OSI Model
TCP/IP Reference Model

Computer Networks

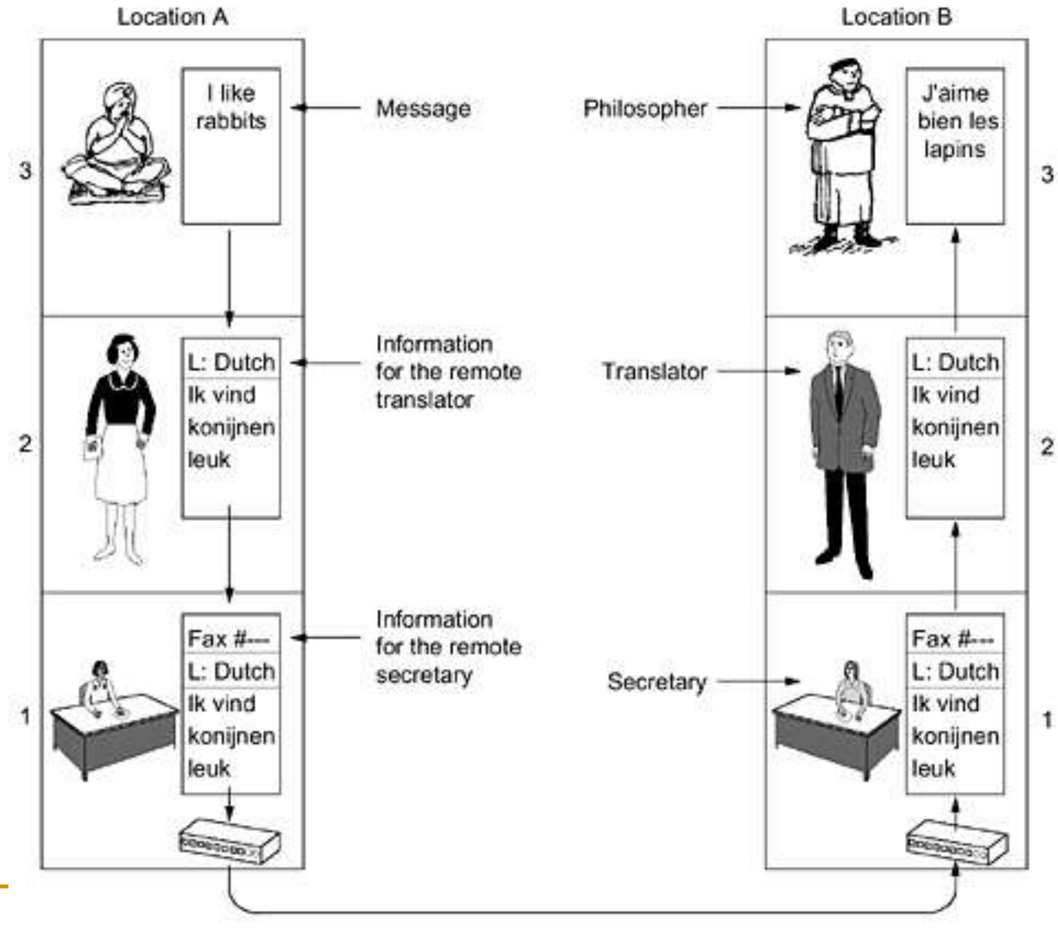
Faculty of Information Technology
Hanoi University

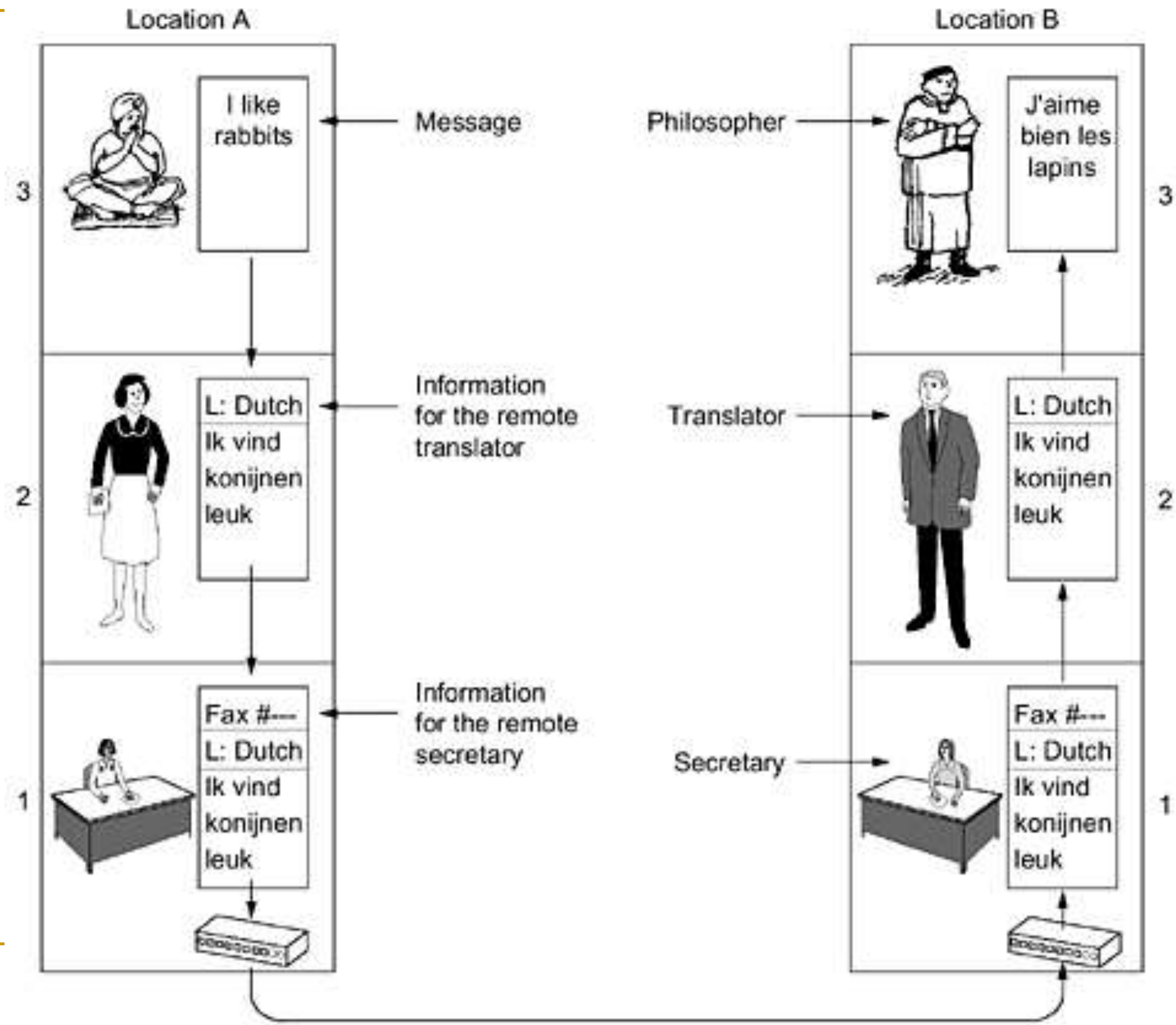
Content

1. Layered structure of network
2. OSI Model
3. Internet and TCP/IP Model

1. Layered structure of network

- An analogy of network layers

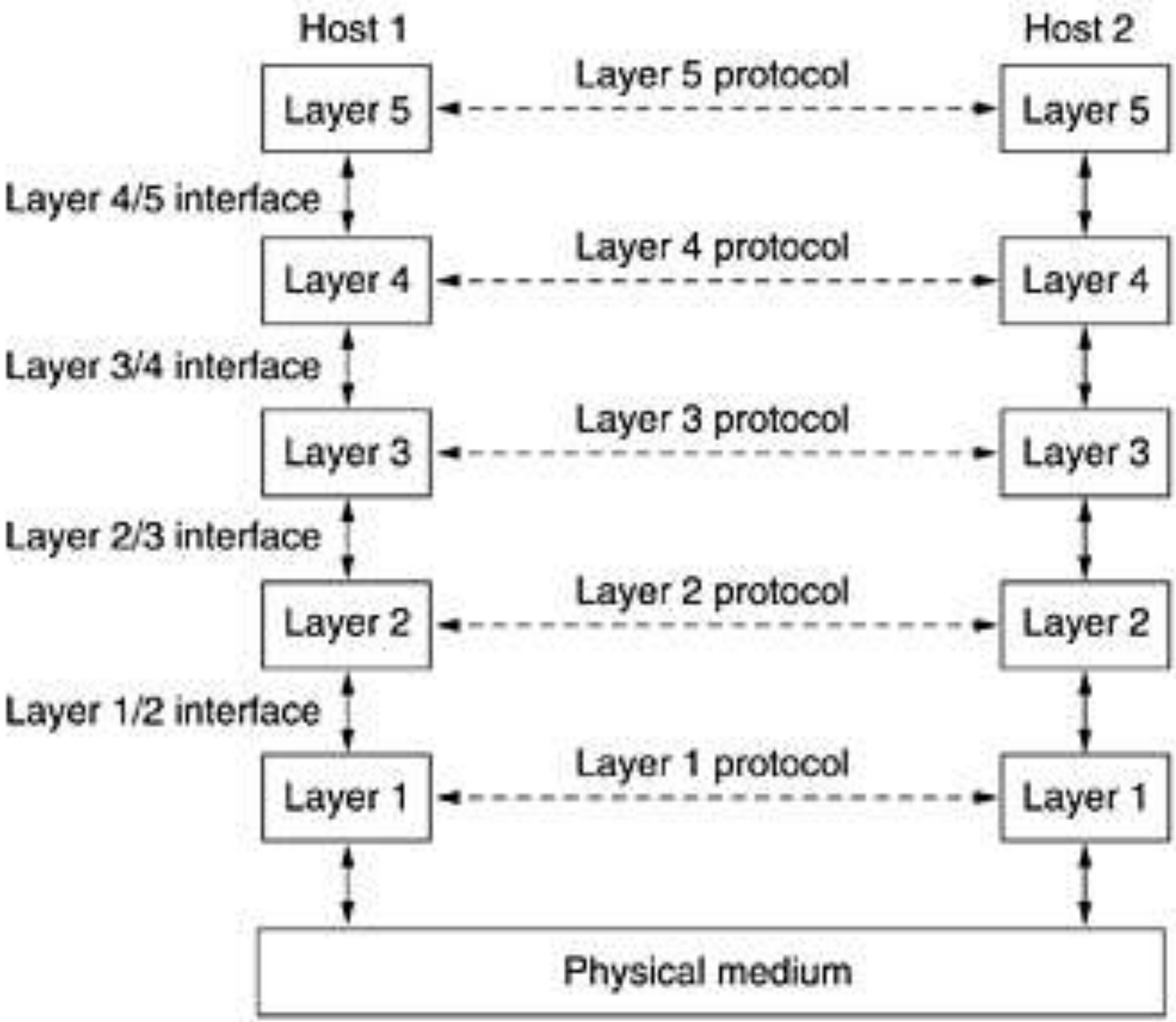



Location A
Location B
I like rabbits
Message
Philosopher
J'aime bien les lapins
3
3
L: Dutch
Ik vind konijnen leuk
Information for the remote translator
Translator
L: Dutch
Ik vind konijnen leuk
2
2
Fax #---
L: Dutch
Ik vind konijnen leuk
Information for the remote secretary
Secretary
Fax #---
L: Dutch
Ik vind konijnen leuk
1
1

Layers, Protocols and Interfaces

- Networks are organized as a stack of layers. Each layer offers certain services to the higher layers.
- Between each pair of adjacent layers is an interface.
- The rules used in conversation between layers n on two machines is called layer n protocol.
- A list of protocols used by a certain system, one protocol per layer, is called a protocol stack.

Example



Host 1
Host 2
Layer 5 protocol
Layer 5
Layer 5
Layer 4/5 interface
Layer 4 protocol
Layer 4
Layer 4
Layer 3/4 interface
Layer 3 protocol
Layer 3
Layer 3
Layer 2/3 interface
Layer 2 protocol
Layer 2
Layer 2
Layer 1/2 interface
Layer 1 protocol
Layer 1
Layer 1
Physical medium

Why layered structure?

- Breaking up the design problem into smaller, more manageable pieces.
- Protocols are designed for each layer.
- Protocols can be changed without affecting higher or lower ones.

