

# CST 303: COMPUTER NETWORKS

Credits :4

A decorative graphic consisting of several horizontal lines of varying lengths and colors (light blue and white) extending from the right side of the slide.

## Expected Outcome

1. Visualise the different aspects of networks, protocols and network design models.
2. Examine various Data Link layer design issues and Data Link protocols.
3. Analyse and compare different LAN protocols.
4. Compare and select appropriate routing algorithms for a network.
5. Examine the important aspects and functions of network layer, transport layer and application layer in internetworking.

# Module I

- Introduction – Uses – Network Hardware – LAN – MAN – WAN, Internetworks – Network Software – Protocol hierarchies – Design issues for the layers – Interface & Service – Service Primitives. Reference models – OSI – TCP/IP.

# Computer Networks

- A computer network is a group of computer systems and other computing hardware devices that are linked together through communication channels to facilitate communication and resource-sharing among a wide range of users.

# Primary objective of Computer Networks

- Transfer data from machine A to machine B
- Facilitate access to remote information
- Facilitate sharing of data
- Facilitates person to person communication
- Facilitate Interactive Entertainment
- Not every machine is connected to every other machine
  - Establish connection between a pair of machines
    - Transfer data
  - Enable machines of different speeds to communicate with each other

# Applications

- World Wide Web
- Email
- Online Social Network
- Streaming Audio/Video
- File Sharing
- Instant Messaging
- 
-

# Uses

1. Business application
  - Resource sharing
  - Client-server model

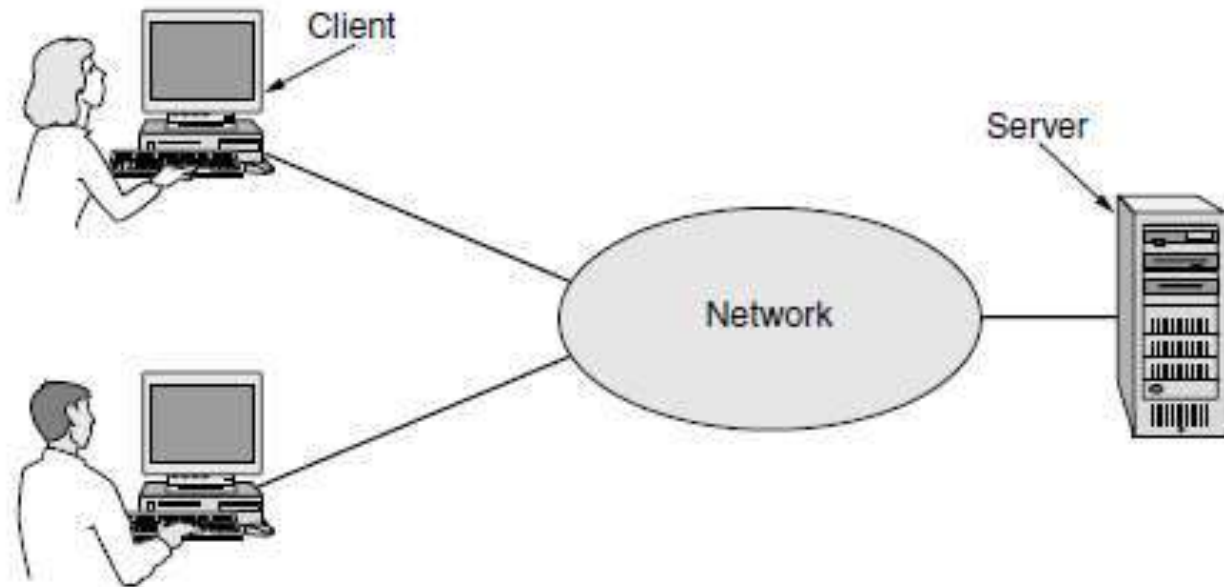


Figure 1-1. A network with two clients and one server.

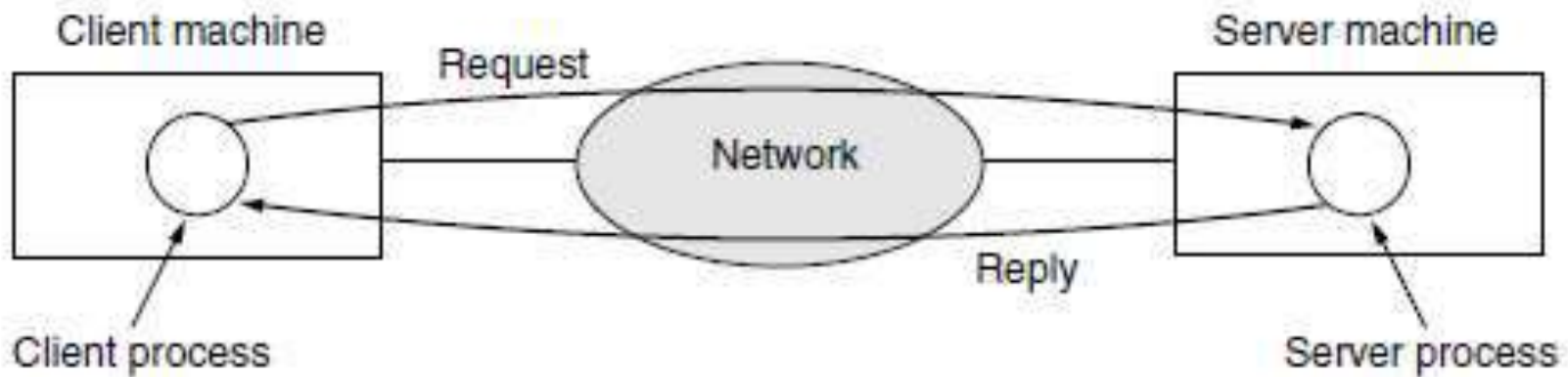


Figure 1-2. The client-server model involves requests and replies.

- Communication medium (e-mail)
- Video conferencing
- Doing business electronically, suppliers & customers
- E-commerce

## 2. Home Applications

- home users with connectivity to remote computers
- use e-commerce
- surfing world wide web
- online news papers
- IEEE, ACM journals

- Peer to peer communication
  - individuals who form a loose group can communicate with others in the group

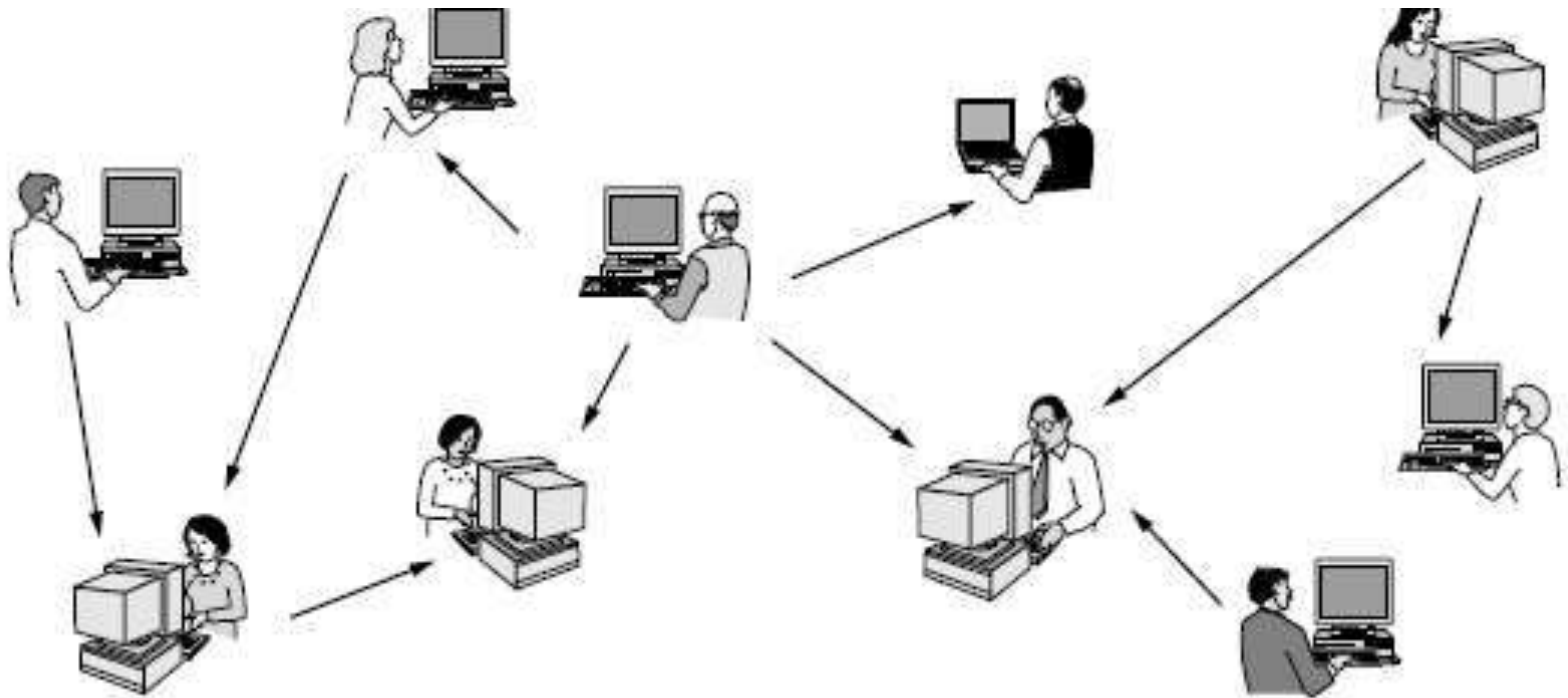


Figure 1-3. In a peer-to-peer system there are no fixed clients and servers.

