CPE 426 Computer Networks

Chapter 1: Introduction

Review 1:Data Communications

Course Outlines

- ดูใน Sheet
- สามารถ Download ได้
  - http://cpe.rsu.ac.th/ut

TOPICS

- **1. Communication/NW Model**
  - Ch.5.1-5.5
- **2. Communication Protocols OSI and TCP/IP**
  - Ch.1.1-1.10
- **3. Communication/NW Topology**
  - Ch. 13.8

TOPICS

- **4. Signal/Power/Loss**
  - Ch.6.1-6.10
- **5. Data Coding(Line Coding)**
  - Ch.6.11-6.20
- **6. Trasmission Media/Noise/Channel Capacity**
  - Ch.7.1-7.10 & 7.20-7.22

TOPICS

- **7. Multiplexing & DSL**
  - Ch.11.1-11.3 & 12.1-12.7
- **8. Asynchronous Communication**
  - Ch.9.1-9.8
- **9. Synchronous Communication**
  - Ch.9.9-9.13

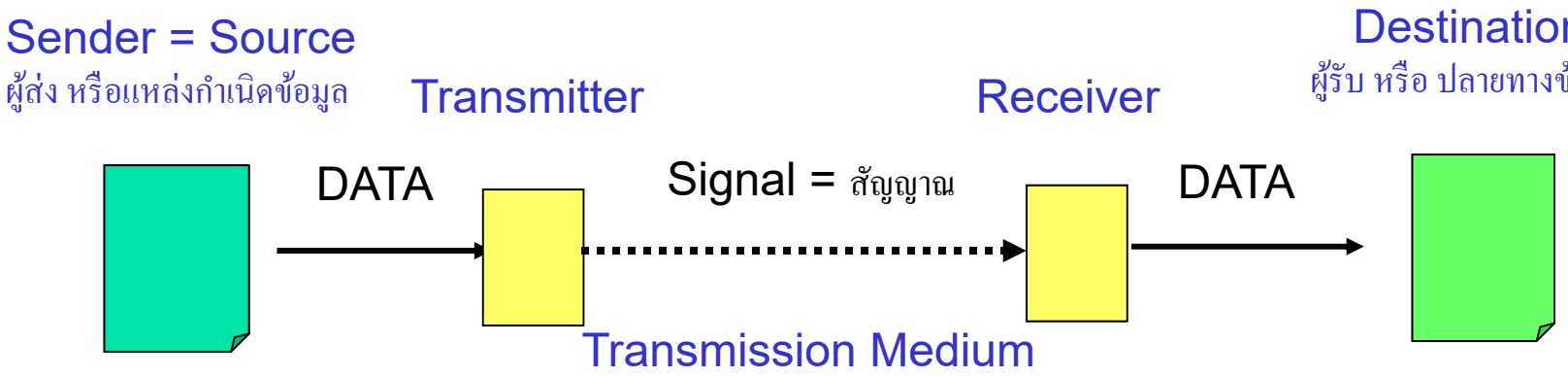
TOPICS

- **10. Flow Control/Error Control/ARQ**
  - Ch. 8.12-8.15
- **11. Circuit vs Packet Switching NW**
  - Ch. 3.1-3.5 % 13.1-13.5
- **ALSO Reference From CPE 326 (Stalling Book)**

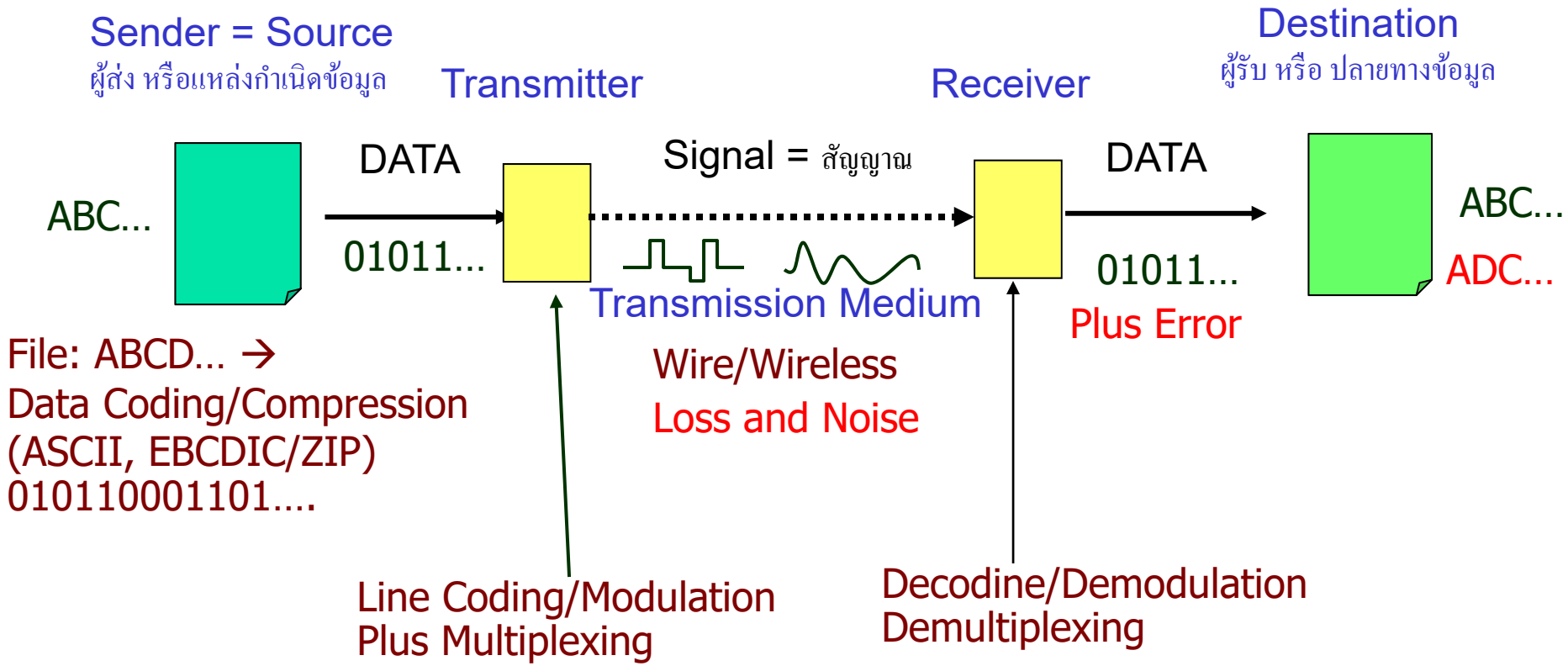
Review

- เพื่อให้รู้ว่าเรียนอะไรไปแล้วบ้าง
- เน้นพื้นฐานที่ต้องใช้หรือต้องเข้าใจ เพื่อต่อในวิชานี้
  - IP Technologies
- Layer 1 และ 2 เป็นส่วนใหญ่ และ Concept ของ Network(Layer 3)

การสื่อสาร ประกอบด้วย 2 Entity



Data comm.Model มี 5 ส่วน



ASCII Code

- **American Standard Code for Information Interchange**
  - ASCII includes definitions for 128 characters: 33 are non-printing control characters (now mostly obsolete) that affect how text and space is processed; 94 are printable characters, and the space is considered an invisible graphic. The most commonly used character encoding on the World Wide Web was US-ASCII until December 2007, when it was surpassed by UTF-8

USASCII code chart

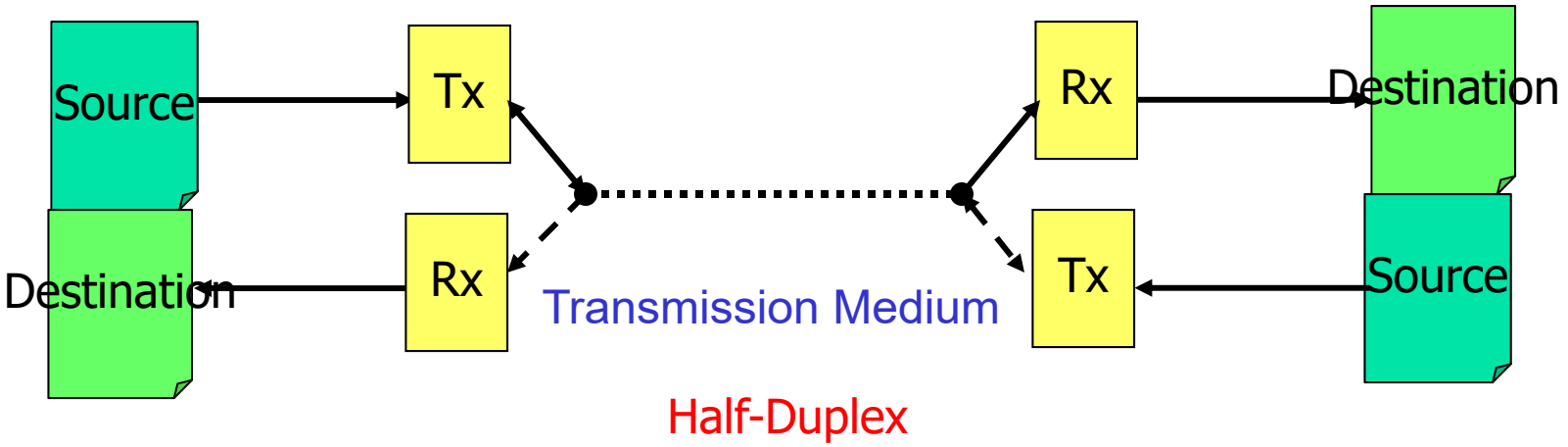
b_7 b_6 b_5 → / Bits ↓	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 b_3 b_2 b_1 Row ↓ / Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

Mode ของการสื่อสาร

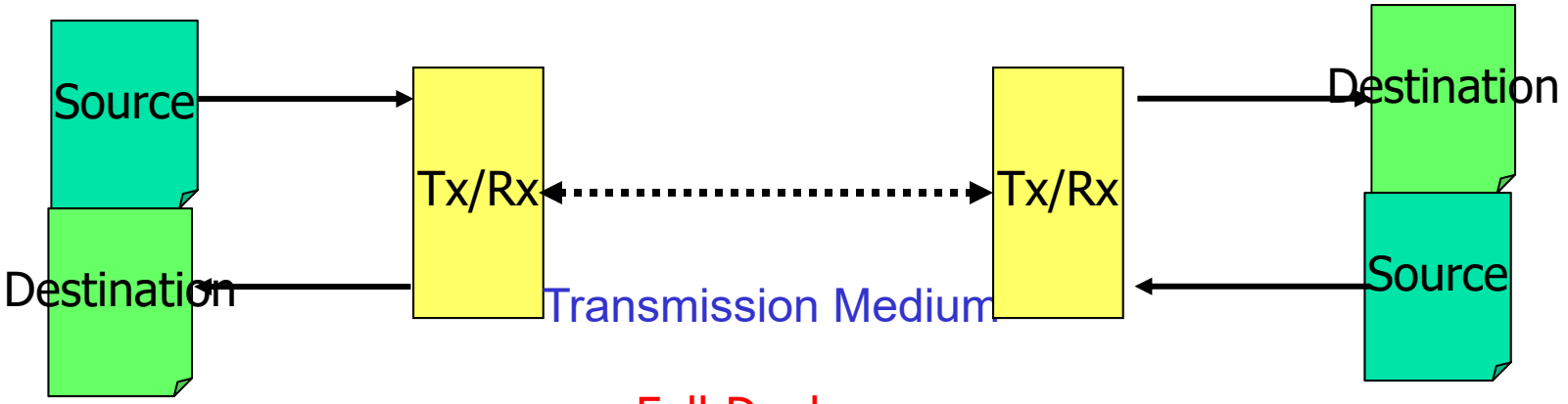
- **Data Communication Model ที่กล่าวถึงใช้สำหรับการสื่อสารสองคน**
  - ถ้ามีวงจรรับและส่งแยกจากกัน โดยใช้ Transmission Medium คนละตัว
    - Simplex
  - ถ้าใช้ Transmission อันเดียวกัน
    - Duplex
    - ถ้าสื่อสารสองทางได้พร้อมกัน
      - Full-Duplex
    - ถ้าสื่อสารสองทางไม่พร้อมกัน
      - Half-Duplex

Simplex
บางครั้งเรียก 4-wire Duplex

Duplex



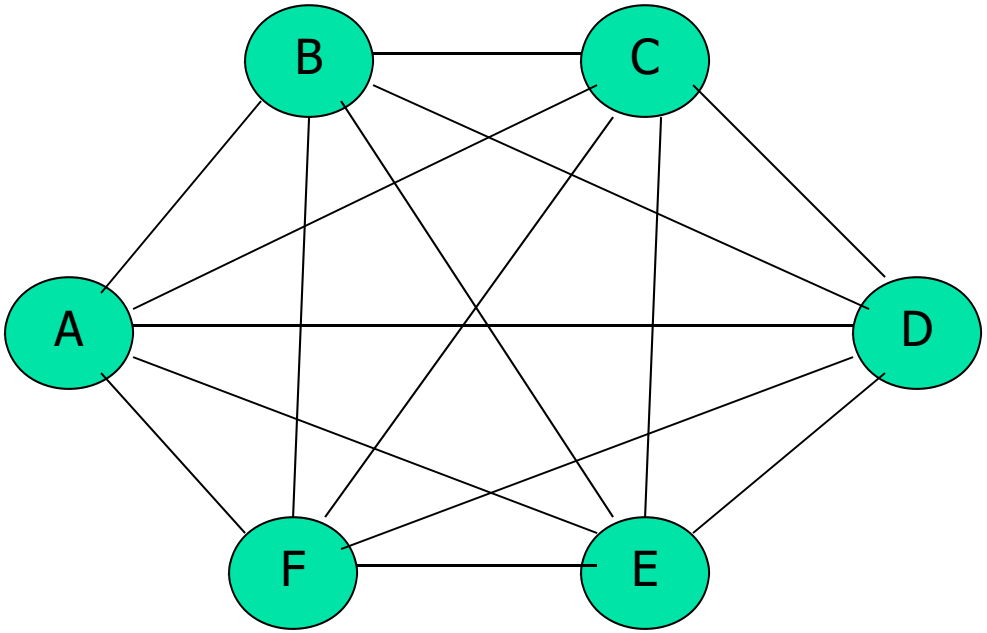
Half-Duplex



Full-Duplex

ถ้าเราต้องการสื่อสารมากกว่า 2 คน

- ใช้วงจร(Duplex)ดังกล่าวตามจำนวนคู่ของการสื่อสาร = Full Mesh Topology



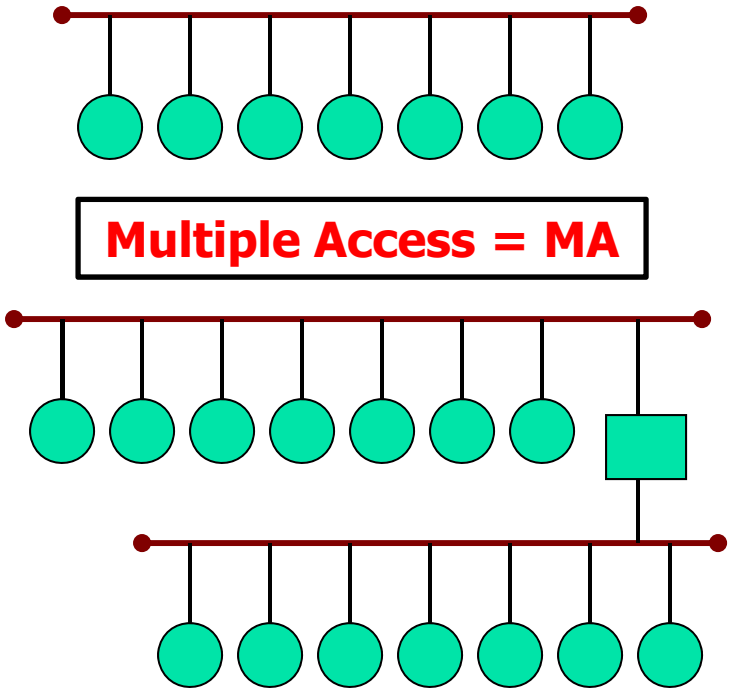
จำนวนวงจร
= n(n-1)/2
=O(n^2)

