Network Design: Layered Architectures

Network Design

- Network must be designed so that it has the flexibility of supporting current and future services.
- To achieve this flexibility, an overall network architecture (plan) is necessary
- The overall process of enabling machines to communicate is very complex
- ***Layering*** partitions related communication functions into manageable groups
- Each layer provides a ***service*** to the layer above
- Each layer operates according to a ***protocol***
- A *protocol* is a set of rules that governs how two or more communicating entities in a layer are to interact. That is,
  - Protocols define format, order of messages sent and received among network entities, and actions taken on message transmission and receipt
- Protocols: building blocks of a network architecture

Network Design…

- Protocols: building blocks of a network architecture

- A layered architecture, therefore, can also be viewed as an architecture for ‘Protocol Layering’.

- That is, in the process of designing network protocols, an important issue is how we organize protocols in layers.

Why Layering?

- Layering simplifies design, implementation, and testing by partitioning overall communications process into parts

- Layering aids modular design
  - Protocol in each layer can be designed separately from those in other layers
  - 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

An Example of Layering

- WWW application

- A framework for accessing documents located in computers in the Internet
  - Documents (text, graphics, and other media) are interconnected by links which appear within the documents

- WWW is accessed through a browser program

- Based on ***client-server*** interaction
  - Server *process* waits for incoming requests by listening to a ***port***
  - Client processes make ***requests***
  - Servers provide ***responds***

- Protocol: HTTP (HyperText Transfer Protocol)
  - Specifies rules by which the client and server interact to retrieve a document

An Example of Layering…

1. User selects a document by clicking on a link with a URL: http://www.cnn.com/world/index.html
2. Client locates the server host and establishes a two-way TCP connection
3. Client uses HTTP to send a message requesting a document

   GET/world/index.html HTTP/1.0
4. Server listening on TCP port 80 interprets the message
5. Server sends a results code and a description

   HTTP/1.1 200 OK

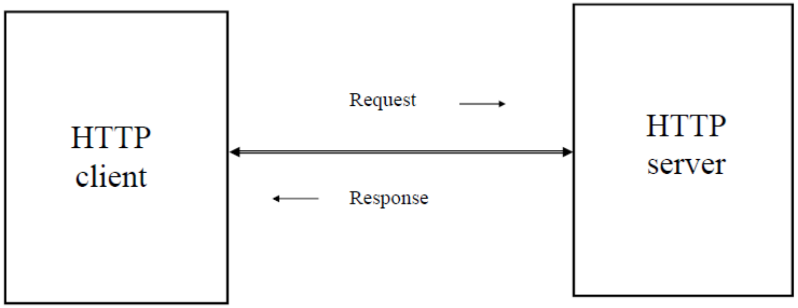
   Server: Apache/1.2.5

   Content-Length: 500

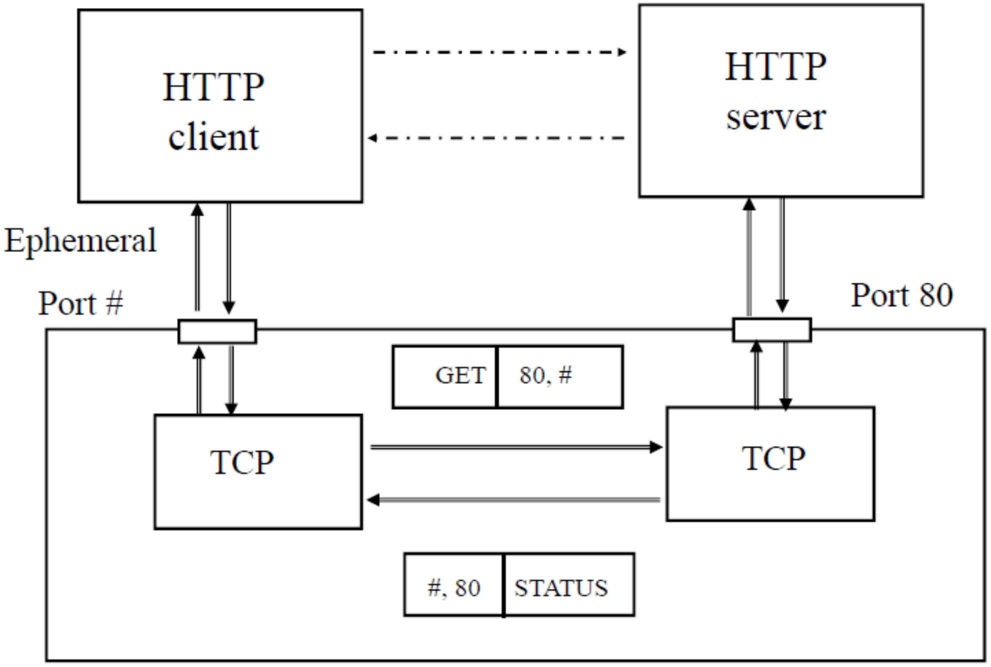
   Content-Type: text/html
6. Server reads the file and sends it through the TCP port

   <html><head><Title> CNN …
7. Server disconnects the connection
8. Client receives the file and displays it (interprets the HTML code and displays the information)

- HTTP assumes that message exchange between peer processes occurs directly



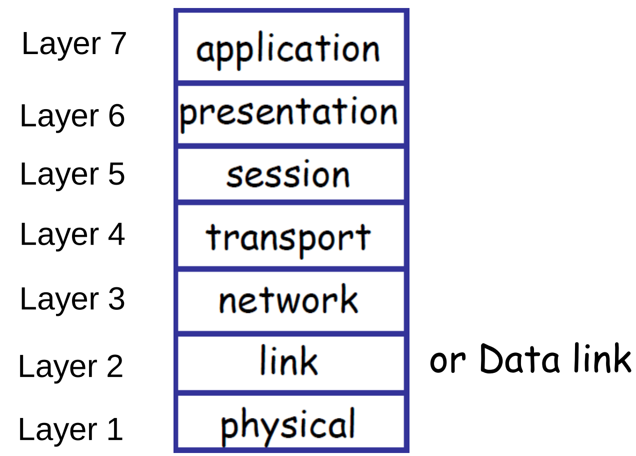
- Transfer of messages between HTTP client and HTTP server is in fact virtual and occurs indirectly via the TCP connection



- HTTP uses the services provided by TCP in an underlying layer
- TCP, in turn, uses the services provided by IP

Network Design: Layered Architectures

- OSI reference model (7 Layers)



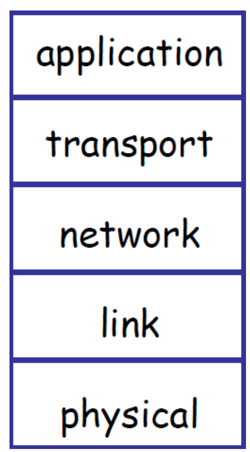
- Internet protocol stack or TCP/IP architecture or Internet Architecture

(There are two representations for this architecture: 5 layers or 4 layers)

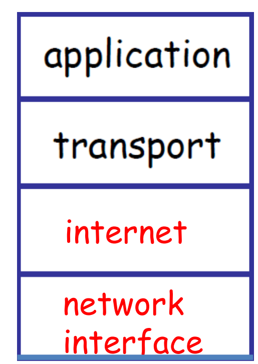
Protocol stack: the protocols of the various layers

TCP: Transmission Control Protocol
IP: Internet Protocol

The Internet = a TCP/IP network



(5 Layers)

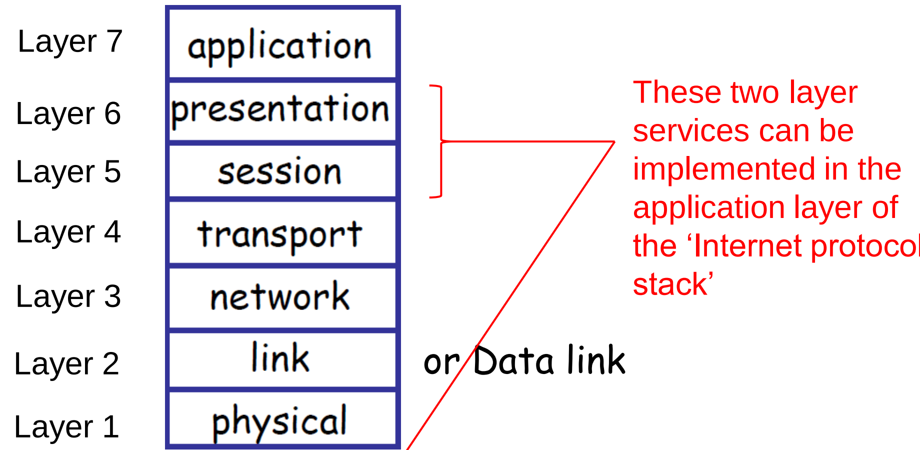


(4 Layers)

Network Design: Layered Architectures

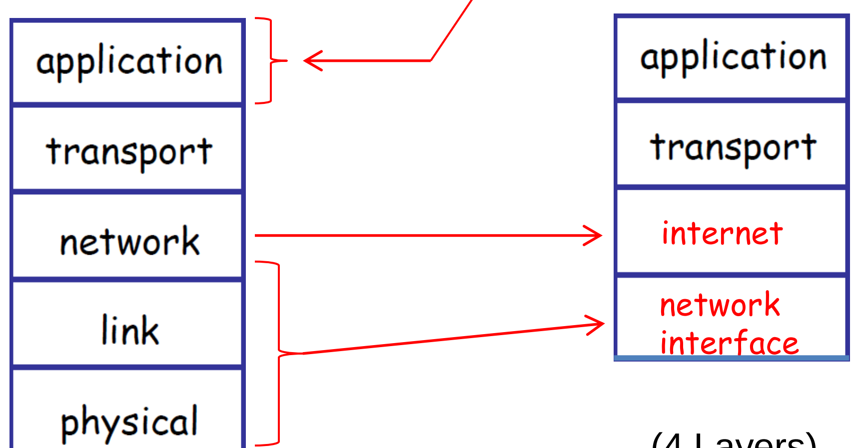
Are they totally different?

- OSI reference model (7 Layers)



- Internet protocol stack or TCP/IP architecture or Internet Architecture

(There are two representations for this architecture: 5 layers or 4 layers)



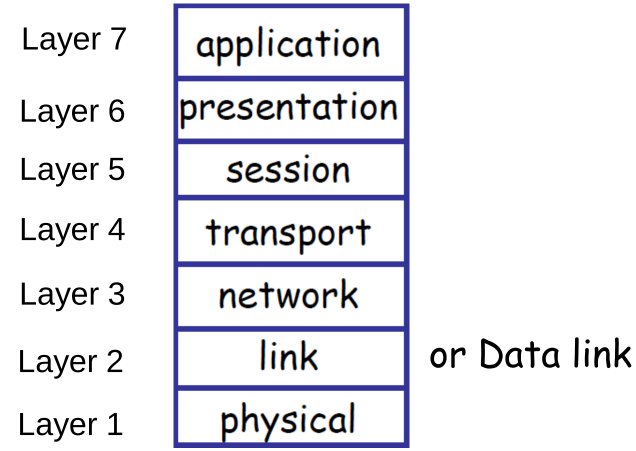
(4 Layers)

(5 Layers)

Network Design: Layered Architectures

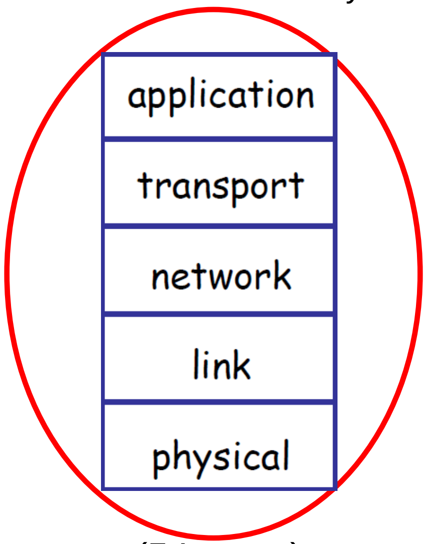
Which architecture we will be using mostly?

- OSI reference model (7 Layers)

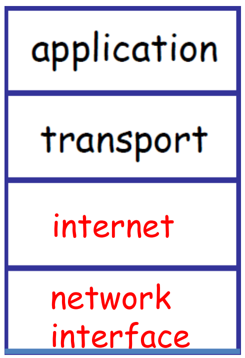


- Internet protocol stack or TCP/IP architecture or Internet Architecture

(There are two representations for this architecture: 5 layers or 4 layers)



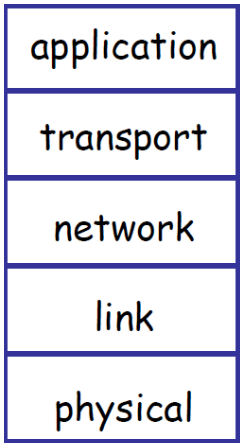
(5 Layers)



(4 Layers)

Network Design: Layered Architectures

Layers, protocols, and services



- Protocol Layering: how we organize protocols in layers.

- Each protocol belongs to one of the layers. E.g.,
  - HTTP (for web browsing): Application layer
  - TCP: Transport layer
  - IP: Network layer

- How a protocol is implemented?
  - Network hardware and software implement the protocol.