1. Access to remote Information
2. Person-to-person communication
3. Interactive entertainment
4. Electronic commerce

Tag	Full name	Example
B2C	Business-to-consumer	Ordering books online
B2B	Business-to-business	Car manufacturer ordering tires from supplier
G2C	Government-to-consumer	Government distributing tax forms electronically
C2C	Consumer-to-consumer	Auctioning second-hand products online
P2P	Peer-to-peer	Music sharing

Figure 1-4. Some forms of e-commerce.

### 3. Mobile Users

- Connectivity to internet
- Wireless hotspots based on the 802.11 standard

Wireless	Mobile	Typical applications
No	No	Desktop computers in offices
No	Yes	A notebook computer used in a hotel room
Yes	No	Networks in unwired buildings
Yes	Yes	Store inventory with a handheld computer

Figure 1-5. Combinations of wireless networks and mobile computing.

- sms – short messaging service
- 4g
- GPS – Global positioning system
- Wearable computers
  - smart watch

## 4. Social Issues

- Social networks, message boards, content sharing sites, and a host of other applications
- People's privacy
- Captcha

# NETWORK HARDWARE

- Two dimensions for network design
  1. Transmission technology
  2. Scale
- two types of transmission technology
  - broadcast links
    - communication channel is shared by all the machines on the network.
    - received by all machines
    - an address field specifies recipient
    - Broadcasting
  - point-to-point links
    - connect individual pairs of machines
    - packets, may have to first visit one or more intermediate machines
    - Unicasting

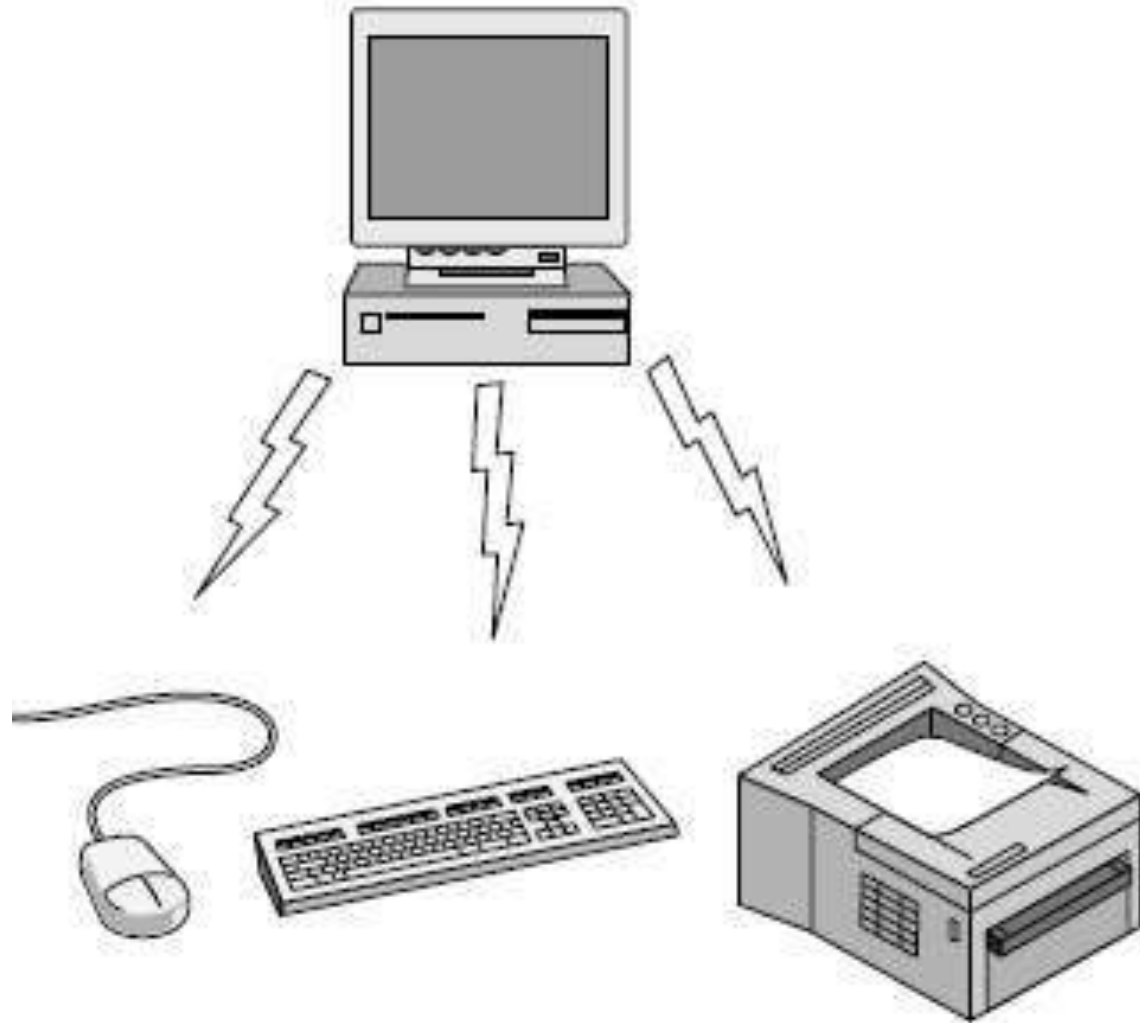
- Multicasting
  - A Source node might want to send a message to some subset of the other nodes, but not all of them.
- Connection of two or more networks is called an **internetwork**, Eg - **Internet**
- Alternate criterion to classify networks is by Scale (Distance)
  - Local
  - Metropolitan
  - Wide Area Networks

- Classification of interconnected networks by Scale

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	
100 m	Building	Local area network
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

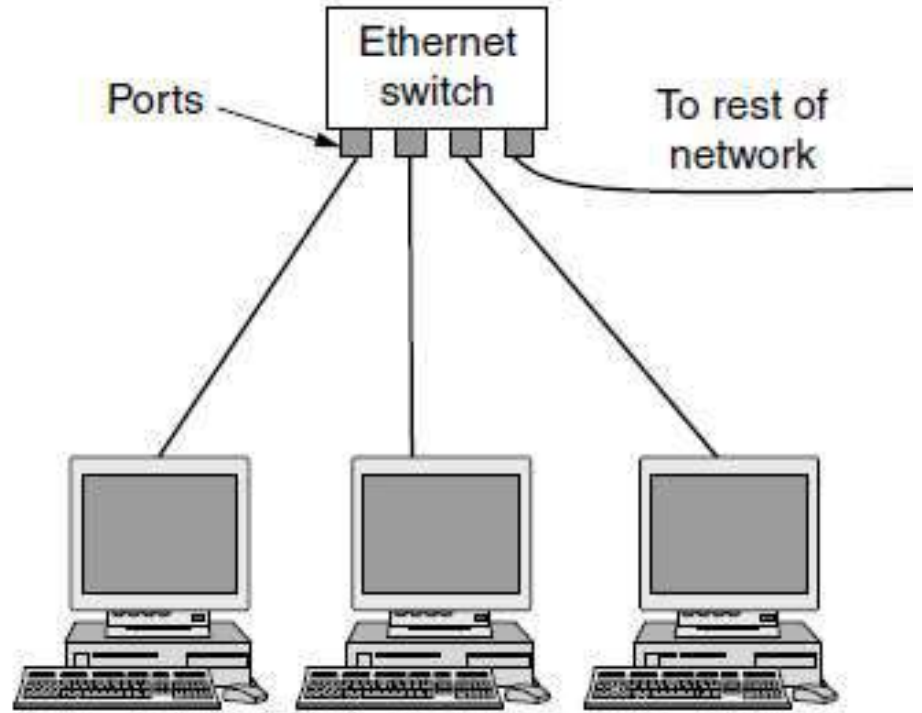
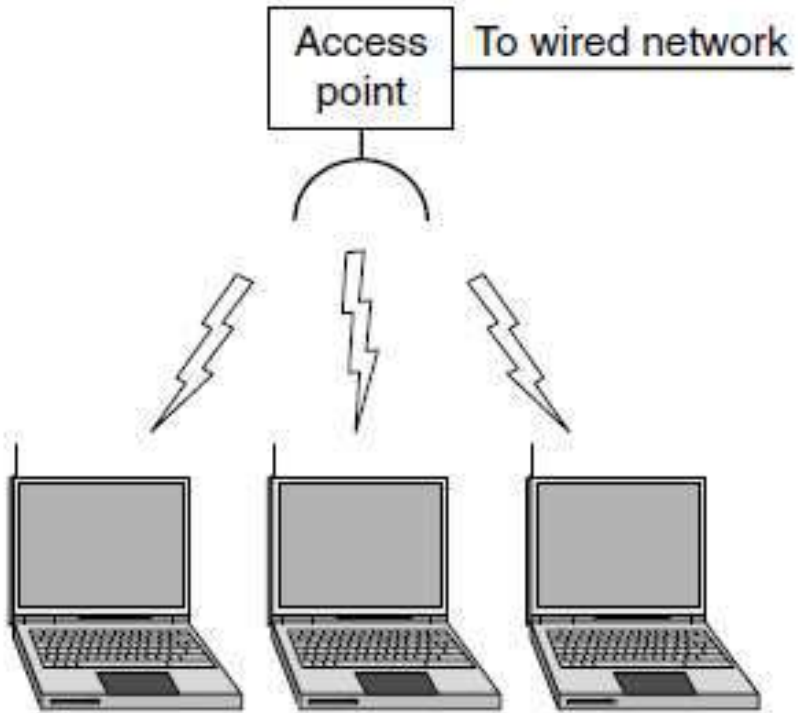
# 1. PAN (Personal Area Networks)

