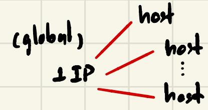
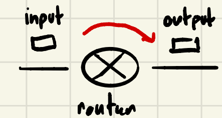
Chapter 4 → Network Layer

- NAT (Network Address Translation) → สามารถทำให้ IP 1 IP ใช้กับ host ได้มากกว่า 1 host.



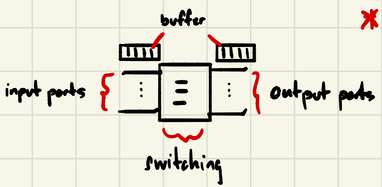
- middleboxes → อุปกรณ์คอมพิวเตอร์ - Firewall กรอง traffic

- NETWORK
  - → Data plane - Hardware, fowarding, Per-router function, พิจารณา packet ที่มาจาก input และพิจารณาจาก ip address ของ host ปลายทาง
  - → Control plane - Software, Routing.



- Router → ประกอบไปด้วย input ports, switching, output ports, buffer management, scheduling. (scheduling: การปล่อย data ไหนออกก่อน.)

[IPv4 → 32 bit, IPv6 → 128 bit.]



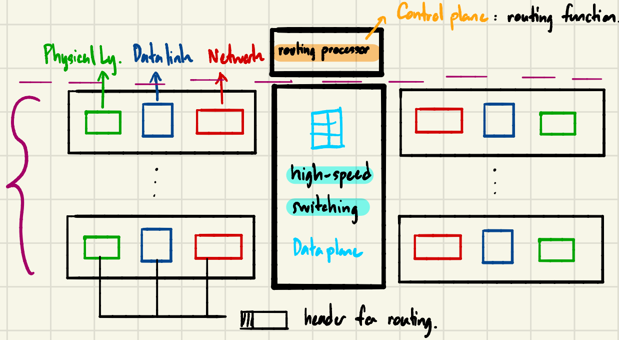
* การทำงานส่วนใดส่วนหนึ่ง อาจจะทำงานเร็วกว่าส่วนอื่นๆ ก็เป็นได้ จึงมี buffer คอยเก็บ data เพื่อไม่ให้ data หาย *

- SDN (Software Define Network) → ช่วยแยก data plane และ control plane ออกจากกัน
  - (data plane) → อยู่ใน router และมี fowarding table → เพื่อบอกว่า data นี้ ต้องออก output ช่องไหน

- Services of Network Layer
  - Sender: encapsulates segments → datagram ใส่หัว (address) ให้กับ data
  - Reciever: deliver segments to transport ถอดหัวและส่ง data ให้กับ transport
  - Router: ส่ง data จาก source → destination (end to end) (โดยผ่านการ routing) (routing: การจัดเส้นทาง.)
  - Guarantee delivery (w/ less than x ms delay)
  - In-order of flow data, guarantee min. bandwidth to flow

- Functions of Network layer. (การเดินทาง)
  - forwarding: ย้าย packet จาก $input_{router}$ → $output_{router}$ { data plane: forwarding table (data plane → router.)
  - routing.: พิจารณาเส้นทางของ packet ว่าจะส่งไปทางไหน { control plane: routing algorithm
    - (routing) แผนการเดินทาง.
    - (control plane) → routing algorithm (router)
    - (control plane) → SDN (remote controlled) → มี Controll Agent

• Router Components



management control plane (SW) (ms)

↑ faster.

forwarding data plane (HW) (ns)

INPUT FUNCTIONS

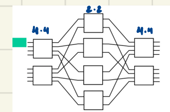
- Physical Layer : NRZ, RZ, manchester...
- Data link layer : stop and wait, go back N, Selective ARQ.

} Data Com.

- Network layer
  - using header fields value (macth plus action)
    - ↳ ใช้ forwarding table เพื่อหา output port.
  - Queuing : เกิดขึ้นเมื่อ datagrams มาถึงเร็วกว่า อัตราการไหลของข้อมูลผ่านข้างใน switch (forwarding rate) → ปัญหา 1. buffer overflow 2. Head of the line blocking (HOL)
    - ↳ ตัวหน้าไปพร้อมกันไม่ได้ต้องรอ
  - Decentralized switching : มี forwarding table อยู่ใน header ของ input.

