DNS แปลงหมายเลขเครื่องส่ง

Socket programmings (โปรแกรมผ่าน Network)

↳ Cilent - Sever

→ IP address เป็น network layer

→ Forwording and Routing หาเส้นทางที่ดีที่สุดไป sever

→ Packet Sniffer = เลือก-เอา Packet คนอื่นมาแปล

คะแนน Midterm
- → Exam 30
  - 20 ทฤษฎี
  - 10 Lab
- → Project 10
  - TCP/UDP Socket program
  - Bonus Basic Chat app
- → Lab and quiz 10

→ Network edge : hosts

→ Network core : packet, internet structure

Internet = Billions of connected computing devices
- host : end systems
- รัน network apps ของ internet

Packet switches = ตัวรับข้อมูล
- routers > switches > hub

⊗ ⊠

Communication links
- fiber, copper, radio, satellite
- transmisson rate : bandwidth (ท่อเท่าไหร่)

Networks
- Collection of devices, routers, links

① → Internet คือ Network of Network
→ IOT คือ devices ที่ใช้ Internet

→ protocols คือ รูปแบบสื่อสารกัน ที่มีกระบวนการ ซึ่ง
ต้อง fix รูปแบบให้เข้าใจกัน

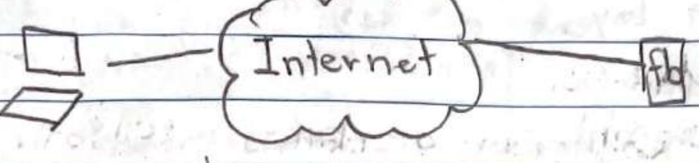
→ Internet standards
RFC : Request for Comments
IETF : Internet Engineering

② Internet คือ → Infra structure (โครงสร้างภายใน)
ที่ จัดหาบริการ ให้แก่ app
→ programming Interface

* Mac Address เลขฐาน 16 ตามรหัสรู้ Devices Internet
* เครื่อง คอมพิวเตอร์ สามารถ ต่อได้หลาย Network
มักเลือกต่อ Banwidth ที่มากกว่า (NIC)

192.168. หรือ 10. เป็น private IP

Config



DHCP จะ set ทั้งหมดนี้ให้
→ ต้องมีหมายเลขเครื่อง subret (วง Network) ไว้ต่อเน็ต
→ IP Default Gateway ของ router ไว้ส่งออกภายนอก
→ ต้อง set DNS เพื่อ map IP

What Transport service app need?

1.) data integrity
2.) timing
3.) throughput ต้องมีการไหลของข้อมูล ห้ามเสียหาย
4.) Security

Network App คือ ส่งข้อมูลผ่าน Network หากัน
2 Architecture
- Client Sever
- Peer to Peer

Internet = Infrastructure จัดหา service ให้ App
↓ = จัดหา programming Interface เพื่อควบคุมให้ Ap
package switching

