CS4450

Computer Networks:
Architecture and Protocols

Lecture 15
Towards the end-to-end ….

Spring 2018
Rachit Agarwal

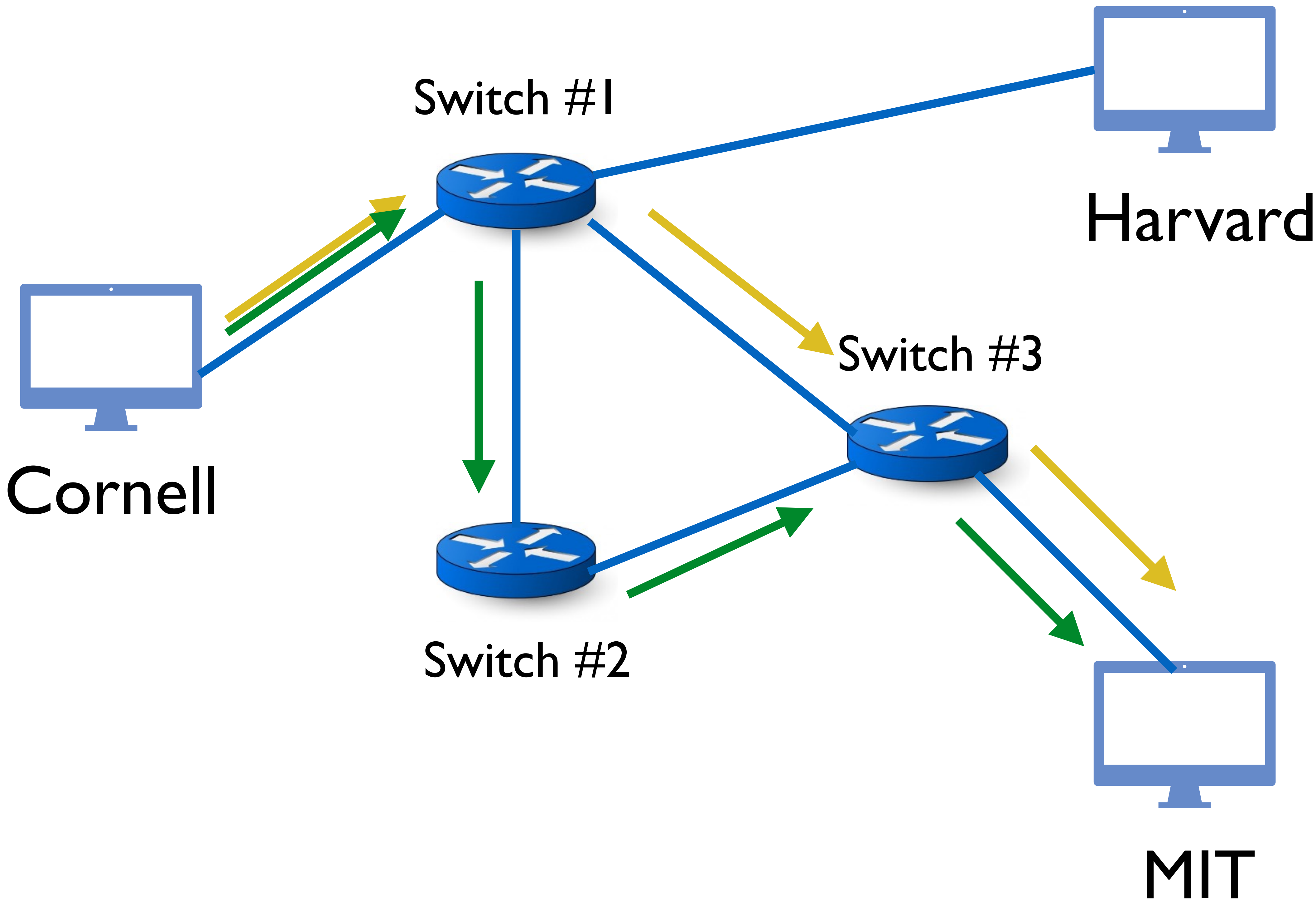
Goal of Today's Lecture

- Switch and Router fabric
  - Longest Prefix match (scalability)
  - Queueing (Input/Output/both)
- Almost there ….

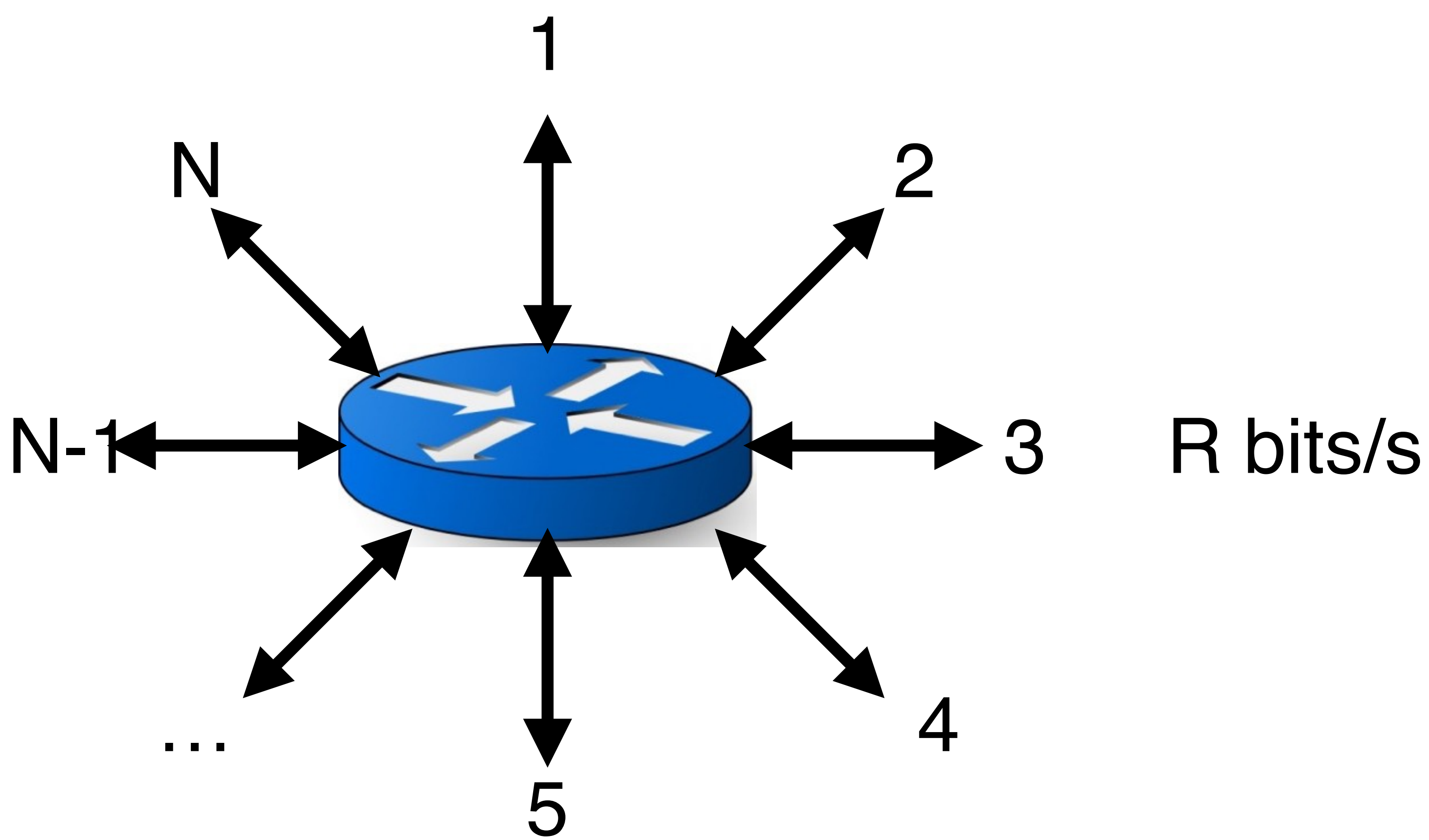
IP Routers and Switches (used interchangeably today)

- Core building block of Internet infrastructure
- $120B+ industry
- Vendors: Cisco, Huawei, Juniper, Alcatel-Lucent (account for >90%)

