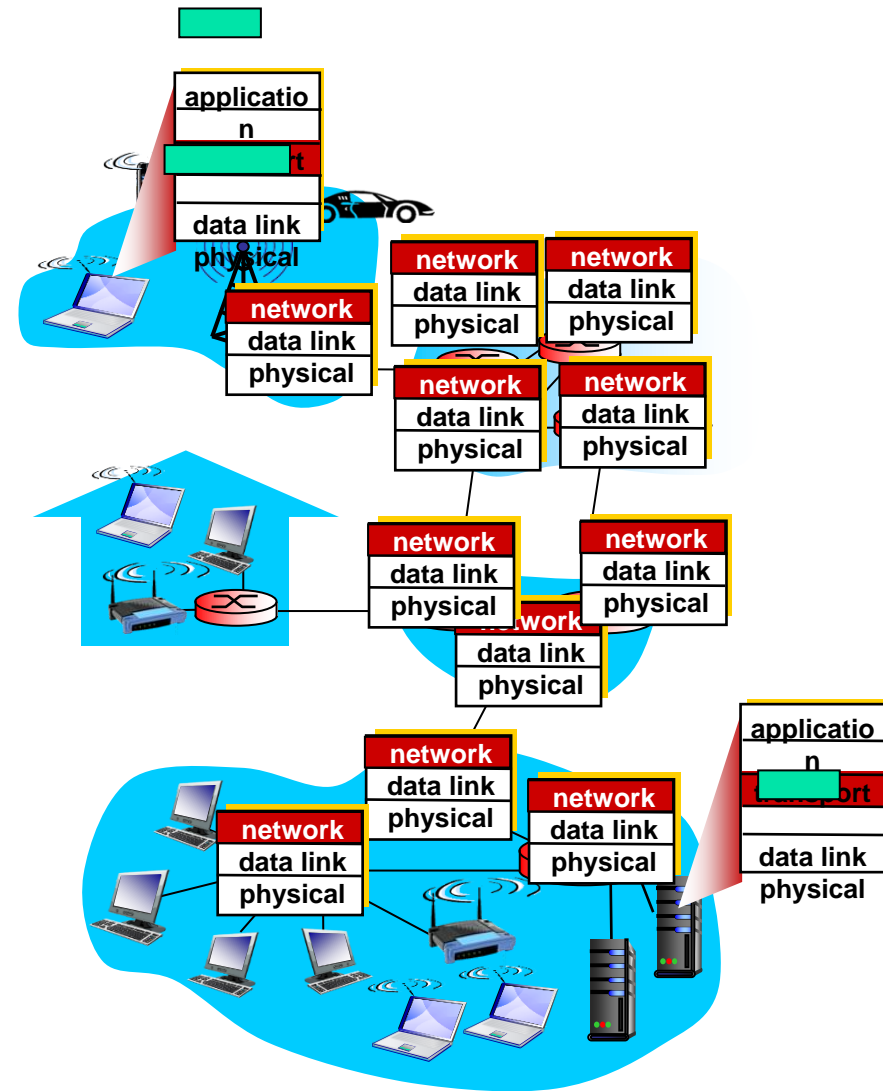


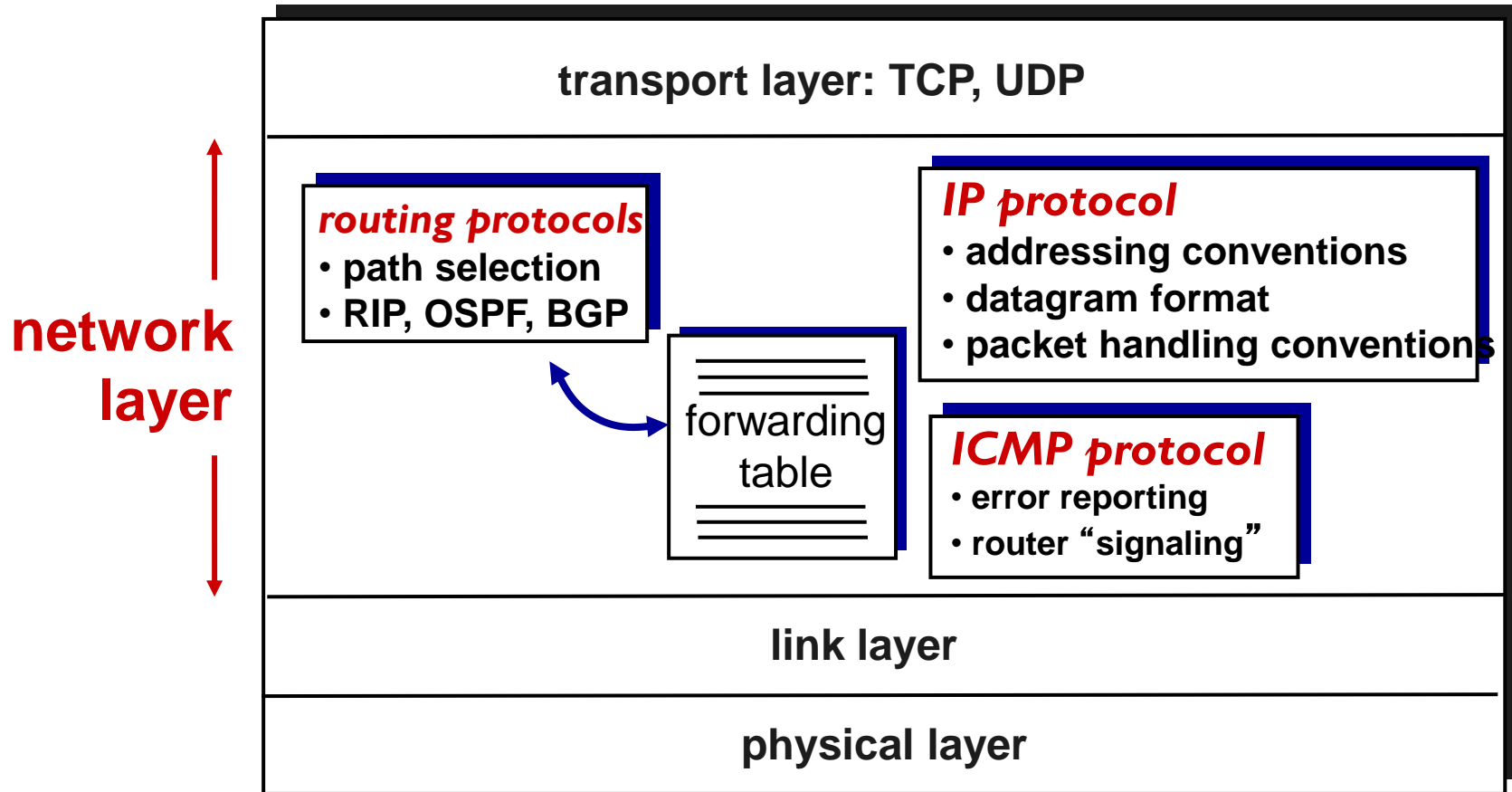
Network layer

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- router examines header fields in all IP datagrams passing through it

