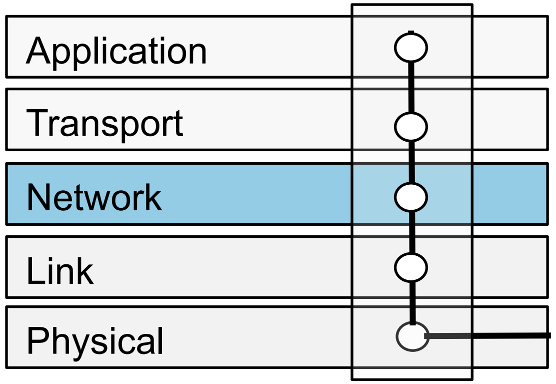
Network Layer | Inside a Router

To do ...

- Data and control planes
- Routers internals

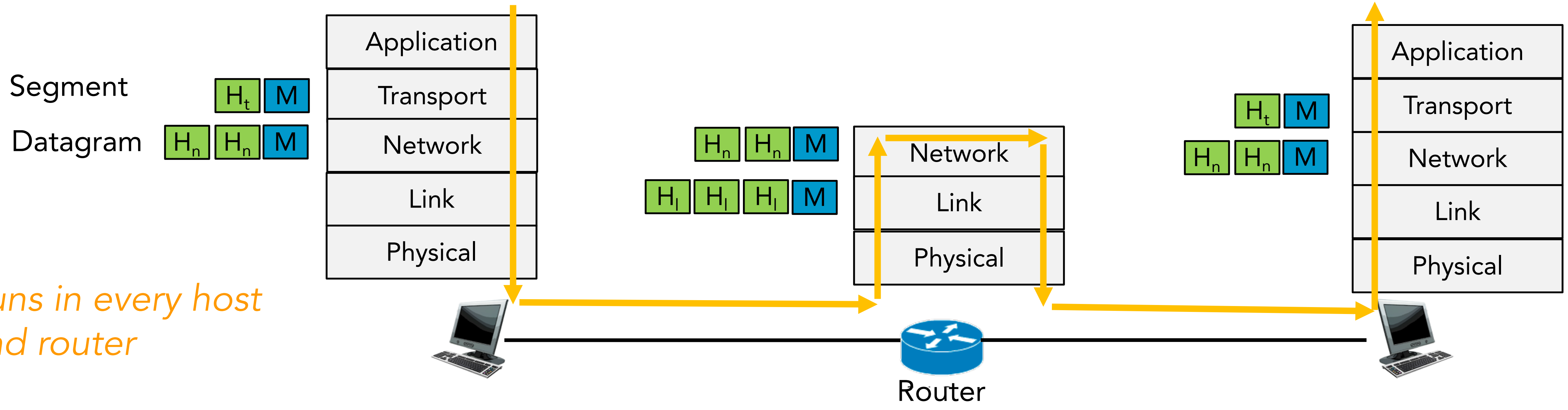


Northwestern

Slides partially based on J. Kurose and K. Ross, "Computer Networking: A Top Down Approach", 7Ed., 2016

Network layer

- Primary goal – to transport segment from end-to-end
  - On sending side, encapsulates segments into datagrams and send them to next router
  - On receiving side, get datagrams from upstream router and delivers segments to transport layer
  - Router examines header fields in all IP datagrams passing through it



