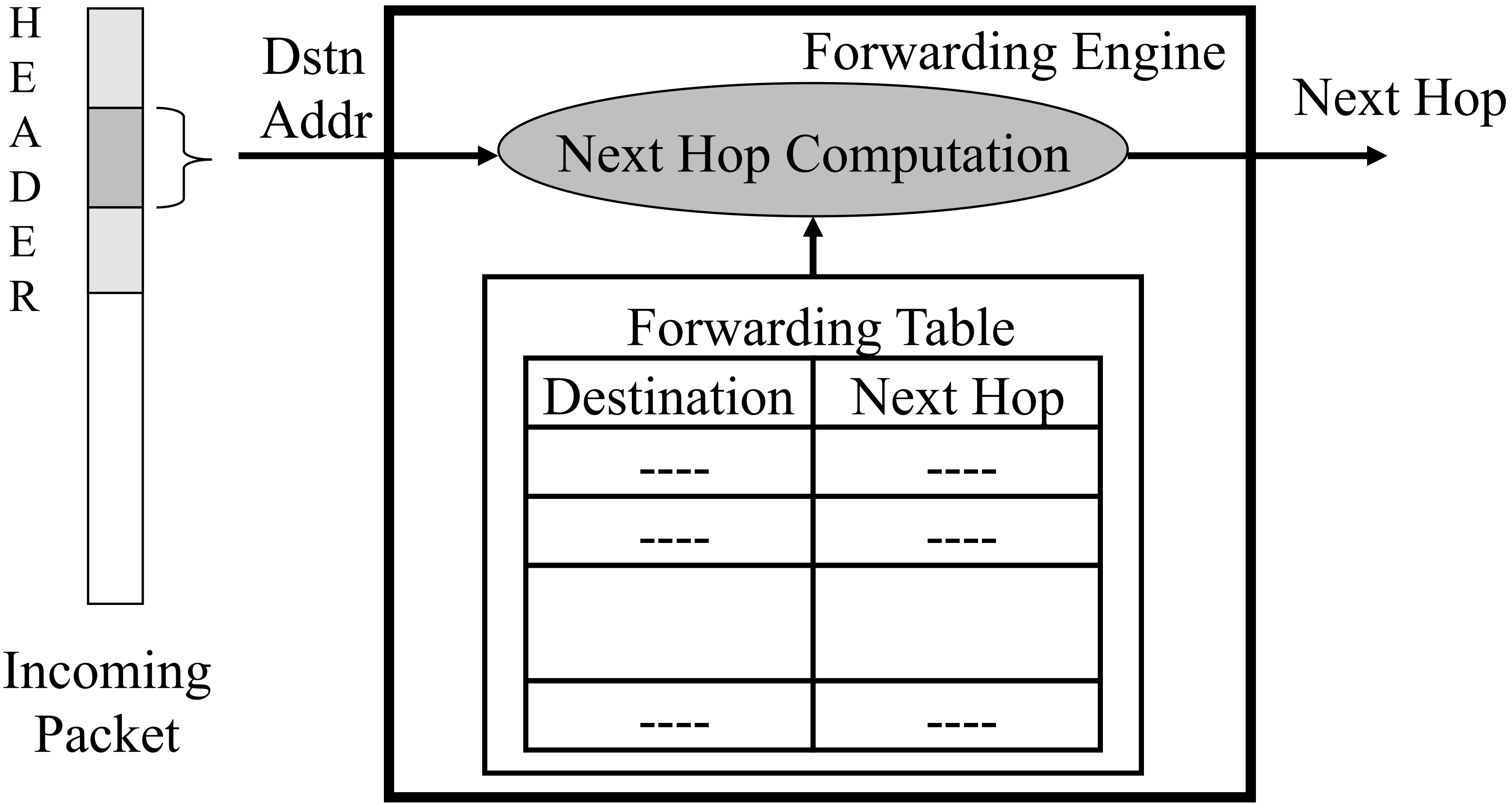


Generic Router Architecture



IP Router

Lookup



Destination	Next Hop
----	----
----	----
----	----

IPv4 unicast destination address based lookup

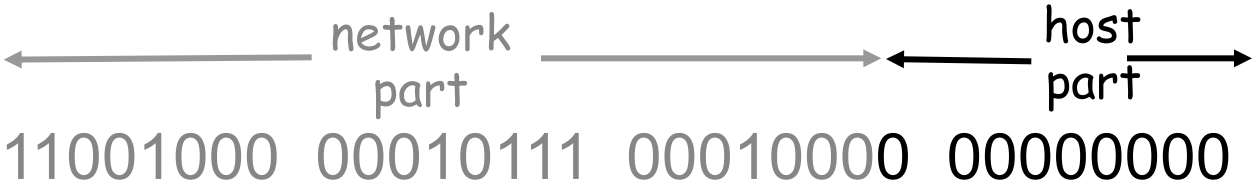
IP Address Lookup

Why it's thought to be hard:

1. It's not an exact match: it's a longest prefix match.
2. The table is large (for Core Routers): about 500,000 entries as of 2013, and growing.
3. The lookup must be fast: about 6.7ns for a 100Gb/s Ethernet (assuming 84byte minimum frame-size).

CIDR: Classless IP addressing

- CIDR: Classless InterDomain Routing
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address



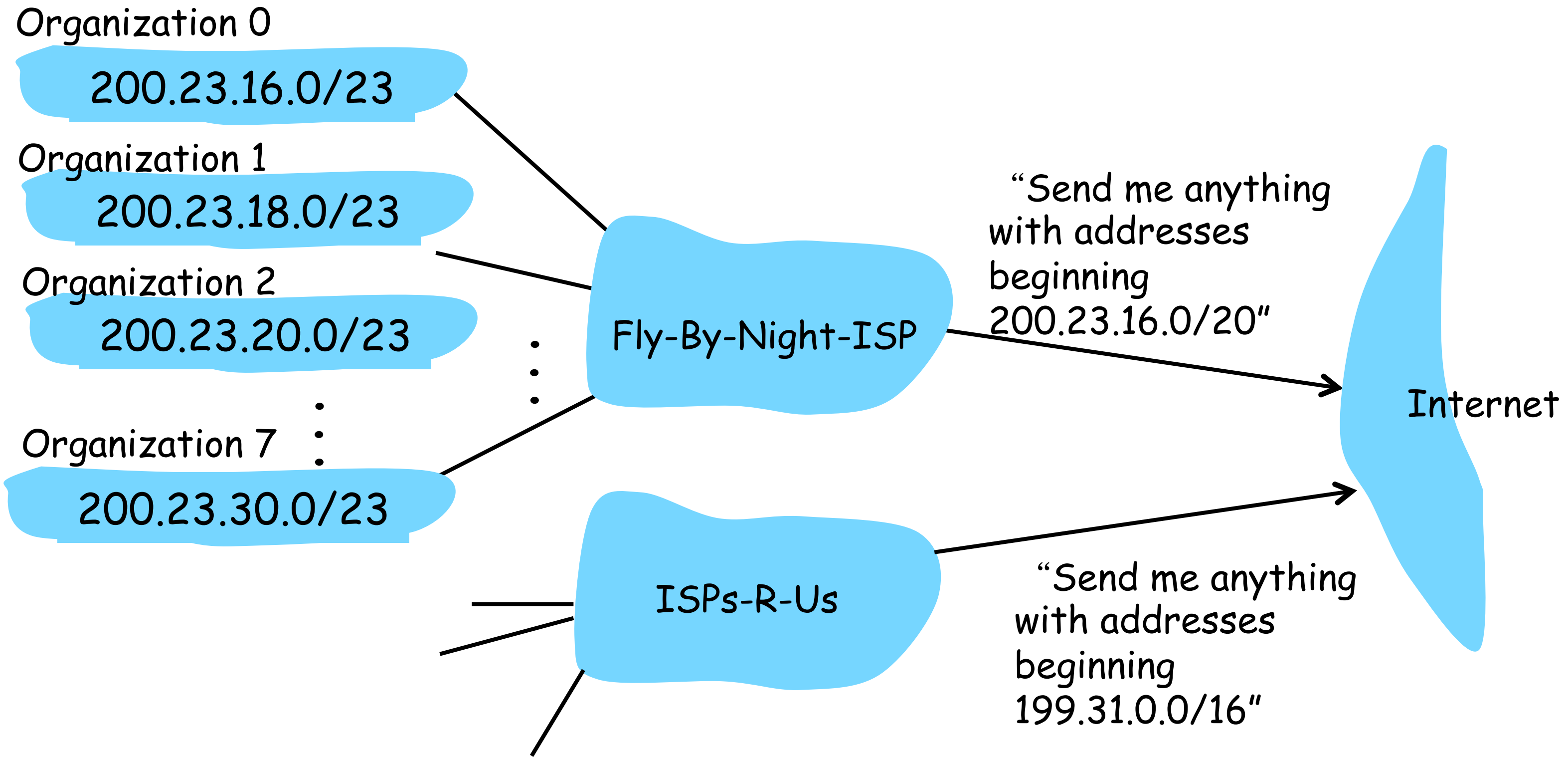
200.23.16.0/23:

A prefix = A contiguous range of IP addresses

= an IP subnetwork

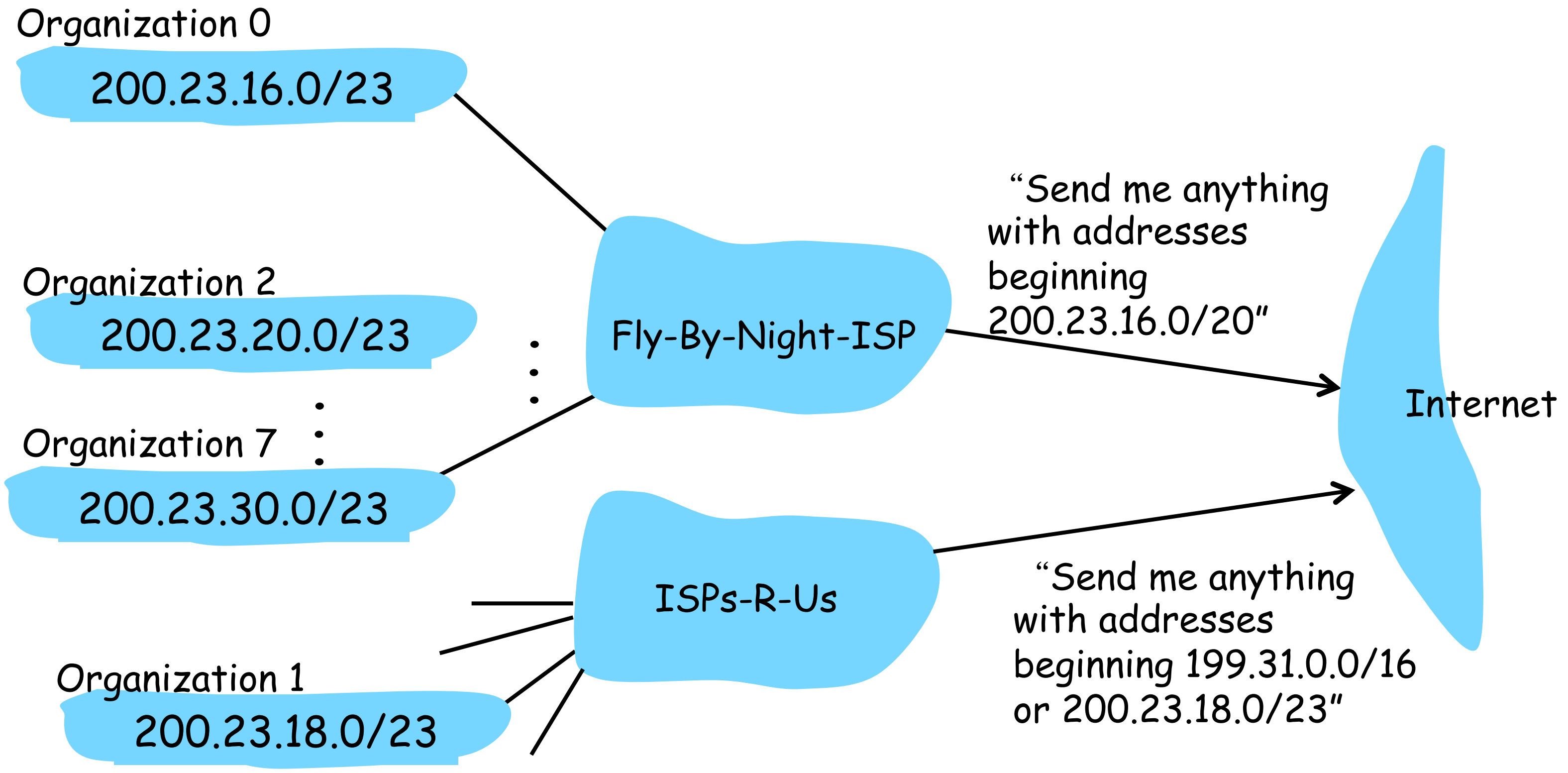
Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

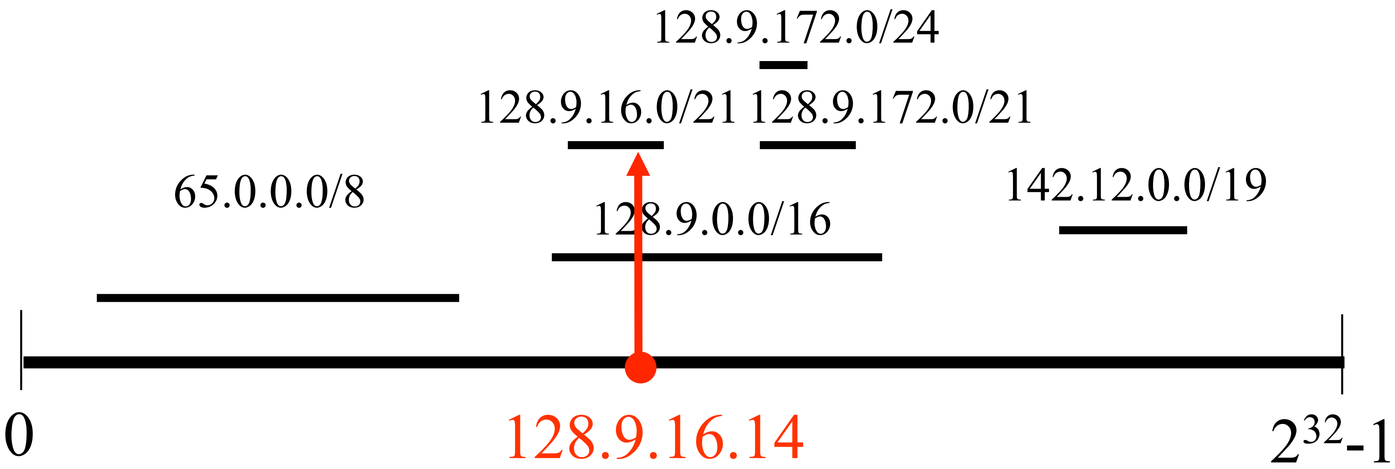
ISPs-R-Us has a more specific route to Organization 1



"Send me anything with addresses beginning 200.23.16.0/20"

"Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23"

IP Lookups find Longest Prefixes



Routing lookup: Find the longest matching prefix (aka the most specific route) among all prefixes that match the destination address.

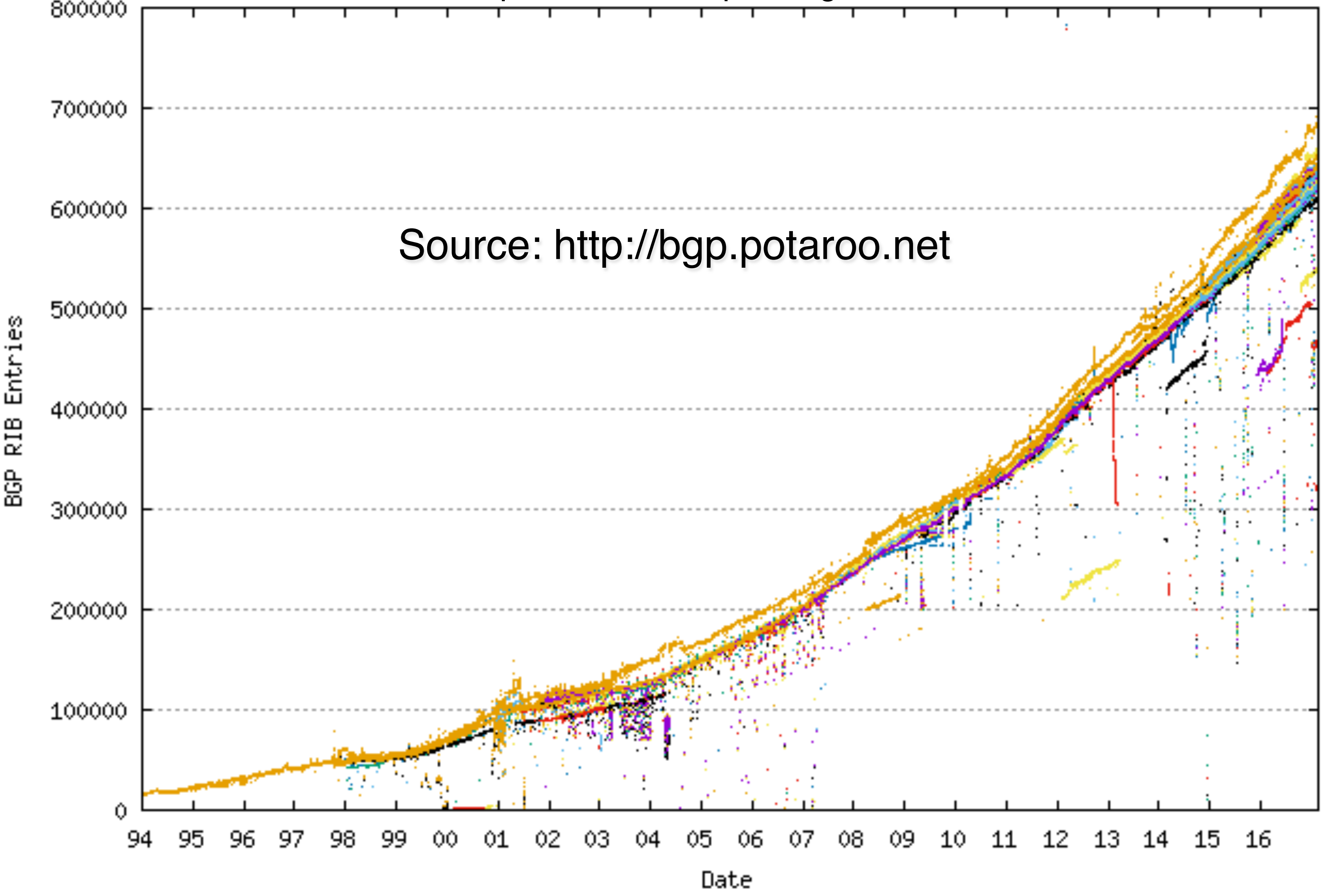
IP Address Lookup

Why it's thought to be hard:

1. It's not an exact match: it's a longest prefix match.
2. The table is large (for Core Routers): about 650,000 entries as of Jan 2017, and growing.
3. The lookup must be fast: about 6.7ns for a 100Gb/s Ethernet (assuming 84byte minimum frame-size).

Address (BGP Routing) Tables are Large (> 650K entries by Jan 2017)

http://www.cidr-report.org/as2.0/



Source: http://bgp.potaroo.net

IP Address Lookup

Why it's thought to be hard:

1. It's not an exact match: it's a longest prefix match.
2. The table is large (for Core Routers): about 500,000 entries as of 2013, and growing.
3. The lookup must be fast: about 6.7ns for a 100Gb/s Ethernet (assuming 84byte minimum frame-size).

Lookup Performance Required

Line	Line Rate	Pkt-size=40Byte	Pkt-size=240Byte
T1	1.5Mbps	4.68 Kpps	0.78 Kpps
OC3	155Mbps	480 Kpps	80 Kpps
OC12	622Mbps	1.94 Mpps	323 Kpps
OC48	2.5Gbps	7.81 Mpps	1.3 Mpps
OC192	10 Gbps	31.25 Mpps	5.21 Mpps

NB: Good Router Performance Requires not only line transmission performance (bps) but ALSO packet processing performance (pps=Packets per Sec)

Lookups Must be Fast

Year	Line	40Byte packets (Mpps)
1997	622Mb/s	1.94
1999	2.5Gb/s	7.81
2001	10Gb/s	31.25
2003	40Gb/s	125 (= 8ns/pkt)

Fortunately, for 100Gbps Ethernet, minimum 100GbE "effective" frame size is:
84 Byte = Preamble (8) + min. Frame length (64) + min.Interframe spacing (12)
> 40Byte => 6.7ns/pkt "only"

IP Routers

Metrics for Lookups

128.9.16.14

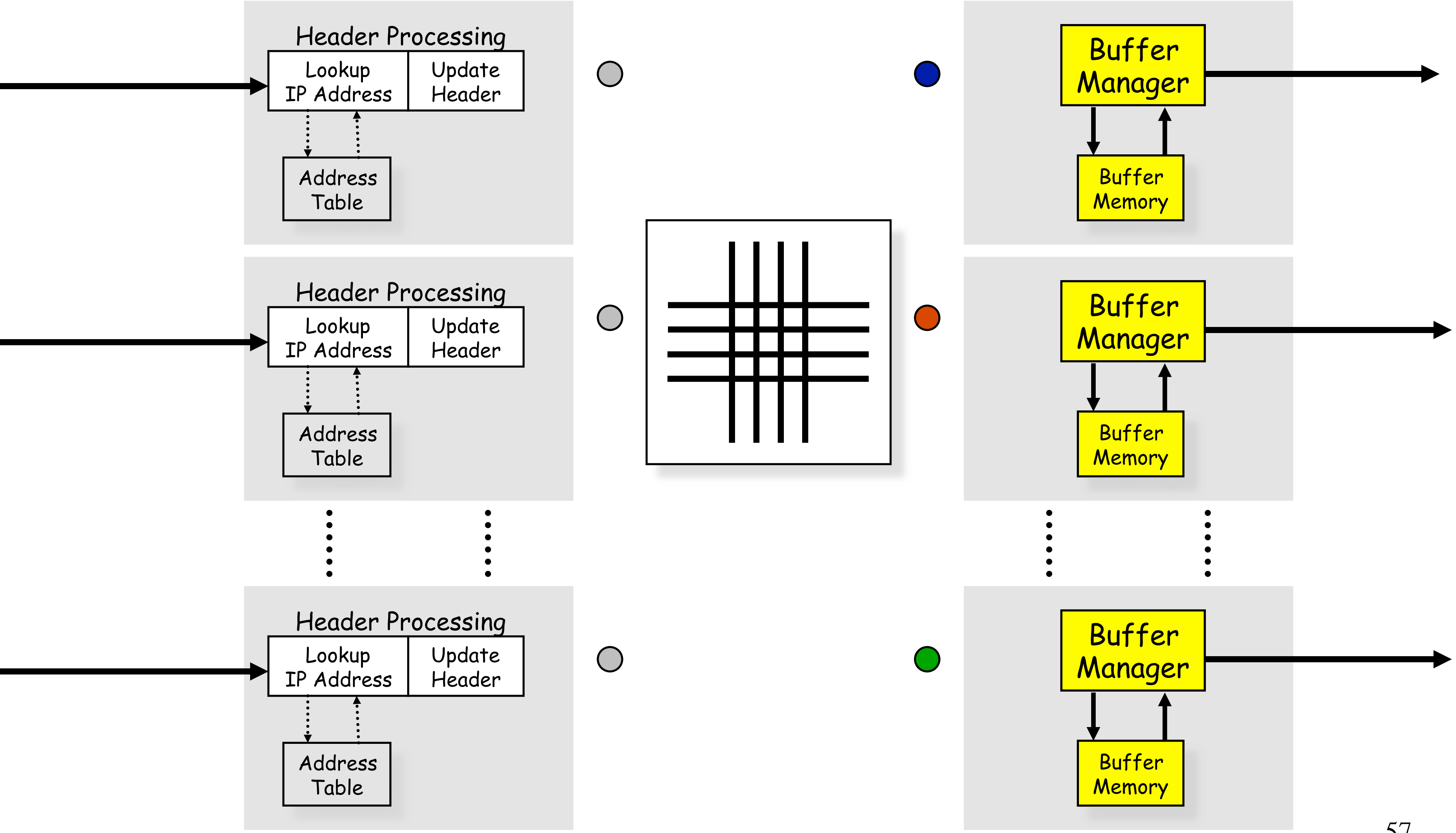
Prefix	Port
65/8	3
128.9/16	5
128.9.16/20	2
128.9.19/24	7
128.9.25/24	10
128.9.176/20	1
142.12/19	3

- Lookup time
- Storage space
- Update time
- Preprocessing time

Outline

- Background
  - What is a router?
  - Why do we need faster routers?
  - Why are they hard to build?
- Architectures and techniques
  - The evolution of router architecture.
  - Packet Classification
  - IP address lookup
  - Packet buffering.
  - Switching.
  - Example Switching Architectures in practice and in theory
  - Future Directions

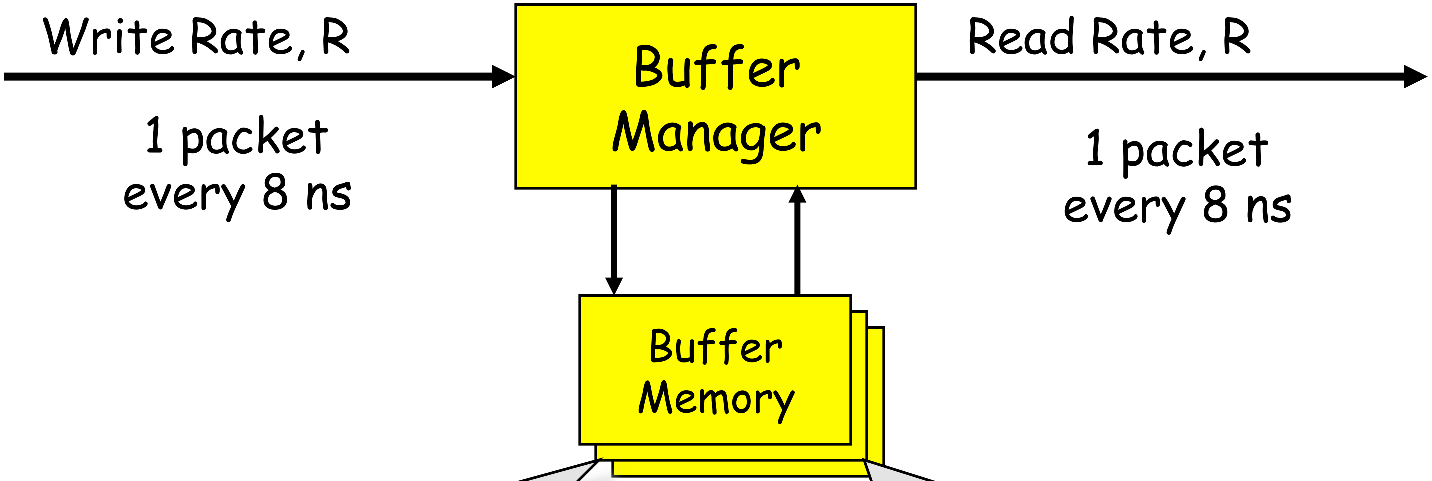
Generic Router Architecture



Fast Packet Buffers

Example: 40Gb/s packet buffer

Size = RTT*BW = 10Gb; 40 byte packets



Write Rate, R

1 packet
every 8 ns

Buffer Manager

Read Rate, R

1 packet
every 8 ns

Buffer Memory

Use SRAM?

+ fast enough random access time, but

- too low density to store 10Gb of data.

Use DRAM?

+ high density means we can store data, but

- too slow (50ns random access time).

Outline

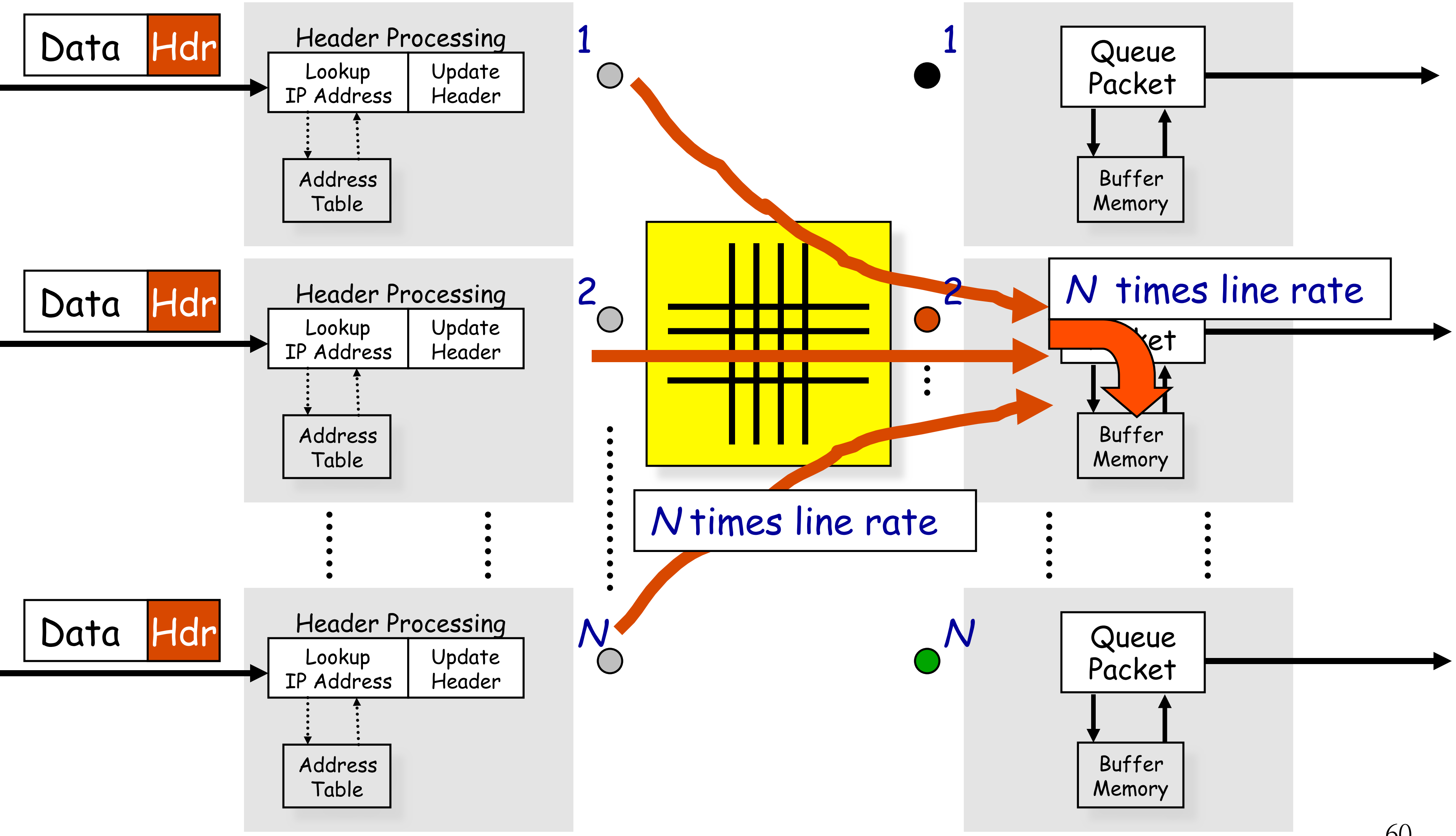
Background

- What is a router?
- Why do we need faster routers?
- Why are they hard to build?

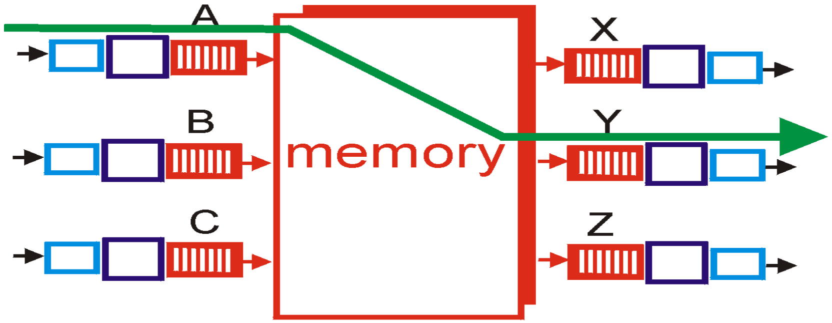
Architectures and techniques

- The evolution of router architecture.
- IP address lookup.
- Packet buffering.
- Switching.