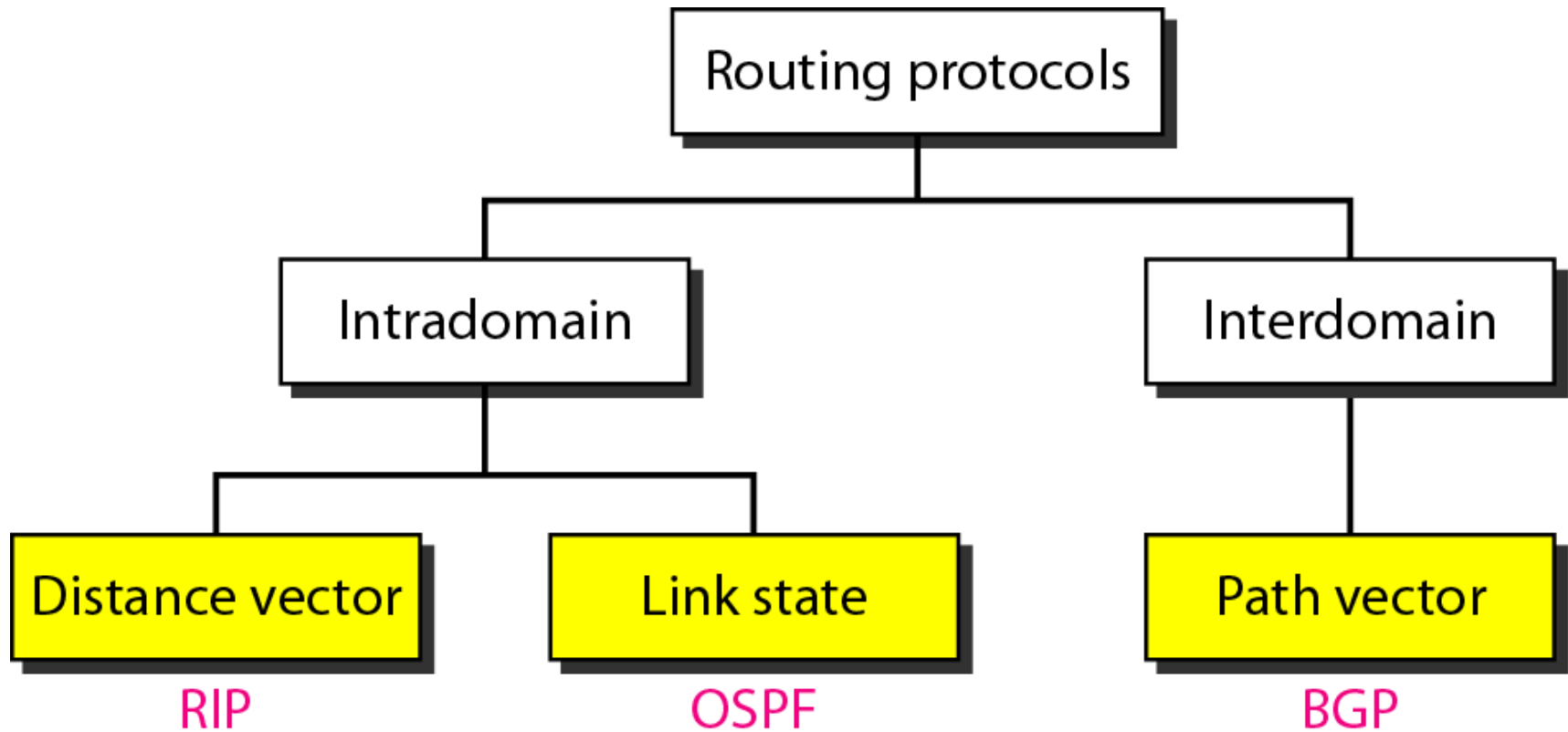


The Internet network layer

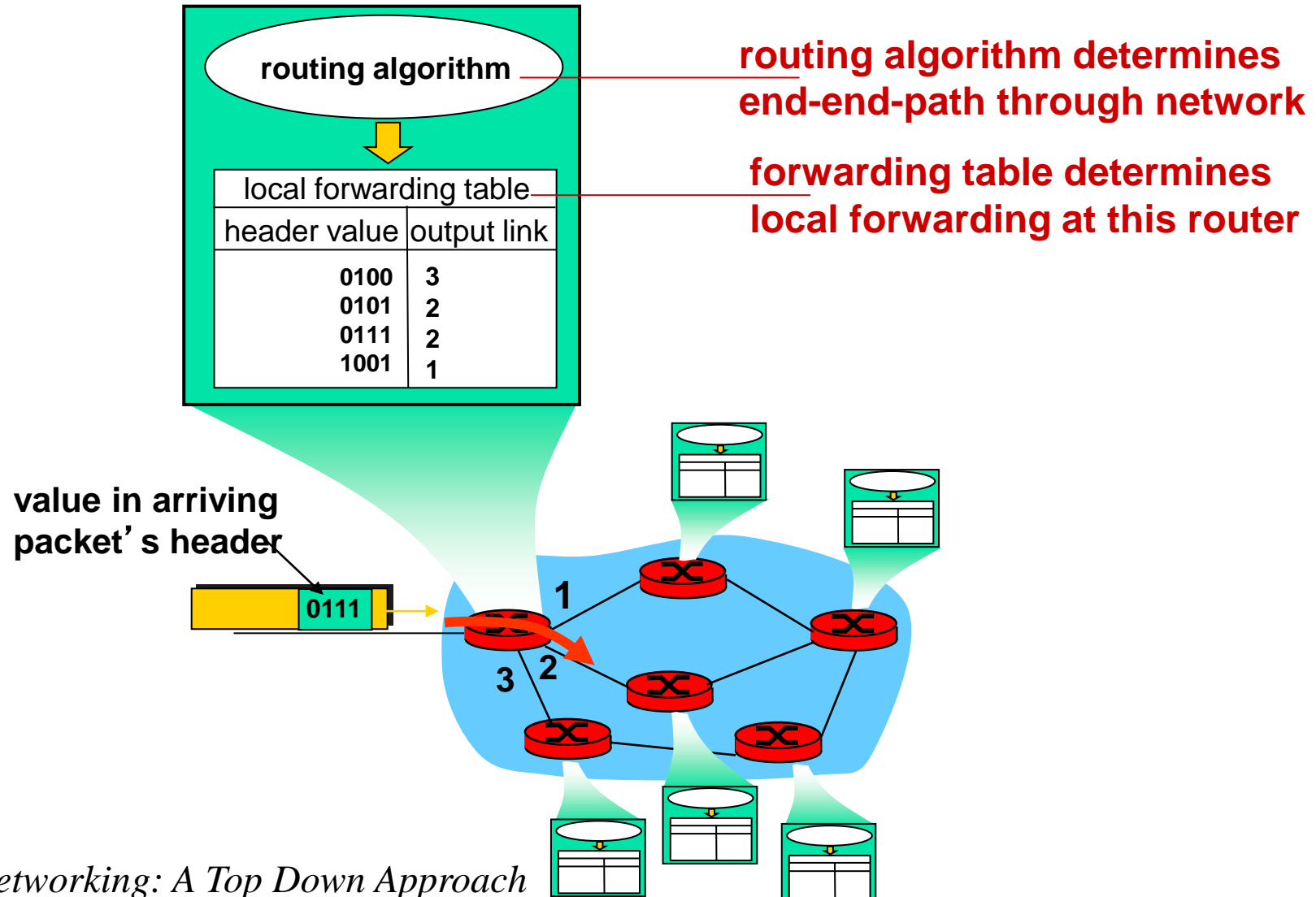
host, router network layer functions:



Popular routing protocols



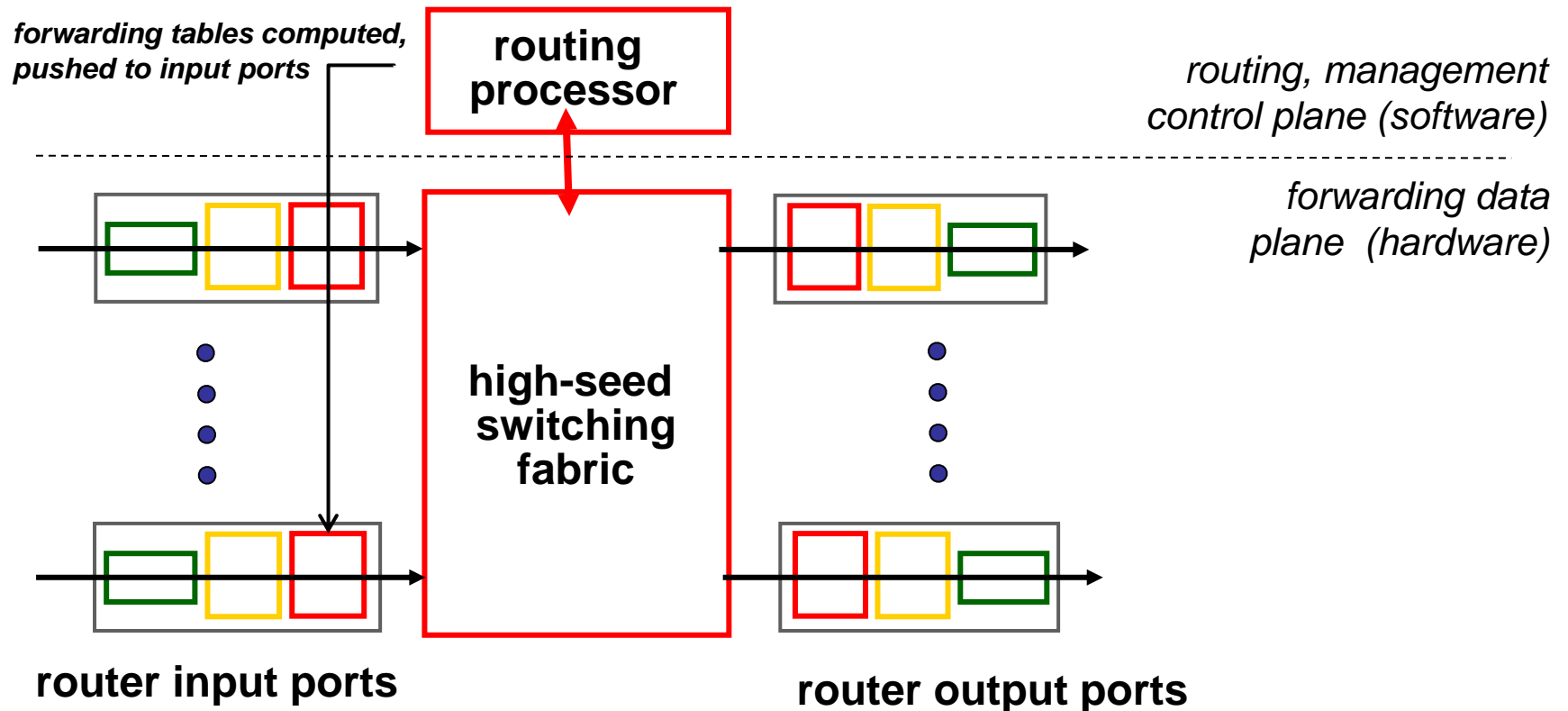
Interplay between routing and forwarding



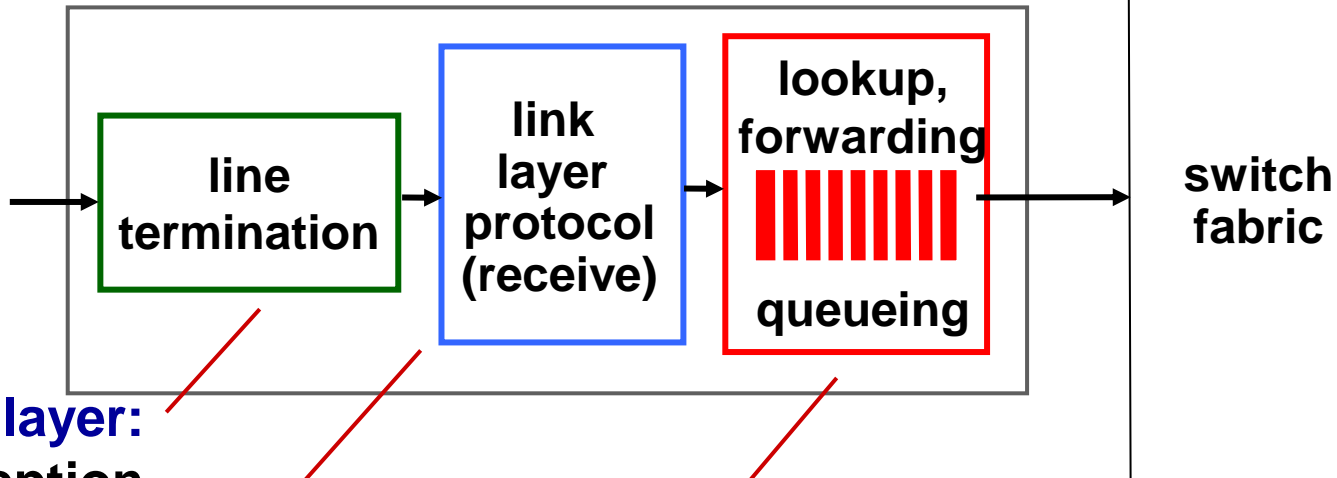
Router architecture overview

two key router functions:

