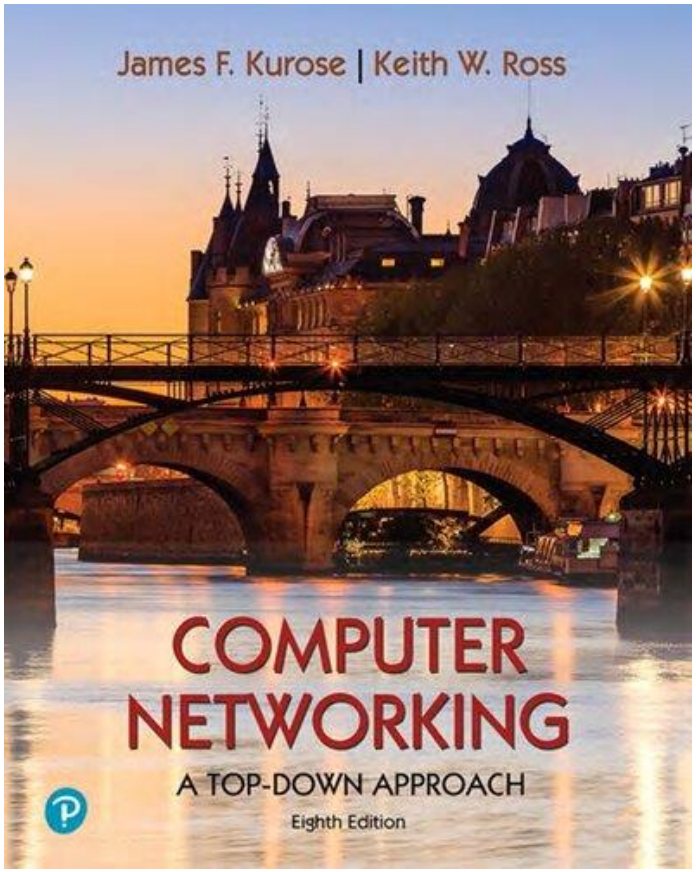
I3304
Network administration and security

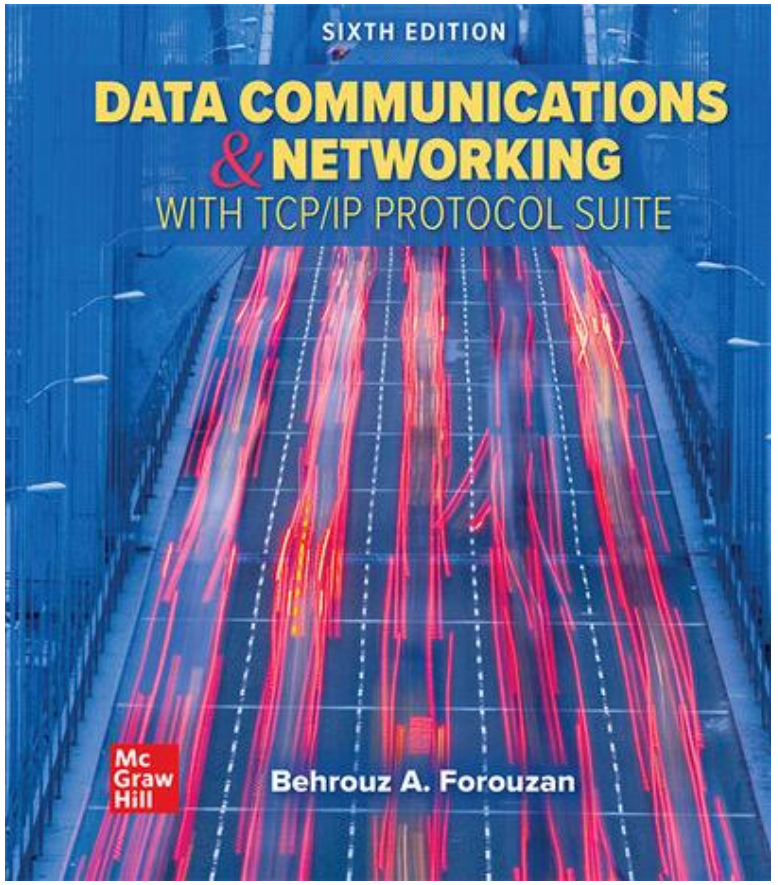
Ahmad Fadlallah

Before to Start

- I3304: 50 Hours course
- Instructor Information
  - Instructor: Ahmad Fadlallah
  - Office Hours: department schedule or by appointment
  - email: ahmad.fadlallah@ul.edu.lb
- Course Information
  - Lectures (Labs included):
    - Wednesday 08:00 –9:40
    - Thursday 16:30-18:00
  - Exercises: integrated in the course

Reference Textbooks

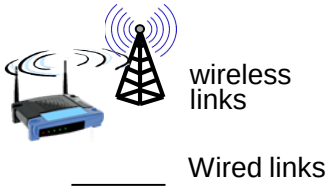


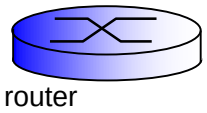


Outline

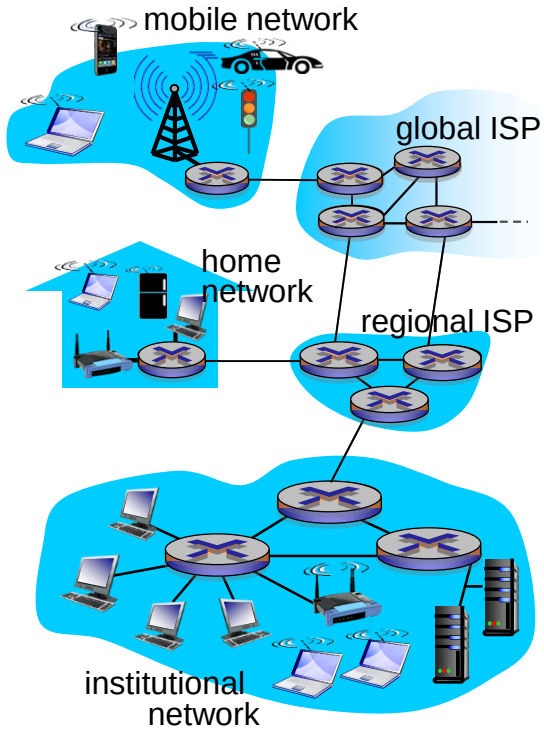
- Introduction
  - Introduction to the course
  - Recall Network Basics (I2208)
- Network Layer
  - IP packet structure (Recall)
  - Static Routing
  - Dynamic Routing Algorithm
  - Dynamic Routing Protocols
  - NAT (Network Address Translation)
- Transport Layer
  - Function of the transport layer
  - UDP Protocol
  - TCP Protocol
    - Connection management
    - Flow control
    - Congestion control
- Application Layer
  - HTTP protocol
  - FTP protocol
  - Mail protocols
  - DNS
- Introduction to Security
  - Security services
  - Cryptography
  - Digital Signature
  - Principle of network security protocols

What's the Internet: "nuts and bolts" view





- Billions of connected computing devices:
  - hosts = end systems
  - running network apps
- Communication links
  - fiber, copper, radio, satellite
  - transmission rate: bandwidth
- Packet switches: forward packets (chunks of data)
  - Routers and switches



“Fun” Internet-connected devices

IP picture frame
http://www.ceiva.com/

Web-enabled toaster +
weather forecaster

Tweet-a-watt:
monitor energy use

Internet
refrigerator

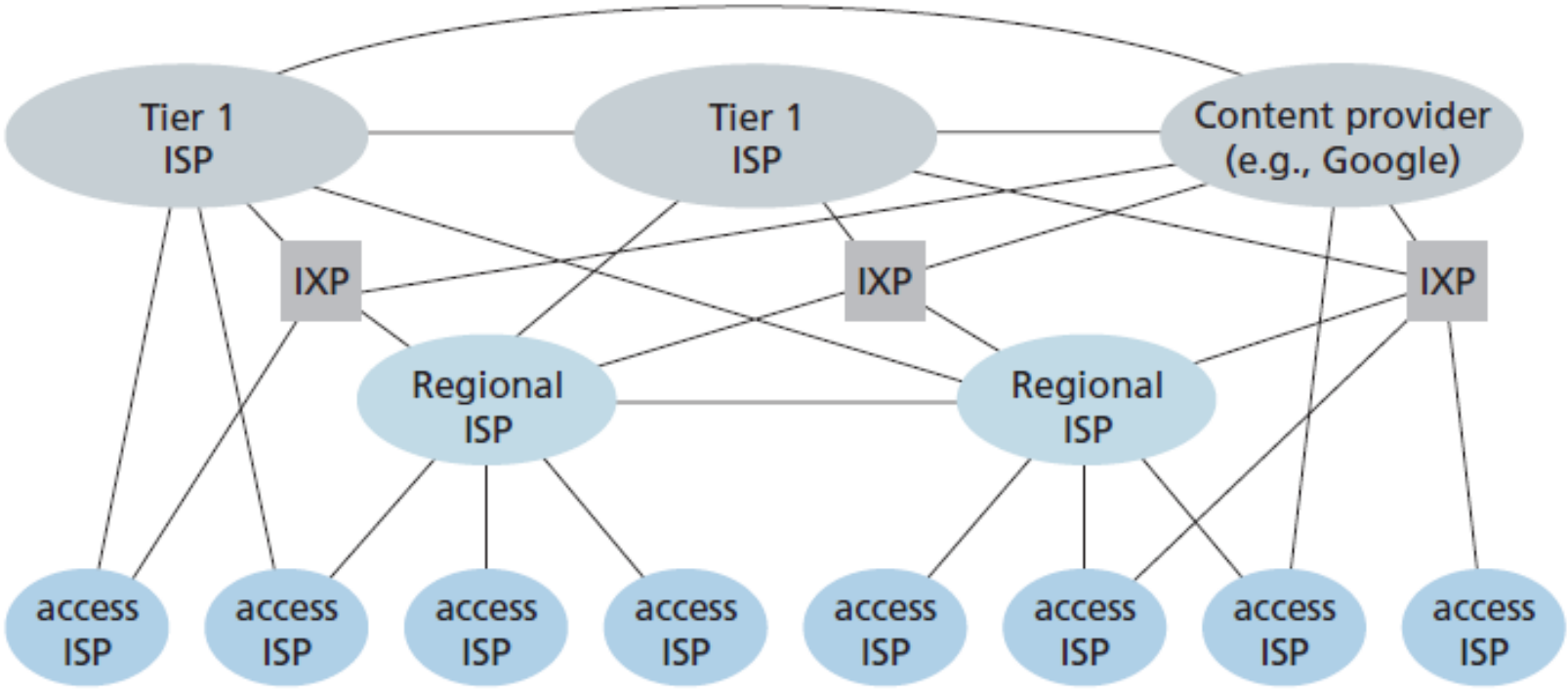
Slingbox: watch,
control cable TV remotely

sensorized,
bed
mattress

Internet phones

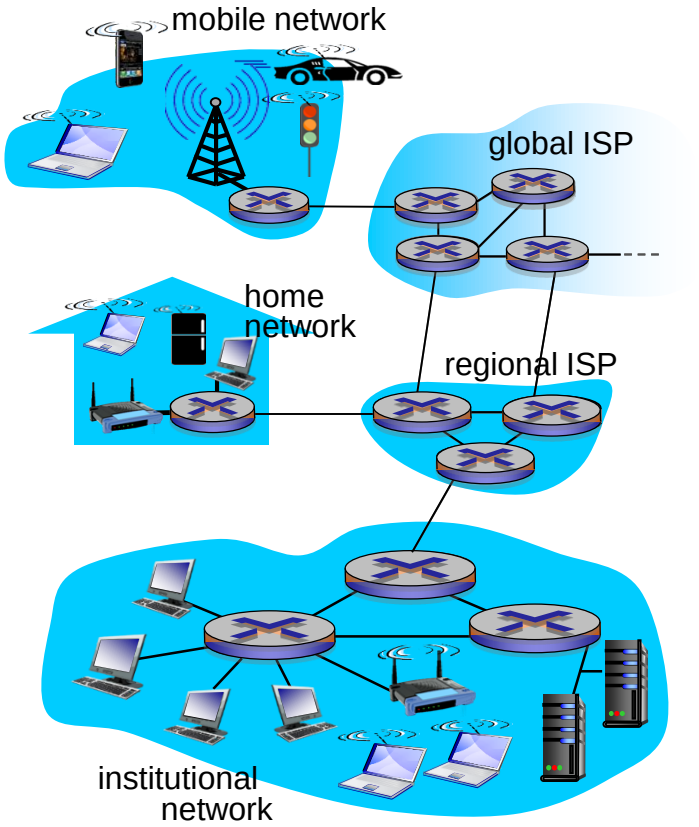
What's the Internet: "nuts and bolts" view

- Internet: "network of networks"
  - Interconnected ISPs



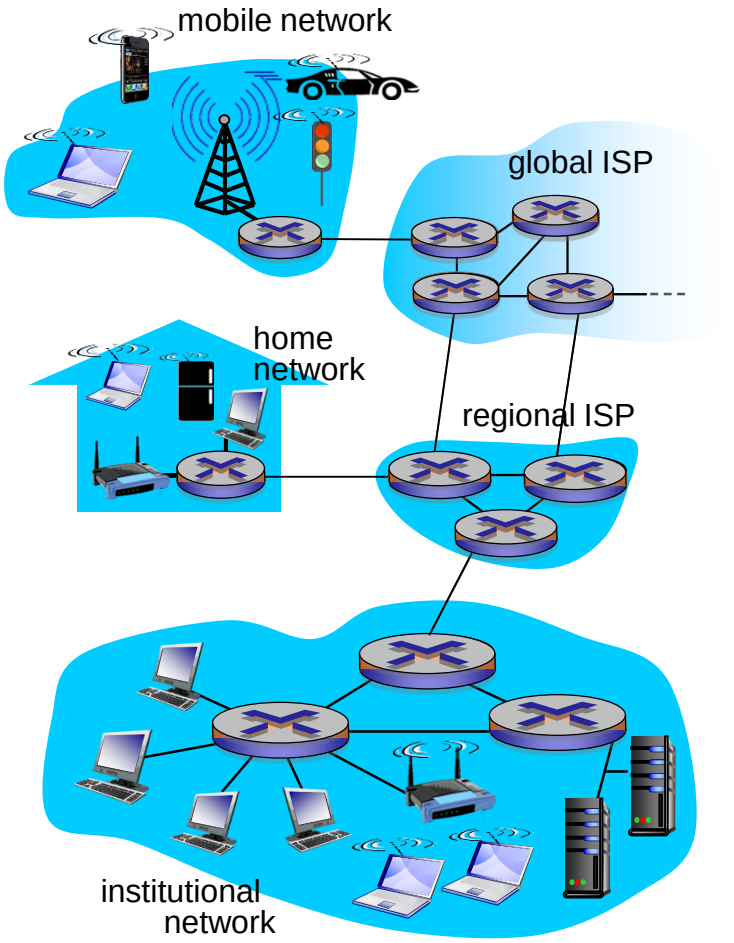
What's the Internet: "nuts and bolts" view

- Protocols control sending, receiving of messages
  - e.g., TCP, IP, HTTP, Skype, 802.11
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



What's the Internet: a service view

- Infrastructure that provides services to applications:
  - Web, VoIP, email, games, e-commerce, social nets, …
- Provides programming interface to apps
  - hooks that allow sending and receiving app programs to "connect" to Internet
  - Provides service options, analogous to postal service



What's a protocol?

Human protocols

- "what's the time?"
- "I have a question"
- introductions

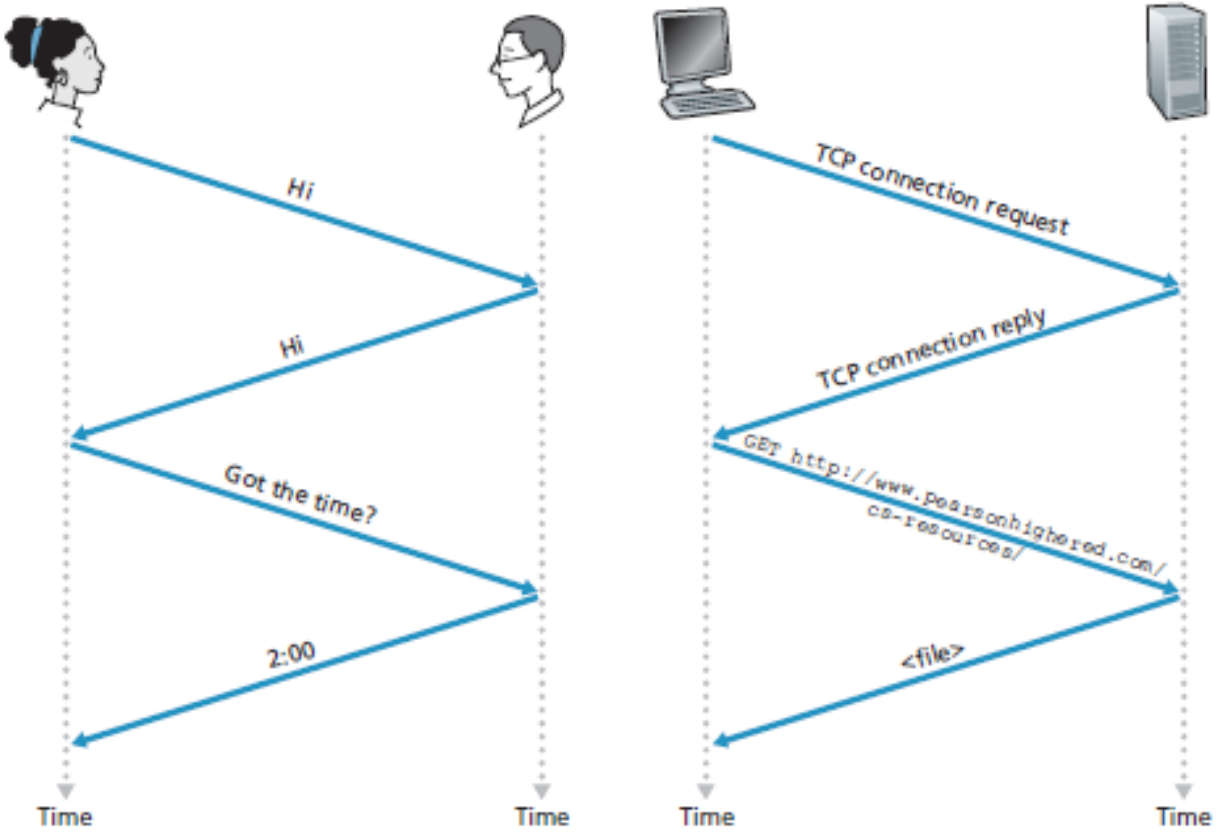
Network protocols

- Machines rather than humans
- All communication activity in Internet governed by protocols

… specific messages sent
… specific actions taken when messages received, or other events

protocols define format, order of messages sent and received among network entities, and actions taken on message transmission, receipt

What's a protocol?



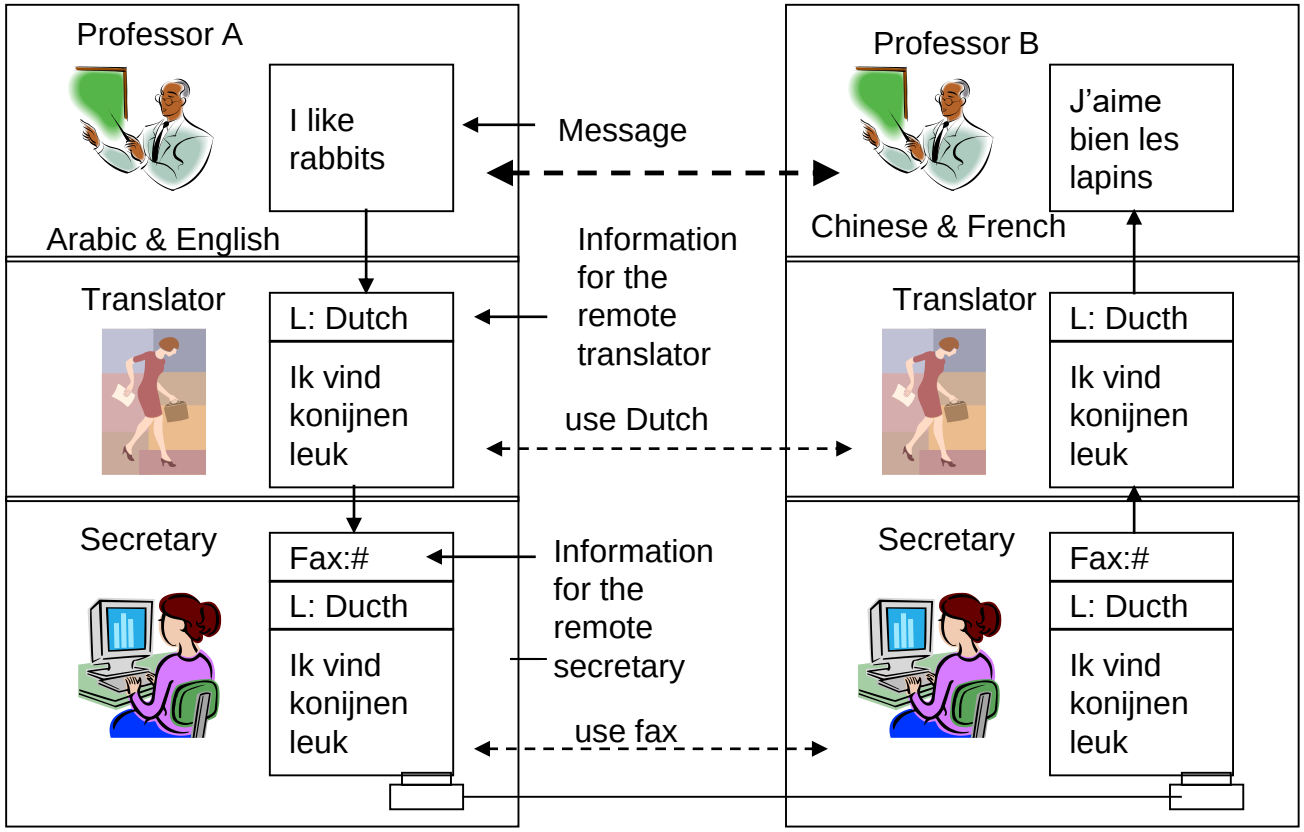
Q: other human protocols?

Why layering?

