CS 352
Network: Addressing

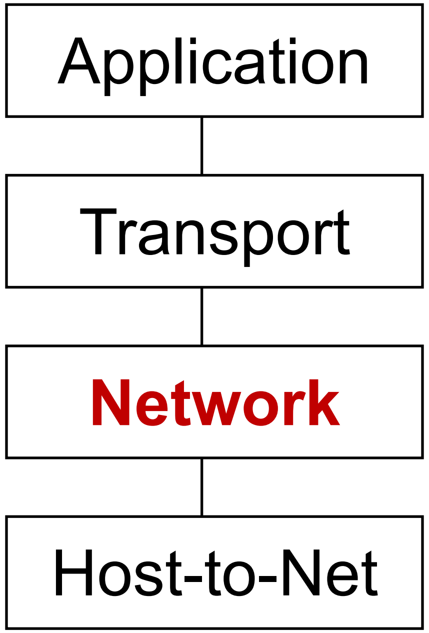
Lecture 19

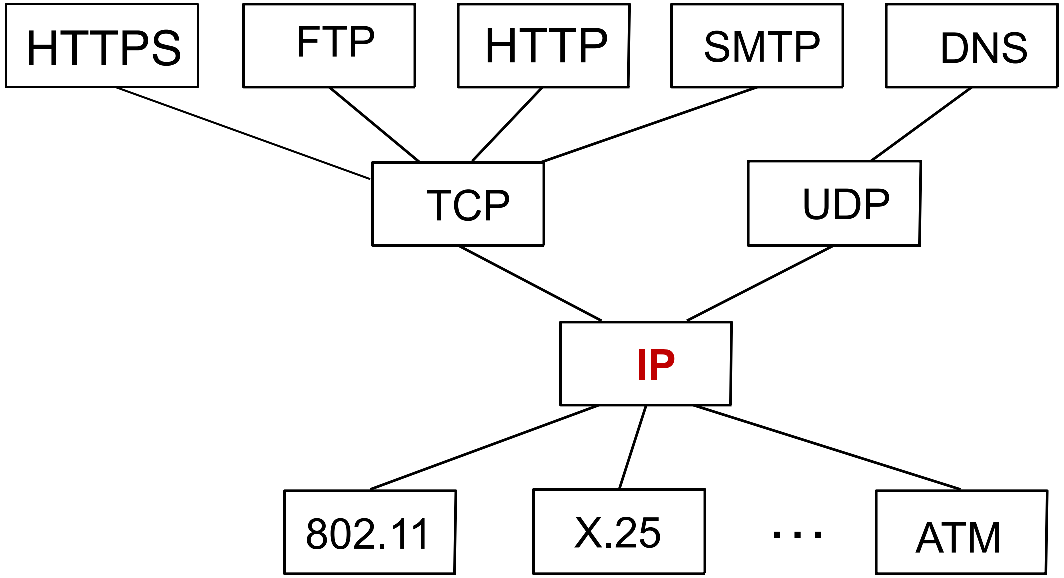
http://www.cs.rutgers.edu/~sn624/352-F22

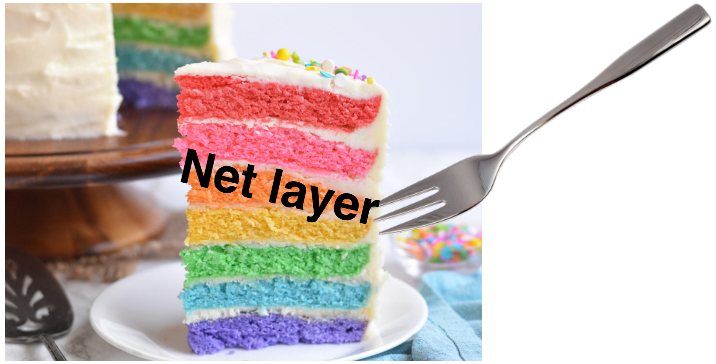
Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Network

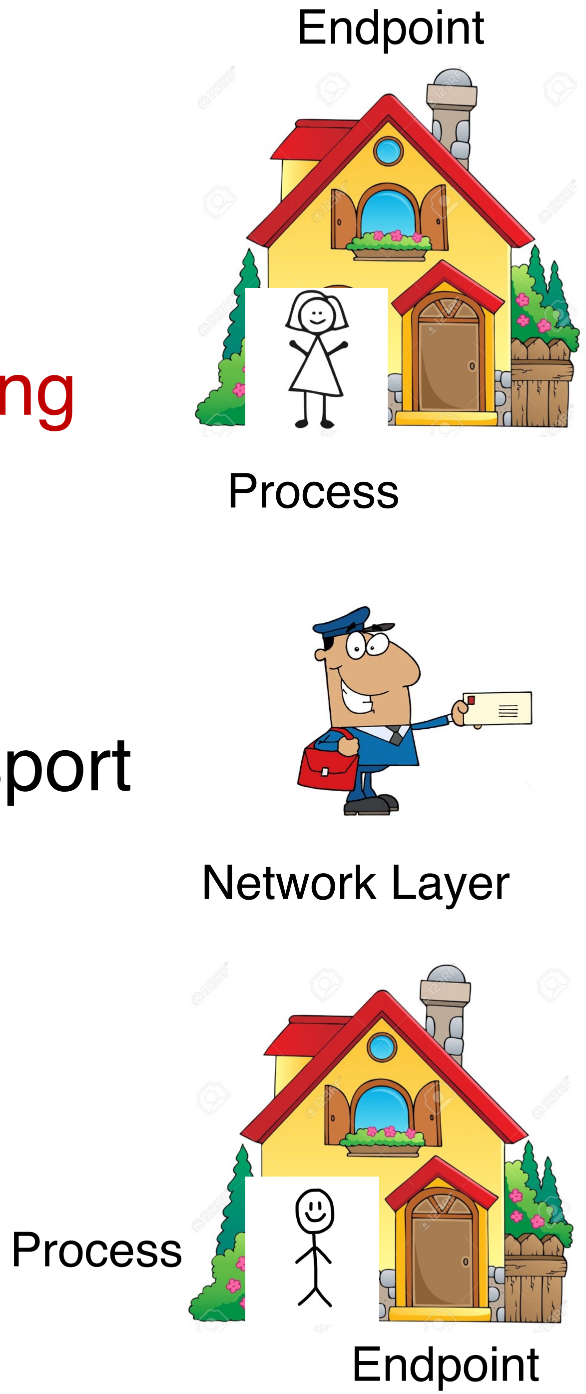

Application
Transport
Network
Host-to-Net


HTTPS
FTP
HTTP
SMTP
DNS
TCP
UDP
IP
802.11
X.25
...
ATM


Net layer

The network layer

- Main function: Move data from sending to receiving endpoint
- on sending endpoint: encapsulate transport segments into datagrams
- on receiving endpoint: deliver datagrams to transport layer
- The network layer also runs in every router
  - Very challenging to evolve the network layer
- Routers examine headers on all passing through them

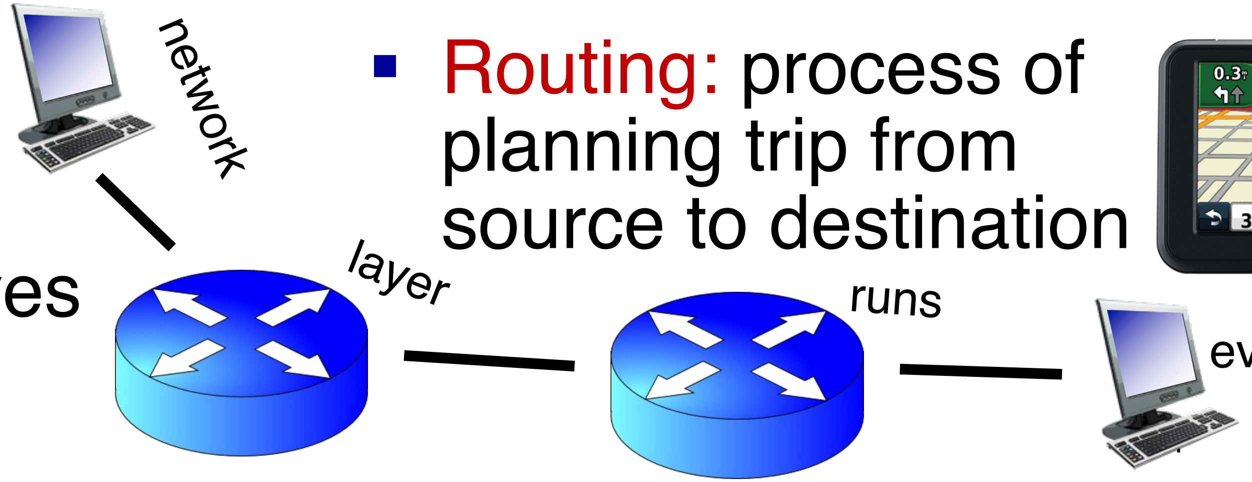


Two key network-layer functions

- Forwarding: move packets from router's input to appropriate router output

- Routing: determine route taken by packets from source to destination
  - routing algorithms

- The network layer solves the routing problem.



Analogy: taking a road trip

- Forwarding: process of getting through single interchange

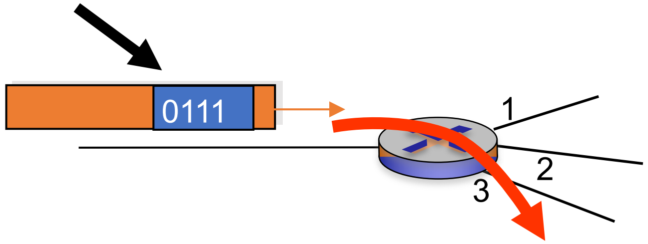
- Routing: process of planning trip from source to destination

Data plane and Control Plane

Data plane = Forwarding

- local, per-router function
- determines how datagram arriving on router input port is forwarded to router output port

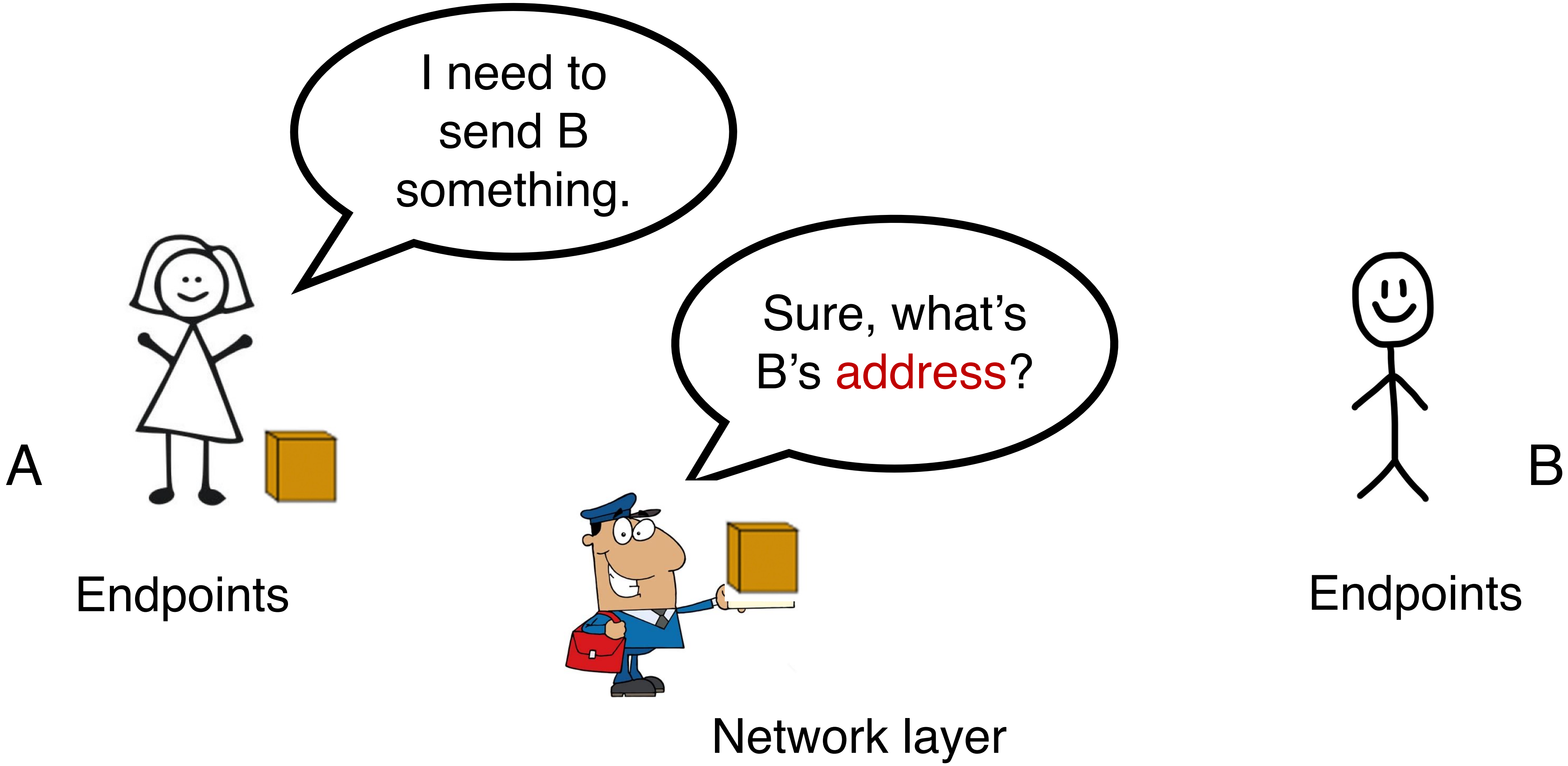
values in arriving packet header



Control plane = Routing

- network-wide logic
- determines how datagram is routed along end-to-end path from source to destination endpoint
- two control-plane approaches:
  - Distributed routing algorithm running on each router
  - Centralized routing algorithm running on a (logically) centralized server