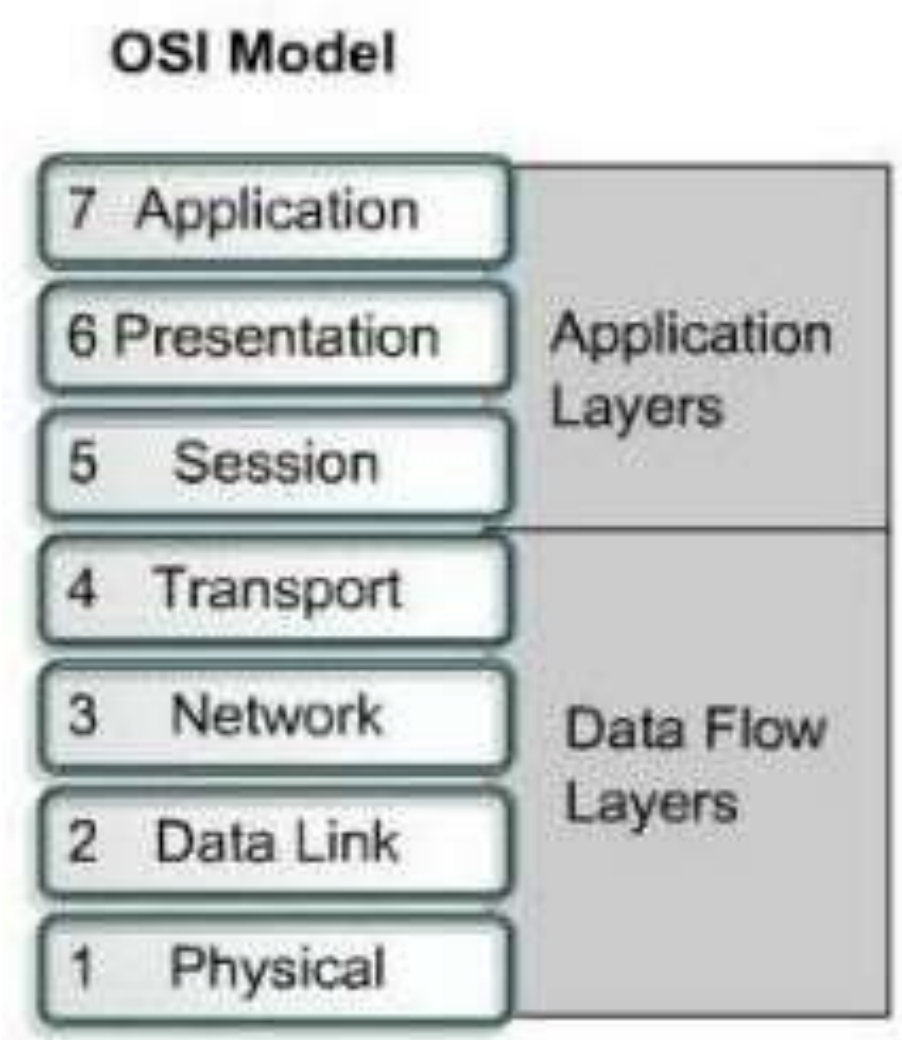
Drawback of layering

- Layering has many conceptual advantages, BUT fanatical adherence to layering as a religion is problematic:
  - Layer N may duplicate lower layer functionality (e.g. Error recovery)
  - Different layers may need same information (e.g. Timestamp)

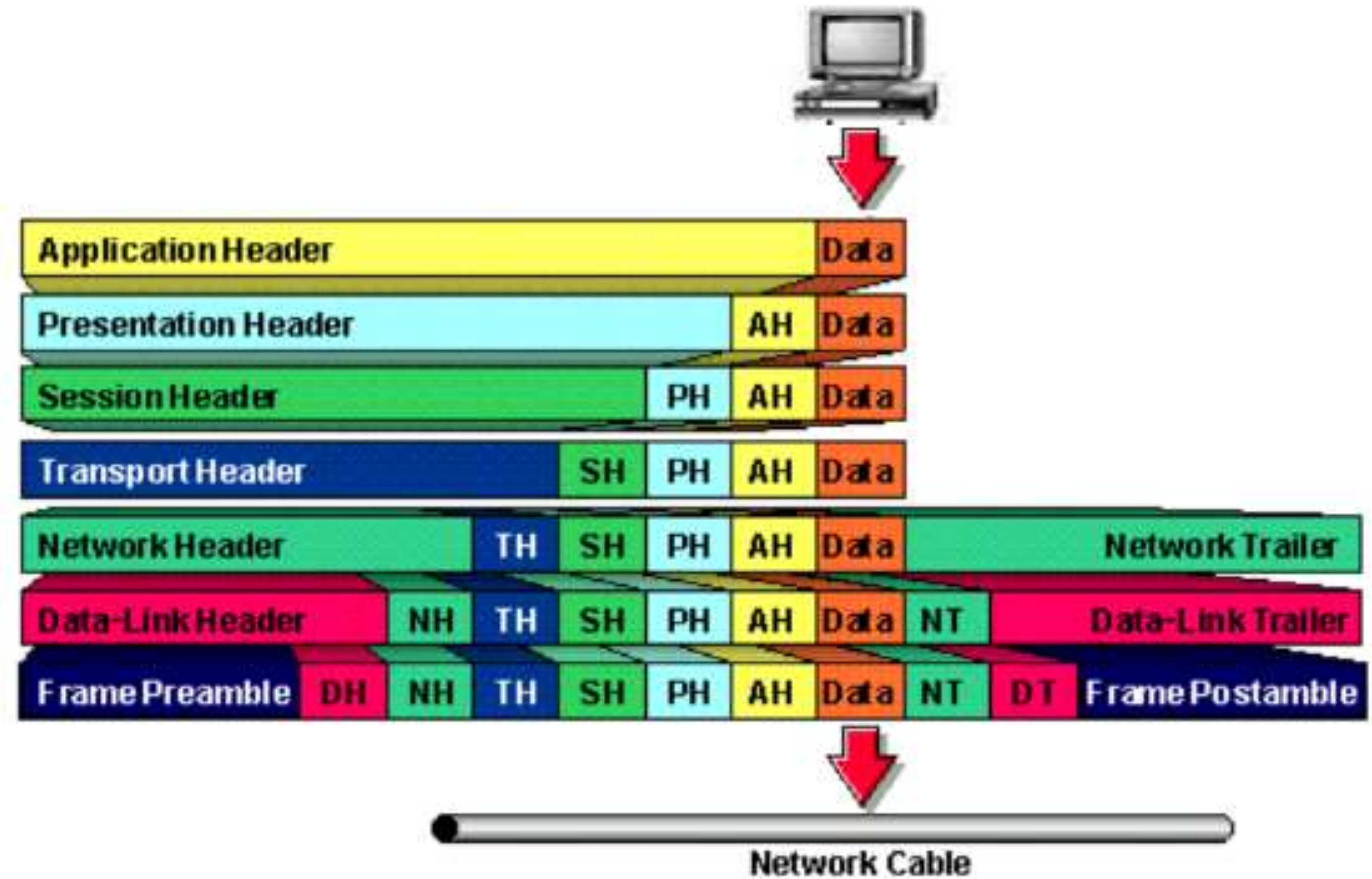
2. OSI Model

- It’s difficult to connect networks of different layered structures -> we must standardize layered model.
- ISO proposed OSI Model (Open Systems Interconnection Model)

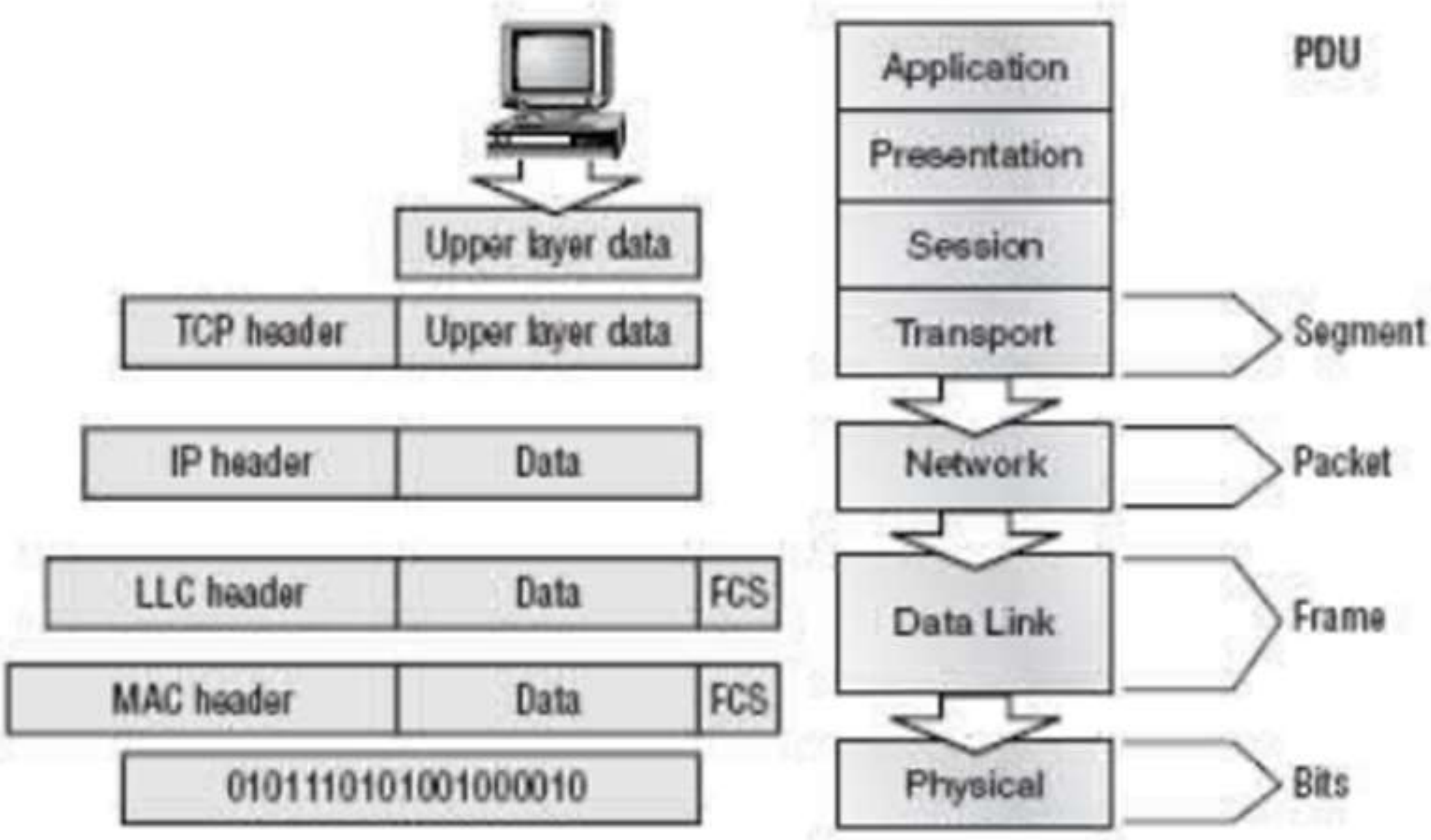
OSI Model


OSI Model
7 Application
6 Presentation
5 Session
4 Transport
3 Network
2 Data Link
1 Physical
Application Layers
Data Flow Layers

