Lecture 2
Applications and Layered Architecture

Protocols, Services & Layering

OSI Reference Model

TCP/IP Architecture

How the Layers Work Together

Lecture 2
Applications and Layered Architecture

Protocols, Services & Layering

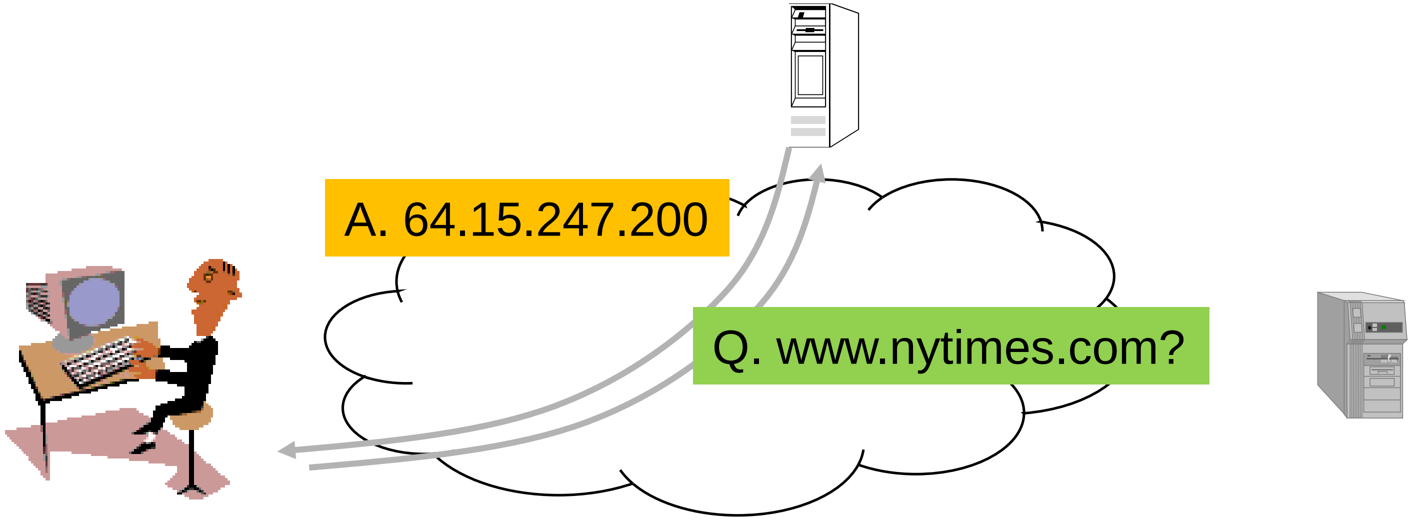
Layers, Services & Protocols

- The overall communications process between two or more machines connected across one or more networks is very complex
- *Layering* partitions related communications functions into groups that are manageable
- Each layer provides a *service* to the layer above
- Each layer operates according to a *protocol*

Web Browsing Application

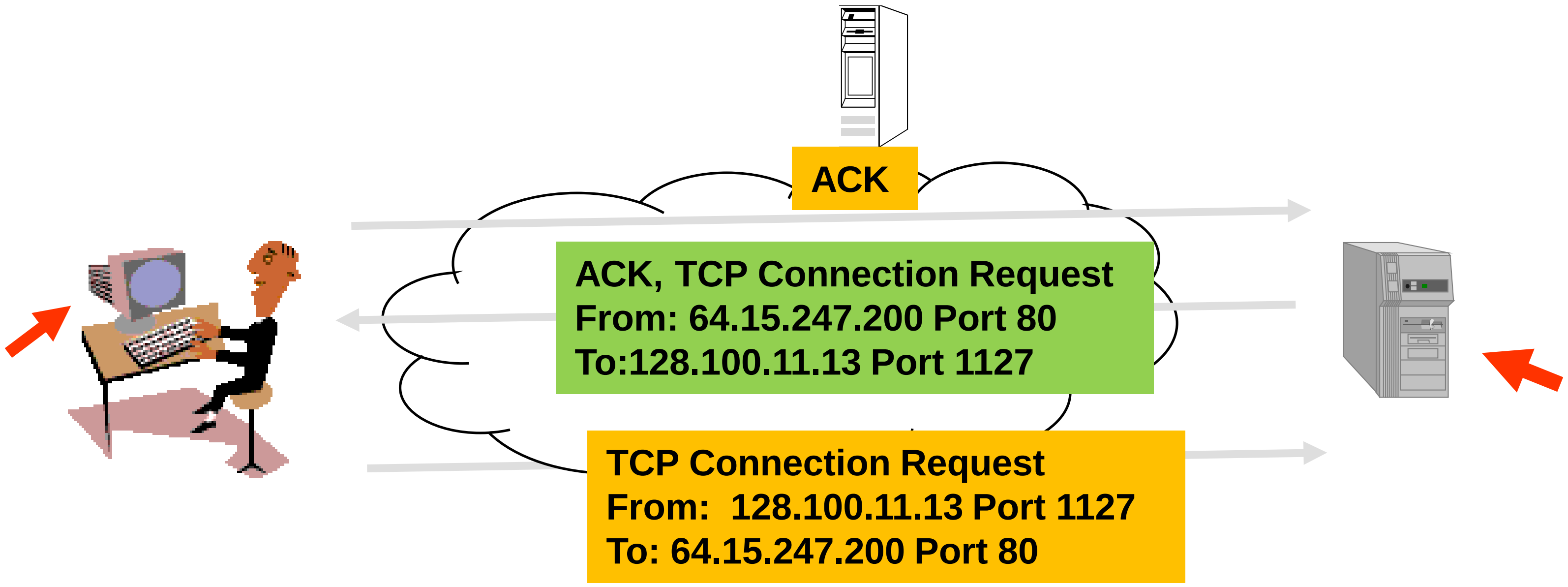
- World Wide Web allows users to access resources (i.e. documents) located in computers connected to the Internet
- Documents are prepared using HyperText Markup Language (HTML)
- A browser application program is used to access the web
- The browser displays HTML documents that include *links* to other documents
- Each link references a *Uniform Resource Locator* (URL) that gives the name of the machine and the location of the given document
- Let's see what happens when a user clicks on a link

1. DNS



- User clicks on http://www.nytimes.com/
- URL contains Internet name of machine (www.nytimes.com), but not Internet address
- Internet needs Internet address to send information to a machine
- Browser software uses Domain Name System (DNS) protocol to send query for Internet address
- DNS system responds with Internet address

2. TCP

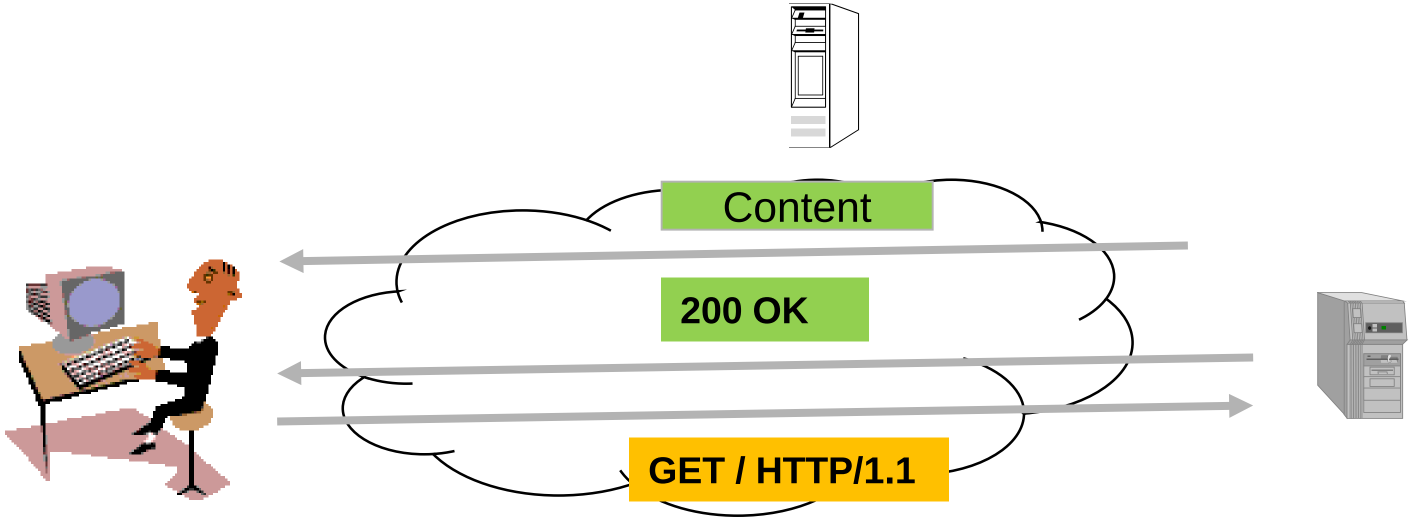

ACK

ACK, TCP Connection Request
From: 64.15.247.200 Port 80
To:128.100.11.13 Port 1127

TCP Connection Request
From: 128.100.11.13 Port 1127
To: 64.15.247.200 Port 80

- Browser software uses HyperText Transfer Protocol (HTTP) to send request for document
- HTTP server waits for requests by listening to a well-known port number (80 for HTTP)
- HTTP client sends request messages through an “ephemeral port number,” e.g. 1127
- HTTP needs a Transmission Control Protocol (TCP) connection between the HTTP client and the HTTP server to transfer messages reliably

3. HTTP



- HTTP client sends its request message: “GET …”
- HTTP server sends a status response: “200 OK”
- HTTP server sends requested file
- Browser displays document
- Clicking a link sets off a chain of events across the Internet!
- Let’s see how protocols & layers come into play…

Example: TCP

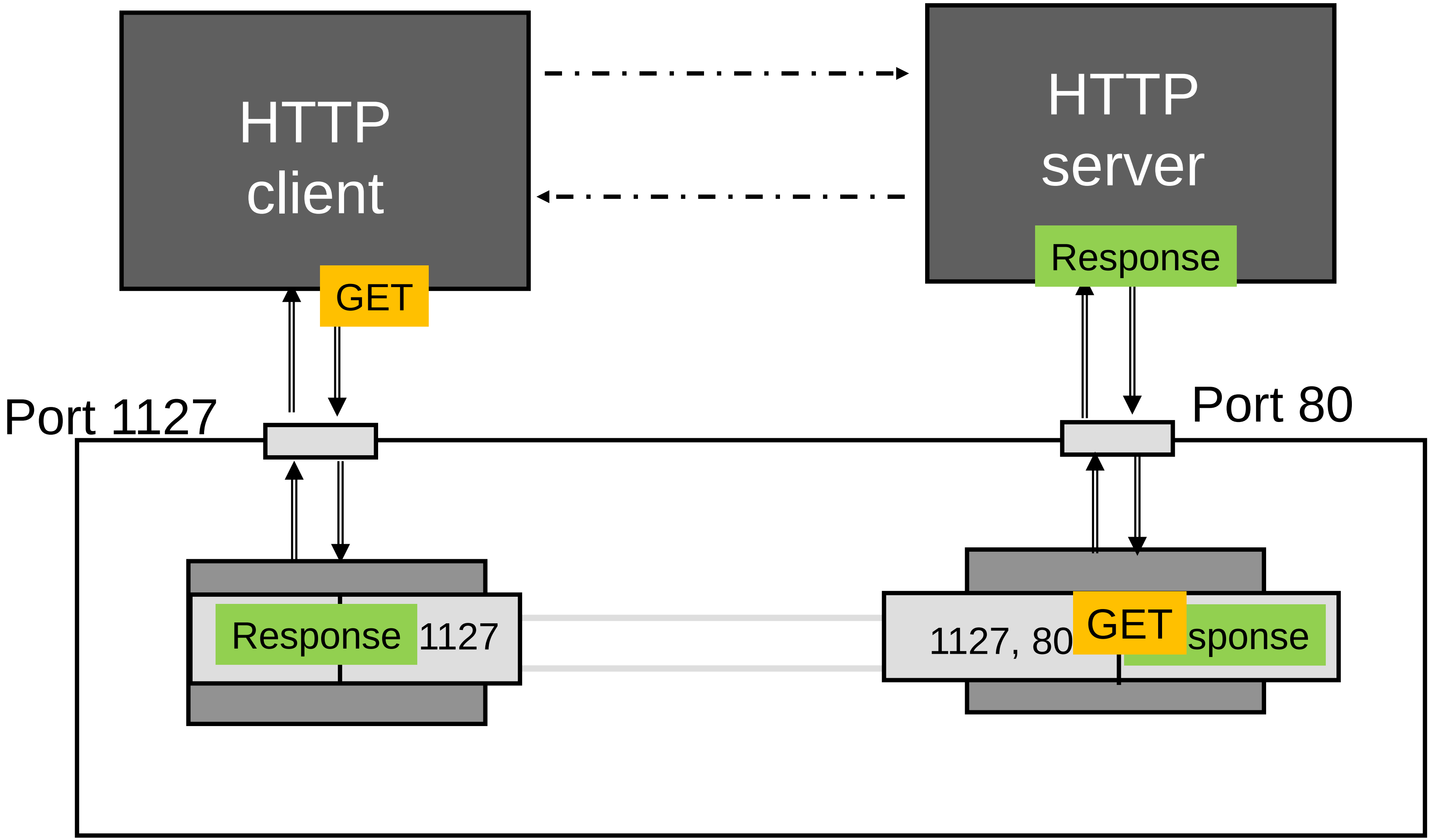
- TCP is a transport layer protocol
- Provides *reliable byte stream service* between two processes in two computers across the Internet
- Sequence numbers keep track of the bytes that have been transmitted and received
- Error detection and retransmission used to recover from transmission errors and losses
- TCP is *connection-oriented*: the sender and receiver must first establish an association and set initial sequence numbers before data is transferred
- Connection ID is specified uniquely by

(send port #, send IP address, receive port #, receiver IP address)

Example: HTTP

- HTTP is an application layer protocol
- Retrieves documents on behalf of a browser application program
- HTTP specifies fields in request messages and response messages
  - Request types; Response codes
  - Content type, options, cookies, …
- HTTP specifies actions to be taken upon receipt of certain messages

