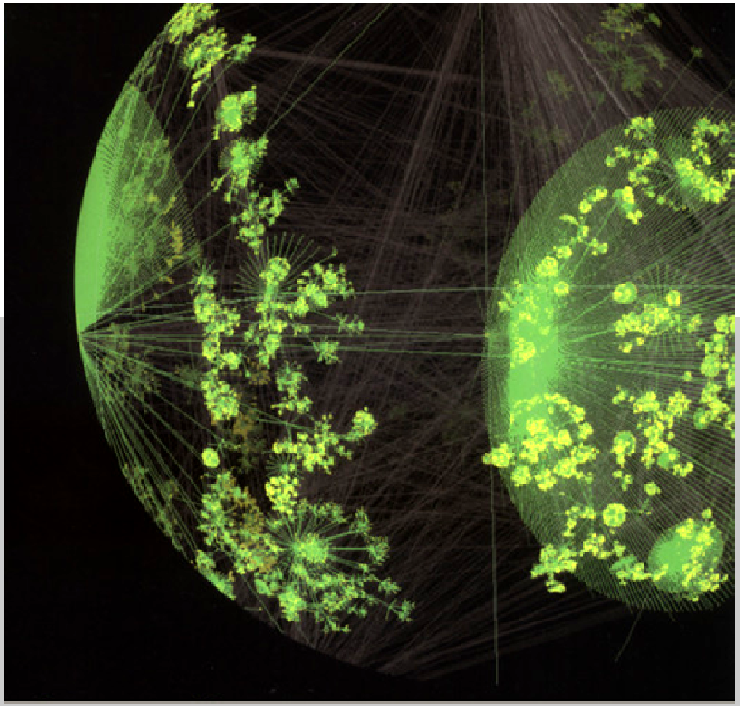
Chapter 0
TCP/IP Overview



TCP/IP Essentials

A Lab-Based Approach

Fall 2015

The Internet

A global information system consisting of millions of private and public, academic, business, and government “computer” networks around the world.

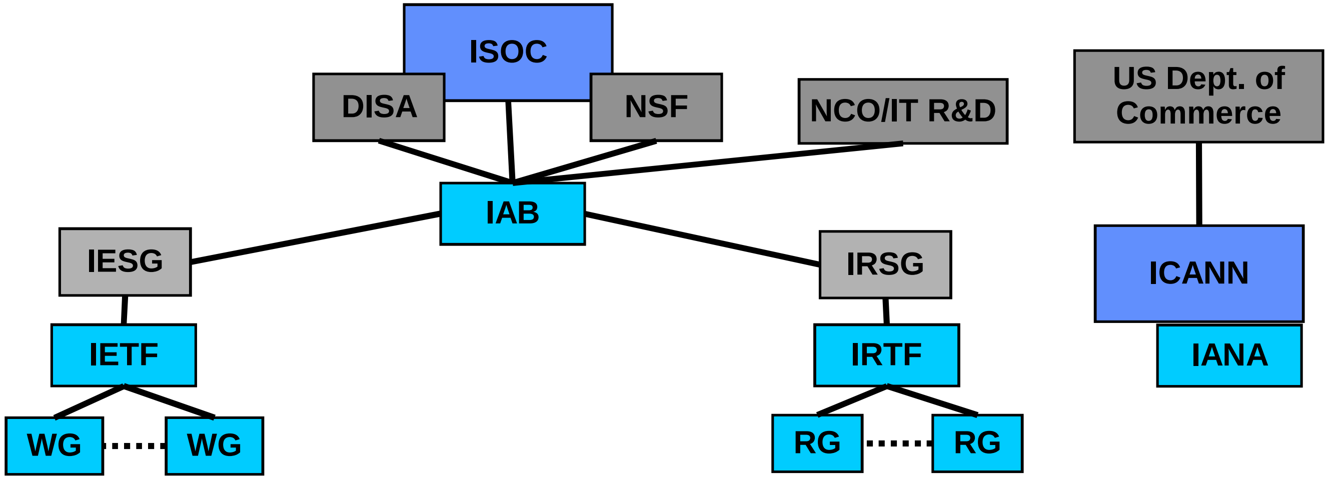
The Internet carries a vast array of information resources and services.

- Users: individuals and organizations in more than 200 countries
- Applications: email, web page, access to remote computer
- Services: broadband Internet access, on-line banking, shopping, social networking

Internet History

- Late 1960s, ARPA sponsors the development of a packet-switching network, called the ARPANET.
- 1974, The TCP/IP protocols and model are proposed by Cerf and Kahn.
- 1983, ARPANET adopts TCP/IP. At this time, the ARPANET has 200 routers.
- 1984, NSF funds a TCP/IP based backbone network. This backbone grows into the NSFNET, which becomes the successor of the ARPANET.
- 1995, NSF stops funding the NSFNET. The Internet is completely commercial.
- 1996, TCP/IP in Windows and Windows NT.
- 2000, conversion to IPv6 standardized.

Internet Standard Control



- Internet SOCiety (ISOC): a professional membership organization that comments on policies, practices, and oversees others dealing with network policy issues.
- Internet Architecture Board (IAB): responsible for defining overall architecture of the Internet with guidance and broad direction to the IETF.
- Internet Engineering Task Force (IETF): responsible for protocol engineering and development.
- Internet Research Task Force (IRTF): responsible for focused, long-term research.
- Internet Corporation for Assigned Names and Numbers (ICANN): in charge of defining all “unique parameters” in the Internet, including domain names and IP addresses.

Internet Standardization Process

A typical (but not only) way of standardization is:

- Internet Drafts
- RFC (Request for Comments)
- Proposed Standard
- Draft Standard (requires 2 working implementation)
- Internet Standard (declared by IAB)

But

> *We reject: kings, presidents, and voting. We believe in: rough consensus and running code.*
>
> *– David Clark, MIT, 1992*

RFCs & FYIs

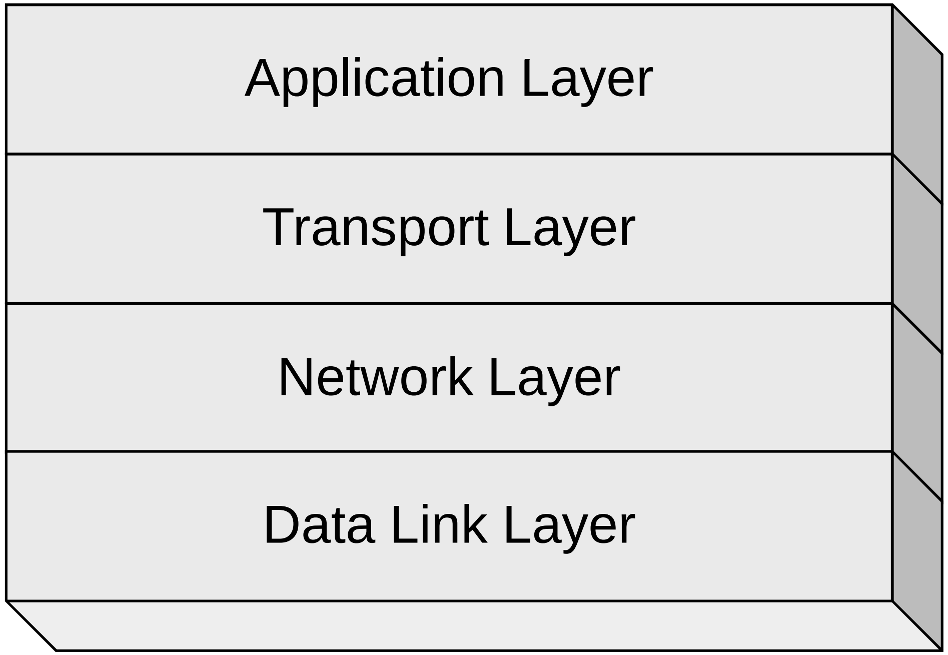
- All standards of the Internet are published as RFC. But not all RFCs are Internet Standards!
- Each new and revised/replacement RFC is assigned a sequential number, and falls in one the six categories:
  - Standards
  - Draft standard
  - Proposed standards
  - Experimental
  - Historical site protocols
  - Informational RFCs
- IETF decides a separate index structure, FYIs, for an important list of RFCs
- Equipment vendors may still implement an Internet draft even before RFC gets issued

Hierarchical Communication Architecture

- Networking can be quite complex and requires a high degree of cooperation between the involved parties.
- Cooperation is achieved by forcing parties to adhere to a set of rules and conventions (protocols).
- The complexity of the communication task is reduced by using multiple protocol layers:
  - Each protocol is implemented independently.
  - Each protocol is responsible for a specific subtask.
  - Protocols are grouped in a hierarchy.
- A structured set of protocols is called a communications architecture or protocol suite.

TCP/IP Protocols

- Internet Protocol Suite
- A combination of different protocols
- Organized into four layers



Internet – An FNC's Definition

On October 24, 1995, the Federal Networking Council* (FNC) unanimously passed a resolution defining the term Internet.

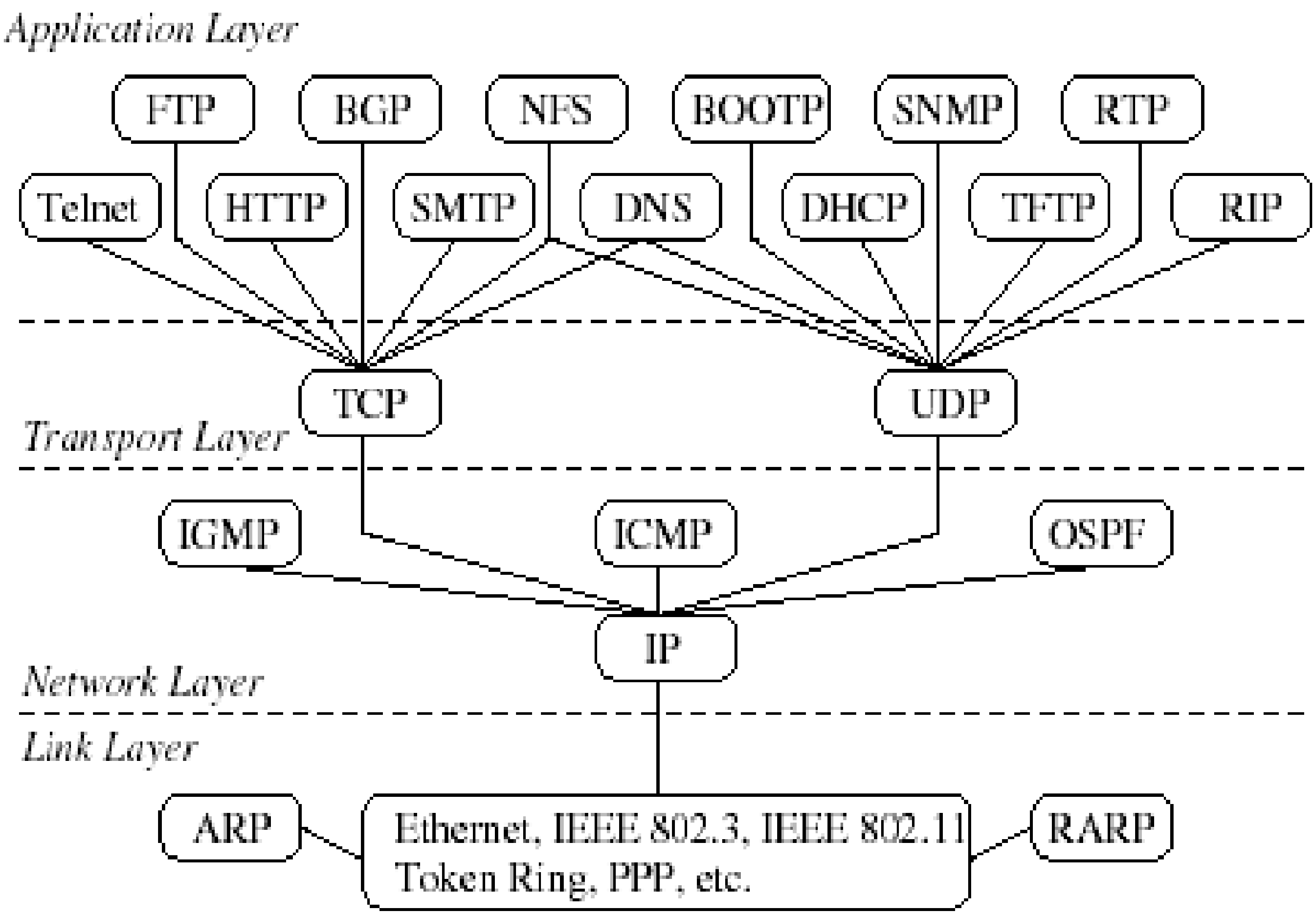
RESOLUTION: "Internet" refers to the global information system that

- is logically linked together by a globally unique address space based on the Internet Protocol (IP) or its subsequent extensions/ follow-ons;
- is able to support communications using the Transmission Control Protocol/Internet Protocol (TCP/IP) suite or its subsequent extensions/follow-ons, and/or other IP-compatible protocols; and
- provides, uses or makes accessible, either publicly or privately, high level services layered on the communications and related infrastructure described herein.

Functions of the Layers

- Application Layer: HTTP, telnet, DNS, SNMP, DHCP
  - Service: Handles details of application programs.
  - Functions: Everything is application specific.
- Transport Layer: UDP, TCP
  - Service: Controls delivery of data between hosts.
  - Functions: Connection establishment/termination, error control, flow control.
- Network Layer: IP, ICMP, IGMP
  - Service: Moves packets inside the network.
  - Functions: Routing, addressing, switching, congestion control.
- Data Link Layer: Ethernet, Wi-Fi, PPP, ARP, etc
  - Service: Reliable transfer of frames over a link.
  - Functions: Synchronization, error control, flow control.