This lecture: Addresses in the Internet

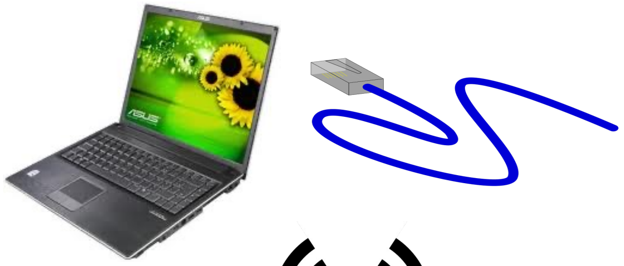


Internet Addressing

The Internet needs addresses

- Addresses are a precursor to communication.
  - Allow endpoints to locate each other
- Internet addresses are neither endpoint names nor identify them
- Addresses help routers determine how to move a packet
  - Like street address for the postal system
- Network layer addresses are designed to help routers perform the forwarding and routing functions efficiently
  - Specifically, we'll look at Internet Protocol (IP) addresses.
  - Most popular: IP version 4 or IPv4. (later: IPv6)

IPv4 Addresses



- 32 bits long
- Identifier for a network interface
- An IP address corresponds to the point of attachment of an endpoint to the network.
- An IP address is NOT an identifier for the endpoint
  - Changes when endpoint moves
- Dotted quad notation: each byte is written in decimal in MSB order, separated by dots. Example:

10000000 11000011 00000001 01010000

128 . 195 . 1 . 80

Grouping IP addresses by prefixes

- IP addresses can be grouped based on a shared prefix of a specified length

- Example: consider two IP addresses:
  - 128.95.1.80 and 128.95.1.4
  - The addresses share a prefix of (bit) length 24: 128.95.1
  - The addresses have different suffixes of (bit) length 8

- IP addresses: prefix corresponds to the network component and the suffix to an endpoint (host) component of the address

IP addresses use hierarchy to scale routing

- IP addresses of endpoint interfaces in a network (e.g., Rutgers Busch campus) share a prefix of some length
- Each interface/endpoint has a different suffix, and hence a different 32-bit IP address
- Using prefixes reduces the amount of information needed to forward packets over the Internet
- IP prefixes are like zip codes: routers don't need to store info for each endpoint, just each prefix
- Prefixes also allow IP addresses to be delegated from one network to another (more on this later)

IP addresses use hierarchy to scale routing

- Postal envelopes should show clearly delineated zip codes.
- Q: How to identify the prefix from a 32-bit IP address?
  - Two methods:
- Old: Classful addressing. IP address itself is formatted to denote the IP prefix
- New: Classless addressing (also called classless inter-domain routing, or CIDR).
  - Each router independently identifies prefix from IP

Classless IPv4 addressing (CIDR)

Classless IPv4 addressing

- Also called classless inter-domain routing (CIDR)
- Key idea: Network component of the address (ie: prefix) can have any length (usually from 8—32)
- Address format: a.b.c.d/x, where x is the prefix length
  - Customary to use 0s for all suffix bits

network part | host part

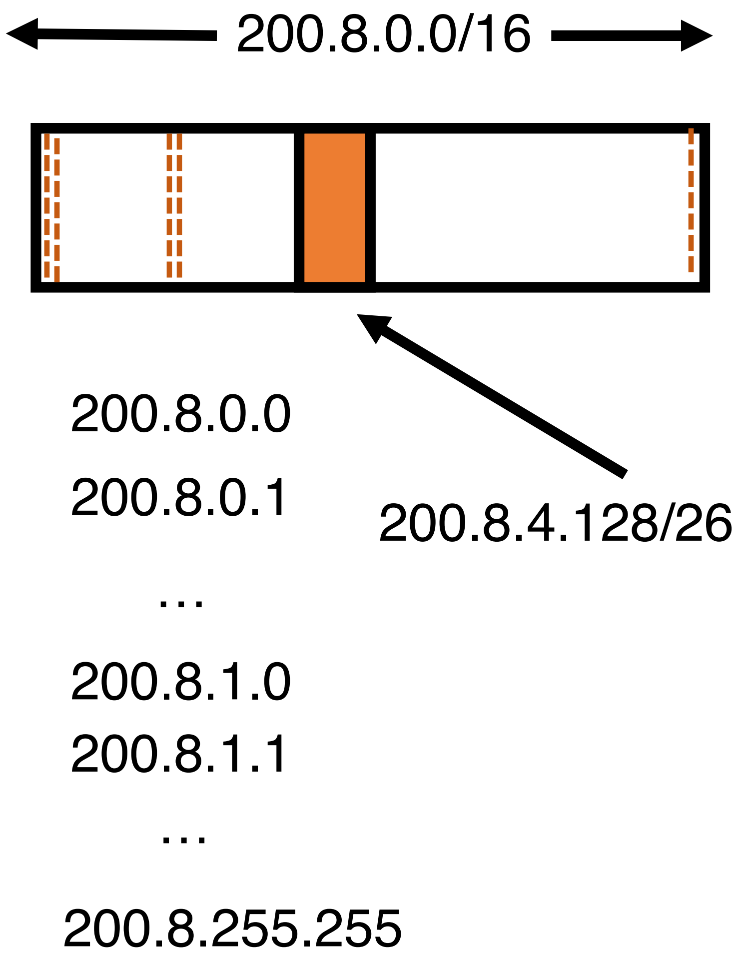
11001000 00010111 00010000 00000000

200.23.16.0/23

https://www.internetsociety.org/resources/deploy360/2015/short-guide-ip-addressing/

CIDR

- An ISP can obtain a block of addresses and partition this further to its customers
- Say an ISP has 200.8.0.0/16 address (65K addresses).
- The ISP has customer who needs only 64 addresses starting from 200.8.4.128
- Then that block can be specified as 200.8.4.128/26
- 200.8.4.128/26 is “inside” 200.8.0.0/16



Netmask (or subnet mask)

- An alternative to denote the IP prefix length of an organization
- 32 bits: a 1-bit denotes a prefix bit position. 0 denotes host bit.

network part | host part

11001000 00010111 00010000 00000000

200.23.16.0/23

network part of mask | host part of mask

11111111 11111111 11111110 00000000

Netmask: 255.255.254.0

Detecting addresses from same network

- Given IP addresses A and B, and netmask M.
  1. Compute logical AND (A & M).
  2. Compute logical AND (B & M).
  3. If (A & M) == (B & M) then A and B are on the same network.

- Ex: A = 165.230.82.52, B = 165.230.24.93, M = 255.255.128.0

- A and B are in the same network according to the netmask
- A & M == B & M == 165.230.0.0

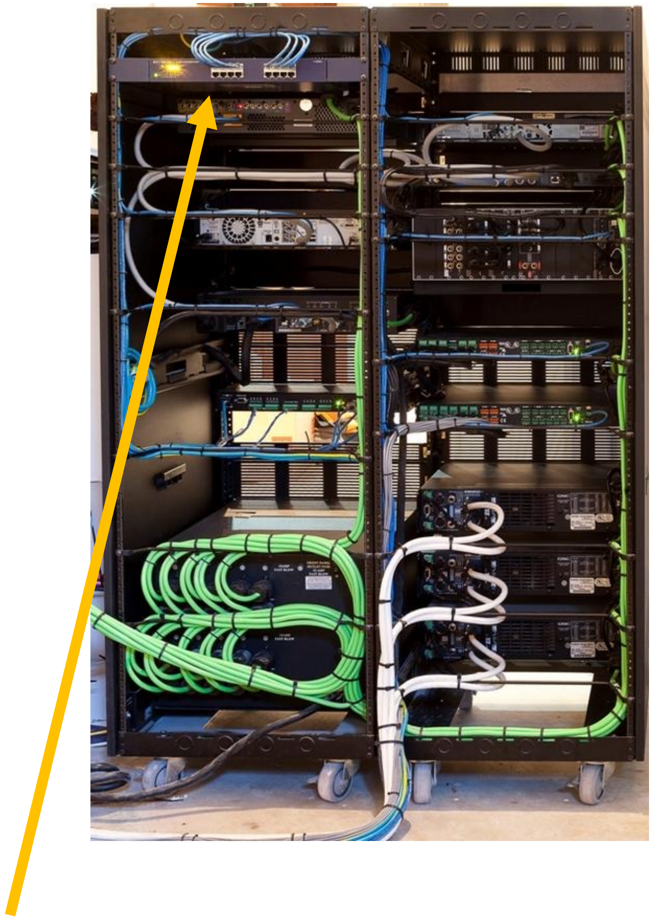
Finding your own IP address(es)

- The old way (still works today on Mac and Linux):
  - `ifconfig —a`
- The new way using “iproute2” tools on Linux:
  - `ip link`
  - `ip addr`
- What else do you see in these outputs?

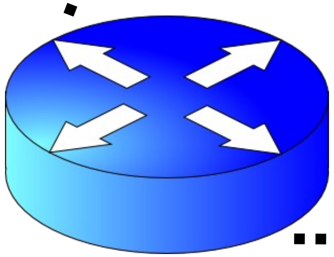
Next we'll talk about routers

Access routers

Core router

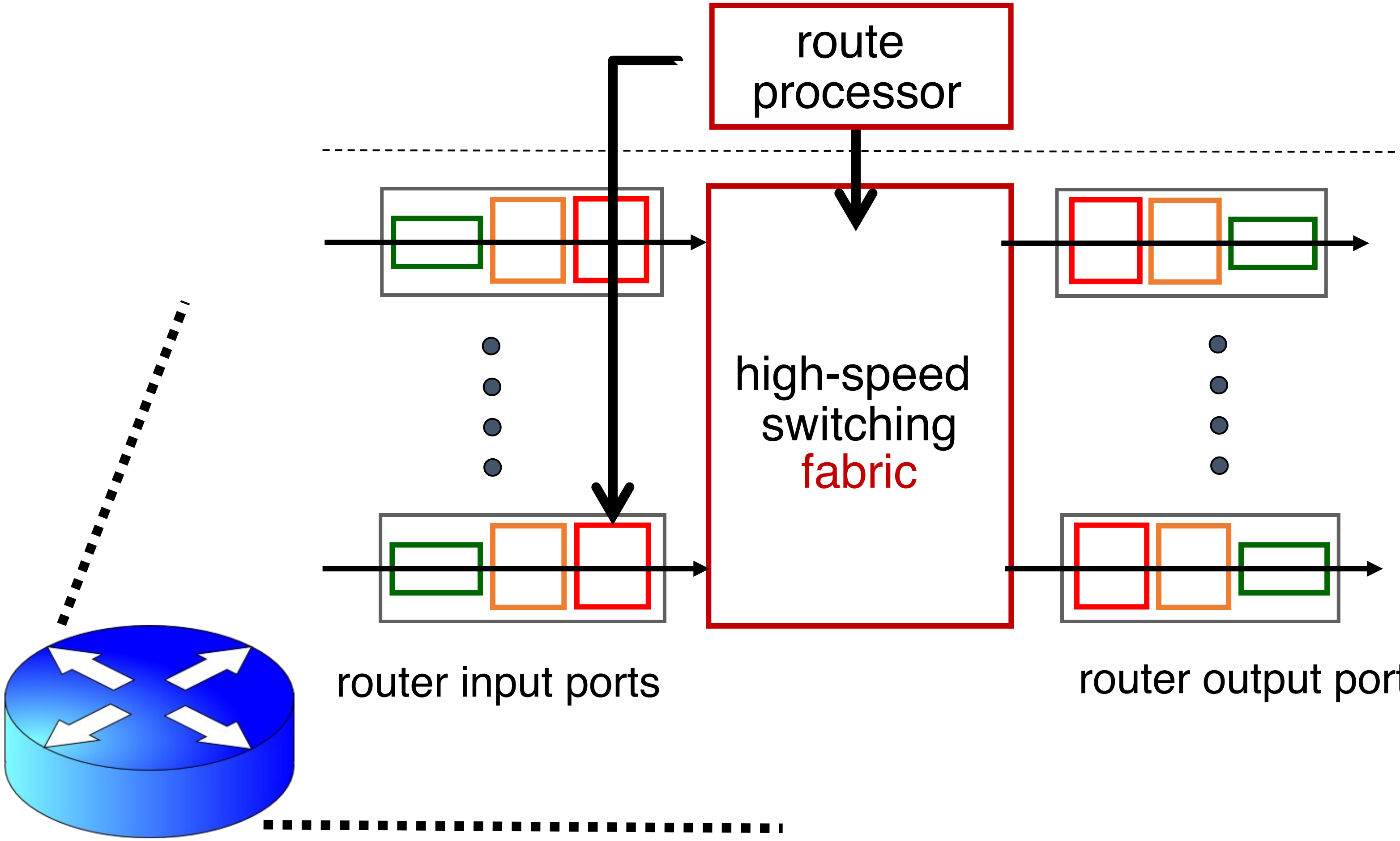


Data center top-of-rack switch



What’s inside a router?