HTTP uses service of TCP

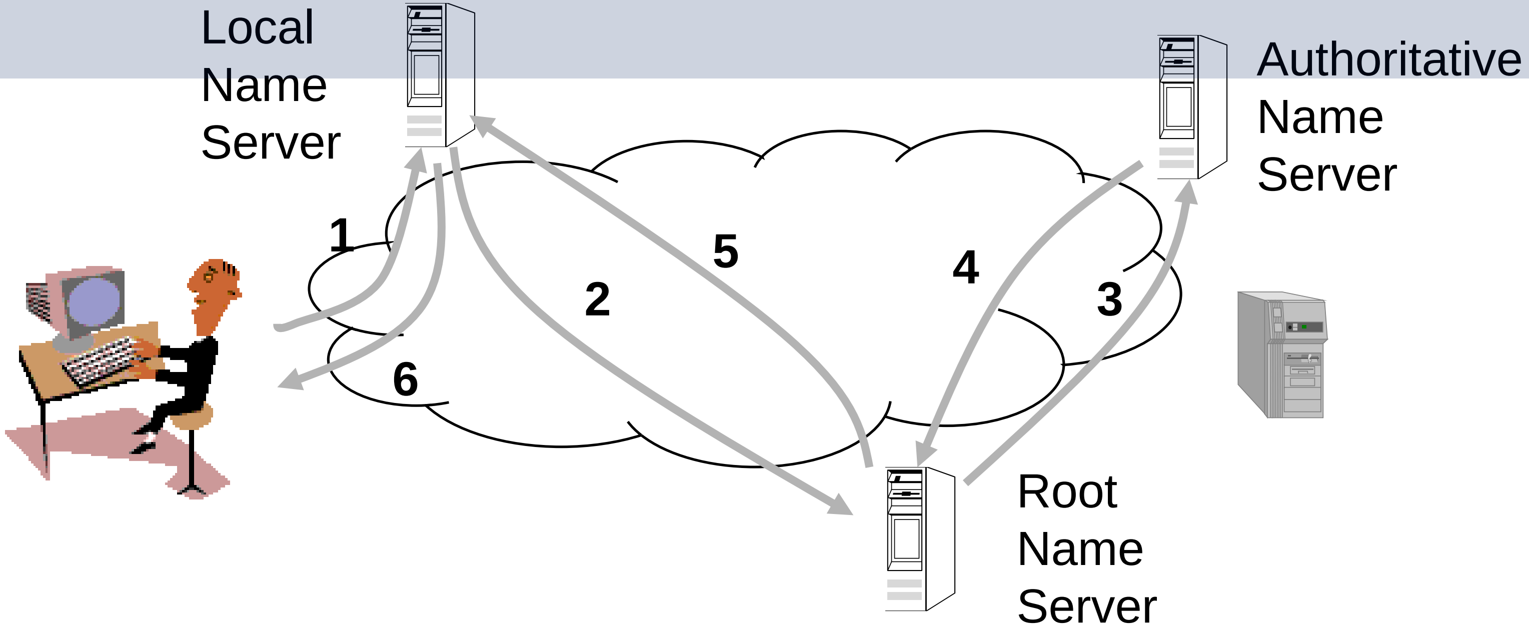


Example: UDP

- UDP is a transport layer protocol
- Provides *best-effort datagram service* between two processes in two computers across the Internet
- Port numbers distinguish various processes in the same machine
- UDP is *connectionless*
- Datagram is sent immediately
- Quick, simple, but not reliable

Example: DNS Protocol

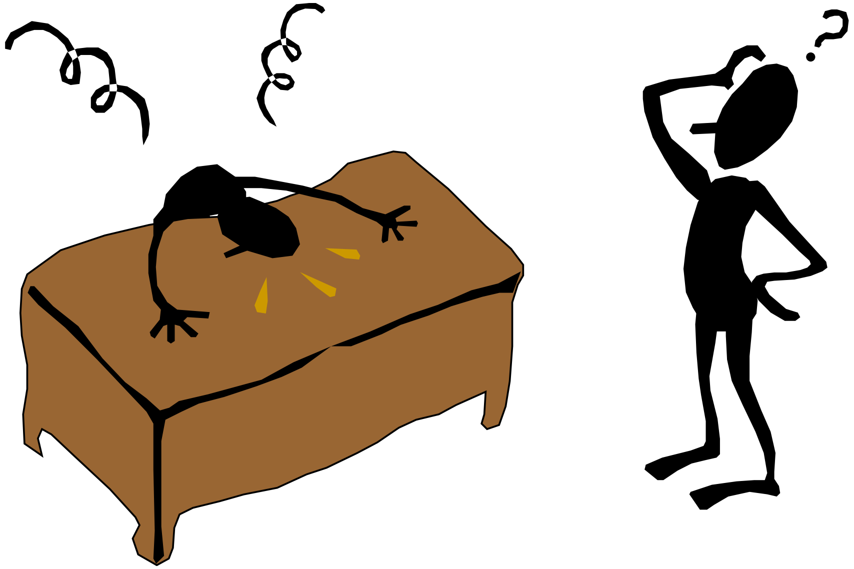
- DNS protocol is an application layer protocol
- DNS is a distributed database that resides in multiple machines in the Internet
- DNS protocol allows queries of different types
  - Name-to-address or Address-to-name
- DNS usually involves short messages and so uses service provided by UDP
- Well-known port 53



- Local Name Server: resolve frequently-used names
  - University department, ISP
  - Contacts Root Name server if it cannot resolve query
- Root Name Servers: 13 globally
  - Resolves query or refers query to Authoritative Name Server
- Authoritative Name Server: last resort
  - Every machine must register its address with at least two authoritative name servers

DNS (More…)

- Click here to open the class note on DNS.



Summary

- Layers: related communications functions
  - Application Layer: HTTP, DNS
  - Transport Layer: TCP, UDP
  - Network Layer: IP
- Services: a protocol provides a communications service to the layer above
  - TCP provides connection-oriented reliable byte transfer service
  - UDP provides best-effort datagram service
- Each layer builds on services of lower layers
  - HTTP builds on top of TCP
  - DNS builds on top of UDP
  - TCP and UDP build on top of IP

Lecture 2
Applications and Layered Architecture

Protocols, Services & Layering

OSI Reference Model

TCP/IP Architecture

How the Layers Work Together

Lecture 2
Applications and Layered Architecture

OSI Reference Model

Why Layering?

- Layering simplifies design, implementation, and testing by partitioning overall communications process into parts
- Protocol in each layer can be designed separately from those in other layers
- Protocol makes “calls” for services from layer below
- Layering provides flexibility for modifying and evolving protocols and services without having to change layers below
- Monolithic non-layered architectures are costly, inflexible, and soon obsolete

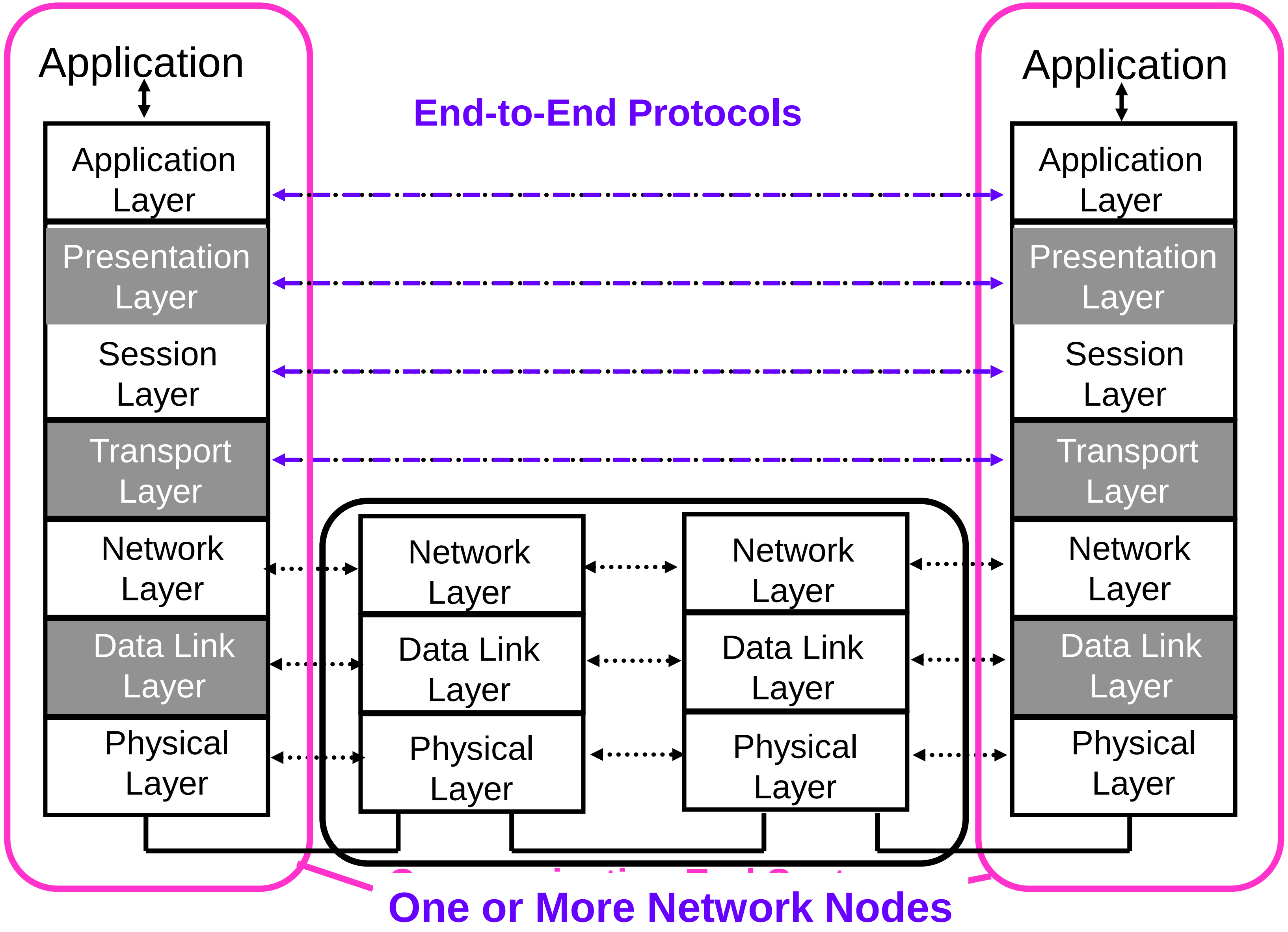
Open Systems Interconnection

- Network architecture:
  - Definition of all the layers
  - Design of protocols for every layer
- By the 1970s every computer vendor had developed its own proprietary layered network architecture
- Problem: computers from different vendors could not be networked together
- Open Systems Interconnection (OSI) was an international effort by the International Organization for Standardization (ISO) to enable multivendor computer interconnection

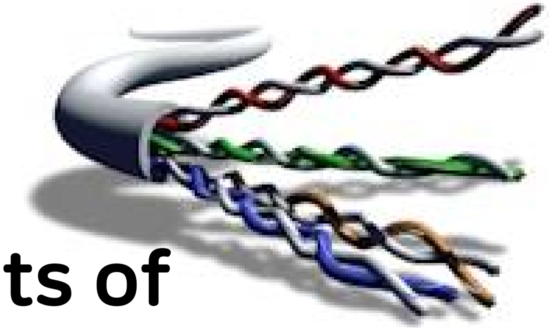
OSI Reference Model

- Describes a seven-layer abstract reference model for a network architecture
- Purpose of the reference model was to provide a framework for the development of protocols
- OSI also provided a unified view of layers, protocols, and services which is still in use in the development of new protocols
- Detailed standards were developed for each layer, but most of these are not in use
- TCP/IP protocols preempted deployment of OSI protocols

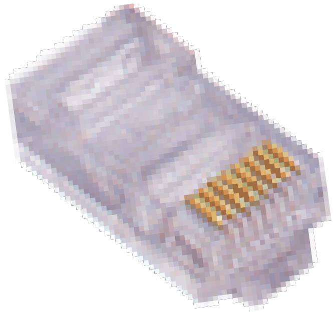
7-Layer OSI Reference Model



Physical Layer



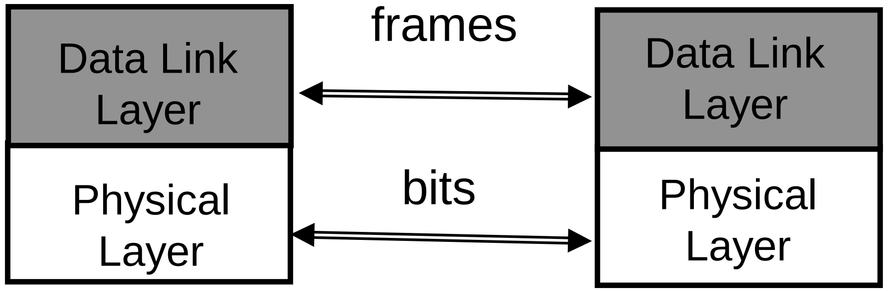
- Transfers bits across link
- Definition & specification of the physical aspects of a communications link
  - Mechanical: cable, plugs, pins...
  - Electrical/optical: modulation, signal strength, voltage levels, bit times, …
  - functional/procedural: how to activate, maintain, and deactivate physical links…



- Ethernet, DSL, cable modem, telephone modems…
- Twisted-pair cable, coaxial cable, optical fiber, radio, infrared, …

Data Link Layer

- Transfers *frames* across *direct* connections
- Groups bits into frames
- Detection of bit errors; Retransmission of frames
- Activation, maintenance, & deactivation of data link connections
- Medium access control for local area networks
- Flow control

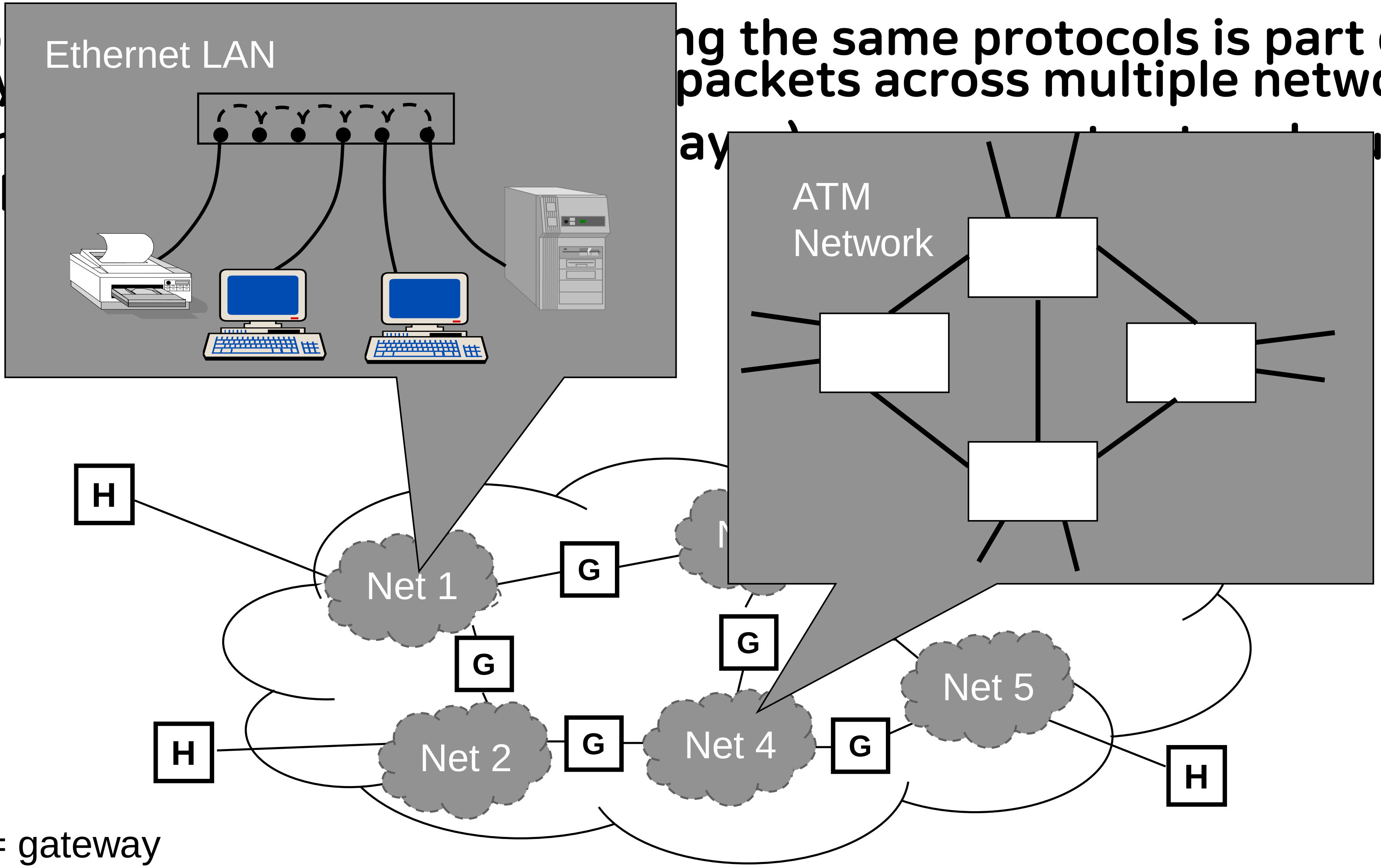


Network Layer

- Transfers *packets* across multiple links and/or multiple networks
- Addressing must scale to large networks
- Nodes *jointly* execute routing algorithm to determine paths across the network
- Forwarding transfers packet across a node
- Congestion control to deal with traffic surges
- Connection setup, maintenance, and teardown when connection-based