OSI Model ออกโดย ISO

↳ Open System Inter connection

งาน สามารถแบ่งให้ต่างคนต่างทำได้ โดยมีความ สัมพันธ์กัน

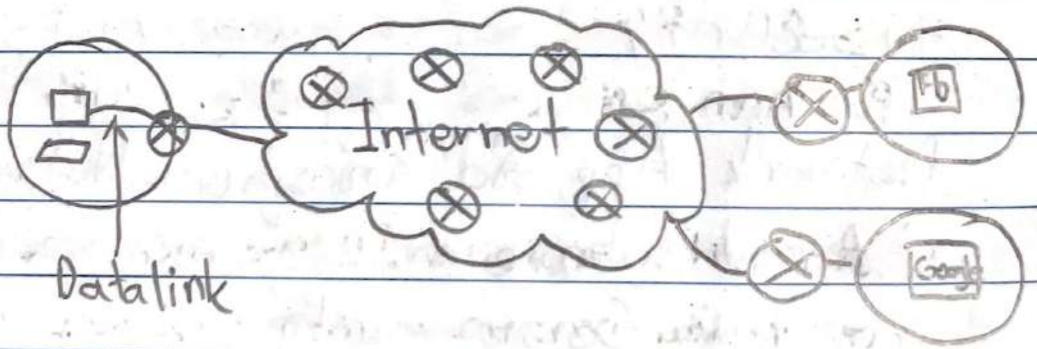
มี 7 layers

TLS → TCP
Quick → UDP

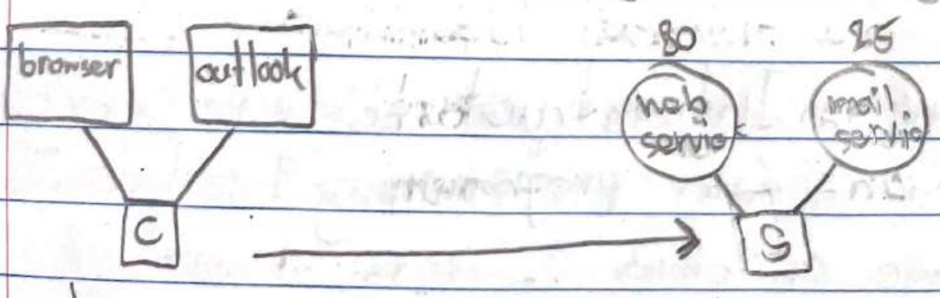
บน			
↓	App	80 443 53 22 25 http, https, DNS, SSH, SMTP (tcp)	7
	presentation → แปล/ถอดรหัสให้ App เข้าใจ		6
ล่าง	session → รอบการทำงาน จดหมาย 1, 2, ...		5.
	Transport → ระบุตัวผู้รับ มาครบไหม	tcp, udp	4
	Network → จัดกระจายข้อมูล	IP	3
	Data Link → กระบวนการตรวจสอบข้อมูล (Erorr Detect)		2
	physical → สาย LAN, Wi-Fi, USB, Fiber		1

* คำสั่งดู Hop → tracert www.facebook.com

Data Link* Protocal มีสาย คือ Ethernet ไร้สาย คือ Wi-Fi (802.11ax)



* หน้าที่ Network คือ routing → Forwarding → IP addressing



* transport ดูจาก Port No. เพื่อให้รู้ว่าเข้าที่ไหน

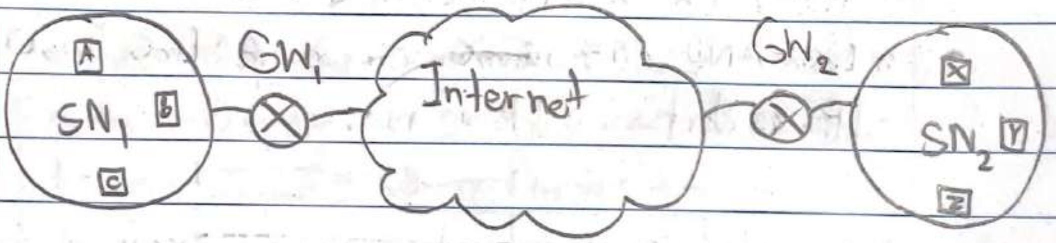
Web Port : 80 , Mail Port เช่น SMTP 25

16 bit $= 2^{16} - 1$

KMITL Public IP
161.246

* Port No. เพื่อใบ้ว่า client ขอ services ไหนของ Sver
* Client Port มีมากสุดถึง 16 bit คือ $2^{16}-1$ Port

4 สิ่งที่ต้อง set เพื่อต่อ Internet
1.) IP addressing
2.) subnet mask
3.) Gateway
4.) DNS



$$\left.\begin{array}{l} IP_A \text{ AND } SN_1 \\ IP_B \text{ AND } SN_1 \\ IP_C \text{ AND } SN_1 \end{array}\right\} NW_1 \quad \left|\quad \left.\begin{array}{l} IP_X \text{ AND } SN_2 \\ IP_Y \text{ AND } SN_2 \\ IP_Z \text{ AND } SN_2 \end{array}\right\} NW_2\right.$$

* วิ่งไปหา DNS เพื่อถามหา IP ของ www. นั้นก่อน
* Sever จะ Run ตลอดเวลา แต่จะไม่ส่งข้อมูลให้ Client ก่อน

▸ http เป็น protocal ไว้คุยกันระหว่าง Browser กับ Sever

req — Verb หรือ Method — post / get
— version

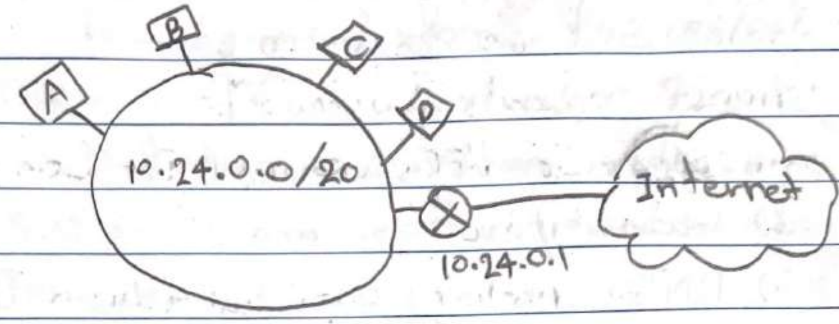
Universal c/s — Sever ไม่ส่งข้อมูลให้ Client ก่อน (client Initial ก่อน)
— Always ON

