hosts = end systems = client + server

<u>what is internet?</u> system of million connected computing devices providing communication links within diff media (fiber, copper wire, radio, satellite)

-diff Transmission rates (bandwidth (bps)) $= 10^3$ Kbps to 10^6 Mbps to 10^9 Gbps

-router/switches = forward packets thru packet switches network

-an infrastructure that enables distributed services/apps (web email, games, file sharing)

units: speed (bps) - kilo (K) $= 10^3 = 1000$
- mega (m) $= 10^6 = 1{,}000{,}000$
- Giga (G) $= 10^9 = 1{,}000{,}000{,}000$

capacity/size (bytes B) - kilo (K) $= 2^{10} = 1024$
- mega (m) $= 2^{20} = 1048576$
- Giga (G) $= 2^{30} = 1073741824$

Time (s) - milli (m) $= 10^{-3} = 0.001$
- micro (μ) $= 10^{-6} = 0.000001$

8 bits = 1 Byte

<u>access technologies:</u>

homeaccess	enterprise access	wide area network
DSL	wifi	3G, 4G, 5G, LTE
dialup modem	ethernet	
wifi	Telephone line	
cable internet	FTTH (fiber to the home)	
Twisted copper wire	Twisted copper wire	

approaches to moving data thru networks:

circuit switching	Packet switching
dedicated circuit paths per call used by all data	data sent in packets
end-end resources reserved for call b/w source to dest.	resources not reserved; on demand and may have to queue to access comm. link; forwards packets from routers to next on path from source to dest.
ex: Telephone networks	ex: internet
when network establishes a circuit connection, it reserves a constant R in networks link for duration of connection R = Transmission rate	packet sent into network w/o reservation so if one link is congested bc packets are being Transmitted at the same time then packet will have to wait in buffer at sending side of Trans link and suffer delays
R is reserved for sender-receiver connection; sender can transfer data to receiver at guaranteed constant rate; guaranteed performance	internet will make constant effort to deliver packets in Timely manner but it cannot make any guarantees; no performance guaranty.
R of 100 Kbps must be reserved for all users; Trans capacity of a link is 1mbps so reserved for 10 users	Trans. rate: L/R; Packets of L bits onto link at R bps; if 35 users are allowed to send packets at a time then low probab. of more than 10 users to be active at the same time
segments sit idle if not used on call; no sharing of resources	link Trans capacity is shared on packet-by-packet basis among users
each user is allocated a time slot per frame	allows more than 3 times the number of users

4 sources of delay at each hop along the path from src. to dest.

① propagation ② Transmission ③ processing ④ queuing

$d_{hop} = d_{proc} + d_{queue} + d_{Trans} + d_{prop}$

<u>Trans delay:</u> Time to Transmit packet onto link $= L/R$
L (size of packet)/R (Trans speed); units in sec(s) or millisec(m)
-when last bit of packet is transmitted

<u>prop. delay:</u> time a packet takes to travel across a link $= d/s$
d (distance from one end to other)/s (prop. speed) $= 2.5\times10^8$ m/s
-units in sec(s) or millisec(m); when first bit of packet will reach the other end of link; depends on physical medium (fiber, coaxial cable, air, etc)

<u>end-end delay:</u> Time it takes for entire packet to reach receiver $= d_{Trans} + d_{prop} = L/R + d/s$ in seconds (adds dproc + dqueue if needed)

<u>proc delay:</u> checks for bit errors;

<u>queue delay:</u> Time spent waiting for Transmission; depends on congestion
-units in micro(μ) or millisecs(m); packets can get lost if buffer is full
-if packet arrival rate (La) to link exceeds trans. rate capacity (R) then packets will queue; $La/R \sim 0$ = avg small queue delay
$La/R \to 1$ = avg large queue delay; $La/R > 1$ = avg delay infinite

<u>Protocol:</u> defines the format, order of messages sent + received among network entities, action taken on msg. Trans + receipt.