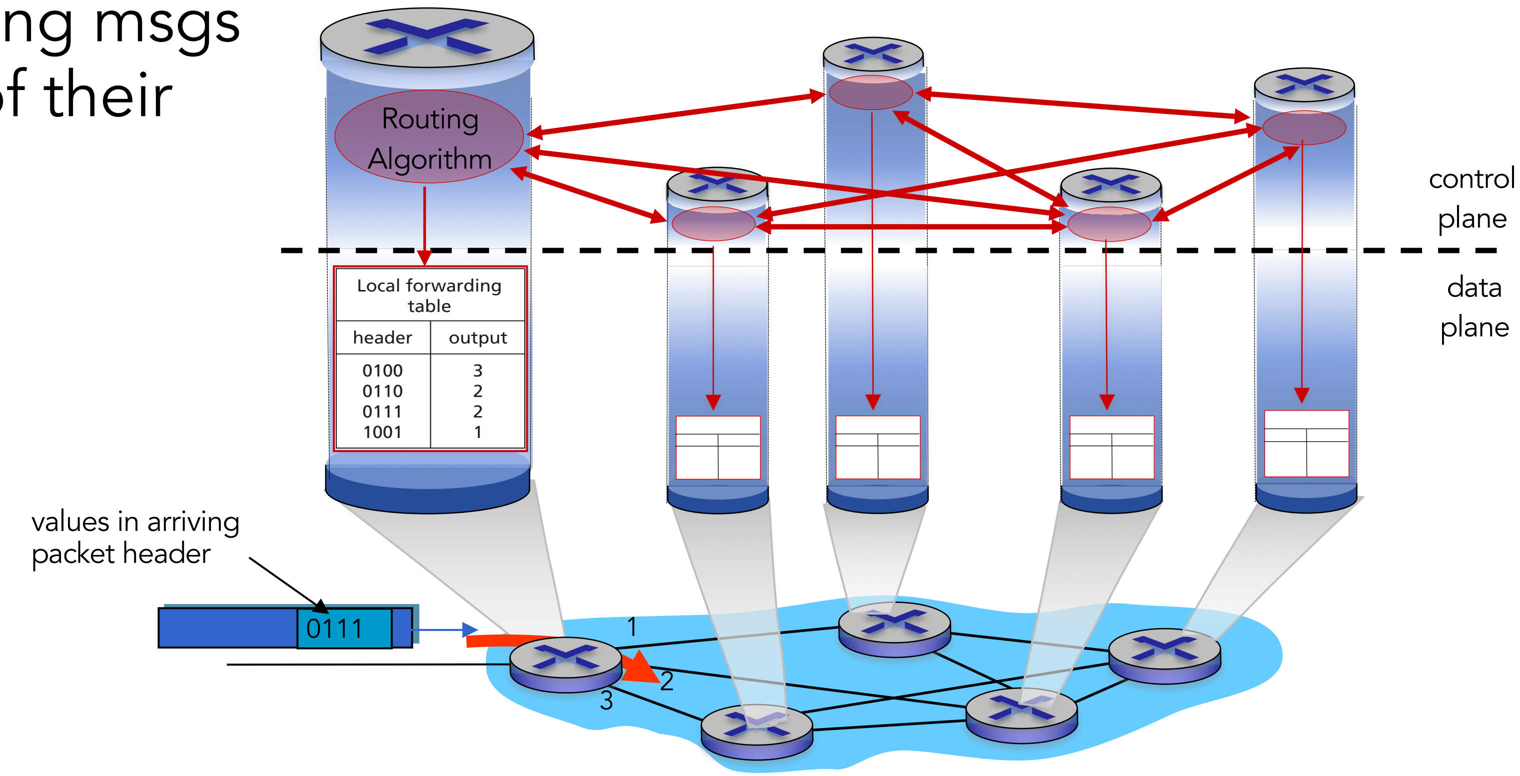
Runs in every host and router

Two network-layer functions

- Forwarding – Move packets from router's input to appropriate router output, a key function of the "data plane"
  - Router-local action, takes places in short timescales (nanosecs), in HW
  - *~Taking a trip, the process of getting through a single interchange*
- Routing – Determine the route taken by packets from src to dest
  - Routing algorithms
  - Network-wide process, longtime scales (secs), typically in SW
  - … *Process of planning trip from origin to destination*
- Key to routers' function – a *forwarding table*
  - Using value in an arriving packet header, index into the table to decide how to forward