Recap: Routers Forward Packets

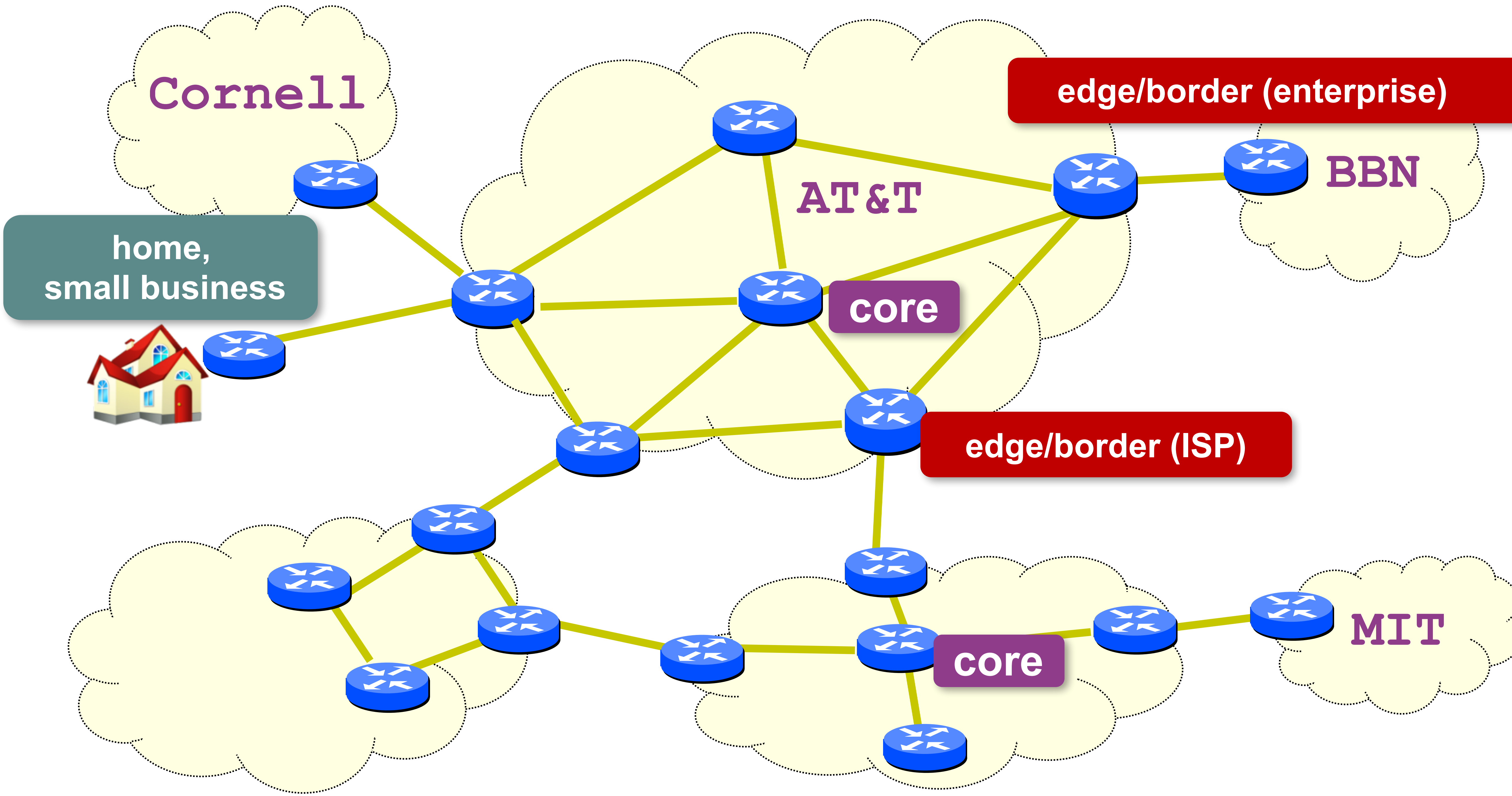


Router Definitions



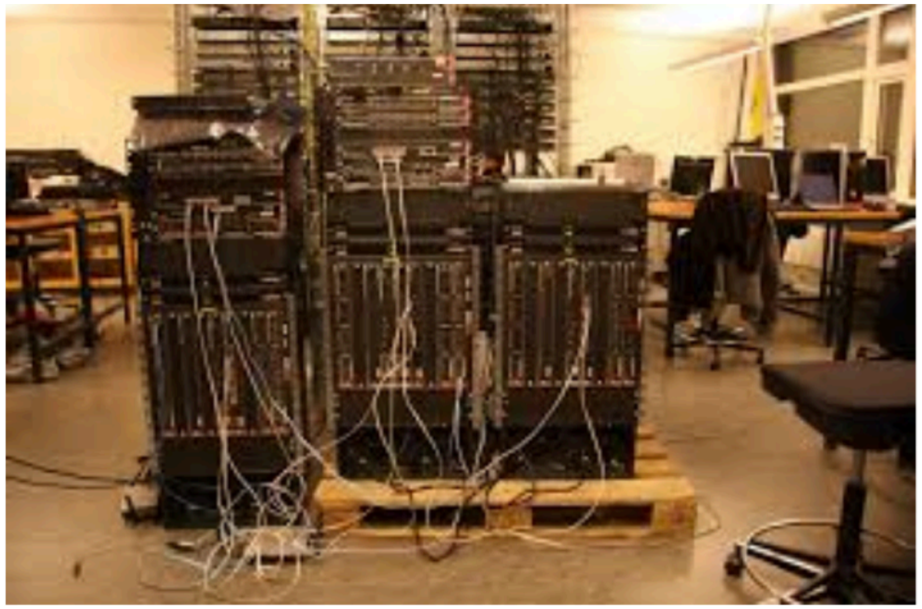
- N = No. Of external router ports
- R = speed ("line rate") of a port
- Router capacity = NxR

Networks and Routers



Examples of Routers (core)

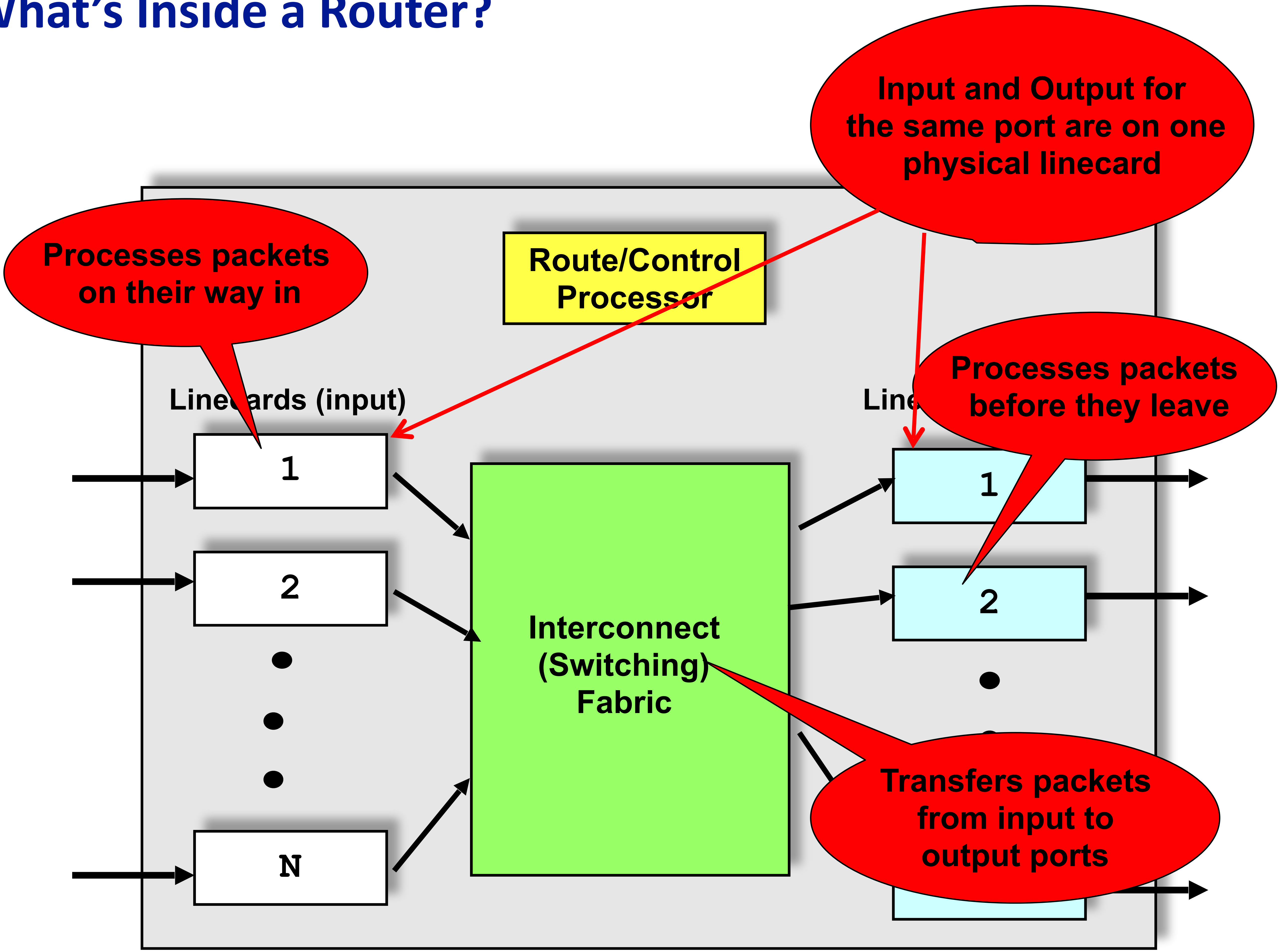
- **Core:** Cisco CRS
  - R = 10/40/100 Gbps
  - NR = 922 Tbps
  - Netflix: 0.7 GB/hr (1.5Mb/s)
  - ~600 million concurrent Netflix users



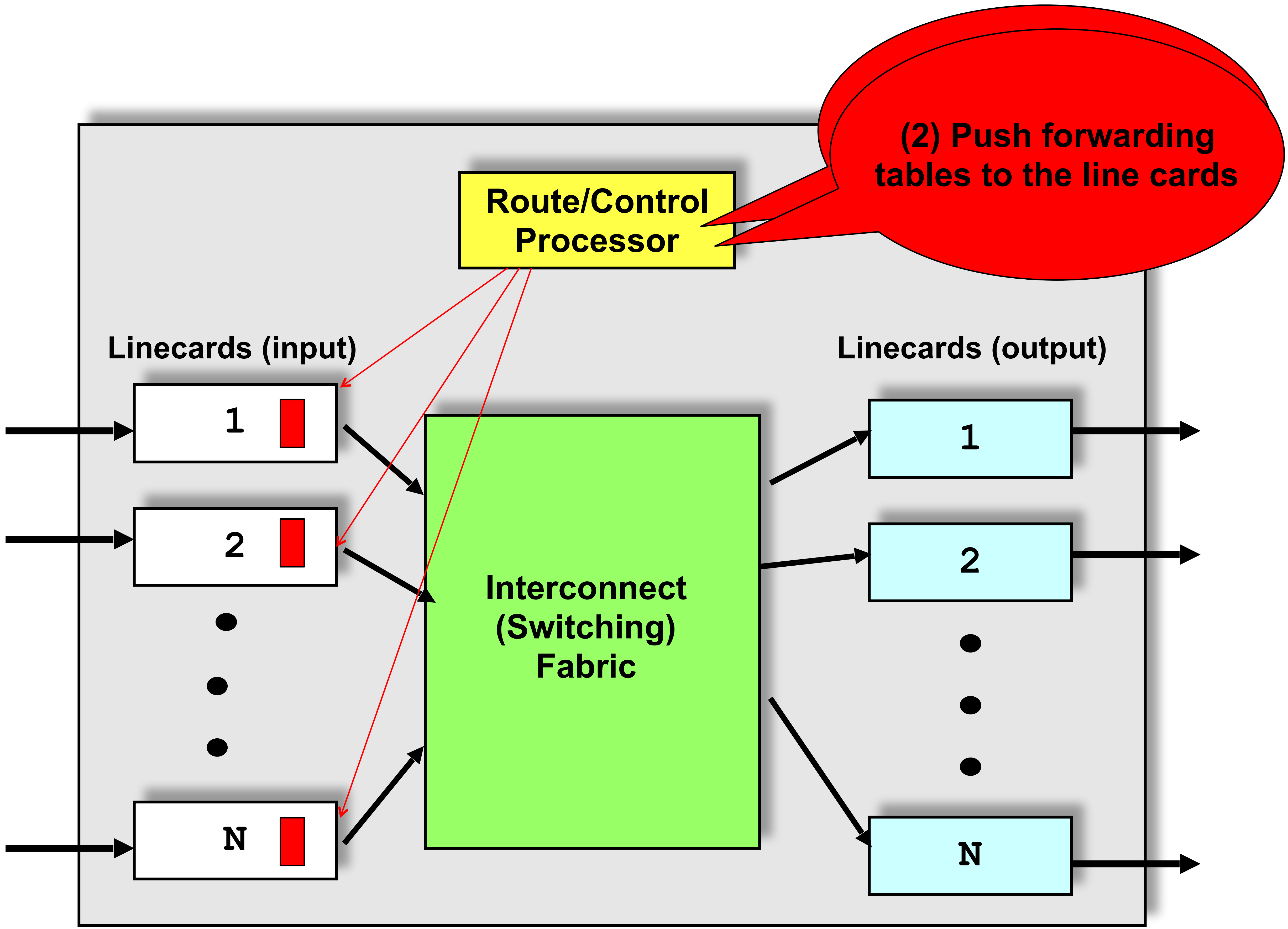
- **Edge (ISP):** Cisco ASR
  - R = 1/10/40 Gbps
  - NR = 120 Gbps