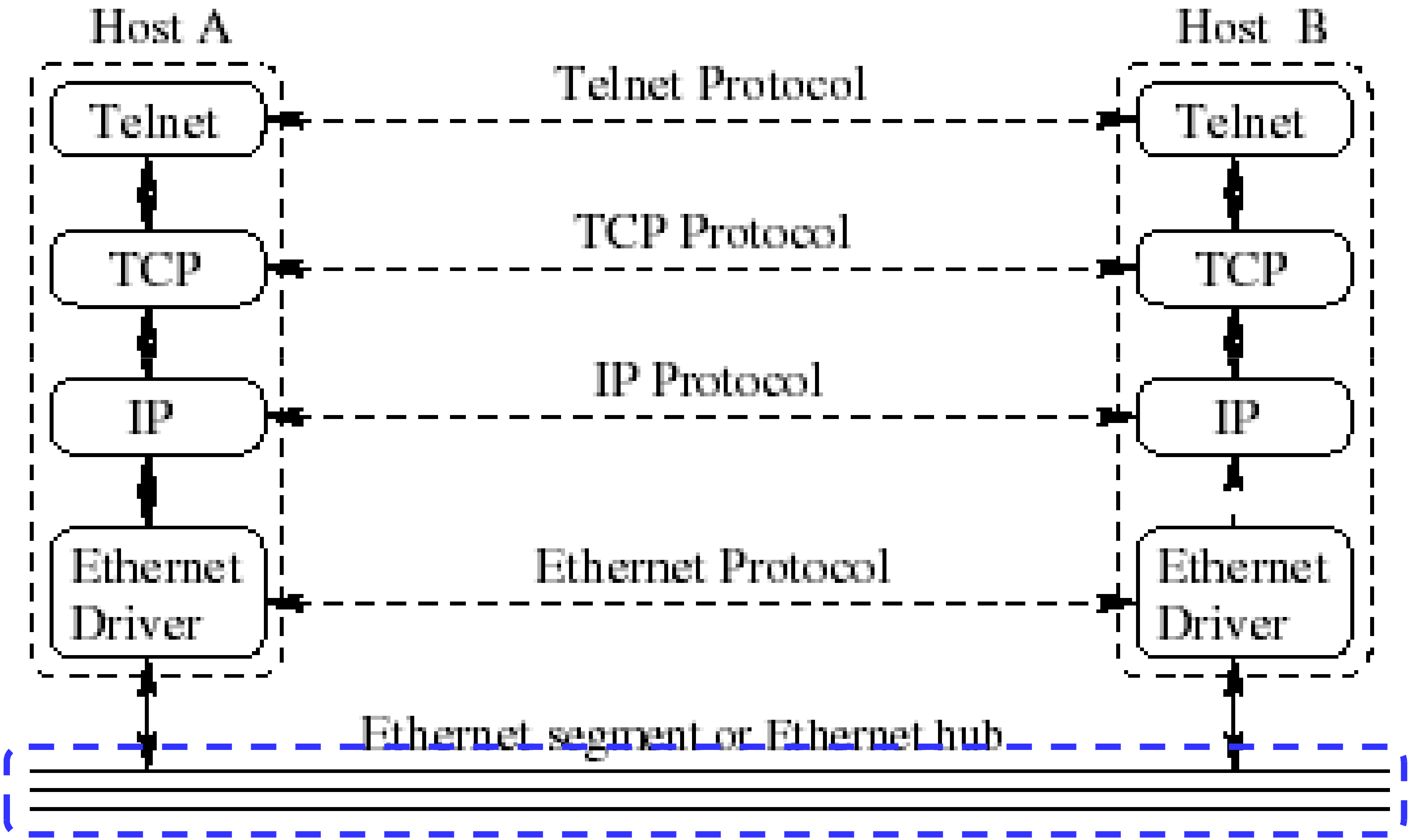
Protocols in Different Layers



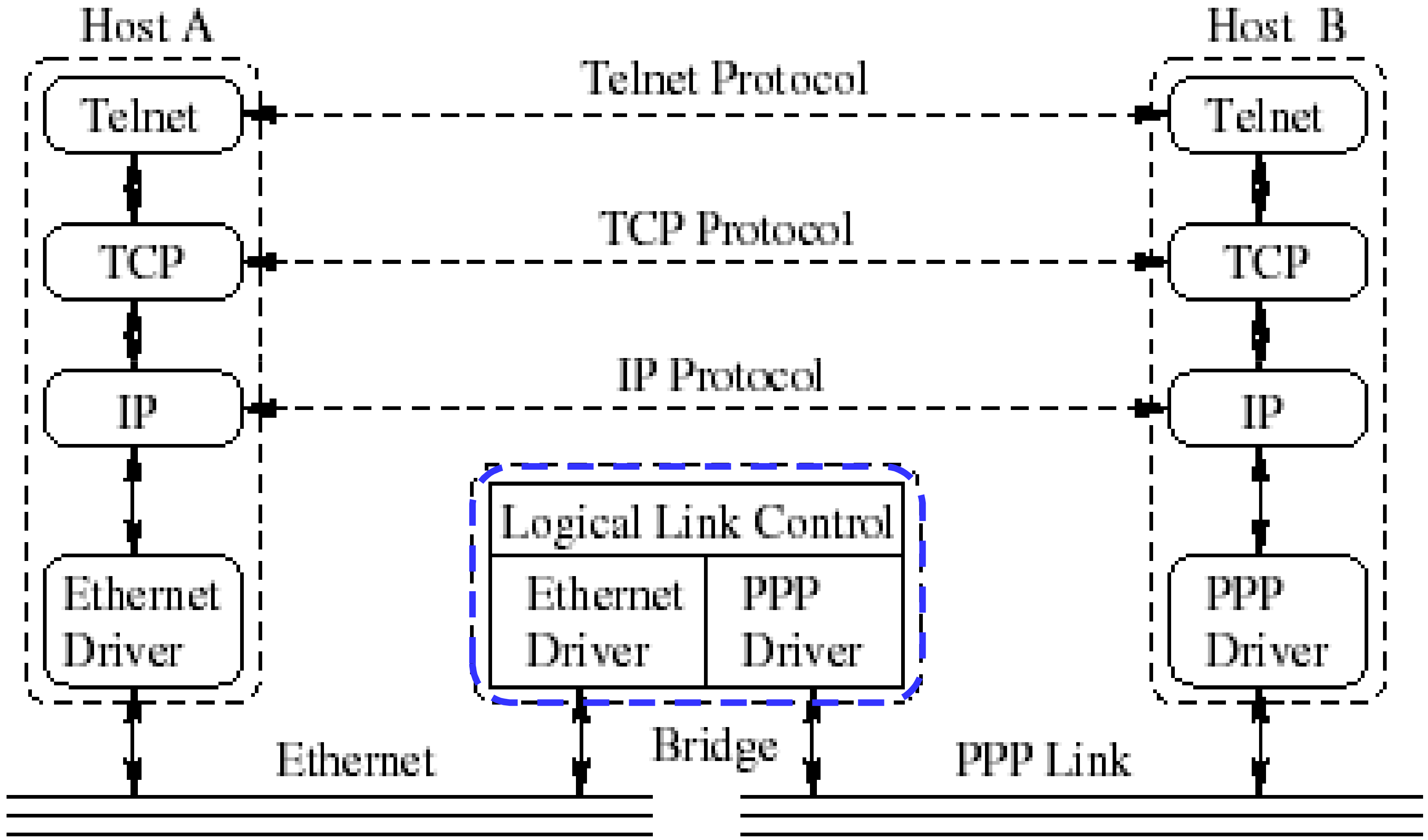
Internetworking Devices

- Computers are connected by internetworking devices.
- Classified according to their functionality and layers
  - –Hub
  - –Bridge
  - –Switch
  - –Router
  - –Gateway

