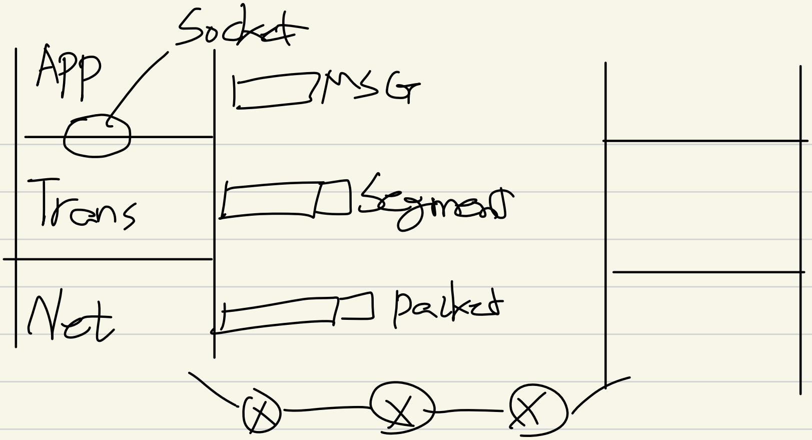


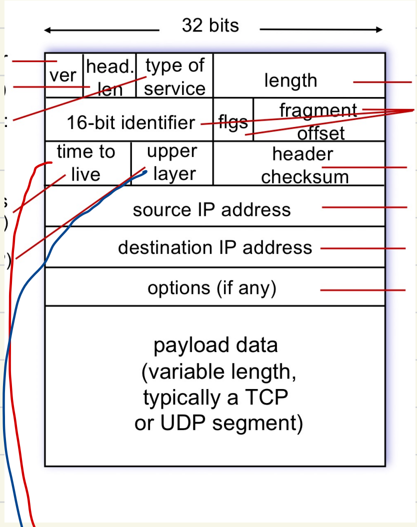
## Two key network-layer functions

- forwarding : 목적지 주소를 보고 알맞은 곳으로 보냄

- routing : forwarding table을 채워 놓는 일

Longest prefix matching : forwarding table에서 가장 길게 매칭 되는 것을 선택

## IP (Internet Packet)



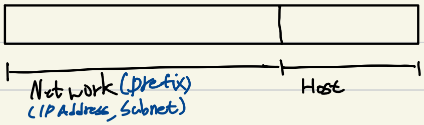
TTL : 영원히 존재 못함

Upper layer : TCP or UDP

## IP addressing (32bit)

- Interface를 지칭함 (머신 자체X)
- Scalability Challenge (확장성 문제) → hierarchical addressing

ex) 12.34.158.0/24
↳ Network 24bit
Host 8bit



12 34 158 5 12.34.158.0/24
00001100 00100010 10011110 00000101 (human readable)
11111111 11111111 11111111 00000000 (machine readable)
255 255 255 0

↳ "AND" operation → Network ID
bitwise

# Classful Addressing

- 클래스를 정해서 /8, /16, /24 이런식으로 배정 → 비효율적

# Classless Inter Domain Routing (CIDR)

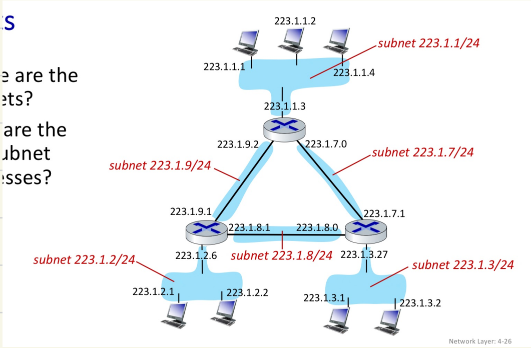
- 필요한 만큼 가변적으로 사용할 수 있음

# Longest Prefix Match Forwarding

- 가장 길게 match하는 prefix로 출력 포트 배정

# Subnets

- Network Id = Prefix = IP Address = Subnet
- 라우터를 거치지 않고 직접 접근할 수 있는 집합

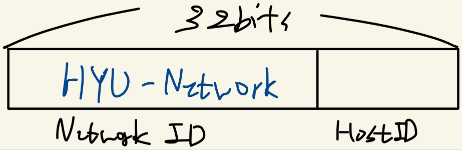


s

e are the
ets?

are the
ubnet
esses?