Dealing with complex systems:

- Explicit structure allows *identification*, relationship of complex system's pieces
- Modularization eases maintenance, updating of system
  - Change of implementation of layer's service transparent to rest of system

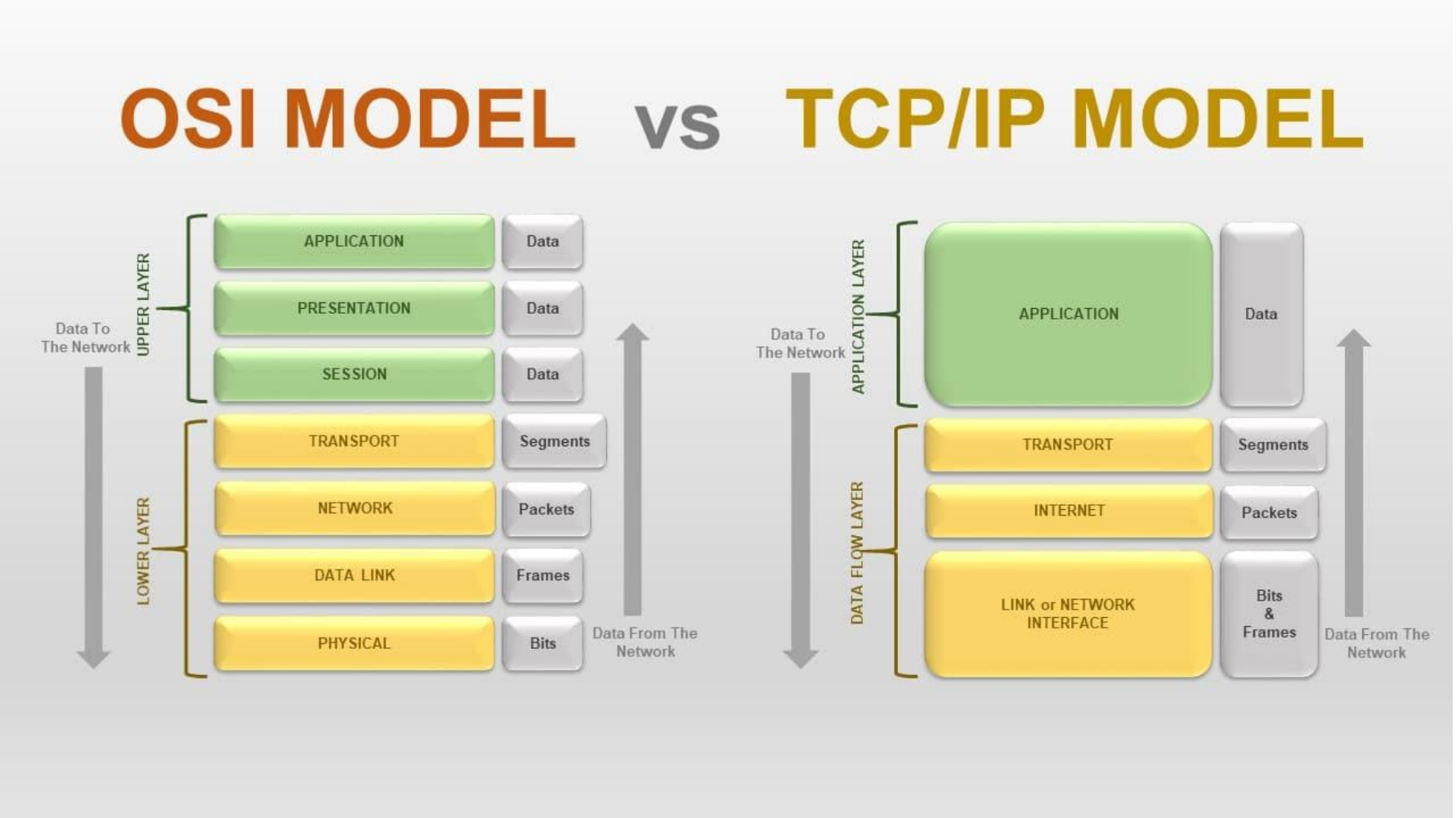
Analogy


Professor A
I like rabbits
Message
Professor B
J'aime bien les lapins
Arabic & English
Chinese & French
Information for the remote translator
Translator
L: Dutch
Ik vind konijnen leuk
use Dutch
Translator
L: Ducth
Ik vind konijnen leuk
Secretary
Fax:#
L: Ducth
Ik vind konijnen leuk
Information for the remote secretary
use fax
Secretary
Fax:#
L: Ducth
Ik vind konijnen leuk

Reference Models

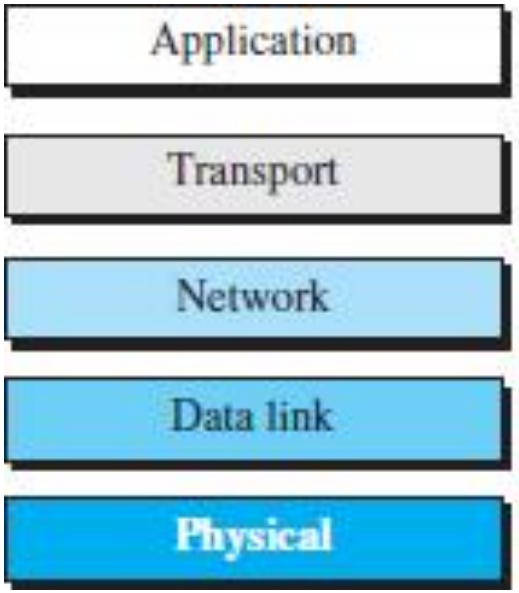
- There are two competing models for how the software is layered: the OSI and the TCP models.
- OSI (Open Systems Interconnection)
  - Developed by ISO (International Standards Organization)
  - 7 layers
- TCP (Transfer Control Protocol)
  - Used in the ARPANET and in the Internet.
  - Common mechanism that is surpassing the OSI Model.
  - 5 layers

Reference Models



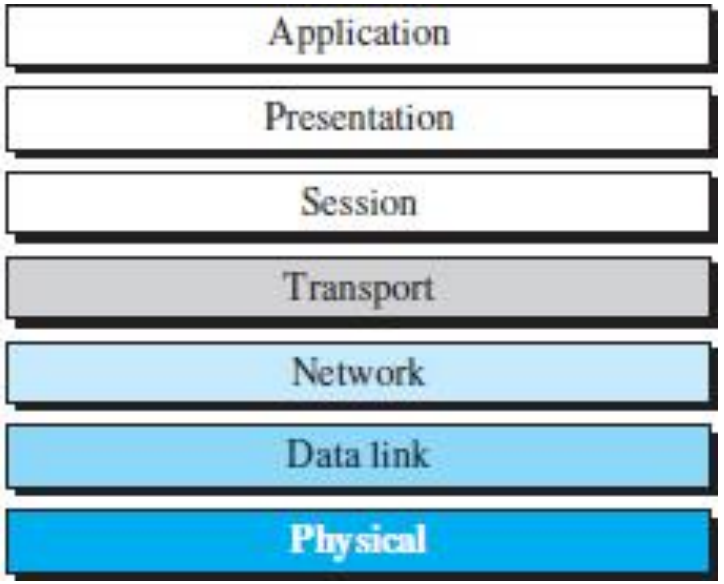
Internet Protocol Stack

- **Application**: supporting network applications
  - FTP, SMTP, HTTP
- **Transport**: process-process data transfer
  - TCP, UDP
- **Network**: routing of datagrams from source to destination
  - IP, routing protocols
- **Link**: data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- **Physical**: bits “on the wire”

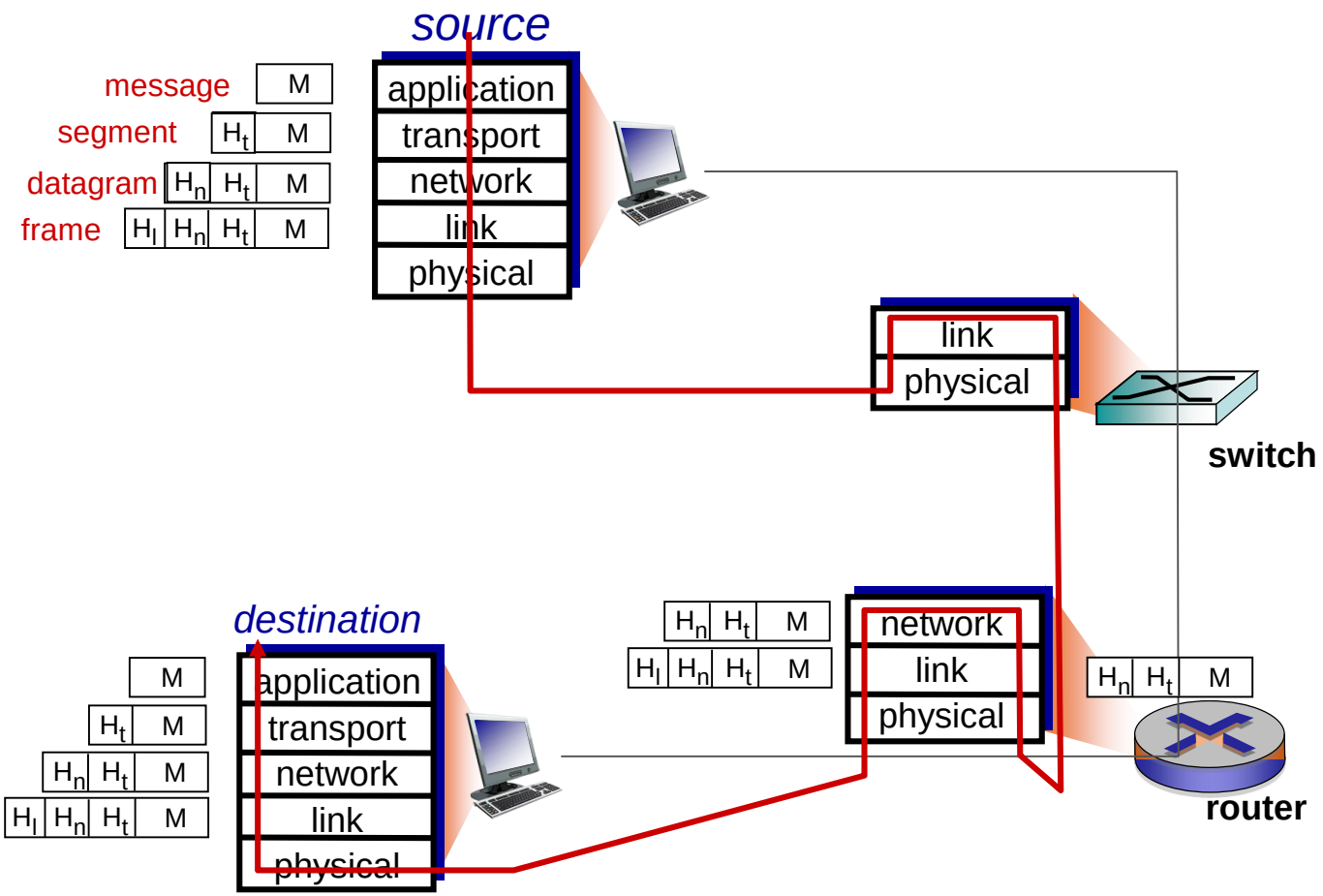


ISO/OSI reference model

- **Presentation**: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- **Session**: *synchronization*, check-pointing, recovery of data exchange
- Internet stack "missing" these layers!
  - these services, if needed, must be implemented in application
  - needed?

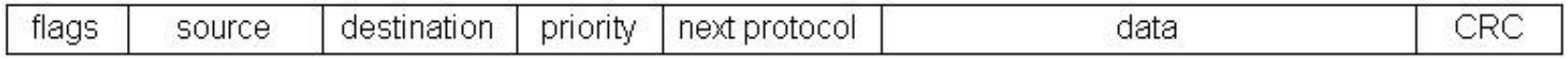


Encapsulation

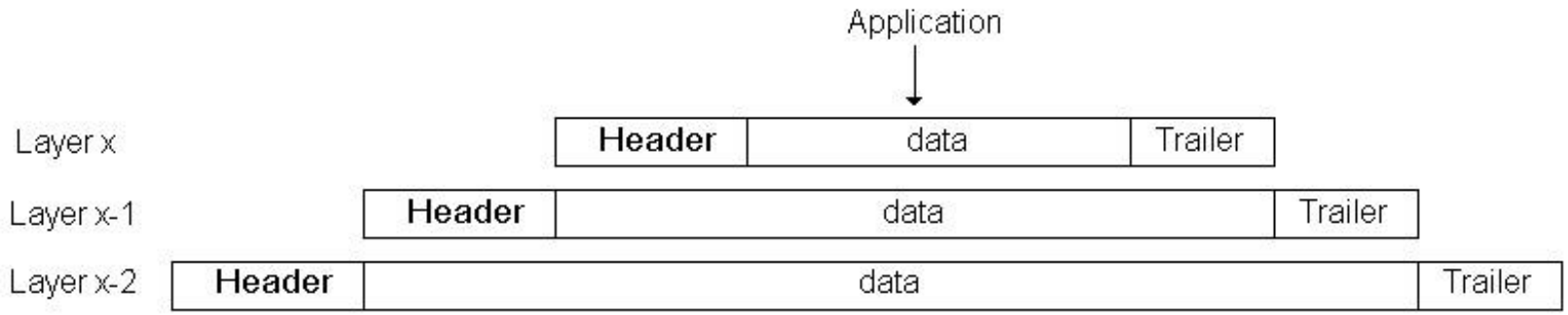


Reference Models

- Headers, Data, and Trailers



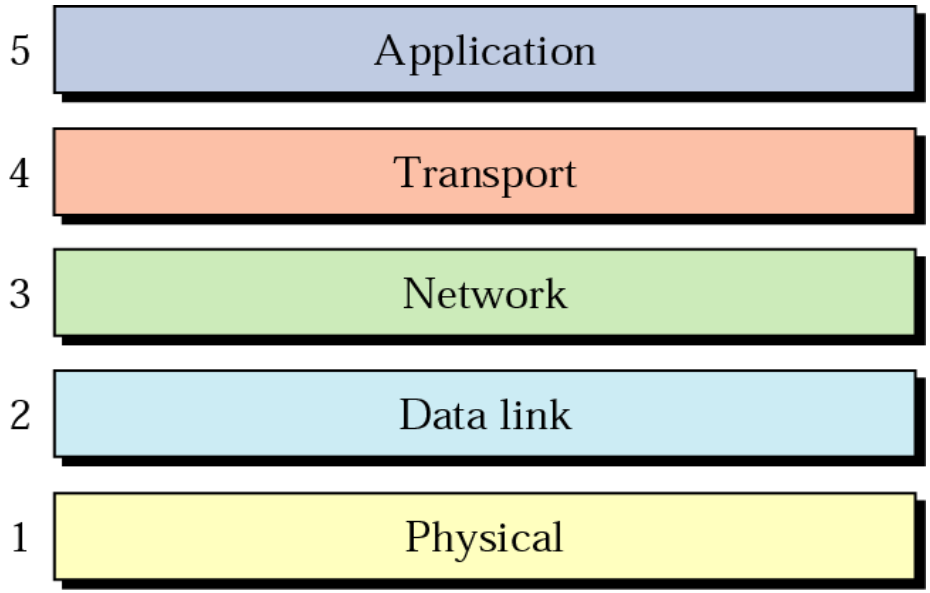
- Encapsulation



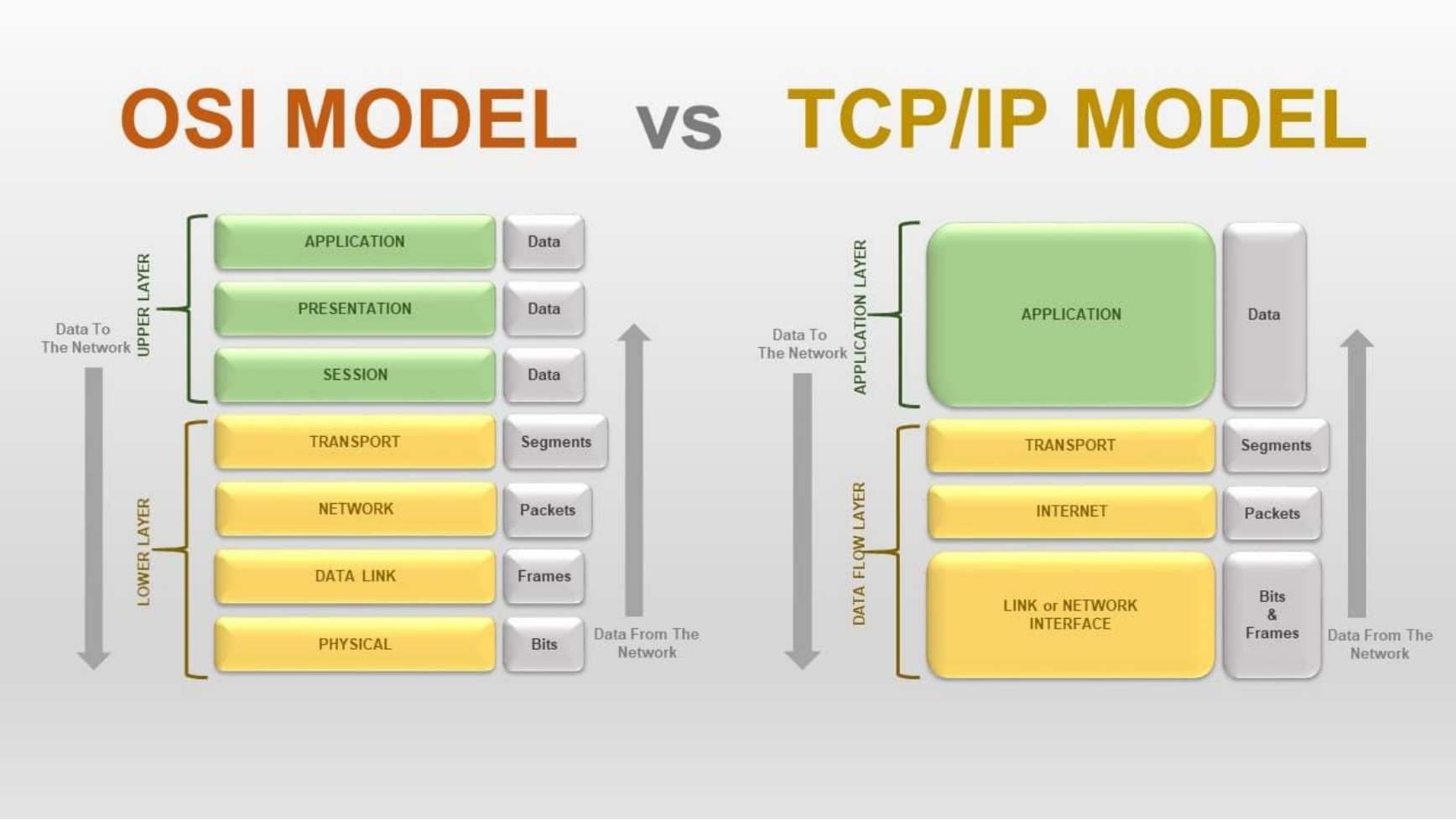
TCP/IP MODEL

Internet Layers

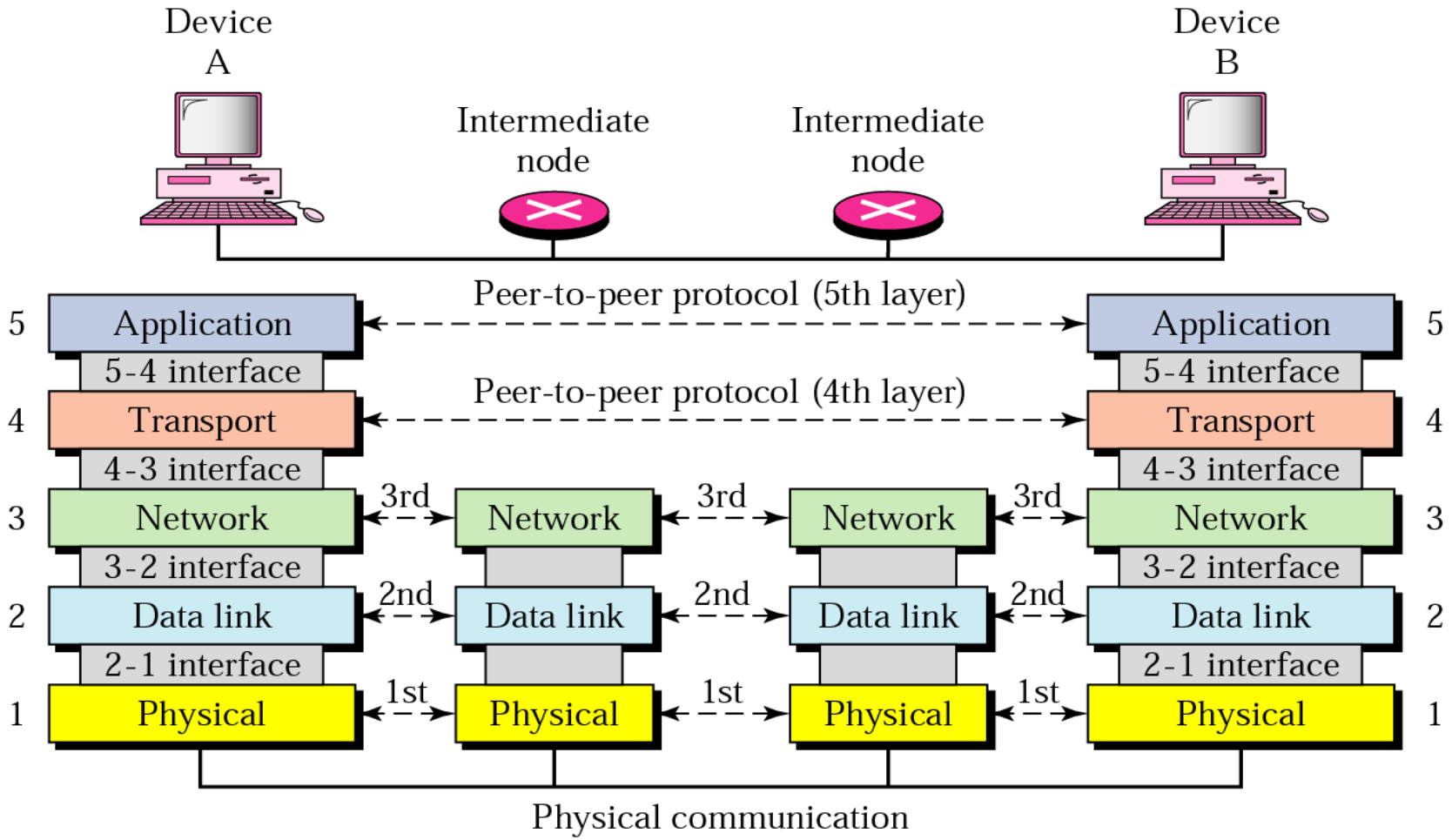
- Internet Model
  - Dominant model in data communications and networking
  - 5 ordered layers; often referred to as TCP/IP protocol suite