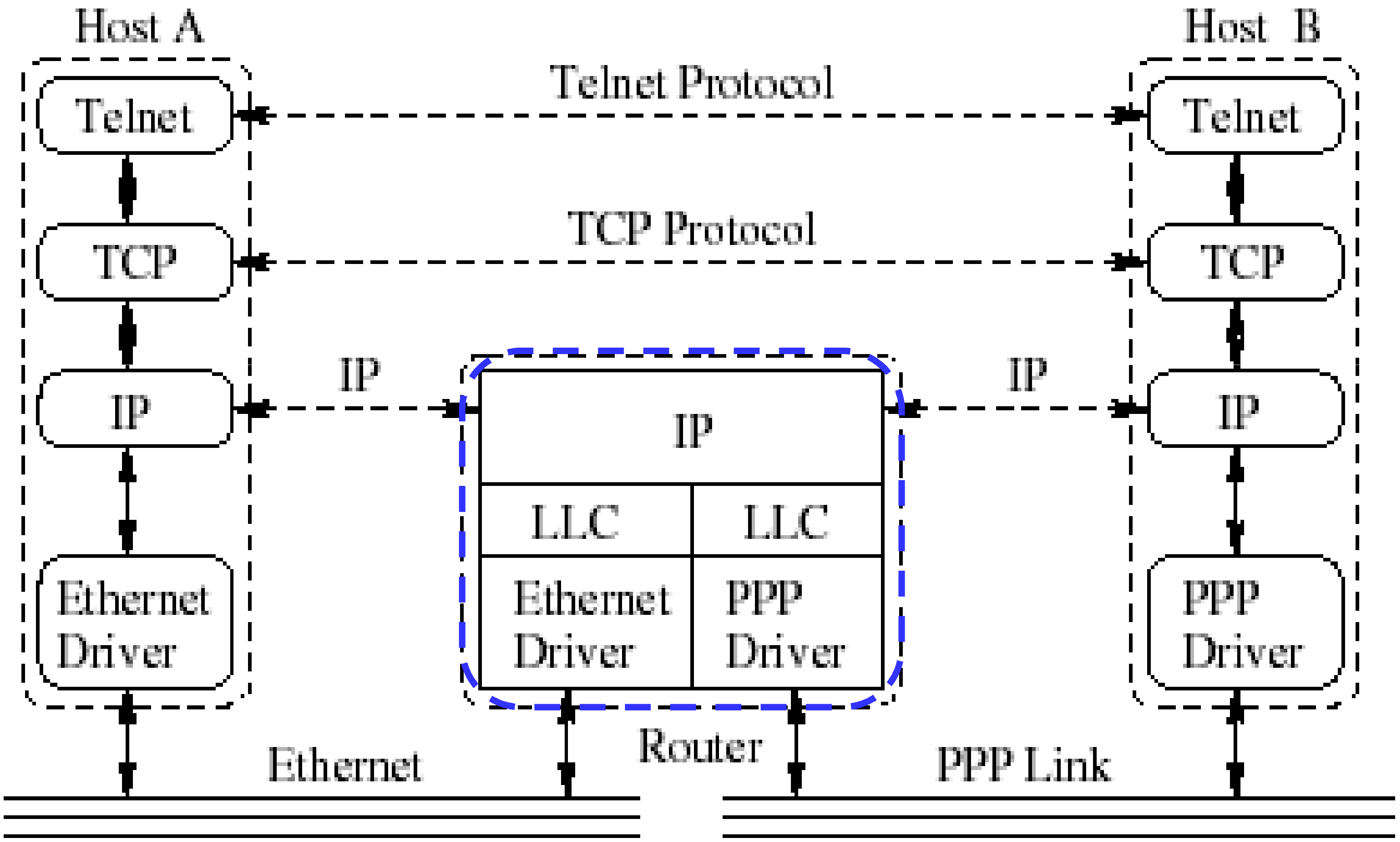
Hosts Communicating over a Hub



Hosts Communicating over a bridge

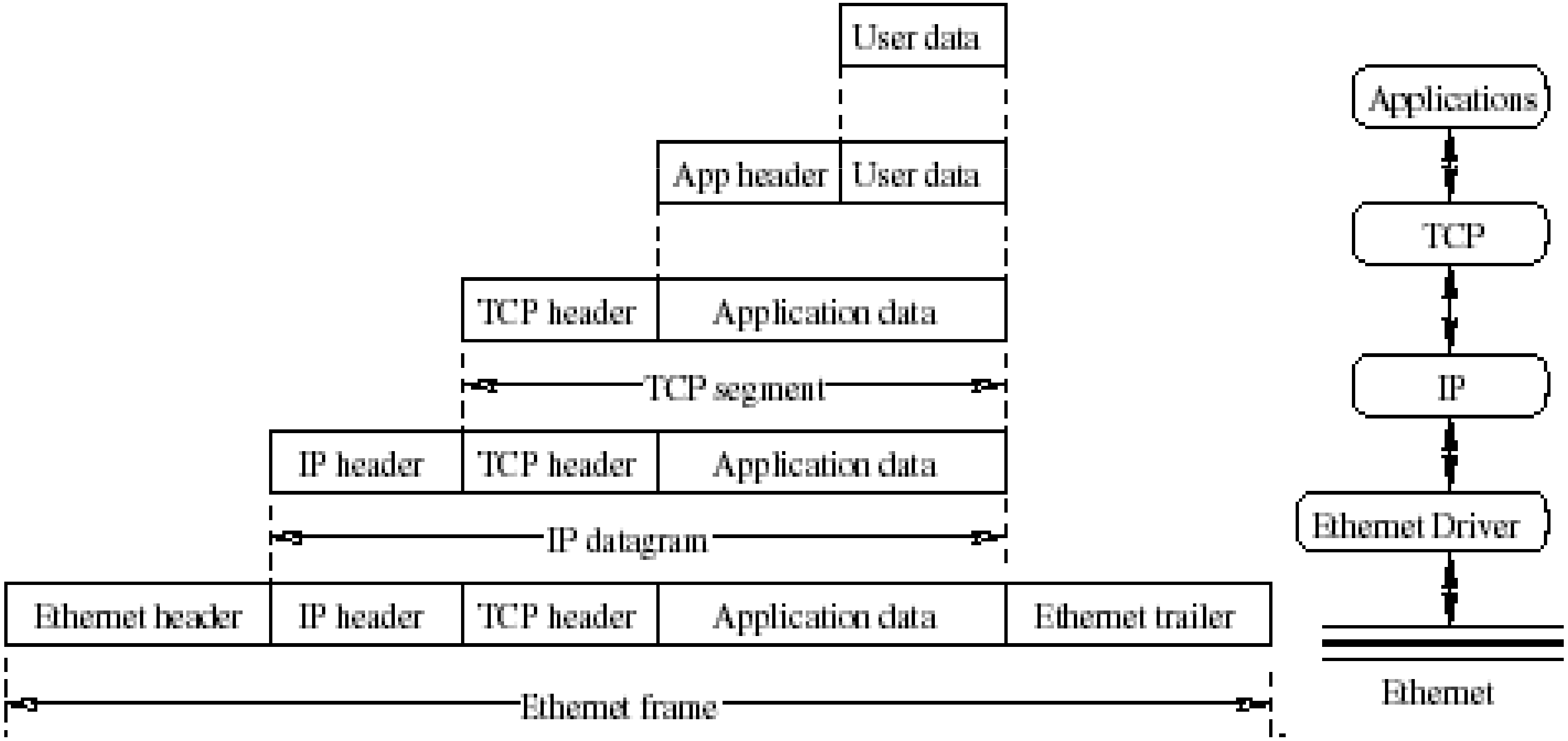


Hosts Communicating over a router



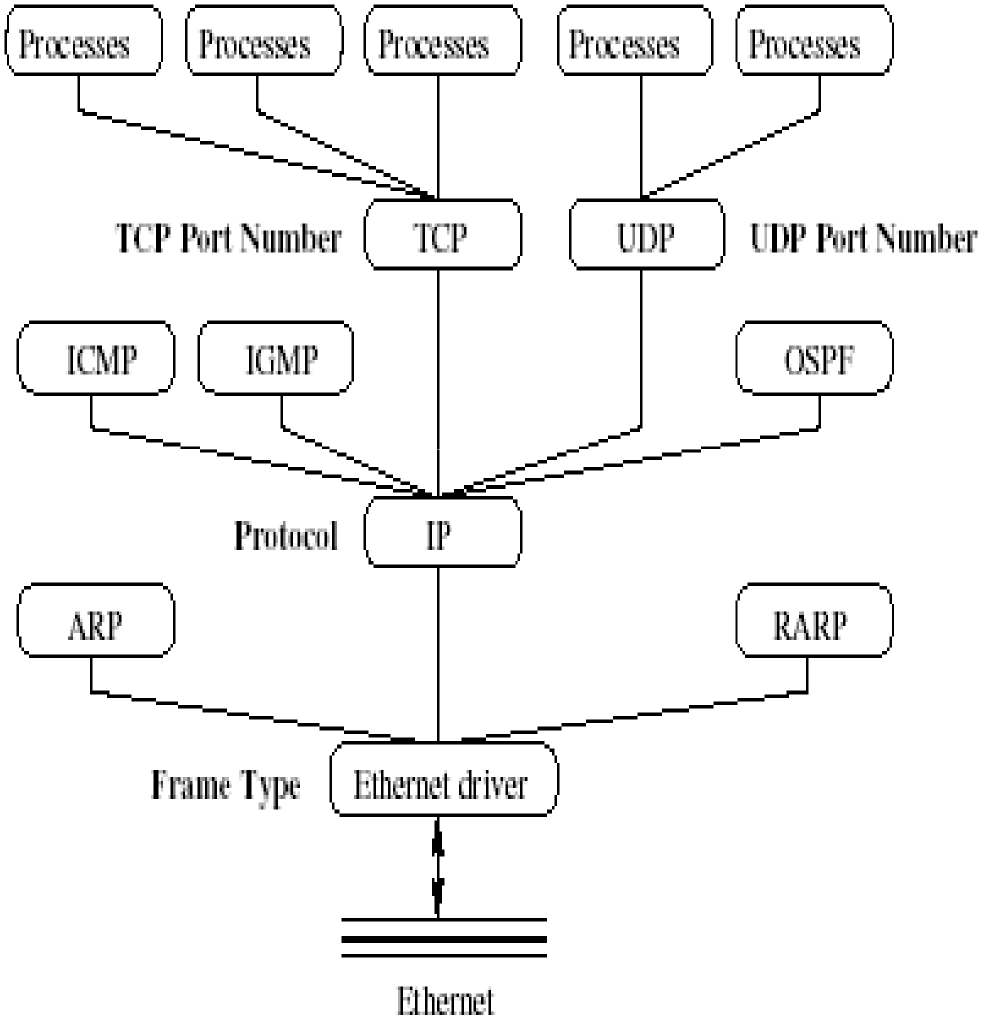
Encapsulation

- The application data is sent down
- Each layer adds a header to the data (PDU) from its higher layer.



Multiplexing and Demultiplexing

- Different higher layer protocols can use the service by the same lower layer protocol.
- The lower layer protocol uses a designated field in its header to identify different higher layer data in encapsulation.
- A higher layer protocol may use the service by different lower layer protocols.



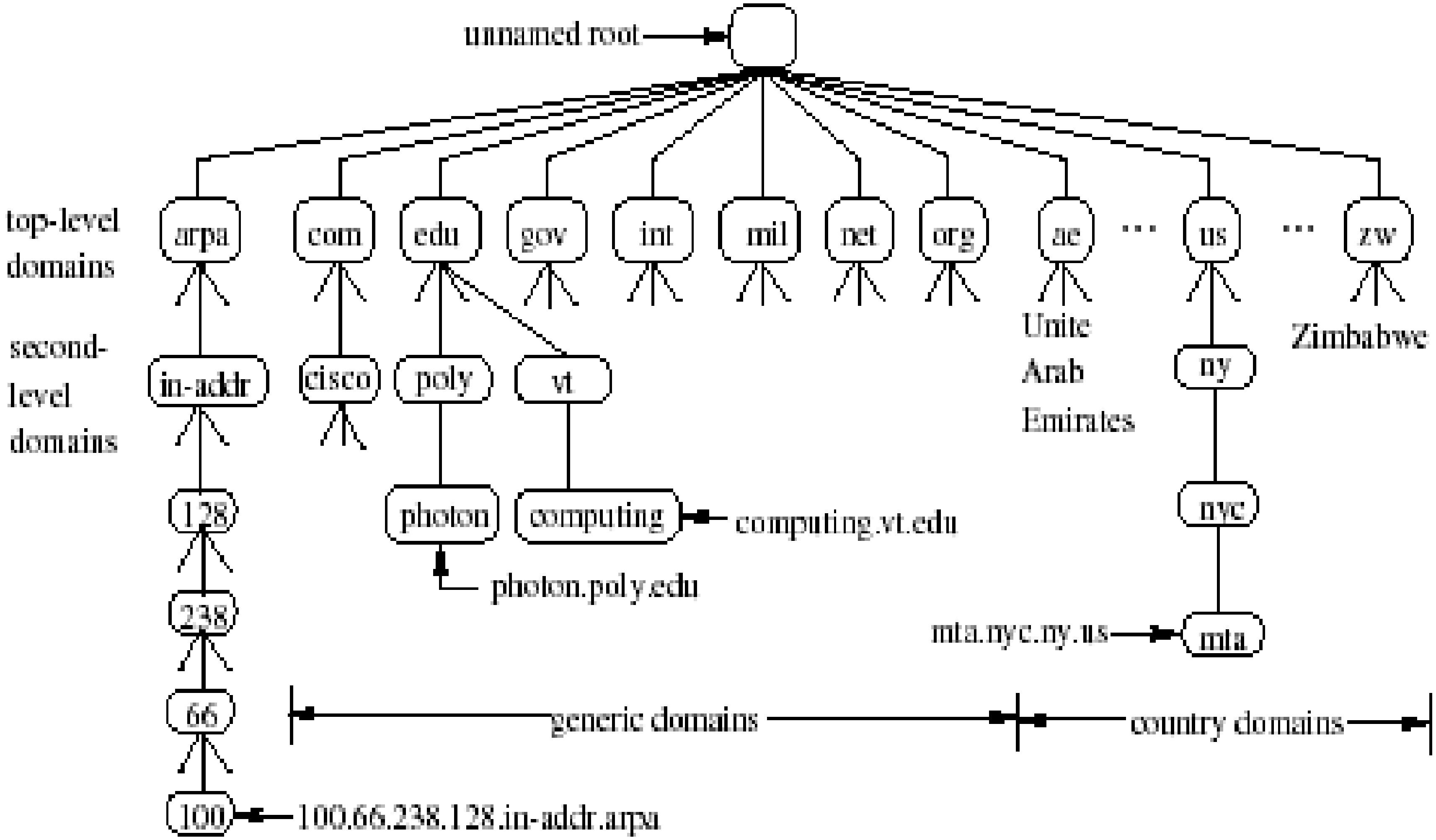
Naming and Addressing

- Uniquely identify processes in different computers for communications.
  - Domain name
  - MAC address
  - IP address
  - Port number

Domain Name

- Identify a host
- User friendly
- Hierarchically organized
  - Examples: eeweb.poly.edu, mta.nyc.ny.us

The Domain Name Space



Domain Name System (DNS)

- Resolves a domain name to the corresponding IP address.
- DNS servers
- Domain name database, distributed
- Name caching
  - Not efficient to query the same name resolution again and again
  - Time-to-Live timer
- DNS query and reply

MAC Address

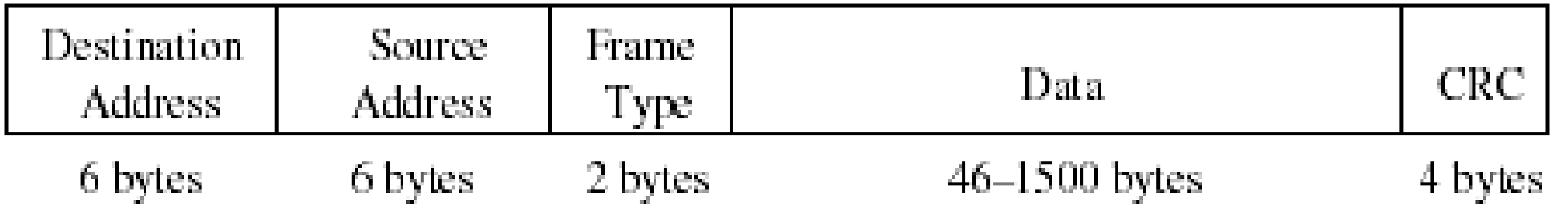
- Medium Access Control (MAC) address, hardware address
- In the layer, identify a network interface
- Usually burned in on the interface, ex. NIC
- Different link layer protocols use different MAC address
  - Ethernet, 48 bits, globally unique
  - Hexadecimal notation, e.g., 0x8:0:20:87:dd:88
- ARP (Address Resolution Protocol) is used to translate an IP address to the corresponding MAC address

Multiple Access

- Network topology
  - Point-to-point → N(N-1)/2 links to connect N nodes
  - Broadcast
- Medium access control (MAC) protocols
  - Rules to share a medium
  - Carrier Sense Multiple Access/Collision Detection (CSMA/CD)
  - Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)

Ethernet Frame Format

- Source Ethernet (MAC) Address
- Destination Ethernet Address
- Frame Type: used to identify the payload
- CRC: used for error control



Destination Address	Source Address	Frame Type	Data	CRC
6 bytes	6 bytes	2 bytes	46–1500 bytes	4 bytes

IP Address

- Each host "interface" in the Internet has a unique IP address.
- IPv4, 32 bits (4 bytes), written in dotted-decimal notation