- Lets devices connect in the range of a person
  - Eg – computer connected to its peripherals.
  - Bluetooth
    - Enables master – slave paradigm
    - PC is the master, talking to
      - mouse
      - keyboard as slaves



## 2. LAN (LOCAL AREA NETWORK)

- LAN is a privately network that works within & nearby a single building like home, office or factory.
  - connected by personal computers & consumer electronics – resource sharing
  - companies, enterprise networks
  - each computer talks to device in the ceiling
    - AP, wireless router or base station
    - Wireless LAN standard : IEEE 802.11
    - 10 to 100 of Mbps.



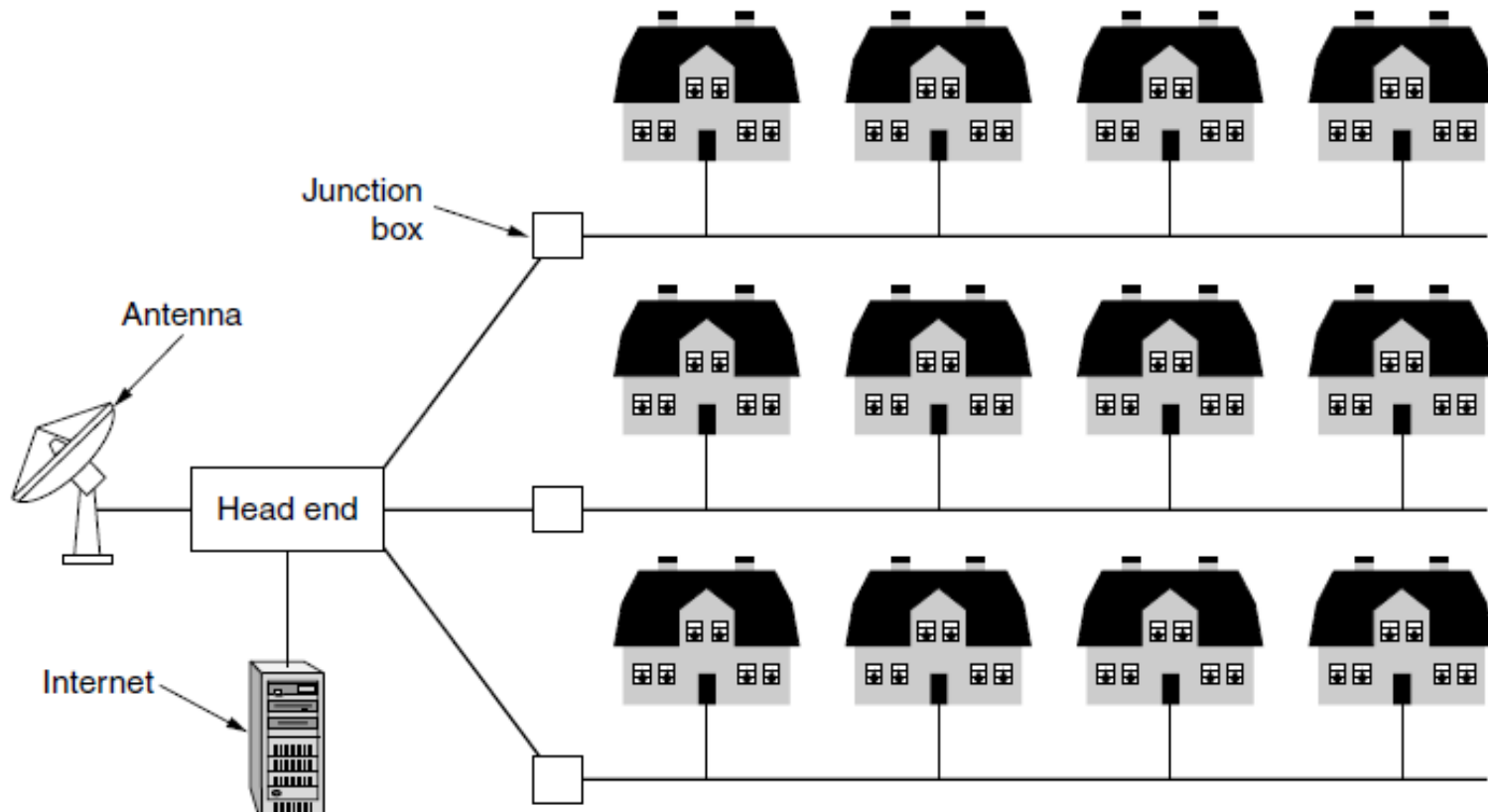
- Transmission technology
  - copper wire
  - optical fibre

LAN has to be

1. networked devices have to be easy to install
2. network & devices have to be foolproof in operation
3. low price is essential for success
4. no format wars
5. security & reliability

