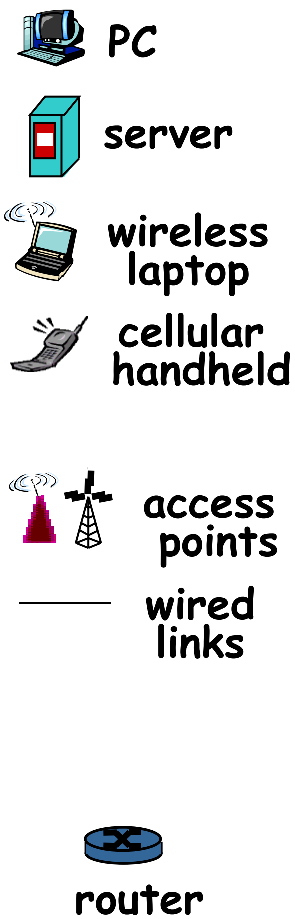
Department of Computer and IT Engineering
University of Kurdistan

Advanced Computer Networks

Network Layer

By: Dr. Alireza Abdollahpouri

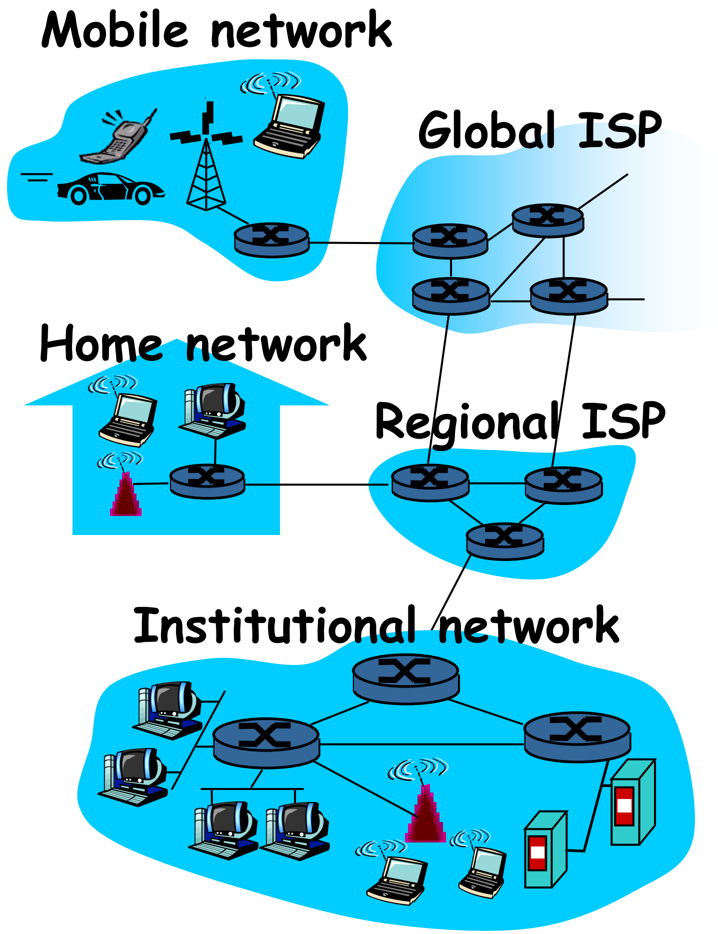
What’s the Internet: “nuts and bolts” view



- millions of connected computing devices: *hosts = end systems*
  - running *network apps*

- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*

- *routers:* forward packets (chunks of data)

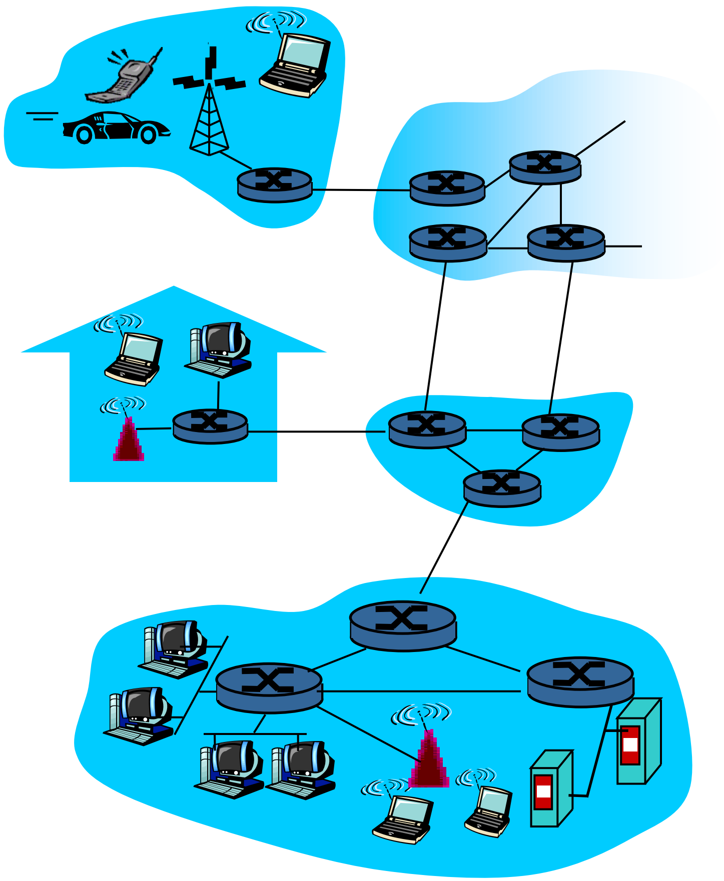


A closer look at network structure:

- network edge: applications and hosts

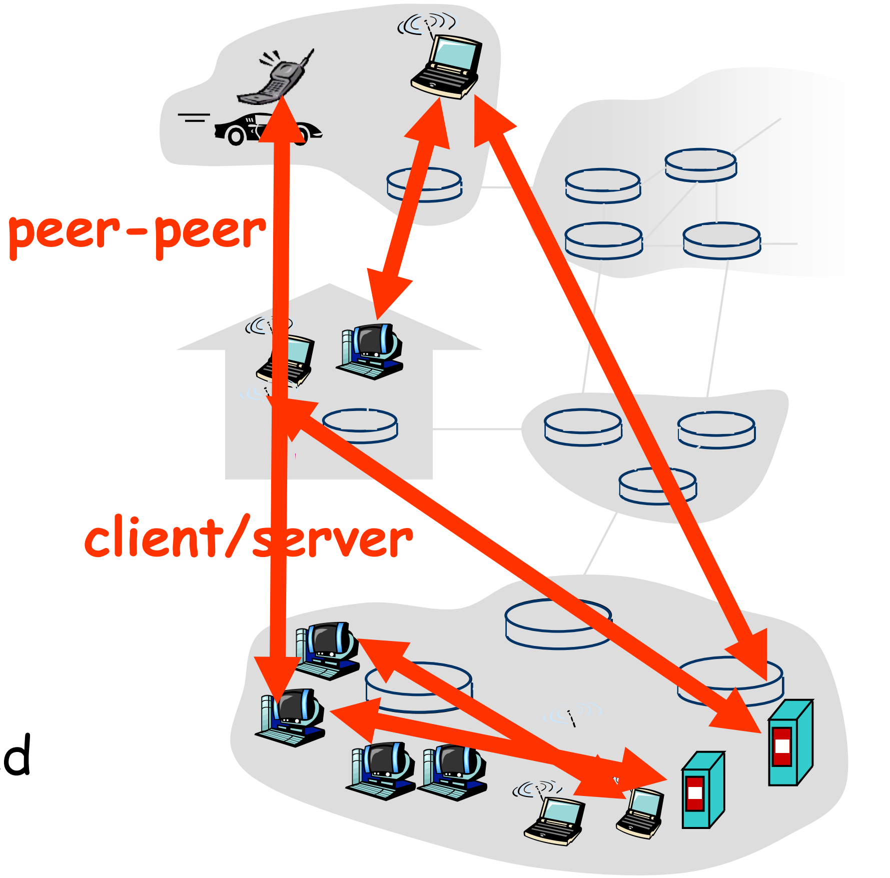
- access networks, physical media: wired, wireless communication links

- network core:
  - interconnected routers
  - network of networks



The network edge:

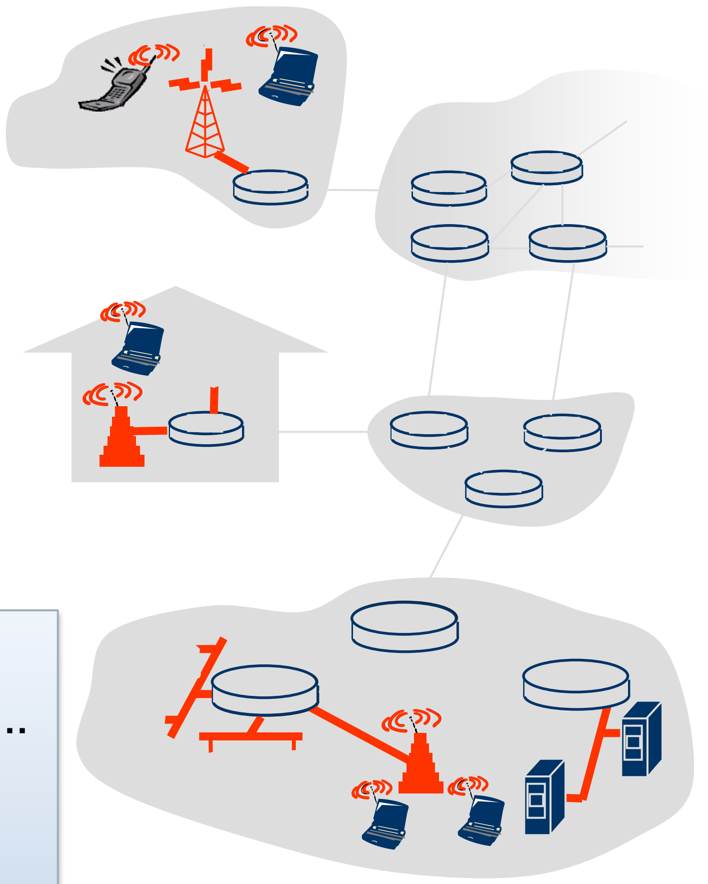
- end systems (hosts):
  - run application programs
  - e.g. Web, email
  - at "edge of network"
- client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- peer-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent



Access networks and physical media

Q: How to connect end systems to edge router?

- residential access nets
- institutional access networks (school, company)
- mobile access networks

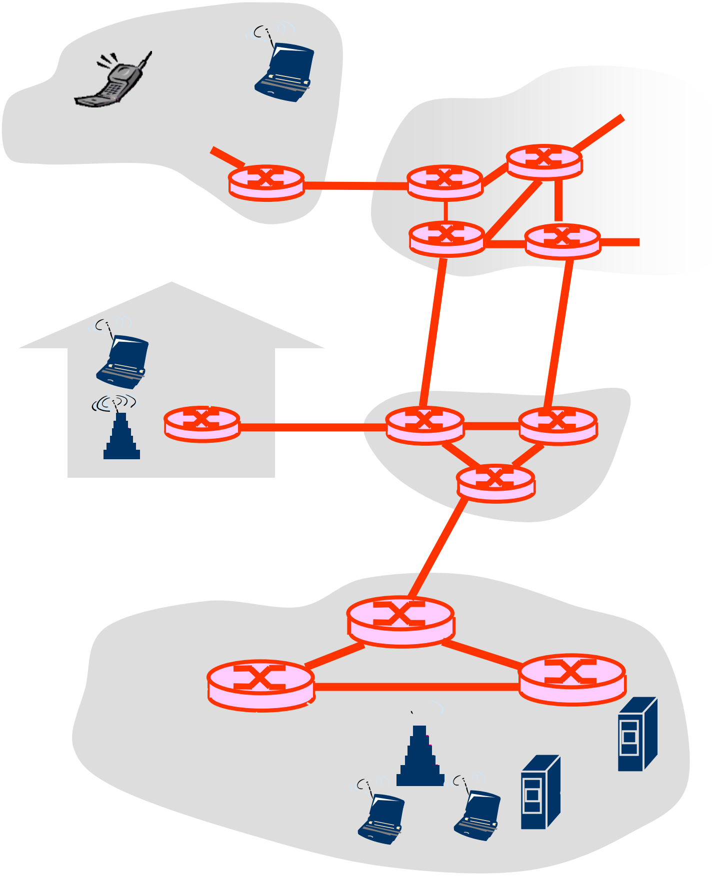


Wired access networks:
xDSL (ADSL, VDSL, SDSL), FTTx (FTTH, FTTC, FTTP), …

Wireless access networks:
WiFi, WiMAX, LTE, …

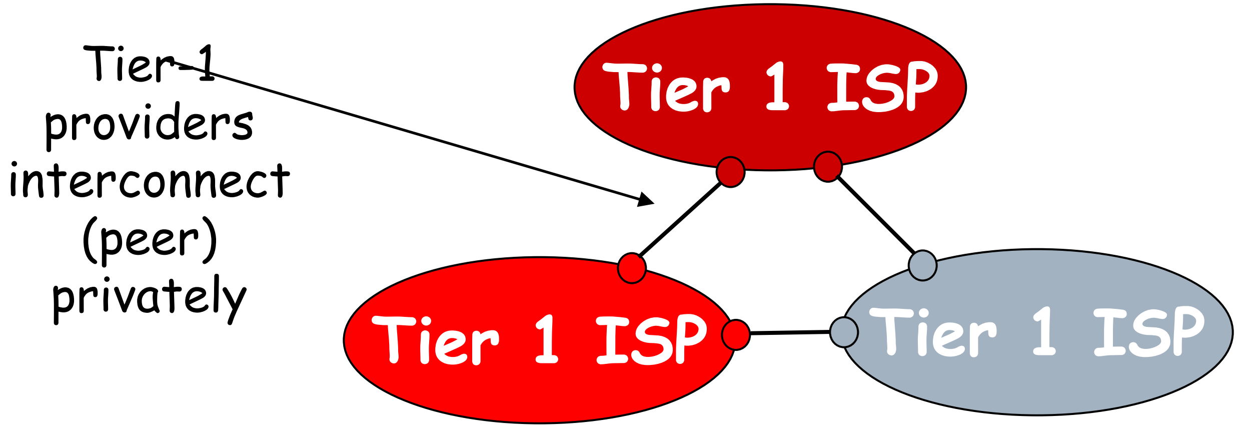
The Network Core

- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete “chunks”

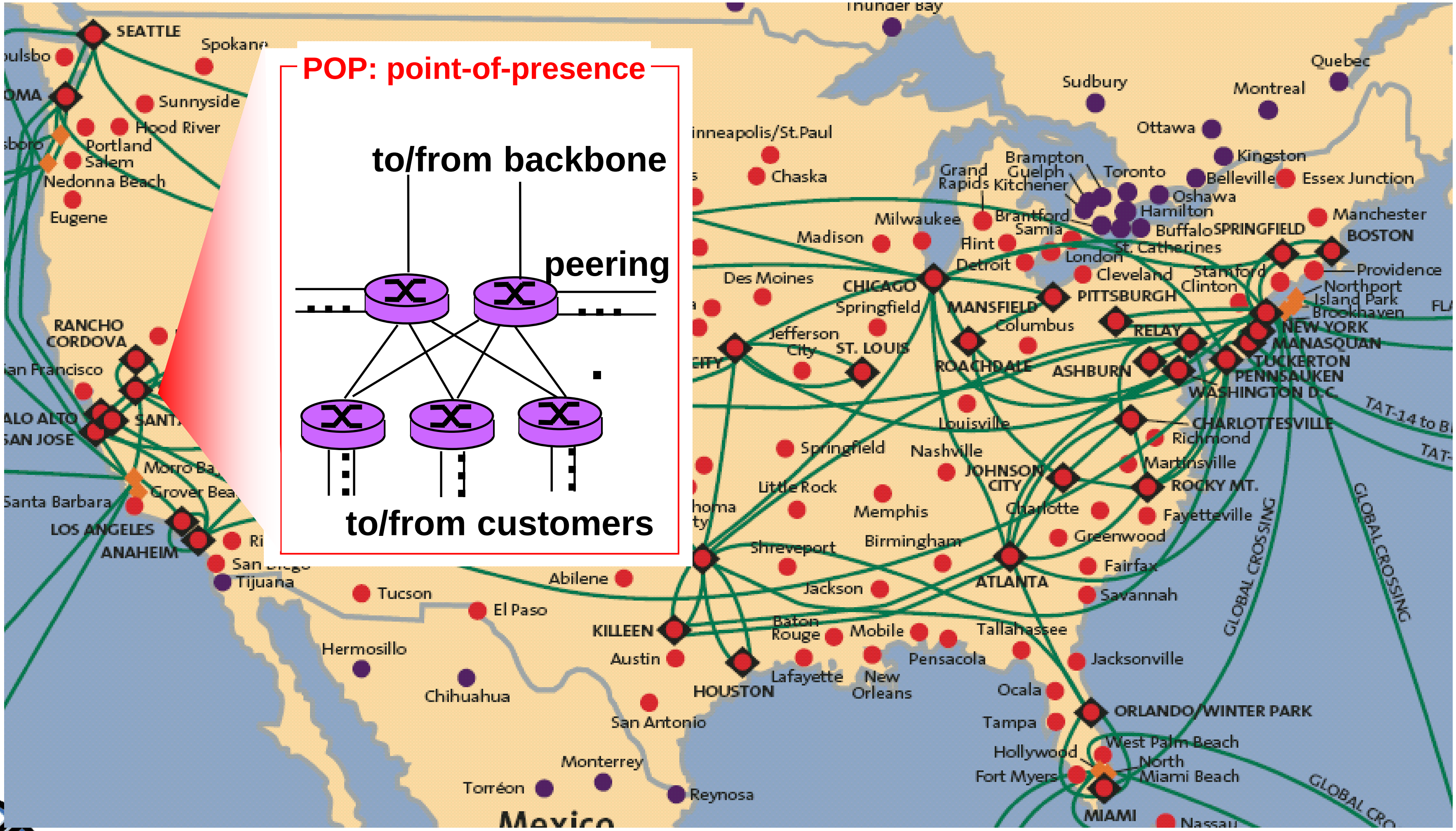


Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (e.g., Verizon, Sprint, AT&T, Cable and Wireless), national/international coverage
  - treat each other as equals

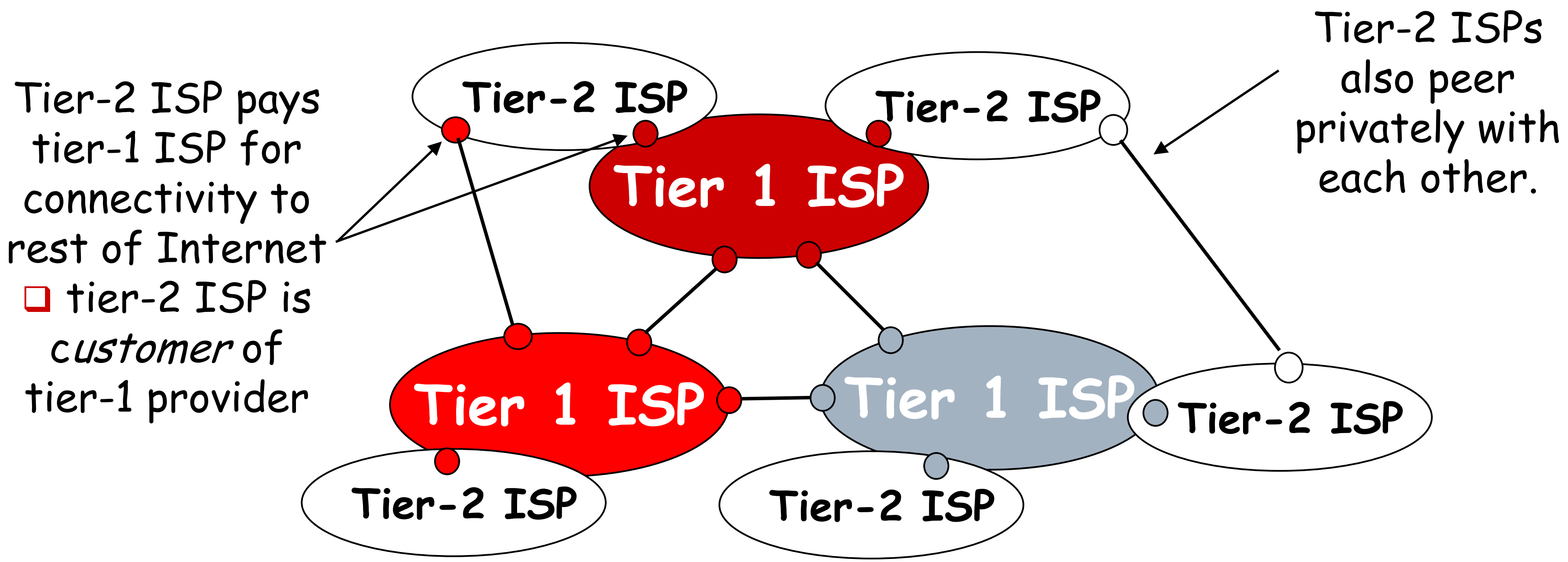


Tier-1 ISP: e.g., Sprint



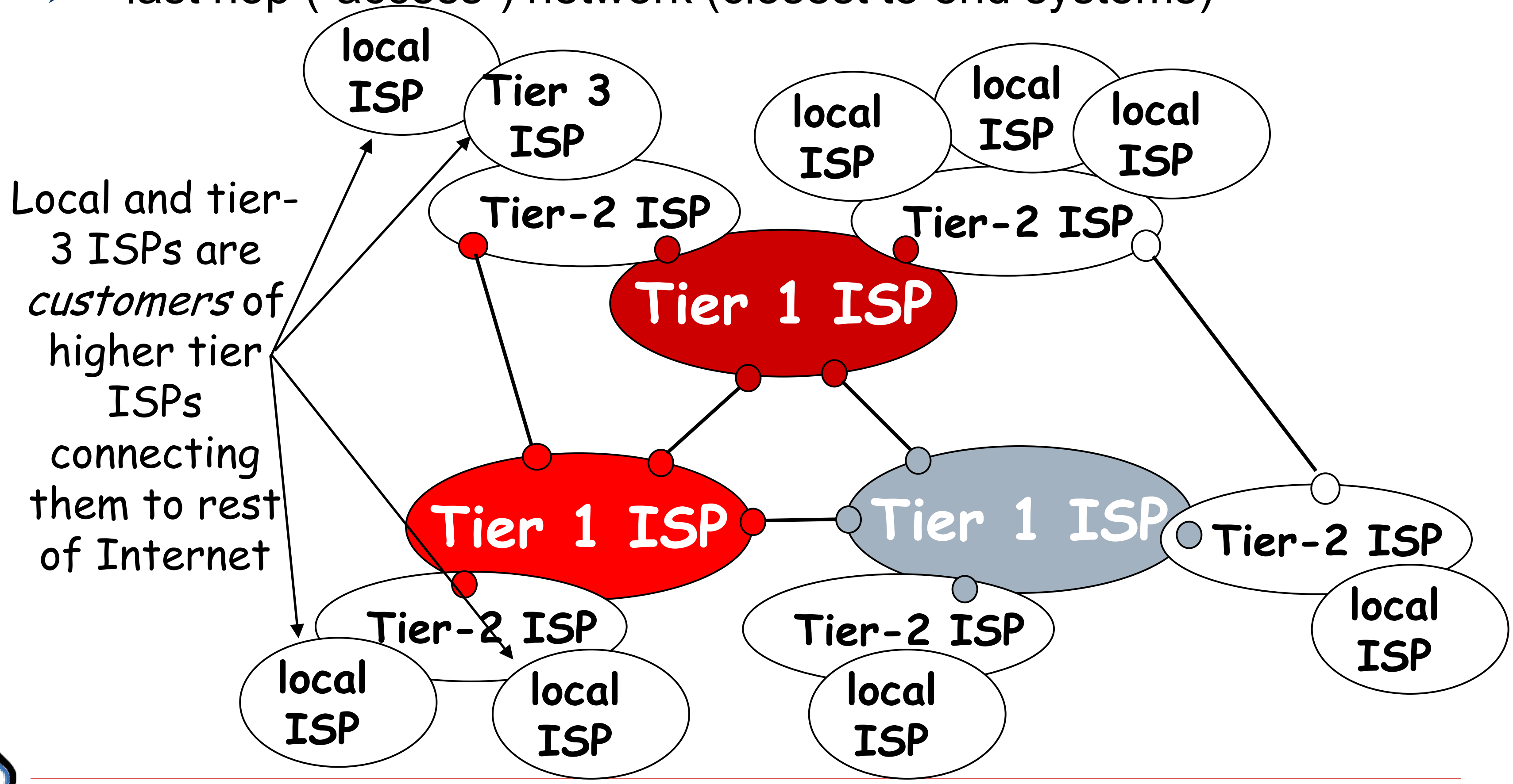
Internet structure: network of networks

- "Tier-2" ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs



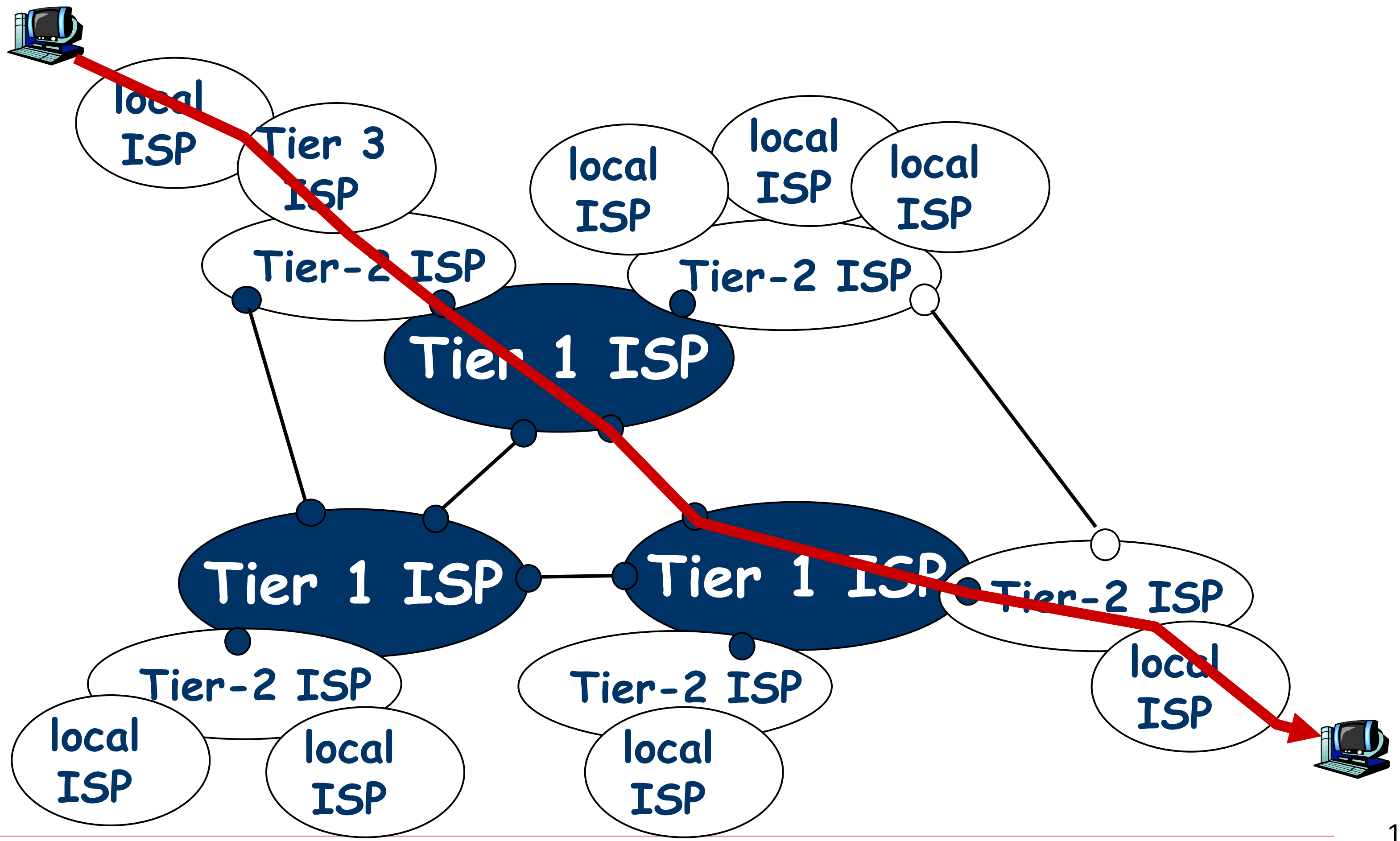
Internet structure: network of networks

- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)



Internet structure: network of networks

- a packet passes through many networks!