SWITCHING FUNCTIONS

- switching rate → N·R → line rate
  - ↳ N input
- 3 major types 1. memory / 2. bus (↱ backbone) / 3. interconnection (↱ crosspoint ↱ จน. crosspoint = $N_{input} \times N_{output}$)
  - multistage switch



8x8 multistage switch
built from smaller-sized switches

จน. crosspoint = 2(4×4) + 4(2×2) + 2(4×4)

  - parallelism : ในทางเข้าแบ่ง datagrams เป็นชิ้นย่อยๆก่อน
    ในทางออกรวม datagrams กลับมาเป็นก้อน

OUTPUT FUNCTIONS

- Network layer
  - Queuing : เกิดขึ้นเมื่อ switch ทำงานเร็วกว่า output port. → ถ้า bufferเต็ม → Drop policy (Data loss)
  - Scheduling : เลือก datagrams ก่อนส่งออก (Priority)

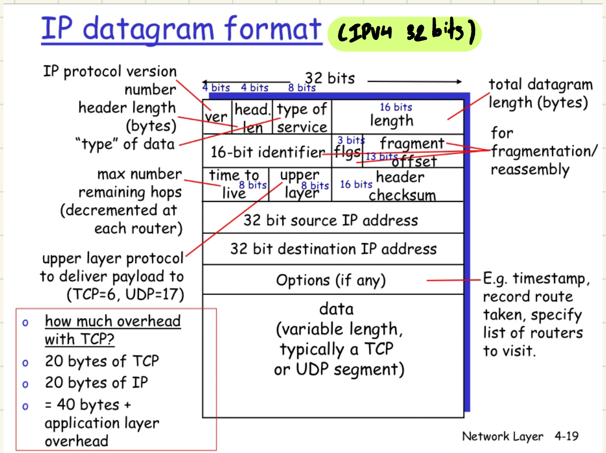
1 flow: average buffering = RTT × $Link_{BW}$

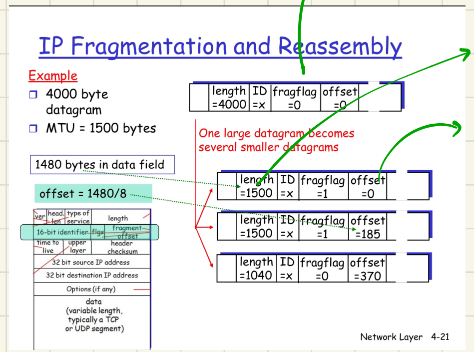
N flow: average buffering = $\frac{RTT \times LinkBW}{\sqrt{N}}$

1.40 หน้าใหม่.

- Internet in network layer.
  - 3 ส่วน
    - IP Protocol - บอก field ของ datagram, องค์ประกอบต่างๆใน packet.
    - Routing Protocol - จัดเส้นทางให้ datagram (Fowarding table)
    - ICMP Protocol - บอกว่าเกิด error หรือไม่, router signaling.

- IP datagram format.





- fragflag - 0 ไม่แบ่ง < initial state / final state | 1 แบ่ง
- length. - data length + header length → 20-60 (no options 20, options)
- offset - 1st offset = 0, 2...n ofset = $\frac{\text{data length}}{8}$

Sol'n
- 1st frame: L = 1500 = 20 + 1480 data, os = 0
- 2nd frame: L = 1500 = 20 + 1480 data, os = 185
- 3rd frame: L = 1040 = 20 + 1020 data, os = 370

Conclu.
- all data = 1480 + 1480 + 1020 = 3980 bytes
- header = 20 no options
- final offset = 370

- head. length - มีขนาด 20 bits (no options), มีขนาด > 20 bits ≤ 60 bits (options)
- length = head. length + data length
- 16 bit identifier / flags / fragment offset
  - fragmentation → แยก datagram ให้เป็นชิ้นเล็กๆ
  - reassemble → รวบรวมกลับ
- Time To Live (TTL) - จำนวน hop ที่สามารถผ่านไปได้
- Upper Layer - 6 (TCP), 17 (UDP)
- Checksum - check error
- Options - มีขนาดสูงสุด 40 bytes

IP Fragmentation, IP reassembly
- MTU (Maximum Transfer Unit) - ขนาดสูงสุดของข้อมูลที่สามารถส่งได้ใน 1 frame (ไม่รวม header, trailer.)
- fragmentation - เกิดขึ้นที่ router, reassembly - เกิดขึ้นที่ host des.