  128.238.42.112 means

  10000000 in 1st byte

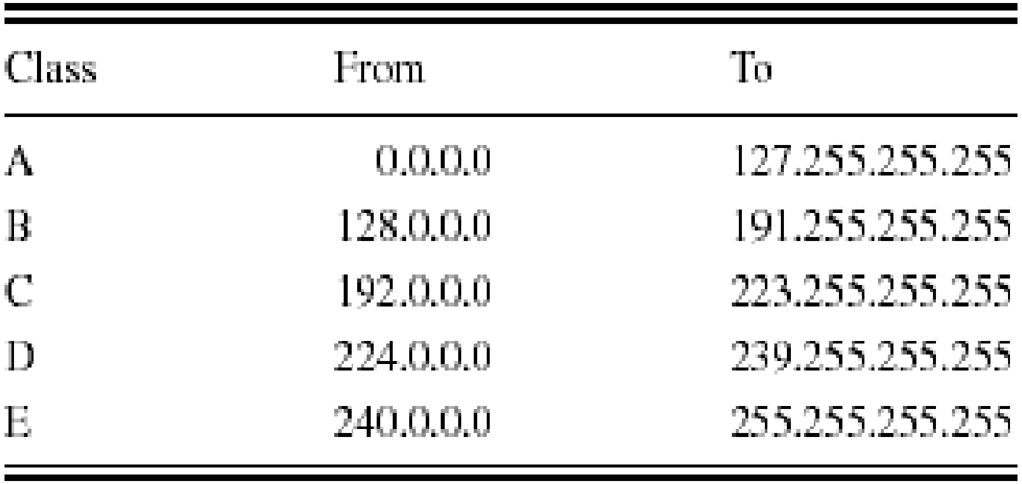
  11101110 in 2nd byte

  00101010 in 3rd byte

  01110000 in 4th byte

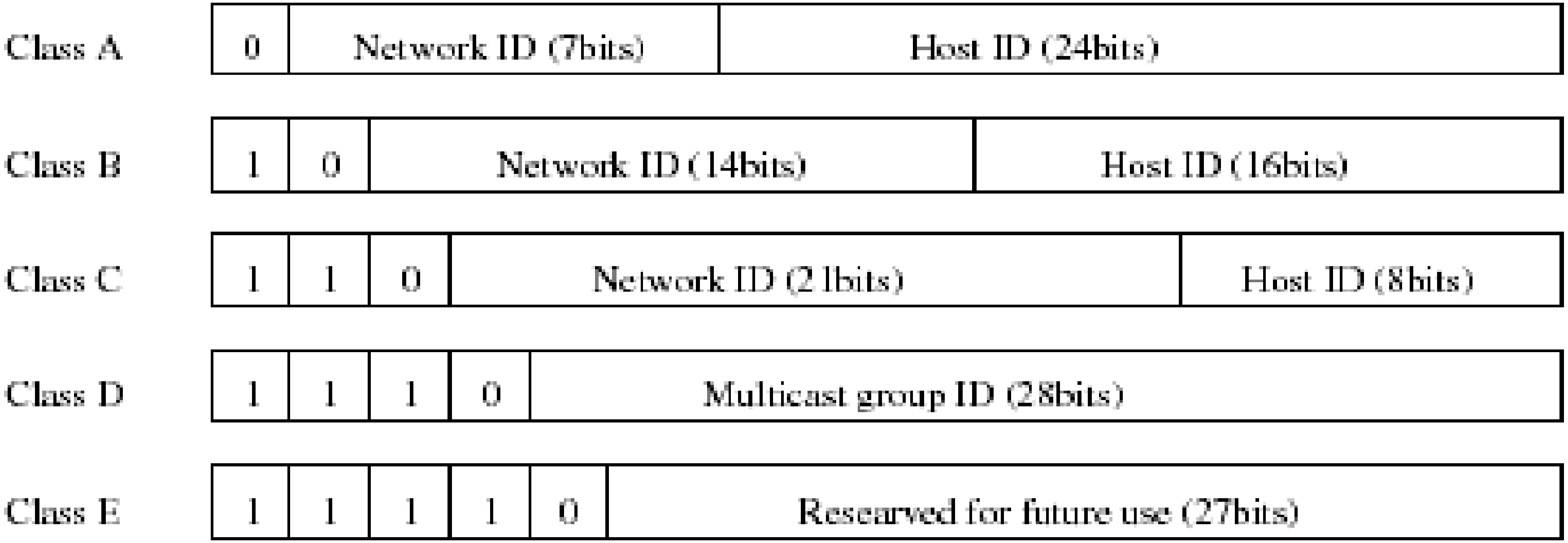
- IPv6, 128-bit address

Five Classes of IP Addresses



Class	From	To
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	255.255.255.255

The end points of each range are not allowed because all zeros and all ones are disallowed for Network ID and Host ID (see later discussion).



Class A	0	Network ID (7bits)	Host ID (24bits)			
Class B	1	0	Network ID (14bits)	Host ID (16bits)		
Class C	1	1	0	Network ID (21bits)	Host ID (8bits)	
Class D	1	1	1	0	Multicast group ID (28bits)	
Class E	1	1	1	1	0	Researved for future use (27bits)

Trade-off of Address Classes

There are a total of 2^{32} = 4,294,967,296 IP addresses.

The network IDs are assigned by Internet Assigned Numbers Authority (IANA).

- Class A: 7 bits for netid → only 128 Class A networks with about 16 million (2^{24}-2) hosts in each network.
- Class B:14 bits for netid → about 16,000 networks with about 65,000 (2^{16} -2) hosts in each network.
- Class C: 21 bits for netid→ about 2 million networks with only 254 hosts in each network.

The address classes make Class A and B addresses much more attractive than Class C addresses. But there are only few Class A and B addresses!

Management of a large number of Class C addresses is cumbersome.

Solution: Subnetting

Subnetting

Goal: divide a large number of IP addresses into groups as subnets interconnected with routers.

Basic Idea:

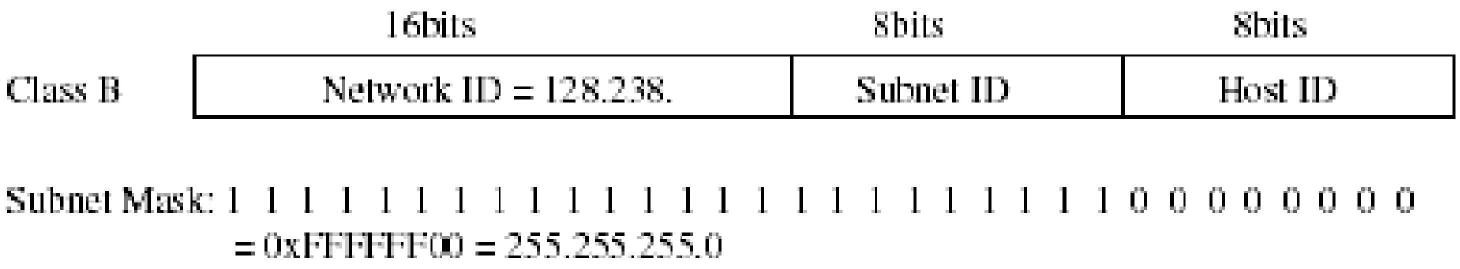
- Split the host id portion of IP addresses into a **subnet id** and a **host id**.
- Assign one **subnet id** to each physical network.

Then:

- Subnets can be freely assigned and be used for many physical networks.
- Distant routers need not be aware of **subnet id**’s.

Subnetting (cont'd)

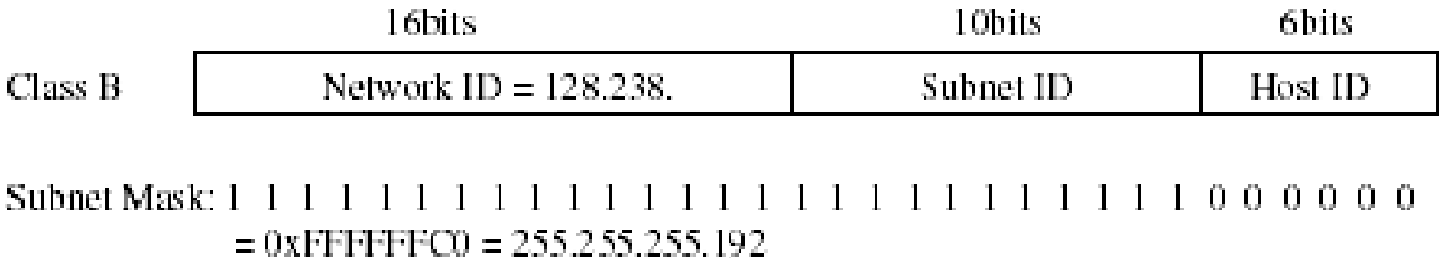
- Use three levels of an IP address:
  - Network ID
  - Subnet ID
  - Host ID
- Subnet mask: separates subnet ID and host ID



	16bits	8bits	8bits
Class B	Network ID = 128.238.	Subnet ID	Host ID

Subnet Mask: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
= 0xFFFFFF00 = 255.255.255.0

128.238.0.0/24 net then contains: 2^8 = 256 subnets with 2^8-2 = 254 hosts in each subnet



	16bits	10bits	6bits
Class B	Network ID = 128.238.	Subnet ID	Host ID

Subnet Mask: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
= 0xFFFFFFC0 = 255.255.255.192

128.238.0.0/26 net then contains: 2^{10} = 1024 subnets with 2^6-2 = 62 hosts in each subnet

IP Address Notation

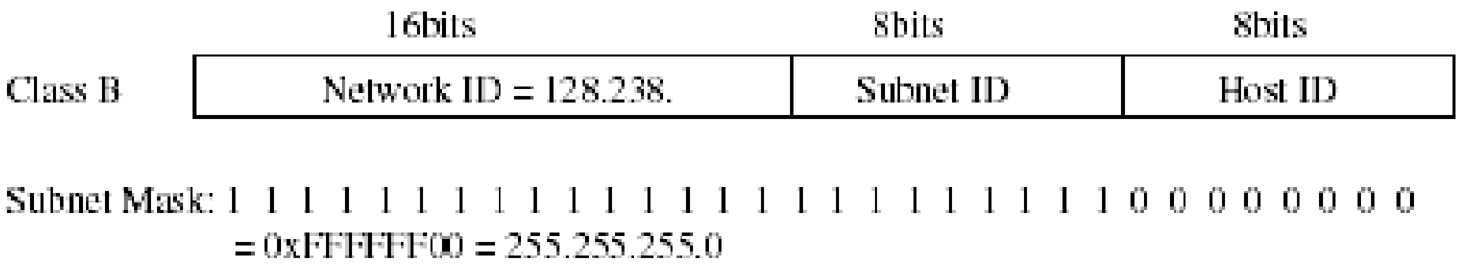
- Network-prefix: network ID
- (Extended-)network-prefix: the combination of network ID and subnet ID
- Slash-notation of IP address:
  - IP address followed by a ‘/’ and number of 1’s in the subnet mask
  - Examples:
    - > 128.238.66.101/24
    - > 128.238.66.101/26
- Socket: the combination of an IP address and a port number that identifies an unique endpoint of a bidirectional inter-process communication flow the Internet.

Subnetting (cont'd)

- Use three levels of an IP address:
  - Network ID
  - Subnet ID
  - Host ID
- Subnet mask: separates subnet ID and host ID

Here is a subnetting example with two masks in a design of 640 subnets.

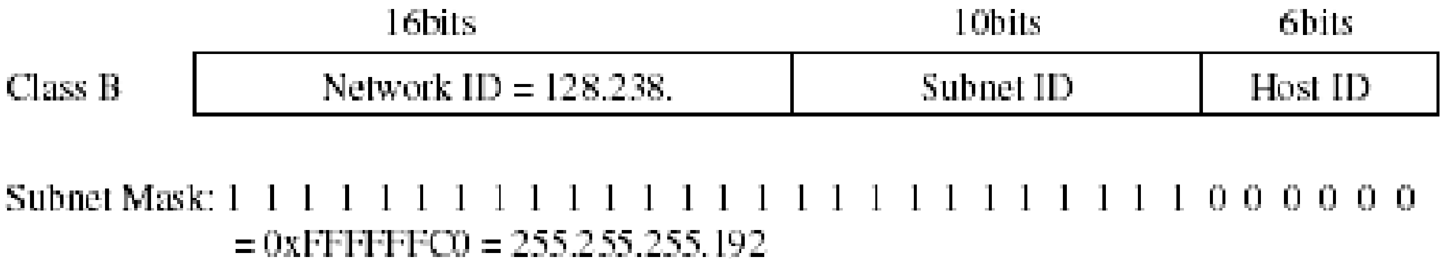
- Use a /24 mask to define 128 subnets, say with subnet address from 128.238.128.0 to 128.238.255.0
- Use a /26 mask to define another 512 subnets, say in this example with subnet address, from 128.238.0.0 to 128.238.127.192



	16bits	8bits	8bits
Class B	Network ID = 128.238.	Subnet ID	Host ID

Subnet Mask: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
= 0xFFFFFF00 = 255.255.255.0

128.238.0.0/24 net then contains: 2^8 = 256 subnets with 2^8-2 = 254 hosts in each subnet



	16bits	10bits	6bits
Class B	Network ID = 128.238.	Subnet ID	Host ID

Subnet Mask: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
= 0xFFFFFFC0 = 255.255.255.192

128.238.0.0/26 net then contains: 2^{10} = 1024 subnets with 2^6-2 = 62 hosts in each subnet

RFC1812 on Special IP Address

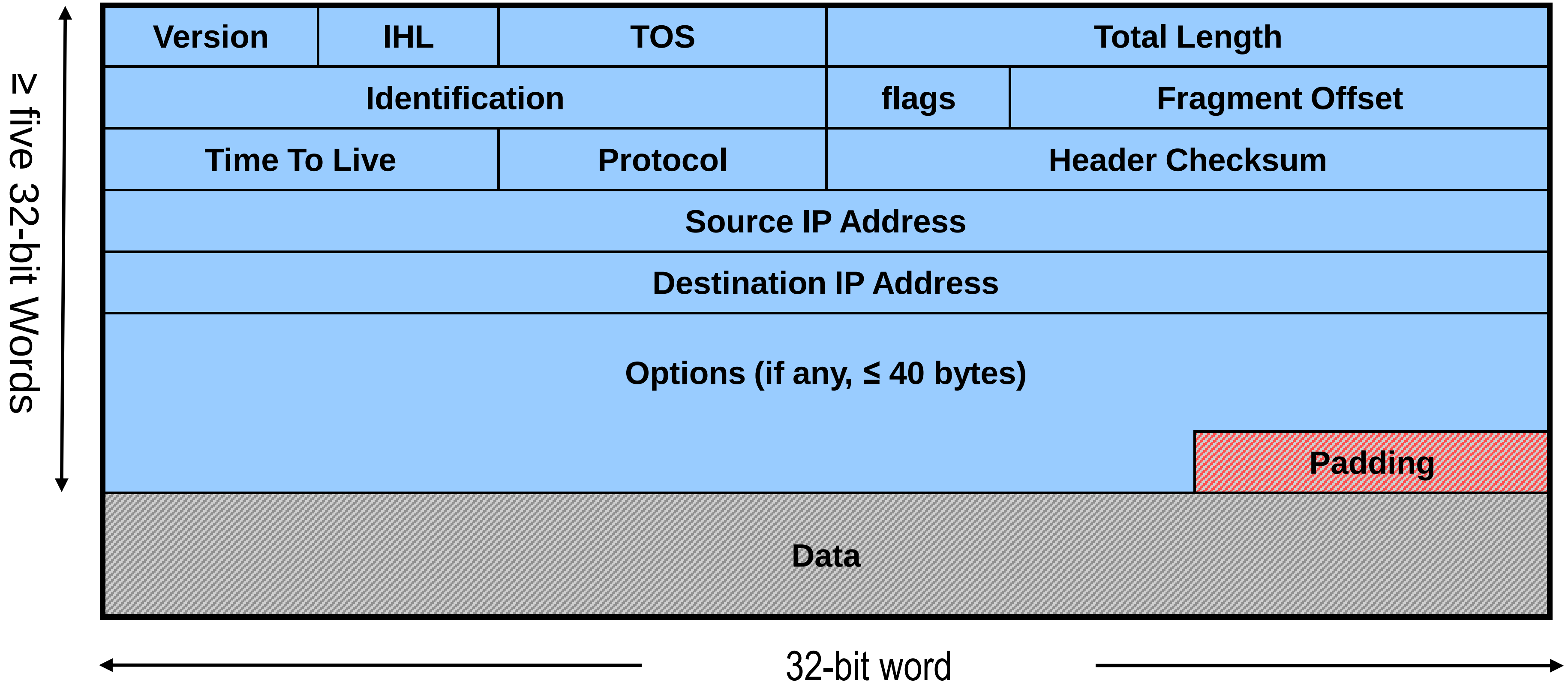
- IP addresses are not permitted to have the value 0 or -1 for the <Host-number> or <Network-prefix> fields except in the special cases listed above. This implies that each of these fields will be at least two bits long.