- **Edge (enterprise):** Cisco 3945E
  - R = 10/100/1000 Mbps
  - NR < 10 Gbps

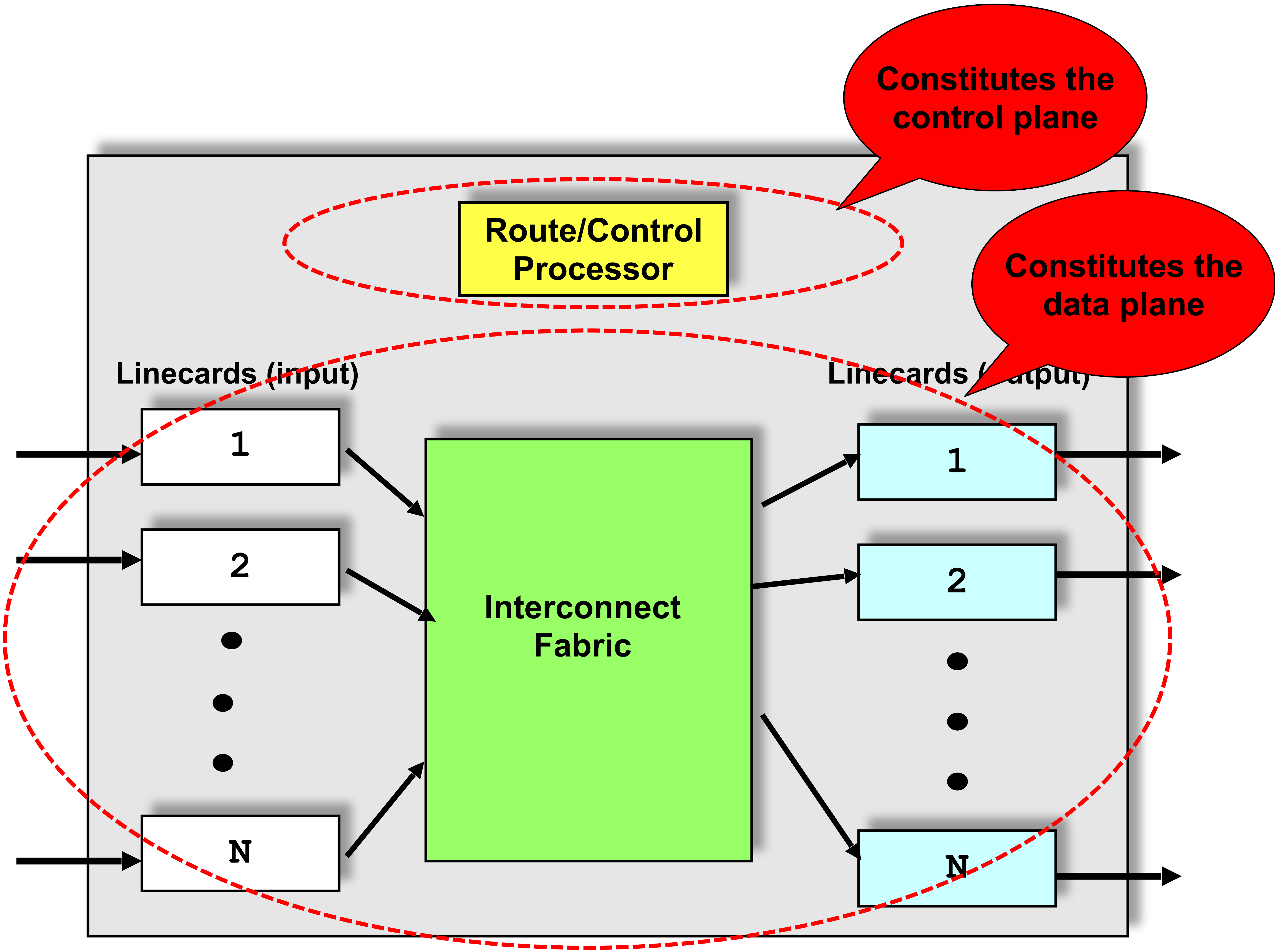
What's Inside a Router?



What's Inside a Router?



What's Inside a Router?



Input Line Cards: Tasks

- Receive incoming packets (physical layer stuff)
- Update the IP header
  - TTL, Checksum (maybe some other fields)
- Lookup the output port for the destination IP address
- Queue the packet at the switch fabric

Challenge: Speed!

- 100B packets @ 40Gbps => packet every 20 nano secs!

- Typically implemented with specialized hardware
    - ASICs, specialized “network processors

Looking up the Output Port

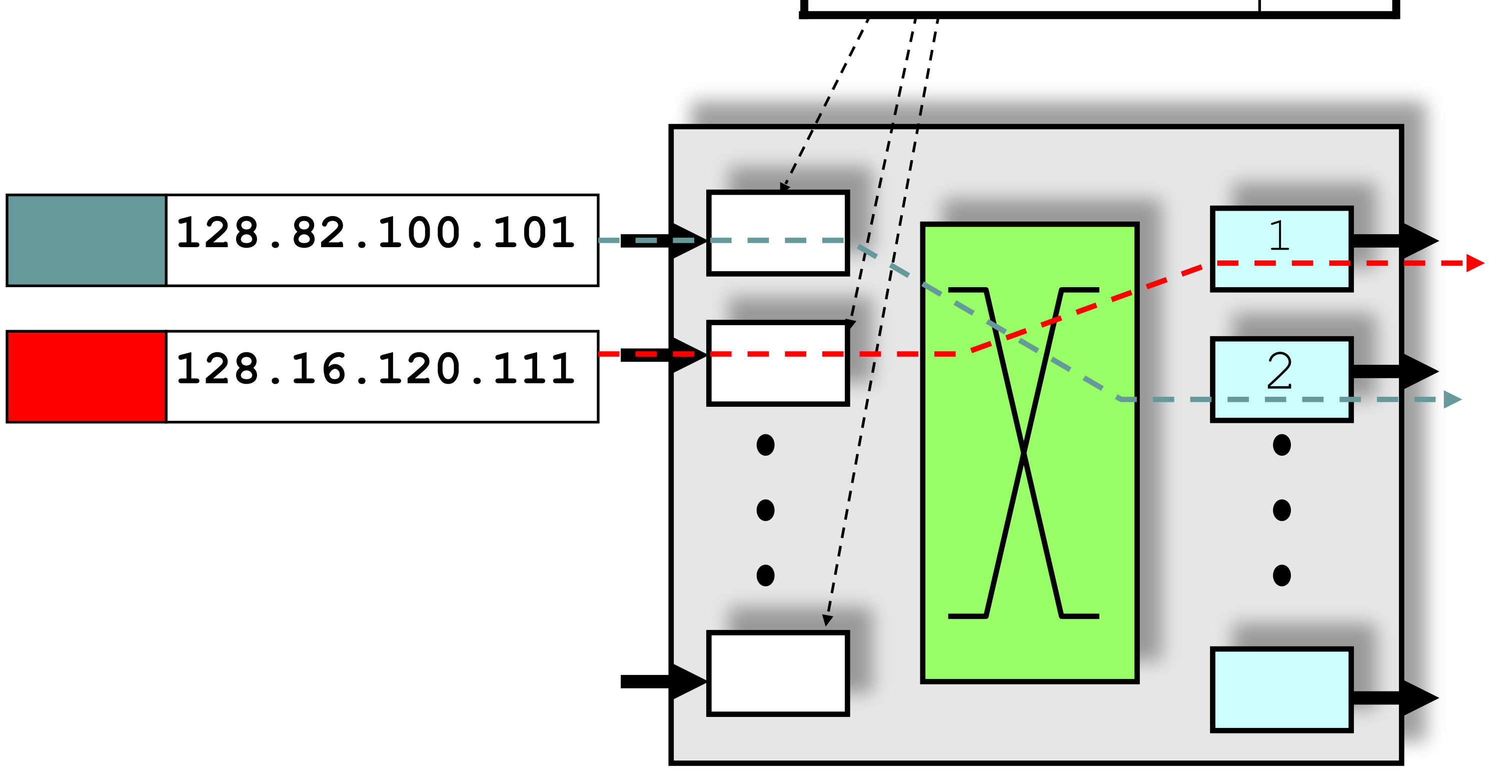
- Upon receiving a packet
  - Inspect the destination IP address in the header
  - Index into the forwarding table
  - If no match, select the **default route**
  - Forward packet out appropriate interface