AND
^
A × B
∩

IP address and Subnet mask

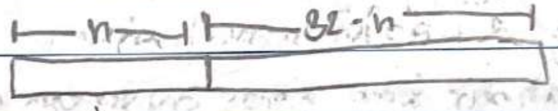
10.24.13.174 225.225.240.0

$\underbrace{\quad}_{10.24.0.0}$ Network Address



* x AND 255 = x , x AND 0 = 0

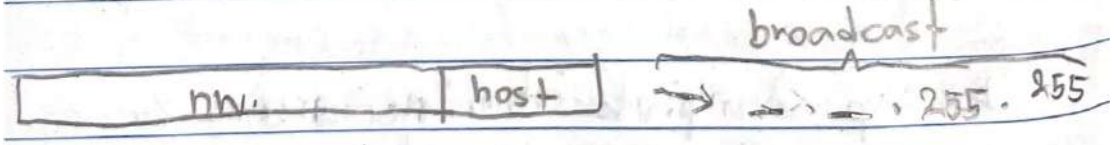
IP 32 bit



↳ นั่นคือ network bit ต้องเหมือนกัน
(subnet mask)

ค่าที่ reserve ไว้คือ 0 กับค่าสุดท้าย นั่นคือ 10.24.255.255
(10.24.0.0)

http : Network Architect ⌐ C/S Model
P2P Model



161.246.4.0 /16

255.255.0.0 subnetmast

161.246.0.0 nw.

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Web App Protocal

- ประเภทข้อความ request, respone
- syntax ข้อความ
- semantics ความหมายของข้อความ
- กฎเกณฑ์ ถามอะไร ต้องตอบอะไร
- open protocals RFCs (Request For Comment)
  ทุกคนสามารถไปอ่าน standard ได้
- Proprietary protocal - protocal ที่เป็นความลับ Zoom, Skype

คุณสมบัติที่ต้องการ

- data integrity (UDP ไม่มี integrity)
- timing ส่งแล้วได้รับภายในทันที
- throughput มีการไหลของข้อมูลตลอด (ภาพออกมาต่อเนื่อง)
  "elastic apps" ลดคุณภาพอัตโนมัติ
- sercurity (ส่วนมากเป็น Add ON)

* web server ประกอบด้วย html file หลาย file
* เนื่องจาก html สามารถเชื่อมไป server อื่นได้ เลยเป็น web

∴ web server เป็นที่เก็บ html file

→ object เป็นไปได้ HTML file, JPEG image, JAVA applet

$$\underbrace{\text{Www.someschool.edu/}}_{\text{host}}\underbrace{\text{someDept/pic.gif}}_{\text{path name}}$$

URL → Uniform Locater

HTTp → hypertext transfer protocal

HTTP 3.0 ⌈ TCP
⌊ UDP port 80

* ก่อนจะส่ง HTTP requst ต้อง TCP connect ก่อน
* stateless คือ sever ไม่จำว่าใคร เคยเข้ามาใช้งานแล้วบ้

(HTTP Requst) request, respone

Request ⌈ ASCII (human-readable format)
⌊ Requst Line / Header Line / empty line (CR LF)

* keep-alive → persistance ยังมีใช้อยู่
* entity body ใส่เฉพาะ Method Post

(HTTP Respone)

- status line : protocal ver. space status code space
- headr line มีหรือไม่มีก็ได้

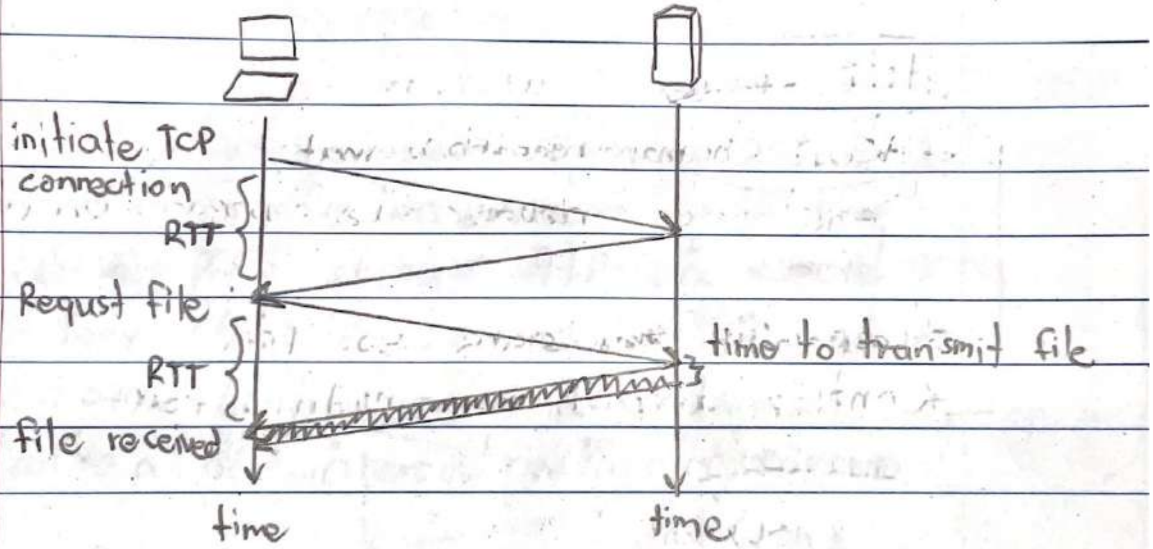
กลุ่ม status code
- 1xx (Information)
- 2xx (successful)
- 3xx (redirection)
- 4xx (Client Error)
- 5xx (server Error)