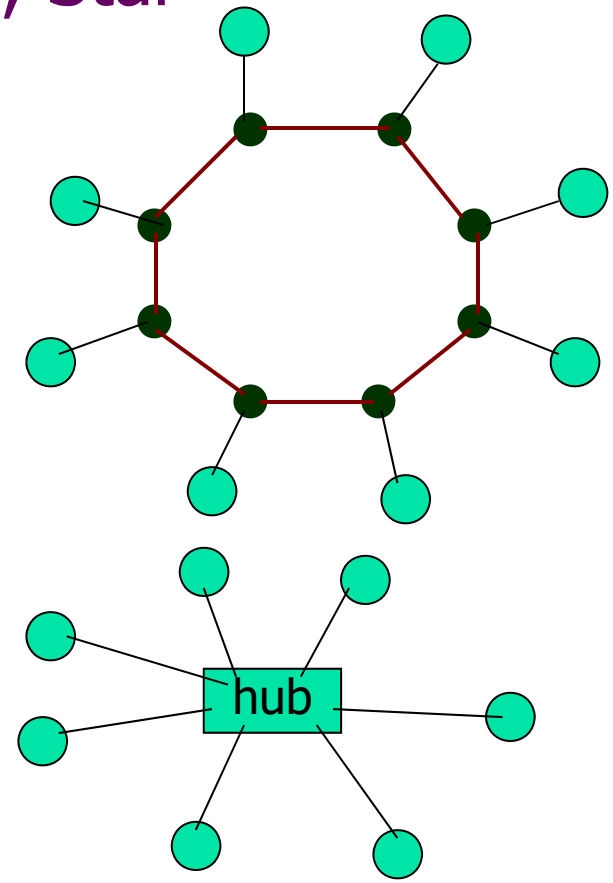
ราคาแพงมากถ้า
n มีค่าสูง
= O(n^2)

Topology ในภาษา Network คือรูปแบบการเชื่อมต่อของอุปกรณ์ต่างๆเข้าด้วยกัน

วิธีแก้คือ Share Medium และทำ Multiple Access Control

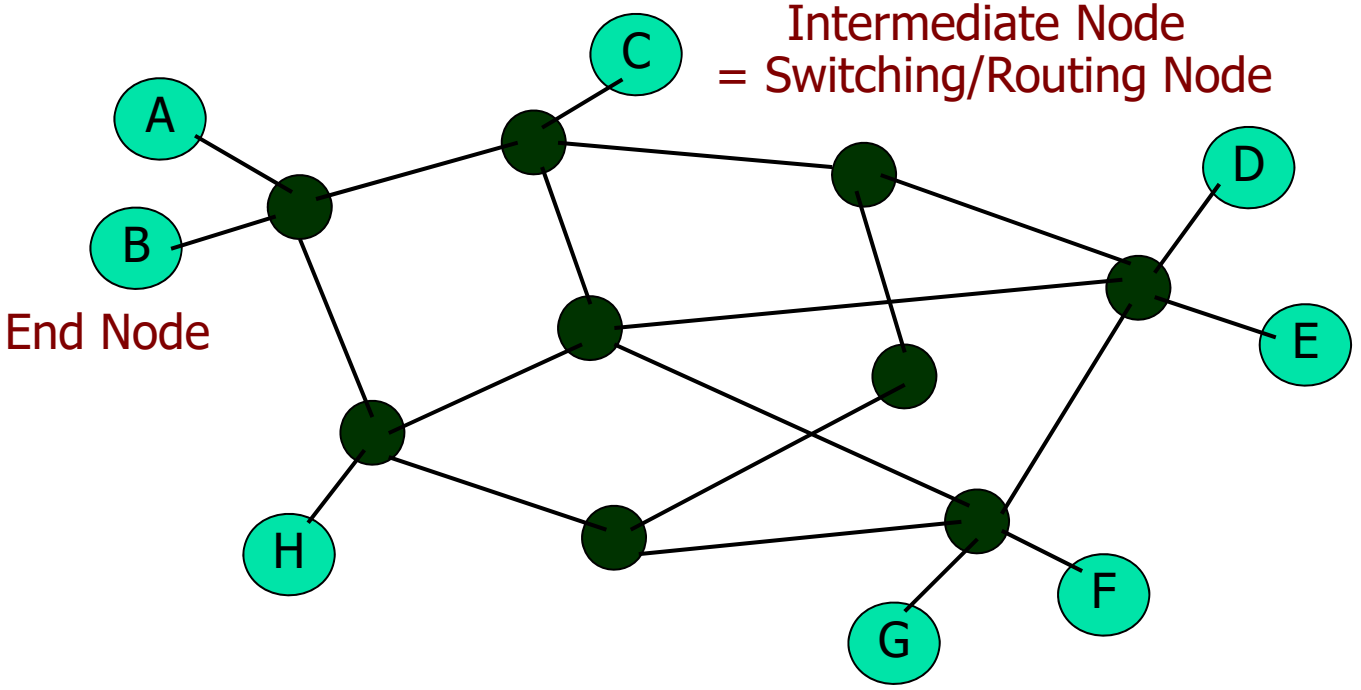
- ใน LAN จะใช้ Topology 3 แบบที่สำคัญ
  - Bus (และ Tree), Ring, Star





วิธีแก้คือ Share Medium และทำ Multiple Access Control

- ใน **WAN** มักจะเป็น **Partial Mesh**
  - Medium จัดได้ว่าเป็น Statistical **Time Division Multiplexing** แบบหนึ่ง



การ Share Medium

- **ต้องมีการควบคุม = Medium Access Control**
- **End Node จะต้องมีการกำหนดชื่อหรือ Address สำหรับอ้างอิง หรือกำหนด Circuit Number**
- **Intermediate Node จะใช้หมายเลขอ้างอิงดังกล่าวในการตัดสินใจส่งข้อมูลต่อออกไป(Forwarding)**
- **ดังนั้น**
  - 1. Data ที่ส่งจะต้องแปะส่วนหัว (Header) ด้วยข้อมูลต่างๆของ Address และการ Control เราเรียกว่าเป็นการทำ Encapsulation ผลลัพธ์ที่ได้เรียกว่า Frame
  - 2. ที่ส่วนท้ายของ Frame จะมีการต่อด้วยข้อมูลช่วยตรวจจับความผิดพลาด (Error Detection) มักจะเป็น CRC Code เรียก Frame Check Sequence(FCS)
  - 3. ก่อนหน้าส่วน Header และหลัง FCS อาจจะมีการเติมบิตสำหรับช่วยตรวจจับหัวและท้ายของ Frame (Frame Delimiter: Pre-amble/Post-amble)
  - 4. สำคัญที่สุดต้องมีการกำหนดกฎเกณฑ์ต่างๆเหล่านี้ให้เป็นมาตรฐาน คือกำหนดเป็น Protocol ของการสื่อสาร

LAN vs WAN Technologies

- **LAN มักจะใช้การ Share Medium แบบ Contention ดังนั้นจะต้องมีขบวนการควบคุมการทำ Multiple Access**
  - Topology ที่เหมาะสมคือ Bus, Ring, Star
- **WAN จะ Share Medium เช่นกัน แต่มักจะใช้วิธีของ Synchronous Multiplexing (TDM) ใน Circuit Switching Networkหรือ Statistical Multiplexing (ใช้ใน Packet Switching Network)**
  - Topology ที่เหมาะสมคือ Mesh Network และมักจะเป็น Partial Mesh
- **Internetworking Technologies มักจะถูกใช้ในการเชื่อมต่อระหว่าง LAN ผ่าน WAN Network**
  - ที่นิยมคือ Internet (IP Network)

Protocol and Protocol Architecture

- **Protocol เป็นตัวกำหนดกฎเกณฑ์สำหรับการสื่อสาร**
- **ถ้ากำหนดเป็นมาตรฐาน หรือ Standard การสื่อสารจะทำได้ง่ายระหว่างอุปกรณ์ที่ต่างกัน**
- **ประกอบด้วย**
  - Syntax
    - Data formats = รูปแบบของข้อมูล, เฟรม, การเข้ารหัส
    - Signal levels=ลักษณะของสัญญาณที่แทนข้อมูล
  - Semantics
    - Control information=การควบคุมการสื่อสาร
    - Error handling=การจัดการกับ Error
  - Timing
    - Speed matching=กำหนดอัตราการส่ง
    - Sequencing=กำหนดลำดับของข้อมูล

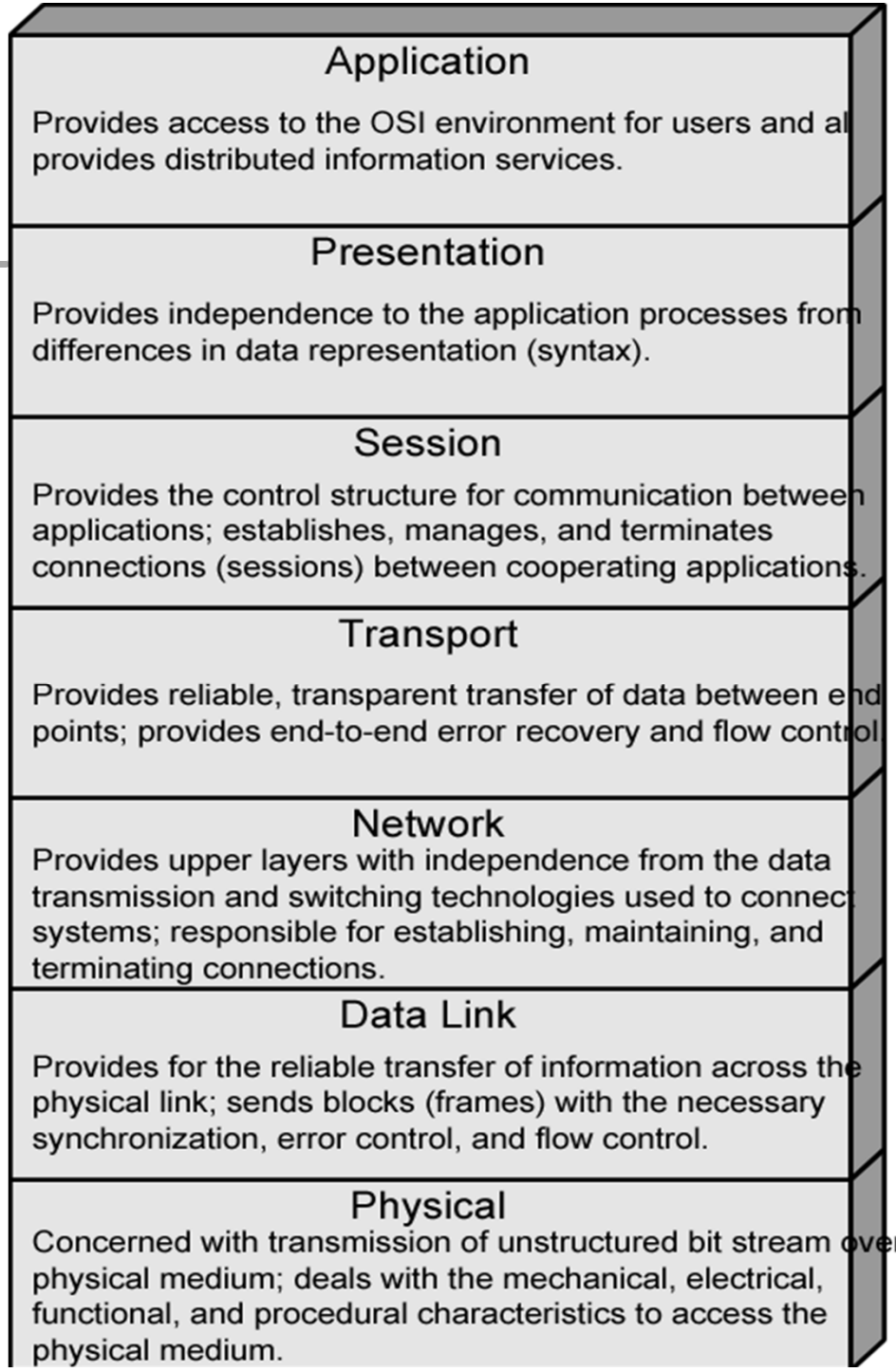
Protocol Architecture (Protocol Stack)