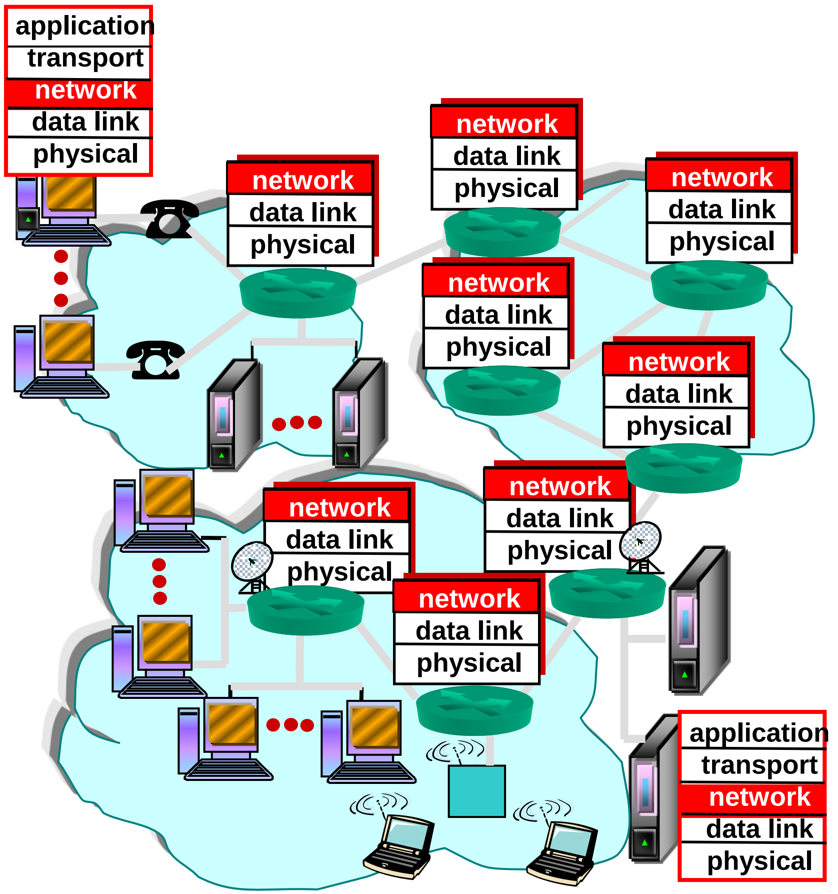


Network Layer Functions

- transport packet from sending to receiving hosts
- network layer protocols in ***every*** host, router

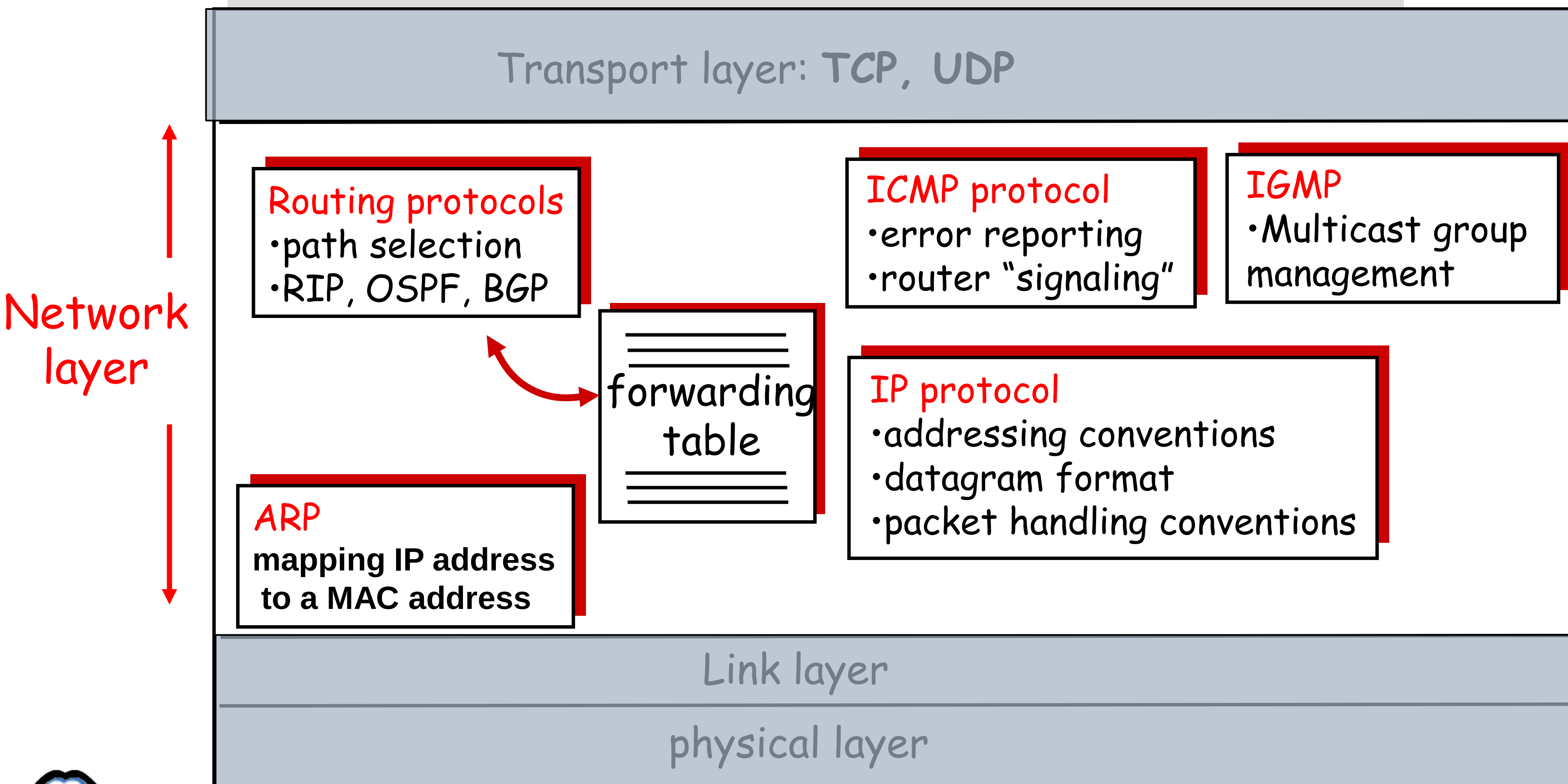
Three important functions:

- *path determination:* route taken by packets from source to dest. (*Routing Algorithms)*
- *forwarding:* move packets from router's input to appropriate router output
- *call setup:* some network architectures require router call setup along path before data flows



The Internet Network layer

Host, router network layer functions:



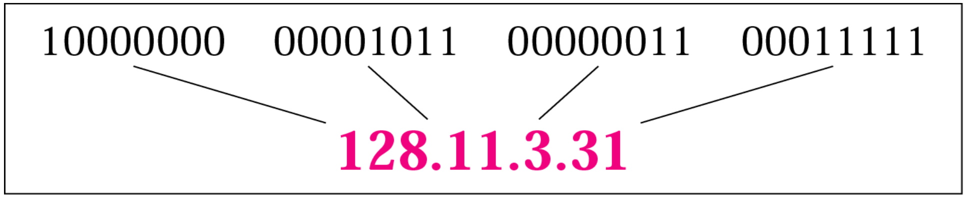
Internet Protocol (IP)

Internet Protocol (IP)

- **Connectionless, unreliable transmission of packets**
- **"Best Effort" Service**
- **IP addressing (IPv4)**
  - Uses logical 32-bit addresses
  - Hierarchical addressing
- **Fragmenting and reassembling of packets**
  - Maximum packet size: 64 kByte
  - In practice: 1500 byte
- **At present commonly used: Version 4 of IP (IPv4)**

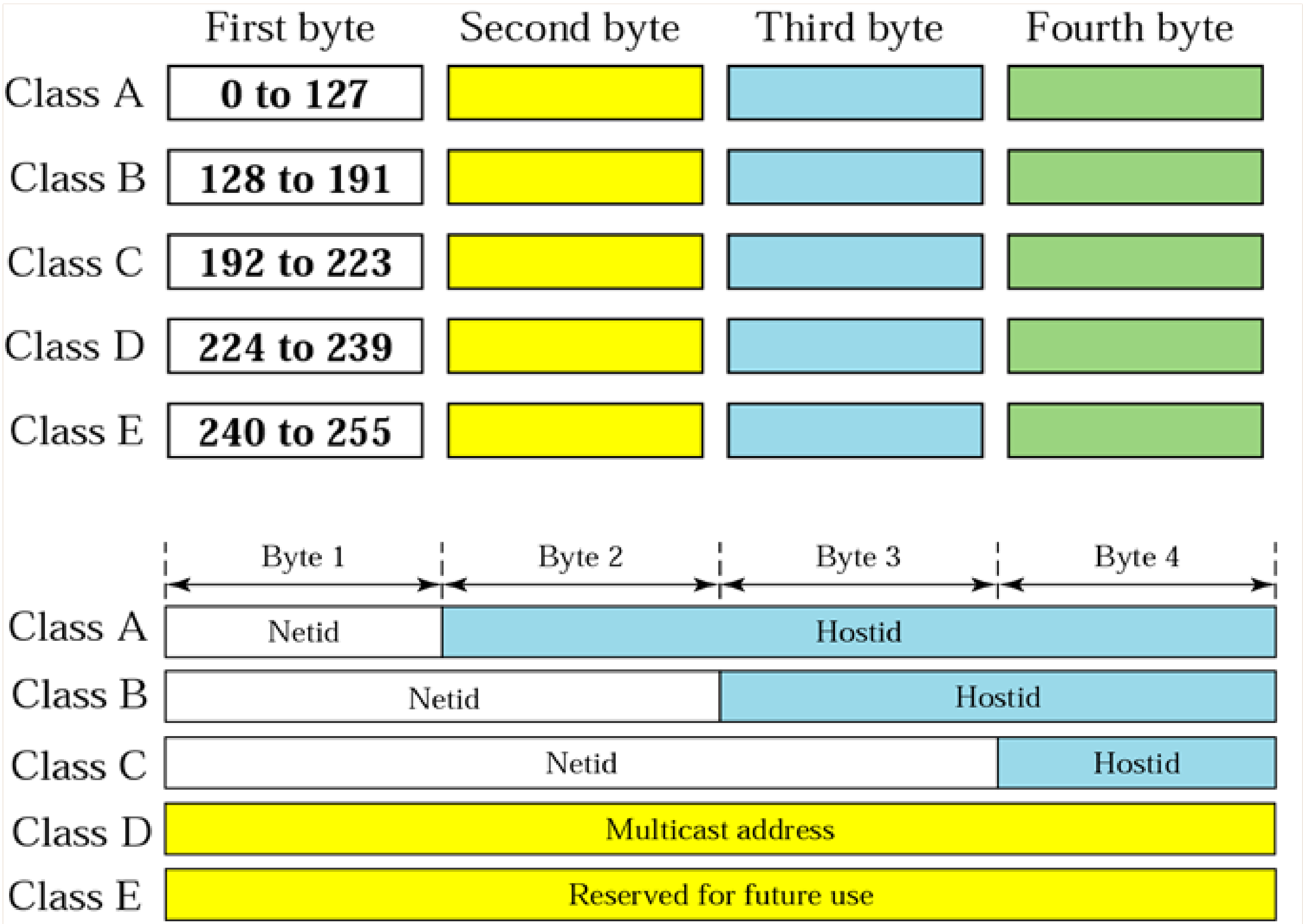
IP Addressing

An IP address is a 32-bit address (dotted decimal notation).



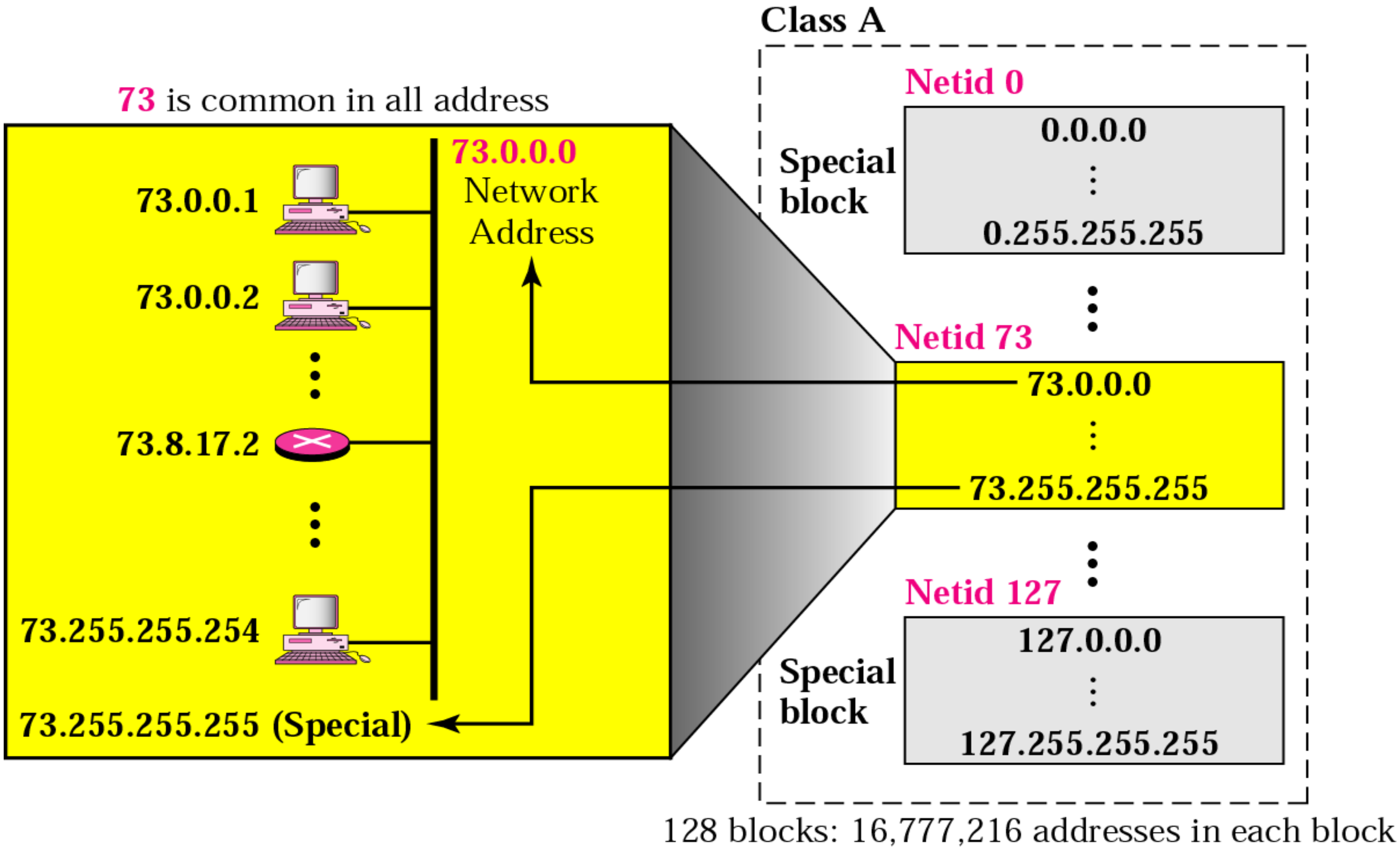
	First byte	Second byte	Third byte	Fourth byte
Class A	**0**			
Class B	**10**			
Class C	**110**			
Class D	**1110**			
Class E	**1111**			

Hierarchical addressing in IP



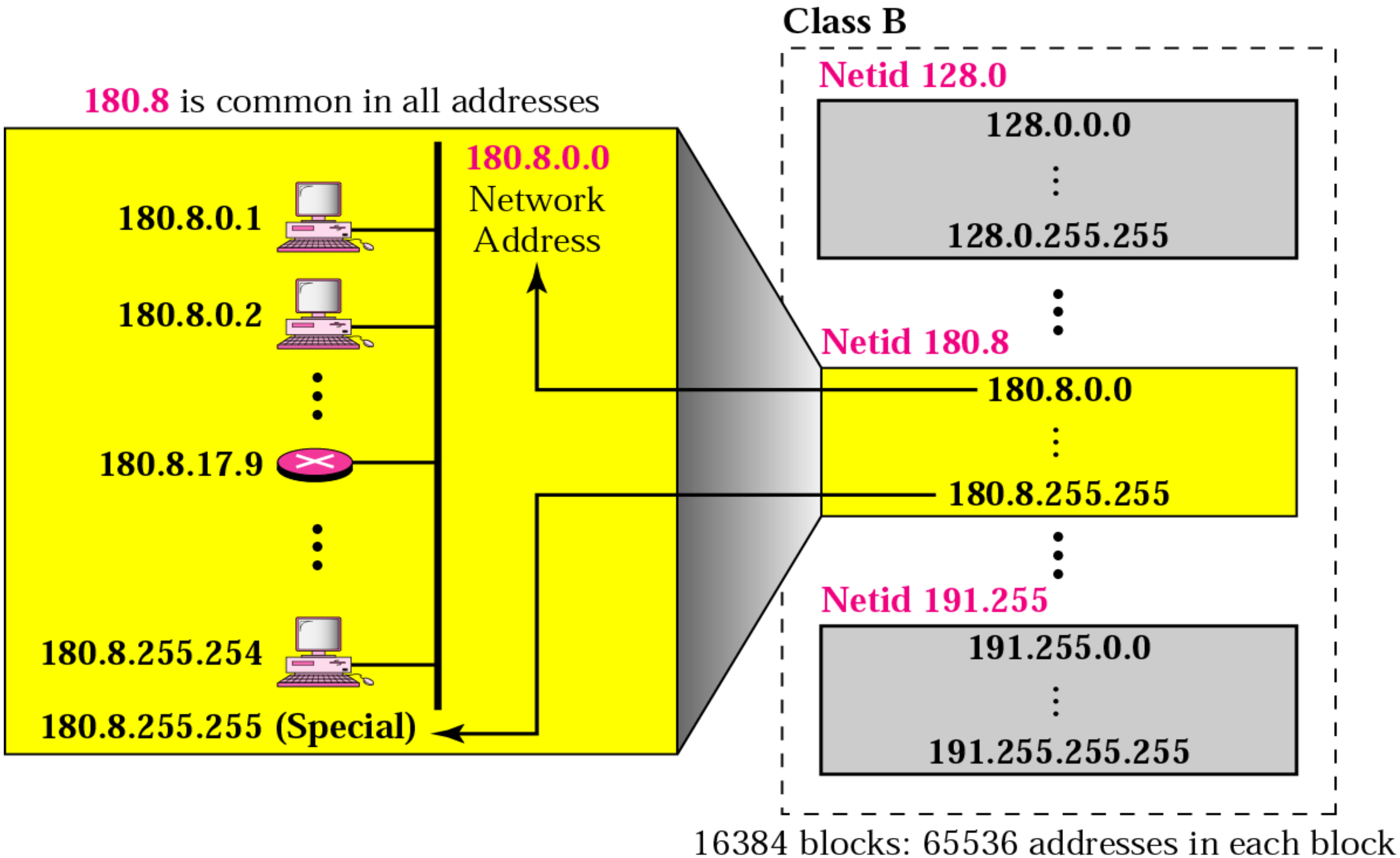
Two-level hierarchy

Blocks in class A



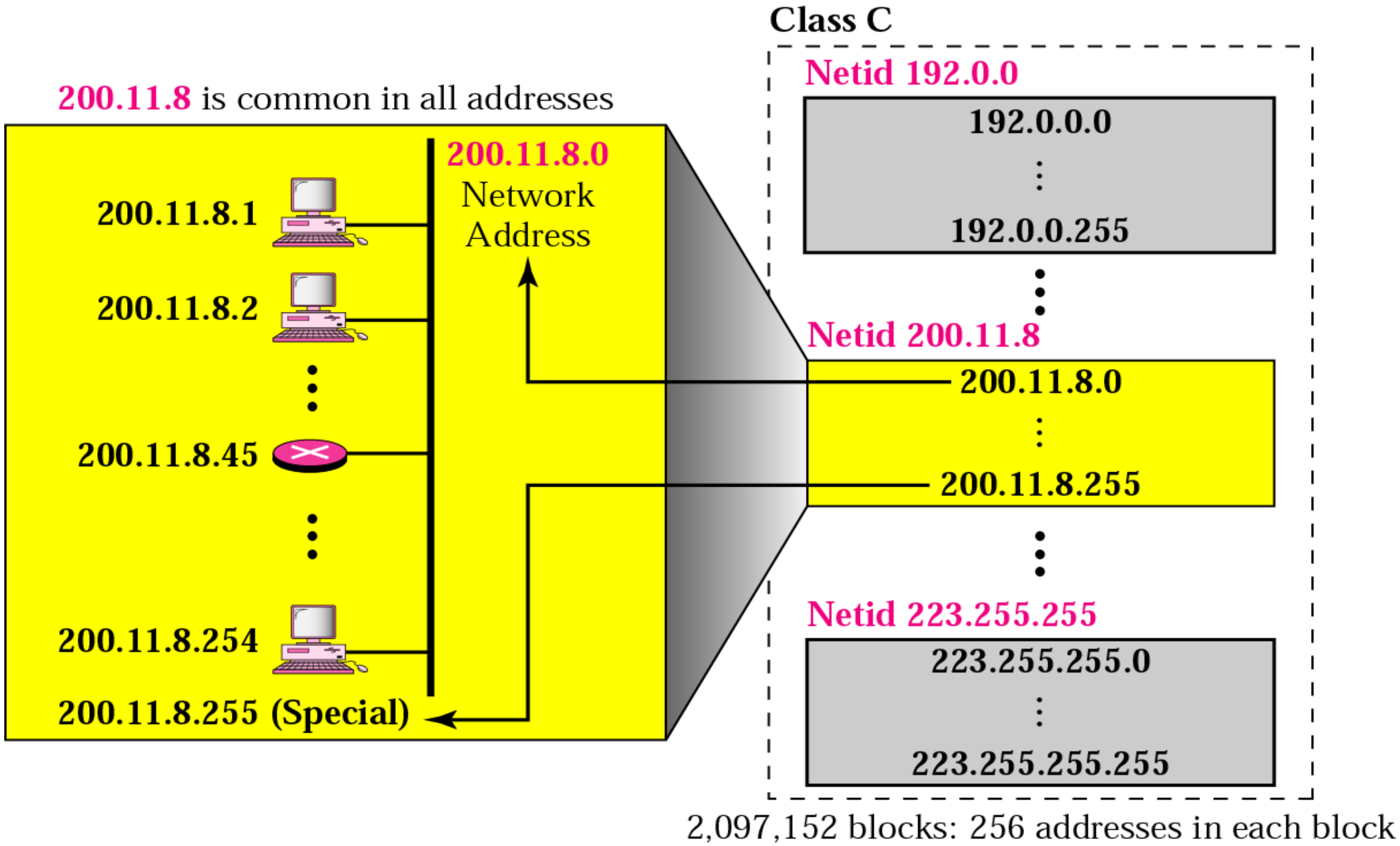
Millions of class A addresses are wasted

Blocks in class B



Many class B addresses are wasted.