Internetworking

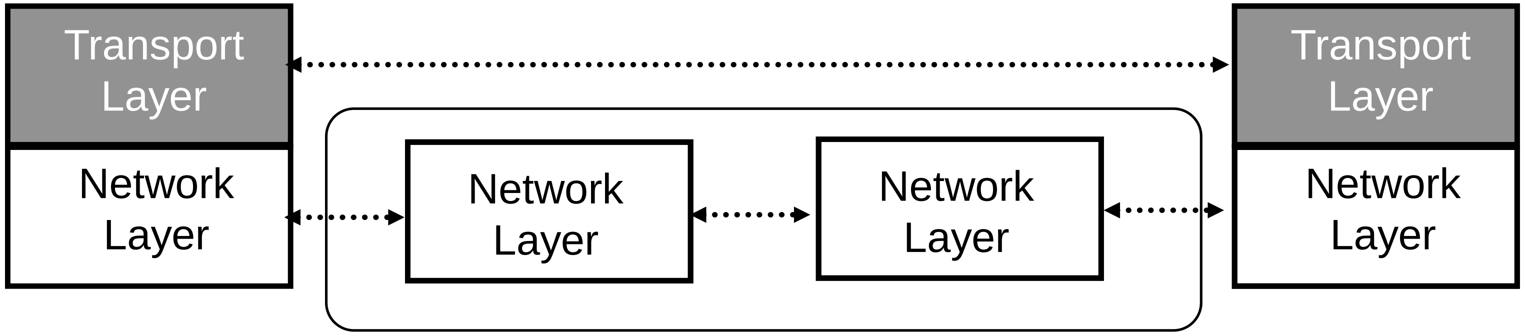
- Internetwo ng the same protocols is part of network lay packets across multiple networks
- Gateway (ir ay using difference



G = gateway
H = host

Transport Layer

- Transfers data end-to-end from process in a machine to process in another machine
- Reliable stream transfer or quick-and-simple single-block transfer
- Multiplexing
- Message segmentation and reassembly
- Connection setup, maintenance, and release

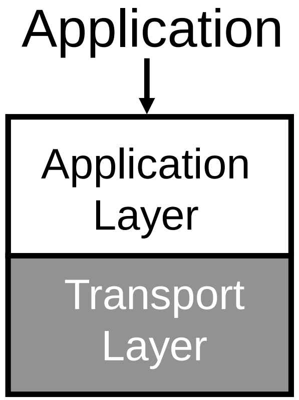


Communication Network

Application & Upper Layers

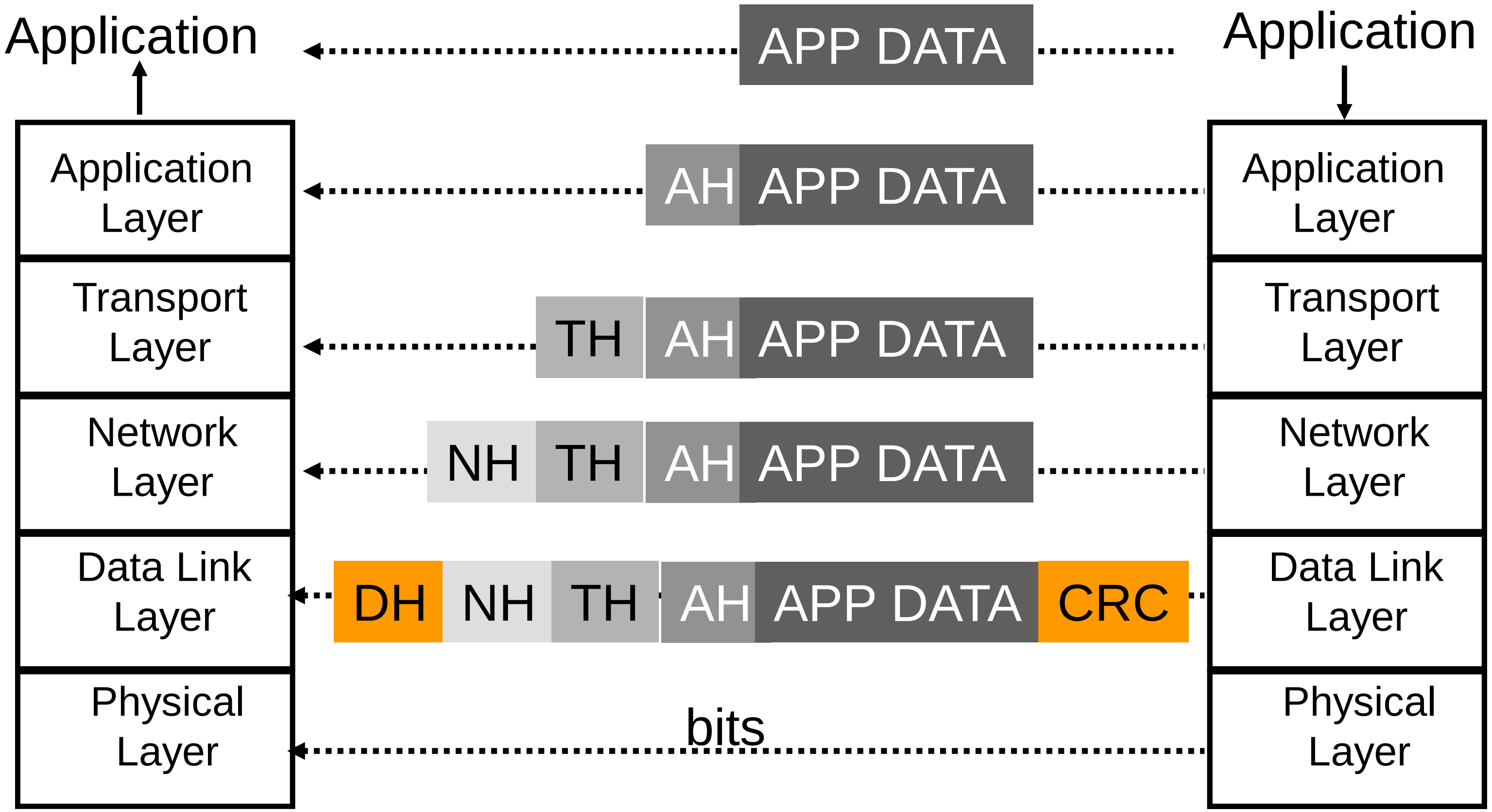
- Application Layer: Provides services that are frequently required by applications: DNS, web access, file transfer, email…
- ~~Presentation Layer: machine-independent representation of data…~~
- ~~Session Layer: dialog management, recovery from errors, …~~

Incorporated into Application Layer



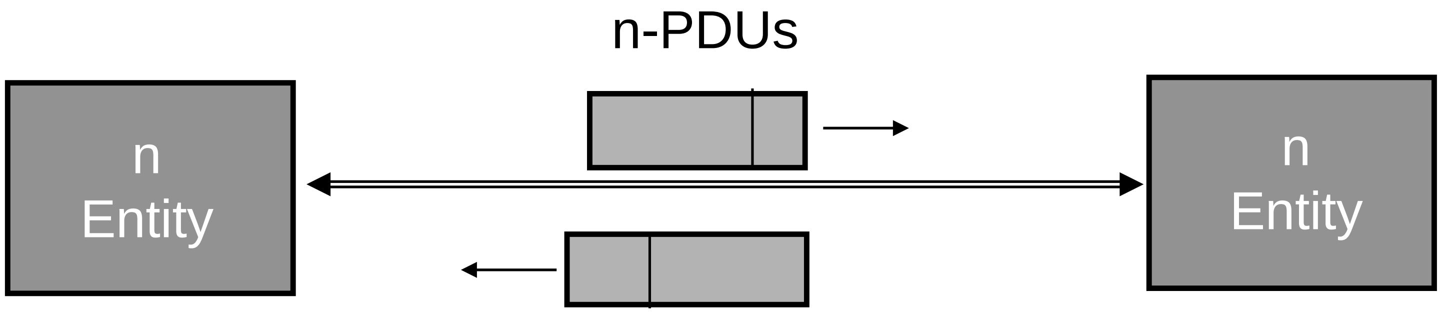
Headers & Trailers

- Each protocol uses a header that carries addresses, sequence numbers, flag bits, length indicators, etc…
- CRC check bits may be appended for error detection



OSI Unified View: Protocols

- Layer n in one machine interacts with layer n in another machine to provide a service to layer n +1
- The entities comprising the corresponding layers on different machines are called *peer processes.*
- The machines use a set of rules and conventions called the *layer-n protocol*.
- Layer-n peer processes communicate by exchanging *Protocol Data Units* (PDUs)