Network ID	Host ID	Special Address
Specific	All 0's	Network address
Specific	All 1's (-1)	Direct broadcast address
All 0's	Specific	Specified host on this network
All 0's	All 0's	This host on this network
All 1's (-1)	All 1's (-1)	Limited broadcast address in LAN
127	Any	Loopback address

- DISCUSSION: Previous versions of this document also noted that subnet numbers must be neither 0 nor -1, and must be at least two bits in length. With Classless InterDomain Routing (CIDR), the subnet number is clearly an extension of the network prefix and cannot be interpreted without the remainder of the prefix. This restriction of subnet numbers is therefore meaningless in view of CIDR and may be safely ignored.

IPv4 Packet Format

- Header size: 20 bytes if without options.
- 5 (20 bytes) ≤ Header Length ≤ 2^4 (up to 60 bytes)
- 20 bytes ≤ Total Length ≤ 2^{16} = 65536 bytes



Version | IHL | TOS | Total Length

Identification | flags | Fragment Offset

Time To Live | Protocol | Header Checksum

Source IP Address

Destination IP Address

Options (if any, ≤ 40 bytes)

Padding

Data

≥ five 32-bit Words

32-bit word

IPv4 Header Fields (1/4)

Version: current version is 4, next version will be 6.

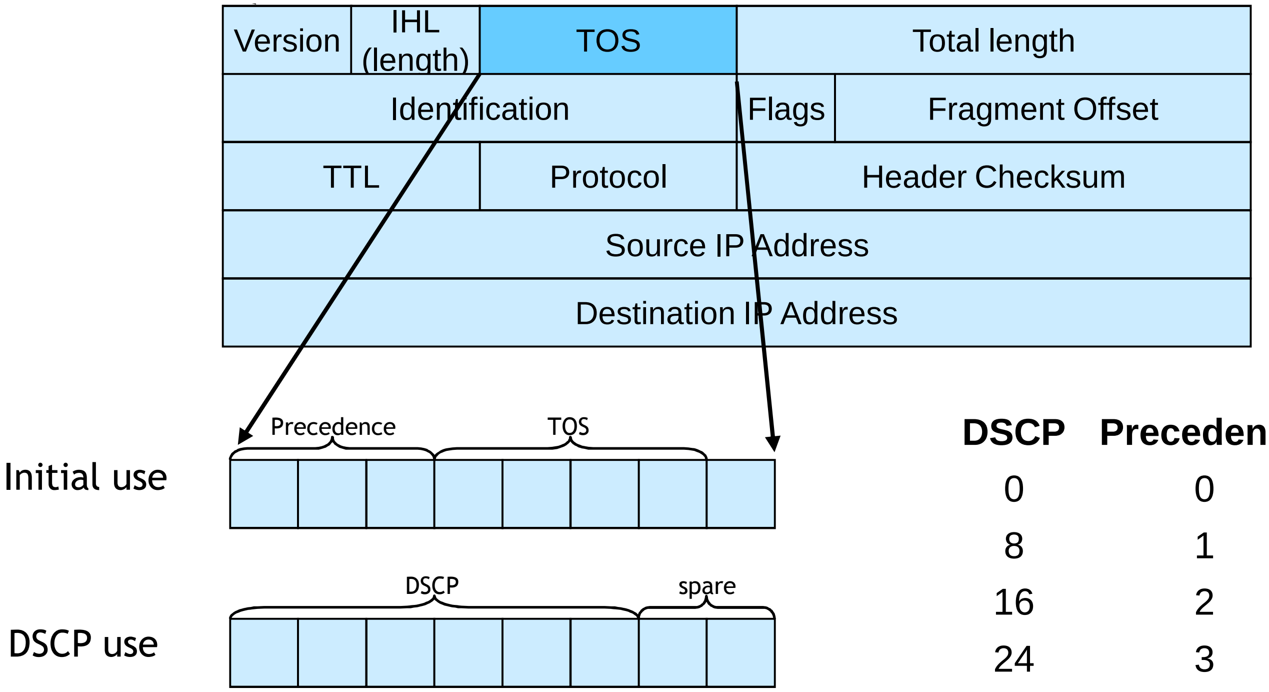
IHL (Internet Header Length, 4 bits): Number of 32-bit words in the IP header

Type of Service (TOS): contains 4 TOS bits, where each bit indicates a desired service.

  - minimize delay
  - maximize throughput
  - maximize reliability
  - minimize cost

- Only one bit can be set! Not supported in all applications.

IP Header TOS Field



DSCP: Differentiated Services Code Point

DSCP	Precedence	Purpose
0	0	Best effort
8	1	Class 1
16	2	Class 2
24	3	Class 3
32	4	Class 4
40	5	Express forwarding
48	6	Control
56	7	Control

IPv4 Header Fields (2/4)

Total Length: Number of bytes in the IP datagram (header+payload)

Identification: unique identification of a datagram from a host.

Incremented whenever a datagram is transmitted.

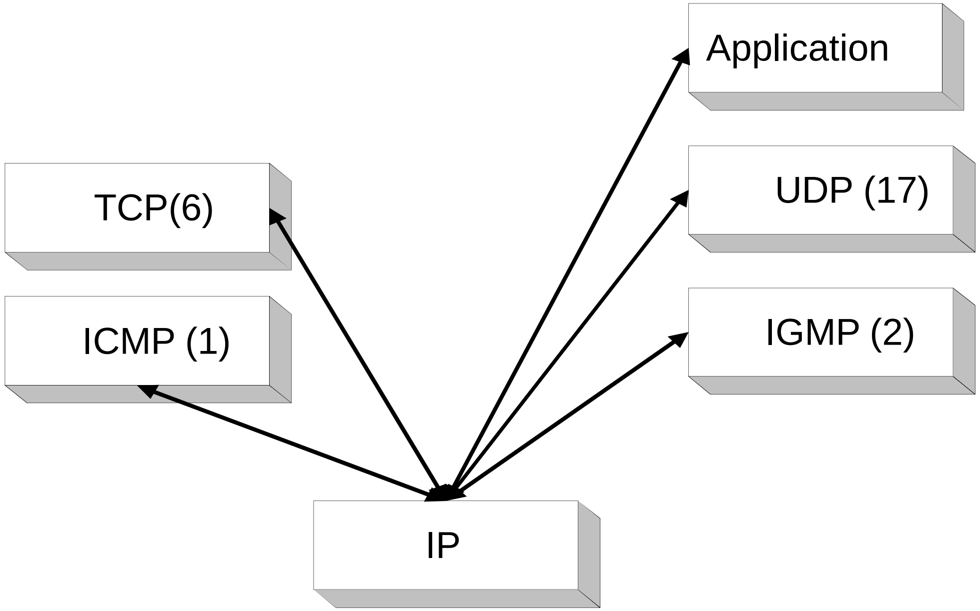
Flags and Fragrment Offset: Associated with fragmentation

Time To Live (TTL): specifies longest path before datagram is dropped.

- Used to prevent infinite looping of packets
- TTL field is set at sending host and is decremented by 1 at each router
- If a router gets a datagram whose TTL is either 0 or 1, the router will drop the packet.
- If a destination host gets a datagram whose TTL is larger than 0, the host will deliver the datagram to the higher layer.

IPv4 Header Fields (3/4)

Protocol: Specifies the higher-layer protocol. Used for demultiplexing to higher layers.



Header Checksum: verifies correctness of header.

IPv4 Header Fields (4/4)

Source and Destination Addresses: identify the interfaces on the sending and receiving hosts

Options:

- Security: indicates security and handling restrictions, …
- Record Route: each router that processes the packet adds its IP address to the header.
- Timestamp: each router that processes the packet adds its IP address and time to the header.
- (loose) Source Routing: specifies a list of routers that must be traversed.
- (strict) Source Routing: specifies a list of the only routers that can be traversed.

Padding: ensures that header ends on a 4-byte boundary

Port Number (per RFC 4340)

- Address for the application layer user process.
- Port Number field in TCP or UDP header.
- The Well-Known Port numbers are ranged from 1 to 1023.
  - Only be used by system (or root) processes or by programs directly executed by privileged users.
- The Registered Port numbers are ranged from 1024 through 49151.
  - Used by ordinary user processes or programs executed by ordinary users.
- The Dynamic and/or Private Port numbers are ranged from 49152 through 65535.
  - Intended for temporary use, including client-side ports, application testing prior to registration of a dedicated port, a.k.a. ephemeral port numbers
  - MUST NOT be registered.

UDP Header Format

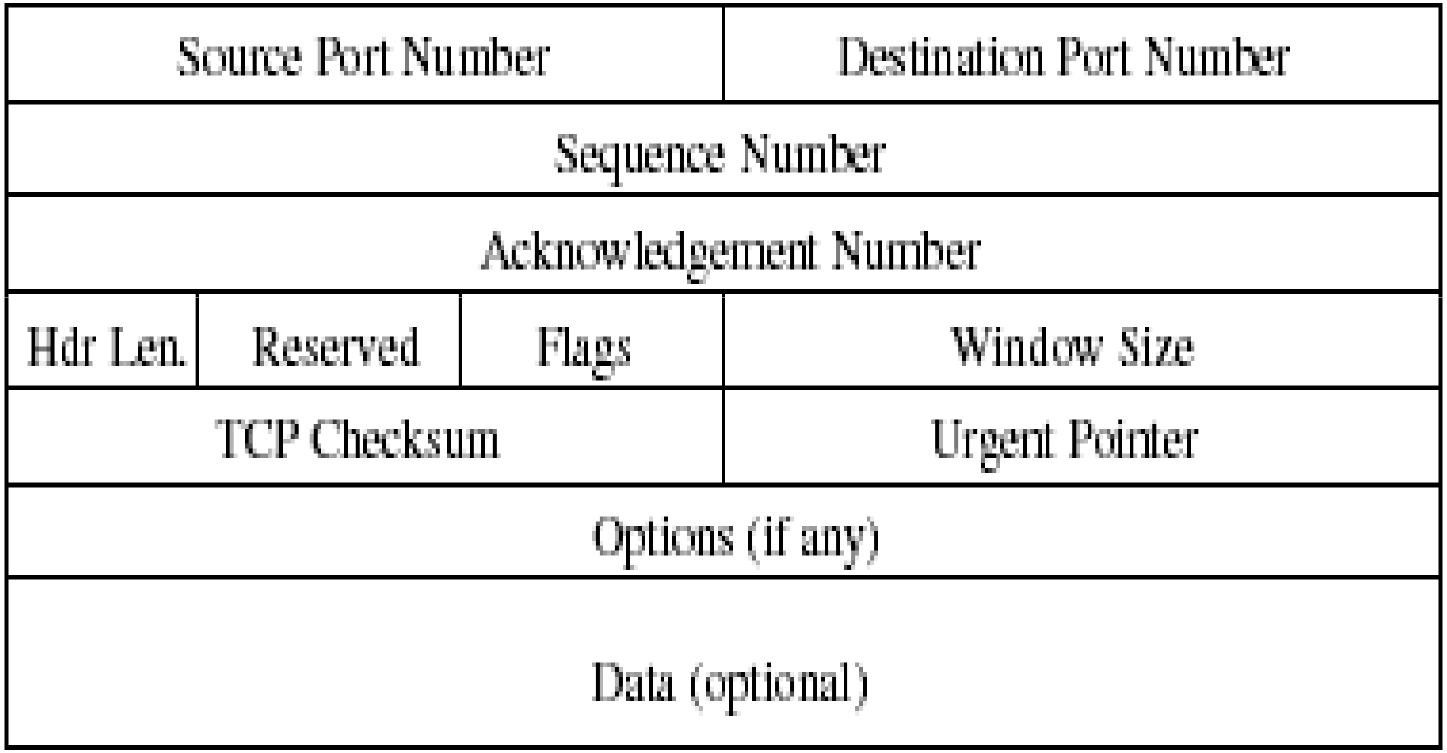
Source Port Number	Destination Port Number
UDP message length	Checksum

- Checksum: computed using
  - a pseudo-header as below

<table>
<tr><td colspan="3">32-bit Source IP Address</td></tr>
<tr><td colspan="3">32-bit Destination IP Address</td></tr>
<tr><td>0x00</td><td>8-bit Protocol (0x17)</td><td>16-bit UDP Length</td></tr>
</table>

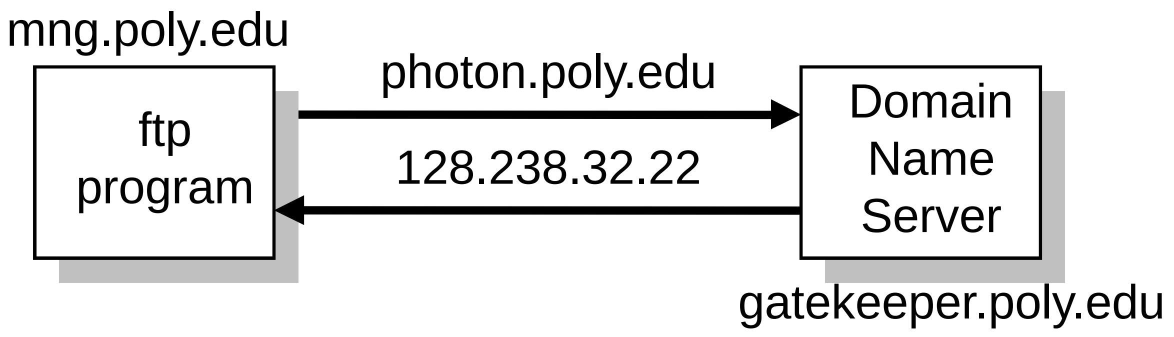
  - the UDP header, and
  - UDP data.