Network Layer: 4-26



assume) HYU-Network

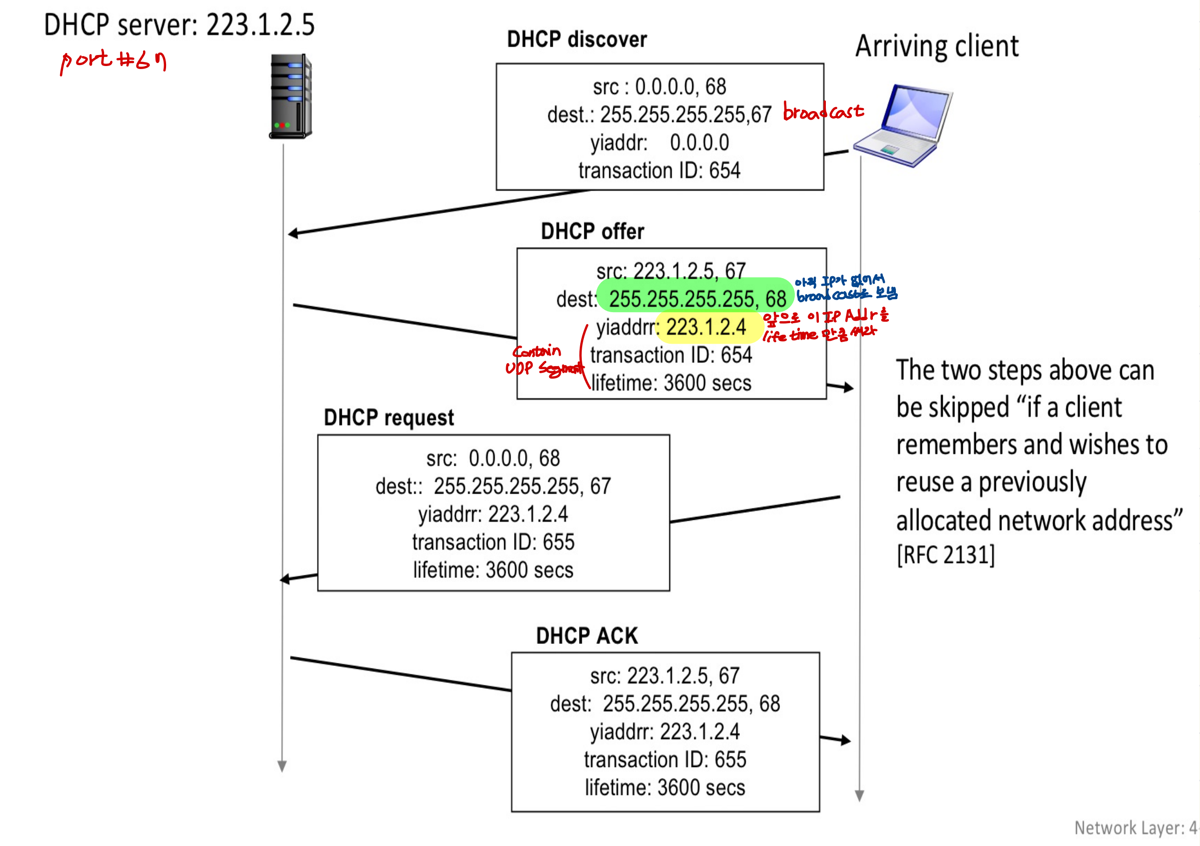
Subnet = 24bit Host = 8bit => $2^8-1$ hosts

20bit 12bit => $2^{12}-1$ hosts

(Subnet이 작을 수록 실제 네트워크는 큼)

# DHCP (Dynamic Host Configuration (구성) Protocol)

- DHCP Client port#68 모든 Subnet은 DHCP Server를 가짐
- DHCP Server port#67



DHCP server: 223.1.2.5
port#67

Arriving client

**DHCP discover**

src : 0.0.0.0, 68
dest.: 255.255.255.255,67 broadcast
yiaddr: 0.0.0.0
transaction ID: 654

**DHCP offer**

src: 223.1.2.5, 67
dest: 255.255.255.255, 68 아직 IP가 없어서 broadcast로 보냄
yiaddrr: 223.1.2.4 앞으로 이 IP Addr을 lifetime 만큼 써라
transaction ID: 654
lifetime: 3600 secs
Contain UDP Segment