- **Default route**
  - Configured to cover cases where no matches
  - Allows small tables at edge (w/o routing algorithms)
    - **if it isn’t on my subnet, send it to my ISP**

Scaling the Lookup

- One entry for each address: billions of entries!
- For scalability, addresses are **aggregated**
- **Longest Prefix match**
  - Find the entry with matching “longest prefix” with destination address

128.16.120.xxx	1
128.82.xxx.xxx	3
128.82.100.xxx	2
…	…



Finding a Match

- Incoming packet destination: 201.143.7.0

Prefix	Port
201.143.0.0/22	Port 1
201.143.4.0.0/24	Port 2
201.143.5.0.0/24	Port 3
201.143.6.0/23	Port 4

Finding a Match: Covert to Binary

- Incoming packet destination: 201.143.7.0

11001001	10001111	00000111	11010010

Routing Table

201.143.0.0/22

11001001	10001111	000000 - -	- - - - - - - -

201.143.4.0/24

11001001	10001111	00000100	- - - - - - - -

201.143.5.0/24

11001001	10001111	00000101	- - - - - - - -

201.143.6.0/23

11001001	10001111	0000011-	- - - - - - - -

Finding a Match: Covert to Binary

- Incoming packet destination: 201.143.7.0

11001001	10001111	00000111	11010010

Routing Table

201.143.0.0/22

11001001	10001111	000000 - -	- - - - - - - -

201.143.4.0/24

11001001	10001111	00000100	- - - - - - - -

201.143.5.0/24

11001001	10001111	00000101	- - - - - - - -

201.143.6.0/23

11001001	10001111	0000011-	- - - - - - - -

Finding a Match: Covert to Binary

- Incoming packet destination: 201.143.7.0

11001001	10001111	00000111	11010010

Routing Table

201.143.0.0/22

~~11001001~~	~~10001111~~	~~000000 - -~~	~~- - - - - - - - -~~

201.143.4.0/24

11001001	10001111	00000100	- - - - - - - - -

201.143.5.0/24

11001001	10001111	00000101	- - - - - - - - -

201.143.6.0/23

11001001	10001111	0000011-	- - - - - - - - -

Longest Prefix Match

- Incoming packet destination: 201.143.7.0

11001001	10001111	00000111	11010010

Routing Table

201.143.0.0/22

~~11001001~~	~~10001111~~	~~000000 - -~~	~~- - - - - - - - -~~

201.143.4.0/24

~~11001001~~	~~10001111~~	~~00000100~~	~~- - - - - - - - -~~

201.143.5.0/24

~~11001001~~	~~10001111~~	~~00000101~~	~~- - - - - - - - -~~

201.143.6.0/23

11001001	10001111	0000011-	- - - - - - - - -

Check an address against all destination prefixes and select the prefix it matches with on the most bits

Finding the Match Efficiently

- Testing each entry to find a match scales poorly
  - Roughly (number of entries) × (number of bits)

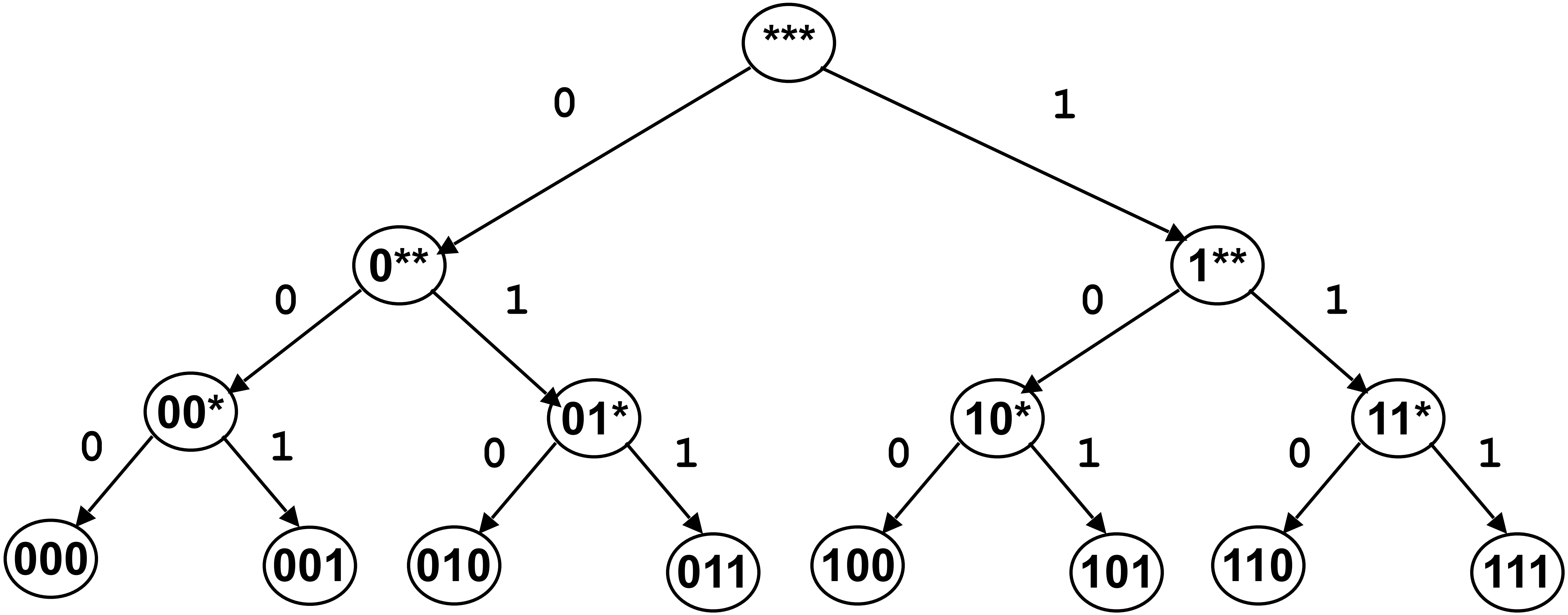
- Must leverage tree structure of binary strings
  - Set up tree-like data structure
  - Called a **TRIE**
    - We will briefly discuss it; more details in text
    - In case you are interested ….

Consider Four 3-Bit Prefixes

- Just focusing on the bits where all the action is….

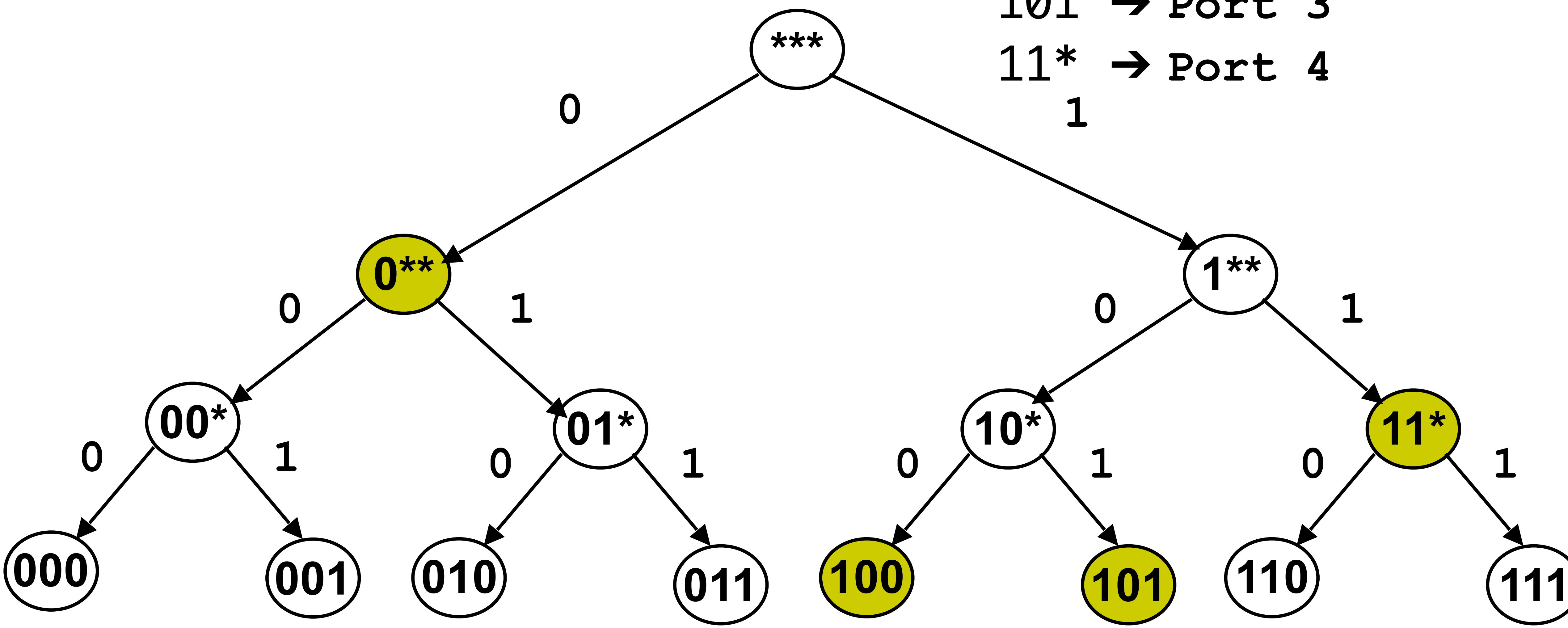
- 0** → Port 1
- 100 → Port 2
- 101 → Port 3
- 11* → Port 4