TCP Packet Format



<table>
<tr><td colspan="3">Source Port Number</td><td>Destination Port Number</td></tr>
<tr><td colspan="4">Sequence Number</td></tr>
<tr><td colspan="4">Acknowledgement Number</td></tr>
<tr><td>Hdr Len.</td><td>Reserved</td><td>Flags</td><td>Window Size</td></tr>
<tr><td colspan="3">TCP Checksum</td><td>Urgent Pointer</td></tr>
<tr><td colspan="4">Options (if any)</td></tr>
<tr><td colspan="4">Data (optional)</td></tr>
</table>

A “Simple” File Transfer at Poly

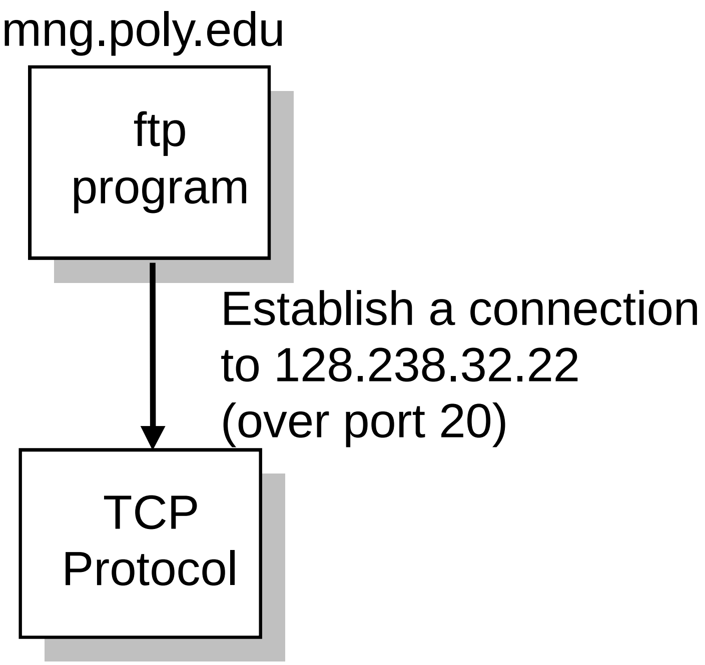


Step 1: The ftp program accesses a database that translates the *Hostname photon.poly.edu* into an *IP Address*.

- The distributed database used is called the *Domain Name System (DNS).*
- All machines on the Internet have at least one IP address. For example:

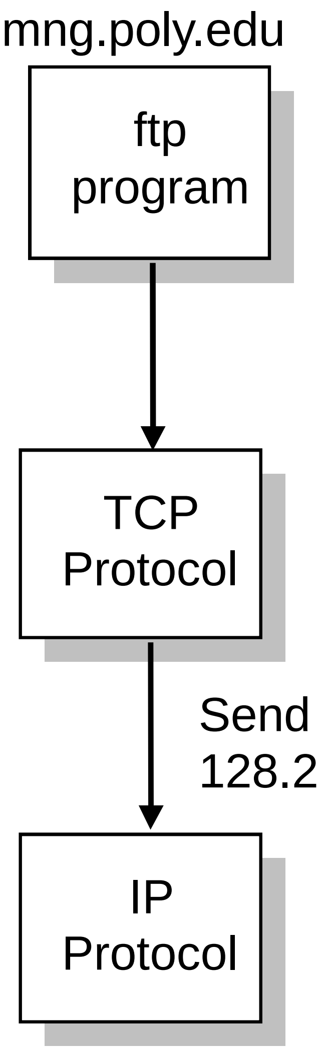
mng.poly.edu	*128.238.42.105/24*
gatekeeper.poly.edu	*128.238.2.38/24*

A “Simple” File Transfer at Poly



Step 2: mng.poly.edu requests the TCP protocol to establish a connection to the machine with address 128.238.32.22.

A “Simple” File Transfer at Poly



Send a datagram to 128.238.32.22 (for protocol 6)

Step 3: TCP sends a connection request by asking its local IP protocol to send an IP datagram to 128.238.32.22.

A “Simple” File Transfer at Poly

Step 4: The IP datagram can only be transmitted directly to 128.238.32.22, if it is on the same local (Ethernet) network as mng.poly.edu (128.238.42.105/24).

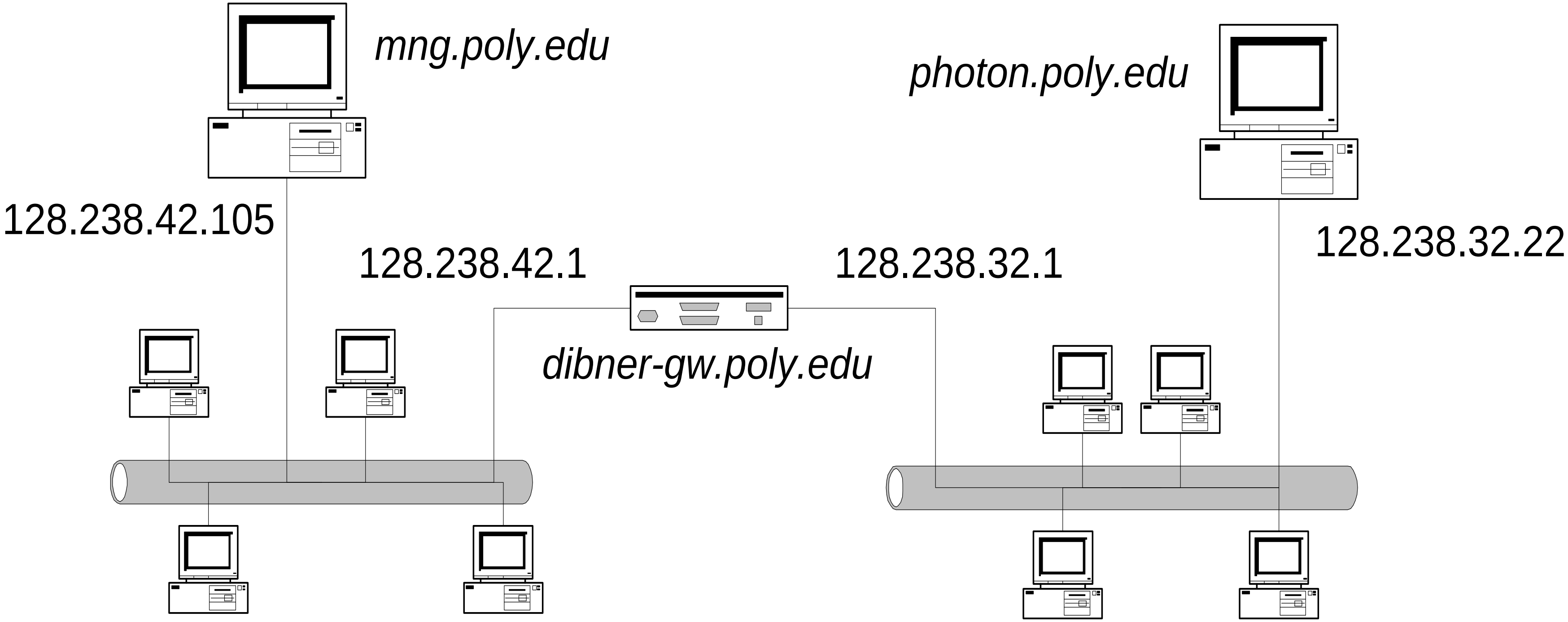
However, *mng* and *photon* are not on the same Ethernet.

(Q: How does *mng* know this?)

In this case, *mng* sends the IP datagram to a *Default Router* that is responsible for forwarding traffic to remote machines.

- The default router for *mng* is *dibner-gw.poly.edu* at 128.238.42.1.

The “Route” from *mng* to *photon*

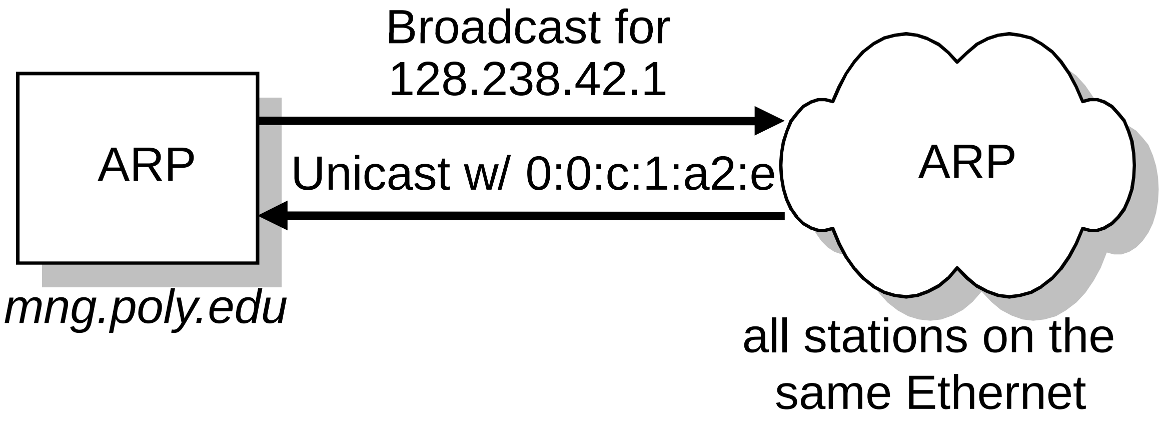


Note that *dibner-gw* has more than one IP address (In fact, it has a total of six IP addresses).

A “Simple” File Transfer at NYU-Poly

Step 5: *mng* must translate the IP address 128.238.42.1 into an *Ethernet Address*.

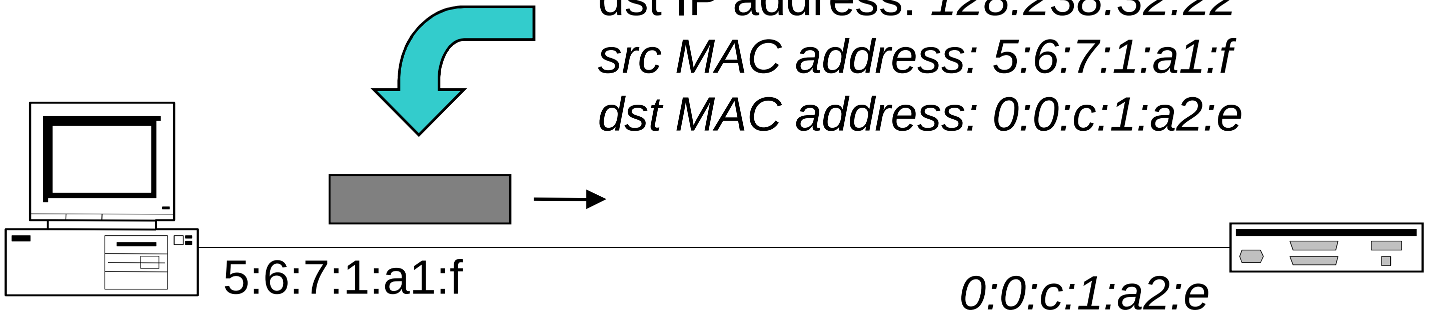
The resolution is performed by the *Address Resolution Protocol (ARP)*.



A “Simple” File Transfer at NYU-Poly

Step 6: The Ethernet device driver sends an *Ethernet Frame* which contains the IP datagram to address 0:0:c:1:a2:e.