- **IPV4 Addressing** → 32 bits, identifying host or router interfaces ↷ การเชื่อมต่อระหว่าง host/router และ physical link
  - host: มักมีแค่ 1 interface; router: มี > 1 interfaces.
  - class-full addressing. - 5 class →
    - class A. byte 1 : network, byte 2-4 : host (0-127)
    - class B. byte 1-2 : network, byte 3-4 : host (128-191)
    - class C. byte 1-3 : network, byte 4 : host (192-223)
    - } host, router interface
    - class D — all byte : multicasting
    - class E — future

* หา จน. network | จน. host *

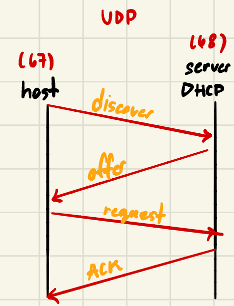
[(Last ip range - first ip range) + 1] × [(Last ip range - first ip range) + 1]

(ดู byte (ที่ใช้))

- **Subnet.** → ถ้า host อยู่ใน subnet เดียวกัน host จะสามารถส่งข้อมูลได้ถึงกันโดยไม่ผ่าน router.

- **CIDR (Classless Inter Domain Routing)** → subnet mask = /x บอก จน. bit ที่เป็น network

- **DHCP (Dynamic Host Configuration Protocol)** → แจก IP ให้กับ host แบบ dynamic
  - หลักการทำงาน
    - → host broadcast (DHCP discover) = [host ส่ง msg ไปหา server]
    - → DHCP server respends (DHCP offer) = [server ตอบกลับ]
    - → host req. IP address (DHCP request) = [host ขอจอง ip จาก server]
    - → DHCP server send address (DHCP ACK) = [host ready to use ip]



- **NAT (Network Address Translation)** (Implement) → ช่วยให้ host หลาย host ใช้ ip มากกว่า 1 ip จาก global ip ได้.
  - (host → global network)
  - Data local network → internet : datagram อยู่ในรูปแบบ private + port →(NAT) global + port
  - (global network → host)
  - Data internet → local network : ⁄————⁄ global + port →(NAT) private port.

- **Address Mapping** → Map จาก logical address (IP) ไป physical address (MAC) โดยใช้ protocol : ARP (Address Resolution Protocol)

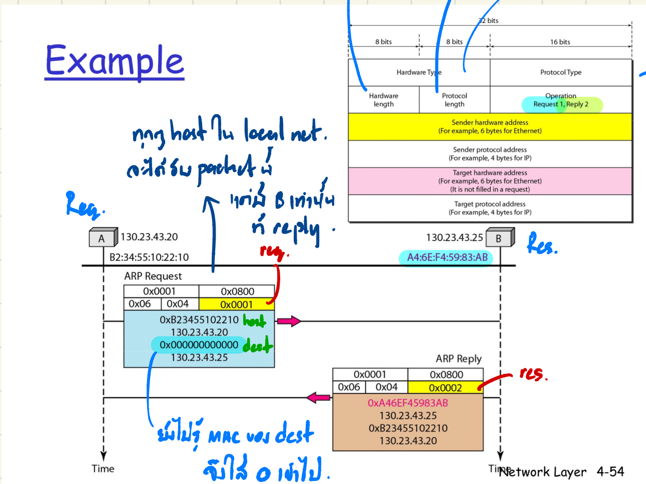
↳ ARP Packet + Example

Ethernet : 6

IPv4 ใช้ : 4

ค่าเป็น 1 : Ethernet , 6 : IEE 802

IPv4 : 0800_{16}



- **ICMP (Internet Control Message Protocol)** → unreliable
  - ↳ ข้อจำกัด! 1. lack of error control (error reported)
    2. host อื่น host is alive/not alive?
  - ↳ message 2 ประเภท
    - → Error-reporting → 5 ชนิด.
      - 3 destination unreachable
      - 4. Source quench. no congestion
      - 11 Time exceeded
      - 12 Parameter problems
      - 5 Redirection
        ↳ บอกว่า Routing ไหนเร็วกว่า
    - → Query message → 4 ชนิด
      - 0 echo reply
      - 8 echo request
      - 9 route adver.
      - 10 route discovery

Time exceeded code → 0 : TTL expired
↳ 1 : Fragment Reassembly

Parameter problems code → 0 : Bad headers
↳ 1 : option part is missing.