- เนื่องจากการสื่อสารเป็นเรื่องที่สลับซับซ้อน เราแบ่งการสื่อสารทั้งหมดออกเป็น Module
- แต่ละ Module มีหน้าที่เฉพาะของมัน
- แต่ละ Module จะมีการสื่อสารระหว่าง Module อื่น
- แต่ละ Module มี Protocol กำกับ
- ปกติจะแบ่งเป็นลำดับชั้น เรียก Protocol Stack หรือ Protocol Architecture

Protocol Architecture (Protocol Stack)

- **Protocol Architecture ที่เป็นมาตรฐานมีสองอัน**
- **7 Layer OSI Reference Model ของ ISO**
  - ปัจจุบันไม่ได้ใช้งานจริง แต่ใช้เป็น Reference
- **TCP/IP Protocol Suite (มี 5 ชั้น)**
  - การสื่อสารเกือบจะทั้งหมด
  - มาตรฐานของ Internet

7 Layer OSI Reference Model



7 Layer

- **Layer 1: Physical Layer**
  - ทำหน้าที่เชื่อมต่อผ่าน Physical Medium รับผิดชอบแปลงบิตเป็นสัญญาณ เรื่องของการ Interface, สายนำสัญญาณ ,มองเห็นข้อมูลในลักษณะ Bit Stream
- **Layer 2: Data Link Layer**
  - ประกอบข้อมูลเป็น Frame, รับผิดชอบในการสื่อสารผ่านแต่ละ Link ทำ Error Control, Flow Control ผ่าน Link
- **Layer 3: Network Layer**
  - รับผิดชอบในการส่งข้อมูลผ่าน Network, หาทิศทางข้อมูล, เชื่อมต่อกับ Layer บนเข้ากับ Network หลายๆ แบบ มองเห็นข้อมูลในลักษณะ Packet

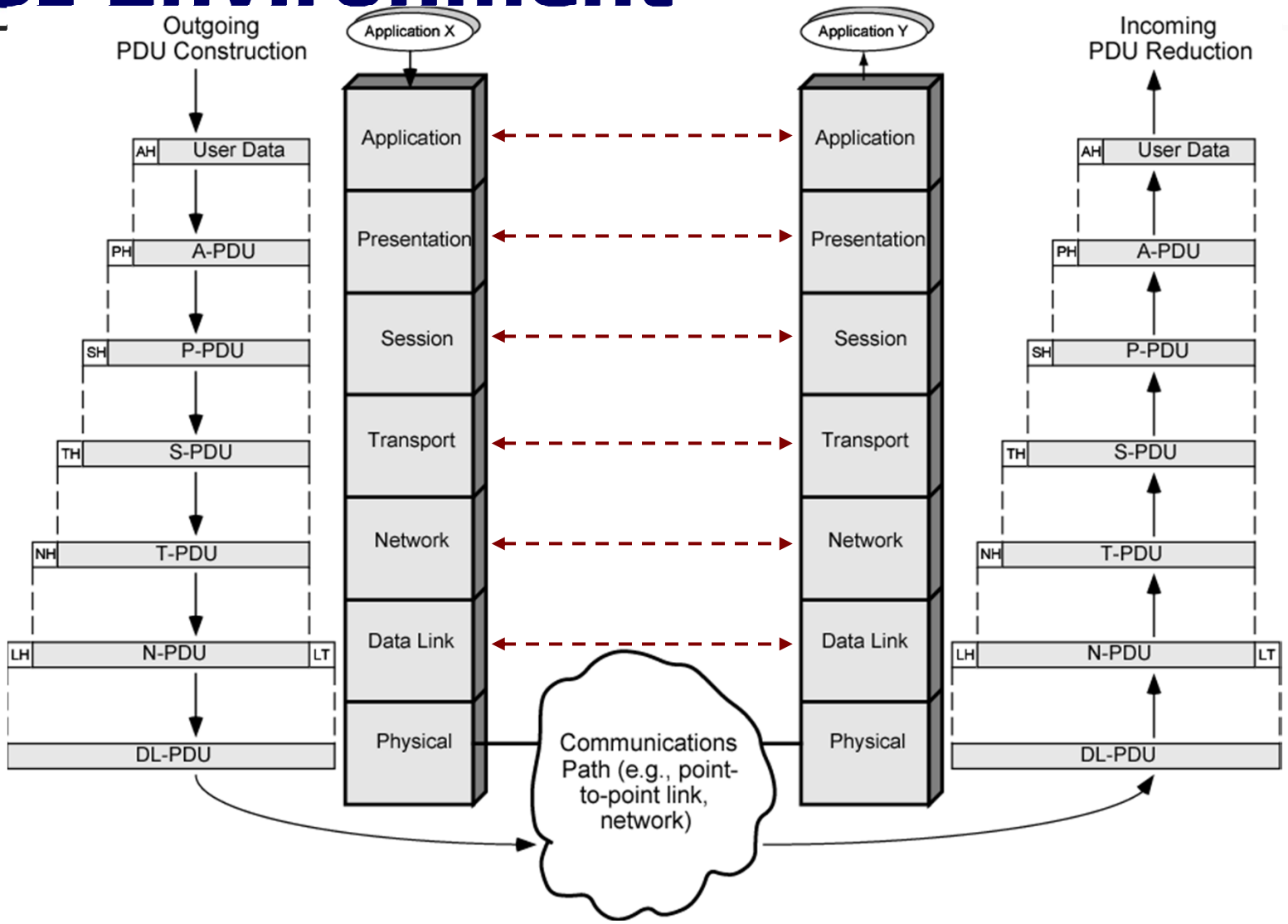
7 Layer

- **Layer 4: Transport Layer**
  - รับผิดชอบการส่งข้อมูลให้ถูกต้องจากต้นทางถึงปลายทาง(End-to-End), จัดการในเรื่อง Error และ Flow Control ในระดับต้นทางถึงปลายทาง ข้อมูลที่ส่งจะถูกแบ่งเป็น Segment
- **Layer 5: Session Layer**
  - ทำหน้าที่จัดตั้ง ดูแล การเชื่อมต่อ(Connection) ระหว่าง Applicationต้นทางและปลายทาง แบ่งการเชื่อมต่อสื่อสารออกเป็น Session

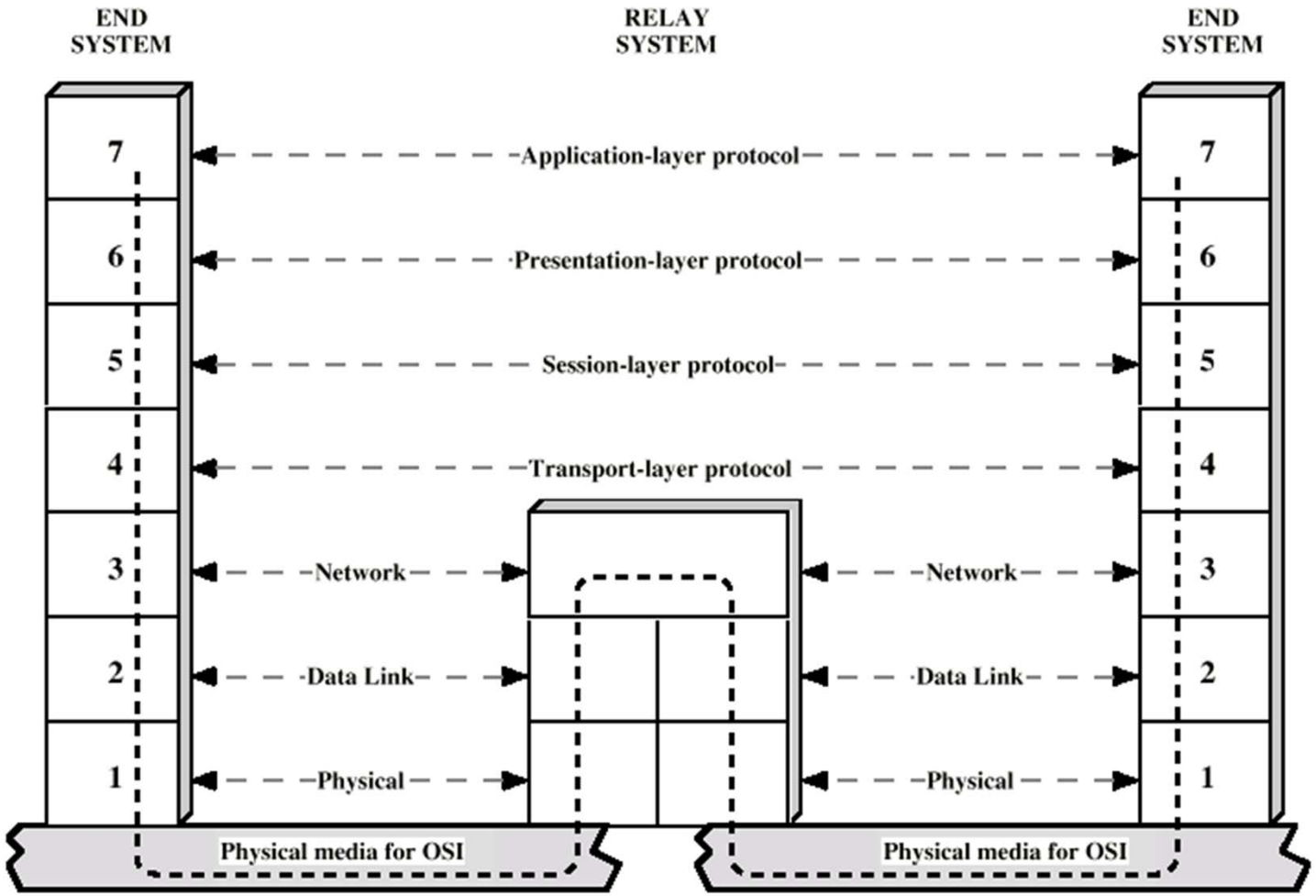
7 Layer

- **Layer 6: Presentation Layer**
  - รับผิดชอบในเรื่องรูปแบบและ Format ของข้อมูล การทำ Encryption รวมถึงการทำ Data Compression ให้อยู่ในรูปแบบที่สื่อสารได้
- **Layer 7: Application Layer**
  - ทำหน้าที่เชื่อมต่อกับ Application และผู้ใช้

OSI Environment



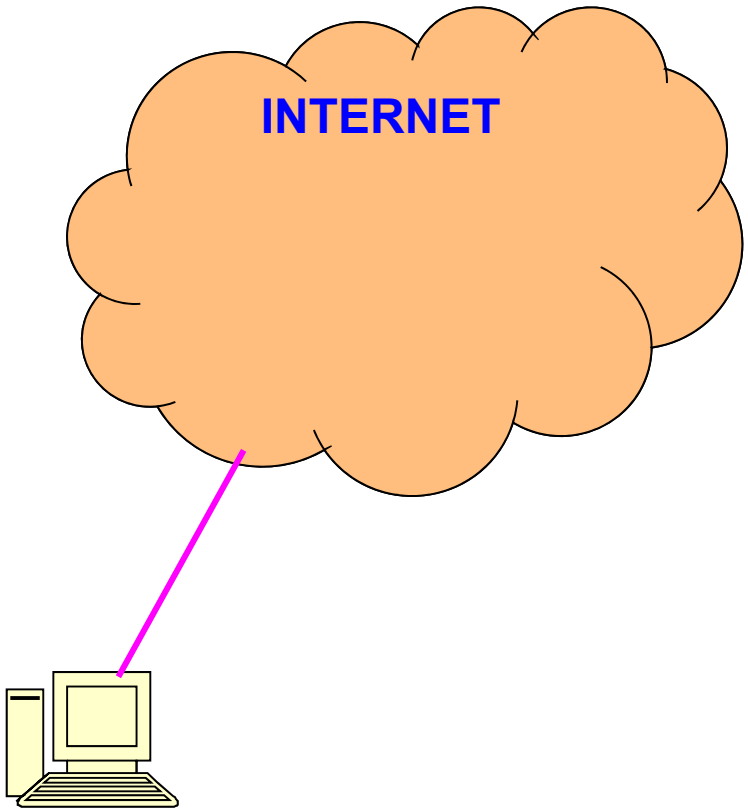
การเชื่อมต่อผ่าน Router

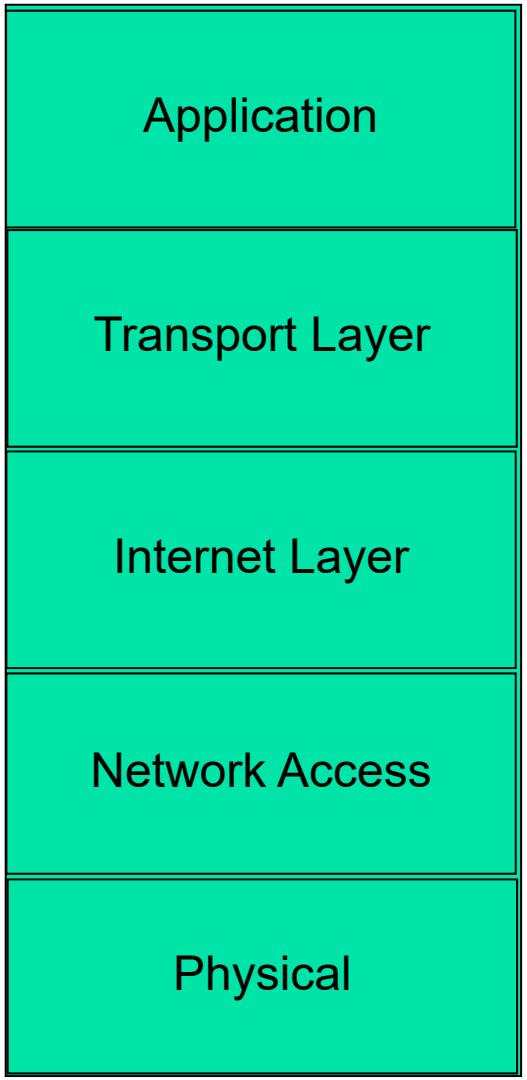


TCP/IP Protocol Architecture

- **Developed by the US Defense Advanced Research Project Agency (DARPA) for its packet switched network (ARPANET)**
- **Used by the global Internet**
- **No official model but a working one.**
  - Application layer
  - Host to host or transport layer
  - Internet layer
  - Network access layer
  - Physical layer

TCP/IP Protocol Architecture





Physical Layer

- Physical interface between data transmission device (e.g. computer) and transmission medium or network
- Characteristics of transmission medium
- Signal levels
- Data rates
- etc.