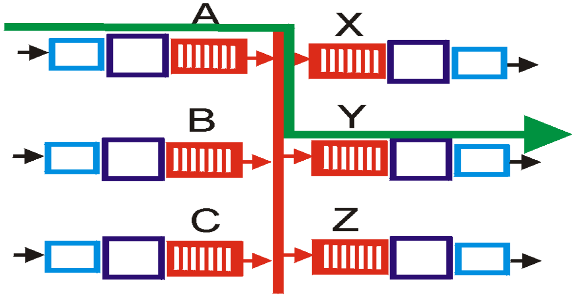
Generic Router Architecture



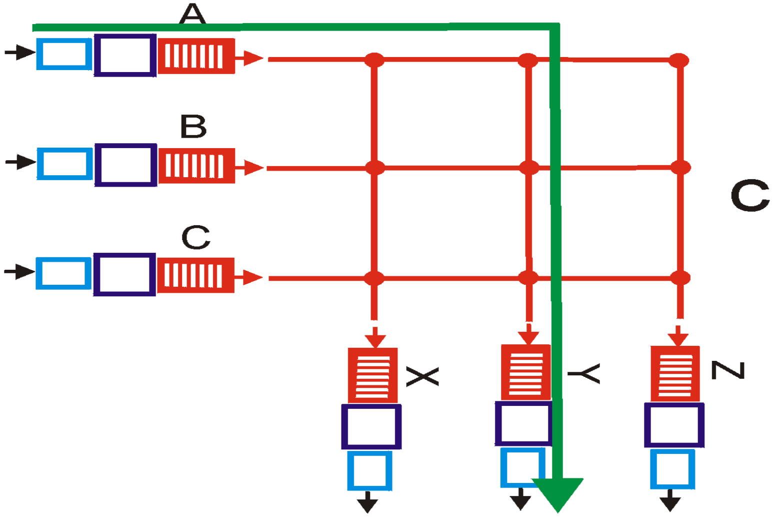
Switching Fabric



memory



bus

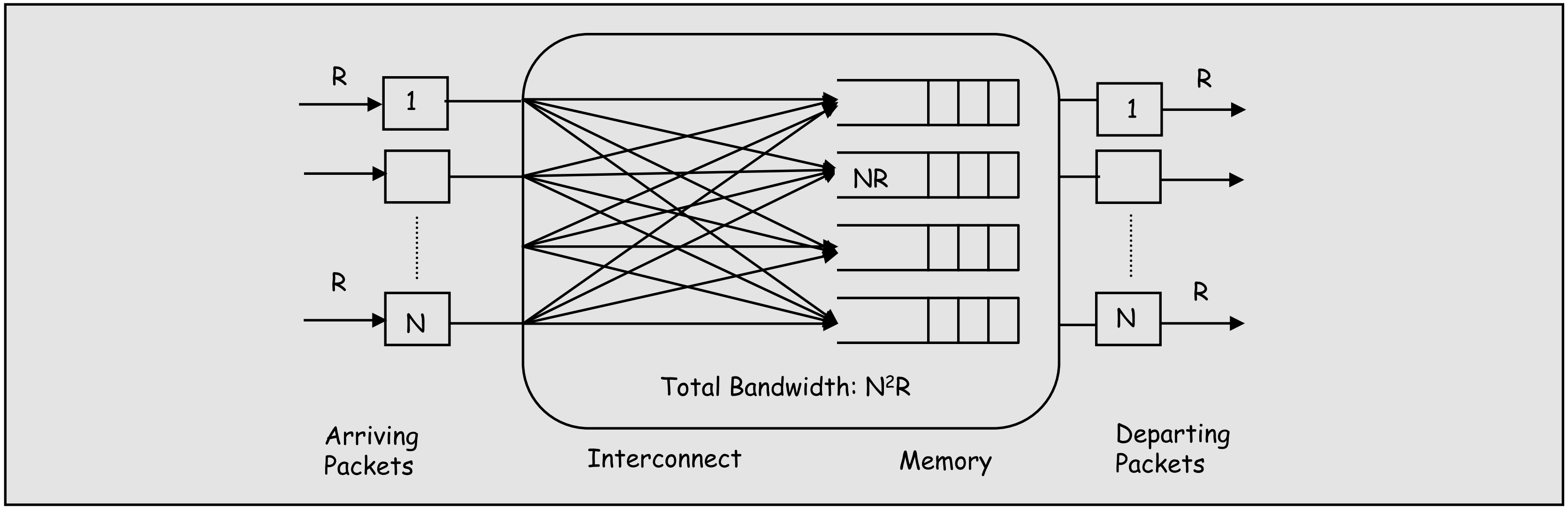


crossbar

Switching Fabrics

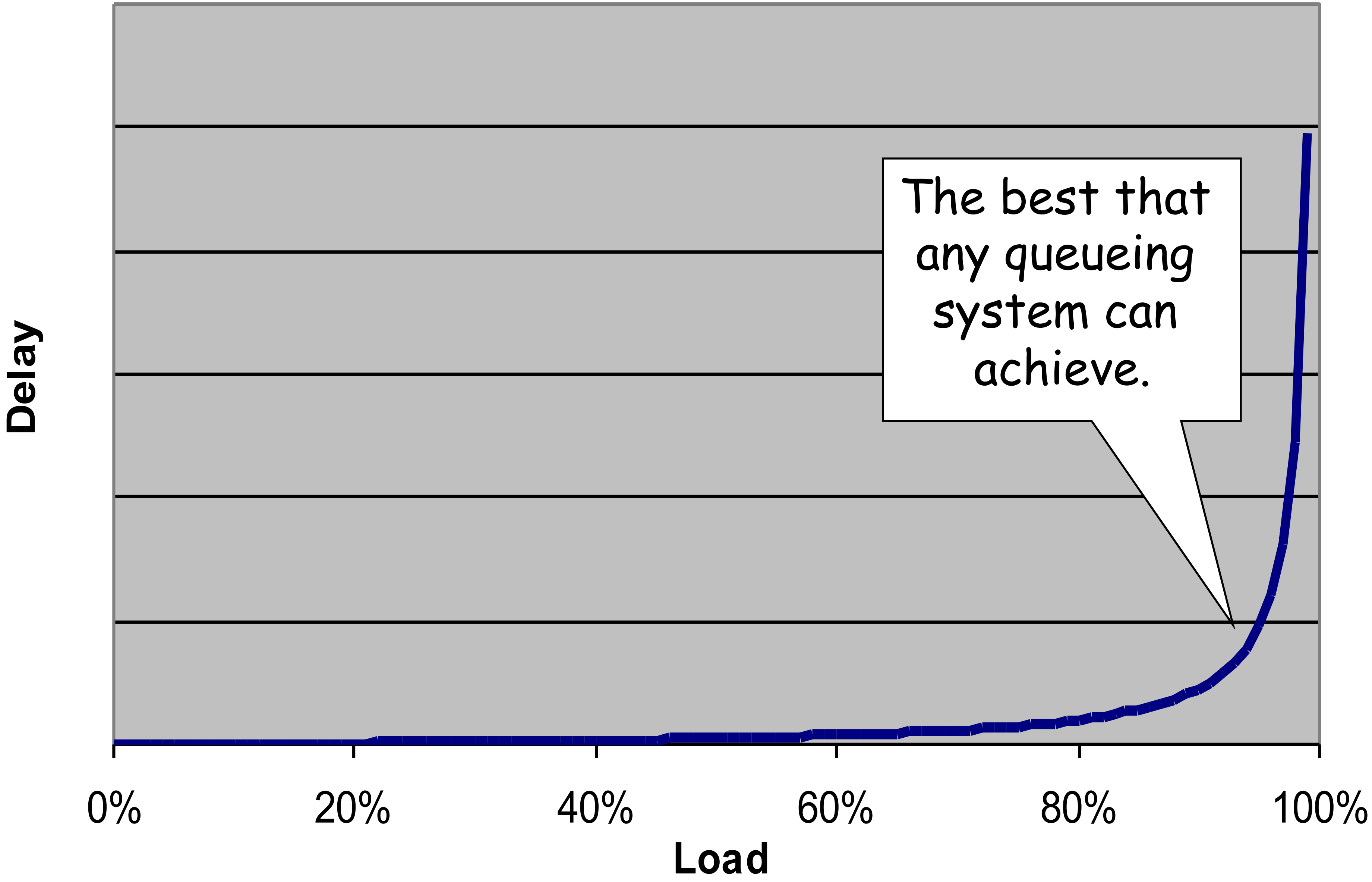
- Output Queueing
- Input Queueing
  - Scheduling algorithms for Fabric
- Combined Input and Output Queueing (CIOQ)

What is an Ideal Router?



- Output Queued (OQ) routers are ideal but not practical
  - ✓ It minimizes the delay faced by a packet
  - ✗ The bandwidth to each output is NR, the total bandwidth is N^2R
  - ✗ The cost and power consumption is prohibitive

A Router without Input Queues

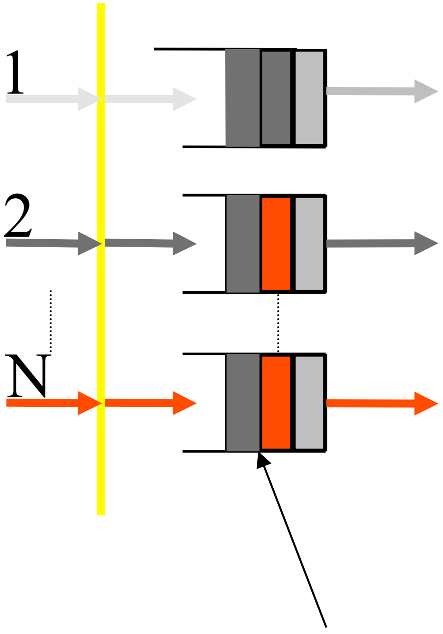


The best that any queueing system can achieve.

Interconnects

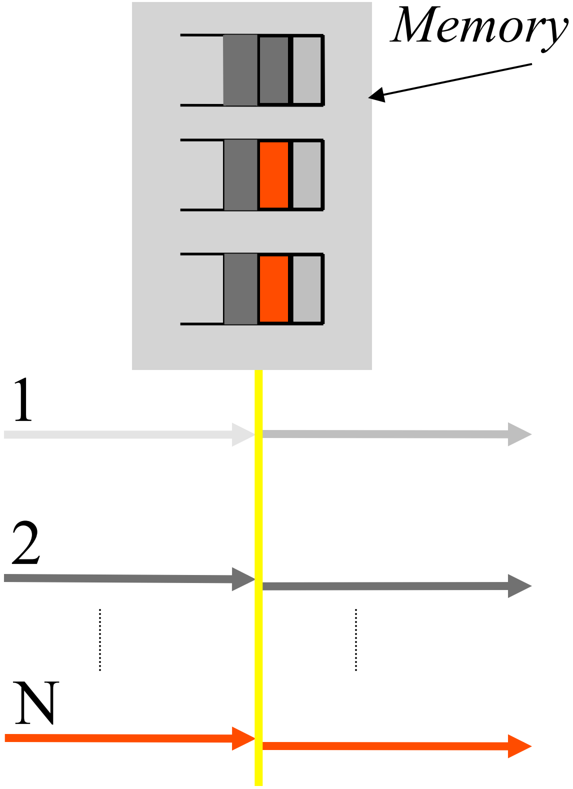
Output Queueing

Individual Output Queues



Memory b/w = (N+1) R

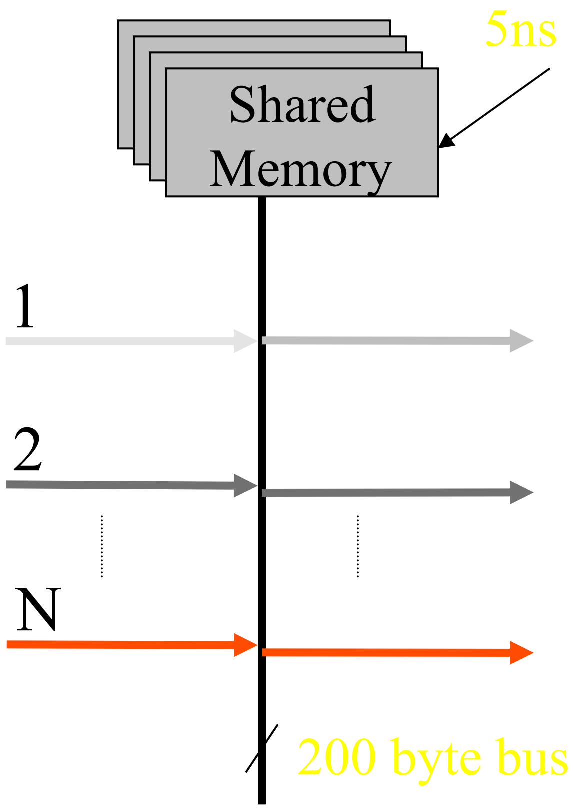
Centralized Shared Memory



Memory b/w = 2N R

Output Queueing

How fast can we make centralized shared memory-based router?



- 5ns per memory operation
- Two memory operations per packet
- Therefore, up to 160Gb/s

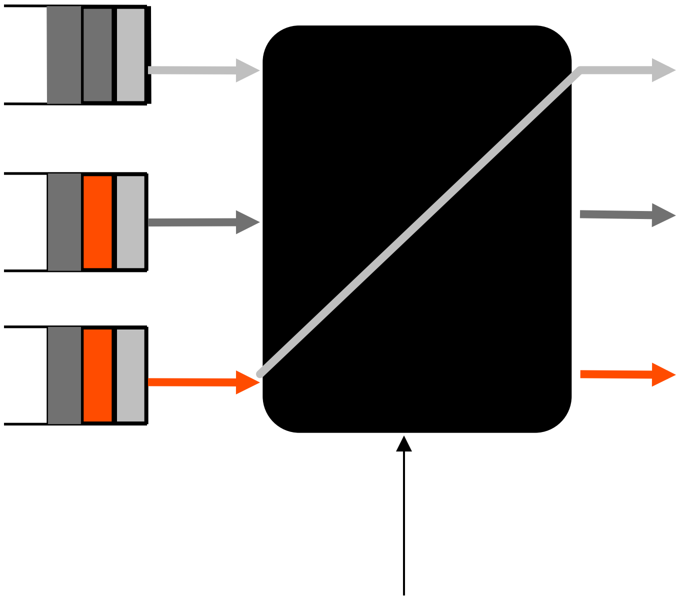
Summary of OQ Switches

- Output queued switches are ideal
  - Work-conserving.
  - Maximize throughput.
  - Minimize expected delay (for fixed length packets).
  - Permit delay guarantees for constrained traffic.
- Output queued switches don't scale well
  - Requires N memory writes per time slot.
  - Memory bandwidth (dictated by the random-access time of a memory) is a bottleneck.
  - Parallelism is not straightforward.

Interconnects

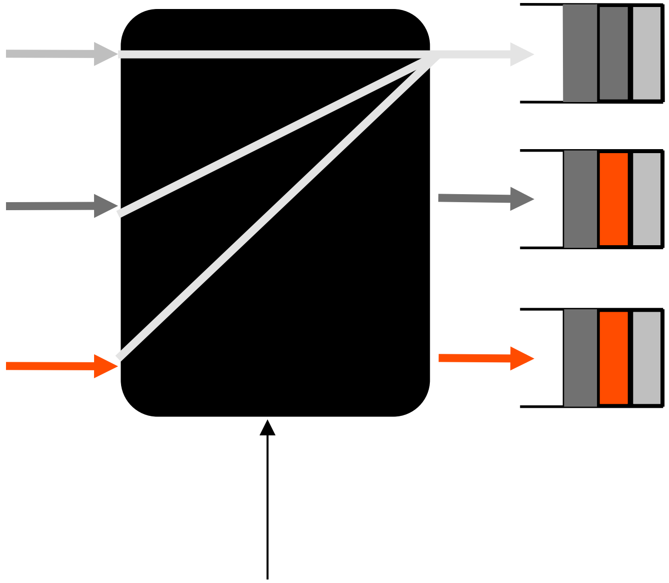
Two basic techniques

Input Queueing



Usually a non-blocking switch fabric (e.g. crossbar)

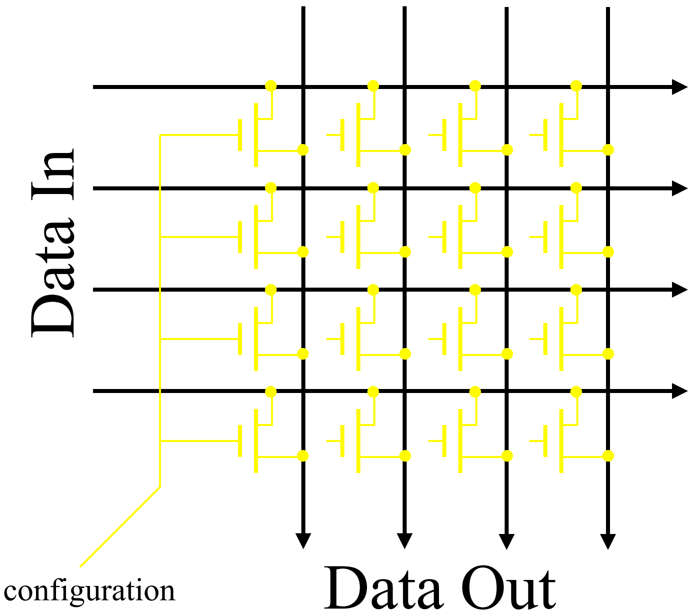
Output Queueing

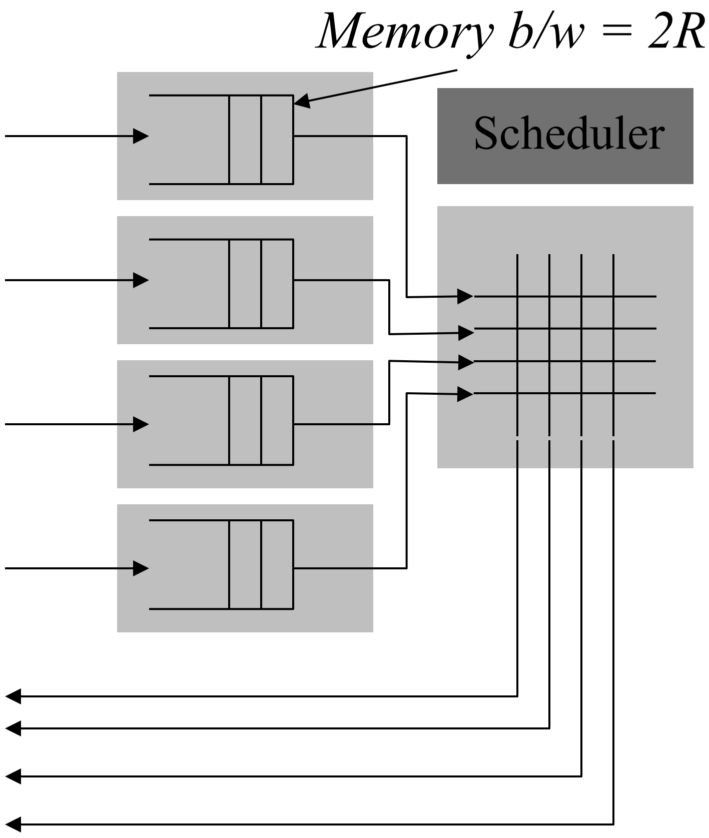


Usually a fast bus

Interconnects

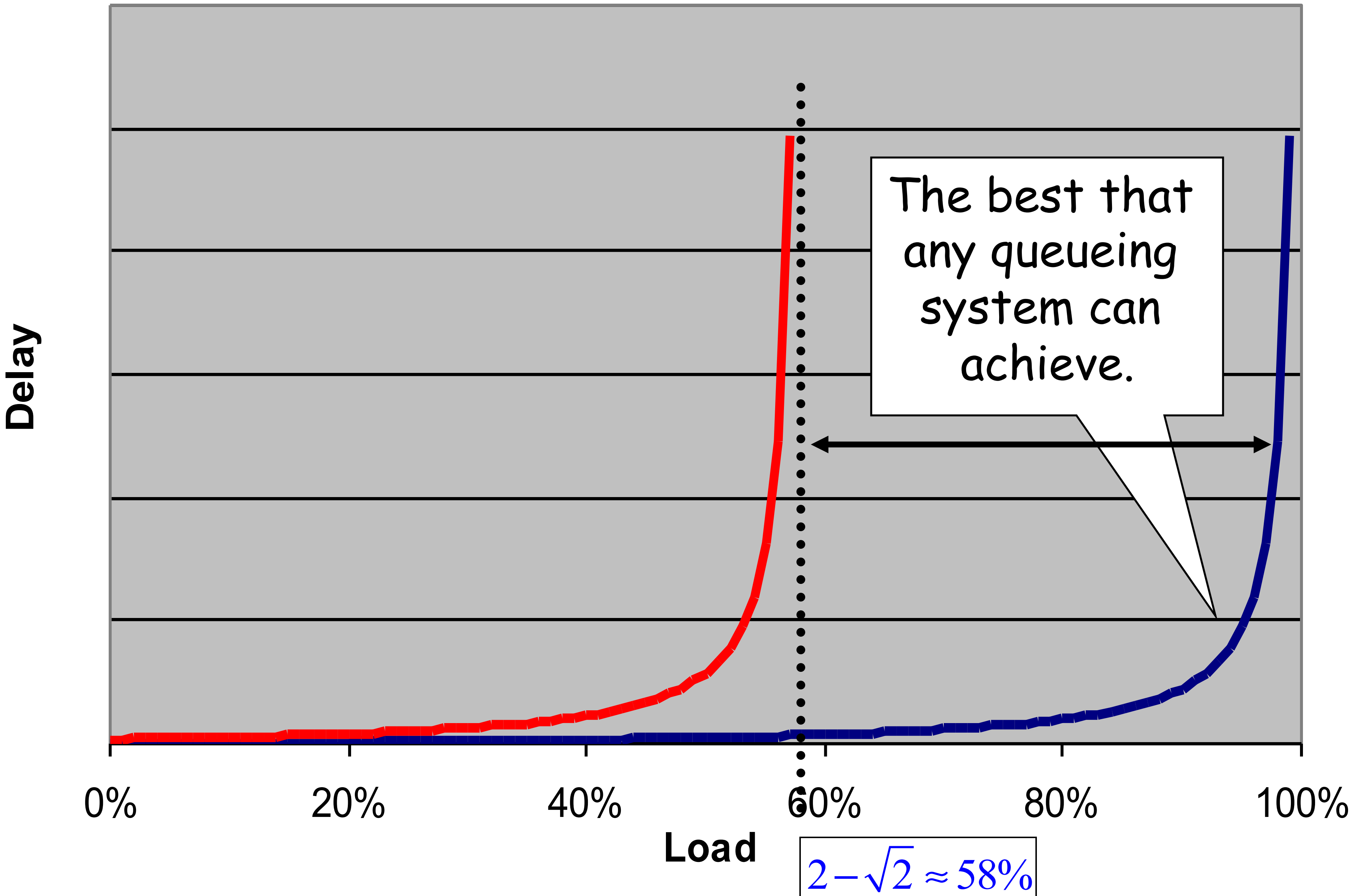
Input Queueing with Crossbar





A Router with Input Queues

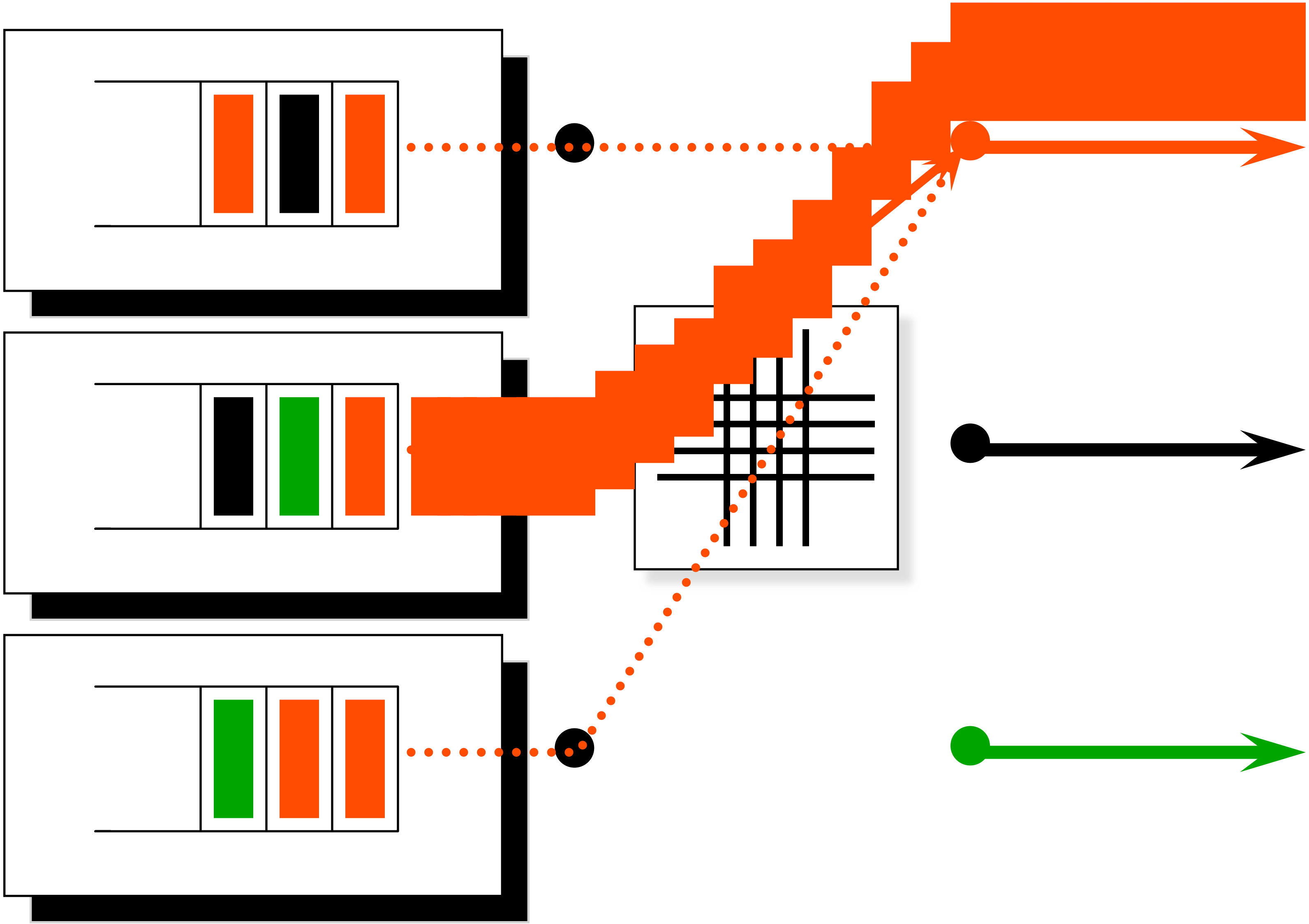
Head of Line Blocking



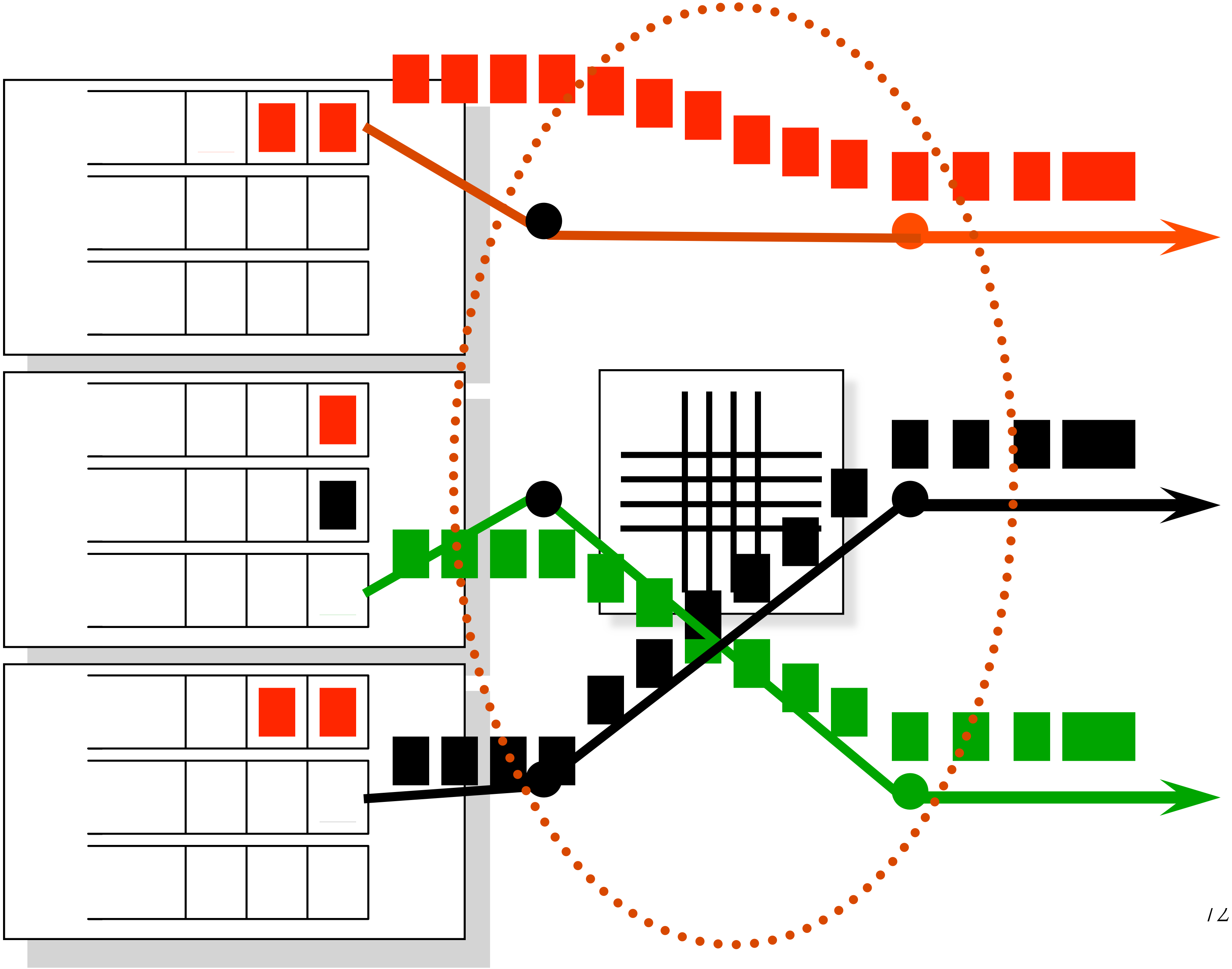
The best that any queueing system can achieve.

$\boxed{2-\sqrt{2} \approx 58\%}$

Head of Line Blocking

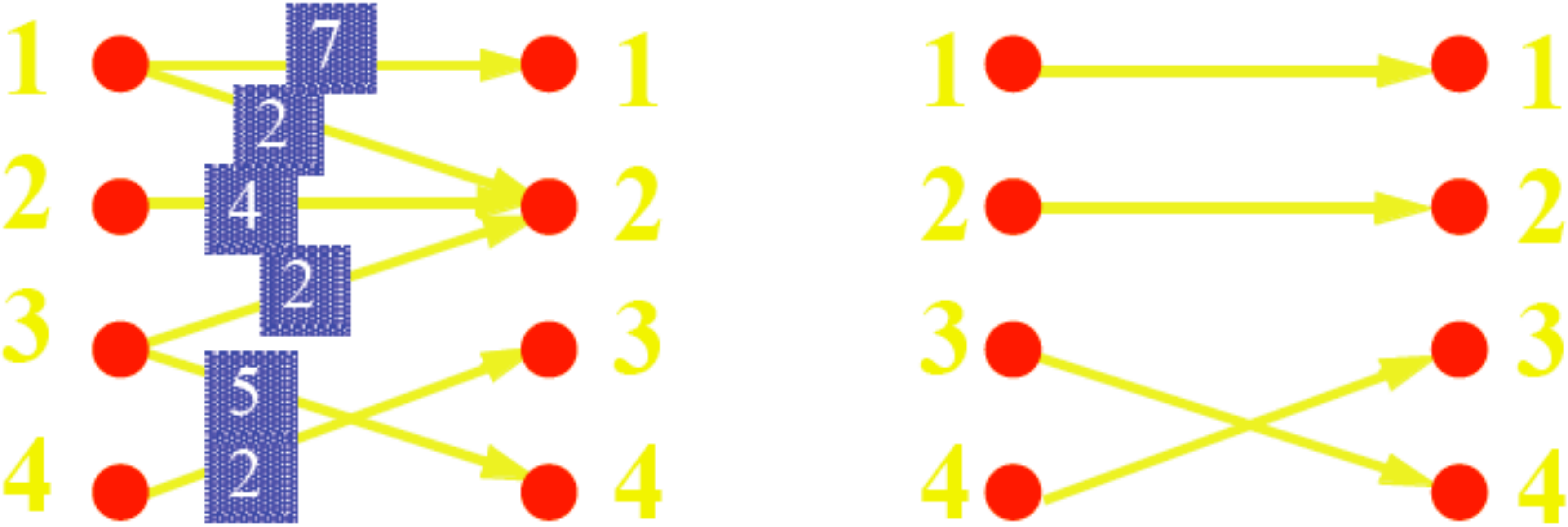


Virtual Output Queues



Input Queueing with virtual output queues

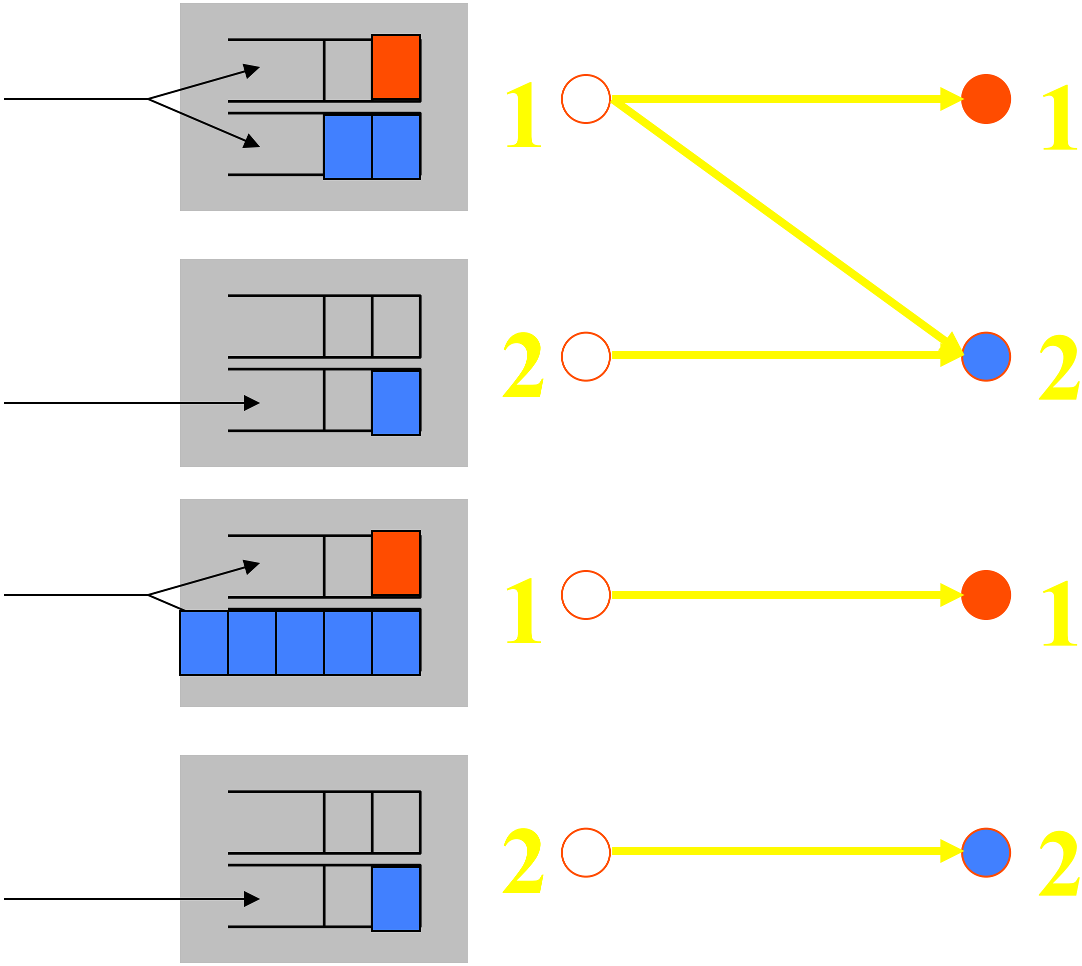
Scheduling



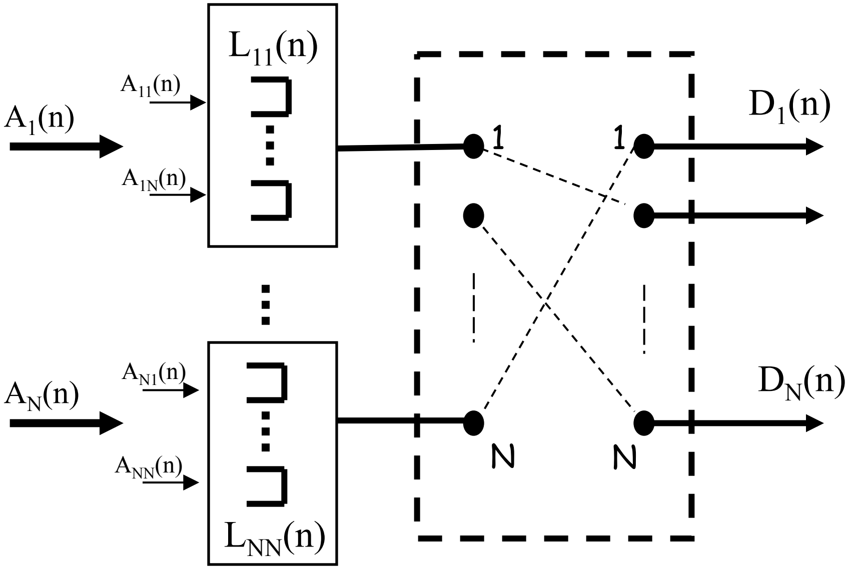
Question: *Maximum weight or maximum size?*

