* chapter one:

* network is the system that connects 2 or more devices and allow data to move through them. (example: internet)
  ↳ interconnected ISPs

  (1) Software: application

  (2) hardware: router

* (1) LAN: 10m : 1/2 km (within faculty / university)
  ↳ any devices connected to each other

  (2) MAN: 1km : 10 km
  ↳ MAN + MAN ⇝ WAN

  (3) WAN: > 10 km

  (4) PAN: 1m : 10m (bluetooth)

* Router: (1) connect LANs together
  (2) can operate as firewall
  (3) communication server (سنترال)

* firewall: بيعمل check على الـ data ، هل الـ data ليها الصلاحيات انها تمرّ؟
  ↳ block & filter

Security {
  → authentication: verify who you are
  → authorization: decides if you have a permission to access a resorce
}

* Server: provides functionality for other devices, called "clients"
  → DHCP server: automatically provides and assigns IP addresses

* IPs/IDs: IDs detect attacks but does not stop them
  IPS detect attacks and stop them مراقبة سلوك البيانات

* Switch: connect pcs together to creat LANs
  connect other devices (IP phones / camera)

* Access point: allow other wifi devices to connect to a wired network
  ↳ can be standalone device or integral component of the router

* Types of topologies

→ Star: Adv: less expensive

Dis: if server goes down, everything goes down

→ Bus: Adv: easy installation

Dis: Difficulty in fault detection

→ Ring: Adv: easy installation (add or remove devices)

Dis: Data traffic issues, since all data is circulating in a ring

→ mesh: Adv: network congestion is minimal

Dis: expensive $(n^2 - 1)$ → # of interface cards

→ hierarchical: Adv: efficient

Dis: if server goes down, everything goes down

* cables are faster than DSL

---

* hosts - end system (ex: pc/ laptop)
* globle ISP → connects two countries
* regional ISP → connects within the country
* internet standards → (1) RFC (2) IETF
* internet service:
  - infastructure that provides service to applications (distributed apps)
  - provides programming interface to applications
* protocol: it's a format, order of msgs sent or received among network entities and actions taken on msg transmission
  ↳ rules for interaction

* Network Structure:

→ network edge: hosts & servers in data centers

→ access network: wired or wireless communication links

→ network core: interconnected routers

* Access network and physical media:

(1) Digital Subscriber line (DSL): [voice, data transmitted at different frequencies]

It uses a deticated line to center office

⟹ 1 Mbps < upload < 2.5 Mbps

⟹ 10 Mbps < download < 24 Mbps

(2) cable network: (fiber caldes)

* frequency division multiplexing: different channels transmitted in different frequency bands

* homes share access network to calde headed (lower actual rate)

⟹ upload = 2 Mbps download = 30 Mbps

(3) Enterprise access network (ethernet): [illegible]

* used in companies & universities

* ethernet switches have 100 Mbps access

* Servers have 1 Gbps or even 10 Gbps access

---

* packet transmission delay = time needed to transmit L-bits packet into link = L(bits) / R(bits/sec)

L(bits) → length

R(bits/sec) → transmission rate

physical media:

(1) guided media (wired)

(2) unguided media (wireless)

* twisted pair (tp) = two parallel copper wires
* coaxial cable: two concentric copper wires
* fiber optic cable: (1) high speed (2) Low error rate

Radio:

* Terrestrial microwave
* LAN (WiFi)
* WAN (cellular)
* Satellite

---

(1) circuit switching:

- depends on dedicated connection

⇒ 3 levels:

(A) circuit establishment

(B) Data transfer

(C) circuit disconnection

- Disadvantages:

(1) may be inefficient (as the channel capacity (بتفضل محجوزة طول ما في اتصال)

(2) delay prior signal transfer establishment

- Advantages:

Data is transmitted at a fixed rate with only propagation delay

propagation + process ← packet عكس ال

(2) packet switching:

- Data is transmitted in chunks, blocks called packets
- Before send the message is broken into packets
- packet consists of data + header
- node ⇒ router or switch
- Advantages:

(1) line is more efficient ⇒ many packets can share the same node to node link (no overloading)

(2) Different data routes for data exchanging.