- ❖ run routing algorithms/protocol (RIP, OSPF, BGP)
- ❖ *forwarding* datagrams from incoming to outgoing link



Input port functions



physical layer:
bit-level reception

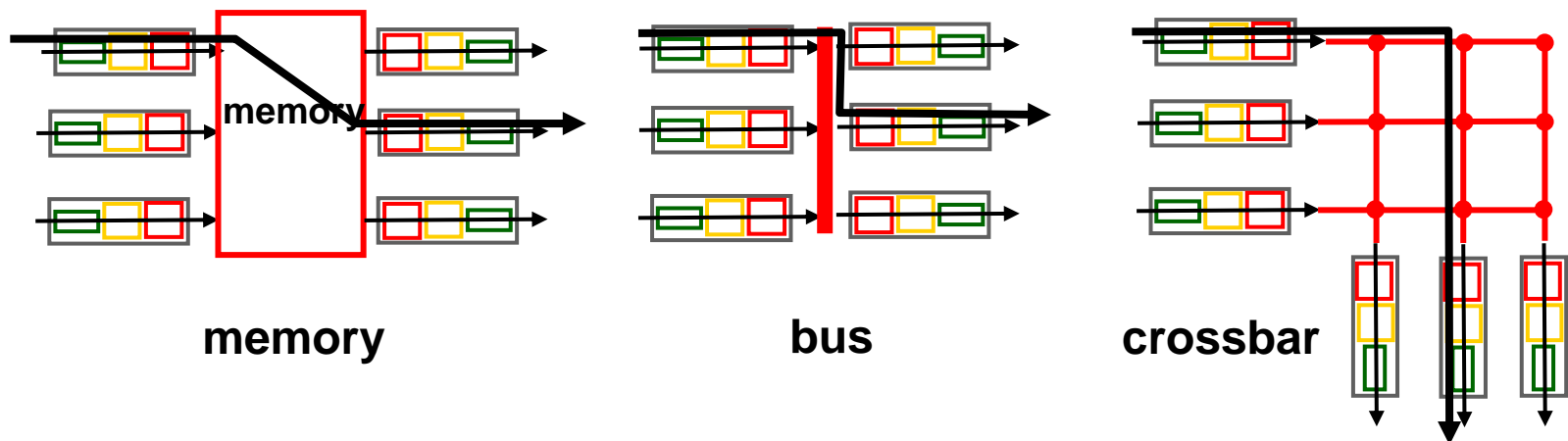
data link layer:
e.g., Ethernet

decentralized switching:

- given datagram dest., lookup output port using forwarding table in input port memory (*"match plus action"*)
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Switching fabrics

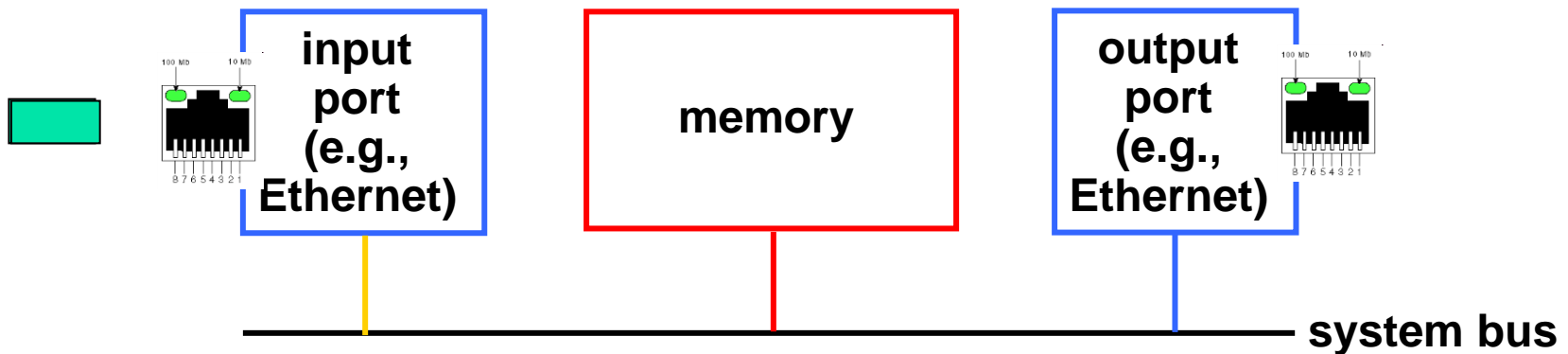
- ❖ transfer packet from input buffer to appropriate output buffer
- ❖ switching rate: rate at which packets can be transfer from inputs to outputs
 - often measured as multiple of input/output line rate
 - N inputs: switching rate N times line rate desirable
- ❖ three types of switching fabrics



Switching via memory

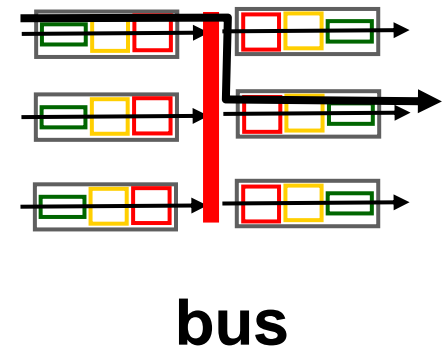
first generation routers:

- 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)



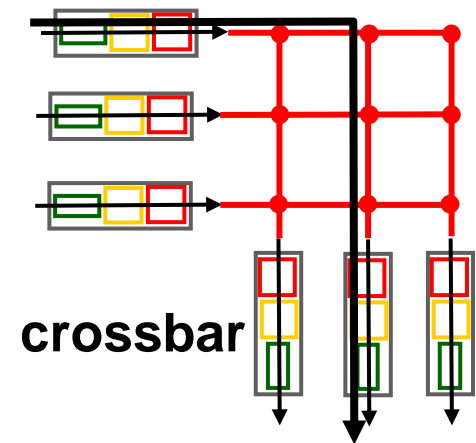
Switching via a bus

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



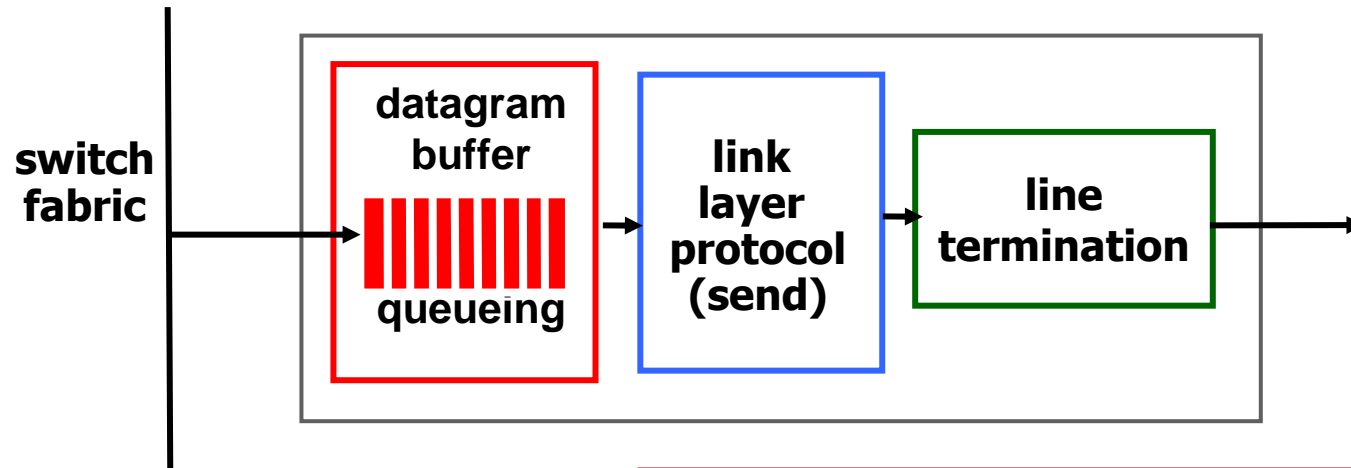
Switching via interconnection network

- ❖ 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.
- ❖ Ex:
 - ❖ Cisco 12000: switches 60 Gbps through the interconnection network



Output ports

This slide important!