Redirection code → 0 : network redirection
↳ 1 : host redirection

- **Ping** → โปรแกรมที่ใช้ตรวจสอบว่า host เปิดหรือปิด? และสามารถส่งข้อมูลได้หรือไม่?

- **Traceroute** → โปรแกรมใช้ตรวจสอบเส้นทางที่ใช้ส่งไปไหนบ้างใน network
  ↳ ใน router มี data อะไรบ้าง.

- IPv6
  - ขนาด Header 40 bytes
  - ไม่มีการแยกย่อย datagram (fragmentation)
  - Tunneling
- Components / field
  - traffic class → congestion control, Non-congestion control.
  - flow label → QoS management.
  - payload range → ความยาวข้อมูลที่บรรจุ.
  - next header → UDP=17, TCP=6
  - hop limit → TTL (in ipv4)

* ไม่มีการแยกย่อย *
- Checksum
- Options

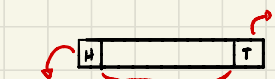
Chapter 6 Data link → implement in Network Adapter.

Introduction.

- node → hosts (PC), router
- link → เส้นที่อยู่ระหว่าง node ได้แก่ wired, wireless, LANs.
- data (unit) ของ datalink → frame (encapsulated datagram)
- หน้าที่หลัก → ส่ง datagram จาก node - node
  - เพิ่ม header และ trailer ให้กับ datagram = frame
- hop - hop delivery
  - Algorithm in flow control: Go back N, Selective, Repeat } Data Com.
- MAC Address เป็นที่อยู่แบบ Physical Address.

- error detection (การตรวจสอบ) เกิดจาก noise, etc. (trailer)
- error correction → แก้ไขด้วยตัวเอง.

★ ข้อมูลระดับ.



} ทั้งก้อนเรียกว่า `frame'

interface communication

- sending → encapsulate (add H, T)
- recieving → look for errors
  - extracts datagram to network layer.

Error detection and Correction. → trailer field

(มีหลายรูปมีการทดสอบ ระเอียด)

- Parity bit checking
  - single bit
    - odd check: ถ้า data bit (1) เป็น คู่ ∴ parity bit = 1
      - ที่ ∴ parity bit = 0
    - even check: คู่ ∴ parity bit = 0
      - ที่ ∴ parity bit = 1
  - two dimensional → check both row + column.

- CRC → มีหลายแบบหลายรูป
  - Analysis formula: $\frac{x^q \cdot D(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$
    - D(x) — Dataword (initial code)
    - highest expo in D(x)
    - Q(x) — Quotient.
    - R(x) — remainder.
    - G(x) — Divisor.
  - Codeword C(x): $x^q \cdot D(x) + R(x)$
    - ชุดข้อมูลที่ถูกต้อง.

- check sum → เช็คที่ trailer ของ reciever. (16 bits)
  - ถ้าตรงกับที่ส่งไป: no errors, pass
  - ถ้าไม่ตรง: errors!

Multiple Access Protocol.

- 2 types
  - point to point → host to host.
  - broadcast → shared
- collision → การชนกันระหว่าง signals 2 สัญญาณขึ้นไป.
- calculate with multiple access channel (MAC) [กำหนดให้ ช่องนี้ส่งด้วย rate R bps]
  - ส่ง 1 node → ส่งเร็วในการส่ง "R bps"
  - ส่ง m node → ส่งเร็วในการส่ง "$\frac{R}{m}$ bps"

Type of Multiple Access Channel

- 3 Groups

Channel partitioning → การแบ่งช่องทางการสื่อสาร TDMA, FDMA, CDMA.

1. TDMA - ใช้ช่วงเวลาในการแบ่งในการส่งข้อมูล.
2. FDMA - ใช้ความถี่ในการแบ่งในการใช้ช่องทางติดต่อสื่อสาร.
3. CDMA - ใช้ code ในการเข้ารหัส.

Random access protocol

(ทำให้ไม่รู้ว่าจะส่งข้อมูลเวลาไหนดี.