**DHCP request**

src: 0.0.0.0, 68
dest:: 255.255.255.255, 67
yiaddrr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs

**DHCP ACK**

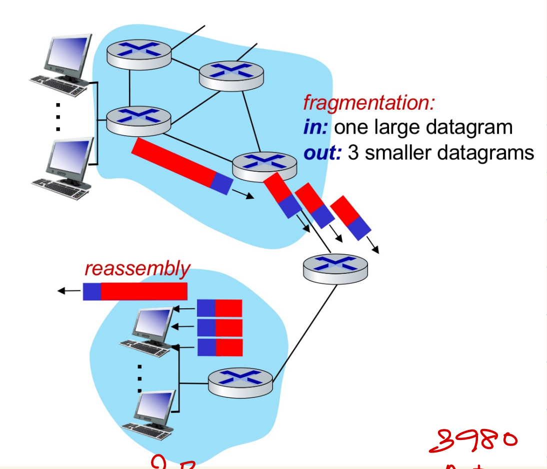
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddrr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs

The two steps above can be skipped "if a client remembers and wishes to reuse a previously allocated network address" [RFC 2131]

Network Layer: 4

- DHCP에서 IP를 제공할때
  - IP Address
  - SubNet Mask
  - DNS Server IP Address
  - Gate way Router IP Address

# IP fragmentation/reassembly

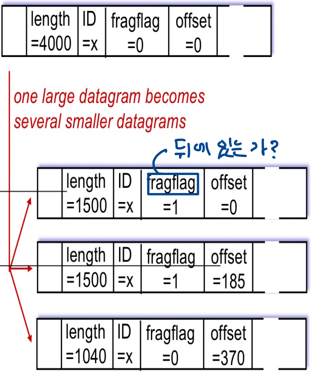


MTU(Maxium Transfer Unit)

- packet size가 MTU보다 크면 쪼개서 보냄

example:

- 4000 byte datagram
- MTU = 1500 bytes



| length | ID | fragflag | offset |
|---|---|---|---|
| =4000 | =x | =0 | =0 |

one large datagram becomes several smaller datagrams

1480 bytes in data field

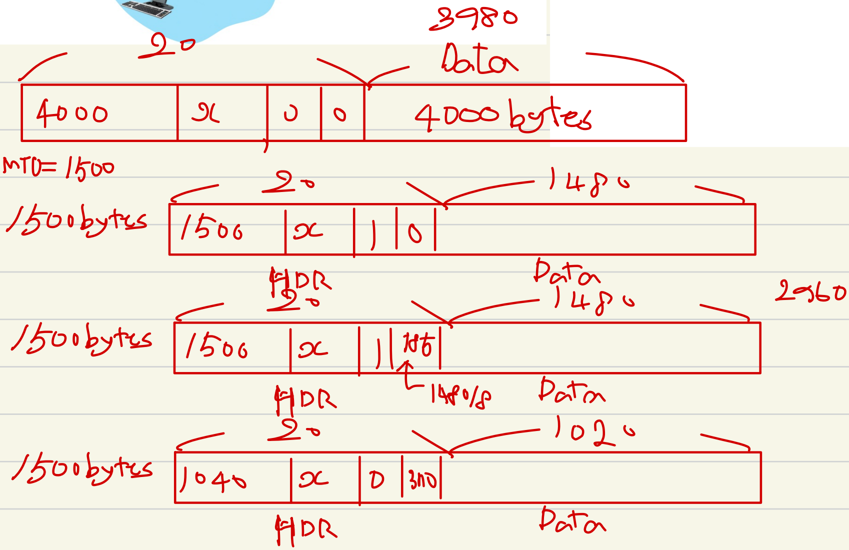
offset = 1480/8

뒤에 있는가?