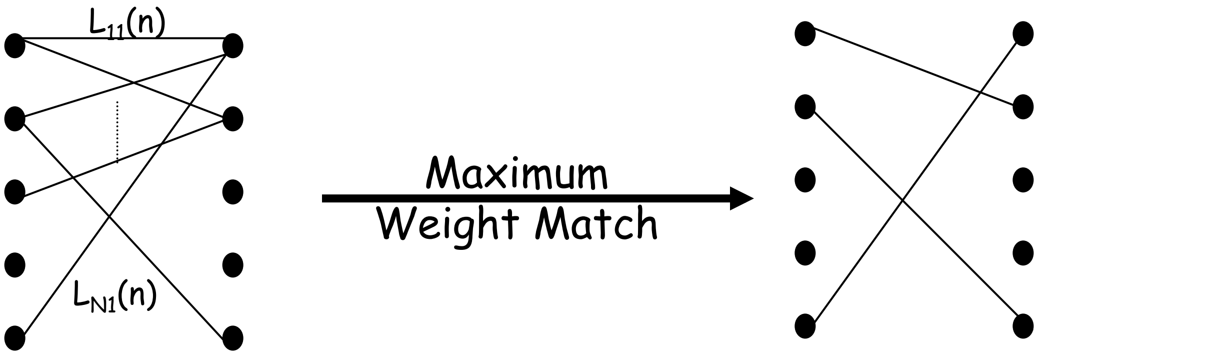
Input Queueing with virtual output queues

Scheduling



Maximum Weight Matching can achieve 100% throughput as the output queueing

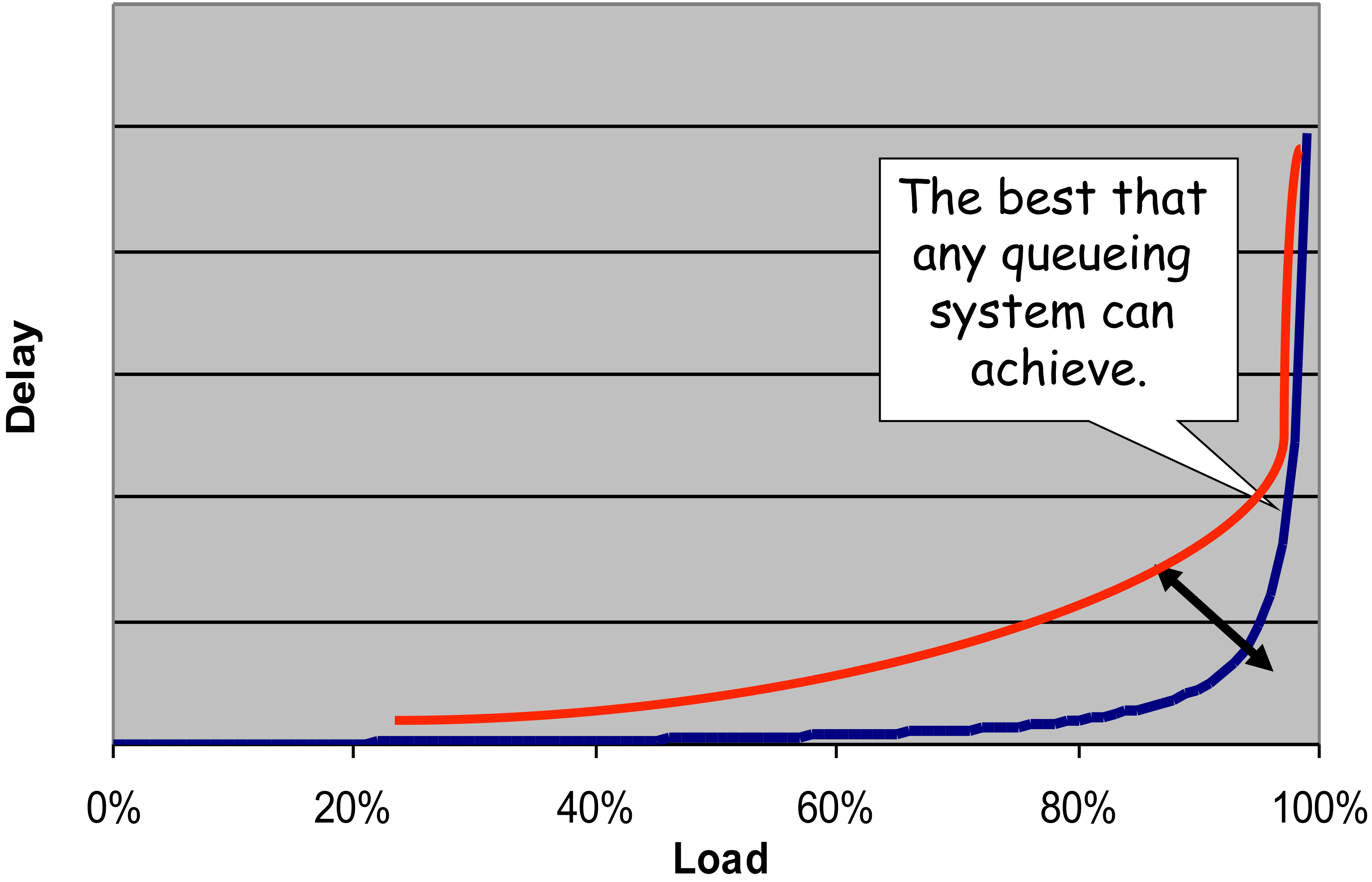




“Request” Graph

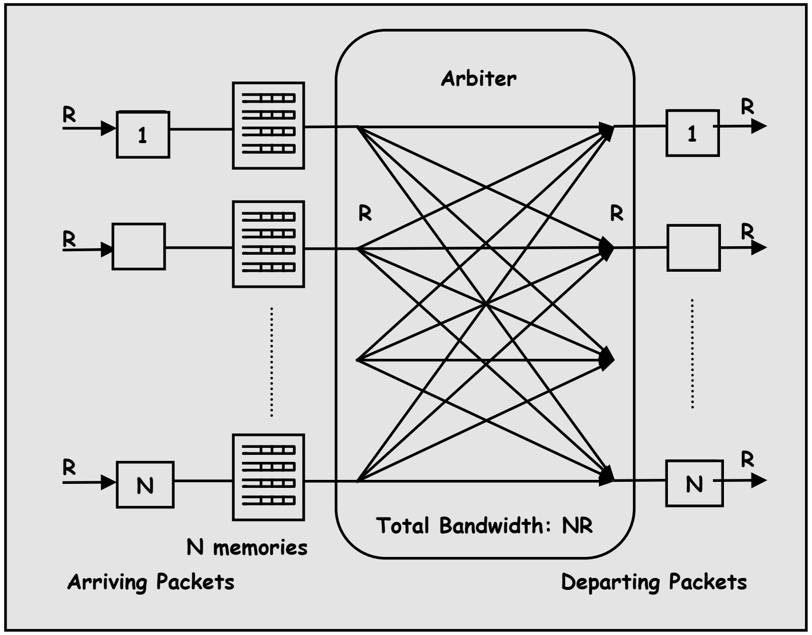
Bipartite Match

A Router with Virtual Output Queues

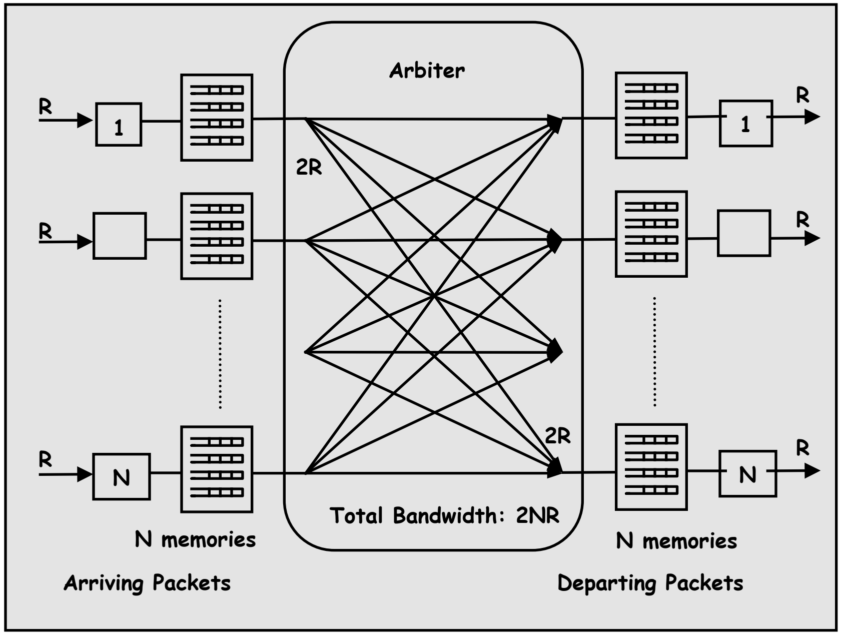


The best that any queueing system can achieve.

CIOQ Router Model



Input Queued Switch

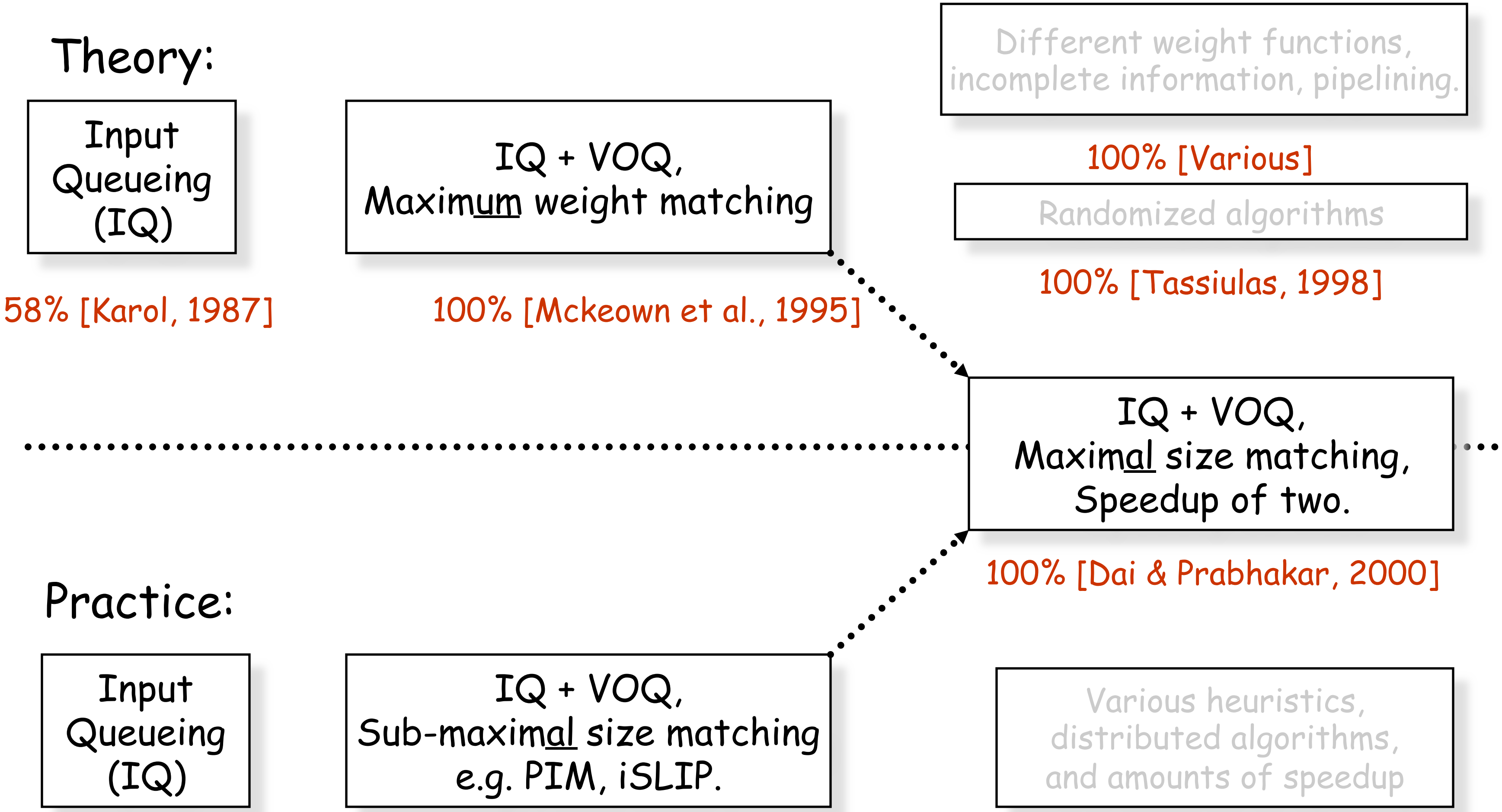


Combined Input-Output Queued Switch

- CIOQ switches offer some advantages over OQ switches, but are still not practical
  - ✓ They can give the same delay guarantees as OQ switches
  - ✓ They need a switching bandwidth of only 2NR

  - ✗ They have high computational complexity
  - ✗ The model does not capture many different architectures

The Evolution of Input Queueing Switching



Theory:

Input Queueing (IQ)

58% [Karol, 1987]

IQ + VOQ, Maximum weight matching

100% [Mckeown et al., 1995]

Different weight functions, incomplete information, pipelining.

100% [Various]

Randomized algorithms

100% [Tassiulas, 1998]

IQ + VOQ, Maximal size matching, Speedup of two.

100% [Dai & Prabhakar, 2000]

Practice:

Input Queueing (IQ)

IQ + VOQ, Sub-maximal size matching e.g. PIM, iSLIP.

Various heuristics, distributed algorithms, and amounts of speedup

Input Queueing References

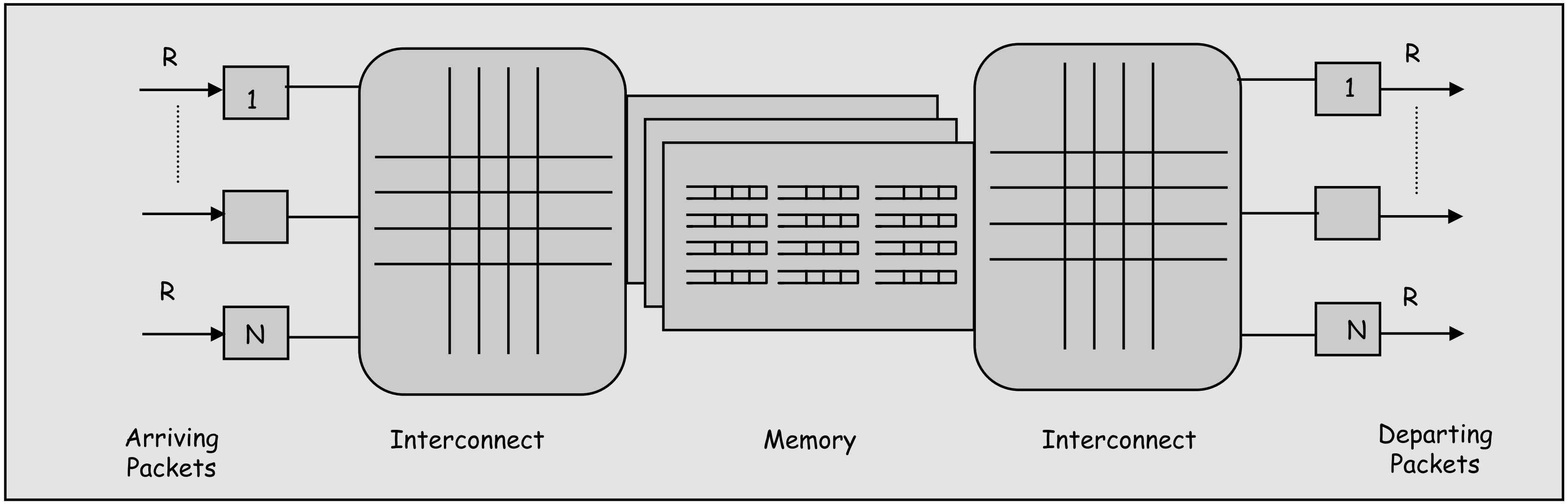
References

- M. Karol et al. “Input vs Output Queueing on a Space-Division Packet Switch”, IEEE Trans Comm., Dec 1987, pp. 1347-1356.
- Y. Tamir, “Symmetric Crossbar arbiters for VLSI communication switches”, IEEE Trans Parallel and Dist Sys., Jan 1993, pp.13-27.
- T. Anderson et al. “High-Speed Switch Scheduling for Local Area Networks”, ACM Trans Comp Sys., Nov 1993, pp. 319-352.
- N. McKeown, “The iSLIP scheduling algorithm for Input-Queued Switches”, IEEE Trans Networking, April 1999, pp. 188-201.
- C. Lund et al. “Fair prioritized scheduling in an input-buffered switch”, Proc. of IFIP-IEEE Conf., April 1996, pp. 358-69.
- A. Mekkitikul et al. “A Practical Scheduling Algorithm to Achieve 100% Throughput in Input-Queued Switches”, IEEE Infocom 98, April 1998.

Other approaches

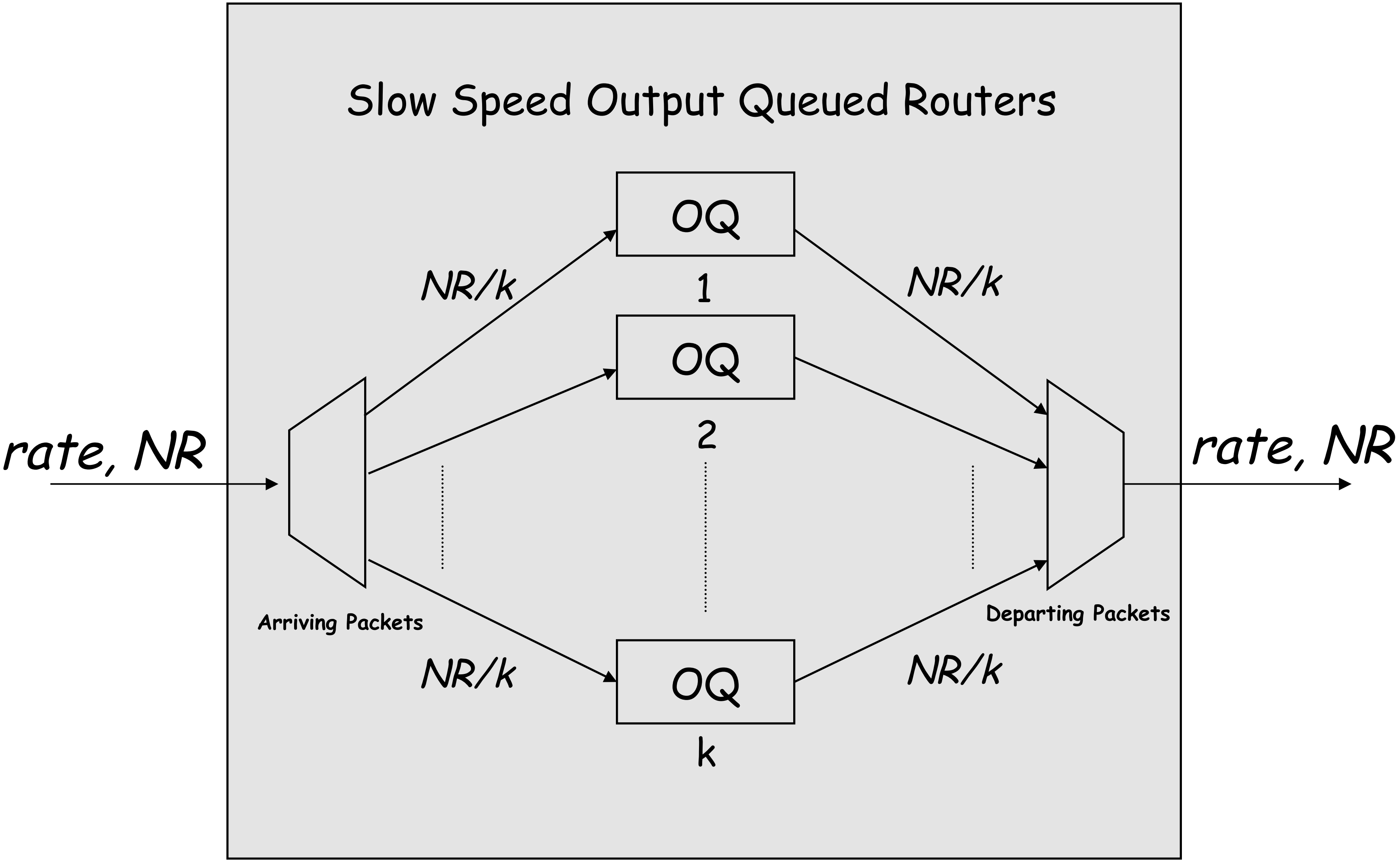
- [Tassiulas 1998] 100% throughput possible for simple randomized algorithm with memory.
- [Giaccone et al. 2001] “Apsara” algorithms.
- [Iyer and Mckeown 2000] Parallel switches can achieve 100% throughput and emulate an output queued switch.
- [Chang et al. 2000, Keslassy et al. Sigcomm 2003] A 2-stage switch with no scheduler can give 100% throughput.
- [Iyer, Zhang and Mckeown 2002] Distributed shared memory switches can emulate an output queued switch.

An Alternative Single Buffered Router Model



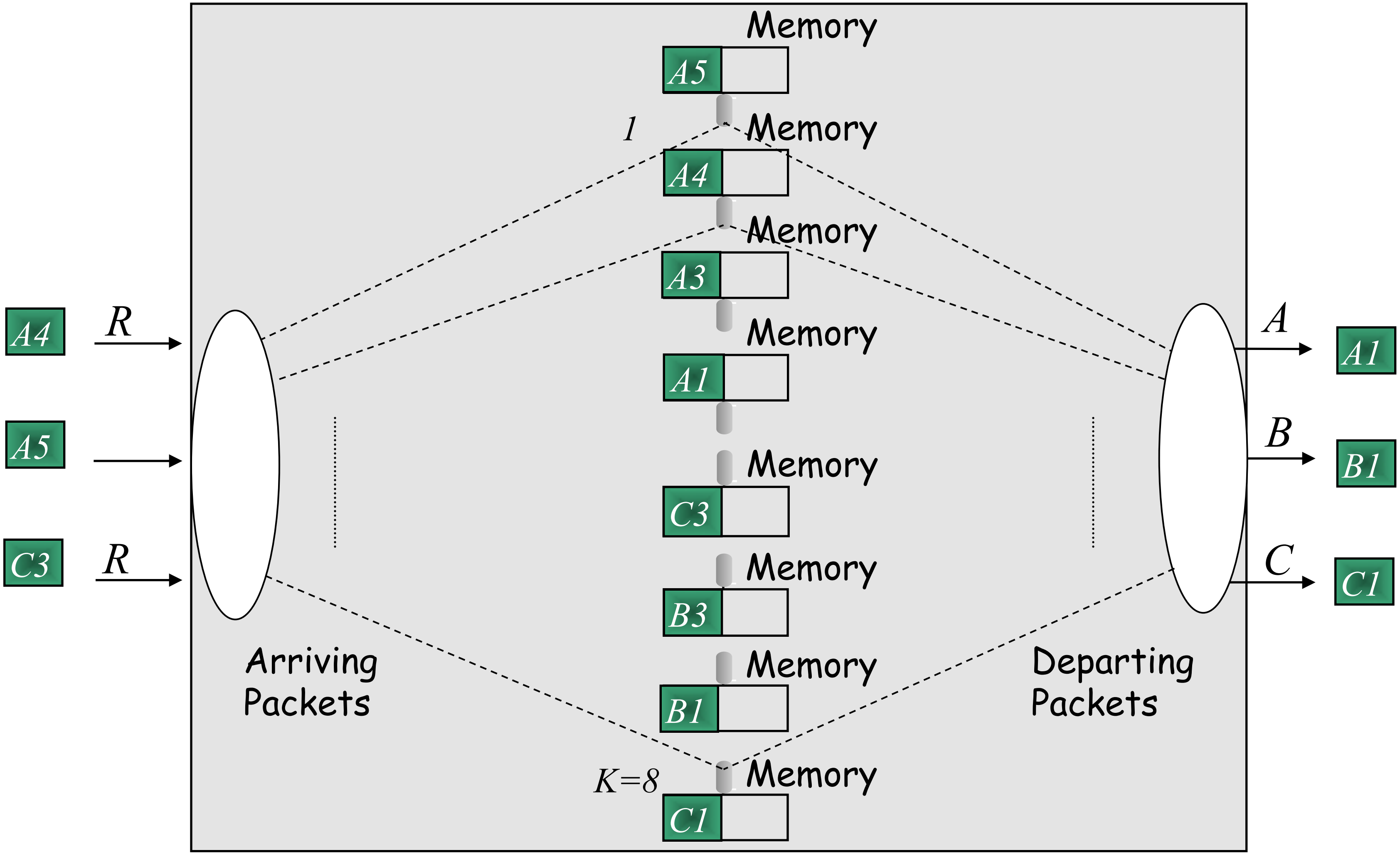
- Single Buffered Routers buffer packets only once
  - The interconnects may be
    - physically separate or merged
    - one of the interconnects may be a simple pass through
  - The memory can be
    - centralized or distributed
    - one or many
    - statically or dynamically allotted amongst all ports

Parallel Packet Switch



The Parallel Shared Memory (PSM) Router

At most ONE operation – a write OR a read per time slot



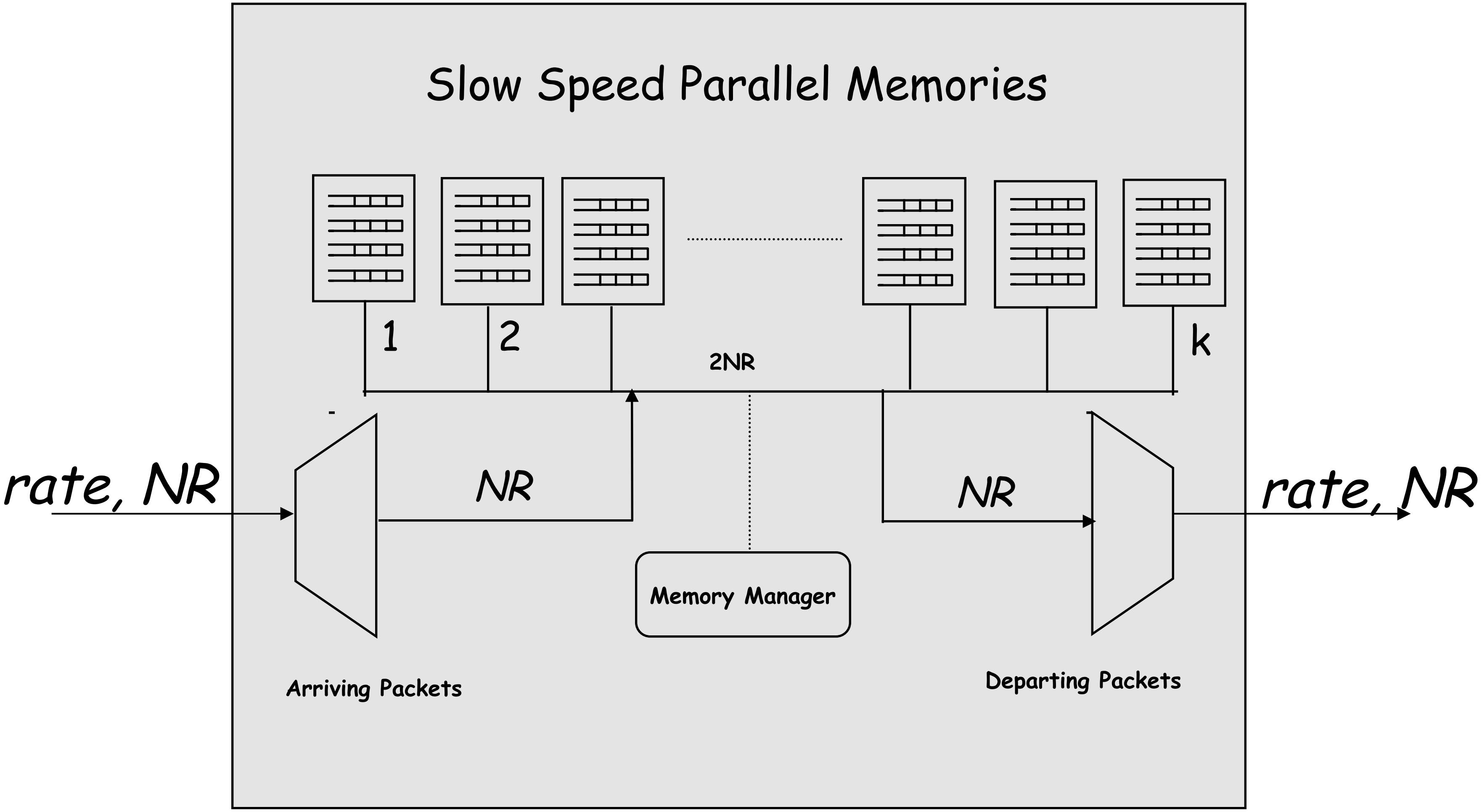
It can be shown that if $k = \# \, of$ Memory pool > 3N -1 (where N=# of linecards), there will be enough Memory pools to avoid conflicts => 100% throughput can be achieved.

In this e.g. N = 3, so but 8 memories suffice

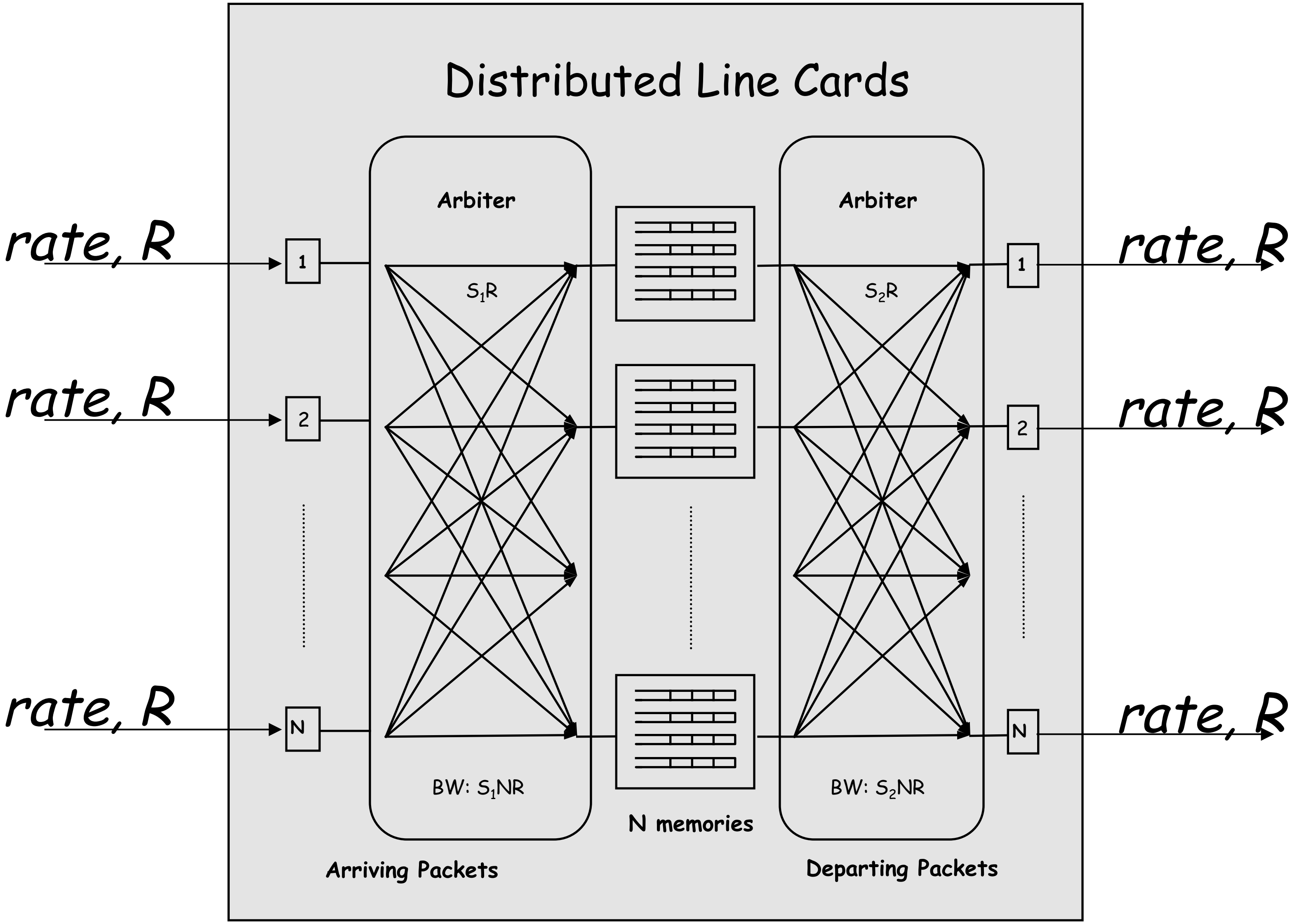
Why k >= 3N -1 suffices ?

- When a packet arrives in a timeslot, it must choose a memory NOT chosen by:
  1. The N-1 other packets that arrive at that timeslot
  2. The N other packets that depart at that timeslot
  3. The N-1 other packets that can depart at the same time as this packet departs (in the future)

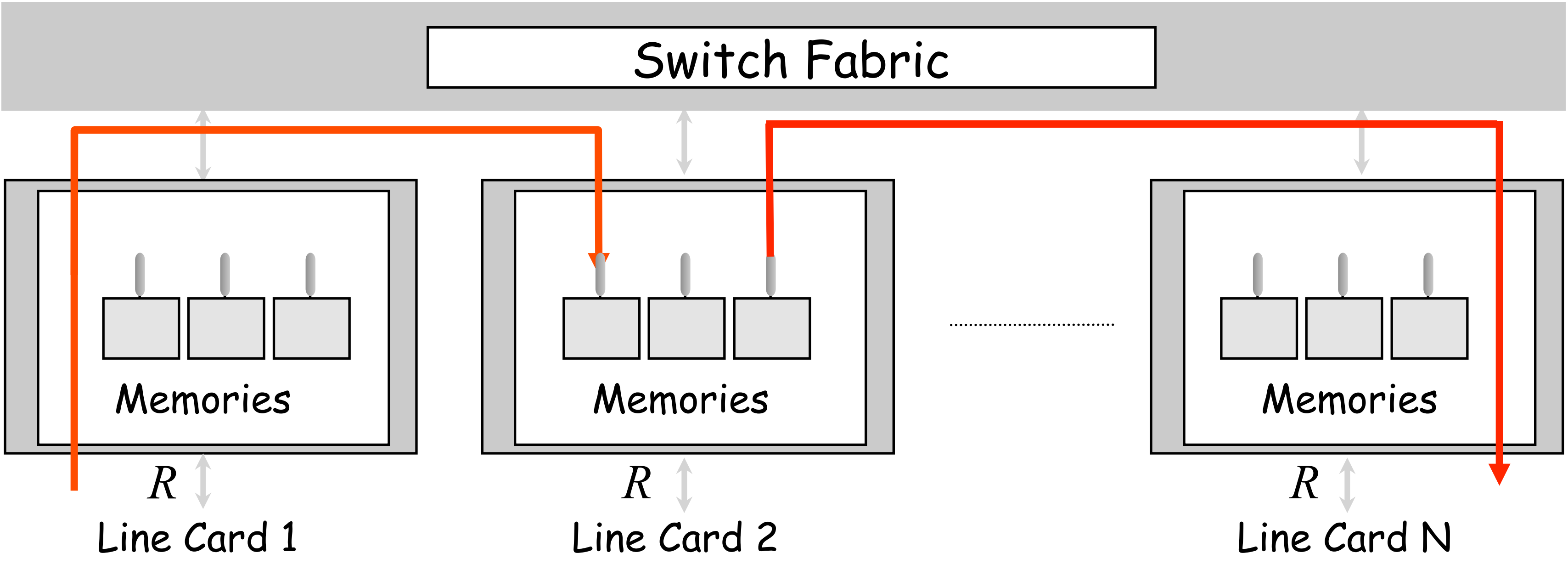
Parallel Shared Memory Router



Distributed Shared Memory Router



Distributed Shared Memory Router



- The central memories are moved to distributed line cards and shared.
- Memory and line cards can be added incrementally.
- The bandwidth in & out of each memory is at the line rate *R*
- From the PSM theorem, $3N-1$ memories which can perform one operation per time slot i.e. a total memory bandwidth of ⍰ $3NR$ suffices for the router to be work-conserving.