- An important issue is **the services a layer offers to the layer above** (service model of a layer). E.g., transport layer services offered to the application layer.

- Each layer provides its service (e.g., transport layer service) by,
  1) Performing certain actions within that layer (e.g., within the transport layer)

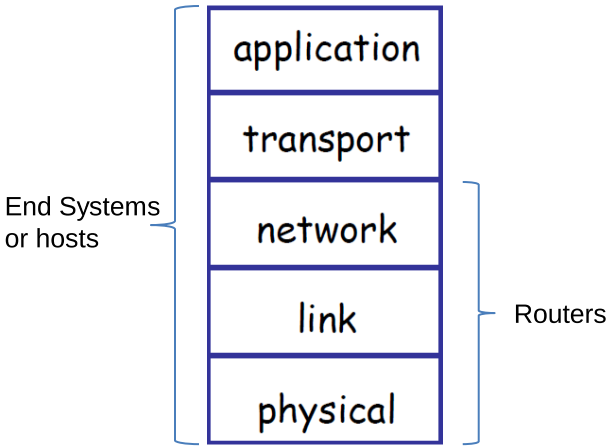
     \+

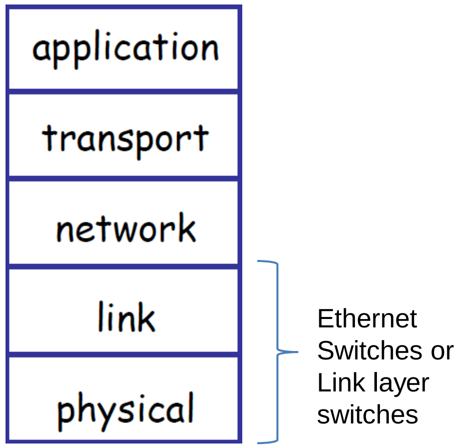
  2) Using the services of the layer below it (e.g., using network layer services)

- E.g., the services offered by the layer *n* (transport layer) may include **reliable delivery** (no errors or loss) of messages from one edge to the other edge. This might be implemented by
  - Using an unreliable edge-to-edge message delivery service of the layer *n*-1 (network layer), and
  - Adding a layer *n* functionality (at the transport layer) to detect and retransmit messages if there are errors (or lost).

Network Design: Layered Architectures

Where each of the protocol layers implemented in the Internet?



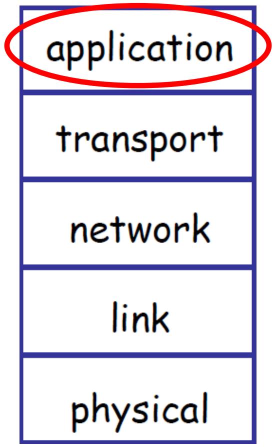


- How a protocol is implemented?
  - Network hardware and software implement the protocol.
- That is, a protocol layer can be implemented in software, in hardware, or in a combination of the two. Where implemented?
  - Application layer protocols (e.g., HTTP) and transport layer protocols (e.g., TCP) are almost always implemented in software at the end systems (e.g., PCs).
  - Network layer is often a mixed implementation of software and hardware at the routers, and other devices. It is typically implemented in software at the end systems (PCs).
  - Data link layer and physical layer are typically implemented in network interface cards (e.g., Ethernet network interface cards (NICs) in PCs, in Ethernet switches, and in routers. WiFi interface cards).

    (TCP/IP protocols are typically implemented as part of the operating system kernel in the end systems such as PCs)
- Layer *n* protocol is distributed among the end systems, packet switches, other components that make up the Internet.
  - That is, there is often a piece of the layer *n* protocol in each of these network components.
- Too many layers in the end systems?

Internet protocol stack

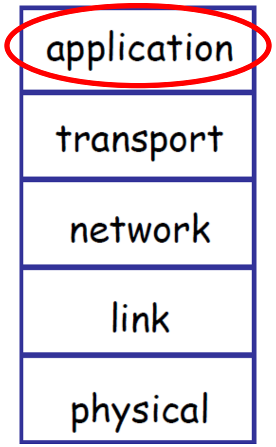
Application Layer



- Networks are driven by applications such as WWW, file transfer, email, etc.
- Communication for a network application takes place between ***end systems (hosts)*** at the application layer
- Application Layer: provides protocols to support different applications such as HTTP, FTP, SMTP, etc.
  - Web access: Browser application uses the HTTP application layer protocol to access a WWW document
  - FTP (file transfer), SMTP (Simple Mail Transfer Protocol for email)
- An application layer protocol is distributed over multiple end systems
- Using the protocol, the application in one end system exchanges ***messages*** with another end system
- Some of the functionalities provided by lower layers can also be implemented at the Application layer. E.g. error control

Internet protocol stack

Application Layer…



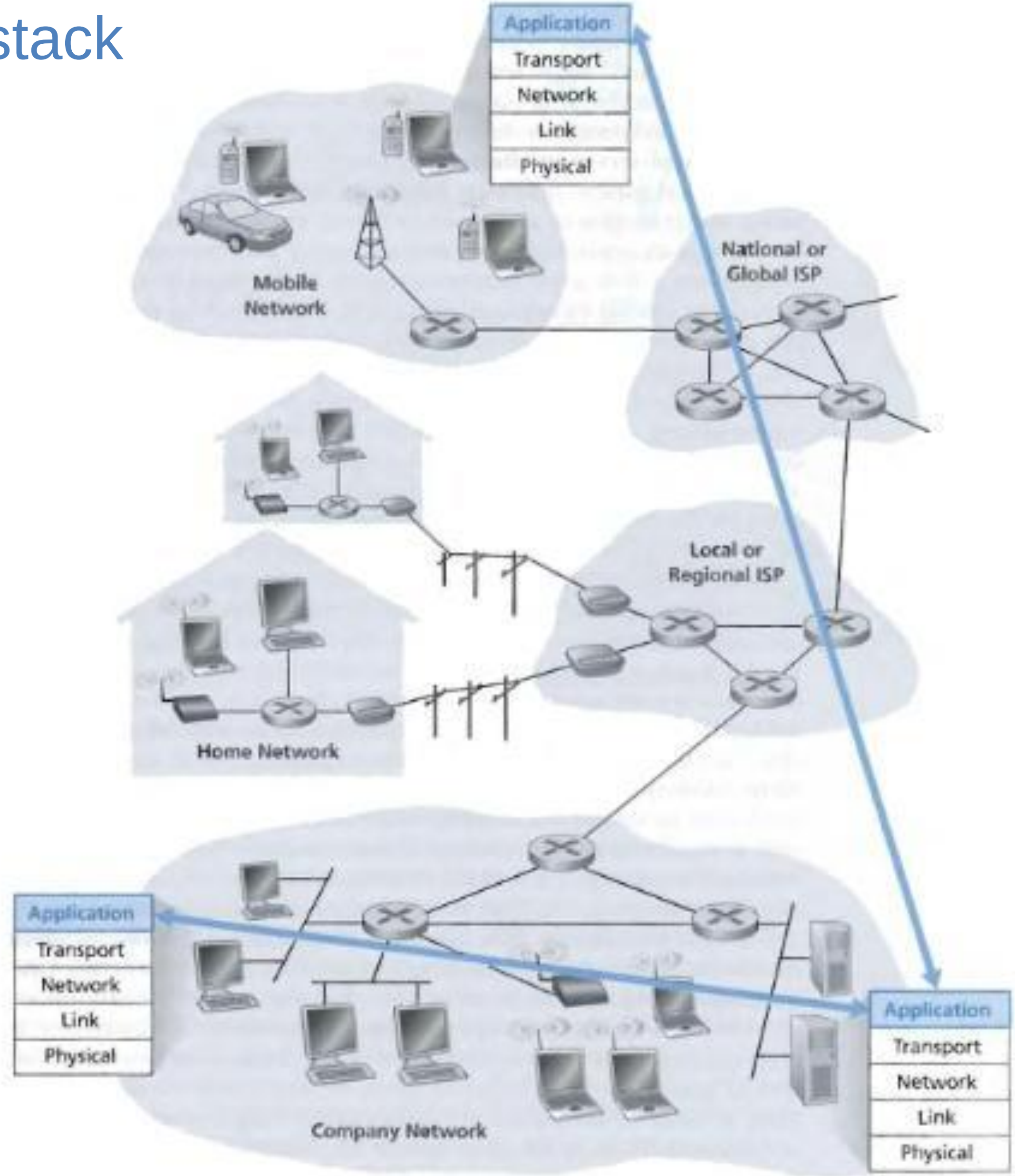
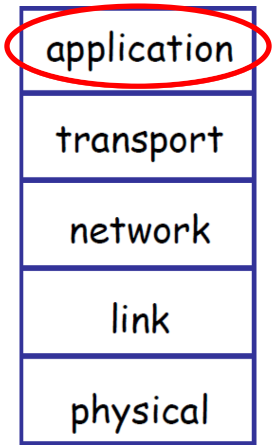


Figure 2.1 • Communication for a network application takes place between end systems at the application layer.

Kurose & Ross, Computer Networking A Top Down Approach Featuring the Internet

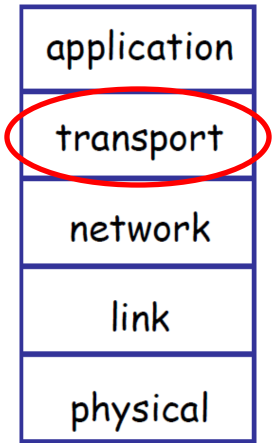
Internet protocol stack

Application Layer: processes communicating



- A network application consists of pairs of **processes** that send messages to each other over the network.
  - A process can be thought of as a program which is running within an end system
- E.g.,
  - Web application: a client browser process exchanges messages with the Web server process.
  - P2P file sharing: a file is transferred from a process in one peer to a process in another peer.
- There may be many processes running in an end system such as a PC (Web browser process, email client process, P2P process, etc.), a server (Web server process, email server process).
- How do we uniquely identify a process associated with a network application such as web browser process in an end system?
  - An identifier that specifies a process in an end system = **port number**
  - E.g., Web server is identified by port number 80.
- In the Internet, there are many end systems/hosts. How do we identify a process on a particular host?
  - In addition to the port numbers, we need a name or address of a host/end system
  - We need: **IP address** + port numbers
- How messages are transmitted between processes?
  - The application layer uses the transport layer's services for this!

Internet protocol stack
Transport Layer



- Transports application data end-to-end from a process in a host to a process in another host
  - Recall that a host/machine may have many processes!
- In the Internet, there are two transport layer protocols:
  - TCP: Transmission Control Protocol
  - UDP: User Datagram Protocol
- It provides two types of services to its applications: connection oriented service (TCP) and connectionless service (UDP)
  - Services provided by TCP include:
    - Reliable stream transfer (error control)
    - Flow control
    - Congestion control
  - UDP services:
    - Quick-and-simple single-block transfer
- Port numbers enable multiplexing. Transport layer uses the port number to passes the data to the respective process in a machine (many processes in a host!).
- Message segmentation and reassembly
- Data entities transmitted at the transport layer are referred to as **segments**.