Tree Structure



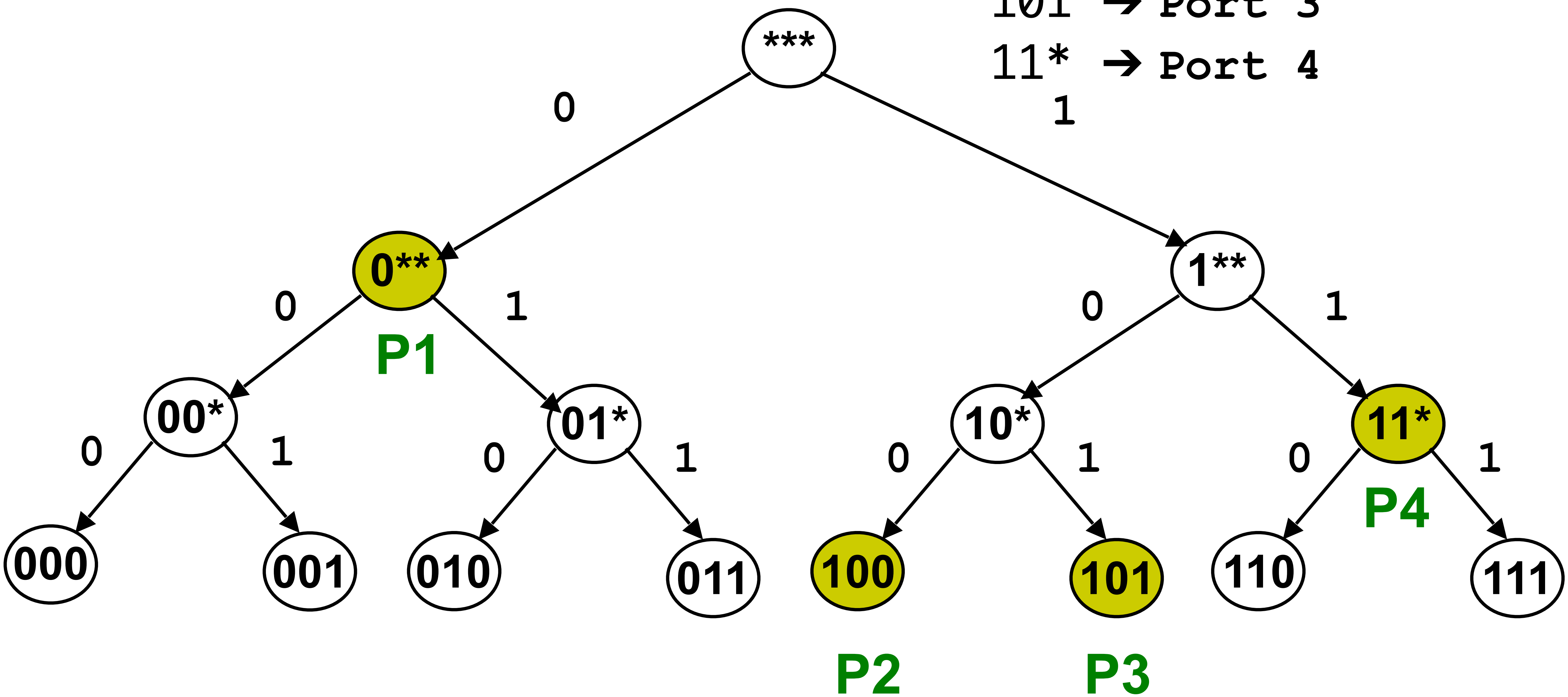
Walk Tree: Stop at Prefix Entries

0** → Port 1
100 → Port 2
101 → Port 3
11* → Port 4



Walk Tree: Stop at Prefix Entries

0** → Port 1
100 → Port 2
101 → Port 3
11* → Port 4



walking trees takes O(#bits)

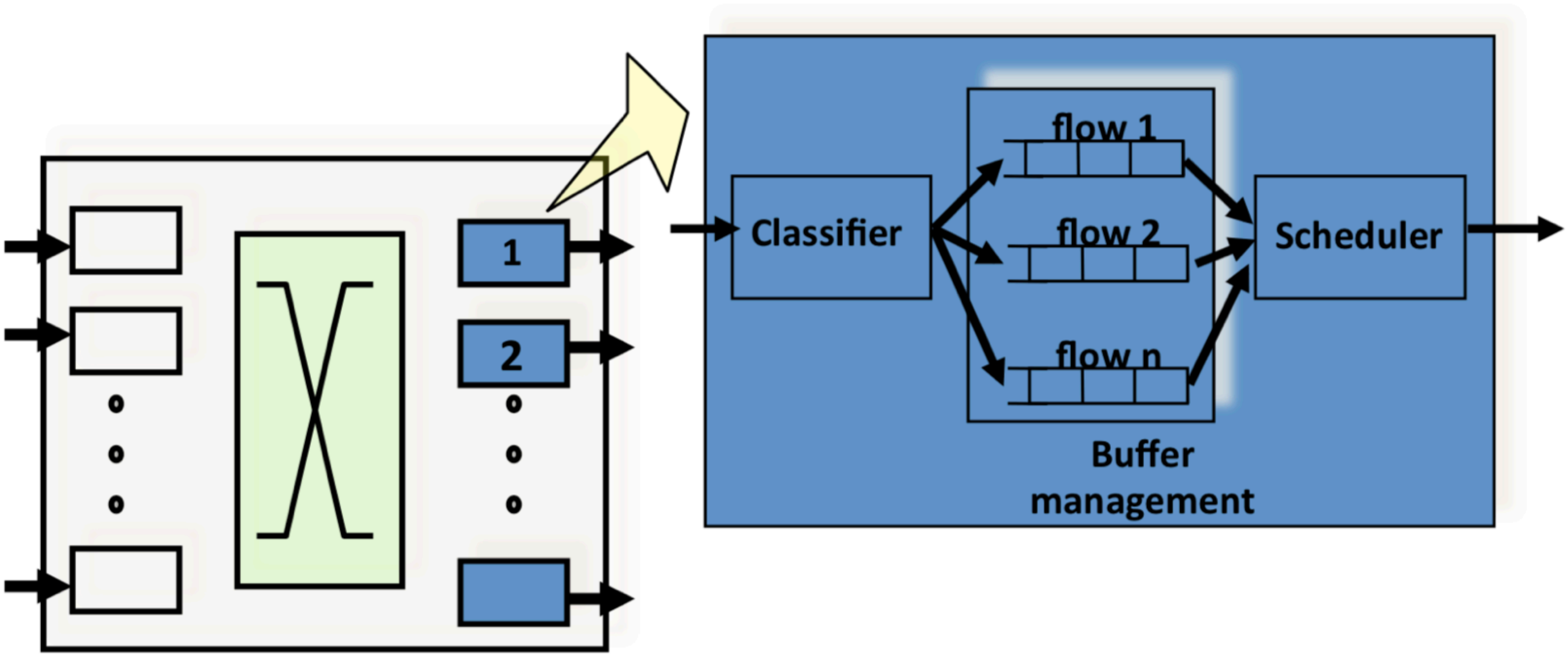
Longest Prefix Match in Real Routers

- Real routers use far more advanced/complex solutions
    - But what we discussed is the starting point
- With many heuristics and optimizations that leverage real-world patterns
    - Some destinations more popular than others
    - Some ports lead to more destinations
    - Typical fix granularities

Recap: Input Linecards

- Main challenge is processing speed
    - But what we discussed is the starting point
- Tasks involved
    - Update packet header (easy)
    - Longest prefix match lookup on destinations address (harder)
- Mostly implemented with specialized hardware

Output Linecard



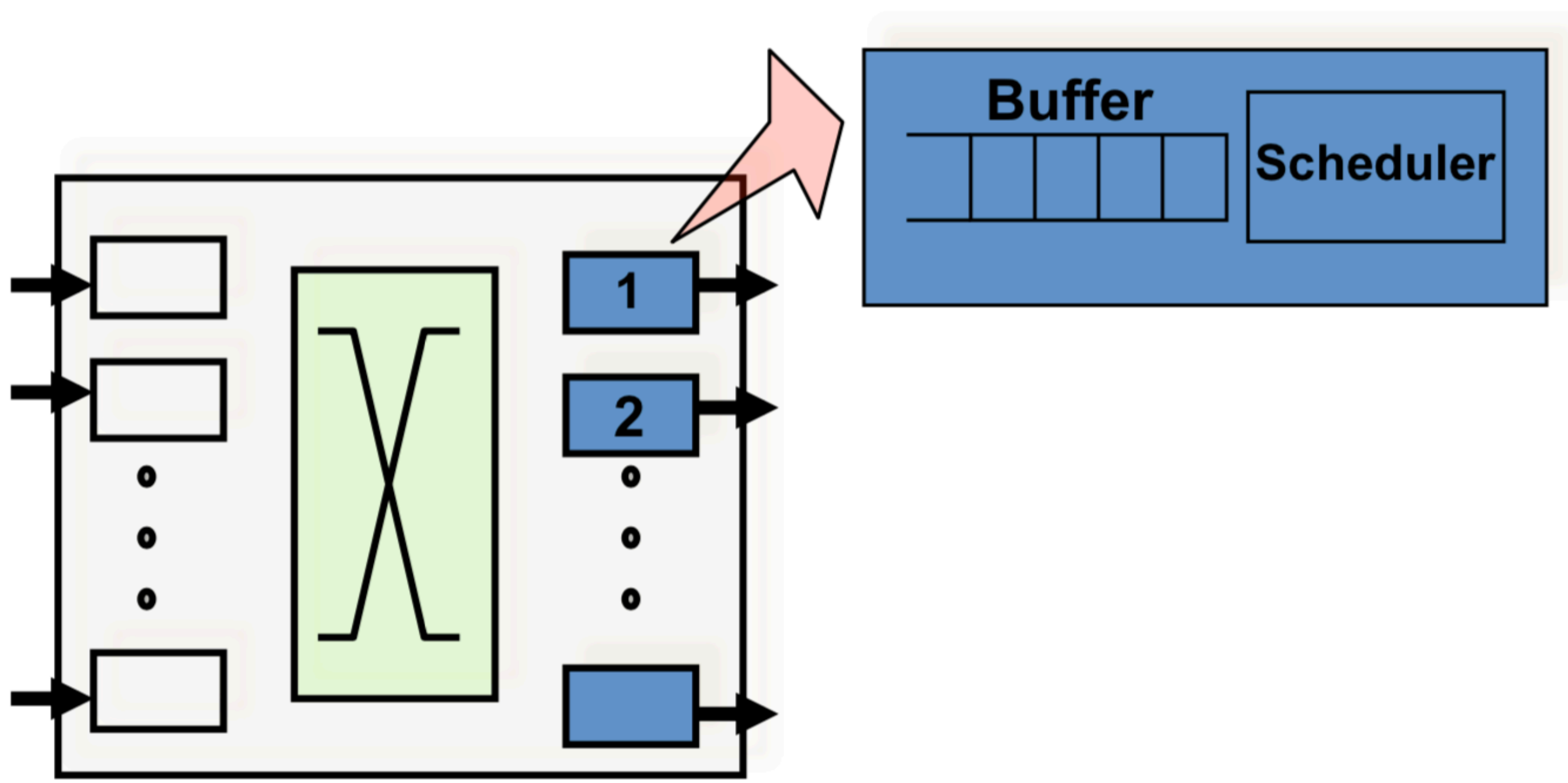
- **Packet Classification**: map each packet to a "flow"
    - Flow (for now): set of packets between two particular endpoints
- **Buffer Management**: decide when and which packet to drop
- **Scheduler**: decide when and which packet to transmit