Network Access Layer

- Exchange of data between end system and network
- Destination address provision
- Invoking services like priority
- ปกติมาตรฐานของ TCP/IP จะไม่ครอบคลุมถึง Layer 1-2
- ทั่วไปเรานำ TCP/IP เป็น WAN และวางบน LAN คือ Ethernet

Internet Layer (IP)

- Systems may be attached to different networks
- Routing functions across multiple networks
- Implemented in end systems and routers
- คือ IP Protocol
  - มีการทำงานแบบ Datagram

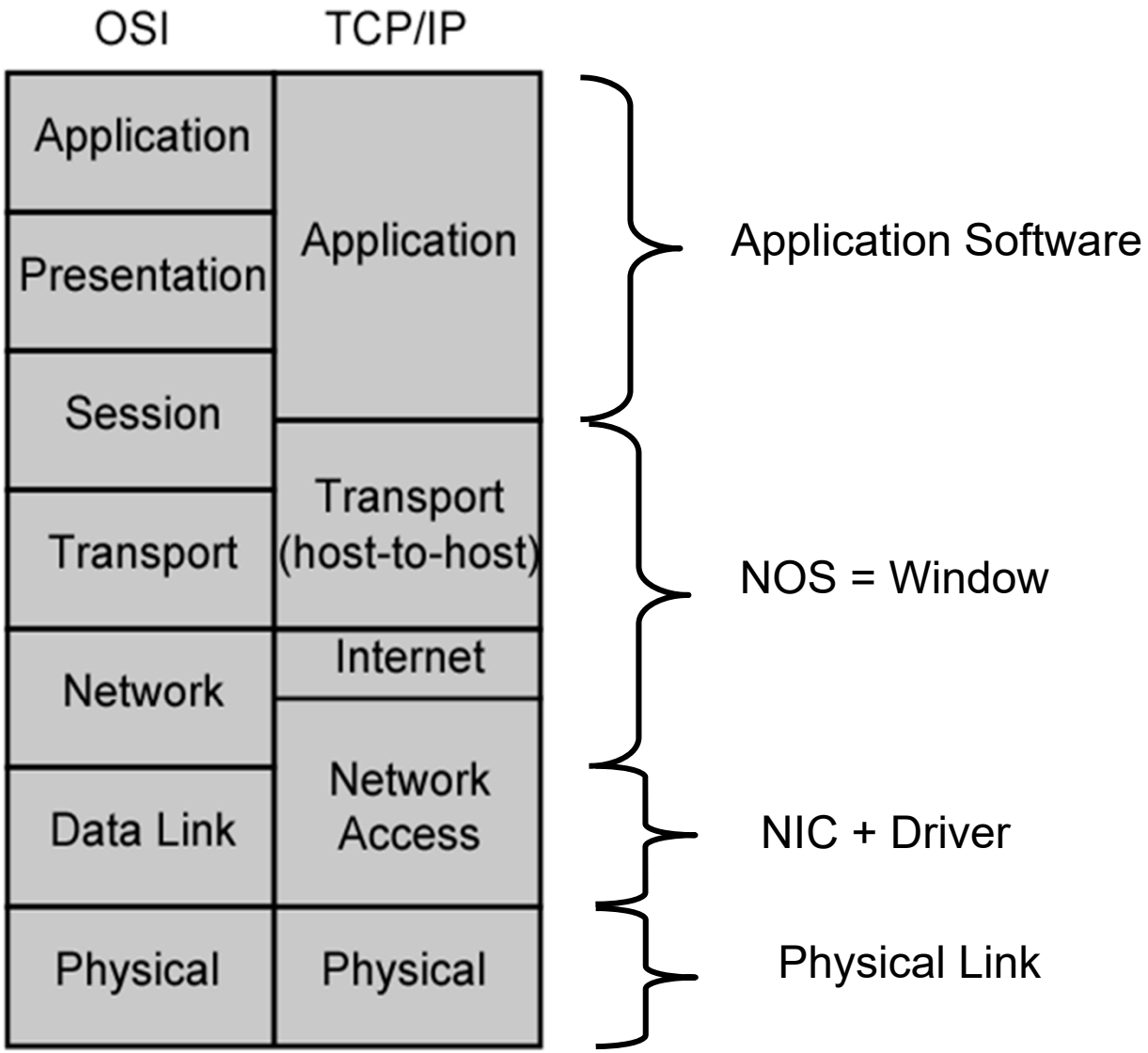
Transport Layer (TCP)

- **Reliable delivery of data**
- **Ordering of delivery**
- **ที่สำคัญมี 2 Protocol**
  - TCP = Transport Control Protocol
    - Connection Oriented
    - Guarantee Delivery
  - UDP = User Datagram Protocol
    - Connectionless
    - Best Effort

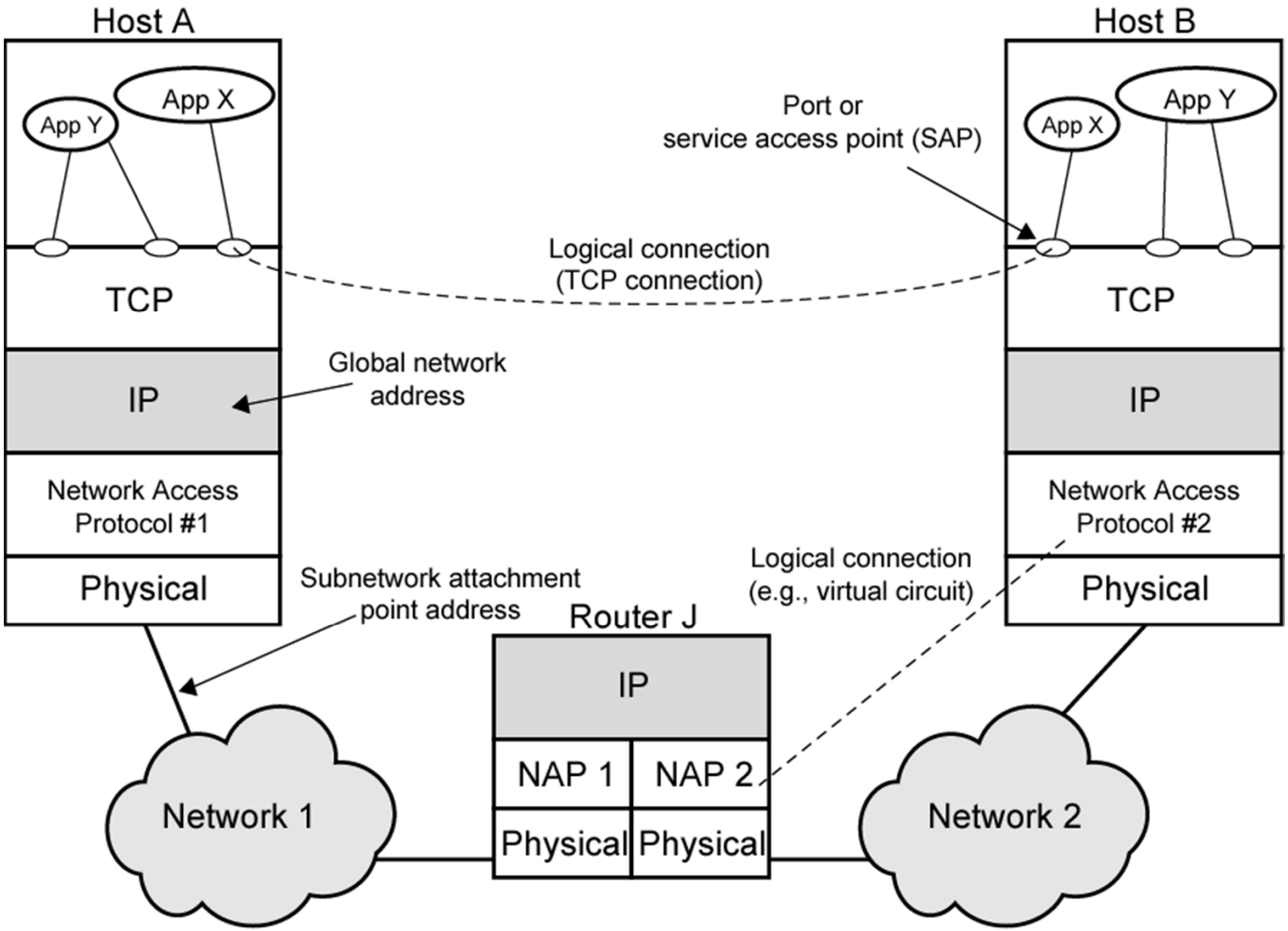
Application Layer

- Support for user applications
- e.g. http, SMPT

TCP/IP VS OSI


OSI
TCP/IP
Application
Presentation
Session
Transport
Network
Data Link
Physical
Application
Transport (host-to-host)
Internet
Network Access
Physical
Application Software
NOS = Window
NIC + Driver
Physical Link