Internet protocol stack

Transport Layer…

application
transport
network
link
physical

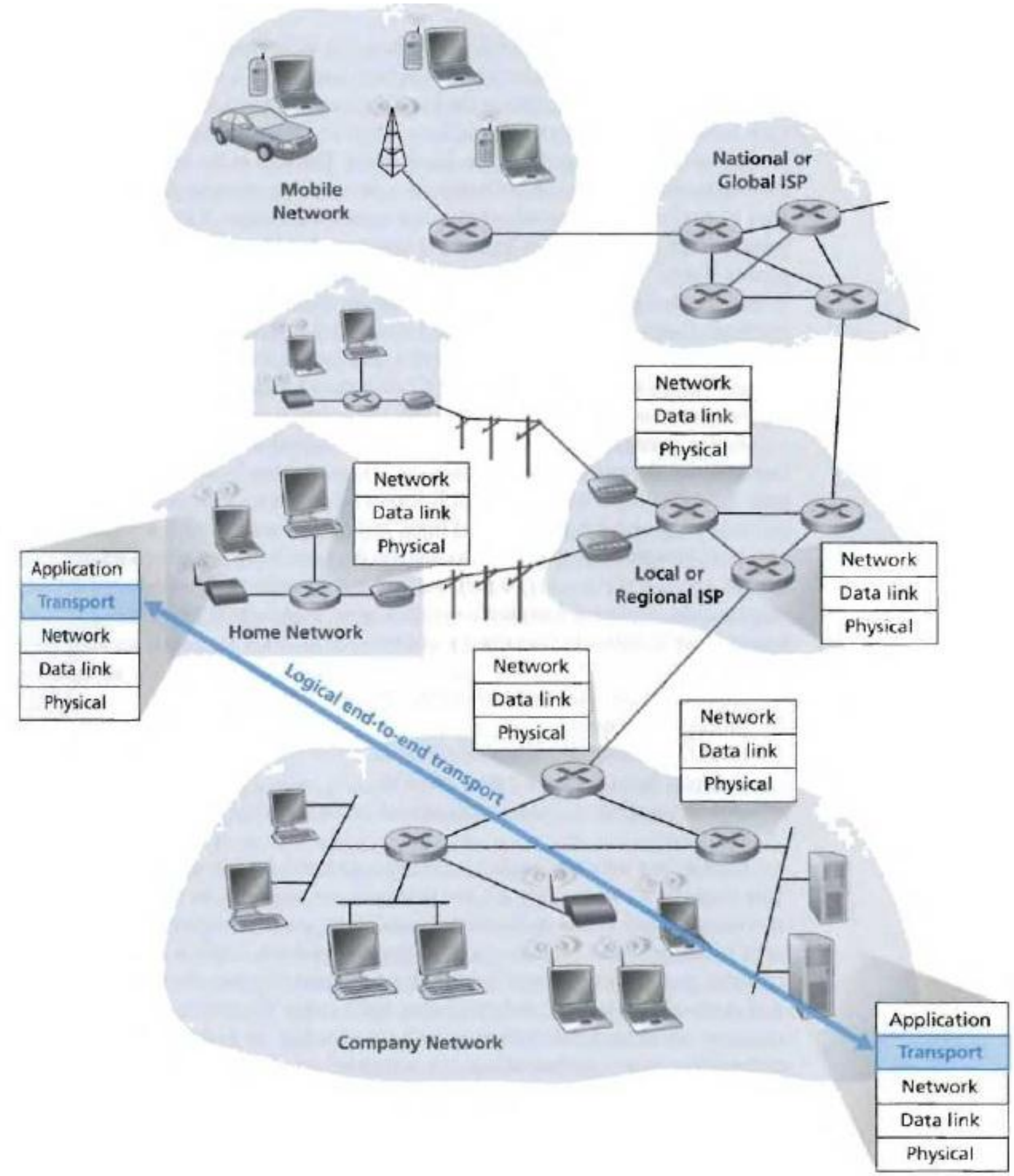
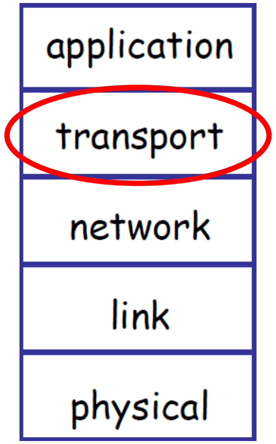


Figure 3.1 ♦ The transport layer provides logical rather than physical communication between application processes.

Internet protocol stack

Transport Layer…

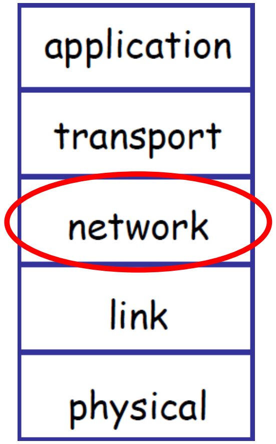


- Transports application data end-to-end from a process in a host to a process in another host

- When transporting segments between processes, **they must leave the source host and traverse through many intermediate devices and reach the destination host** before transmitted onto the respective process at the destination host.
  - How this host-to-host (or machine-to-machine) transmission is done?
- The transport layer uses the network layer's services for this!

Internet protocol stack

Network Layer

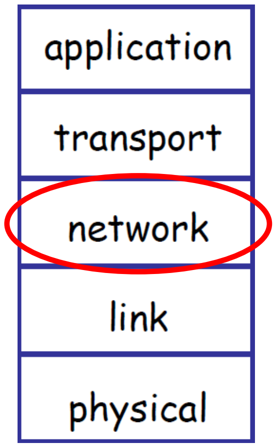


Note: The term ‘packets’ might, sometimes, be used to refer to any data chunks.

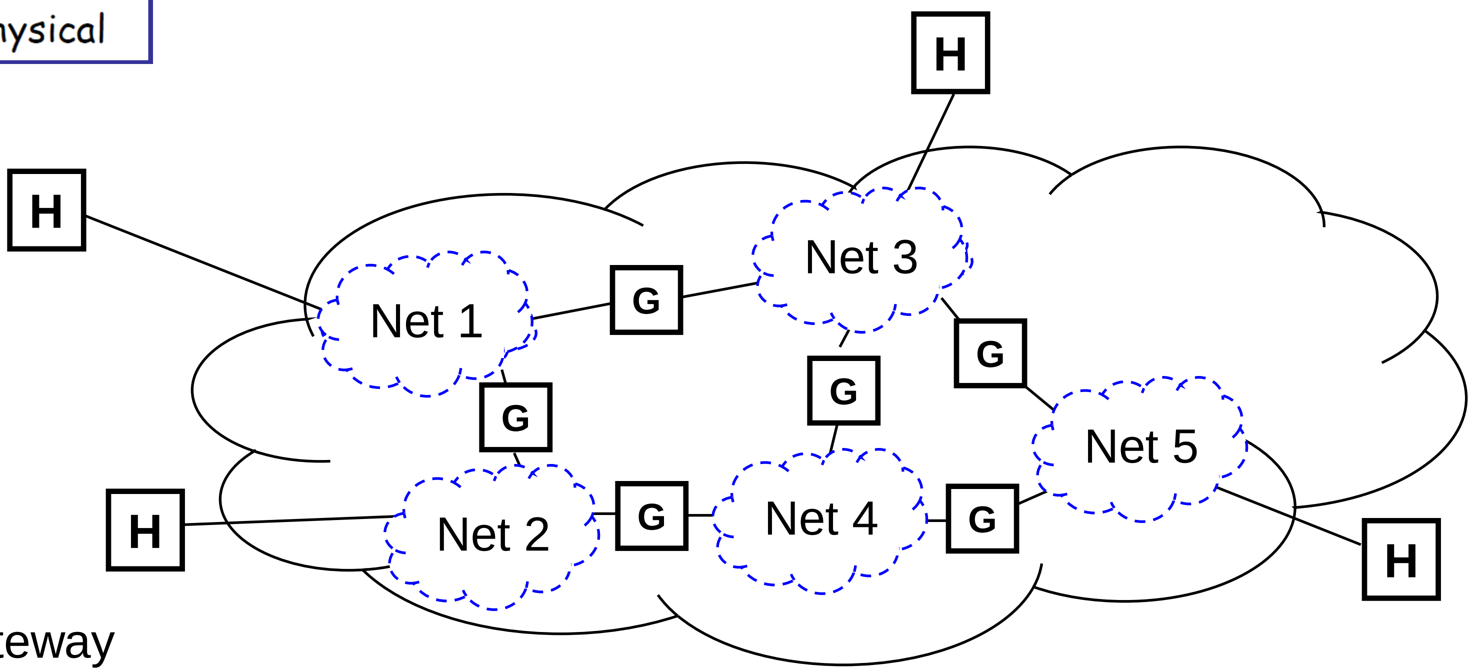
- When transferring application messages, the transport layer provides process-to-process communication service whereas the network layer provides host-to-host communication service
- Transfers ***packets or datagrams*** from one host to another (passing through several routers!).
- The transport layer protocol (TCP or UDP) in a source host passes a segment and a destination address (got from the user inputs) to the network layer. The network layer then provides the service of delivering the segment to the transport layer in the destination host.
- The Internet’s network layer provides ‘connectionless service’ which is ‘best-effort connectionless packet transfer’.
- The Internet’s network layer includes the key protocol, IP (Internet Protocol).
  – IP defines the fields in the packet/datagram and how the end systems and routers act on these fields.
  – There is only one IP protocol and all Internet components which have a network layer must run the IP protocol.
  – The Internet’s network layer also contains ‘routing protocols’ that determine the routes/paths that packets/datagrams take between sources and destinations.
  – Although the network layer contains both the IP protocol and many routing protocols, it is often simply referred to as the IP layer (since **IP is the one which glues different networks in the Internet together!**)

Internet protocol stack

Network Layer…



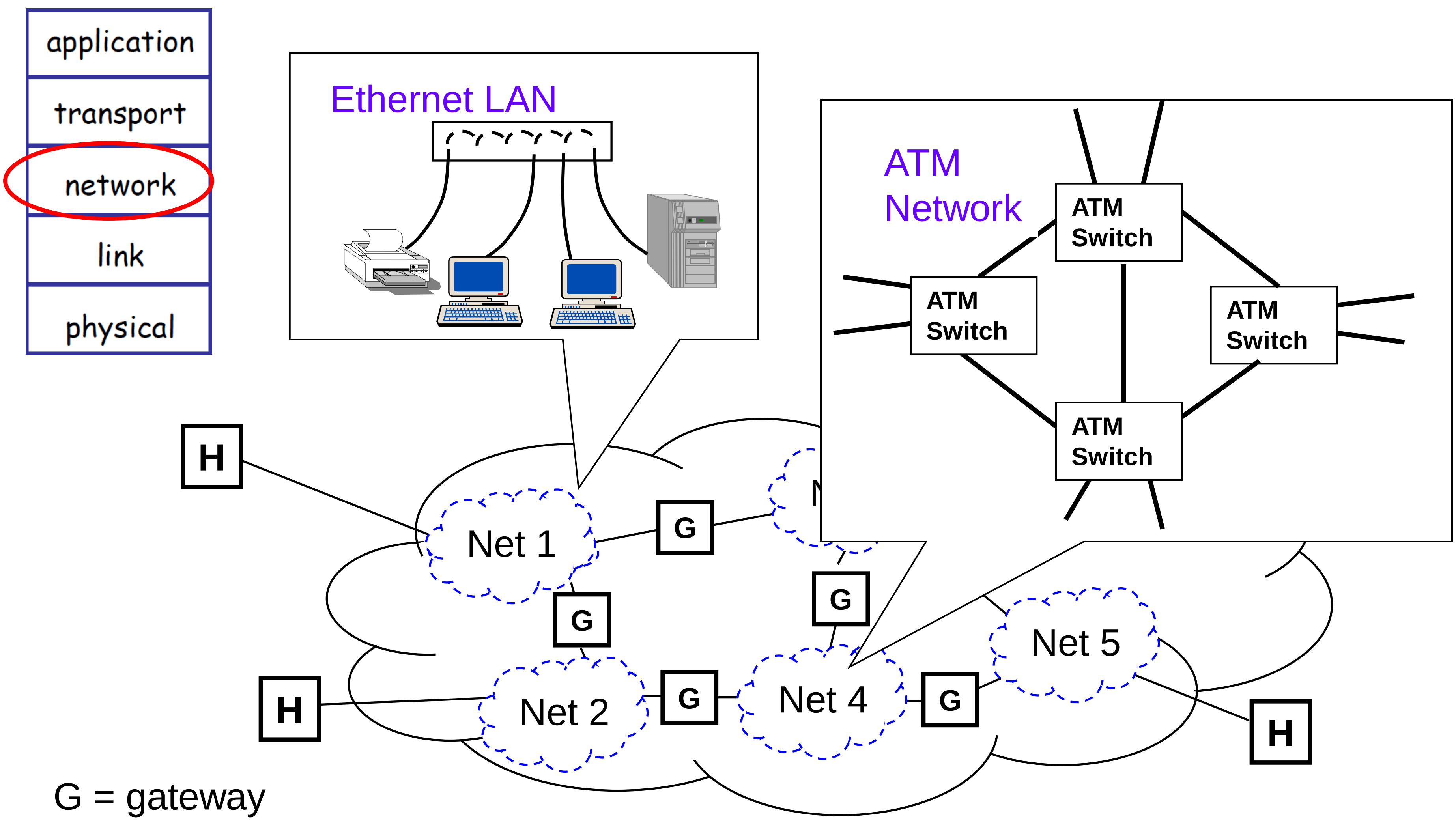
- Internetworking is necessary to route packets across multiple possibly dissimilar networks
- Gateways (routers) direct packets across networks