- Disadvantages:

More processing time is needed at each node

---

* jitter ⇒ معيار بيقيس كفاءة الشبكة، لو المعيار كبير تبقى الشبكة وحشة

(3) packet and circuit نيه نوع تالت بيجمع بين

↳ data وده بيكون أفضل من حيث زمن نقل ال

- throughput = min {Rs, Rc}

* chapter two:

- no need to write software for network-core devices
- network-core devices do not run user application
- process is the program within a host
- within same host, two process communicate using inter-process communication (defined by OS), so there is no need for network
- processes in different hosts communicate by exchanging messages
- socket ⟵ layers transport ال و application ال بين الاتصال نقطة
- protocol definition: (1) type of message
  (2) message syntax
  (3) message semantics
  (4) rules for when & how process send & respond to
- open protocol ⟵ HTTP/SMTP
  proprietary protocol ⟵ Skype
- what transport service does application need?
  → data integrity → timing → throughput → security
- identifier has IP address (32 bit) & port number
  ↳ not enough as there may be many processes on same host
- port number of HTTP server: 80
  mail server: 25
  FTP: 21 [using TCP (control connection)
  20 [data connection]
  POP3 → 110
- TCP & UDP ⟵ Security مفيش ⟵ so we use SSL
  اسرع بكتير
- Web pages consists of objects
- www.facebook.com/profile/photos
  domain name path name
- HTTP uses TCP
- HTTP is stateless (server maintains no information about past client requests)

- non-persistant HTTP: one object sent over TCP connection
  persistant HTTP: multiple objects sent over TCP connection (one RTT)
- non-persistant HTTP response time = 2RTT + file transmission time
- request line: Get / post / Head RTT: from client → server ←
  \r : carriage return character
  \n : line feed character
  \r\n at the start of line indicates end of header lines
- uploading form input: post method / URL method
  - head: asks server to leave requested object out of response
  - put: upload file in entity body to path specified in the URL field
  - delete: delete file specified in the URL field
- There is no method in the response message (Status line)
- Status codes: (1st line) : 200 ok
  301 move permanently
  400 Bad request
  404 not found
  505 HTTP version not supported
- cookies: (1) header line of HTTP response
  (2) header line of HTTP request
  (3) file kept on user's host
  (4) back end database at web site
- cookies can be used for: (1) authorization
  (2) shopping carts
  (3) recommendations
- cookies permit sites to learn a lot about you.

- cache: satisfy client request without involving origin server but through proxy server.
- cache acts as a server for original requesting clients
- cache acts as a client to original server
- cache is useful as it reduces response time and traffic / Security ميزة في
- conditional GET: don't send object if cache has up to date cached version
- FTP uses TCP (as non presistant http) → control: (send information) → data
- connection control (port 20) => out of band
- FTP is state => save current directory & earlier authentication
- 331 username ok, password required

125 data connection already open; transfer starting

425 can't open data connection

452 Error writing file

- SMTP: protocol between mail servers (uses TCP) → (sending / receiving)
- 3 phases of transfer: (1) handshaking (greeting)
  (2) transfer of message
  (3) closure
- HTTP / FTP / SMTP use ASCII text (command) & Status code (response) → 7-bit
- user agent: g-mail / yahoo ... etc
- SMTP is persistant
- CRLF. CRLF is used by SMTP to determine end of message

HTTP	presistent / transfer files	SMTP
pull	TCP	push
ASCII		ASCII
each object in its own message		multiple objects sent in multipart msg
web server → web client		mail server → mail server

in band	persistent
client-server	

• pop3 & IMAP (pull)

↳ authorization & download (transaction)
↳ more features, manipulation of stored messages
↳ State across sessions

↳ (1) download & delete — ممكن اخذ ال header بس لو ال band width قليل

(2) download & keep

↳ Stateless across sessions

• host/domain name → IP address (32 bit) [to resolve names]

google.com has so many IP addresses (to equalize traffic)

• why no centralize DNS? UDP

→ Single point of failure → traffic volume → distant centralized database → maintenance

• 13 root name "servers" worldwide, 10 in the US

• Top level domain responsible for .com, .org, .net ... etc & .uk, .eg & .edu

• authoritative DNS: maintained by organization or service provider

• iterated query: contacted server replies with name of server to contact
↳ heavy load on local DNS

• recursive query: puts burden of name resolution on contacted name server
↳ heavy load on root DNS (memory cache)

• Local DNS → Root DNS → TLD DNS → authoritative DNS (اخر)