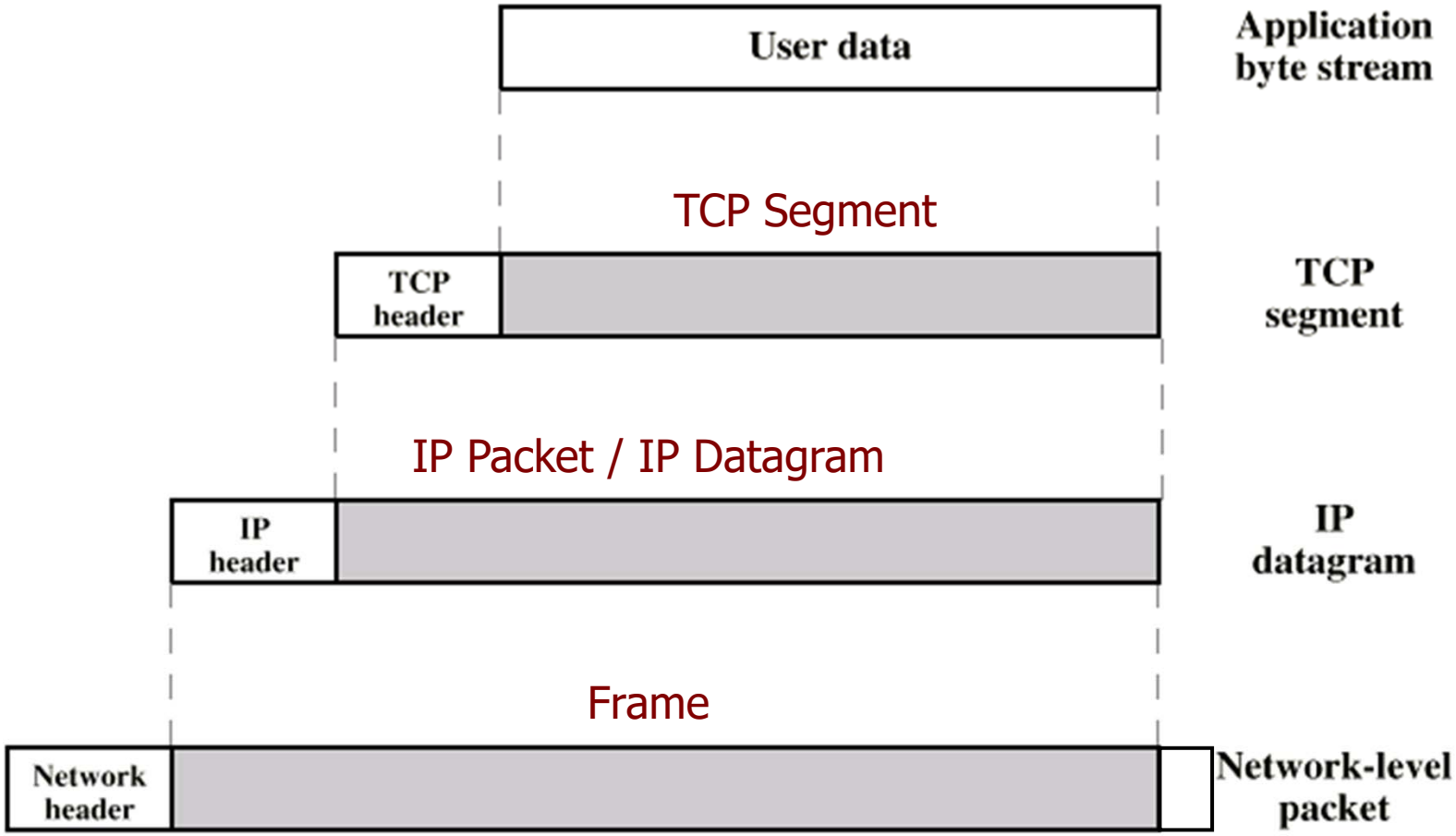
รูปแบบการเชื่อมต่อด้วย TCP/IP



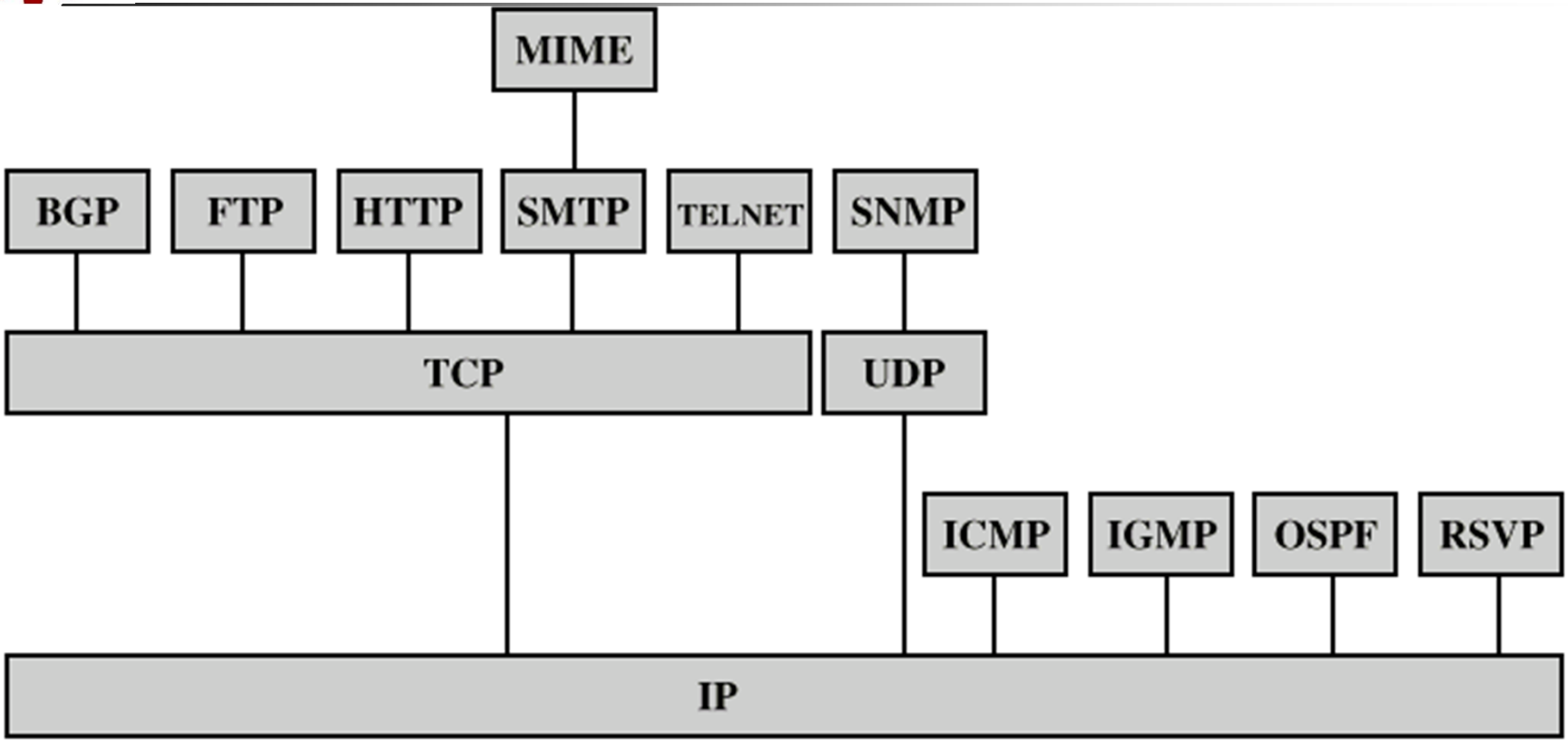
Addressing ใน TCP/IP

- **TCP Port หรือ UDP Port = 16 Bit**
- **IP Address, IPv4 = 32 Bit หมายเลขเครื่อง และหมายเลข Network**
- **Physical Hardware Address**
  - ถ้าใช้ TCP/IP บน Ethernet LAN อันนี้คือ Address ของ NIC หรือ MAC Address = 48 Bit

PDU = Protocol Data Unit



Protocol ที่สำคัญของ TCP/IP



BGP = Border Gateway Protocol
FTP = File Transfer Protocol
HTTP = Hypertext Transfer Protocol
ICMP = Internet Control Message Protocol
IGMP = Internet Group Management Protocol
IP = Internet Protocol
MIME = Multi-Purpose Internet Mail Extension
OSPF = Open Shortest Path First
RSVP = Resource ReSerVation Protocol
SMTP = Simple Mail Transfer Protocol
SNMP = Simple Network Management Protocol
TCP = Transmission Control Protocol
UDP = User Datagram Protocol

Standard

- **LAN:**
  - IEEE 802
  - Ethernet IEEE 802.3 มีย่อยอีกหลายตัว
  - WLAN IEEE 802.11, 802.11b, 802.11g, 802.11n,802.11i
  - PAN-Bluetooth IEEE 802.15
  - www.ieee.org
- **WAN**
  - มีหลายตัว ที่สำคัญมักจะถูกดูแลโดย OSI (ITU)
- **TCP/IP**
  - RFC = Request for Comments
  - มีมากกว่า 4000 RFCs อันใหม่จะแทนอันเก่า (Obsolete)
  - www.faqs.org/rfcs

Summary Physical Layer

- **Physical Layer จะสลับซับซ้อนที่สุด ปกติจะเกี่ยวกับไฟฟ้าสื่อสาร**
  - กำหนด Medium, Signal, Coding, Connector รวมถึงกระบวนการ
  - การสื่อสารจะถูกจำกัดจาก Layer นี้
    - Bit Rate/Baud Rate ~ Power, Noise, Distortion, Interference, Cross Talk
      - ที่สำคัญ SNR และ Eb/No
    - ขีดจำกัดตาม Channel Capacity

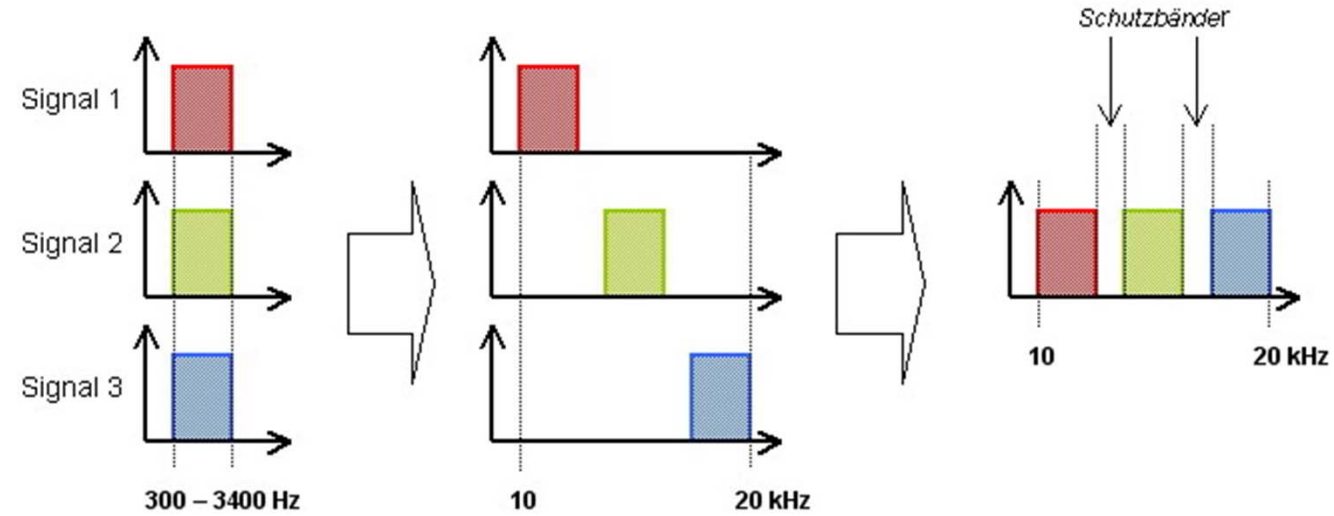
Line Coding

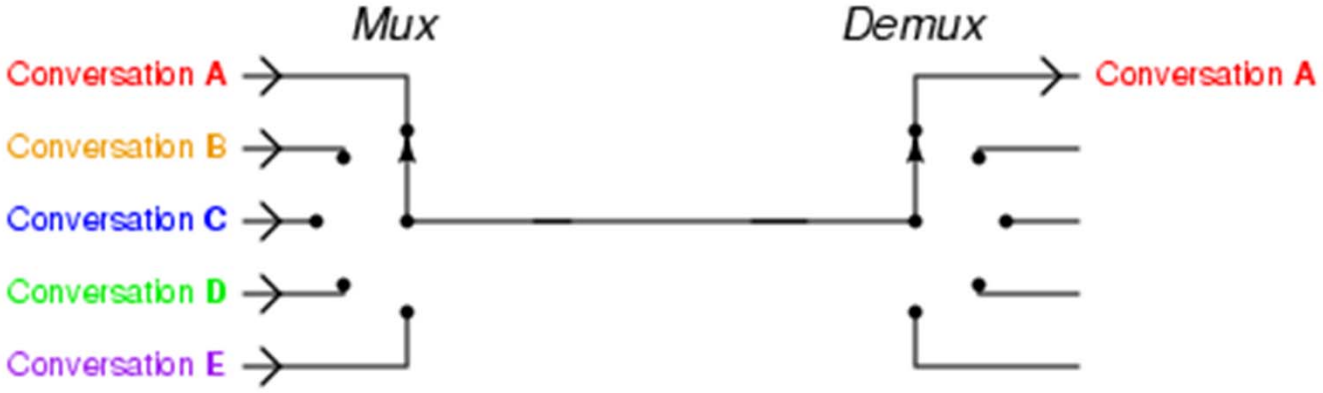
- **การส่ง Pulse เพื่อที่จะแทน Data แต่ละบิต**
  - Pulse 2 ระดับ = Binary Signal
  - M-ary Signal จะใช้ M ระดับ
- **ข้อควรคำนึง**
  - Average DC เป็นศูนย์หรือไม่
  - Signal Transition มากเพียงพอ
- **NRZ, AMI, Pseudoternary, Manchester, Differential Manchester และ อื่นๆ**
  - อาจจะร่วมกับการทำ Scrambling
    - HDB3, B8ZS

Multiplexing

- **เป็นวิธีการที่จะสามารถส่งสัญญาณได้หลายคู่บน Transmission Medium เดียวกัน**
  - FDM = Frequency Division Multiplexing
    - สัญญาณแต่ละคู่ใน Bandwidth (ช่วงความถี่) ต่างกัน
  - TDM = Time Division Multiplexing
    - สัญญาณแต่ละคู่ส่งที่เวลาต่างกัน
    - แบ่งเป็น
      - Synchronous TDM: แบ่งเวลาเป็น Channel ตามจำนวนคู่ คู่หนึ่งจะใช้ Channel เบอร์ที่กำหนดเท่านั้น
      - Statistical TDM แบ่งเป็น Channel เช่นกัน แต่ไม่กำหนด คู่ใดต้องการส่งข้อมูลให้จอง Channel เพื่อส่ง ดังนั้นในการส่งข้อมูลครั้งหนึ่งๆ อาจจะใช้ Channel แตกต่างกัน

FDM vs TDM


Schutzbänder
Signal 1
Signal 2
Signal 3
300 – 3400 Hz
10
20 kHz
10
20 kHz


Mux
Demux
Conversation A
Conversation B
Conversation C
Conversation D
Conversation E
Conversation A