G = gateway
H = host

Internet protocol stack

Network Layer…



G = gateway
H = host

Internet protocol stack

Network Layer…

application

transport

network

link

physical

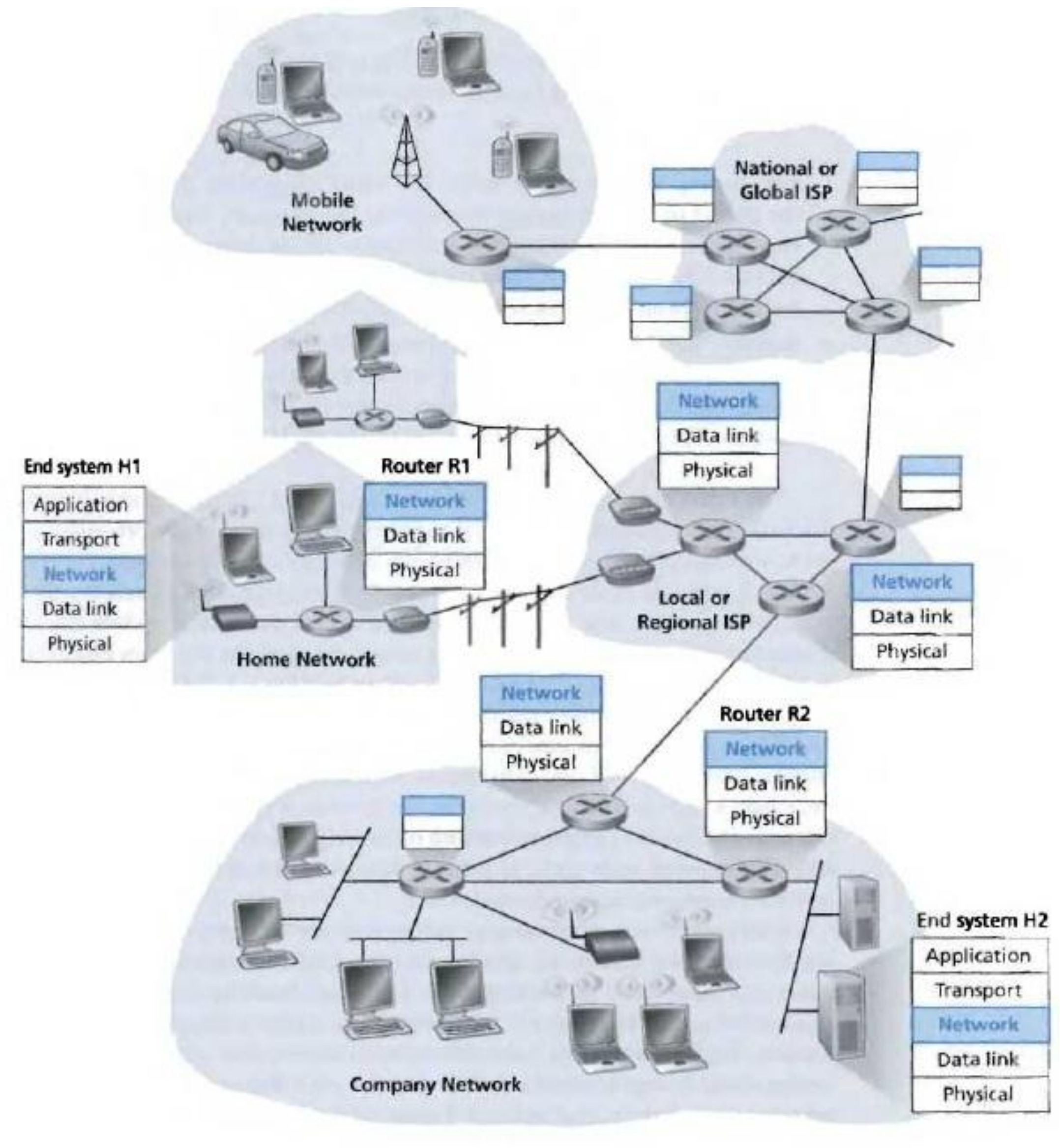
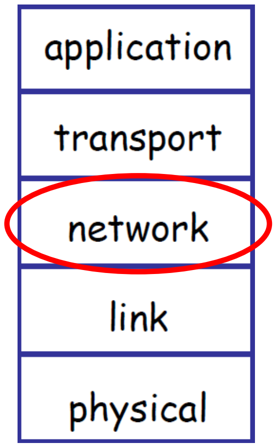


Figure 4.1 ♦ The network layer

Internet protocol stack

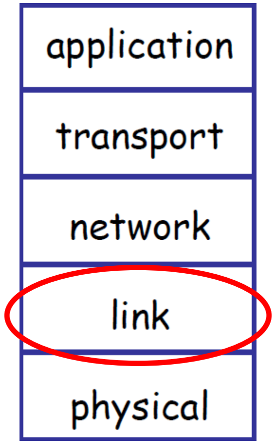
Network Layer…



- Key issues:
  - Addressing: network-layer addressing (IP address)
  - Routing and forwarding
  - Internetworking
  - Congestion control

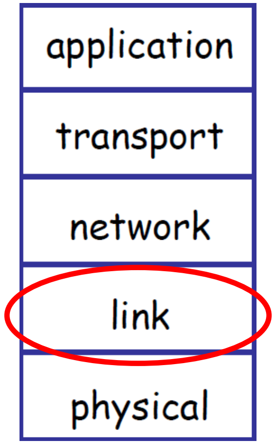
- The network layer provides host-to-host transmission through several intermediate routers. A host-to-host communication path consists many *links.*
  - Link: from one node to another node such as host-router (router-host) link, and router-router link.
  - Recall that the link which connects an end system/host to the first router, host-router (router-host) link, is typically an access network such as Ethernet (an access *link*).
  - How transmission over these *links* (note: these links include access networks/links) is done?
  - The network layer uses the link-layer's services for this!

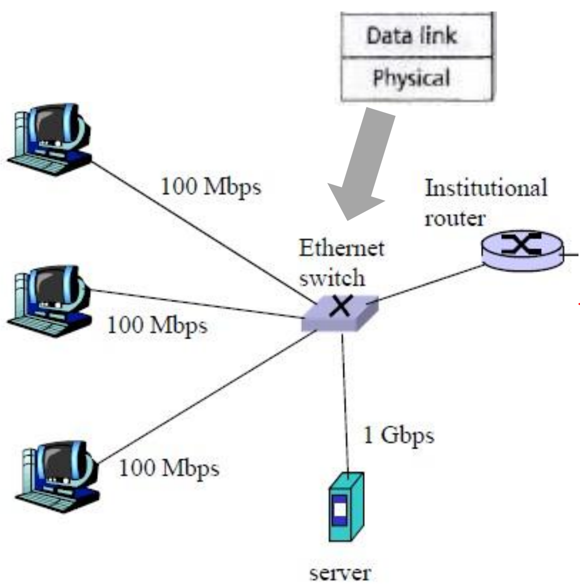
Internet protocol stack
Data Link Layer



- A host-to-host communication path consists many *links.* The Data Link layer deals with how packets are transmitted over each of the links, which connects two nodes.
  - The links include access networks (or LANs) such as Ethernet.
- At each node, the network layer passes the packet/datagram to the link layer which delivers the packet to the next node. At this next node, the link layer passes this packet/datagram to the network layer.
- Link layer transmission entities are called ***frames.***
- Link layer protocols: Ethernet, Wifi, PPP (point-to-point protocol), etc.
- As packets typically need to traverse several links along their paths, the packets may be handled by different link-layer protocols at different links along the route!
  - E.g., a packet may be handled by Ethernet on one link and by PPP on the next link.
  - The network layer will receive a different service from each of the different link-layer protocols.
- Can provide connectionless or connection oriented services
  - E.g. Ethernet : connectionless service (unacknowledged)

Internet protocol stack
Data Link Layer…





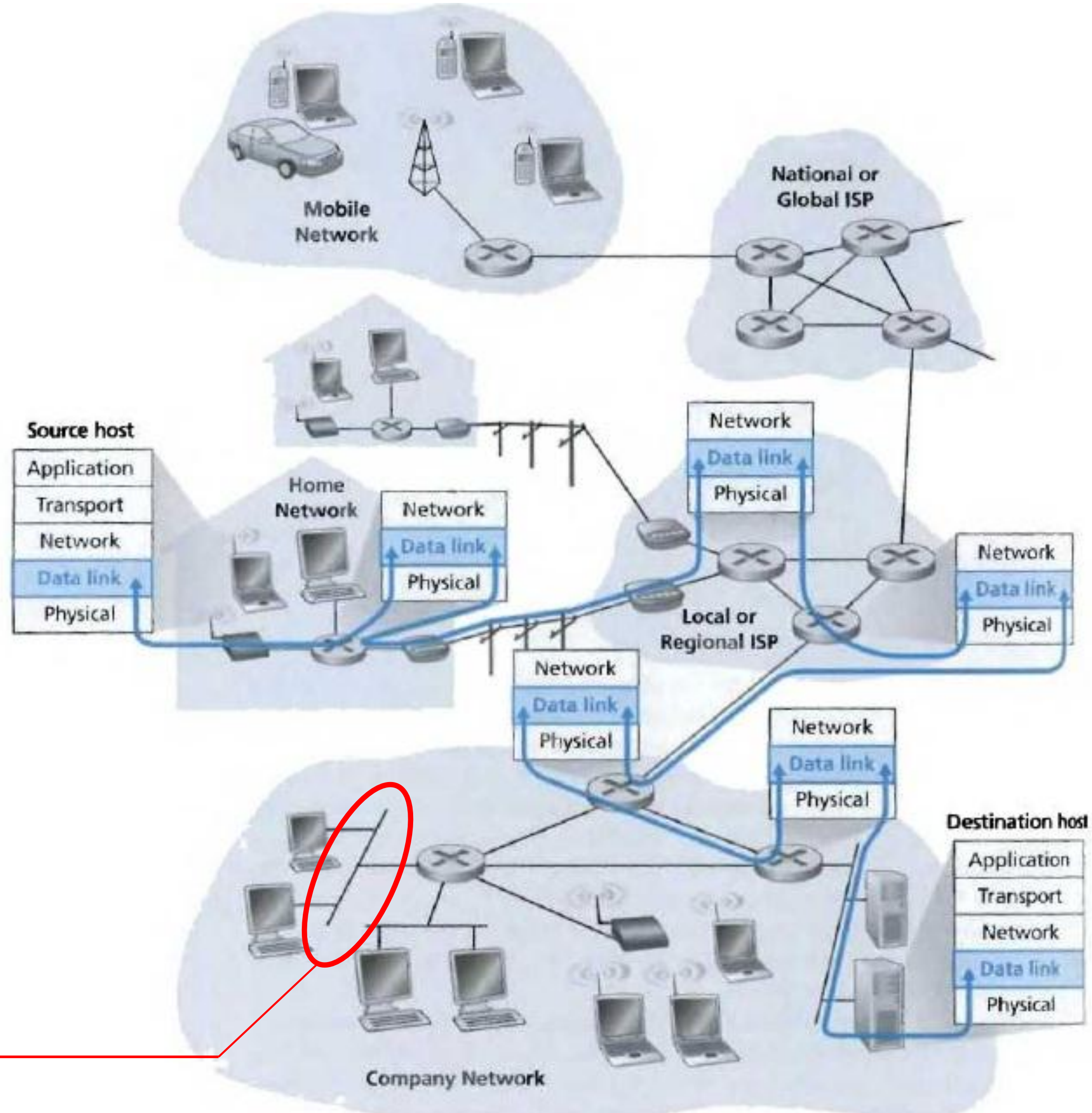
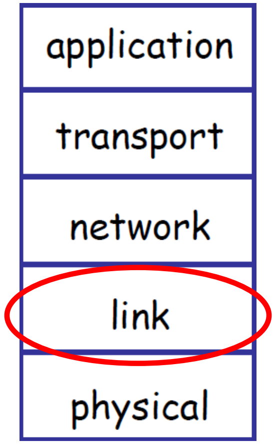


Figure 5.1 ♦ The link layer

Internet protocol stack

Data Link Layer…

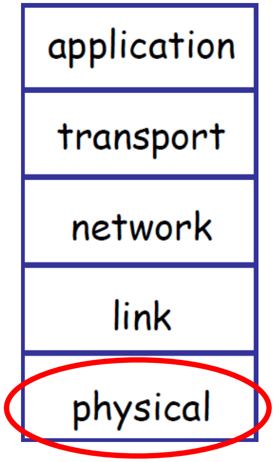


- Key Issues:
  - Framing. Why Framing?
  - Error control or reliable transmission, Flow control (Transport layer already has these services. Why do we need these again in the link-layer?).
  - Medium access control for local area networks
  - Activation, maintenance, & deactivation of data link connections
  - Half duplex and full duplex
    - Full duplex: nodes at both ends of a link may transmit at the same time (i.e., a node can both transmit and receive at the same time)
    - Half duplex: a node cannot both transmit and receive at the same time.
  - Addressing: Link-layer addressing. Why link-layer addresses when we already have network-layer addresses (IP addresses)?

- The link layer is dealing with transmission of frames on each link between two adjacent nodes. On each of those links, how the actual, physical transmission of each bit in the frame is done?
  - The link-layer uses the services of the physical layer for this!