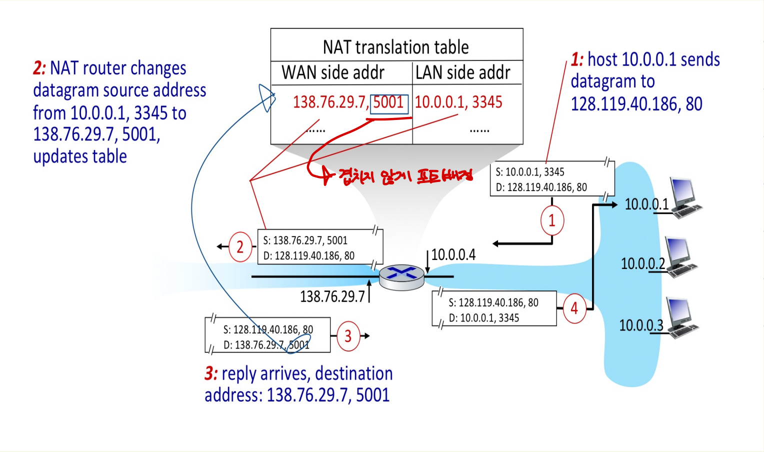
| length | ID | fragflag | offset |
|---|---|---|---|
| =1500 | =x | =1 | =0 |

| length | ID | fragflag | offset |
|---|---|---|---|
| =1500 | =x | =1 | =185 |

| length | ID | fragflag | offset |
|---|---|---|---|
| =1040 | =x | =0 | =370 |



# NAT(network address translation)



| NAT translation table | |
|---|---|
| WAN side addr | LAN side addr |
| 138.76.29.7, 5001 | 10.0.0.1, 3345 |
| …… | …… |

겹치지 않게 포트배정

2: NAT router changes datagram source address from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

3: reply arrives, destination address: 138.76.29.7, 5001

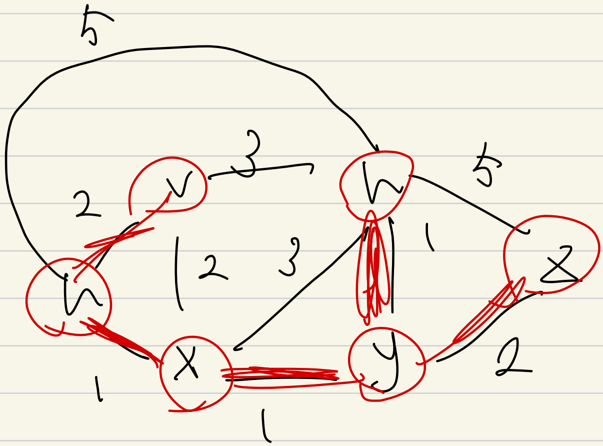
LAN(Local Area Network)

WAN(Wide Area Network)

문제점

- Server의 역할을 하기 힘들어짐
- 계층화의 핵심이 무너짐
  - Network layer에서 packet header에서 IP주소를 수정
  - port# 열람

|   | N' | D(v) P(v) | D(w), P(w) | D(x), P(x) | D(y), P(y) | D(z), P(z) |
|---|---|---|---|---|---|---|
| 0 | u | 2, u | 5, u | 1, u | ∞ | ∞ |
| 1 | ux | 2, u | 4, x |  | 2, x | ∞ |
| 2 | uxv |  | 4, x |  | 2, x | ∞ |
| 3 | uxvy |  | 3, y |  |  | 4, y |
| 4 | uxvyw |  |  |  |  | 4, y |
| 5 | uxvywz |  |  |  |  |  |



4+0

50+0

4+1

$D_x(y) = \min\{c_{xv} + D_v(y)\}$

1 0

50+0

1 +

0

1+5

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 1 | 0 | 1 |
| Z | 5 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 1 | 0 | 1 |
| Z | 2 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 1 | 0 | 1 |
| Z | 5 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 1 | 0 | 1 |
| Z | 5 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 1 | 0 | 1 |
| Z | 2 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

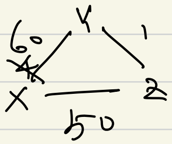
| | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 1 | 0 | 1 |
| Z | 2 | 1 | 0 |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 1 | 0 | 1 |
| Z | 2 | 1 | 0 |