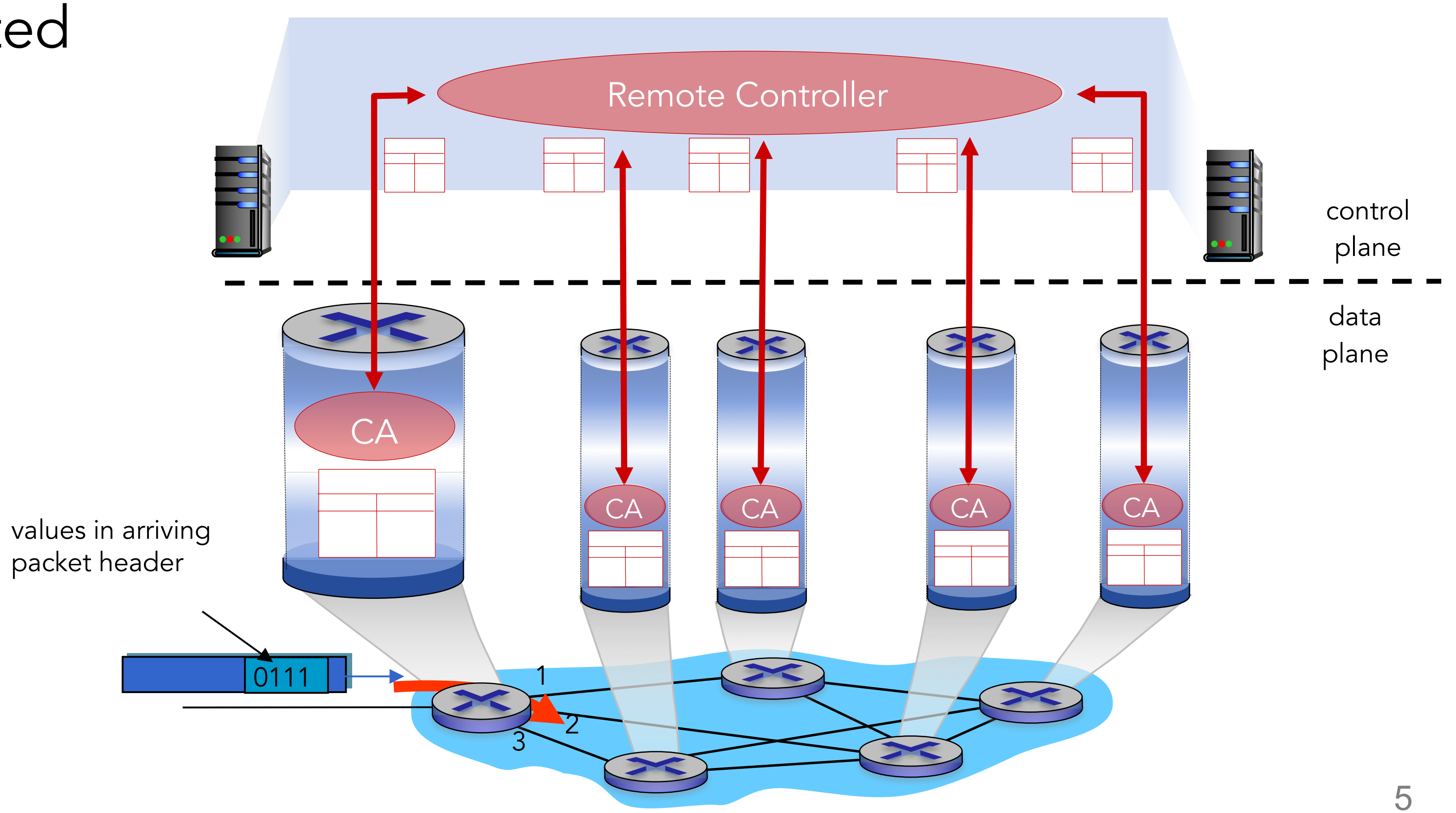
Control plane

- So how are forwarding tables set?
  - The routing algorithm determines the content
- Traditionally
  - Routing algorithm runs in every router
  - Routers exchange routing msgs to compute the value of their forwarding table



An alternative – Software-Defined Networking (SDN)

- Distinct (typically remote) controller interacts with local control agents (CAs)
- Remote controller
  - Computes and distributed the forwarding tables
  - Maybe implemented in a data center
  - Increasingly "open" implementations
- Note the data plane remains the same



Network service model

- What service model for "channel" transporting datagrams from sender to receiver? Some possible services …
  - Guaranteed delivery
  - Guaranteed delivery with bounded delay
  - In-order packet delivery (flow)
  - Guaranteed minimum bandwidth
  - Security
  - …
- Internet's network layer
  - Best-effort service, *so none of the above*
  - *Still, with adequate bandwidth seems good enough for now*

Before looking at the per-router functions

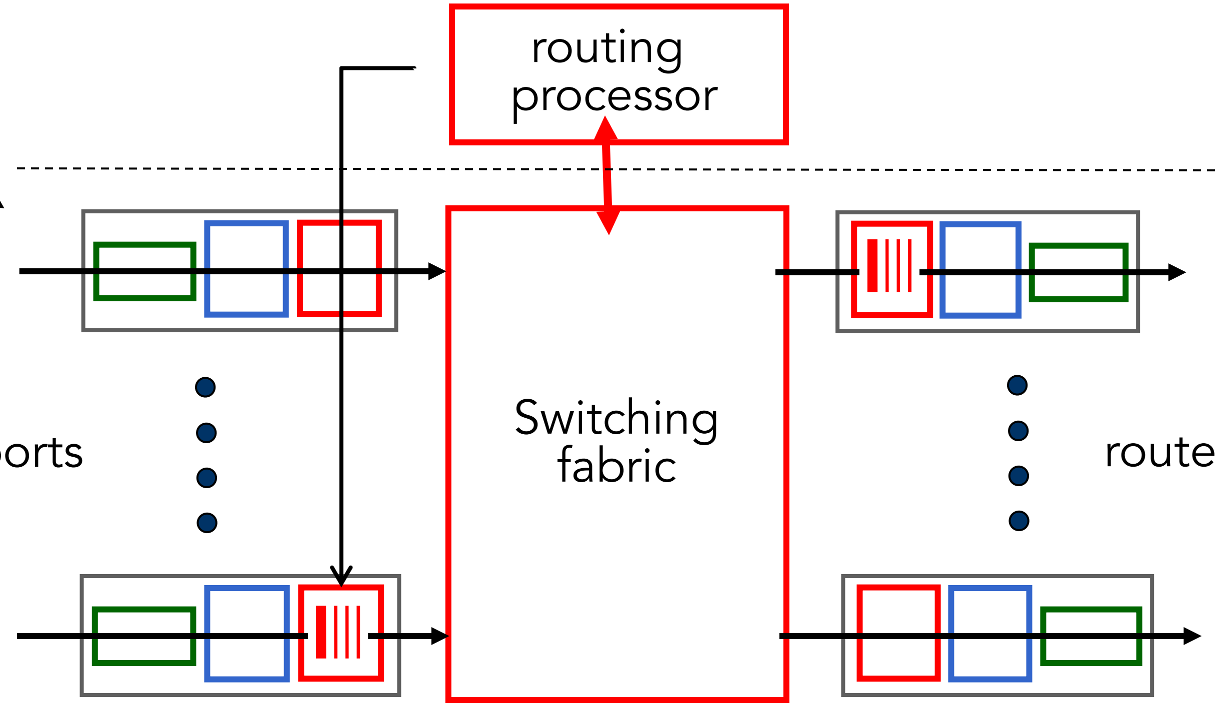
A quick look at routers …

Router architecture overview – A high-level view

Four main components

In traditional routers, executes routing protocol, maintains forwarding table
In SDN, talks to controller to receive forwarding table entries and install entries in router's input ports

