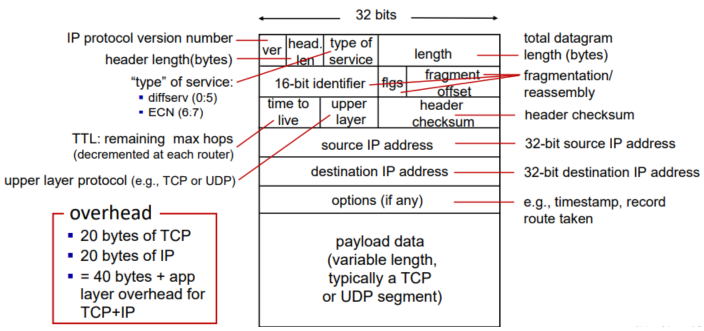
นาย วัฒชัย เตชะลือ Sec53 65015137

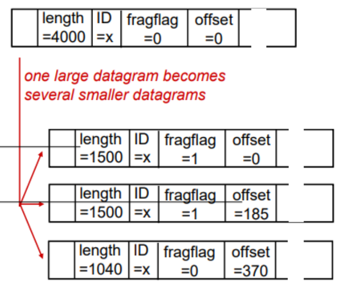
IP Datagram format



No	Network Addr	Subnet ^2	Start IP	Last IP	Broadcast
1	192.168.240.0	255.255.252.0	192.168.240.1	192.168.243.254	192.168.244.255
2	192.168.244.0	255.255.252.0	192.168.244.1	192.168.247.254	192.168.247.255
3	192.168.248.0	255.255.254.0	.248.1	.249.254	.254.255
4	192.168.250.0	255 254.0	250.1	251.254	.251.255
5	252.0	254.0	252.1	253.254	253.255
6	254.0	255.0	254.1	254.254	254.255
7	255.0	255.192	255.1	255.62	255.63
8	255.64	255.192	255.65	126	127
9	128	192	129	190	191
10	192	192	193	254	255

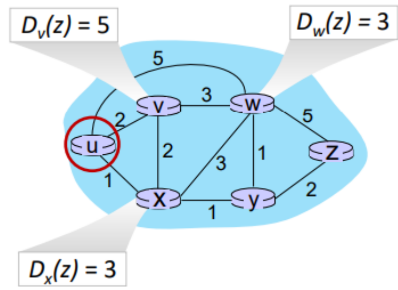
IP fragmentation/reassembly

example:
- 4000 byte datagram
- MTU = 1500 bytes



Bellman-Ford Example

Suppose that *u*'s neighboring nodes, *x,v,w*, know that for destination *z*:



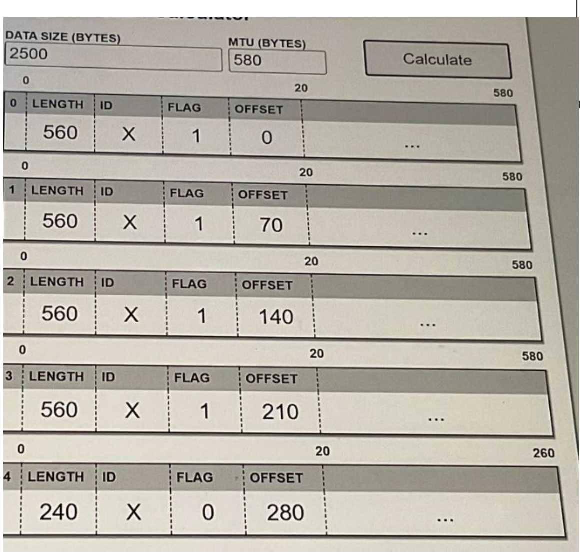
Bellman-Ford equation says:

$$D_u(z) = \min \{ c_{u,v} + D_v(z), c_{u,x} + D_x(z), c_{u,w} + D_w(z) \}$$

$$= \min \{2 + 5, 1 + 3, 5 + 3\} = 4$$

node achieving minimum (x) is next hop on estimated least-cost path to destination (z)

	Hosts	Netmask	Amount of a Class C
/30	4	255.255.255.252	1/64
/29	8	255.255.255.248	1/32
/28	16	255.255.255.240	1/16
/27	32	255.255.255.224	1/8
/26	64	255.255.255.192	1/4
/25	128	255.255.255.128	1/2
/24	256	255.255.255.0	1
/23	512	255.255.254.0	2
/22	1024	255.255.252.0	4
/21	2048	255.255.248.0	8
/20	4096	255.255.240.0	16
/19	8192	255.255.224.0	32
/18	16384	255.255.192.0	64
/17	32768	255.255.128.0	128
/16	65536	255.255.0.0	256