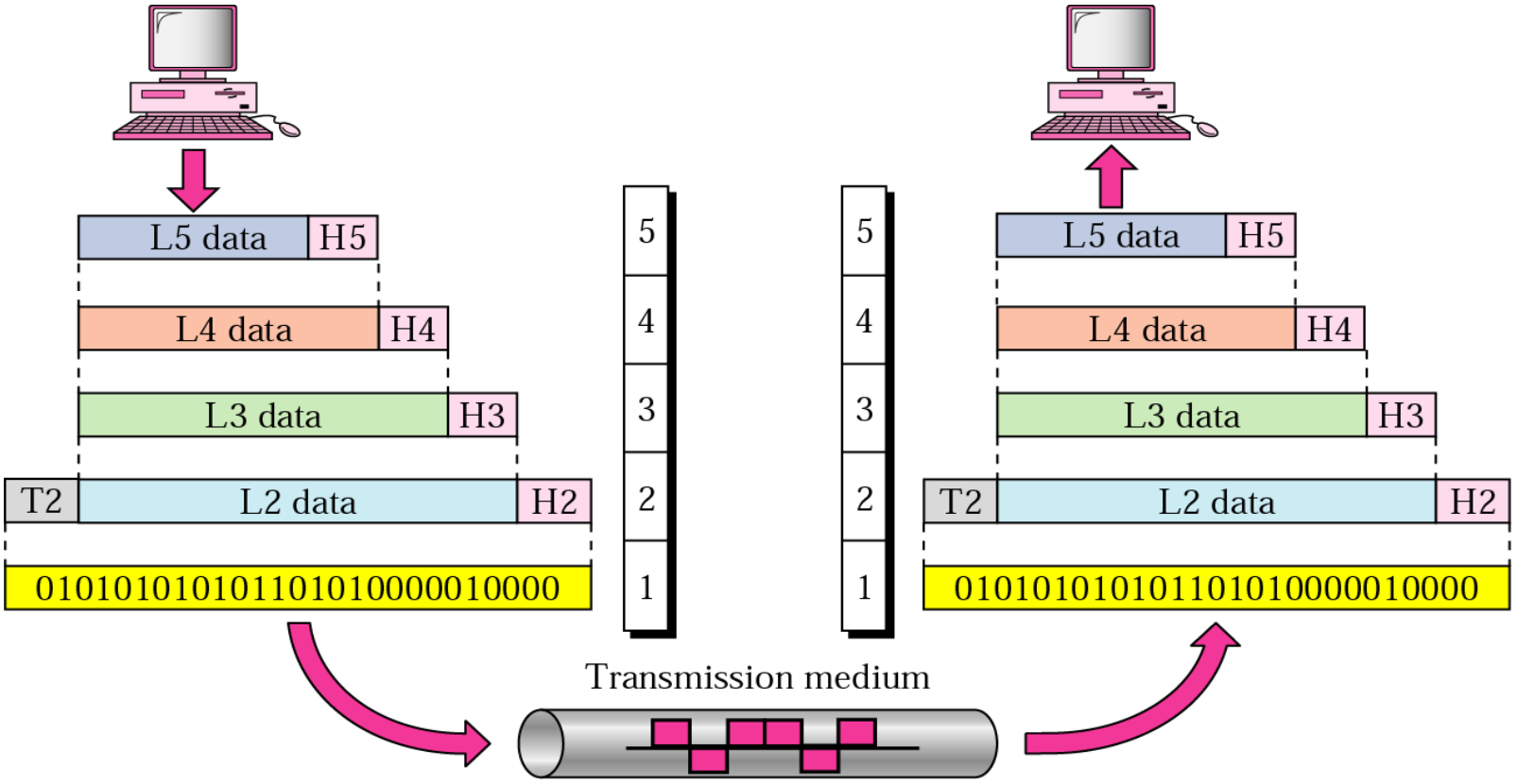
Internet Layers



Peer-to-Peer Processes

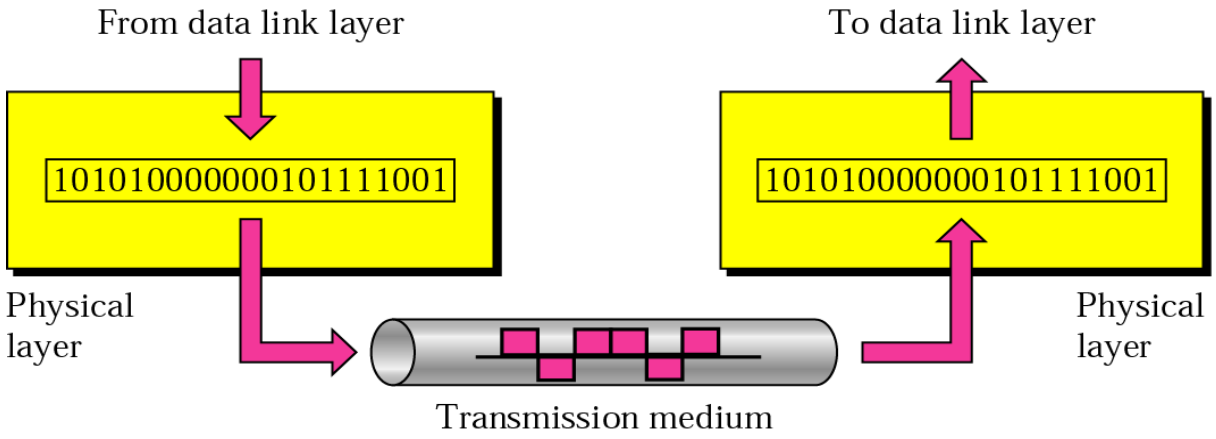


An exchange using the Internet model



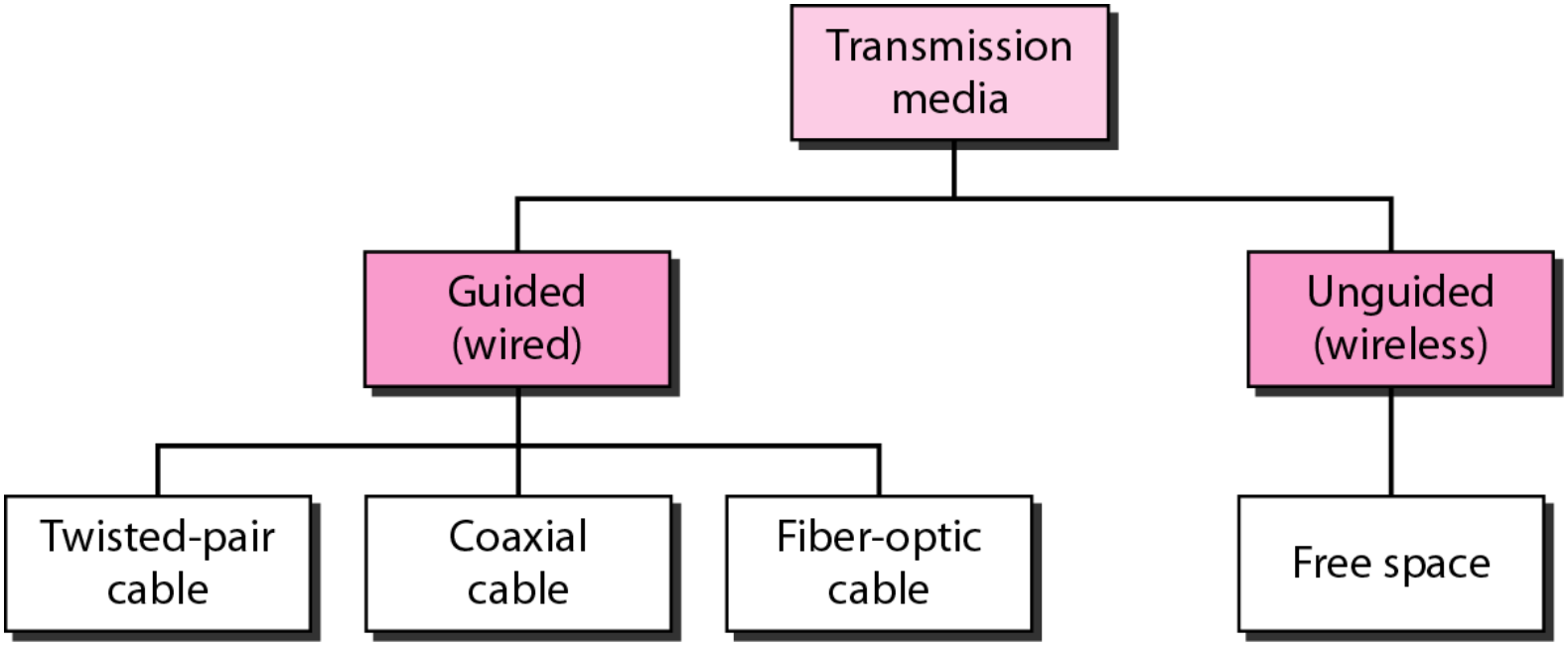
Physical Layer

- Physical characteristics of interfaces and media
  - Representation of bits without interpretation
  - **Data rate**: number of bits per second
  - Synchronization of bits



The physical layer is responsible for transmitting individual bits from one node to the next.

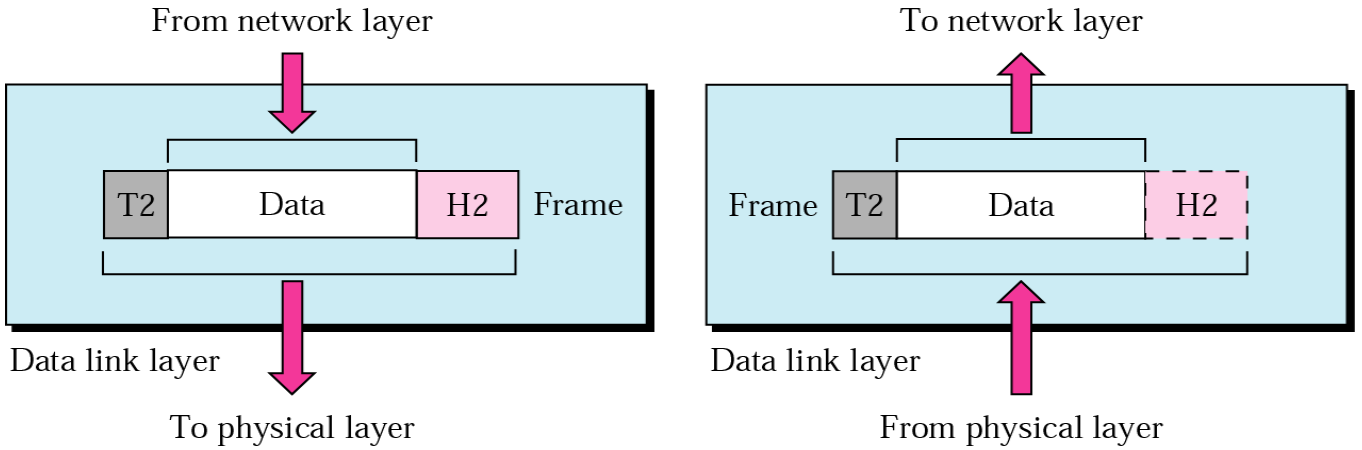
Transmission Media



Data link Layer

Data Link Layer Responsibilities

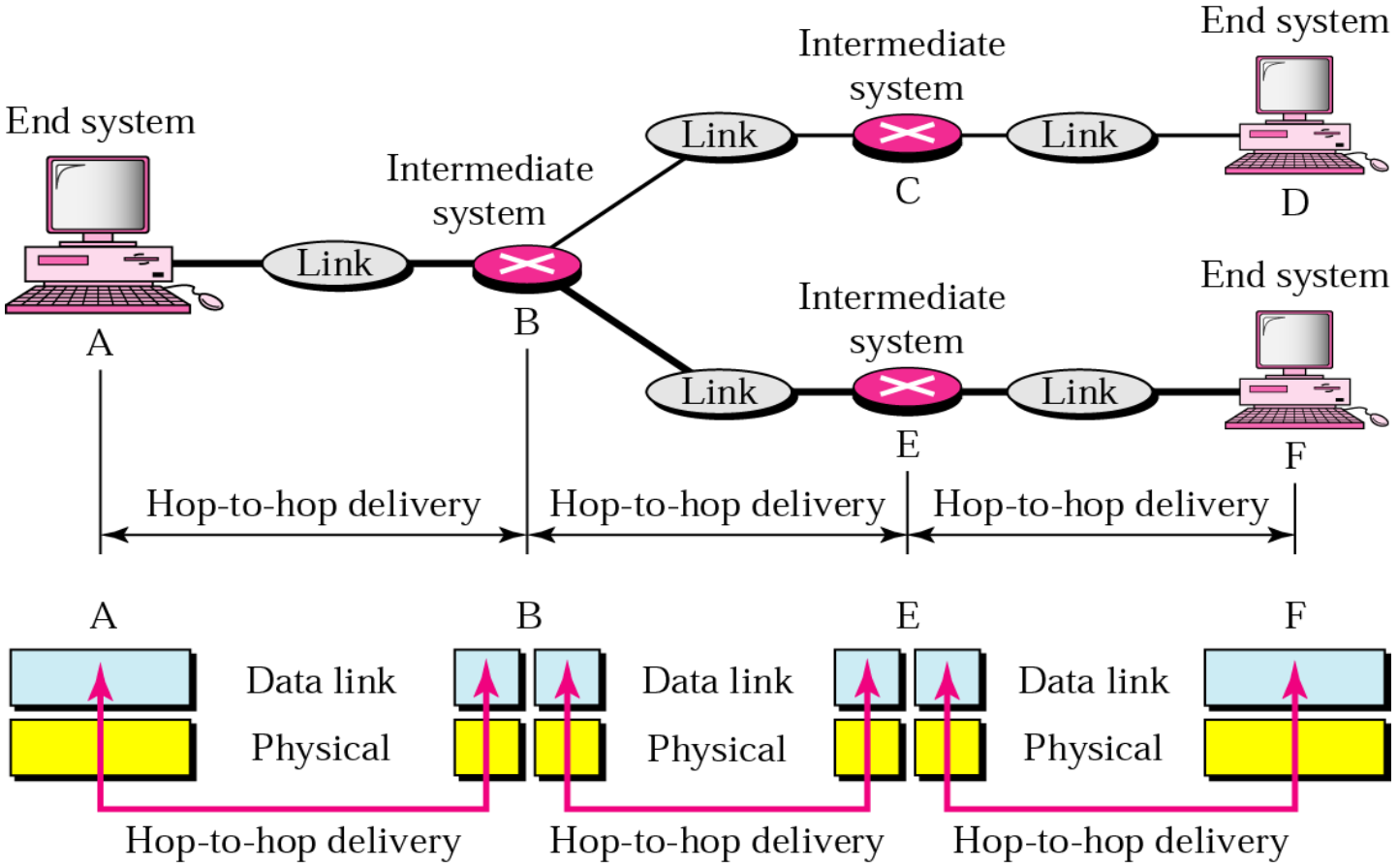
- Defines frames into manageable **data units**
  - Physical addressing
  - Flow control
  - Error control
  - Access control



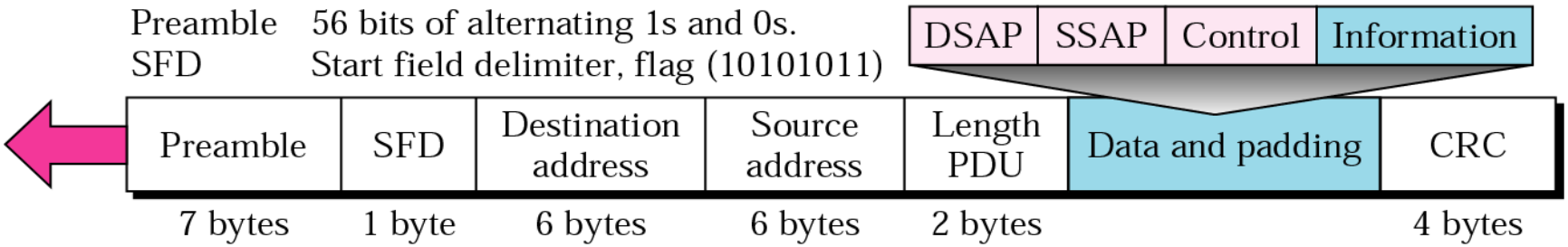
Note:

The data link layer is responsible for transmitting frames from one node to the next.

Node-to-node delivery



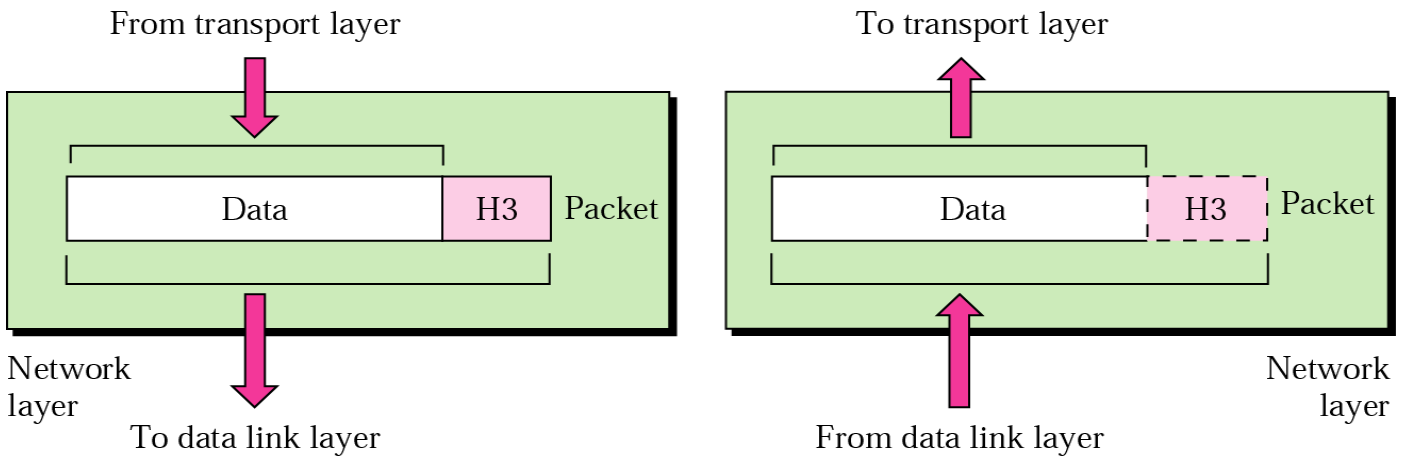
802.3 MAC frame



- Preamble – *7 bytes* of alternating 0s and 1s to alert the receiver and allow it to synchronize
- Start Frame Delimiter (SFD) – *1 byte* – 10101011 signals the beginning of a frame, last chance for synchronization – last 2 bits are 11
- Destination address (DA) – *6 bytes* – contains the physical address of the destination station or stations
- Source address (SA) – *6 bytes* – contains the physical address of the sender
- Length/type – if less than 1518 then it defines the length of the data field – if more than 1536 then it defines the type of the PDU packet that is encapsulated
- Data – data encapsulated from upper-layer protocols : *46 ~ 1500 bytes*
- CRC – CRC-32

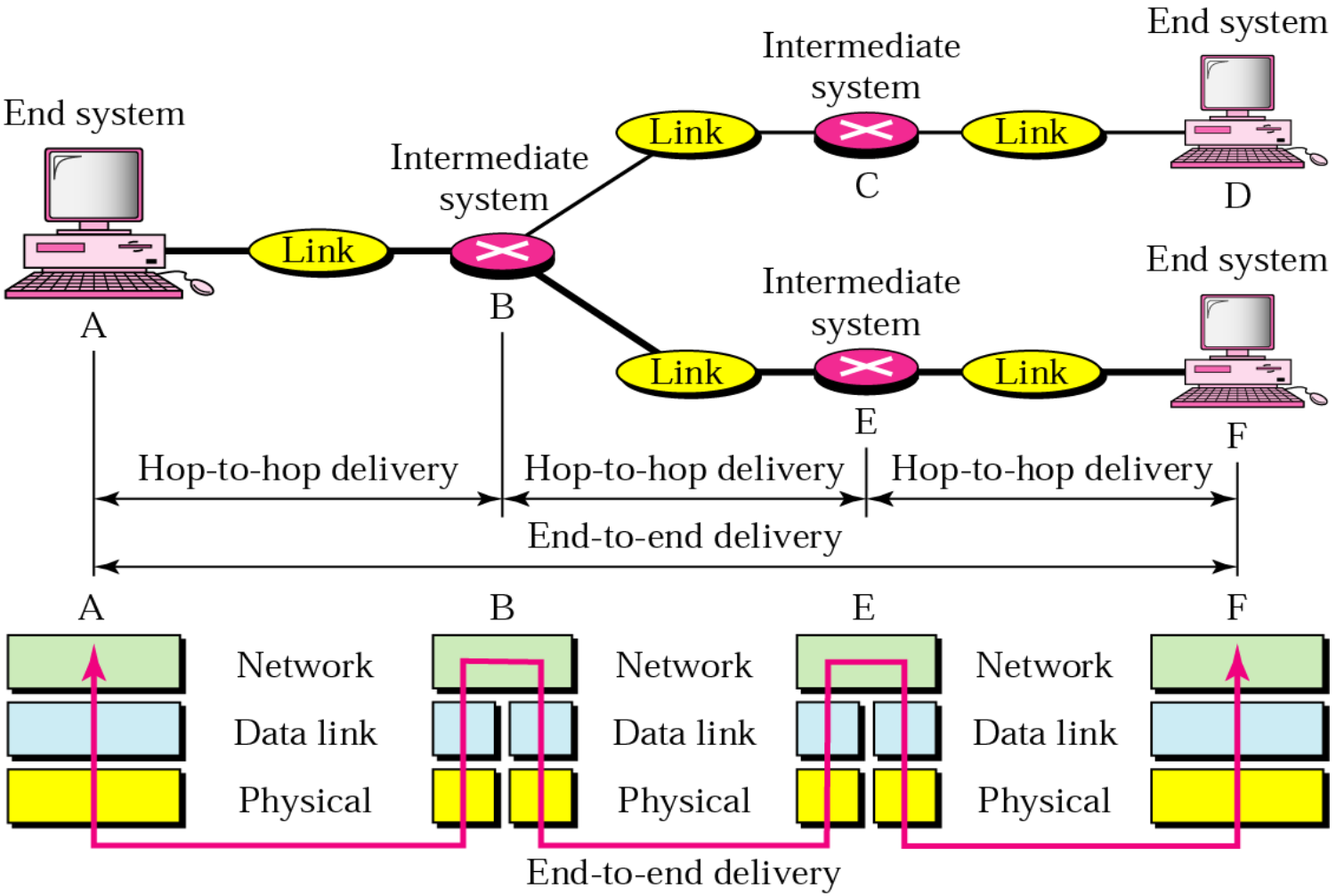
Network layer

- Network Layer Responsibilities
  - Source-to-destination delivery, possibly across multiple networks
  - Logical addressing
  - Routing

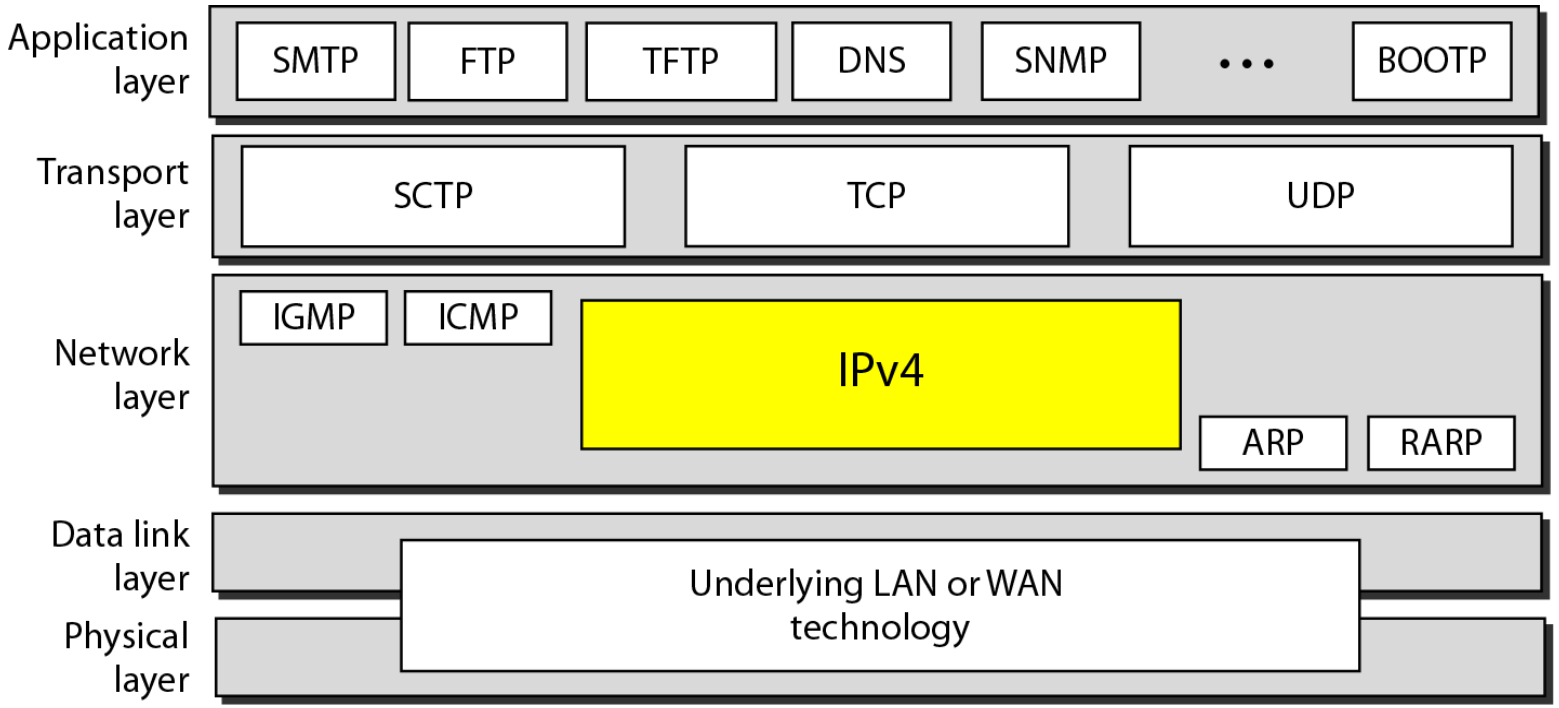


The network layer is responsible for the delivery of packets from the original source to the final destination.