Terminates the physical link and does the lookup in the routing table



routing, management control plane (software)

forwarding data plane

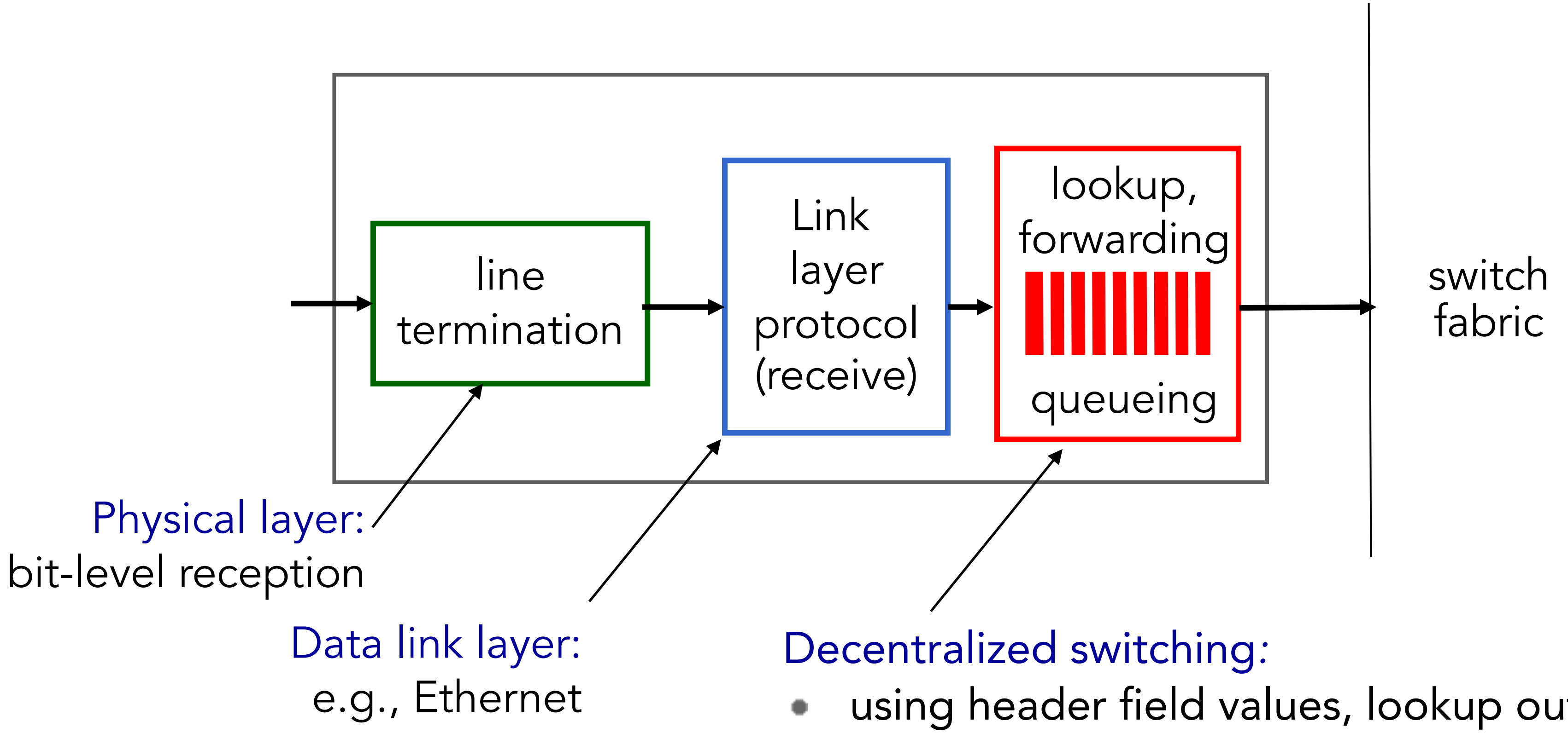
router input ports

router output ports

Connects input to output ports

Stores packets from the switching fabric and forwards them on the outgoing links

Input port functions in more detail



Decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Forwarding table entries updated by routing processor over a separate bus (e.g., a PCI bus), keeping a shadow copy for local processing

Destination-based forwarding

- A naïve, brute-force, one entry/address doesn't work with 4 billion entries
- Ranges to handle scale

Destination Address Range	Link i/f
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
Otherwise	3

- But we don't need all that either

Longest prefix matching

- When looking for forwarding table entry for given destination address, use longest address prefix that matches destination

Which interface for this destination?