Router architecture overview



Review: assuming distributed routing, routing function: decide which ports packets need to exit

Control plane

Data plane

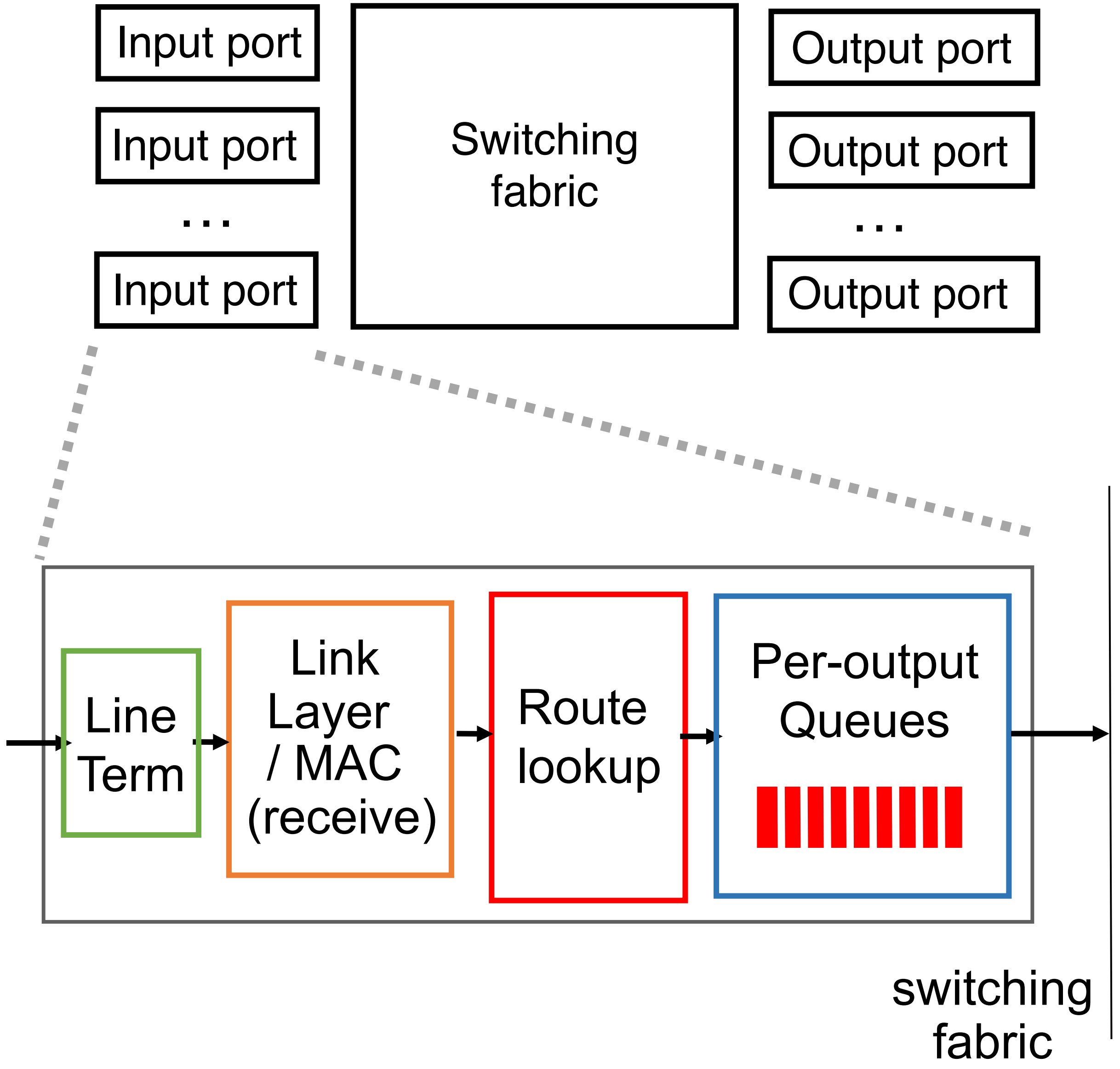
Review: Forwarding function: move packets from one input port to another.

Different and evolving designs

- There are different kinds of routers, with their own designs
  - Access routers (e.g., home WiFi), chassis/core routers, top-of-rack switches
- Router designs have also evolved significantly over time
- For simplicity and concreteness, we will learn about one high-speed router design from the early 2000s.
- Called the MGR (multi-gigabit router). It could support an aggregate rate of 50 Gbit/s (1 G = 10^9)
  - Today's single-chip routers can support aggregate rates of ~10 Tbit/s (1 T = 10^{12})

Input port functions

- Line termination: receives physical (analog) signals and turns them into digital signals
- Rate of link connecting to a single port termed line speed or line rate (modern routers: 100+ Gbit/s)
- Link layer: performs medium access control functions (e.g., Ethernet)

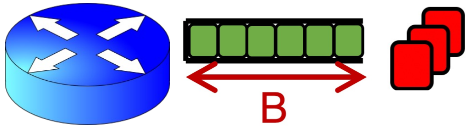


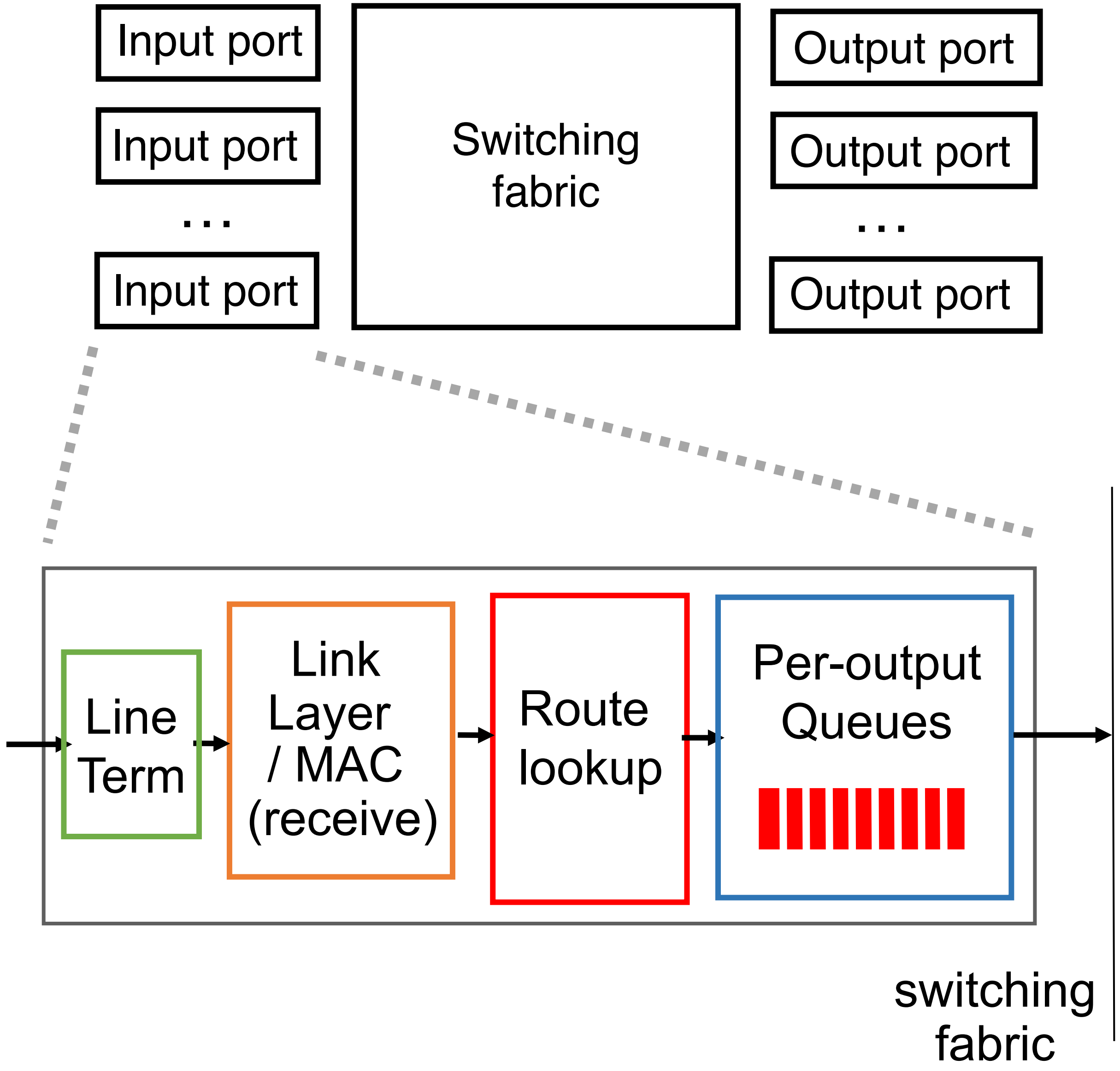
Input port functions

- Route lookup: high-speed lookup of which output port the packet is destined to

- Goal: must complete this processing at the line rate

- Queueing: packets may wait in per-output-port queues if packets are arriving too fast for the switching fabric to send them to the output port





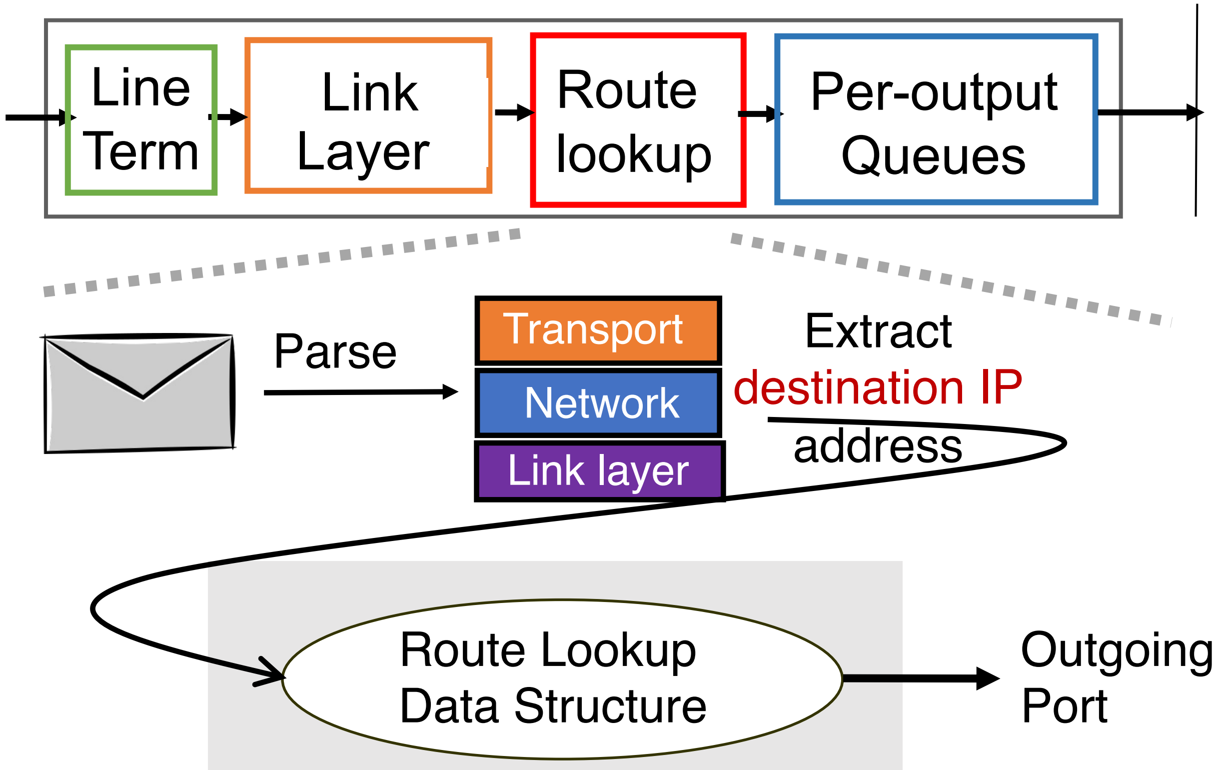
Route lookups

Packet forwarding in the Internet is based on the destination IP address on the packet.