Source-to-Destination delivery



IPv4



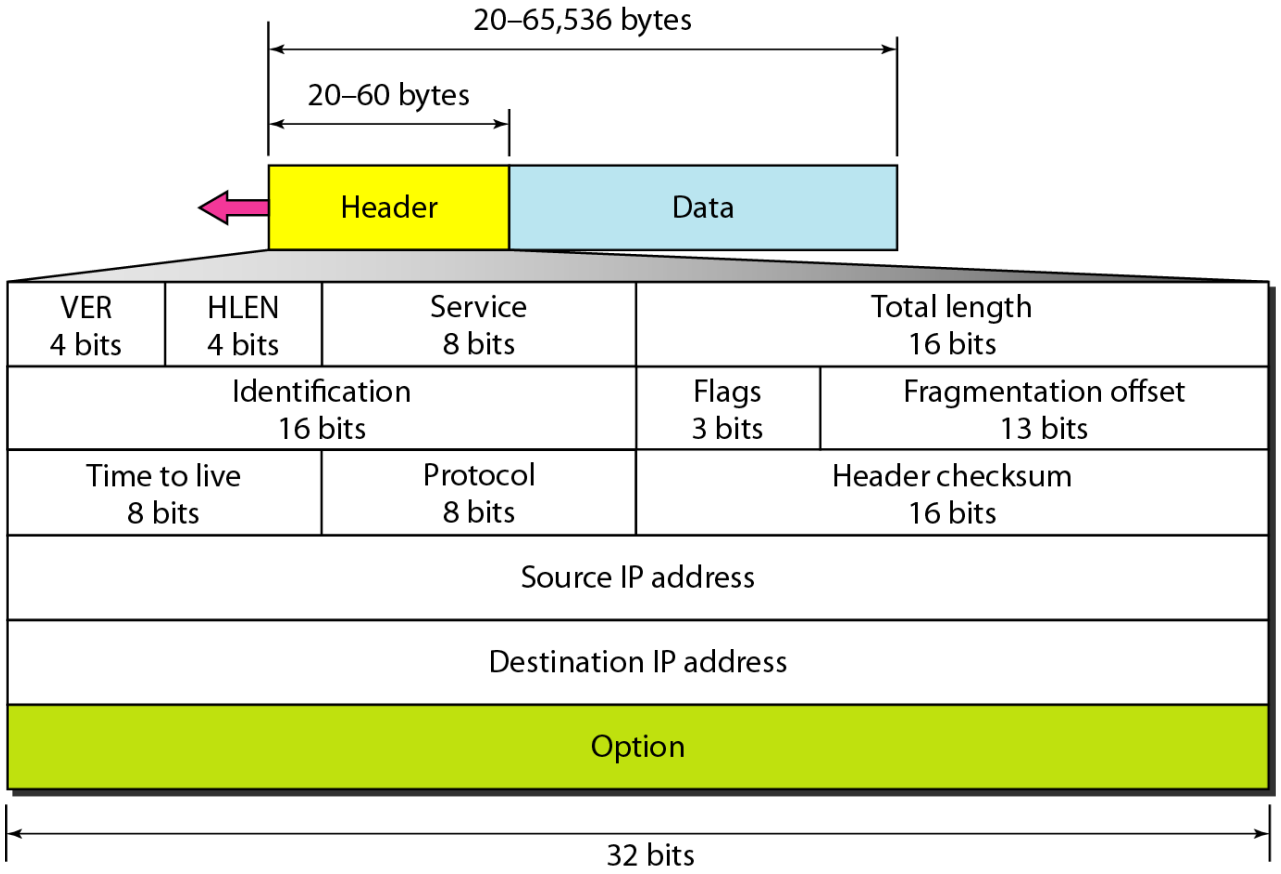
Position of IPv4 in TCP/IP protocol suite

IPv4

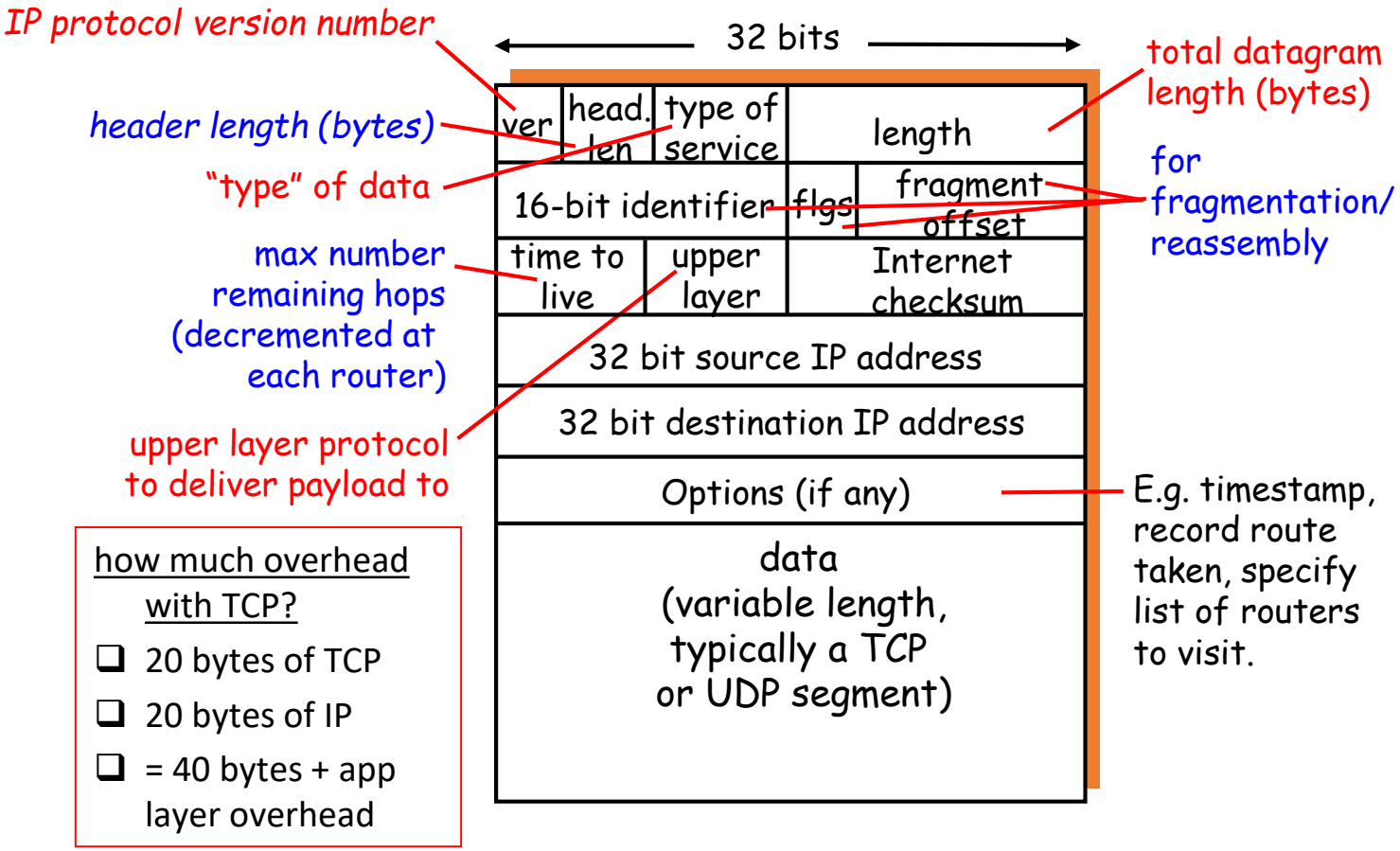
- Best-effort delivery
  - IPv4 is an unreliable and connectionless datagram protocol
  - A best-effort delivery service.
  - best-effort ➔ IPv4 provides no error control or flow control (except for error detection on the header).
- Connectionless protocol
  - Each datagram is handled independently
  - Datagrams sent by the source to the same destination could
    - Arrive out of order
    - be lost or corrupted during transmission.
  - IPv4 relies on a high-level protocol to take of all these problem.

IPv4 Datagrams

- Packets in the IPv4 layer are called Datagrams.



IP datagram format

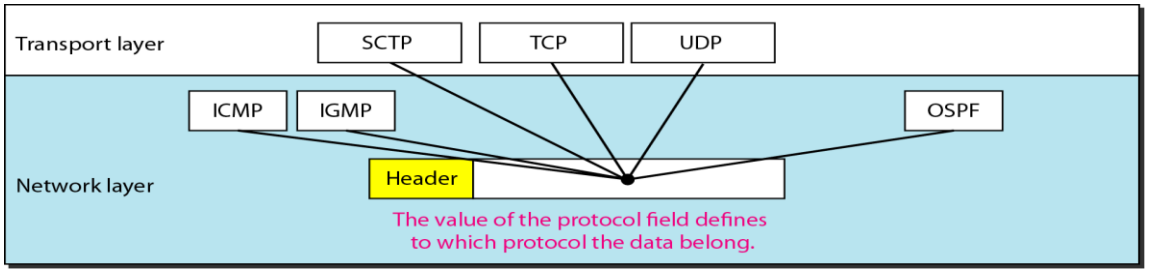


IPv4 Datagram (cont'd)

- A datagram is a variable-length packet consisting of a header and data.
- Header
  - ⊙ Length : 20 – 60 bytes
  - ⊙Contains information essential to routing and delivery.
- Version (VER) : It defines the Version of IPv4. it is 4.
- Header Length (HLEN) : Defining the total length of the datagram header in 4-byte words.

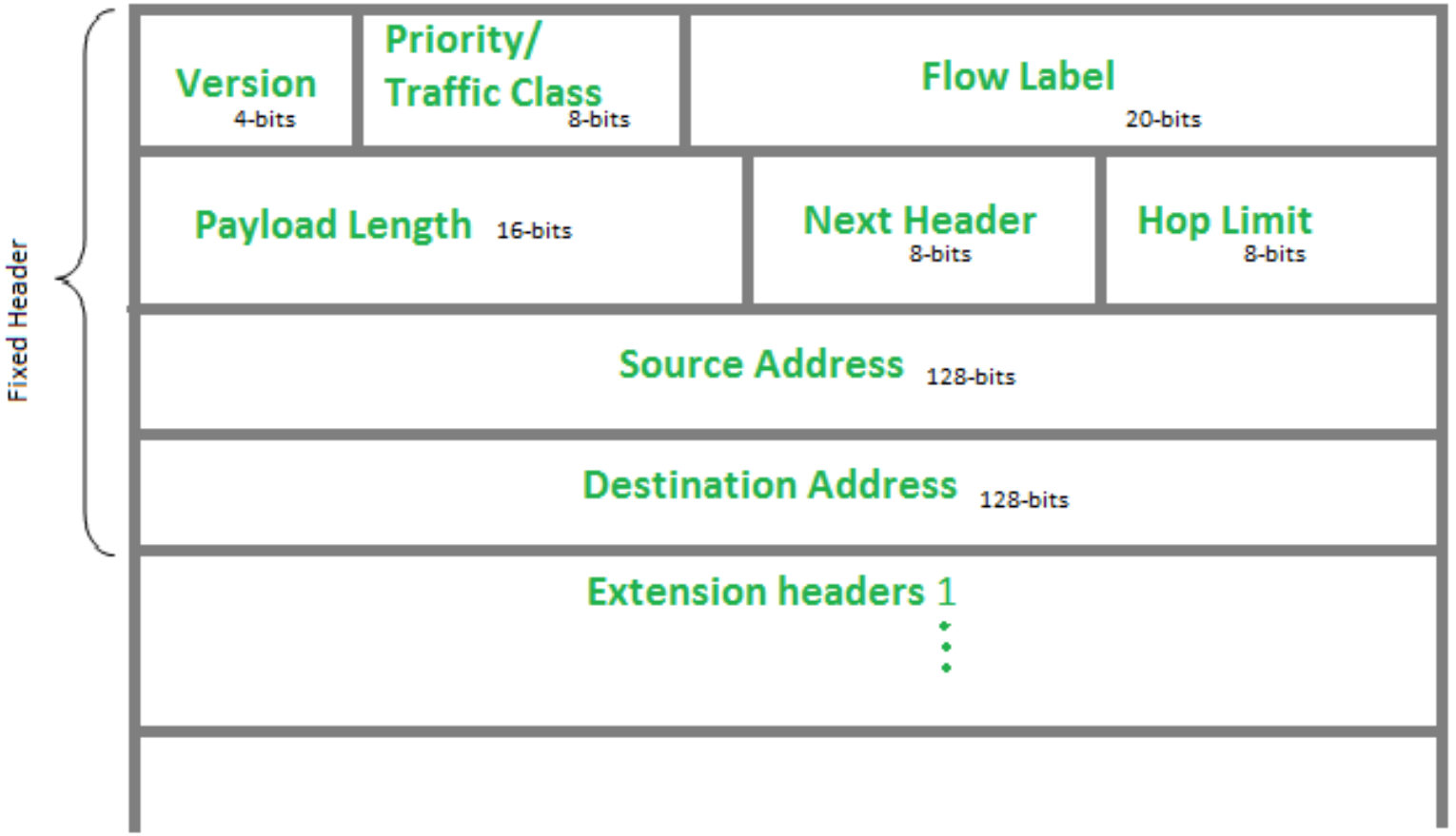
IPv4 Datagram (cont'd)

- Protocol
  - ⊙ Defining the higher level protocol that uses the services of the IP layer
    - TCP, UDP, ICMP, and IGMP
    - Multiplexing data from different higher level protocols



Value	Protocol
1	ICMP
2	IGMP
6	TCP
8	EGP
17	UDP
89	OSPF

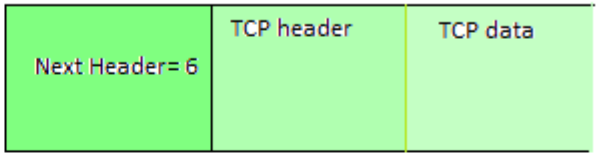
IPv6 Datagram



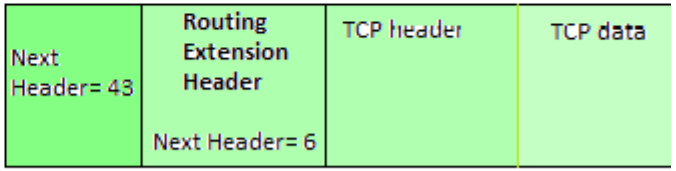
IPv6 Datagram

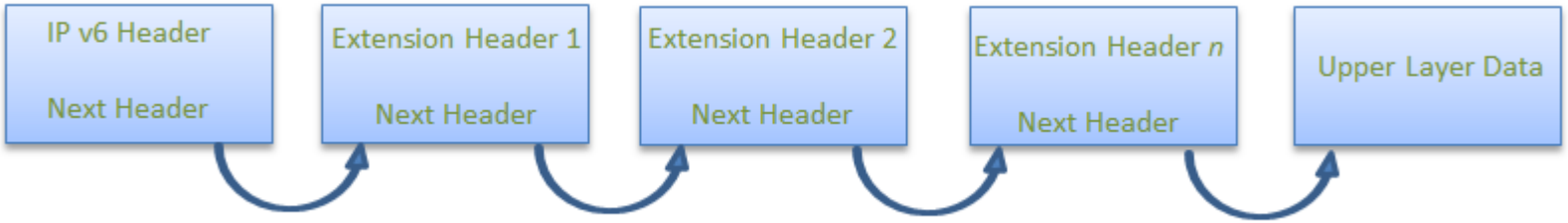
Order	Header Type	Next Header Code
1	Basic IPv6 Header	-
2	Hop-by-Hop Options	0
3	Destination Options (with Routing Options)	60
4	Routing Header	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulation Security Payload Header	50
8	Destination Options	60
9	Mobility Header	135
	No next header	59
Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58

Example: *TCP is used in IPv6 packet*



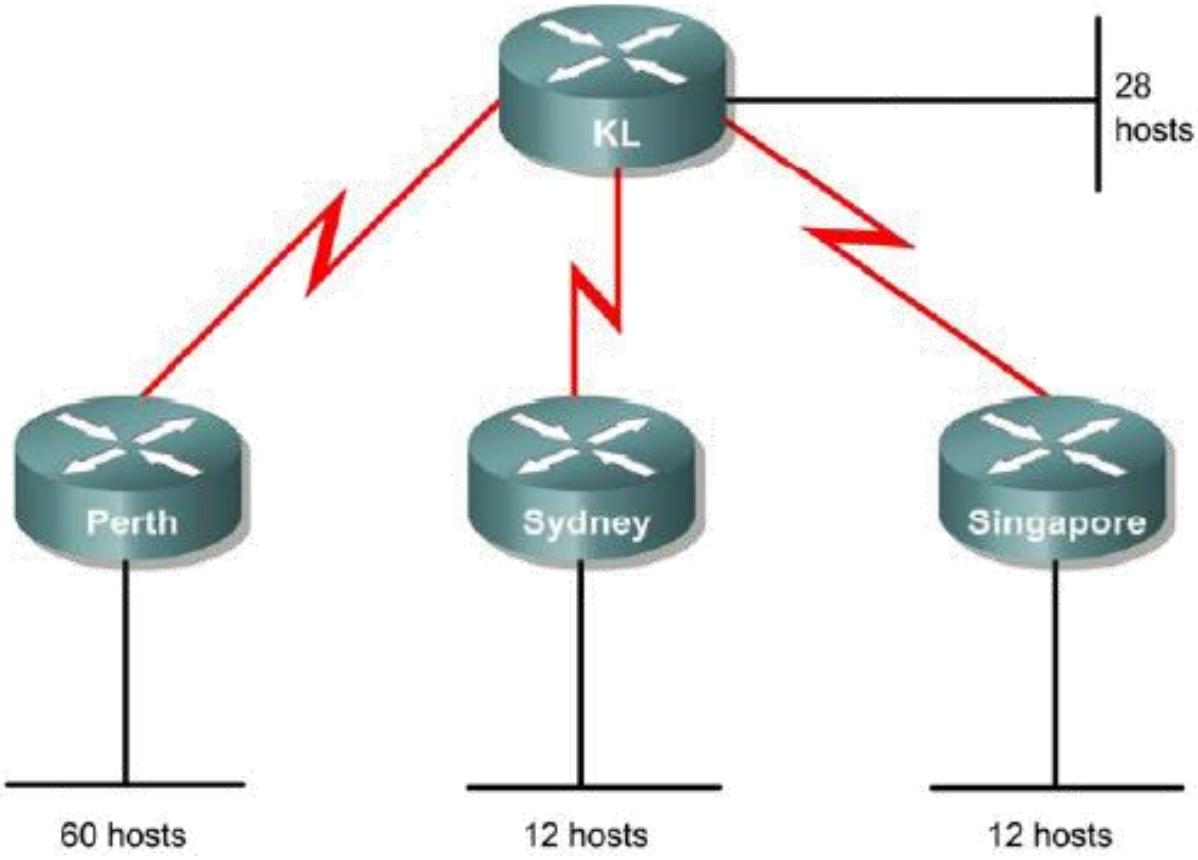
Example2:



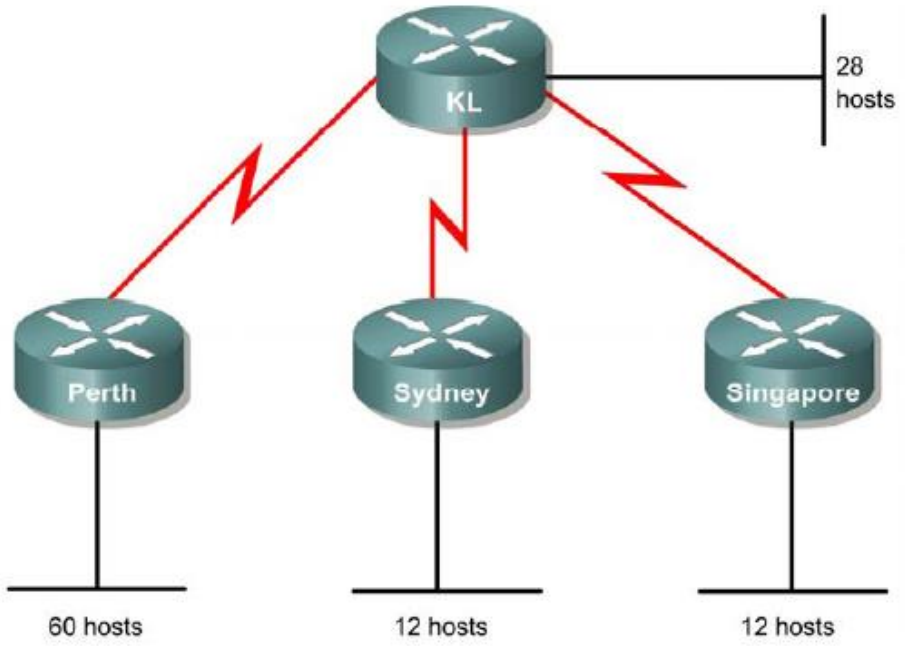


Subnet with VLSM

192.168.10.0/24



Regular Subnet

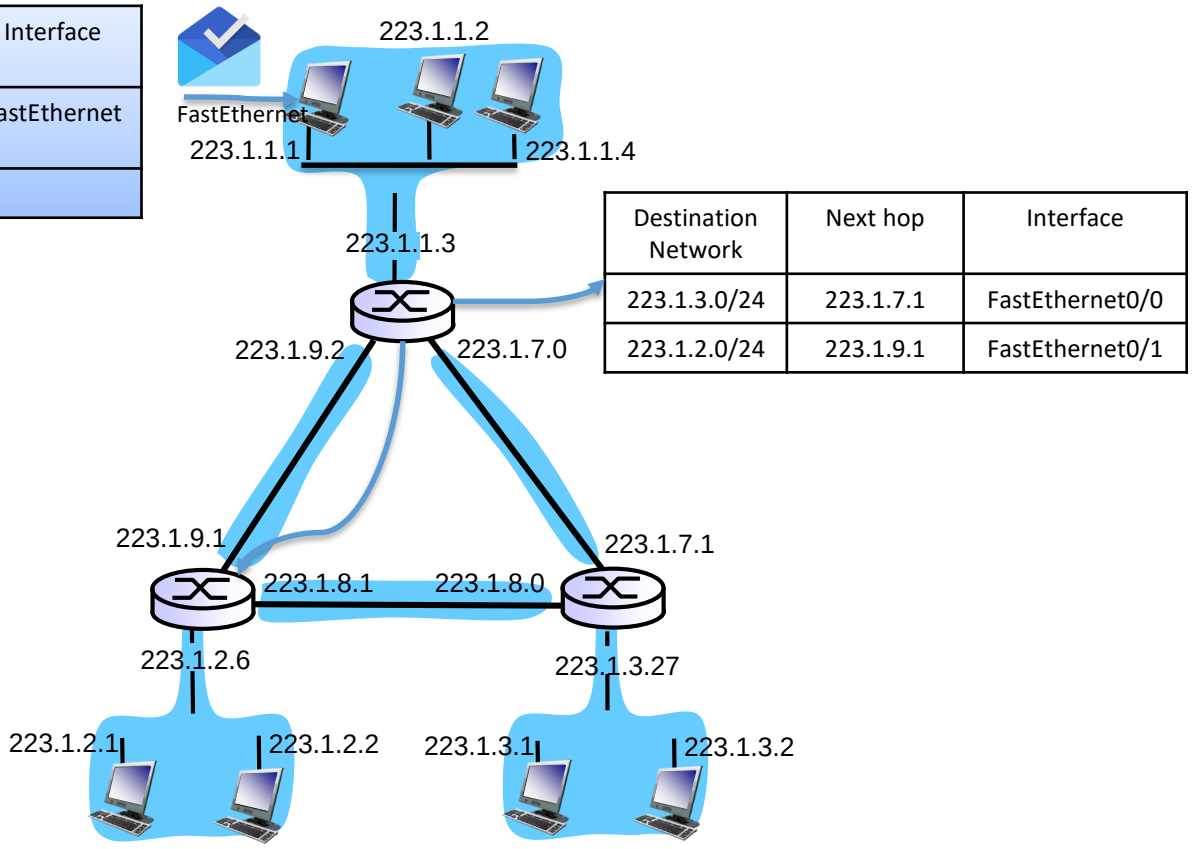


192.168.10.0/24

7 subnets; The largest subnet needs **60** hosts
If 3 bits for subnet (8 subnets) ➔ 5 bits for host (32 hosts)
If 6 bits for host (64 hosts) ➔ 2 bits for subnet (4 subnets)

Forwarding - Routing

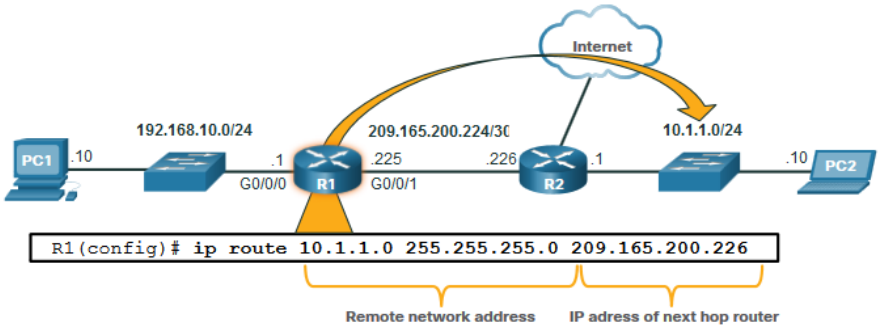
Destination Network	Next hop	Interface
0.0.0.0	223.1.1.3	FastEthernet



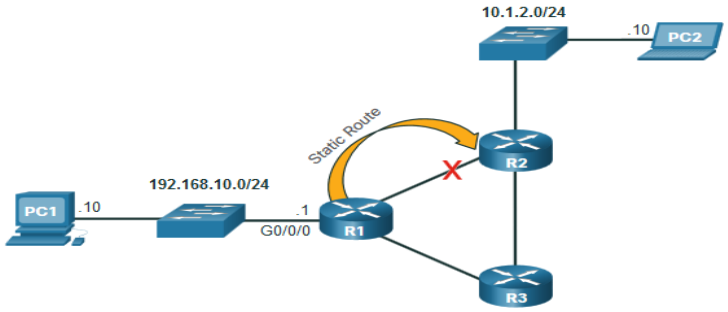
Destination Network	Next hop	Interface
223.1.3.0/24	223.1.7.1	FastEthernet0/0
223.1.2.0/24	223.1.9.1	FastEthernet0/1

Introduction to Routing
Static Routing

- Static Route Characteristics:
  - ⊙ Must be configured manually
  - ⊙ Must be adjusted manually by the administrator when there is a change in the topology
  - ⊙ Good for small non-redundant networks
  - ⊙ **Often used in conjunction with a dynamic routing protocol** for configuring a default route



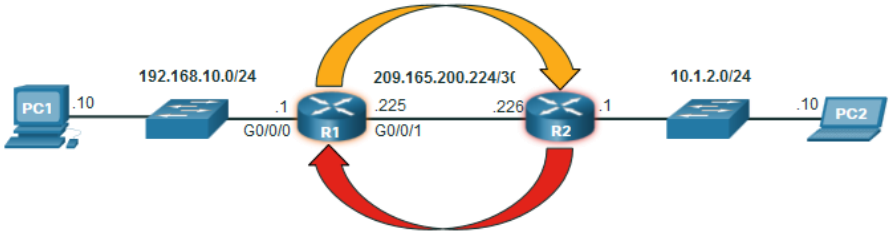
R1 is manually configured with a static route to reach the 10.1.1.0/24 network. If this path changes, R1 will require a new static route.