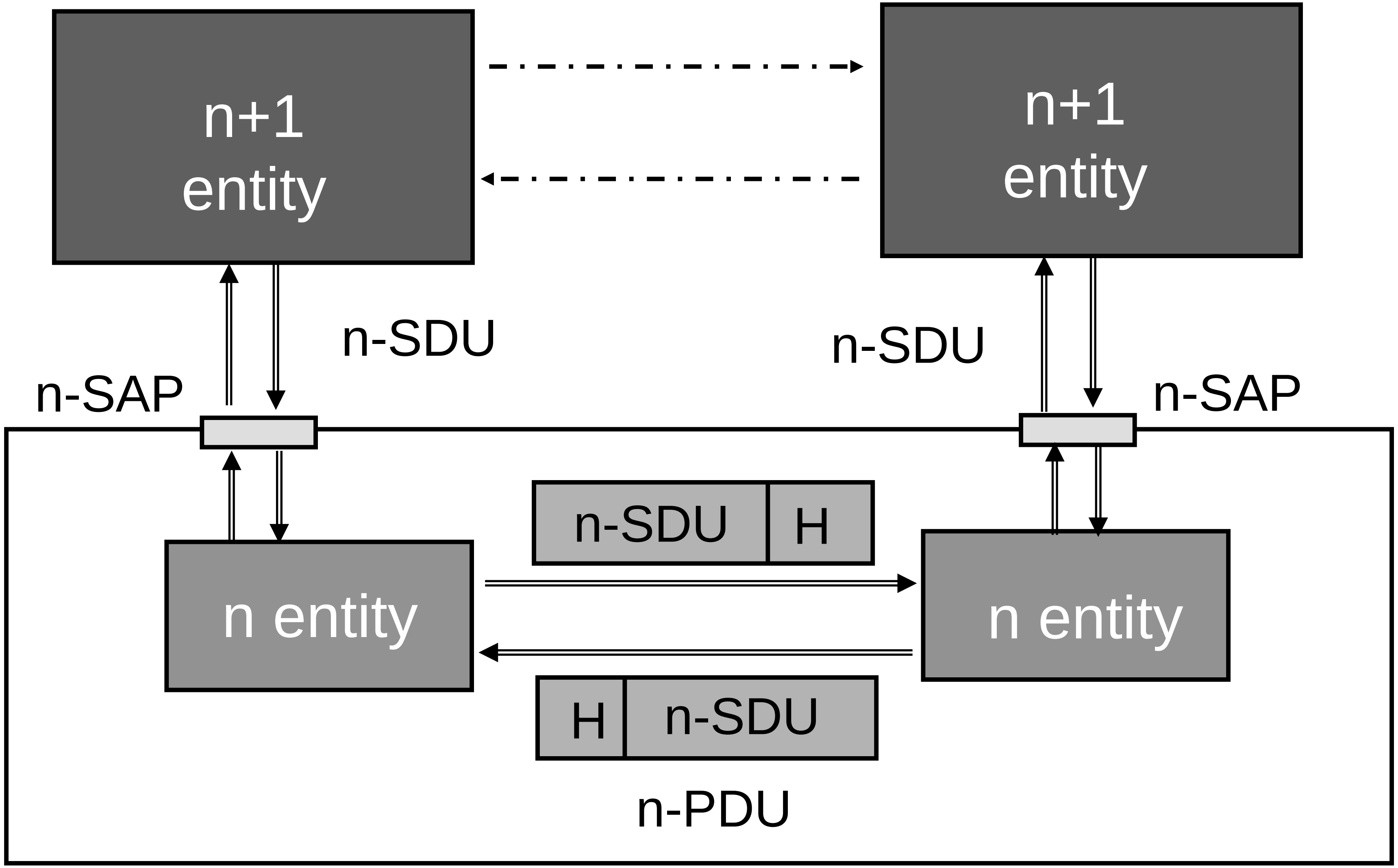


Layer n peer protocol

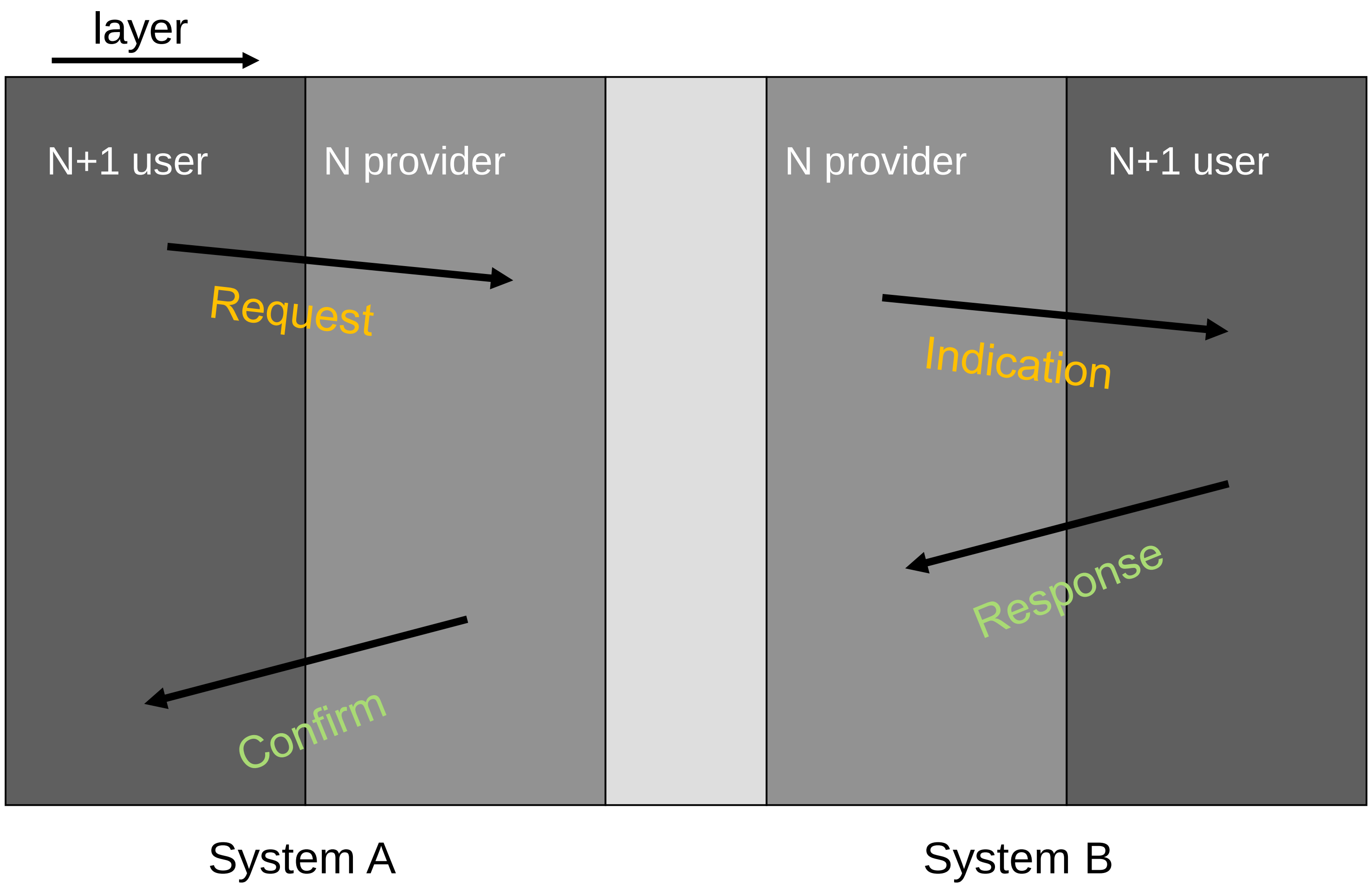
OSI Unified View: Services

- Communication between peer processes is virtual and actually indirect
- Layer n+1 transfers information by invoking the services provided by layer n
- Services are available at *Service Access Points* (SAP's)
- Each layer passes data & control information to the layer below it until the physical layer is reached and transfer occurs
- The data passed to the layer below is called a *Service Data Unit* (SDU)
- SDU's are *encapsulated* in PDU's

Layers, Services & Protocols



Interlayer Interaction



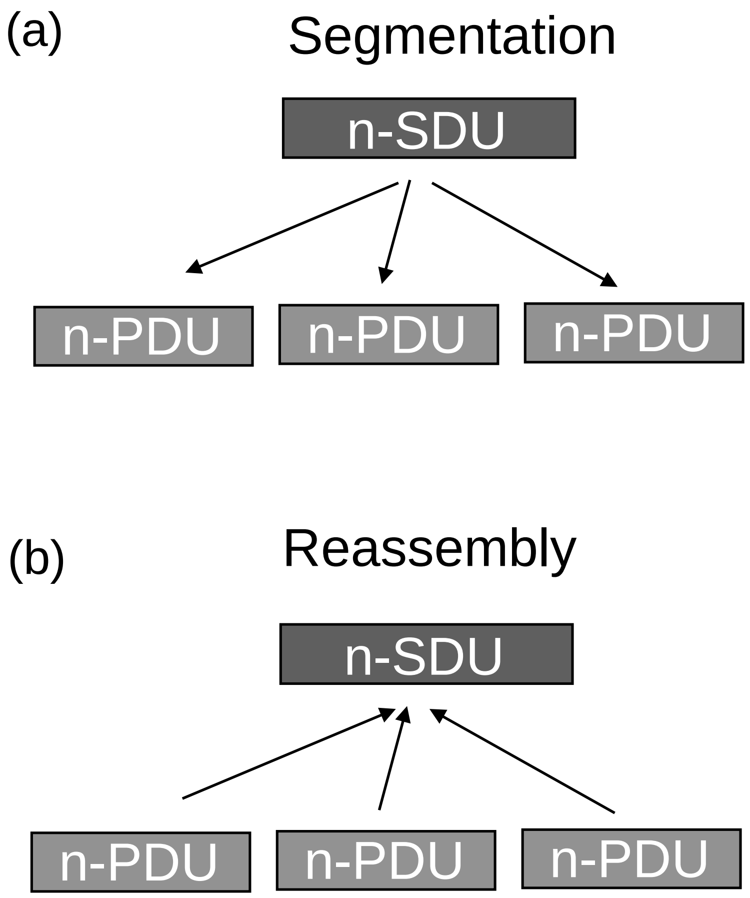
Connectionless & Connection-Oriented Services

- Connection-Oriented
  - Three-phases:
    1. Connection setup between two SAPs to initialize state information
    2. SDU transfer
    3. Connection release
  - E.g. TCP, ATM

- Connectionless
  - Immediate SDU transfer
  - No connection setup
  - E.g. UDP, IP
- Layered services need not be of same type
  - TCP operates over IP
  - IP operates over ATM

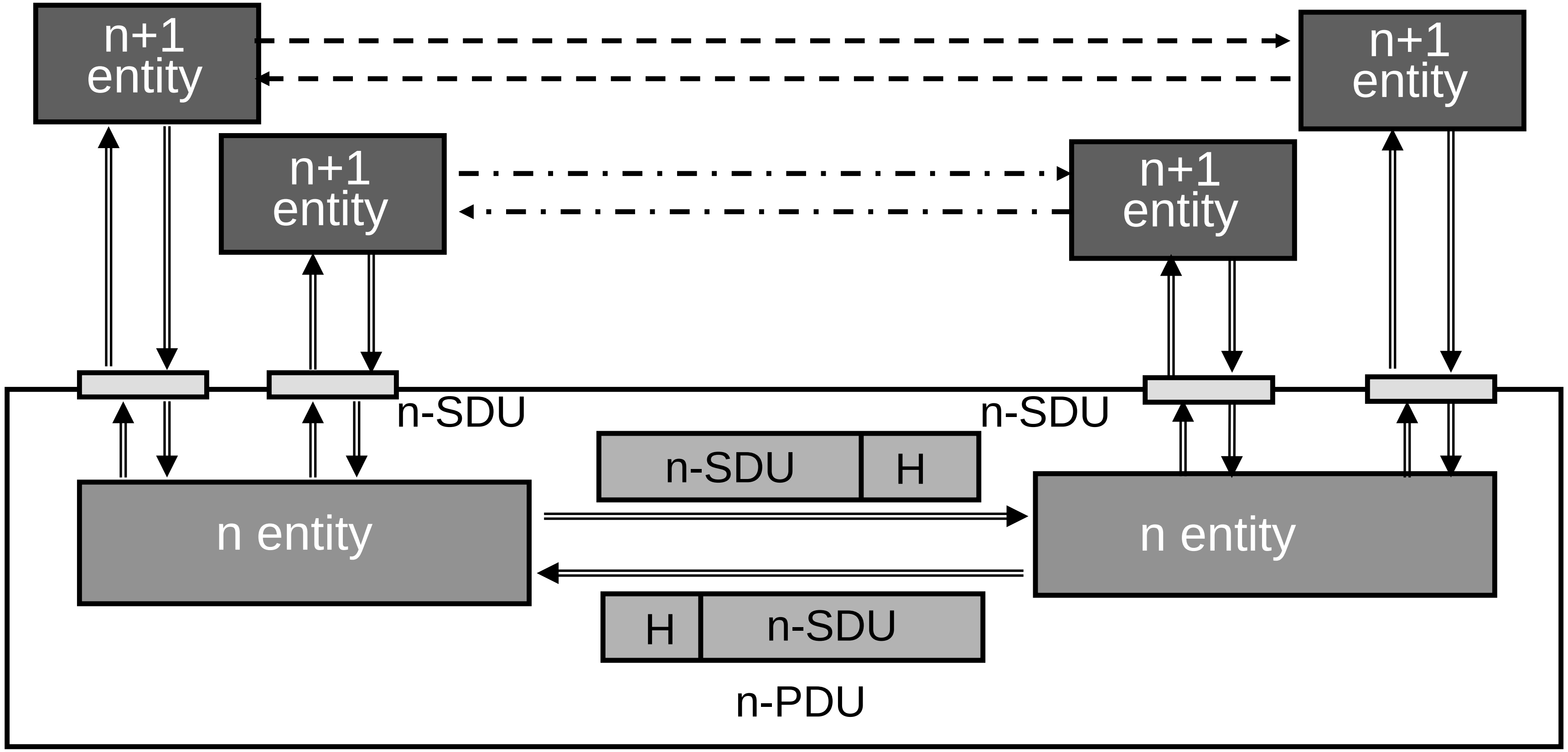
Segmentation & Reassembly

- A layer may impose a limit on the size of a data block that it can transfer for implementation or other reasons
- Thus a layer-n SDU may be too large to be handled as a single unit by layer-(n-1)
- Sender side: SDU is segmented into multiple PDUs
- Receiver side: SDU is reassembled from sequence of PDUs



Multiplexing

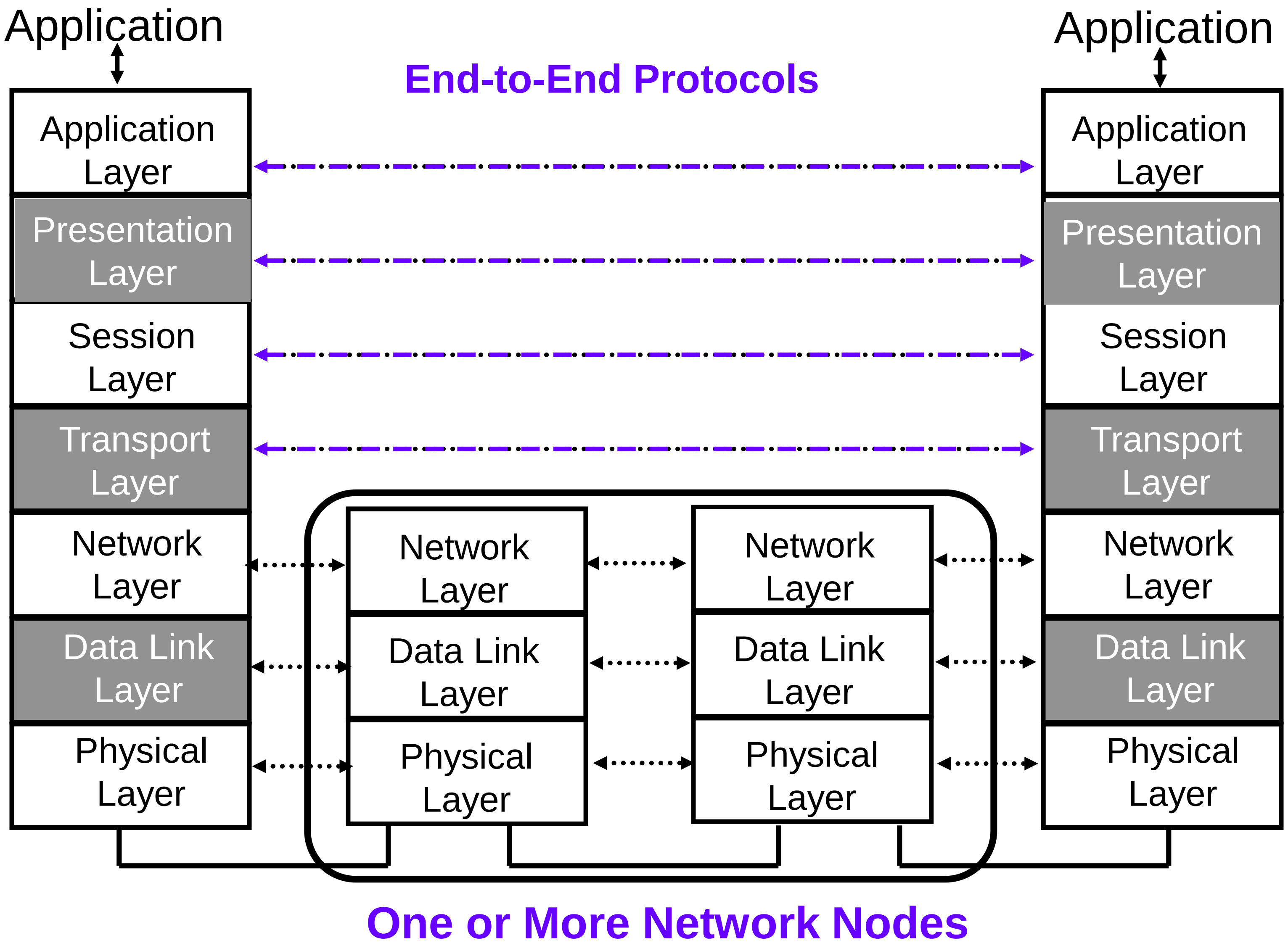
- Sharing of layer n service by *multiple* layer n+1 users
- Multiplexing tag or ID required in each PDU to determine which users an SDU belongs to



Multiplexing

- FDM (Frequency Division Multiplexing)
- TDM (Time Division Multiplexing)
- WDM (Wavelength Division Multiplexing)