Example: if dst IP on packet is 65.45.145.34, it matches the forwarding table prefix 65.0.0.0/8.

The packet is forwarded out port 3.

Example 2: what about dst IP 128.9.5.6?



Forwarding Table

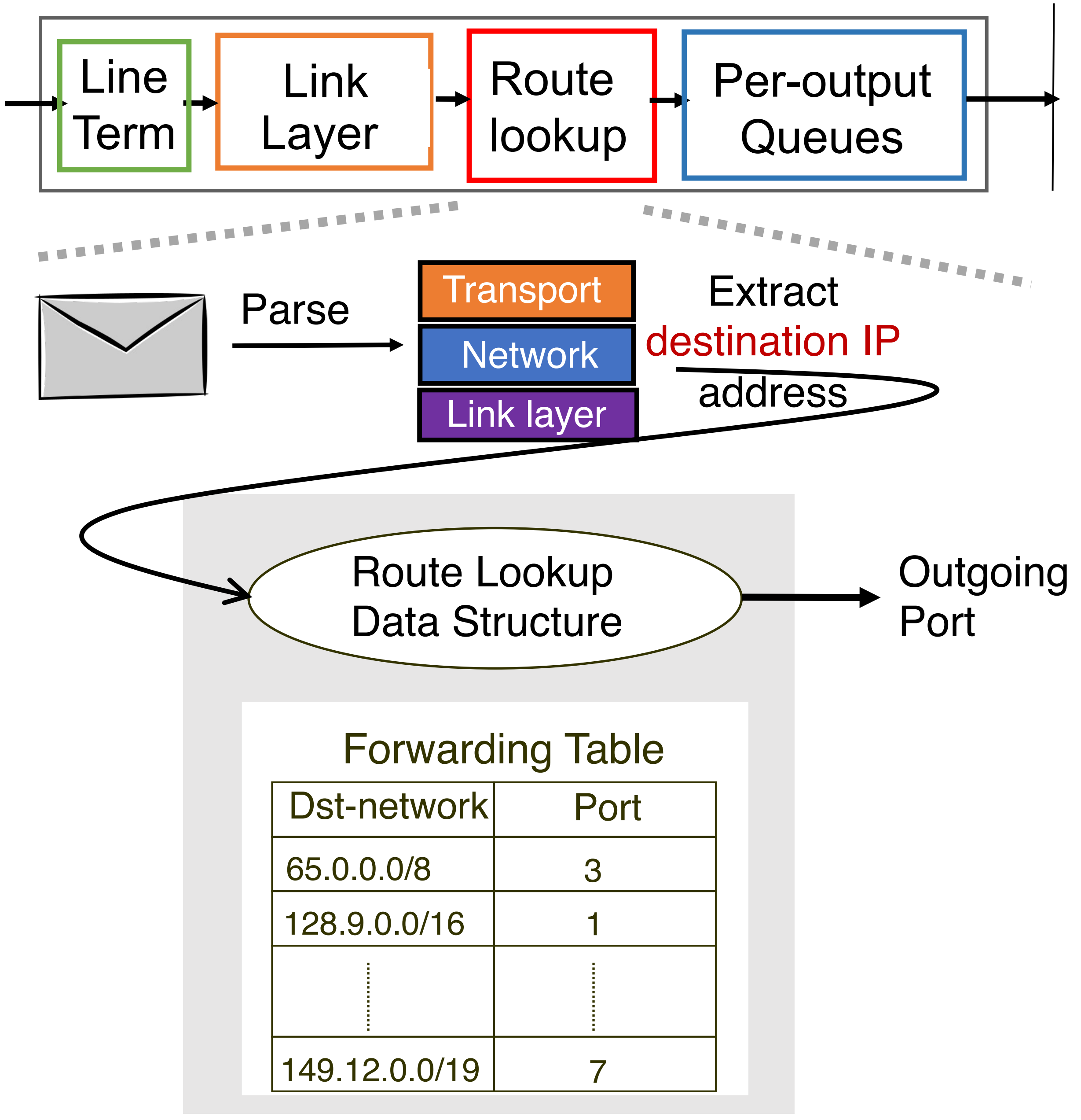
Dst-network	Port
65.0.0.0/8	3
128.9.0.0/16	1
⋮	⋮
149.12.0.0/19	7

Route lookups

Number of entries in the forwarding table matters.

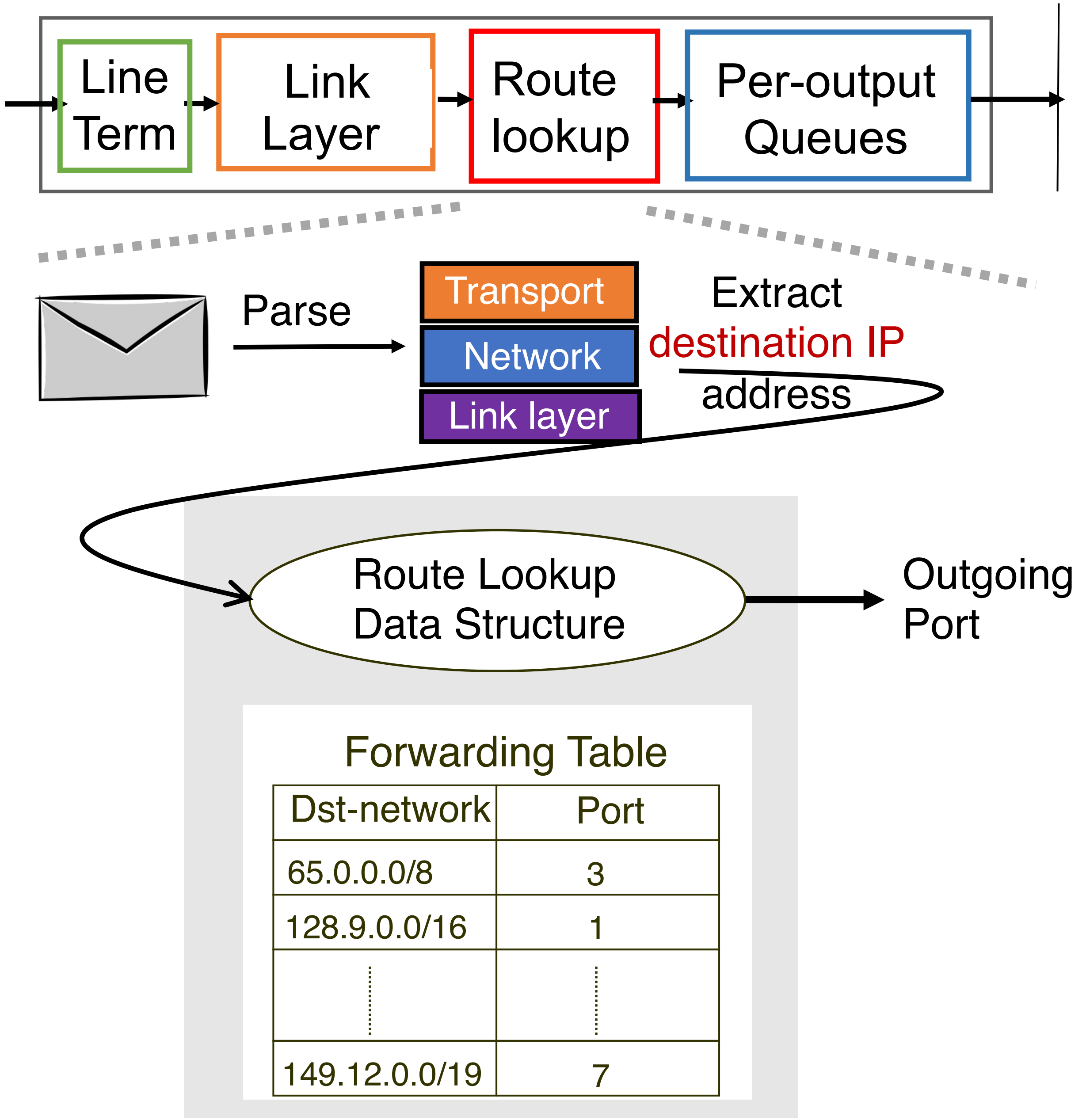
Fitting into router memory

Designing hardware and software for fast lookups



Dst-network	Port
65.0.0.0/8	3
128.9.0.0/16	1
⋮	⋮
149.12.0.0/19	7

Route lookups



Dst-network	Port
65.0.0.0/8	3
128.9.0.0/16	1
⋮	⋮
149.12.0.0/19	7

Recall: IP addresses can be aggregated based on shared prefixes.

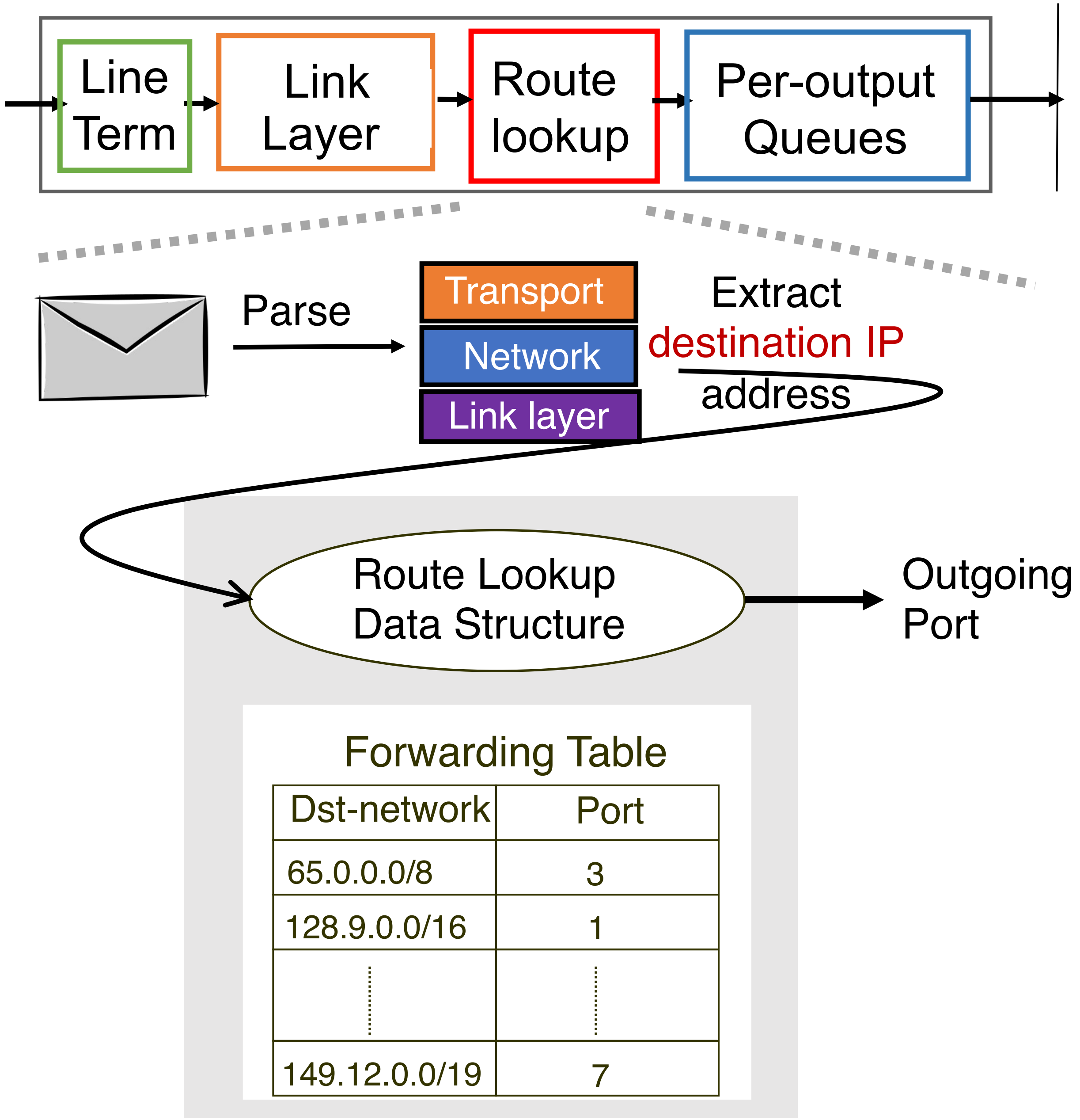
The number of table entries in a router is proportional to the number of prefixes, NOT the number of endpoints.

Today: ~ 1 million prefixes.

Route lookups

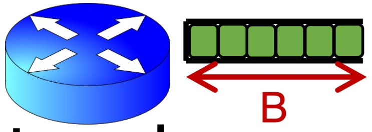
Destination-IP-based forwarding has consequences.