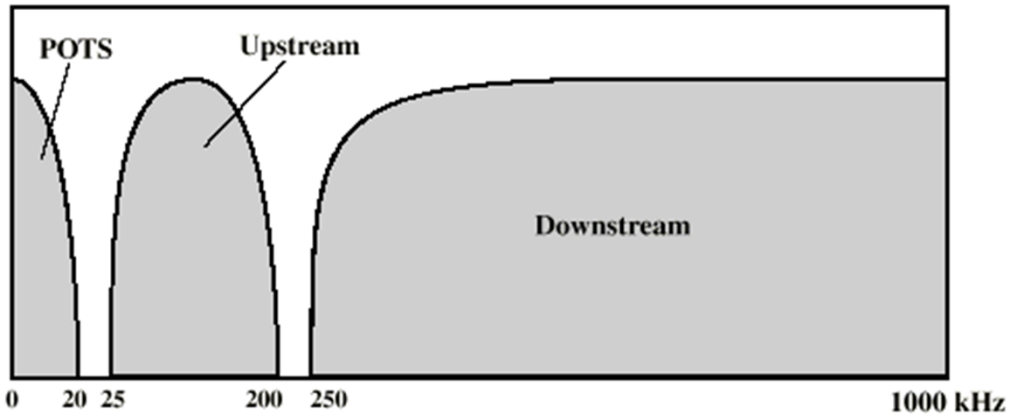
Statistical TDM Frame Formats

(a) Overall frame

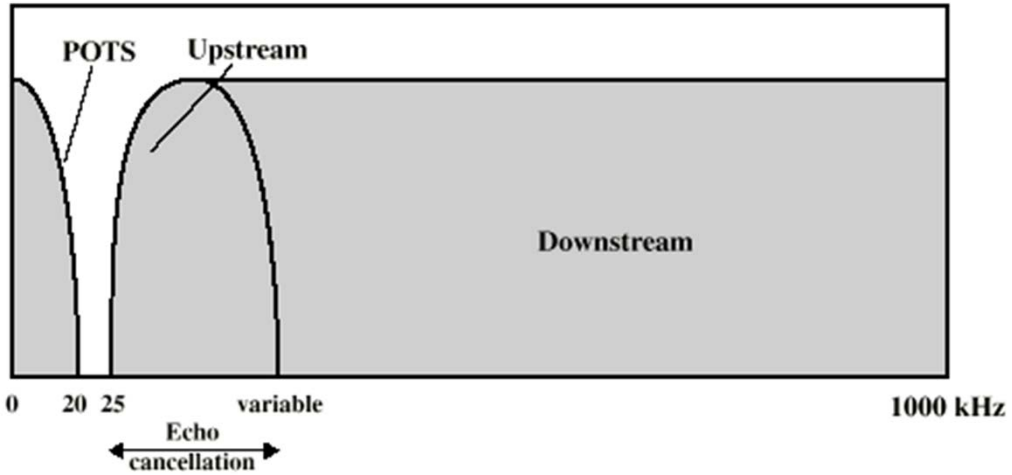
(b) Subframe with one source per frame

(c) Subframe with multiple sources per frame

ADSL Channel Configuration

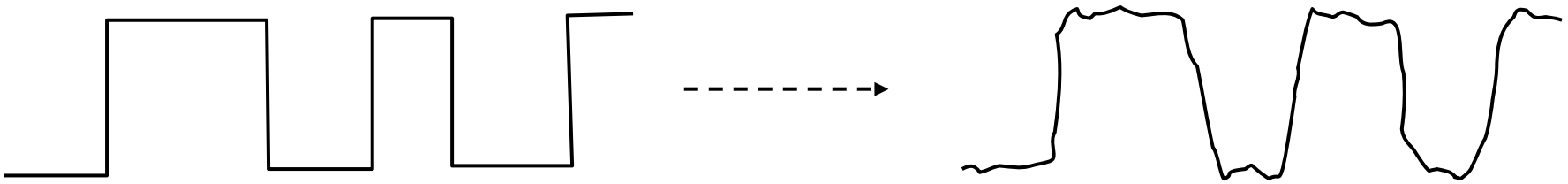


(a) Frequency-division multiplexing



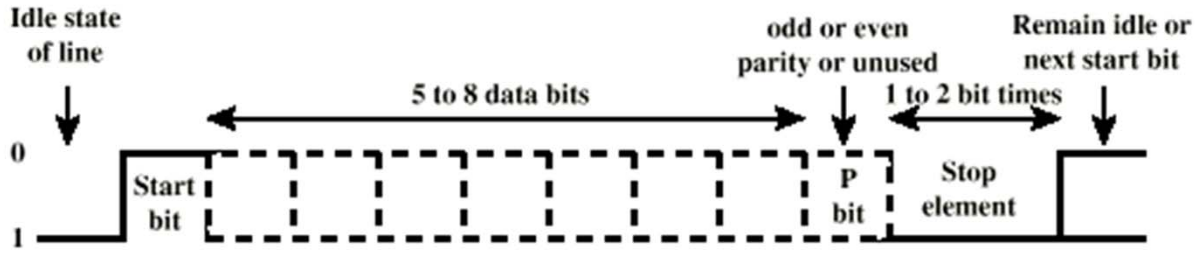
(b) Echo cancellation

Mode ของการส่งข้อมูล Digital

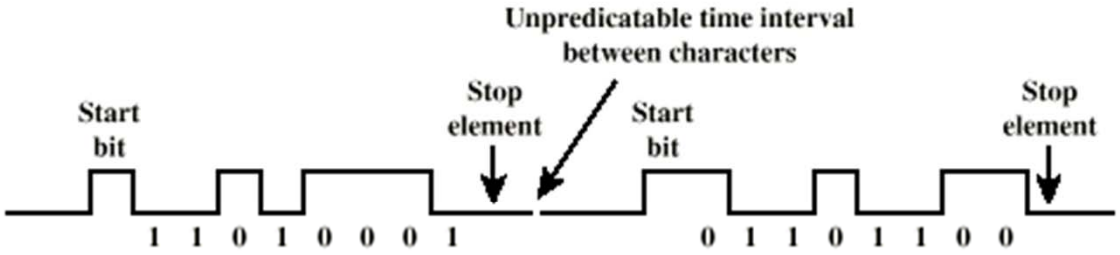


- **Timing problems require a mechanism to synchronize the transmitter and receiver**
- **Two solutions**
  - Asynchronous
  - Synchronous

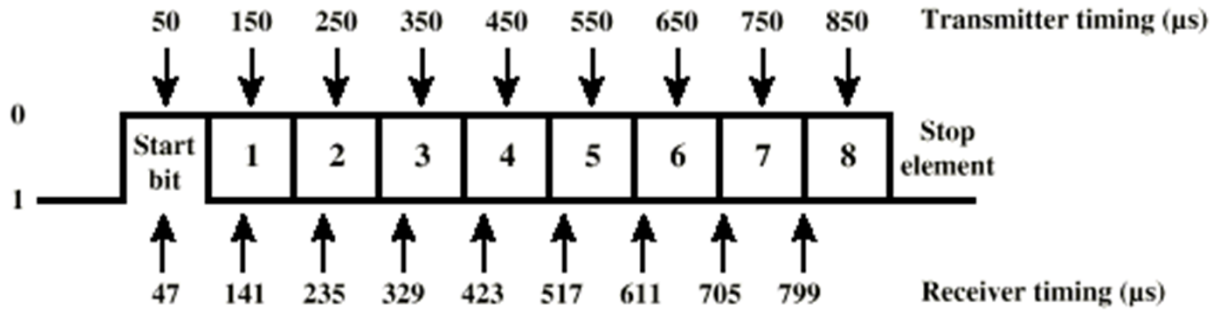
Asynchronous (diagram)



(a) Character format

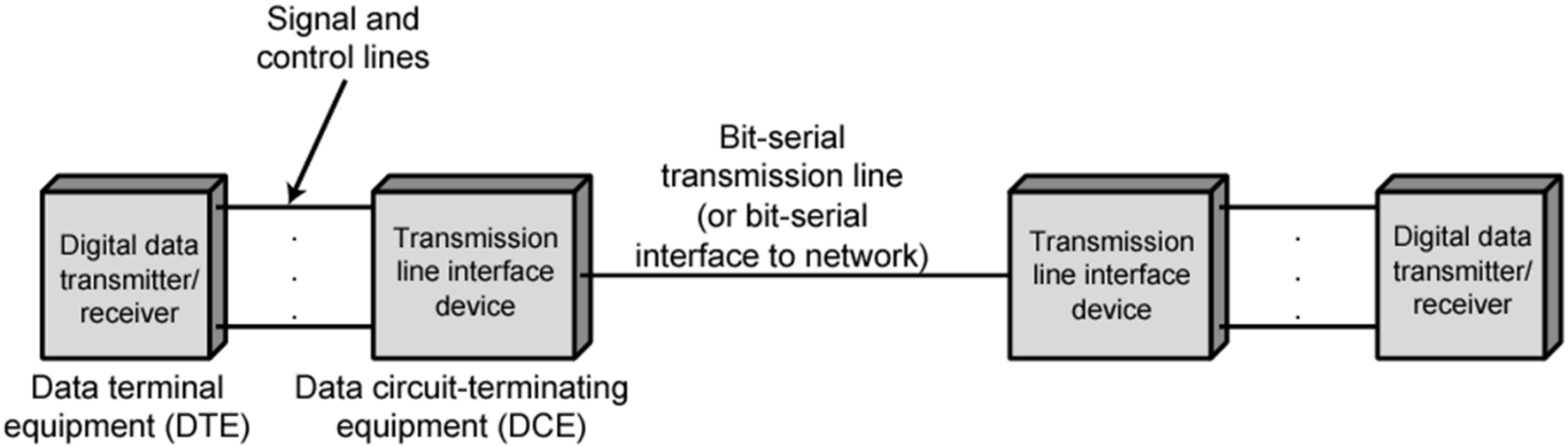


(b) 8-bit asynchronous character stream



(c) Effect of timing error

Data Communications Interfacing (DTE-DCE Concept)



(a) Generic interface to transmission medium

(b) Typical configuration

Mechanical Specification

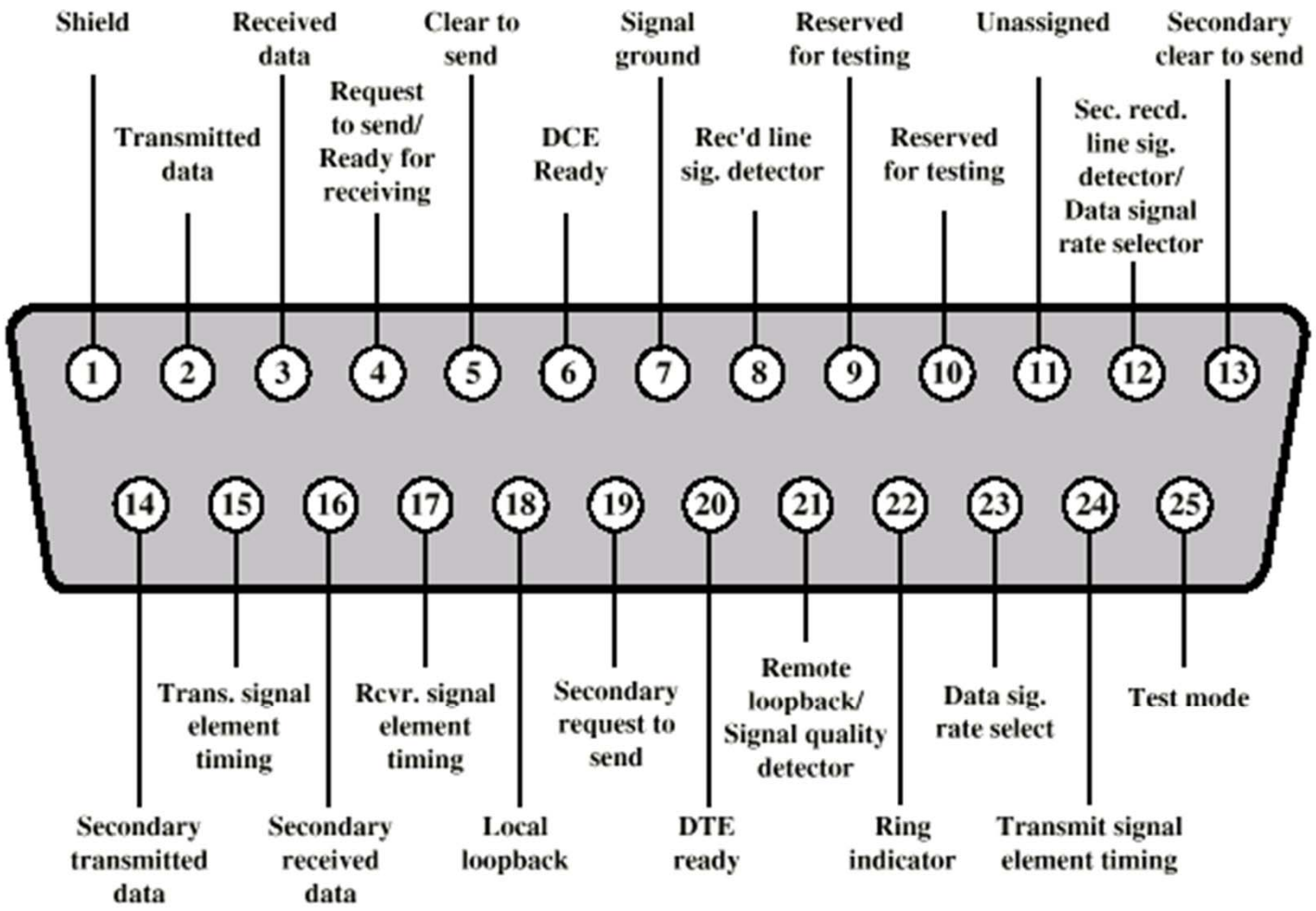
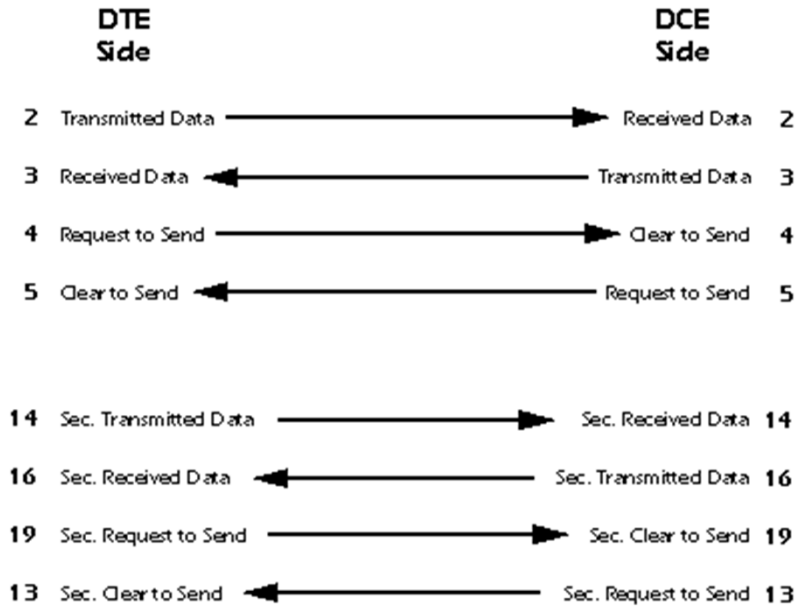


Figure 6.5 Pin Assignments for V.24/EIA-232 (DTE Connector Face)

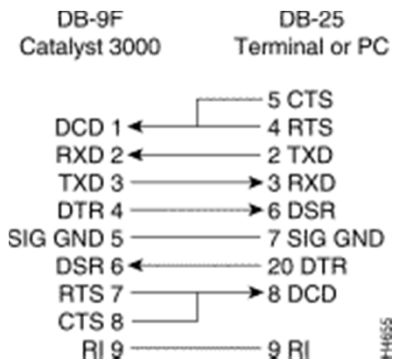
Electrical Specification

- Digital signals
- Values interpreted as data or control, depending on circuit
- Less than -3v is binary 1, more than +3v is binary 0 (NRZ-L)
- Signal rate < 20kbps
- Distance <15m
- For control, Less than-3v is off, +3v is on

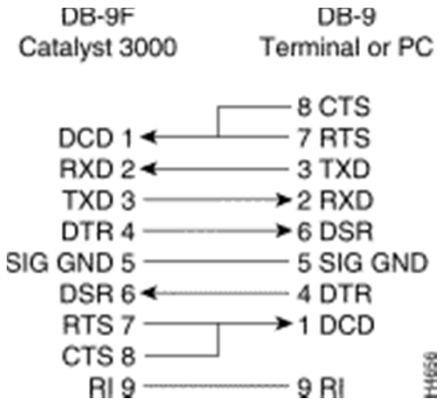


Null Modem: DTE to DTE

- **RS-232-C Null Modem Cable (for Terminal/PC with 25-pin Connector)**

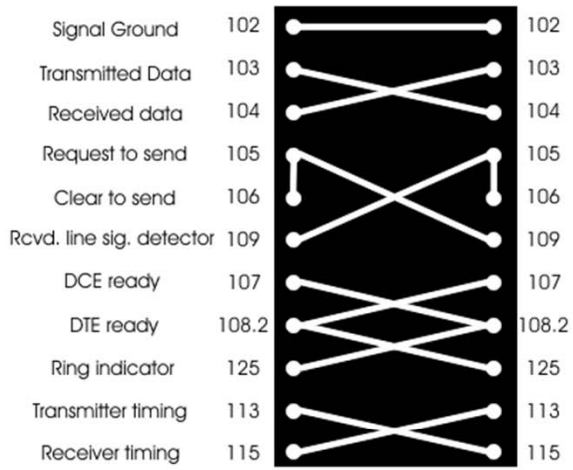


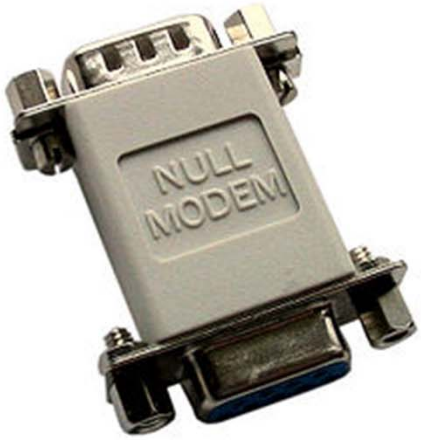
- **RS-232-C Null Modem Cable (for Terminal/PC with 9-pin Connector)**