505 HTTP Not support

Persistent

→ Non
- 1. TCP connection opened
- 2. at most one object sent over TCP connection
- 3) TCP connection closet

*1.1 เป็นค่าเริ่มต้น→ Persistent HTTP เปิด TCP รอบเดียว ส่งหลาย object



*RTT เวลาที่ packet เดินทางจาก client ไป Server แล้วกลับมา

Cookies ทำให้ Sever เป็น stateful

↳ จำ state ได้ ไม่ต้อง Authen ซ้ำ ใส่ของลงตะกร้าได้เลย

สามารถดูแล cookie ได้จาก
- HTTP respone แรก
- HTTP requst หลังจาก respone แรก
- web Database
- Client Database

* ทั่วไปเป็น first Party cookie

google invent quic บน udp
Ladd บน Layer Application

Name servers เก็บ หมายเลข server

nslookup -type=ns

w.x.y.z

ชื่อ host

organize
- com
- edu
- gov
- mil

Web caches อยู่ในเครื่อง Web cache จะได้ไม่ต้อง
ตอบหลายรอบไม่เกี่ยวข้องกับ origin server

* broner เก็บ HTTP requests ไว้ที่ caches

* ปัญหาคือ web cache ต้อง update real time

Conditional GET → If-modified-since : <date>
ตรวจสอบ ว่า วันที่ แก้ไข ตรงกันหรือไม่ จะได้ไม่ต้องส่งใหม่

(HOL)
Head of Line Blocking รอ Obj ใหญ่ ต้องโหลดเสร็จก่อน
HTTP/1.1 ถึงจะโหลด Obj ต่อไปได้

HTTP/2 รับ order แล้ว แบ่ง obj. ย่อยๆ เป็น frames

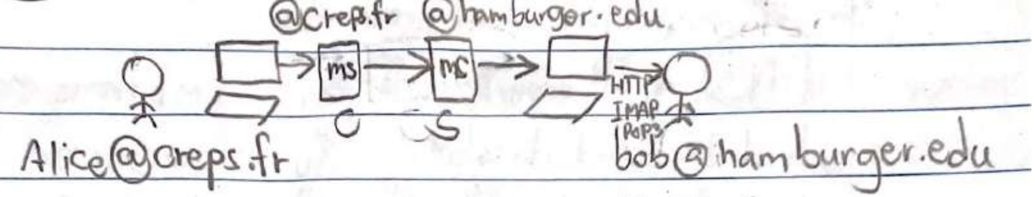
HTTP/2
- ไม่มี sercurity เรียกว่า vanilla TCP connection
- recovery จาก packet loss

HTTP/3 - เพิ่ม sercurity โดยเพิ่ม quic เข้าไปเพื่อ
ทำงานบน UDP

SMTP 25   HTTPs 443

SMTP mail server

คนส่งจะเป็น client, คนรับจะเป็น sever



Outlook IPhone Mail Client

องค์ประกอบ 1) User gaents (โปรแกรมเขียน แก้ไข อ่านEmail)