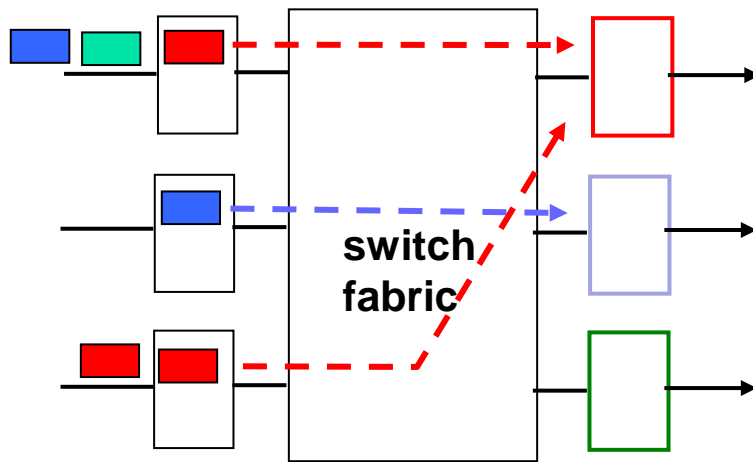


Datagram (packets) can be lost due to congestion, lack of buffers

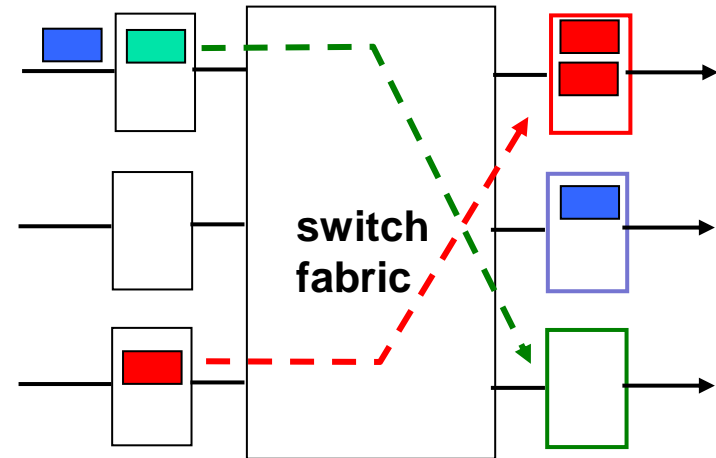
- ❖ *buffering* required when datagrams arrive from fabric faster than the transmission rate
- ❖ *scheduling discipline* chooses among queued datagrams for transmission

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!*

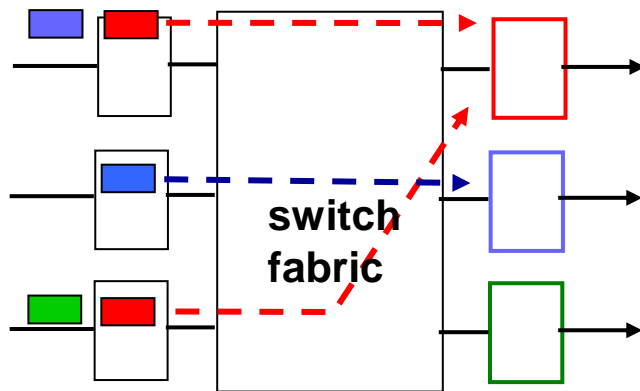
How much buffering?

- RFC 3439 rule of thumb:
 - average buffering equal to “typical” RTT (say 250 msec) times link capacity C
 - e.g., C = 10 Gpbs link: 2.5 Gbit buffer
- recent recommendation: with N flows, buffering equal to

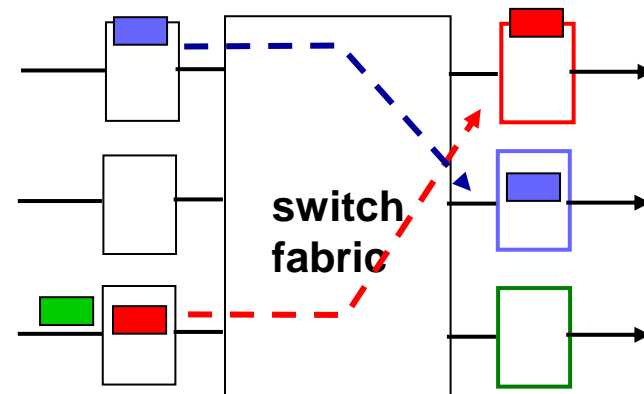
$$\frac{RTT \cdot C}{\sqrt{N}}$$

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*

Routing Strategies

- static routing
 - Fixed
 - Flooding
 - Random
- adaptive (dynamic)
 - continuous
 - periodic
 - considerable load change
 - architecture updates (topology, throughput, etc.)

Static Routing

■ Features

- fixed routes: no use of status variables; DGs and VCs follow the same route
- all static routes are maintained in Central Routing Directory (Routing Control Center)
- local part of the routes are stored at each node directory (consists of destinations and next node to them)
- simple, smallest overload, no flexibility
- application for reliable networks with small variation of traffic parameters

■ Examples

- Generic fixed routing
- (Shortest path routing
- Flooding
- Flow-based routing
- Multipath random routing

Fixed Routing

- Single permanent route for each source to destination pair
- Determine routes using a least cost algorithm
- Route fixed, at least until a change in network topology

CENTRAL ROUTING DIRECTORY

		From Node					
		1	2	3	4	5	6
To Node	1	—	1	5	2	4	5
	2	2	—	5	2	4	5
	3	4	3	—	5	3	5
	4	4	4	5	—	4	5
	5	4	4	5	5	—	5
	6	4	4	5	5	6	—

Node 1 Directory

Destination	Next Node
2	2
3	4
4	4
5	4
6	4

Node 2 Directory

Destination	Next Node
1	1
3	3
4	4
5	4
6	4

Node 3 Directory

Destination	Next Node
1	5
2	5
4	5
5	5
6	5

Node 4 Directory

Destination	Next Node
1	2
2	2
3	5
5	5
6	5

Node 5 Directory

Destination	Next Node
1	4
2	4
3	3
4	4
6	6

Node 6 Directory

Destination	Next Node
1	5
2	5
3	5
4	5
5	5

Shortest Path Routing

■ Features





- based on graph representation of the subnet (node/router, arc/link)
- shortest path is based on
 - the number of arcs between two routers (i.e. number of hops) - uses unmarked graph
 - the accumulated “length” of links (length is measured by the transmission time, transmission price, distance, etc.) - uses weighted graph (labeled arcs)
- algorithms for shortest path from given node to ALL other:
 - **Dijkstra's algorithm** - developing path in order of increasing path length; each node is labeled with the couple (S_L, N_{i-1}) , iterations follow the shortest path to the next node
 - **Bellman-Ford algorithm** - iterative search for all the nodes that are distanced by 0, 1, 2, ... MAX hops
- Dijkstra vs. Bellman-Ford
 - Computational Complexity: Dijkstra's $O(N^2)$ computations, Bellman-Ford $O(N^3)$
 - **D's** algorithm requires knowledge of all costs - better suited to centralized routing decisions
 - **BF's** only requires knowledge of link costs to neighboring nodes - may be implemented in a distributed way
 - Both can adapt to slowly changing link costs

Flooding

■ Features

- based on propagation of every incoming packet to each output line excluding the delivery line
- modification: propagation is pointed to some “proper” subset of output lines - **selective flooding**

■ Effects:

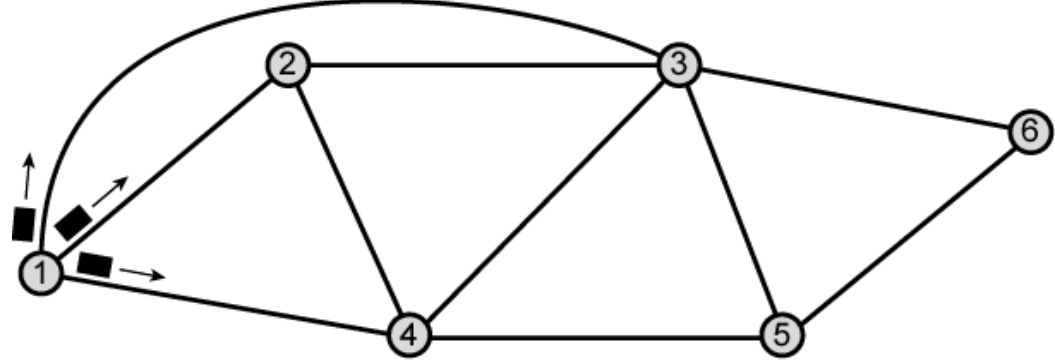
-  exponential growth of the number of packets
-  duplicate packets □ hop limitation
-  guaranteed finding of the shortest path
-  the flooding gives all the shortest paths in the graph

■ Application:

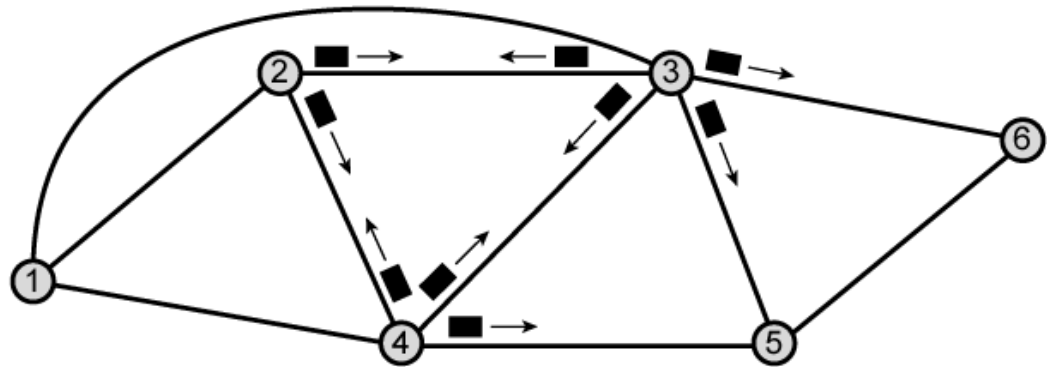
- trouble sensitive applications
- concurrent updates in distributed databases