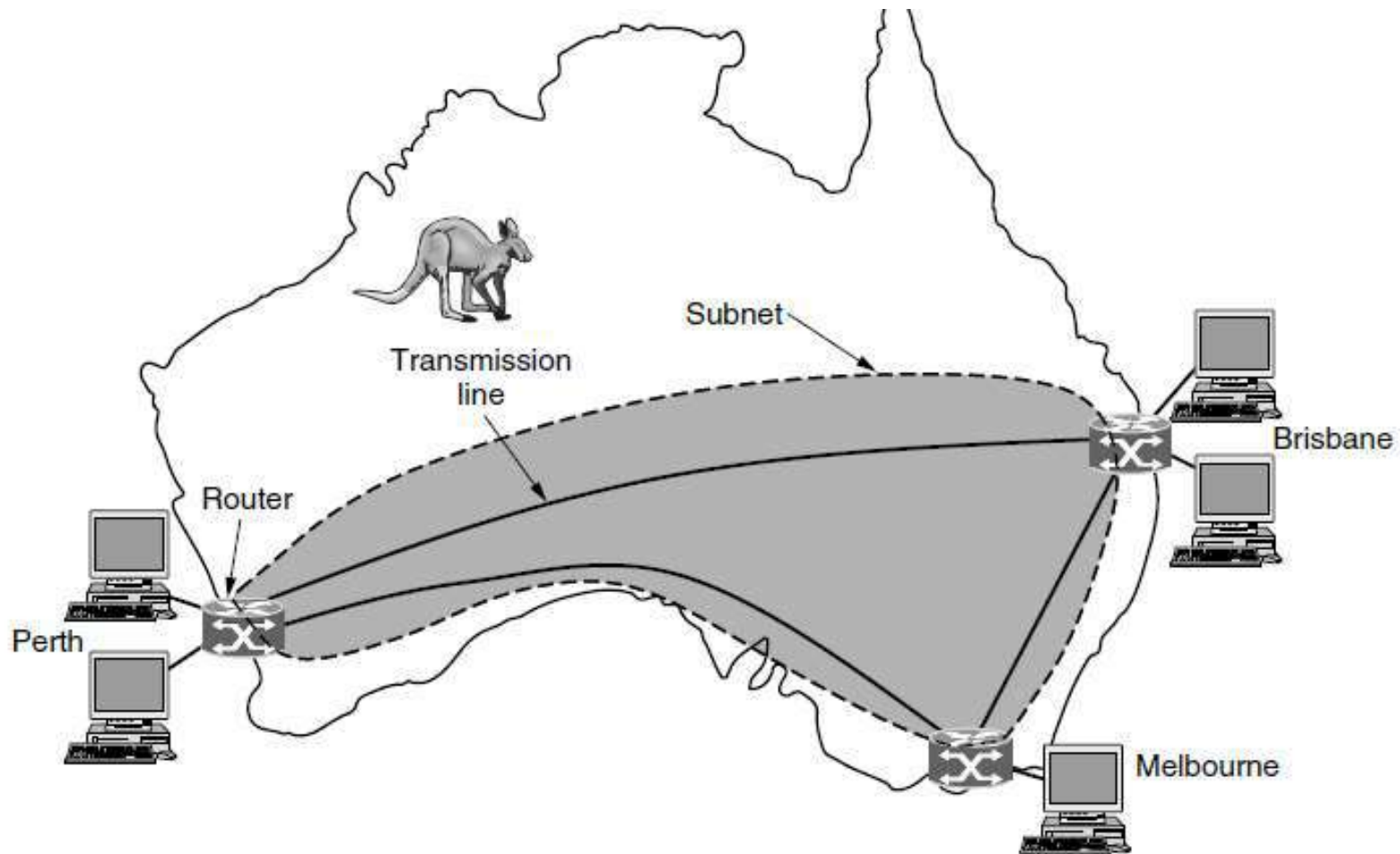
# 3. METROPOLITAN AREA NETWORKS

- Covers a city
  - Eg : Television network



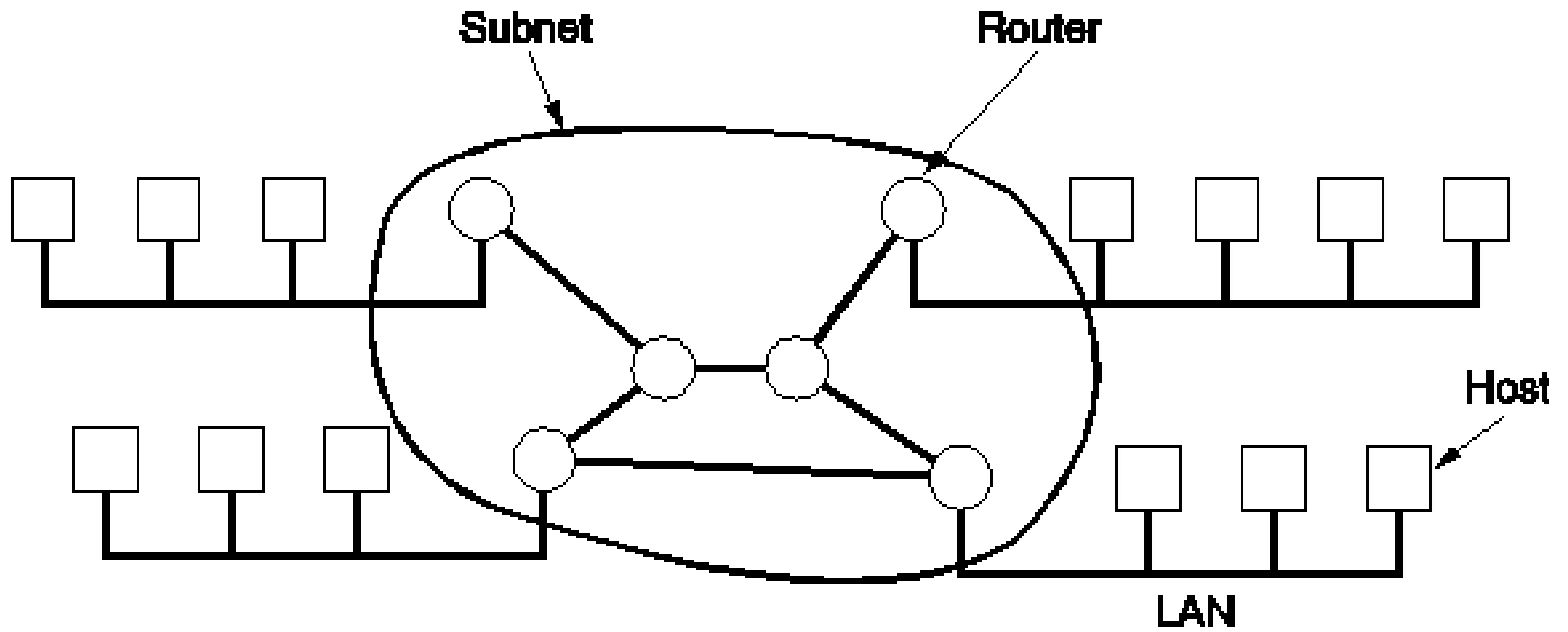
# 4. WIDE AREA NETWORKS

- Spans a large geographical area
- Eg – offices with different branches



host

- rest of the network that connects these hosts is communication subnet
  - subnet carries messages from hosts to hosts
  - 2 distinct components
    - transmission line (copper wire, optical fiber)
      - move bits between machines
      - leased telephone lines
    - switching elements (router)
      - switches are specialized computers connects two or more transmission lines



# INTERNETWORKS

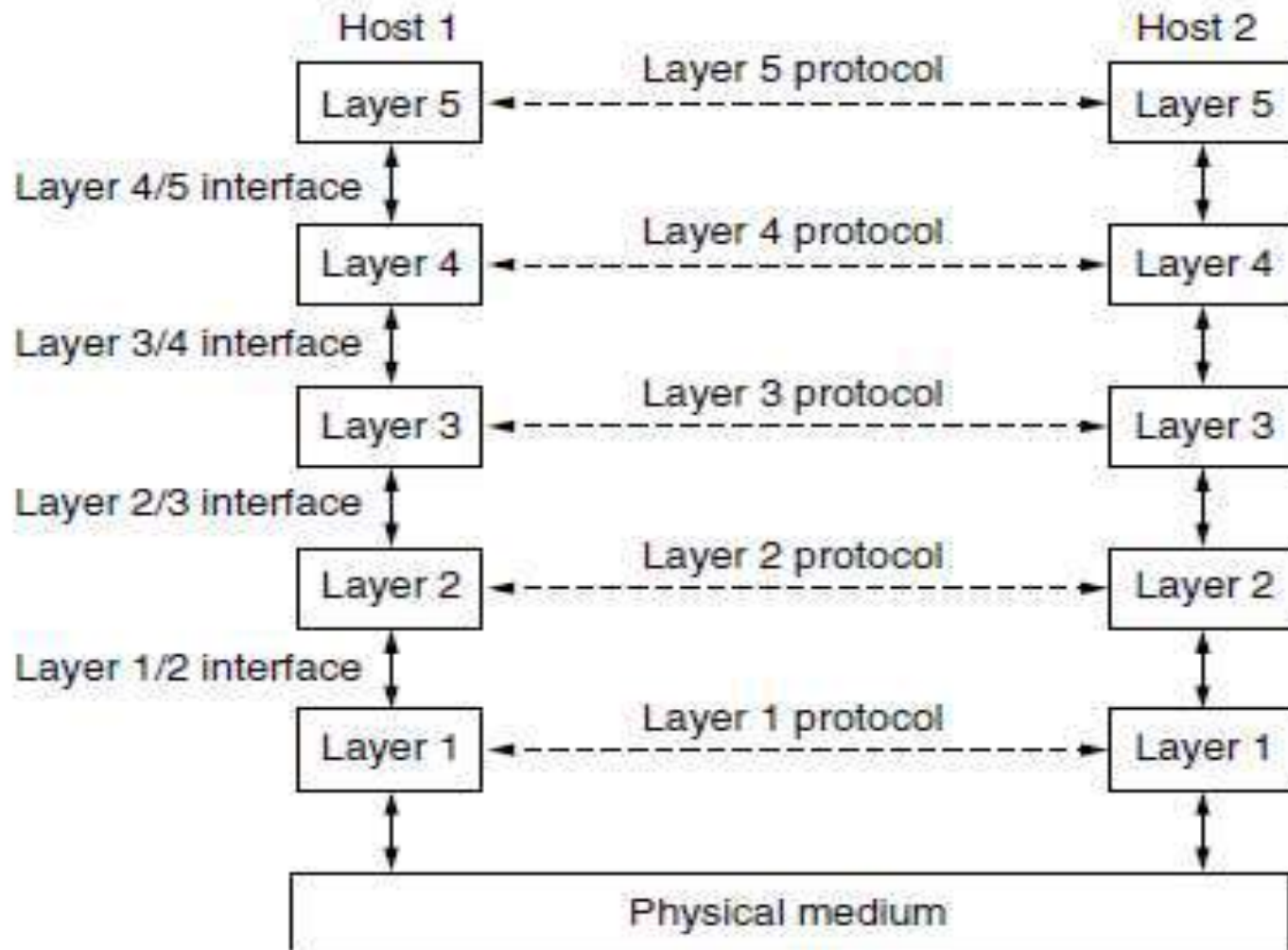
- A collection of interconnected networks is called an internetwork or internet
  - Collection of LANs connected by a WAN
  - Connecting a LAN and WAN or 2 LANs
- **GATEWAY**
  - The general name for a machine that makes a connection between two or more networks and provides the necessary translation, both in terms of hardware and software.