Ref: Sundar Iyer, Rui. Zhang Nick Mckeown, “Routers with a Single Stage of Buffering,” ACM Sigcomm 2002.

A Commercial DSM Router

The Juniper M and T series Routers

Capacity per Router: from 10's to 1000's Gbps

Scaling the Cross-Bar switch fabric

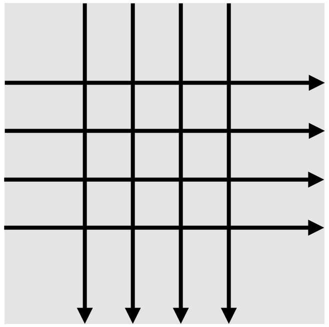
- Up until now, we have focused on high performance packet switches with:
  - A crossbar switching fabric,
  - Input queues (and possibly output queues as well),
  - Virtual output queues, and
  - Centralized arbitration/scheduling algorithm.
  - Even Distributed Shared Memory Router/Switch needs a Cross-bar fabric

- Now, let's talk about the implementation of the crossbar switch fabric itself.
  - How are they built ?
  - How do they scale ?
  - What limits their capacity ?

Crossbar switch

Limiting factors

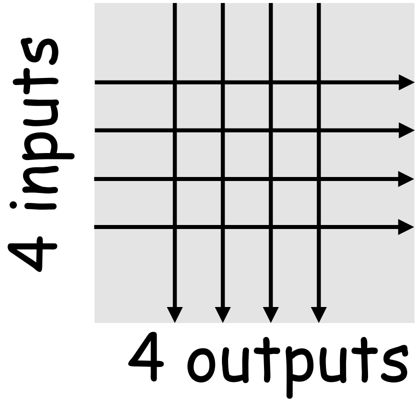
- N^2 crosspoints per chip, or N x N-to-1 multiplexers
- It's not obvious how to build a crossbar from multiple chips,
- Capacity of "I/O"s per chip.
  - A Practical Example: About 300 pins each operating at 3.125Gb/s ~= 1Tb/s per chip.
  - About 1/3 to 1/2 of this capacity available in practice because of overhead and speedup.
  - Crossbar chips today are limited by "I/O" capacity.



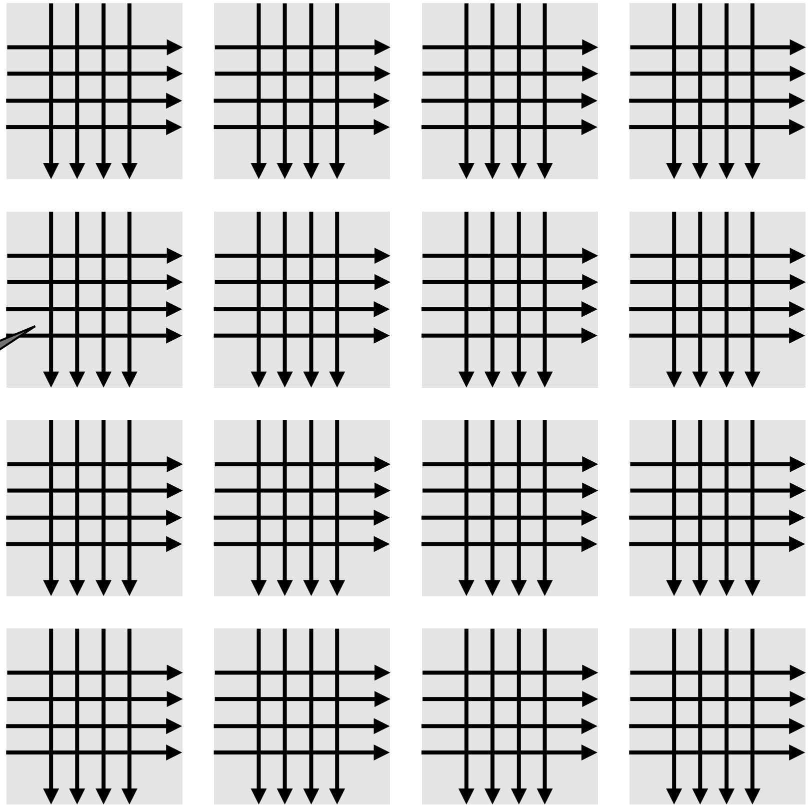
Scaling number of outputs:

Trying to build a crossbar from multiple chips

Building Block:

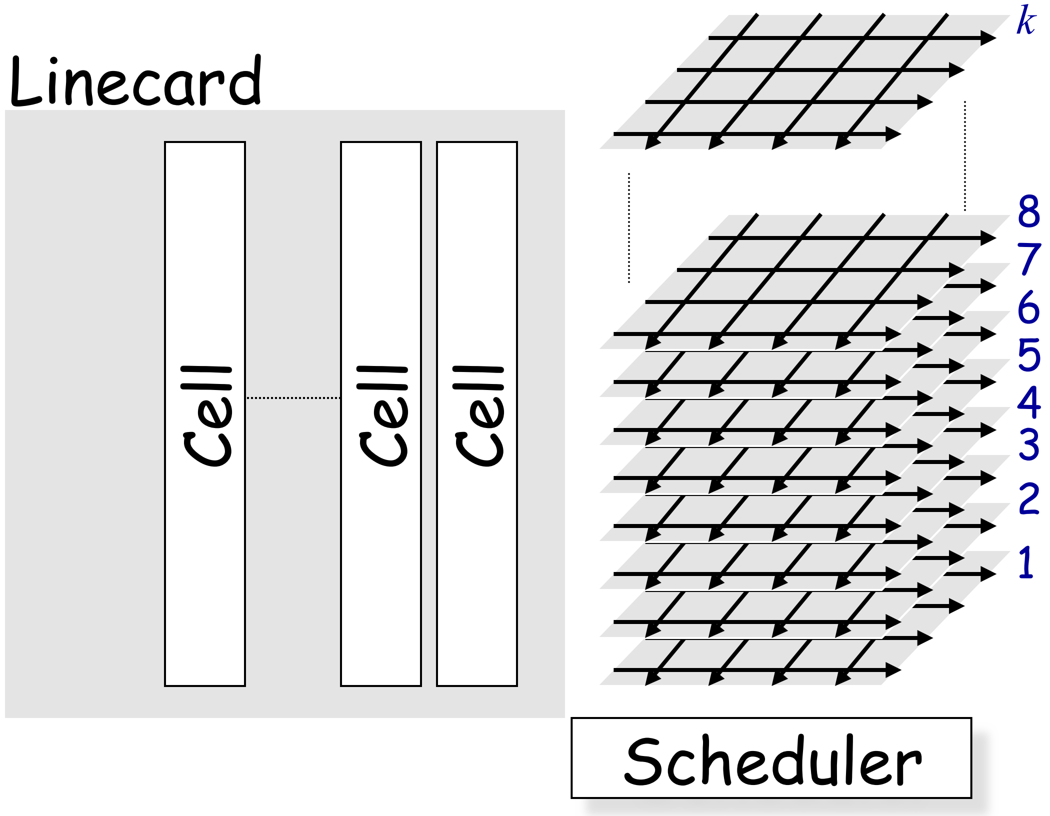


16x16 crossbar switch:



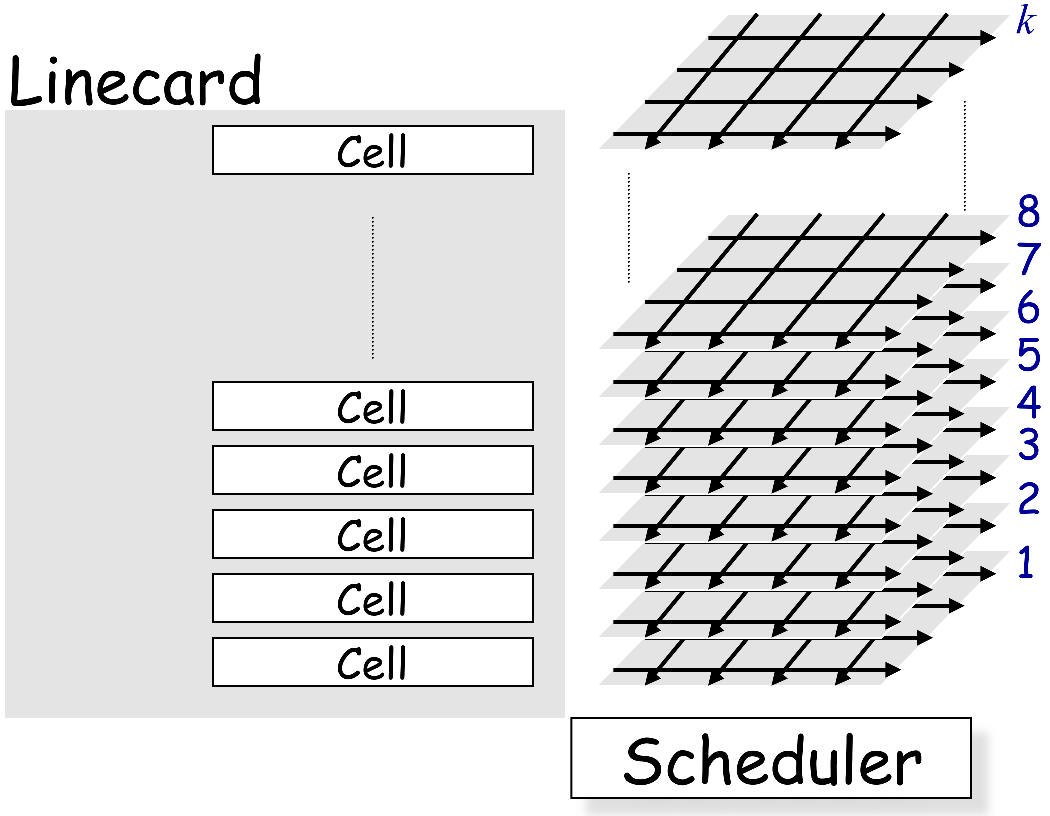
Eight inputs and eight outputs required!

Scaling line-rate:
Bit-sliced parallelism



- Cell is "striped" across multiple identical planes.
- Crossbar switched "bus".
- Scheduler makes same decision for all slices.

Scaling line-rate:
Time-sliced parallelism

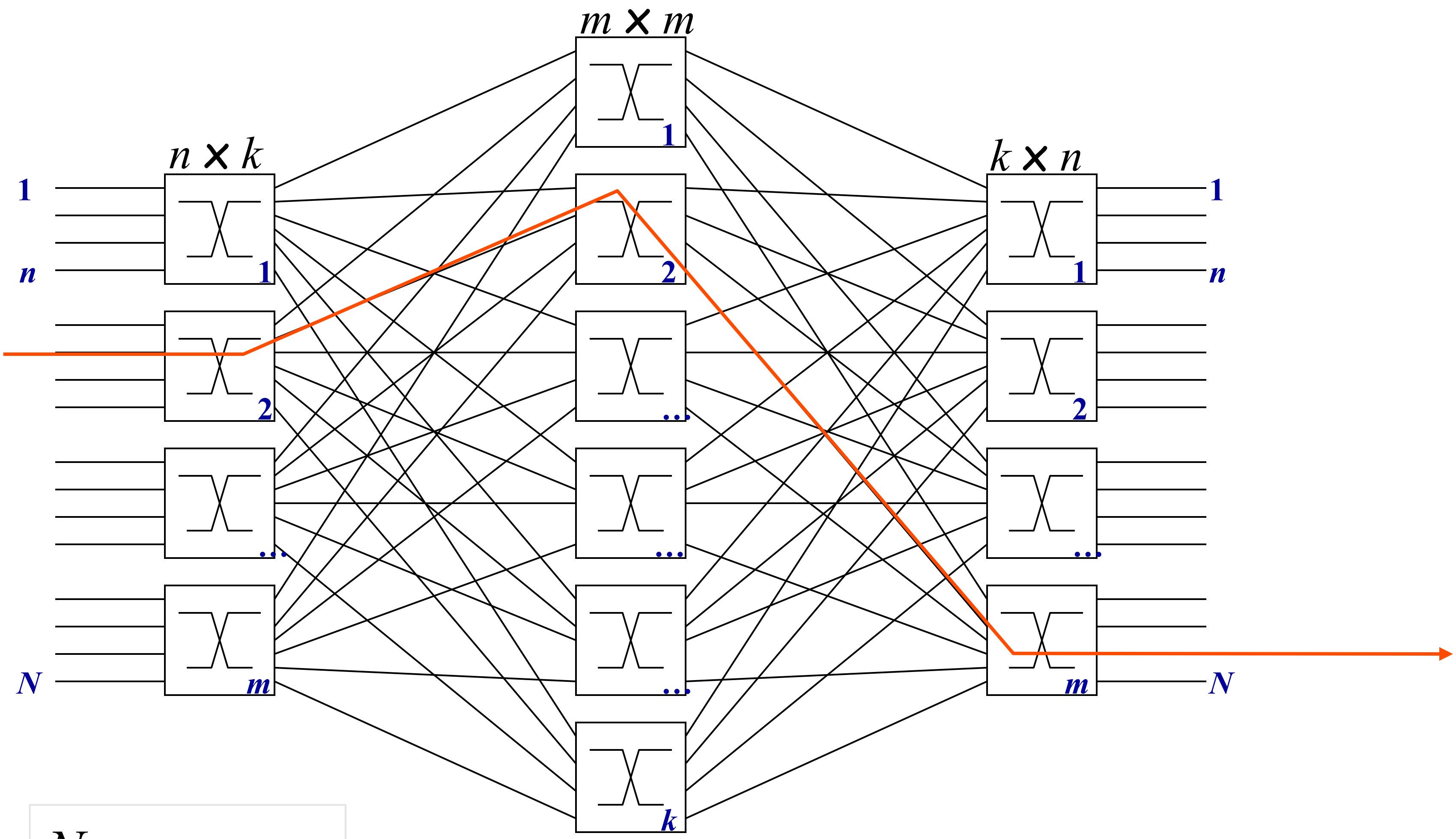


- Cell carried by one plane; takes k cell times.
- Scheduler is unchanged.
- Scheduler makes decision for each slice in turn.

Scaling a crossbar

- Conclusion: scaling the capacity is relatively straightforward (although the chip count and power may become a problem).
- What if we want to increase the number of ports?
- Can we build a crossbar-equivalent from multiple stages of smaller crossbars?
- If so, what properties should it have?

3-stage Clos Network

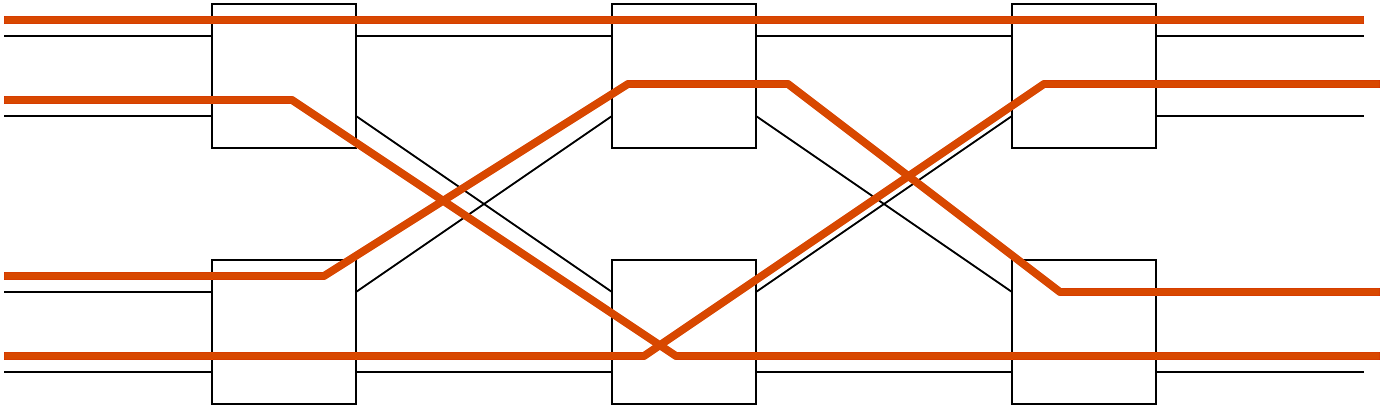


$N = n$ x m

$k >= n$

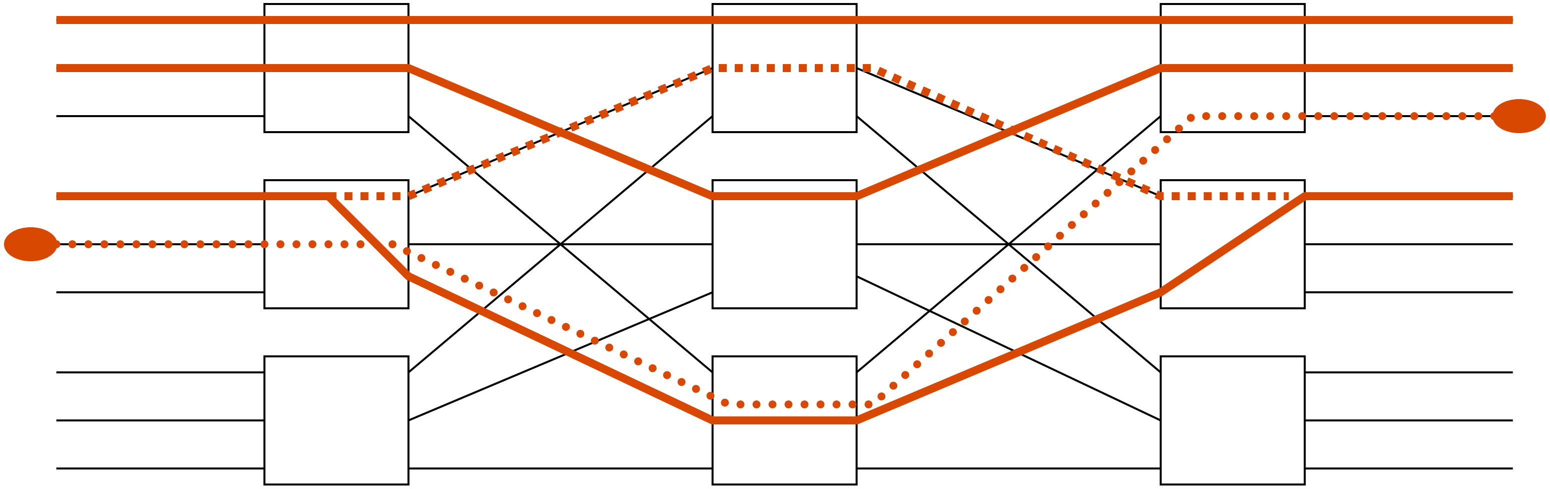
With $k = n,$ is a Clos network non-blocking like a crossbar?

Consider the example: scheduler chooses to match (1,1), (2,4), (3,3), (4,2)



With $k = n$ is a Clos network non-blocking like a crossbar?

Consider the example: scheduler chooses to match (1,1), (2,2), (4,4), (5,3), …



By rearranging matches, the connections could be added. It can be shown that k>=n makes the Clos network "rearrangeably non-blocking"!

Implementation

Pros

- A rearrangeably non-blocking switch can perform any permutation
- A cell switch is time-slotted, so all connections are rearranged every time slot anyway

Cons

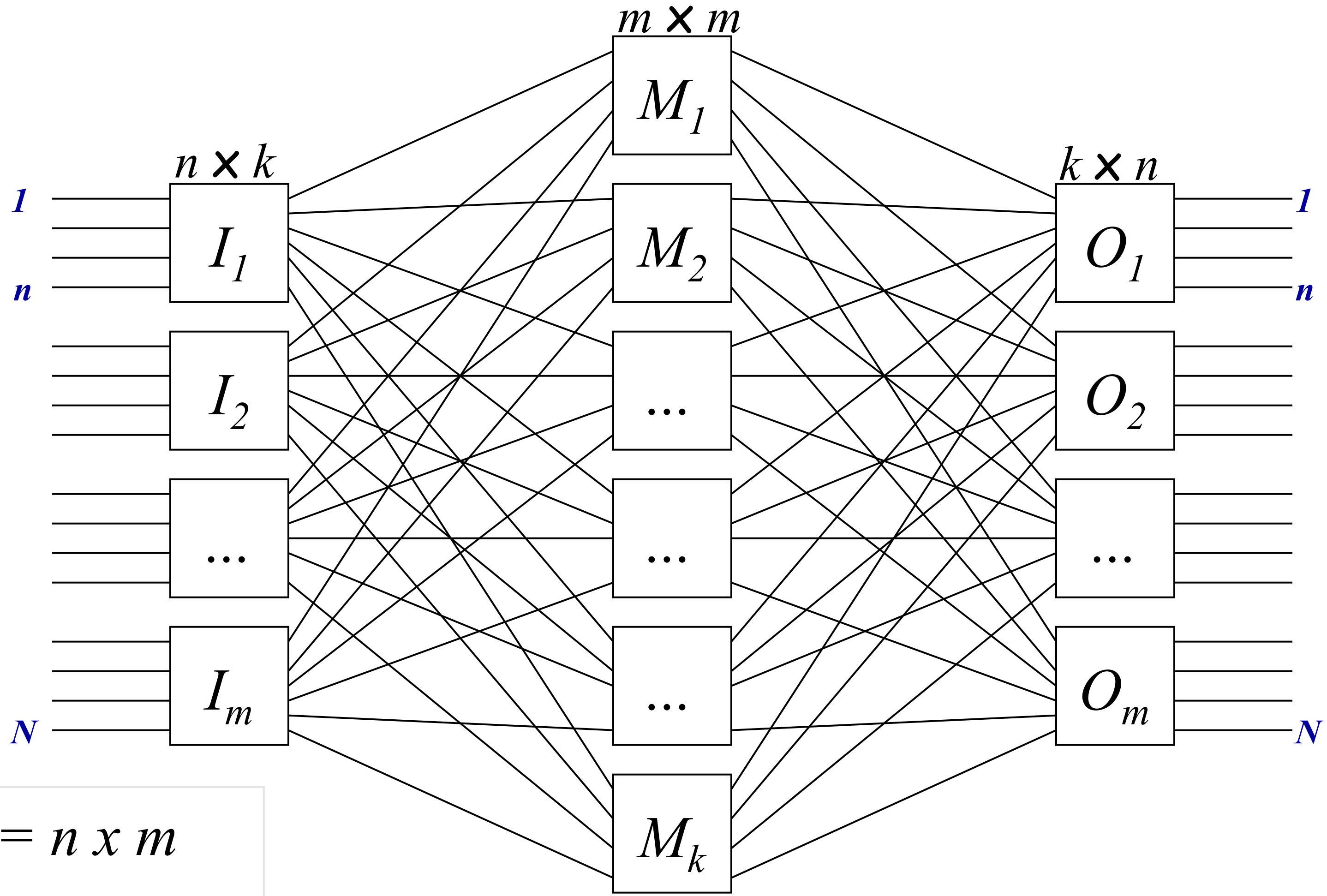
- Rearrangement algorithms are complex (in addition to the scheduler)

Can we eliminate the need to rearrange?

Strictly non-blocking Clos Network

Clos' Theorem:

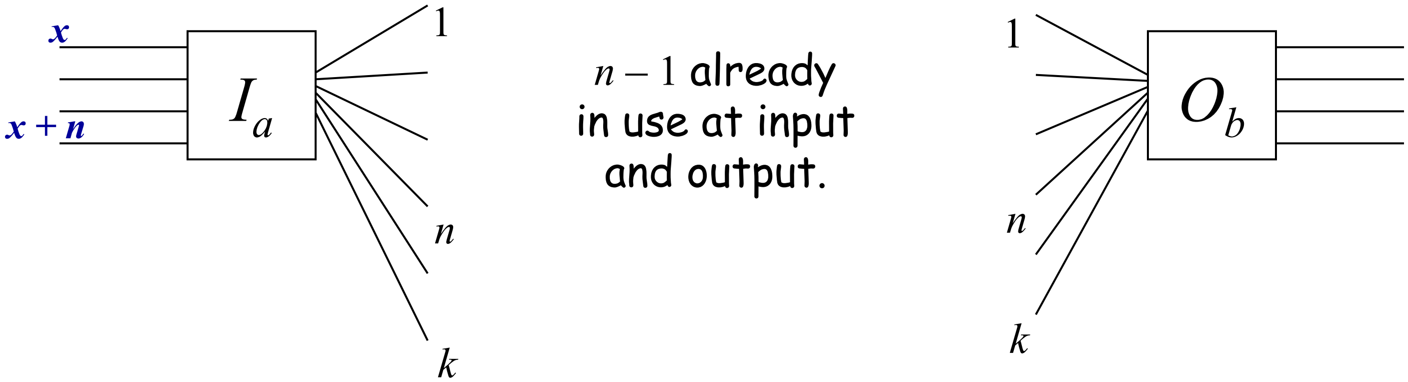
If $k >= 2n - 1$, then a new connection can always be added without rearrangement.



$N = n\ x\ m$

$k >= n$

Clos Theorem



$n-1$ already in use at input and output.

1. Consider adding the n-th connection between 1st stage I_a and 3rd stage O_b.
2. We need to ensure that there is always some center-stage M available.
3. If $k > (n-1) + (n-1)$, then there is always an M available. i.e. we need $k >= 2n-1$.

Recall: Multi-stage (Multi-chassis) Core Routers

T4000

	Ports
208	10 Gbps
16	40 Gbps
16	100 Gbps

T1600

	Ports
80	10 Gbps
16	40 Gbps
8	100 Gbps

T640

	Ports
40	10 Gbps
8	40 Gbps

TX Matrix Plus

	Ports
832	10 Gbps
64	40 Gbps
64	100 Gbps

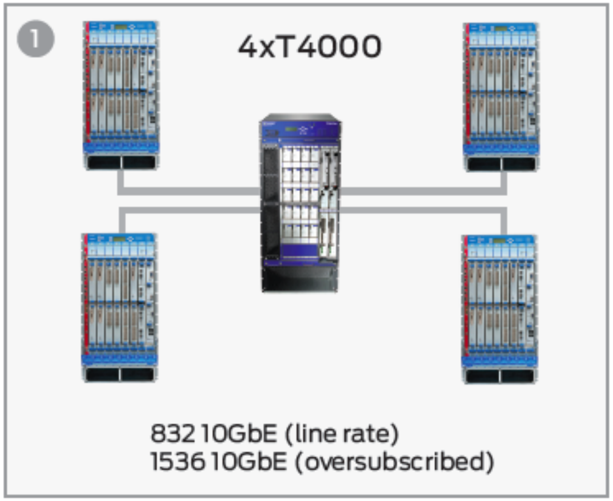
TX Matrix

	Ports
160	10 Gbps
32	40 Gbps

Table 1: Juniper Networks T Series Single Chassis Scaling Characteristics

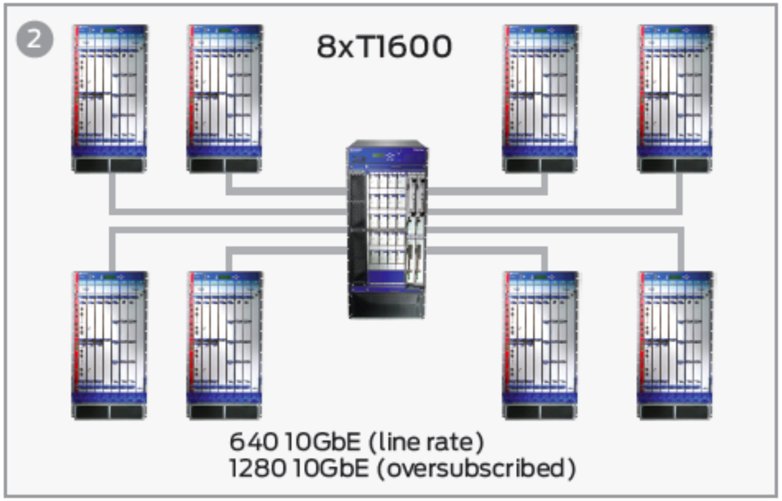
Platform	Throughput	Rack Space	10-Gigabit Ethernet Density	Fully Redundant Hardware	Multichassis Capable
T640	640 Gbps	1/2 rack (19 in)	40	Yes	Yes
T1600	1.6 Tbps	1/2 rack (19 in)	80 (line rate) 160 (oversubscribed)	Yes	Yes
T4000	4 Tbps	1/2 rack (19 in)	208 (line rate) 384 (oversubscribed)	Yes	Yes

Current Generation
T-series Routers from Juniper



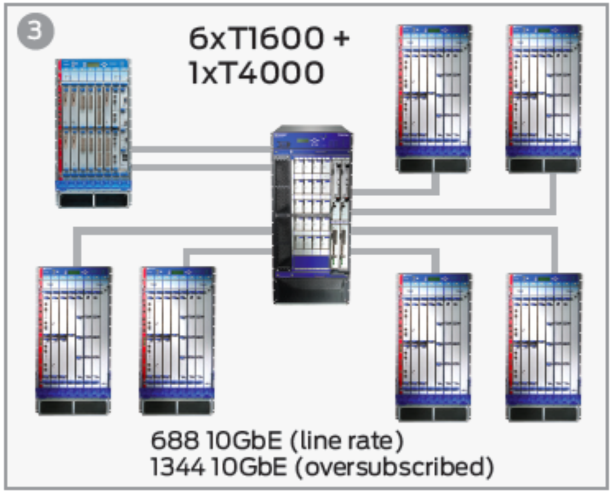
1 4xT4000

832 10GbE (line rate)
1536 10GbE (oversubscribed)



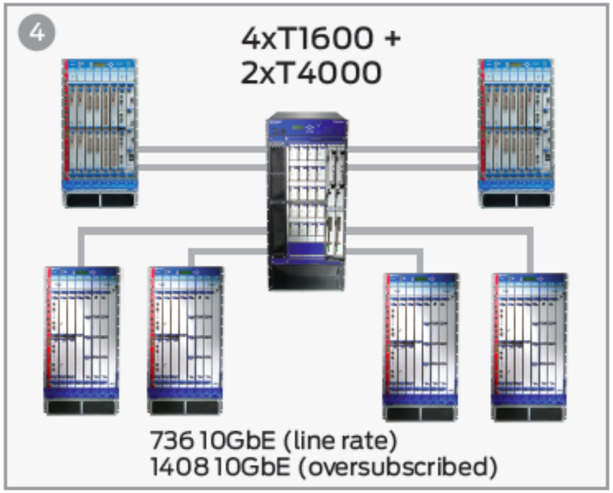
2 8xT1600

640 10GbE (line rate)
1280 10GbE (oversubscribed)



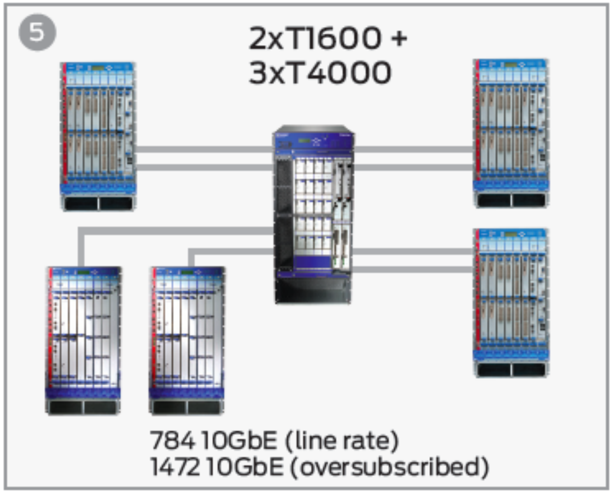
3 6xT1600 +
1xT4000

688 10GbE (line rate)
1344 10GbE (oversubscribed)



4 4xT1600 +
2xT4000

736 10GbE (line rate)
1408 10GbE (oversubscribed)



5 2xT1600 +
3xT4000

784 10GbE (line rate)
1472 10GbE (oversubscribed)