- Ethernet address is for an interface (not a system)
- IP address is also per interface



src IP address: *128.238.42.105*
dst IP address: *128.238.32.22*
src MAC address: 5:6:7:1:a1:f
dst MAC address: 0:0:c:1:a2:e

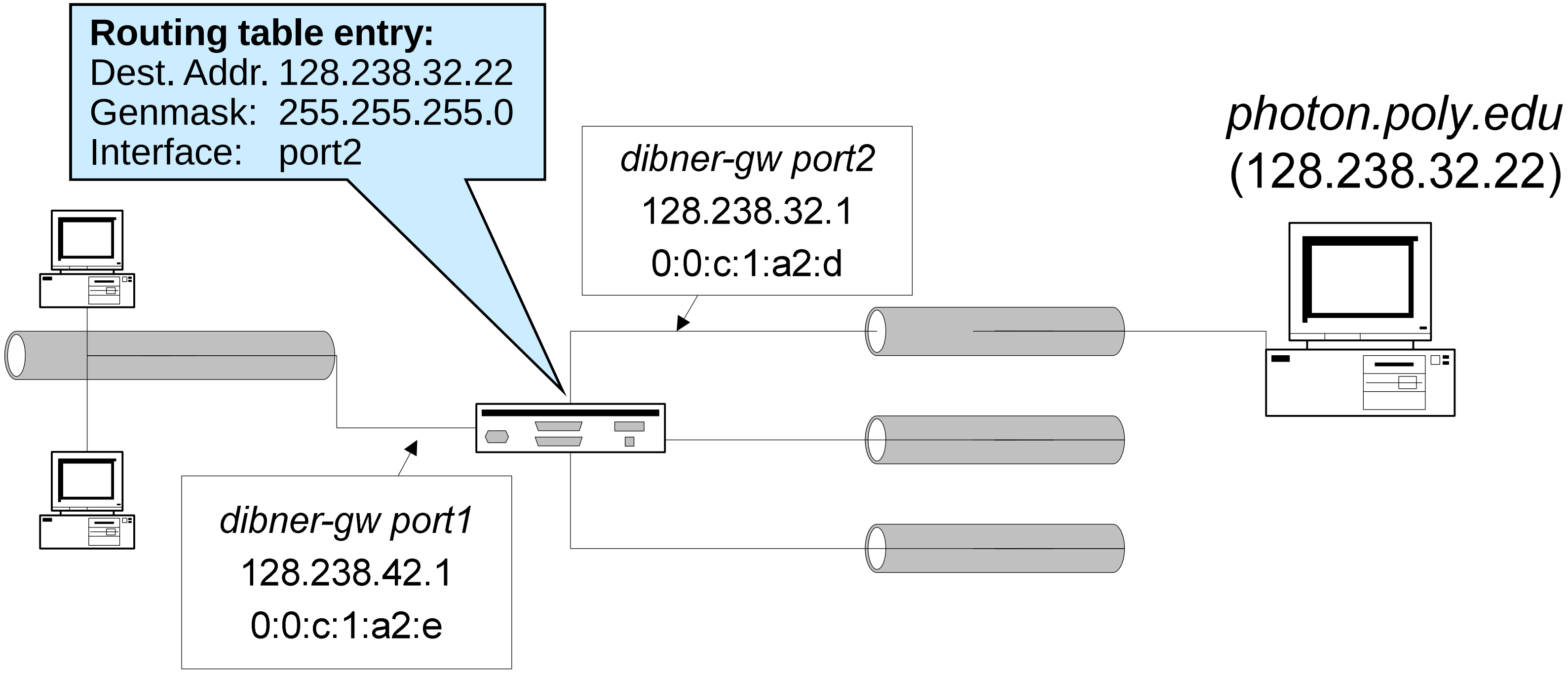
mng.poly.edu

dibner-gw.poly.edu

A “Simple” File Transfer at NYU-Poly

Step 7: *dibner-gw* receives the Ethernet frame, recovers the IP datagram and determines that the IP datagram should be forwarded to a host on one of the Ethernet segments connected to *dibner-gw*.

Note: *dibner-gw* performs a *Routing* function.



Routing table entry:
Dest. Addr. 128.238.32.22
Genmask: 255.255.255.0
Interface: port2

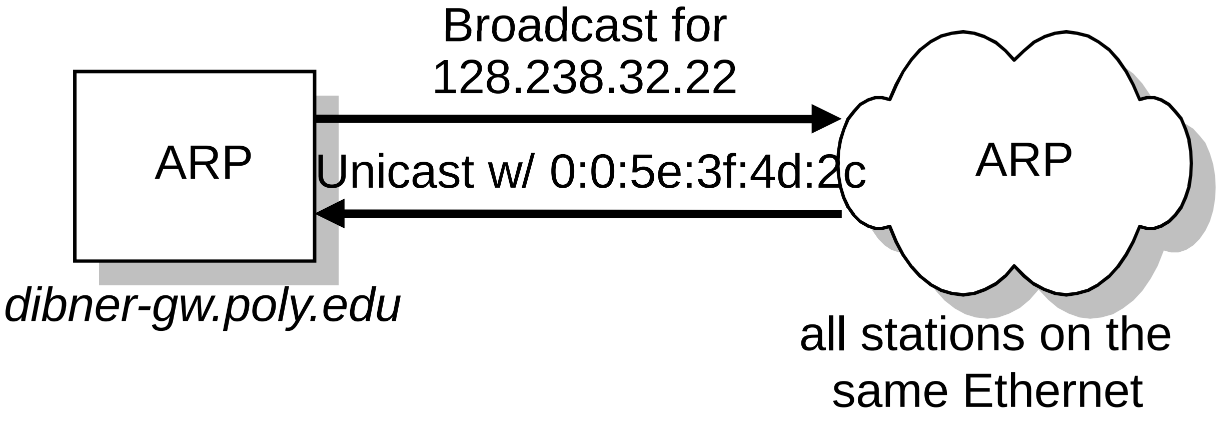
dibner-gw port2
128.238.32.1
0:0:c:1:a2:d

photon.poly.edu
(128.238.32.22)

dibner-gw port1
128.238.42.1
0:0:c:1:a2:e

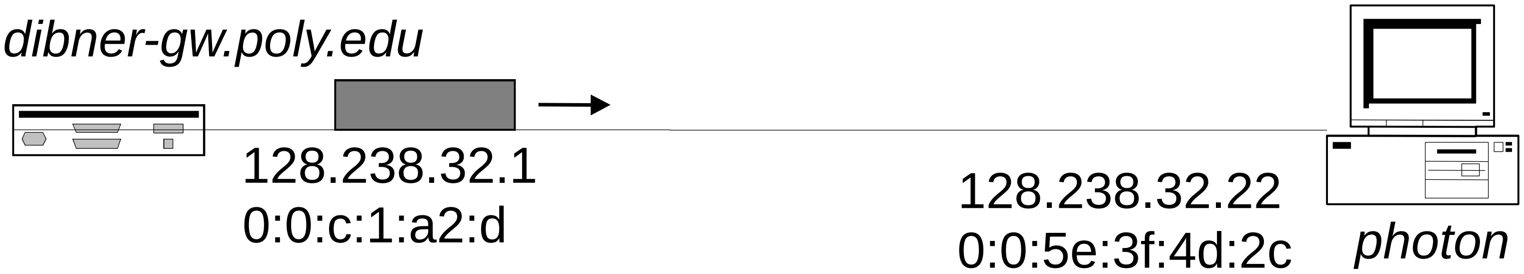
A “Simple” File Transfer at NYU-Poly

Step 8: *dibner-gw* performs the steps performed by *mng in Step 5;* it must translate the IP address 128.238.32.22 into an Ethernet Address (using ARP).



A “Simple” File Transfer at NYU-Poly

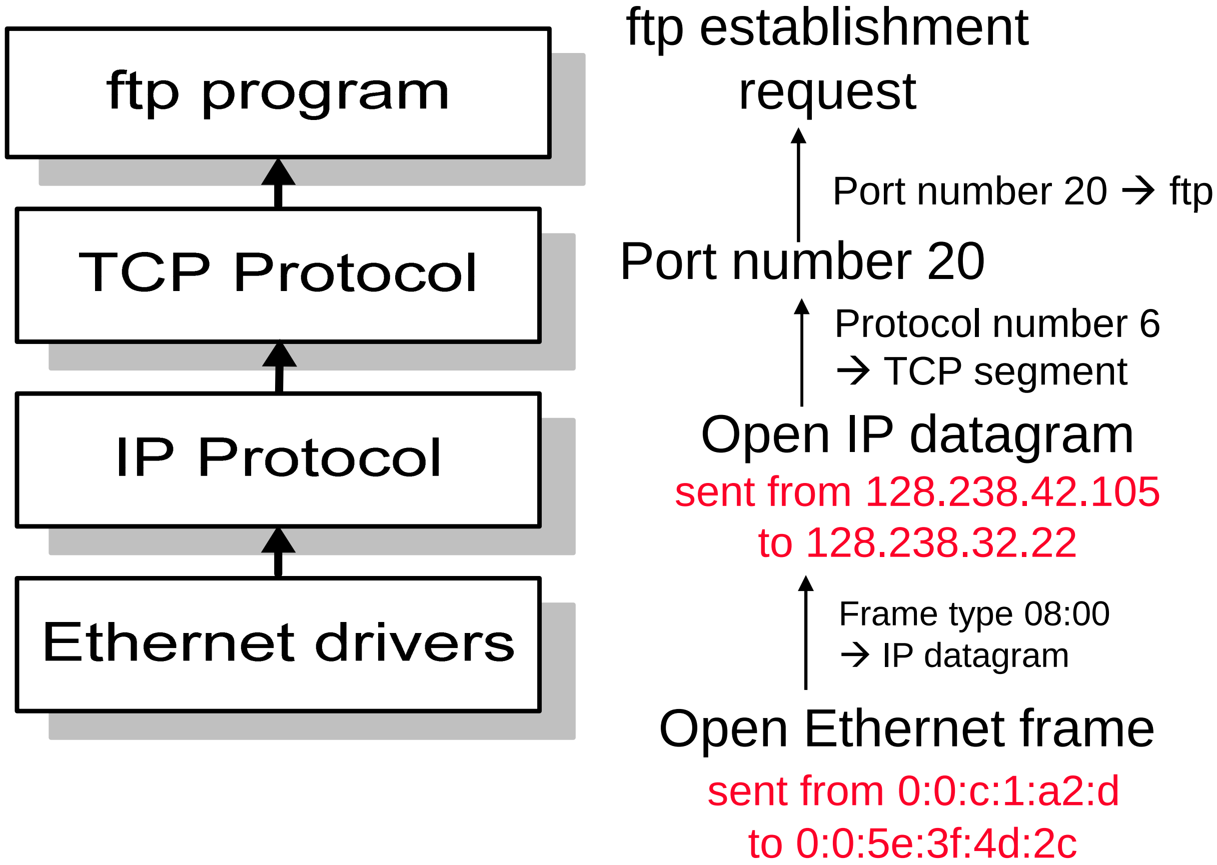
Step 9: The Ethernet device driver sends an Ethernet frame which contains the IP datagram to address 0:0:5e:3f:4d:2c.



A “Simple” File Transfer at NYU-Poly

The Ethernet frame is passed up the layers in photon with each layer processing its corresponding header.

photon.poly.edu



ftp program

TCP Protocol

IP Protocol

Ethernet drivers

ftp establishment request

Port number 20 → ftp

Port number 20

Protocol number 6 → TCP segment

Open IP datagram

sent from 128.238.42.105 to 128.238.32.22

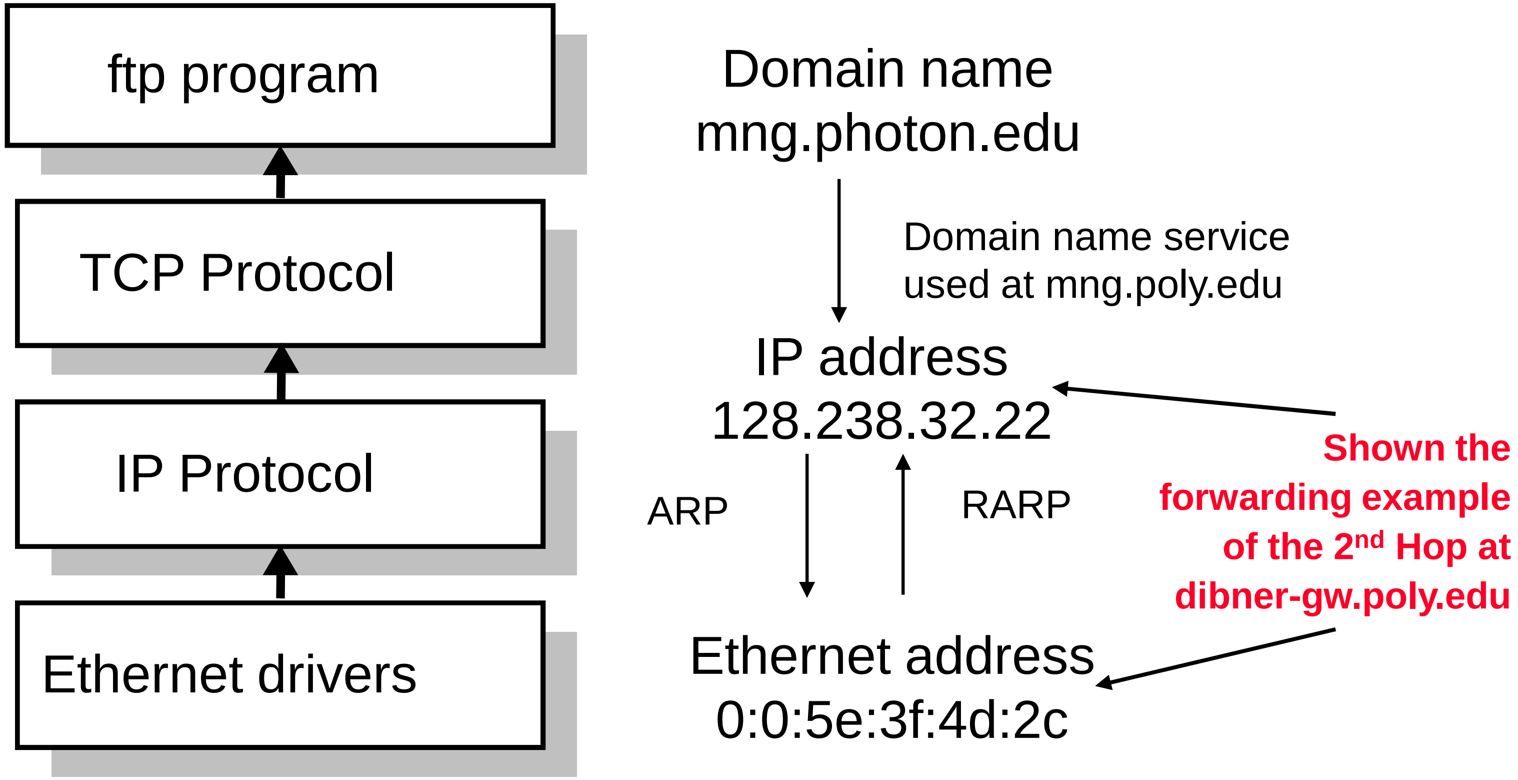
Frame type 08:00 → IP datagram

Open Ethernet frame

sent from 0:0:c:1:a2:d to 0:0:5e:3f:4d:2c

Protocol Layers and Node Identifiers

Node identifiers: Domain Name, IP address, Ethernet Addresses



Shown the forwarding example of the 2nd Hop at dibner-gw.poly.edu

Wrapping-Up the Example

So far, photon has only obtained a single packet.

Much more work is required to establish an actual ftp connection (or even the data transfer).