• DNS records: RR format: (name, value, type, ttl) → time to live (صلاحية)

type = A
• name is hostname
• value is IP address

type = CNAME
• name is alias
• value is conical name

type = NS
• name is domain
• value is hostname of authoritative name server

type = MX
• value is name of mailserver

- DNS message header identification consists of 16 bit
- 3way handshake (request, acknowledge, close)
- Disadvantages of non-presistent:
  (1) new connection must be established
  (2) TCP buffer must be allocated

- انا لما ابعت رسالة بيبعتها الـ mail server و بتتخزن في الـ MSG queue لحد ما الشخص الآخر يشوفها.
- سابتها الـ user agent بتاعة هيبعت استعادة عندو و يحطها في الـ mail server

- traket of torrent: • registration of peers
  • gives IP for peers

(1) rarest chunks
(2) highest rate
] when downloading from torrent

- hash table: change key into integers
- bit torrent: 256 kb chunk

Tracker: tracks peers participating in torrent

Torrent: groups of peers
exchange chunks of files

while downloading
↳ peers upload chunks to other chunks
↳ peers may come & go : churn

- once peer has entire file → leaves (selfish)
  → remain in torrent

*Revision:

- Dial-up modem over telephone line — 56 Kbps
  DSL over telephone line — home 24Mbps↓ 2.5Mbps↑
  cable to HFC — 42.8↓ 30.7↑
- 100 Mbps switched ethernet
  Wifi — enterprise
  3G or 4G — wide area network
- Bandwith = throughput (more general) = transmission rate
- HFC, in the downstream all packets emanate from a single source named (the head end)
- Ethernet can run over twisted pair copper wire or HFC
- most popular wirless Internet access technologies .. wifi & 3G/4G
- circuit switch guarantee certain amount of bandwidth
- FDM needs sophisticated analog hardware.
- ISPs reduce their payments to their provider ISP when they peer with each other.
- IXP earn money from the multiple ISPs connected to it.
- private network (google) : (1) more private (2) more control (3) save
- propagation delay & transmission & queuing (variable) & processing
- header in each packet includes the IP address of the destination
- traffic intensity = $\frac{\#\text{ of Packets/sec}}{\text{transmission rate}}$
- error control, flow control, segmentation & reassembly, multiplexing, connection setup
- logical addressing → network
- Bursty data better in packet switching
- end to end → seconds

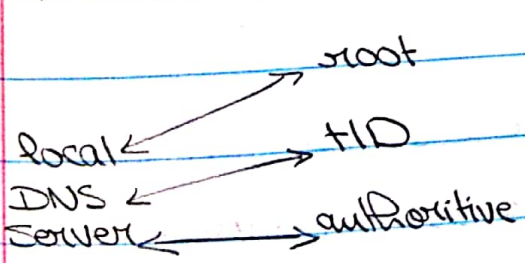
G → 10^9, M → 10^6, k → 10^3 M → 10^{-6} μ → 10^{-3}

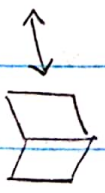
- HTTP
  - app layer protocol
  - port 80
  - TCP
  - stateless ⟶ cookies (sort will state)
  - client server
- Non-presistent
  - at most a object
  - parallel conection
- 1 RTT (initiating)
  - 1RTT / object

persistent (default)
- multiple objects
- 1 RTT (for all)

- DNS

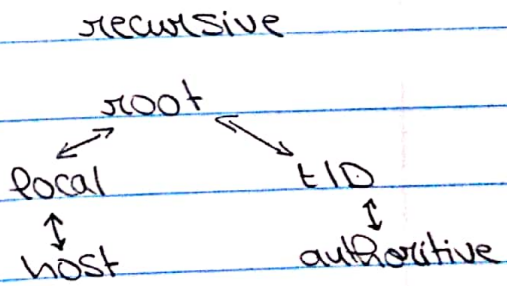
iterative





DNS server is queried & returns an answer without querying other DNS servers even if it can't provide definitive answer

recursive



DNS server is queried & subsequent DNS until definitive answer is returned

- HTTP
  - inband
  - 1 TCP connection for both control & data
  - Stateless

FTP
- out of band
- 2 parallel TCP connection both control & data
- State

29/9 TCP/IP (Stack)