Output Linecard

- **Packet Classification**: map each packet to a “flow”
  - Flow (for now): set of packets between two particular endpoints
- **Buffer Management**: decide when and which packet to drop
- **Scheduler**: decide when and which packet to transmit
- Used to implement various forms of policy
  - Deny all e-mail traffic from ISP X to Y (**access control**)
  - Route IP telephony traffic from X to Y via PHY_CIRCUIT (**policy**)
  - Ensure that no more than 50 Mbps are injected from ISP-X (**QoS**)

Simplest FIFO Router

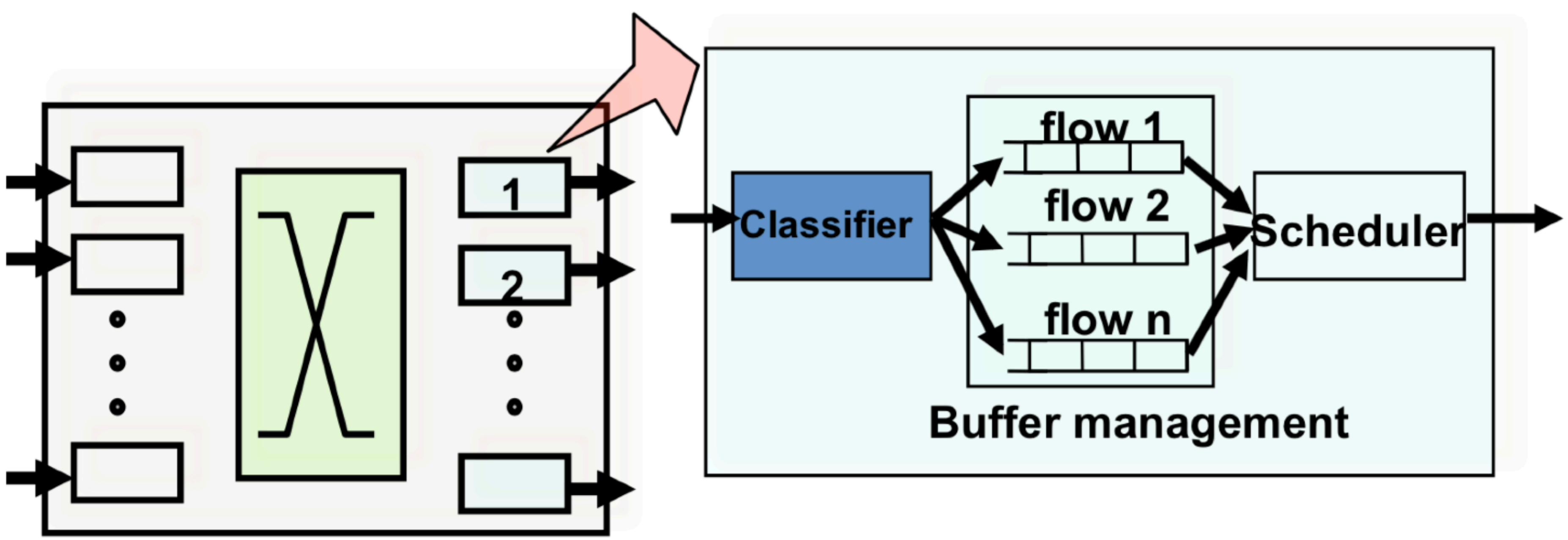
- **No classification**
- **Drop tail buffer management:** when buffer is full drop incoming packet
- **First In First Out (FIFO) Scheduling:** schedule packets in order of arrival



Packet Classification

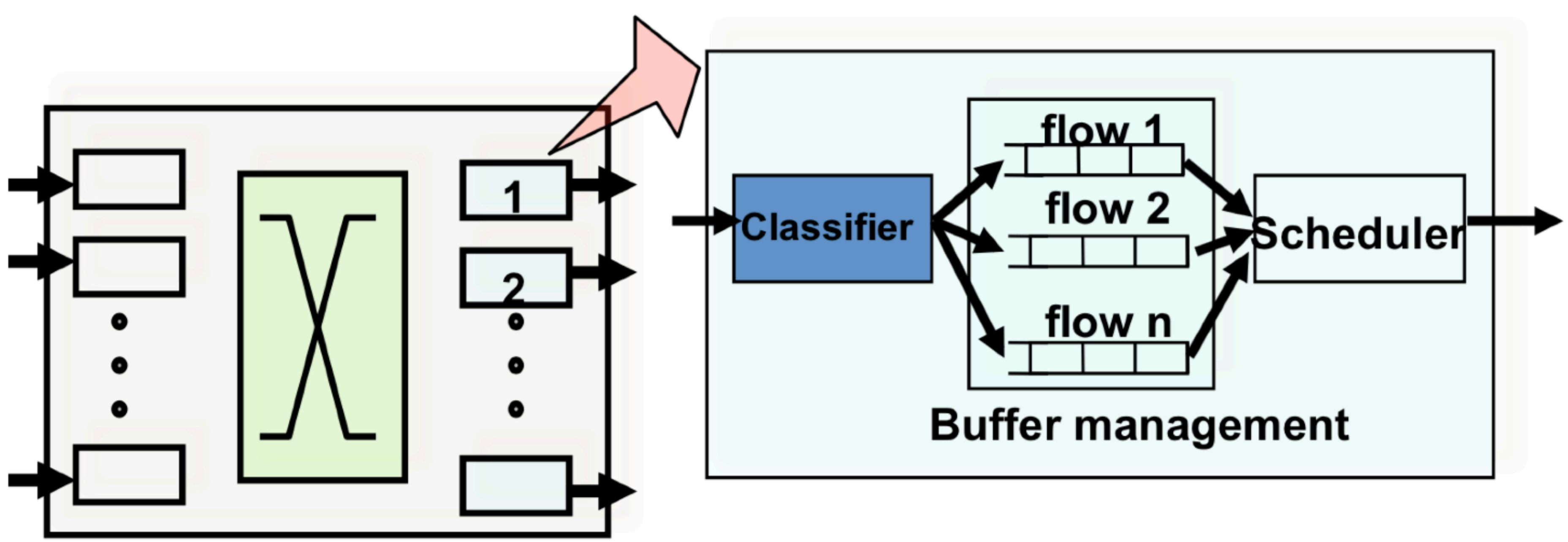
- Classify an IP packet based on the number of fields in the packet header
  - Source/destination IP address (32 bits)
  - Source/destination TCP port number (16 bits)
  - Type of Service (TOS) byte (8 bits)
  - Type of Protocol (8 bits)

- In general fields are specified by range
  - Classification requires a multi-dimensional range search



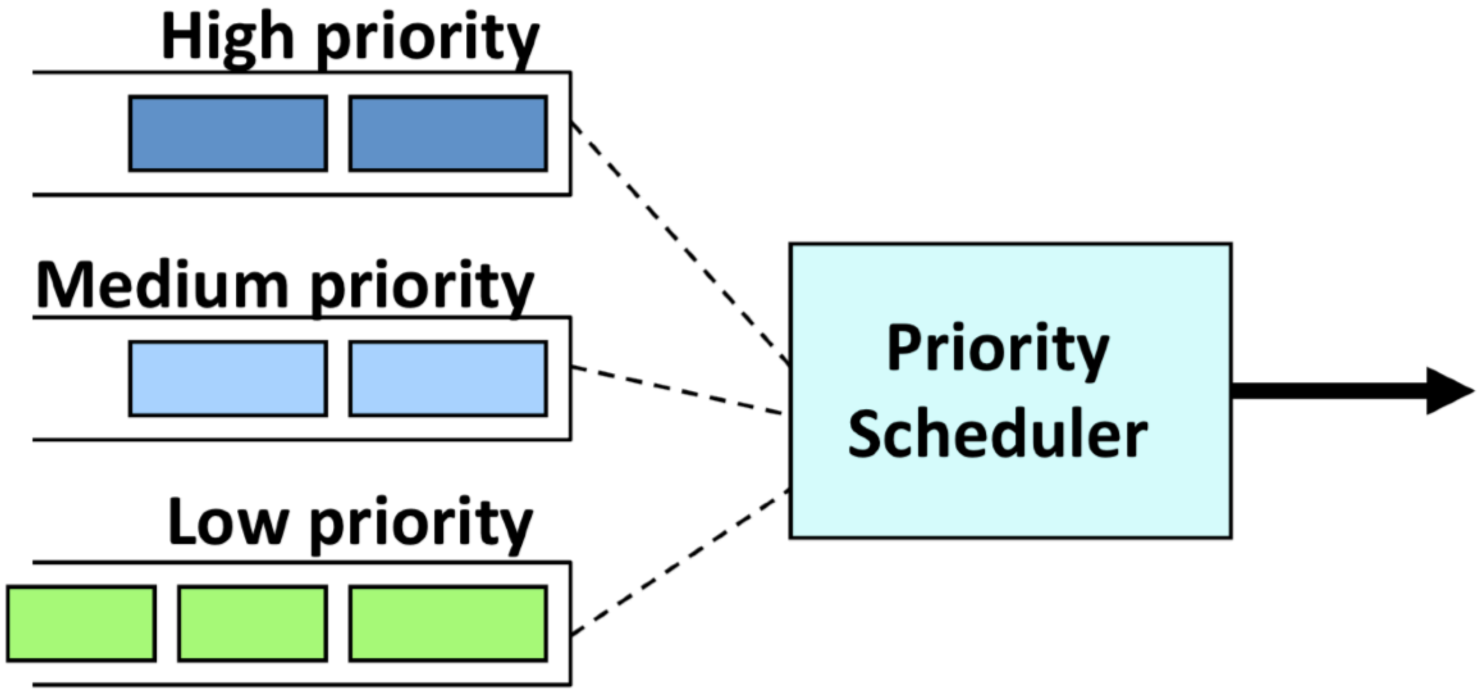
Scheduler

- One queue per flow
- Scheduler decides from which queue to send a packet
- Goals of scheduling algorithm
  - Fast!
  - Depends on the policy being implemented (fairness, priority, etc.)