Properties of Flooding

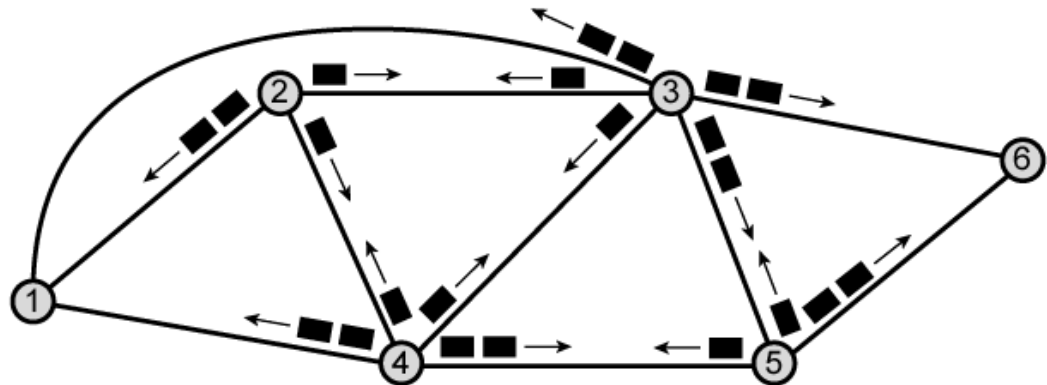
- All possible routes are tried
 - Very robust
- At least one packet will have taken minimum hop count route
 - Can be used to set up virtual circuit
- All nodes are visited
 - Useful to distribute information (e.g. routing)



(a) First hop



(b) Second hop



(c) Third hop

Random Routing

- Node selects one outgoing path for retransmission of incoming packet
- Selection can be random or round robin
- Can select outgoing path based on probability calculation
- No network info needed
- Route is typically not least cost nor minimum hop

Adaptive Routing

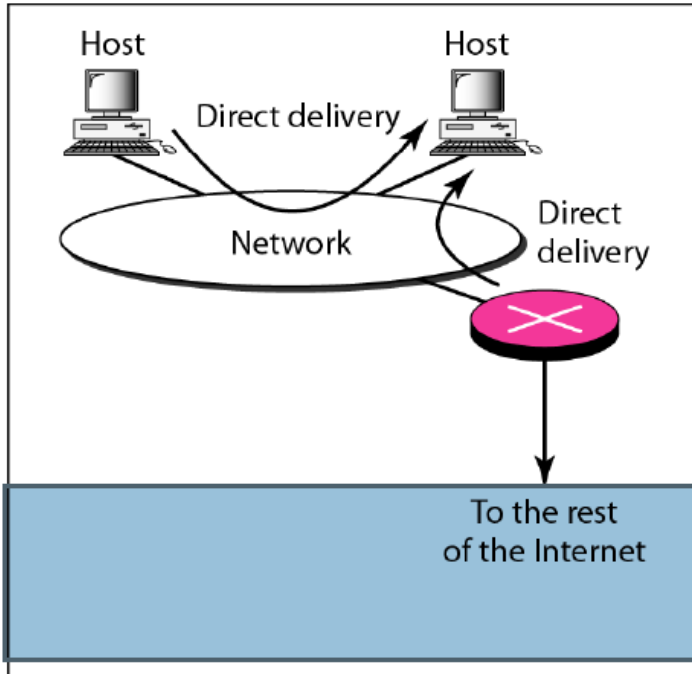
- Used by almost all packet switching networks
- Routing decisions change as conditions on the network change
 - Failure
 - Congestion
- Requires info about network
- Decisions more complex
- Tradeoff between quality of network info and overhead
- Reacting too quickly can cause oscillation
- Too slowly to be relevant

Adaptive Routing - Advantages

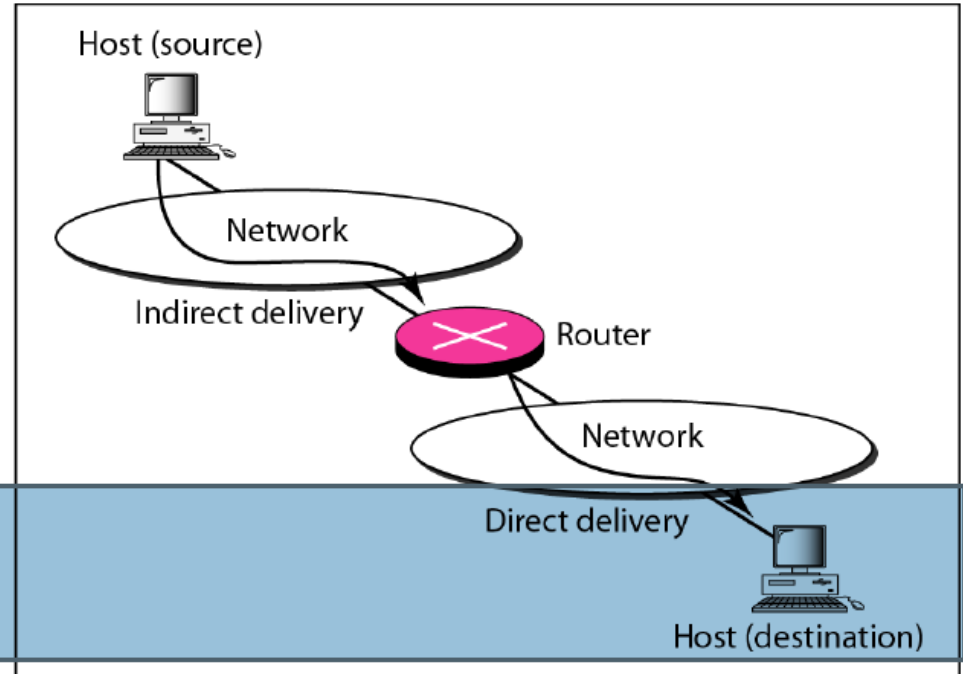
- Improved performance
- Aid congestion control
- Complex system
 - May not realize theoretical benefits

- Routing approaches
 - Next-hop routing
 - Network-specific routing
 - Host-specific routing
 - Default routing
 - Classful addressing routing table

Direct and indirect delivery



a. Direct delivery



b. Indirect and direct delivery

Next-hop routing

Routing table for host A

Destination	Route
Host B	R1, R2, Host B

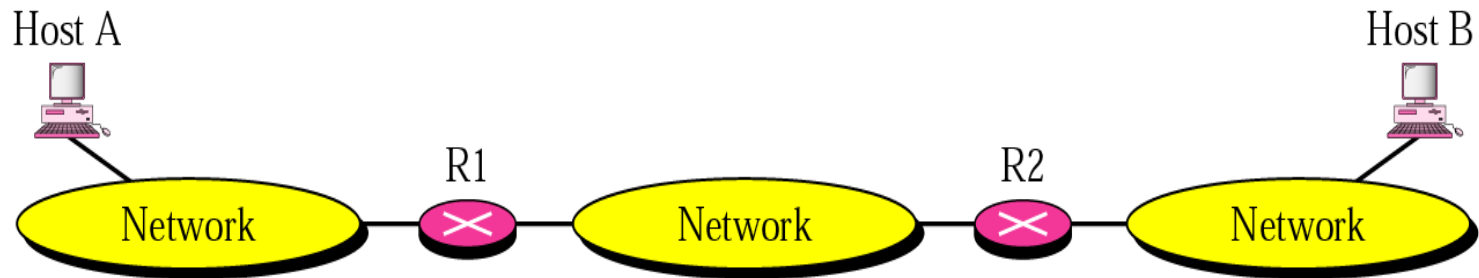
Routing table for R1

Destination	Route
Host B	R2, Host B

Routing table for R2

Destination	Route
Host B	Host B

a. Routing tables based on route



Routing table for host A

Destination	Next Hop
Host B	R1

Routing table for R1

Destination	Next Hop
Host B	R2

Routing table for R2

Destination	Next Hop
Host B	—

b. Routing tables based on next hop

Next Hop Routing is a technique to reduce the contents of a routing table.

The routing table holds only the information that leads to the next hop instead of holding information about the complete route.

Network-specific routing

Network-specific routing is a technique to reduce the routing table and simplify the searching process.

Instead of having an entry for every host connected to the same physical address, we have only one entry to define the address of network itself.

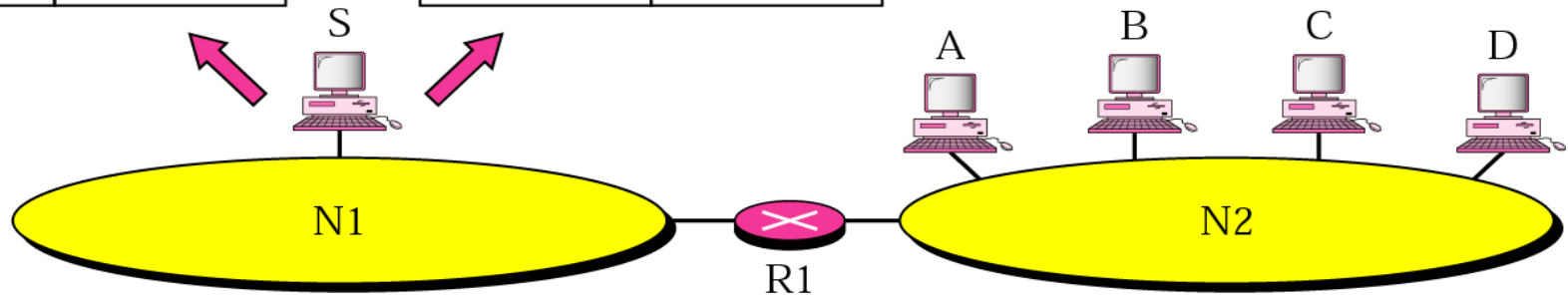
We treat all hosts to the same network as one single entry.

Routing table for host S based
on host-specific routing

Destination	Next Hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based
on network-specific routing

Destination	Next Hop
N2	R1



Host-specific routing

In **host-specific routing** the destination host address is given in the routing table.