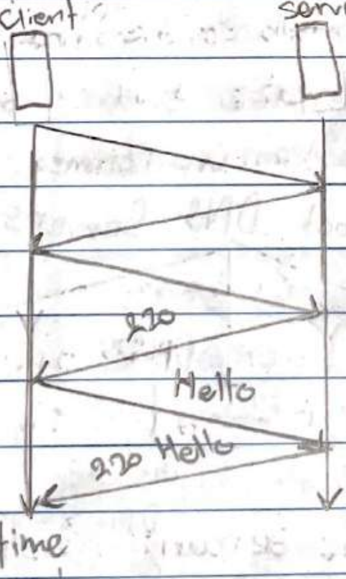
2) mail server

3.) simple mail trasfer protocol : SMTP

mail server — mailbox กล่องรับข้อความ อยู่บน mail server

message queue คิวรอส่ง email

SMTP เป็น protocol ใช้ส่งระหว่าง e-mail server



HTTP

POP3 * Status อยู่บนฝั่ง server ทั้งหมด

* IMAP ก็อป mail จาก mail box มาให้อ่าน

HTTP : client pull

SMTP : client push ส่งของไปยัง

ASCII, persitance

To ต้องการคำตอบ , CC ต้องการให้ทราบ

Bcc ให้คนอื่นไม่รู้ว่าเราส่งด้วย

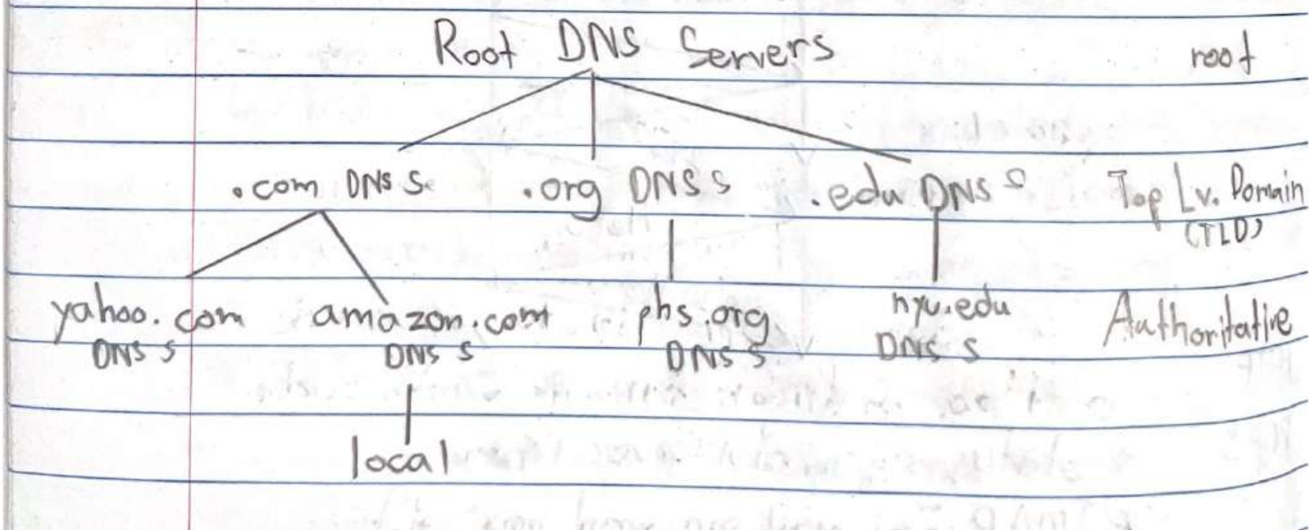
Syntax - header lines → To: , From: , subject

DNS รู้ IP ขอ host name , รู้ host name ขอ IP

- distributed database เก็บ name server หาคำตอบจากเครื่องอื่น ที่แบ่งกันเก็บข้อมูล
- host aliasing ขอชื่อเล่นของหลายเครื่อง
- load distributing หลาย IP เข้ามาใน ชื่อเดียวกัน
- mail server aliasing

→ ทำไมไว้ในเครื่องเดียว

- พังแล้ว เจ๊งหมด
- มีคนเข้าช้า เพราะทุกคนเข้าที่เดียวหมด
- ระยะทาง ไปเครื่อง กลางไกล
- mantainace



* ICANN จัดการ root DNS domain
* ทุกๆ country domains มีแค่ 2 ตัวเท่านั้น .th .jp .us

thainic → co.th

63050180@k

Local ทำงานหนัก

* iterated query ถามทุกชั้นของ server เริ่มจาก
root → TLD → Authoritative

core Network ทำงานหนัก

recursive query ถามแค่ root ครั้งเดียว แล้ว server
จะไปถามกันเอง แล้วตอบกลับขึ้นไป

Chching DNS Non-auhor

- เก็บ entries timeout (TTL)
- หมดเวลา แล้วต้องลบทิ้งไป

www.kmitl.ac.th

DNS records resource records (RR)

format (name, value, type, ttl)

type: A	type = CNAME	type = NS
name = host nane	name = alise name ชื่อเล่น	name = domain kmitl.ac.th
value = IP	value = cononical name ชื่อจริง	value = host nane
		ของ Authoritative

type = MX
value = SMTP mail

name	value	type	ttl
CS18@co.th	dn1.cs18.co.th	NS	xxx
dn1.cs18.co.th	161.246.4.1	A	yyy

DNS query, reply ใช้ format เดียวกัน
ดู flag ว่าเป็นประเภท query หรือ reply

* ต้อง

ขั้นตอนการตั้ง DNS ไปหา TLD ที่ตรง
→ มีชื่อ Domain, IP ของ Authoritative ลงทะเบียน
→ ใส่ NS, A ลงใน TLD ใน Database
→ ตั้งเครื่อง DNS ของตัวเอง ให้ IP ตรง

type record IPv4 เป็นการ mapping IP กับ host name

ยิงระดับ TB ชื่อ mirai

nslookup hostname DNS-Server

DNS security

→DDoS attack ถล่ม DNS จนทำให้ Overload

→Spoofing attack ตั้ง DNS ปลอม แล้วส่ง record ปลอมไป

nslookup เพื่อดู IP ของ Host-name

*มี IP เดียวมักใช้ cloud service

Socket เลือกให้ถูกว่าใช้ tcp/udp

└→ ยัด app ลง socket → เหมือนประตูสำหรับ App ไป transport

UDP : Unreliable datagram

TCP : reliable, byte stream-oriented

ไม่มี handshaking ก่อนส่ง

P2P ทุกคนแชร์ resource (No always-on)