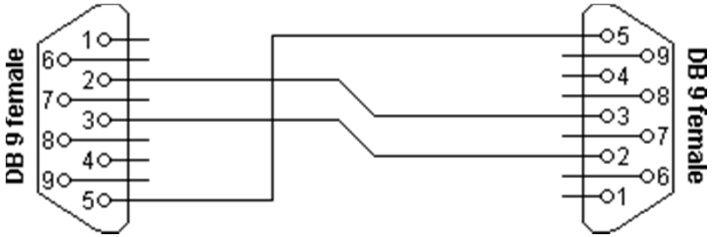
Null Modem

Example of a Null Modem

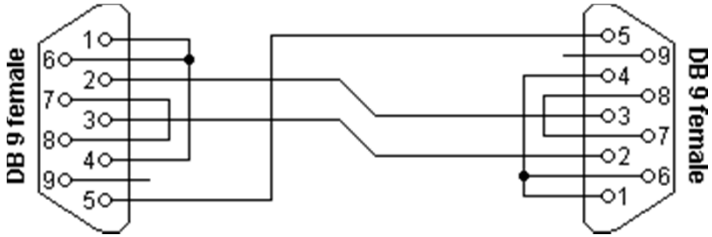




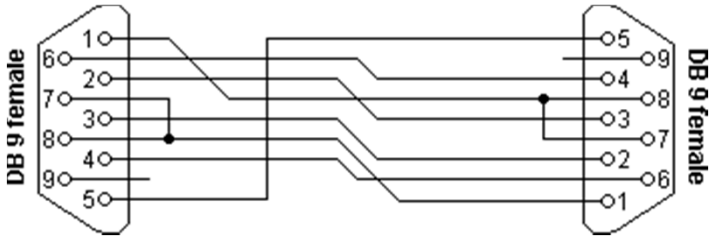
Summary Null Modem



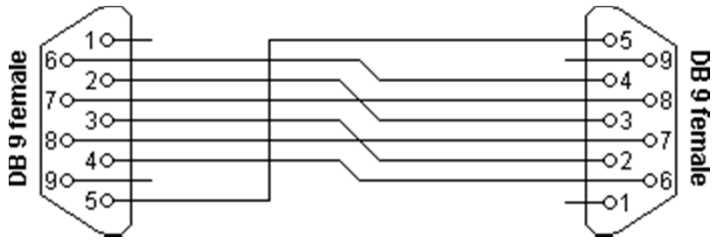
Simple Null Modem
without Handshaking



Null Modem
With Loop-Back Handshaking



Null Modem
With Partial Handshaking



Null Modem
With Full Handshaking

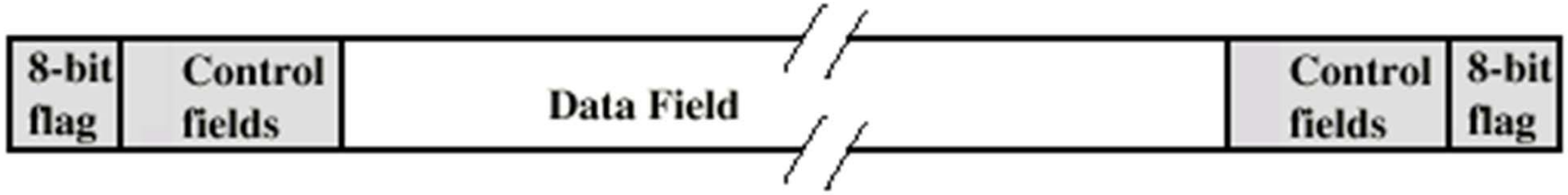
Synchronous - Bit Level

- **Block of data transmitted without start or stop bits**
- **Clocks must be synchronized**
- **Can use separate clock line**
  - Good over short distances
  - Subject to impairments
- **Embed clock signal in data**
  - Manchester encoding
  - Carrier frequency (analog)

Synchronous - Block Level

- **Need to indicate start and end of block**
- **Use preamble and postamble**
  - e.g. series of SYN (hex 16) characters
  - e.g. block of 11111111 patterns ending in 11111110
- **More efficient (lower overhead) than async**

Synchronous (diagram)



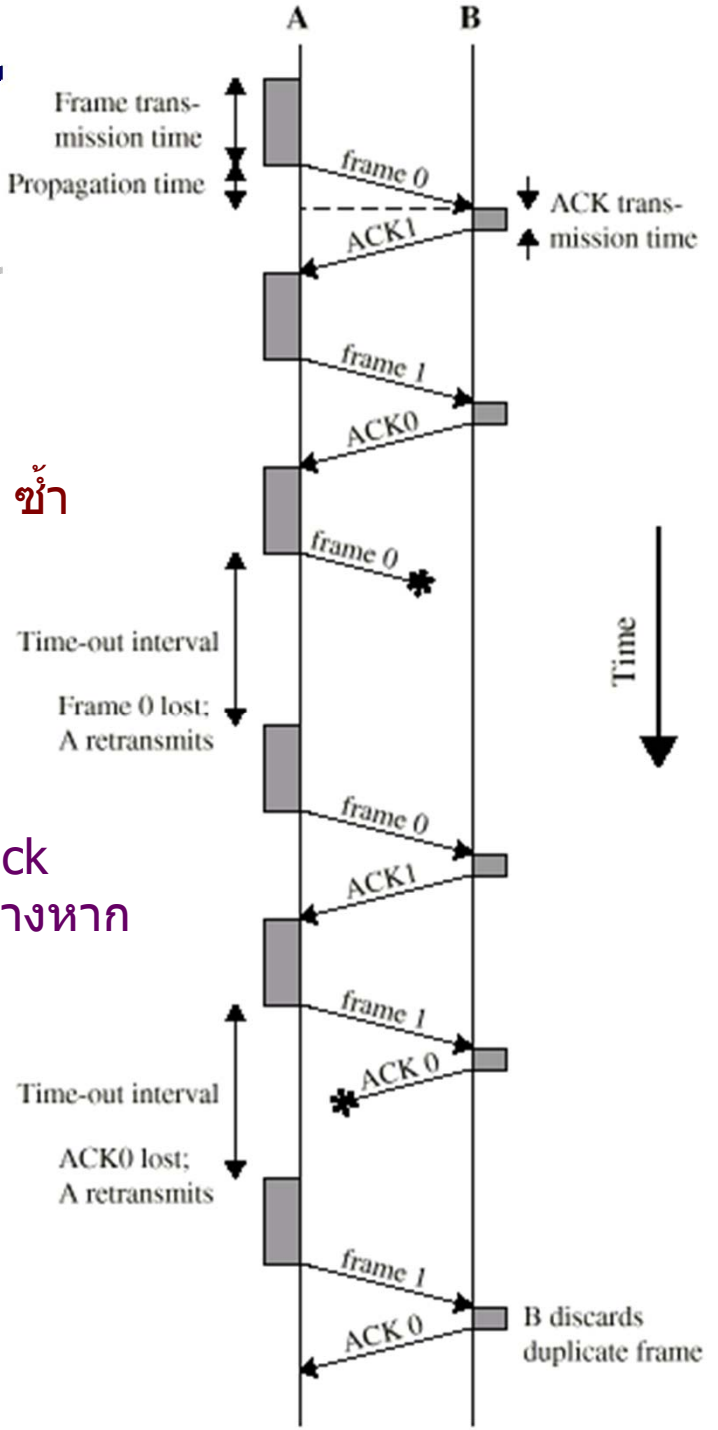
Flow Control/Error Control

- **เราควบคุมการไหลข้อมูลเพื่อไม่ให้ผู้ส่งส่งข้อมูลเกินกว่าผู้รับจะรับได้**
  - เมื่อส่งข้อมูลแล้ว ให้รอสัญญาณพร้อมที่จะรับข้อมูลอันต่อไปจากผู้รับ
- **ปกติ Mechanism นี้จะใช้ร่วมกับ Error Control โดยเมื่อมี Error จะใช้วิธีการ Retransmission**
  - เราเรียกรวมว่า ARQ = Automatic Repeat Request

Stop and Wait · Diagram

1. แต่ละ Frame ที่ส่ง กำหนด Timer
2. Frame Sequence ใช้ 1 Bit สำหรับตรวจ Frame ซ้ำ

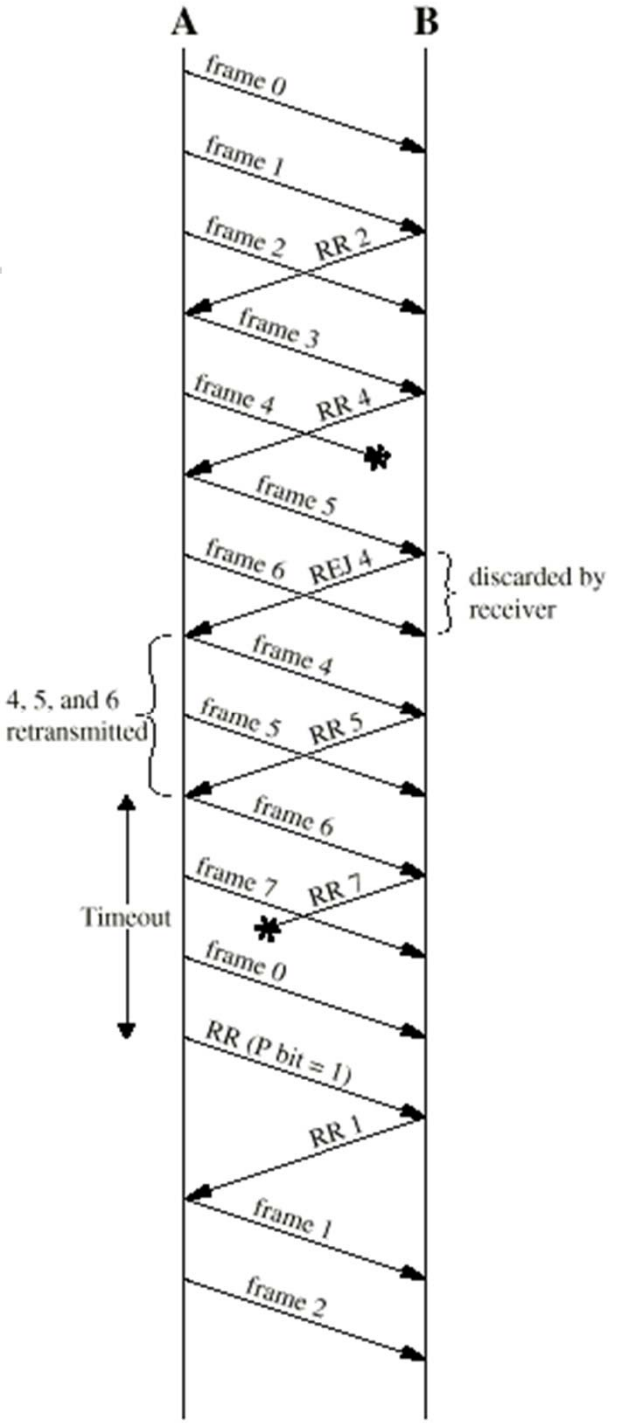
ในทางปฏิบัติ การ Acknowledge จะใช้ Piggyback
สำหรับใน Microprocessor อาจจะใช้สาย Ack แยกต่างหาก



Go Back N - Diagram

1. แต่ละ Frame ที่ส่ง กำหนด Timer
2. ขนาด Window สูงสุดไม่เกิน 2^n-1
3. เมื่อมี Error ให้เริ่มส่งใหม่ตั้งแต่ Frame นั้น

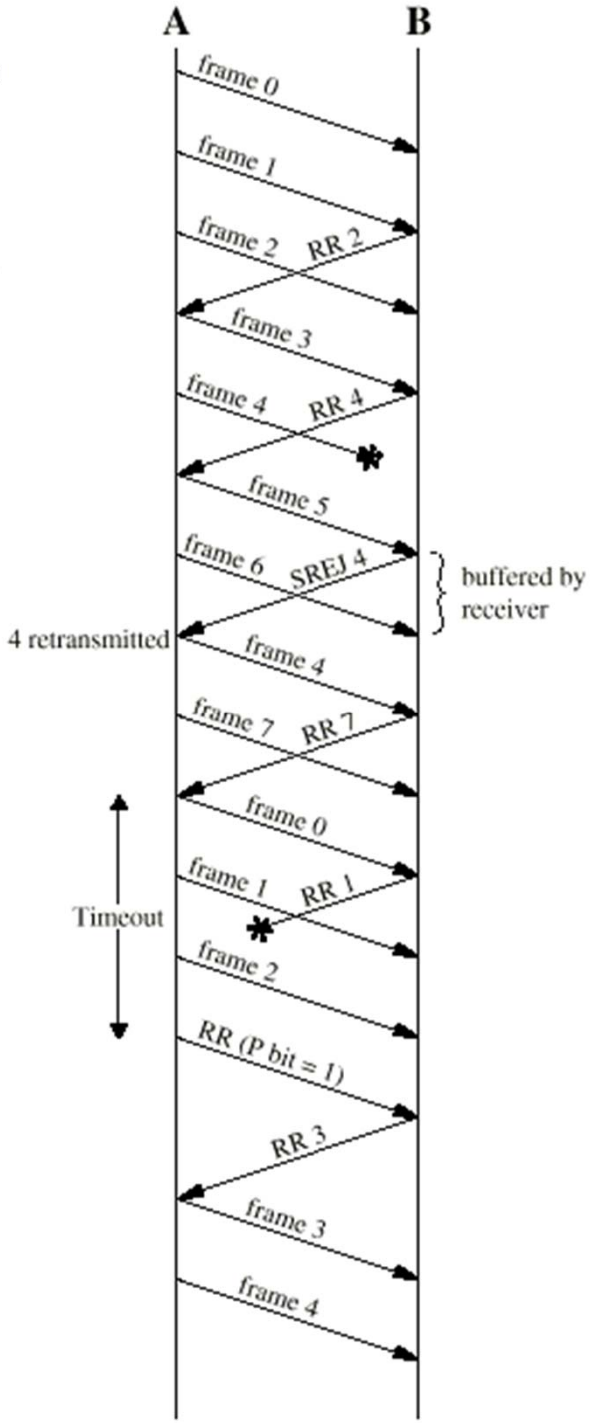
ในทางปฏิบัติ การ Acknowledge จะใช้ Piggyback