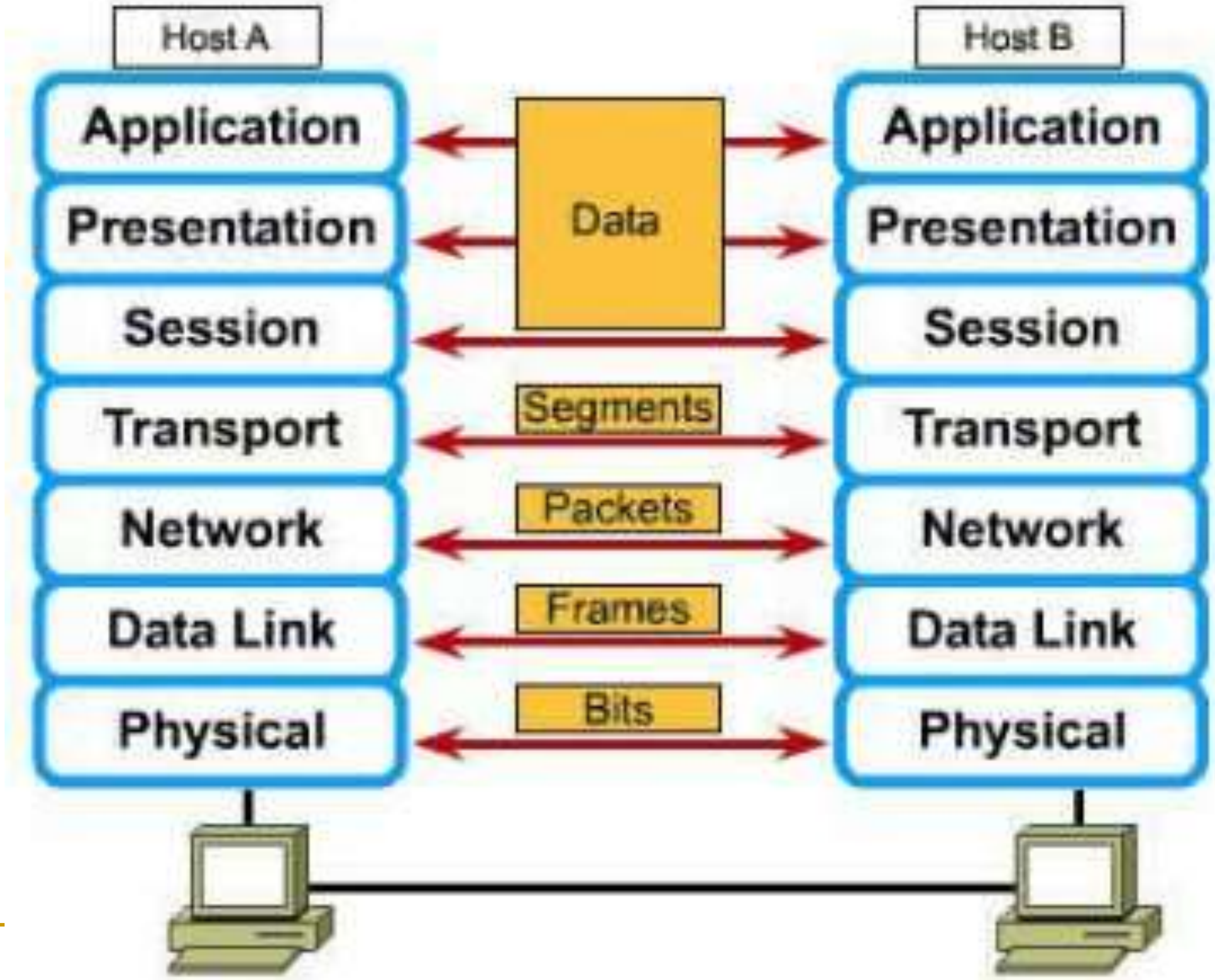
OSI Model : Data encapsulation



OSI Model: PDU

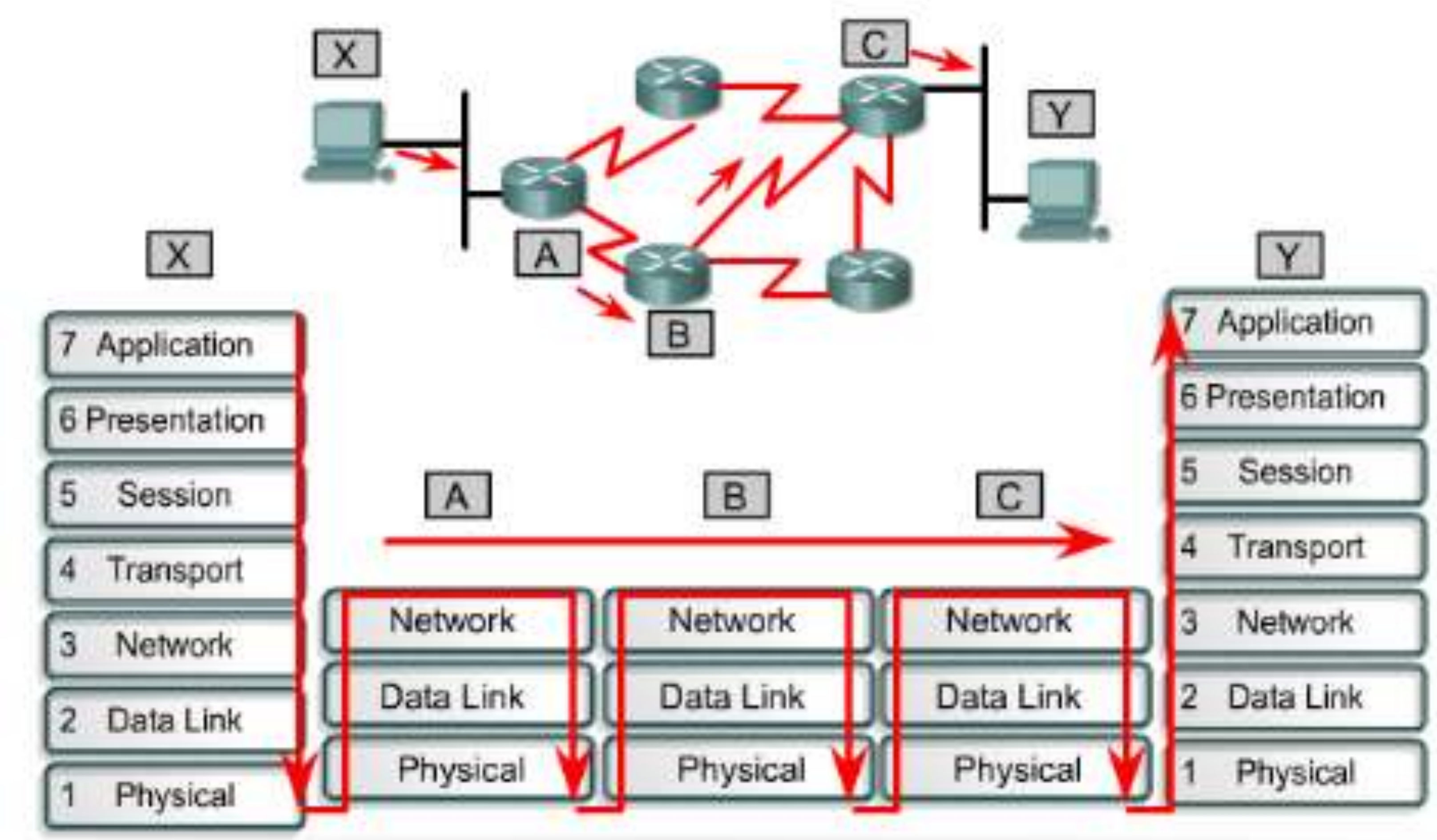


OSI Model



Host A
Host B
Application
Presentation
Session
Transport
Network
Data Link
Physical
Data
Segments
Packets
Frames
Bits
Application
Presentation
Session
Transport
Network
Data Link
Physical

Data flow in network



3. Internet and TCP/IP Model

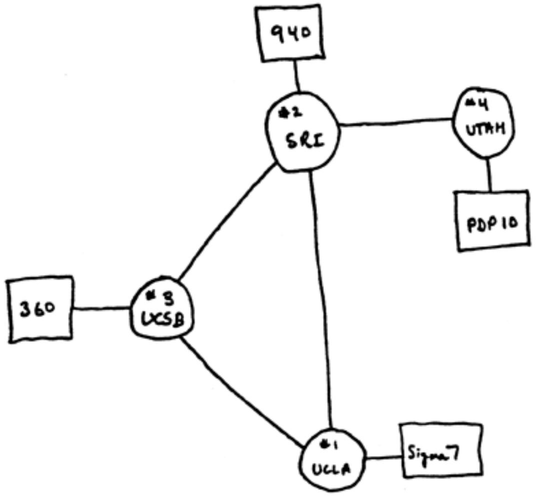
Selected Internet Milestones

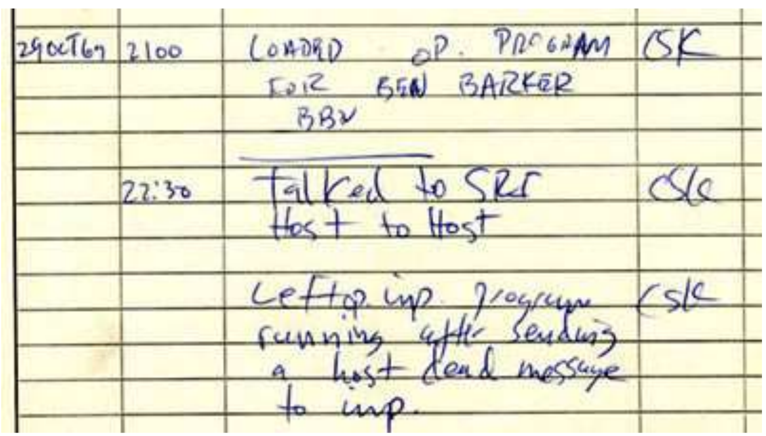
- 1969 – Basic packet net research (ARPANET)
- 1974 - First paper on packet switching (Vint Cerf)
- 1983 – The first major Internet deployment (TCP/IP)
- 1984 - first router company (CISCO)
- 1991 – www (Tim Berners-Lee)
- 1995 – US commercial Internet
- 1996 – US next generation Internet
- 1997 – First Internet service in Vietnam

Internet - The Beginning

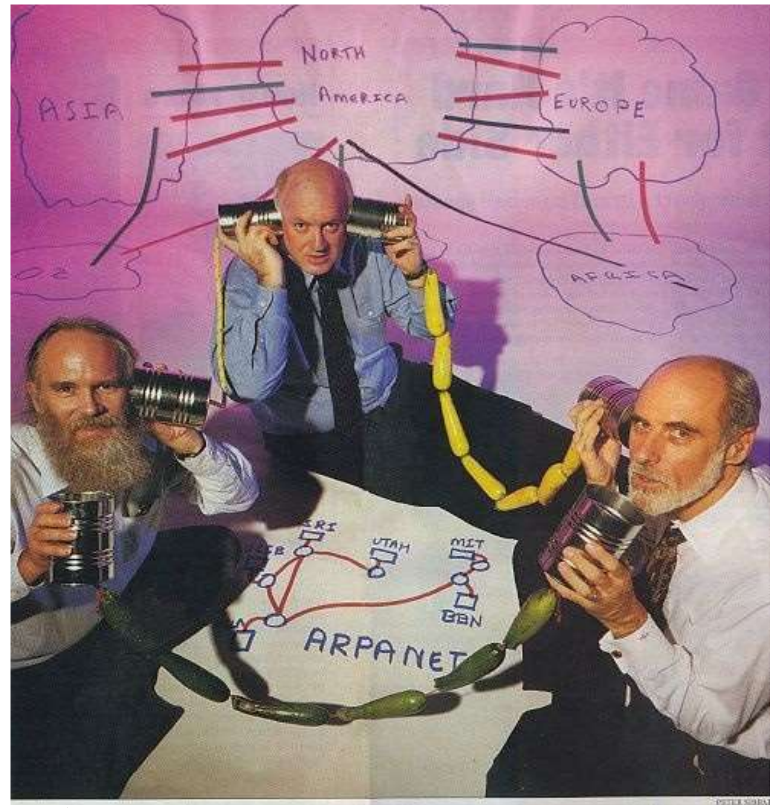
- Fall, 1968
  - US Department of Defense Advanced Research Projects Administration envisions what would become the ARPANet Project.
    - Official justification: create a communications network capable of surviving nuclear war
    - Real reason: An ARPA administrator had 3 terminals on his desk connecting to 3 different timesharing systems and he thought this was stupid.

Draft design in 1969





29 OCT 69	2100	LOADED OP. PROGRAM FOR BEN BARKER BBN	CSK
	22:30	Talked to SRI Host to Host	CSK
		Left op. imp. program running after sending a host dead message to imp.	CSK



Crocker

Jon Postel

Vint Cerf

The design philosophy of TCP/IP

- Fundamental goal
  - Packet switching
- Second level goals
  - Survivability in the face of failure
  - Multiple types of service
  - Varieties of networks

Connection-oriented service (Circuit switching)

- Establish a connection
- Use the connection
- Release the connection

- Example: The telephone system.

- Drawbacks?
- Advantages?

Connectionless service (Packet switching)

- Each message carries the full destination address
- Each one is routed through the system independent of all the others.

- Example: The postal system.

- Drawbacks?
- Advantages?

Why packet switching?

- Save/make money
- Example: 1Mbps link; each user requires 100Kbps when transmitting; each user has data to send only 10% of the time.
  - Circuit-switching
    - Given each called 100 Kbps capacity
    - Can support 10 callers
  - Packet switching
    - With 35 ongoing calls, probability that 10 or more callers are simultaneously active is less than 0.0004
    - Can support many more callers with small probability of contention.
- If users are burst (on/off), then packet-switching is advantageous

Non Goal

*"the lesson of the Internet is that **efficiency is not the primary consideration**. Ability to grow and adapt to changing requirements is the primary consideration. This makes simplicity and uniformity very precious indeed."*

Bob Braden

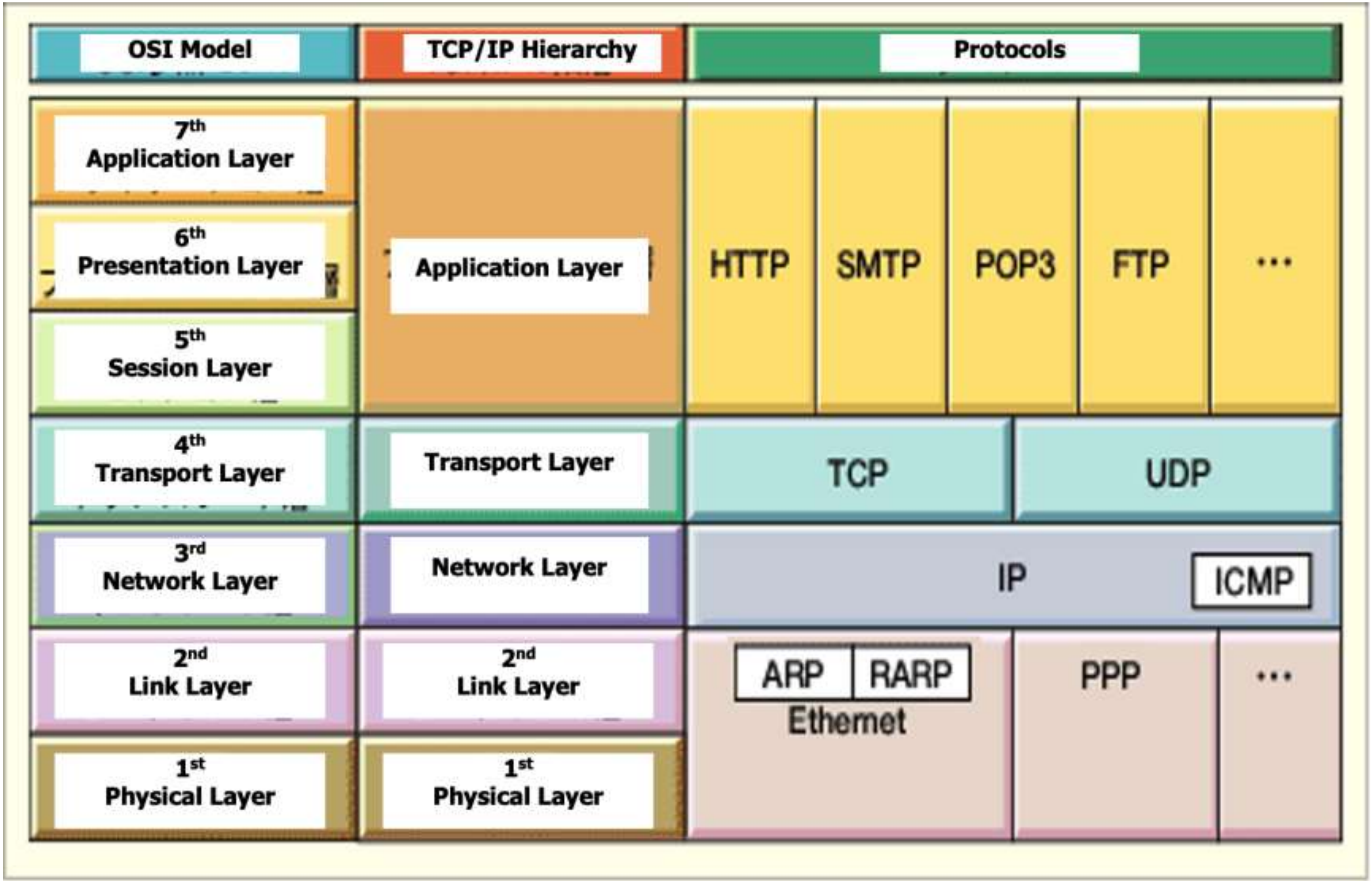
IETF mailing list 2-Feb-2001

"Huh?" (says the phone world)

TCP/IP Protocol Stack

- Internet is based on “TCP/IP protocol suite”.
  - TCP (Transmission Control Protocol of Transport layer)
  - IP (Internet Protocol of Network layer)