- เครื่องเป็นได้ทั้ง client และ server
- เพิ่มขนาด Network เองได้
- complex management

$\frac{F}{d_{min}}$

$\frac{nF}{u_s} = t_s$

* ข้อดี ถ้าจัดการ complex management ได้

c/s ความเร็วในการส่งขึ้นอยู่กับ ปริมาณ client

แต่ P2P Client ช่วยกันอัพโหลด ยิ่ง client มากยิ่งเร็ว

→ File นั้นแบ่งเป็น chucnk 256Kb

→torrent คือ กลุ่มของ peers ที่แลกเปลี่ยน chunks ของ File

→คนใหม่เข้ามาจะ register ตัวเองเป็น tracker

* ดัน top 4 ที่ bandwidth มากที่สุด ทุกๆ 10 วินาที

* เลือก 4 คนแรกที่ส่งข้อมูลเยอะที่สุด ถ้าไม่ให้เลยจะถูก kick
tit-for-tat (ตาต่อตา ฟันต่อฟัน)

* ทุก 30 วินาที จะสุ่มเลือกคนที่ไม่ใช่ top 4 ถ้าให้เยอะพอ
เราจะกลายเป็น top 4

Ex domain name ที่ใช้คือไร www.google.com

1. เลือก 1 host จาก Ip address : 172.217.174.164
2. ns look up จาก domain name
3. ns look up จาก mail server ชื่อไร

- ns1.google.com, IP: 216.239.32.10
- mx203.inbound-mx.org, IP: 164.90.197.143

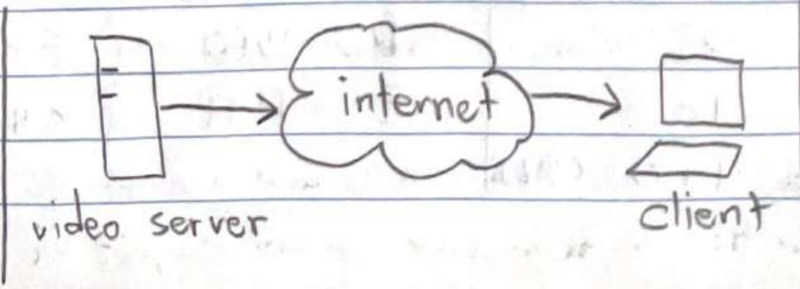
mx203.inbound-nx.net, IP: 147.182.189.184

CNDs

→ Multimedia : video → sequence of images
display in constance time

- จำนวน bit ใน 1วินาที 57,600,000 bits
- ลดจำนวน bit โดยการส่งรูปที่เห็นความแตกต่างจริงๆ
  - spatial เก็บสีเดียวที่มี pixel เหมือนกัน } coding
  - temporal ดูความแตกต่างระหว่าง 2 ภาพ
- CBR (constant bit rate) คือ 1วินาที encode อัตราเท่าเดิม
- VBR (variable BR) คือ encode เปลี่ยนตาม spatial, temporal

-network
ไม่บริสุทธิ์
มี data loss



-bandwidth
เปลี่ยนตาม
สถานที่,
Network

Streaming stored video

1. วิดีโอส่งเป็น chunks (เช่น 30 frames/sec)
2.) วิดีโอถูกส่งแล้ว แล้วมี network delay
3.) วิดีโอได้รับแล้ว cilent เล่นได้ (30 frames/sec)

* บางครั้ง มี network delay ทำให้เกิด buffed video

→ google พัฒนา DASH เป็น TCP
server แบ่ง chunks ของ video หลาย group
client ดูว่า server ส่งด้วย bandwidth เท่าไร
แล้ว cilent รับได้เท่าใด แล้ว sever ส่งตามจำนวนนั้น
ตาม encoding rate ที่เหมาะสม

→ Content distribution network
- option 1 มี server ใหญ่ศูนย์กลางเดียว
- option 2 มี server กระจายอยู่ทุกประเทศ เก็บหลายที่

$0000101\boxed{0}$
$1111111\boxed{0}$

a. 192.168.10.11	c. 510
b. 192.168.11.254	d. 192.168.10.0
	e. 192.168.11.255

→ Class Binary

/8	A	0	D	1110	First Byte
/16	B	10	E	1111	(8 bit แรก)
/24	C	110 (192)			