# NETWORK SOFTWARE

- **Protocol Hierarchies**

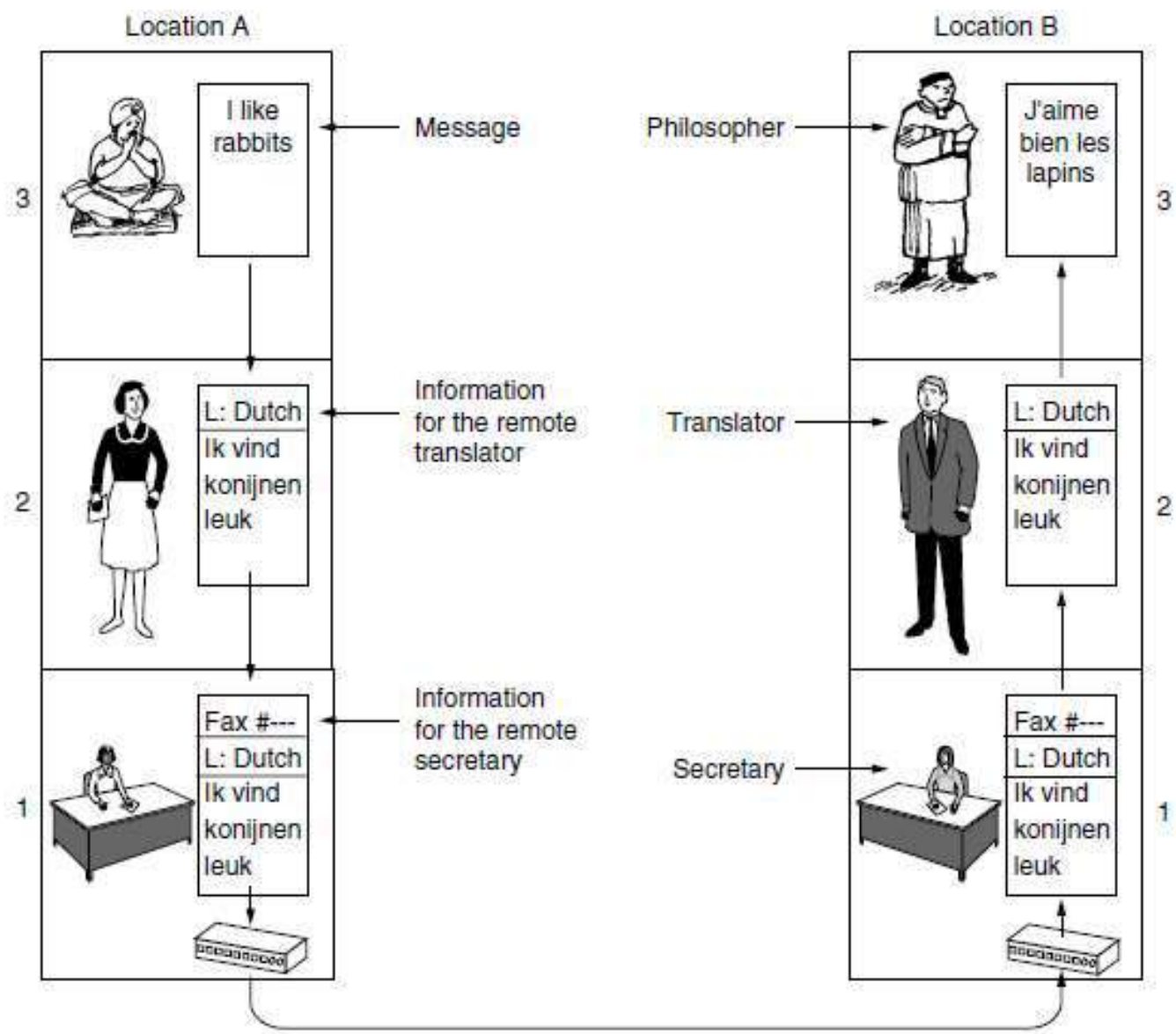
- To reduce design complexity, networks are designed as stack of layers or levels
- Purpose of each layer is to offer certain services to the higher layers while shielding those layers from the details of how the offered services are actually implemented
- **Protocol** is an agreement between the communicating parties on how communication is to proceed.

# Layers, protocols, interfaces

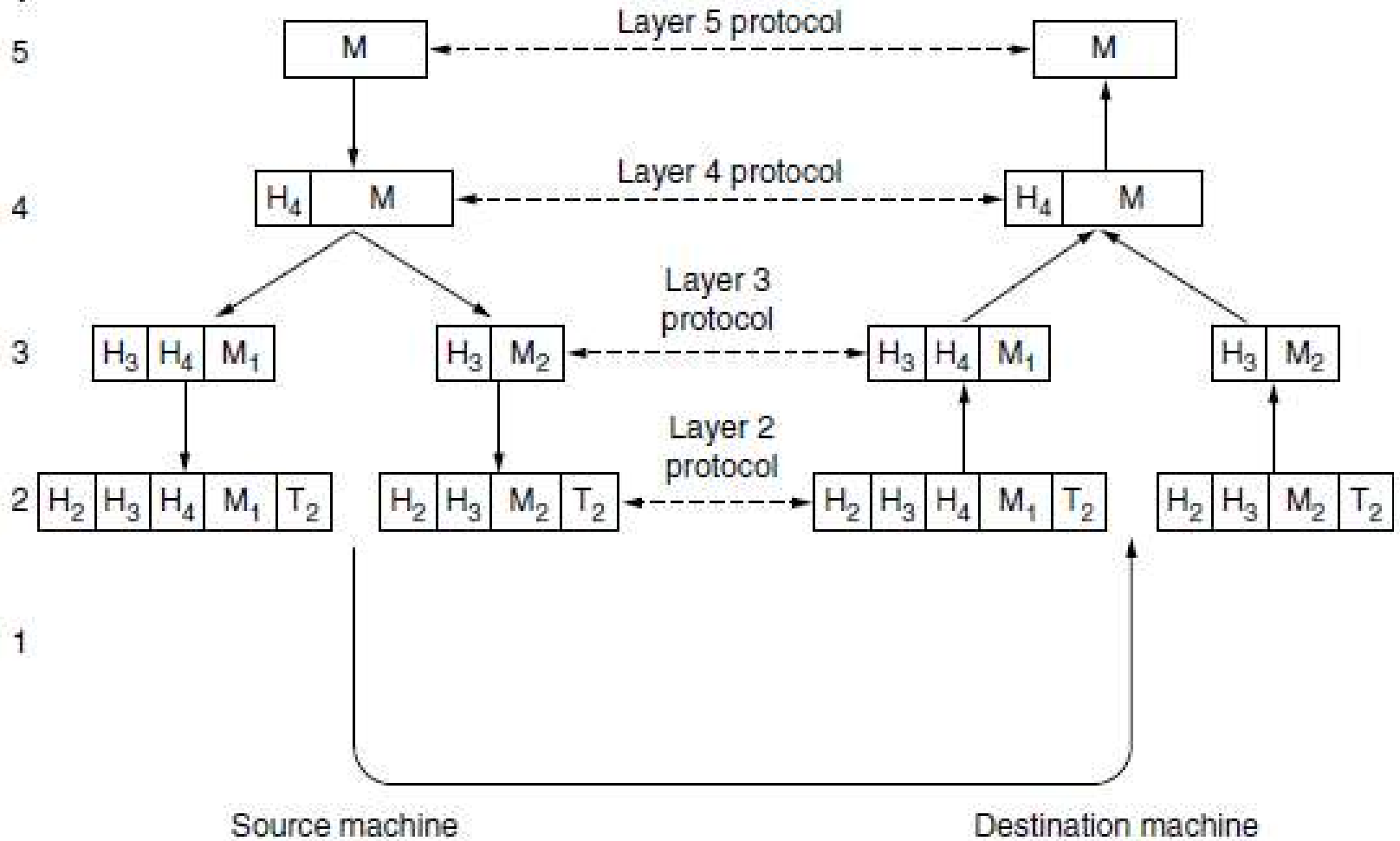


- Peer
  - software processes, hardware devices, or even human beings.
  - it is the peers that communicate by using the protocol to talk to each other.
- Each layer passes data and control information to the layer immediately below it, until the lowest layer is reached.
  - physical medium through which actual communication occurs
  - The interface defines which primitive operations and services the lower layer makes available to the upper one.

- Between each pair of adjacent layers is an interface.
- A set of layers and protocols is called a **network architecture**
- A list of the protocols used by a certain system, one protocol per layer, is called a **protocol stack**



Layer



# Design Issues for the Layers

## 1. RELIABILITY

- bits of packets/ inverted
- One mechanism for finding errors in received information uses codes for **error detection**
- Retransmitted/Error correction

## 2. Finding a working path through a network

- **Routing**

- Over time, networks grow larger/ new emerging designs needed, protocol layering
- In a message, senders & receivers need to be identified, **Addressing** or naming
- Different network technologies often have different limitations.
  1. order of messages
  2. maximum size of a message
    - Mechanisms (multiplexing, demultiplexing) for disassembling, transmitting, and then reassembling messages. This overall topic is called **internetworking**.

## Resource allocation

- capacity of transmission lines.
- **statistical multiplexing**, meaning sharing based on the statistics of demand.
- applied at low layers for a single link, or at high layers for a network or even applications that use the network.
- An **allocation problem** that occurs at every level is how to keep a fast sender from swamping a slow receiver with data.
  - Solution is feedback, thereby **flow control**
  - Overloading the network is **congestion**

# CONNECTION-ORIENTED & CONNECTIONLESS SERVICES

- Connection-Oriented Services
  - Service user first establishes a connection, uses the connection & releases the connection.
  - Acts like a pipe
    - sender pushes bits at one end
    - receiver takes them out at the other end
  - Sender, Receiver, subnet conduct a negotiation about parameters to be used
    - Max msg size
    - QoS required etc
    - One side makes proposal, & other may or may not reject it.