TCP/IP Protocol Stack



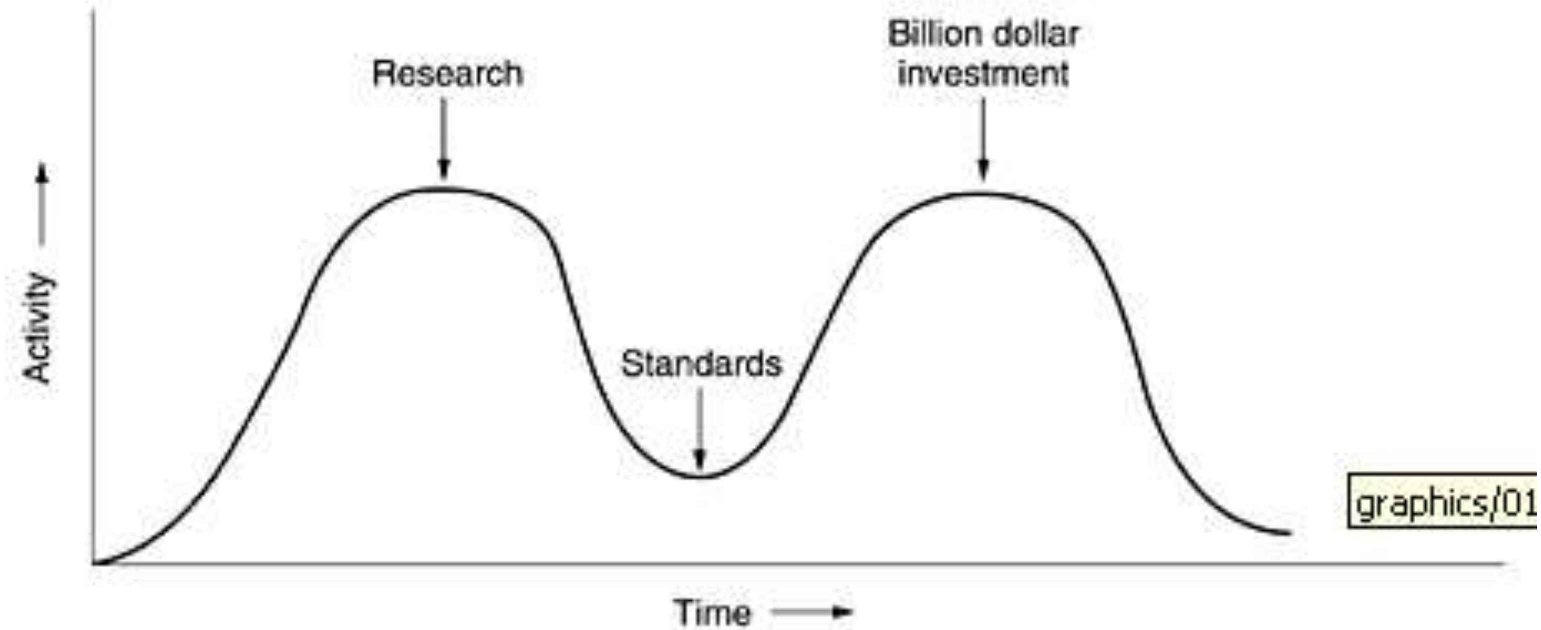
TCP/IP Protocol Stack

1. Link Layer : includes device driver and network interface card
2. Network Layer : handles the movement of packets, i.e. Routing
3. Transport Layer : provides a reliable flow of data between two hosts
4. Application Layer : handles the details of the particular application

A Critique of the OSI Model

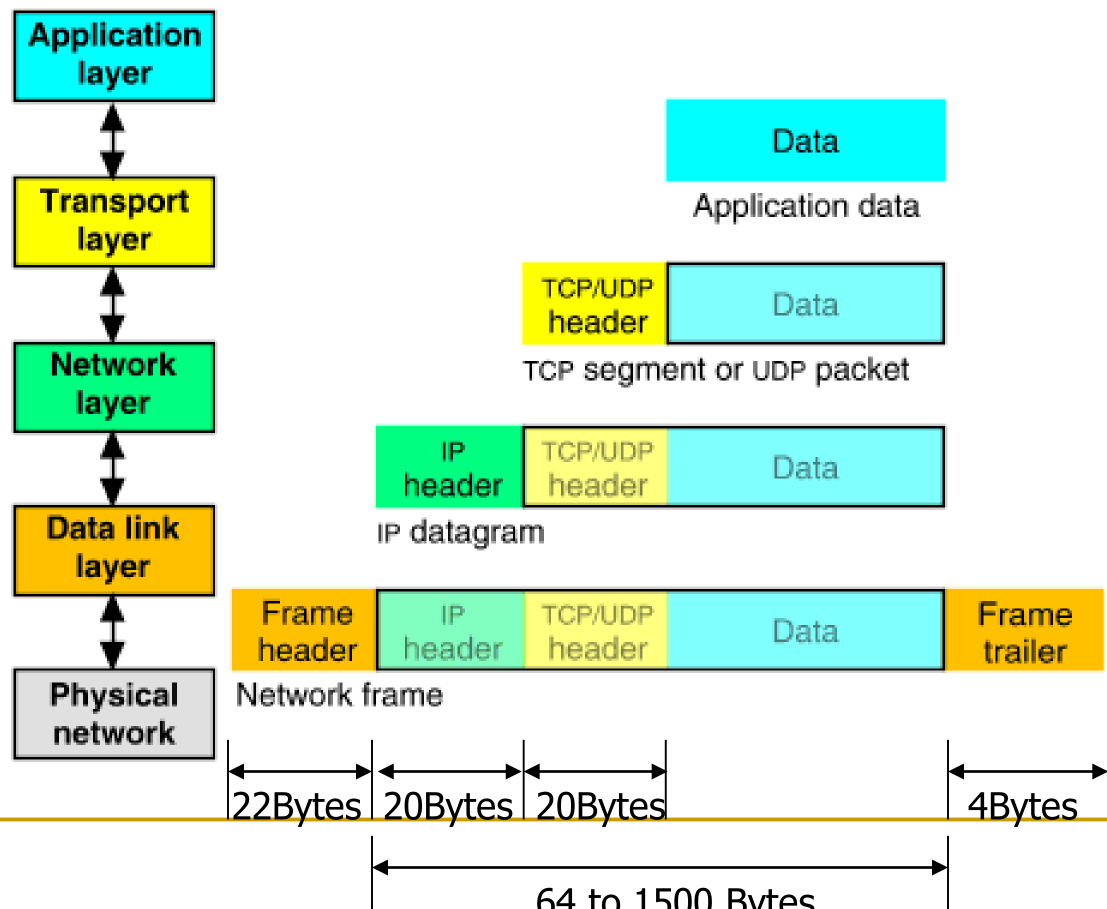
- Why OSI did not take over the world
  - Bad timing
  - Bad technology
  - Bad implementations
  - Bad politics

The apocalypse of the two elephants



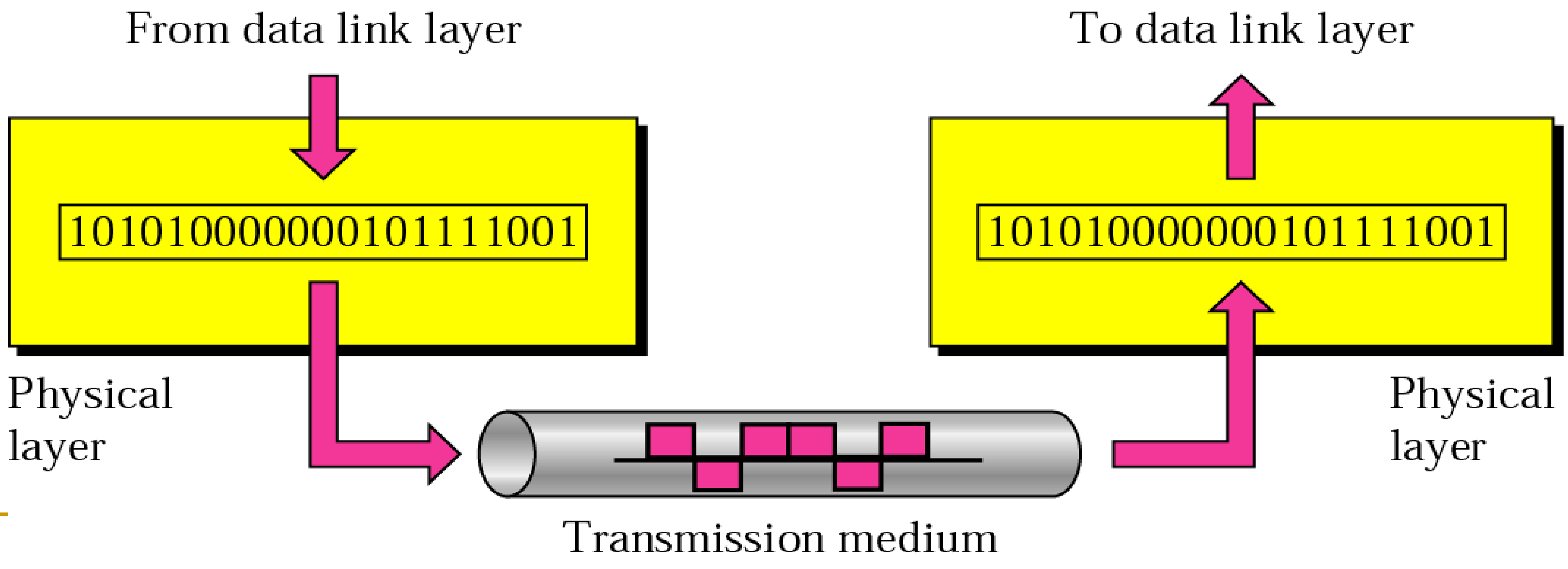
Data Encapsulation

- The data is sent down the protocol stack
- Each layer adds to the data by prepending headers



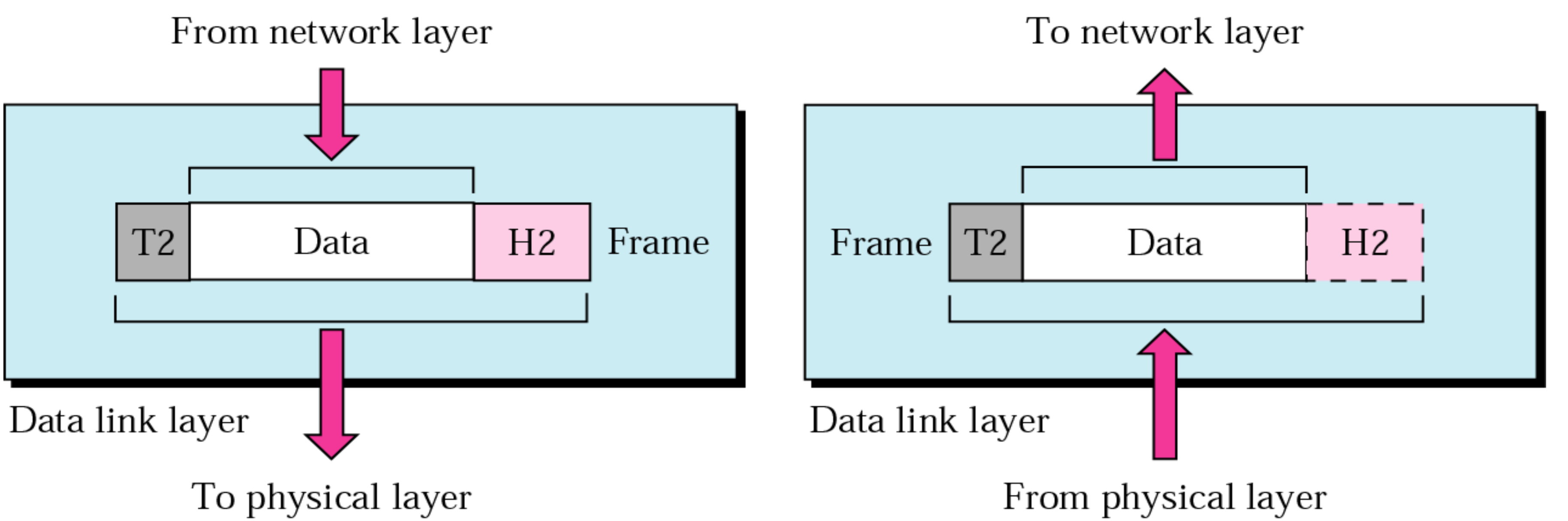
Physical layer

- Representation of bits
- Data Rate

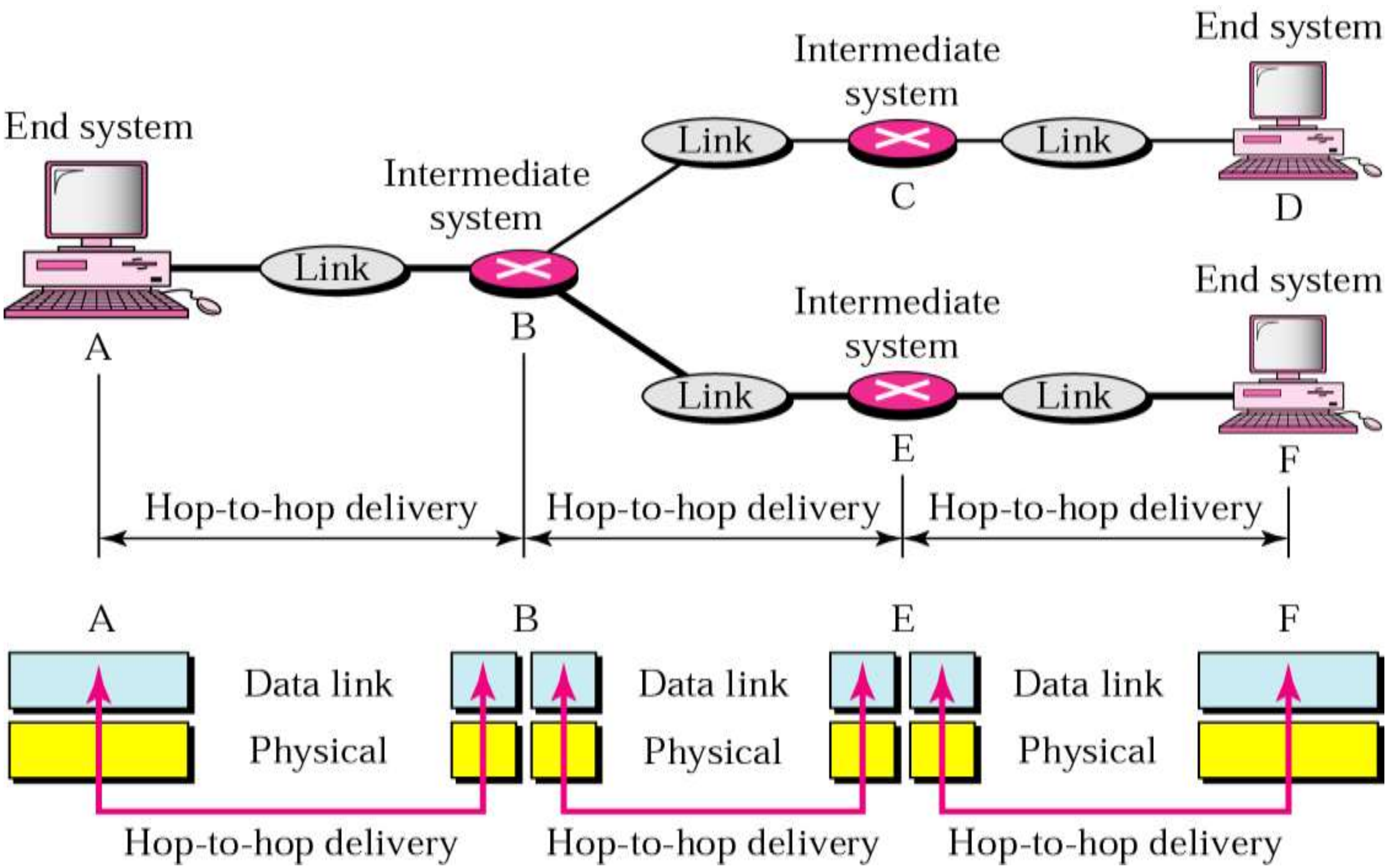


Data Link Layer

- Hop-to-Hop Delivery
- Framing
- Physical Addressing



Data Link Layer



Ethernet

- Computer <-> Computer communication on same network
- Each device has unique MAC address (48-bit)

  example: 00-C0-4F-48-47-93

Ethernet Frame:

Preamble	Dest. address	Source address	Type	Data	CRC
8bytes	**6bytes**	**6bytes**	**2bytes**	**64 - 1500bytes**	**4bytes**

MAC: Media Access Control

ARP : Address Resolution Protocol

- ARP provides mapping

  32bit IP address <-> 48bit MAC address

  128.97.89.153 <-> 00-C0-4F-48-47-93
- ARP cache

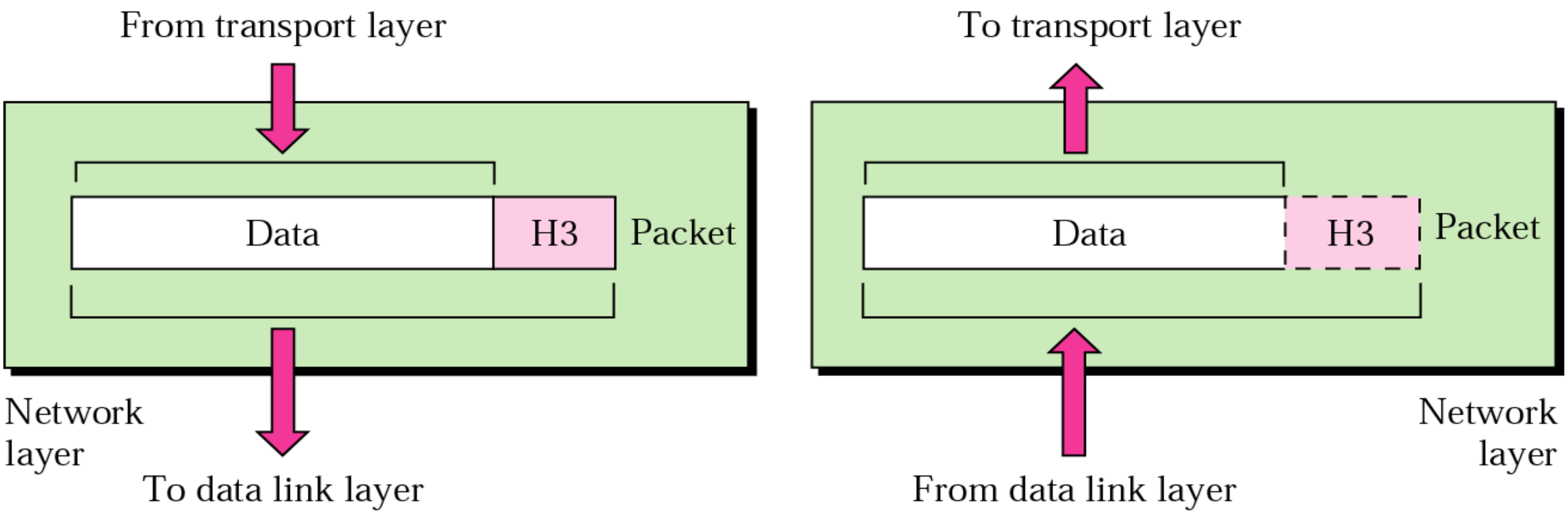
  maintains the recent mappings from IP addresses to MAC addresses

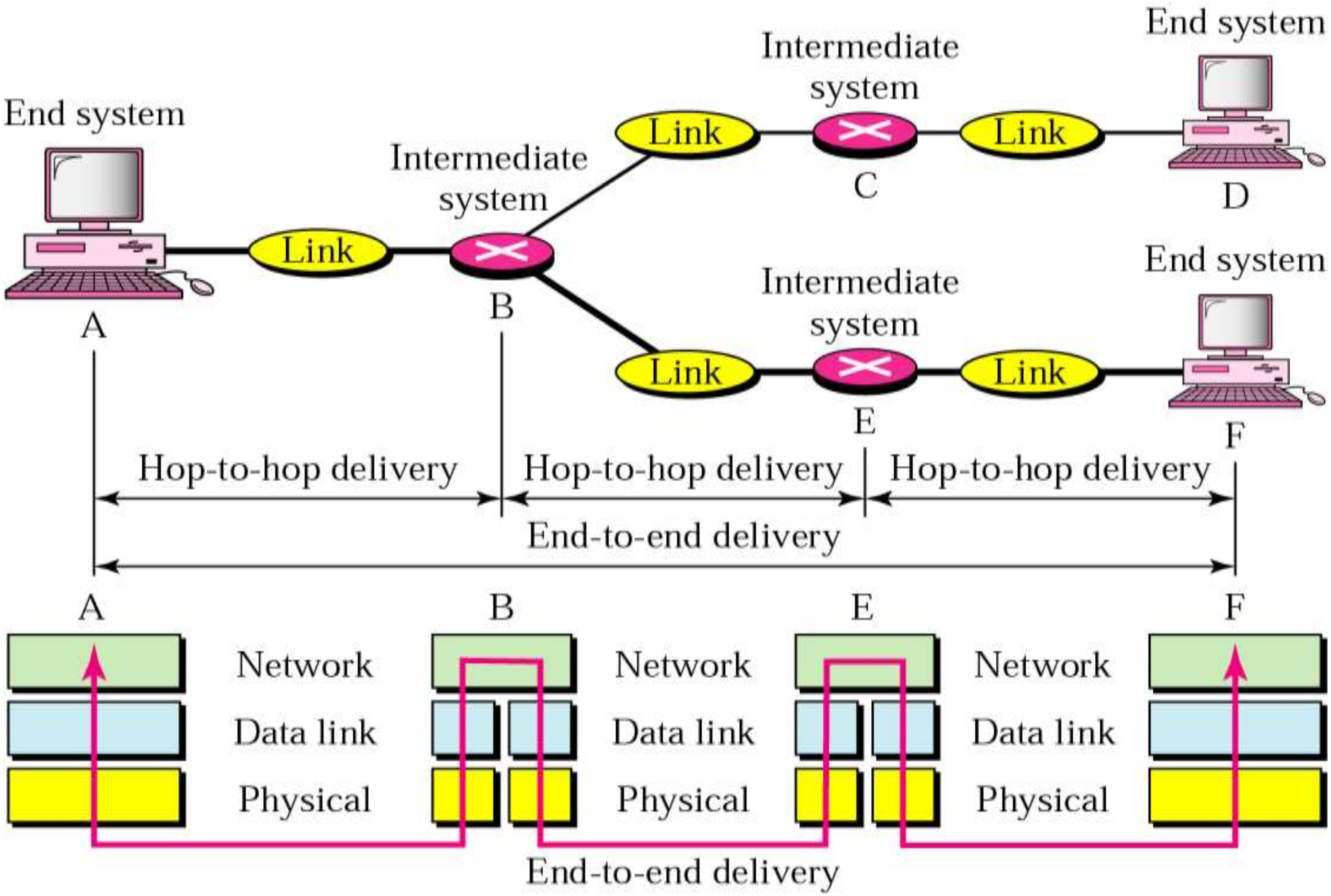
<u>Protocol</u>

1. ARP request broadcast on Ethernet
2. Destination host ARP layer responds

Network Layer

- End-to-End Delivery
- Logical Addressing
- Routing

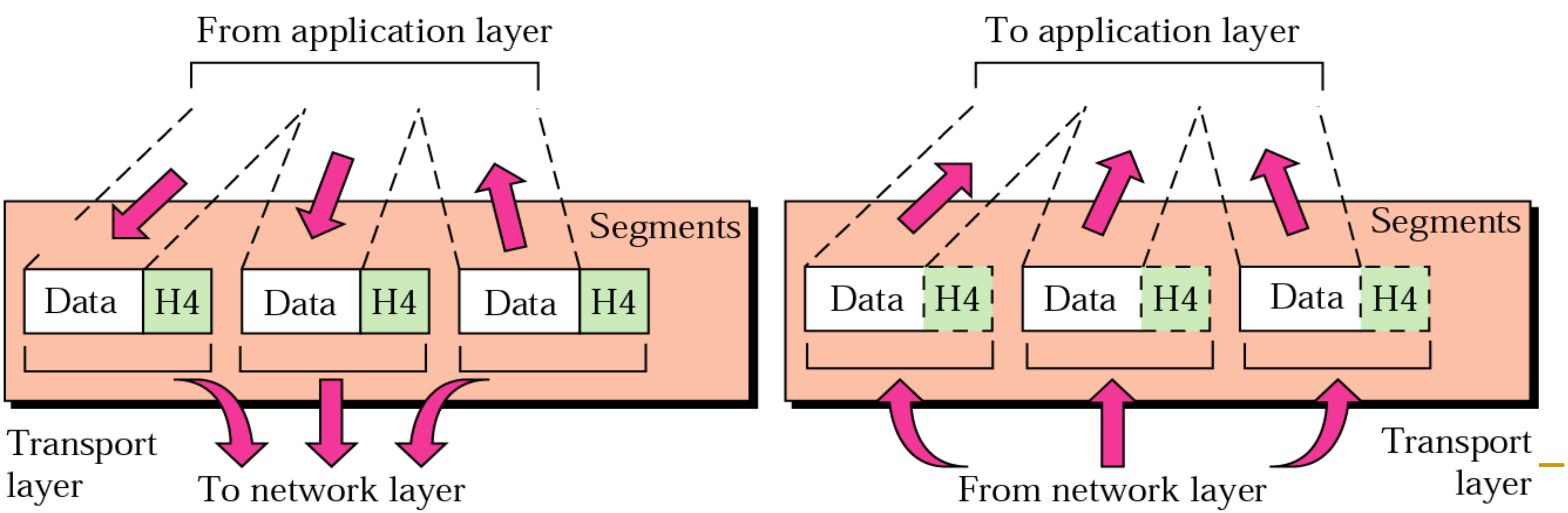


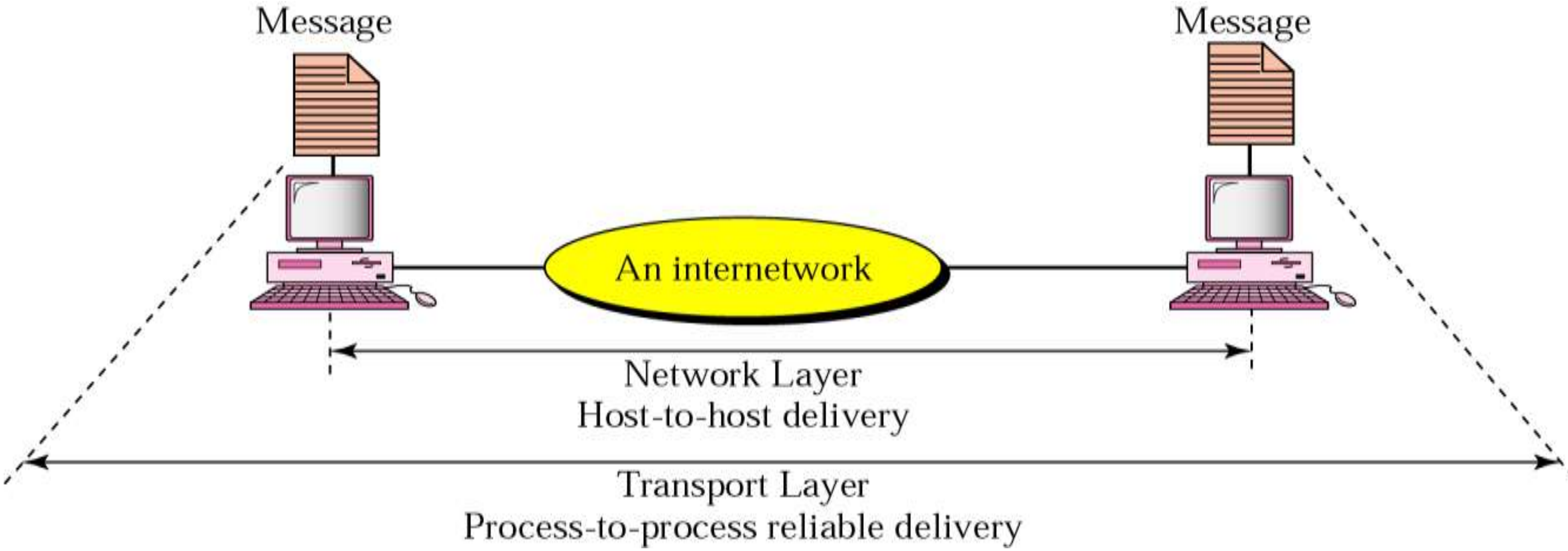


End system
Intermediate system
End system
Link
Link
Link
Intermediate system
C
D
Intermediate system
End system
A
B
Link
Link
E
F
Hop-to-hop delivery
Hop-to-hop delivery
Hop-to-hop delivery
End-to-end delivery
A
B
E
F
Network
Network
Network
Data link
Data link
Data link
Physical
Physical
Physical
End-to-end delivery