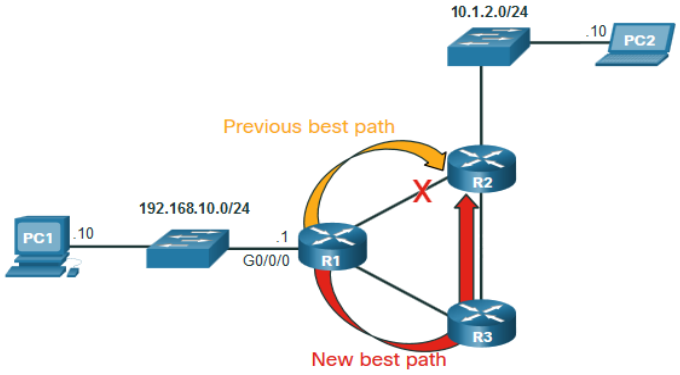
If the route from R1 via R2 is no longer available, a new static route via R3 would need to be configured. A static route does not automatically adjust for topology changes.

Introduction to Routing
Dynamic Routing

- Dynamic Routes Automatically:
  - Discover remote networks
  - Maintain up-to-date information
  - **Choose the best path** to the destination
  - **Find new best paths** when there is a topology change
  - Dynamic routing can also share static default routes with the other routers.



- R1 is using the routing protocol OSPF to let R2 know about the 192.168.10.0/24 network.
- R2 is using the routing protocol OSPF to let R1 know about the 10.1.1.0/24 network.



R1, R2, and R3 are using the dynamic routing protocol OSPF. If there is a network topology change, they can automatically adjust to find a new best path.

Address Resolution - ARP

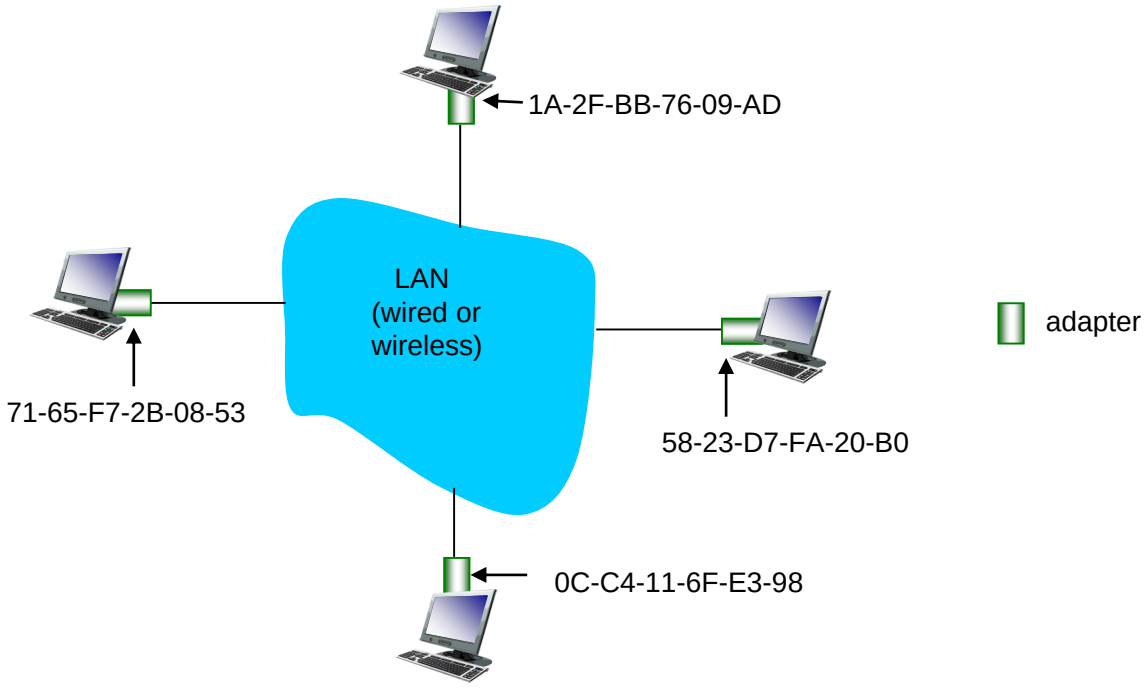
MAC addresses and ARP

- IP address:
  - "Low" Network-layer address for interface
  - Used for layer 3 (network layer) forwarding
- MAC (or LAN or physical or Ethernet) address:
  - Function: used "locally" to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)
  - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each "numeral" represents 4 bits)

LAN addresses and ARP

each adapter on LAN has unique *LAN* address



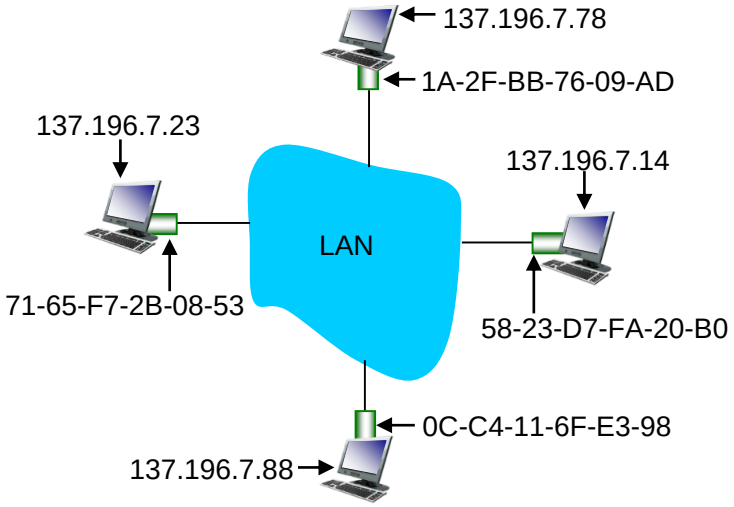
LAN addresses (more)

- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address → portability
  - can move LAN card from one LAN to another
- IP hierarchical address not portable
  - address depends on IP subnet to which node is attached

ARP: Address Resolution Protocol

- ARP table: each IP node (host, router) on LAN has a ARP table
  - IP/MAC address mappings for some LAN nodes:
- < IP address; MAC address; TTL>
  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

Question: how to determine interface's MAC address, knowing its IP address?



IP address	MAC address	TTL
137.196.7.14	58-23-D7-FA-20-B0	20

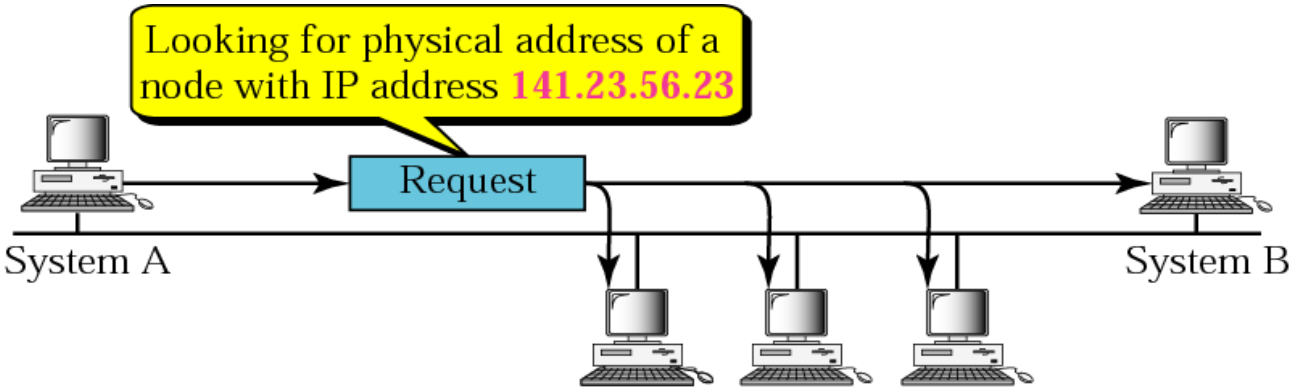
Use arp command to display ARP table on Windows or Linux

ARP protocol: same LAN

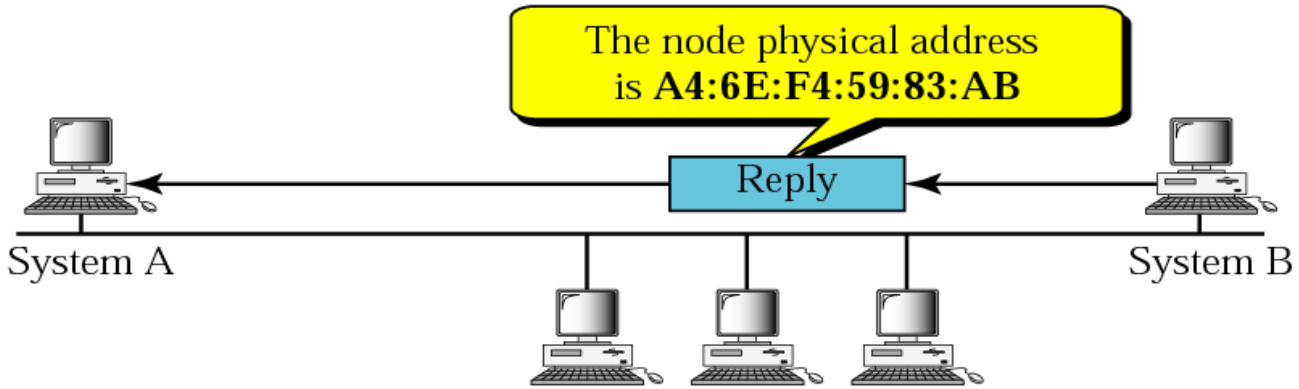
1. **A wants to send datagram to B**
   - B's MAC address not in A's ARP table.
2. **A broadcasts ARP query packet**, containing B's IP address
   - Destination MAC address = FF-FF-FF-FF-FF-FF
   - All nodes on LAN receive ARP query
3. B receives ARP packet, replies to A with its (B's) MAC address
   - frame sent to A's MAC address (unicast)
4. A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
   - soft state: information that times out (goes away) unless refreshed

- ARP is "plug-and-play":
  - Nodes create their ARP tables without intervention from net administrator

ARP operation



a. ARP request is broadcast



b. ARP reply is unicast

ARP packet

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

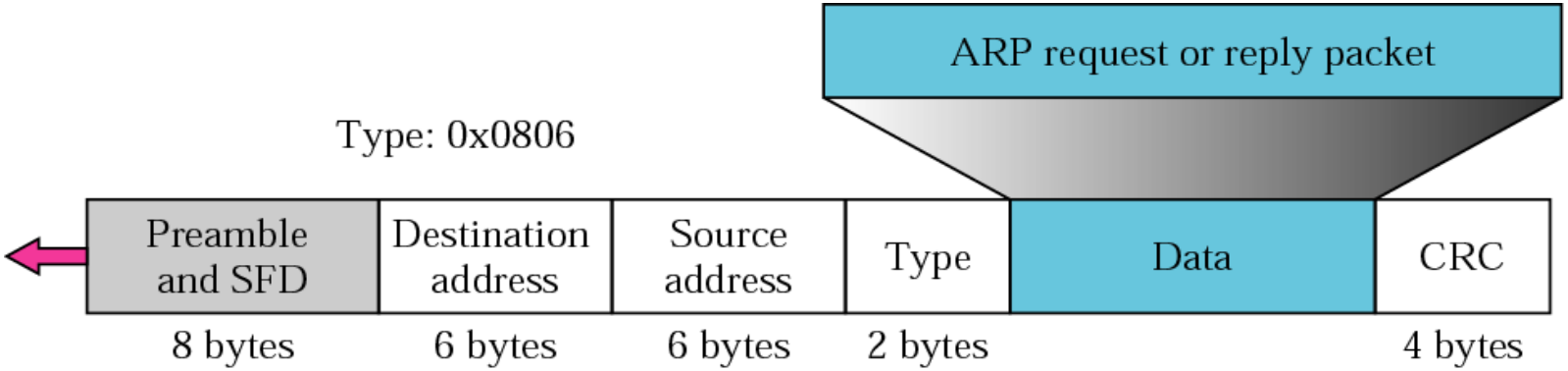
Hardware Type – Ethernet is type 1

Protocol Type- IPv4=x0800

Hardware Length: length of Ethernet Address (6)

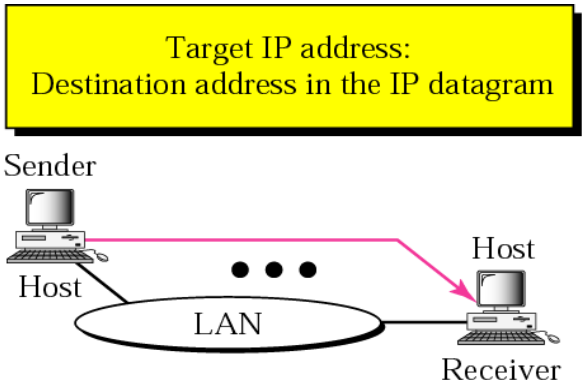
Protocol Length: length of IPv4 address (4)

Encapsulation of ARP packet

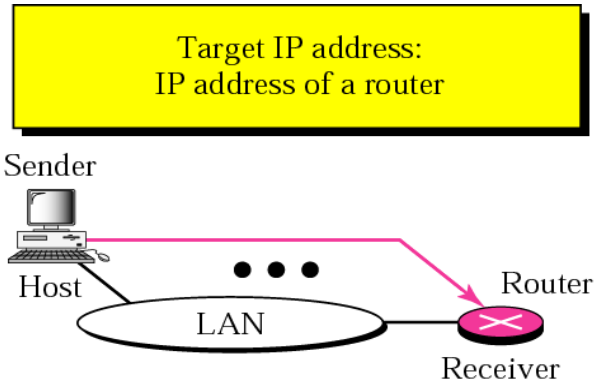


The ARP packet is encapsulated within an Ethernet packet.
Note: Type field for Ethernet is x0806

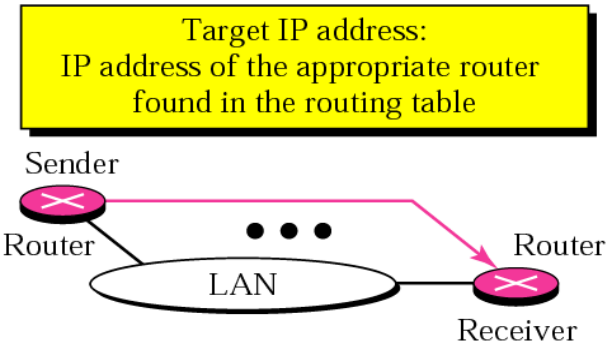
Four cases using ARP



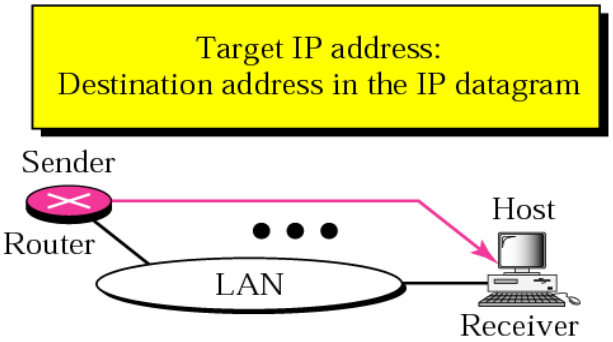
Case 1. A host has a packet to send to another host on the same network.



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

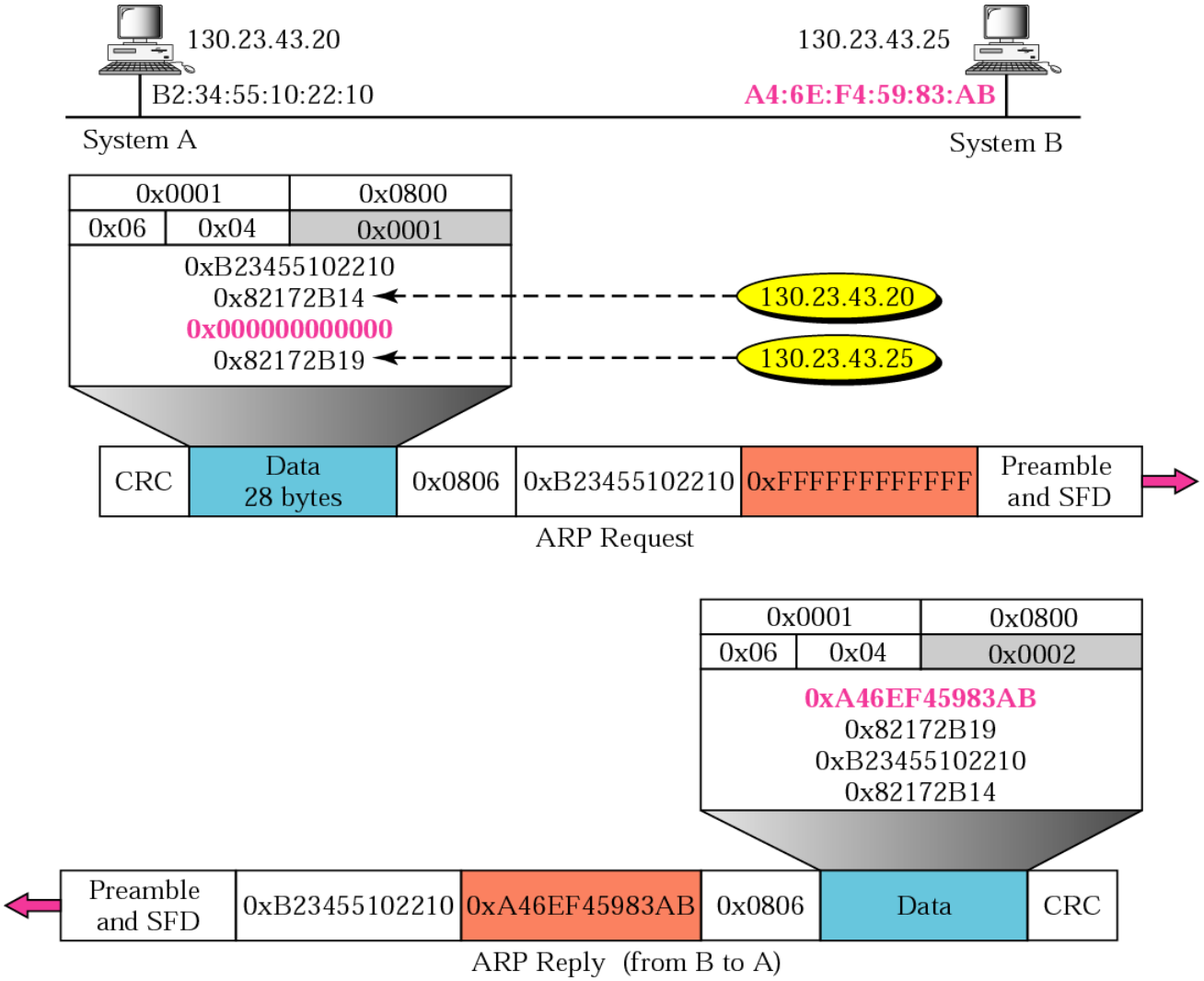


Case 3. A router receives a packet to be sent to a host on another network.
It must first be delivered to the appropriate router.

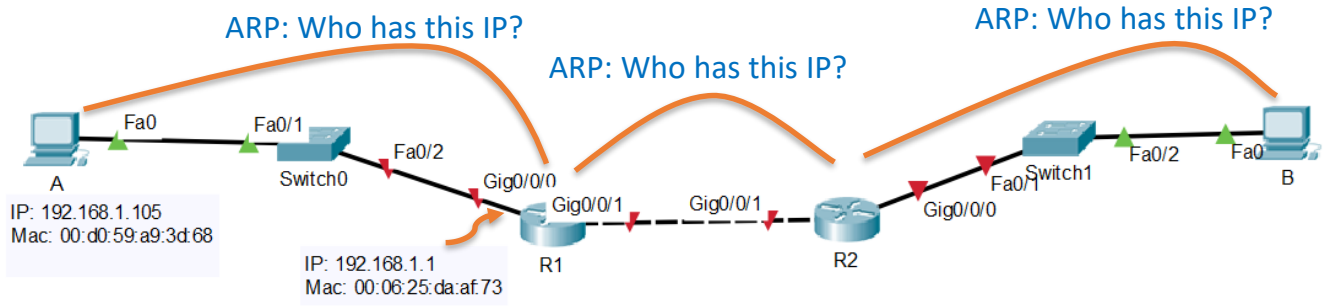


Case 4. A router receives a packet to be sent to a host on the same network.