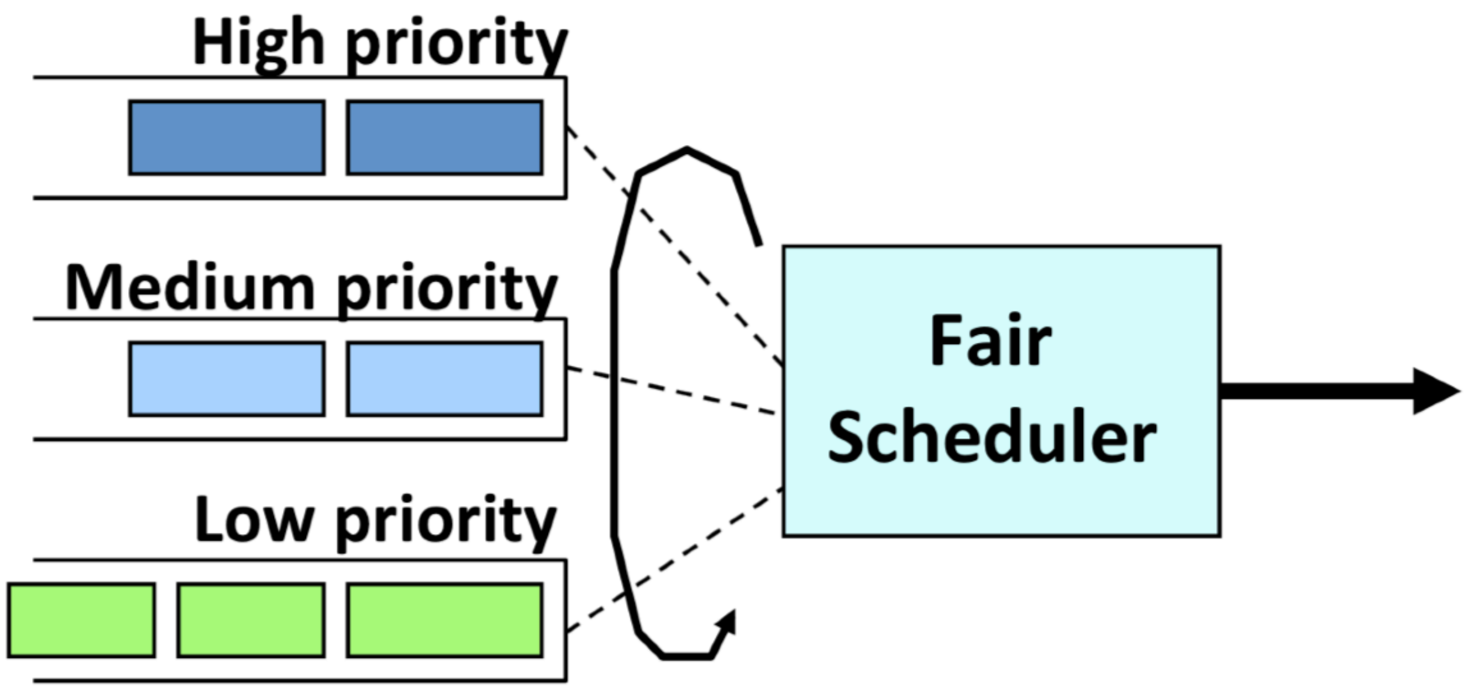
Example: Priority Scheduler

- Packets in the highest priority queue are always severed before the packets in the lower priority queues