<u>five layer TCP/IP protocol stack:</u> ① application layer, ② Transport, ③ network ④ link layer, ⑤ physical layer

<u>seven layer OSI reference model:</u> addition of ⑥ presentation + ⑦ session

<u>app layer:</u> used to send data over multiple end systems
- HTTP, FTP, SMTP (email), DNS; referred to as <u>'message'</u>

<u>Transport:</u> host-host data transfer
- TCP, UDP; referred to as <u>'segment'</u>

<u>network:</u> routing of packets (datagrams) from src. to dest.
- IP protocol, routing protocols; refer to as 'IP layer'

<u>link:</u> moves packets from one node to another in route
- reliable delivery; ethernet, wifi, cable access network DOCSIS protocol (not TCP); referred to as <u>'frames'</u>

<u>physical:</u> moves individual bits within from from 1 node to next
-link dependent protocols, depends on physical medium

<u>presentation:</u> allows apps to interpret meaning of data;
-data compression, encryption, description

<u>session:</u> supports synchronization and delimiting of data exchanged; includes building checkpoint + recovery scheme

<u>encapsulation:</u> (process from app layer to link layer)

<u>throughput:</u> min {Rs, Rc, R/m}; rate at which hosts exchange bits

<u>bottleneck:</u> link on end-end path that constraints end-end throughput
-link with smallest capacity

<u>Circuit Switching:</u> physical path is obtained and reserved to a single connection b/w two endpoints for duration of connection

<u>Socket:</u> 'endpoint' in connection for sending + receiving data b/w two programs running on network; used w set of request or 'function calls'

<u>Port:</u> uniquely identify diff apps/processes running on single computer and enable to share a single physical connection to packet-switched network

<u>DNS:</u> domain naming system for computers, services, resources connected to Internet

<u>Proxy:</u> machine to satisfy client HTTP request w/o contacting origin server
nearby clients send their HTTP request to this machine

<u>Trace route:</u> comp. network diagnostic Tool for displaying route/path + measuring Transit delays of packets across an IP network

<u>Pipelining:</u> Sender is allowed to send multiple packets w/o waiting for ACK

if network sources send data at constant bit rate, Circuit Switching more desirable because there is no statistical multiplexing gains to be had, by using circuit each connection will get constant amount of bandwidth

<u>App Architecture:</u> designed by app dev who decides how app is structured over end systems. <u>Client-Server Architecture:</u> server that services requests for client

<u>Server:</u> always on host; permanent IP Address; server forms for scaling

<u>Client:</u> comm. w server; may be intermittently connected; dynamic IP add. do not communicate directly w each other

<u>Data Center:</u> virtual server to keep up with client requests

<u>P2P Architecture:</u> min or no reliance on dedicated servers in data centers
-peers: laptop, desktops; peers communicate w/o passing thru server; cost effective
-not always on server; arbitrary end systems directly communicate
-peers intermittently connected and change IP address; highly scalable
-hard to manage

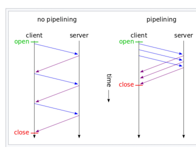
<u>Hybrid of Client-Server + P2P:</u> <u>Napster:</u> File Transfer P2P
File search centered at
-Peer register content at central server
-Peers query same central server to locate content

<u>Processes:</u> communication by sending/receiving messages across comp. network
to identify a process running on another host: IP address + port #
↳ 32 bit

<u>Handshaking Protocol:</u> server + client exchange control info before sending data to each other

<u>TCP</u> uses three-way handshake to create reliable connection; has full-duplex connection and guarantees reliable data transfer over connection simultaneously and after server-client finish, conn. terminates

Neither UDP and TCP provide encryption; TCP with (SSL) secure socket layer provides encryption, security, etc.