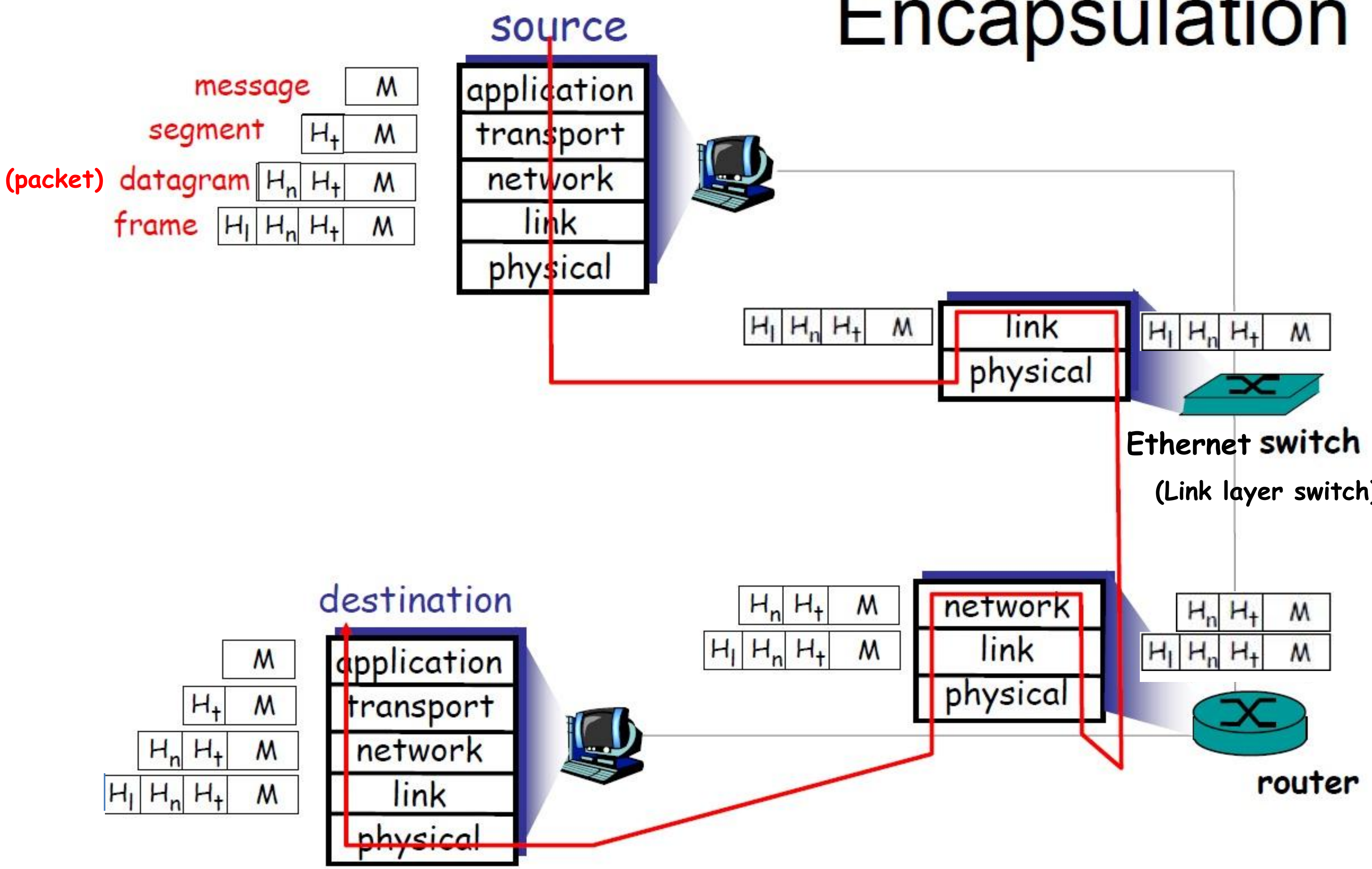
Internet protocol stack

Physical Layer



- Move individual bits in a frame across a link
- The protocols in this layer are link dependent and further depend on the actual transmission medium of the link (twisted-pair copper wire, single-mode fiber optics)
  - E.g., Ethernet has many physical layer protocols: one for twisted-pair copper wire, another for coaxial cable, another for fiber.
  - In each case, a bit is moved across the link in a different way.
- 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…
- Key issues:
  - Digital transmission fundamentals
  - Properties of media: copper, optical fiber, wireless media
  - Line coding
  - modulation

Encapsulation

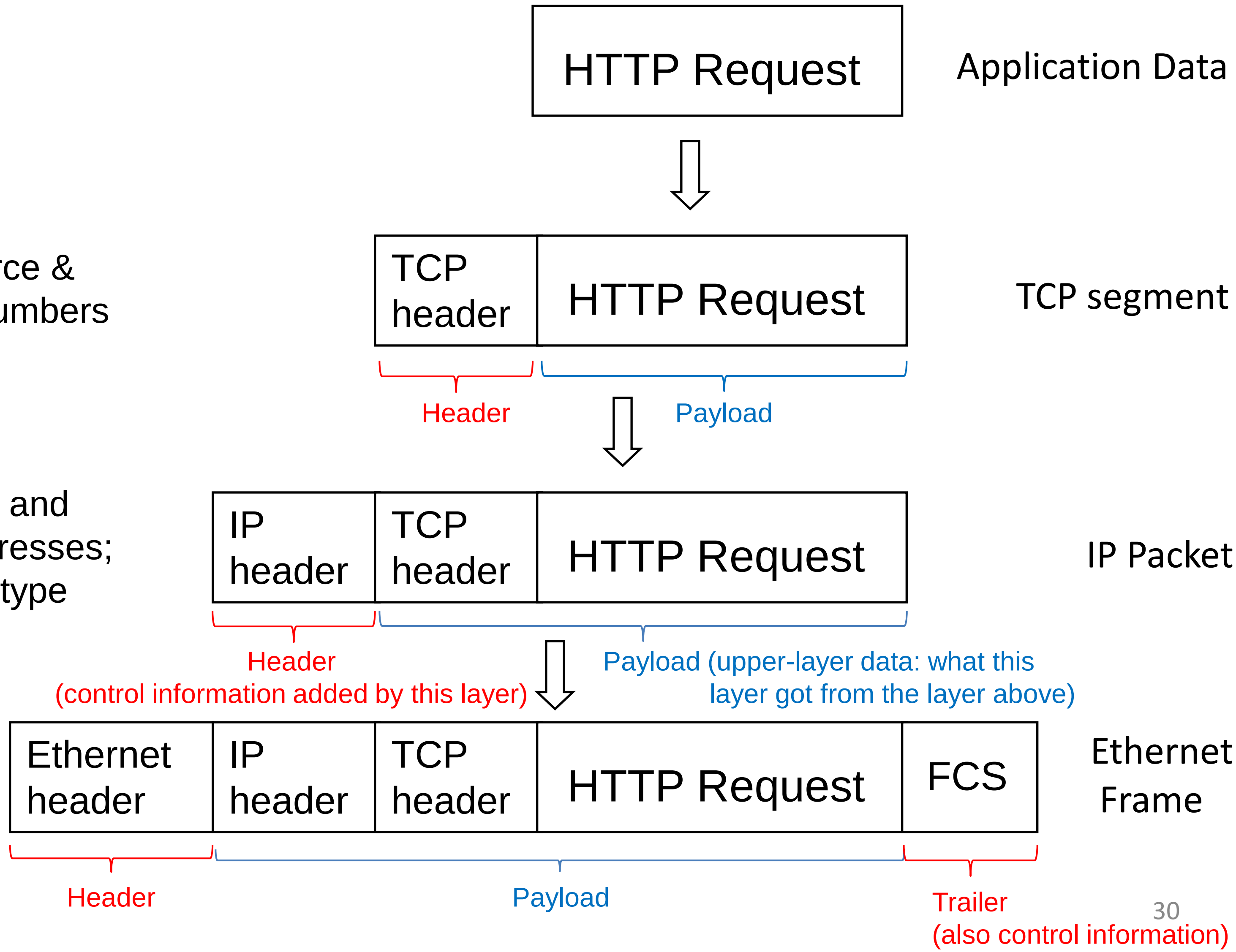


Encapsulation: HTTP request

TCP Header: source & destination port numbers

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

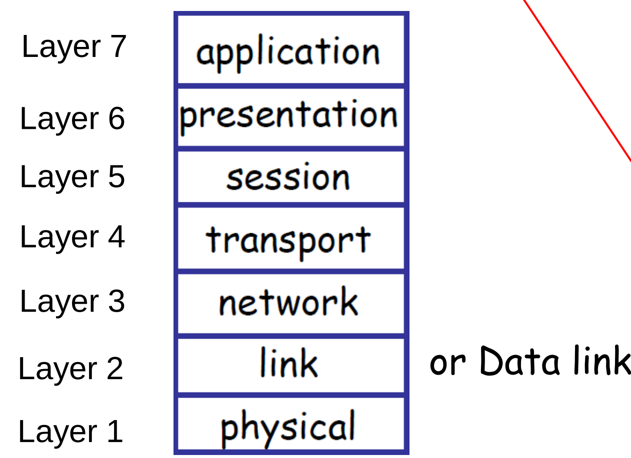
Ethernet Header: source & destination MAC addresses; network protocol type



Network Design: Layered Architectures

The Internet protocol stack or TCP/IP architecture with **4-layer representation**

- OSI reference model (7 Layers)



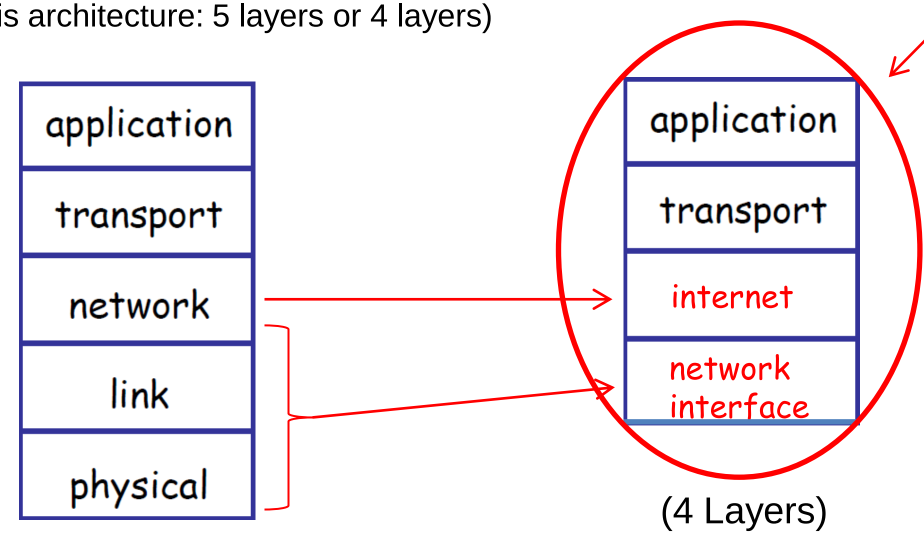
- Internet protocol stack or TCP/IP architecture or Internet Architecture

(There are two representations for this architecture: 5 layers or 4 layers)

Protocol stack: the protocols of the various layers

TCP: Transmission Control Protocol
IP: Internet Protocol

The Internet = a TCP/IP network

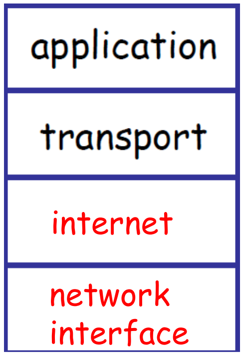


(5 Layers)

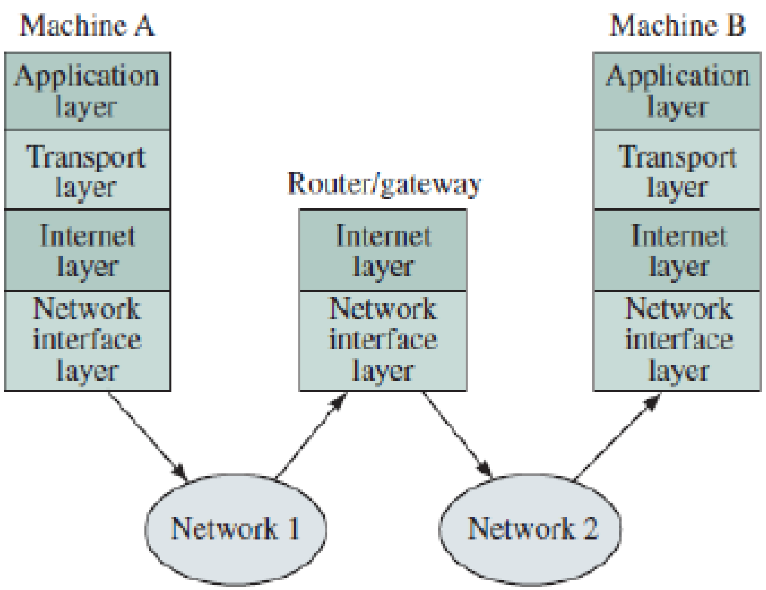
(4 Layers)

Network Design: Layered Architectures

The Internet protocol stack or TCP/IP architecture with **4-layer representation**….