- ★ Pure Aloha → no synchronization, unslotted
  - easy to get collision.
- ★ Slotted Aloha → frame มีขนาดเท่ากันทุกๆ frame
  - ถ้าเกิดการ 'collision' จะมีโอกาส p ในการส่งใหม่
- ★ CSMA (Carrier Sense Multiple Access)
  - ถ้าช่องว่าง(ปกติ cidle) → ส่งเลย. (เพราะรู้ว่าเปลี่ยนการส่งเสร็จ, เลย)
  - เกิดการ collision, busy → ยกเลิกและเลื่อนการส่งข้อมูล
- ★ CSMA/CD algo
  1. NIC ได้รับ datagram จาก NW. และ สร้าง frame ขึ้นมา (เพิ่ม header, trail)
  2. CH ว่าง → ส่ง, CH ไม่ว่าง → รอจนว่าง ค่อยส่งอีกที
  3. หลังจากการหยุดส่ง NIC จะทำการ 'binary exponential backoff'
     - มีการวน m ครั้ง และครั้งที่ m จะมีการสุ่มเลข K
     - จาก $0 - 2^{m-1}$ และทำการรอด้วย 'K·512 bit times'

back to step 2

(เลื่อนเวลาให้ node ส่งข้อมูลแบบไม่ชนกัน.

- node ทุกช่องมีอัตราเท่าเทียมกัน.
- เมื่อเกิด 'collision' ก็ต้องมีวิธี 'recover from collision' (delay retransmission)
- ตัวอย่างของ R.A.P. → Aloha, Slotted Aloha, CSMA, CSMA/CD, CSMA/CA (no collision, collision)

Taking turns → polling → loop checking devices Master จะเป็นตัวถาม Slaves ว่า Slaves ต้องการส่งข้อมูลหรือไม่ yes → ส่งมา no → ผ่าน.

- 2 types 1. central sites ปกติ, 2. token pass ลูปวงกลม.

LANs

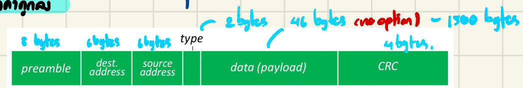
- IP Address → 32 bit address, use for interface. (network layer) (dec no.)
- MAC Address → 48 bit address, hop to hop, subnet เดียวกัน (datalink layer) (hex no.)
  → MAC ไม่ซ้ำกันเหมือนกัน จัดการโดย IEEE
  ↳ 24 bit แรก - fixed, 24 bit หลัง - random.
- switch ไม่มี interface ที่เป็น MAC Address.
- ใน ARP table จะมีตาราง IP, MAC, TTL → บอกว่าจะเก็บ MAC ของ IP นี้ไปถึงตอนไหน?
  ↳ แรกเริ่มยังไม่รู้ว่า dest. MAC คือเครื่องไหน ∴ ใส่ boardcast เข้าไปแทน.
  ทุก node ได้รับ 'ARP query' แต่ใน boardcast นั้นมี target ip address แฝงอยู่ / node ที่มี ip นี้ก็จะตอบกลับ ไปหาตัว request.

Ethernet

- การทำงานของ token ring, FDDI, ATM ซับซ้อน, ราคาสูง
- มีการปรับปรุง version + data rate ที่ดีขึ้น
- Hardware เช่น Adapter, switch ราคาถูกลง

} สาเหตุที่ยังใช้ ethernet อยู่.

- Ethernet frame Structure



preamble - มีเพื่อ synchronize กับ reciever 7 bytes แรก. → 10101010..., bytes สุดท้าย. 10101011

dest / source address - บอก MAC ของ host ทั้ง S/D, ถ้า target ip ไม่ตรง host จะ discard frame ทิ้ง.

type - บอกประเภทของ protocol เช่น ipv4, ipv6, arp ...etc.

CRC - error detection.

Switch

→ 2 แบบ ① Switch layer 2 - ทำงานที่ physical + data link
② Switch layer 3 - 〃 physical + data l. + network

- Store, selectively foward
  L ตารางสอบ   L ส่งไปหา port ที่เหมาะสม.
- มีกระบวนการ self-learning
  ↳ switch จะมีข้อมูล (MAC, Port, TTL) ก็ต่อเมื่อ host ส่งข้อมูลมายัง switch.
- filtering / fowarding.
  ถ้าไม่มีข้อมูลในตาราง → broadcast ไปทุก port. (flooding)
  ถ้ามีข้อมูล → ส่งไปที่ port นั้นเลย.
  (ถ้า host ได้รับ frame พร้อมกัน = collision → drop frame)