Transport Layer

- Process-to-Process delivery
- Port Addressing
- Reliable or Unreliable Services

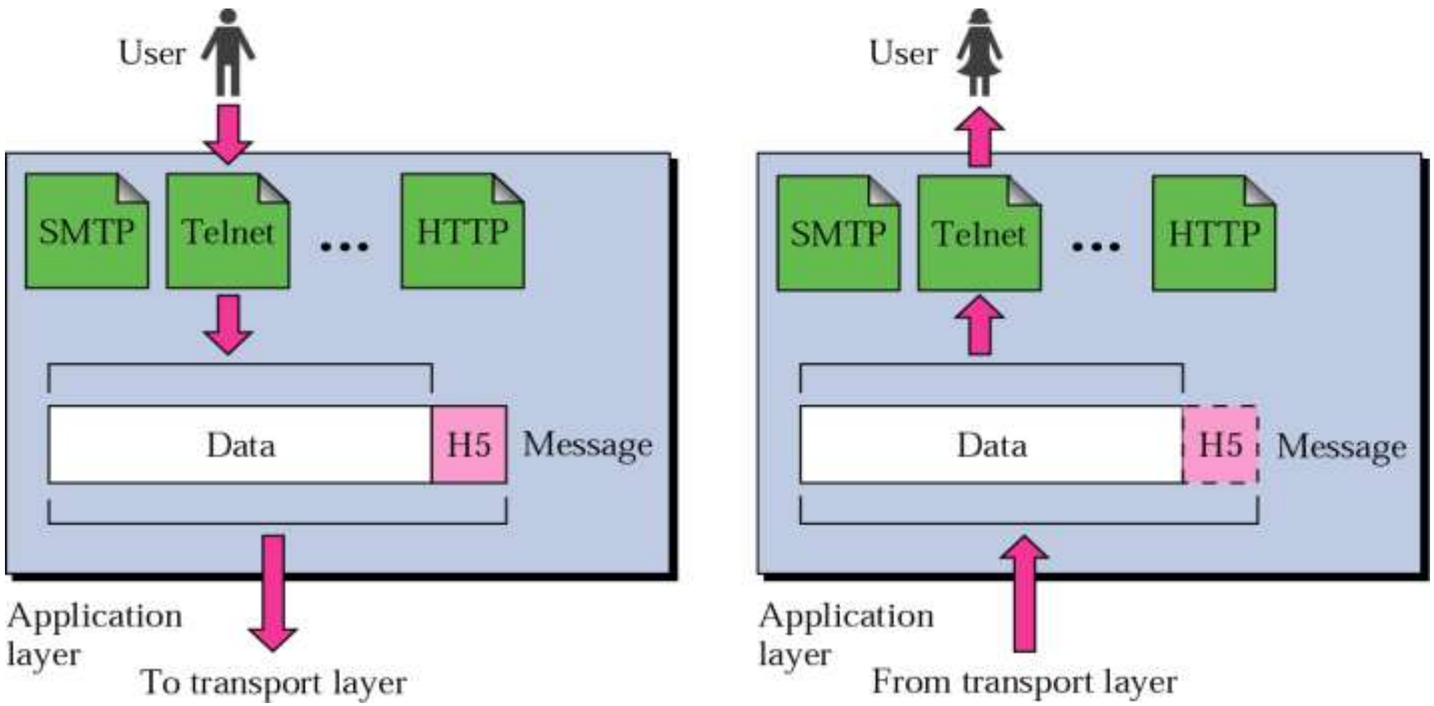




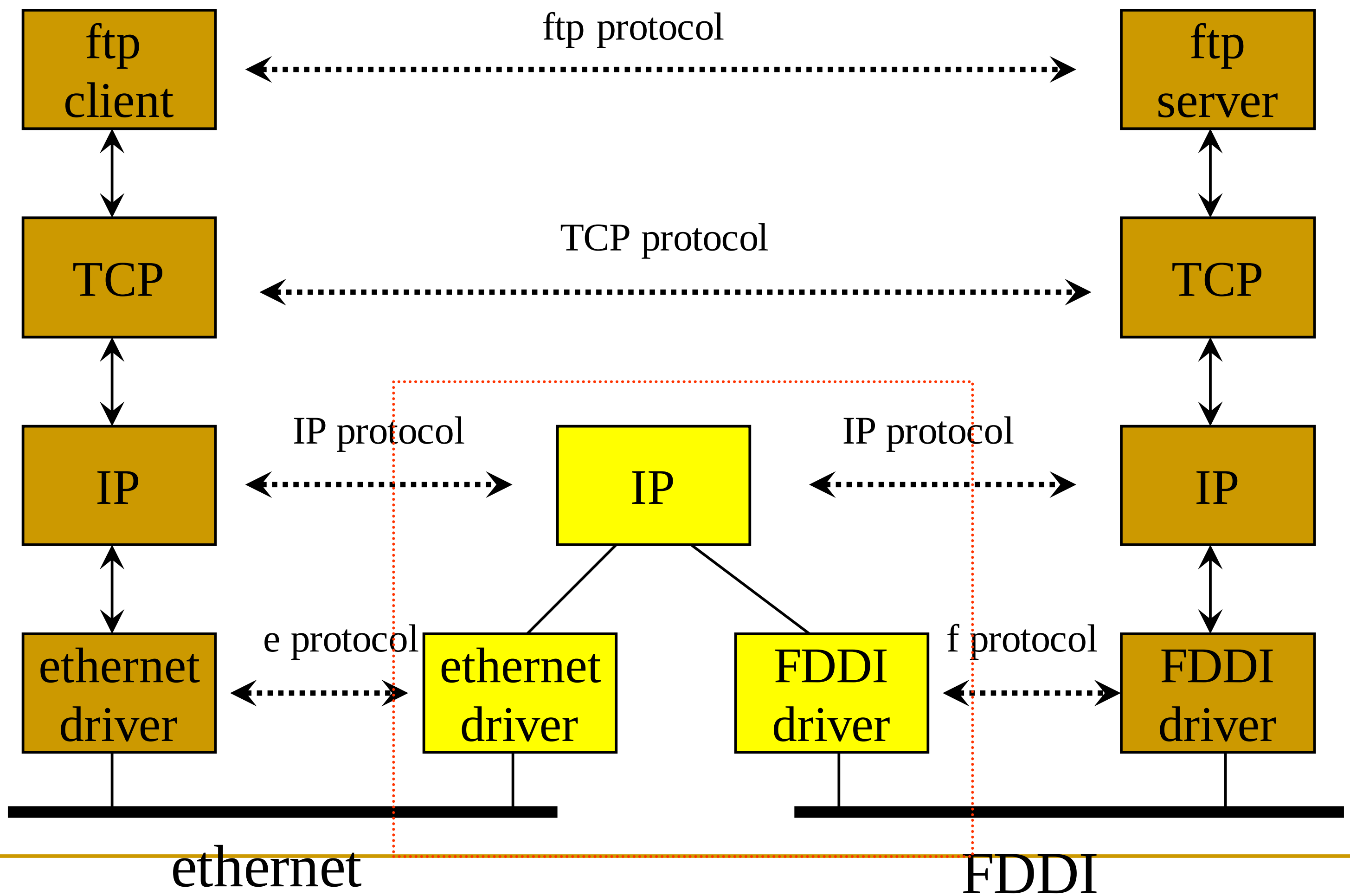
Message
Message
An internetwork
Network Layer
Host-to-host delivery
Transport Layer
Process-to-process reliable delivery

Application Layer

- Provide services to the user
  - Email (SMTP), File transfer (FTP, NFS), Remote Login(Telnet), Name manager(DNS), WWW(http) etc.



TCP/IP Architecture



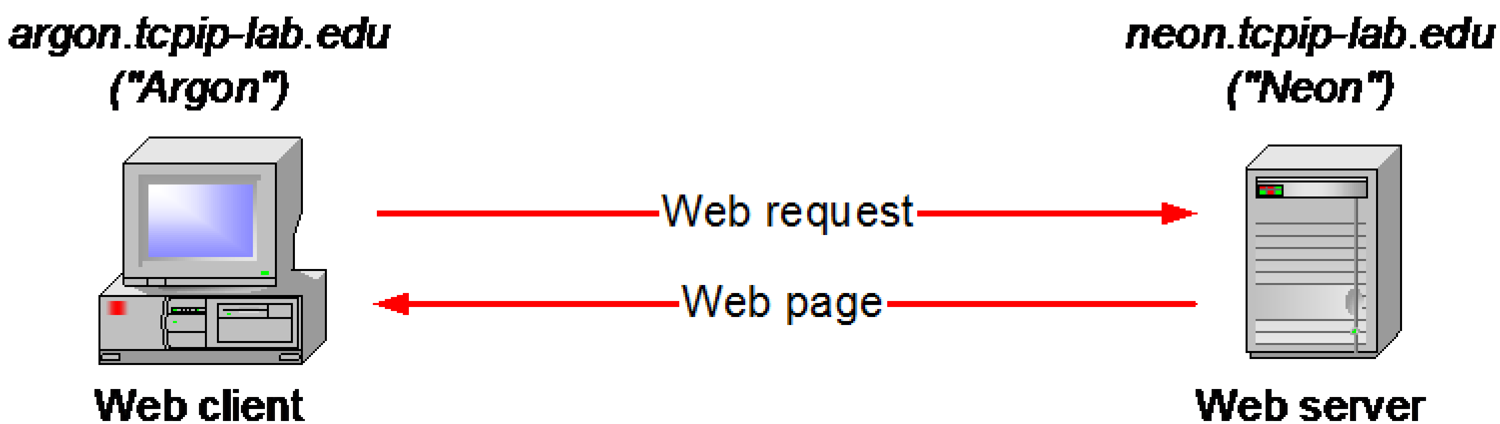
A simple TCP/IP example

- Source:
  - *www.cs.virginia.edu/~cs458/slides/module01-exampleV2.**ppt***

A simple TCP/IP Example

- A user on host *argon.tcpip-lab.edu* (“*Argon*”) makes a web access to URL

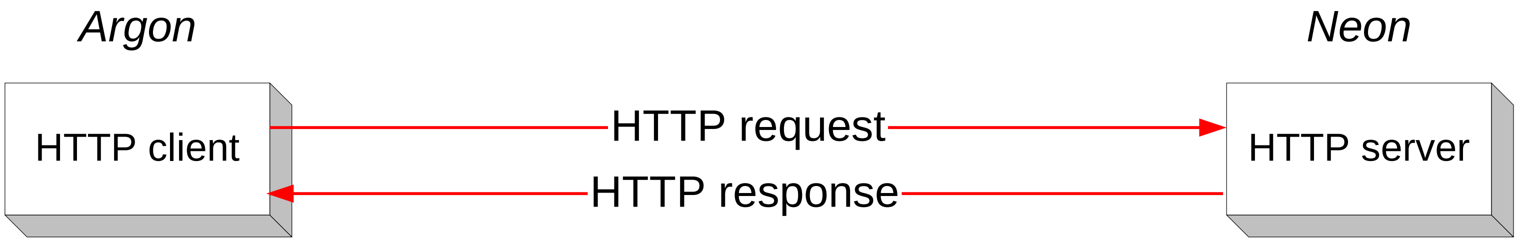
http://neon.tcpip-lab.edu/index.html.



- What actually happens in the network?

HTTP Request and HTTP response

- Web browser runs an HTTP client program
- Web server runs an HTTP server program
- HTTP client sends an HTTP request to HTTP server
- HTTP server responds with HTTP response



HTTP Request

```
GET /index.html HTTP/1.1
Accept: image/gif, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Host: neon.tcpip-lab.edu
Connection: Keep-Alive
```

HTTP Response

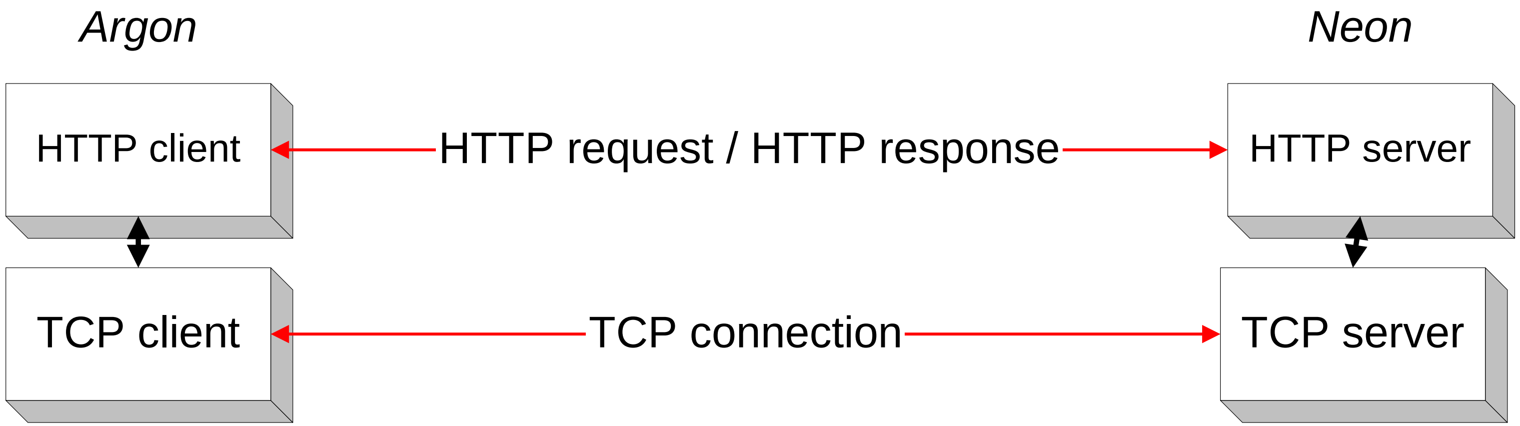
```
HTTP/1.1 200 OK
Date: Sat, 25 May 2002 21:10:32 GMT
Server: Apache/1.3.19 (Unix)
Last-Modified: Sat, 25 May 2002 20:51:33 GMT
ETag: "56497-51-3ceff955"
Accept-Ranges: bytes
Content-Length: 81
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<HTML>
<BODY>
<H1>Internet Lab</H1>
Click <a href="http://www.tcpip-
lab.net/index.html">here</a> for the Internet Lab webpage.
</BODY>
</HTML>
```

- How does the HTTP request get from Argon to Neon ?