ARP Example


130.23.43.20
B2:34:55:10:22:10
System A
130.23.43.25
A4:6E:F4:59:83:AB
System B
0x0001
0x0800
0x06
0x04
0x0001
0xB23455102210
0x82172B14
0x000000000000
0x82172B19
130.23.43.20
130.23.43.25
CRC
Data
28 bytes
0x0806
0xB23455102210
0xFFFFFFFFFFFF
Preamble
and SFD
ARP Request
0x0001
0x0800
0x06
0x04
0x0002
0xA46EF45983AB
0x82172B19
0xB23455102210
0x82172B14
Preamble
and SFD
0xB23455102210
0xA46EF45983AB
0x0806
Data
CRC
ARP Reply (from B to A)

ARP – A each step



ethernet-ethereal-trace-1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	AmbitMic_a9:3…	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.105
2	0.001018	LinksysG_da:a…	AmbitMic_a9:3d…	ARP	60	192.168.1.1 is at 00:06:25:da:af:73
6	13.542…	CnetTech_73:8…	Broadcast	ARP	60	Who has 192.168.1.117? Tell 192.168.1.104

```
› Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
▾ Ethernet II, Src: AmbitMic_a9:3d:68 (00:d0:59:a9:3d:68), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  › Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  › Source: AmbitMic_a9:3d:68 (00:d0:59:a9:3d:68)
    Type: ARP (0x0806)
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: AmbitMic_a9:3d:68 (00:d0:59:a9:3d:68)
    Sender IP address: 192.168.1.105
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.1
```

ethernet-ethereal-trace-1

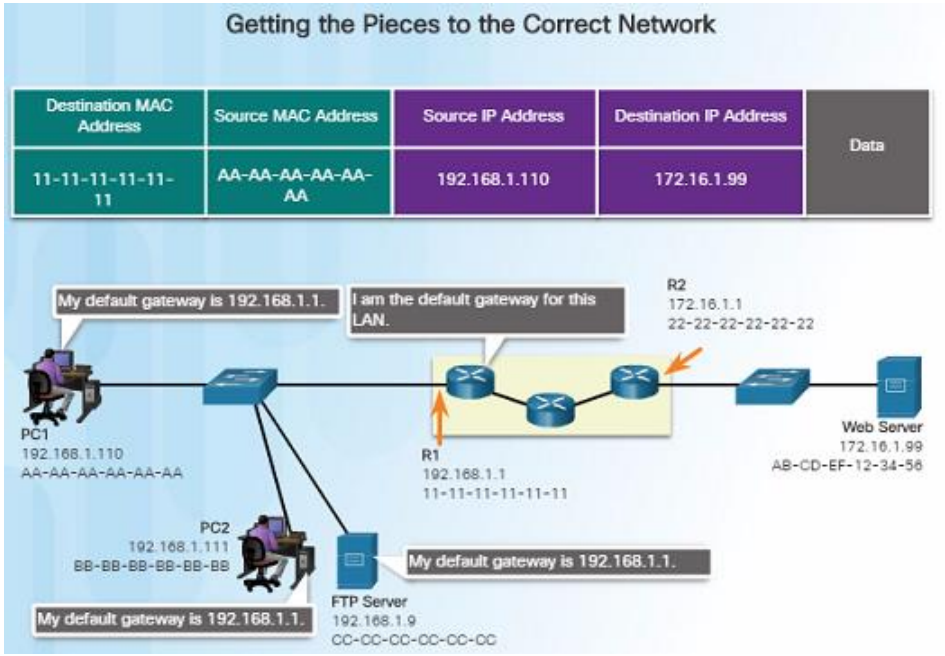
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	AmbitMic_a9:3…	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.105
2	0.001018	LinksysG_da:a…	AmbitMic_a9:3d…	ARP	60	192.168.1.1 is at 00:06:25:da:af:73
6	13.542…	CnetTech_73:8…	Broadcast	ARP	60	Who has 192.168.1.117? Tell 192.168.1.104

```
› Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
▾ Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: AmbitMic_a9:3d:68 (00:d0:59:a9:3d:68)
  › Destination: AmbitMic_a9:3d:68 (00:d0:59:a9:3d:68)
  › Source: LinksysG_da:af:73 (00:06:25:da:af:73)
    Type: ARP (0x0806)
    Padding: 00000000000000000000000000000000000000000000
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: LinksysG_da:af:73 (00:06:25:da:af:73)
    Sender IP address: 192.168.1.1
    Target MAC address: AmbitMic_a9:3d:68 (00:d0:59:a9:3d:68)
    Target IP address: 192.168.1.105
```

Default Gateways

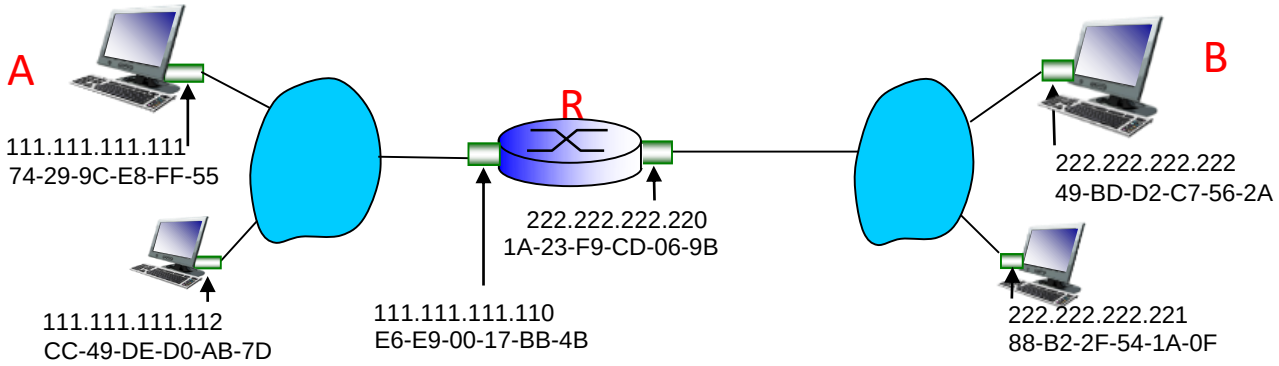


- Routers are also usually configured with their own default gateway.

- Devices need the following information for network access: *IP address*, *subnet mask*, and *default gateway*.
- When a host sends a packet to a device that is on the same IP network, the packet is forwarded out the host interface to the destination device. The router does not need to get involved.
- When a host sends a packet to a device on a different IP network, the packet is forwarded to the default gateway because the host device cannot communicate with devices outside of the local network.
- The default gateway is the device that routes traffic from the local network to devices on remote networks, such as devices on the Internet.

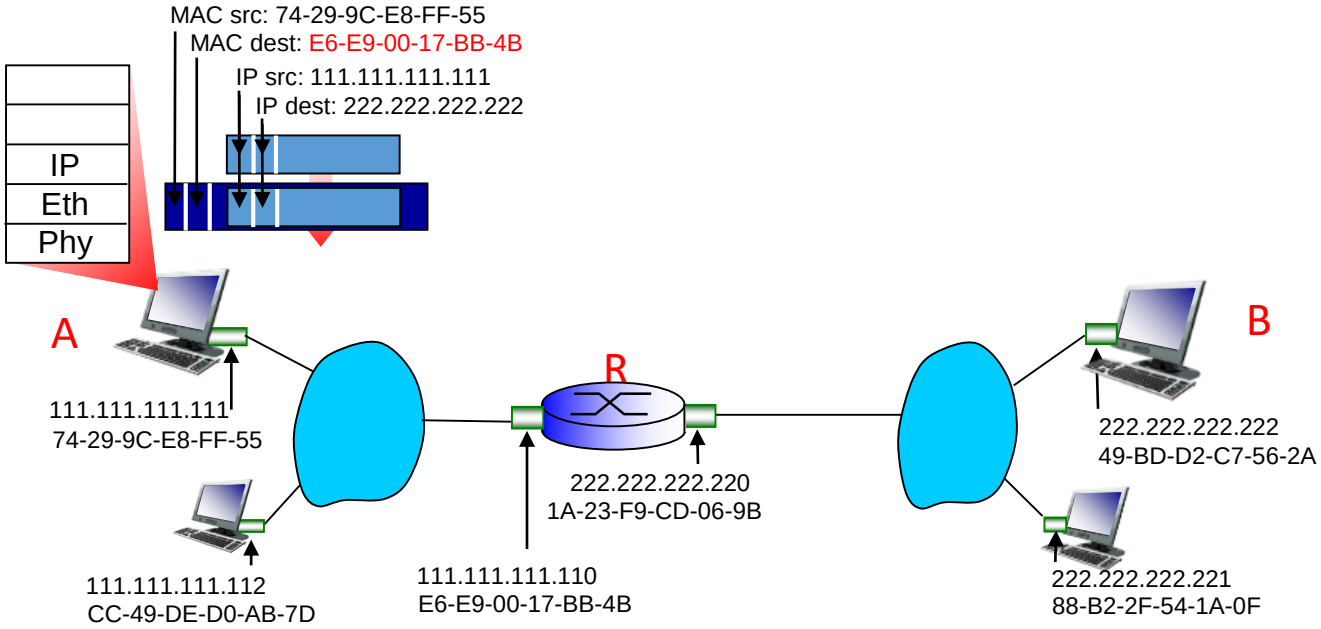
Addressing: routing to another LAN

- walkthrough: send datagram from A to B via R
  - Focus on addressing – at IP (datagram) and MAC layer (frame)
  - Assume A knows B's IP address
  - Assume A knows IP address of first hop router, R (how?)
  - Assume A knows R's MAC address (how?)



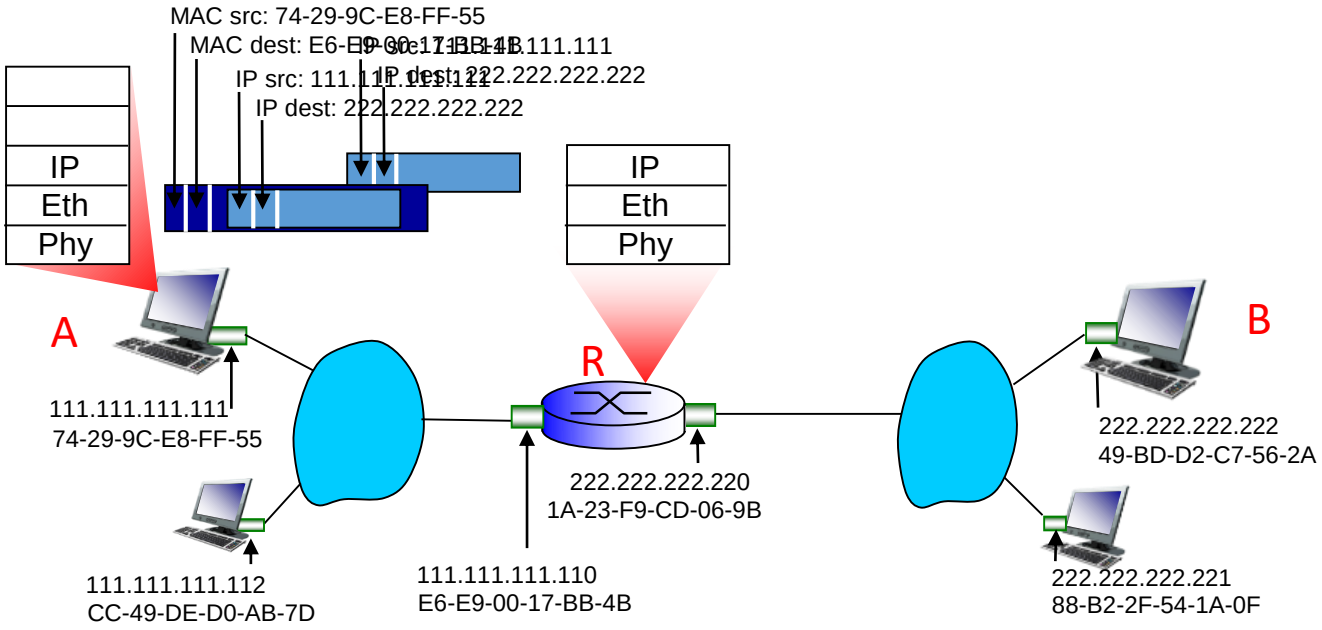
Addressing: routing to another LAN

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as destination address, frame contains A-to-B IP datagram



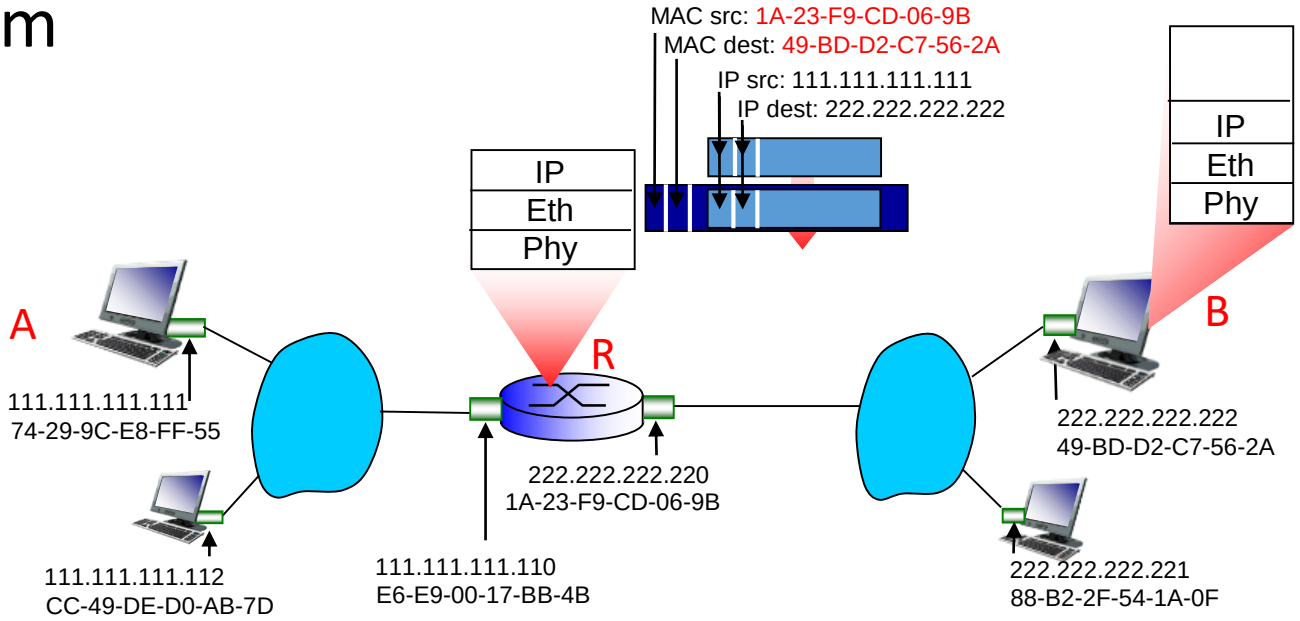
Addressing: routing to another LAN

- Frame sent from A to R
- Frame received at R, datagram removed, passed up to IP



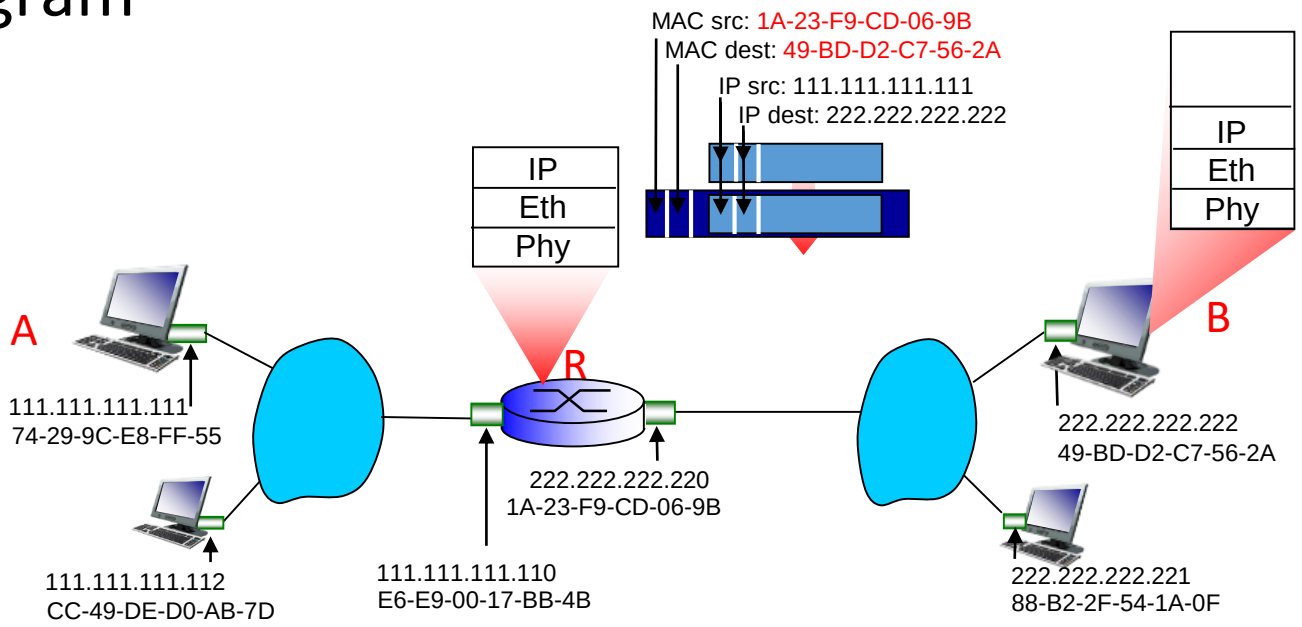
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram



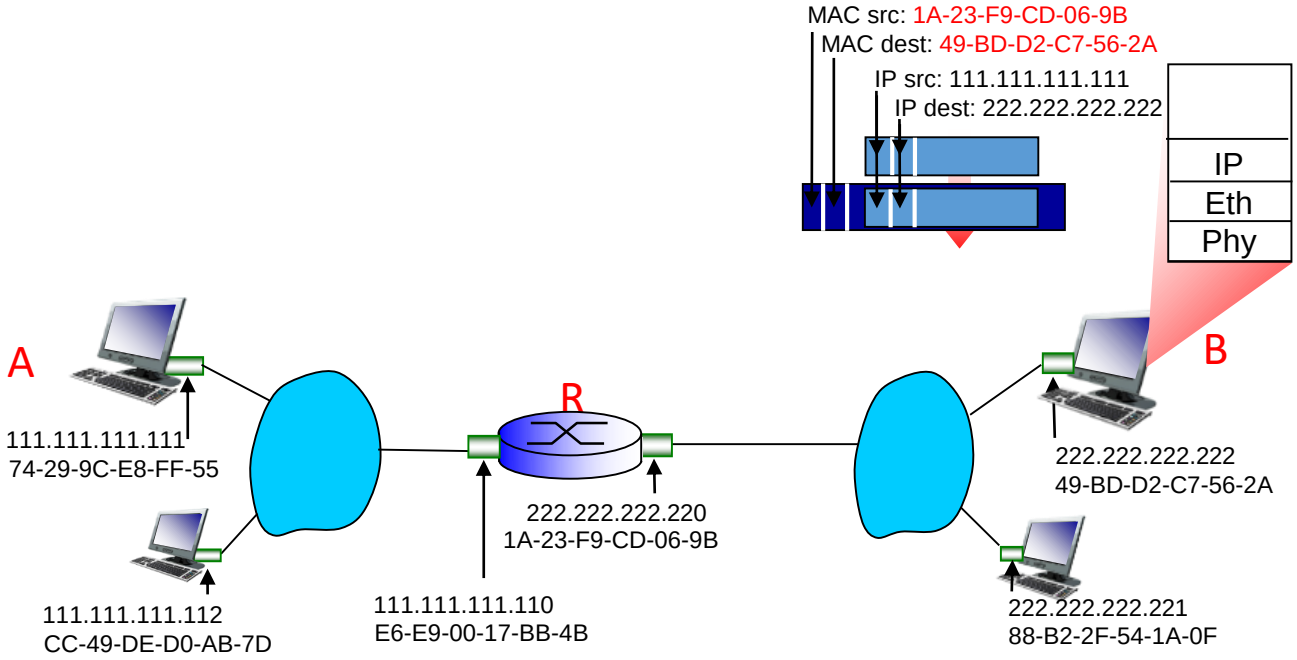
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram



Addressing: routing to another LAN

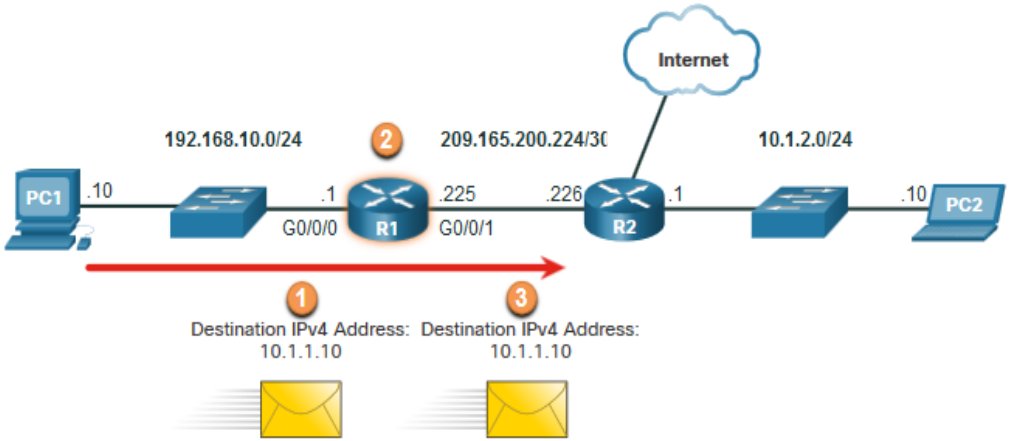
- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Introduction to Routing
Router Packet Forwarding Decision

- What happens when the router receives the frame from the host device?

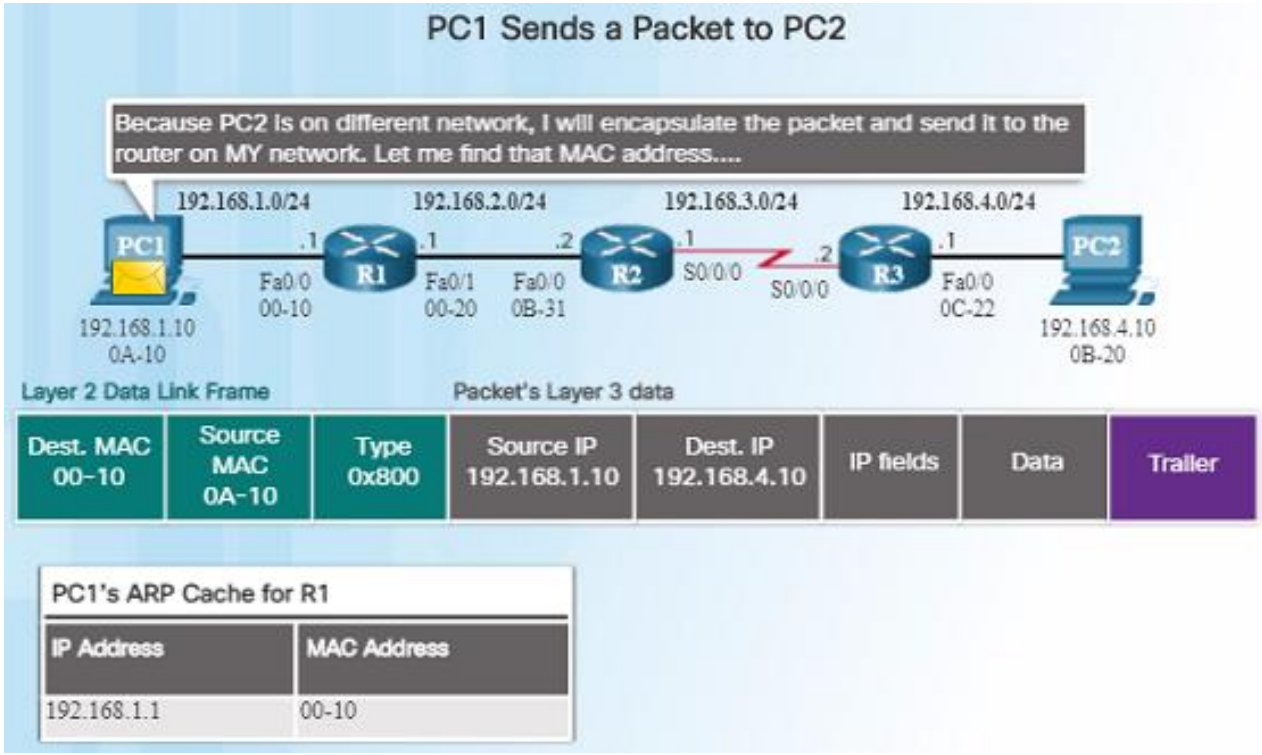


1. Packet arrives on the Gigabit Ethernet 0/0/0 interface of router R1. R1 de-encapsulates the Layer 2 Ethernet header and trailer.
2. Router R1 examines the destination IPv4 address of the packet and searches for the best match in its IPv4 routing table. The route entry indicates that this packet is to be forwarded to router R2.
3. Router R1 encapsulates the packet into a new Ethernet header and trailer, and forwards the packet to the next hop router R2.