- Forwarding behavior is independent of the source: legitimate source vs. malicious attack traffic
- Forwarding behavior is independent of the application: web traffic vs. file download vs. video
- IP-based packet processing is “baked into” router hardware: evolving the IP protocol faces tall deployment hurdles

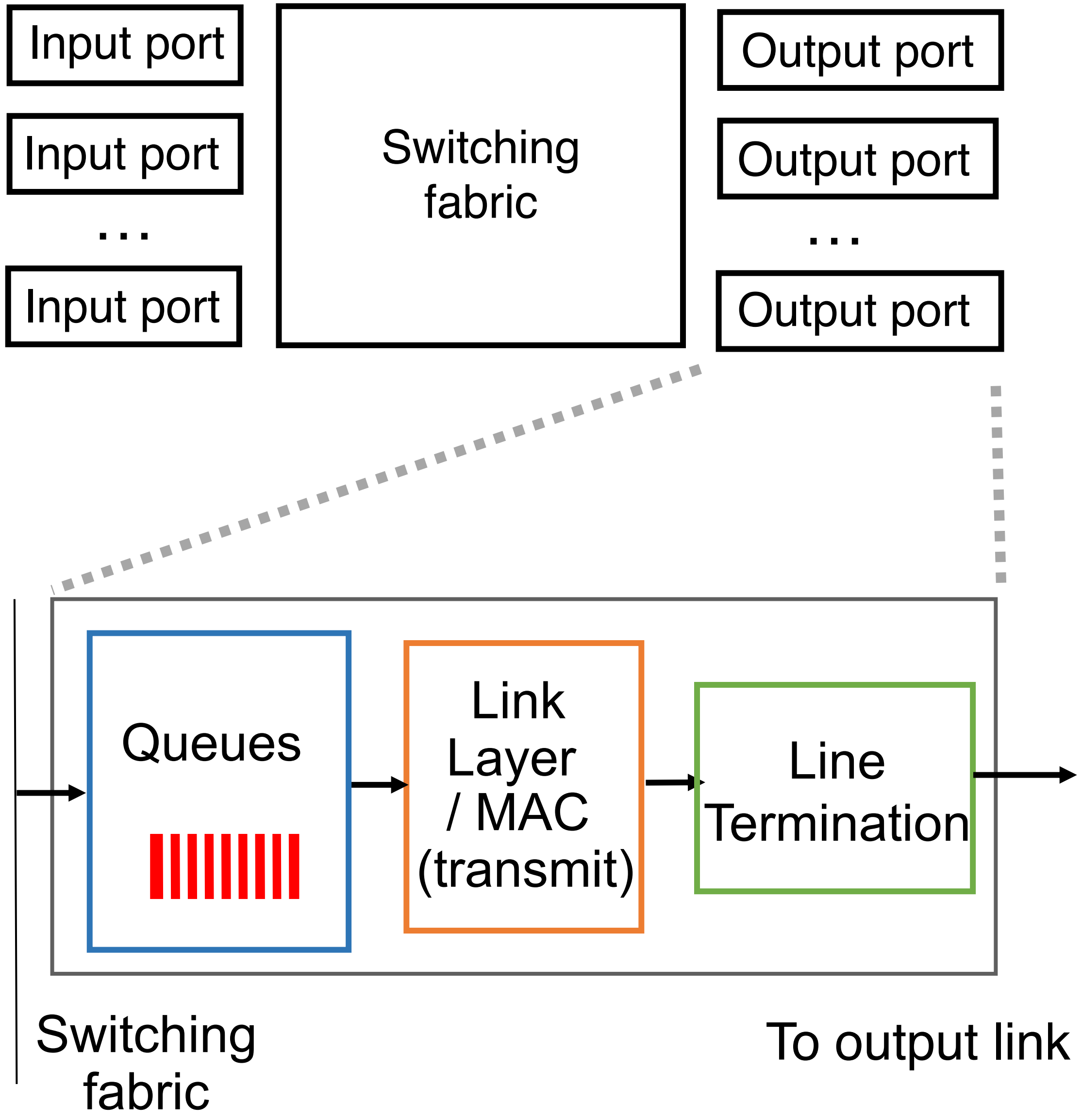


Output port functions

- Components in reverse order of those in the input port

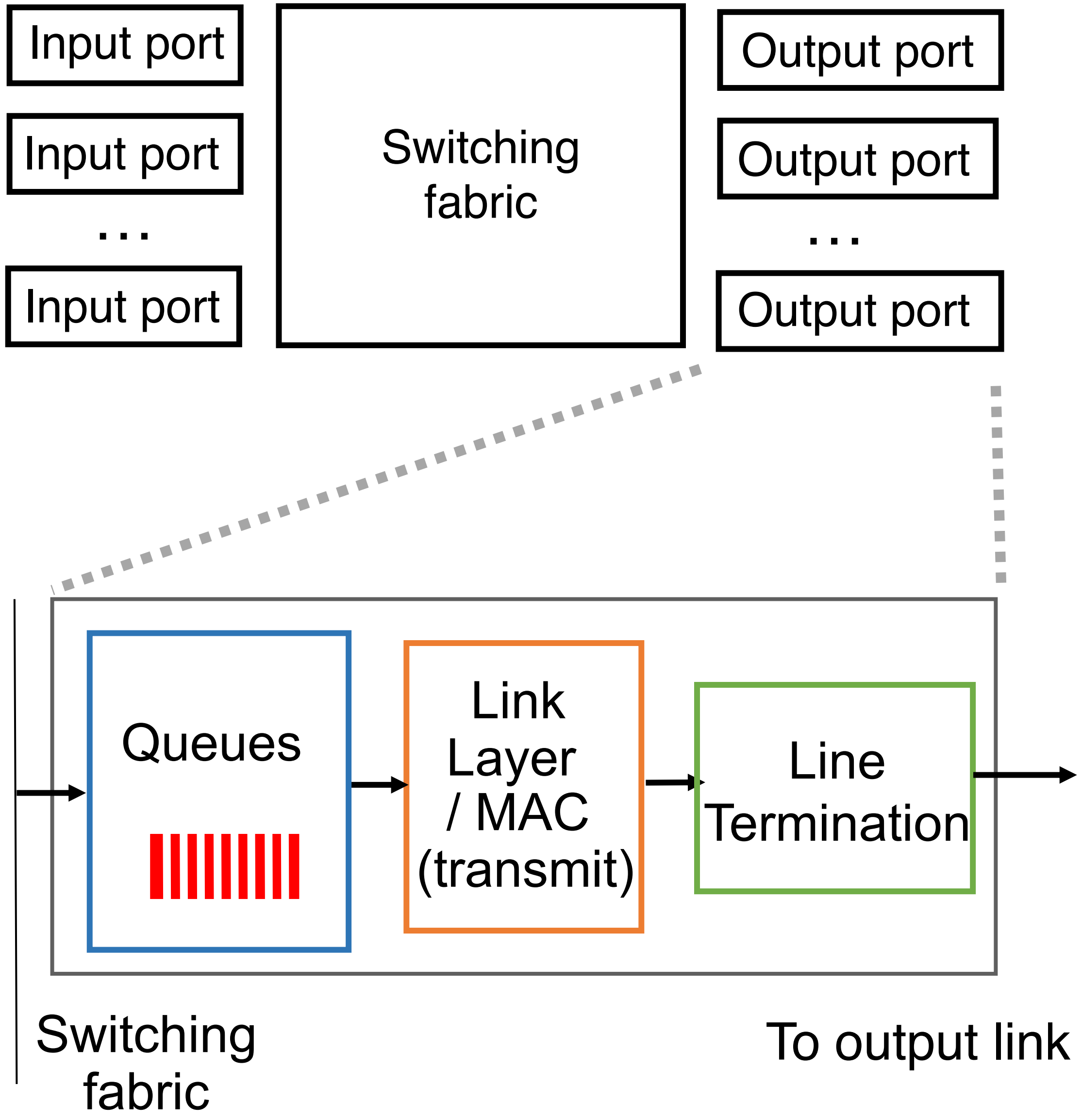


- This is where most routers have the bulk of their packet buffers
  - Recall discussions regarding router buffers from transport
- MGR uses per-port output buffers, but modern routers have shared memory buffers
  - More efficient use of memory under varying demands



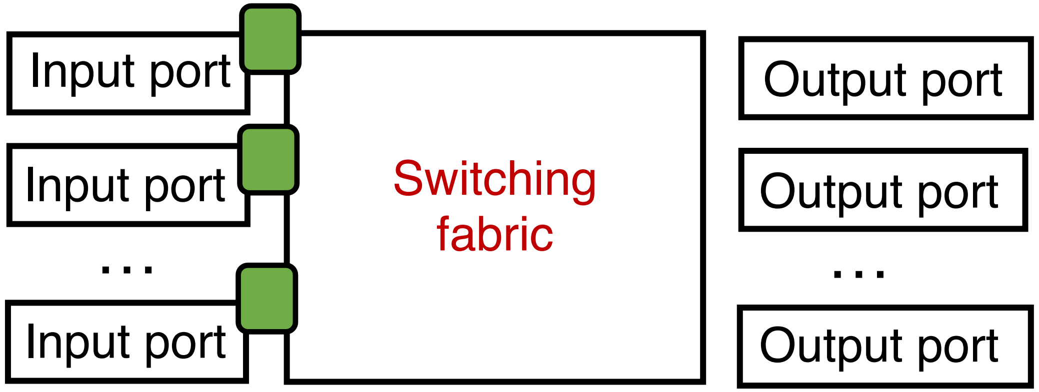
Output port functions

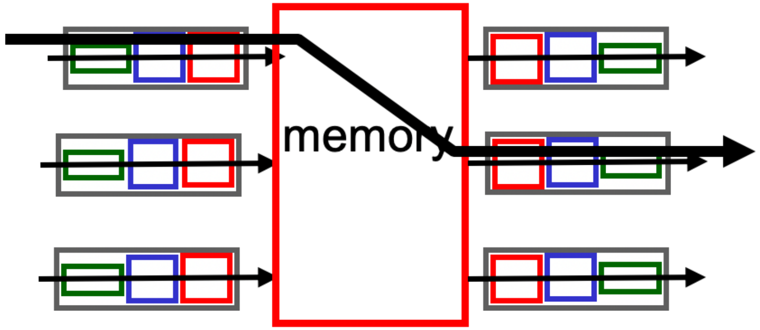
- Two important policy decisions
- Scheduling: which among the waiting packets gets to be transmitted out the link?
  - Ex: First-In-First-Out (FIFO)
- Buffer management: which among the packets arriving from the fabric get space in the packet buffer?
  - Ex: Tail drop: later packets dropped first



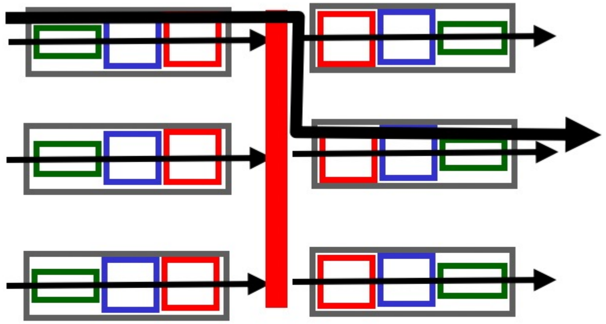
Fabrics: Types

Fabric goal: Ferry as many packets as possible from input to output ports as quickly as possible.

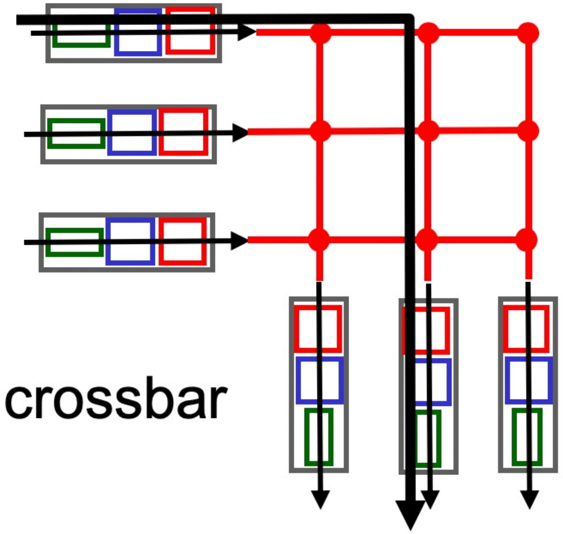




memory



bus



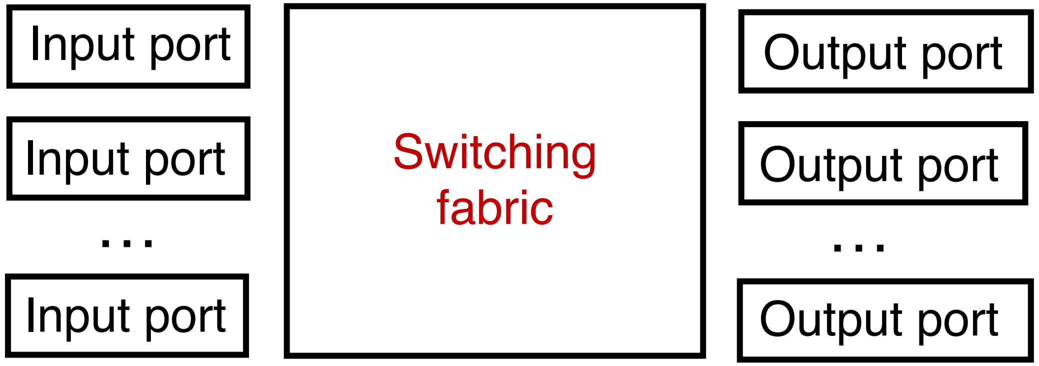
crossbar

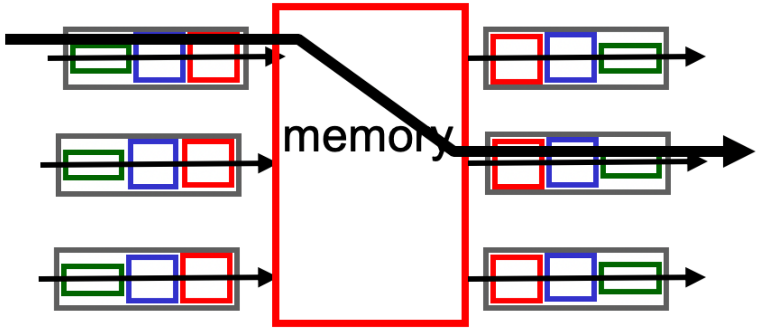
Input port writes packets into shared memory.
Output port reads the packet when output link ready to transmit.