Blocks in class C

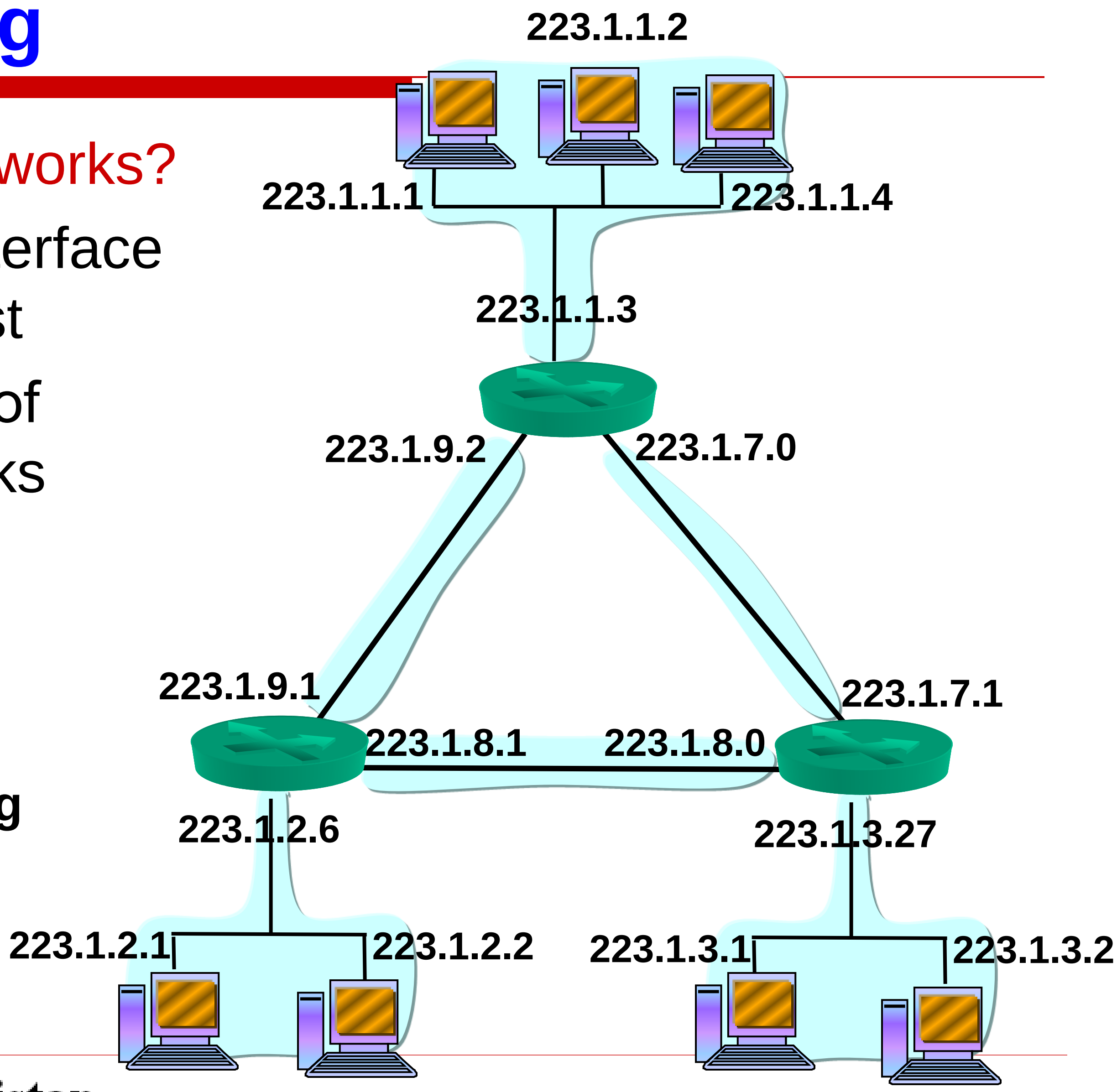


The number of addresses in class C is smaller than the needs of most organizations

IP Addressing

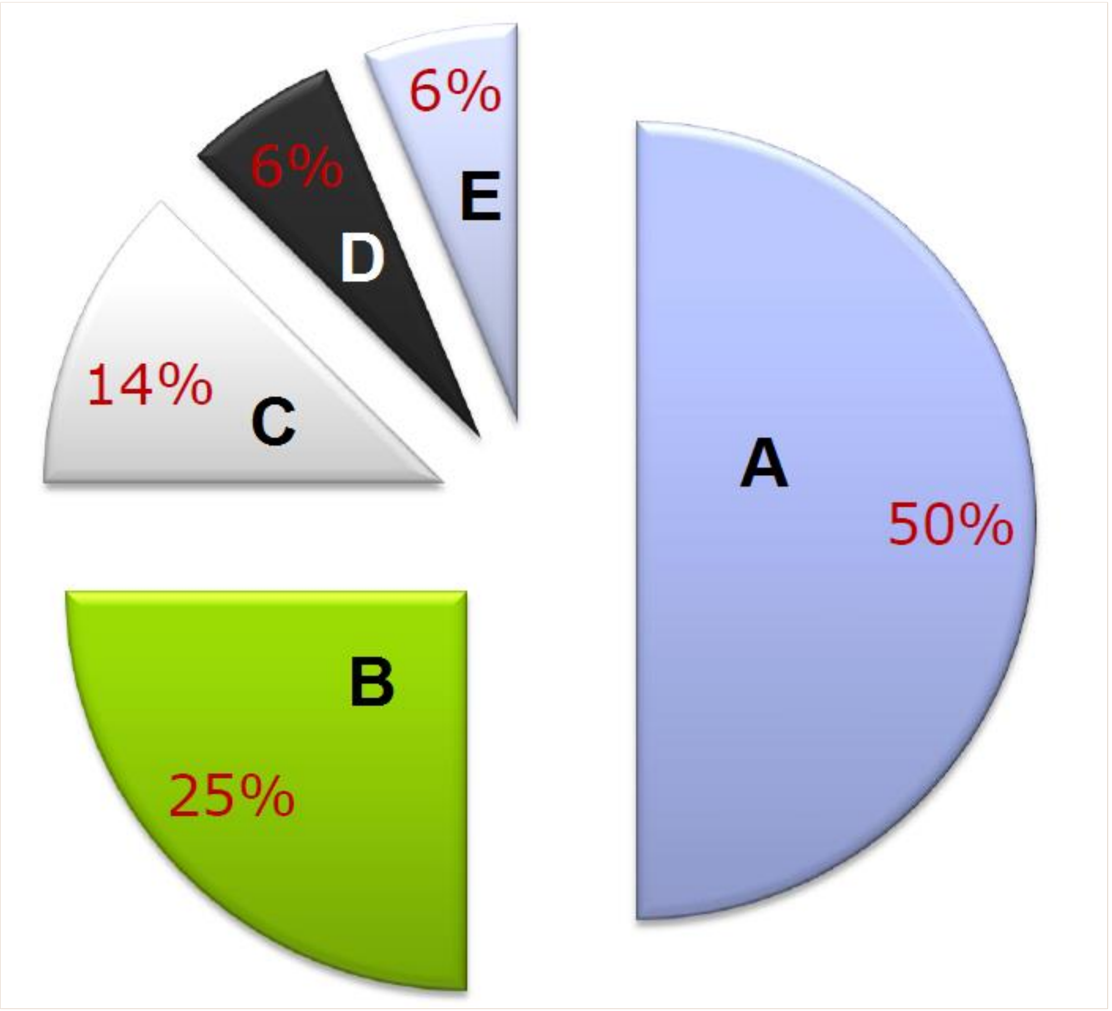
How to find the networks?

- Detach each interface from router, host
- create "islands of isolated networks



Interconnected system consisting of six networks.

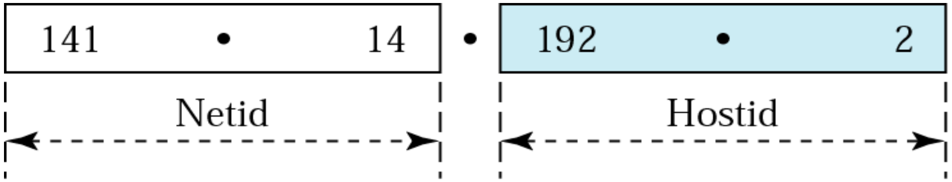
Address Space



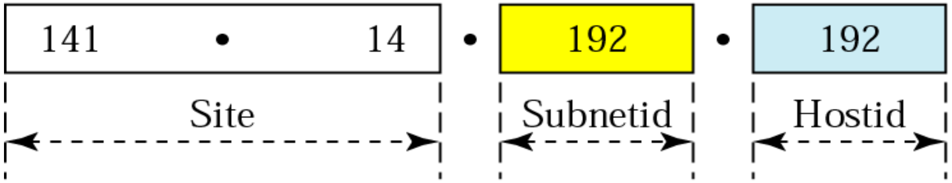
IP subnetts

Class C-networks (256 hosts) are very small and Class B-networks (65536 hosts) often too large. Therefore, divide a network into **subnets**

Dividing networks into smaller parts
more levels of hierarchy



a. Without subnetting

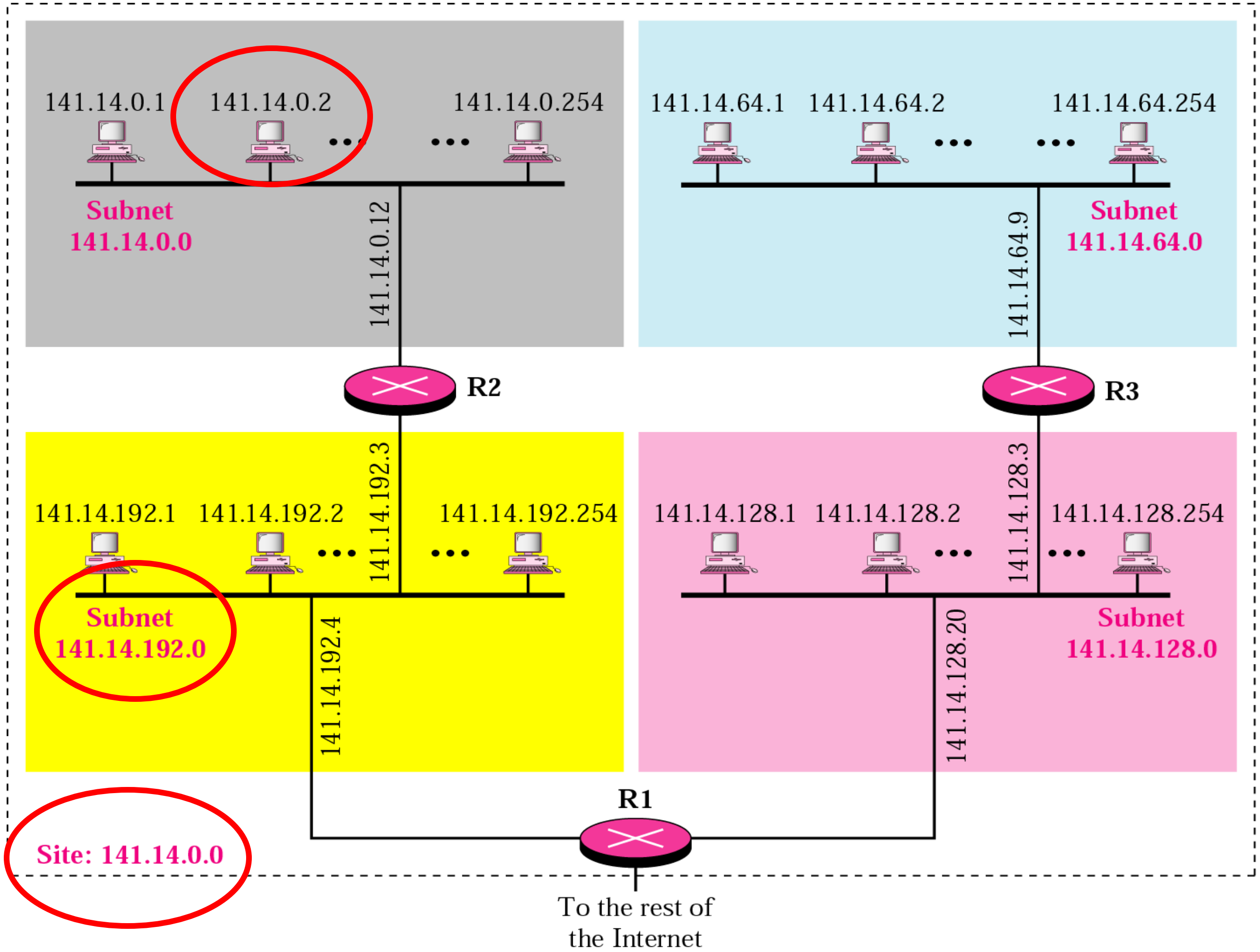


b. With subnetting

Subnetting (cnt'd)

3 hierarchy levels

- **Site**
- **Subnet**
- **Host**



Subnetting (Extended Network Prefix)

The ISP have been allocated the address block

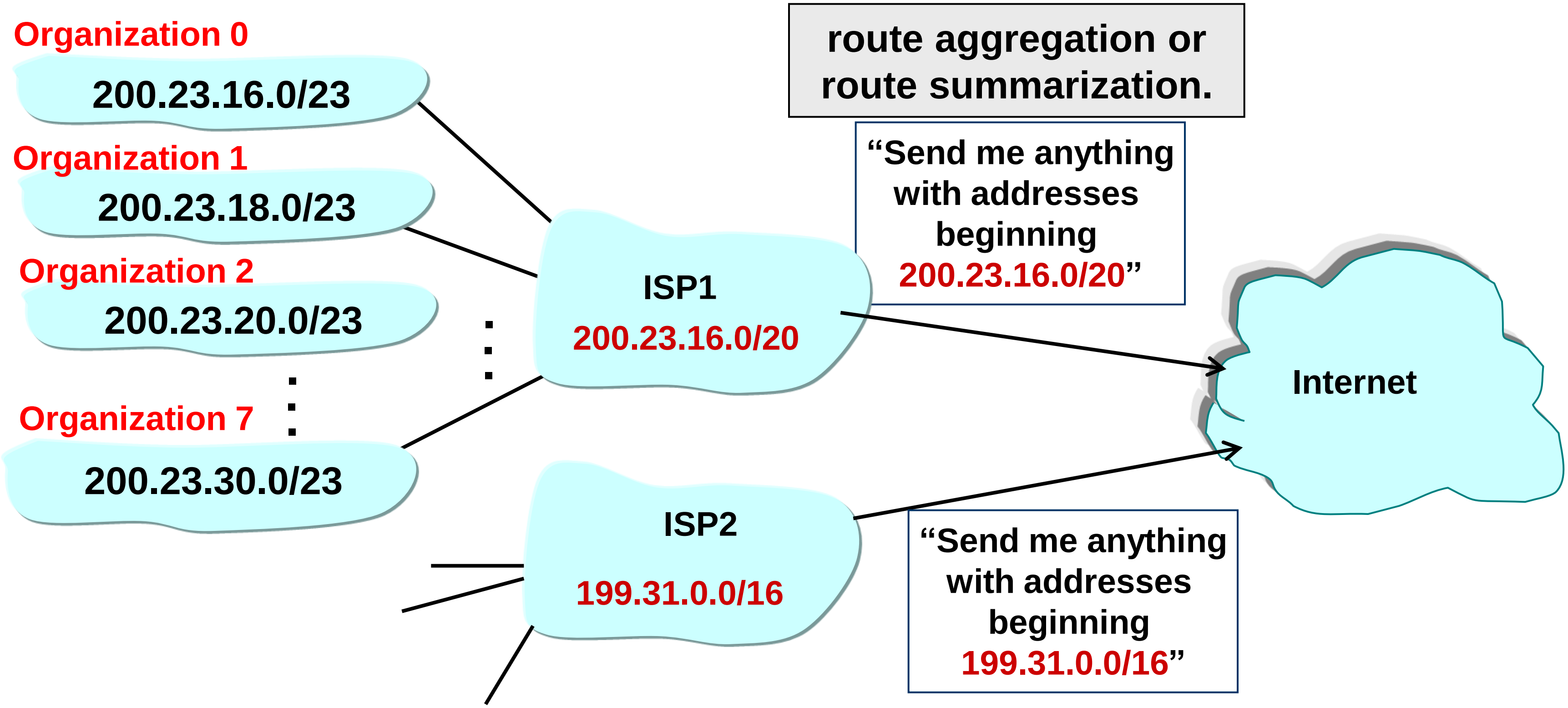
ISP's block	<u>11001000 00010111 0001</u>0000	00000000			200.23.16.0/20
Organization 0	<u>11001000 00010111 0001000</u>0	00000000			200.23.16.0/23
Organization 1	<u>11001000 00010111 0001001</u>0	00000000			200.23.18.0/23
Organization 2	<u>11001000 00010111 0001010</u>0	00000000			200.23.20.0/23
...	.....	....			....
Organization 7	<u>11001000 00010111 0001111</u>0	00000000			200.23.30.0/23

The ISP divides the block into 8 smaller addr. blocks (subnets) and gives them to 8 organization.

University of Kurdistan

Hierarchical addressing- route aggregation

Hierarchical addressing allows efficient advertisement of routing information



Addressing - mask

- *Routin*g is based on both *network* and *subnetwork* addresses
  - Analogy: Parcel delivery –> zip code and street address
- How can a router find the network or the subnetwork address to route the packet?
- Default mask: 32-bit binary number ANDed with the address in the block
  - if the bit in the mask = 1, then retain the bit in the address
  - if the bit in the mask ≠ 1, then put 0

Class	*In Binary*	*In Dotted-Decimal*	*Using Slash*
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 111111111 11111111 00000000	255.255.255.0	/24

number of 1's

Subnet Mask

ISP's block	11001000	00010111	00010000	00000000	200.23.16.0/20
ISP’s subnet mask	11111111	11111111	11110000	00000000	255.255.240.0

Organization 0	11001000	00010111	00010000	00000000	200.23.16.0/23
Organization 1	11001000	00010111	00010010	00000000	200.23.18.0/23
Organization 2	11001000	00010111	00010100	00000000	200.23.20.0/23
...	…..			….	….
Organization 7	11001000	00010111	00011110	00000000	200.23.30.0/23
Or’s subnet mask	11111111	11111111	11111110	00000000	255.255.254.0

Network part of an IP address= subnet mask & IP address

Classless addressing

- **Solving problems with classful addressing:**
  - 256 < the number of IP address < 16 777 216
  - what if one needs at home only 2 addresses? 254 wasted?
- **Solution: Classless addressing**
  - addresses provided by Internet Service Provider
  - ISP divides blocks of addresses into groups of 2, 4, 8 or 16
  - the household devices are connected to ISP via dial-up, DSL, …
- **Variable-length blocks that belong to no class**
  - the number of address block must be power of 2
- **Classless InterDomain Routing (CIDR)**

Longest prefix match forwarding

- Forwarding tables in IP routers
  - Maps each IP prefix to next-hop link(s)
- Destination-based forwarding
  - Packet has a destination address
  - Router identifies longest-matching prefix
  - Cute algorithmic problem: very fast lookups

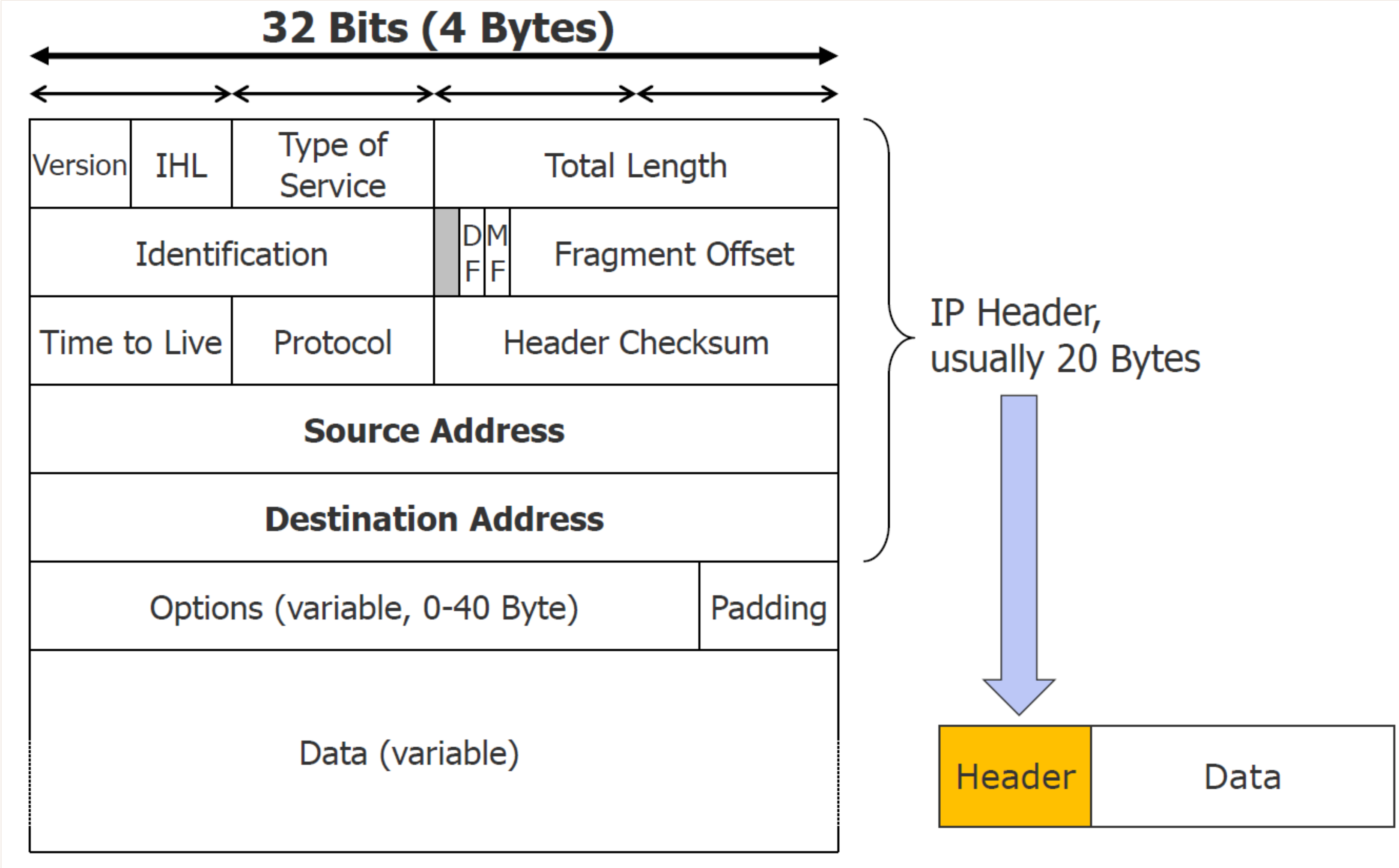
forwarding table

destination
145.13.52.63

Address/Mask	Outgoing Interface
145.13.56.0/22	E0
145.13.60.0/22	E1
192.13.52.0/23	S0
145.13.54.0/22	S1

outgoing link
Port S1

IP Header

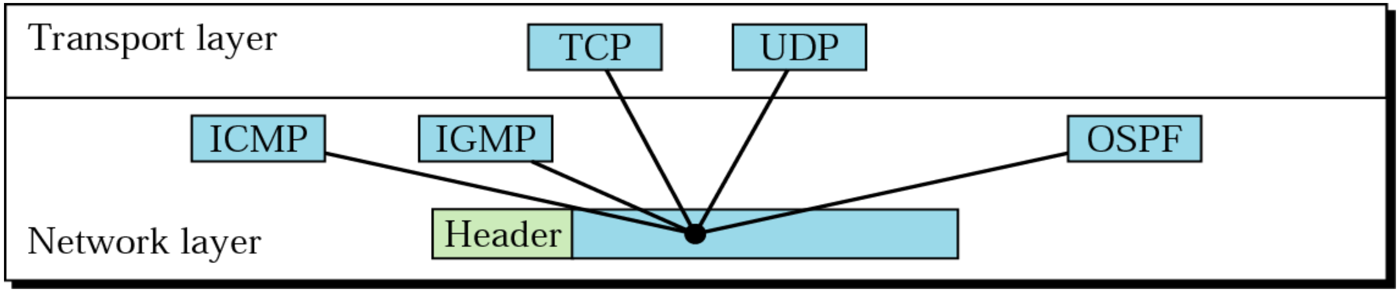

32 Bits (4 Bytes)
Version
IHL
Type of Service
Total Length
Identification
DF
MF
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options (variable, 0-40 Byte)
Padding
Data (variable)
IP Header, usually 20 Bytes
Header
Data

IP Header fields

- **Version number (4 bits)**
  - Indicates the version of the IP protocol
  - Necessary to know what other fields to expect
  - Typically “4” (for IPv4), and sometimes “6” (for IPv6)
- **IP Header length (4 bits)**
  - Number of 32-bit words in the header
  - Typically “5” (for a 20-byte IPv4 header)
  - Can be more when IP options are used
- **Total length (16 bits)**
  - Number of bytes in the packet
  - Maximum size is 65,535 bytes (2^{16} -1)
  - … though underlying links may impose smaller limits

IP Header fields - protocol

- **Protocol (8 bits)**
  - Identifies the higher-level protocol
  - Important for demultiplexing at receiving host



value	protocol
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

IP Header fields

- **Two IP addresses**
  - Source IP address (32 bits)
  - Destination IP address (32 bits)