The idea is inverse of **network-specific network**.

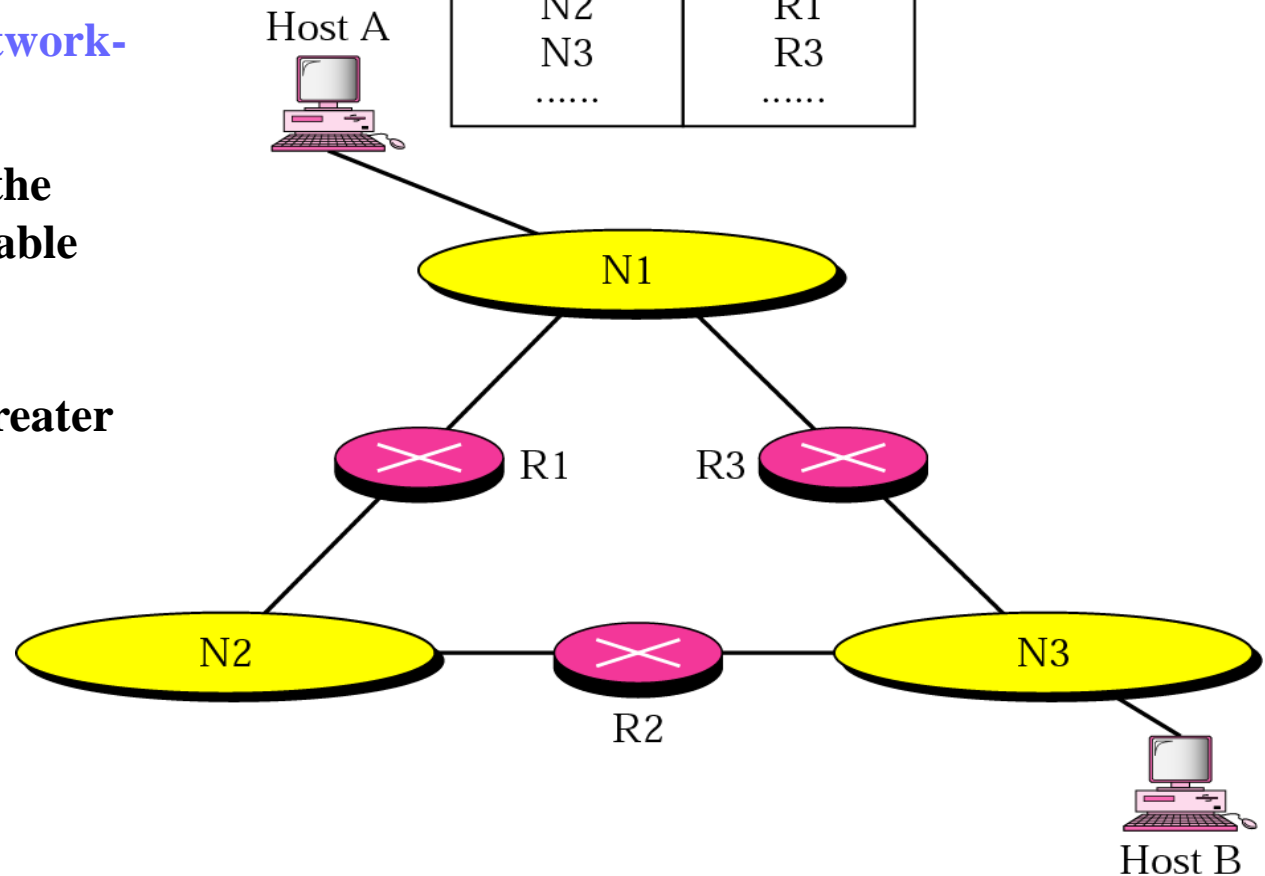
It is not efficient to put the host address in routing table

BUT

The administrator has greater control over routing

Routing table for host A

Destination	Next Hop
Host B	R3
N2	R1
N3	R3
.....



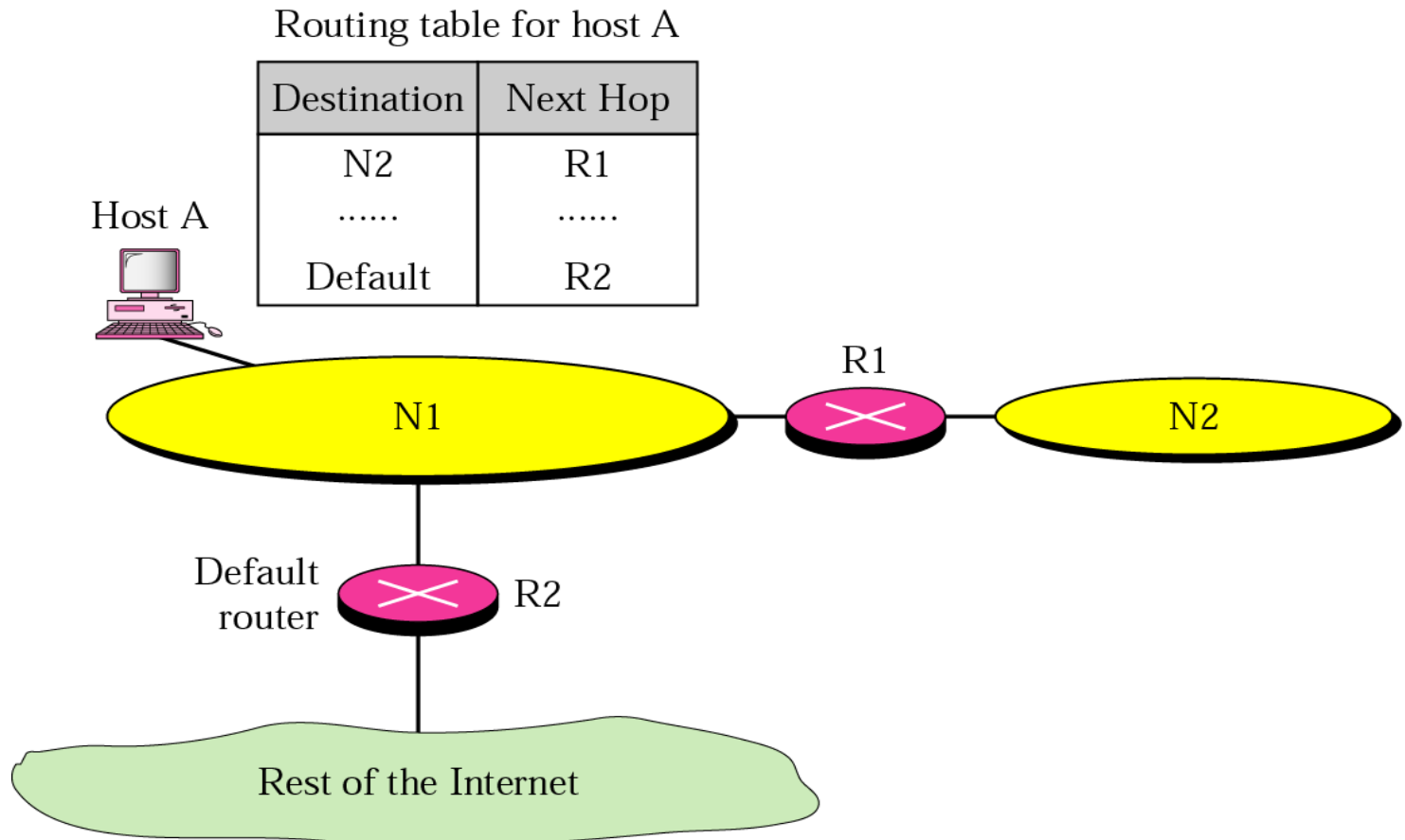
Default routing

Host A is connected to a network with two routers

R1 is used to route the packets to holds connected to network N2

R2 is used to connect to the rest of INTERNET.

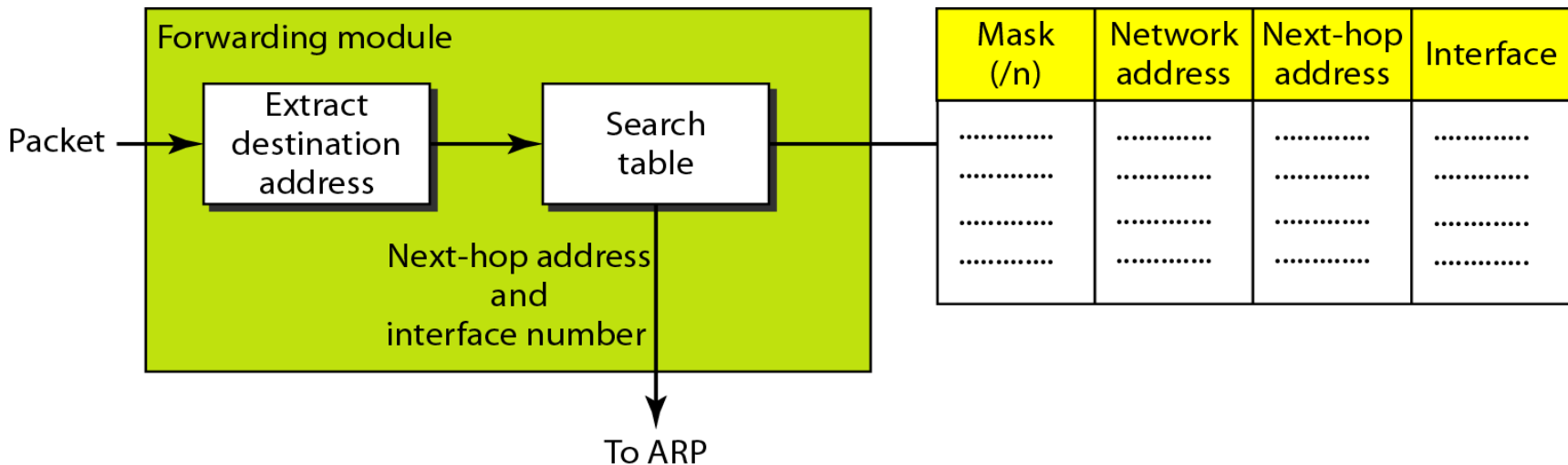
Instead of listing all networks in the entire INTERNET , host A can just have one entry call **DEFAULT** (network address 0.0.0.0)