Pipelining: sender can send multiple packets without waiting for ACKs ⤵ cons.
-range of seq. #s must be increased
-sender + receiver may have to buffer more than one packet

Two approaches for pipeline error recovery:
Go Back N + selective repeat

<u>Go-Back-N Protocol:</u> link layer protocol that uses sliding window method for reliable + sequential delivery of data frames.
GBN can have performance issues, when window size + bandwidth delay are large, many packets can be in pipeline
A single packet error can cause GBN to retransmit large number of packets
GBN can allow a max num of packets, it must wait one or more packet to be ACK before proceeding so if packet 2 is lost then rest after are out of order and dropped.

hosts = end sys ... client + server
what is internet? system of million connected computing devices providing communication links within diff media (fiber, copper wire, radio, satellite)
-diff Transmission rates (bandwidth (bps)) = 10^3 Kbps to 10^6 Mbps to 10^9 Gbps
-router/switches = forward packets thru packet switches network
-an infrastructure that enables distributed services/apps (web email, games, file sharing)
nits: speed (bps) - kilo (K) = 10^3 = 1000; mega (m) = 10^6 = 1,000,000; Giga (G) = 10^9 = 1,000,000,000
pacity/size (bytes B) - kilo (K) = 2^10 = 1024; mega (m) = 2^20 = 1048576; Giga (G) = 2^30 = 1073741824
me (s) - milli (m) = 10^-3 = 0.001; micro (μ) = 10^-6 = 0.000001
bits = 1 Byte
access technologies:
homeaccess | enterprise access | wide area network
DSL | wifi | 3G, 4G, 5G, LTE
dialup modem | ethernet
wifi | Telephone line
cable internet | FTTH (fiber to the home)
Twisted copper wire | Twisted copper wire
approaches to moving data thru networks:
circuit switching | Packet switching
dedicated circuit paths per call used by all data | data sent in packets
end-end resources reserved for call b/w source to dest. | resources not reserved; on demand and may have to queue to access comm. link; forwards packets from routers to next on path from source to dest.
ex: Telephone networks | ex: internet
when network establishes a circuit connection, it reserves a constant R in networks link for duration of connection; R = Transmission rate | packet sent into network w/o reservation so if one link is congested bc packets are being Transmitted at the same time then packet will have to wait in buffer at sending side of Trans link and suffer delays
R is reserved for sender-receiver connection; sender can transfer data to receiver at guaranteed constant rate; guaranteed performance | internet will make constant effort to deliver packets in Timely manner but it cannot make any guarantees; no performance guaranty.
R of 100 Kbps must be reserved for all users; Trans capacity of a link is 1mbps so reserved for 10 users | Trans. rate: L/R; Packets of L bits onto link at R bps; if 35 users are allowed to send packets at a time then low probab. of more than 10 users to be active at the same time
segments sit idle if not used on call; no sharing of resources | link Trans capacity is shared on packet-by-packet basis among users
each user is allocated a time slot per frame | allows more than 3 times the number of users
4 sources of delay at each hop along the path from src. to dest.
① propagation ② Transmission ③ processing ④ queuing
d_hop = d_proc + d_queue + d_Trans + d_prop
Trans delay: Time to Transmit packet onto link = L/R
L (size of packet)/R (Trans speed); units in sec(s) or millisec(m)
-when last bit of packet is transmitted
prop. delay: time a packet takes to travel across a link = d/s
d (distance from one end to other)/s (prop. speed) = 2.5x10^8 m/s
-units in sec(s) or millisec(m); when first bit of packet will reach the other end of link; depends on physical medium (fiber, coaxial cable, air, etc)
end-end delay: Time it takes for entire packet to reach receiver

hosts = end sys ℓ en1 + server

what is internet? system of million connected computing devices providing communication links within diff media (fiber, copper wire, radio, satellite)