Lab #1

	Layer	
قسمناهم layers علشان أعرف فين المشكلة (5)	Application layer	* end user (google chrome) * Session ID * HTTP / HTTPs / FTP / SMTP
(4)	Transport layer (Transmition protocol)	* Insure reliabity * port number * TCP / udp صورة rotated ولا لأ؟
Bridge - Router - modem (3)	network or internet layer	*اعرف اكلم ناس من network مختلفة * LAN (أنت) ——⊗—— LAN (مكان تاني) using IP address → بيتكرر [router] بيتغير (logical address)
Switch) (2)	→ data link D.L or network access layer	(bits) * flow control الاك بتتأكد أن الـ data صح * Ethernet / PPP لـ صورة نفسها وصلت مش محتاج نت * MAC : media access control → (مش بيتكرر) ~~> unique [switch] → physical address (local)
(Hub) (1)	physical layer	media (wifi - wires)

↓ Encapsulation:

(5) Application	data (01010101) Binary form		data الـ PDU
(4) Transport	application header	data	PDU ↓ packet data unit
(3) network x → x	transport header operating system → network card	data	TCP segment/socket UDP datagram
(2) Data link x → o o → x	network card → physical layer network header	data	packet
(1) physical	d.l header	data	Trailer — frame

↑ decapsulation

circuit switching

* Data transferred in a dedicated channel (same route)
* conn. oriented — conn. setup
  data transfer
  conn. close
* waste of resources idle → مستهلك
* more reliable
* pre-allocate use of transmission link
* reserving circuit (call)

packet switching

* Data is split into packets
  each packet travels independently (different route)
* connection less
* less reliable
  يرتبها قبل ما توصل الـ end-user
* allocate link use on demand
  عند الطلب

---

different peers can be client or server
على حسب هي عايزة ايه

client-server

* client intiates conn.
* Server waits for the connection
* ~ always on host
* ~ Service requests from many clients
* fixed, well known, IP address
* dynamic IP address → client
* clients can't comm. directly

peer to peer

* peer can comm. without passing through a server
* Self scalability network اكبر
  ↳ adding new services
* peers are intermittently connected
  ↳ متقطع
  thus complex management

---

protocols of transport layer

TCP

* connection oriented ↑
* reliable
* flow control ⇒ source doesn't overwhelm receiver عشان يبعت data
* error detection ↳ receiver بتحصل بسبب
* congestion control
  ↳ assignment ↳ network بسبب

UDP

* connection less
* unreliable (packets may be stopped)
"Best effort delivery" ⇒ doesn't guarantee delivery
* doesn't provide — flow control
  — congestion control
  ↳ live video & audio app
  games

20/10

Lab #4

That affects the quality → traffic intensity: β L/R

كثافة/ضغط

L → datasize or object size

R → transmission rate

β: avg. request rate → avg. packet arrival → كل مرة بكلم السيرفر (server)

Delta

- $\Delta = \frac{L}{R}$ (sec/min/hr)

  → avg. transmit of object time (transmission delay)

- Total delay or total avg. response time

(1) = LAN delay + Access delay + Internet delay

LAN delay → too small value, so most of the time we make it equal zero.

Internet delay → RTT (round trip time)

[X] delay

X: Access or LAN or internet

∴ Access delay

$= \Delta / [1 - \beta\Delta]$ → $\beta\Delta$: traffic intensity

$= \frac{L}{R} / [1 - \beta \frac{L}{R}]$ → هي دي اللي بتفرق بين ال delays

R → Access transmission rate

- utilization = x delay × 100

Access utilization = access delay × 100

- Total delay or total avg. response time

(2) = [hit rate × delay on cache] + [miss rate × delay on origin]

most of time equal 0 ↑

⟹ 1 - hit rate = miss rate && miss rate = 1 - hit rate

- Delay on origin = RTT + [Δ / 1 - miss rate × BΔ]

= RTT + [$\frac{L}{R}$ / 1 - miss rate × B $\frac{L}{R}$]

- proxy: (1) asra3

(2) wayn3sh tet3a3, 3andak copy mawgoda

- hit rate: tallot haga wgatly mn el proxy

miss rate: tallot haga w mal2thash fl proxy so, it request from server

- End to end delay: propagation delay + transmission delay

$= \frac{\text{distance}}{\text{speed}} + \frac{L}{R}$

+ queuing delay + processing delay

27/10

Lab #5

network engineer { Time ↓ capacity↑ Speed↑

- client-server distribution:

- server must send a copy for each peer "client".

