

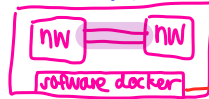
Networking overview

Docker's networking plug ได้โดยใช้ driver เหล่านี้

① bridge (bridge network)

เป็น default, data link layer

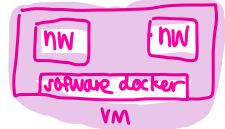
ใช้กับ container ที่ run บน docker daemon พิจารณาจาก MAC เดียวกัน



docker daemon

② host

container จะไม่เบรกออกจาก host ใช้ network host โดยตรง
เหมาะกับการที่จัดการกับ port บน host



③ overlay

ใช้เพื่อ Docker daemon หลายตัว
เข้าด้วยกัน และเปิด service ให้ communicate ได้

คล้าย ingress เมื่ออยู่บน host docker อื่น

④ macvlan

สามารถกำหนด MAC Address ให้กับ container เพื่อเชื่อมกับ physical network
Docker daemon กำหนดเส้นทางการรับ-ส่งข้อมูลไปยัง container ตามที่เหมือน container บน host

⑤ none

ปิดการใช้งาน network



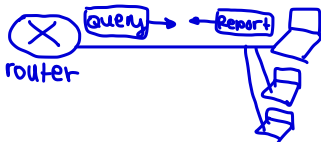
Internet group management protocol IGMP

(multicast routing) in local area nw

Query message ส่งจาก router ไป host

เพื่อเก็บข้อมูลว่าใครอยู่ใน group บ้าง
224.0.0.1

Report message ส่งจาก host ไป router 224.0.0.22



รูปแบบ Address

0000:0000

= ::

3223:0000:2212:0000

= 3223::2212:0000

** บั๊วได้ 1 ครั้งต่อ 1 IP

** IPv4 & IPv6 Compare

มี source Address เหมือนกัน
* ต้องมี

ทำเรื่อง security ได้
มากกว่า IPv4

IPv6

ต่อเลขระบุใน network ซึ่ง default จะเป็นอยู่กับ OS
แบ่งตามการทำ routing

-unicast one-to-one

-multicast one-to-many

-Anycast

จะก่อนแล้ว duplicate ระบุทาง

unicast & multicast รวมกัน

จะส่งไปทั่วแทน แล้ว ในตัวแทนส่งต่อ
(load solution)

TRANSPORT LAYER (LAYER 4) UDP/TCP

UDP ^{connection less} ส่งข้อมูลผ่าน พวไปปลายทาง (จาก IP Address)
แต่ไม่มีความเชื่อถือ, ไม่รับประกัน แต่ใจ
ว่าจะถึงปลายทาง? เหมาะกับ TFTP, RIP

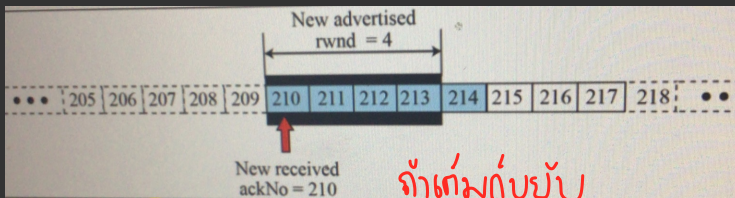
TCP ทำเหมือน UDP แต่จะมี Handshake ไปยังปลายทางก่อน
ว่าจะทำการส่ง เพื่อให้แน่ใจว่าข้อมูลจะถึงจริง
และมี Timeout เพื่อส่งใหม่ **connection oriented**

Control field (segment header)

- Source port 16 bit ต้นทาง
- Destination port
- Sequence number เฉพาะ TCP
- Control Bit flags ควบคุมการเชื่อมต่อ, การส่งต่อ
- URG (Urgent pointer) ด่วน
- ACK (Acknowledgement) การรับทราบ
- PSH (PUSH) Request for push ข้อมูล
- RST (Reset) reset connection เพราะ errors
ทั้ง 2 ฝ่าย อาจทำให้ข้อมูลที่ส่งไปรบกวนทางนาฬิกา
- SYN (Synchronize sequence number)
- FIN ปิดการเชื่อมต่อ No data from sender



Flow Controls → sliding windows ขยับไปเรื่อยๆ



Error Controls → ถ้า sender ส่งไปแล้ว ไม่มี response
response จะส่งมาแจ้ง sender ก็จะส่งใหม่

TCP congestion control (ความแออัด)



เมื่อมีคองเจสชัน และ TCP ก็พยายามส่งข้อมูลทุกก้อน
detection

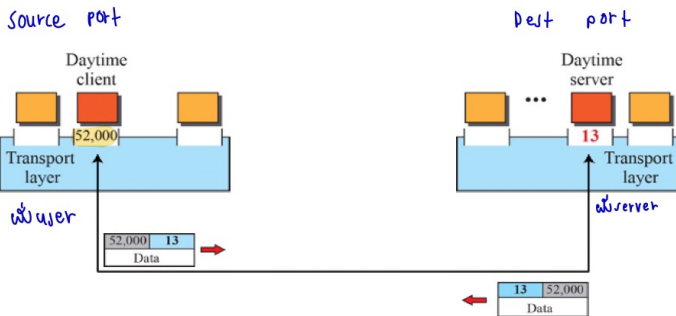
- Timeout ถ้า Timeout ก็ไม่ได้ message กลับมา
- ได้รับ Ack มา 3 times อาจได้ Ack มาล่าช้า
 - RTT (Round trip time) reach destination and ack to be received