Competing Product (current Gen.) from Juniper

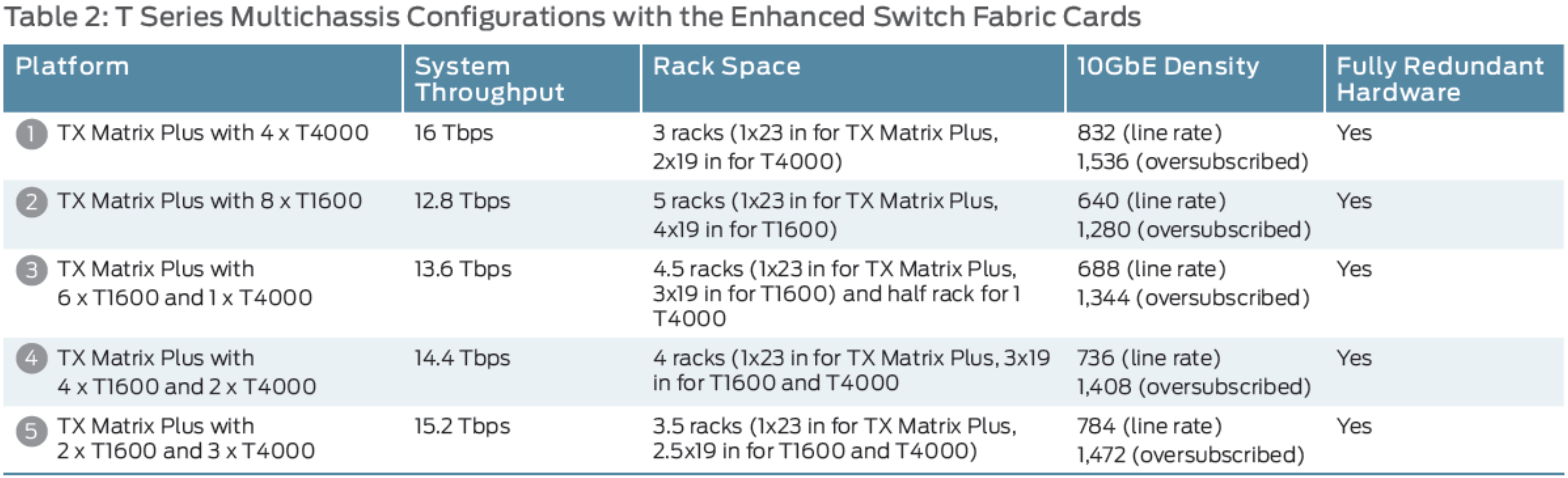
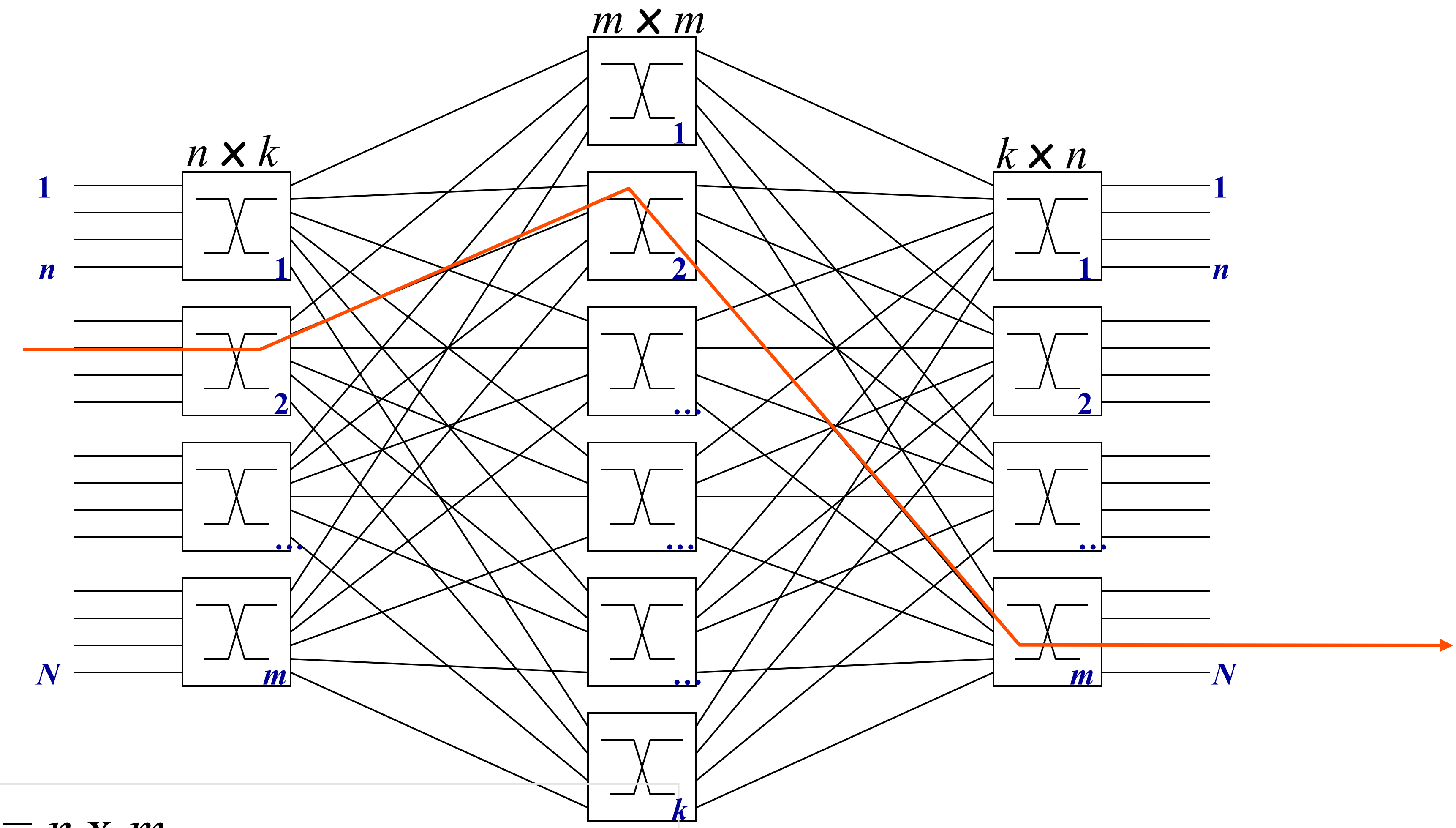


Table 2: T Series Multichassis Configurations with the Enhanced Switch Fabric Cards

Platform	System Throughput	Rack Space	10GbE Density	Fully Redundant Hardware
① TX Matrix Plus with 4 x T4000	16 Tbps	3 racks (1x23 in for TX Matrix Plus, 2x19 in for T4000)	832 (line rate) 1,536 (oversubscribed)	Yes
② TX Matrix Plus with 8 x T1600	12.8 Tbps	5 racks (1x23 in for TX Matrix Plus, 4x19 in for T1600)	640 (line rate) 1,280 (oversubscribed)	Yes
③ TX Matrix Plus with 6 x T1600 and 1 x T4000	13.6 Tbps	4.5 racks (1x23 in for TX Matrix Plus, 3x19 in for T1600) and half rack for 1 T4000	688 (line rate) 1,344 (oversubscribed)	Yes
④ TX Matrix Plus with 4 x T1600 and 2 x T4000	14.4 Tbps	4 racks (1x23 in for TX Matrix Plus, 3x19 in for T1600 and T4000	736 (line rate) 1,408 (oversubscribed)	Yes
⑤ TX Matrix Plus with 2 x T1600 and 3 x T4000	15.2 Tbps	3.5 racks (1x23 in for TX Matrix Plus, 2.5x19 in for T1600 and T4000)	784 (line rate) 1,472 (oversubscribed)	Yes

3-stage Clos Network

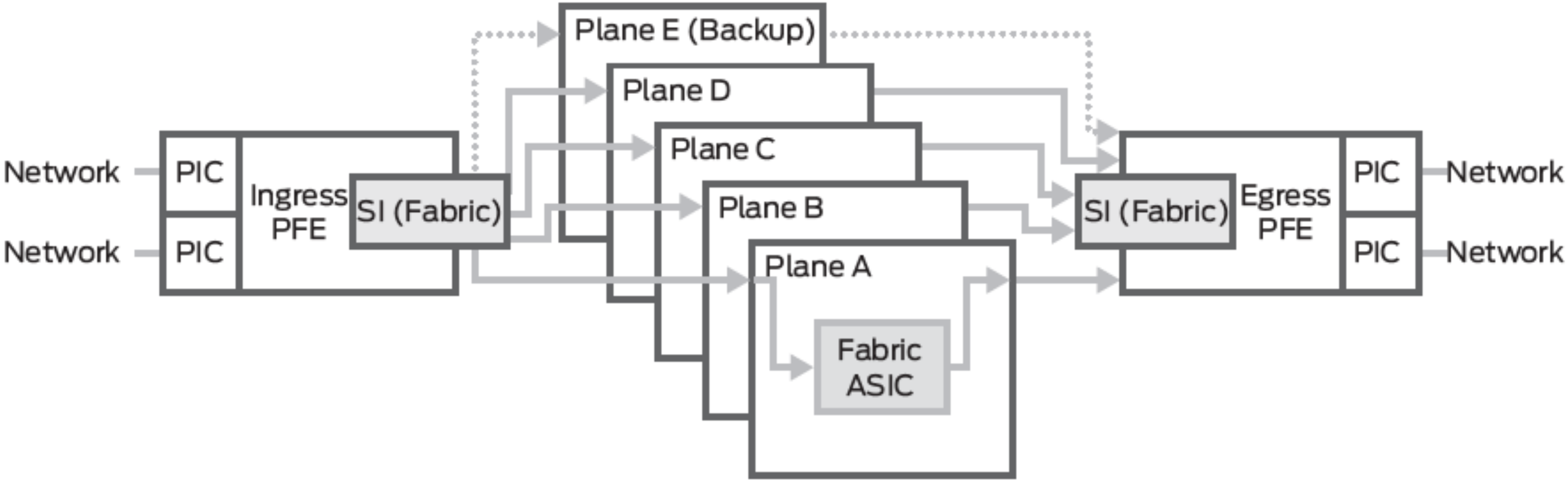


$N = n \text{ x } m$

$k >= n$

$k = 16, n = 16, m = 4,$

5 Switching Planes for a Juniper T-series Single Chassis Core Router

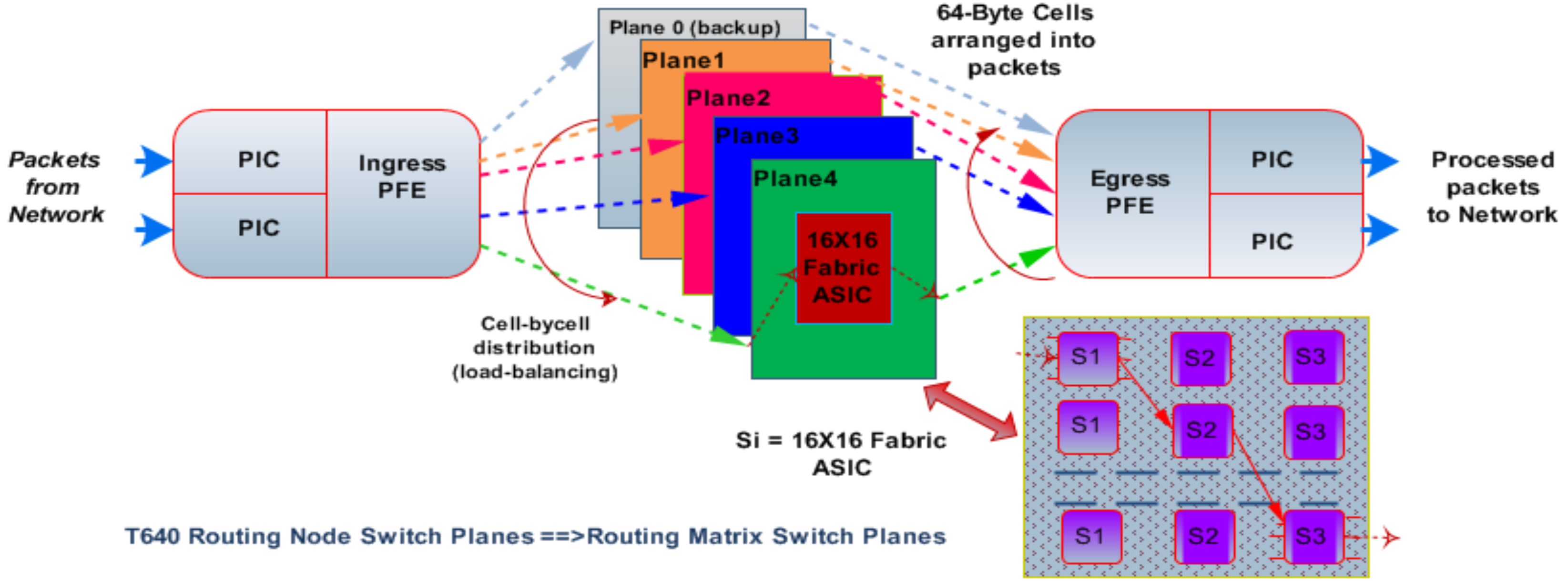


- Note the Distributed Shared Memory (DSM) Architecture with Buffering in the Line Cards (PIC/PFE)

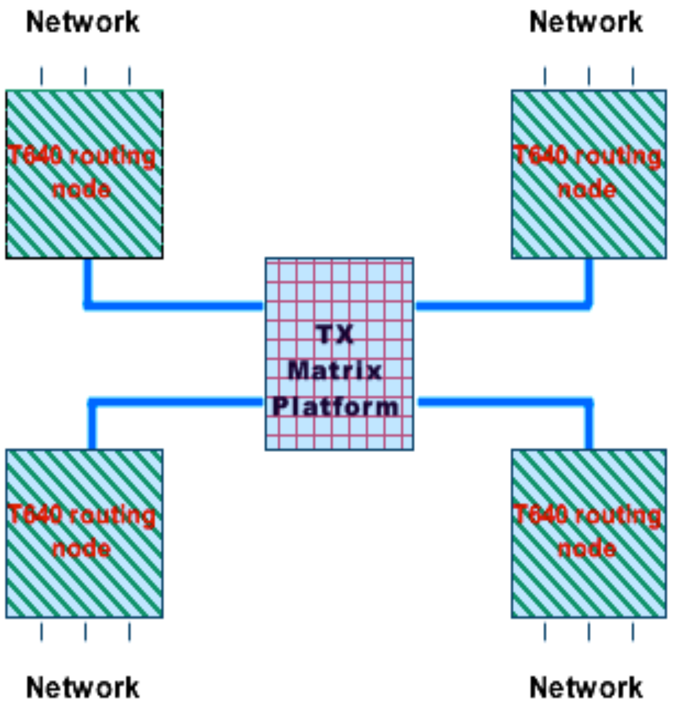
Source: “T-series Core Router Architecture Overview,” Juniper Networks white paper, 2012.

Switch Fabric Implementations

- Maintains data plane connectivity among all of the PFEs.
- Four operationally independent, identical and active switch planes.
- The fifth plane that acting as a hot spare to provide redundancy.


Packets from Network
PIC
PIC
Ingress PFE
Plane 0 (backup)
Plane1
Plane2
Plane3
Plane4
16X16 Fabric ASIC
64-Byte Cells arranged into packets
Egress PFE
PIC
PIC
Processed packets to Network
Cell-bycell distribution (load-balancing)
Si = 16X16 Fabric ASIC
S1
S2
S3
S1
S2
S3
S1
S2
S3
T640 Routing Node Switch Planes ==>Routing Matrix Switch Planes

3-stage Clos Network Realization of the Juniper T-series Multi-chassis Core Routers



The Routing Matrix

TX Matrix (Plus) Platform

(Maximum of 4 T4000 Routing Nodes)

TX Matrix (Plus):

-Performs Routing Functions

-Stage 2 of CLOS Switch Fabric (F2-stage)

-Single Management Interface

-16 4x4 (5-plane) Switching Fabrics in TX routing matrix

T Routing Nodes:

-Distributed Packet Forwarding

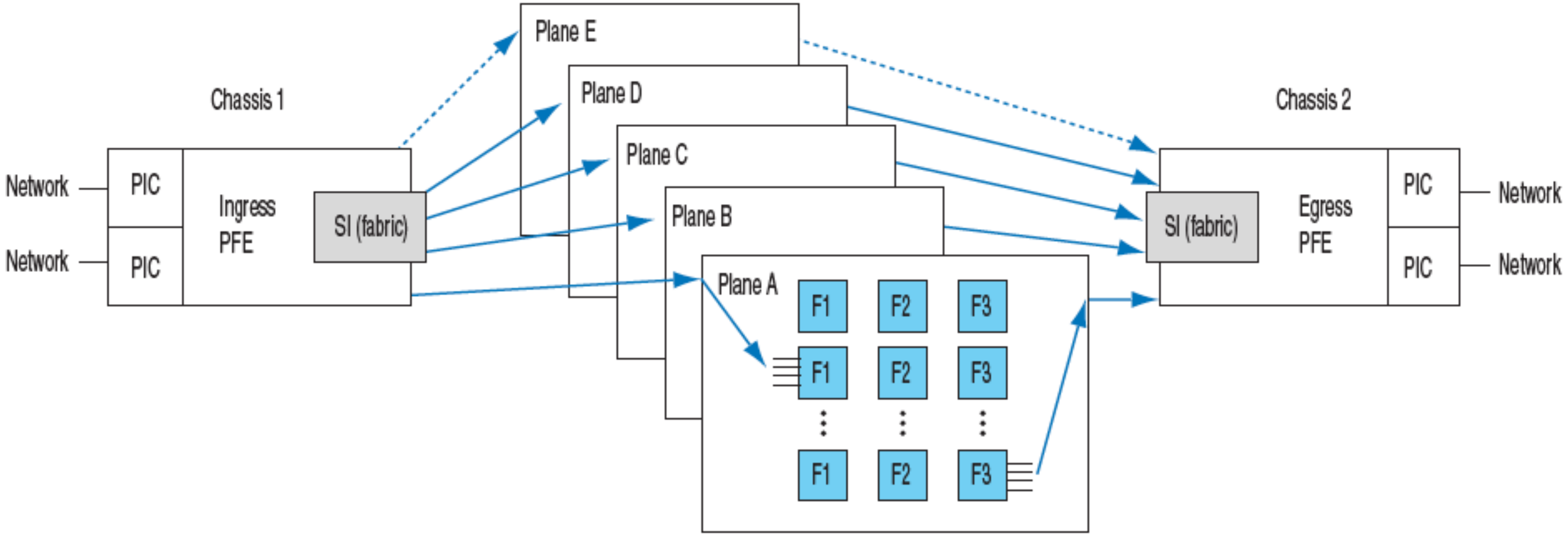
-Stages 1 & 3 of CLOS Fabric (F1&F3-stage)

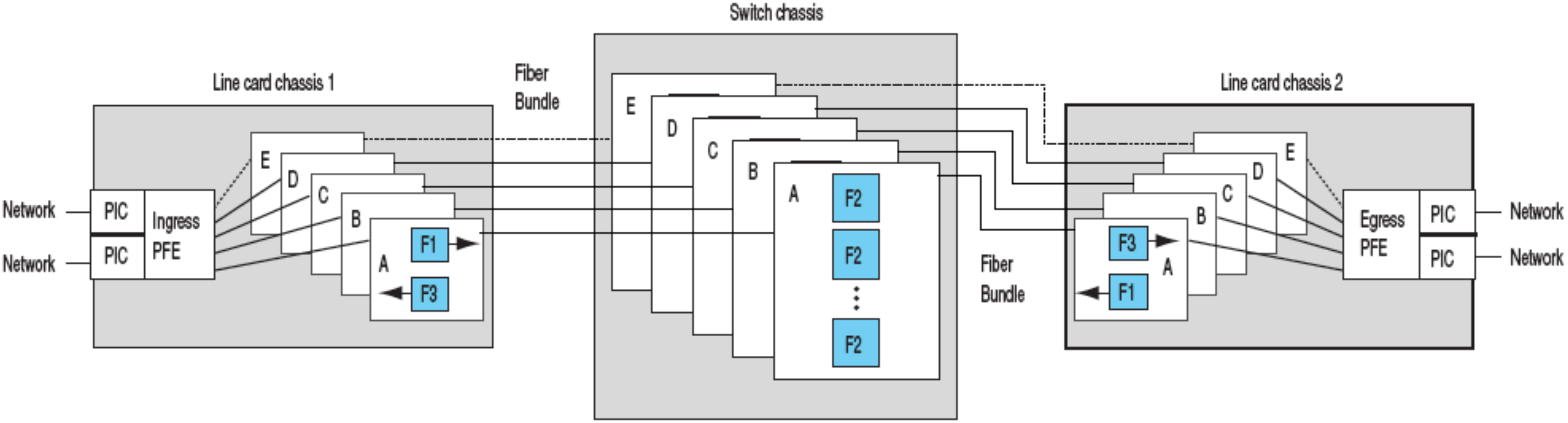
-REs: local chassis management

-16 PFEs in T640 routing node

=> 16 x16 (5-plane) Fabrics [single-chassis]

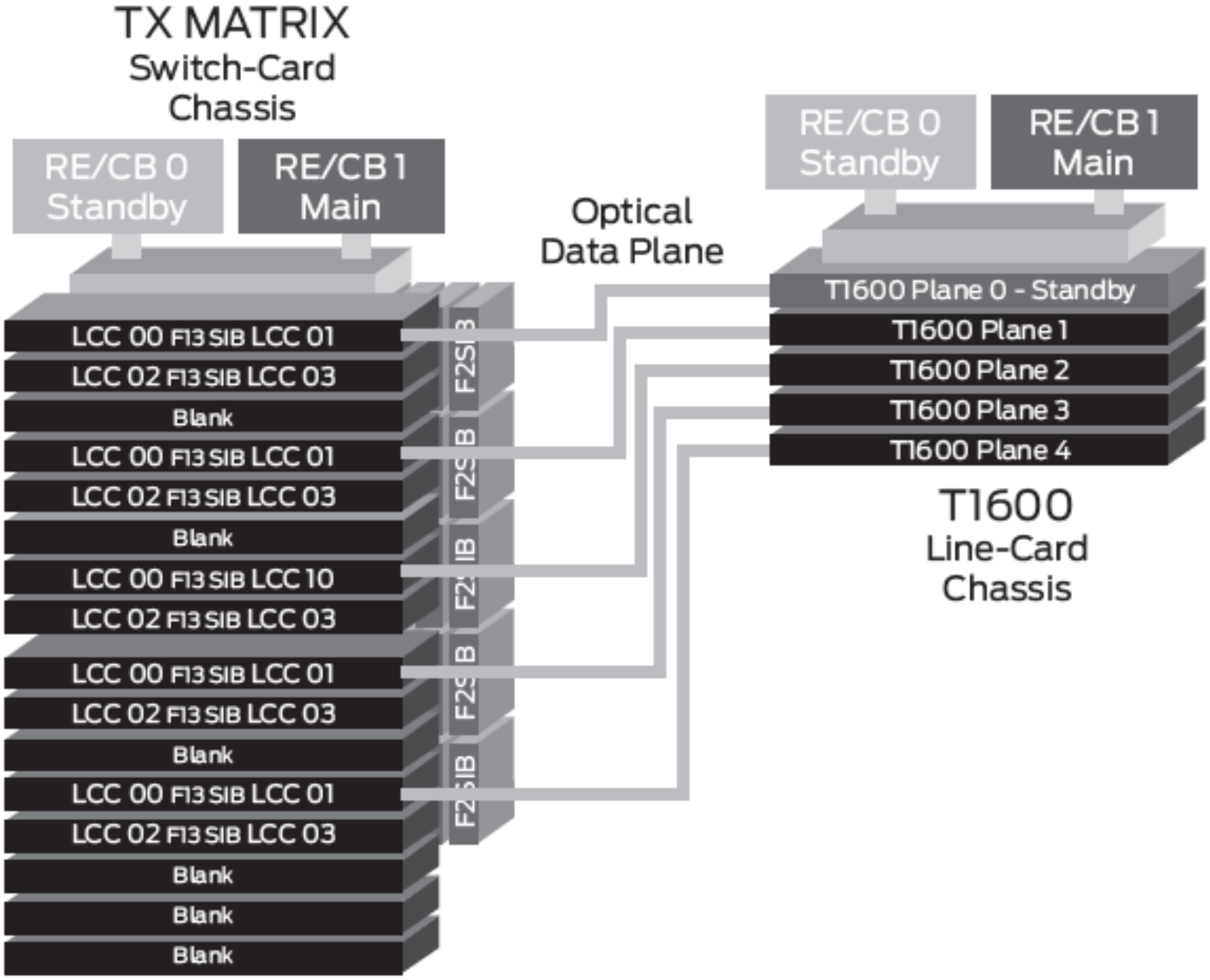
Multi-chassis Realization of 3-stage Clos Network in Juniper T-series Routers





Source: “T-series Routing Platforms: System and Packet Forwarding Architecture,” Juniper Networks white paper, 2002.

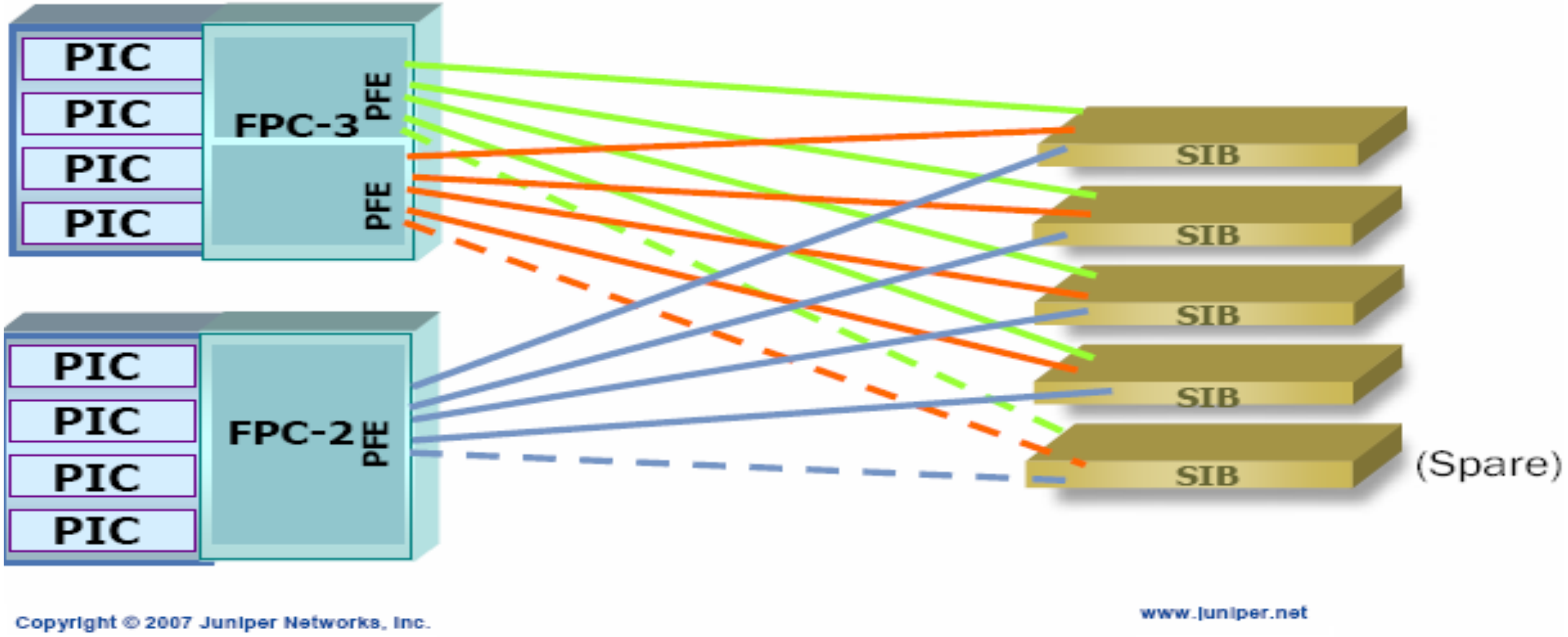
Switch-Card Chassis to Line-Card Chassis Interconnection

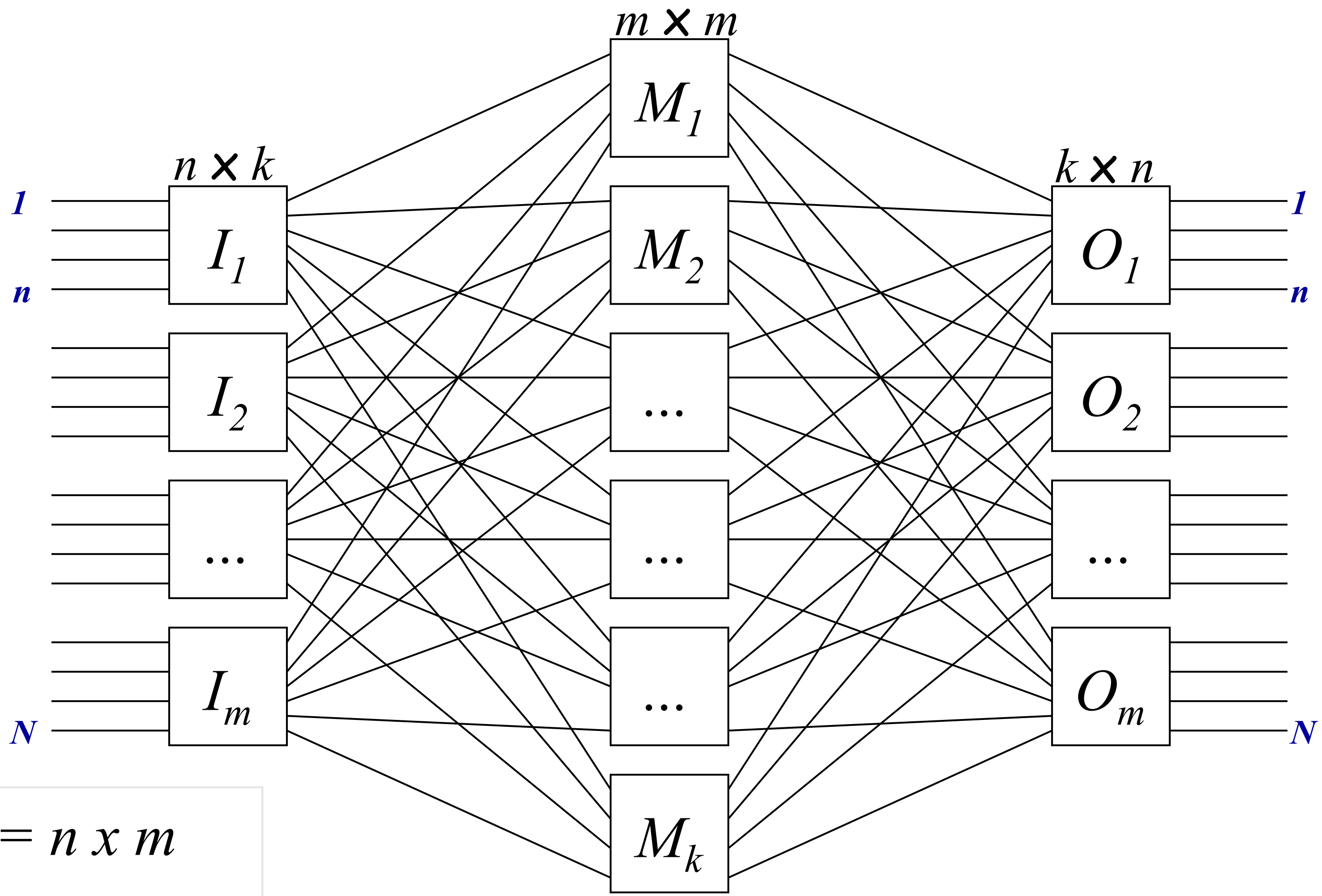


Source: “T-series Routing Platforms: System and Packet Forwarding Architecture,” Juniper Networks white paper, 2002.

TX-SIB, T640 Node and Clos Switch Fabric

- The TX Matrix platform contains five SIB cards connected to the T640-SIB cards in each T640 routing node by way of inter-chassis fiber-optic array cables.
- Each TX Matrix SIB provides connectivity between the ingress and egress T640 routing nodes delivering high performance switching capacity.
- Each T640 routing node is connected to the TX Matrix platform by a five fiber-optic array cable set [VCSEL].
- A fully populated routing matrix requires a total of 20 VCSEL fibers for switch plane interconnect (four T640 => 20 cables total).

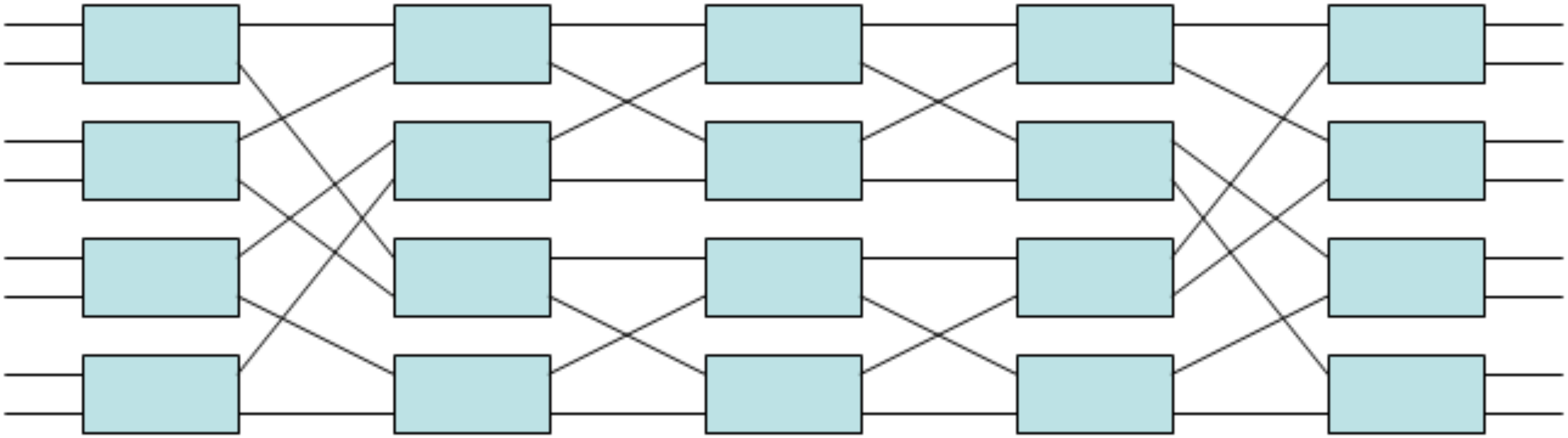




$N = n \ x \ m$
$k >= n$

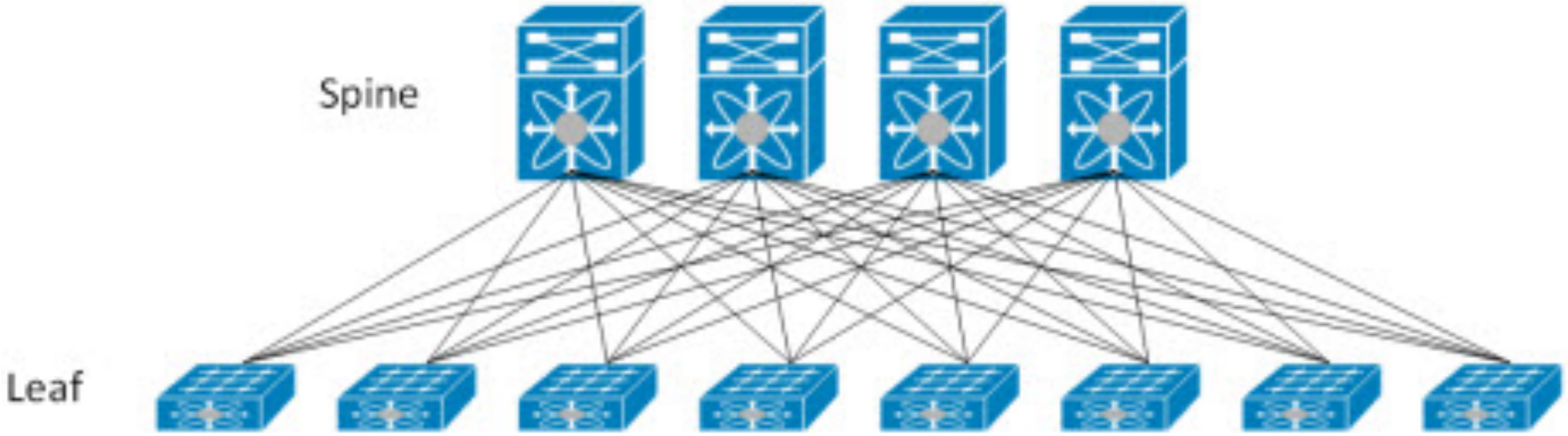
Note: The concept of Clos Network can be generalized to more than 3-stage by replacing each 2nd-stage m x m cross-bar with another 3-stage Clos Network and continue recursively ; e.g. Benes Network (k=n=2) used by 4th Generation Commercial (Cisco) Routers.