Subnet 1 request 300+1 host
Start : 202.107.16.0/23
End : 202.107.17.255/23
Can use : 202.107.16.1-202.107.17.254

Subnet 2 request 300+1 host
Start : 202.107.18.0/23
End : 202.107.19.255/23
Can use : 202.107.18.1-202.107.19.254

Subnet 3 request 100+1 host
Start : 202.107.20.0/25
End : 202.107.20.127/
Can use : 202.107.20.1-202.107.20.126

Subnet 4 request 100+1 host
Start : 202.107.20.128/25
End : 202.107.20.255/25
Can use : 202.107.20.129-202.107.20.254

Subnet 5 request 100+1 host
Start : 202.107.21.0/25
End : 202.107.21.127/25
Can use : 202.107.21.1-202.107.21.126

Subnet 6 request 50+1 host
Start : 202.107.21.128/26
End : 202.107.21.191/26
Can use : 202.107.21.129-202.107.21.190

Subnet 7 request 50+1 host
Start : 202.107.21.192/26
End : 202.107.21.255/26
Can use : 202.107.21.193-202.107.21.254

Subnet 8 request 50+1 host
Start : 202.107.22.0/26
End : 202.107.22.63/26
Can use : 202.107.22.1-202.107.22.62

Subnet 9 request 50+1 host
Start : 202.107.22.64/26
End : 202.107.22.127/26
Can use : 202.107.22.65-202.107.22.126

Subnet 10 request 30+1 host
Start : 202.107.22.128/26
End : 202.107.22.191/26
Can use : 202.107.22.129-202.107.22.190

Subnet 11 request 30+1 host
Start : 202.107.22.192/26
End : 202.107.22.255/23
Can use : 202.107.22.193-202.107.22.254

Subnet 12 request 30+1 host
Start : 202.107.23.0/26
End : 202.107.23.63/26
Can use : 202.107.23.1-202.107.23.62

Subnet 13 request 16+1 host
Start : 202.107.23.64/27
End : 202.107.23.95/27
Can use : 202.107.23.65-202.107.23.94

Subnet 14 request 16+1 host
Start : 202.107.23.96/27
End : 202.107.23.127/27
Can use : 202.107.23.97

Subnet 15 request 16+1 host S
tart : 202.107.23.128/27
End : 202.107.23.159/27
Can use : 202.107.23.129-202.107.23.158

Subnet 16 request 16+1 host
Start : 202.107.23.160/27
End : 202.107.23.191/27
Can use : 202.107.23.161 -202.107.23.160

นาย วัฒชัย เตชะลือ Sec53 65015137

ทออกมา

IP header 20 bytes

ส่งข้อมูล 4,000 bytes MTU = 1,500 bytes

lenght	IP	fragflag	offset
4000	=X	= 0	= 0

4000 - 20 = 3980 bytes

0 = สุดท้าย
1 = มีต่อ

1480/8 = 185

ส่งได้ครั้งละ 1500 - IP header = 1480

ครั้งต่อไป = (1480 * 2)/8

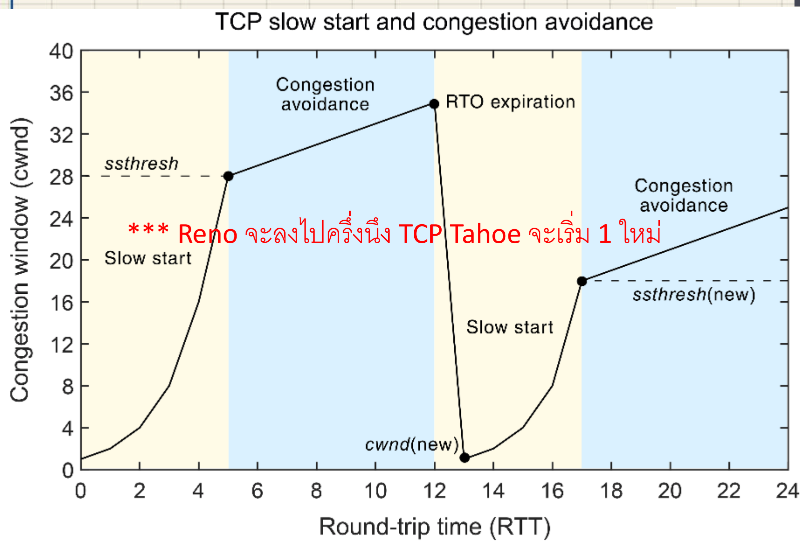
ตอนแบ่ง ข้อมูล 1480

lenght	IP	fragflag	offset
1500	=X	= 1	0

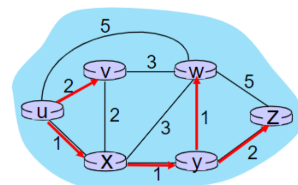
lenght	IP	fragflag	oflse
1500	=X	= 1	185