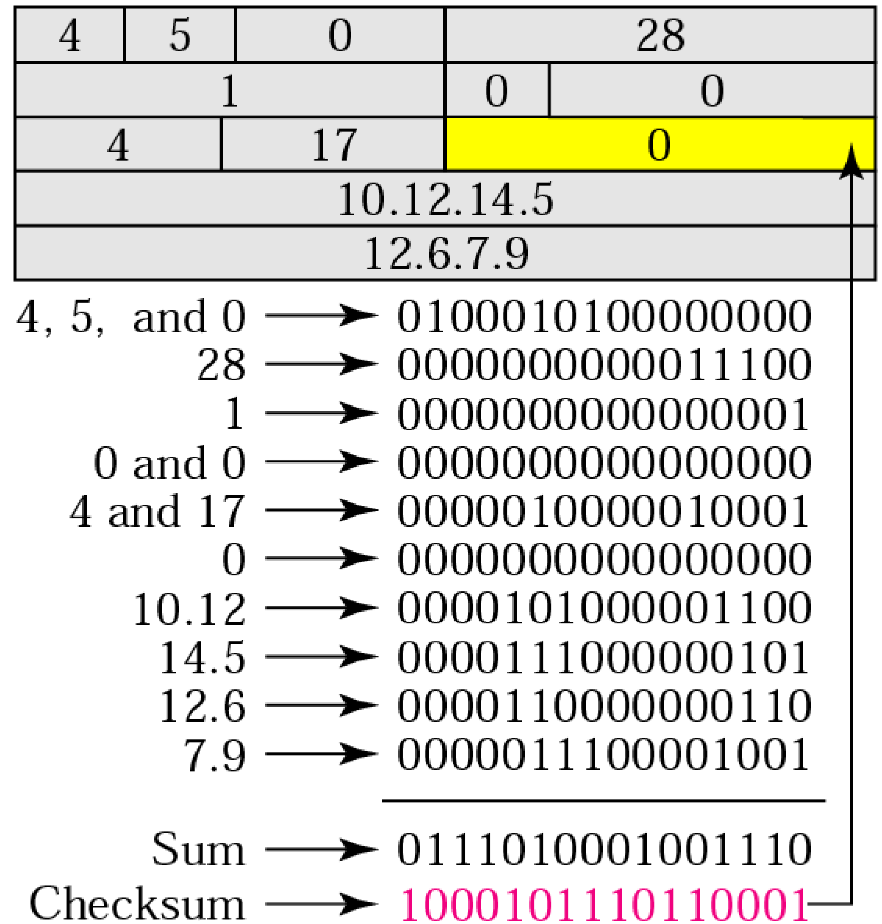
- **Type-of-Service (8 bits)**
  - Allow packets to be treated differently based on needs
  - E.g., low delay for audio, high bandwidth for bulk transfer
  - Has been redefined several times, will cover late in QoS

- **Options**

IP Header fields - checksum

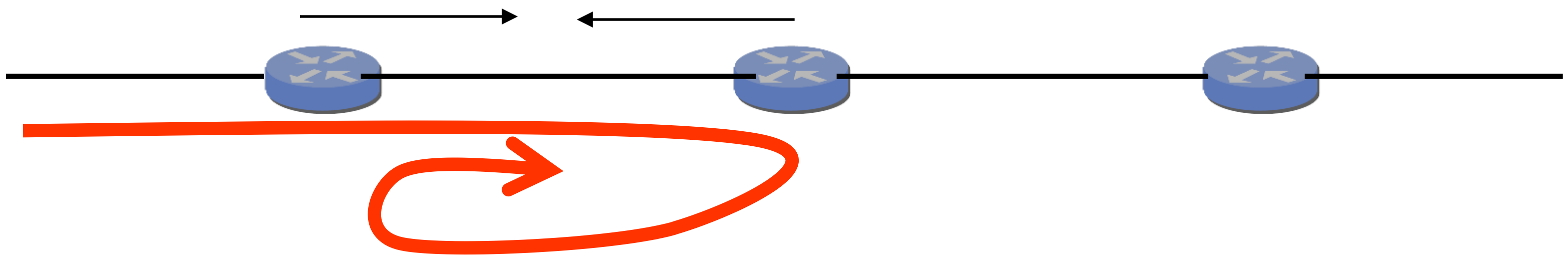
- **Header Checksum**
  for error detection

- **If not correct, router discards packets**



IP Header fields - TTL

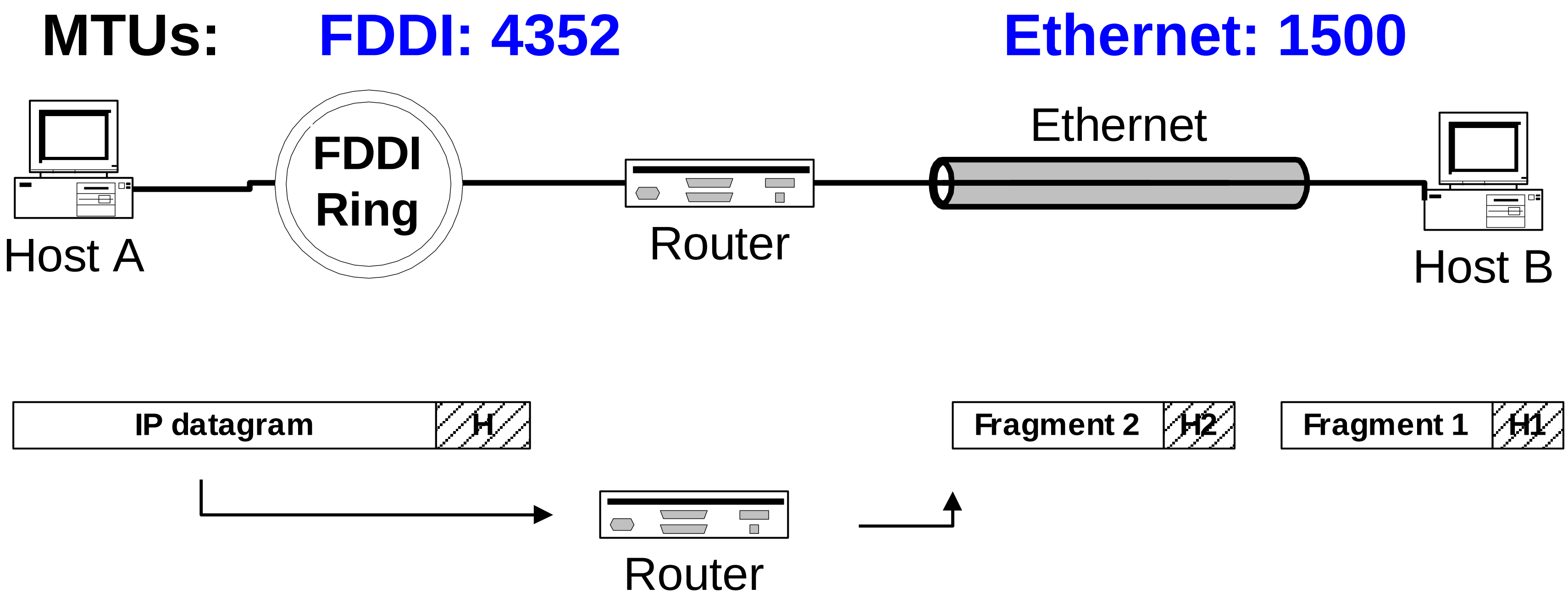
- **Forwarding loops cause packets to cycle forever**
  - As these accumulate, eventually consume **all** capacity



- **Time-to-Live (TTL) Field (8 bits)**
  - Decremented at each hop, packet discarded if reaches 0
  - …and "time exceeded" message is sent to the source

IP Header fields

Fragmentation: when forwarding a packet, an Internet router can split it into multiple pieces ("fragments") if too big for next hop link



IP Header fields – fragmentation fields

- **Identifier (16 bits):** used to tell which fragments belong together
- **Flags (3 bits):**
  - **Don't Fragment (DF):** instruct routers to not fragment the packet even if it won't fit
    - Instead, they drop the packet and send back a "Too Large" ICMP control message
    - Forms the basis for "Path MTU Discovery", covered later
  - **More (MF):** this fragment is not the last one
- **Offset (13 bits):** what part of datagram this fragment covers in 8-byte units

Example of Fragmentation

- Suppose we have a 4000 byte datagram sent from host 1.2.3.4 to host 3.4.5.6 …

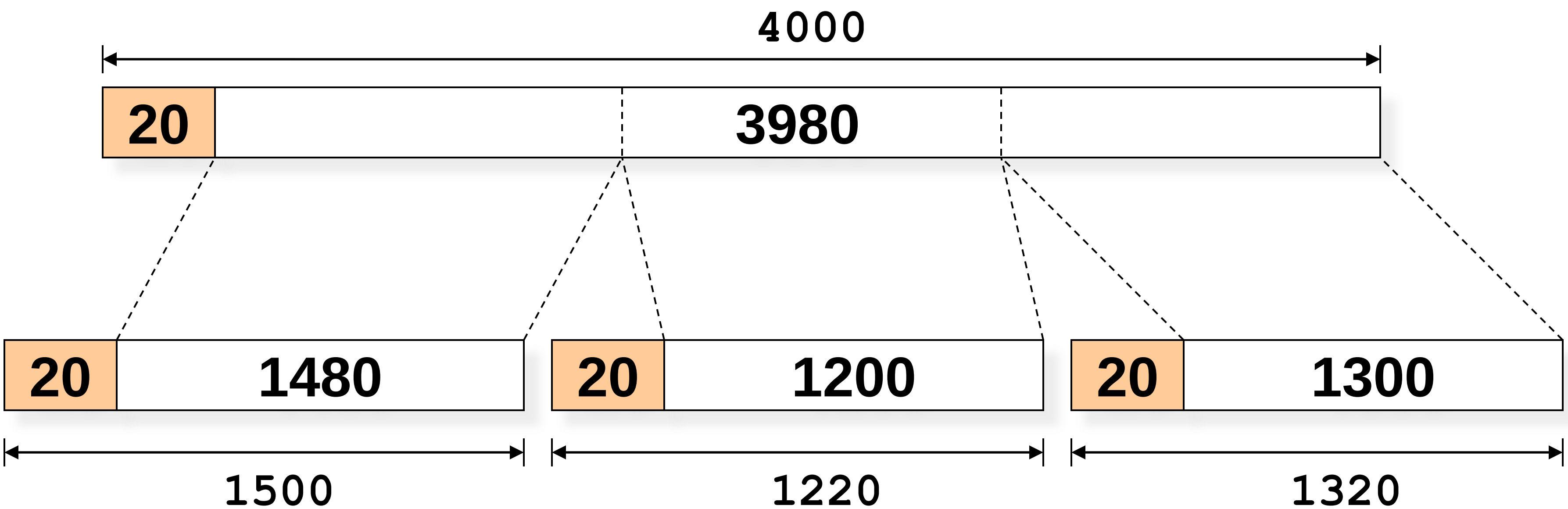
<table>
<tr><td>Version
4</td><td>Header Length
5</td><td>Type of Service
0</td><td colspan="2">Total Length: 4000</td></tr>
<tr><td colspan="3">Identification: 56273</td><td>D/M
0/0</td><td>Fragment Offset: 0</td></tr>
<tr><td colspan="2">TTL
127</td><td>Protocol
6</td><td colspan="2">Checksum: 44019</td></tr>
<tr><td colspan="5">Source Address: 1.2.3.4</td></tr>
<tr><td colspan="5">Destination Address: 3.4.5.6</td></tr>
</table>

(3980 more bytes here)

- … and it traverses a link that limits datagrams to 1,500 bytes

Example of Fragmentation (con’t)

Datagram split into 3 pieces, for example:



Example of Fragmentation (con’t)

Datagram split into 3 pieces. Possible first piece:

Version 4	Header Length 5	Type of Service 0	Total Length: 1500
Identification: 56273		D/M 0/1	Fragment Offset: 0
TTL 127	Protocol 6	Checksum: xxx	
Source Address: 1.2.3.4			
Destination Address: 3.4.5.6			

Example of Fragmentation (con’t)

Possible second piece:

<table>
<tr><td>Version
4</td><td>Header Length
5</td><td>Type of Service
0</td><td colspan="3">Total Length: 1220</td></tr>
<tr><td colspan="3">Identification: 56273</td><td colspan="2">D/M
0/1</td><td>Fragment Offset: 185
(185 * 8 = 1480)</td></tr>
<tr><td colspan="2">TTL
127</td><td>Protocol
6</td><td colspan="3">Checksum: yyy</td></tr>
<tr><td colspan="6">Source Address: 1.2.3.4</td></tr>
<tr><td colspan="6">Destination Address: 3.4.5.6</td></tr>
</table>

Example of Fragmentation (con’t)

Possible third piece:

Version 4	Header Length 5	Type of Service 0	Total Length: 1320	
Identification: 56273			D/M 0/0	Fragment Offset: 335 (335 * 8 = 2680)
TTL 127	Protocol 6	Checksum: zzz		
Source Address: 1.2.3.4				
Destination Address: 3.4.5.6				

University of Kurdistan

Where is Fragmentation done?

- **Fragmentation** can be done at the **sender** or at **intermediate routers**
- The same datagram can be fragmented **several times**.
- **Reassembly** of original datagram is only done at **destination hosts** !!

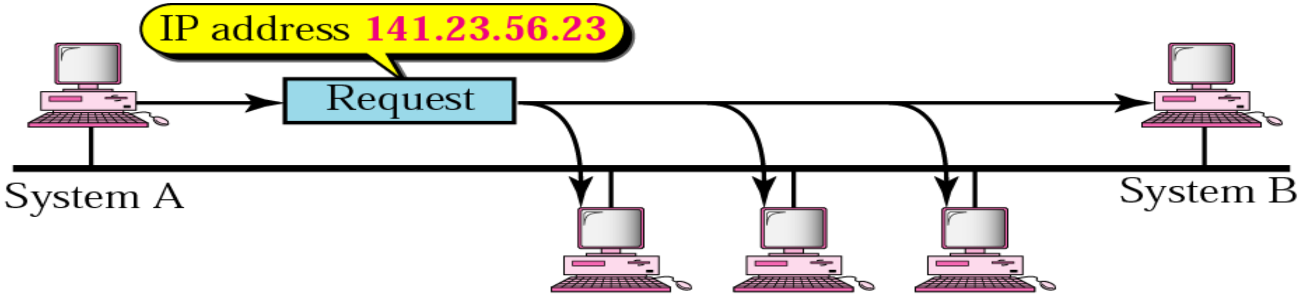
Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP)

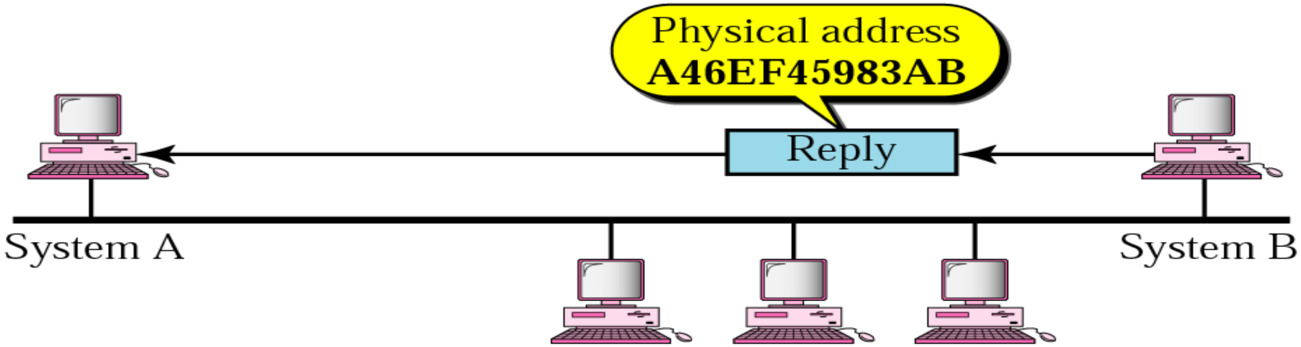
- Two levels of addresses: IP and MAC
- Need to be able to map an IP address to its corresponding MAC address
- Two types of mapping : static and dynamic
- Static mapping has some limitations and overhead against network performance
- Dynamic mapping: ARP and RARP
- ARP: mapping IP address to a MAC address
- RARP (replaced by DHCP): mapping a MAC address to an IP address

ARP operation

- ARP associates an IP address with its MAC addresses
- An ARP request is broadcast; an ARP reply is unicast.



a. ARP request is broadcast



b. ARP reply is unicast

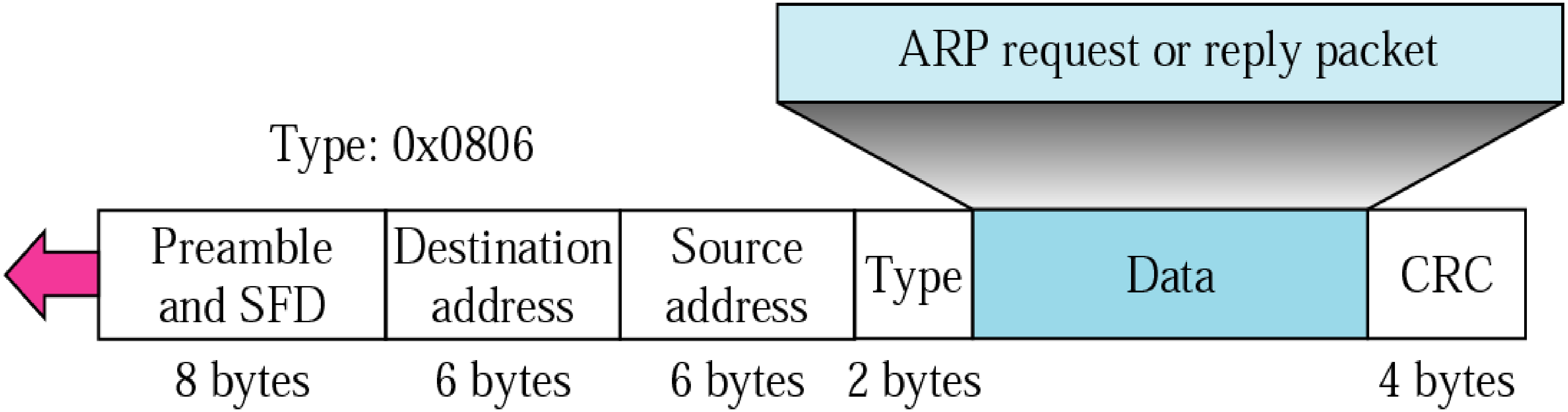
ARP packet format

- Protocol Type: 0800 for IPv4, Hardware length: 6 for Ethernet, Protocol length: 4 for IPv4

<table>
<tr><td colspan="2">Hardware Type</td><td>Protocol Type</td></tr>
<tr><td>Hardware
length</td><td>Protocol
length</td><td>Operation
Request 1, Reply 2</td></tr>
<tr><td colspan="3">Sender hardware address
(For example, 6 bytes for Ethernet)</td></tr>
<tr><td colspan="3">Sender protocol address
(For example, 4 bytes for IP)</td></tr>
<tr><td colspan="3">Target hardware address
(For example, 6 bytes for Ethernet)
(It is not filled in a request)</td></tr>
<tr><td colspan="3">Target protocol address
(For example, 4 bytes for IP)</td></tr>
</table>

Encapsulation of ARP packet

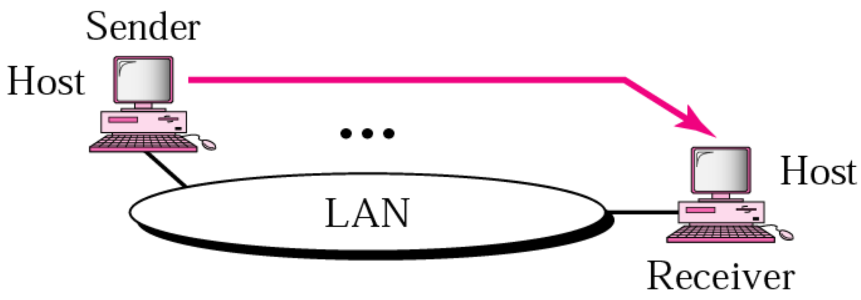
- ARP packet is encapsulated ***directly*** into a data link frame (example: Ethernet frame)



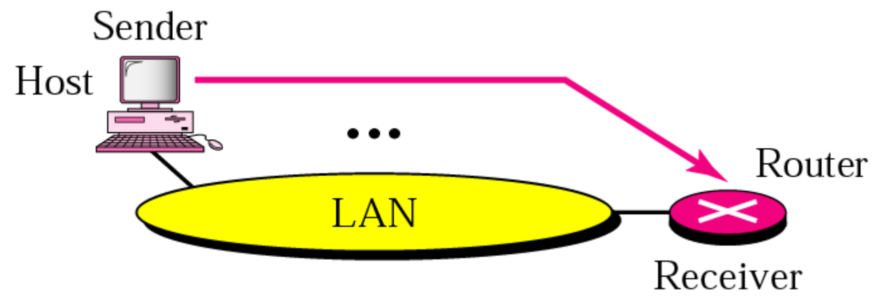
ARP Operation

- The sender knows the IP address of the target
- IP asks ARP to create an ARP request message
- The message is encapsulated in a frame (destination address = broadcast address)
- Every host or router receives the frame. The target recognizes the IP address
- The target replies with an ARP reply message (unicast with its physical address)
- The sender receives the reply message knowing the physical address of the target
- The IP datagram is now encapsulated in a frame and is unicast to the destination

Four different cases using ARP

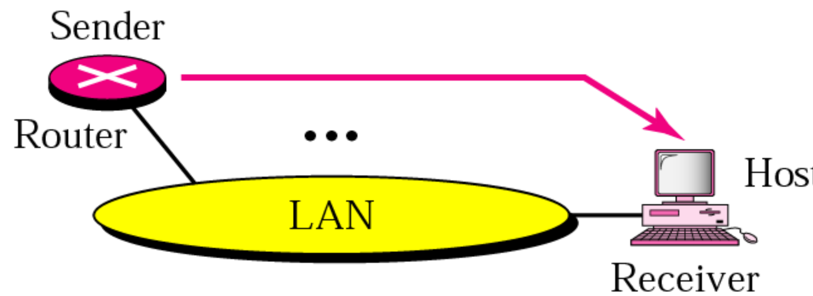


Case 1. A host has a packet to send to another host on the same network.



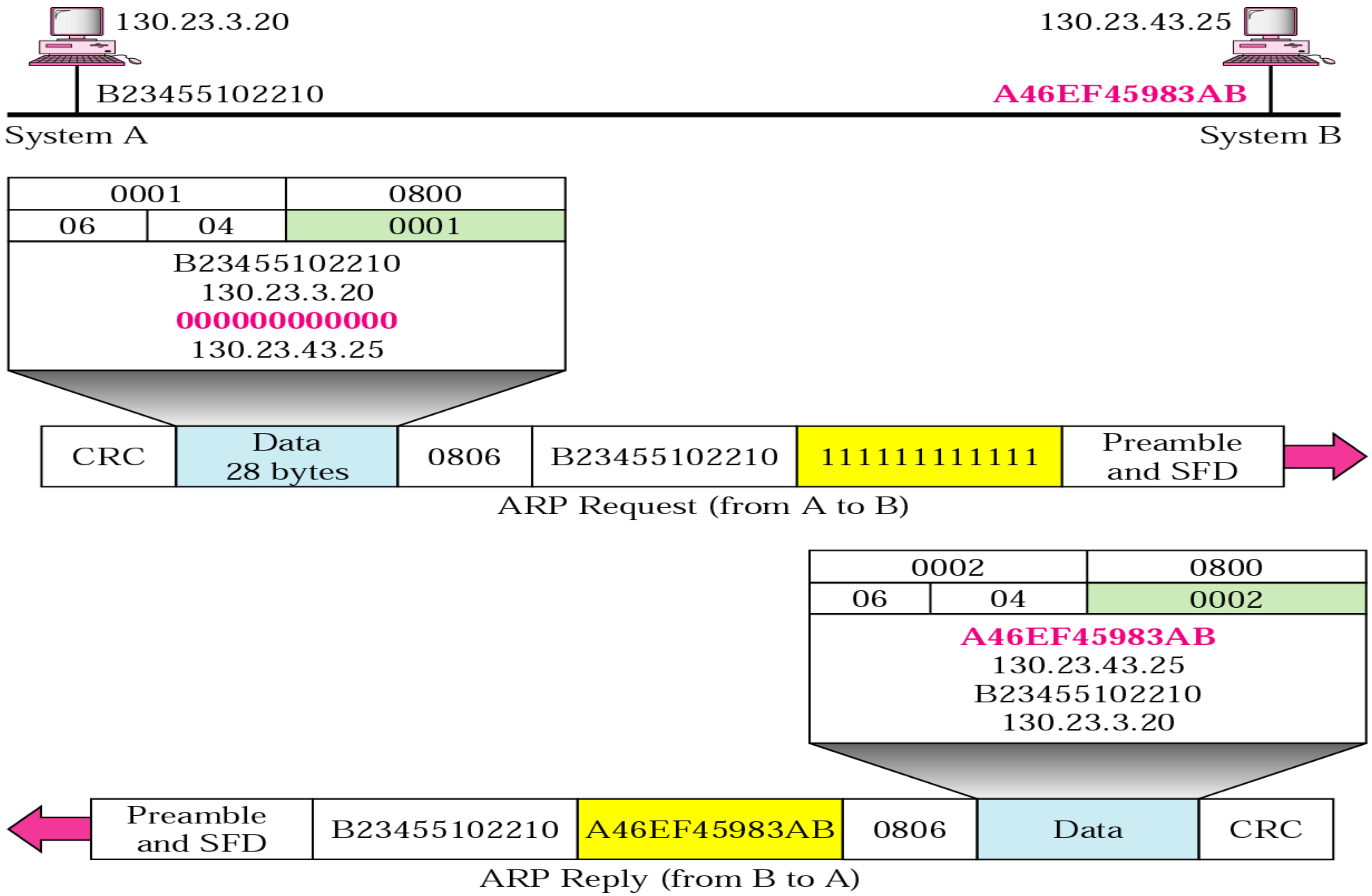
Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to the appropriate router.

Case 3. A router receives a packet to be sent to a host on another network.
It must first be delivered to the appropriate router.



Case 4. A router receives a packet to be sent to a host on the same network.