Summary:
7-Layer OSI Reference Model



Lecture 2
Applications and Layered Architecture

Protocols, Services & Layering

OSI Reference Model

TCP/IP Architecture

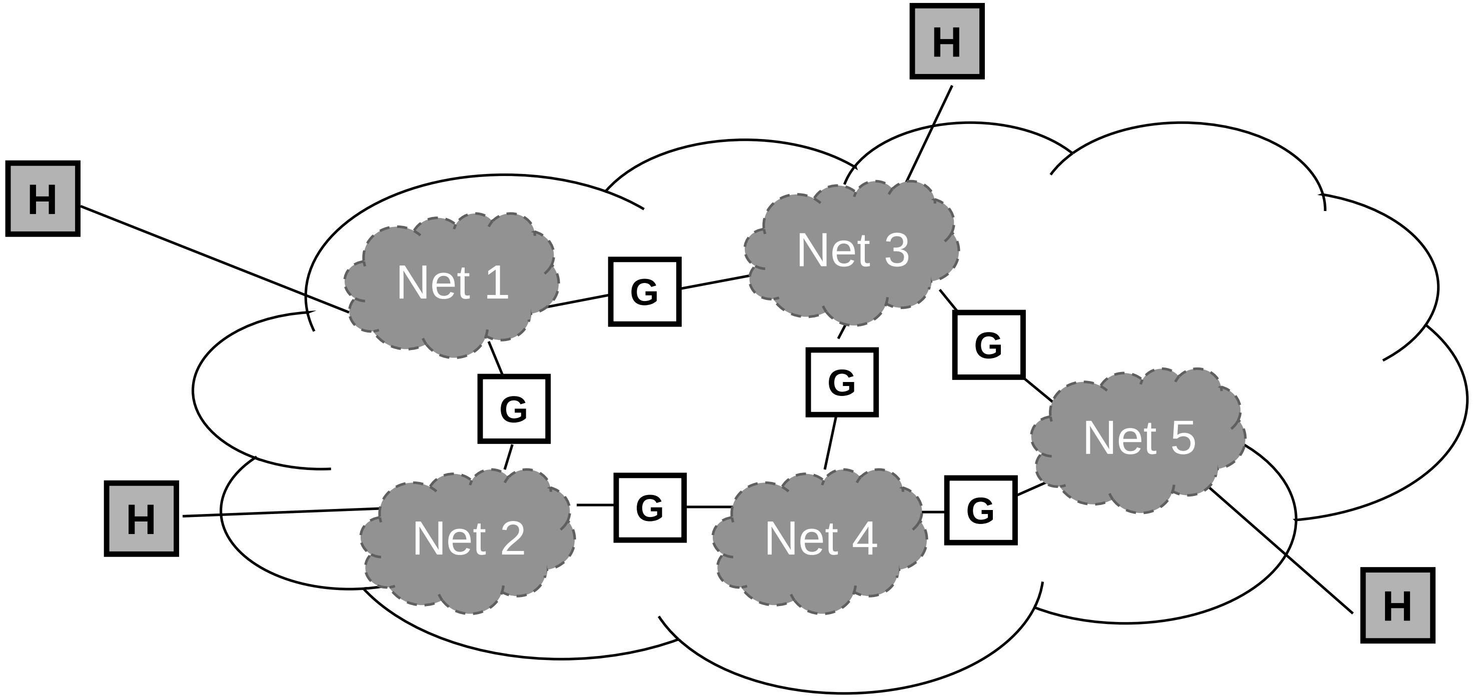
How the Layers Work Together

Lecture 2
Applications and Layered Architecture

TCP/IP Architecture
How the Layers Work Together

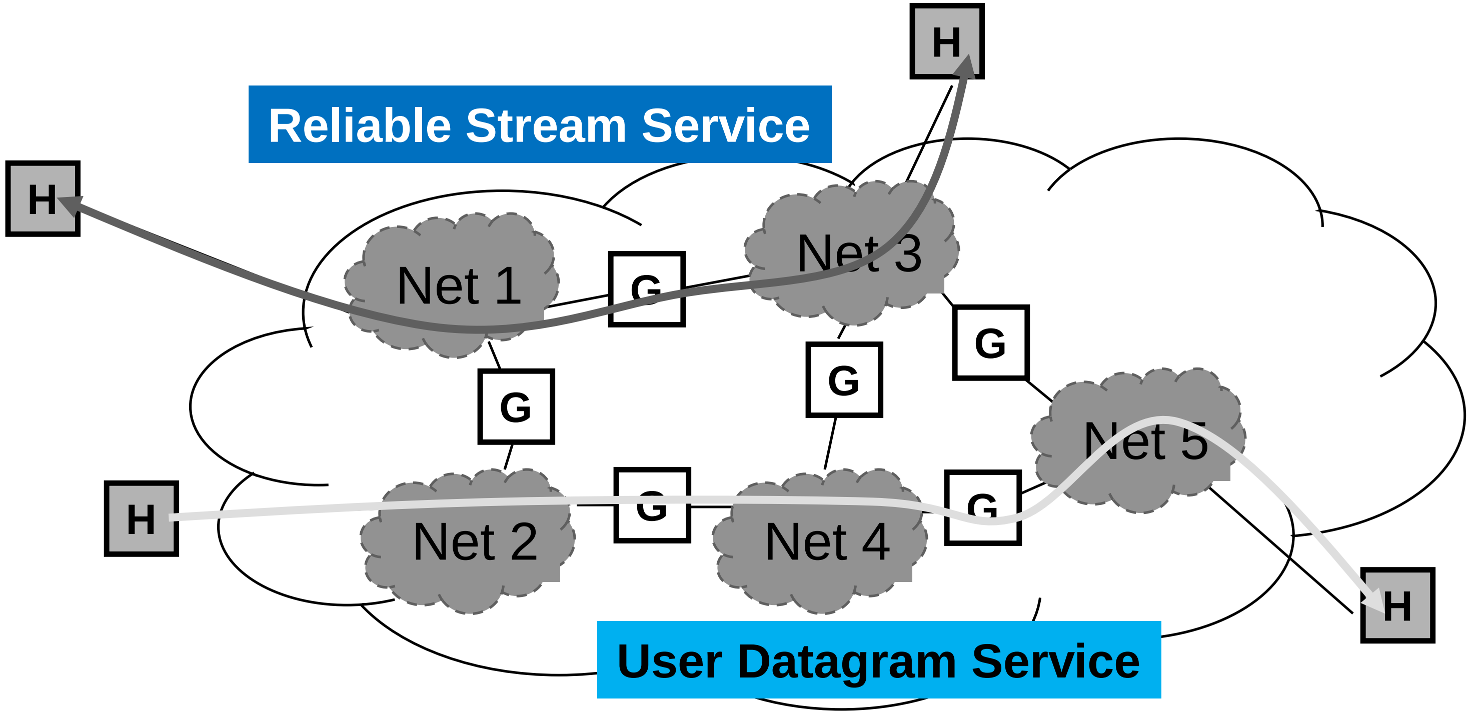
Why Internetworking?

- To build a “network of networks” or internet
  - operating over multiple, coexisting, different network technologies
  - providing ubiquitous connectivity through IP packet transfer
  - achieving huge economies of scale



Why Internetworking?

- To provide universal communication services
  - independent of underlying network technologies
  - providing common interface to user applications

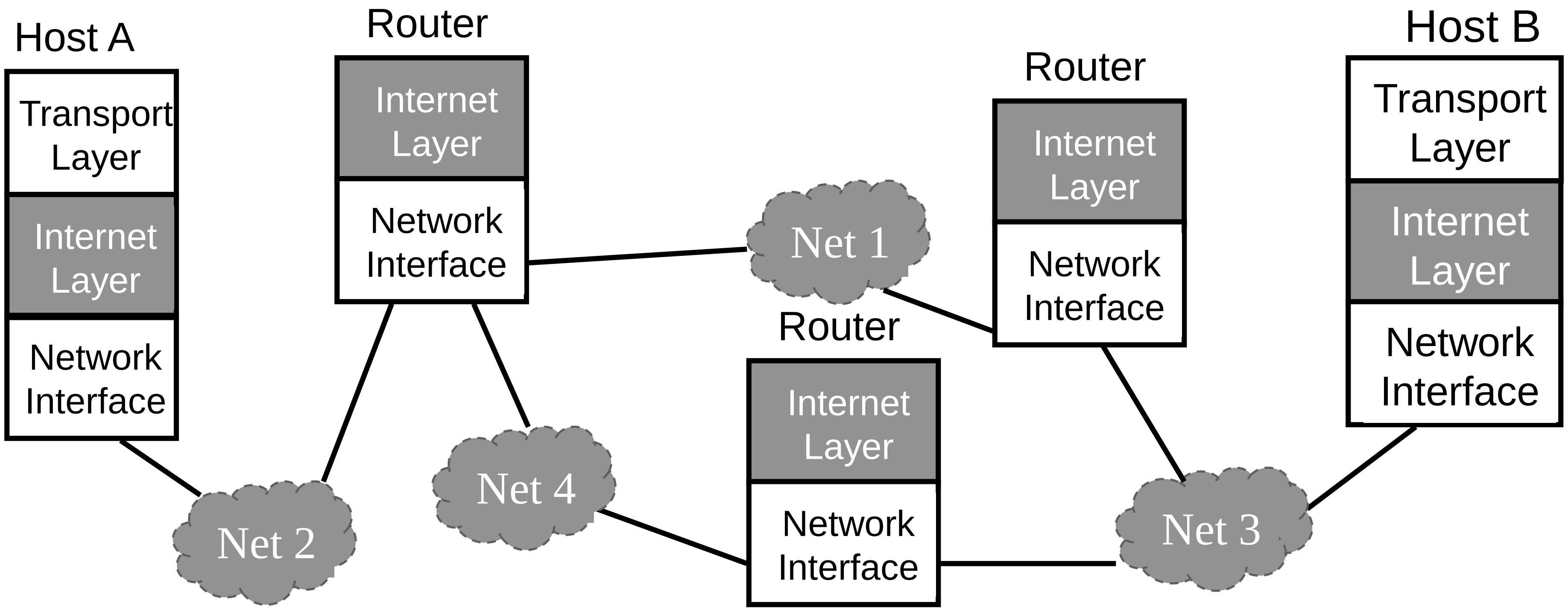


Why Internetworking?

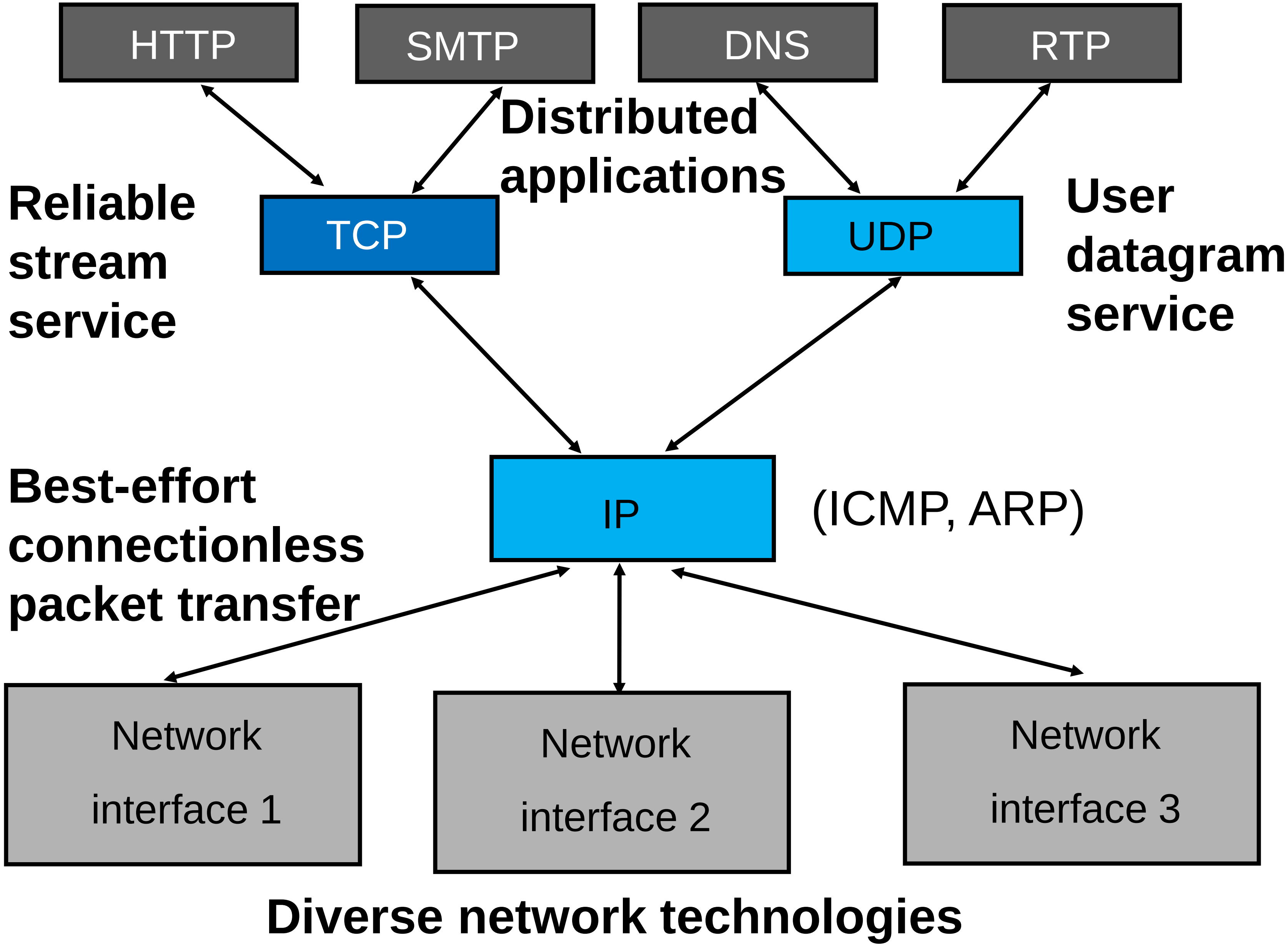
- To provide distributed applications
    - Any application designed to operate based on Internet communication services immediately operates across the entire Internet
    - Rapid deployment of new applications
        - Email, WWW, Peer-to-peer
    - Applications independent of network technology
        - New networks can be introduced below
        - Old network technologies can be retired

Internet Protocol Approach

- IP packets transfer information across Internet
- Host A IP → router→ router…→ router→ Host B IP
- IP layer in each router determines next hop (router)
- Network interfaces transfer IP packets across networks



TCP/IP Protocol Suite



Internet Names & Addresses

Internet Names

- Domain Name
  - Unique name
  - Independent of physical location
  - Facilitate memorization by humans
  - Organization under single administrative unit
- Host Name
  - Name given to host computer
- User Name
  - Name assigned to user

leongarcia@comm.utoronto.ca

Internet Addresses

- Each host has globally unique *logical* 32 bit IP address
- Separate address for each physical connection to a network
- Routing decision is done based on destination IP address
- IP address has two parts:
  - *netid* and *hostid*
  - *netid* unique
  - *netid* facilitates routing
- Dotted Decimal Notation:

  int1.int2.int3.int4

  (intj = jth octet)

  128.100.10.13

DNS resolves IP name to IP address

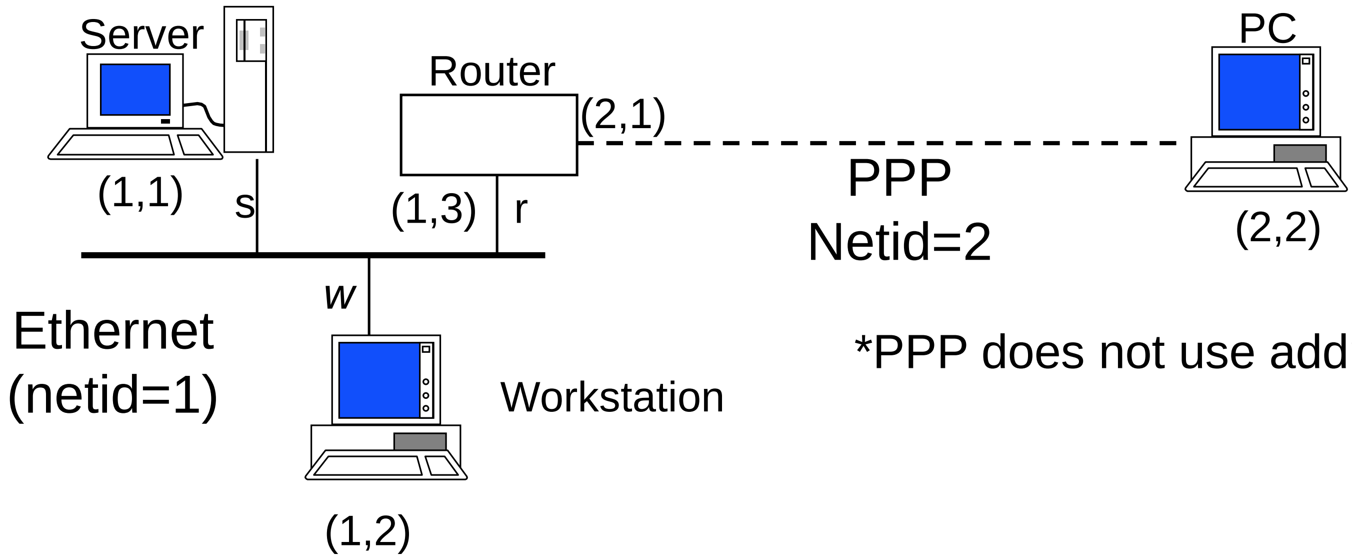
Physical Addresses

- LANs (and other networks) assign physical addresses to the physical attachment to the network
- The network uses its own address to transfer packets or frames to the appropriate destination
- IP address needs to be resolved to physical address at each IP network interface
- Example: Ethernet uses 48-bit addresses
  - Each Ethernet network interface card (NIC) has globally unique Medium Access Control (MAC) or physical address
  - First 24 bits identify NIC manufacturer; second 24 bits are serial number
  - $00{:}90{:}27{:}\underbrace{96{:}68{:}07}_{\text{Intel}}$ 12 hex numbers