Single shared channel to move data from input to output port.
Easy to build buses; technology is quite mature.

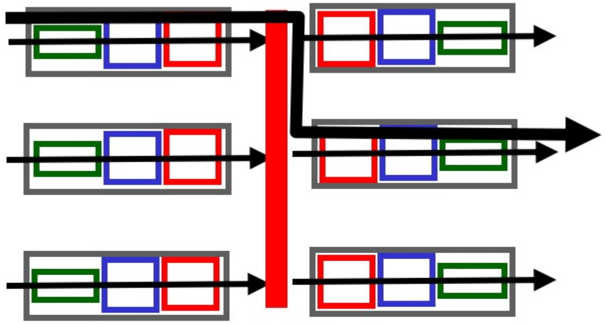
Each input port has a physical data path to every output port.
Switch at the cross-over points turns on to connect pairs of ports.

Fabrics: Types

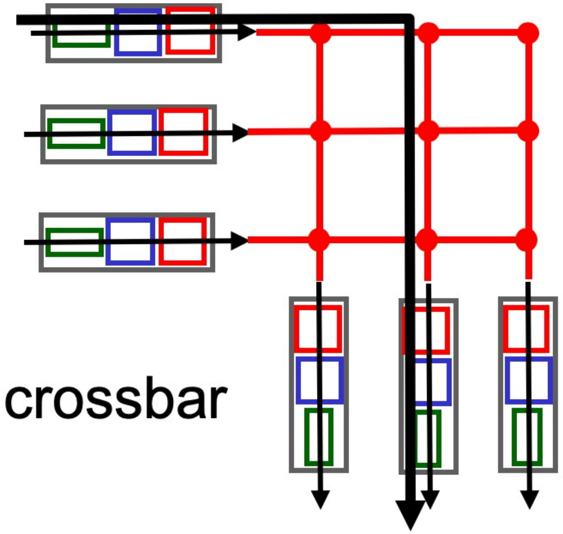




memory



bus

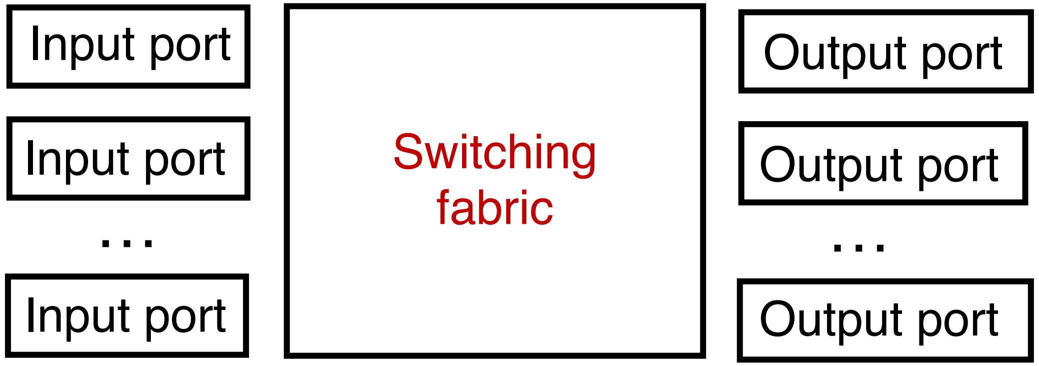


crossbar

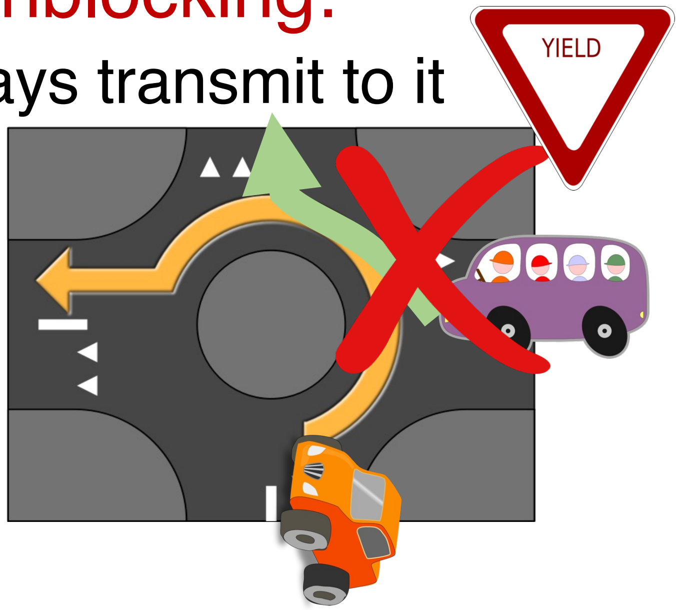
Modern high-speed routers use highly optimized shared-memory-based interconnects.

Crossbars can get expensive as the number of ports grows (N^2 connections for N ports)
MGR uses a crossbar and schedules (in,out) port pairs.

Nonblocking fabrics

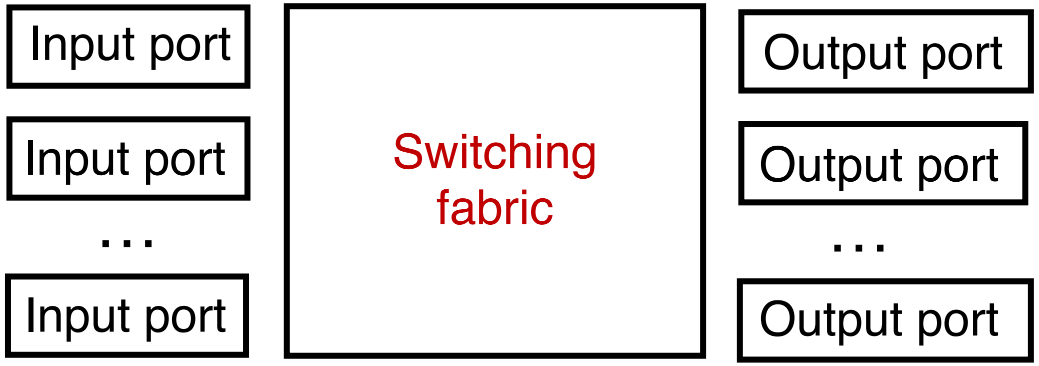


- High-speed switching fabrics designed to be nonblocking:
  - If an output port is “available”, an input port can always transmit to it without being blocked by the switching fabric itself
  - Nontrivial to achieve
- Crossbars are nonblocking by design



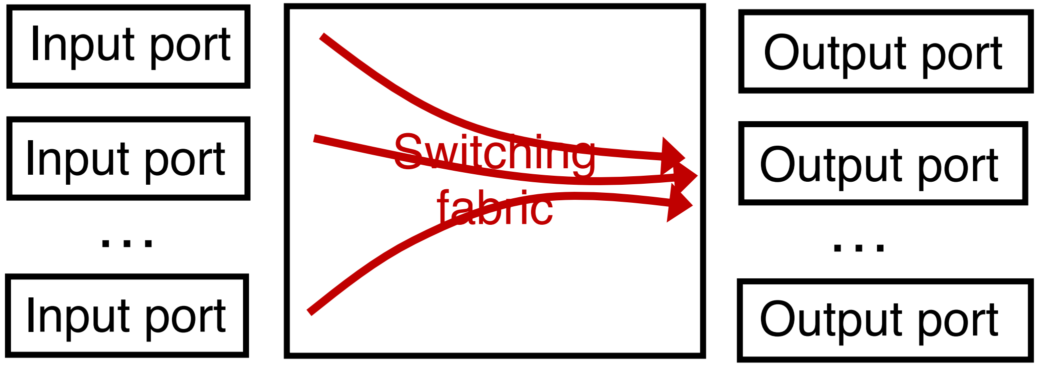
- Shared memory can be designed to be nonblocking if the memory access is fast enough

Nonblocking fabrics



- With a nonblocking fabric, queues aren’t formed due to the switching fabric.
  - With a nonblocking fabric, there are no queues due to inefficiencies at the input port or the switching fabric
- Queues only form due to contention for the output port
  - Fundamental, unavoidable, given the route

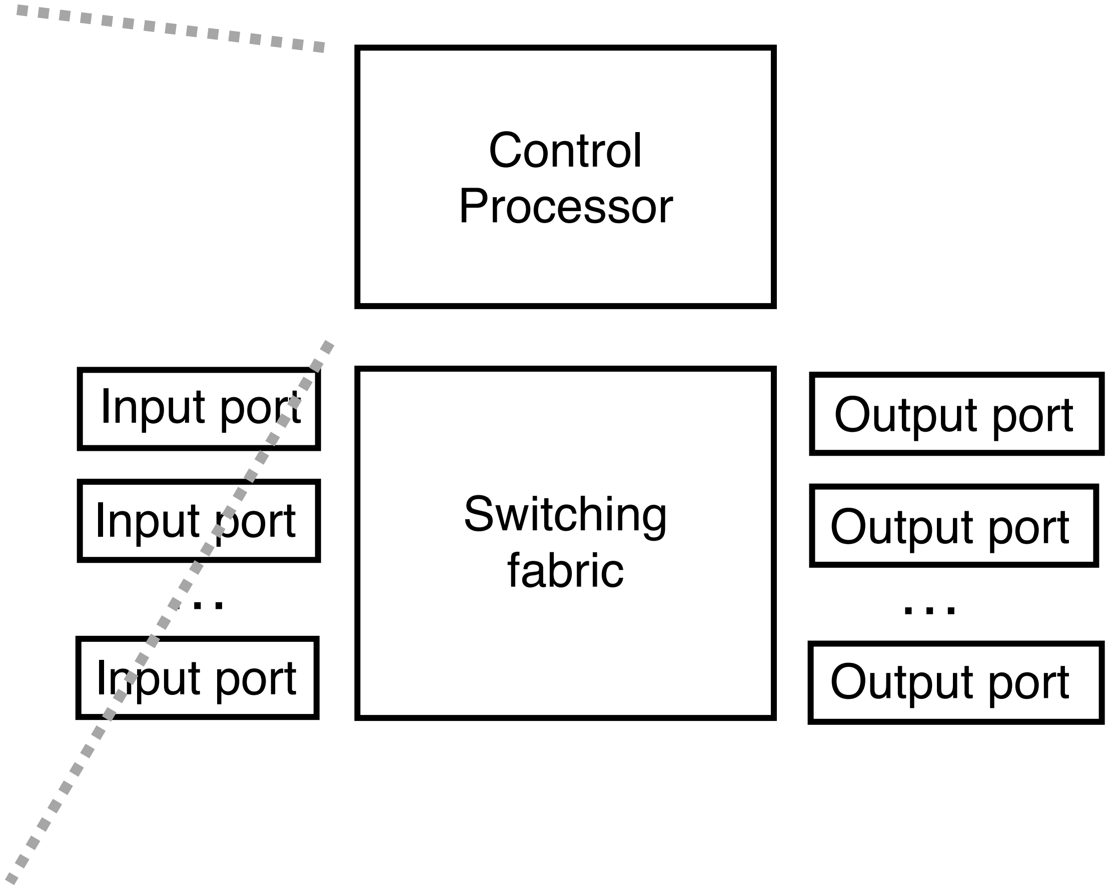
Nonblocking fabrics



- With a nonblocking fabric, queues aren't formed due to the switching fabric.
  - With a nonblocking fabric, there are no queues due to inefficiencies at the input port or the switching fabric
- Queues only form due to contention for the output port
  - Fundamental, unavoidable, given the route
- Typically, these queues form on the output side
  - But can also "backpressure" to the input side if there is high contention for the output port
  - i.e.: can't move pkts to output Qs since buffers full, so buffer @ input

Control (plane) processor

- A general-purpose processor that "programs" the data plane:
  - Forwarding table
  - Scheduling and buffer management policy
- Implements the routing algorithm by processing routing protocol messages
  - Mechanism by which routers collectively solve the Internet routing problem
  - More on this soon.