- Connectionless Service
  - Each message carries full destination address, & each one is routed through the system independent of all the others
  - if 2 messages are send, first one sent will be the first to arrive
  - also, first one can be delayed so that second one arrives first.
- Each service can be characterized by a Quality of Service.
  - reliable – never lose data
    - receiver ack data, so sender is sure that its arrived
    - overhead & delays

- Reliable connection oriented service has 2 minor variations:

### 1. Message Sequences

- message boundaries are preserved
- two 1024 byte messages are sent, they are two distinct messages at receiver
- eg - sequence of pages

### 2. Byte Streams

- no message boundaries
- one 2048 or two 1024 message not reliable
- eg - remote login

- **Unreliable connection** oriented service (telephone users hearing bit of noise than delay in ack)
- **Unreliable Connectionless** service is called datagram service (analogy to telegram service)
- **Acknowledged datagram** service (sending registered letter, ack is send back to sender)
- **Request-reply service** (client-server model)

# Six different types of services

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages of a book
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connectionless	Unreliable datagram	Network test packets
	Acknowledged datagram	Registered mail
	Request-reply	Database query

Sl no	Connection-oriented Service	Connection-less Service
1.	Connection-oriented service is related to the telephone system.	Connection-less service is related to the postal system.
2.	Connection-oriented service is preferred by long and steady communication.	Connection-less Service is preferred by bursty communication.
3.	In connection-oriented Service, Congestion is not possible.	In connection-less Service, Congestion is possible.
4.	Connection-oriented Service gives the guarantee of reliability.	Connection-less Service does not give the guarantee of reliability.
5.	In connection-oriented Service, Packets follow the same route.	In connection-less Service, Packets do not follow the same route.
6.	Connection-oriented Services requires a bandwidth of high range.	Connection-less Service requires a bandwidth of low range.

# Service Primitives

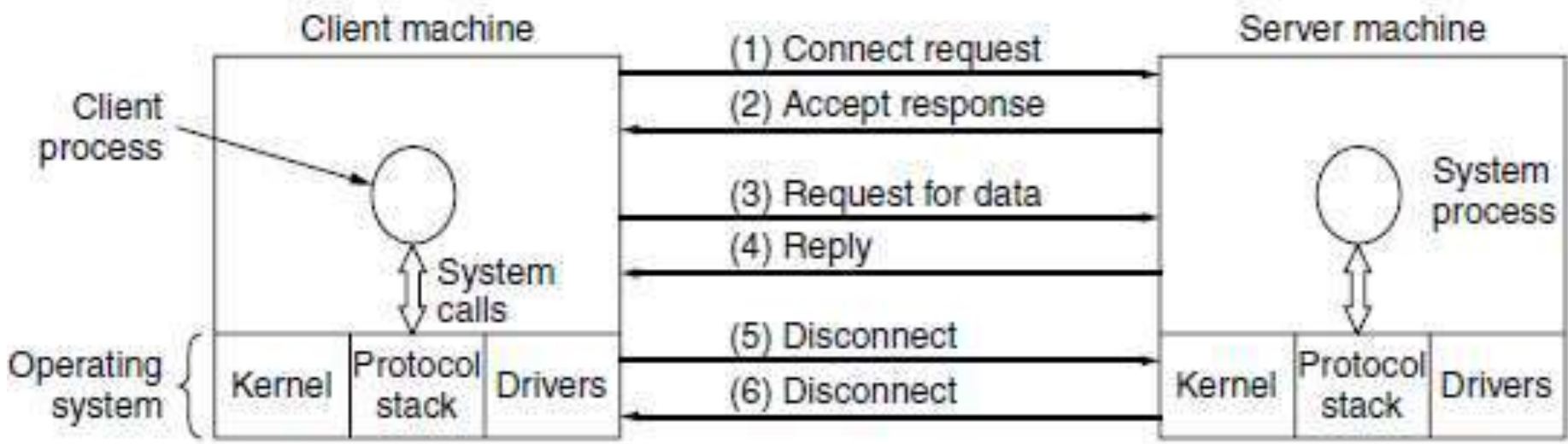
Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

To Implement a reliable byte stream in a Client-server model:

1. Server executes LISTEN to show its prepared to accept incoming connections
  - implement LISTEN is to make it a blocking system call
2. Client process executes CONNECT to establish a connection with the server.
  - The CONNECT call needs to specify who to connect to, so it might have a parameter giving the server's address
  - operating system sends a packet to the peer asking it to connect (1)
- 3.

3. When the packet arrives at the server, the operating system sees that the packet is requesting a connection.
  - It checks to see if there is a listener, and if so it unblocks the listener.
  - The server process can then establish the connection with the `ACCEPT` call.
4. Arrival of `ACK` unblocks the client.
  - Connection is established
5. Server to execute `RECEIVE` to prepare to accept the first request.
  - client executes `SEND` to transmit its request (3) followed by the execution of `RECEIVE` to get the reply

6. Server uses SEND to return the answer to the client (4).
7. When the client is done, it executes DISCONNECT to terminate the connection (5).
8. When the server gets the initial DISCONNECT packet, it also issues a DISCONNECT of its own, acknowledging the client and releasing the connection (6).
9. When the server's packet gets back to the client machine, the client process is released and the connection is broken.



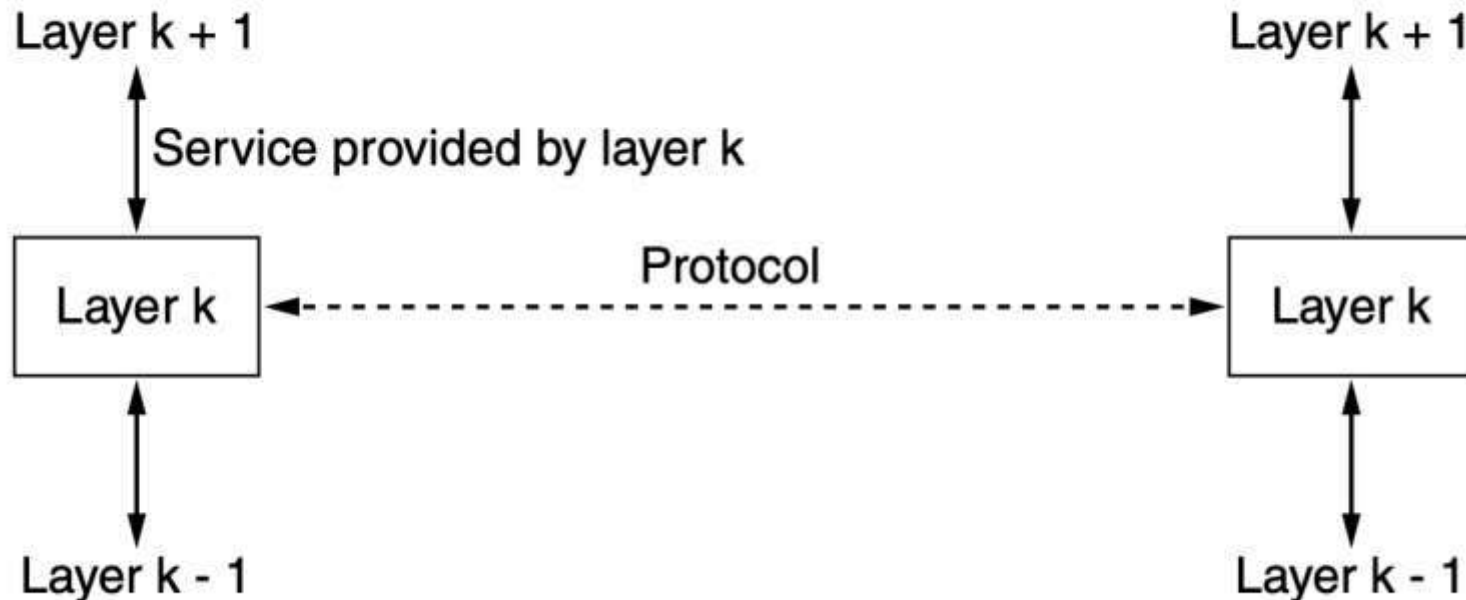
**Figure 1-18.** A simple client-server interaction using acknowledged datagrams.

## Relationship between Services to Protocols

- A service is a set of primitives (operations) that a layer provides to the layer above it.
  - defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented.
  - relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user.

- A protocol, is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer.
  - Entities use protocols to implement their service definitions.
  - free to change their protocols at will, provided they do not change the service visible to their users.

- Services relate to the interfaces between layers
- Protocols relate to the packets sent between peer entities on different machines



# Reference models

- **The OSI Reference Model**
- **TCP/IP Reference Model**

# The OSI Reference Model

- ISO (International Standards organisation)
- **OSI (Open Systems Interconnection) Reference Model** because it deals with connecting open systems.
- It just tells what each layer should do.
- 7 layers

## Principles that were applied to arrive at the 7 layers:

- A layer should be created where a different abstraction is needed
- Perform a well-defined function
- Toward defining internationally standardized protocols
- The layer boundaries should be chosen to minimize the information flow
- The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.