More Information on IP Address and Subnetting

- Click here or go to Lecture 2-1 Note for more information on IP addressing and Subnetting

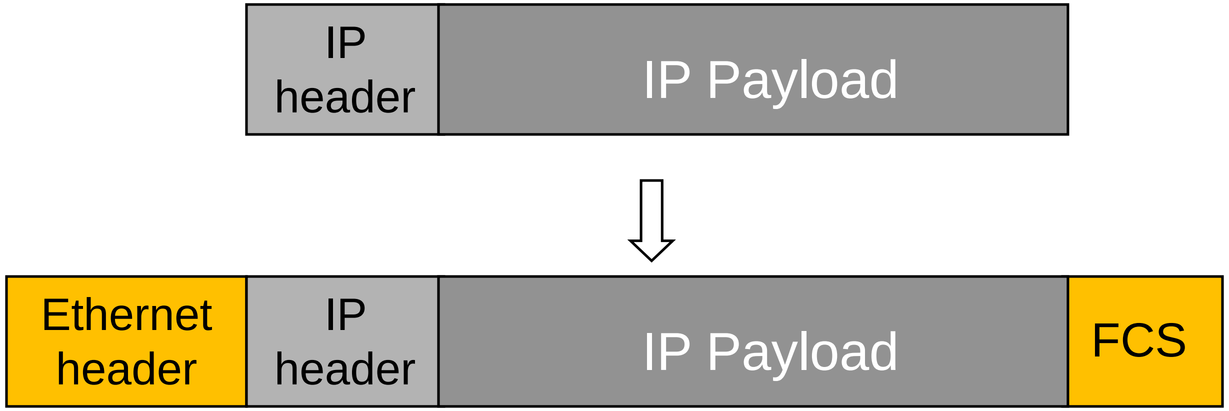
Example internet



*PPP does not use addresses

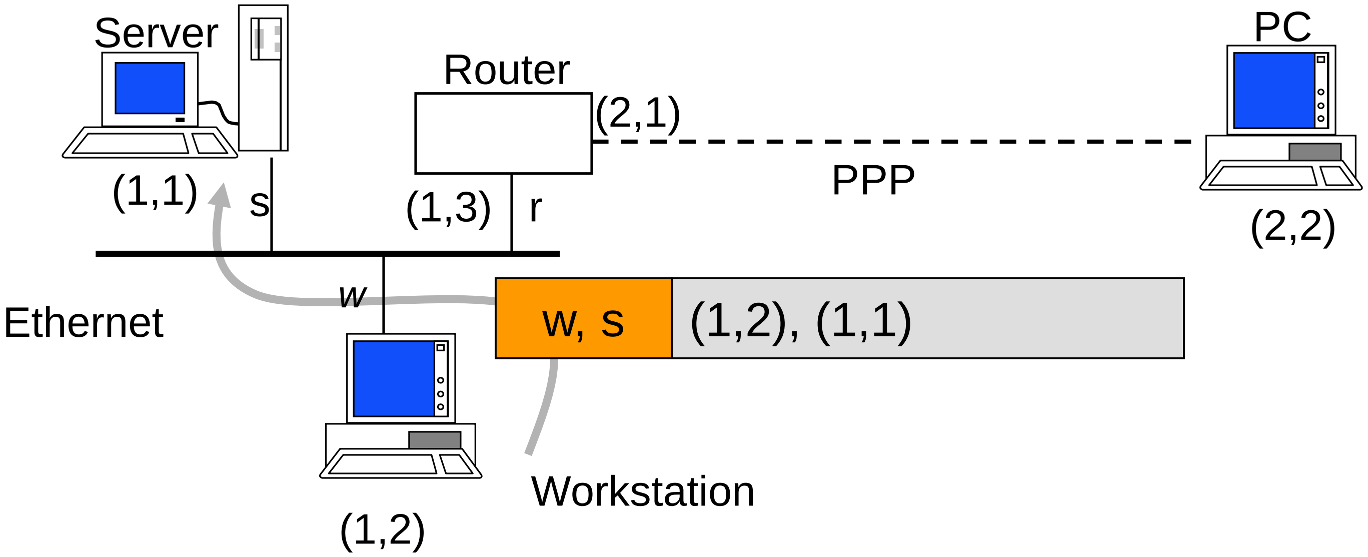
	netid	hostid	Physical address
server	1	1	s
workstation	1	2	w
router	1	3	r
router	2	1	-
PC	2	2	-

Encapsulation



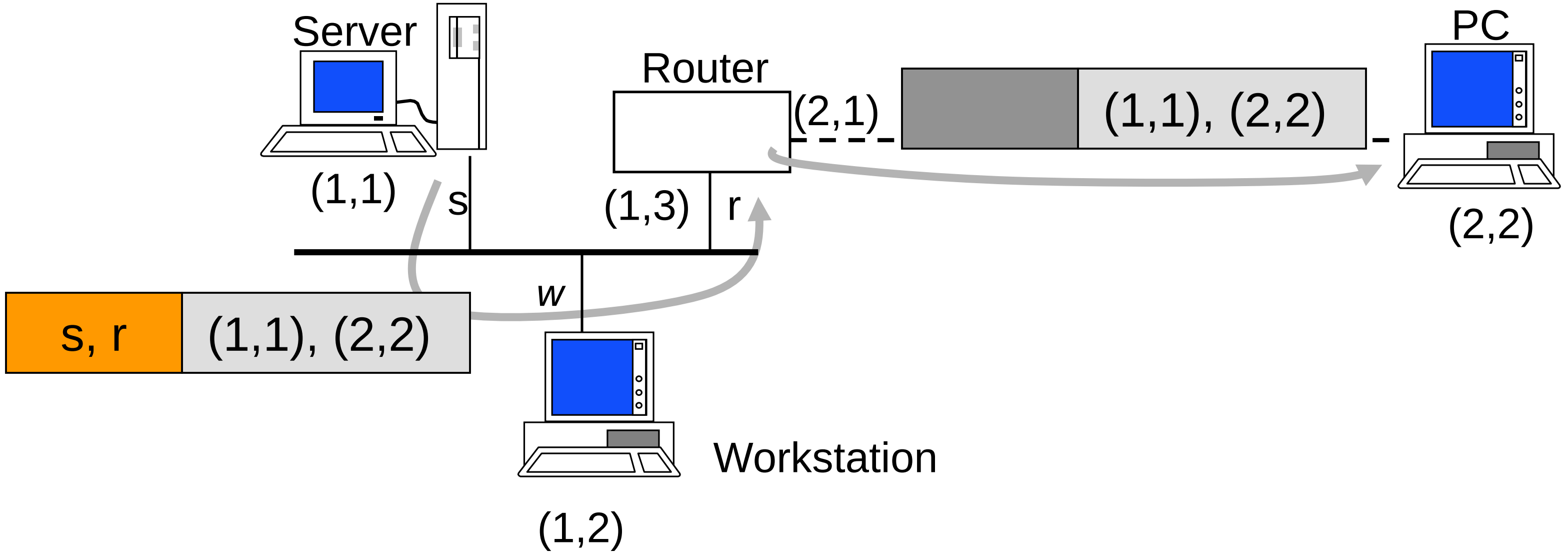
- Ethernet header contains:
  - source and destination physical addresses
  - network protocol type (e.g. IP)

IP packet from workstation to server



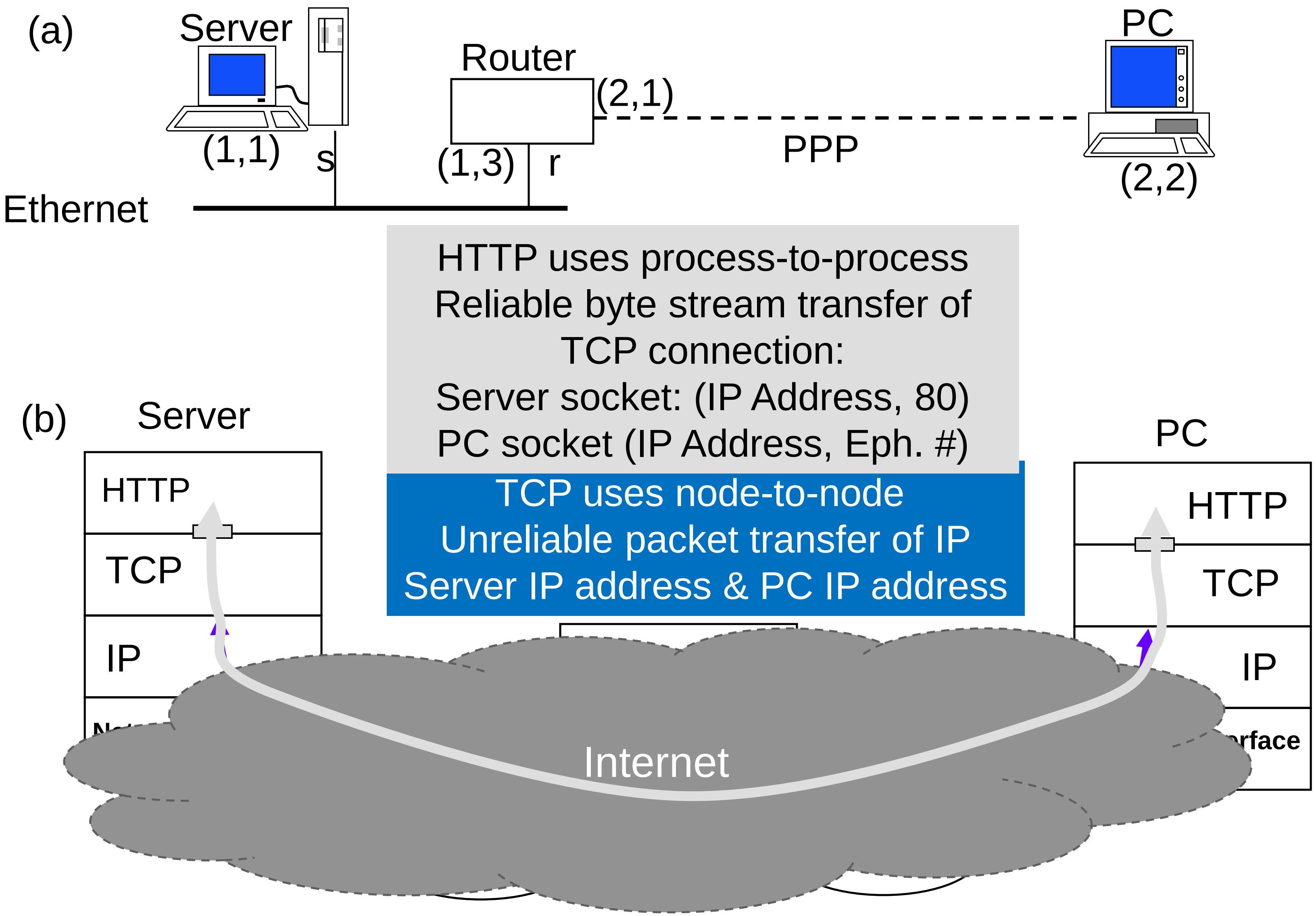
1. IP packet has (1,2) IP address for source and (1,1) IP address for destination
2. IP table at workstation indicates (1,1) connected to same network, so IP packet is encapsulated in Ethernet frame with addresses w and s
3. Ethernet frame is broadcast by workstation NIC and captured by server NIC
4. NIC examines protocol type field and then delivers packet to its IP layer

IP packet from server to PC



1. IP packet has (1,1) and (2,2) as IP source and destination addresses
2. IP table at server indicates packet should be sent to router, so IP packet is encapsulated in Ethernet frame with addresses s and r
3. Ethernet frame is broadcast by server NIC and captured by router NIC
4. NIC examines protocol type field and then delivers packet to its IP layer
5. IP layer examines IP packet destination address and determines IP packet should be routed to (2,2)
6. Router's table indicates (2,2) is directly connected via PPP link
7. IP packet is encapsulated in PPP frame and delivered to PC
8. PPP at PC examines protocol type field and delivers packet to PC IP layer

How the layers work together



HTTP uses process-to-process Reliable byte stream transfer of TCP connection:
Server socket: (IP Address, 80)
PC socket (IP Address, Eph. #)

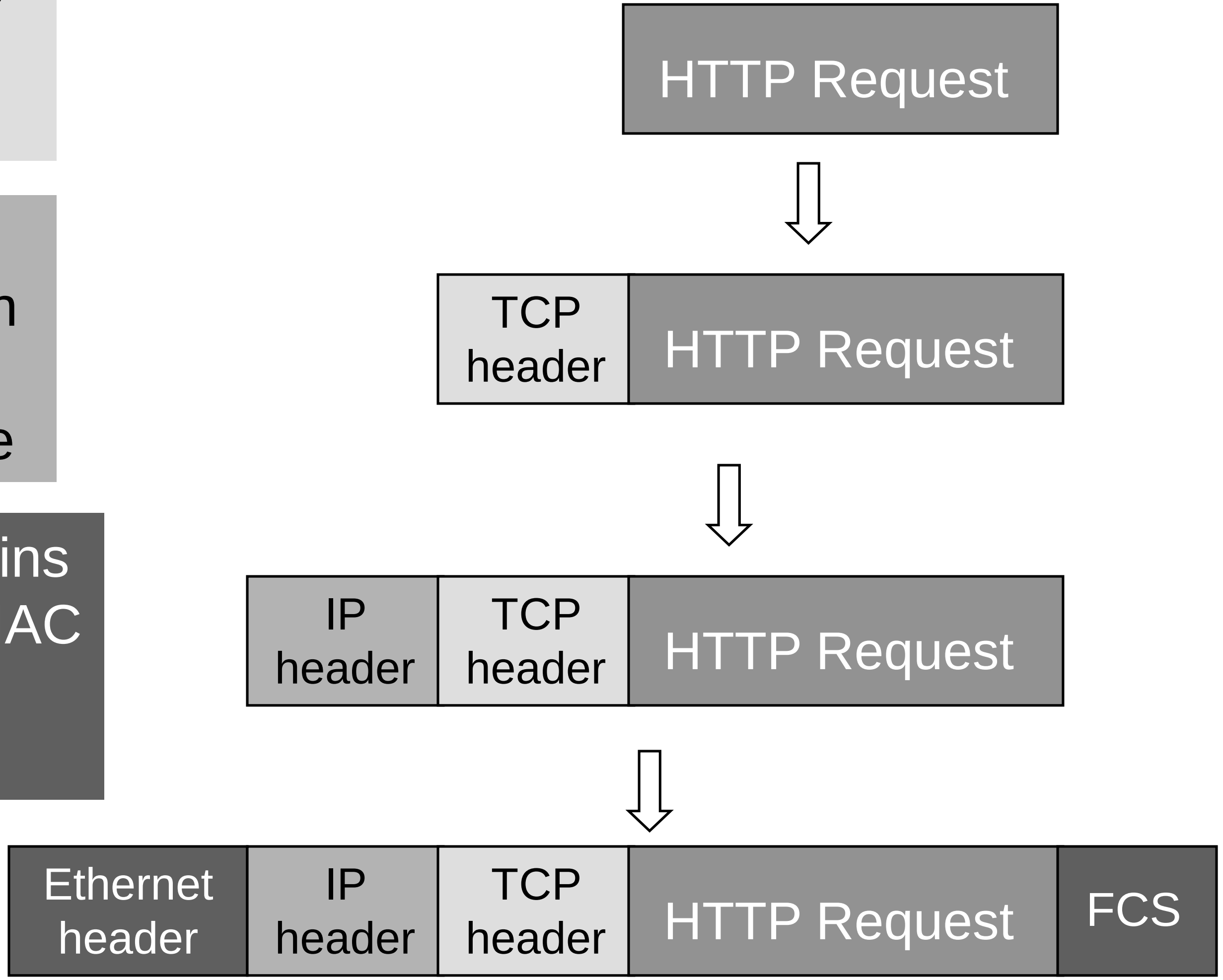
TCP uses node-to-node Unreliable packet transfer of IP
Server IP address & PC IP address