From HTTP to TCP

- To send request, HTTP client program establishes an TCP connection to the HTTP server Neon.
- The HTTP server at Neon has a TCP server running



Resolving hostnames and port numbers

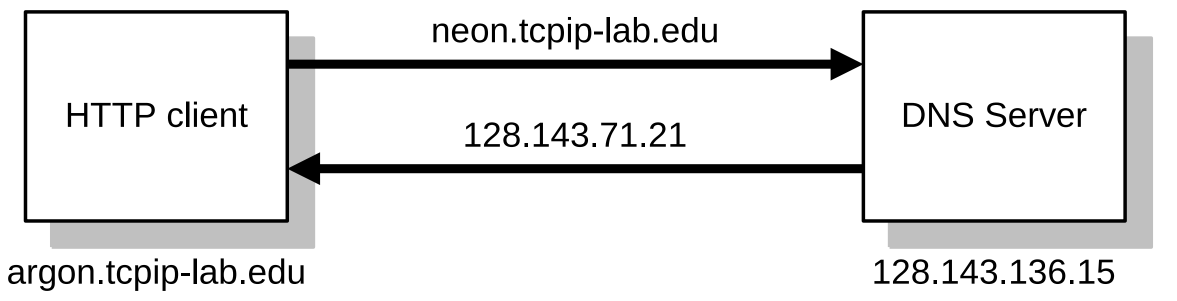
- Since TCP does not work with hostnames and also would not know how to find the HTTP server program at Neon, two things must happen:

    1. The name "neon.tcpip-lab.edu" must be translated into a 32-bit **IP address.**

    2. The HTTP server at Neon must be identified by a 16-bit **port number**.

Translating a hostname into an IP address

- The translation of the hostname *neon.tcpip-lab.edu* into an IP address is done via a database lookup



- The distributed database used is called the ***Domain Name System (DNS)***
- All machines on the Internet have an IP address:

  argon.tcpip-lab.edu *128.143.137.144*

  neon.tcpip-lab.edu *128.143.71.21*

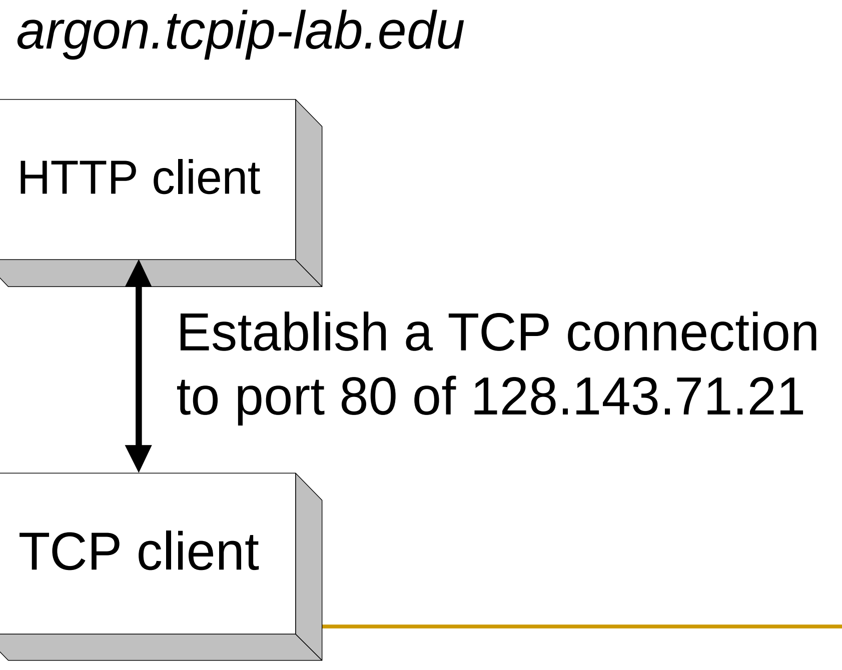
Finding the port number

- **Note:** Most services on the Internet are reachable via well-known ports. E.g. All HTTP servers on the Internet can be reached at port number “80”.
- **So:** Argon simply <u>knows</u> the port number of the HTTP server at a remote machine.
- On most Unix systems, the well-known ports are listed in a file with name /etc/services. The well-known port numbers of some of the most popular services are:

ftp	21	finger	79
telnet	23	http	80
smtp	25	nntp	119

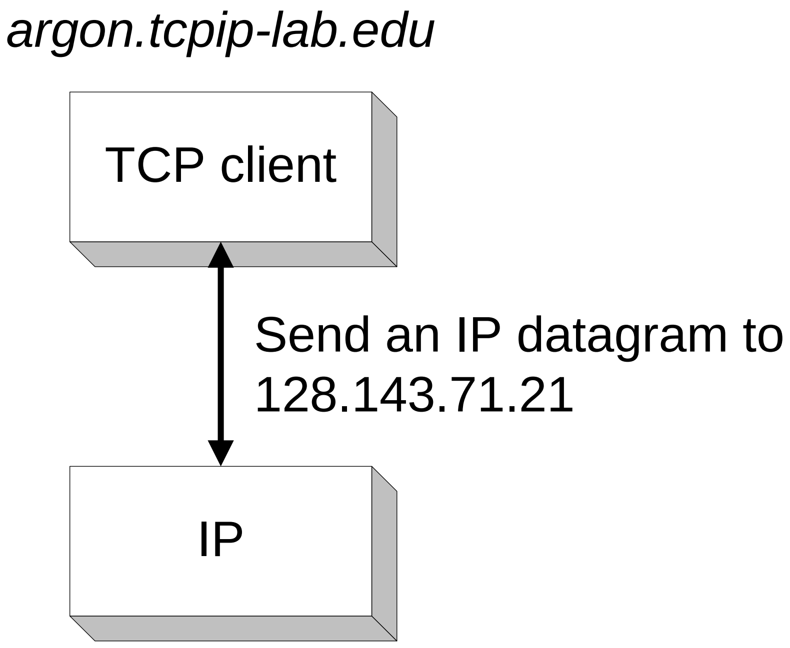
Requesting a TCP Connection

- The HTTP client at *argon.tcpip-lab.edu* requests the TCP client to establish a connection to port 80 of the machine with address 128.141.71.21



Invoking the IP Protocol

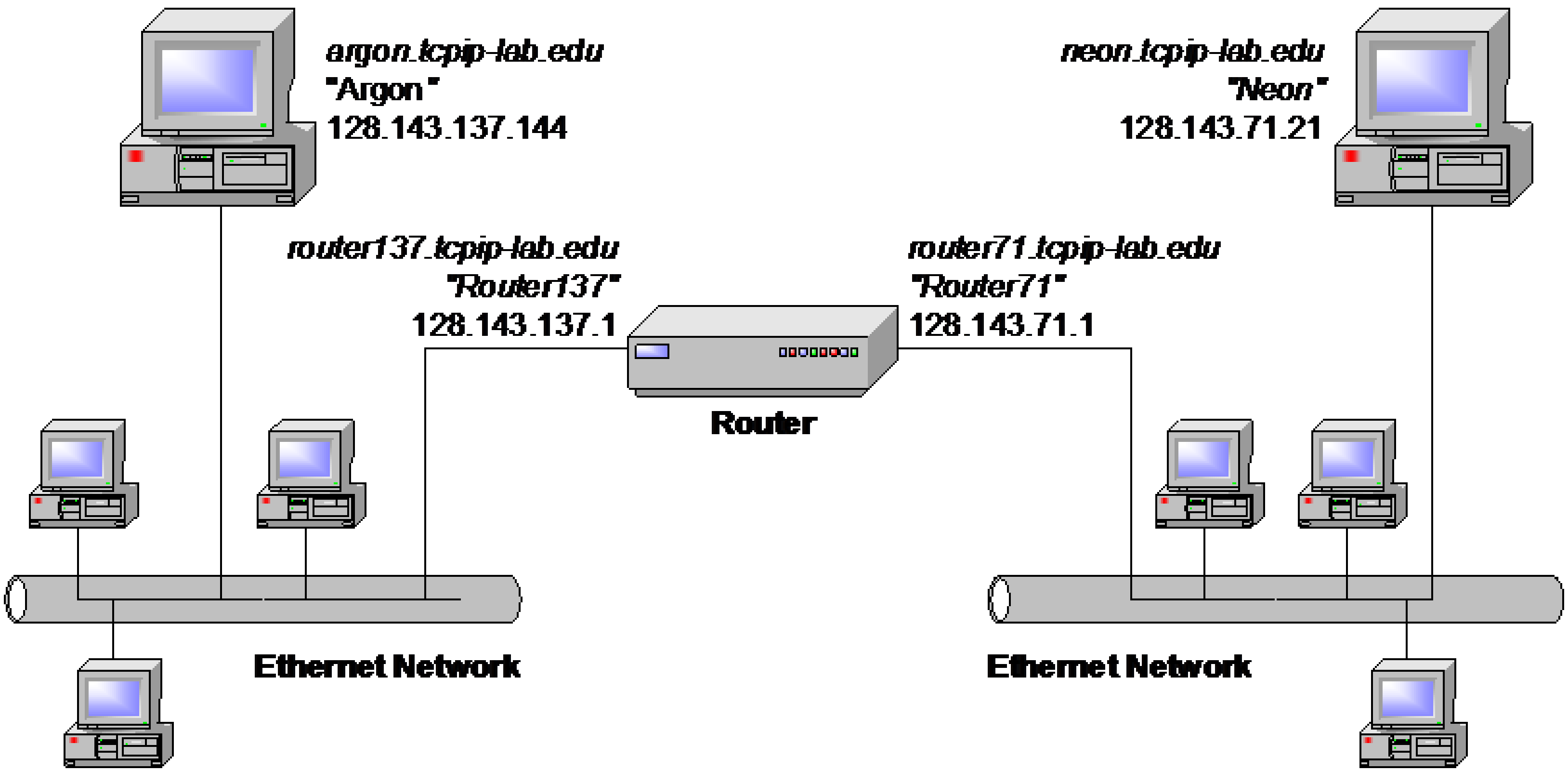
- The TCP client at *Argon* sends a request to establish a connection to port 80 at *Neon*
- This is done by asking its local IP module to send an IP datagram to *128.143.71.21*
- *(The data portion of the IP datagram contains the request to open a connection)*



Sending the IP datagram to an IP router

- *Argon (128.143.137.144)* can deliver the IP datagram directly to *Neon (128.143.71.21*), only if it is on the same local network ("subnet")
- But *Argon* and N*eon* are <u>not</u> on the same local network
  (**Q**: How does *Argon* know this?)
- So, *Argon* sends the IP datagram to its default gateway
- The default gateway is an IP router
- The default gateway for *Argon* is *Router137.tcpip-lab.edu* (128.143.137.1).

The route from ***Argon*** to ***Neon***

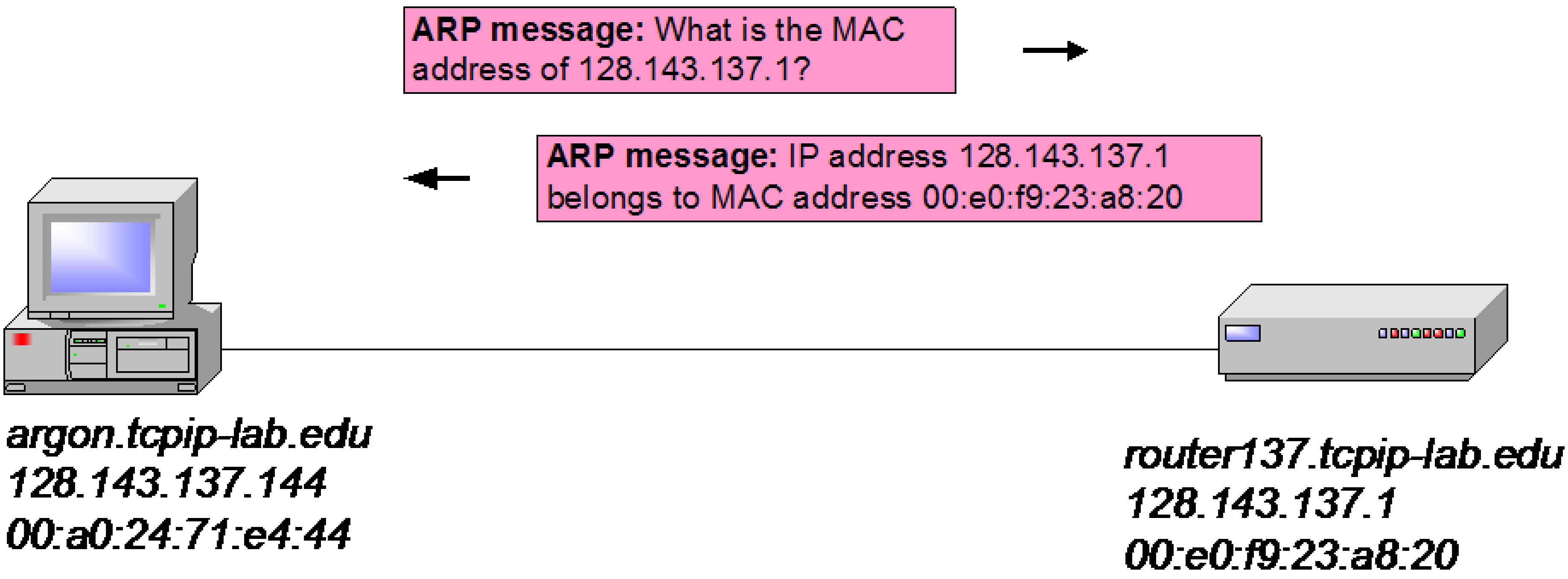


- Note that the gateway has a different name for each of its interfaces.