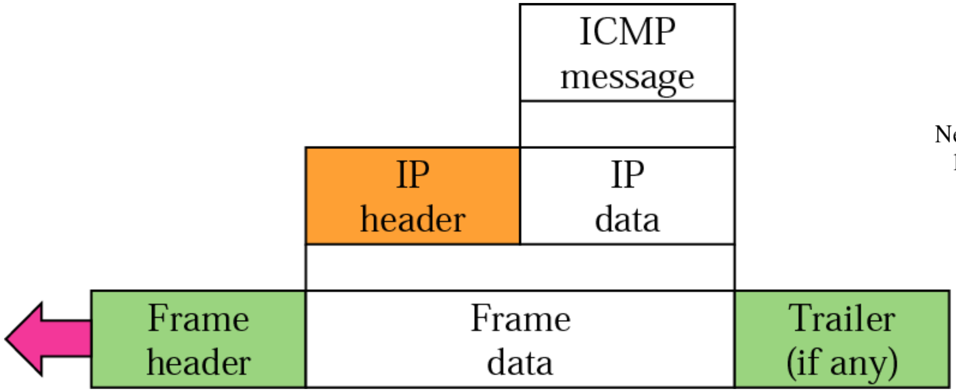
ARP: Example

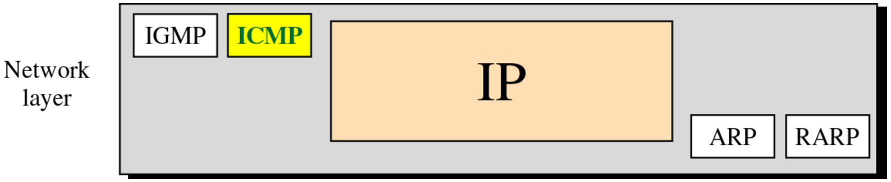

130.23.3.20
B23455102210
System A
130.23.43.25
A46EF45983AB
System B
0001
0800
06
04
0001
B23455102210
130.23.3.20
000000000000
130.23.43.25
CRC
Data
28 bytes
0806
B23455102210
111111111111
Preamble
and SFD
ARP Request (from A to B)
0002
0800
06
04
0002
A46EF45983AB
130.23.43.25
B23455102210
130.23.3.20
Preamble
and SFD
B23455102210
A46EF45983AB
0806
Data
CRC
ARP Reply (from B to A)

Internet Control Message Protocol (ICMP)

ICMP

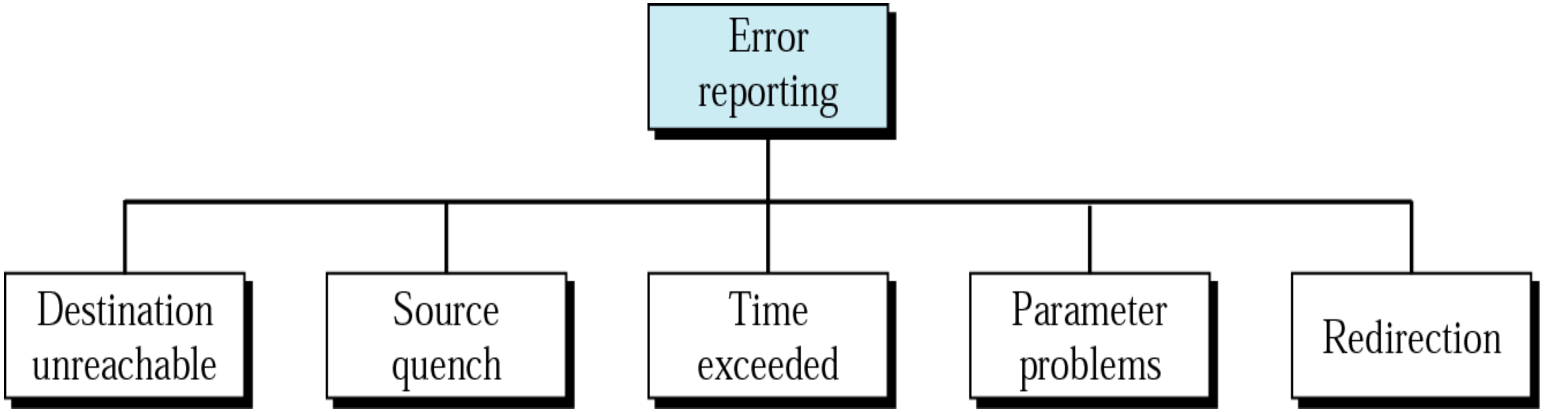
- IP has no error-reporting or error-correcting mechanism
- IP also lacks a mechanism for host and management queries
- Internet Control Message Protocol (ICMP) is designed to compensate for two deficiencies, which is a companion to the IP
- Two types messages: error-reporting messages and query messages



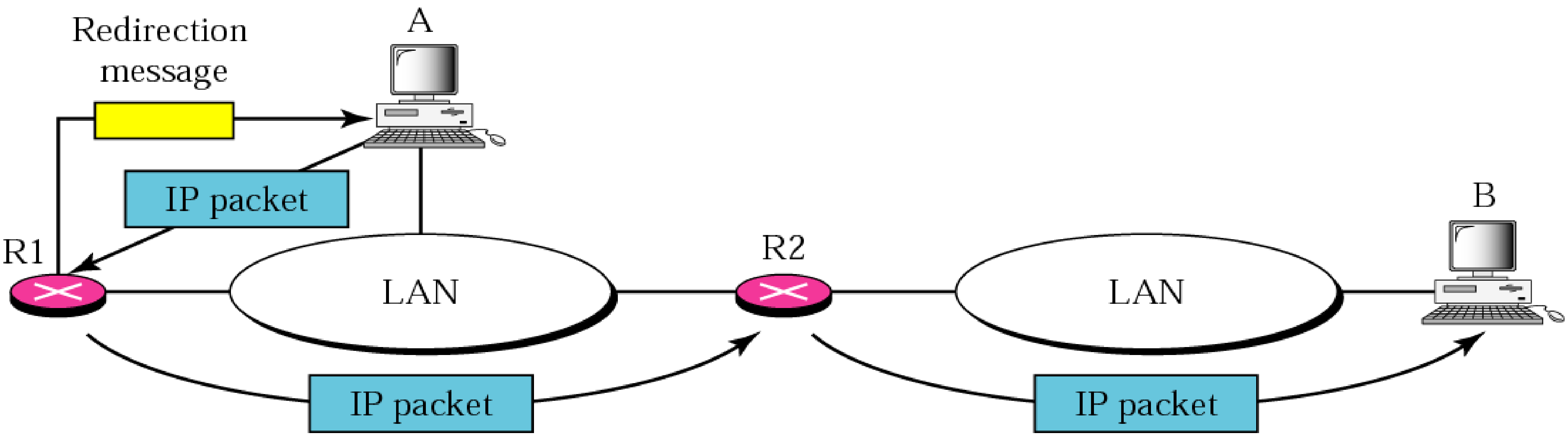


Error-reporting messages

- ICMP always reports error messages to the original source.
- Source quench: There is no flow control or congestion control mechanism in IP. Source Quench requests that the sender decrease the rate of messages
- Time exceed: (1) TTL related, (2) do not receive all fragments with a certain time limit
- Redirection: To update the routing table of a host

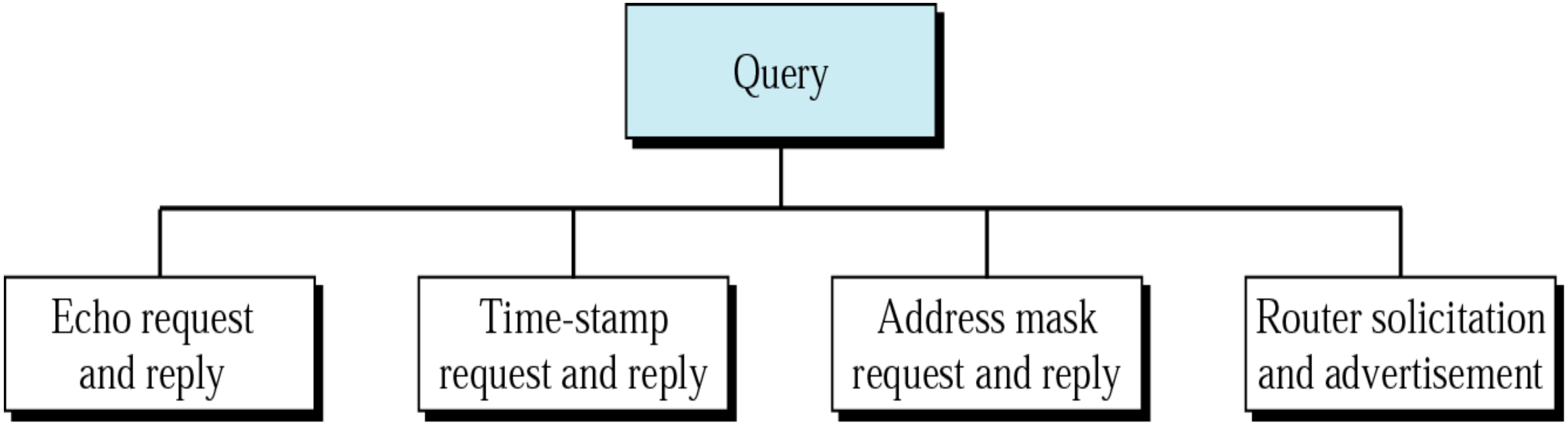


Redirection concept

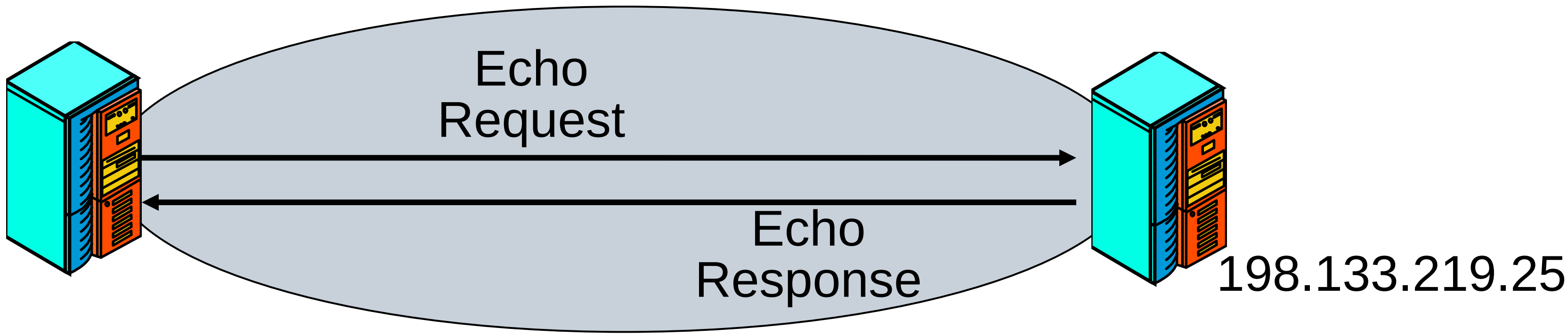


Query messages

- To diagnose some network problems
- A node sends a message that is answered in a specific format by the destination node
- Echo for diagnosis; Time-stamp to determine RTT or synchronize the clocks in two machines; Address mask to know network address, subnet address, and host id; Router solicitation to know the address of routers connected and to know if they are alive and functioning



ICMP Query usage (Ping)




```
C:\WINNT\System32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
<C> Copyright 1985-2000 Microsoft Corp.

C:\> ping 198.133.219.25

Pinging 198.133.219.25 with 32 bytes of data:

Reply from 198.133.219.25: bytes= 32 time= 16ms TTL=247
Reply from 198.133.219.25: bytes= 32 time= 16ms TTL=247
Reply from 198.133.219.25: bytes= 32 time= 16ms TTL=247
Reply from 198.133.219.25: bytes= 32 time= 16ms TTL=247

Ping statistics for 198.133.219.25:
    Packets: Sent = 4, Recieived = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 16ms, Maximum = 16ms, Average = 16vms
C:\>
```

Traceroute and ICMP

- Source sends series of UDP segments to dest
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router& IP address

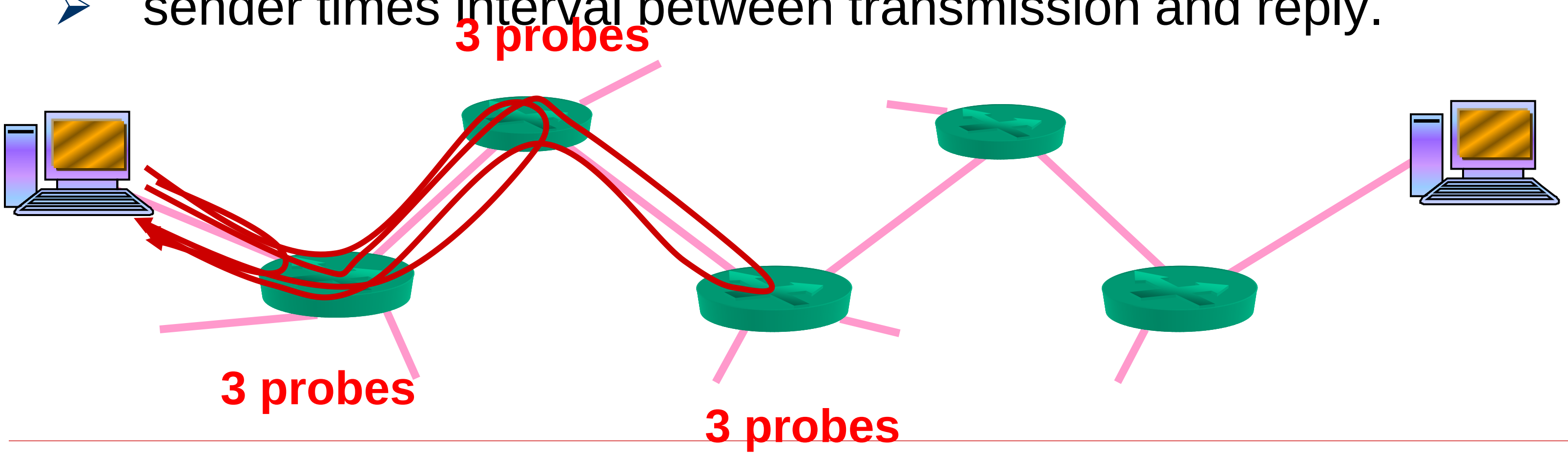
- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times

Stopping criterion

- UDP segment eventually arrives at destination host
- Destination returns ICMP “host unreachable” packet (type 3, code 3)
- When source gets this ICMP, stops.

"Real" Internet delays and routes

- What do "real" Internet delay & loss look like?
- **Traceroute** program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - sends three packets that will reach router *i* on path towards destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.

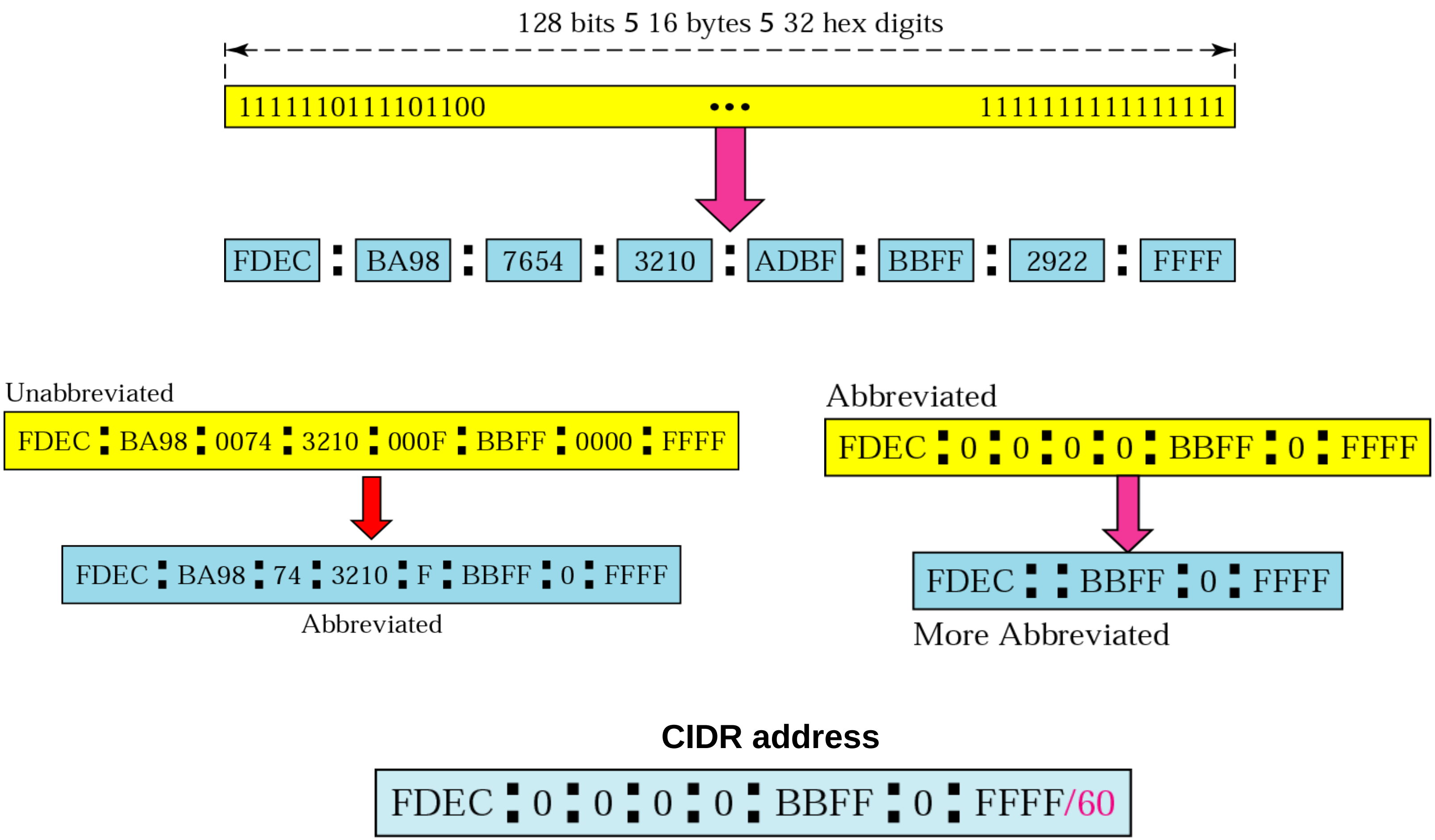


IP Version 6 (IPV6)

IPv6 address

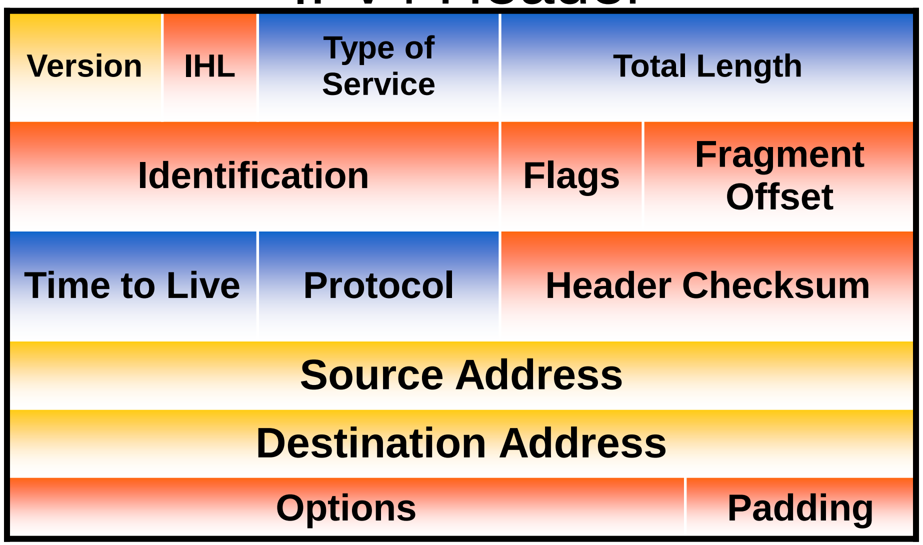
- The use of address space is inefficient
- Minimum delay strategies and reservation of resources are required to accommodate real-time audio and video transmission
- No security mechanism (encryption and authentication) is provided
- IPv6 (IPng: Internetworking Protocol, next generation)
  - Larger address space (128 bits)
  - Better header format
  - New options
  - Allowance for extension
  - Support for resource allocation: flow label to enable the source to request special handling of the packet
  - Support for more security

IPv6 address

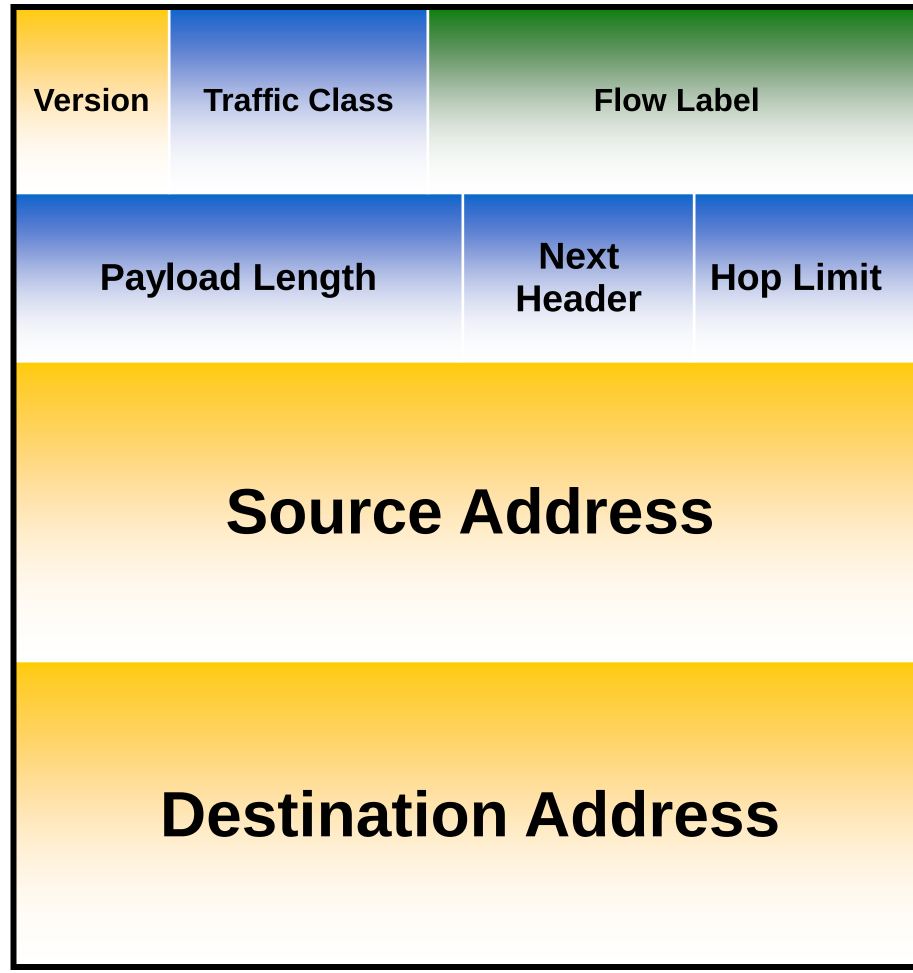


IPv4 & IPv6 Header Comparison

IPv4 Header



IPv6 Header



Legend

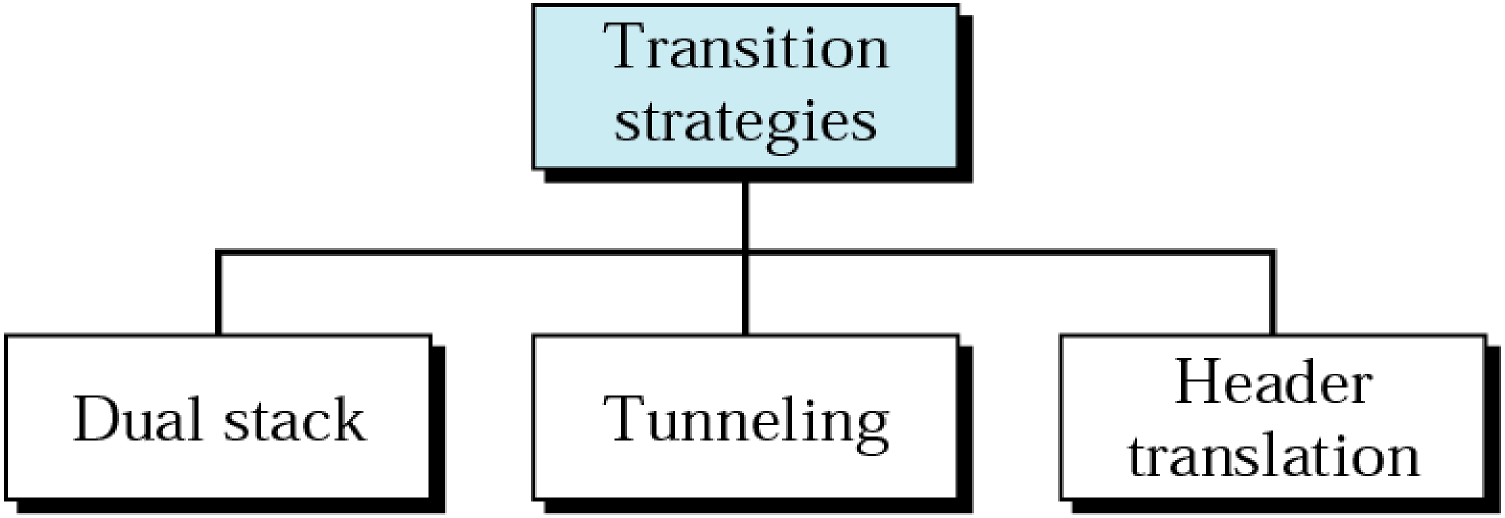
- field's name kept from IPv4 to IPv6
- fields not kept in IPv6
- Name & position changed in IPv6
- New field in IPv6