A Sample Benes Network



Note: The concept of Clos Network can be generalized to more than 3-stage by replacing each 2nd-stage m x m cross-bar with another 3-stage Clos Network and continue recursively ; e.g. Benes Network (k=n=2) used by 4th Generation Commercial (Cisco CRS-X) Routers.

Clos Networks’ Reappearance in Data Center Networks (aka the Spine and Leaf Topology, or Folded Clos, or Fat-Trees)



The Top of Rack (ToR) switches are the Leaf switches

Each ToR is connected to multiple Core switches which represent the Spine.

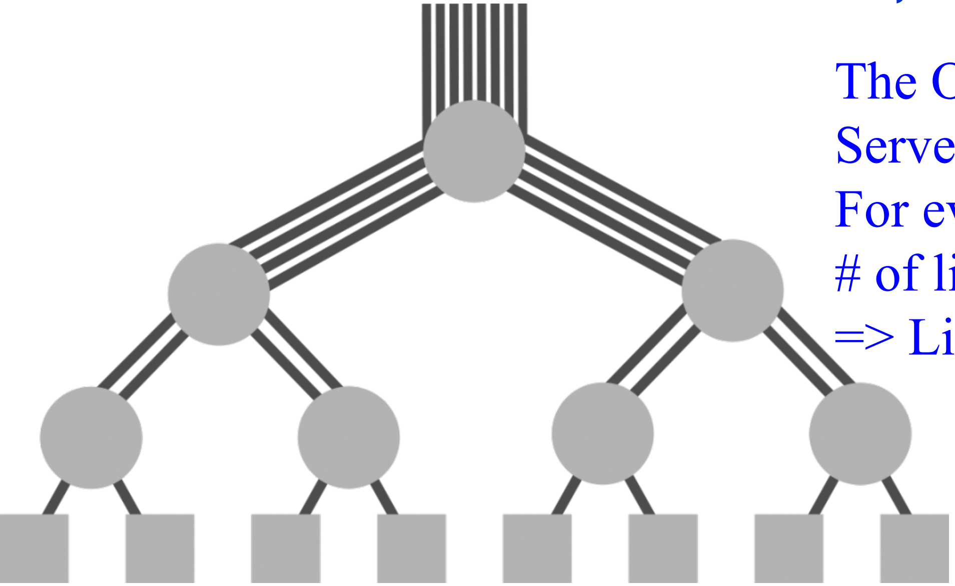
# of Uplinks (of each ToR) = # of Spine switches

# of Downlinks (of each Spine switch) = # of Leaf switches

Multiple ECMP exists for every pair of Leaf switches

Support Incrementally “Scale-out” by adding more Leaf and Spine switches

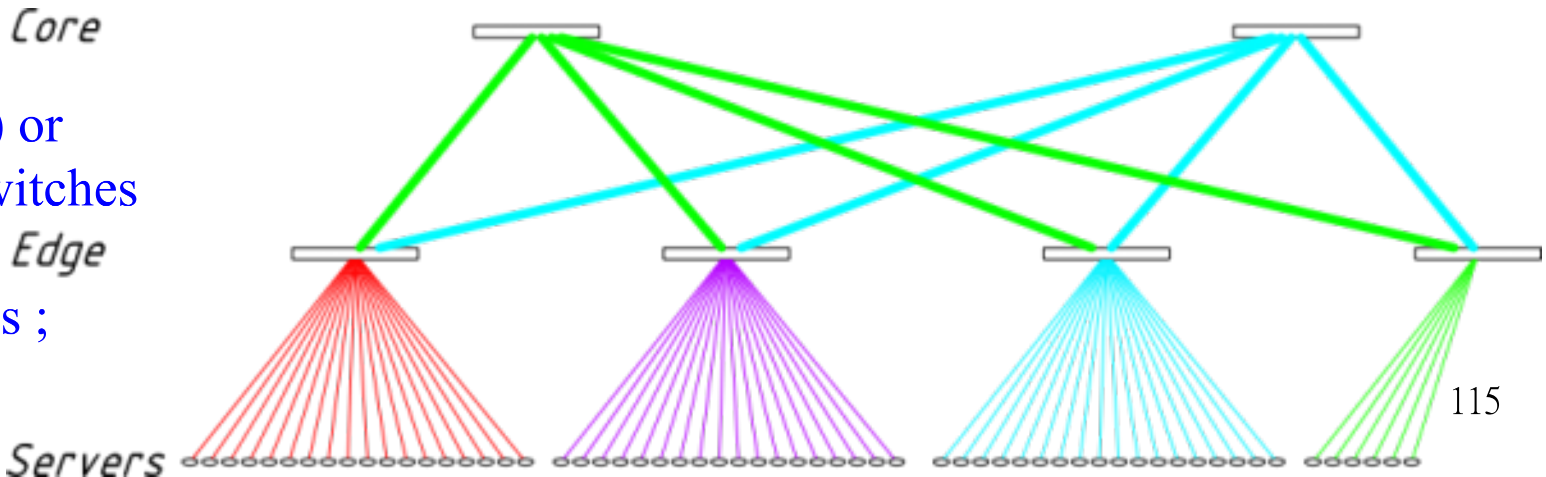
Clos Networks' Reappearance in Data Center Networks (aka the Spine and Leaf Topology, or Folded Clos, or Fat-Trees)



The Original Fat-Tree Topology [Leiserson 85]:
Servers (Processors) are the leafs ;
For every non-leaf node (Switch) in the tree,
# of links to its Parent = # of links to its Children
=> Links at "Fatter" towards the top of the tree

Source: http://clusterdesign.org/fat-trees/

Example:
All Leaf (Edge) or
Spine (Core) switches
are identical
36-port switches ;



Scaling Crossbars: Summary

- Scaling capacity through parallelism (bit-slicing and time-slicing) is straightforward.
- Scaling number of ports is harder…
- Clos network:
  - Rearrangeably non-blocking with $k = n$, but routing is complicated,
  - Strictly non-blocking with $k >= 2n - 1$, so routing of circuit is simple BUT requires more bisection bandwidth.
    - become a moot-point with packet-by-packet routing/switching.

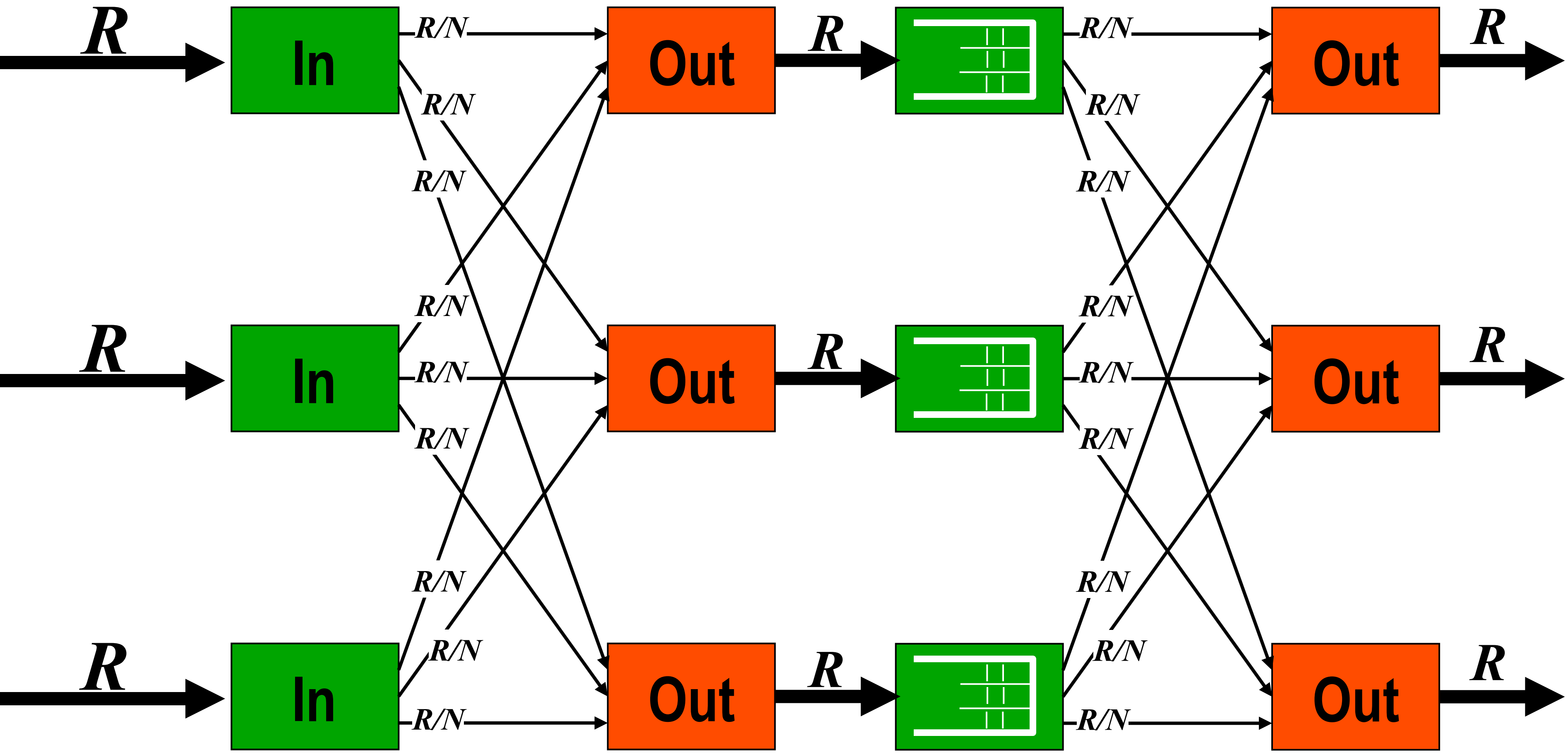
Summary - Routers which give 100% throughput

	Fabric	# Mem.	Mem. BW	Total Memory BW	Switch BW	Scheduling Algorithm
Output-Queued	Bus	N	(N+1)R	N(N+1)R	NR	None
Shared Mem.	Bus	1	2NR	2NR	2NR	None
Input Queued	Crossbar	N	2R	2NR	NR	MWM
CIOQ (Cisco)	**Crossbar**	2N	3R	6NR	2NR	Maximal
		2N	**3R**	**6NR**	**3NR**	**Time Reserve***
PSM	**Bus**	**k**	**3NR/k**	**3NR**	**3NR**	**C. Sets**
DSM (Juniper)	**Xbar**	**N**	**3R**	**3NR**	**4NR**	**Edge Color**
		N	**3R**	**3NR**	**6NR**	**C. Sets**
		N	**4R**	**4NR**	**4NR**	**C. Sets**
PPS - OQ	**Clos**	**Nk**	**2R(N+1)/k**	**2N(N+1)R**	**4NR**	**C. Sets**
PPS –Shared Memory	**Clos**	**Nk**	**4NR/k**	**4NR**	**4NR**	**C. Sets**
		Nk	**2NR/k**	**2NR**	**2NR**	**None**

Summary - Routers with delay guarantees

	Fabric	# Mem.	Mem. BW	Total Memory BW	Switch BW	Scheduling Algorithm
Output-Queued	Bus	N	(N+1)R	N(N+1)R	NR	None
Shared Mem.	Bus	1	2NR	2NR	2NR	None
Input Queued	Crossbar	N	2R	2NR	NR	-
CIOQ (Cisco)	**Crossbar**	2N	3R	6NR	2NR	Marriage
		2N	**3R**	**6NR**	**3NR**	**Time Reserve**
PSM	**Bus**	**k**	**4NR/k**	**4NR**	**4NR**	**C. Sets**
DSM (Juniper)	**Xbar**	**N**	**4R**	**4NR**	**5NR**	**Edge Color**
		N	**4R**	**4NR**	**8NR**	**C. Sets**
		N	**6R**	**6NR**	**6NR**	**C. Sets**
PPS - OQ	**Clos**	**Nk**	**3R(N+1)/k**	**3N(N+1)R**	**6NR**	**C. Sets**
PPS –Shared Memory	**Clos**	**Nk**	**6NR/k**	**6NR**	**6NR**	**C. Sets**
		Nk	**2NR/k**	**2NR**	**2NR**	**-**

Future Directions: Two-stage load-balancing switch

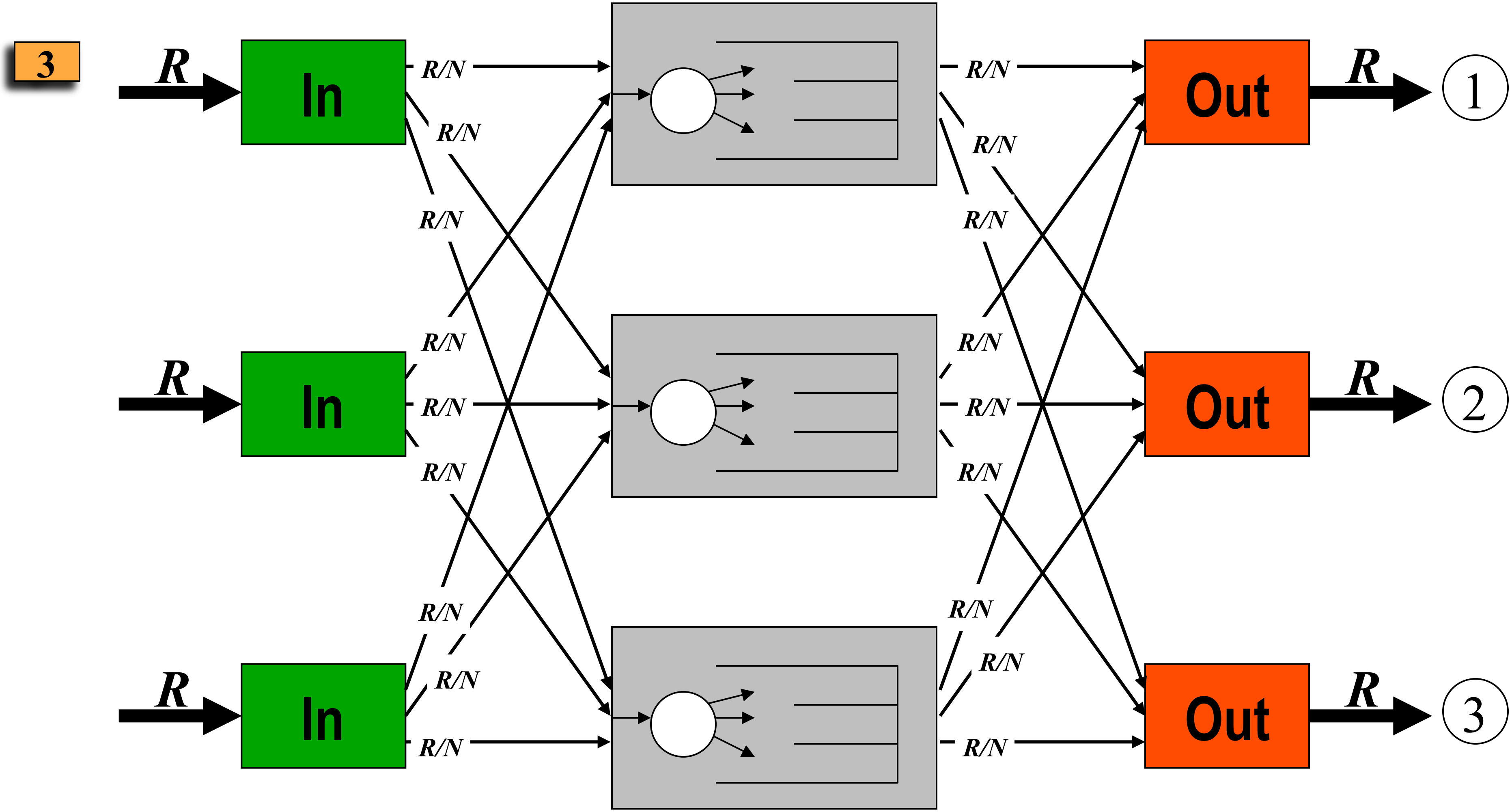


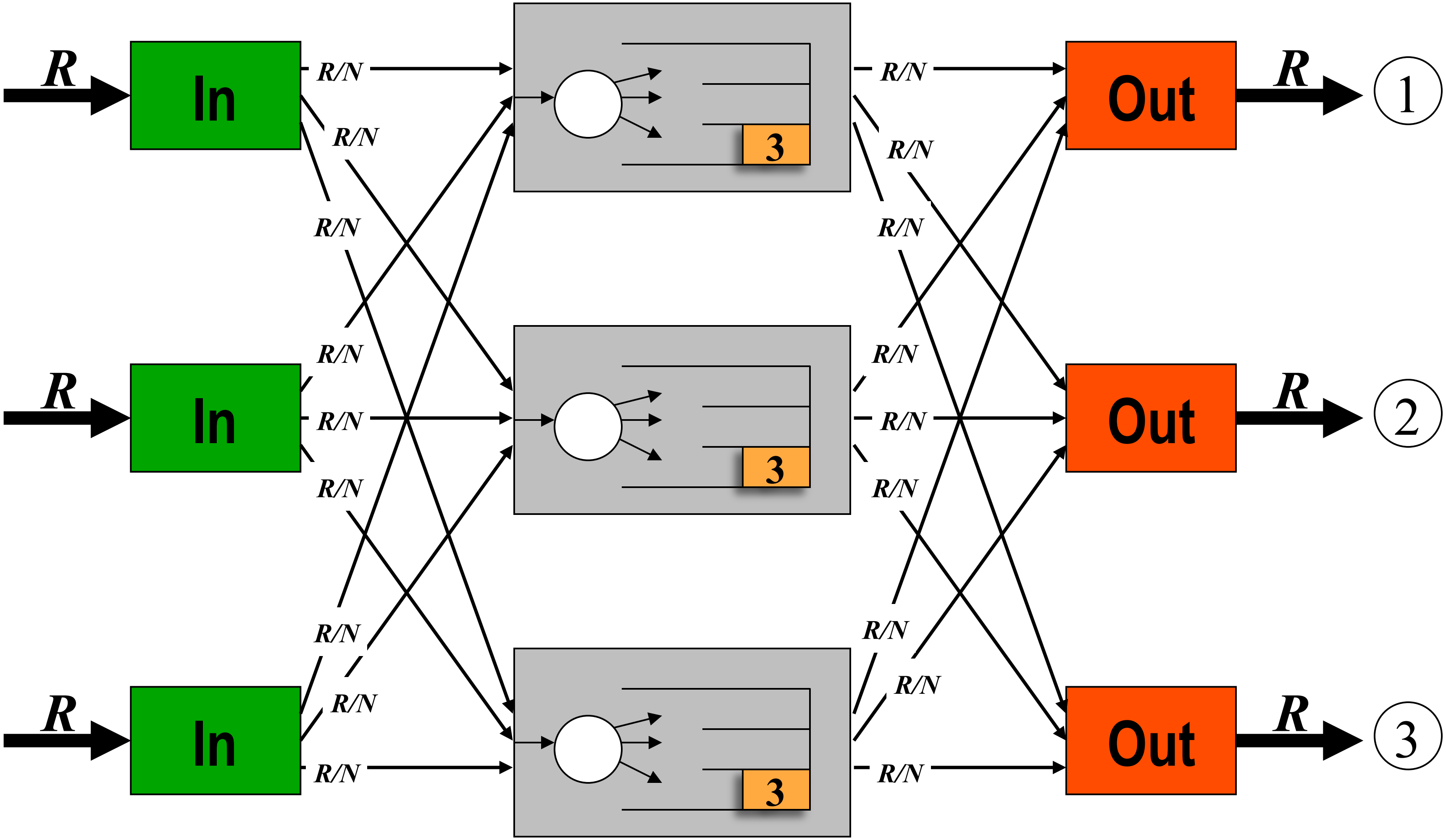
Load-balancing stage

Switching stage

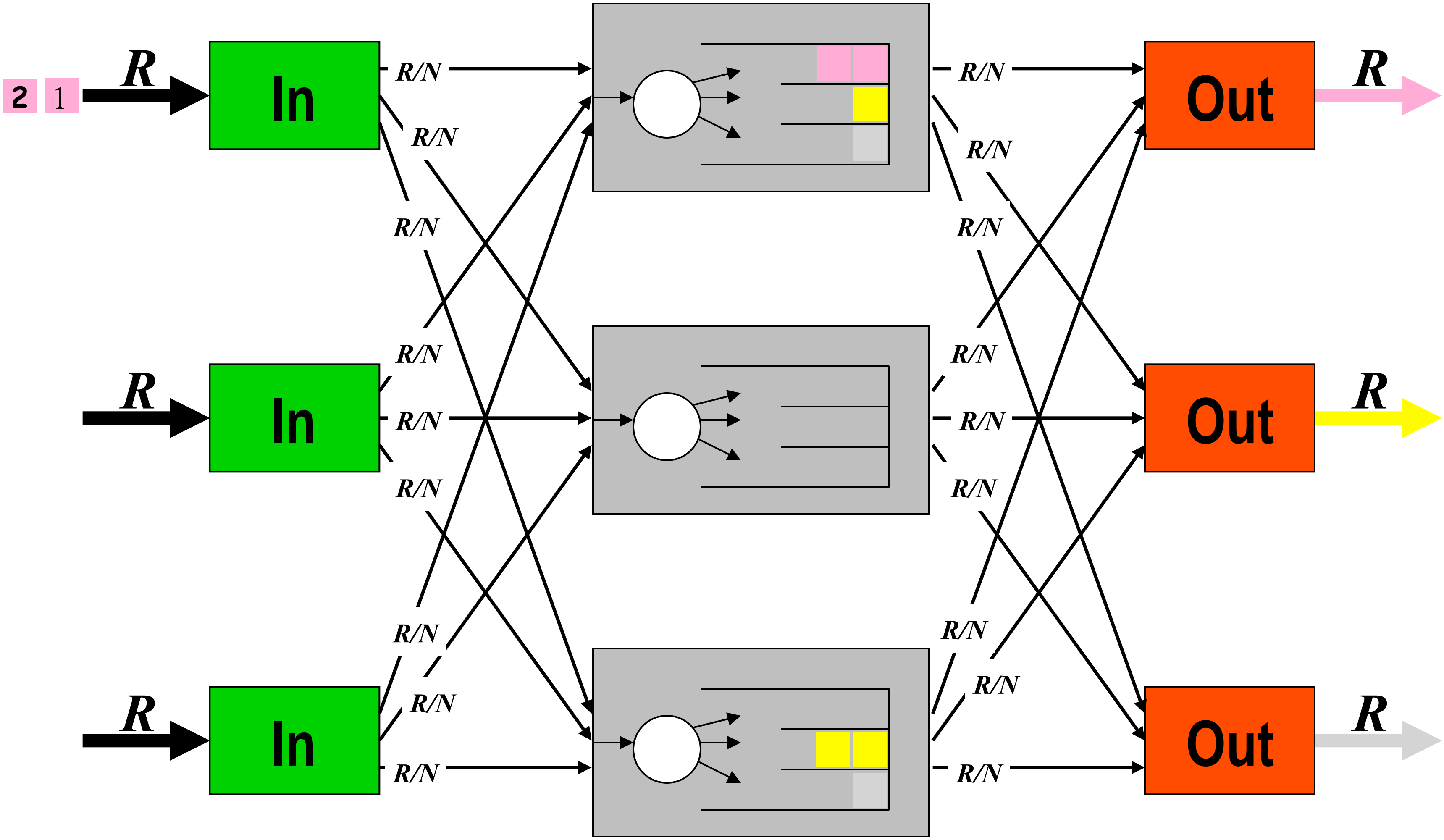
100% throughput for weakly mixing, stochastic traffic.

[C.-S. Chang, Valiant]


3
R
In
R
In
R
In
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
R/N
Out
Out
Out
R
R
R
1
2
3

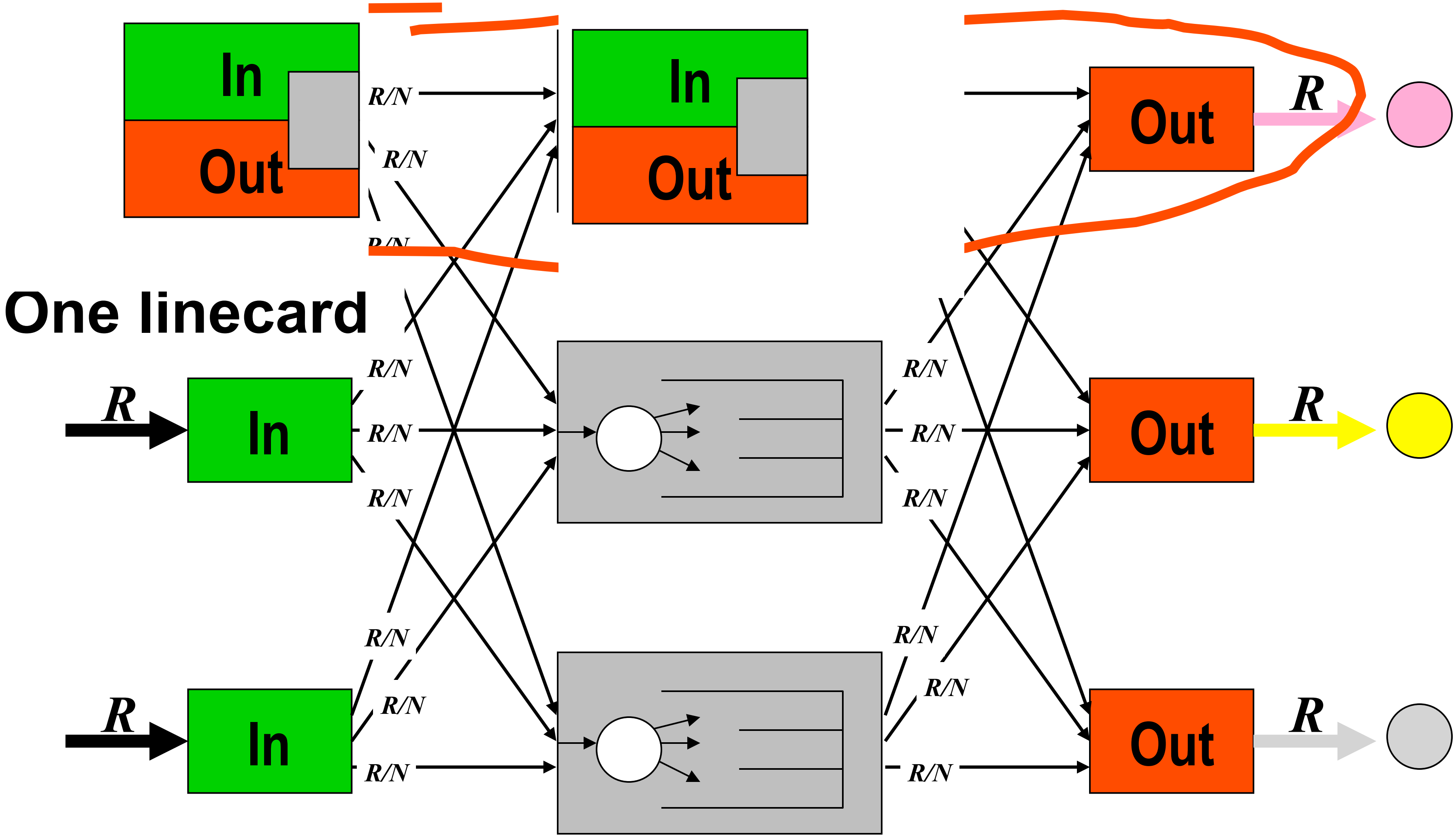

R
R
R
In
In
In
R/N
3
3
3
Out
Out
Out
1
2
3

Packet Reordering Problem



Re-ordering Problem can be fixed by a simple, fully distributed algorithm

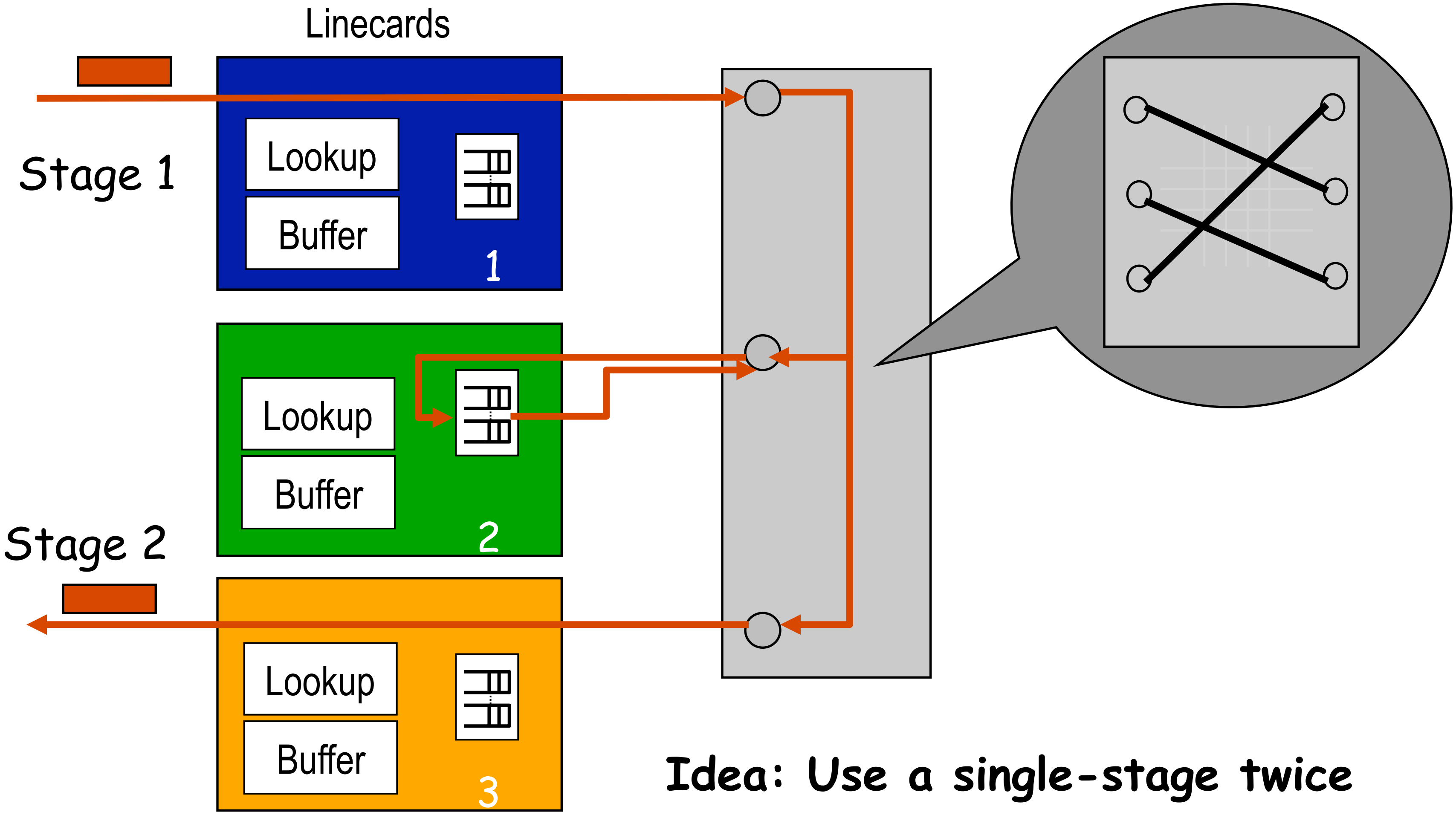
Combining the Two Meshes



Combining the 2 Meshes

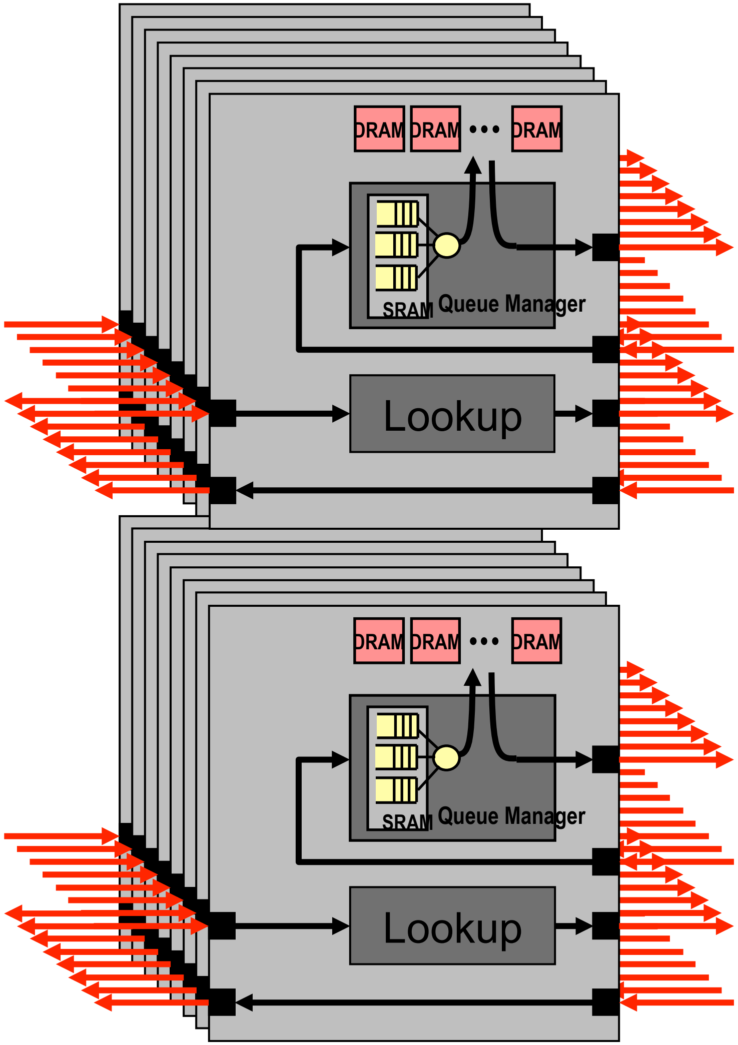
Load-balancing 2-stage Switch

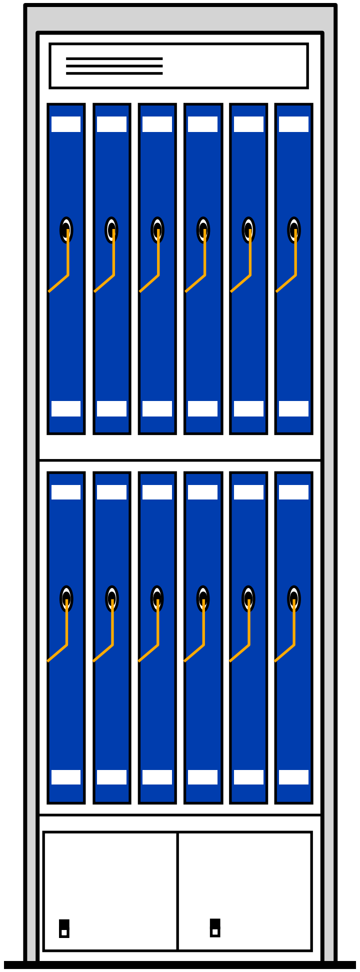
[Isaac Keslassy et al, Sigcomm 2003]