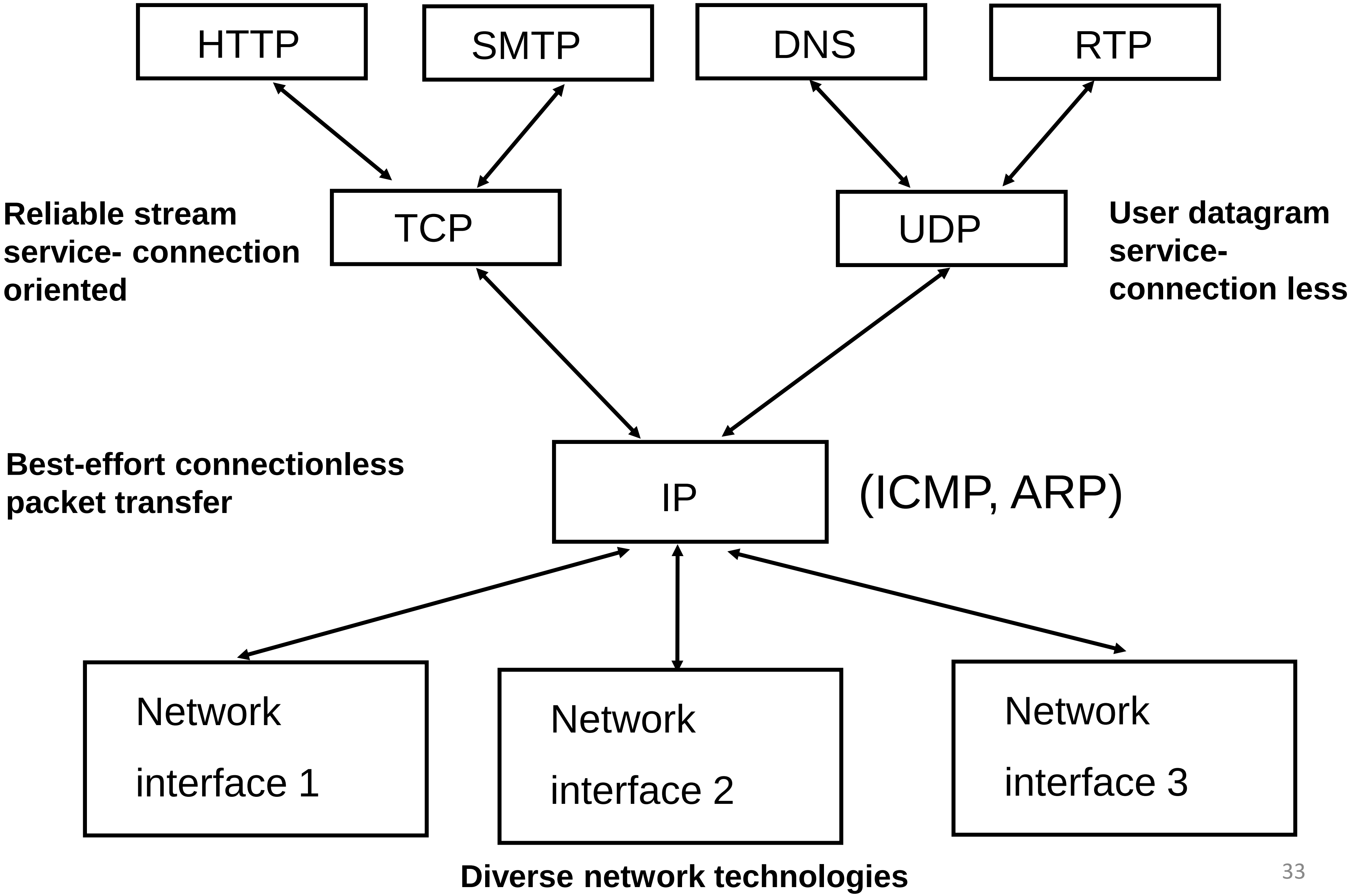
(4 Layers)



Layers marked in red (see the figure):

- Internet layer ~ network layer (in the 5 layer representation).
- Network Interface layer:
  - Related to network-specific aspects of the transfer of packets
  - Various network interfaces related to networks such as Ethernet, ATM, token ring, frame relay, X.25
  - Concerned with the protocols that access the intermediate networks
    - The protocol encapsulates an IP packet into a frame (or packet) of the underlying network or link (e.g., Network 1)
    - The IP packet is recovered at the exit gateway of the given network
    - The exit gateway encapsulates the IP packet into a frame (or packet) of the next network or link (e.g., Network 2)
- Provides a clear separation of the internet layer from the technology-dependent network interface layer
- Internet layer provides a packet transfer service that is transparent to the underlying networks (i.e., not depending on the details of the underlying networks)

TCP/IP Protocol Suite



TCP/IP Protocol Suite…

- IP protocol is complemented by additional protocols (ICMP, IGMP, ARP, RARP)
- Hourglass shape of the TCP/IP protocol graph
  - A single IP protocol over various networks
  - It provides independence from the underlying network technologies
- Services of TCP and UDP provide a network independent platform on which applications can be developed
- By allowing multiple network technologies to coexist, the Internet provides ubiquitous connectivity and achieves enormous economies of scale.

IP addresses and MAC addresses

IP addresses (Network layer address/Layer-3 address/Internet Addresses)

- Each host in the Internet has (at least) a globally unique IP address.
- Separate IP address for each network interface to a network.
- Routing decision is done based on destination IP address.
- IP address: *network-id* (unique) + *host-id*
- IP version 4 (IPv4): 32 bits, e.g., 128.100.10.13 (Dotted Decimal Notation)

  IP version 6 (IPv6): 128 bits

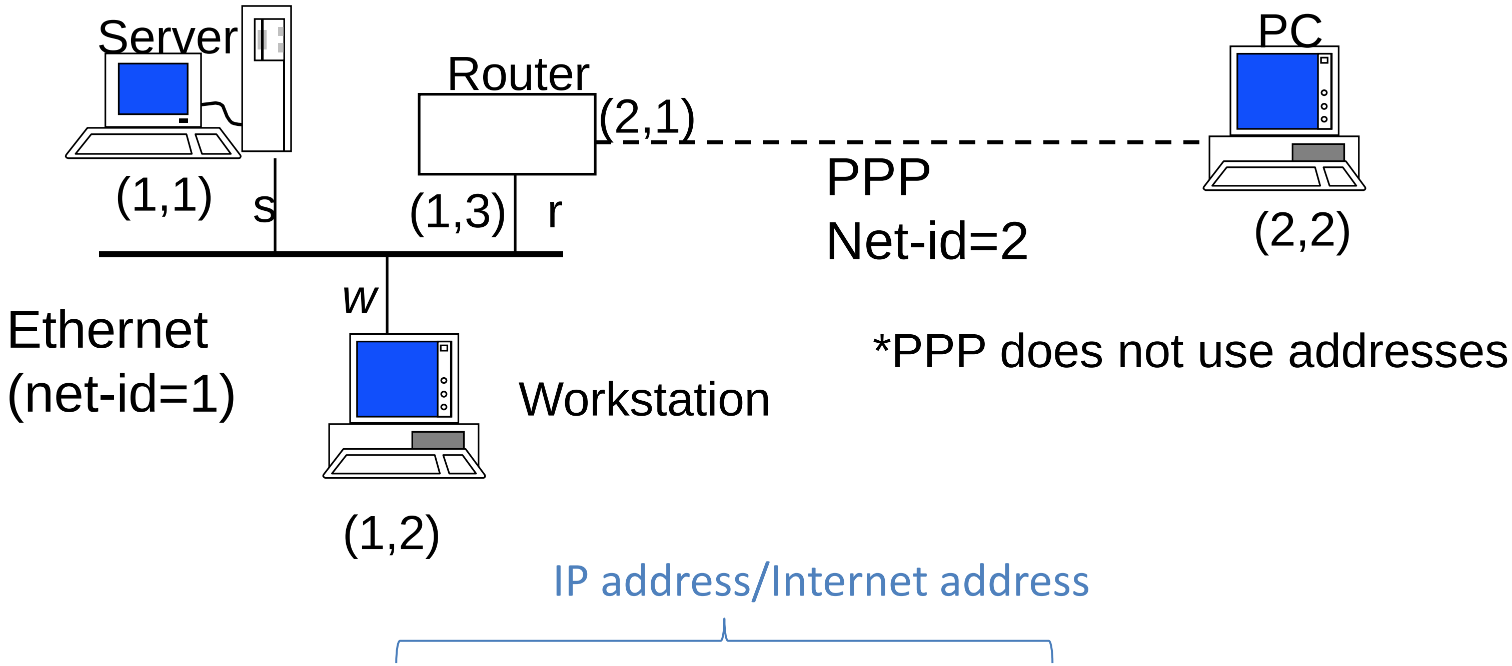
  e.g., 2001:0DB8:85A3:0000:0000:8A2E:0370:7334 (hexadecimal)

MAC addresses (Link layer address/Layer-2 address/Physical Addresses)

- On a physical network (e.g., Ethernet) attachment of a device to the network is identified by a physical address or MAC address.
- The physical network uses the physical address to transfer frames or packets 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
  - E.g., 00:90:27:96:68:07 (12 hex numbers)(24 bits NIC manufacturer+ 24 bits serial No.)

Why do we need MAC addresses when we already have IP addresses?

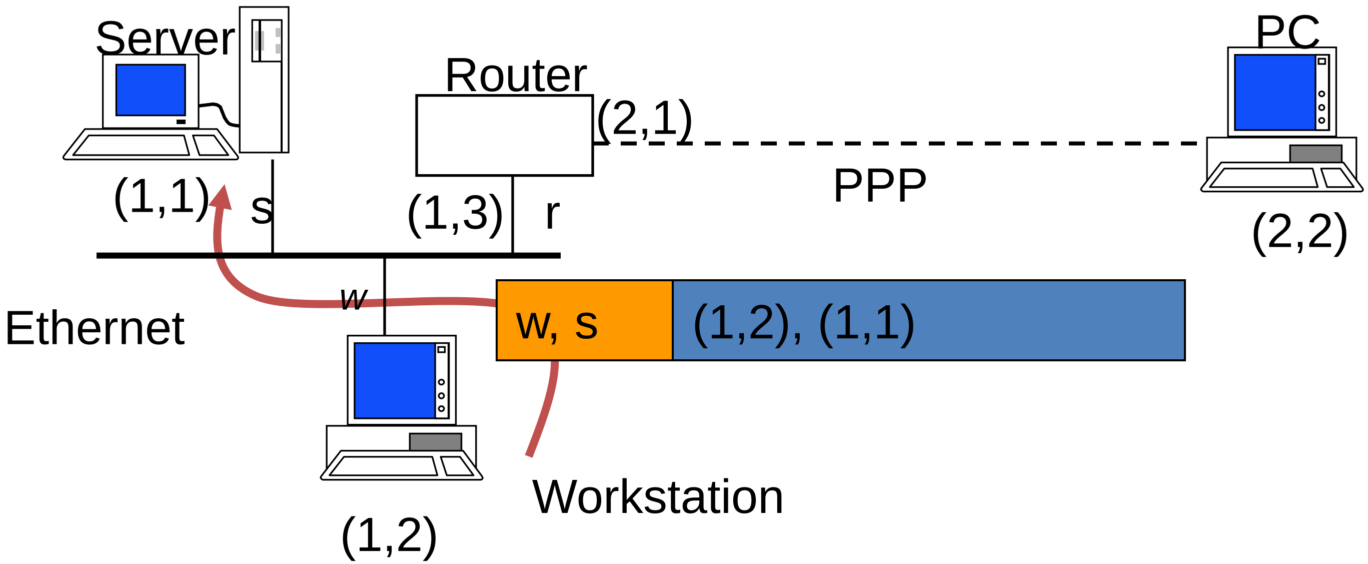
Example internet



	network-id	host-id	MAC/Physical address
server	1	1	s
workstation	1	2	w
router	1	3	r
router	2	1	-
PC	2	2	-

Example internet

IP packet from workstation to server



1. IP packet has the IP address of the source (1,2) and the IP address of the destination (1,1)

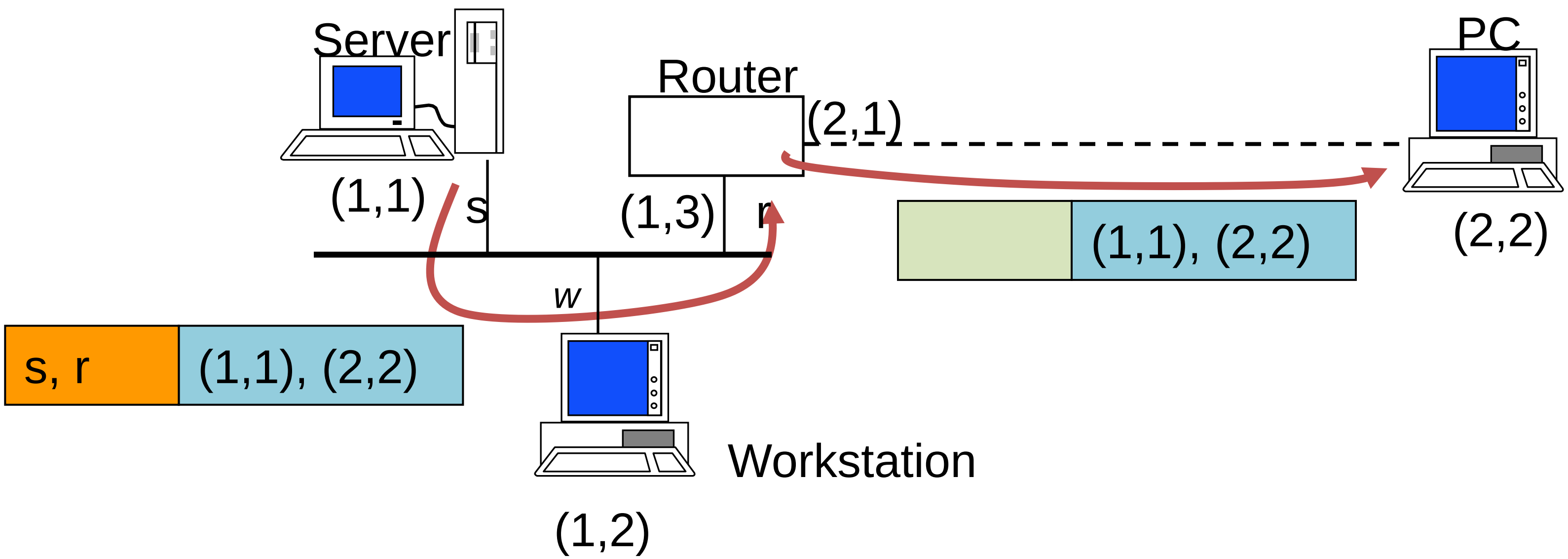
2. IP table at workstation indicates (1,1) connected to same network. Assume that the physical address of the server (s) is known. IP packet is encapsulated in Ethernet frame with addresses w and s

