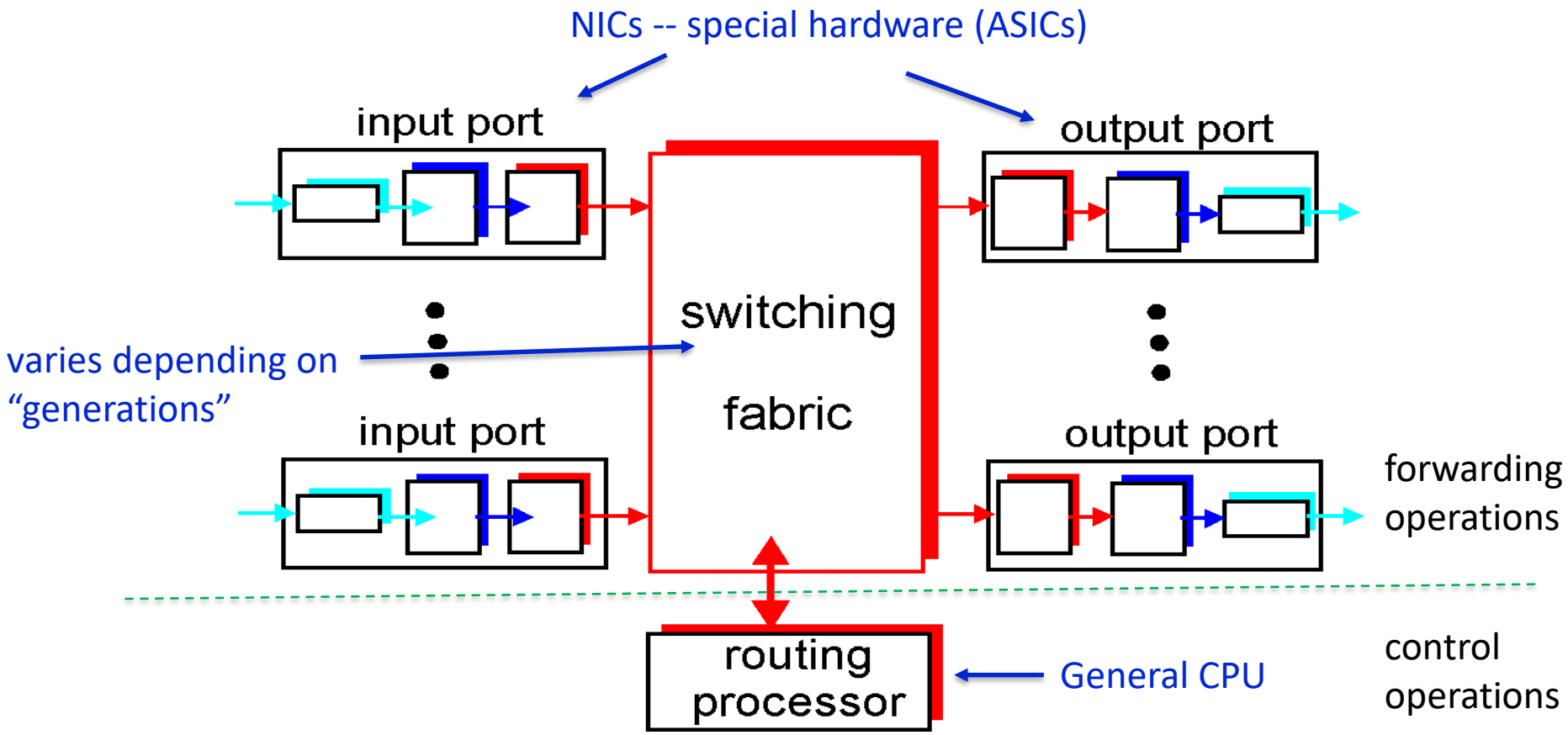
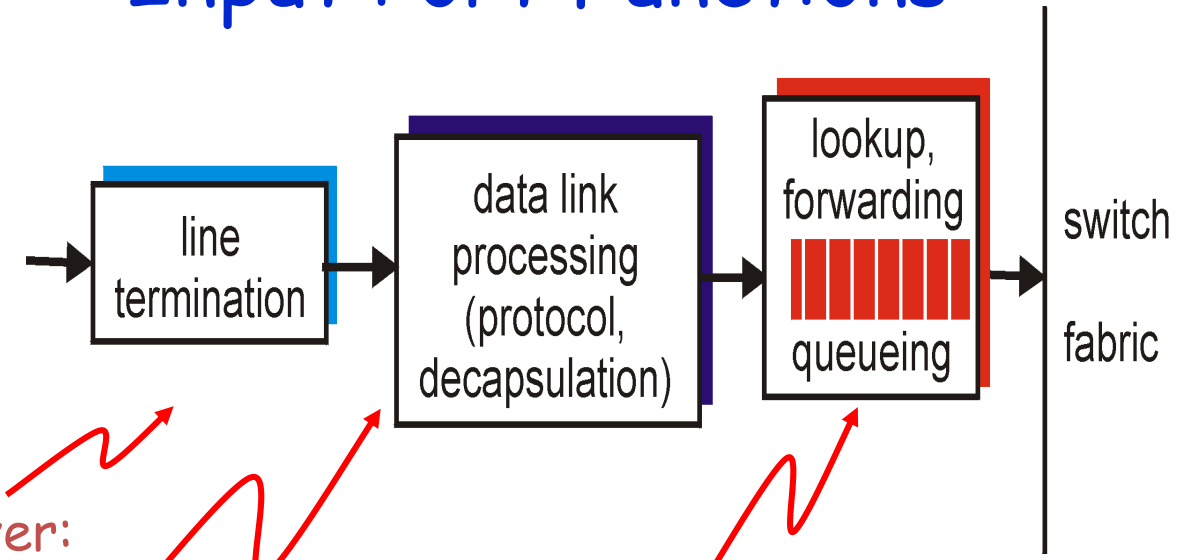
Router/Switch Architecture Overview