11001000 00010111 00010110 10100001

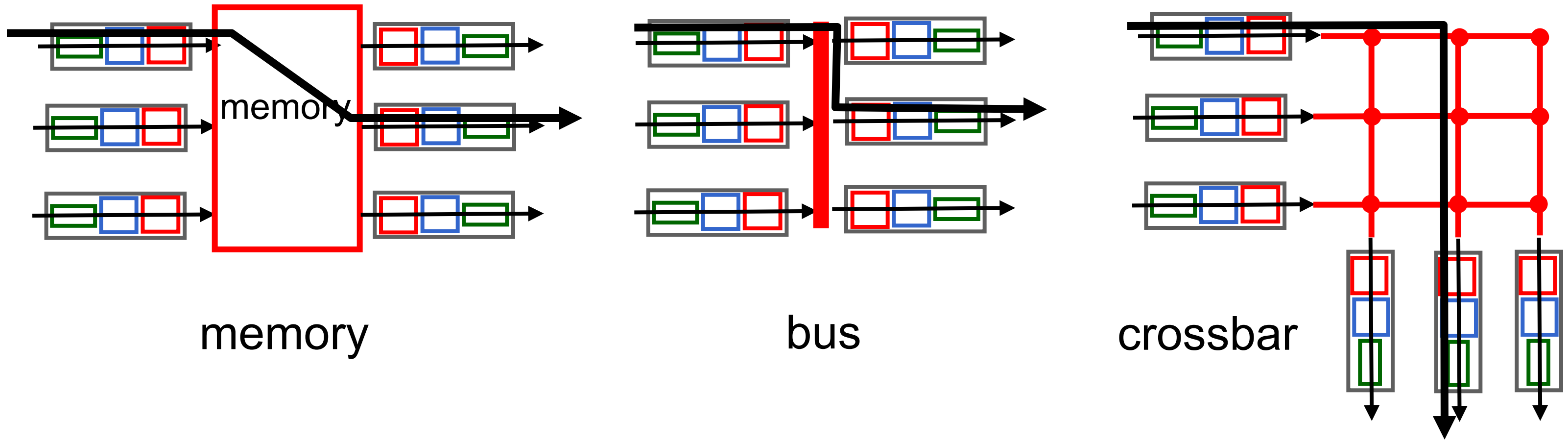
11001000 00010111 00011000 10101010

Destination Address Range	Link i/f
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
Otherwise	3

- Often using ternary content addressable memories (TCAMs)
  - Present address to TCAM: retrieve address in one clock cycle, regardless of table size (e.g., Cisco Catalyst, up ~1M routing table entries in TCAM)

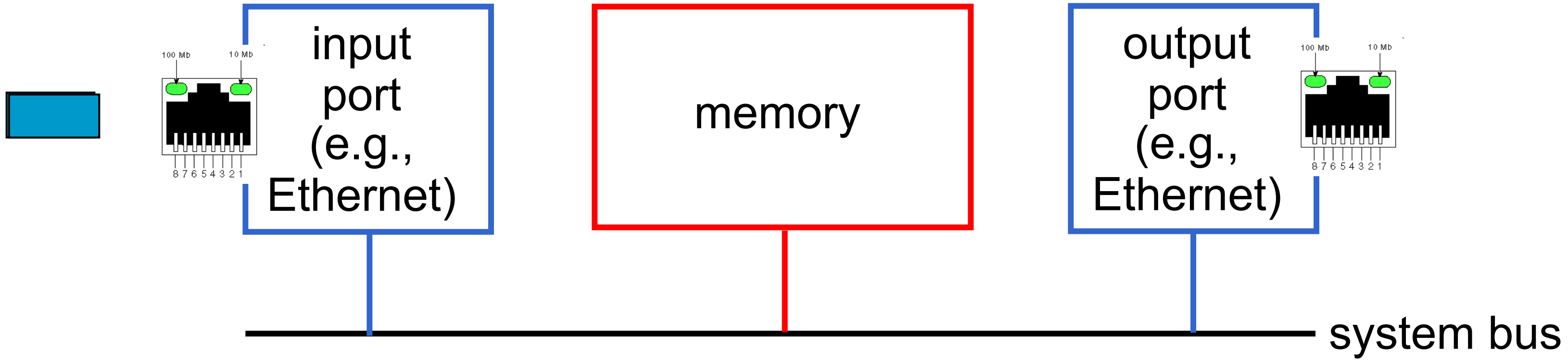
Switching fabrics

- Transfer packet from input buffer to appropriate output buffer
- Switching rate – rate at which packets can be transferred
  - Often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- Three types of switching fabrics



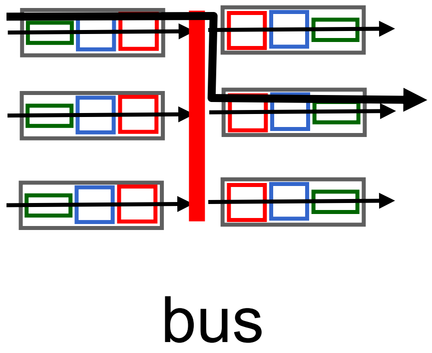
Switching via memory

- Simplest and earliest routers
- Traditional computers with switching under direct CPU control
  - Input/output ports functioned like traditional I/O devices
  - Packet copied to system's memory
  - Speed limited by (half of) memory bandwidth
- Some modern routers adopt it, but processing in the input line
  - Basically a shared memory multiprocessor with processing on a line switching/writing into the memory of an output port