R1 Routing Table

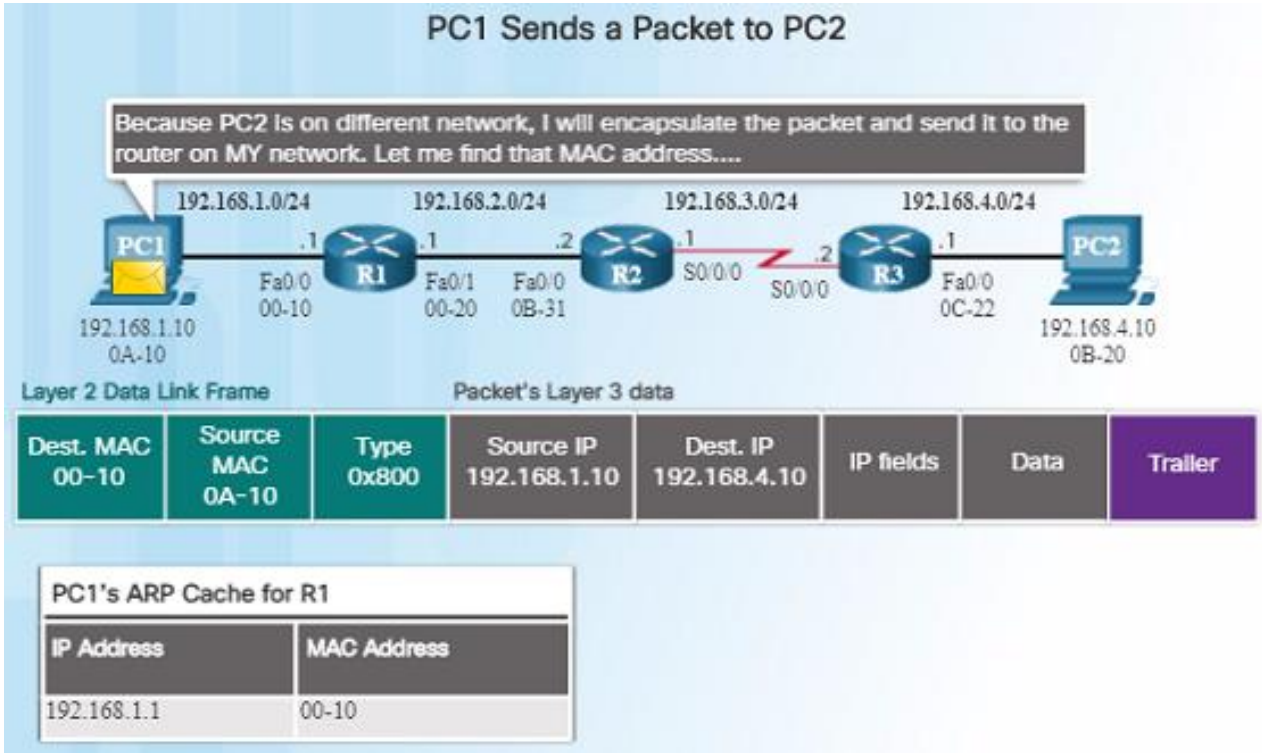
Route	Next Hop or Exit Interface
192.168.10.0 /24	G0/0/0
209.165.200.224/30	G0/0/1
10.1.1.0/24	**via R2**
Default Route 0.0.0.0/0	via R2

Switching Packets Between Networks: Send a Packet



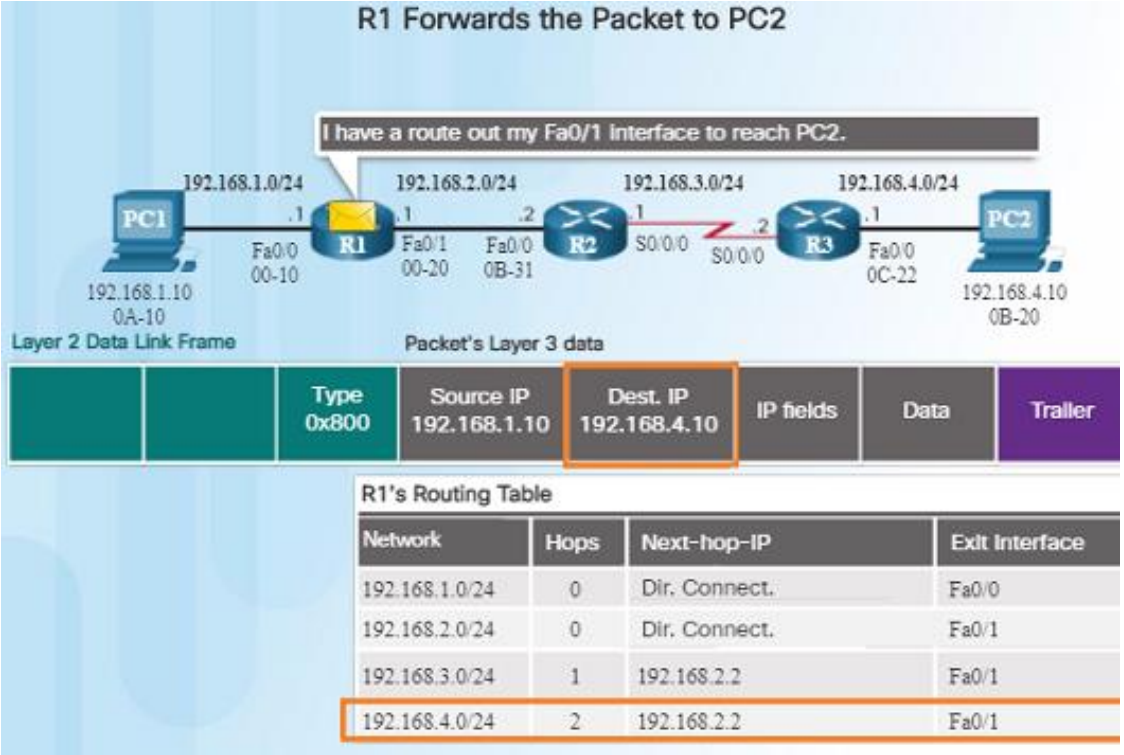
- PC1 must determine if the destination IPv4 address is on the same network.
- If it is on the same network, PC1 will obtain the destination MAC address from its ARP cache or use an ARP request.
- Because the destination network is on a different network, PC1 forwards the packet to its default gateway.

Switching Packets Between Networks: Send a Packet



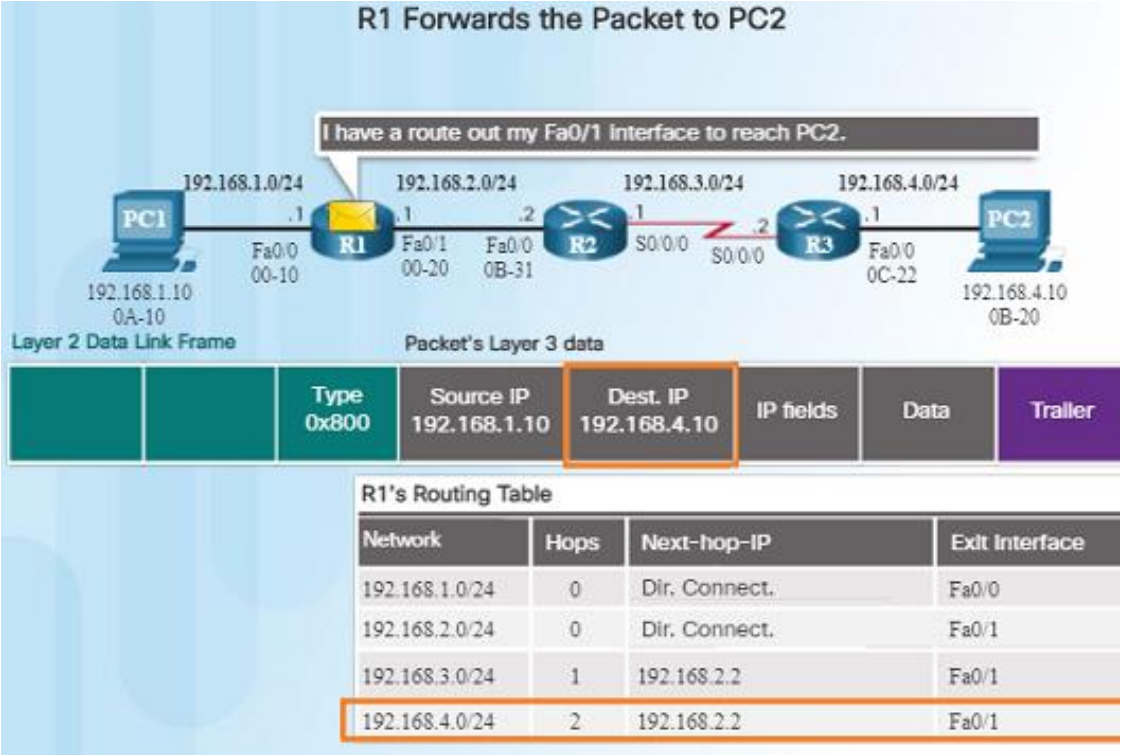
- **To determine the MAC address of the default gateway**, PC1 checks its ARP table for the IPv4 address of the default gateway and its corresponding MAC address.
  - An ARP request is sent if it is not found.
- When PC1 has the MAC address of Router R1, it can forward the packet.

Switching Packets Between Networks: Forward to the Next Hop



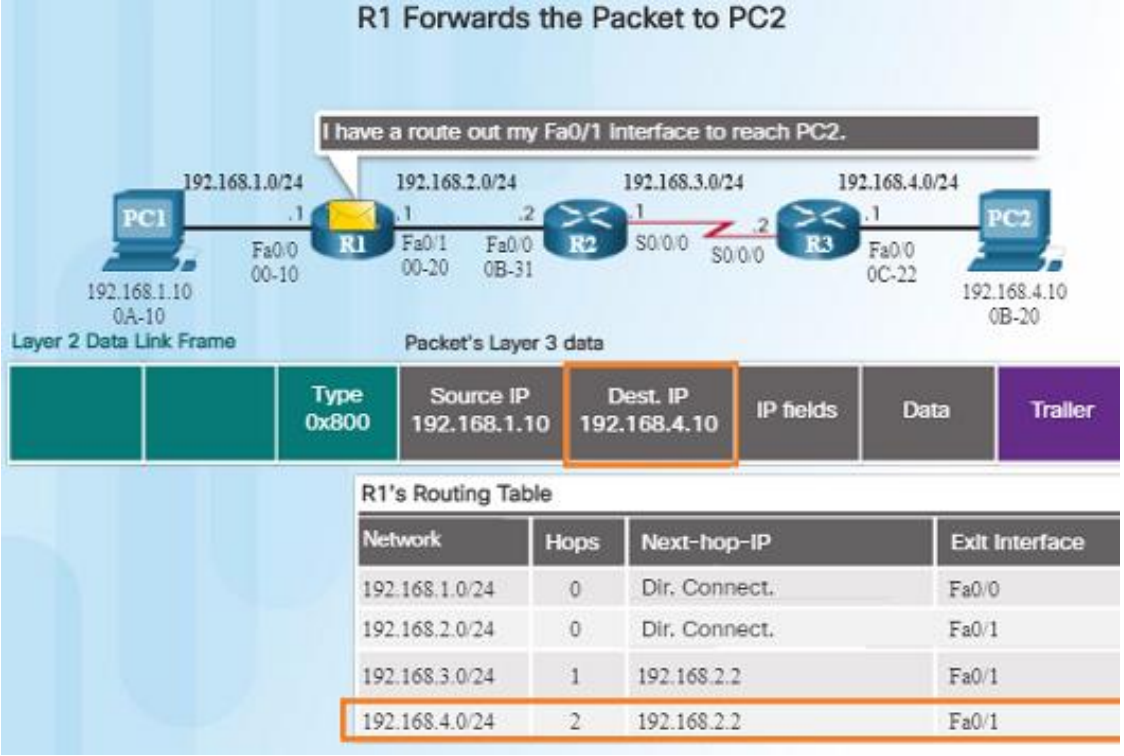
- R1 **examines the destination MAC address** which matches the MAC address of the receiving interface and copies the frame into its buffer.
- R1 **identifies the Ethernet Type field as 0x800** which indicates that the Ethernet frame contains an IPv4 packet in the data portion of the frame.

Switching Packets Between Networks: Forward to the Next Hop



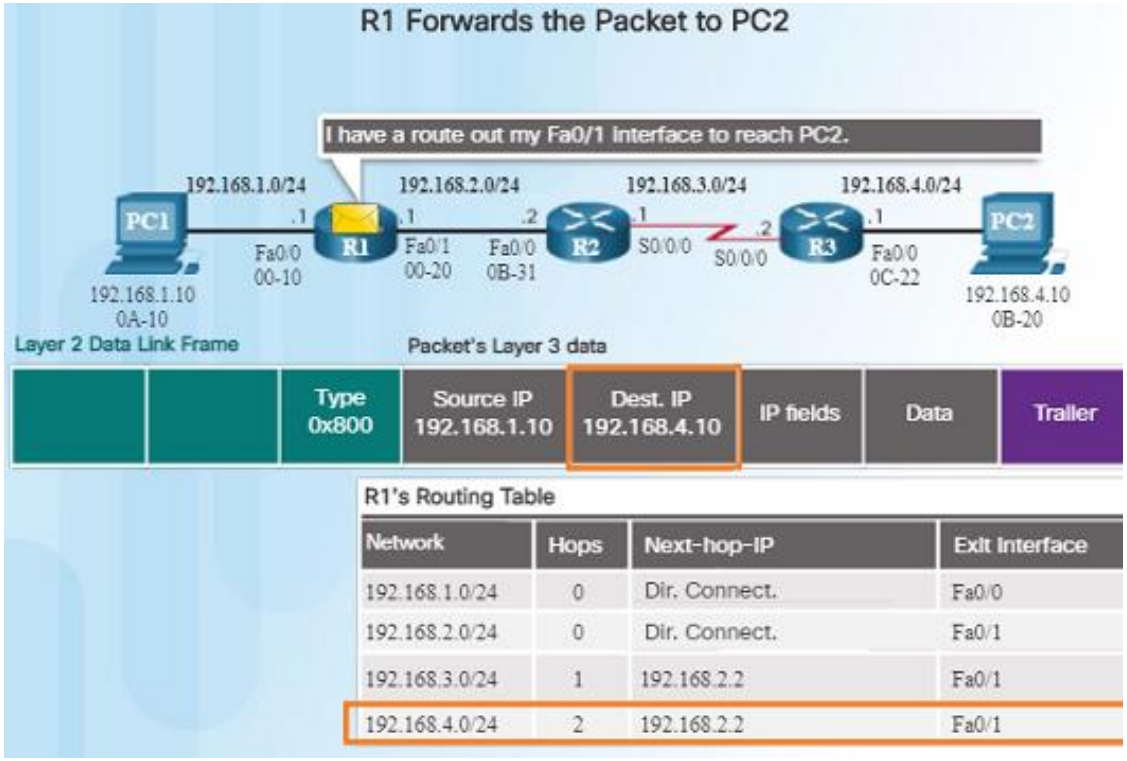
- R1 de-encapsulates the Ethernet frame.
- Because the destination IPv4 address of the packet, 192.168.4.10, does not match any of the directly connected networks on R1, R1 searches the routing table for a corresponding route.
  - R1's Routing Table has a route for the 192.168.4.0/24 network.

Switching Packets Between Networks: Forward to the Next Hop



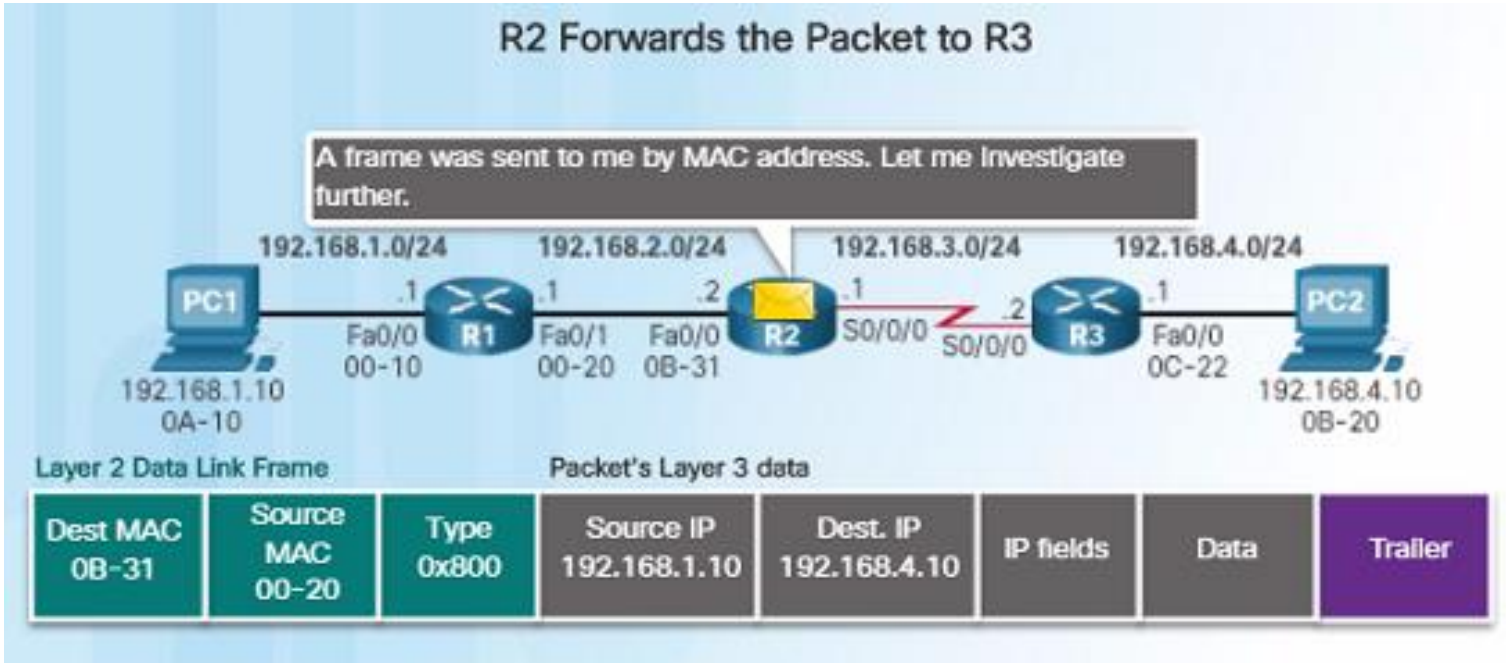
- The **route** that R1 finds to the 192.168.4.0/24 network has a **next-hop address** of 192.168.2.2 and an **exit interface** of FastEthernet 0/1.
- This will require that the IPv4 packet be encapsulated in a new Ethernet frame with the destination MAC address of the IPv4 address of the next-hop router, 192.168.2.2

Switching Packets Between Networks: Forward to the Next Hop



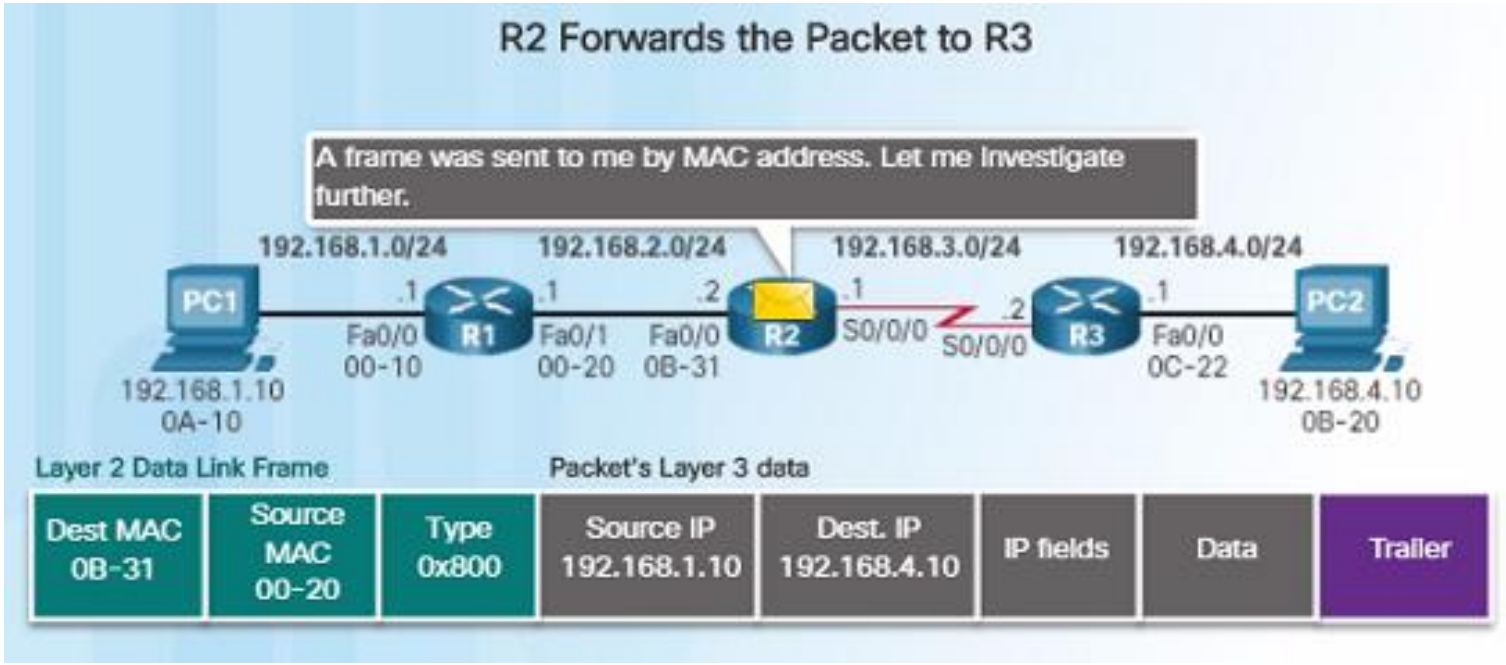
- Because the exit interface is on an Ethernet network, R1 must resolve the next-hop IPv4 address with a destination MAC address using ARP, assuming it is not in its ARP cache.
- When R1 has the MAC address for the next-hop, the Ethernet frame is forwarded out of the FastEthernet 0/1 interface of R1.

Switching Packets Between Networks
Packet Routing



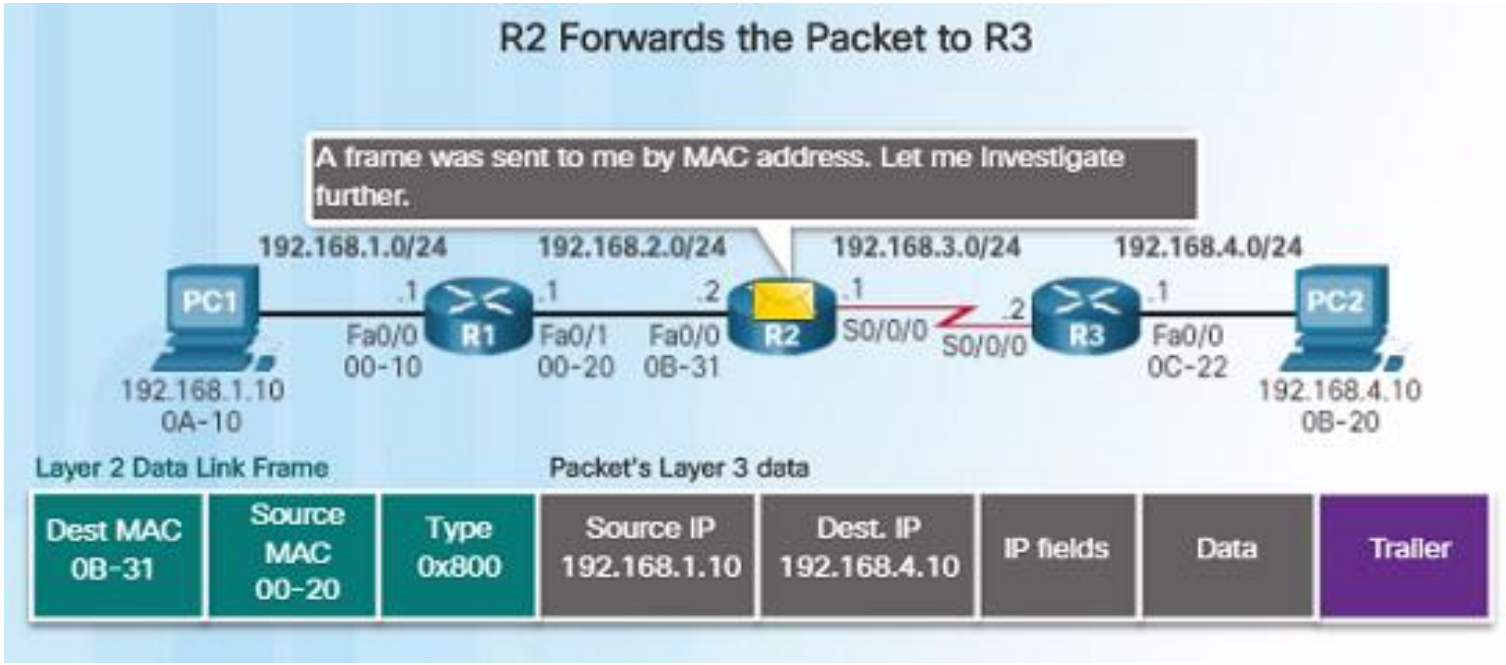
- R2 examines the destination MAC address
  - Because it matches the MAC address of its receiving interface, **R2 copies the frame into its buffer**.
- R2 determines that that **frame contains an IPv4 packet** in the data portion of the frame.