A key router/switch *data plane* function:

- *forwarding* packets (e.g., IP datagrams, or Ethernet frames) from incoming to outgoing link (or drop them if no forwarding table entries, or flood them if no switching table entries



Input Port Functions



Physical layer:
bit-level reception

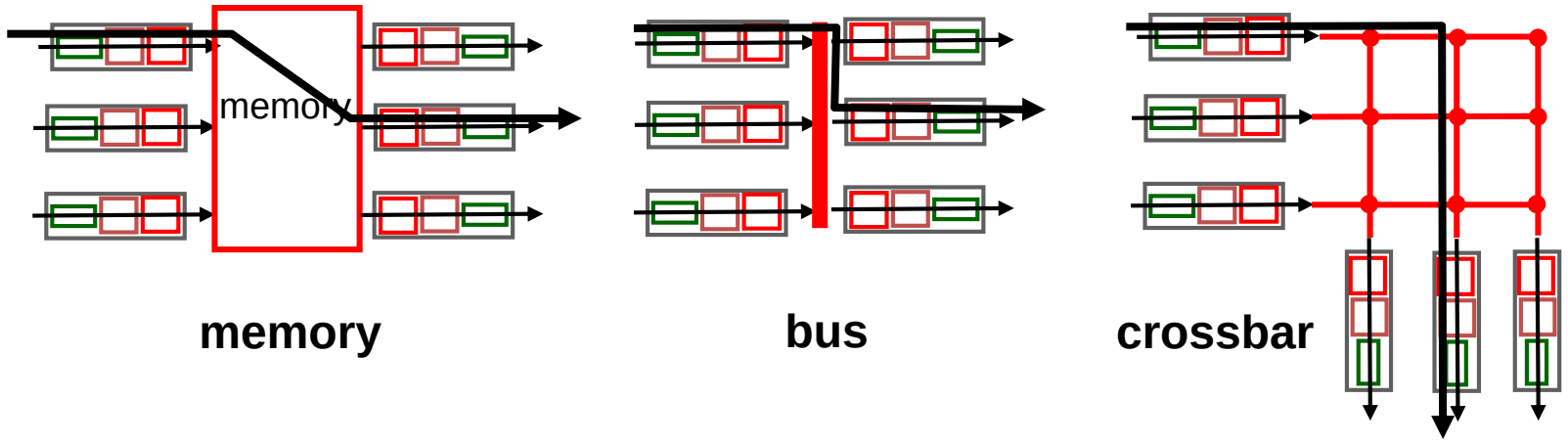
Data link layer:
e.g., Ethernet
see chapter 6

Decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Switching Fabrics

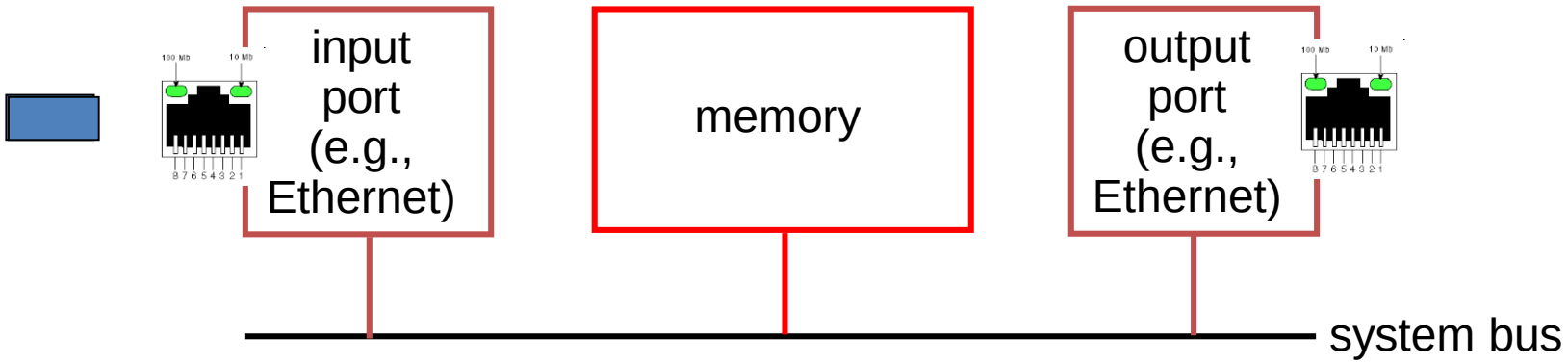
- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- three types of switching fabrics



Switching Via Memory

First generation routers:

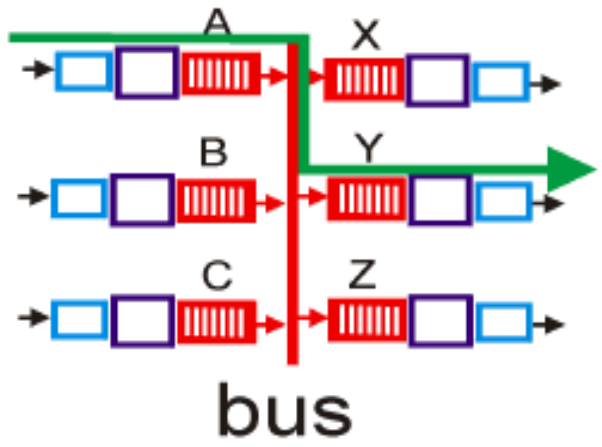
- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings per datagram)



Switching Via a Bus

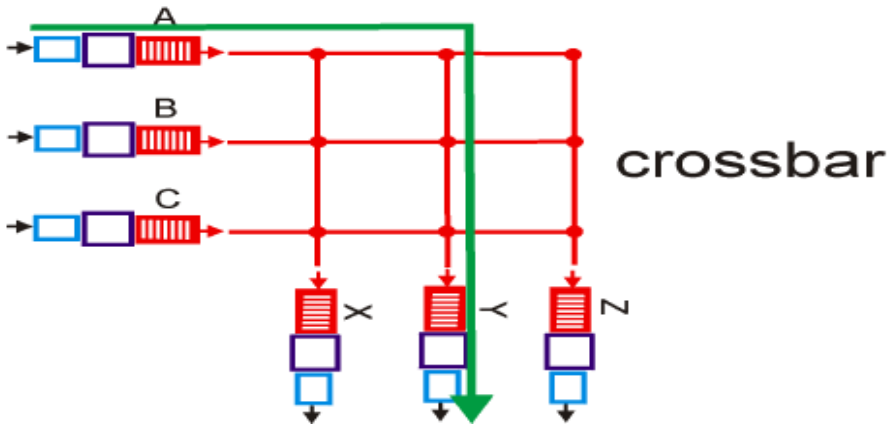
- datagram from input port memory

  to output port memory via a shared bus
- bus contention: switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers



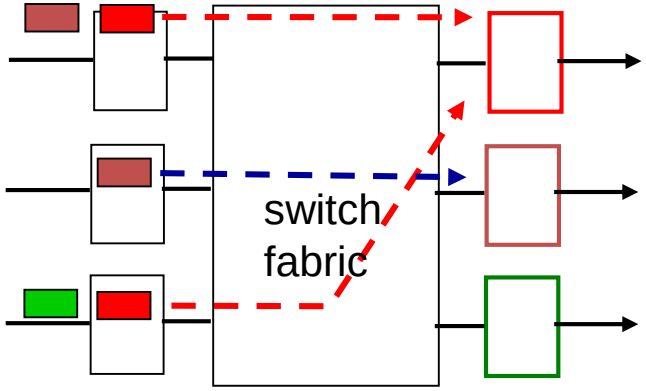
Switching Via An Interconnection Network

- overcome bus bandwidth limitations
- banyan networks, crossbar, other interconnection nets initially developed to connect processors in multiprocessor
- advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches 60 Gbps through the interconnection network

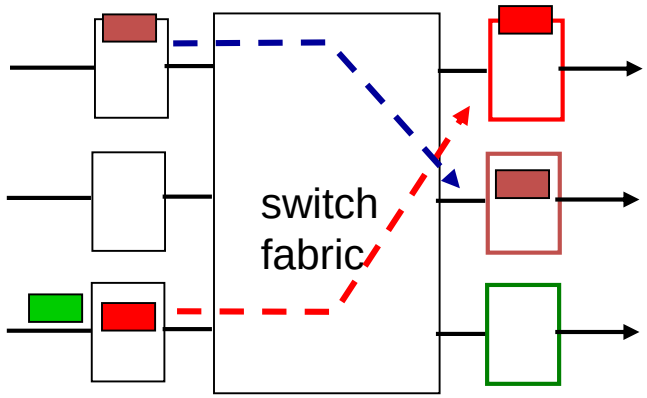


Input Port Queuing

- Fabric slower than input ports combined -> queueing may occur at input queues
  - *queueing delay and loss due to input buffer overflow!*
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward

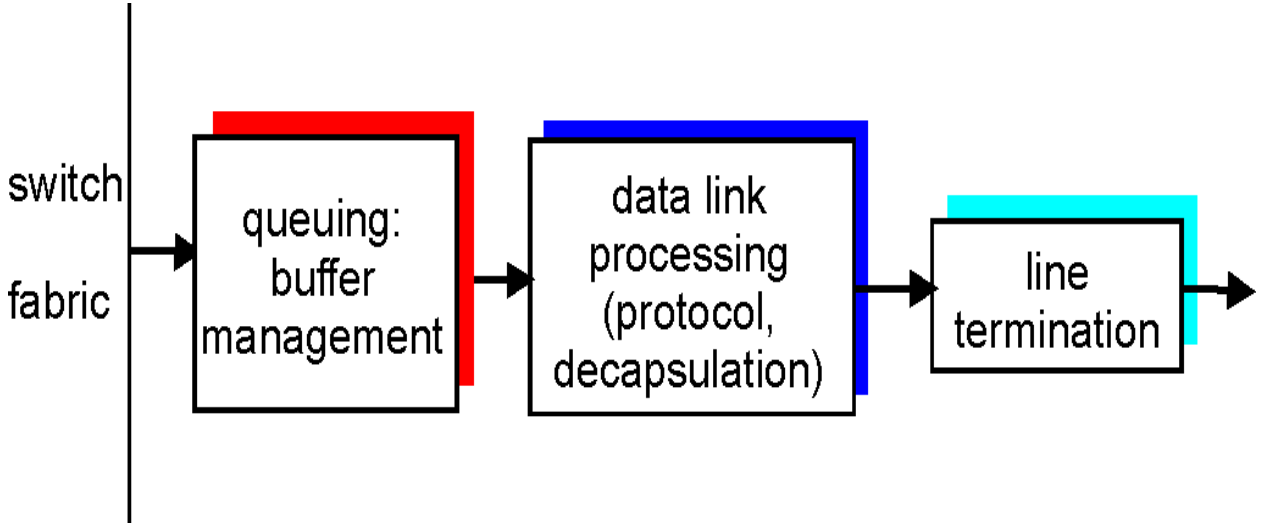


output port contention:
only one red datagram can be transferred.
lower red packet is blocked