IPv6 Header

- Version: IPv4, IPv6
- Priority (4 bits): the priority of the packet with respect to traffic congestion
- Flow label (3 bytes): to provide special handling for a particular flow of data
- Payload length
- Next header (8 bits): to define the header that follows the base header in the datagram
- Hop limit: TTL in IPv4
- Source address (16 bytes) and destination address (16 bytes): if source routing is used, the destination address field contains the address of the next router

Three transition strategies from IPv4 to IPv6

- Transition should be smooth to prevent any problems between IPv4 and IPv6 systems

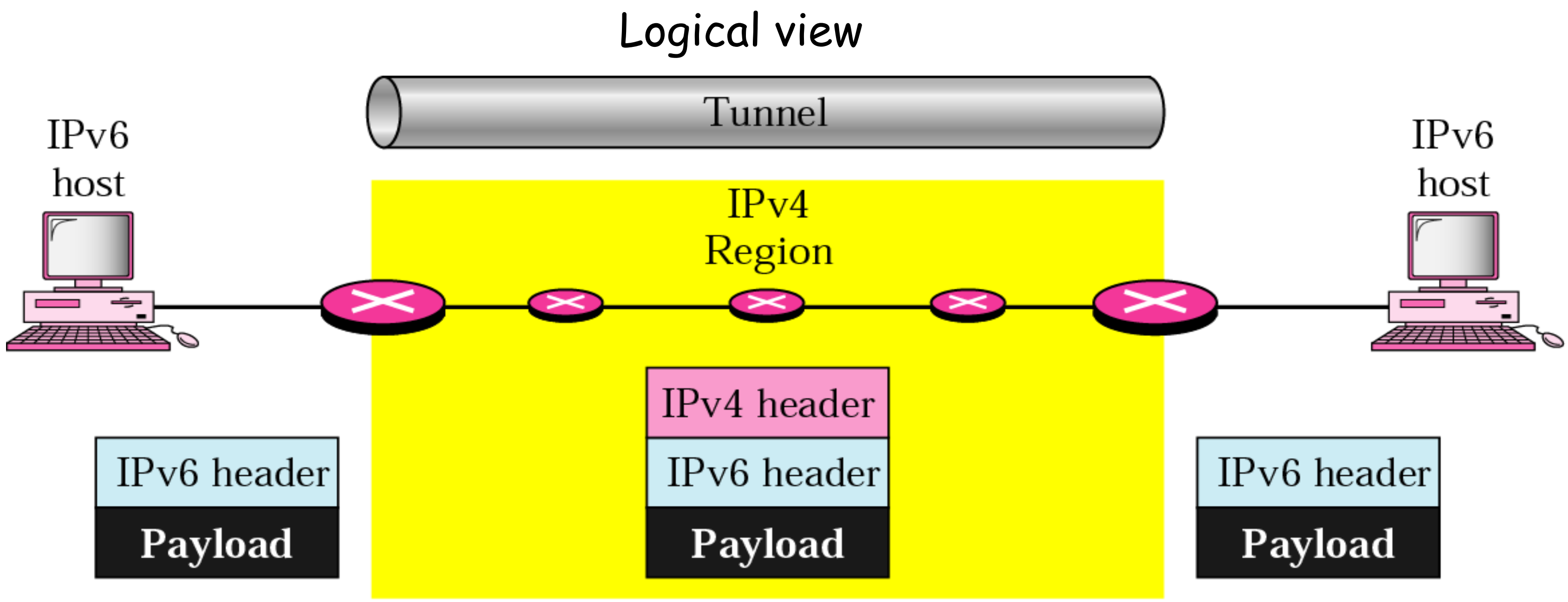


Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneous
  - no “flag days”
  - How will the network operate with mixed IPv4 and IPv6 routers?
- *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers

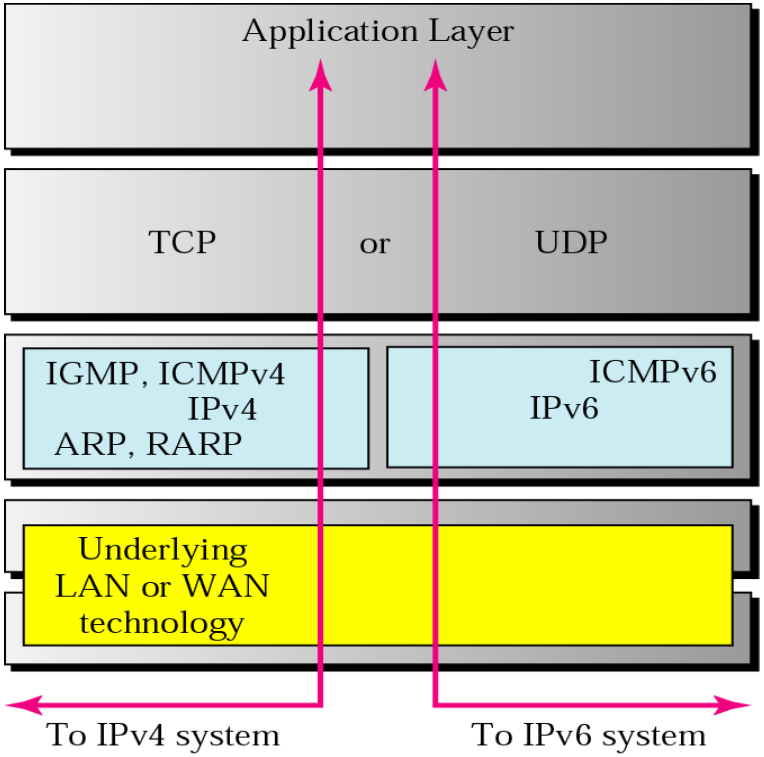
Tunneling

- IPv6 packet is encapsulated in an IPv4 packet



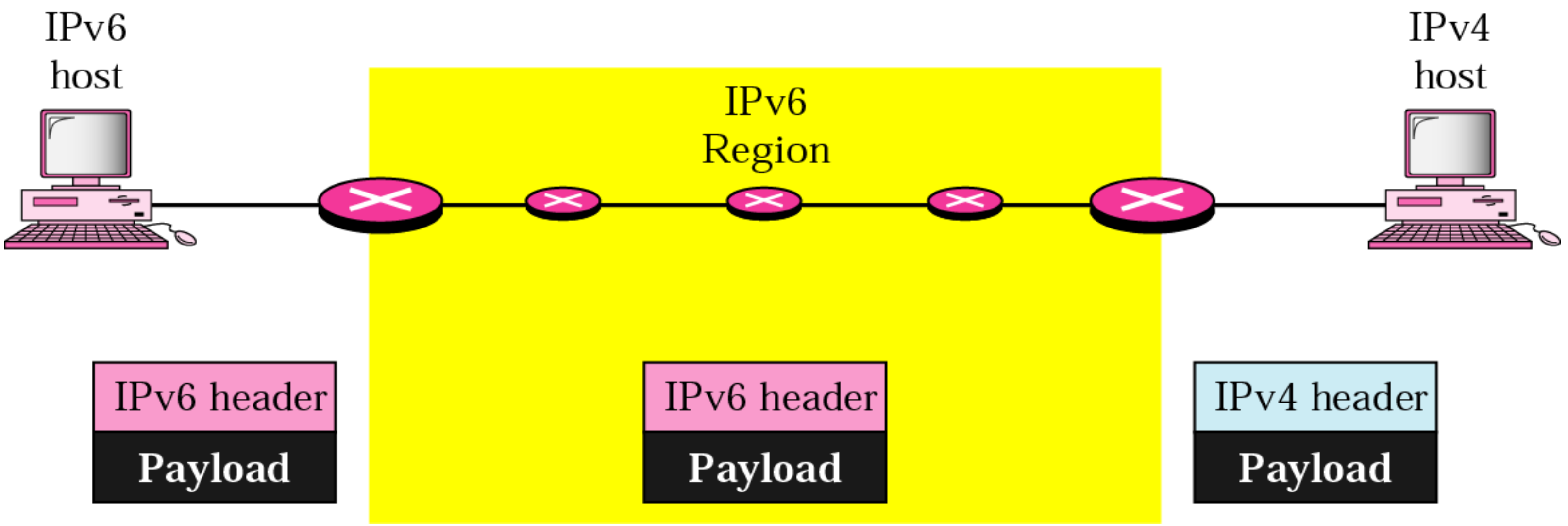
Dual stack

- All hosts have a dual stack of protocols before migrating completely to version 6



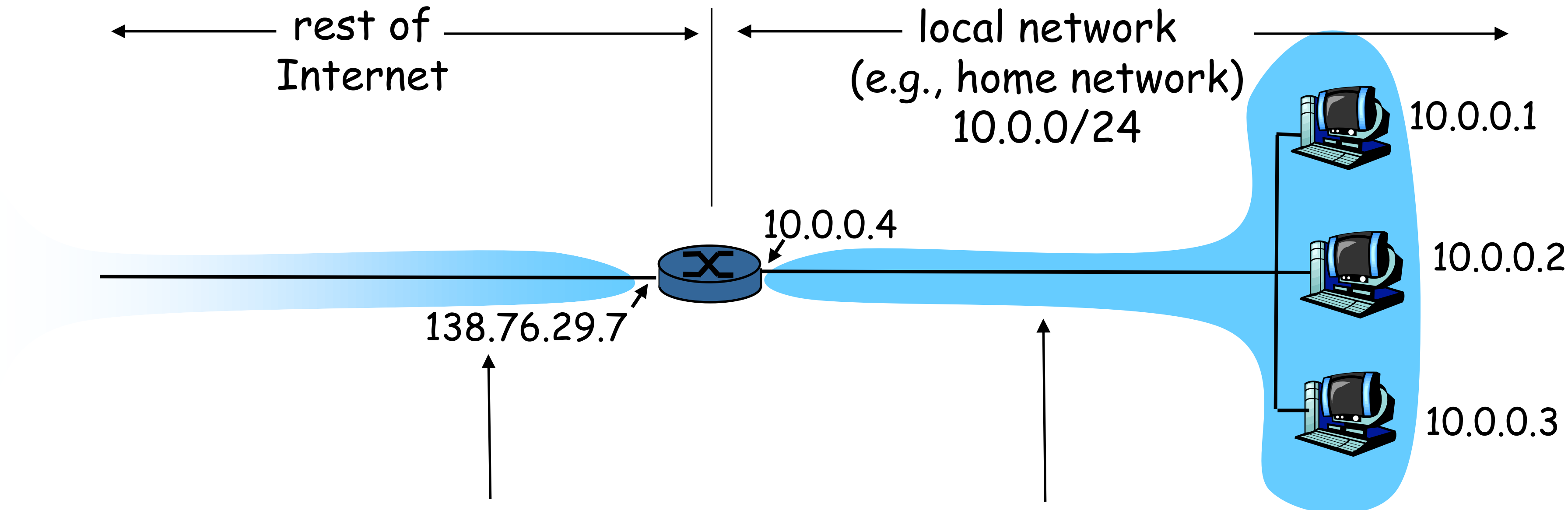
Header translation

- Necessary when the majority of the Internet has moved to IPv6 but some systems still use IPv4
- Header format must be changed totally through header translation



Network Address Translation (NAT)

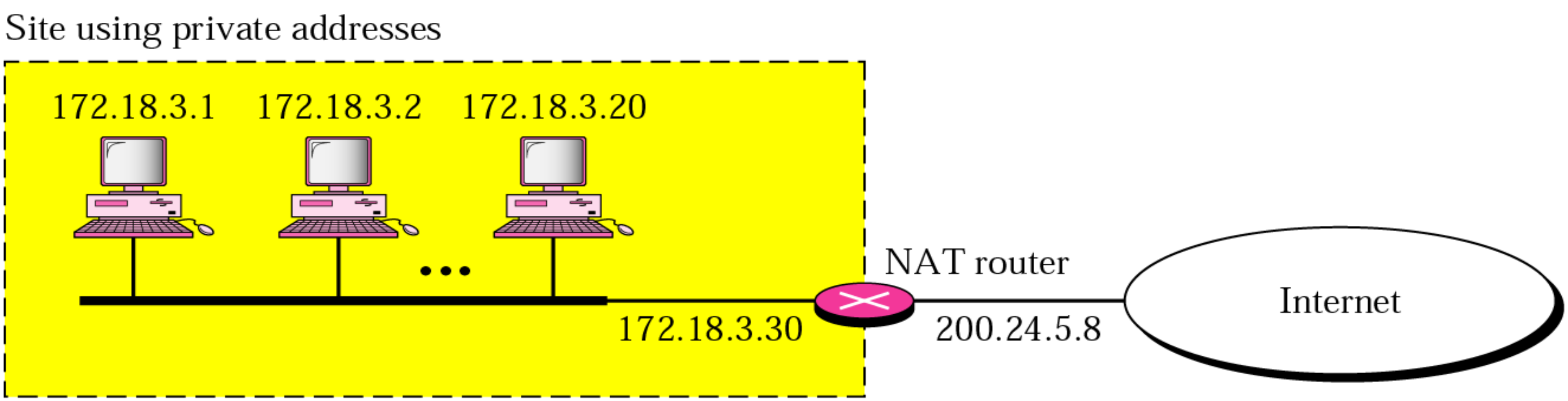
NAT: Network Address Translation



All datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: Network Address Translation



Name	IP address range	Number of IPs
24-bit block	10.0.0.0 – 10.255.255.255	16,777,216
20-bit block	172.16.0.0 – 172.31.255.255	1,048,576
16-bit block	192.168.0.0 – 192.168.255.255	65,536

NAT: Network Address Translation

- Motivation: local network uses just one IP address as far as outside word is concerned:
  - no need to be allocated range of addresses from ISP: - just one IP address is used for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).

NAT: Network Address Translation

Implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr.

- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair

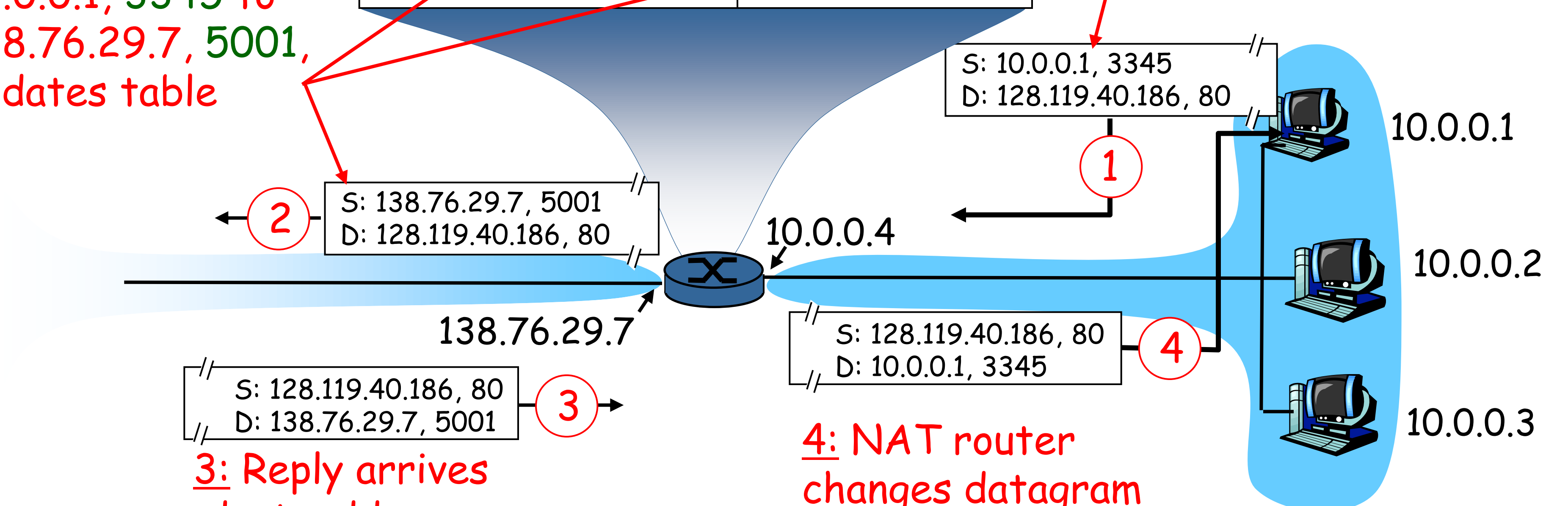
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: Network Address Translation

NAT translation table	
WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

1: host 10.0.0.1 sends datagram to 128.119.40, 80



3: Reply arrives dest. address: 138.76.29.7, 5001

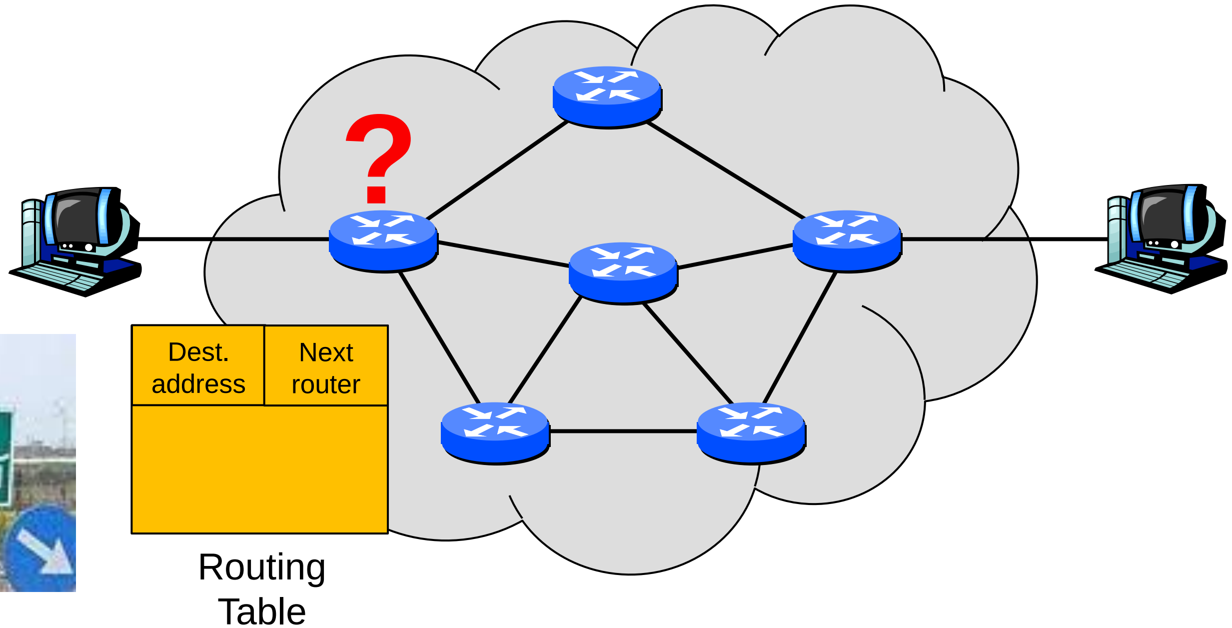
4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

NAT: Network Address Translation

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, eg, P2P applications
  - address shortage should instead be solved by IPv6

Routing

Routing

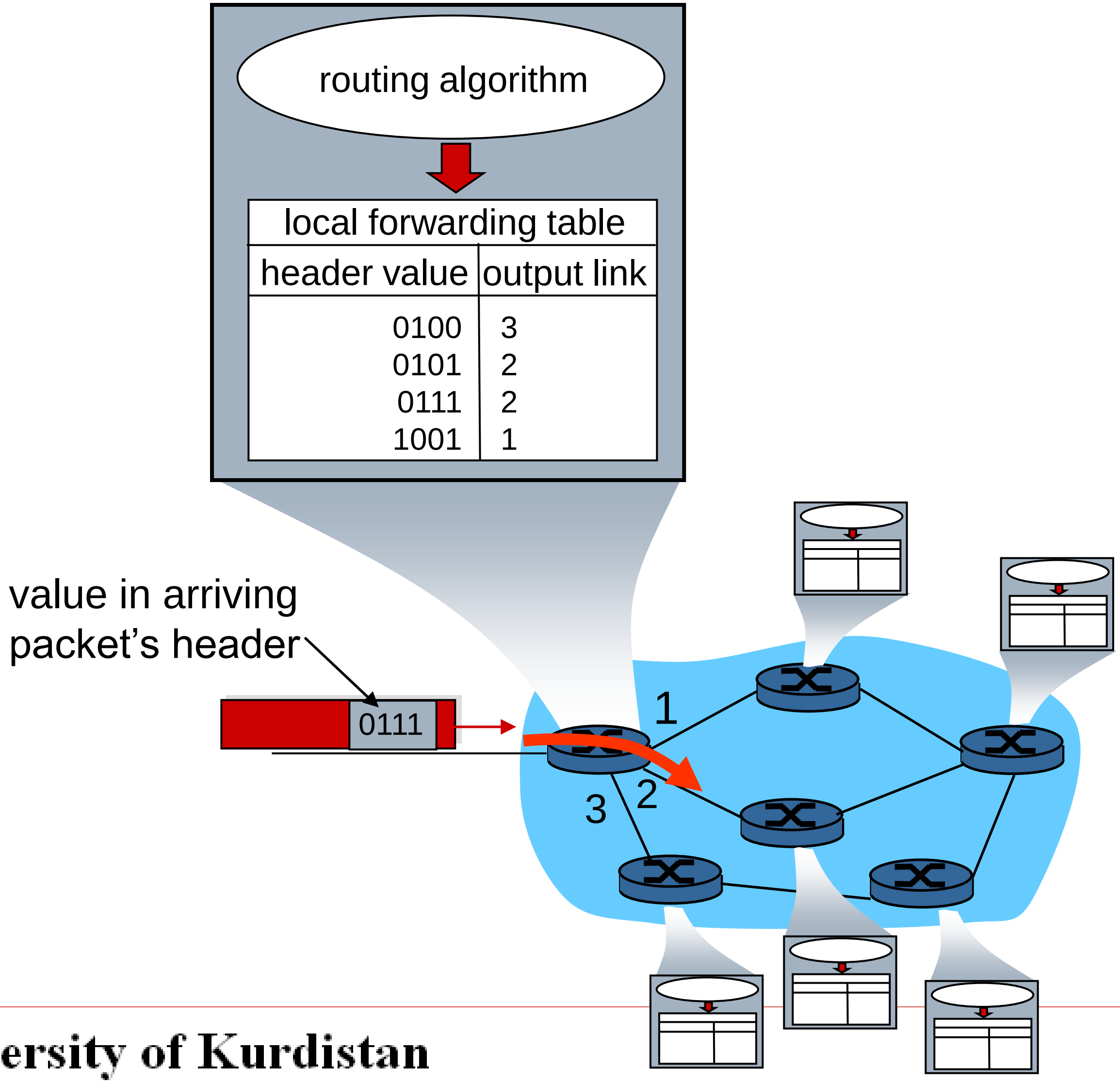


determining the most favorable path from the source of a message to its destination

Routing – most favorable route

- Short response times
- High throughput
- Avoidance of local overload situations
- Security requirements
- Shortest path

Interplay between routing and forwarding



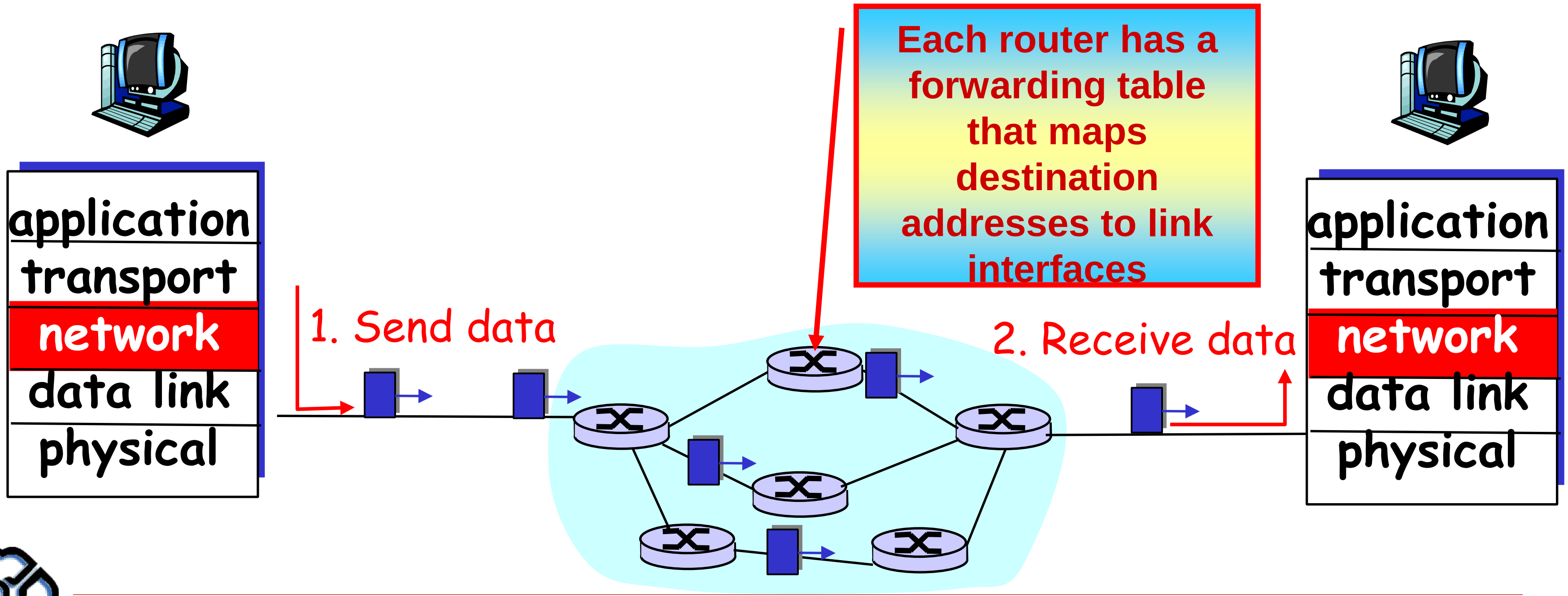
Routing & forwarding

- Not the same thing!
- Routing- filling the routing tables
- Forwarding – handling the packets based on routing tables