Finding the MAC address of the gateway

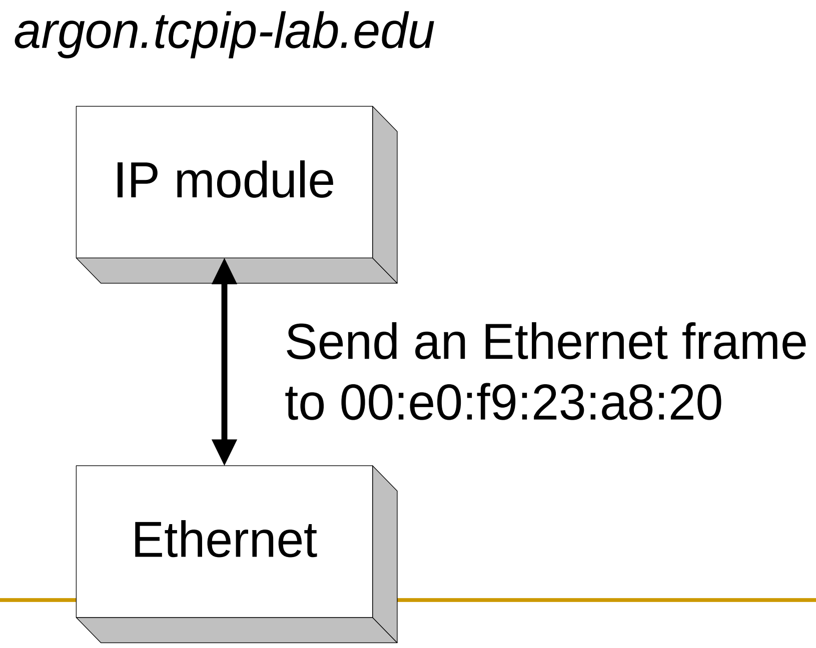
- To send an IP datagram to Router137, *Argon* puts the IP datagram in an Ethernet frame, and transmits the frame.
- However, Ethernet uses different addresses, so-called Media Access Control (MAC) addresses (also called: physical address, hardware address)
- Therefore, *Argon* must first translate the IP address 128.143.137.1 into a MAC address.
- The translation of addressed is performed via the Address Resolution Protocol (ARP)

Address resolution with ARP



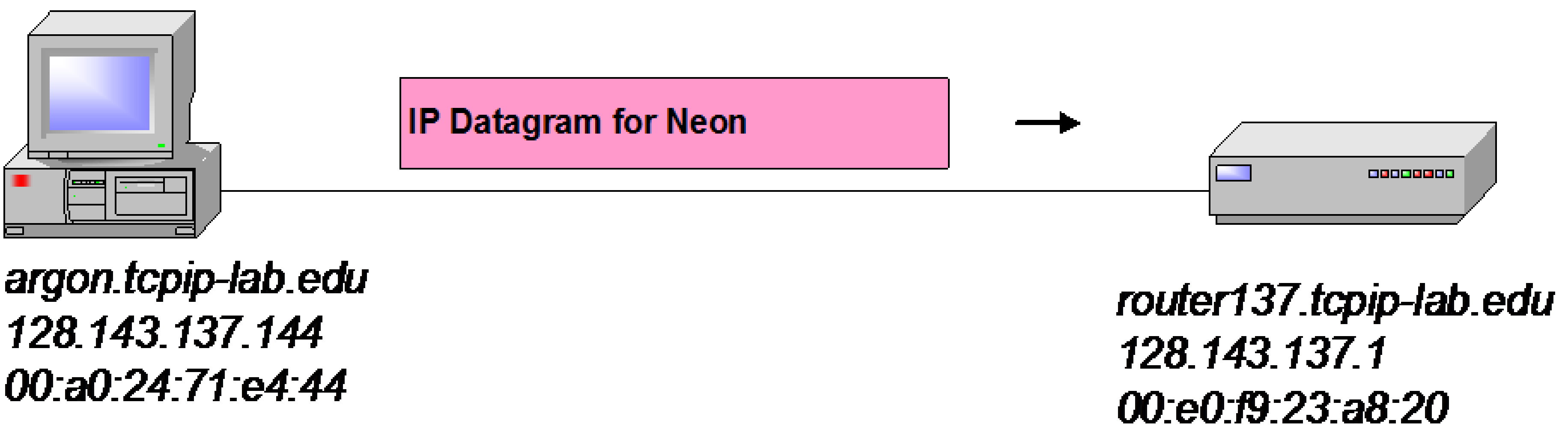
Invoking the device driver

- The IP module at *Argon*, tells its Ethernet device driver to send an Ethernet frame to address *00:e0:f9:23:a8:20*



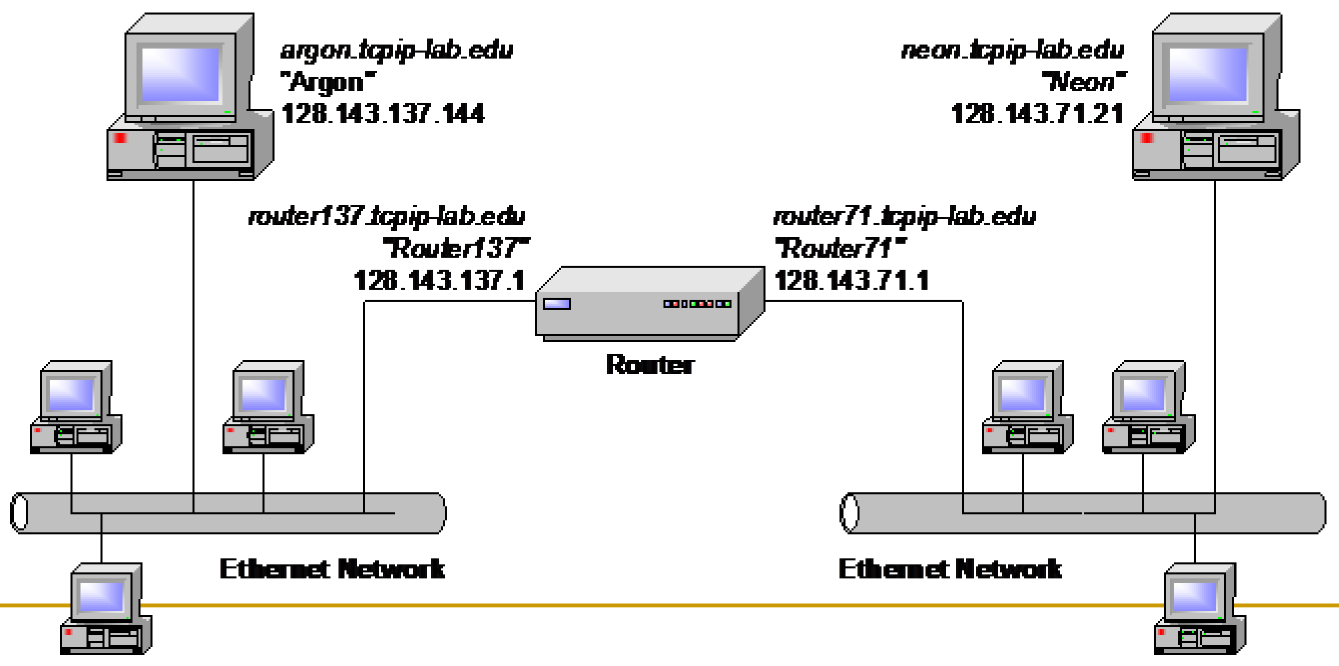
Sending an Ethernet frame

- The Ethernet device driver of *Argon* sends the Ethernet frame to the Ethernet network interface card (NIC)
- The NIC sends the frame onto the wire



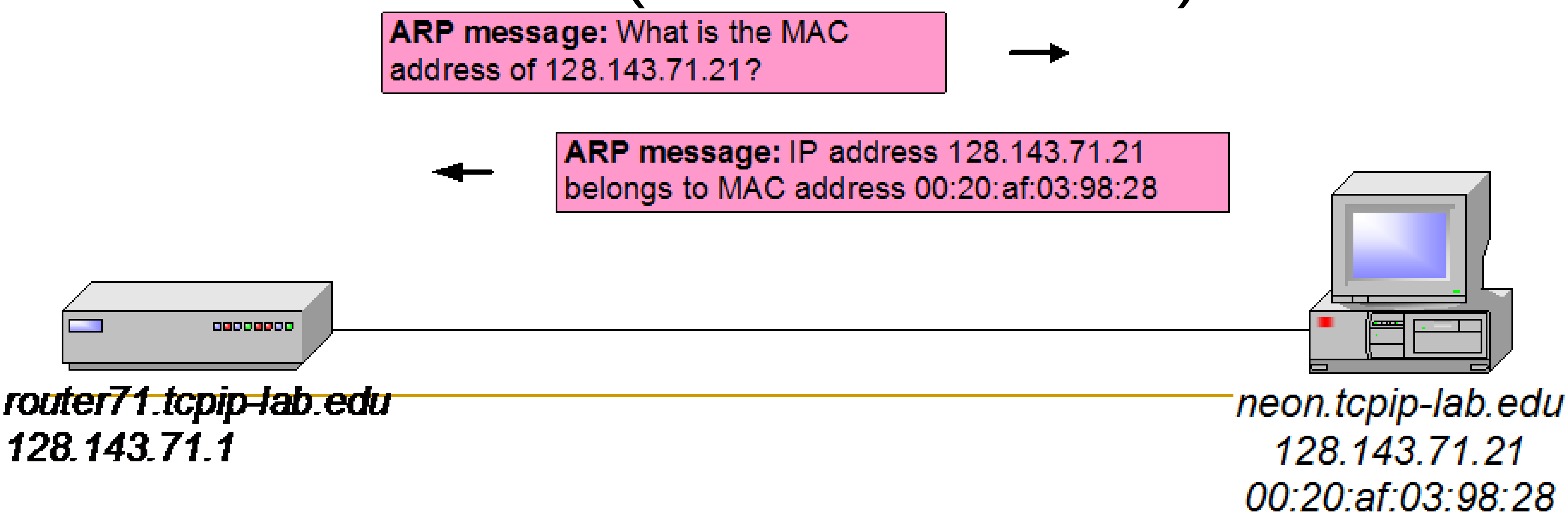
Forwarding the IP datagram

- The IP router receives the Ethernet frame at interface 128.143.137.1, recovers the IP datagram and determines that the IP datagram should be forwarded to the interface with name 128.143.71.1
- The IP router determines that it can deliver the IP datagram directly



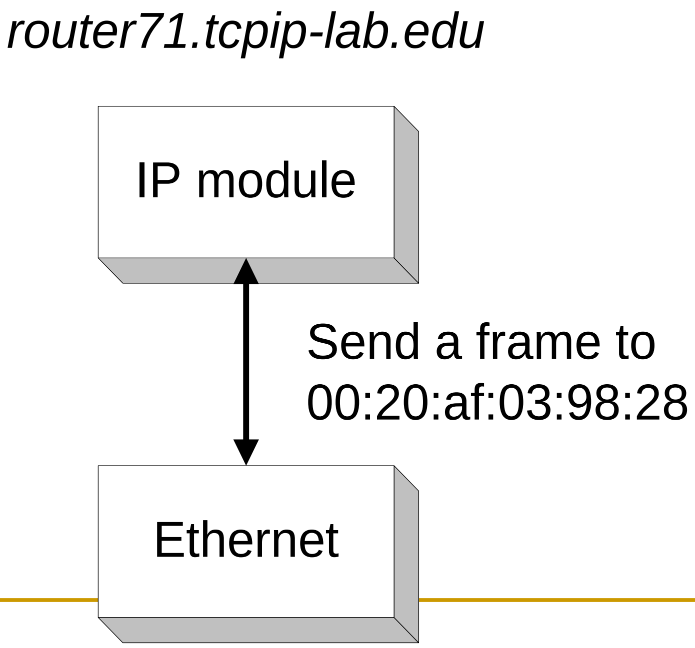
Another lookup of a MAC address

- The rouer needs to find the MAC address of *Neon*.
- Again, ARP is invoked, to translate the IP address of *Neon* (128.143.71.21) into the MAC address of neon (00:20:af:03:98:28).



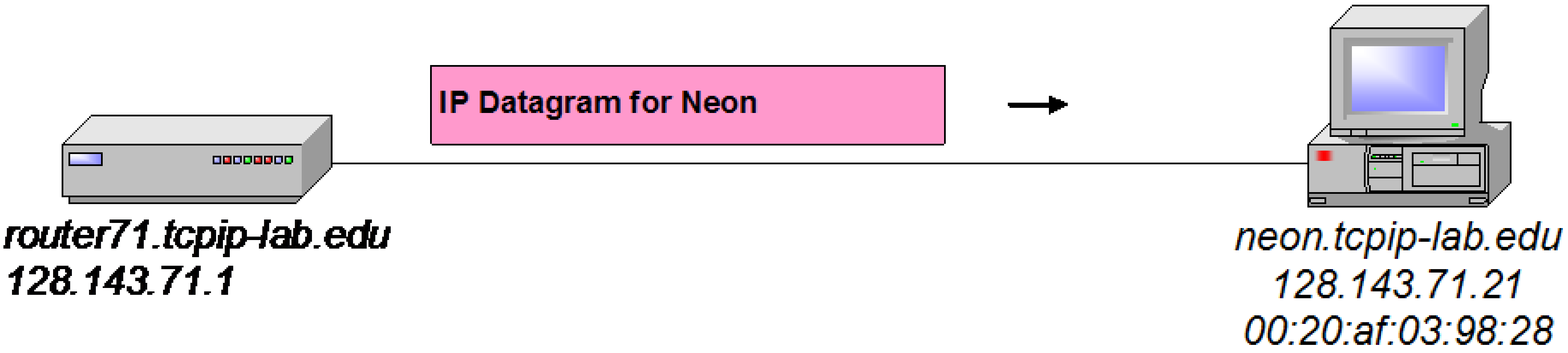
Invoking the device driver at the router

- The IP protocol at *Router71*, tells its Ethernet device driver to send an Ethernet frame to address *00:20:af:03:98:28*



Sending another Ethernet frame

- The Ethernet device driver of *Router71* sends the Ethernet frame to the Ethernet NIC, which transmits the frame onto the wire.



Data has arrived at Neon

- *Neon* receives the Ethernet frame
- The payload of the Ethernet frame is an IP datagram which is passed to the IP protocol.
- The payload of the IP datagram is a TCP segment, which is passed to the TCP server
- **Note**: Since the TCP segment is a connection request (SYN), the TCP protocol does not pass data to the HTTP program for this packet. Instead, the TCP protocol at neon will respond with a SYN segment to *Argon*.