lenght	IP	fragflag	oflse
1040	=X	= 0	370

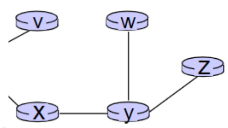
No.	Network Address	Subnet Mask	IP address แรกที่ host หรือ router สามารถนำไปใช้ได้	IP Address สุดท้ายที่ host หรือ router สามารถนำไปใช้ได้
1	192.168.240.0	/22	192.168.240.1	192.168.243.254
2	192.168.244.0	/22	192.168.244.1	192.168.247.254
3	192.168.248.0	/23	192.168.248.1	192.168.249.254
4	192.168.250.0	/24	192.168.250.1	192.168.250.254
5	192.168.251.0	/24	192.168.251.1	192.168.251.254
6	192.168.252.0	/26	192.168.252.1	192.168.252.62
7	192.168.252.64	/26	192.168.252.65	192.168.252.126
8	192.168.252.128	/27	192.168.252.129	192.168.252.158
9	192.168.252.160	/27	192.168.252.161	192.168.252.190
10	192.168.252.192	/27	192.168.252.193	192.168.252.222



stra's algorithm: an example



east-cost-path tree from u:



resulting forwarding table in u:

destination	outgoing link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
x	(u,x)

route from u to v directly

route from u to all other destinations via x

jkstra's algorithm: an example

Step	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	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

10000011 11101101 00100111

กำหนด IP address 131.237.39.76 ให้หา Network ID สำ

137.0.0.0 กรณี /8

137.224.0.0 กรณี /12

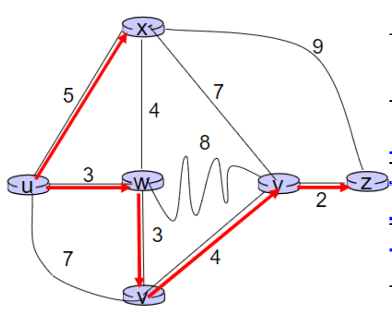
137.237.32.0 กรณี /20

137.237.36.0 กรณี /22

กรณี /8
กรณี /12
กรณี /20
กรณี /22

Dijkstra's algorithm: another example

Step	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					



01001100

ถ้า 12 ให้ไล่ไป 12 Bit

/8 /12 /20 /22

11111111.11111111.11111100.00000000 } and กัน
10000011 11101101. 00100111 01001100

10000011.00000000.00000000.00000000
10000011.11100000.00000000.00000000
10000011.11101101.00100000.00000000
10000011.11101101.00100100.00000000

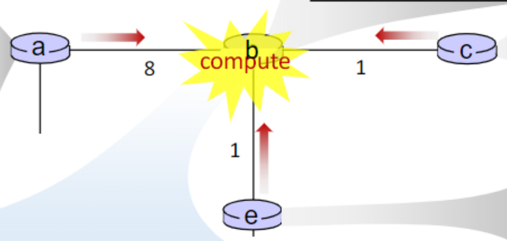
128 64 32 16 8 4 2 1

นาย วัฒชัย เตชะลือ Sec53 65015137

Distance vector example: (

t=1

- b receives DVs from a, c, e, computes:



DV in a:
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

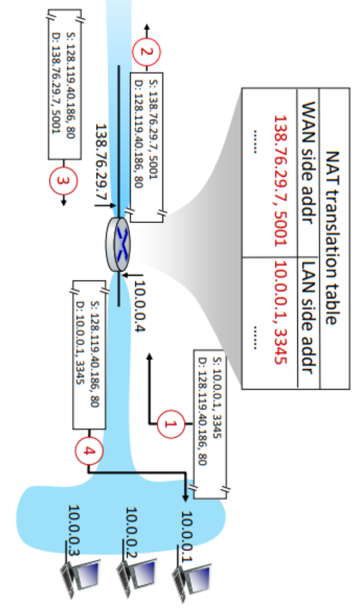
DV in b:
$D_b(a) = 8$ $D_b(f) =2$
$D_b(c) = 1$ $D_b(g) = \infty$
$D_b(d) = 2$ $D_b(h) = 2$
$D_b(e) = 1$ $D_b(i) = \infty$