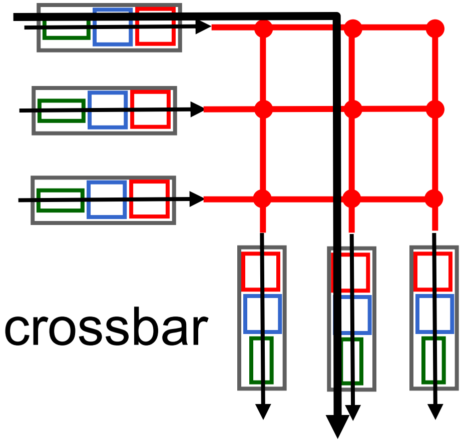
Switching via a bus

- Input port transfer packets directly into output port over a shared bus
  - No processor intervention
  - Input port pre-pend a switch-internal label to the packet
  - All ports get it, only the target one keeps it and removes the label
- Bus contention – switching speed limited by bus bandwidth
- Fast enough for local area and enterprise networks
  - Cisco 6500 @ 32 Gbps



Switching via interconnection network

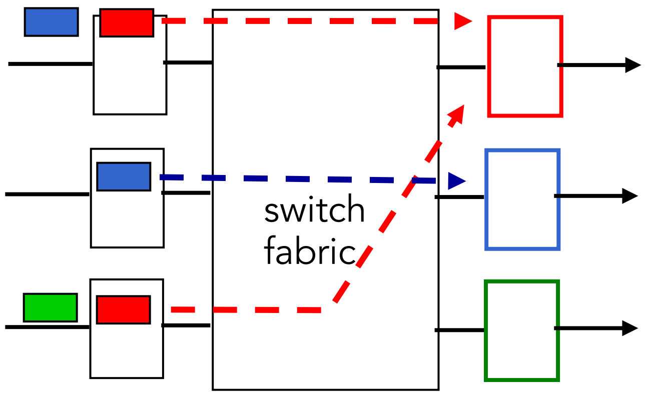
- Overcome bus bandwidth limitations, different, more sophisticated interconnection networks (from multiprocessors)
  - Banyan networks, crossbar, …
- Advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000 @ 60 Gbps

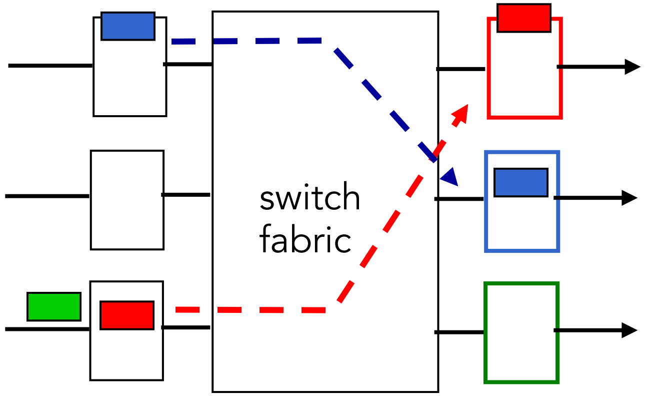


Input port queuing

- Queues may form at input or output ports
- Fabric slower than input ports combined → queueing at input
  - Queueing delay and loss due to input buffer overflow
- And not just the second red packet waits, queued datagram at front of queue prevents others in queue from moving forward - Head-of-the-Line (HOL) blocking

output port contention: only one red datagram can be transferred; *lower red packet is blocked*



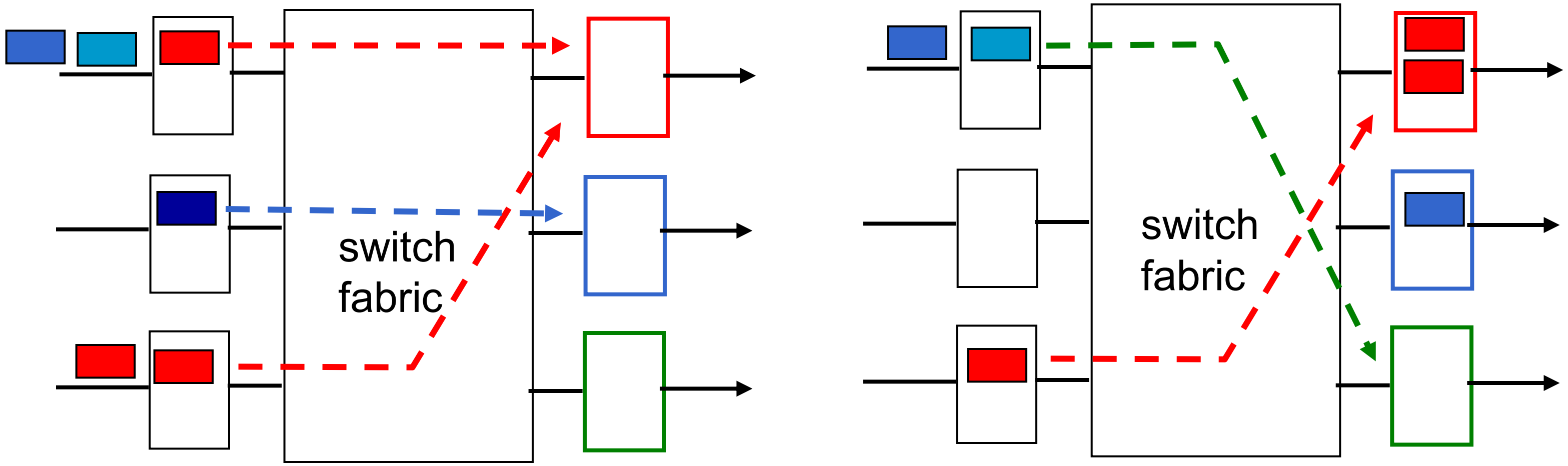


one packet time later: green packet experiences HOL blocking even if there is no contention for the green port