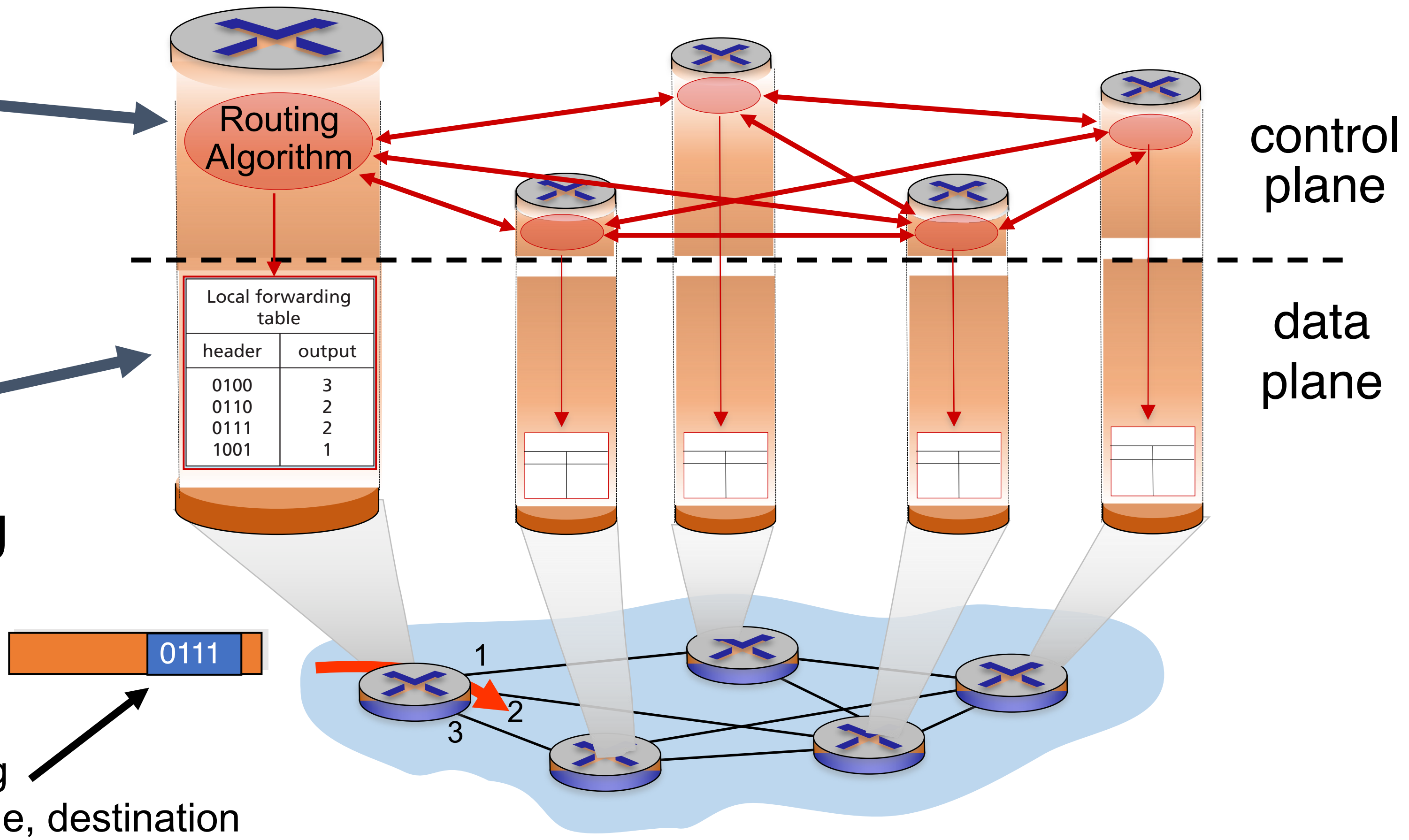
Router design: the bigger picture

Control plane

Traditional distributed routing: per route-change processing (~ a few tens of seconds)

Data plane

per-packet processing (~ tens of nanoseconds)

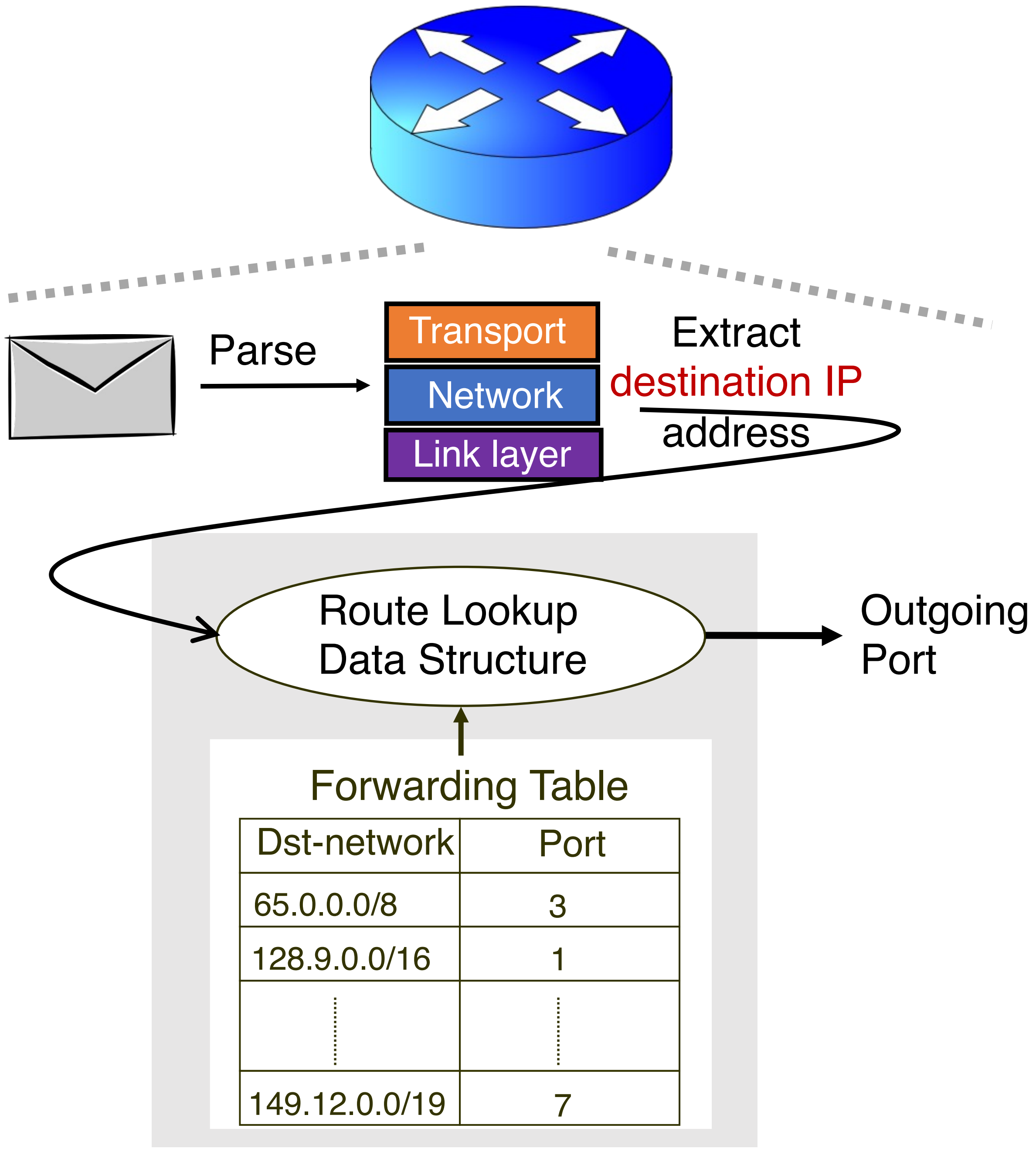


values in arriving packet header, i.e, destination IP address

Longest Prefix Matching

Review: Route lookup

- Table lookup matches a packet against an IP prefix
  - Ex: 65.12.45.2 matches 65.0.0.0/8
- Prefixes are allocated to organizations by Internet registries
- But organizations can reallocate a subset of their IP address allocation to other orgs

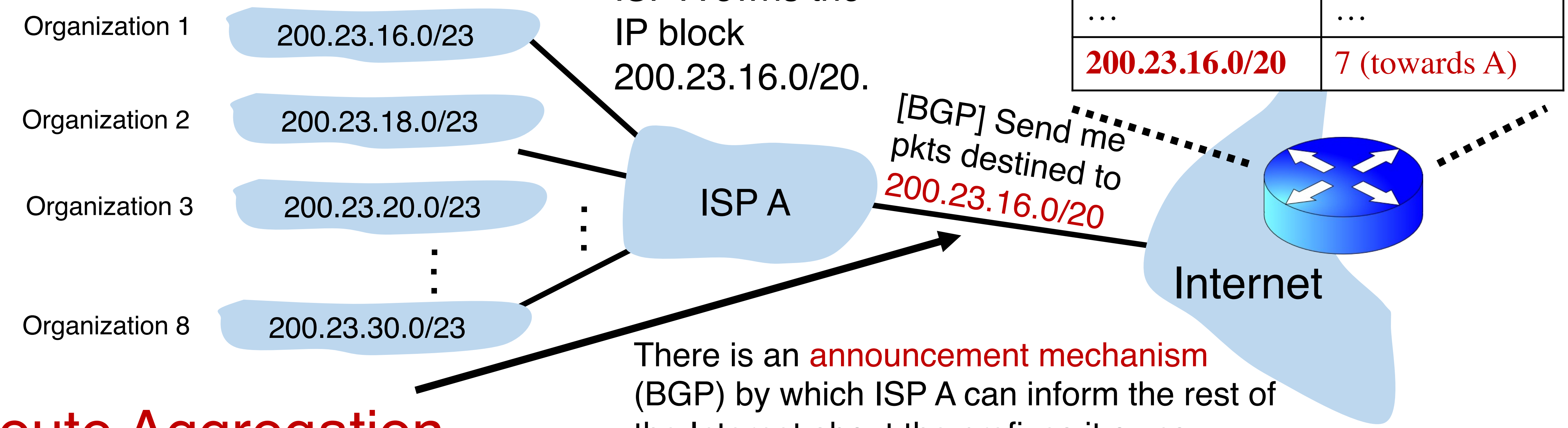


Dst-network	Port
65.0.0.0/8	3
128.9.0.0/16	1
⋮	⋮
149.12.0.0/19	7

Example of IP block reallocation

Suppose ISP A reallocates a part of its IP block to orgs 1… 8

Dst IP Prefix	Output port
65.0.0.0/8	3
128.9.0.0/16	1
…	…
200.23.16.0/20	7 (towards A)



Route Aggregation

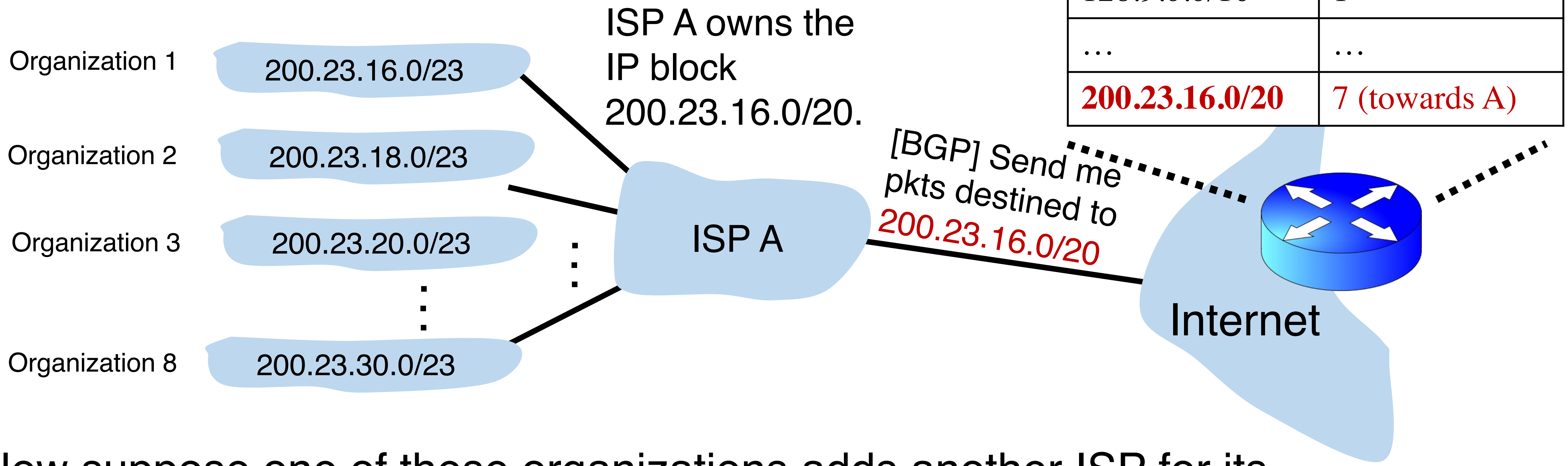
Save forwarding table memory
Fewer routing protocol msgs

There is an announcement mechanism (BGP) by which ISP A can inform the rest of the Internet about the prefixes it owns.
It is enough to announce a coarse-grained prefix 200.23.16.0/20 rather than 8 separate sub-prefixes.