- place an overhead on server and consume large amount of server bandwidth.

- $D_{C-S} = Max\left\{\frac{NF}{U_S}, \frac{F}{di}\right\}$ [UPLOAD]

F → file size

$\frac{F}{di}$ → download time of client = download rate of peer (min)

U_S → upload rate of server

N → number of peers or file copies

upload time of server increase linearly in N [Max]

أقوى جزء في السلسلة = أضعف جزء فيها

- P2P distribution:

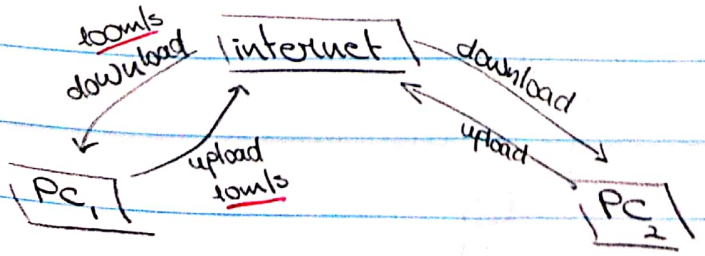
- Any peer can redistribute part of file it has received to other peers. _delay a2al

- $D_{P2P} = Max\left\{\frac{F}{US}, \frac{F}{di}, \frac{NF}{U_S + \sum_{i=1}^{i=N} U_N}\right\}$

di → min

U_N → upload rate of peers

⇒ wayn f3sh nafs el link yemel upload and download



* Question (9) Sheet 9:

Given:

bytes → 1024

bit → 100

file size: 15 gbits → 15 x 1000 mbits

upload rate of server → 30 mbits

download rate of users → 2 mbits

Number of users → 10, 100, 1000

upload rate of users → 300 kbps, 700 kbps, 2 Mbps

D_{c-s} ← مش بستفاد منها في اد

$\frac{300}{1000}$ Mbits, $\frac{700}{1000}$ Mbits

- $D_{c-s} = Max\left\{\frac{Nf}{U_s}, \frac{f}{di}\right\}$

$\frac{Nf}{U_s} = \frac{10 \times 15000}{30} = 5000$ sec

$\frac{f}{di} = \frac{15000}{2} = 7500$ sec → مش بيتغير

- $D_{P2P} = Max\left\{\frac{f}{U_s}, \frac{f}{di}, \frac{Nf}{U_s + \sum_{n=1}^{N} U_n}\right\}$ $\frac{f}{U_s} = \frac{15000}{30} = 500$ sec

$\frac{f}{di} = \frac{15000}{2} = 7500$ sec

$\frac{Nf}{U_s + \sum_{n=1}^{N} U_n} = \frac{15000 \times 100}{30 + 100(0.3)}$

D_{c-s}

U \ N	10	100	1000
0.3	7500	50,000	500,000
0.7	~	~	~
2	~	~	~

D_{P2P}

U \ N	10	100	1000
0.3	7500		
0.7			
2			

← مش واردة لان $U_s > 2$

3/11

Lab #6 Server-client socket code

* TCP

Server.Java Run 1	client.Java Run 2
ServerSocket(port);	Socket client (IP, port);
Establish connection	
Input Stream (read UTF)	Input stream receive; ← receive data server
Data ⇒ output Stream (write UTF)	rec. read UTF;
output ↲ String يحولها	output stream send; → Send data
input ↲ bytes يحولها	Send. write UTF;
close ⇒ close();	client. close()
	receive. close()
	close اي حاجة عرفتها ب

* UDP — الـ data بتتبعت أي وقت ومفيش Ack عليها

Server.Java Run 1	client.Java Run 2
Establish connection: Datagram socket server(port);	Datagram socket client(); ** IP, port on the packet not here ↑

بيشتغل بس: { byte rec[] = new byte[] .receive(); client من السيرفر لـ
byte send[] = new byte[] .Send();

Data:
Send = = input.get byte(); — تحويل لـ bytes
String output = rec.get data();
Datagram psend (IP, port, Send[], size) client.Send();
Datagram prec (rec[], recSize);
client.receive();

close → .close()	.close()

* input/output stream — بيعمل زي buffer بيخزن فيه الـ data