DV in c:
$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$

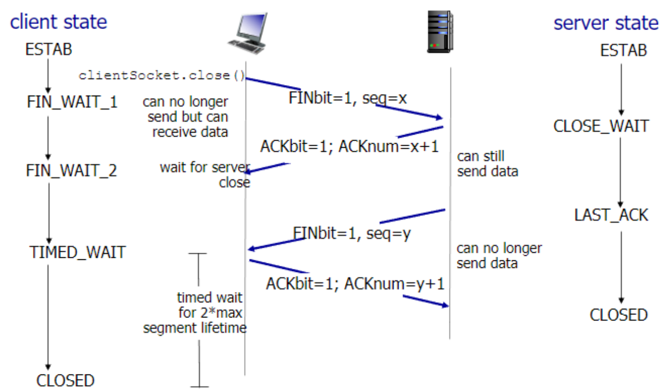
DV in e:
$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

$D_b(a) = \min\{c_{b,a}+D_a(a),\ c_{b,c}+D_c(a),\ c_{b,e}+D_e(a)\} = \min\{8,\infty,\infty\} = 8$
$D_b(c) = \min\{c_{b,a}+D_a(c),\ c_{b,c}+D_c(c),\ c_{b,e}+D_e(c)\} = \min\{\infty,1,\infty\} = 1$
$D_b(d) = \min\{c_{b,a}+D_a(d),\ c_{b,c}+D_c(d),\ c_{b,e}+D_e(d)\} = \min\{9,2,\infty\} = 2$
$D_b(e) = \min\{c_{b,a}+D_a(e),\ c_{b,c}+D_c(e),\ c_{b,e}+D_e(e)\} = \min\{\infty,\infty,1\} = 1$
$D_b(f) = \min\{c_{b,a}+D_a(f),\ c_{b,c}+D_c(f),\ c_{b,e}+D_e(f)\} = \min\{\infty,\infty,2\} = 2$
$D_b(g) = \min\{c_{b,a}+D_a(g),\ c_{b,c}+D_c(g),\ c_{b,e}+D_e(g)\} = \min\{\infty, \infty, \infty\} = \infty$
$D_b(h) = \min\{c_{b,a}+D_a(h),\ c_{b,c}+D_c(h),\ c_{b,e}+D_e(h)\} = \min\{\infty, \infty, 2\} = 2$
$D_b(i) = \min\{c_{b,a}+D_a(i),\ c_{b,c}+D_c(i),\ c_{b,e}+D_e(i)\} = \min\{\infty, \infty, \infty\} = \infty$

etwork address translation



Closing a TCP conne



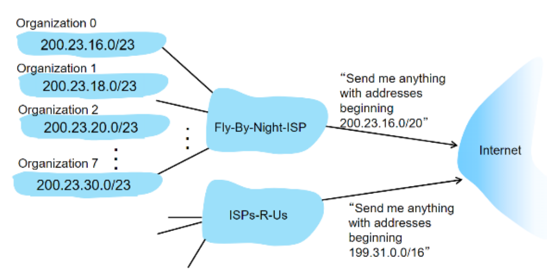
ICMP: internet control message protocol

- used by hosts and routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP messages carried in IP datagrams
- *ICMP message:* type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

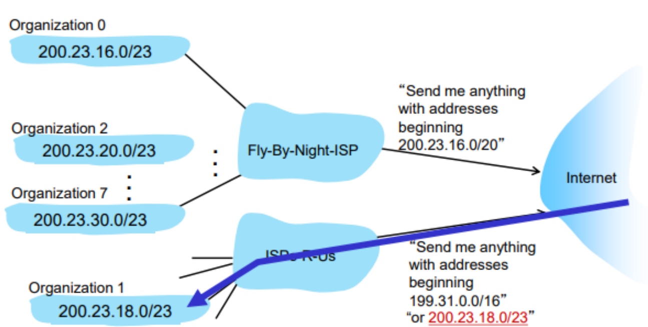
Hierarchical addressing: route aggregation

hierarchical addressing allows efficient advertisement of routing information:

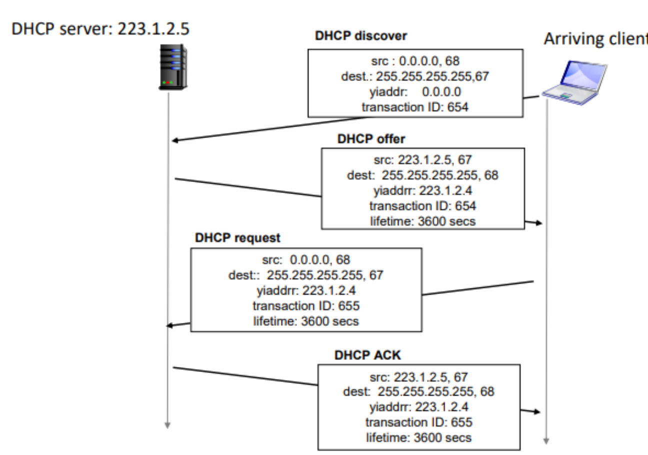


Hierarchical addressing: more specific routes

- Organization 1 moves from Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us now advertises a more specific route to Organization 1



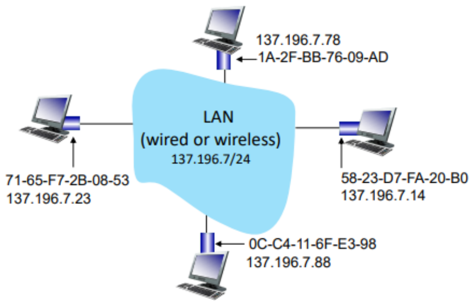
DHCP client-server scenario



นาย วัฒชัย เตชะลือ Sec53 65015137

MAC addresses

each interface on LAN
- has unique 48-bit MAC address
- has a locally unique 32-bit IP address (as we've seen)

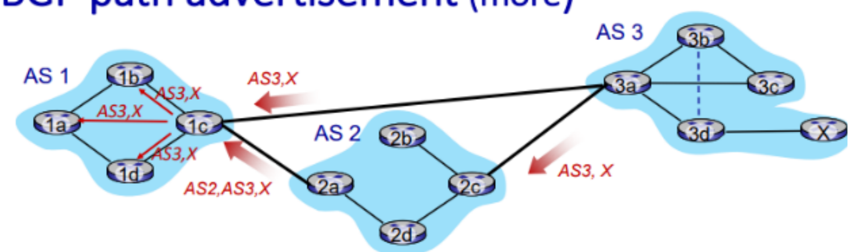


MAC addresses

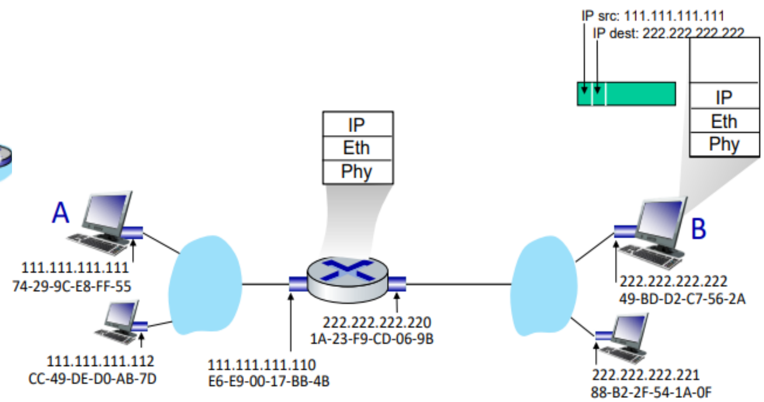
- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
  - e.g.: 128.119.40.136
- MAC (or LAN or physical or Ethernet) address:
  - function: used "locally" to get frame from one interface to another physically-connected interface (same subnet, in IP-addressing sense)
  - 48-bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation (each "numeral" represents 4 bits)

BGP path advertisement (more)



gateway router may learn about multiple paths to destination:
- AS1 gateway router 1c learns path *AS2,AS3,X* from 2a
- AS1 gateway router 1c learns path *AS3,X* from 3a
- based on *policy*, AS1 gateway router 1c chooses path *AS3,X* and advertises path within AS1 via iBGP

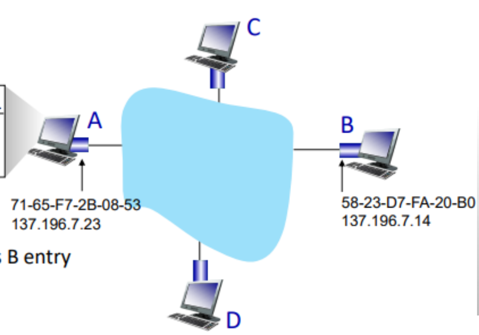


ARP protocol in action

example: A wants to send datagram to B
- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

ARP table in A

IP addr	MAC addr	TTL
137.196.7.14	58-23-D7-FA-20-B0	500



3 A receives B's reply, adds B entry into its local ARP table