- diff Transmission rates/bandwidth(bps) $= 10^3$ kbps to 10^6 Mbps to 10^9 Gbps
- router/switches = forward packets thru packet switches network
- an infrastructure that enables distributed services/apps (web email, games, file sharing)

units: speed (bps) – kilo (K) $= 10^3 = 1000$

– mega (M) $= 10^6 = 1{,}000{,}000$

– Giga (G) $= 10^9 = 1{,}000{,}000{,}000$

capacity/size (Bytes B) – kilo (K) $= 2^{10} = 1024$

– mega (M) $= 2^{20} = 1048576$

– Giga (G) $= 2^{30} = 1073741824$

time (s) – milli (m) $= 10^{-3} = 0.001$

– micro (μ) $= 10^{-6} = 0.000001$

bits = 1 Byte

access technologies:

home access	enterprise access	wide area network
DSL	wifi	3G, 4G, 5G, LTE
dialup modem	ethernet	
wifi	Telephone line	
cable internet	FTTH (fiber To the home)	
Twisted copper wire	Twisted copper wire	

approaches to moving data thru networks:

circuit switching	Packet switching
dedicated circuit paths per call used by all data	data sent in packets
end-end resources reserved for call b/w source To dest.	resources not reserved; on demand and may have to queue To access comm. link; forwards packets from 1 router To next on path from source To dest.
ex: Telephone networks	ex: internet
when network establishes a circuit connection, it reserves a constant R in networks link for duration of connection R = Transmission rate	packet sent into network w/o reservation so if one link is congested bc packets are being Transmitted at the same Time then packet will have To wait in buffer at sending side of Trans link and suffer delays
R is reserved for sender-receiver connection; sender can Transfer data to receiver at guaranteed constant rate; guaranteed performance	internet will make constant effort To deliver packets in Timely manner but it cannot make any guarantees; no performance guaranty.
R of 100 Kbps must be reserved for all users; Trans capacity of a link is 1 mbps so reserved for 10 users	Trans. rate: L/R; packets of L bits onto link at R bps; if 35 users are allowed to send packets at a time, then low probab. of more than 10 users To be active at The same time
segments sit idle if not used on call; no sharing of resources	link Trans capacity is shared on packet-by-packet basis among users
each user is allocated a Time slot per frame	allows more than 3 Times the number of users

4 sources of delay at each hop along the path from src. To dest.

① propagation ② Transmission ③ processing ④ queuing

$$\boxed{d_{hop} = d_{proc} + d_{queue} + d_{Trans} + d_{prop}}$$

Trans delay: Time To Transmit packet onto link $= L/R$

L (size of packet) / R (Trans speed); units in sec(s) or millisec(ms)

- when last bit of packet is Transmitted

prop. delay: time a packet Takes To Travel across a link $= d/s$

d (distance from one end To other) / s (prop. speed) $= 2.5 \times 10^8$ m/s

- units in sec(s) or millisec(ms); when first bit of packet will reach the other end of link; depends on physical medium (fiber, coaxial cable, air, etc)

end-end delay: Time it Takes for entire packet To reach receiver $+ d_{prop} = L/R + d/s$ in seconds (adds $d_{proc} + d_{queue}$

proc delay: checks for bit errors;

queue delay: Time spent waiting for Transmission; depends on congestion

- units in micro(μ) or millisecs(m); packets can get lost if buffer is full
- if packet arrival rate (La) To link exceeds trans. rate capacity (R) then packets will queue; $La/R \sim 0$ = avg small queue delay; $La/R \to 1$ = avg large queue delay; $La/R > 1$ = avg delay infinite

Protocol: defines the format, order of messages sent + received among network entities, action Taken on msg. Trans + receipt.

five layer TCP/IP protocol stack: ① application layer, ② Transport, ③ network ④ link layer, ⑤ physical layer

seven layer OSI reference model: addition of ⑥ presentation + ⑦ session

app layer: used To send data over multiple end systems

- HTTP, FTP, SMTP (email), DNS; referred To as 'message'

Transport: host-host data Transfer

- TCP, UDP; referred To as 'segment'

network: routing of packets (datagrams) frm src. To dest.