Policies

- slow start (เริ่มอย่างช้าๆ) ค่อยๆส่งไปที่ละอัน เพิ่มเรื่อยๆ
- congestion Avoidance ส่งไปเรื่อยๆที่เดียว

** ปัญหาใช้ TCP

- เพราะสร้าง connection ค่อนข้างมาก
- เว้นลำดับการทำงาน, เกิด error ตลอด (ในระบบทางหลวง)



A มี source 52000 ไปที่ B port 13
ก็ไปอยู่ที่ C port 80
ทั้งสองที่เหมือนกัน

ถ้า process หนึ่ง
ก็ gen port หนึ่ง
จะไม่ใช้ 52000 - 4 แล้ว
มัน เวลาใช้มันก็

ไม่ gen port
ใน process เดียวกัน

1.10

socket address (ฝั่ง server)
- รอรับ service จาก client

port 69 → TFTP service
(UDP/TCP) เอาไว้ Transfer
ไฟล์เล็ก ๆ เช่น config
อัปเดตไฟล์ คล้าย drive

** Network layer (layer 3)

→ เป็น connection less
เพราะจะถามทีละ hop
ทีละ router ต่อถัดไปเรื่อยๆ
(next hop)

Sliding window

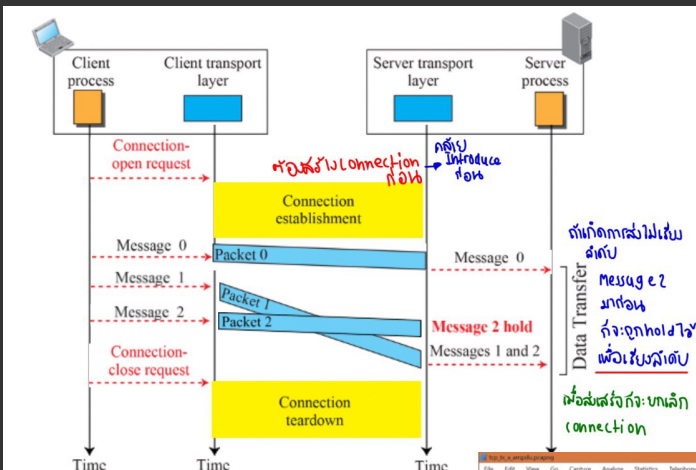
เหมือน Buffer ที่พักข้อมูล (ไม่รับข้อมูล ระบายไป)



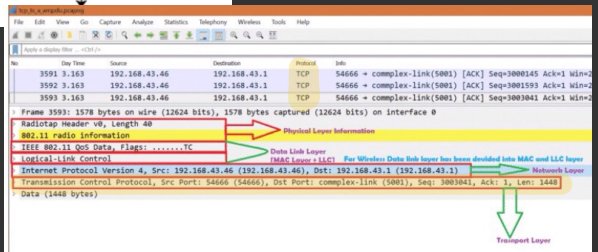
ถ้าเต็มก็จะ drop ทั้ง
เมื่อ 0 ฉุกเฉินไปรับ ก็จะขยับไปเรื่อยๆ

Connection

Connection less → ส่งข้อมูลไม่เรียงลำดับ ขึ้นอยู่กับ Application
 ** Connection oriented → สร้าง connection ก่อน
 recommend เหมือน introduce



HTTP → Application Layer



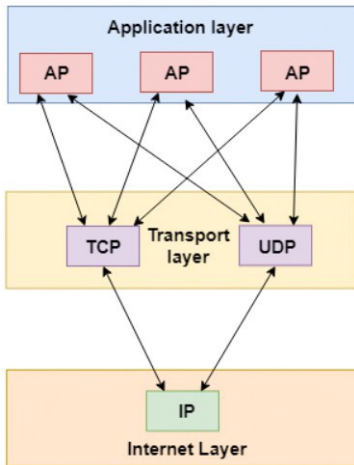
TRANSPORT LAYER PROTOCOL

- Stop & wait protocol - ส่งไปถ้าไม่ได้ ACK ก็จะ resnet
- Go-Back - ถ้า lost จะกลับไปเริ่มใหม่ในครั้ง แต่จะส่งในครั้งก่อน ไม่หยุด จากนั้นค่อย resnet ใหม่ตั้งแต่แรก
- Selective repeat - resnet อันที่ lost เท่านั้น (ส่งในครั้งก่อนค่อย repeat)

APPLICATION LAYER (LAYER 5) ใกล้เคียงกับ user มากที่สุด

Show program ที่มีการส่งข้อมูลผ่าน NW, ตอบกลับ Application
PROTOCOL ที่รับส่งข้อมูล

- ① HTTP ส่งข้อมูลรูปแบบ ข้อความ กับ WWW. server
- ② FTP บน port 20, 21 ส่งข้อมูลระหว่าง client & server
- ③ DNS แปลง domain name to IP Address (www.domainname.com)
- ④ UDP
- ⑤ POP
- ⑥ SNMP (Simple network management protocol)
เป็น protocol ให้อุปกรณ์ device ใน IP Network



LAYER 5 (Application layer)
สามารถส่งข้อมูลโดยใช้ TCP/UDP
ทั้ง TCP/UDP จะ communicate
กับ internet