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

Example internet

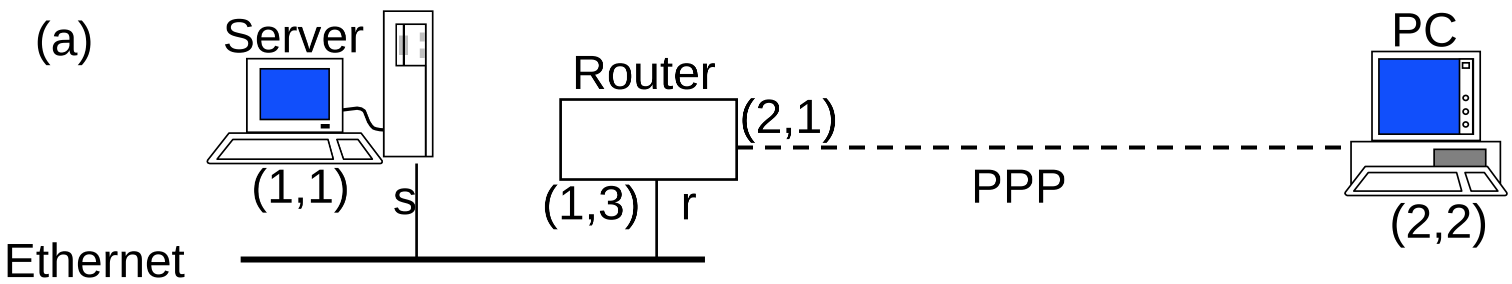
IP packet from server to PC

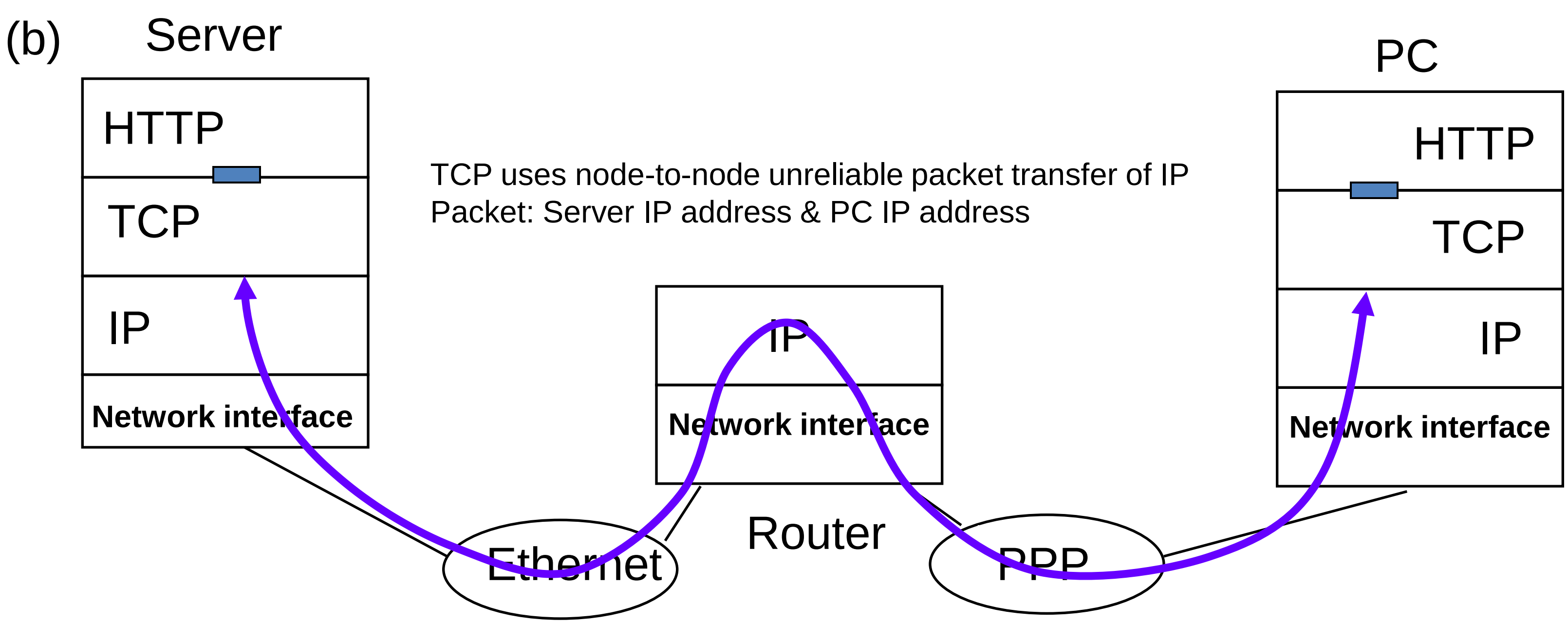


1. IP packet has (1,1) and (2,2) as IP source and destination addresses
2. IP table at server indicates packet should be sent to router, so IP packet is encapsulated in Ethernet frame with addresses s and r
3. Ethernet frame is broadcast by server NIC and captured by router NIC
4. NIC examines protocol 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

Example internet

How the layers work together

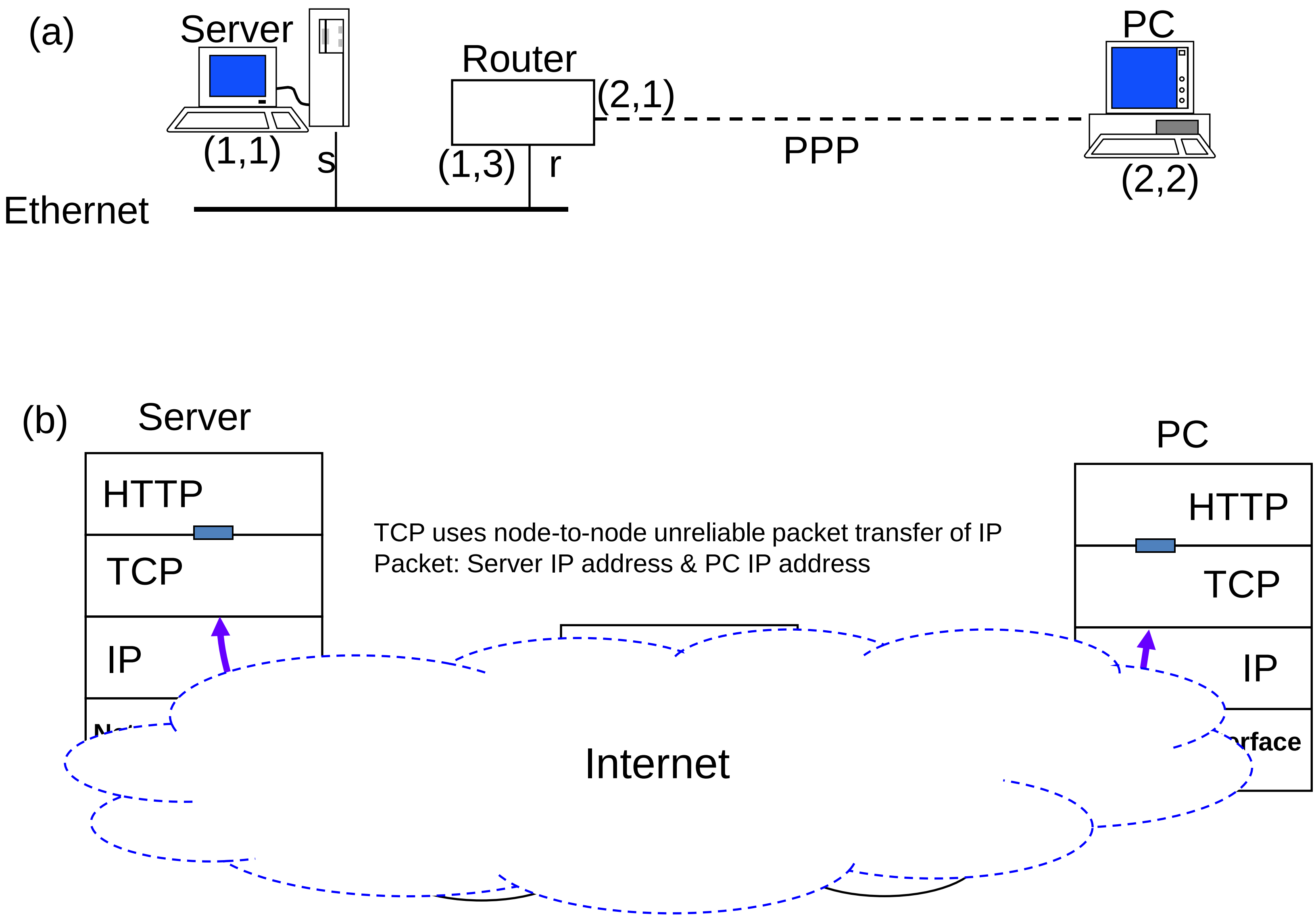




TCP uses node-to-node unreliable packet transfer of IP
Packet: Server IP address & PC IP address

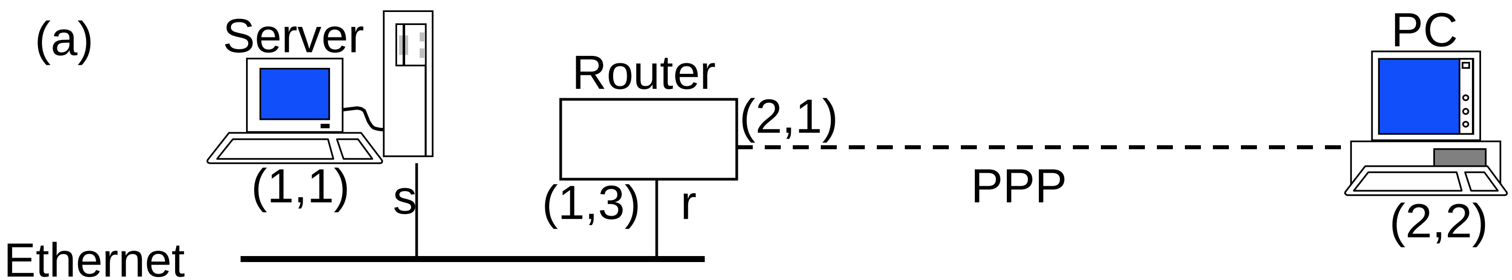
Example internet

How the layers work together



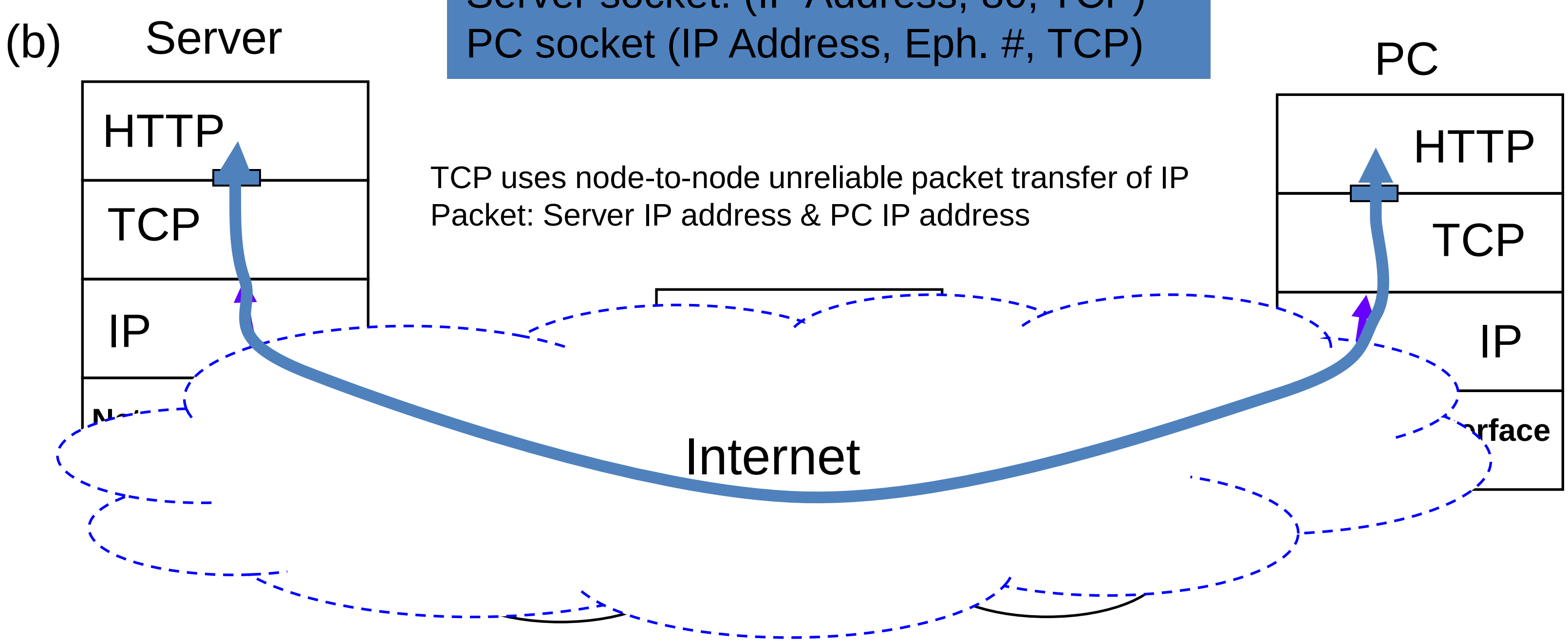
Example internet

How the layers work together



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

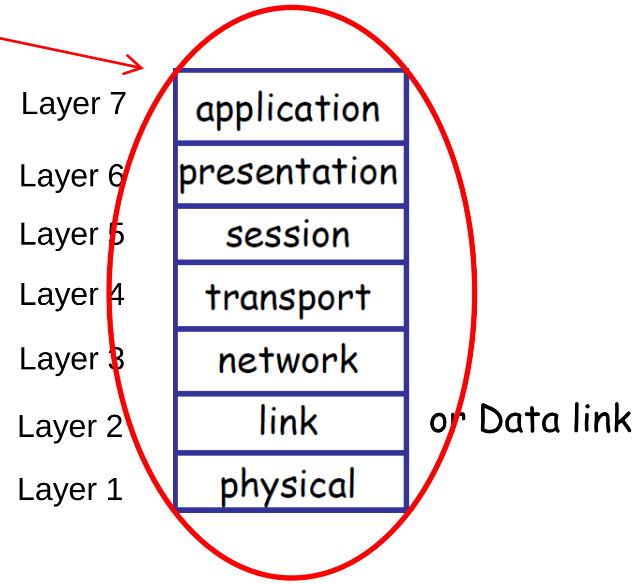
TCP uses node-to-node unreliable packet transfer of IP
Packet: Server IP address & PC IP address