Addressing routing table

In addressing, with or without subnetting, a routing table needs a minimum of columns (it normally has more):

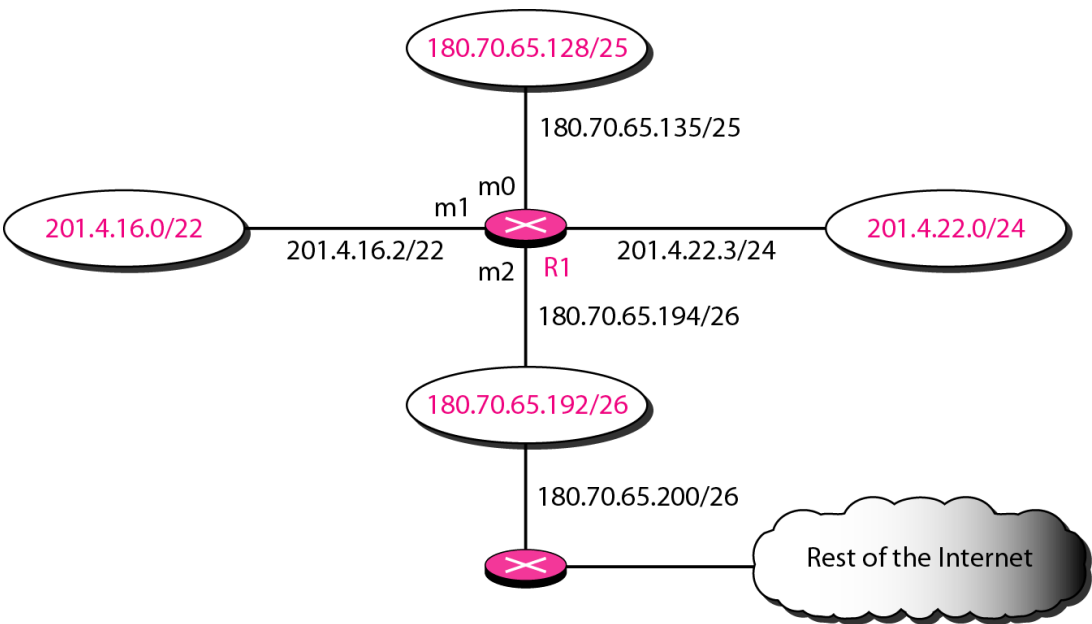
- mask,
- destination network address,
- next hop address
- interface.



When a packet arrives, the router applies the **mask to the destination address** to **find network address**.

If found, the packet is sent out from the **corresponding interface** in the table.

If not found, the packet is **delivered to the default interface** which carries the packet to the default router.



Mask	Network Address	Next Hop	Interface
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	m1
Any	Any	180.70.65.200	m2

B. Show the forwarding process if a packet arrives at R1 with the destination address 201.4.22.35.

Solution

The router performs the following steps:

1. The first mask (/26) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address.
2. The second mask (/25) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 2).
3. The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address. The destination address of the packet and the interface number m3 are passed to ARP.

A. Show the forwarding process if a packet arrives at R1 with the destination address 180.70.65.140.

Solution

The router performs the following steps:

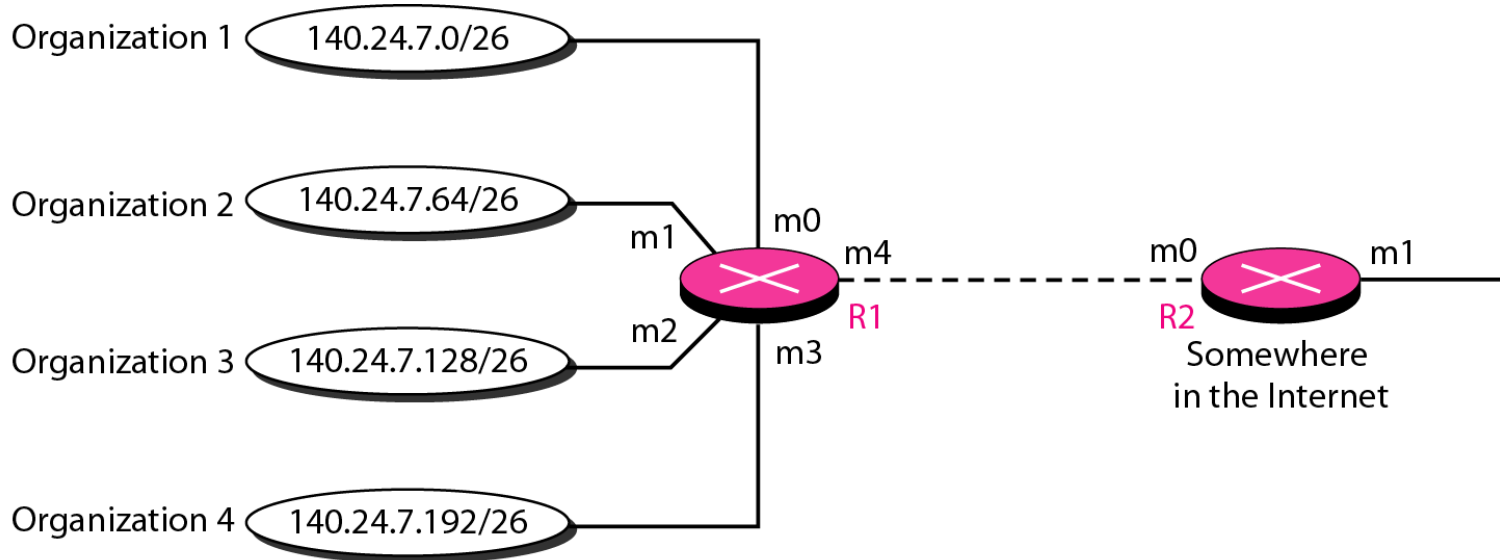
1. The first mask (/26) is applied to the destination address. The result is 180.70.65.128, which does not match the corresponding network address.
2. The second mask (/25) is applied to the destination address.
3. The result is 180.70.65.128, which matches the corresponding network address.
4. The next-hop address and the interface number m0 are passed to ARP for further processing.

C. Show the forwarding process if a packet arrives at R1 with the destination address 18.24.32.78.

Solution

This time all masks are applied, one by one, to the destination address, but no matching network address is found. When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP. This is probably an outgoing package that needs to be sent, via the default router, to someplace else in the Internet.

Address aggregation



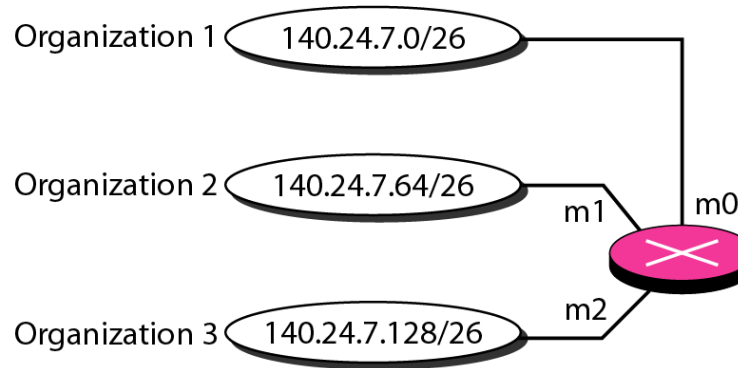
Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	-----	m0
/26	140.24.7.64	-----	m1
/26	140.24.7.128	-----	m2
/26	140.24.7.192	-----	m3
/0	0.0.0.0	Default	m4

Routing table for R1

Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	-----	m0
/0	0.0.0.0	Default	m1

Routing table for R2

Longest mask matching

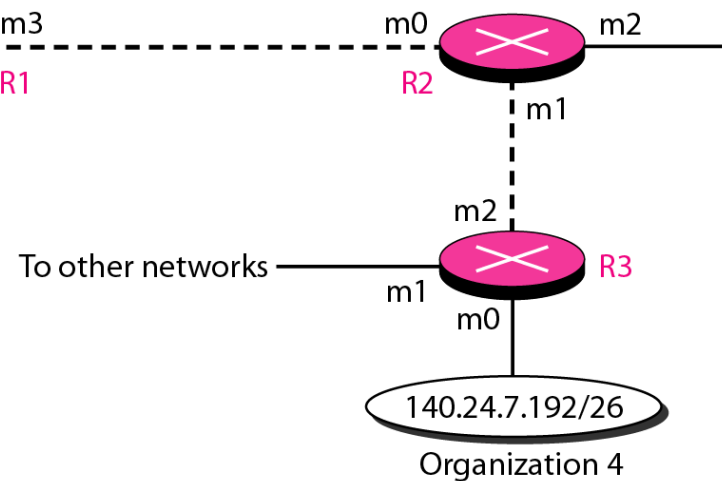


Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	-----	m0
/26	140.24.7.64	-----	m1
/26	140.24.7.128	-----	m2
/0	0.0.0.0	Default	m3

Routing table for R1

Routing table for R2

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	-----	m1
/24	140.24.7.0	-----	m0
/??	???????	?????????	m1
/0	0.0.0.0	Default	m2



Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	-----	m0
/??	???????	?????????	m1
/0	0.0.0.0	Default	m2