Idea: Use a single-stage twice

Separate Racks for Linecards and Optical Core providing Mesh connectivity





Two ways to Realize Load-balanced Switch via a Single Fixed-Rate Uniform Mesh

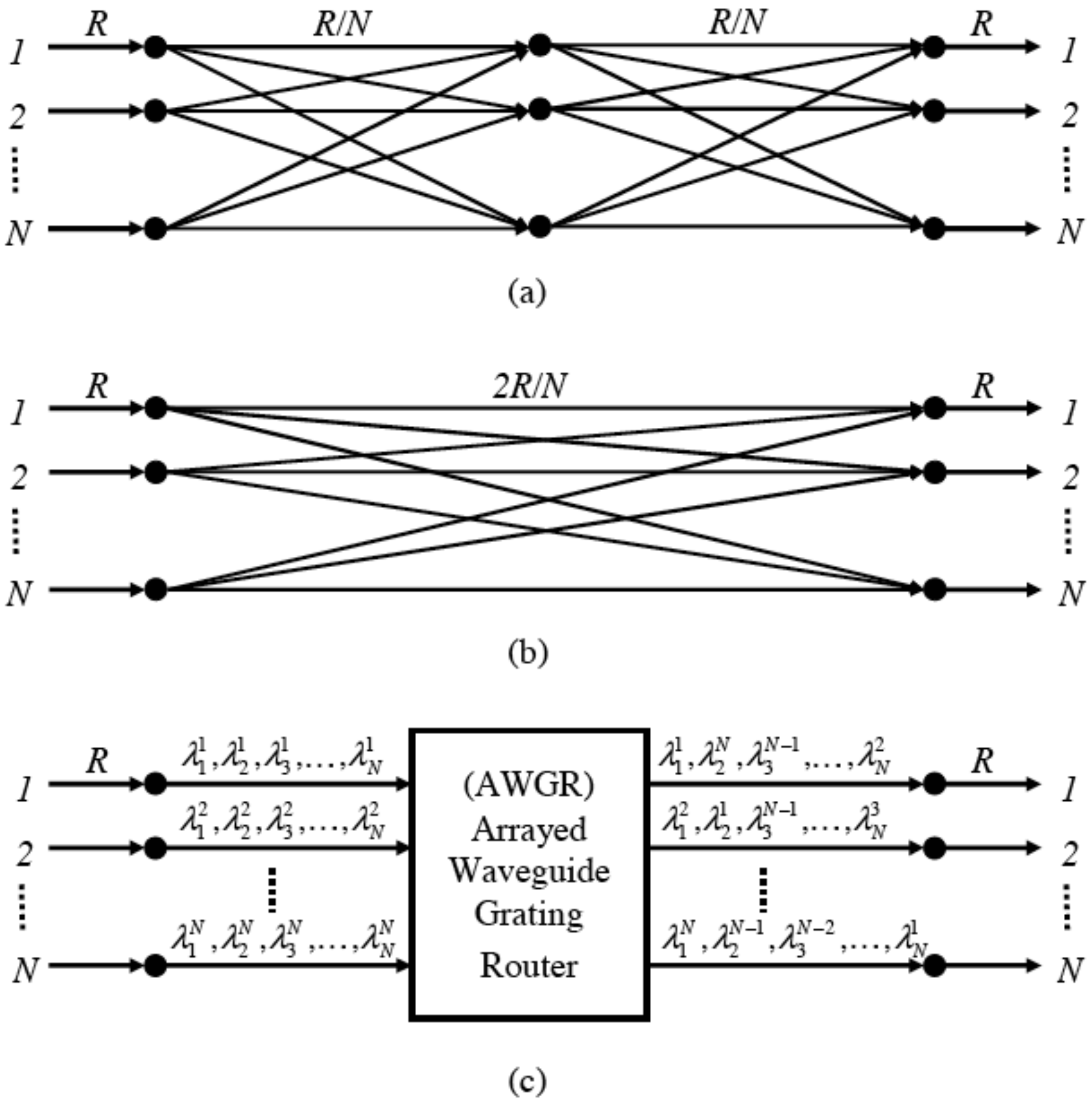
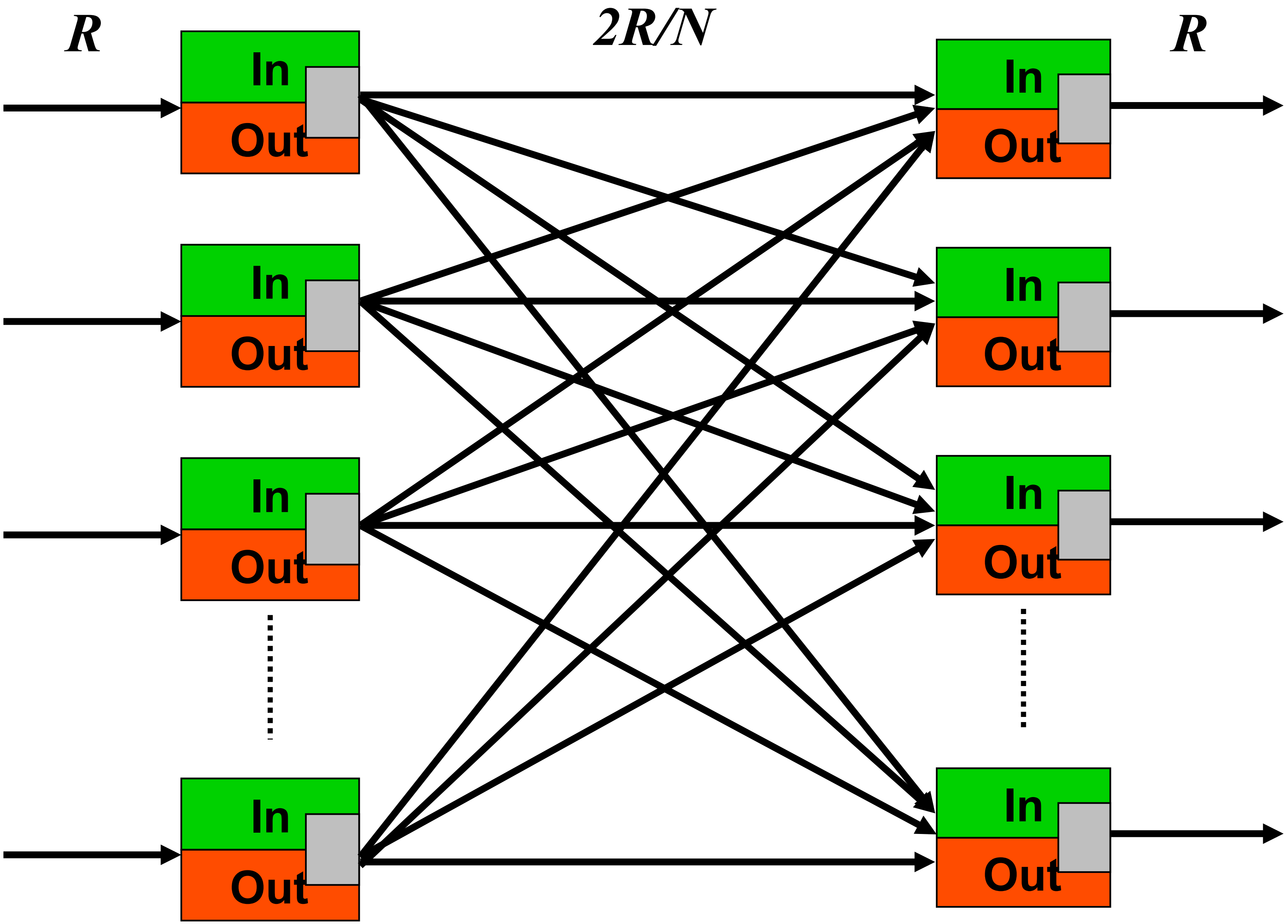
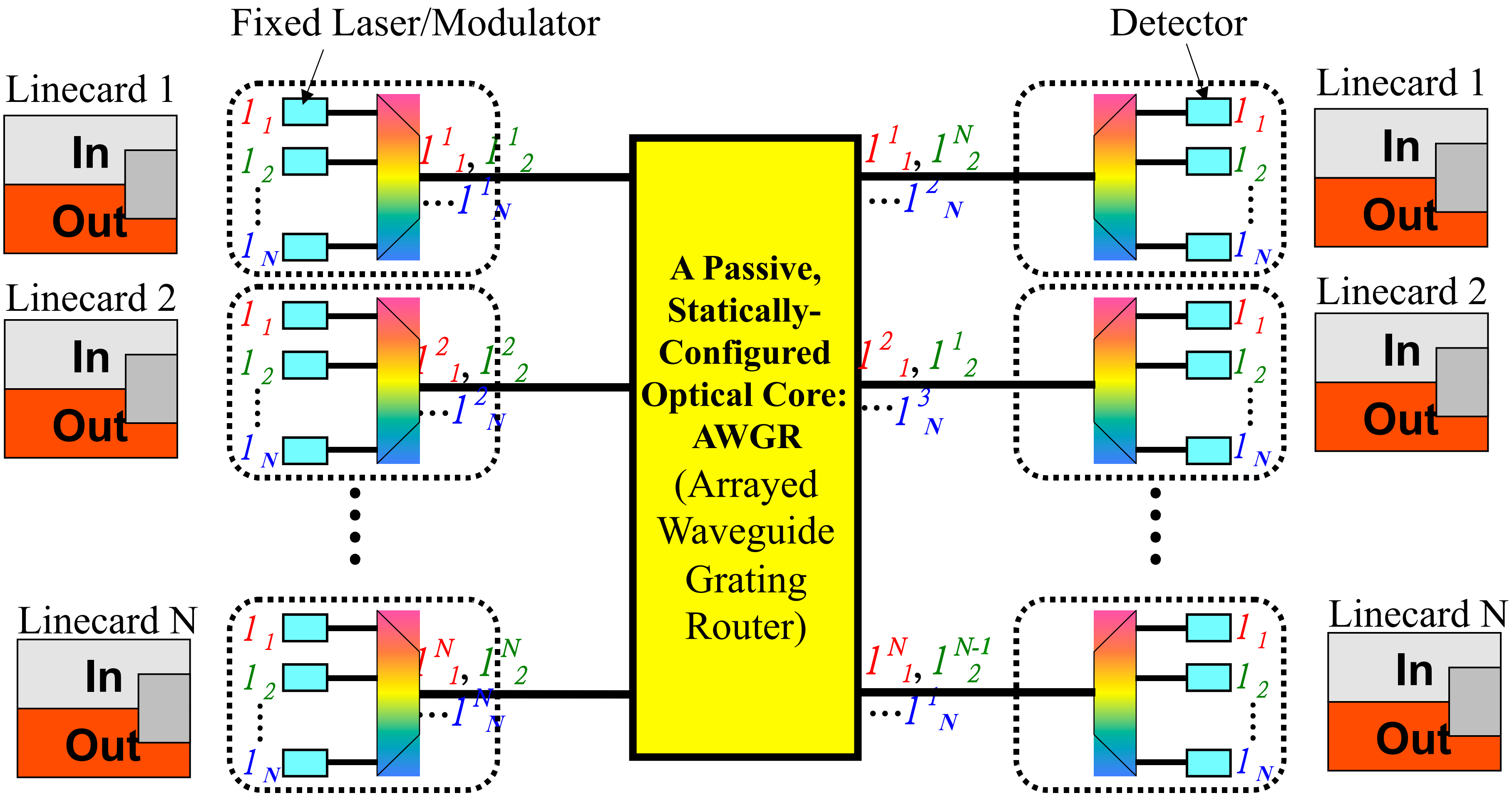

R
R/N
R/N
R
1
2
N
1
2
N
(a)
R
2R/N
R
1
2
N
1
2
N
(b)
R
$\lambda_1^1, \lambda_2^1, \lambda_3^1, \ldots, \lambda_N^1$
$\lambda_1^2, \lambda_2^2, \lambda_3^2, \ldots, \lambda_N^2$
$\lambda_1^N, \lambda_2^N, \lambda_3^N, \ldots, \lambda_N^N$
(AWGR)
Arrayed
Waveguide
Grating
Router
$\lambda_1^1, \lambda_2^N, \lambda_3^{N-1}, \ldots, \lambda_N^2$
$\lambda_1^2, \lambda_2^1, \lambda_3^{N-1}, \ldots, \lambda_N^3$
$\lambda_1^N, \lambda_2^{N-1}, \lambda_3^{N-2}, \ldots, \lambda_N^1$
R
1
2
N
1
2
N
(c)

Figure 5: Two ways in which the load-balanced switch can be implemented by a single fixed-rate uniform mesh. In both cases, two stages operating at rate R/N, as shown in (a), are replaced by one stage operating at $2R/N$, and every packet traverses the mesh twice. In (b), the mesh is implemented by N^2 fibers. In (c), the mesh is N^2 WDM channels interconnected by an AWGR. λ_w^i is transmitted on wavelength λ_w from input i and operates at rate $2R/N$.

A Single Combined Mesh



AWGR: A Mesh of WDM Channels



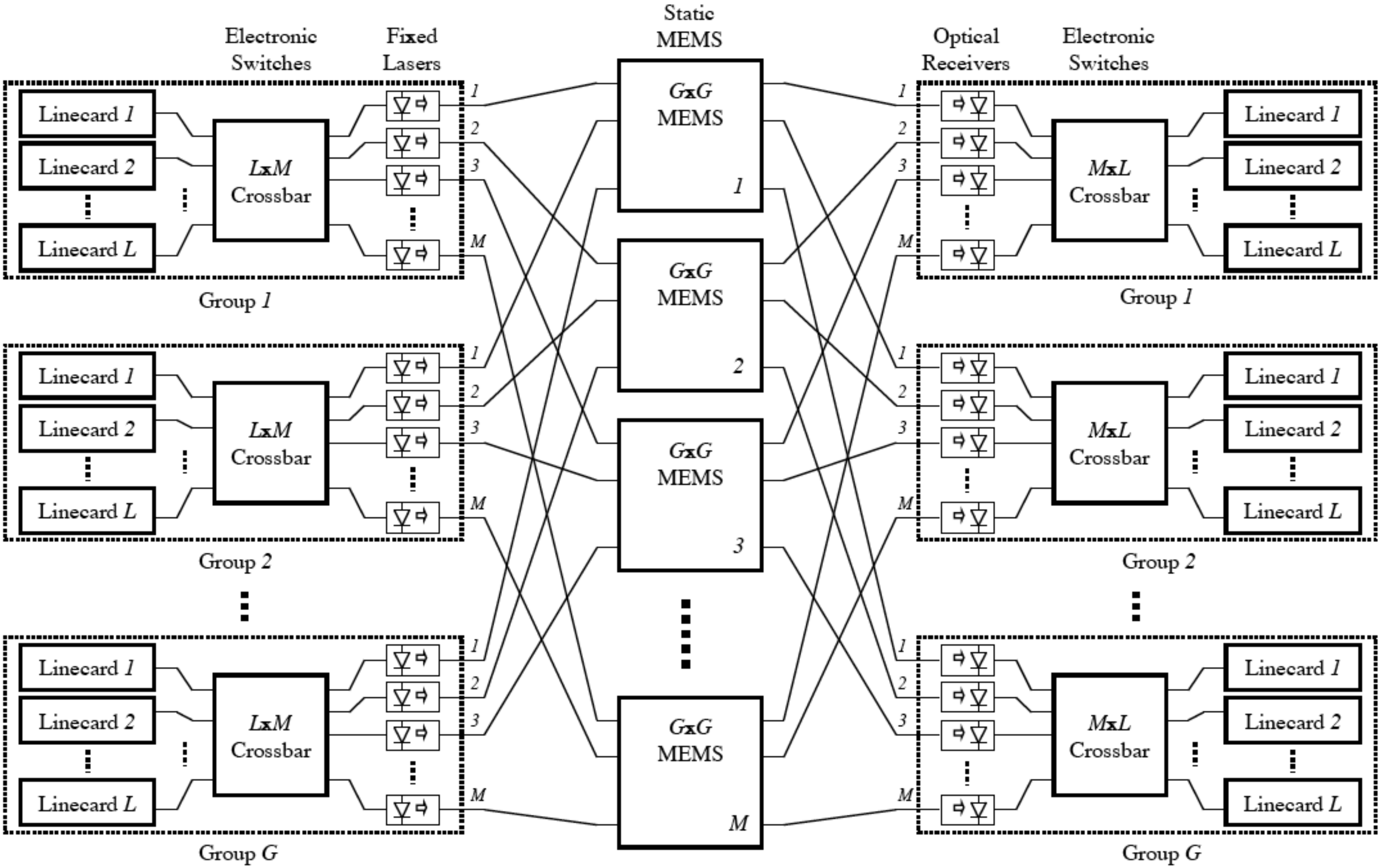
Fixed Laser/Modulator

Detector

A Passive, Statically-Configured Optical Core: AWGR (Arrayed Waveguide Grating Router)

Linecard 1
In
Out
Linecard 2
In
Out
Linecard N
In
Out
l_1
l_2
l_N
l^1_1, l^1_2, …l^1_N
l^2_1, l^2_2, …l^2_N
l^N_1, l^N_2, …l^N_N
l^1_1, l^N_2, …l^2_N
l^2_1, l^1_2, …l^3_N
l^N_1, l^{N-1}_2, …l^1_N
Linecard 1
In
Out
Linecard 2
In
Out
Linecard N
In
Out

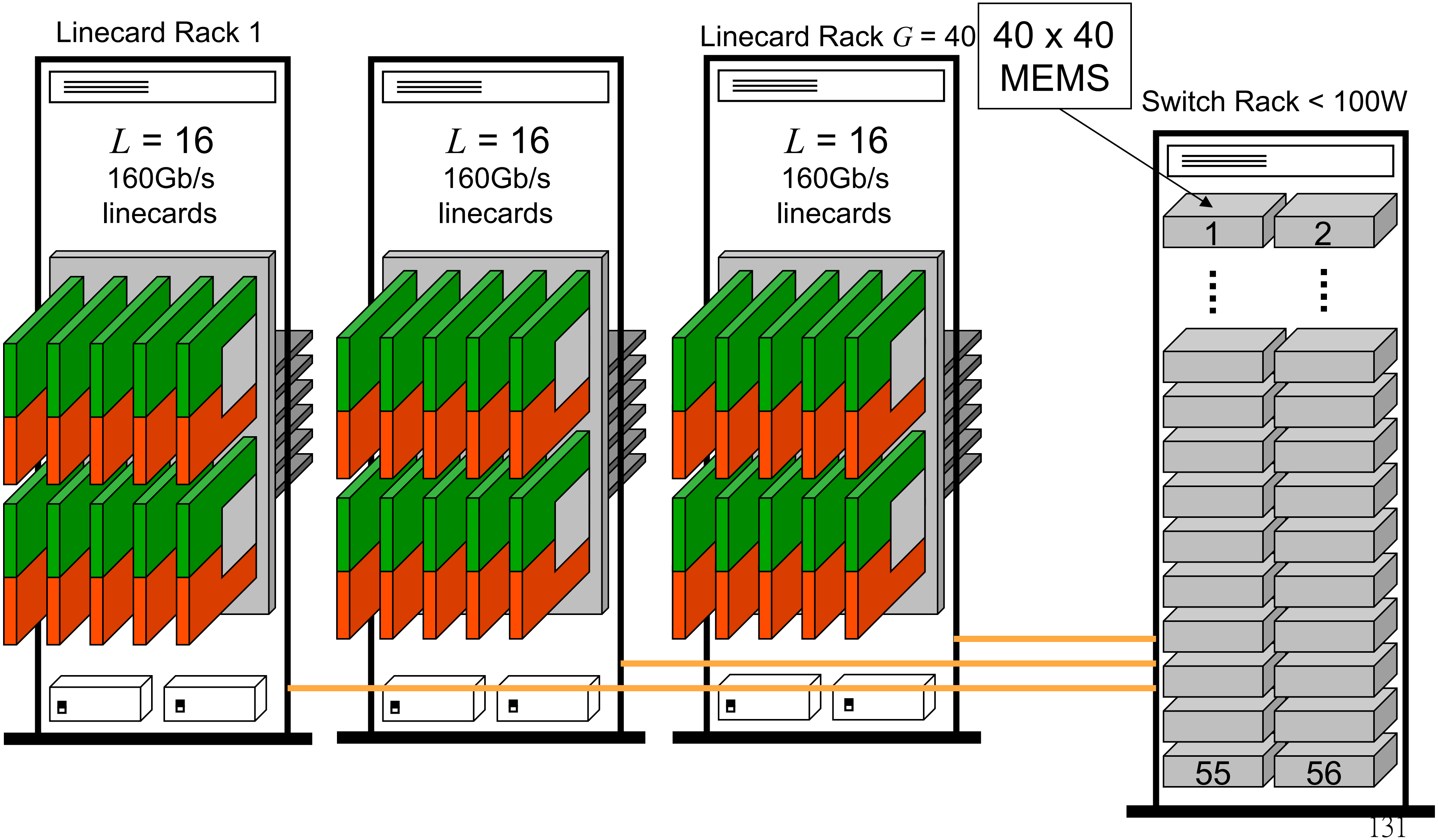
Hybrid Optical and Electrical Switch Fabric



Hybrid Electro-Optical Switch Fabric

- Thm: There is a polynomial time algorithm that finds a static configuration for each MEMS switch, and a fixed-length sequence of permutations for the electronic crossbars to spread packets over the paths.

100Tb/s Load-Balanced Router



A Pure Optical Switch Fabric

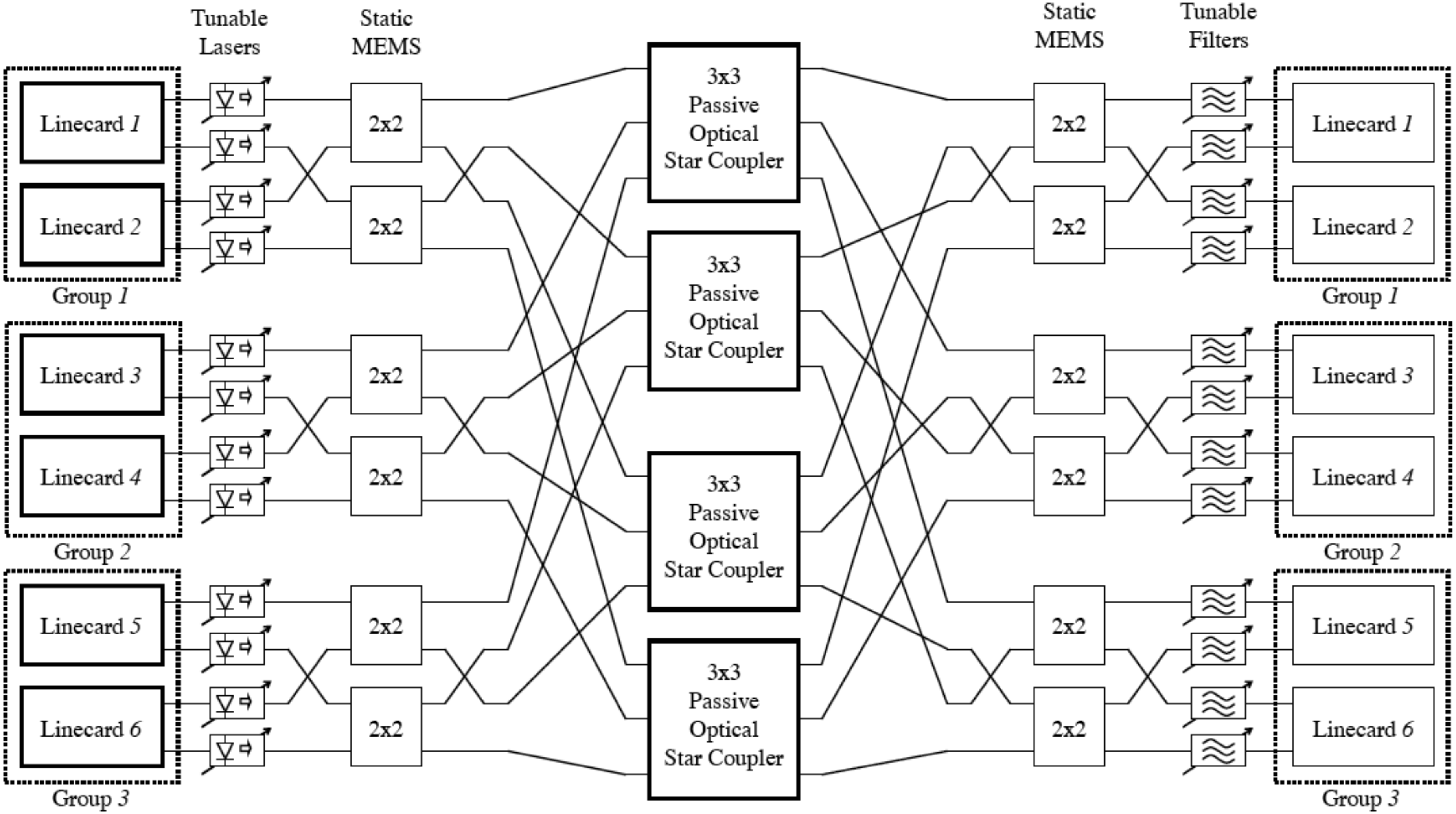
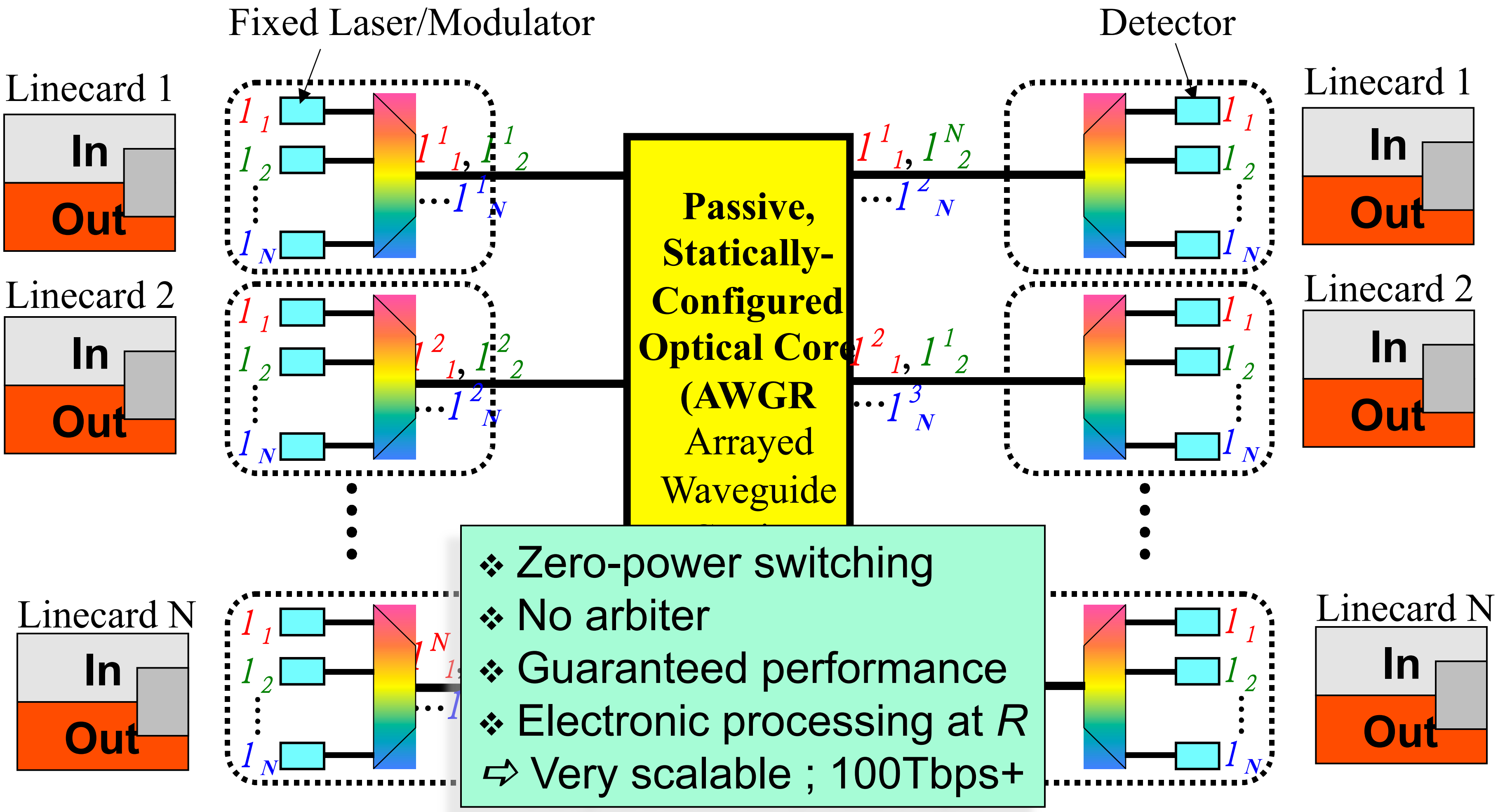


Figure 8: An optical switch fabric for $G = 3$ groups with $L = 2$ linecards per group.

5th Generation routers?

Load-balancing over passive optics (in research stage)



- ❖ Zero-power switching
- ❖ No arbiter
- ❖ Guaranteed performance
- ❖ Electronic processing at *R*

➪ Very scalable ; 100Tbps+

Summary of Router Architecture

- Multi-rack routers
- Single router POPs
- No commercial router provides 100% throughput guarantee.
- Address lookups
  - Not a problem to 400Gb/s per linecard.
- Packet buffering
  - Imperfect; loss of throughput above 10Gb/s.
- Router/ Switch Architecture
  - Centralized schedulers up to about 6.4Tb/s
    - CIOQ: Combined Input Output Queueing Architecture (Cisco)
    - DSM: Distributed Shared Memory Architecture (Juniper)
  - High capacity (Slicing) and High port-count (Multi-stage Clos/Benes Network) Crossbar Fabric can scale the aggregate switch capacity to 100's of Tb/s
  - There are experimental research which uses Load-balanced 2-stage optical switches with 100% throughput
    - with a statically-configured optical core potentially can scale aggregate switch capacity beyond 100's Tb/s.

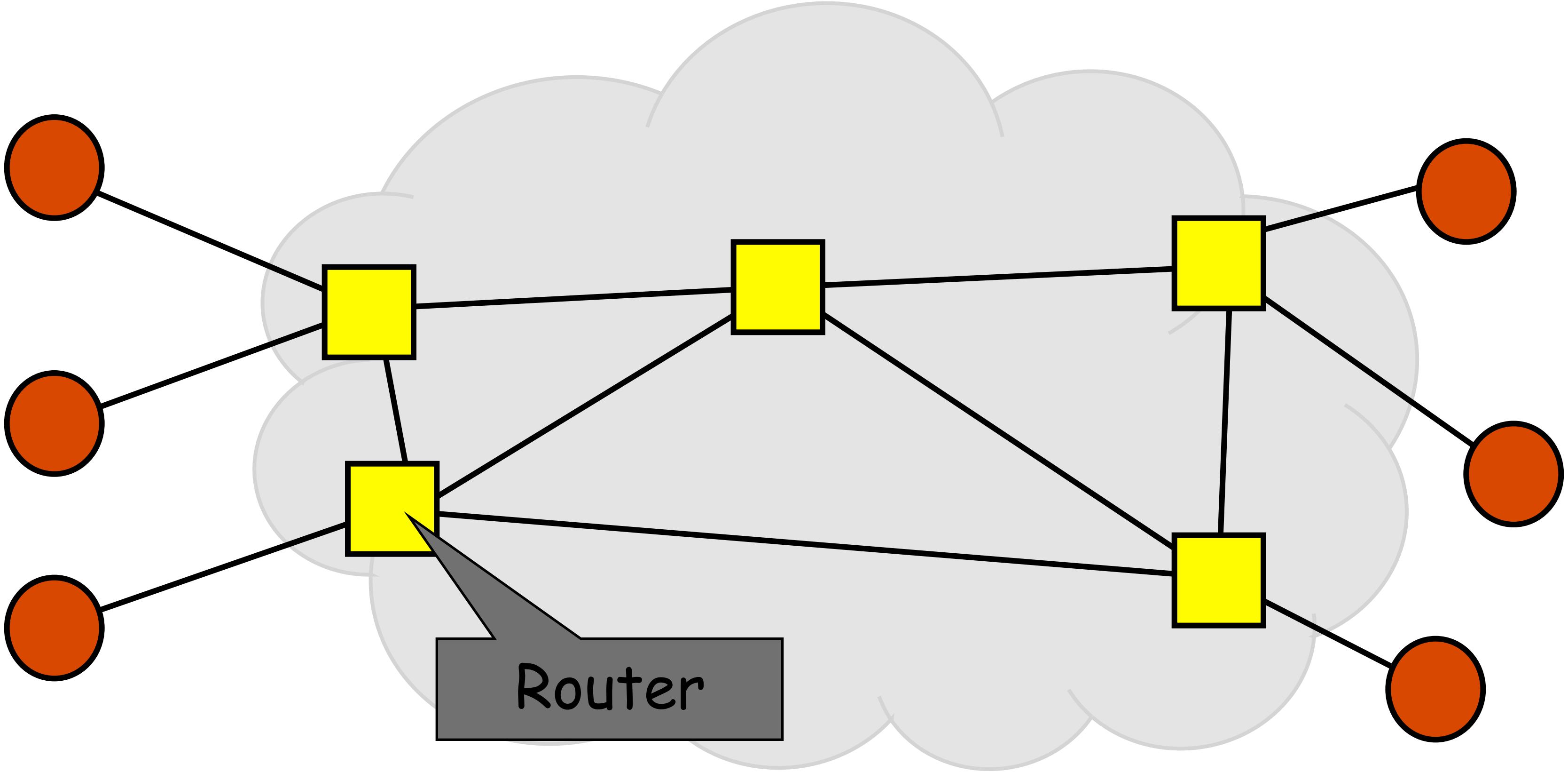
Current Internet Router Technology
Summary

- Techniques exist today for:
  - 100Gbps line-rate IP address lookup
  - 400Gb/s (4x 100GE ports) linecards.
  - 6.4 Tb/s capacity for single-rack (chassis) router
  - 100+Tb/s capacity for multiple-stage architectures, with

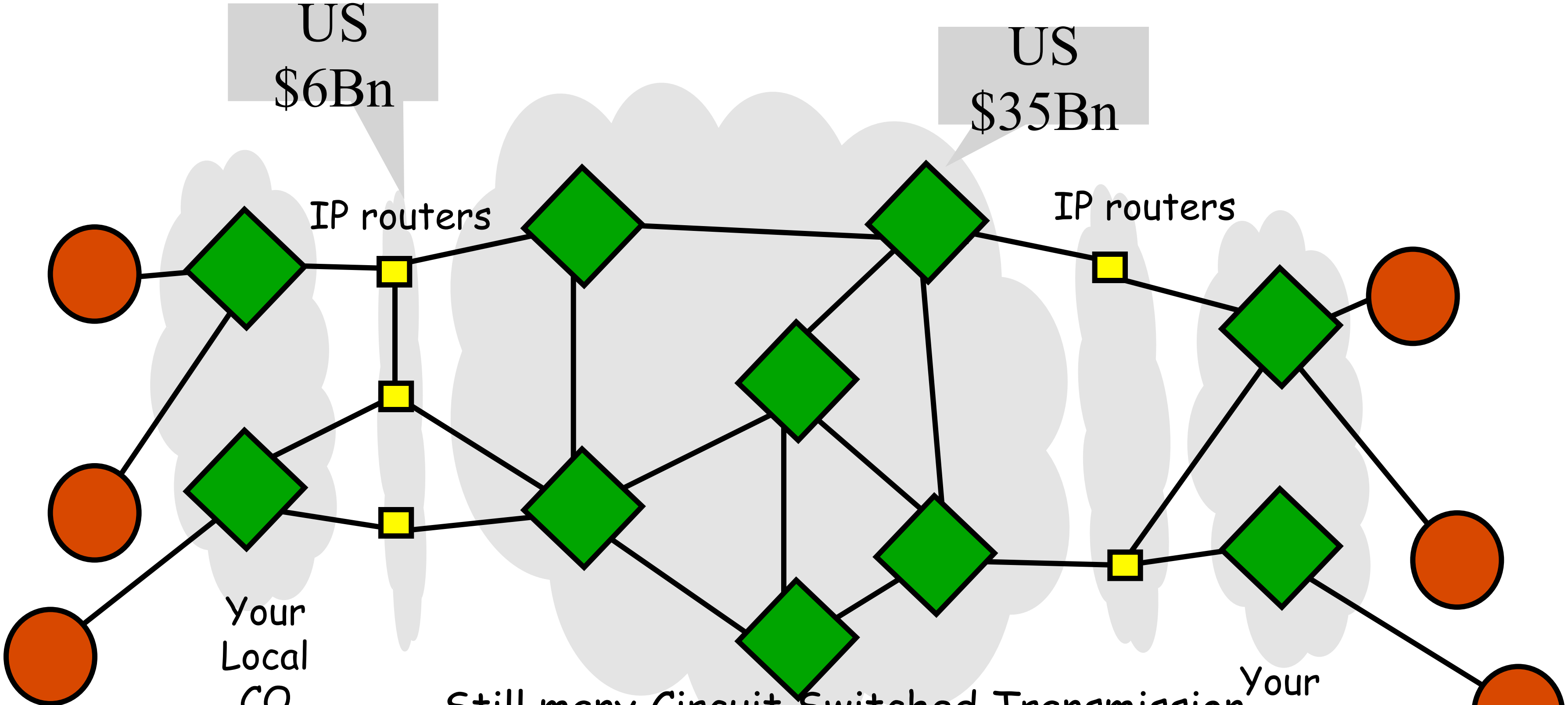
- According to Cisco, they have sold more than 10,000 of their Carrier-class Routing Systems (CRS-1 and CRS-3) to 750+ customers world-wide since 2004.