Output port queueing

- Queueing when datagrams arrive from fabric faster than the transmission rate (output line speed)
  - Queueing (delay) and loss due to output port buffer overflow
  - Drop tail (last arrived packet) or make room for it
  - Maybe drop proactively (before it is full)? Active Queue Management (AQM) algorithms (e.g., Random Early Detection, RED)

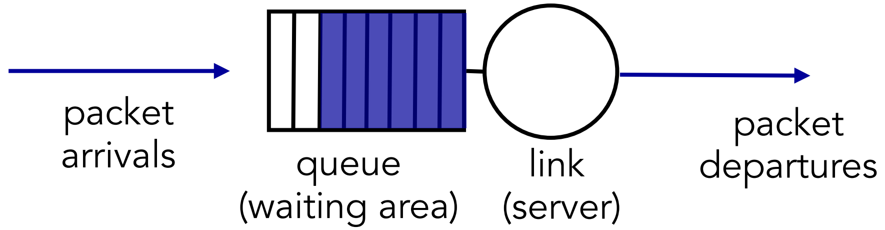
at *t*, packets move from input to output

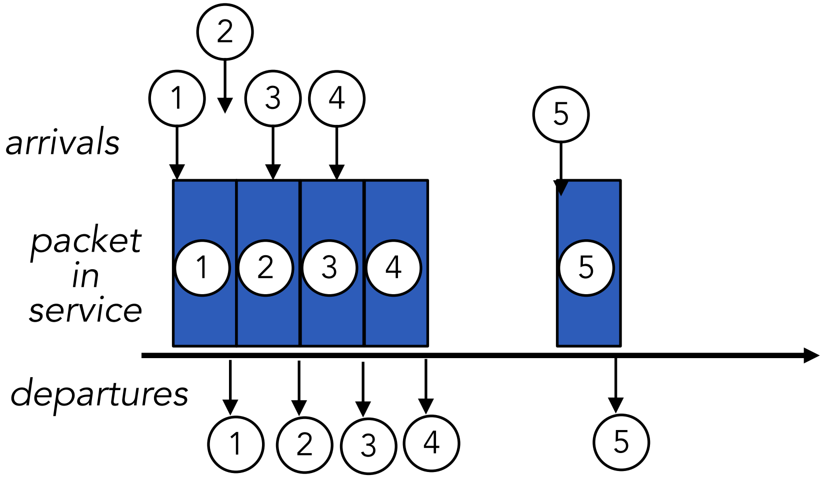


one packet time later

Scheduling mechanisms

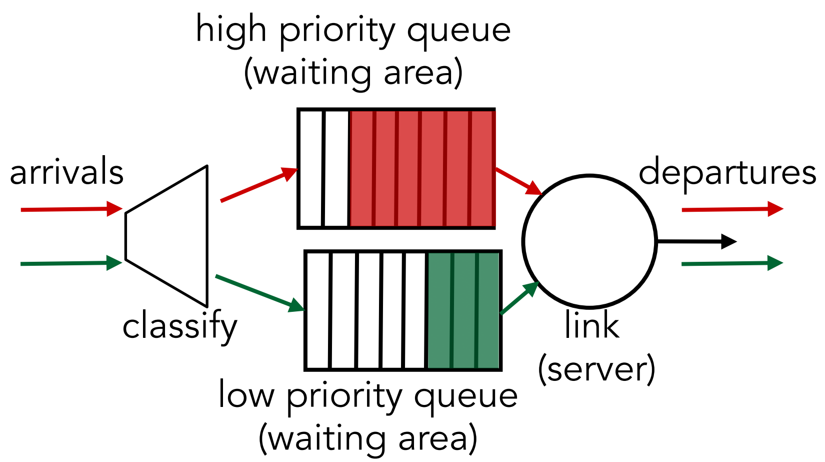
- Queued datagrams means you can choose what to send …
- FIFO scheduling: send in order of arrival to queue
  - Discard policy: if packet arrives to full queue: which one to discard?
    - tail drop: drop arriving packet
    - priority: drop/remove on priority basis
    - random: drop/remove randomly