Example of IP block reallocation

Suppose ISP A reallocates a part of its IP block to orgs 1… 8

Dst IP Prefix	Output port
65.0.0.0/8	3
128.9.0.0/16	1
…	…
200.23.16.0/20	7 (towards A)

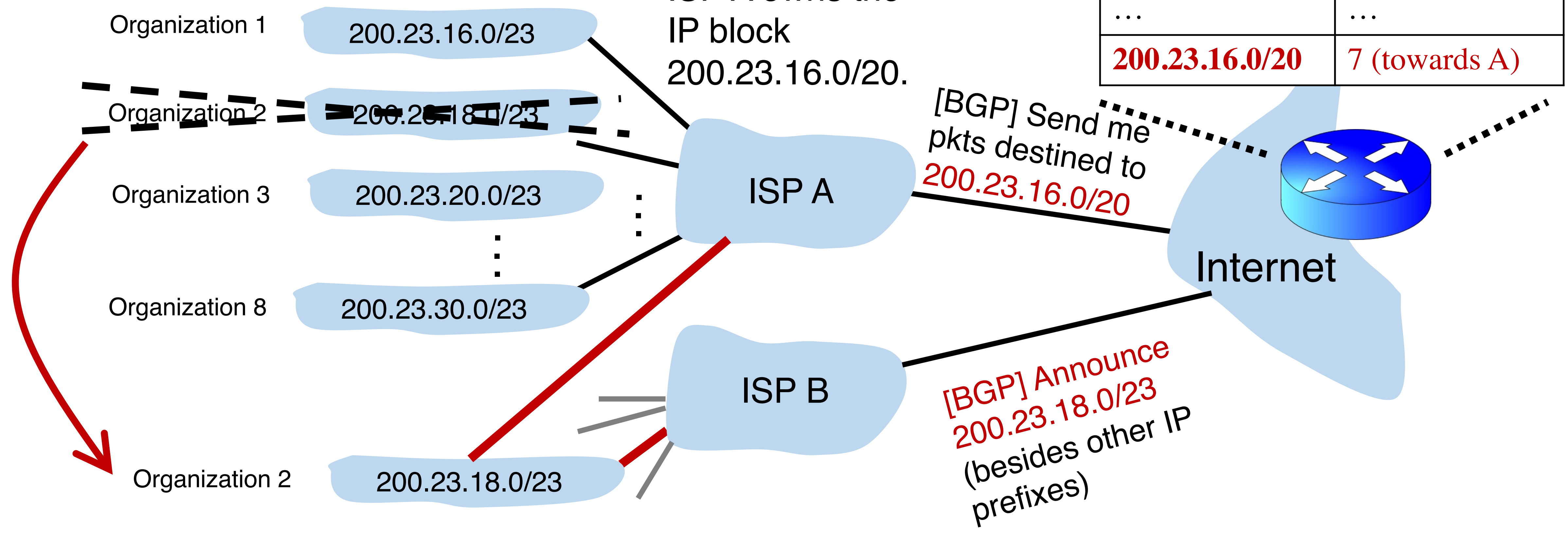


Now suppose one of these organizations adds another ISP for its Internet service and prefers using the new ISP.
Note: it's possible for the organization to retain its assigned IP block.

Example of IP block reallocation

Suppose ISP A reallocates a part of its IP block to orgs 1… 8

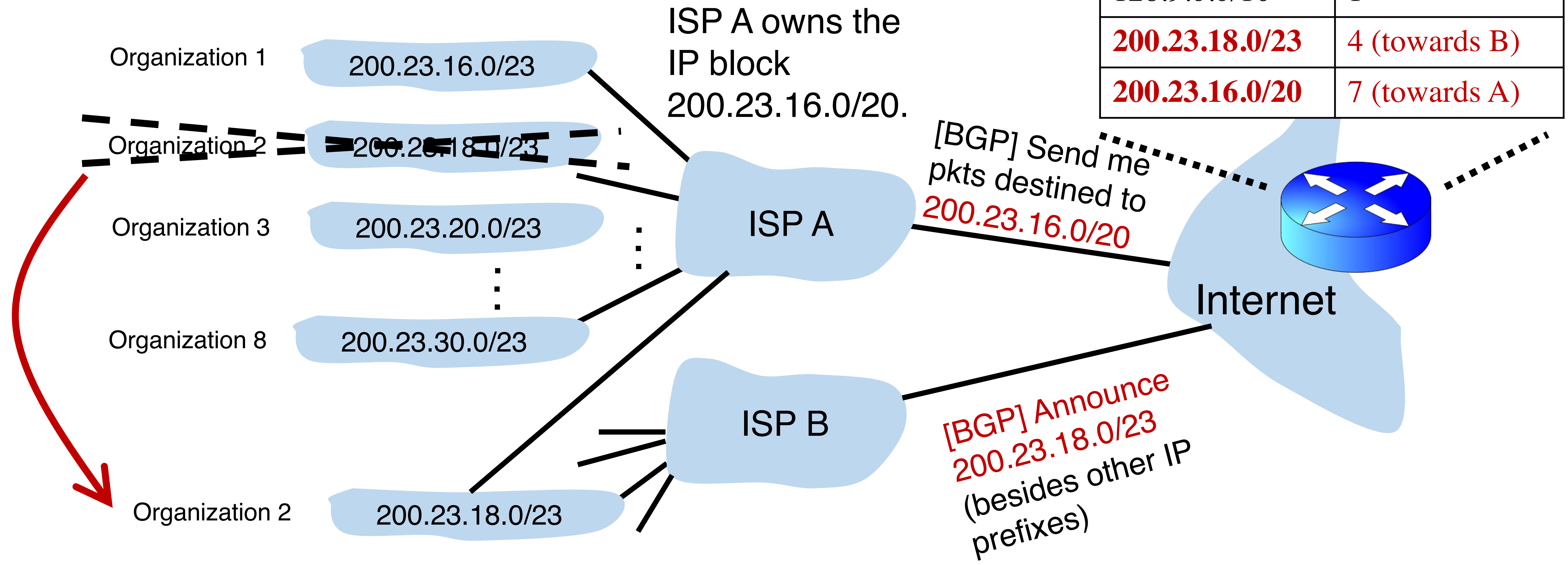
Dst IP Prefix	Output port
65.0.0.0/8	3
128.9.0.0/16	1
…	…
200.23.16.0/20	7 (towards A)



Example of IP block reallocation

Suppose ISP A reallocates a part of its IP block to orgs 1… 8

Dst IP Prefix	Output port
65.0.0.0/8	3
128.9.0.0/16	1
200.23.18.0/23	4 (towards B)
200.23.16.0/20	7 (towards A)



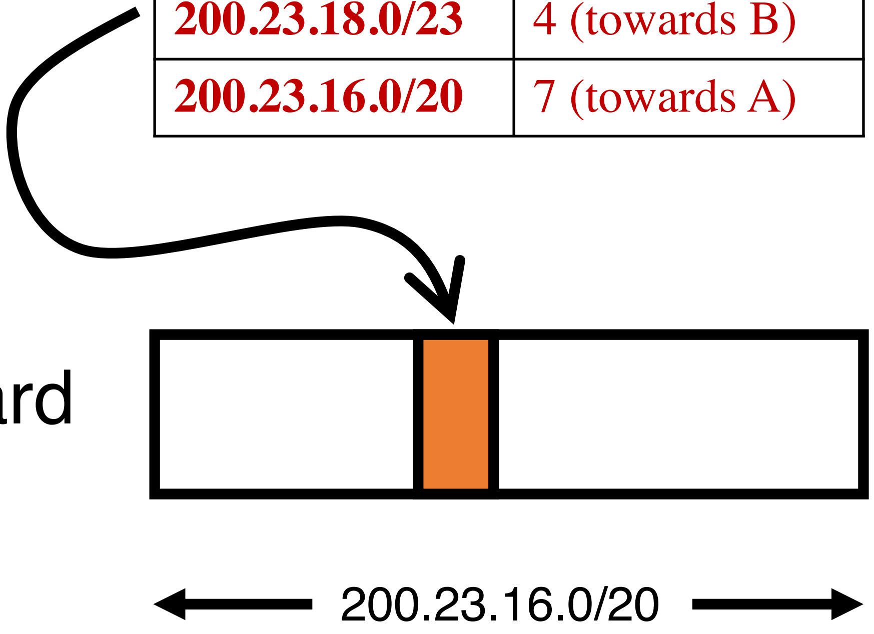
A closer look at the forwarding table

- 200.23.18.0/23 is inside 200.23.16.0/20

- A packet with destination IP address 200.23.18.xx is in both prefixes
  - i.e., both entries match

- Q: How should the router choose to forward the packet?
  - The org prefers B, so should choose B

Dst IP Prefix	Output port
65.0.0.0/8	3
128.9.0.0/16	1
200.23.18.0/23	4 (towards B)
200.23.16.0/20	7 (towards A)



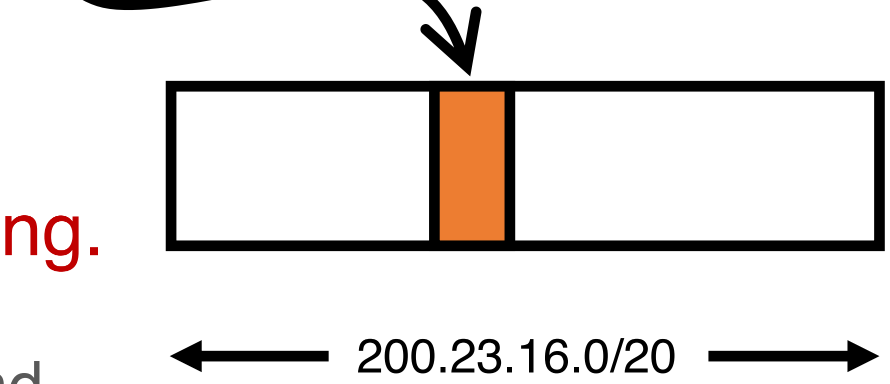
Longest Prefix Matching (LPM)

- Use the longest matching prefix, i.e., the most specific route, among all prefixes that match the packet.

- Policy borne out of the Internet's IP allocation model: prefixes and sub-prefixes are handed out

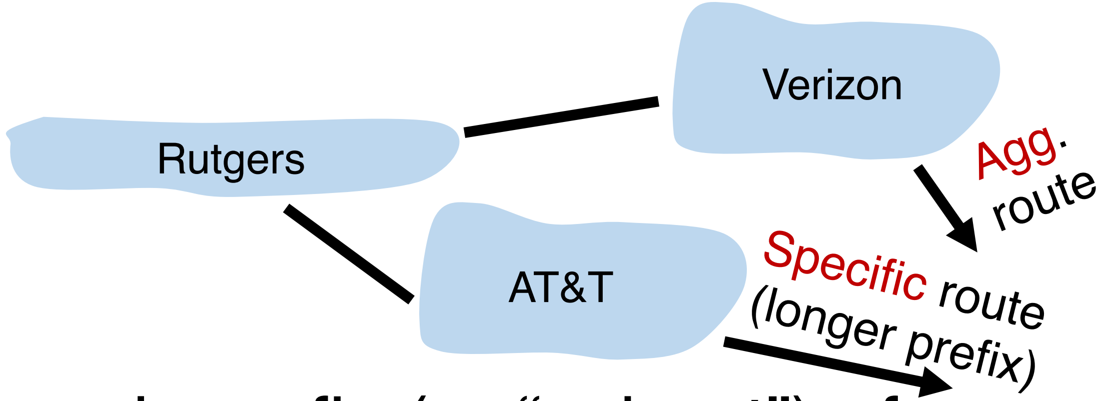
- Internet routers use longest prefix matching.
  - Very interesting algorithmic problems
  - Challenges in designing efficient software and hardware data structures

Dst IP Prefix	Output port
65.0.0.0/8	3
128.9.0.0/16	1
200.23.18.0/23	4 (towards B)
200.23.16.0/20	7 (towards A)



Internet routers perform longest-prefix matching on destination IP addresses of packets.

Why is LPM prevalent?



- An ISP (e.g., Verizon) has allocated a sub-prefix (or “subnet”) of a larger prefix that the ISP owns to an organization (e.g., Rutgers)
- Further, the ISP announces the aggregated prefix to the Internet to save on number of forwarding table memory and number of announcements
- The organization (e.g., Rutgers) is reachable over multiple paths (e.g., through another ISP like AT&T)
- The organization has a preference to use one path over another, and expresses this by announcing the longer (more specific) prefix
- Routers in the Internet must route based on the longer prefix