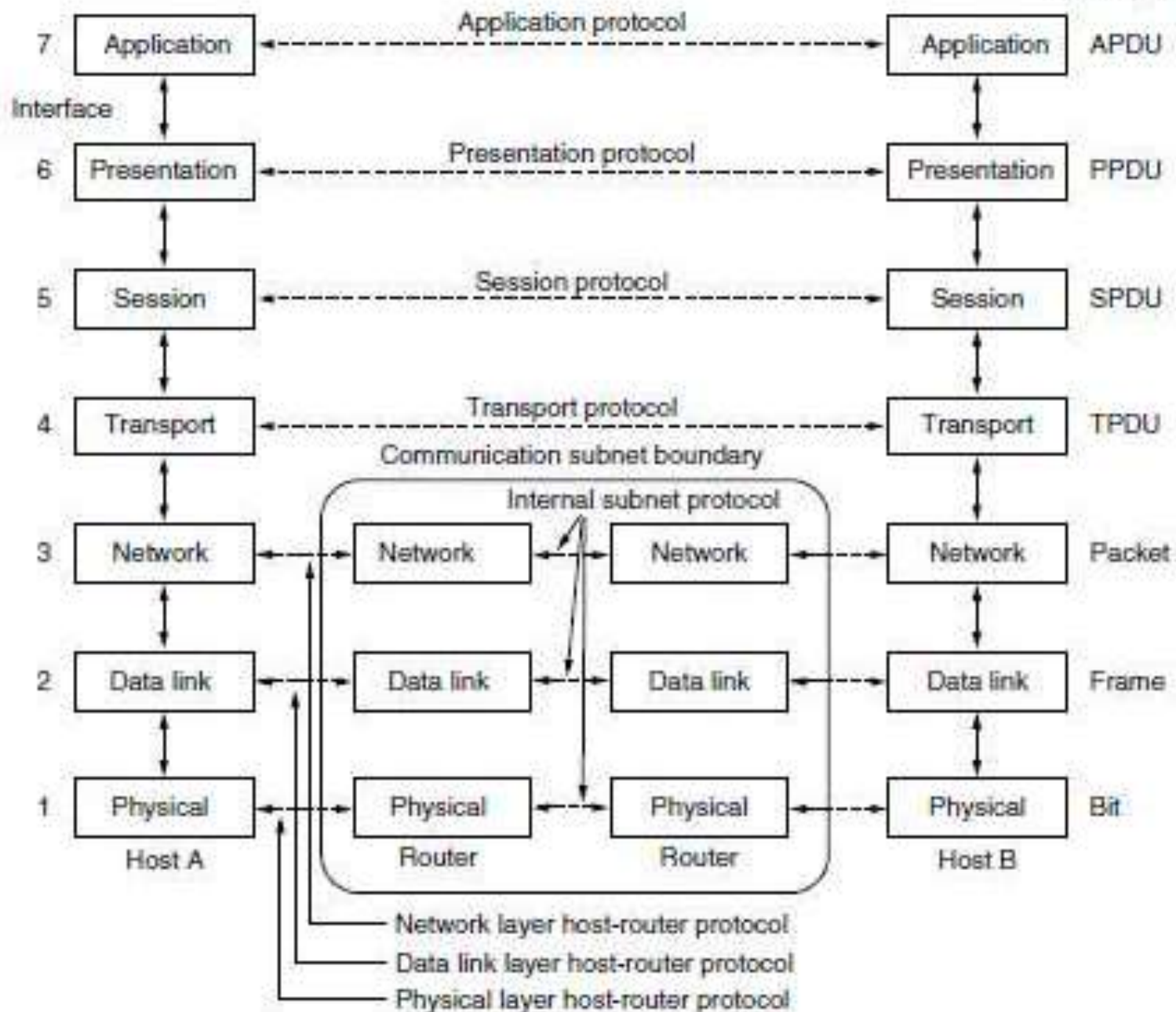


Figure 1-20. The OSI reference model.

# 1. The Physical Layer

- Concerned with transmitting raw bits over a communication channel
  - design issue □ 1 bit is send as 1 bit not received as 0 bit
  - how many volts to be used for 1 bit
  - how many nanosec 1 bit lasts
  - how initial connection is established
  - how it is torn down
  - how many pins network connector has,
  - what each pin is used for

## 2. The Data Link Layer

- Transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer
- Done by sender breaking up the input data into data frames & transmit frames subsequently
- Service reliable  ACK frame
- Design issue  fast transmitter drowning the slow receiver

### 3. The Network Layer

- Controls the operation of subnet
- Design issue □ how packets are routed from source to destination
- Too many packets, they will get to one another's way forming bottlenecks
- Quality of the network
- Addressing issues, Large packets, protocols different in a destination network

## 4. The Transport Layer

- Basic function □ accept data from above layers & split up into smaller units if needed
- Determines type of service to be provided to the session layer & to the users of the network.
- Transport layer is true end to end layer, from source to destination.
- Lower layers, protocols are between each machine & its immediate neighbors.

## 5. The Session Layer

- Allows users on different machines to establish sessions between them.
  
- Sessions offers various services
  - Dialog control [track of whose turn to transmit]
  
  - Token management [preventing 2 parties from attempting the same critical operation at the same time]
  
  - Synchronization [ checkpointing long transmissions]

## 6. The Presentation Layer

- Concerned with syntax & semantics of info transmitted
- Abstract data structures & high level data structures
- eg: Banking Records

## 7. The Application Layer

- Standardize common type of exchanges
- Contains a variety of protocols used by users
- HTTP basis for www
- FTP, EMAIL etc

# TCP/IP Reference Model

- Reference model in ARPANET, world wide web
- ARPANET : network model of US Dept. of Defense
- Connected hundreds of universities and government installations, using leased telephone lines.
- satellite & radio networks were added later
- created trouble, internetworking □ new reference model, TCP/IP reference model

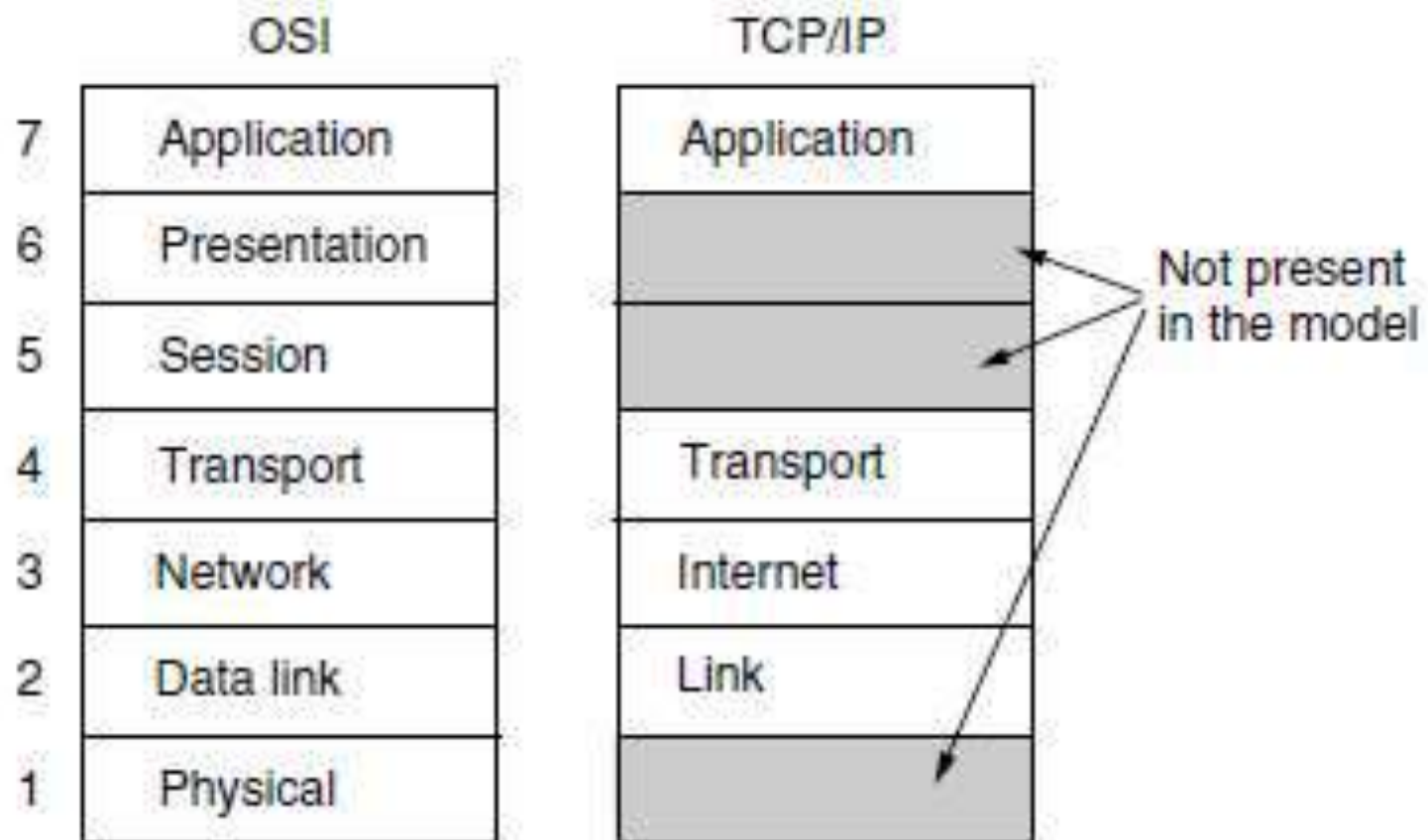
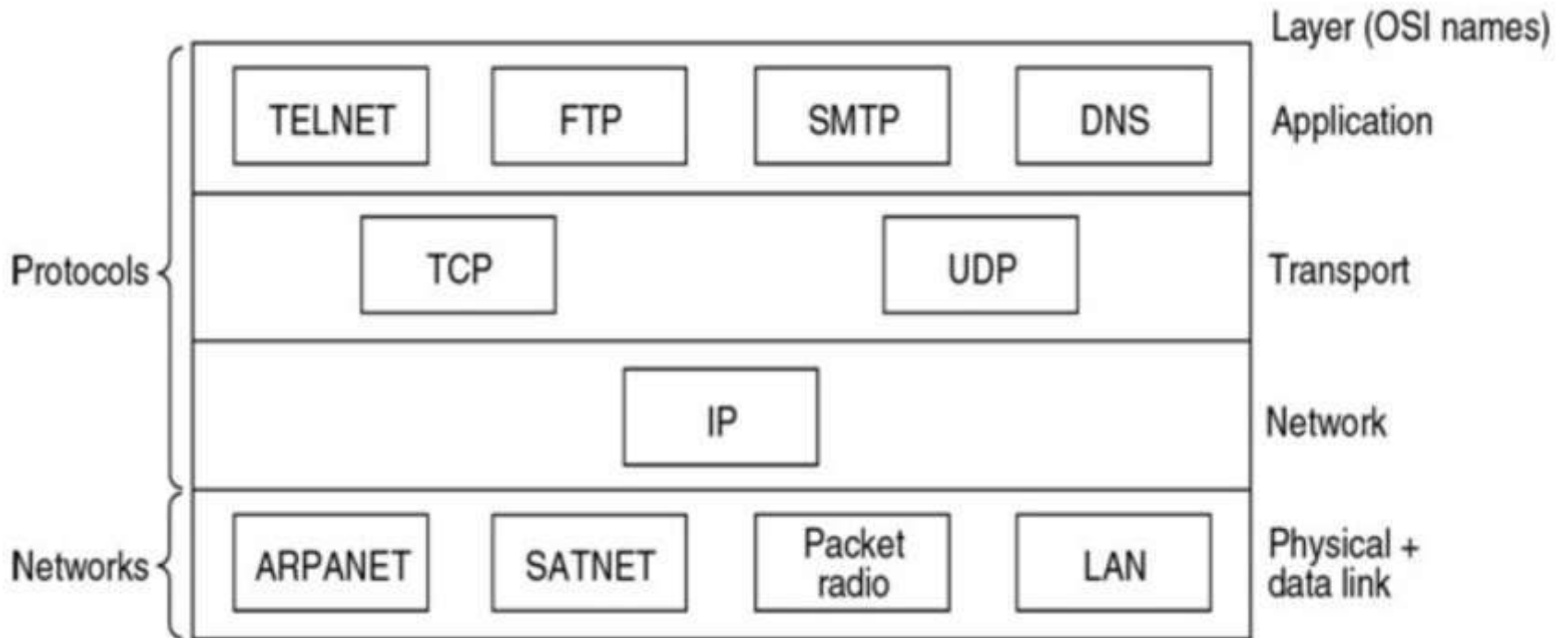