The example was simplified in several ways:

- No transmission errors
- *mng* and *photon* are close to another
- ARP involves all machines on the Ethernet
- *dibner-gw* had the routing information it needed
- ….

An Example:

How TCP/IP Protocols Work Together

- Bob, a user, wants to book an air ticket from an online booking website.
- Bob knows the domain name www.expedia.com
- The remote computer with the domain name is a web server.

An Example – the Application

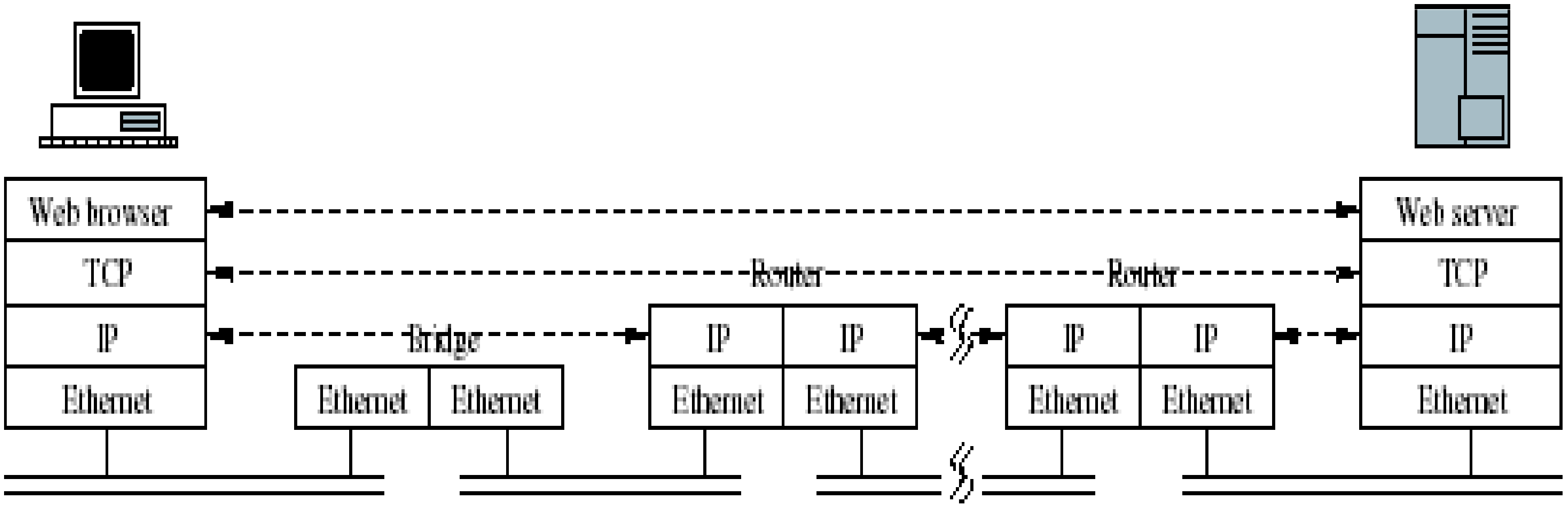
- The web server provides the web service.
- Bob uses a web browser, which is a web client, to request and receive web service.
- The HyperText Transfer Protocol (HTTP), an application layer protocol, is used by the web server and browser.

An Example – DNS

- Bob starts MS Internet Explorer in his computer, and types http://www.expedia.com/index.html.
- The domain name needs to be translated to an IP address.
- A DNS query is sent to a DNS server.
- A DNS reply will return to the client with the IP address.

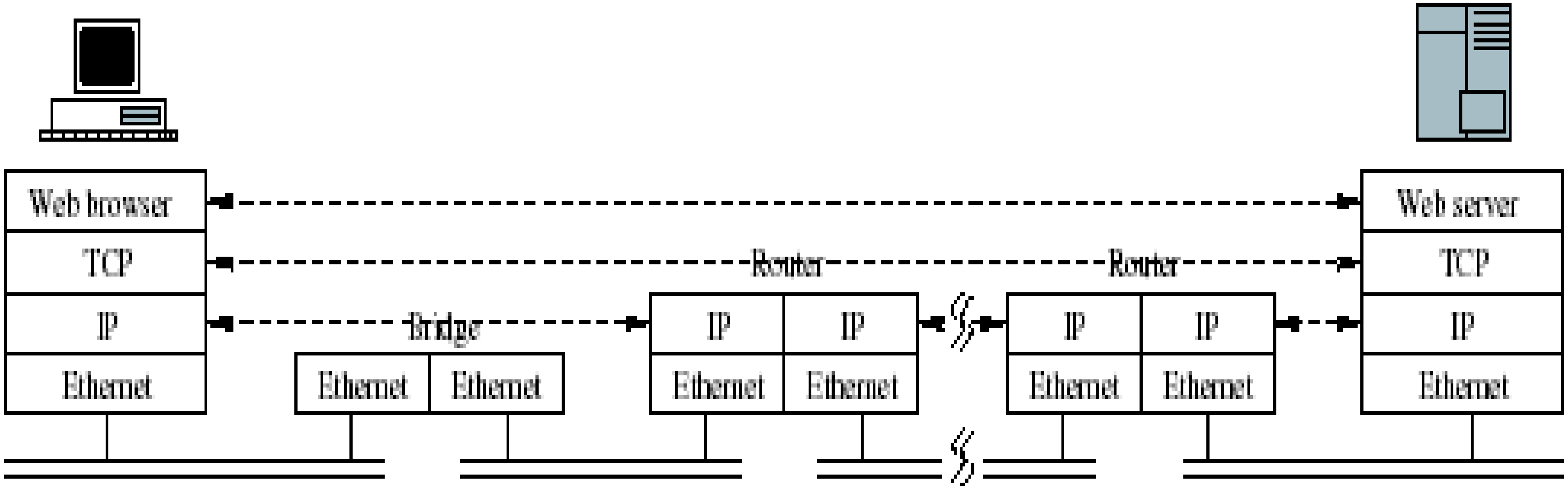
An Example – TCP Connection

- The client establishes a TCP connection to the web server.
- A port number is carried in all packets in this process.
- Application data (an HTTP request message for the index.html file) is sent over the TCP connection, encapsulated in a TCP segment.



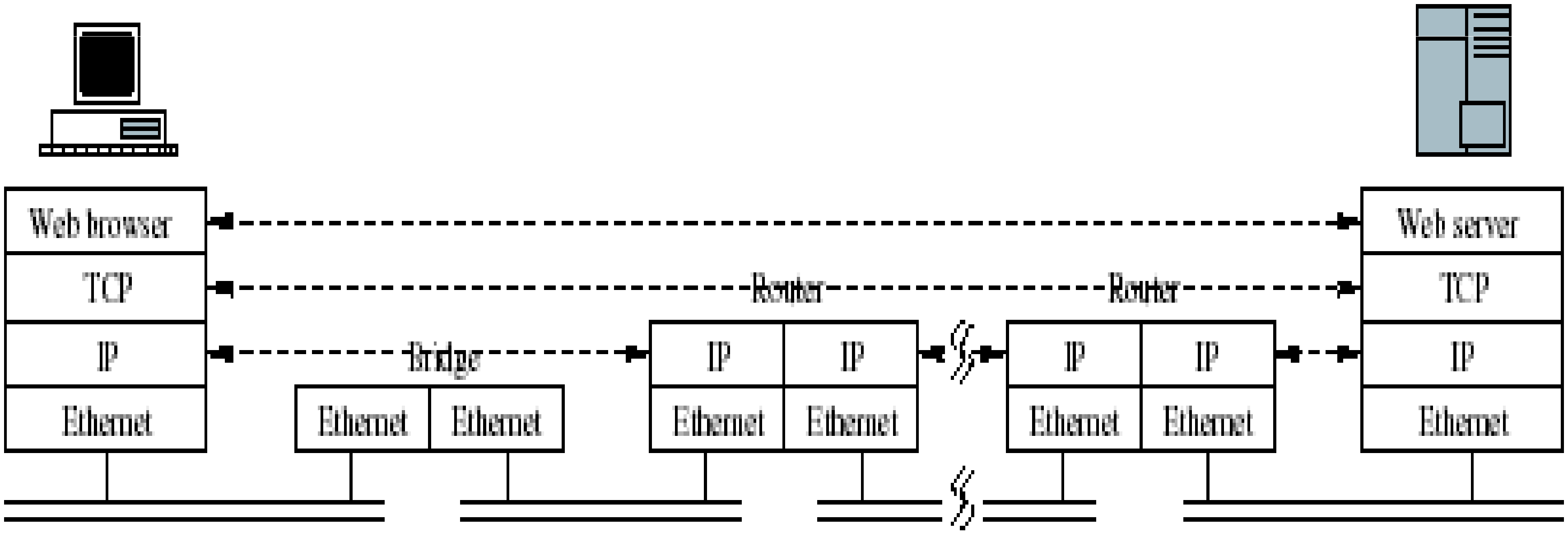
An Example – IP Layer

- The TCP segment is sent down to the IP layer and encapsulated in an IP datagram.
- Routers will forward the IP datagram hop by hop to the web server by checking their routing tables.



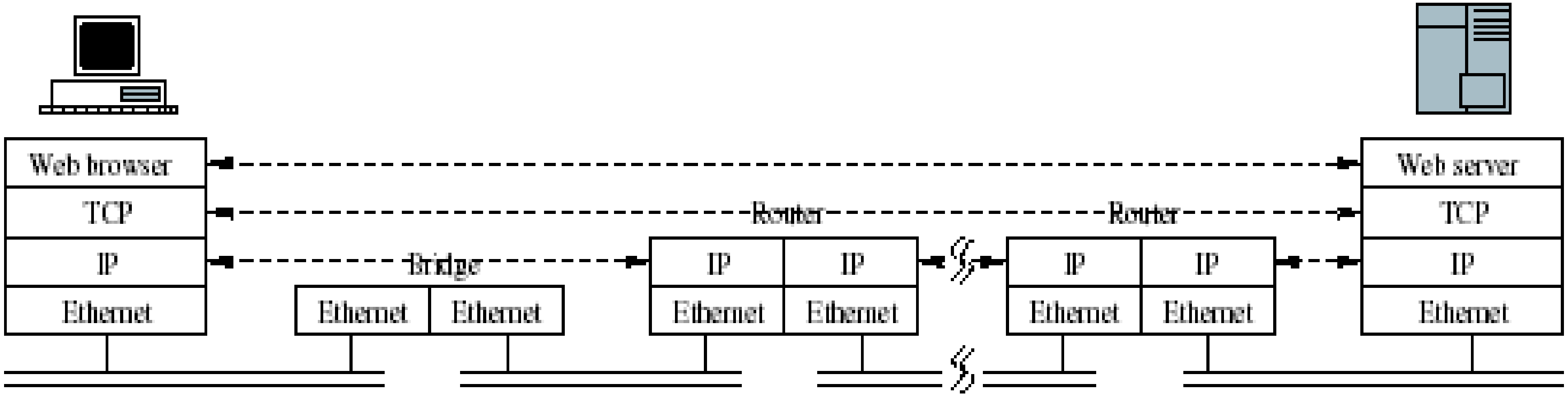
An Example – MAC Layer

- The IP datagram and the next-hop router's IP address down to the MAC layer.
- The IP datagram is encapsulated in an Ethernet frame.



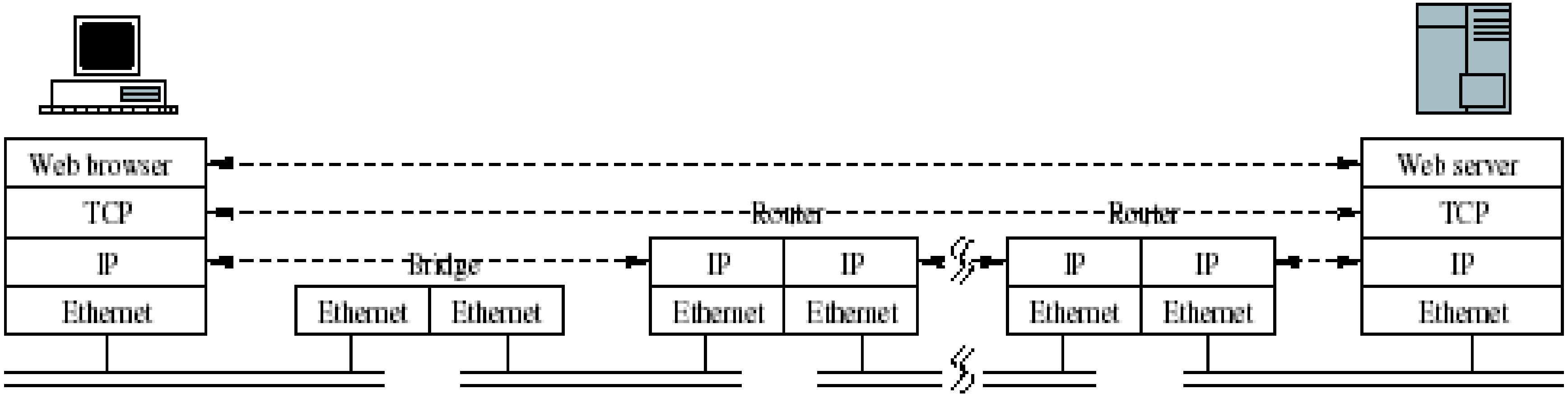
An Example – ARP

- The Ethernet frame needs to be sent to the interface of the next-hop router.
- Only Ethernet MAC address can be recognized.
- ARP request/reply is used to resolve the MAC address.



An Example – on the Other End

- The web server receives the Ethernet frame.
- The packet is delivered to the upper layers.
- When the application layer receives the HTTP request message, it sends an HTTP response message to the client.


Web browser
TCP
IP
Ethernet
Bridge
Ethernet
Ethernet
Router
IP
IP
Ethernet
Ethernet
Router
IP
IP
Ethernet
Ethernet
Web server
TCP
IP
Ethernet