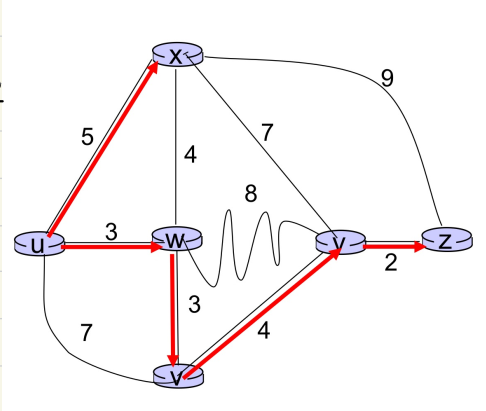
| $D_x$ | x | y | z |
|---|---|---|---|
| x | | | |
| y | | | |
| z | | | |

| $D_y$ | x | y | z |
|---|---|---|---|
| x | ∞ | ∞ | ∞ |
| y | 60 | 0 | 1 |
| z | ∞ | ∞ | ∞ |

| $D_z$ | x | y | z |
|---|---|---|---|
| x | | | |
| y | | | |
| z | | | |



| | N' | D(v), p(v) | D(w), p(w) | D(x), p(x) | D(y), p(y) | D(z), p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 7, u | 3, u | 5, u | ∞ | ∞ |
| 1 | uw | 6, w | | 5, u | 11, w | ∞ |
| 2 | uwx | 6, w | | | 11, w | 14, x |
| 3 | uwxv | | | | 10, v | 14, x |
| 4 | uwxvy | | | | | 12, y |
| 5 | uwxvyz | | | | | |



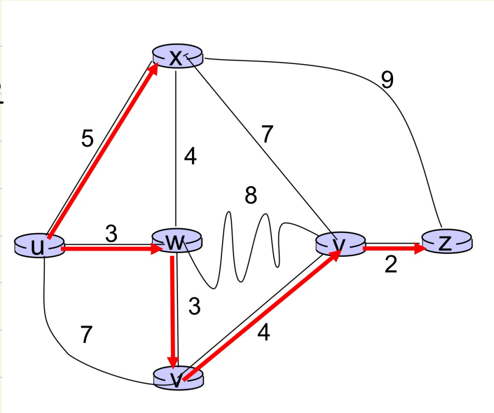
# Routing Protocol

- link state (Dijkstra's Algorithm) - interation

  - $C_{x,y}$ : link cost from $x$ to $y$
  - $D(v)$ : least cost path from source to destination $v$
  - $p(v)$ : predecessor node along path from source to $v$
  - $N'$ : set of nodes whose least cost path definitively known (내가 알고 있는 최단경로의 집합)

$$D(b) = \min(D(b), D(a) + C_{a,b})$$

|  | N' | D(v), P(v) | D(w), P(w) | D(x), P(x) | D(y), P(y) | D(z), P(z) |
|---|---|---|---|---|---|---|
| 0 | u | 7, u | 3, u | 5, u | ∞ | ∞ |
| 1 | uw | 6, w |  | 5, u | 11, w | ∞ |
| 2 | uwx | 6, w |  |  | 11, w | 14, x |
| 3 | uwxv |  |  |  | 10, v | 14, x |
| 4 | uwxvy |  |  |  |  | 12, y |
| 5 | uwxvyz |  |  |  |  |  |



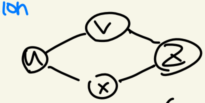
resulting forwarding table in u

| destination | outgoing link |
|---|---|
| v | (u, w) |
| w | (u, w) |
| x | (u, x) |
| y | (u, w) |
| z | (u, w) |

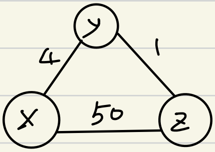
## - Distance vector (Bellman-Ford) - recursion

$$D_x(y) = \min_v \{ \underline{c_{x,v}} + D_v(y) \}$$

↳ 원래초기값



$$D_u(z) = \min \begin{cases} c_{u,v} + D_v(z) \\ c_{u,x} + D_x(z) \end{cases}$$



$$D_x(y) = \min \begin{cases} c_{x,y} + D_y(y) & 4 \\ c_{x,z} + D_z(y) & 51 \end{cases}$$

to

from

| node x | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 50 |
| Y | ∞ | ∞ | ∞ |
| Z | ∞ | ∞ | ∞ |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | $\boxed{4}$ | 5 |
| Y | 4 | 0 | 1 |
| Z | 50 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

| node Y | X | Y | Z |
|---|---|---|---|
| X | ∞ | ∞ | ∞ |
| Y | 4 | 0 | 1 |
| Z | ∞ | ∞ | ∞ |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 50 |
| Y | 4 | 0 | 1 |
| Z | 50 | 1 | 0 |