- Routing differs in datagram and VC networks

Datagram Routing (The internet model)

- routers: no state about end-to-end connections
  - no network-level concept of 'connection'
- packets are typically routed using destination host ID
  - packets between same source-destination pair may take different paths

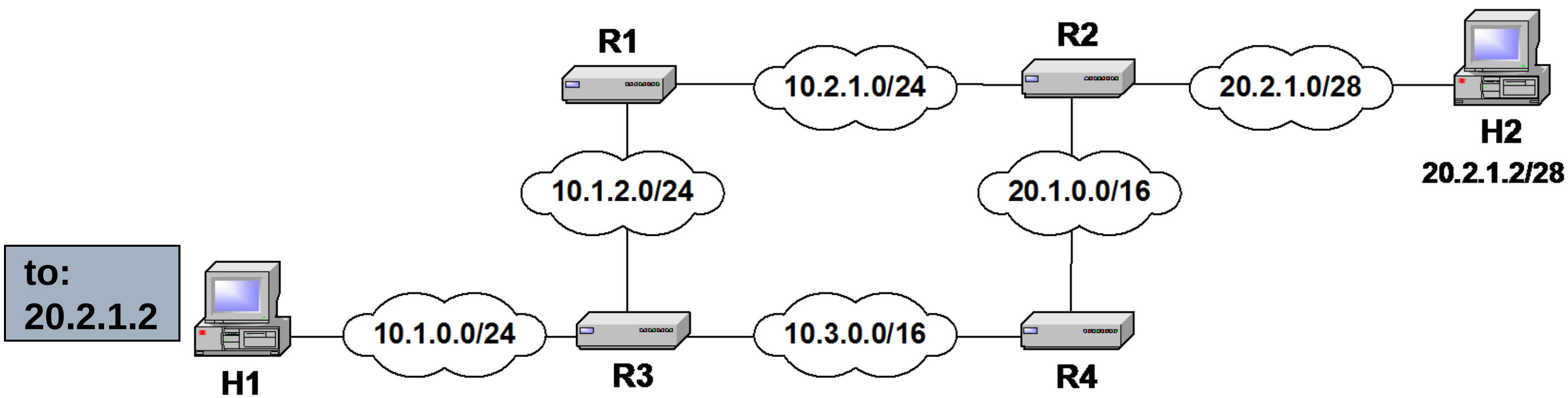


Delivery with routing tables

Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	direct
10.2.1.0/24	direct
10.3.1.0/24	R3
20.2.0.0/16	R2
30.1.1.0/28	R2

Destination	Next Hop
10.1.0.0/24	R1
10.1.2.0/24	R1
10.2.1.0/24	direct
10.3.1.0/24	R4
20.1.0.0/16	direct
20.2.1.0/28	direct

Destination	Next Hop
10.1.0.0/24	R2
10.1.2.0/24	R2
10.2.1.0/24	R2
10.3.1.0/24	R2
20.1.0.0/16	R2
20.2.1.0/28	direct



Destination	Next Hop
10.1.0.0/24	direct
10.1.2.0/24	R3
10.2.1.0/24	R3
10.3.1.0/24	R3
20.1.0.0/16	R3
20.2.1.0/28	R3

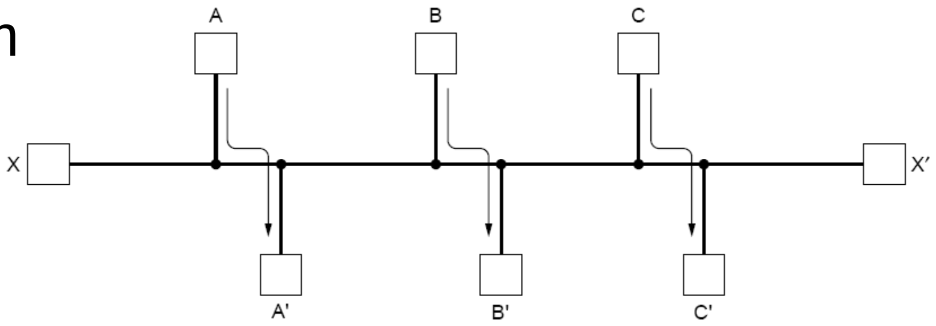
Destination	Next Hop
10.1.0.0/24	direct
10.1.2.0/24	direct
10.2.1.0/24	R4
10.3.1.0/24	direct
20.1.0.0/16	R4
20.2.1.0/28	R4

Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	R3
10.2.1.0/24	R2
10.3.1.0/24	direct
20.1.0.0/16	direct
20.2.1.0/28	R2

Routing - properties

1. correctness
2. simplicity
3. robustness
   - updating possibility
   - should cope with changes in the topology and traffic
4. stability
   - must converge to equilibrium
5. fairness
6. optimality
   - ➢ min mean packet delay
   - ➢ max total network throughput

➢ 5 & 6 often contradictory



Routing algorithms

- **DYNAMIC**
  - change routing decisions to reflect changes in the topology
  - adapt for changes in the traffic (load change)
  - ALGORITHMS: where routers get the information from?
    - locally
    - from adjacent routers
    - from all routers
  - ALGORITHMS: when they change their routes?
    - every ΔT sec
    - when the load changes
    - when topology changes
- **STATIC**
  - routes computed in advance
    - node failures, current load etc. not taken into account

Global & decentralized routing algorithms

1. **Global routing algorithm**
   - least-cost path calculated using global knowledge about network
   - input: connectivity between all nodes & link costs nodes
   - link state algorithms
2. **Decentralized routing algorithm**
   - least-cost path calculated in an iterative, distributed manner
   - no node has complete info about the cost of all network links
   - begins with cost of directly attached links
   - info exchange with neighbouring nodes
   - distance vector algorithms

Two basic dynamic algorithms

- **<u>Distance Vector Routing</u>**

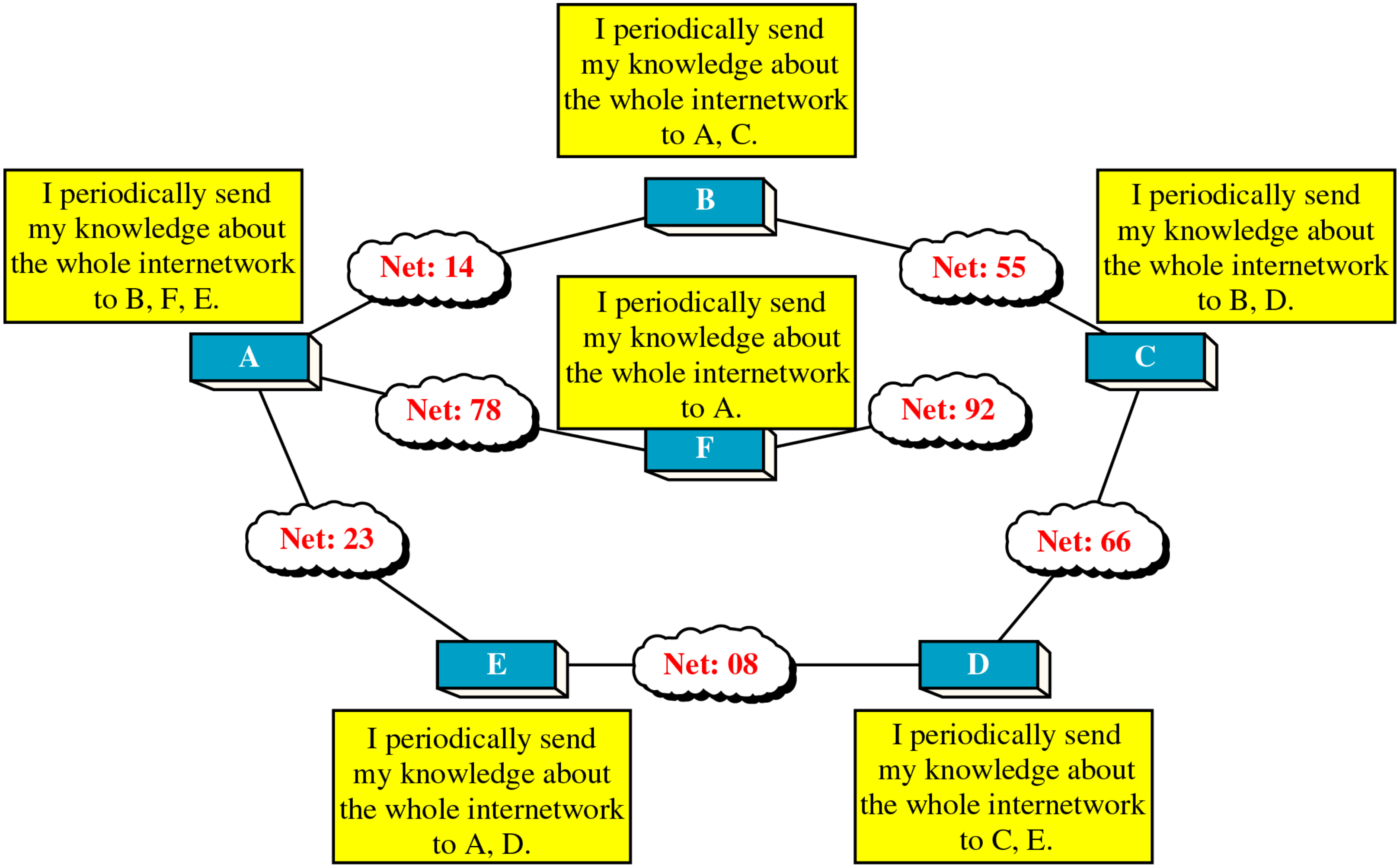
  - routing protocols are like road signs
  - used in the ARPANET

- **Link State Routing**
  - routing protocols are more like a road map
  - used in the newer Internet Open Short Path First (OSPF) protocol

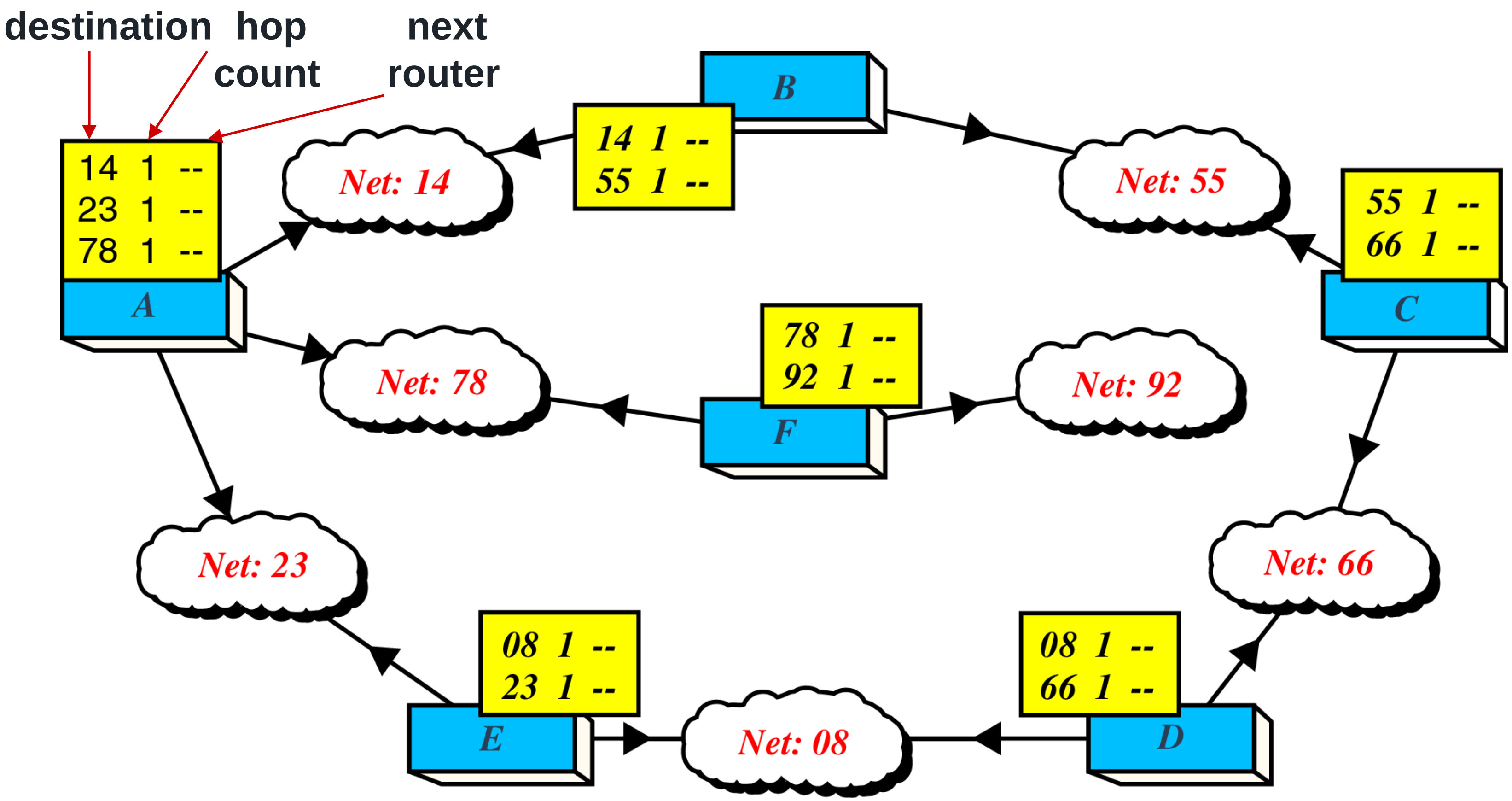
The Distance Vector Routing

- **dynamic algorithm**
  - takes current network load into account
- **distributed**
  - each node receives information from its directly attached neighbours, performs a calculation, distribute the results back to neighbours
- **iterative**
  - alg performed in steps until no more information to change
  - initially, each node knows only about its adjacent nodes
- **asynchronous**
  - nodes do not operate in lockstep with each other

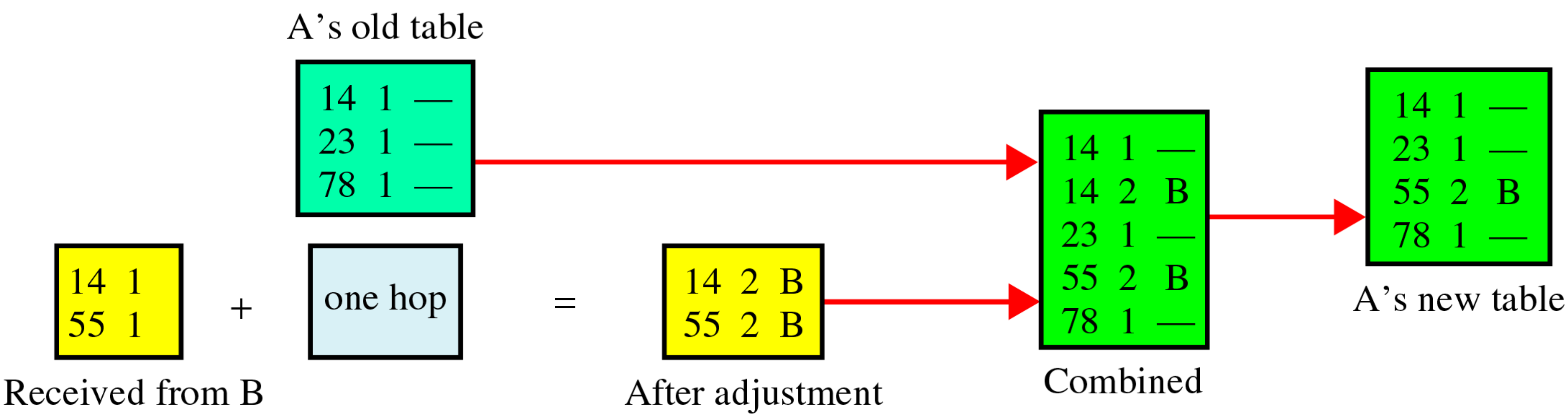
The concept of distance vector routing



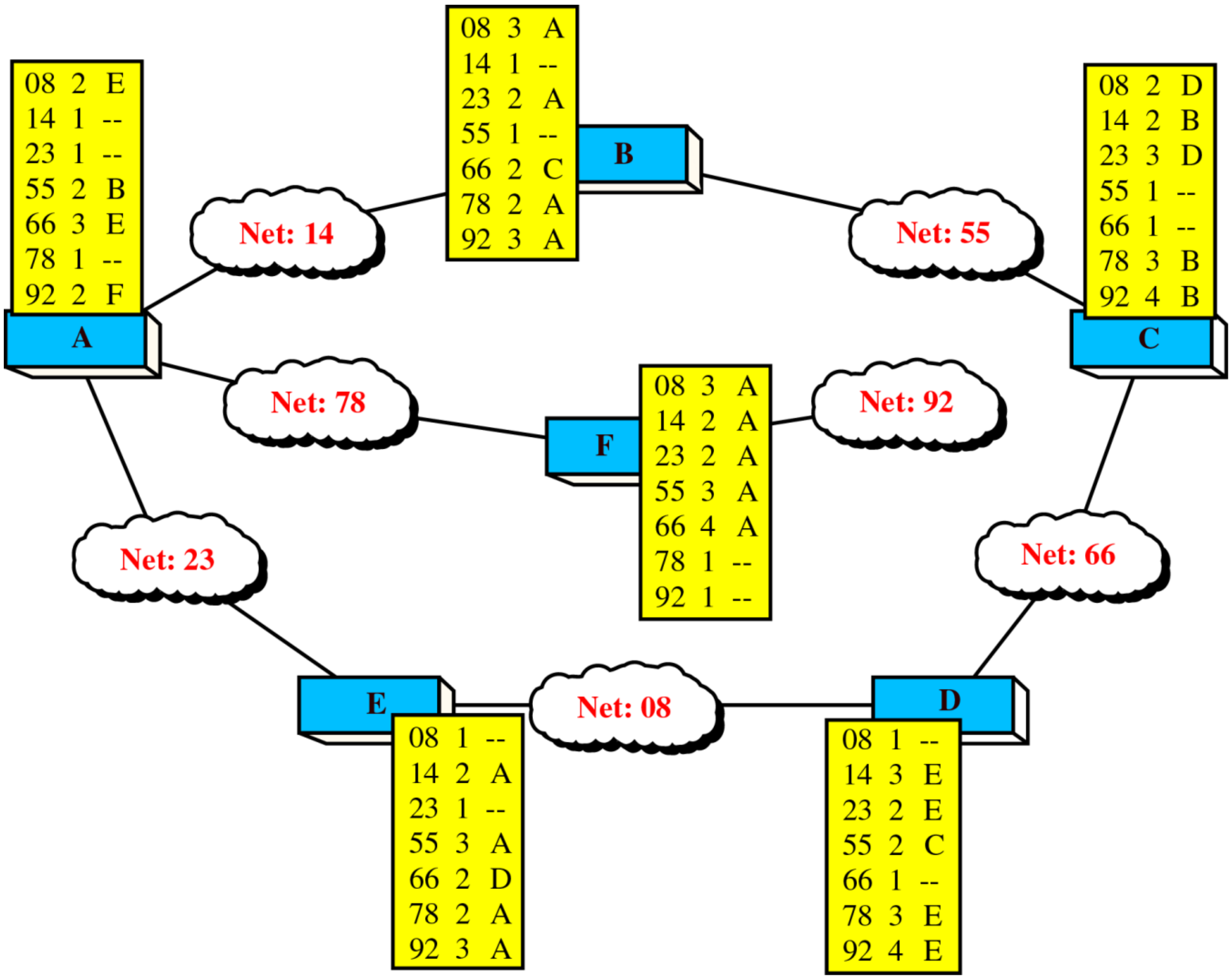
Routing Table Distribution



Updating Routing Table for Router A



Final Routing Tables



Problems in distance vector routing

- Two problems
  1. Link bandwidth not taken into account for metric, only the queue length
     - all the lines at that time 56 Kbps
  2. Too long time to converge
     - QUESTION: when the algorithm converges?
     - ANSWER: when every node knows about all other nodes and networks and computes the shortest path to them

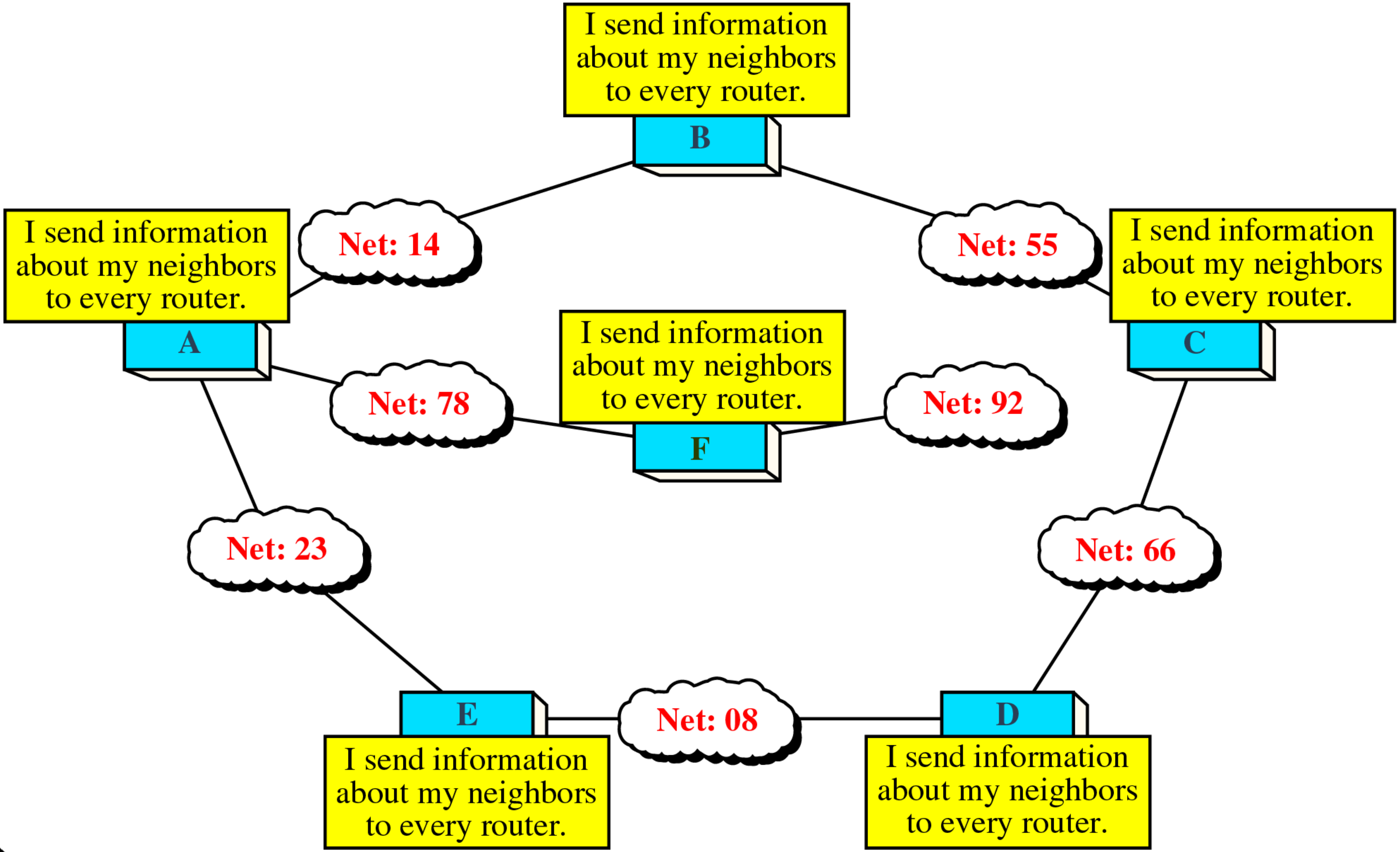
Two basic algorithms

- Distance Vector Routing
- **<u>Link State Routing</u>**

A Link state routing algorithm

- link state broadcast – node learn about path costs from its neighbors
- inform the neighbors whenever the link cost changes
  - hence the name link state

The concept of link state routing



Link state routing

- Each router does the following (repeatedly):

  1- **discover neighbors**, particularly, learn their network addresses
    - A router learns about its neighbours by sending a special HELLO packet to each point-to-point line. Routers on the other end send a reply

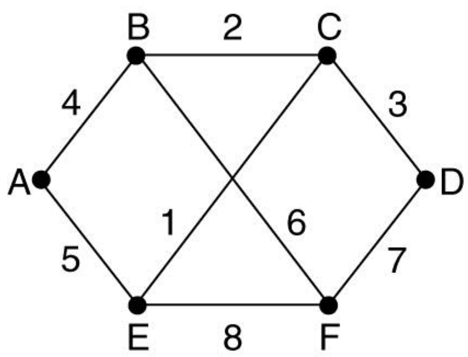
  2- **measure cost** to each neighbor
    - e.g. by exchanging a series of packets
    - sending ECHO packets and measuring the average round-trip-time
    - include traffic-induced delay?

  3- construct a link state packet

  4- send this packet **to all other routers**
    - using what route information? chicken / egg
    - what if re-ordered? or delayed?