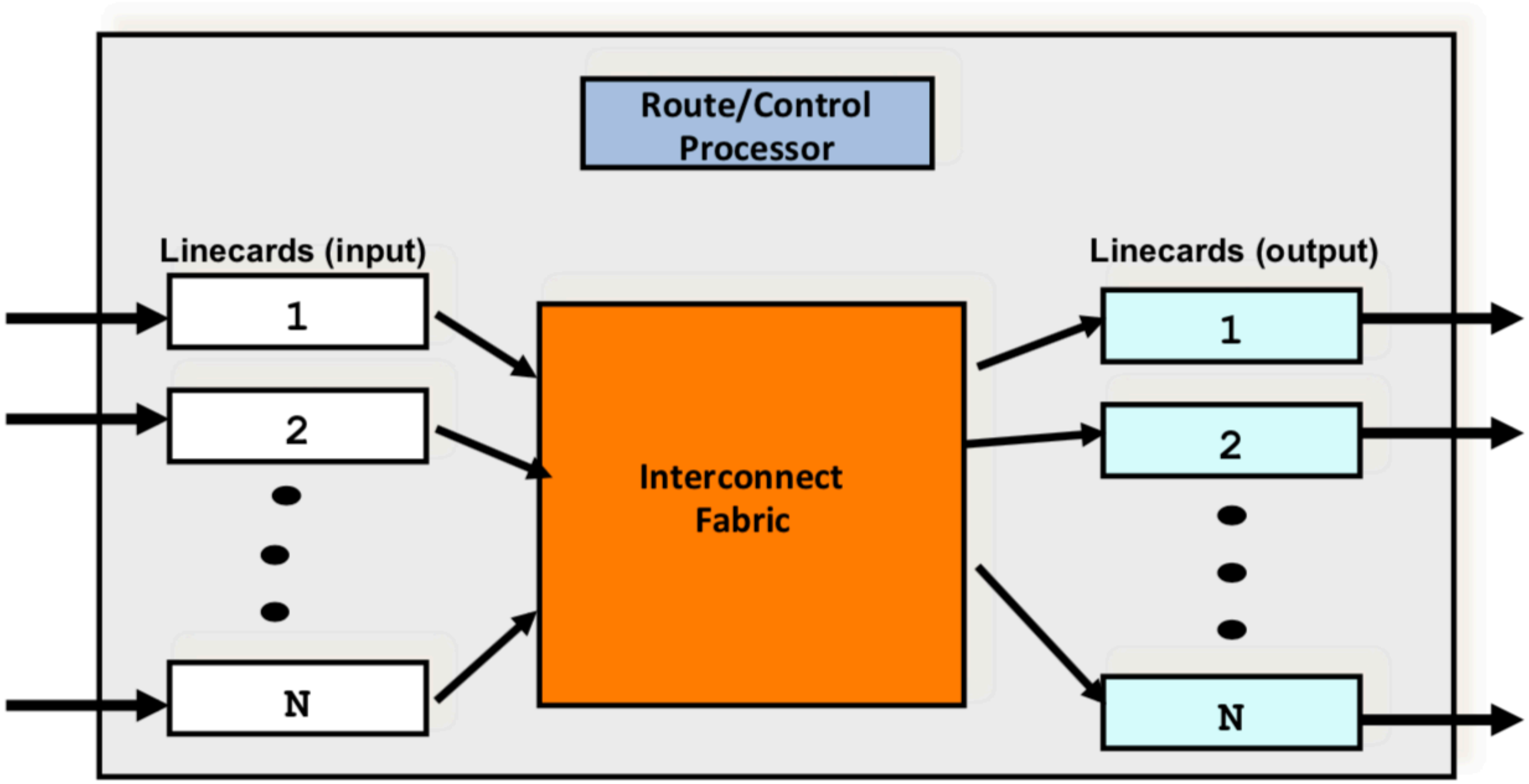
Example: Round Robin Scheduler

- Packets are served from each queue in turn

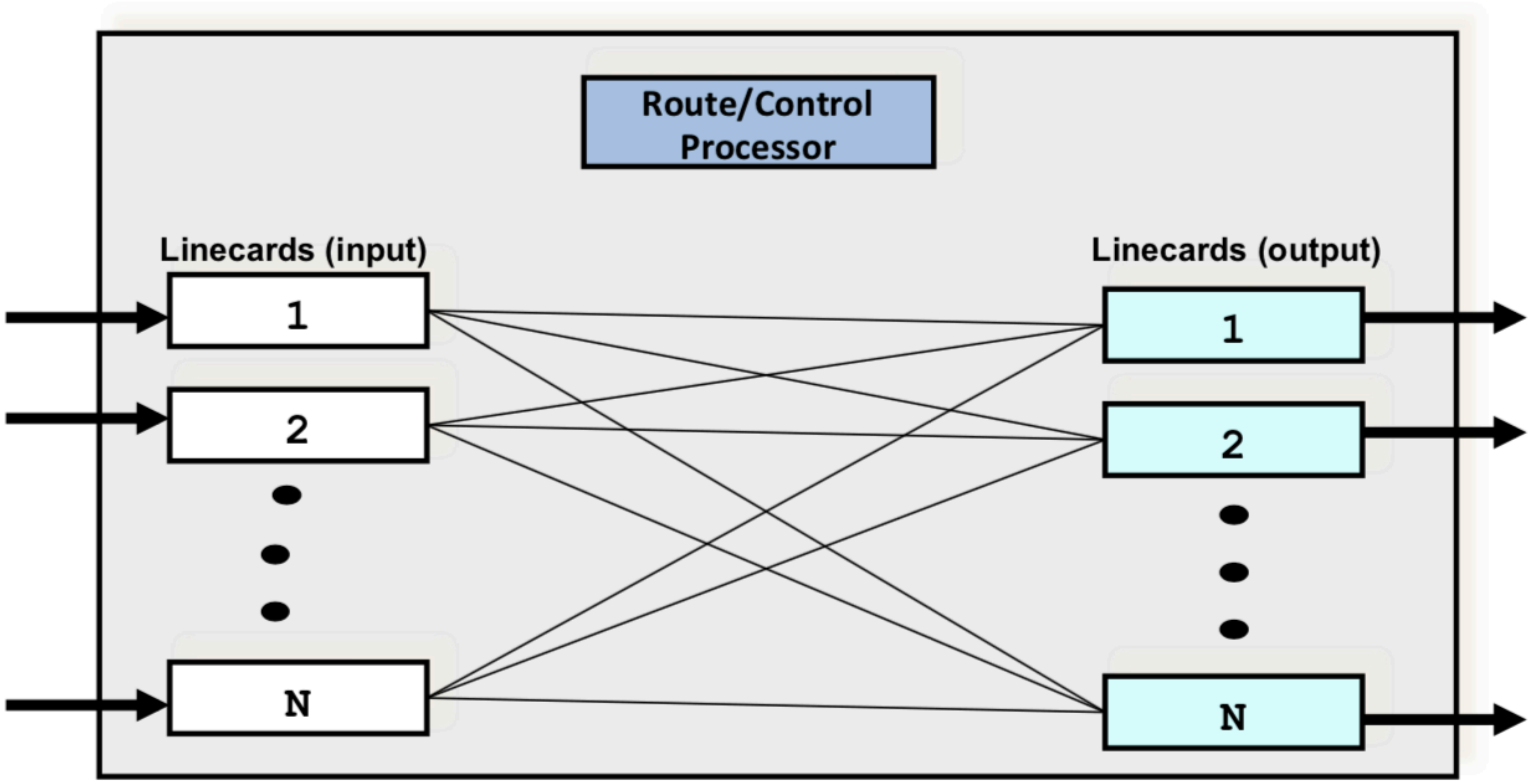


Connecting Input to Output: Switch Fabric

- Priority Scheduler: packets are served from each queue in turn



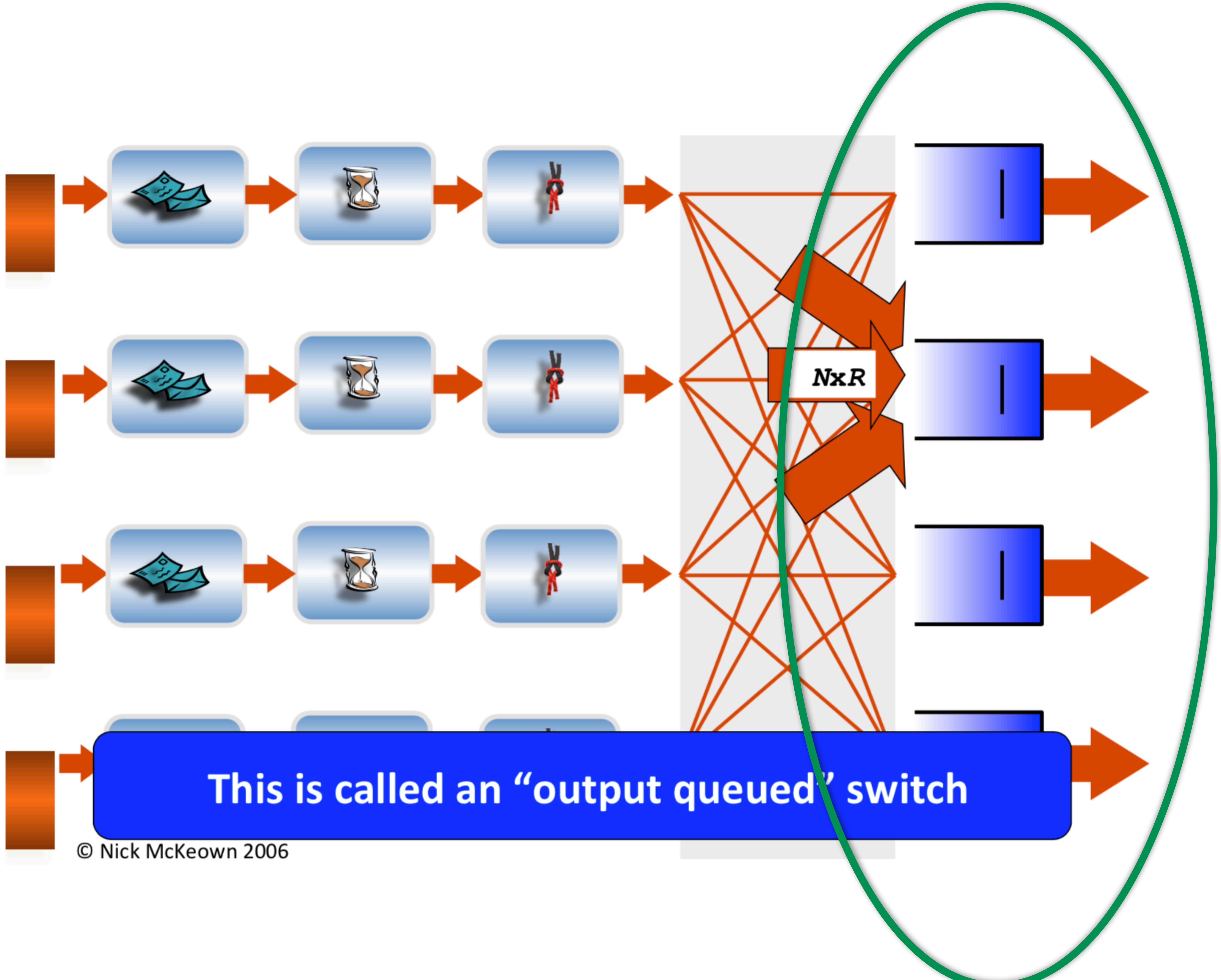
Today's Switch Fabrics: Mini Network!



What's Hard About the Switch Fabric?

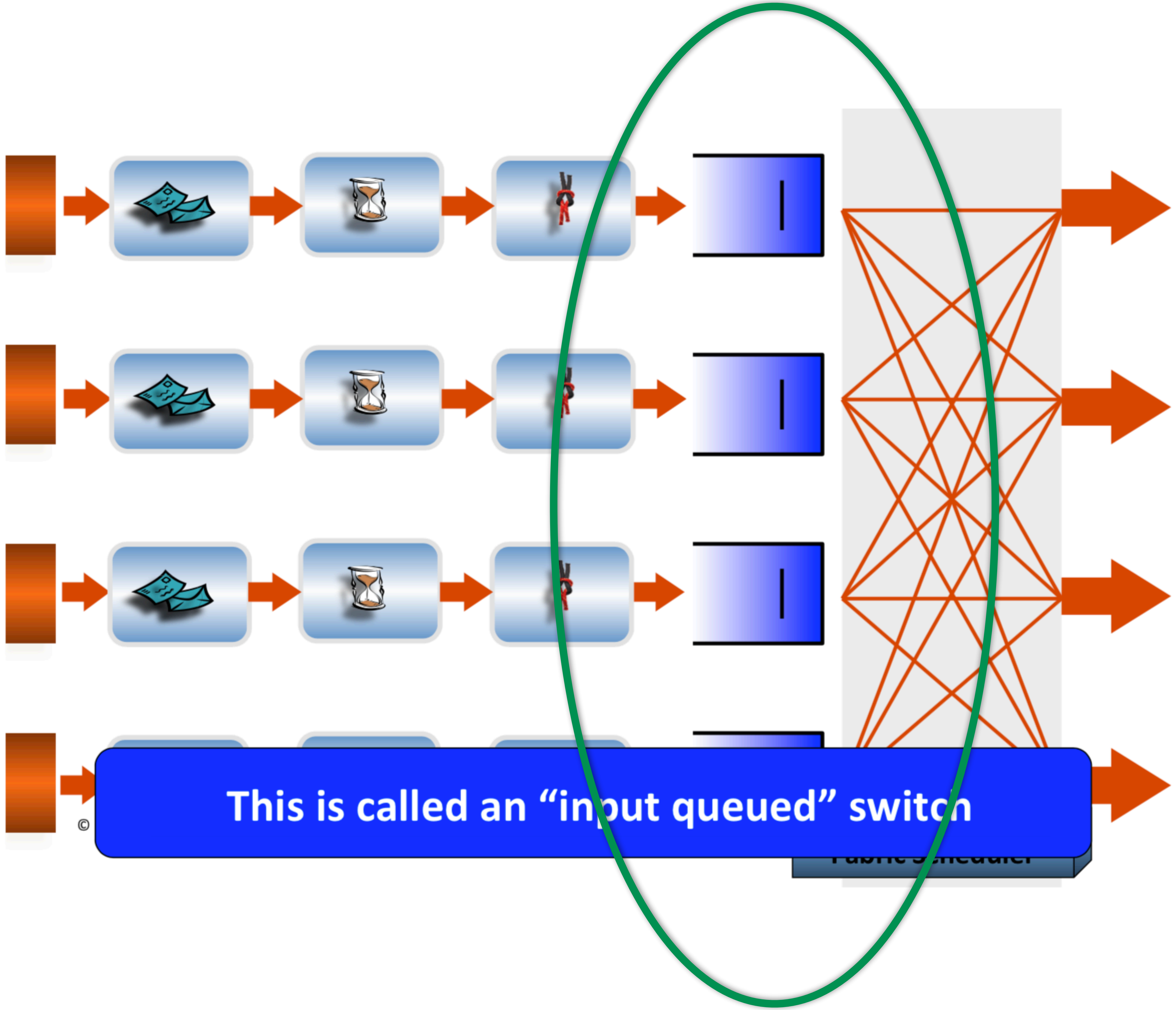
Queueing!

Third Generation Router: Switched Interconnects



This is called an "output queued" switch

Third Generation Router: Switched Interconnects



Reality is More Complicated

- Commercial high-speed routers use
    - Combination of input and output queueing
    - Complex multi-stage “topologies”
    - Distributed multi-stage schedulers (for scalability)

IP Routers Recap

- Core building block of Internet infrastructure
- Scalable Routing -> Longest Prefix Matching
- Need fast implementations for
    - Longest prefix matching
    - Switch fabric scheduling