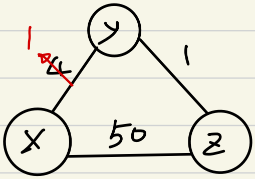
|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

| node z | X | Y | Z |
|---|---|---|---|
| X | ∞ | ∞ | ∞ |
| Y | ∞ | ∞ | ∞ |
| Z | 50 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 50 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

## link cost changes



good news travels fast

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 1 | 0 | 1 |
| Z | 5 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 1 | 0 | 1 |
| Z | 5 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 1 | 0 | 1 |
| Z | 5 | 1 | 0 |

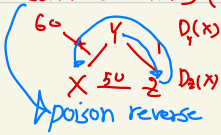
|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 4 | 5 |
| Y | 4 | 0 | 1 |
| Z | 5 | 1 | 0 |

|  | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 2 |
| Y | 1 | 0 | 1 |
| Z | 2 | 1 | 0 |

bad news travels slow

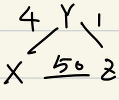
↳ count-to-infinity problem



$D_y(x) = 6$

$D_z(x) = 5$

poisoned reverse



$\underline{D_z(x) = 5}$

y에 의존

y에게 의존한 값을 넘겨줄때
∞로 넘겨줌

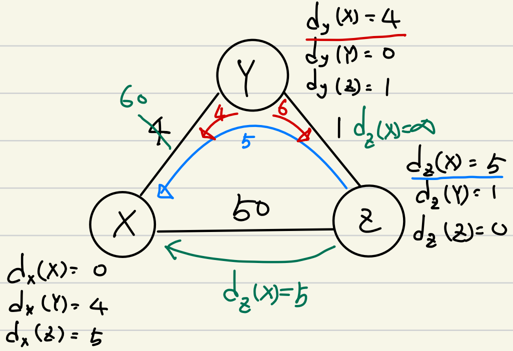
# link Cost Changes

- link cost가 줄어들면

: 금방 안정화가 됨

- link cost가 늘어나면

: 안정화되는데 시간이 오래걸림

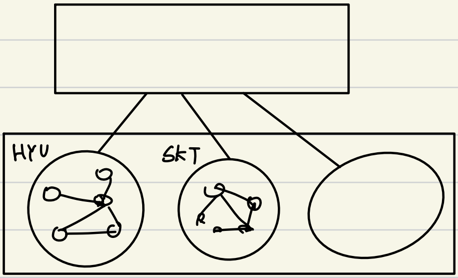


$$D_Y(X) = \min \left\{ \begin{array}{c} C_{YX} + D_X(X) \\ 60 + 0 \\ C_{YZ} + D_Z(X) \end{array} \right\}$$

$$1 + 5$$

poison reverse

# Hierarchical routing



C D

| | x | y | z |
|---|---|---|---|
| x | 0 | 4 | 50 |
| y | $\infty$ | $\infty$ | $\infty$ |
| z | $\infty$ | $\infty$ | $\infty$ |

| | x | y | z |
|---|---|---|---|
| x | 0 | 4 | 5 |
| y | 4 | 0 | 1 |
| z | 50 | 1 | 0 |

| | x | y | z |
|---|---|---|---|
| x | 0 | 4 | 5 |
| y | 4 | 0 | 1 |
| z | 5 | 1 | 0 |

| | x | y | z |
|---|---|---|---|
| x | $\infty$ | $\infty$ | $\infty$ |
| y | 4 | 0 | 1 |
| z | $\infty$ | $\infty$ | $\infty$ |

| | x | y | z |
|---|---|---|---|
| x | 0 | 4 | 50 |
| y | 4 | 0 | 1 |
| z | 50 | 1 | 0 |

| | x | y | z |
|---|---|---|---|
| x | 0 | 4 | 5 |
| y | 4 | 0 | 1 |
| z | 5 | 1 | 0 |

| | x | y | z |
|---|---|---|---|
| x | $\infty$ | $\infty$ | $\infty$ |
| y | $\infty$ | $\infty$ | $\infty$ |
| z | 50 | 1 | 0 |

| | x | y | z |
|---|---|---|---|
| x | 0 | 4 | 50 |
| y | 4 | 0 | 1 |
| z | 5 | 1 | 0 |

| | x | y | z |
|---|---|---|---|
| x | 0 | 4 | 5 |
| y | 4 | 0 | 1 |
| z | 5 | 1 | 0 |