Encapsulation

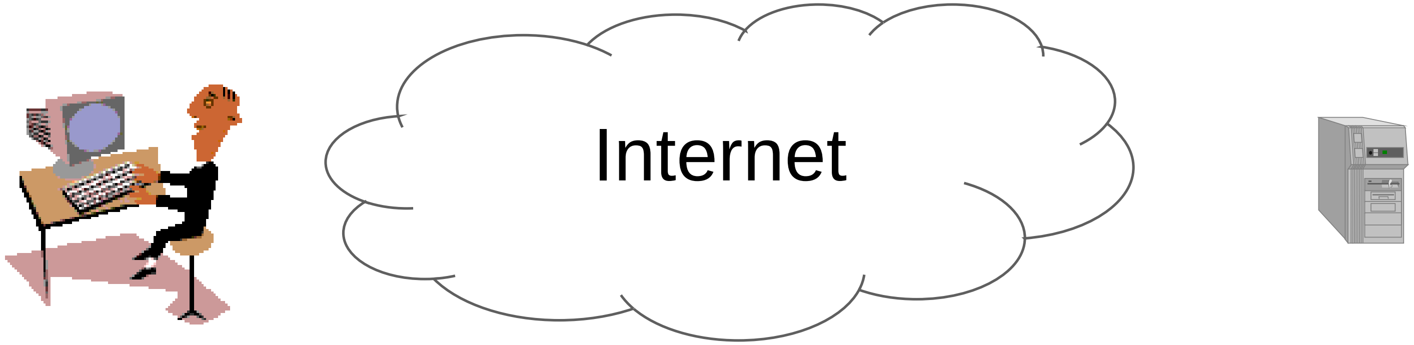
TCP Header contains source & destination port numbers

IP Header contains source and destination IP addresses; transport protocol type

Ethernet Header contains source & destination MAC addresses;
network protocol type

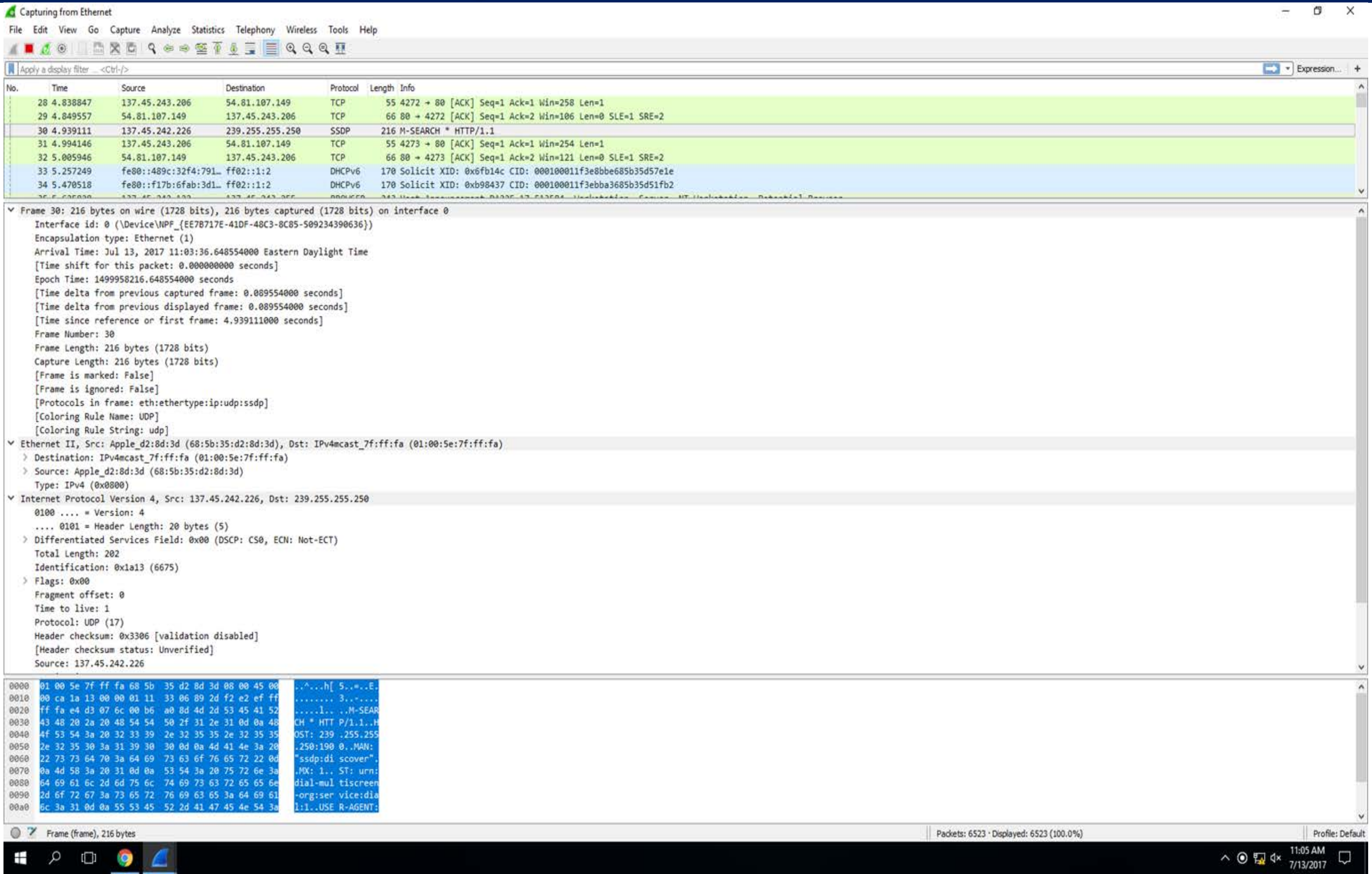


How the layers work together: Network Analyzer



- *Wireshark* network analyzer captures all frames observed by its Ethernet NIC
- Sequence of frames and contents of frame can be examined in detail down to individual bytes

Wireshark.org



```
No.   Time        Source               Destination       Protocol  Length Info
   28 4.838847    137.45.243.206       54.81.107.149     TCP          55 4272 → 80 [ACK] Seq=1 Ack=1 Win=258 Len=1
   29 4.849557    54.81.107.149        137.45.243.206    TCP          66 80 → 4272 [ACK] Seq=1 Ack=2 Win=106 Len=0 SLE=1 SRE=2
   30 4.939111    137.45.242.226       239.255.255.250   SSDP        216 M-SEARCH * HTTP/1.1
   31 4.994146    137.45.243.206       54.81.107.149     TCP          55 4273 → 80 [ACK] Seq=1 Ack=1 Win=254 Len=1
   32 5.005946    54.81.107.149        137.45.243.206    TCP          66 80 → 4273 [ACK] Seq=1 Ack=2 Win=121 Len=0 SLE=1 SRE=2
   33 5.257249    fe80::489c:32f4:791… ff02::1:2         DHCPv6      170 Solicit XID: 0x6fb14c CID: 000100011f3e8bbe685b35d57e1e
   34 5.470518    fe80::f17b:6fab:3d1… ff02::1:2         DHCPv6      170 Solicit XID: 0xb98437 CID: 000100011f3ebba3685b35d51fb2
```

```
˅ Frame 30: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits) on interface 0
    Interface id: 0 (\Device\NPF_{EE7B717E-41DF-48C3-8C85-509234390636})
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul 13, 2017 11:03:36.648554000 Eastern Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1499958216.648554000 seconds
    [Time delta from previous captured frame: 0.089554000 seconds]
    [Time delta from previous displayed frame: 0.089554000 seconds]
    [Time since reference or first frame: 4.939111000 seconds]
    Frame Number: 30
    Frame Length: 216 bytes (1728 bits)
    Capture Length: 216 bytes (1728 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:ssdp]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
˅ Ethernet II, Src: Apple_d2:8d:3d (68:5b:35:d2:8d:3d), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  › Destination: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  › Source: Apple_d2:8d:3d (68:5b:35:d2:8d:3d)
    Type: IPv4 (0x0800)
˅ Internet Protocol Version 4, Src: 137.45.242.226, Dst: 239.255.255.250
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  › Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 202
    Identification: 0x1a13 (6675)
  › Flags: 0x00
    Fragment offset: 0
    Time to live: 1
    Protocol: UDP (17)
    Header checksum: 0x3306 [validation disabled]
    [Header checksum status: Unverified]
    Source: 137.45.242.226
```

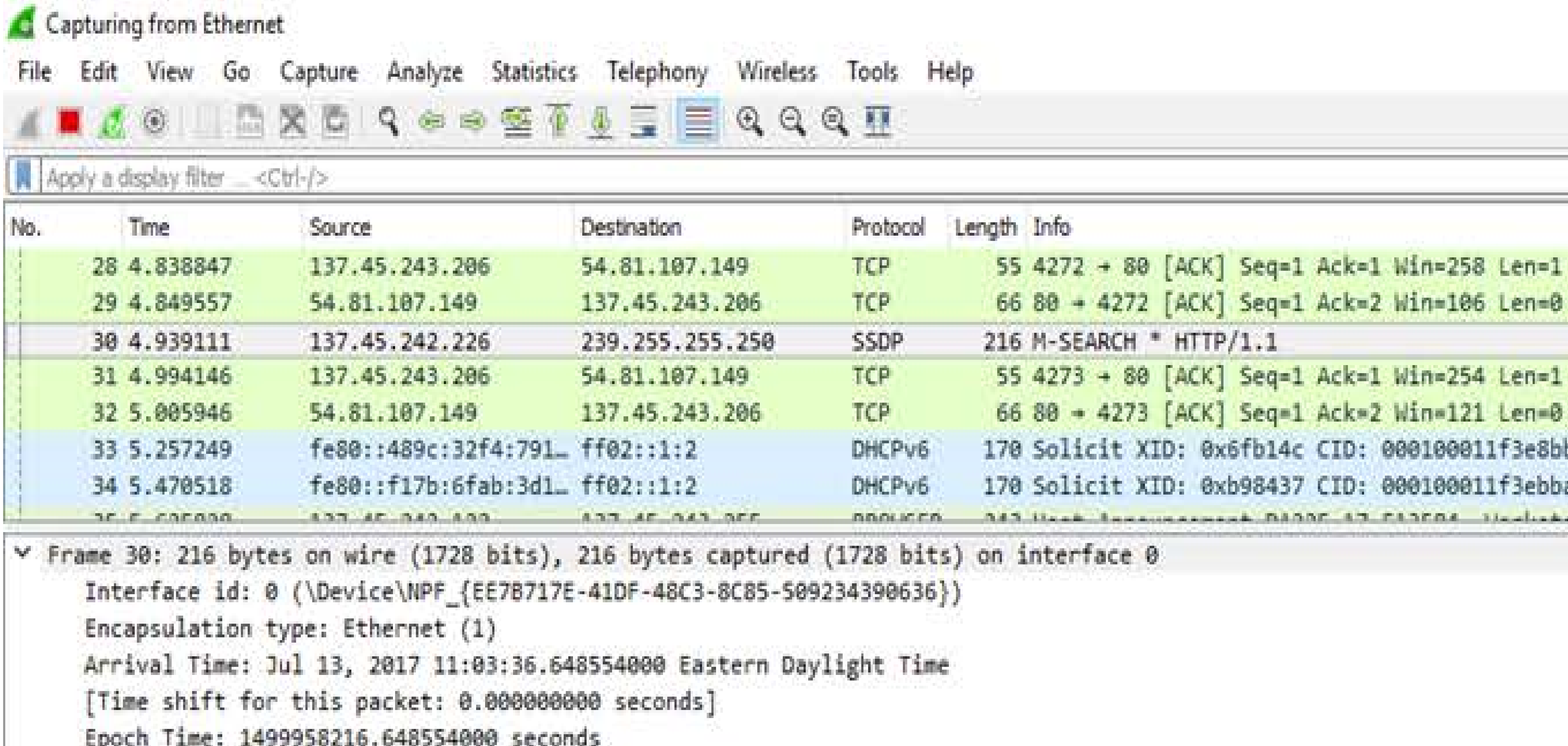
```
0000  01 00 5e 7f ff fa 68 5b  35 d2 8d 3d 08 00 45 00   ..^...h[ 5..=..E.
0010  00 ca 1a 13 00 00 01 11  33 06 89 2d f2 e2 ef ff   ........ 3..-....
0020  ff fa e4 d3 07 6c 00 b6  a0 8d 4d 2d 53 45 41 52   .....l.. ..M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1..H
0040  4f 53 54 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   OST: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 4d 41 4e 3a 20   .250:190 0..MAN: 
0060  22 73 73 64 70 3a 64 69  73 63 6f 76 65 72 22 0d   "ssdp:di scover".
0070  0a 4d 58 3a 20 31 0d 0a  53 54 3a 20 75 72 6e 3a   .MX: 1.. ST: urn:
0080  64 69 61 6c 2d 6d 75 6c  74 69 73 63 72 65 65 6e   dial-mul tiscreen
0090  2d 6f 72 67 3a 73 65 72  76 69 63 65 3a 64 69 61   -org:ser vice:dia
00a0  6c 3a 31 0d 0a 55 53 45  52 2d 41 47 45 4e 54 3a   l:1..USE R-AGENT:
```

Frame (frame), 216 bytes

Packets: 6523 · Displayed: 6523 (100.0%)