Network Design: Layered Architectures

The OSI reference model (7 Layers)

- OSI reference model (7 Layers)



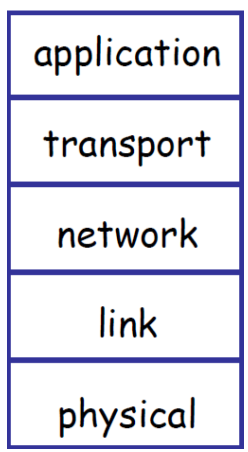
- Internet protocol stack or TCP/IP architecture or Internet Architecture

(There are two representations for this architecture: 5 layers or 4 layers)

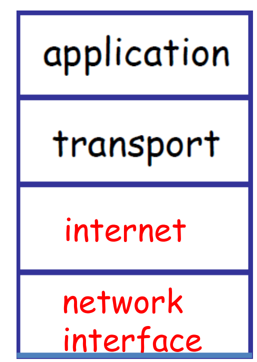
Protocol stack: the protocols of the various layers

TCP: Transmission Control Protocol
IP: Internet Protocol

The Internet = a TCP/IP network



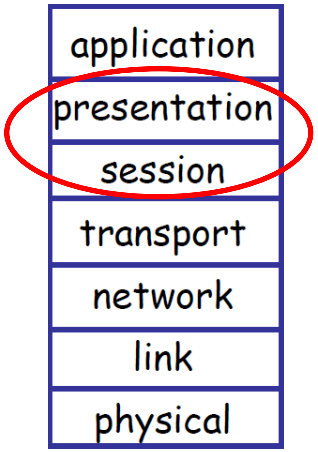
(5 Layers)



(4 Layers)

Network Design: Layered Architectures

The OSI reference model (7 Layers)…



The two layers not seen in the Internet protocol stack:

- Presentation Layer: This layer is intended to provide the application layer with independence from differences in the representation of data
  - For example, when communicating from application-A to application-B, it converts the machine-dependent information provided by application-A into a machine-independent form and then converts this back into machine-dependent form suitable for application-B
  - Different computers use different data representation (character, integer), and use different conventions such as how data is transmitted MSB first or LSB first

- Session Layer: dialog management…
  - provide name space to collectively refer to different transport streams that are a part of a single application eg. Video conferencing (audio and video streams)

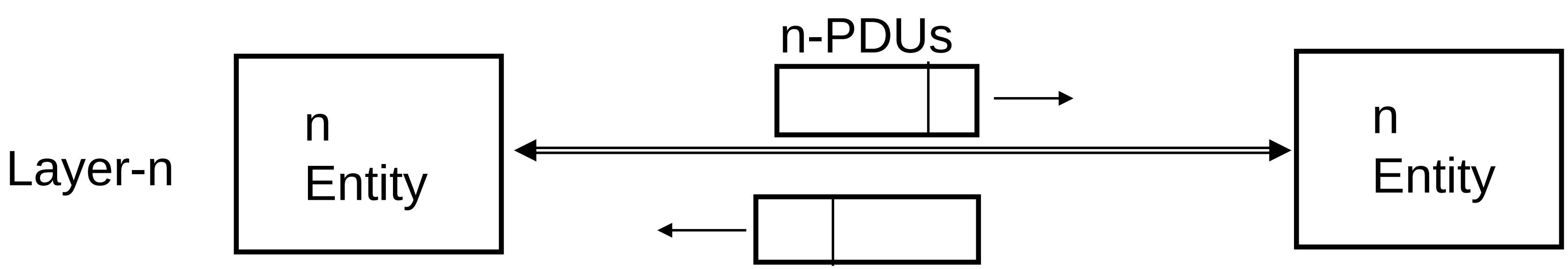
Internet stack “missing” these layers?
–these services, *if needed,* must be implemented in the application

History: OSI reference model, TCP/IP architecture…

- By the 1970s every computer vendor 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.
- It describes a seven-layer abstract reference model for a network architecture (OSI reference model).
- 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 protocol standards were developed for each layer, but most of these are not in use.
- TCP/IP protocols preempted the deployment of OSI protocols.
- TCP/IP architecture: Successor of ARPAnet (Defense Advanced Research Agency).
  - Principles:
    - Robustness: survive failures in the network
    - Flexibility: operate over diverse networks
- How come the Internet, which is based on the TCP/IP architecture, has become so successful?

OSI Unified View: Layers, Protocols, and Services

- On each layer, a process on one machine carries out a conversation with peer process on the other machine
- Processes at layer n are referred to as layer n entities
- Layer-n peer processes communicate by exchanging *Protocol Data Units* (PDUs)
  - PDU = Header (protocol control information) + Service Data Unit (SDU) (User information: data received from the higher layer)
- The behavior of layer-n entities is governed by a set of rules called layer-n protocol

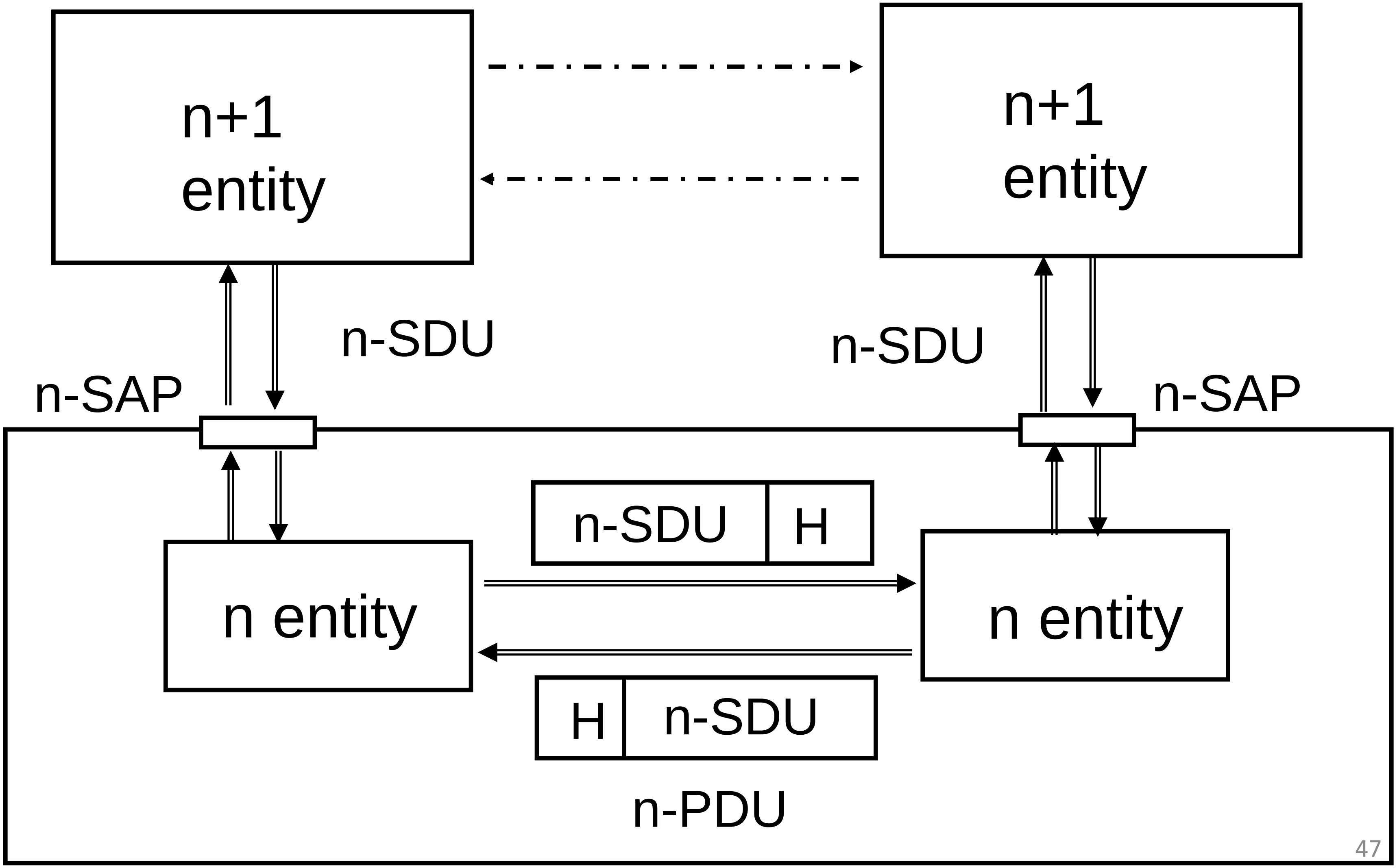


Peer-to-peer communication

OSI Unified View…

- Communication between peer processes is virtual and actually indirect.
- For communication to take place, layer n+1 entities make use of the services provided by layer-n.
- Layer n+1 PDU is passed from layer n+1 to layer n through a software port called the layer n *Service Access Point (SAP)* (Services are available at *Service Access Points).*
- 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

OSI Unified View…



OSI Unified View…
Connectionless & Connection-Oriented Services

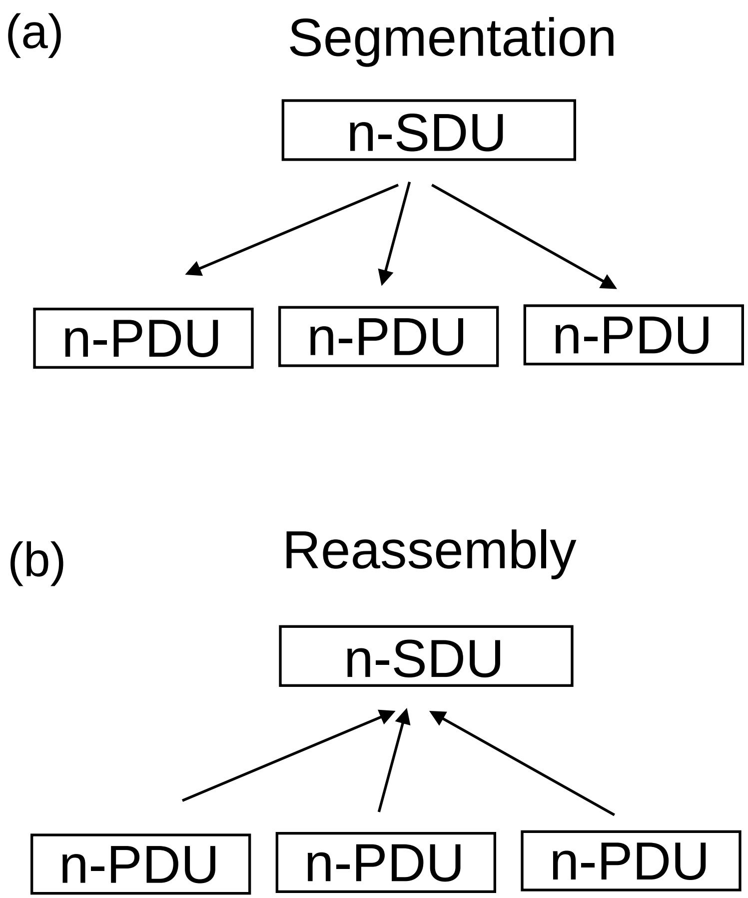
- Connection-Oriented
  - Three-phases:
    1. Connection setup between two SAPs to initialize state information
    2. SDU transfer
    3. Connection release
  - E.g. TCP, ATM

- 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

OSI Unified View…
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.



OSI Unified View…
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