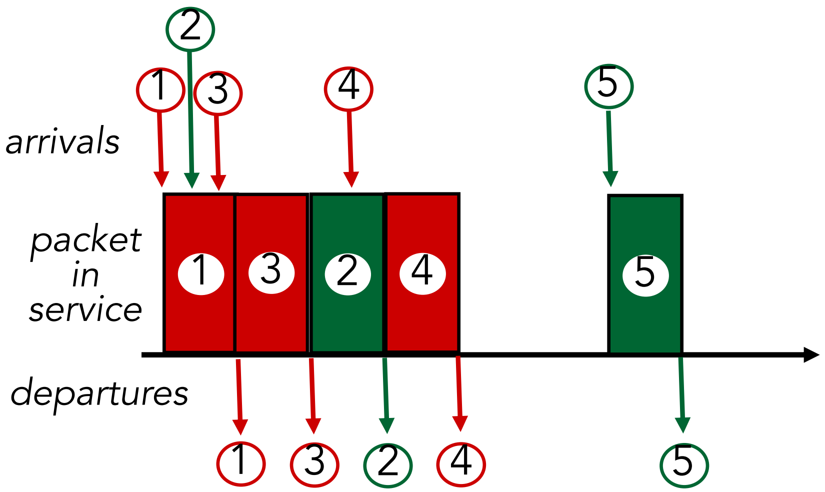




Scheduling policies: Priority

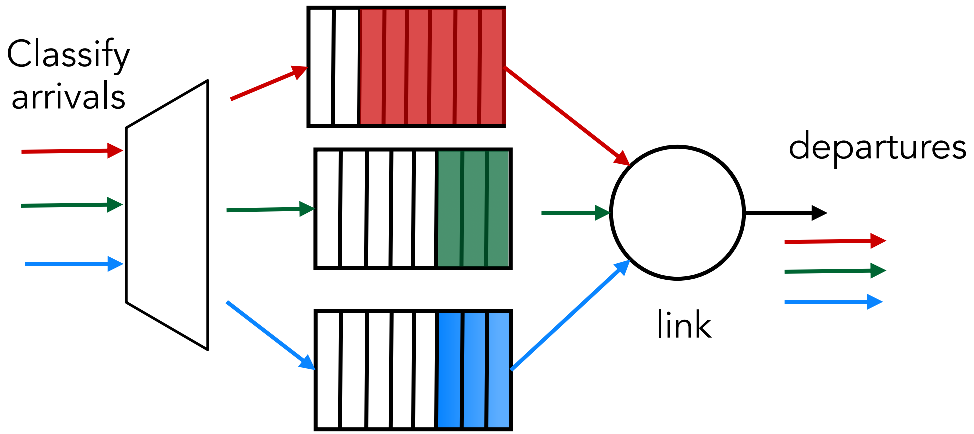
- Multiple classes, with different priorities
  - Class may depend on marking or other header info, e.g. IP srce/dst, port numbers, …
- Send highest priority queued packet
- Non-preemptive – the transmission of a packet is not interrupted by the arrival of a higher-priority one





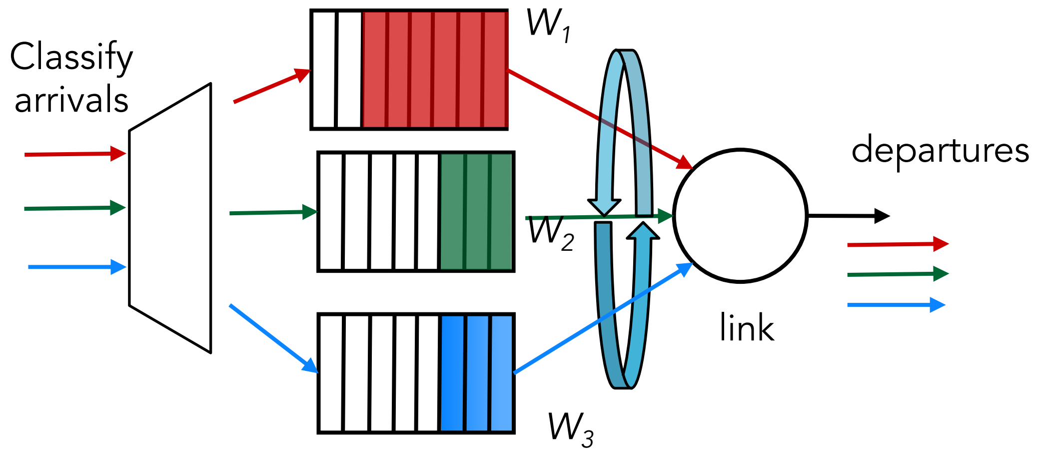
Scheduling policies: Round robin

- Multiple classes
- Cyclically scan class queues, sending one complete packet from each class (if available)
  - *Work-conserving queueing discipline* – don't go idle if there are packets of any class, if a queue is empty, check next class



Scheduling policies: Weighted Fair Queuing (WFQ)

- Generalized Round Robin
- Each class gets weighted amount of service in each cycle
  - Also work-conserving



class *i* will get a fraction of service equal to $w_i / \Sigma w_j$ where the sum is over all classes with packets queued for transmission

Recap

- Data plane, *what we have seen so far*, and control plane
  - The per-router functions that determine how packets arriving on one of the router input ports is forwarded to one of the output ports
- Routers internal, mechanisms and policies
- So far, all without reference to any specific network architecture or protocol
  - Next, we'll look at the Internet's network layer and the Internet Protocol