one packet time later:
green packet
experiences HOL
blocking

Output Ports

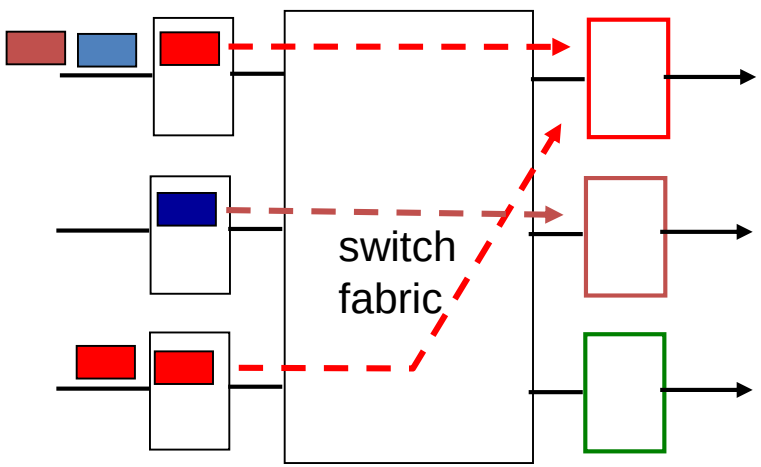


- *Buffering* required when datagrams arrive from fabric faster than the transmission rate
- *Scheduling discipline* chooses among queued datagrams for transmission

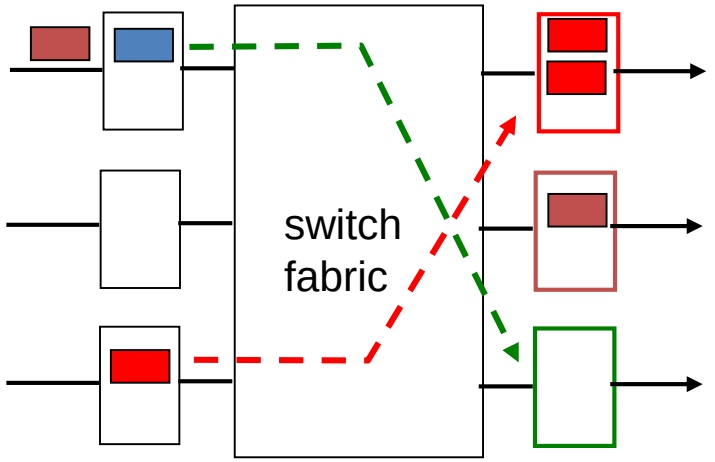
Datagram (packets) can be lost due to congestion, lack of buffers

Priority scheduling – who gets best performance, network neutrality

Output Port Queueing



at *t*, packets more from input to output



one packet time later

- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

Router/Switch Data Plane Operations: Forwarding Abstraction

We have learned:

- How IP routers forward IP datagrams based on destination IP addresses
  - take the dest. IP address from an IP packet
  - look up the IP forwarding table using the longest prefix matching
  - if there's a match, forwarding the packet to the outgoing port specified; otherwise drop it
- How layer 2 switches forward (Ethernet) frames based on destination MAC addresses
  - take the dest. MAC address from an Ethernet frame
  - look up the switch table using the exact matching
  - if there's a match, forwarding the packet to the outgoing port specified; otherwise flood it to all other ports
- What are common and what are different?
  - Common: match a header field against a table, then take an action
  - Different: how table is constructed, what header fields to match, how to match it, and what actions to take

Router/Switch Data Plane Operations: Forwarding Abstraction

We have learned:

- How IP routers forward IP datagrams based on destination IP addresses
  - take the dest. IP address from an IP packet
  - look up the IP forwarding table using the longest prefix matching
  - if there's a match, forwarding the packet to the outgoing port specified; otherwise drop it
- How layer 2 switches forward (Ethernet) frames based on destination MAC addresses
  - take the dest. MAC address from an Ethernet frame
  - look up the switch table using the exact matching
  - if there's a match, forwarding the packet to the outgoing port specified; otherwise flood it to all other ports
- What are common and what are different?
  - Common: match a header field against a table, then take an action
  - Different: how table is constructed, what header fields to match, how to match it, and what actions to take

Generalized Forwarding and SDN

Each switch contains a *flow table* that is computed and distributed by a *logically centralized* routing controller