- IP protocol, routing protocols; refer To as 'IP layer'

link: moves packets from one node To another in route

- reliable delivery; ethernet, wifi, cable access network DOCSIS protocol (not TCP); referred To as 'frames'

physical: moves individual bits within from from 1 node To next

- link dependent protocols, depends on physical medium

presentation: allows apps to interpret meaning of data;

- data compression, encryption, description

session: supports synchronization and delimiting of data exchanged; includes building checkpoint + recovery scheme

encapsulation: (process from app layer To link layer)

throughput: min{Rs, Rc, R/M}; rate at which hosts exchange bits

bottleneck: link on end-end path that constraints end-end thruput

- link with smallest capacity

Circuit Switching: physical path is obtained and reserved To a single connection b/w Two endpoints for duration of connection

Socket: 'endpoint' in connection for sending + receiving data b/w Two programs running on network; used w set of request or 'function calls'

Port: uniquely identify diff apps/processes running on single computer and enable To share a single physical connection To packet-switched network

DNS: domain naming system for computers, services, resources connected To Internet

Proxy: machine To satisfy client HTTP request w/o contacting origin server; nearby clients send their HTTP request To this machine

Trace route: comp. network diagnostic Tool for displaying route/path + measuring Transit delays of packets across an IP network

Pipelining: Sender is allowed To send multiple packets w/o waiting for ACK

if network sources send data at constant bit rate, Circuit Switching is more desirable because there is no statistical multiplexing gains To be had, by using circuit each connection will get constant amount of bandwidth

App Architecture: designed by app dev who decides how app is structured over end systems. Client-Server Architecture: server that services requests for client

Server: always on host, permanent IP Address; server forms for scaling

Client: comm. w server; may be intermittently connected; dynamic IP add. do not communicate directly w each other

Data Center: virtual server To keep up with client requests

P2P Architecture: min or no reliance on dedicated servers in data centers

- peers: laptop, desktops; peers communicate w/o passing thru server; cost effective
- not always on server; arbitrary end systems directly communicate
- peers intermittently connected and change IP address; highly scalable
- hard To manage

Hybrid of Client-Server + P2P: Napster: File Transfer P2P

File search centered at

- Peer register content at central server
- Peers query same central server To locate content

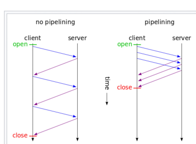
Processes: communication by sending/receiving messages across comp. network

To identify a process running on another host: IP address + port # ↳ 32 bit

Handshaking Protocol: server + client exchange control info before sending data To each other

TCP uses three-way handshake To create reliable connection; has full-duplex connection and guarantees reliable data Transfer over connection simultaneously and after server-client finish, conn. Terminates

Neither UDP and TCP provide encryption; TCP with (SSL) secure socket layer provides encryption, security, etc.



Pipelining: sender can send multiple packets without waiting for ACKs. cons.

- range of seq. #s must be increased
- sender + receiver may have To buffer more than one packet

Two approaches for pipeline error recovery:

Go-Back-N Protocol: link layer protocol that uses sliding window method for reliable + sequential delivery of data frames.

GBN can have performance issues, when window size + bandwidth delay are large, many packets can be in pipeline

A single packet error can cause GBN To reTransmit large number of packets

GBN can allow a max num of packets, it must wait one or more packet To be ACK before proceeding so if packet 2 is lost then rest after are out of order a dropped

Total amount of Time To get IP add: $RTT_1 + RTT_2 + \ldots RTT_n$

once IP add is known, RTT_0 elapses To set up TCP, another RTT_0 elapses To request and receive object: $2RTT_0 + RTT_1 + RTT_2 + \ldots RTT_n$

Non-persistent HTTP w. no parallel TCP (w. 8 small obj):

$2RTT_0 + RTT_1 + RTT_2 + \ldots RTT_n + 8 \times 2 \cdot RTT_0$

Persistent HTTP conn. w. pipelining: $RTT_1 + \ldots + RTT_n + 2RTT_0 + RTT_0$

default mode of HTTP.

Persistent HTTP conn. w/o pipelining: $RTT_1 + \ldots + RTT_n + 2RTT_0 + 8RTT_0$

w/o parallel connections