  5- compute ***locally*** the shortest path to every other router when this information is received (**using dijkstra's algorithm**)

Constructing link state packets



(a)

subnet

Link State Packets

sender

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b)

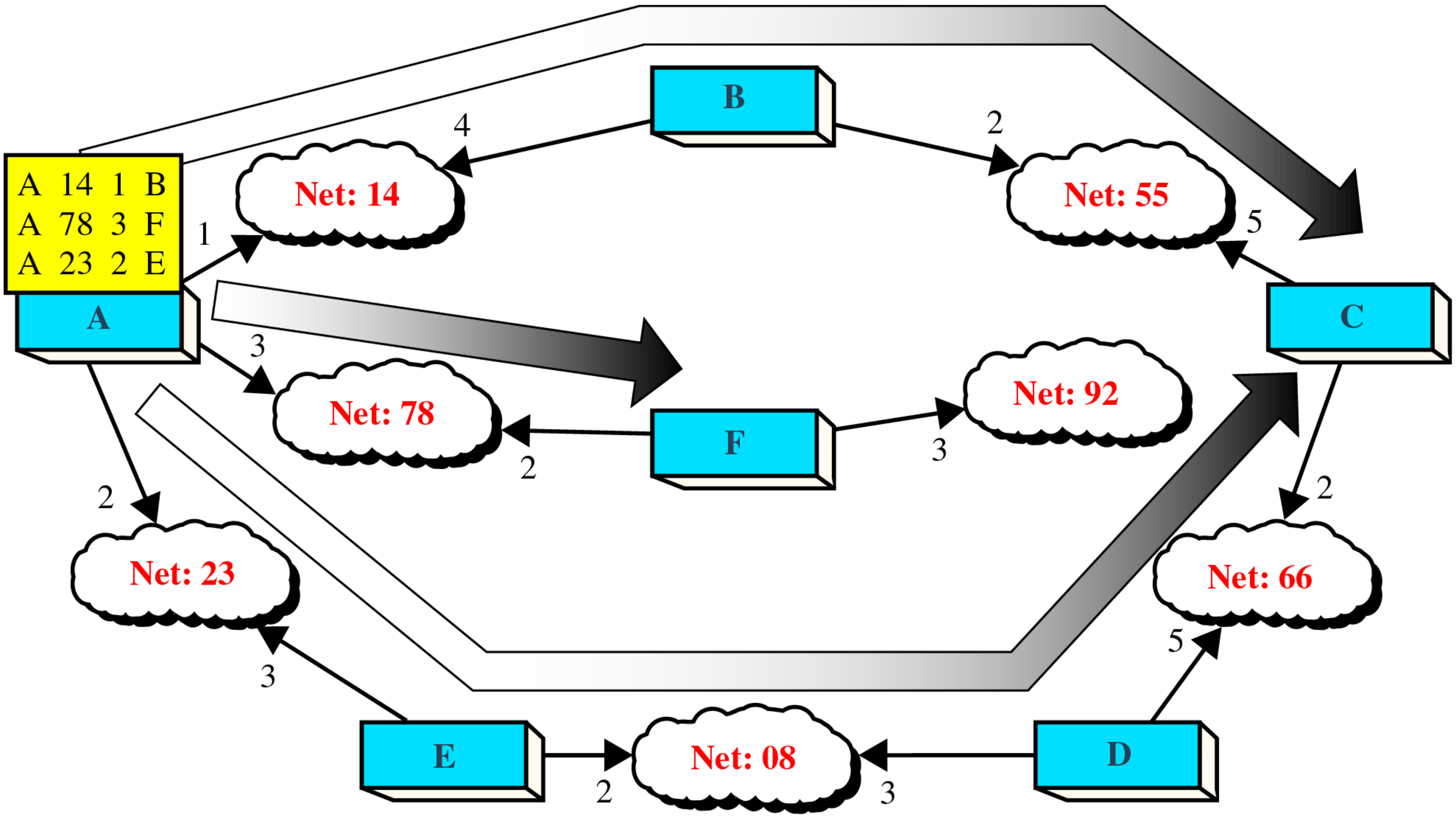
link state packets for this subnet

- **When to build these packets?**
  - at regular time intervals
  - on occurrence of some significant event

Distributing the link state packets

- Typically, flooding
  - routers recognize packets passed earlier
    - sequence number incremented for each new packet sent
    - routers keep track of the (source router, sequence) pair
    - thus avoiding the exponential packet explosion
  - first receivers start changes already while changes are being reported
  - sequence numbers wrap around or might be corrupted (a bit inversed – 65540 instead of 4)
    - 32 bit sequence number (137 years to wrap)
    - To avoid corrupted sequences (or a router reboot) and therefore prevent any update, the state at each router has an *age* field that is decremented once a second
    - but, need additional robustness in order to deal with errors on router-to-router lines
      - acknowledgements

Distributing the link state packets



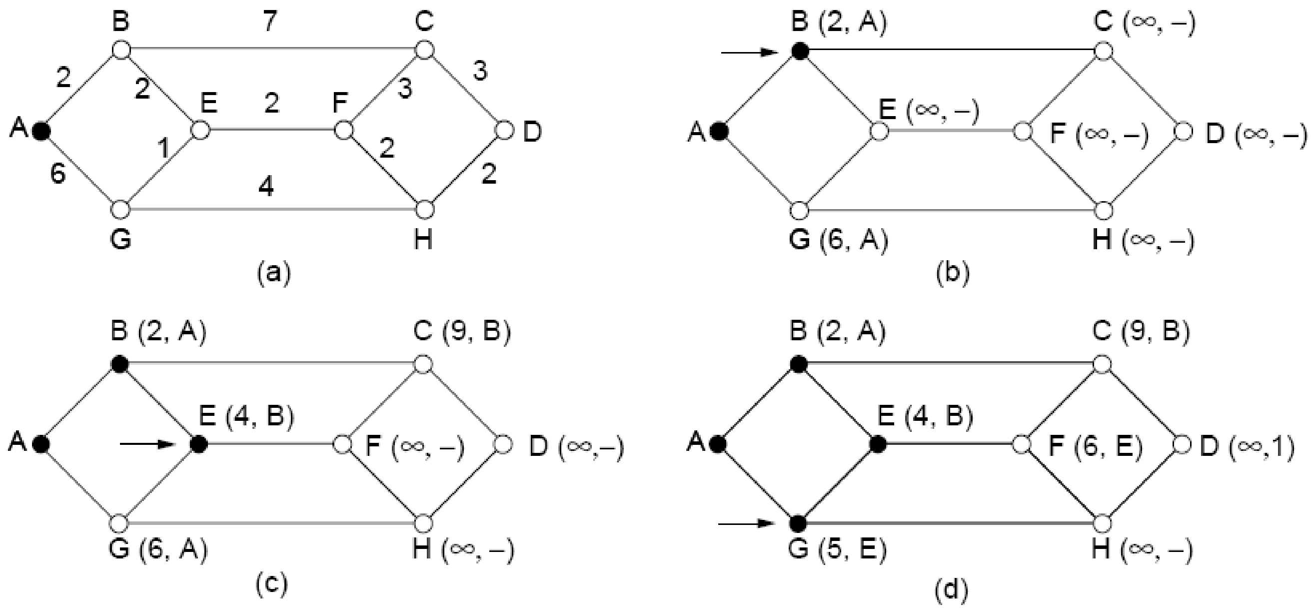
Dijkstra's algorithm to compute the shortest path

```
Initialization:
  N = {A}
  for all nodes v
    if v adjacent to A
      then D(v) = c(A,v)
      else D(v) = infty

Loop
  find w not in N such that D(w) is a minimum
  add w to N
  update D(v) for all v adjacent to w and not in N:
     D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   shortest path cost to w plus cost from w to v */
until all nodes in N
```

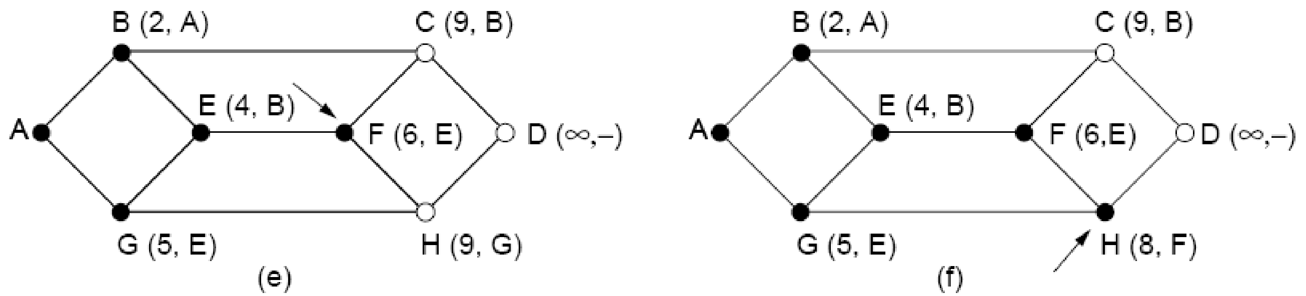
- $c(i,j)$ link cost from node i to j
- $c(i,j)=\infty$ if i & j not directly conn
- $D(v)$ cost of the path from the source node to destination v
- N set of nodes whose least-cost path from the source is definitely known

Dijkstra’s algorithm - sketch



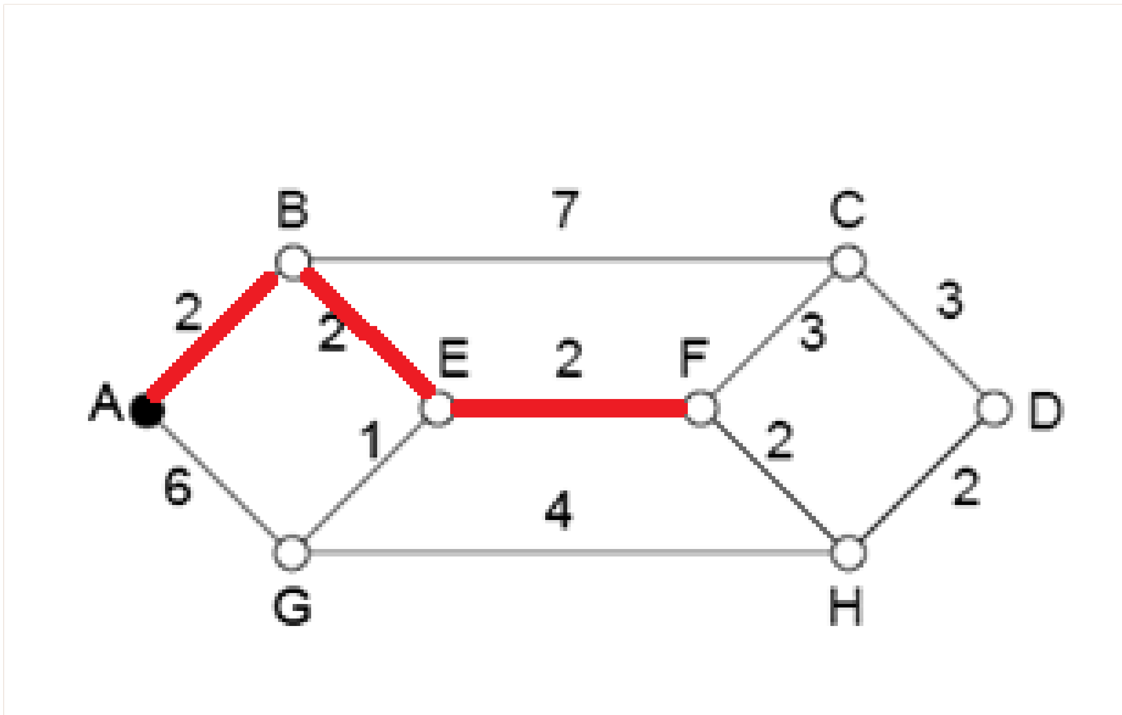
step	N	D(B),p(B)	D (C),p (C)	D(D),p(D)	D(E), p(E)	D(F), p(F)	D(G),p(G)	D(H),p(H)
0	A	2,A	∝	∝	∝	∝	6,A	∝
1	AB		9,B	∝	4,B	∝	6,A	∝
2	ABE		9,B	∝		6,E	5,E	∝
3	ABEG		9,B	∝		6,E		9,G
4	ABEGF		9,B	∝				8,F

Dijkstra’s algorithm - sketch



step	N	D(B),p(B)	D (C),p (C)	D(D),p(D)	D(E), p(E)	D(F), p(F)	D(G),p(G)	D(H),p(H)
0	A	2,A	∝	∝	∝	∝	6,A	∝
1	AB		9,B	∝	4,B	∝	6,A	∝
2	ABE		9,B	∝		6,E	5,E	∝
3	ABEG		9,B	∝		6,E		9,G
4	ABEGF		9,B	∝				8,F
5	ABEGFH		9,B	10,H				
6	ABEGFHC			10,H				
5	ABEGFHCD							

Shortest path



Shortest path from A to F using Dijkstra’s algorithm

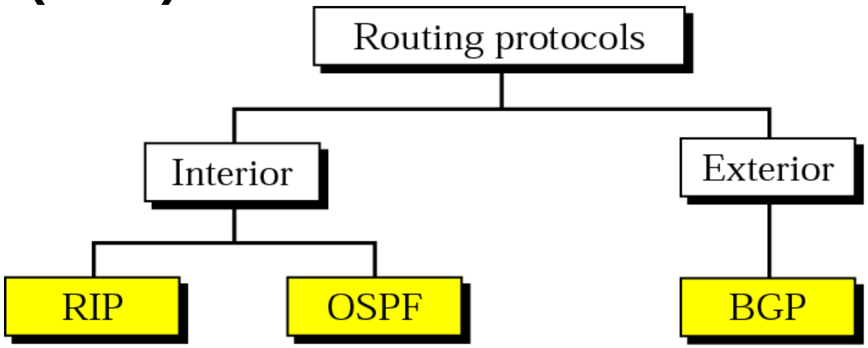
Routing in the Internet

- What would happen if hundreds of millions of routers execute the same routing algorithm to compute routing paths through the network?
- Scale
  - large overhead
  - enormous memory space in the routers
  - no bandwidth left for data transmission
  - would DV algorithm converge?
- Administrative autonomy
  - an organization should run and administer its networks as wishes but must be able to connect it to "outside" networks

Hierarchical routing

- The Internet uses hierarchical routing
  - it is split into Autonomous Systems **(AS)**
    - routers at the border: <u>gateways</u>
    - gateways must run both <u>intra</u> & <u>inter</u> AS routing protocols
  - routers within AS run the same routing algorithm
    - the administrator can chose any <u>Interior Gateway Protocol</u>
      - Routing Information Protocol **(RIP)**
      - Open Shortest Path First **(OSPF)**
  - between AS gateways use <u>Exterior Gateway Protocol</u>
    - Border Gateway Protocol **(BGP)**

Why do we have different protocols for inter & intra AS routing?

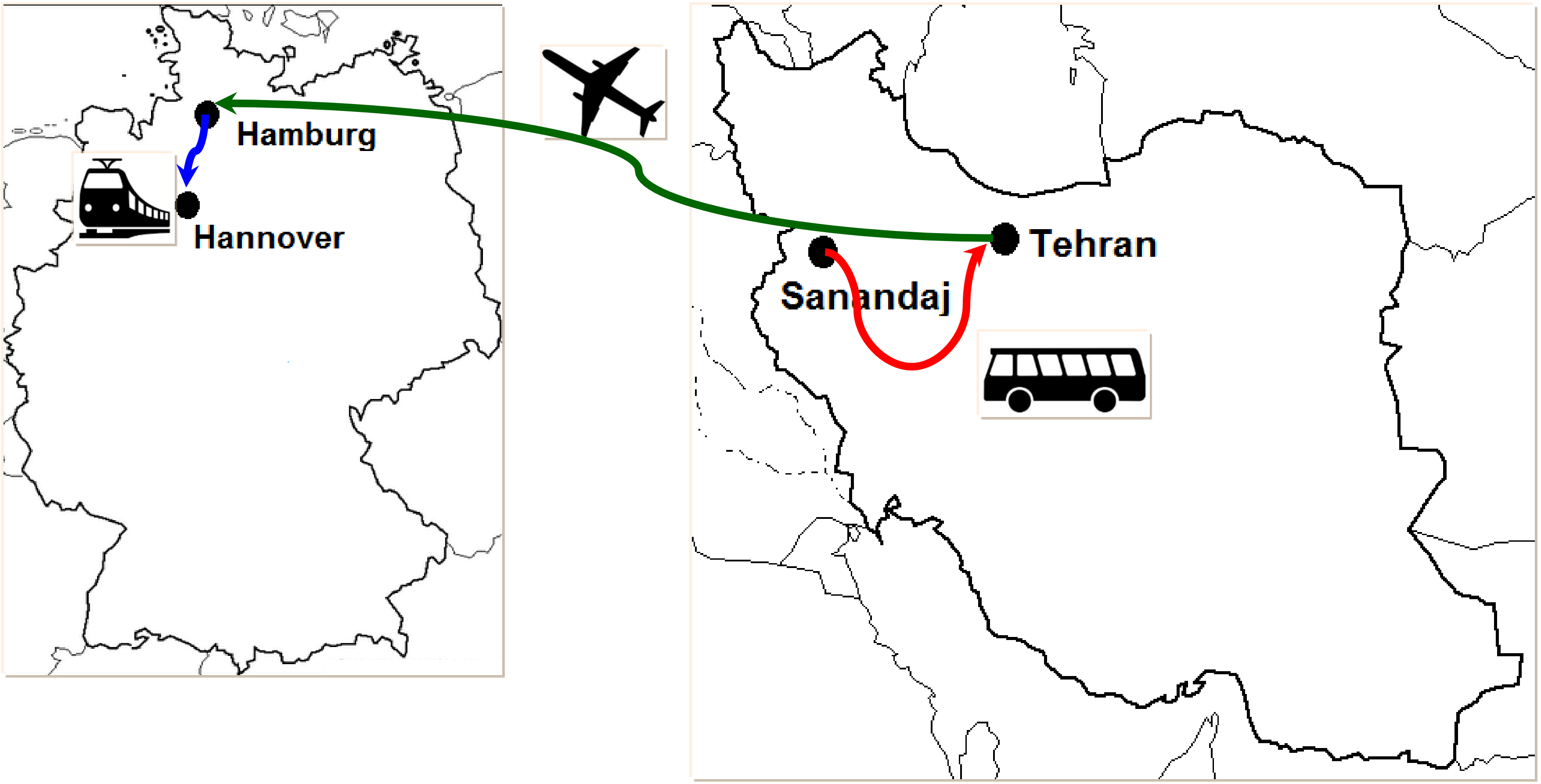


Autonomous Systems

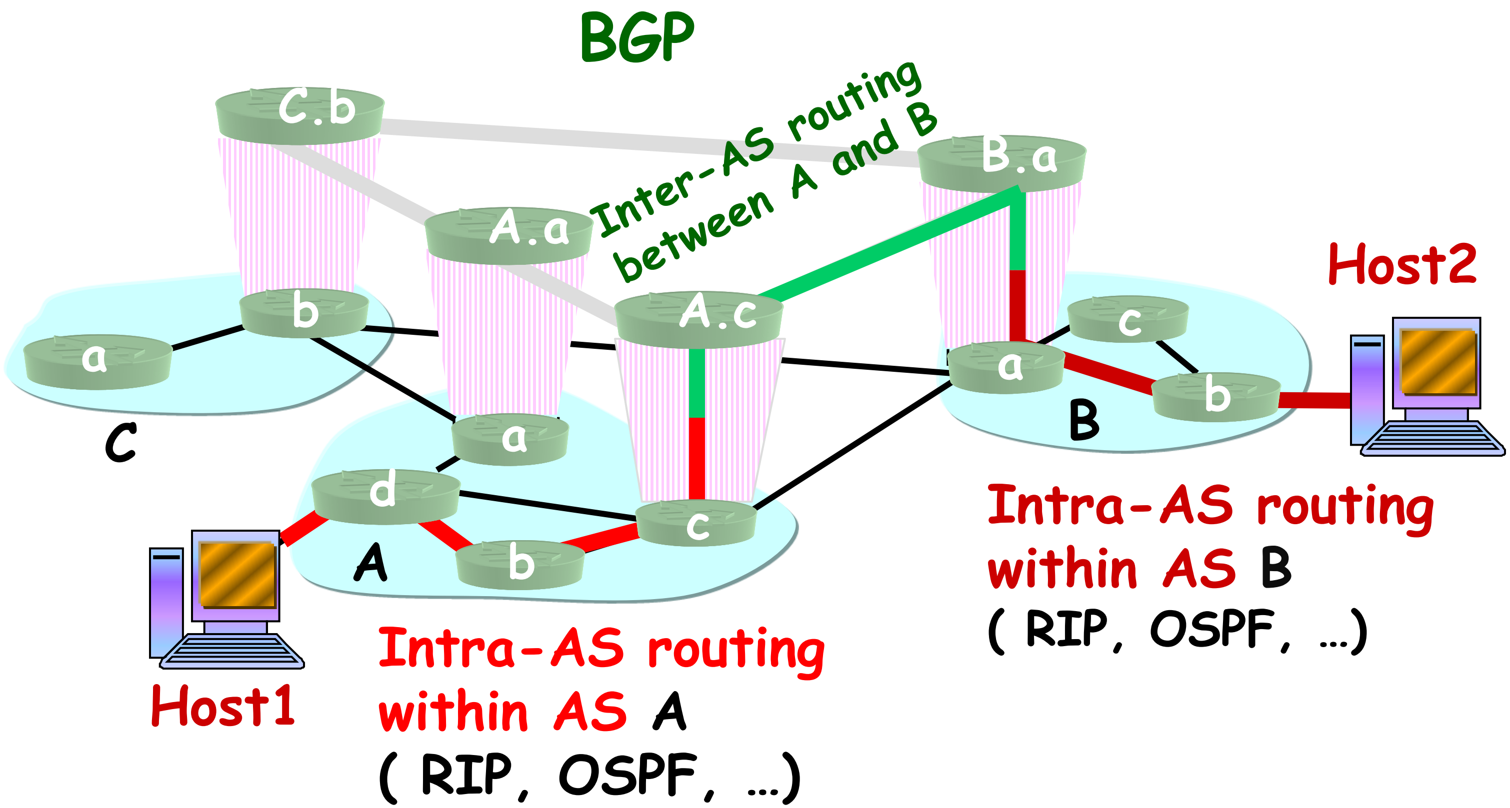
- An **autonomous system** is a region of the Internet that is administered by a single entity.
- Examples of autonomous regions are:
  - UVA’s campus network
  - MCI’s backbone network
  - Regional Internet Service Provider

- Routing is done differently within an autonomous system (**intradomain routing**) and between autonomous system **(interdomain routing**).

Hierarchical routing (analogy)

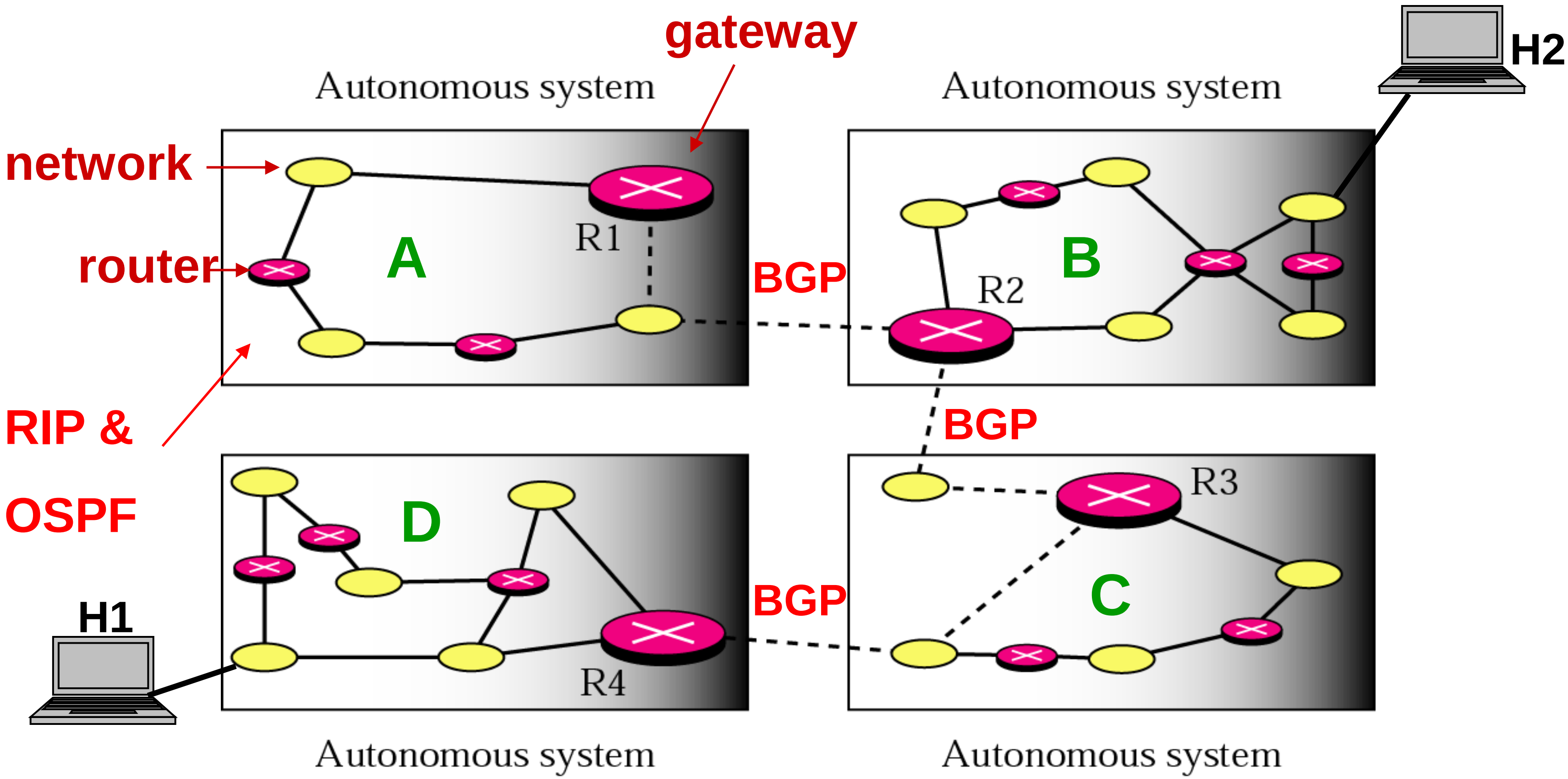


Intra-AS and Inter-AS routing



Inter AS routing Border Gateway Protocol

it is *de facto* standard interdomain routing protocol in today's Internet




Questions