Figure 1-21. The TCP/IP reference model.



## Protocols and networks in the TCP/IP model initially

# 1. Link Layer(Host-to-Network Layer)

- describes what links must do to meet the needs of this layer.
- protocol is used to connect to the host, so that the packets can be sent over it.
- interface between host and transmission links

## 2. Internet Layer

- layer which holds the whole architecture together.
- permit hosts to inject packets into any network and have them travel independently to the destination.
- different order of arrival, higher layers have to rearrange if in-order delivery is desired.
- defines a packet format & protocol, called, **Internet Protocol, Internet Control Message Protocol.**

- The various functions performed by the Internet Layer are:
  - Delivering IP packets
  - Performing routing
  - Avoiding congestion

# 3. Transport Layer

- The layer above the internet layer in the TCP/IP model
  - allow peer entities on the source and destination hosts to carry on a conversation
  - 2 end-to-end protocols
    - TCP (Transmission Control Protocol)
    - UDP (User Datagram Protocol)

- TCP

- reliable connection-oriented protocol
- allows a byte stream originating on one machine to be delivered without error on any other machine in the internet
- fragments the incoming byte stream into discrete messages and passes each one on to the internet layer
- destination □ receiving TCP process reassembles the received messages into the output stream
- TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle

- **UDP**

- unreliable, connectionless protocol

- applications that do not want TCP's sequencing or flow control and wish to provide their own.

- used for client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video.

## 4. Application Layer

- Do not have presentation / session layers
- higher level protocols:
  - Virtual terminal (TELNET)
  - File transfer (FTP)
  - Electronic mail (SMTP)
  - DNS
  - HTTP
  - RTP

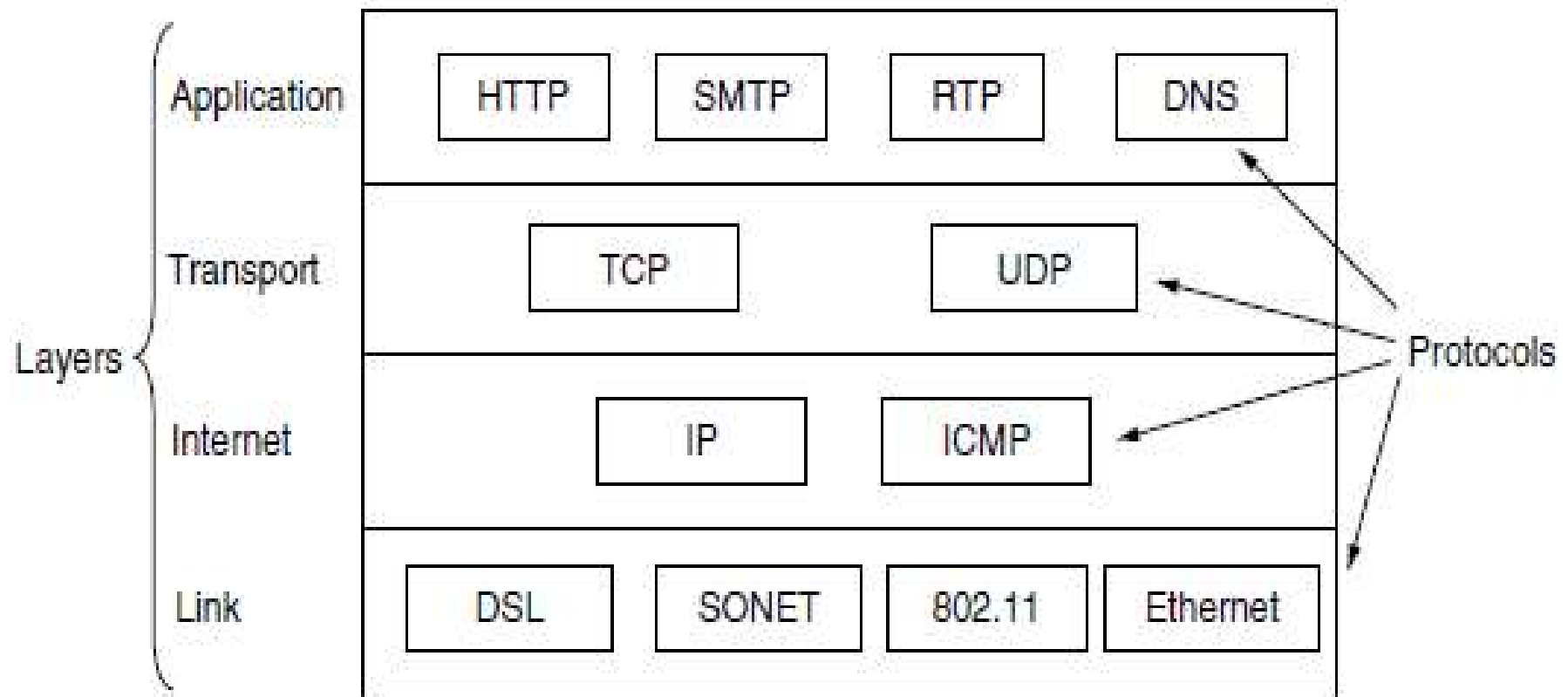


Figure 1-22. The TCP/IP model with some protocols we will study.

TCP	UDP
Connection-oriented protocol.	Connectionless protocol.
TCP reads data as streams of bytes, and the message is transmitted to segment boundaries.	UDP messages contain packets that were sent one by one. It also checks for integrity at the arrival time.
TCP messages make their way across the internet from one computer to another.	It is not connection-based, so one program can send lots of packets to another.
TCP rearranges data packets in the specific order.	UDP protocol has no fixed order because all packets are independent of each other.
The speed for TCP is slower.	UDP is faster as no error recovery
Header size is 20 bytes	Header size is 8 bytes.
TCP is heavy-weight.	UDP is lightweight.
TCP does error checking and also makes error recovery.	UDP performs error checking, but it discards erroneous packets.
Acknowledgment segments	No Acknowledgment segments
Using handshake protocol like SYN, SYN-ACK, ACK	No handshake (so connectionless protocol)
TCP is reliable as it guarantees delivery of data to the destination router	The delivery of data to the destination can't be guaranteed in UDP