ยิง 1-n

→ การส่งแบบ boadcast คือส่งข้อมูลให้ทุกเครื่องใน network

→ multicast จัด IP เป็นกลุ่มเพื่อส่งข้อมูล

→ เหลือ 7 bit จาก 32

Network Address 10.1.1.0/25

1st Host 10.1.1.1

Last Host 10.1.1.126

Broatcast Host 10.1.1.127

จำนวน Host 2^7-2 = 126 host

↳ จำนวน bit ที่เหลือ

↳ NW กับ Broatcast

* ถ้ามี router ต้องเก็บ IP ไว้ 1 เบอร์ นั่นคือ gateway

subnet คือ Network ย่อย ใน class เดียวกัน

* router ต่อกับ router จะนับเป็น 1 network เหมือนกัน

CIDR : Classless InterDomain Routing

- subnet portion ของ address
- a.b.c.d/x , x คือ subnet portion bits

bit ที่ต่อจาก /20

000	0
001	1
010	2
011	3
⋮	
111	7

subneting แบ่ง host ที่เหลือเป็น network ย่อยๆ

เช่น ปัจจุบันเป็น /20 → /21 ได้ 2 network

→ /22 ได้ 4 network

→ /23 ได้ 8 network

* ถ้าซับเป็น /23 จะเหลือ host ไว้ใช้ 2^9 host ในแต่ละ network ทั้ง 8 network

ICANN จัดการ IP address ครอบ root อีกที

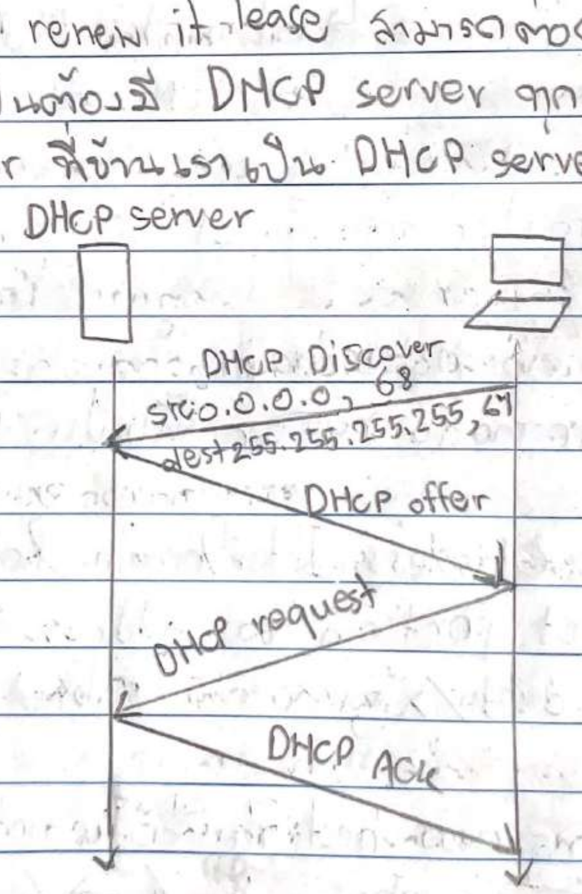
NAT ช่วยกระจาย IPv4 จาก NW address ไป NW ย่อย

DHCP server ใช้ DHCP protocal

- เหมือ set IP GW DNS subnet ให้ USER
- ด้วย concept แบ่งๆ กันใช้ นั่นคือ มี เวลาให้ใช้
  - allows reuse of address ไม่ใช้ ไปให้คนอื่น
  - can renew it lease สามารถต่ออายุ เองได้

* ไม่จำเป็นต้องมี DHCP server ทุก Network

* router ที่บ้านเรา เป็น DHCP server ด้วย



* เป็น UDP ทำงานใน App Layer

*** ข้อสอบ ให้เบอร์ Host x.y.z.a/xx พร้อม subnet

nw add
bc add
IP

UDP ไม่ลงทะเบียน

TCP ส่งแบบลงทะเบียน

Transport layer — TCP / UDP

- Logical มองการการส่งระหว่าง App เป็น Black Block
- transport แบ่งเป็น segment ไปยัง network Layer

* Network คือ จดหมายถึงบ้านเลยๆ จากไปรษณี
transport คือ ป้าที่เอาจดหมายให้เด็กแต่ละคน

Th	app.msg

Transport header + App = Segment
↳ เพื่อเข้า socket ไหน / Length เท่าใด

3-way handshake

TCP : Transmission Control Protocal
- มีความน่าเชื่อถือ ส่งตามลำดับ
- มีกระบวนการจัดการ data loss (congestion control)
- flow control (จัดการตอบรอด ของข้อมูล
- connection setup

UDP : User Datagram Protocal
- ไม่น่าเชื่อถือ unorder
- best-effort ส่งอะไรก็เหมือนเดิม (ยัดๆ เข้าไป)

* ทั้งสอง ไม่การันตี deley และ bandwidth

Transport รวมหลายๆ socket เป็นเส้นเดียว เรียกว่า “Multiplexing” โดยแปะ transport header เอาไว้

16 bits | 16 bits

header มี 32 bit แบ่งเป็น src port | dest port

server มี 1 socket — Conectionless demultiplexing คือ UDP socket

- ต้องรู้ IP กับ dest port
- เวลาข้อมูลวิ่งเข้ามา เป็น Service App ไหน ผ่าน port
- ถ้าข้อมูลมาจากหลายเครื่อง แต่ dest port เดียวกัน ก็มาที่เดียวกัน

server มี socket ตาม จน. client — Connection-oriented คือ TCP socket

- ต้องรู้ src IP, src port, dest IP, dest port
- แยกเครื่อง แต่ละเครื่องได้ (ทุก connection มี socket ส่ง-รับ)
- ไม่มีการ share data (ส่งของใครของมัน)