Switching Packets Between Networks
Packet Routing



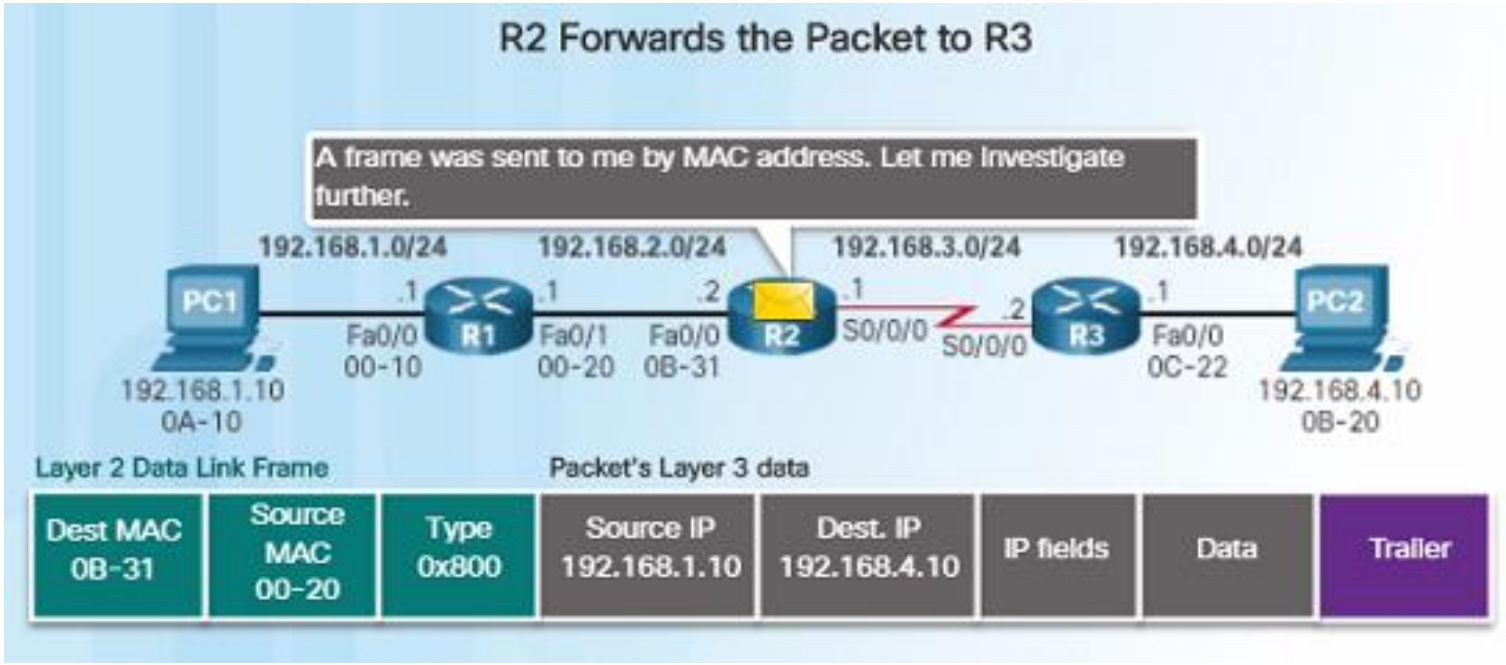
- R2 **de-encapsulates the Ethernet frame**.
- Because the destination IP address is on a different network, the **routing table is searched** to find a **corresponding route** for the destination IPv4 address.

Switching Packets Between Networks
Packet Routing



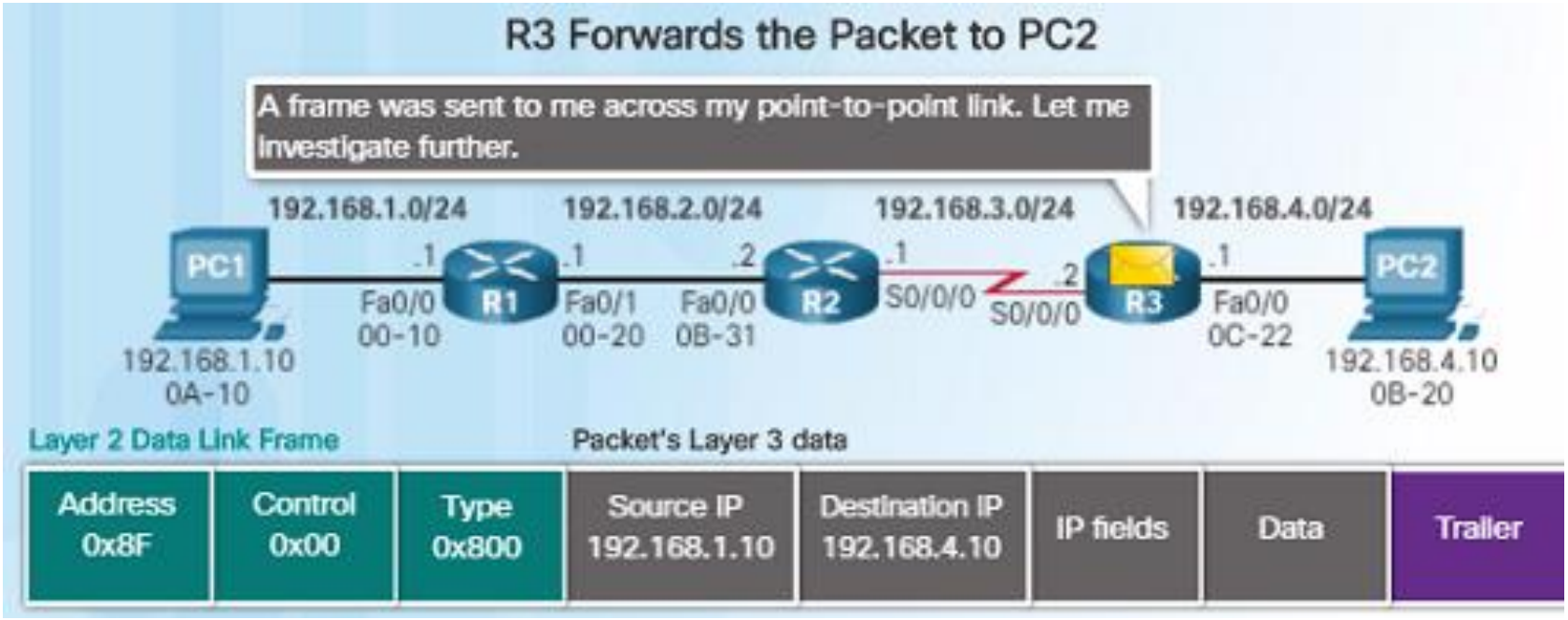
- The routing table of R2 has a **route** to the 192.168.4.0/24 network with a **next-hop IPv4 address** of 192.168.3.2 and an **exit interface** of Serial 0/0/0.
- Because the **exit interface is not Ethernet**, R2 does not have to resolve the next-hop IP-v4 address with a destination MAC address.

Switching Packets Between Networks
Packet Routing



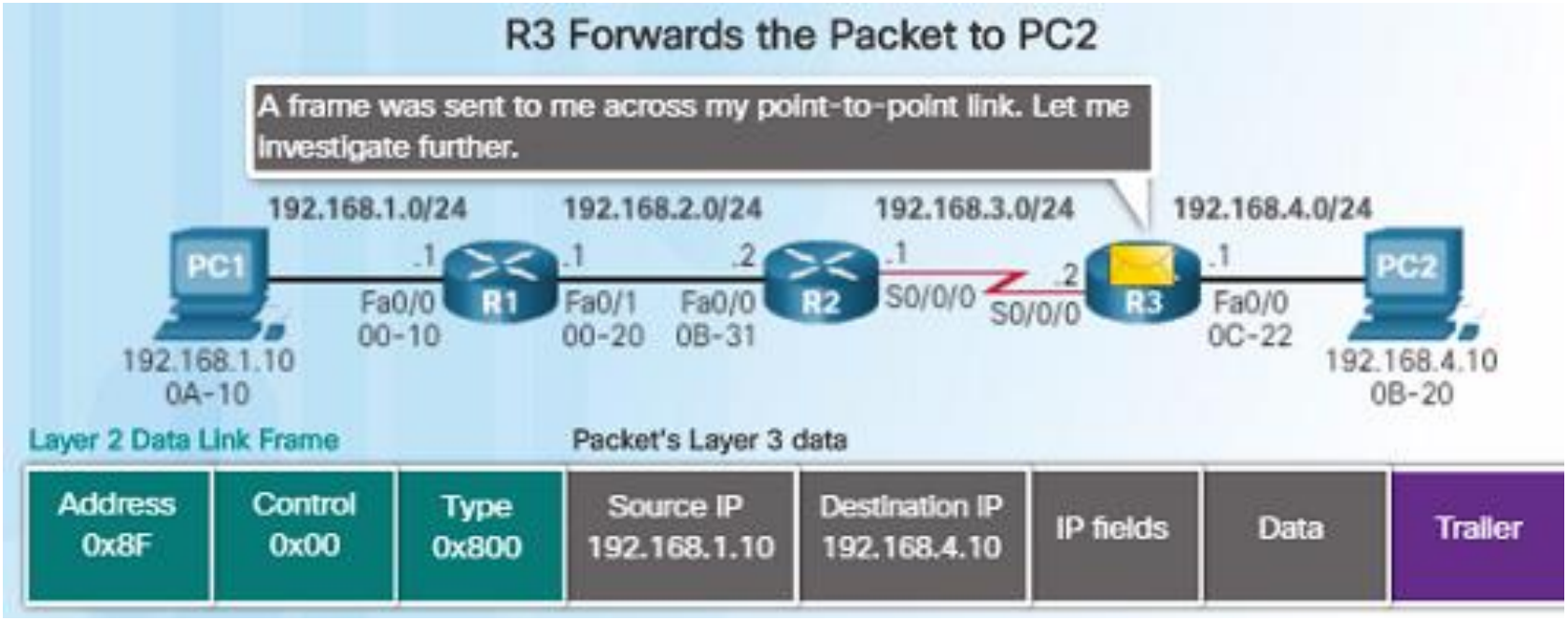
- The IPv4 packet is **encapsulated** into a **new data link frame** used by the exit interface and sent out the Serial 0/0/0 exit interface.
- Because there are no MAC addresses on serial interfaces, R2 sets the data link destination address to an equivalent of a broadcast.

Switching Packets Between Networks
Reach the Destination



- R3 copies the data link PPP frame into its buffer.
- R3 **de-encapsulates** the data link PPP frame.
- R3 **searches the routing table** for the destination IPv4 address of the packet.

Switching Packets Between Networks
Reach the Destination



- Because the destination network is on R3's directly connected network, the packet can be sent directly and does not need to be sent to another router.
- Because the **exit interface is a directly connected Ethernet network**, R3 must resolve the destination IPv4 address of the packet with a destination MAC address by either finding it in its ARP cache or send out an ARP request.

Internet Control Message Protocol

- The Internet Control Message Protocol (ICMP)
- Specified in [RFC 792],
- Used by hosts and routers to communicate network-layer information to each other.
- Most typical use of ICMP is for error reporting
- Often considered part of IP, but architecturally it lies just above IP, as ICMP messages are carried inside IP datagrams.
  - ICMP messages are carried as IP payload

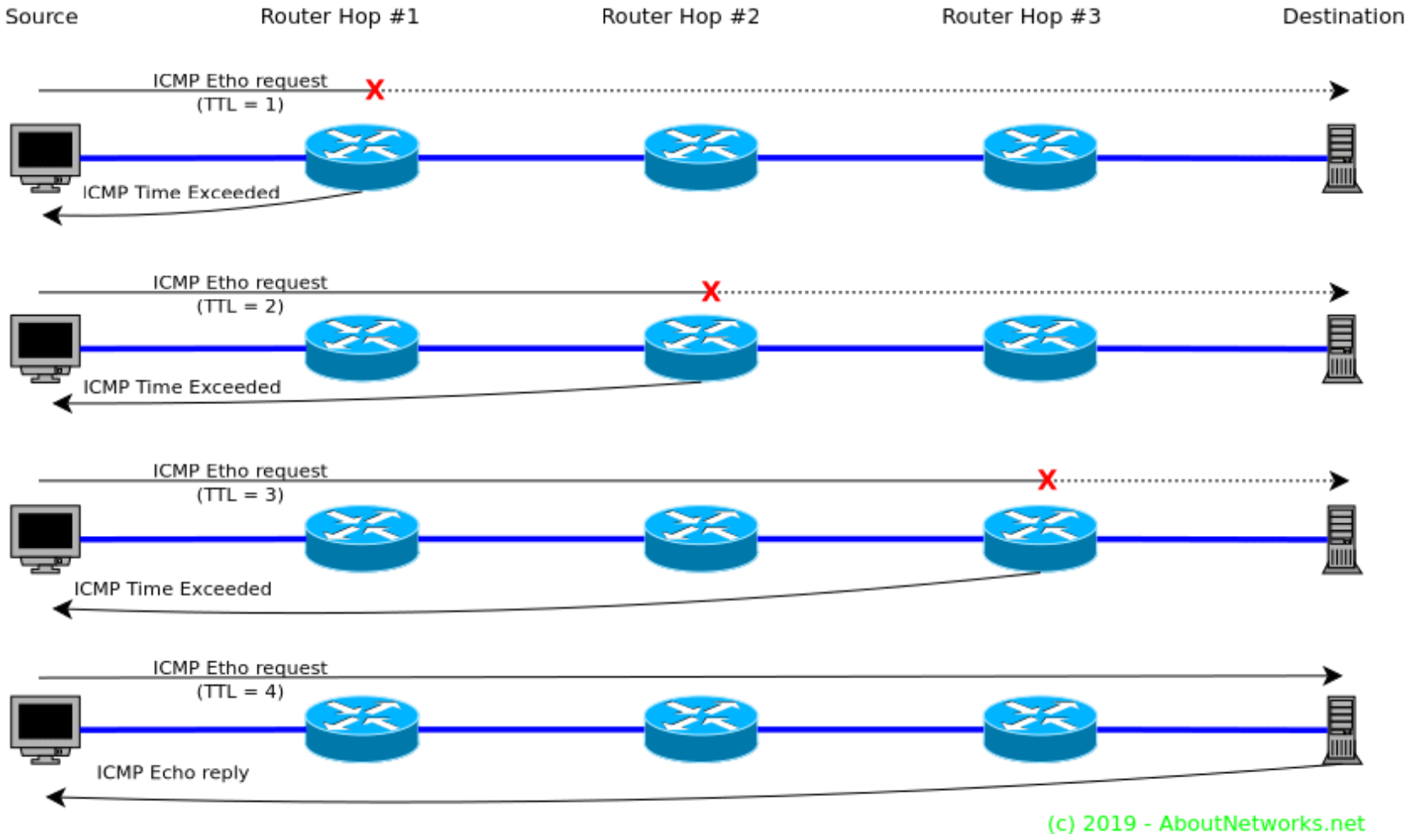
ICMP Type and Code

Type	Code	Description
0	0	echo reply (ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

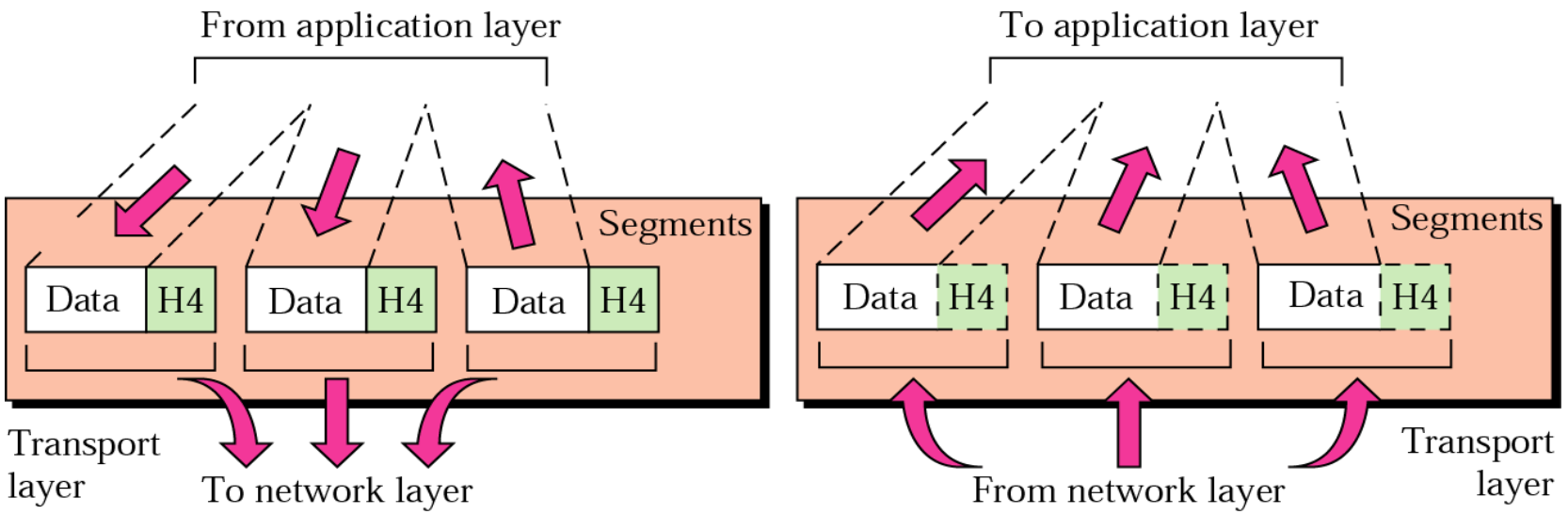
Trace Route – ICMP

- Source sends series of UDP segments (or ICMP Echo request) to destination host
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When n^{th} datagram arrives to n^{th} router:
  - Router discards datagram
  - Sends to source an ICMP message (type 11, code 0)
  - Datagram includes router IP address. Traceroute does DNS lookup to find name of router (if any)
- When ICMP message arrives, source calculates RTT
- Traceroute repeat the process 3 times (To get average RTT)
- Stopping criterion
  - UDP segment eventually arrives at destination host
  - Destination returns ICMP “port unreachable” packet (type 3, code 3)

Trace Route – ICMP



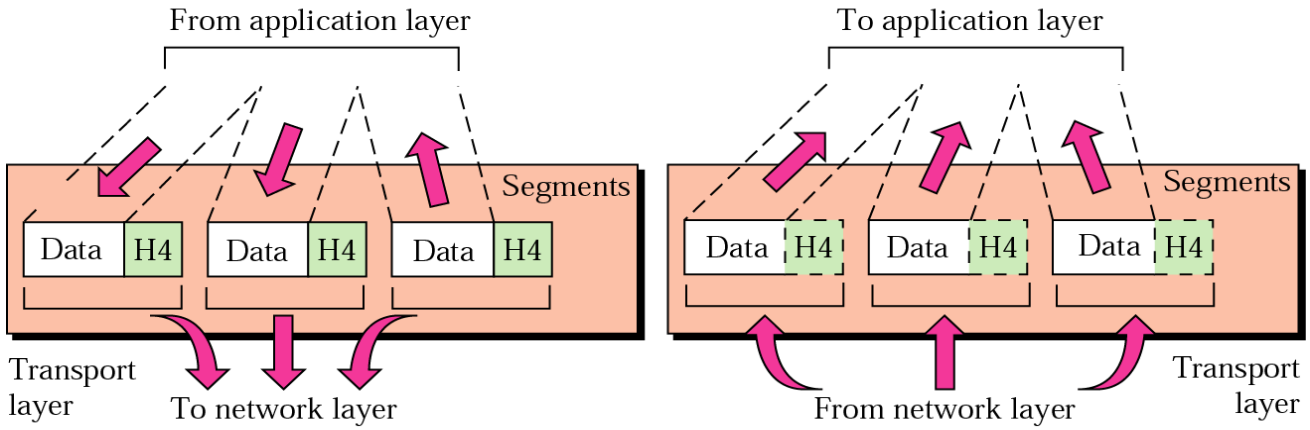
Transport layer



The transport layer is responsible for delivering segments from a source process to a final destination process.

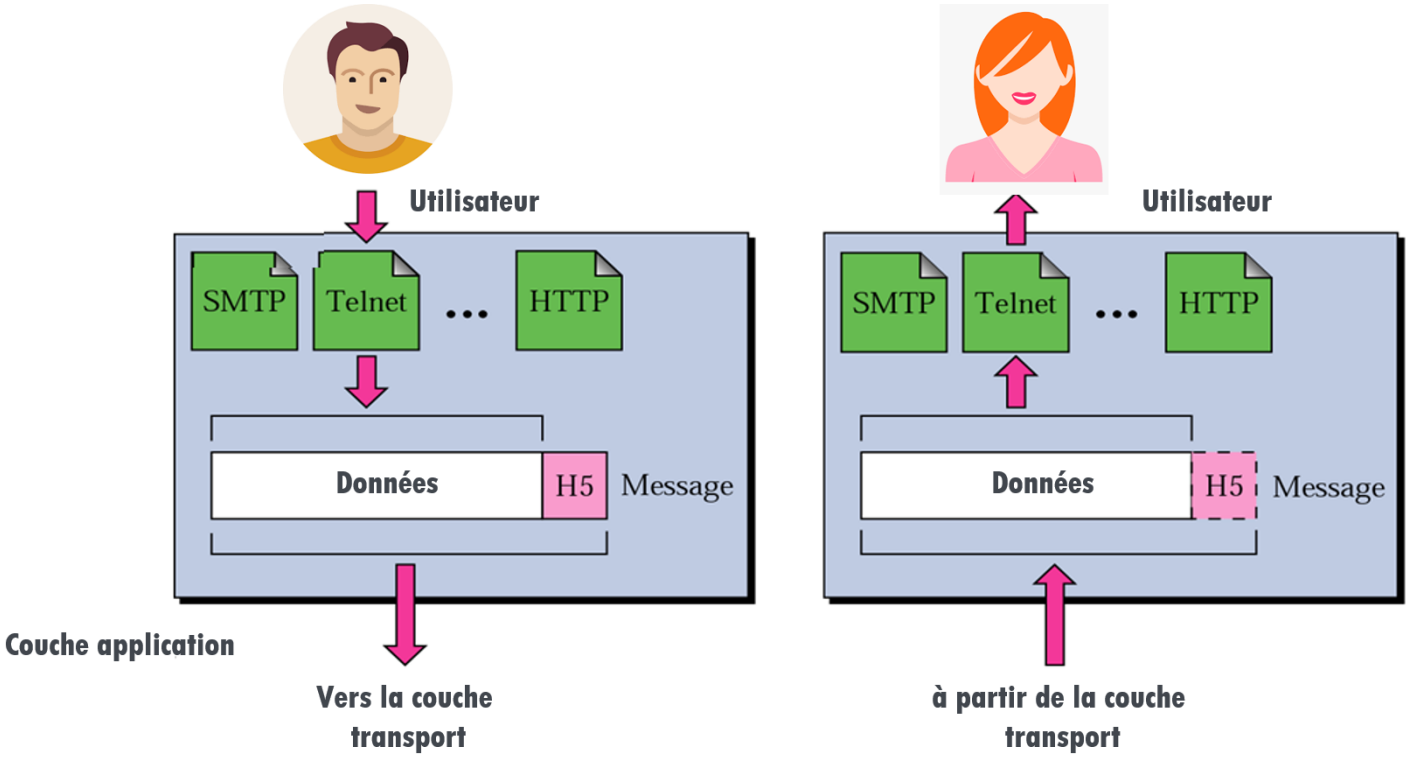
Transport layer

- Transport Layer Responsibilities
  - Process-to-process delivery of entire message
  - Port addressing
  - Segmentation and reassembly
  - Connection control: connectionless or connection-oriented
  - End-to-end flow control
  - End-to-end error control



Application Layer

- The application layer is responsible for providing services to the user.



Thanks !