Beware that Routers are only a SMALL part of the Telecommunications Infrastructure (The OLD view of Transport vs. Switching)

How datacom/ networking people think the Internet is



The Network Core: How the Internet really looks like:



US $6Bn

US $35Bn

IP routers

IP routers

Your Local CO

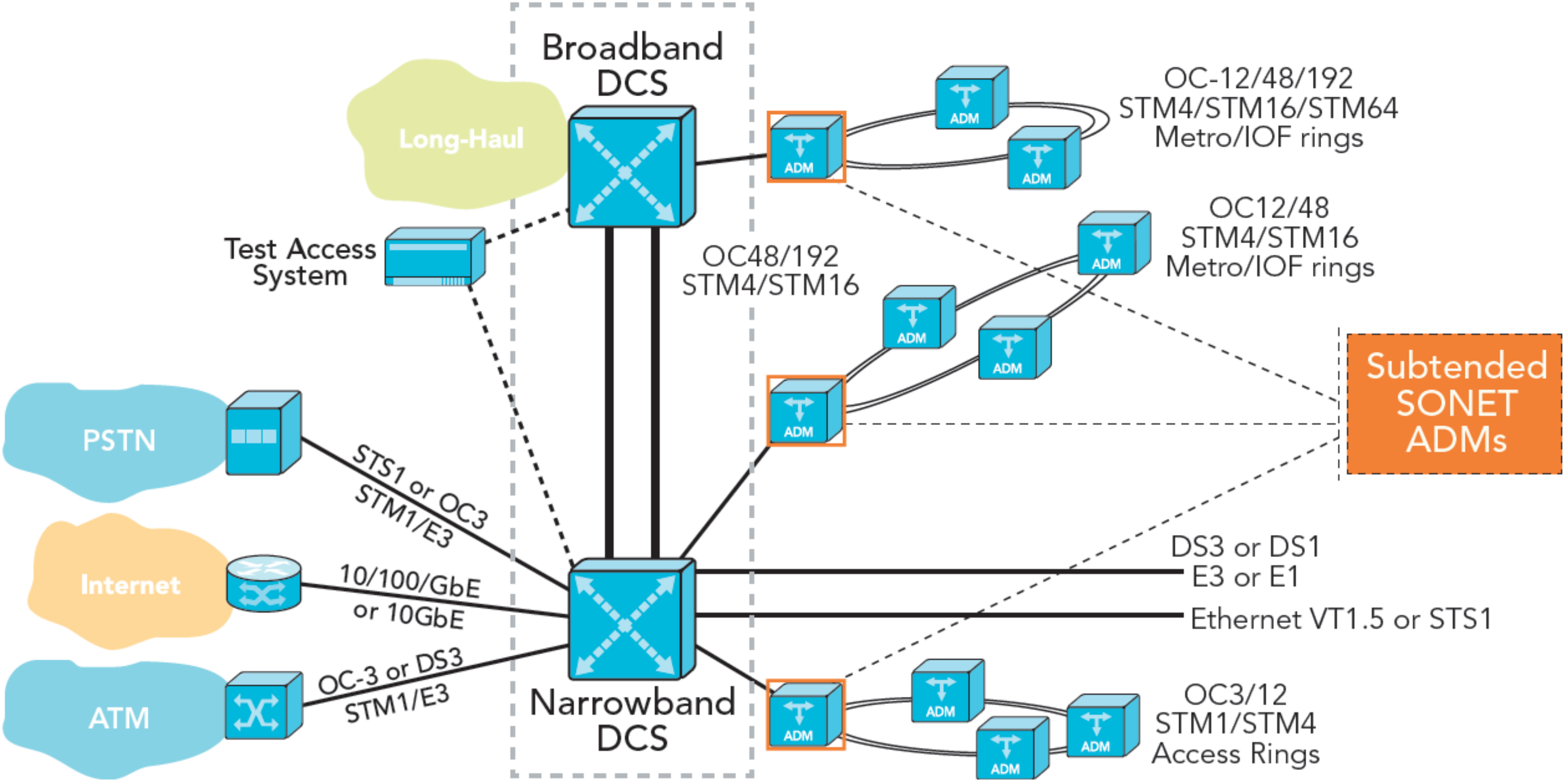
and its access networks

Your Local CO

and its access networks

Still many Circuit Switched Transmission Network, using Optical networking technologies e.g. SONET Mux, ADMs, Cross-connects to build SONET rings/meshes ; DWDM ; some packet-switched ATM/MPLS Cross-connects/ MetroEthernet Switches/ Resilient Packet Rings (RPR), etc

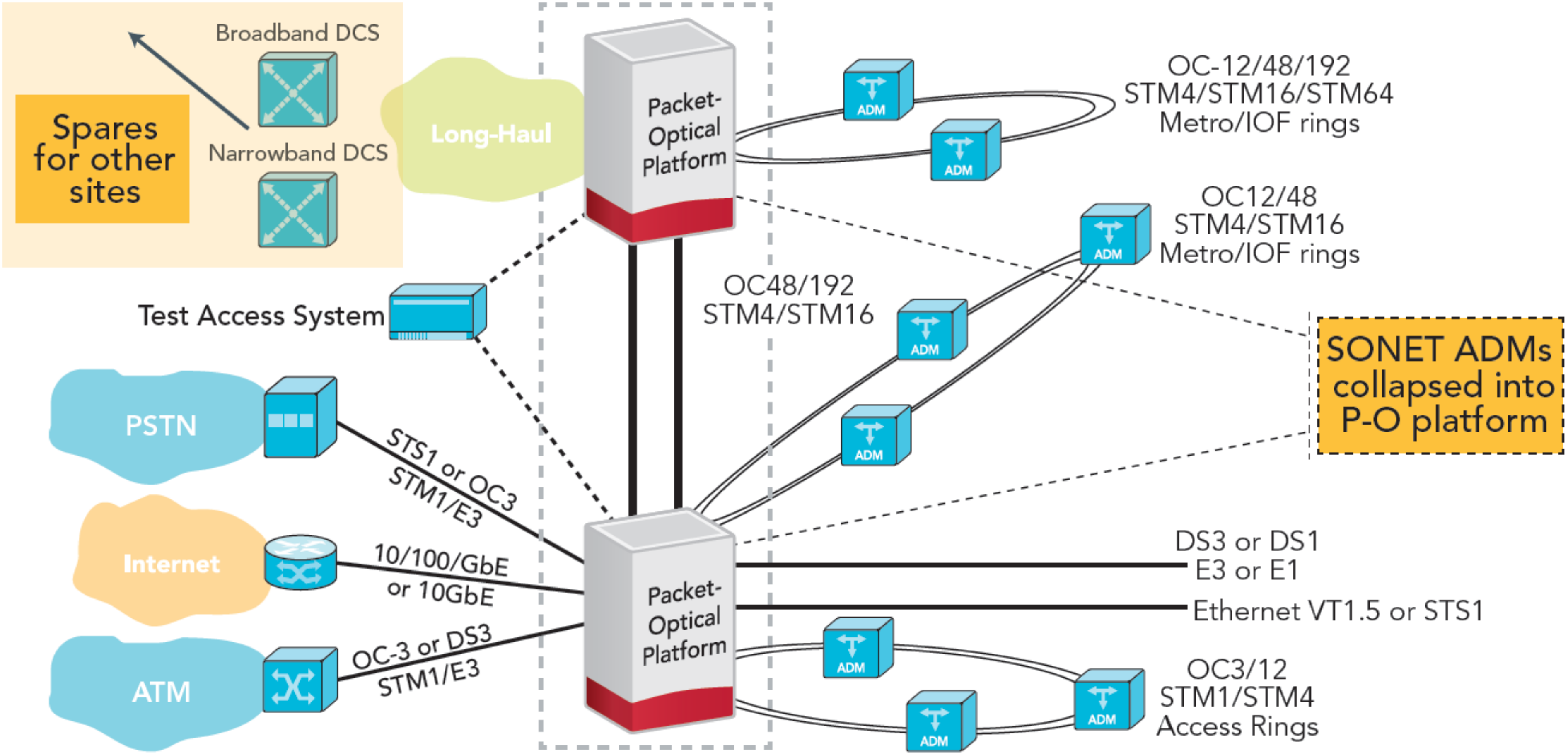
Today's (Legacy) Digital Cross-Connect (DCS) and SONET/SDH Architecture



Source: Ciena

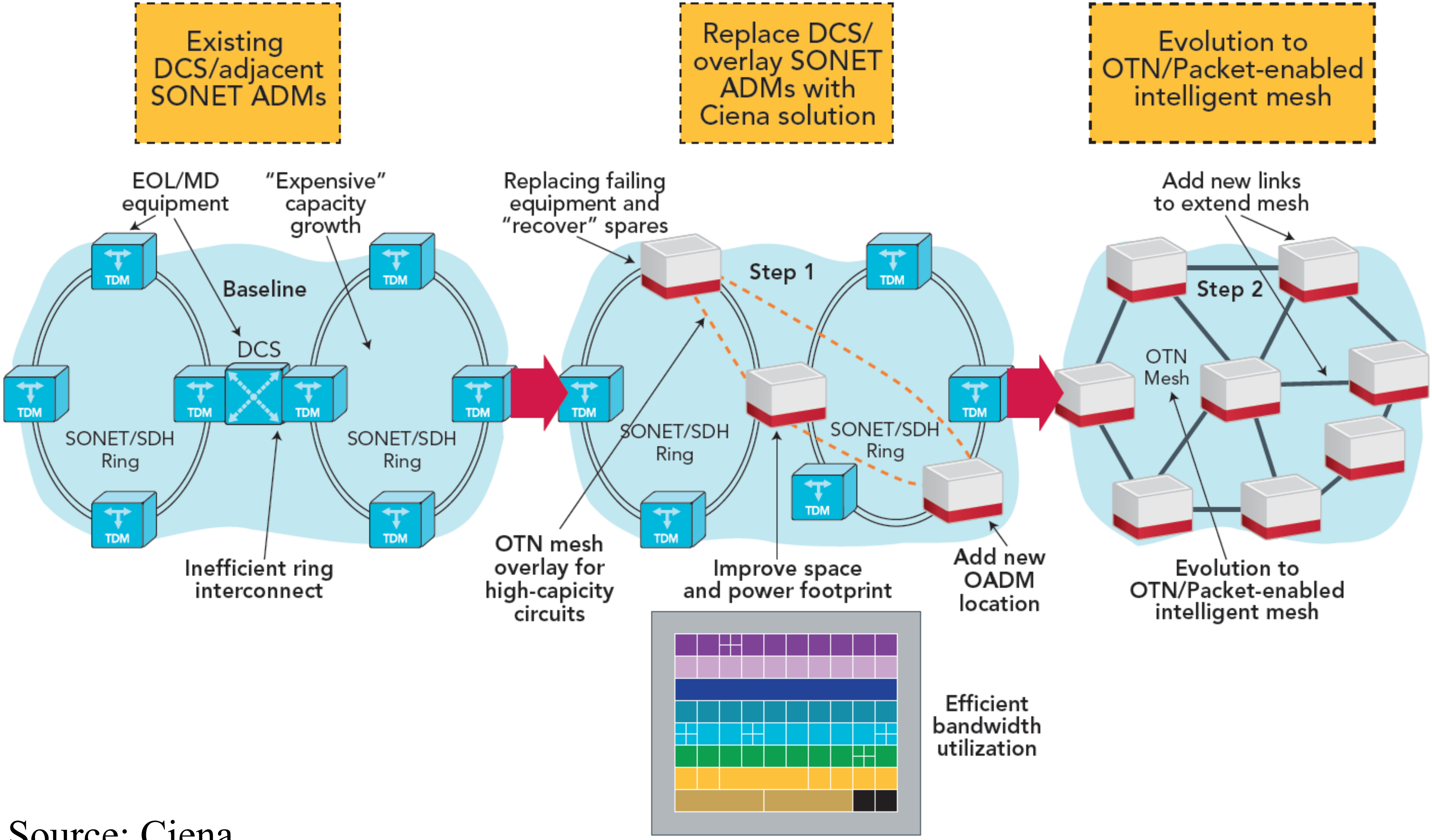
http://media.ciena.com/documents/SONET_SDH_Network_Modernization_Is_Long_Overdue_WP.pdf

Evolving towards Packet-Optical Network (OTN)



Source: Ciena

Migration to Optical Transport Network (OTN) with Mesh Restoration



Existing DCS/adjacent SONET ADMs
Replace DCS/ overlay SONET ADMs with Ciena solution
Evolution to OTN/Packet-enabled intelligent mesh
EOL/MD equipment
"Expensive" capacity growth
Baseline
DCS
TDM
SONET/SDH Ring
SONET/SDH Ring
Inefficient ring interconnect
Replacing failing equipment and "recover" spares
Step 1
SONET/SDH Ring
SONET/SDH Ring
OTN mesh overlay for high-capicity circuits
Improve space and power footprint
Add new OADM location
Add new links to extend mesh
Step 2
OTN Mesh
Evolution to OTN/Packet-enabled intelligent mesh
Efficient bandwidth utilization

Source: Ciena

Reference: http://media.ciena.com/documents/Experts_Guide_to_OTN_ebook.pdf

Netflow: A tool for Network flow/Traffic Measurement

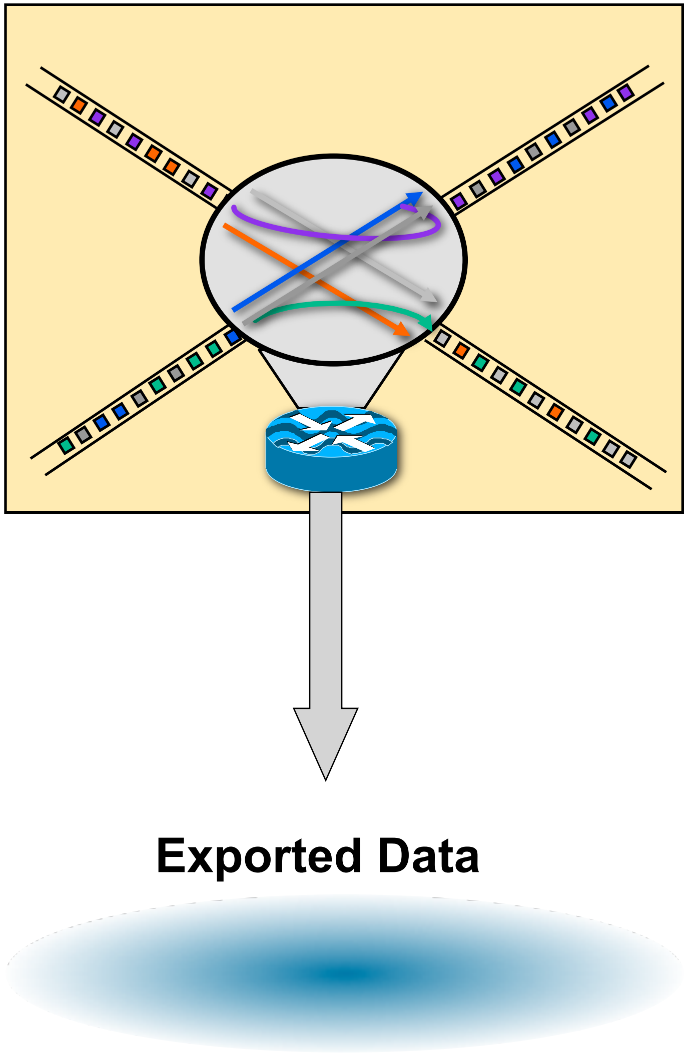
Network Traffic Monitoring tools

- Flow-based solution
  - Implemented in routers
  - E.g. Netflow
    - implemented in Cisco routers
    - Available since mid-90s ; Cisco's version upto ver.9,
  - Competing products include Jflow from Juniper and Sflow from HP
  - Get standardized by IETF as IPFIX (IP Flow Information Export) ;
    - RFCs 5101-5103, 5153,5470-5473
- Packet-based solution
  - Implemented in a stand-alone box, listening promiscuously on a multiple access medium (e.g. Ethernet)
  - E.g. Wireshark (used to call Ethereal)

What is a flow?

Defined by seven unique keys (according to Netflow):

- Source IP address
- Destination IP address
- Source port
- Destination port
- Layer 3 protocol type
- TOS byte (Type of Service)
- Input logical interface (ifIndex)



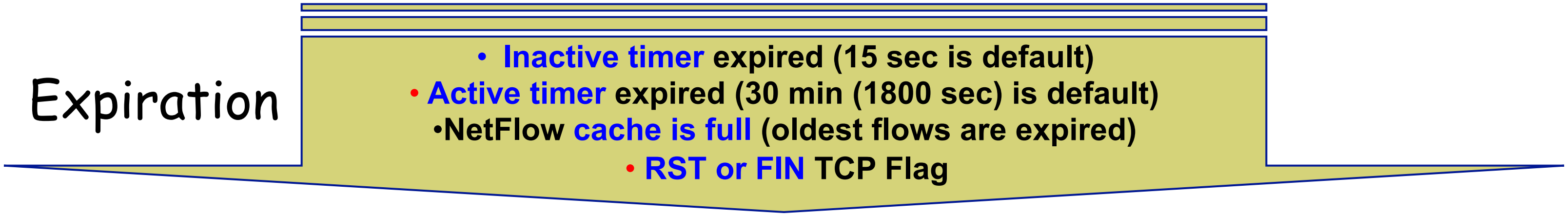
What is a flow?

- TCP flows – delineated by special packets SYN, FIN or RST etc
- Could be any sequence of packets (between the same source-destination addr/port)
  - Such flows assumed to end after some period of inactivity (default inactive timer=15 seconds)
  - Such flows are also terminated after a sufficiently long time of activity, for keeping track of what is going on (default active timer=30 minutes)
- When monitor’s cache is full, some flows may also be terminated

1. Create and update flows in NetFlow Cache

SrcIf	SrcIPadd	DstIf	DstIPadd	Protocol	TOS	Flgs	Pkts	SrcPort	SrcMsk	SrcAS	DstPort	DstMsk	DstAS	NextHop	Bytes/Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A2	/24	5	00A2	/24	15	10.0.23.2	1528	1745	4
Fa1/0	173.100.3.2	Fa0/0	10.0.227.12	6	40	0	2491	15	/26	196	15	/24	15	10.0.23.2	740	41.5	1
Fa1/0	173.100.20.2	Fa0/0	10.0.227.12	11	80	10	10000	00A1	/24	180	00A1	/24	15	10.0.23.2	1428	1145.5	3
Fa1/0	173.100.6.2	Fa0/0	10.0.227.12	6	40	0	2210	19	/30	180	19	/24	15	10.0.23.2	1040	24.5	14

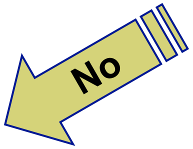
2. Expiration

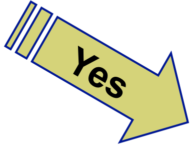


- **Inactive timer expired (15 sec is default)**
- **Active timer expired (30 min (1800 sec) is default)**
- **NetFlow cache is full (oldest flows are expired)**
- **RST or FIN TCP Flag**

SrcIf	SrcIPadd	DstIf	DstIPadd	Protocol	TOS	Flgs	Pkts	SrcPort	SrcMsk	SrcAS	DstPort	DstMsk	DstAS	NextHop	Bytes/Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A2	/24	5	00A2	/24	15	10.0.23.2	1528	1800	4

3. Aggregation?





e.g. Protocol-Port Aggregation Scheme becomes

Protocol	Pkts	SrcPort	DstPort	Bytes/Pkt
11	11000	00A2	00A2	1528

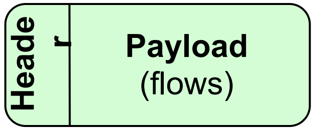
4. Export Version

Non-Aggregated Flows – export **Version 5 or 9**

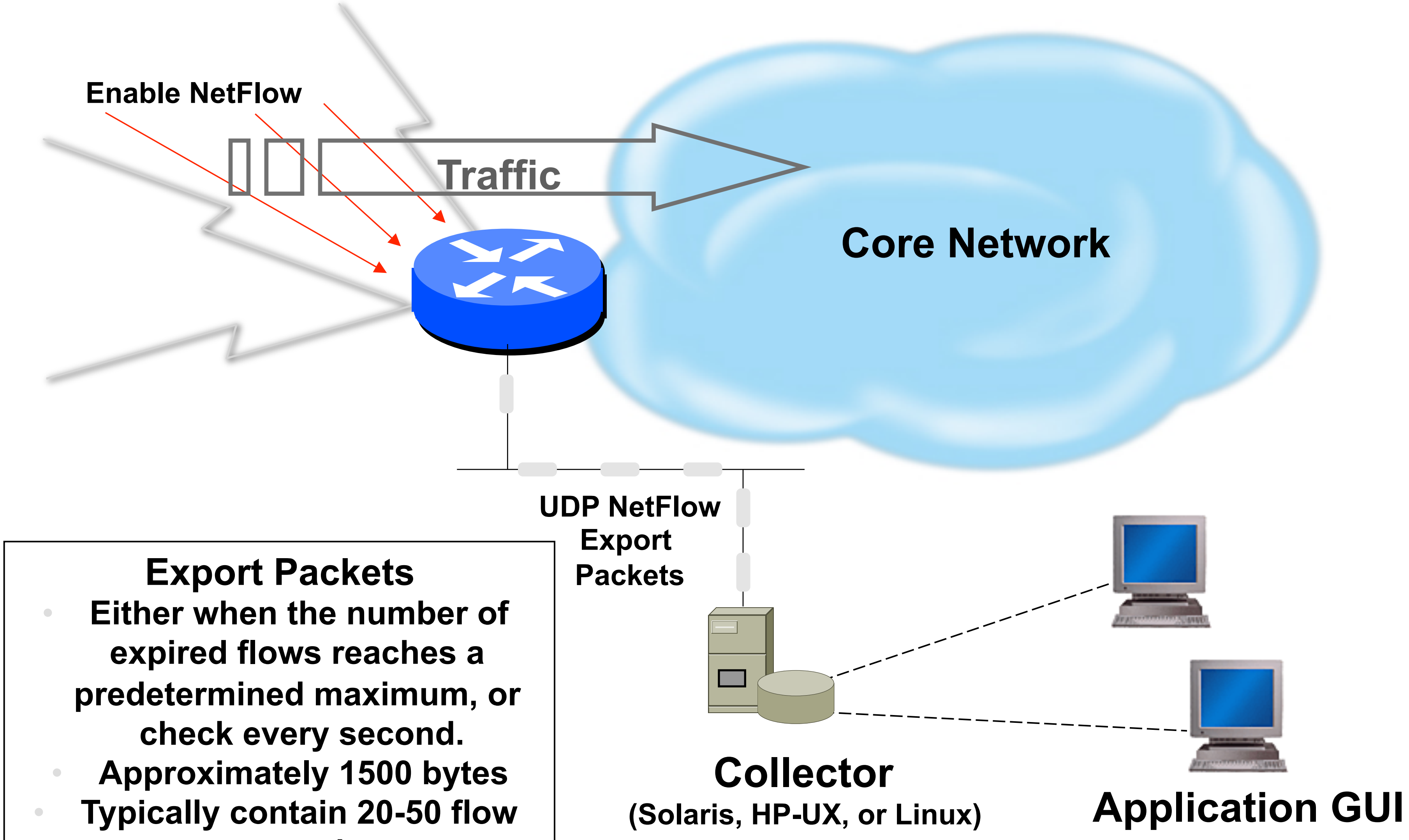
Aggregated Flows – export **Version 8 or 9**

5. Transport Protocol

Export Packet



Export Packets



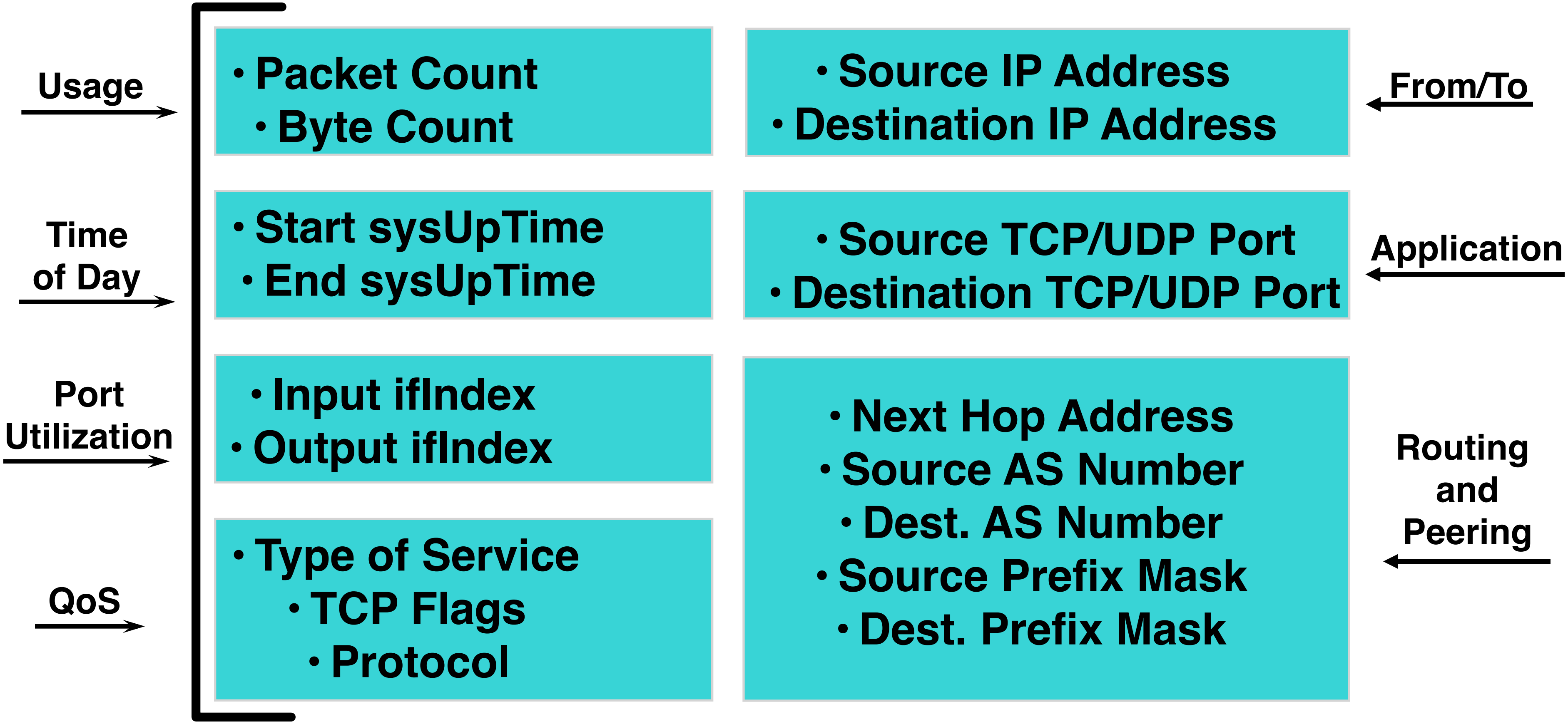
Export Packets

- Either when the number of expired flows reaches a predetermined maximum, or check every second.
- Approximately 1500 bytes
- Typically contain 20-50 flow records

NetFlow Versions

NetFlow Version	Comments
1	Original
5	Standard and most common
7	Specific to Cisco Catalyst 6500 and 7600 Series Switches Similar to Version 5, but does not include AS, interface, TCP Flag & TOS information
8	Choice of eleven aggregation schemes Reduces resource usage
9	Flexible, extensible file export format to enable easier support of additional fields & technologies; coming out now MPLS, Multicast, & BGP Next Hop

Version 5 - Flow Format



Usage

- Packet Count
- Byte Count

Time of Day

- Start sysUpTime
- End sysUpTime

Port Utilization

- Input ifIndex
- Output ifIndex

QoS

- Type of Service
- TCP Flags
- Protocol

From/To

- Source IP Address
- Destination IP Address

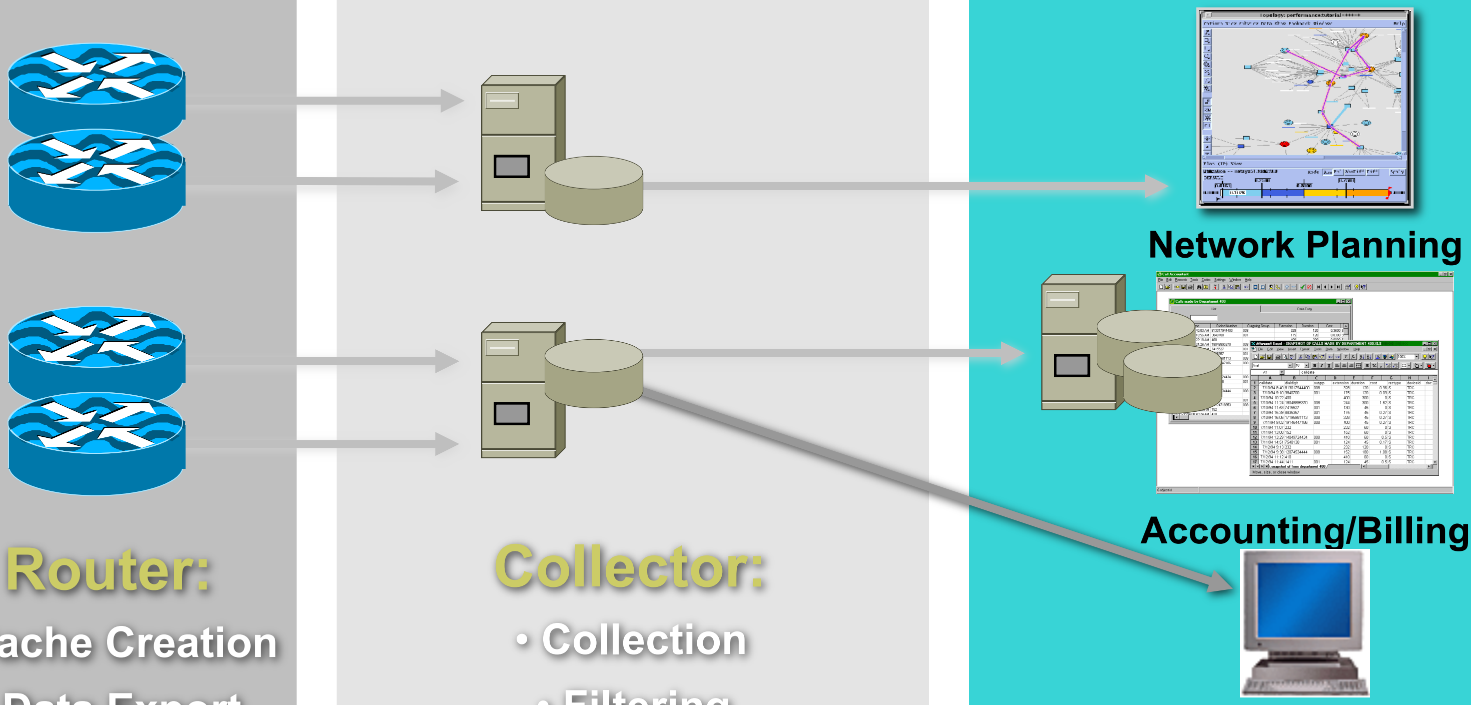
Application

- Source TCP/UDP Port
- Destination TCP/UDP Port

Routing and Peering

- Next Hop Address
- Source AS Number
- Dest. AS Number
- Source Prefix Mask
- Dest. Prefix Mask

NetFlow Infrastructure



Network Planning

Accounting/Billing

Cisco

Router:

- Cache Creation
- Data Export
- Aggregation

Cisco & Partners

Collector:

- Collection
- Filtering
- Aggregation
- Storage
- File System Management

Partners

Applications:

Data Presentation

Principle Netflow Benefits

Service Provider

- Peering arrangements
- Network Planning
- Traffic Engineering
- Accounting and billing
- Security Monitoring

Enterprise

- Internet access monitoring (protocol distribution, where traffic is going/coming)
- User Monitoring
- Application Monitoring
- Charge Back billing for departments
- Security Monitoring

Billing

- Flat-rate billing does not necessarily scale
  - Competitive pricing models can be created with usage-based billing
- Usage-based billing considerations
  - Time of day
  - Within or outside of the network
  - Application
  - Distance-based
  - Quality of Service (QoS) / Class of Service (CoS)
  - Bandwidth usage
  - Transit or peer
  - Data transferred
  - Traffic class

Tracking Users

- Who are my top N talkers, and what percentage of traffic do they represent?
- How many users are on the network at a given time?
  - When will upgrades affect the least number of users?
- How long do users spend connected to the network?
- Which Internet sites do they use?
- What is a typical pattern of usage between sites?
- Are users staying within an Acceptable Usage Policy (AUP)?
- Alarm DOS attacks like smurf, fraggle, and SYN flood

Sampled Netflow

- For high speed interfaces, the processor and the memory cannot keep up with the packet rate, Cisco introduced sampled NetFlow.
- Packet-based (one of every N packets is sampled)

  Deterministic or random
- Time-based

  uses traffic from the first 64 milliseconds every 4096 milliseconds.