Length, check sum

UPD ง่ายที่สุดแล้วเท่านี้

- segment หายได้, ไม่ส่งเป็นลำดับ
- no handshaking แต่ละ segment เป็นอิสระต่อกัน

ข้อดี - เร็ว มีแค่ผู้ส่ง-ผู้รับ ไม่เสีย overhead, header มีข้อมูลน้อย

* ต้องเพิ่ม cogestion กับ reriable บน App Layer

- checksum - บางทีการส่ง สัญญาณอาจมีความผิด เช่น flipped bits
  - จึงมีกระบวนการตรวจ bits ด้วย checksum
  - ใช้วิธี one's complement sum

rdt-send() → udt_send → rdt_rcv()→deliver_data()

Reliable Tranfer ทำให้ unreliable เป็น reliable
L ใช้ FSM ในการกำหนด state ของ Network

rdt 3.0 รออย่างมีเหตุผล < เหมือนได้รับ / เหมือนไม่ได้รับ
ใช้ timeout ไม่การแก้ package loos แต่อาจจะ
เจอ dupicate เพราะ network delay
*แก้ไขโดยใช้ timeout เมื่อ pkt หรือ ack ส่งไม่ถึง

(คำนวณ)
Usender: utilization - เวลาที่เสียไปตอนส่ง

Pipeling เพิ่มให้ข้อมูลวิ่งในท่อพร้อมกัน โดยไม่รอ ACK
(ส่งเป็น series หรือ ส่งเป็น windows)
window size N
เขียว-ack แล้ว, เหลือง-ยังไม่ ack, ฟ้า-ยังไม่ส่ง, ขาว-ยังไม่ได้ใช้
cumulative ack : ได้รับ ack ตัวเลขมากไม่สน loss เก่า
(go-back-N) ให้ resend ใหม่ทั้งหมดตั้งแต่ loss

ส่ง ack1 ต้องการ pkt2

Go-back-N ถ้ารับ pkt0 แล้ว รอ pkt1 ถ้าไม่ได้ pkt
ตัวถัดไป จะ retransmit ทั้ง windows
rcv pkt0, deliver, send ack0

Selective repeat ถ้ายังไม่ได้รับ pkt ที่รอ จะสร้าง buffer
เก็บ pkt ที่รับ ack แล้ว (เป็น queue)
* จะไม่ส่งซ้ำ เพราะ เมื่อได้รับ pkt ที่รอแล้ว ดึง pkt ใน
buffer แล้ว ปล่อย record ใน sender

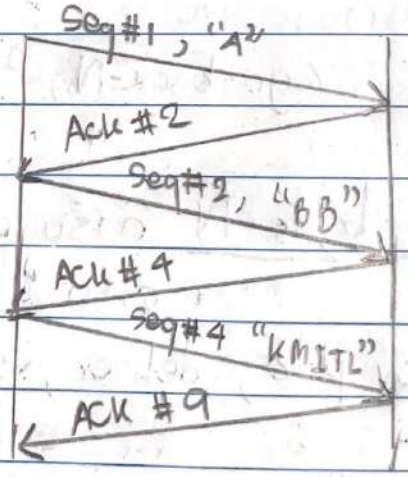
- reliable transfer concept (ประกอบอยู่ใน TCP,
  - ✓ → Seq no
  - ✓ → Ack
  - → Time out
  - → Pipelining
  - → check sum

ส่งข้อมูลเป็น Bytestream TCP

- point to point → one sender, one reciver
- reliable, inorder byte stream
- full duplex → ส่ง 2 ทาง, MSS
- มี Ack ทับซ้อนกัน (เป็น Go-back-N ,
- pipelining → set window size
- connection oriented → 3 way hand sake
- flow control

→ TCP segment

- header อย่างน้อย 40 byte ประกอบด้วย Src port, dest port, seq No., ack No., Length, rcv window, checksum



เป็น Ack ว่า รับมาถึง Byte ที่ +1 เพราะคาดหวังเอา byte ถัดไป

TCP fast retransmit ถ้าได้รับ Ack เดิม 4 ครั้ง มั่นใจได้ว่า pkt Loss แน่นอน retransmit ทันทีไม่รอ time out

TCP flow control เพื่อไม่ให้ข้อมูลกระฉอก

3 way handshake : Listen → SYN → ESTAB

Closing Fin bit = 1

127.0.01 Local Host

แนวข้อสอบ

- IP address
- TCP exchage msg
- ถ้า CHCP เอ็ง ให้ ตั้ง config IP
- ถ้าเปิดร้านกาแฟ กว่าจะเปิด Fb ได้ มี protocal อะไรบ้าง เกี่ยว ข้องกันอย่างไร