Routing table for R3

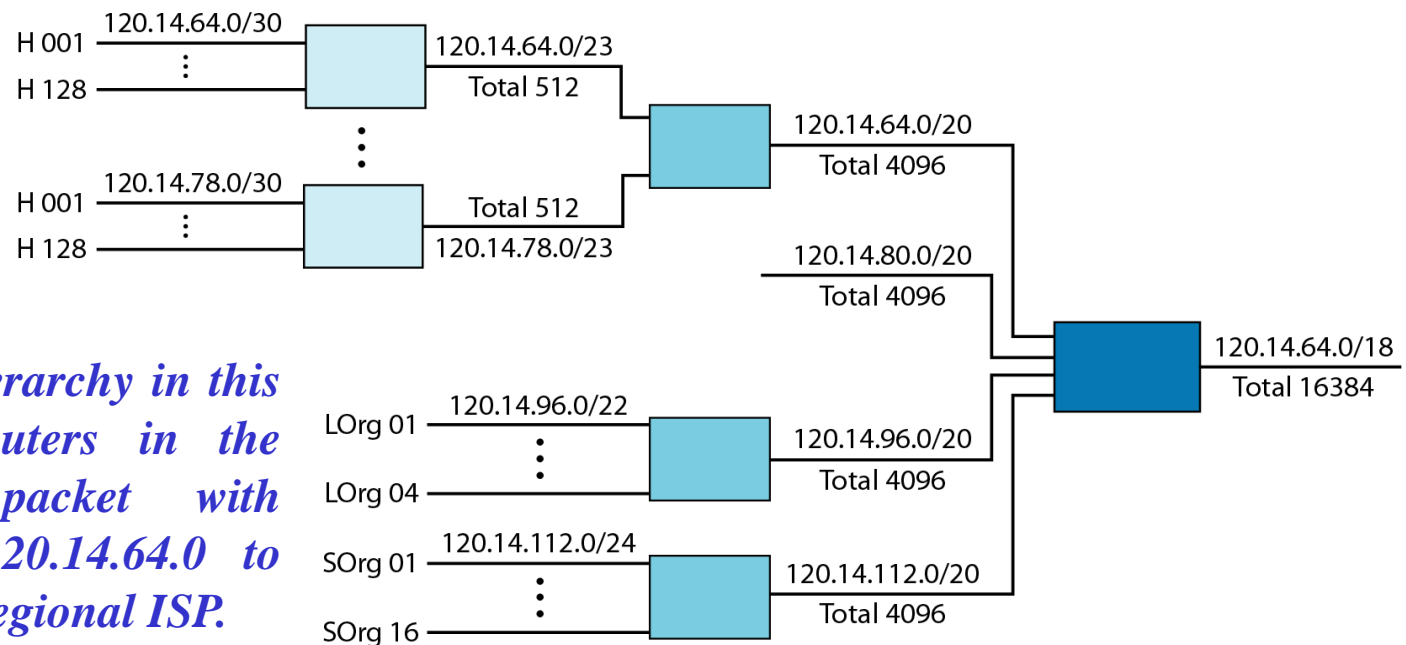
Example 22.5

As an example of hierarchical routing, let us consider regional ISP is granted **16,384** addresses starting from **120.14.64.0**.

The regional ISP has decided to divide this block into four subblocks, each with **4096** addresses.

Three of these subblocks are assigned to three local ISPs; the second subblock is reserved for future use.

Note that the mask for each block is **/20** because the original block with mask **/18** is divided into 4 blocks.



The first local ISP has divided its assigned subblock into 8 smaller blocks and assigned each to a small ISP. Each small ISP provides services to 128 households, each using four addresses.

The second local ISP has divided its block into 4 blocks and has assigned the addresses to four large organizations.

The third local ISP has divided its block into 16 blocks and assigned each block to a small organization. Each small organization has 256 addresses, and the mask is **/24**.

*There is a sense of hierarchy in this configuration. All routers in the Internet send a packet with destination address **120.14.64.0** to **120.14.127.255** to the regional ISP.*

*One utility that can be used to find the contents of a routing table for a host or router is **netstat** in UNIX or LINUX. The next slide shows the list of the contents of a default server.*

*We have used two options, **r** and **n**.*

- the option **r** indicates that we are interested in the routing table,*
- the option **n** indicates that we are looking for numeric addresses.*

Note that this is a routing table for a host, not a router. Although we discussed the routing table for a router throughout the chapter, a host also needs a routing table.

Example (continued)

```
$ netstat -rn
```

```
Kernel IP routing table
```

Destination	Gateway	Mask	Flags	Iface
153.18.16.0	0.0.0.0	255.255.240.0	U	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	lo
0.0.0.0	153.18.31.254	0.0.0.0	UG	eth0

The destination column here defines the network address. The term gateway used by UNIX is synonymous with router. This column actually defines the address of the next hop.

The value 0.0.0.0 shows that the delivery is direct.

The last entry has a flag of G, which means that the destination can be reached through a router (default router). The Iface defines the interface.

*More information about the IP address and physical address of the server can be found by using the **ifconfig** command on the given interface (eth0).*

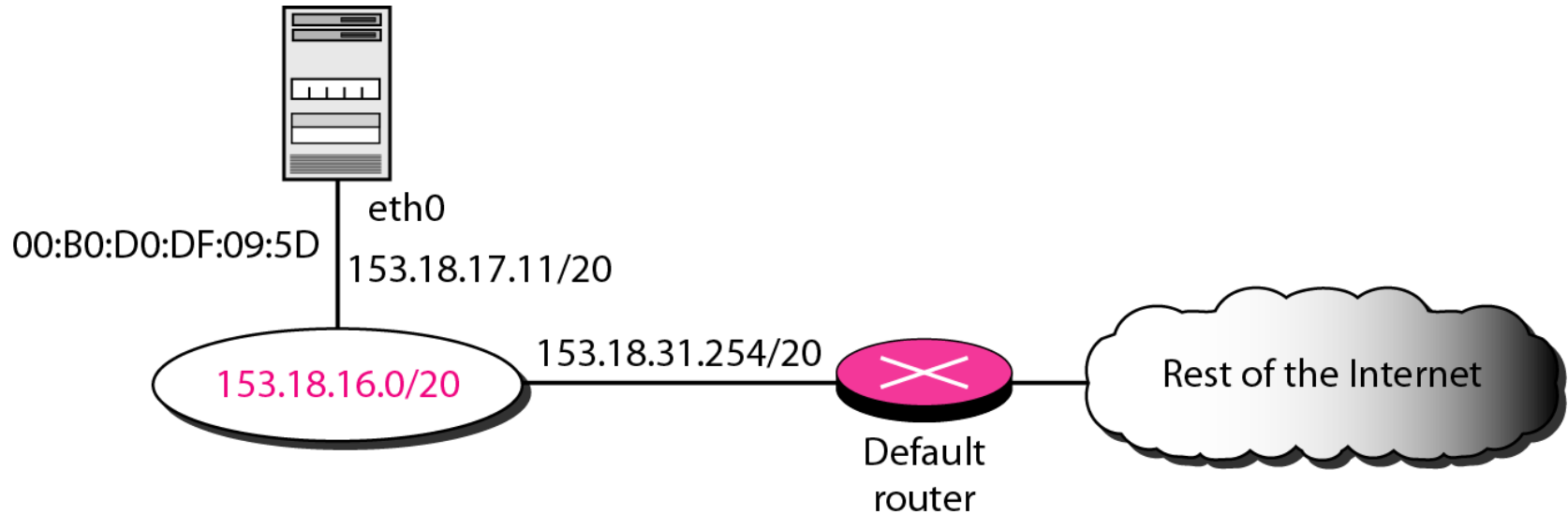
```
$ ifconfig eth0
```

```
eth0  Link encap:Ethernet  HWaddr 00:B0:D0:DF:09:5D
```

```
inet addr:153.18.17.11  Bcast:153.18.31.255  Mask:255.255.240.0
```

```
...
```

Common fields in a routing table



Mask	Network address	Next-hop address	Interface	Flags	Reference count	Use
.....

Example

Using the table 1, the router receives a packet for destination 192.16.7.1.

For each row, the mask is applied to the destination address until a match with the destination address is found.

In this example, the router sends the packet through interface m0 (host specific).

/32= 11111111.11111111.11111111.11111111
192 . 16 . 7 . 1

Table 1

	Mask	Destination address	Next-hop address	Interface
	/8	14.0.0.0	118.45.23.8	m1
Host-specific →	/32	192.16.7.1	202.45.9.3	m0 ←
	/24	193.14.5.0	84.78.4.12	m2
Default →	/0	/0	145.11.10.6	m0

Example

Using the table 2, the router receives a packet for destination 193.14.5.22. For each row, the mask is applied to the destination address until a match with the next-hop address is found.

In this example, the router sends the packet through **interface m2** (network specific).

/32=11111111.11111111.11111111.00000000

193 . 14 . 5 . 22

Table 2:

		Mask	Destination address	Next-hop address	Interface
		/8	14.0.0.0	118.45.23.8	m1
Host-specific	→	/32	192.16.7.1	202.45.9.3	m0
		/24	193.14.5.0	84.78.4.12	m2
Default	→	/0	/0	145.11.10.6	m0

Example

Using the table 3, the router receives a packet for destination 200.34.12.34. For each row, the mask is applied to the destination address, but no match is found. In this example, the router sends the packet through the default interface m0.

Table 3

	Mask	Destination address	Next-hop address	Interface
	/8	14.0.0.0	118.45.23.8	m1
Host-specific →	/32	192.16.7.1	202.45.9.3	m0
	/24	193.14.5.0	84.78.4.12	m2
Default →	/0	/0	145.11.10.6	m0 ←