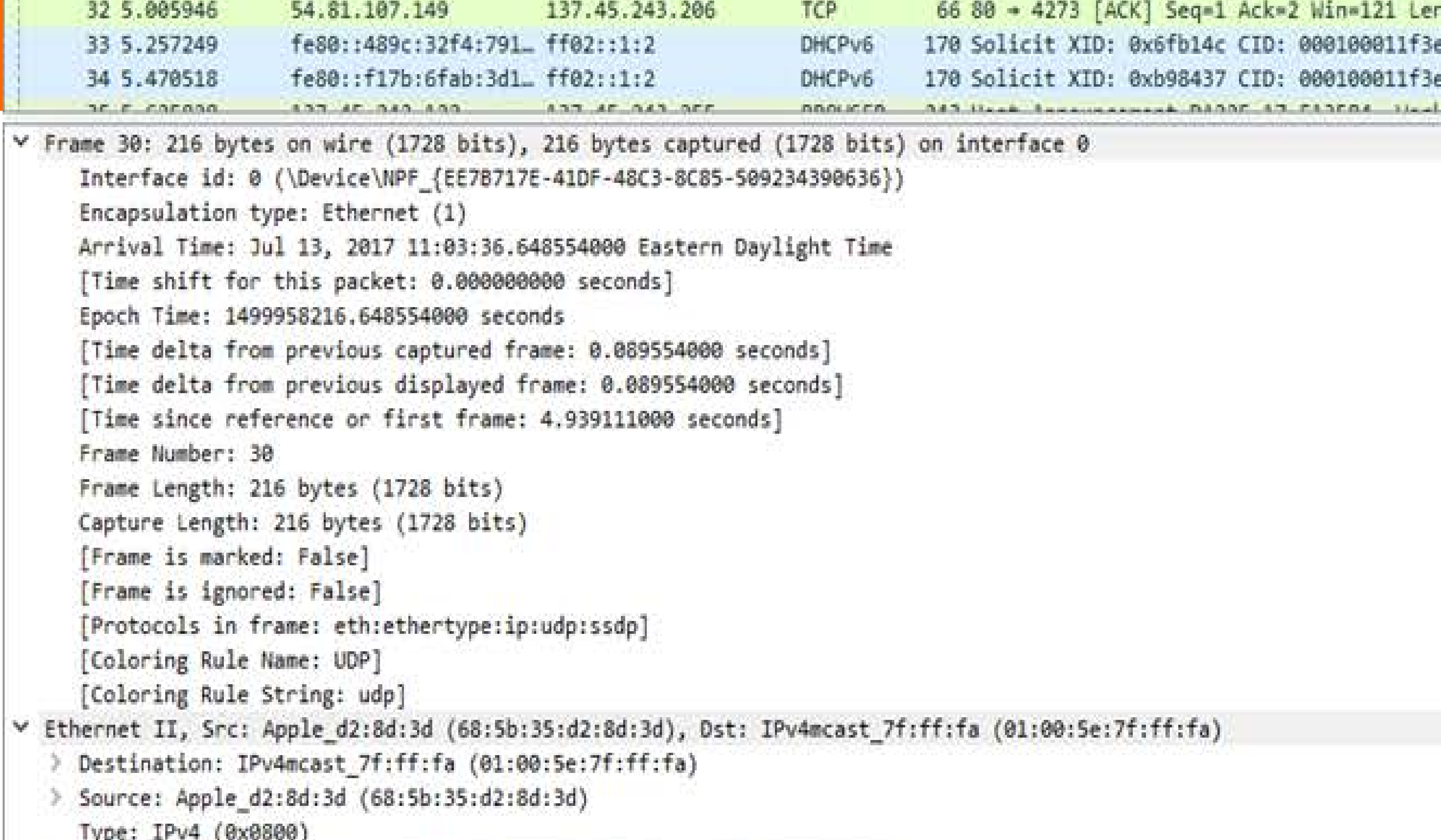
Profile: Default

Wireshark.org



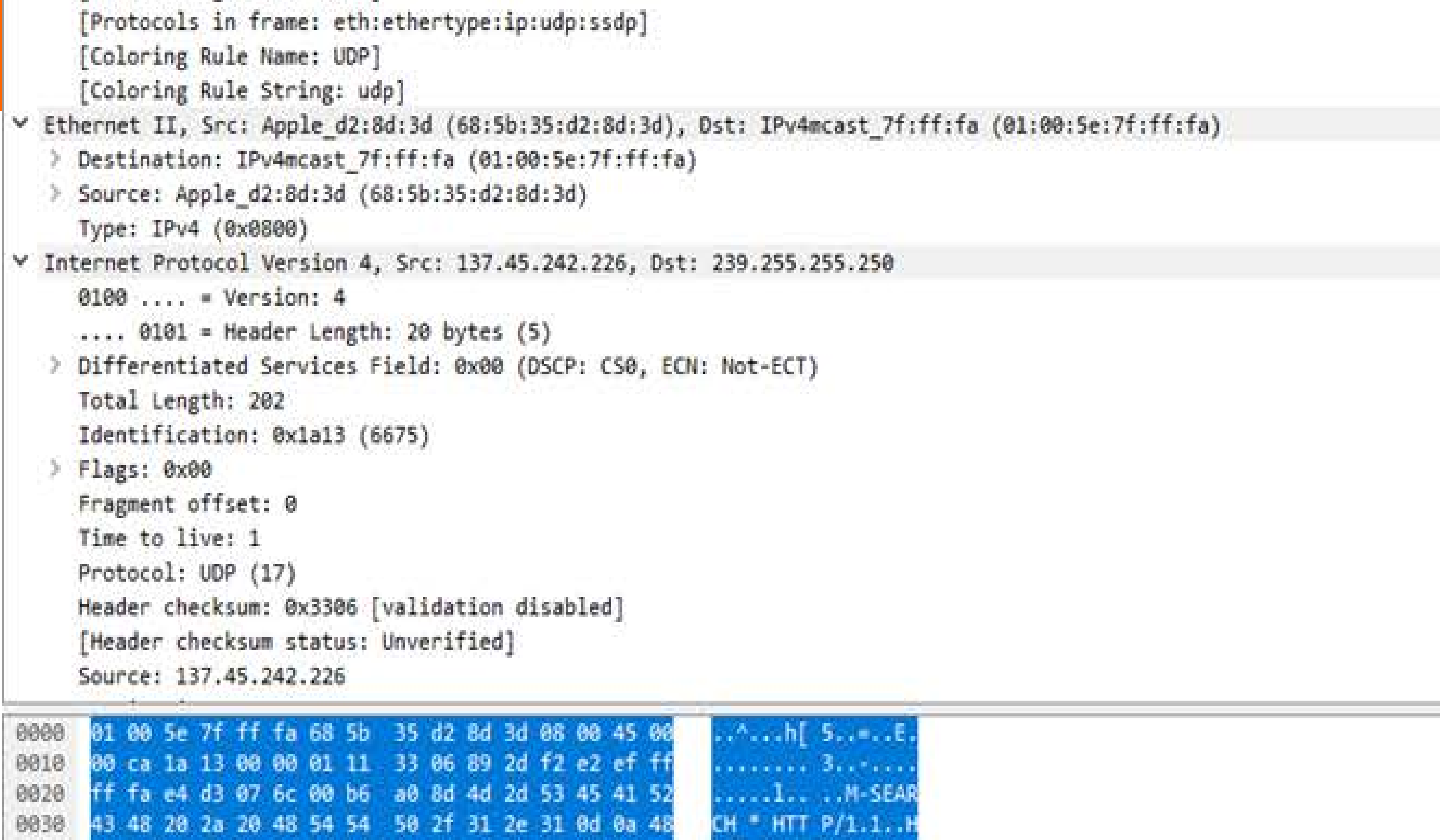
```
No.      Time       Source                   Destination       Protocol  Length  Info
      28 4.838847   137.45.243.206           54.81.107.149     TCP           55 4272 → 80 [ACK] Seq=1 Ack=1 Win=258 Len=1
      29 4.849557   54.81.107.149            137.45.243.206    TCP           66 80 → 4272 [ACK] Seq=1 Ack=2 Win=106 Len=0
      30 4.939111   137.45.242.226           239.255.255.250   SSDP         216 M-SEARCH * HTTP/1.1
      31 4.994146   137.45.243.206           54.81.107.149     TCP           55 4273 → 80 [ACK] Seq=1 Ack=1 Win=254 Len=1
      32 5.005946   54.81.107.149            137.45.243.206    TCP           66 80 → 4273 [ACK] Seq=1 Ack=2 Win=121 Len=0
      33 5.257249   fe80::489c:32f4:791…     ff02::1:2         DHCPv6       170 Solicit XID: 0x6fb14c CID: 000100011f3e8bb
      34 5.470518   fe80::f17b:6fab:3d1…     ff02::1:2         DHCPv6       170 Solicit XID: 0xb98437 CID: 000100011f3ebba
```

```
˅ Frame 30: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits) on interface 0
     Interface id: 0 (\Device\NPF_{EE7B717E-41DF-48C3-8C85-509234390636})
     Encapsulation type: Ethernet (1)
     Arrival Time: Jul 13, 2017 11:03:36.648554000 Eastern Daylight Time
     [Time shift for this packet: 0.000000000 seconds]
     Epoch Time: 1499958216.648554000 seconds
```



```
     32 5.005946    54.81.107.149         137.45.243.206      TCP       66 80 → 4273 [ACK] Seq=1 Ack=2 Win=121 Len
     33 5.257249    fe80::489c:32f4:791… ff02::1:2           DHCPv6   170 Solicit XID: 0x6fb14c CID: 000100011f3e
     34 5.470518    fe80::f17b:6fab:3d1… ff02::1:2           DHCPv6   170 Solicit XID: 0xb98437 CID: 000100011f3e
```

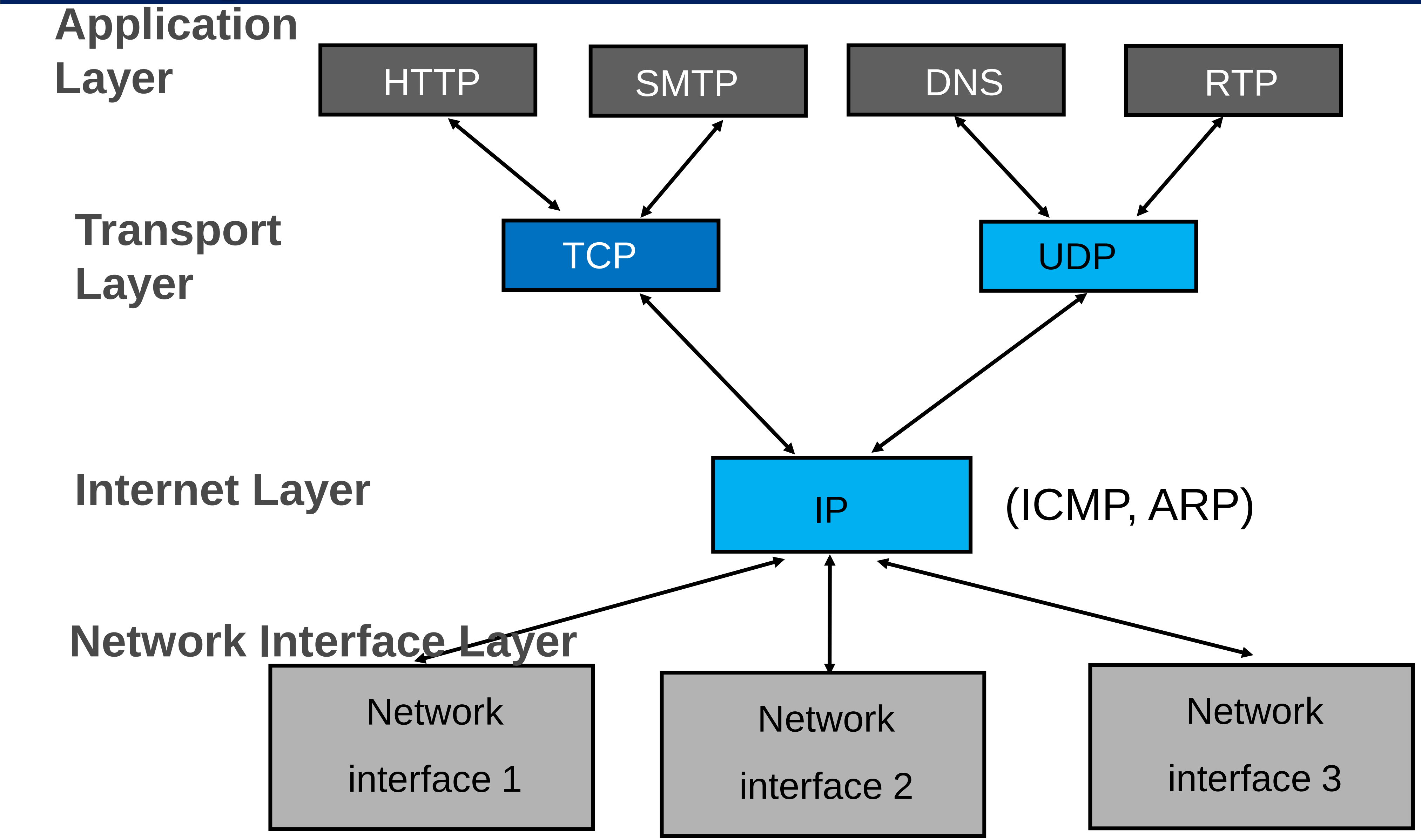
```
∨ Frame 30: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits) on interface 0
     Interface id: 0 (\Device\NPF_{EE78717E-41DF-48C3-8C85-509234390636})
     Encapsulation type: Ethernet (1)
     Arrival Time: Jul 13, 2017 11:03:36.648554000 Eastern Daylight Time
     [Time shift for this packet: 0.000000000 seconds]
     Epoch Time: 1499958216.648554000 seconds
     [Time delta from previous captured frame: 0.089554000 seconds]
     [Time delta from previous displayed frame: 0.089554000 seconds]
     [Time since reference or first frame: 4.939111000 seconds]
     Frame Number: 30
     Frame Length: 216 bytes (1728 bits)
     Capture Length: 216 bytes (1728 bits)
     [Frame is marked: False]
     [Frame is ignored: False]
     [Protocols in frame: eth:ethertype:ip:udp:ssdp]
     [Coloring Rule Name: UDP]
     [Coloring Rule String: udp]
∨ Ethernet II, Src: Apple_d2:8d:3d (68:5b:35:d2:8d:3d), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
   > Destination: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
   > Source: Apple_d2:8d:3d (68:5b:35:d2:8d:3d)
     Type: IPv4 (0x0800)
```



```
    [Protocols in frame: eth:ethertype:ip:udp:ssdp]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
v Ethernet II, Src: Apple_d2:8d:3d (68:5b:35:d2:8d:3d), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  > Destination: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  > Source: Apple_d2:8d:3d (68:5b:35:d2:8d:3d)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 137.45.242.226, Dst: 239.255.255.250
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 202
    Identification: 0x1a13 (6675)
  > Flags: 0x00
    Fragment offset: 0
    Time to live: 1
    Protocol: UDP (17)
    Header checksum: 0x3306 [validation disabled]
    [Header checksum status: Unverified]
    Source: 137.45.242.226

0000  01 00 5e 7f ff fa 68 5b  35 d2 8d 3d 08 00 45 00   ..^...h[ 5..=..E.
0010  00 ca 1a 13 00 00 01 11  33 06 89 2d f2 e2 ef ff   ........ 3..-....
0020  ff fa e4 d3 07 6c 00 b6  a0 8d 4d 2d 53 45 41 52   .....l.. ..M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1..H
```

Recap: TCP/IP Protocol Suite



Diverse network technologies

Summary

- Encapsulation is key to layering
- IP provides for transfer of packets across diverse networks
- TCP and UDP provide universal communications services across the Internet
- Distributed applications that use TCP and UDP can operate over the entire Internet
- Internet names, IP addresses, port numbers, sockets, connections, physical addresses