Selective Reject Diagram

1. แต่ละ Frame ที่ส่ง กำหนด Timer
2. ขนาด Window สูงสุดไม่เกิน 2^{n-1}
3. เมื่อมี Error ส่งใหม่เฉพาะ Error Frame

ในทางปฏิบัติ การ Acknowledge จะใช้ Piggyback



WAN

- **WAN**
  - Public Network
    - การเชื่อมต่อปกติจะผ่าน Network ของผู้ให้บริการ หรือ Service Provider เราไม่ได้เป็นเจ้าของ
    - เป็นลักษณะการเช่า จ่ายตามจำนวนที่ใช้ เวลา/จำนวนข้อมูล
    - ระยะทางไกลกว่า
    - Technologies ที่ใช้ในการเชื่อมต่อ แตกต่างกัน
  - Network
    - Circuit Switching Network
    - Packet Switching Network

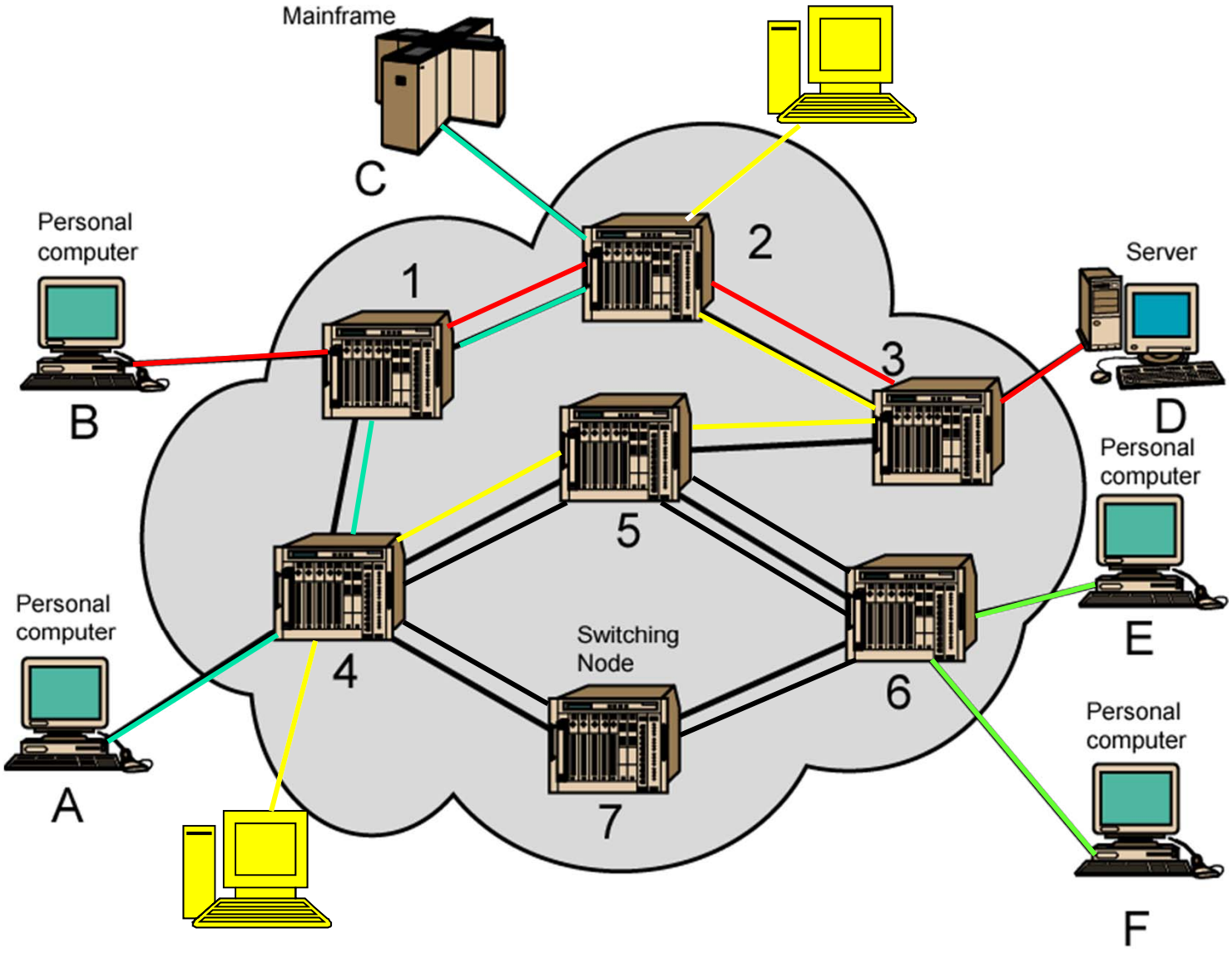
Circuit Switching Network

- สำหรับเครือข่ายโทรศัพท์
- ลักษณะข้อมูลและสัญญาณเป็น Real-Time
  - ยอมให้มี Error ได้บ้าง
  - ค่า Delay และ Delay Variation จะถูกจำกัดไม่ให้เกินค่าที่กำหนด
- ดังนั้นเพื่อให้ NW สามารถรองรับความต้องการได้ ทั้งผู้ส่งและผู้รับจะต้องมีวงจรเชื่อมต่อ (Circuit) ที่เป็นส่วนตัว จะใช้ร่วมกันไม่ได้ = Dedicate Circuit
- อย่างไรก็ตาม เพื่อประหยัด Resource ตัววงจรดังกล่าวจะแบ่งกันใช้ ถ้าผู้ใดไม่ใช้สามารถให้คนอื่นใช้ได้ และการใช้ต้องมีการจอง

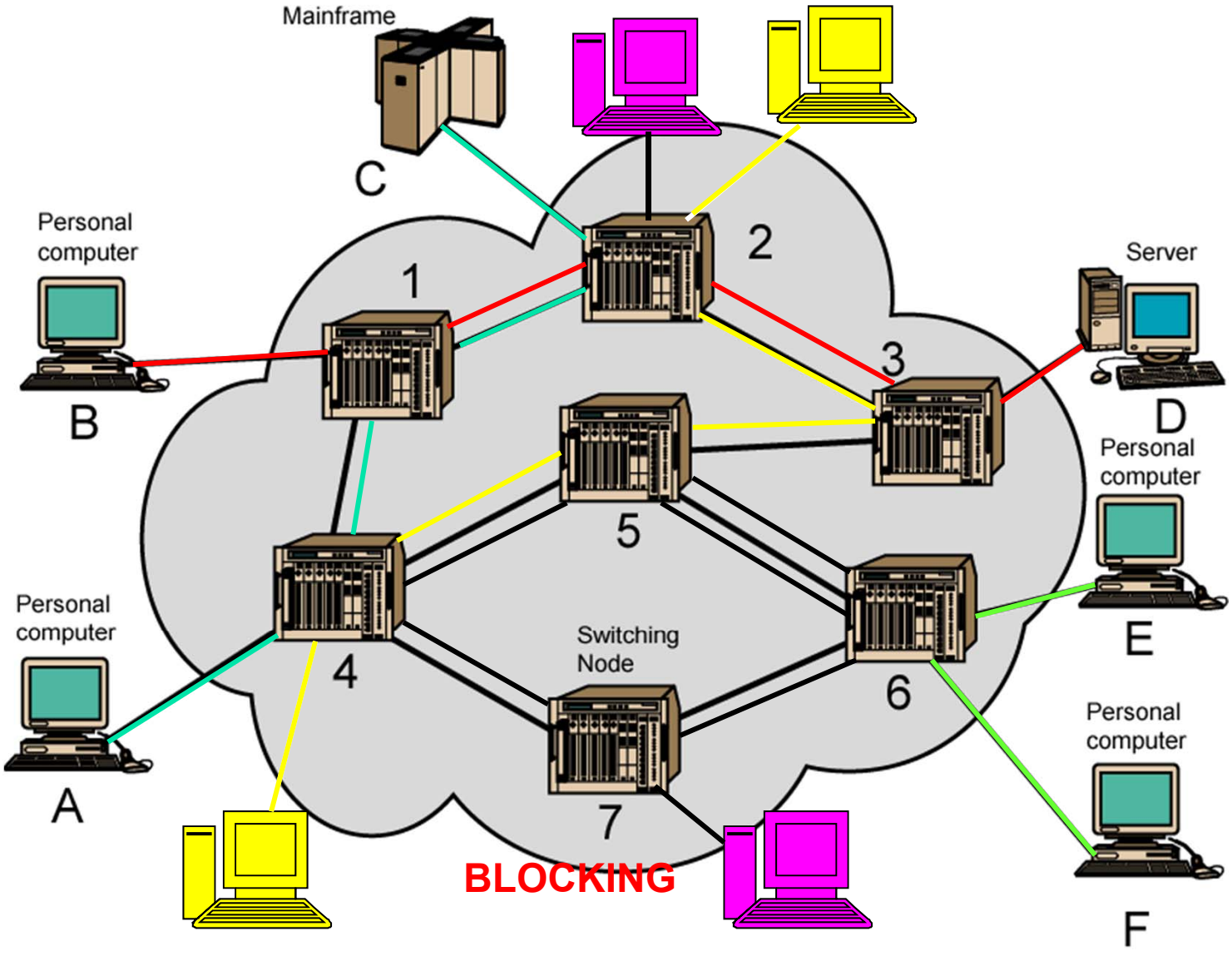
Circuit Switching Network

- **อย่างไรก็ตาม เพื่อประหยัด Resource ตัววงจรดังกล่าวจะแบ่งกันใช้ ถ้าผู้ใดไม่ใช้สามารถให้คนอื่นใช้ได้ และการใช้ต้องมีการจอง**
- **ดังนั้นการใช้งานจะแบ่งเป็น 3 Phase**
  - 1. Connection เพื่อขอ Circuit โดยการหมุนเลขหมายไปยังปลายทาง ตัว Network จะหาทิศทาง กำหนดว่าใช้ Link ไหนและผ่าน Node = Switch อะไรบ้าง ถ้าทิศทางว่าง และผู้รับทำการรับสาย วงจรนั้นจะถูกจองไว้
  - 2. Data Transfer ในกรณีโทรศัพท์คือเสียง
  - 3. Disconnection เมื่อไม่ใช้แล้ว ส่วน Resource ต่างๆที่ถูกจองไว้จะถูกส่งคืน และ Network สามารถนำไปให้ผู้อื่นใช้ได้

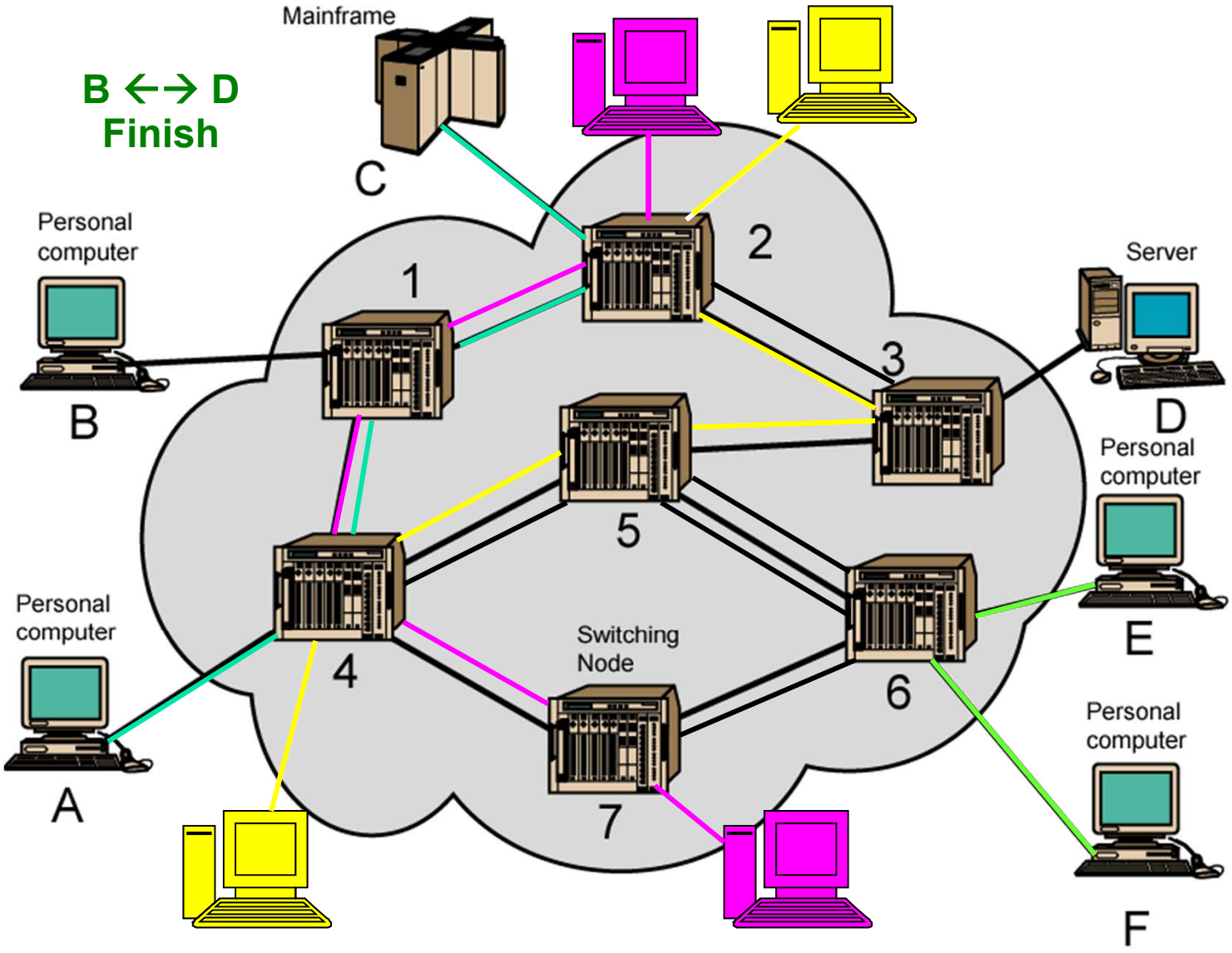
Simple Switched Network



Simple Switched Network



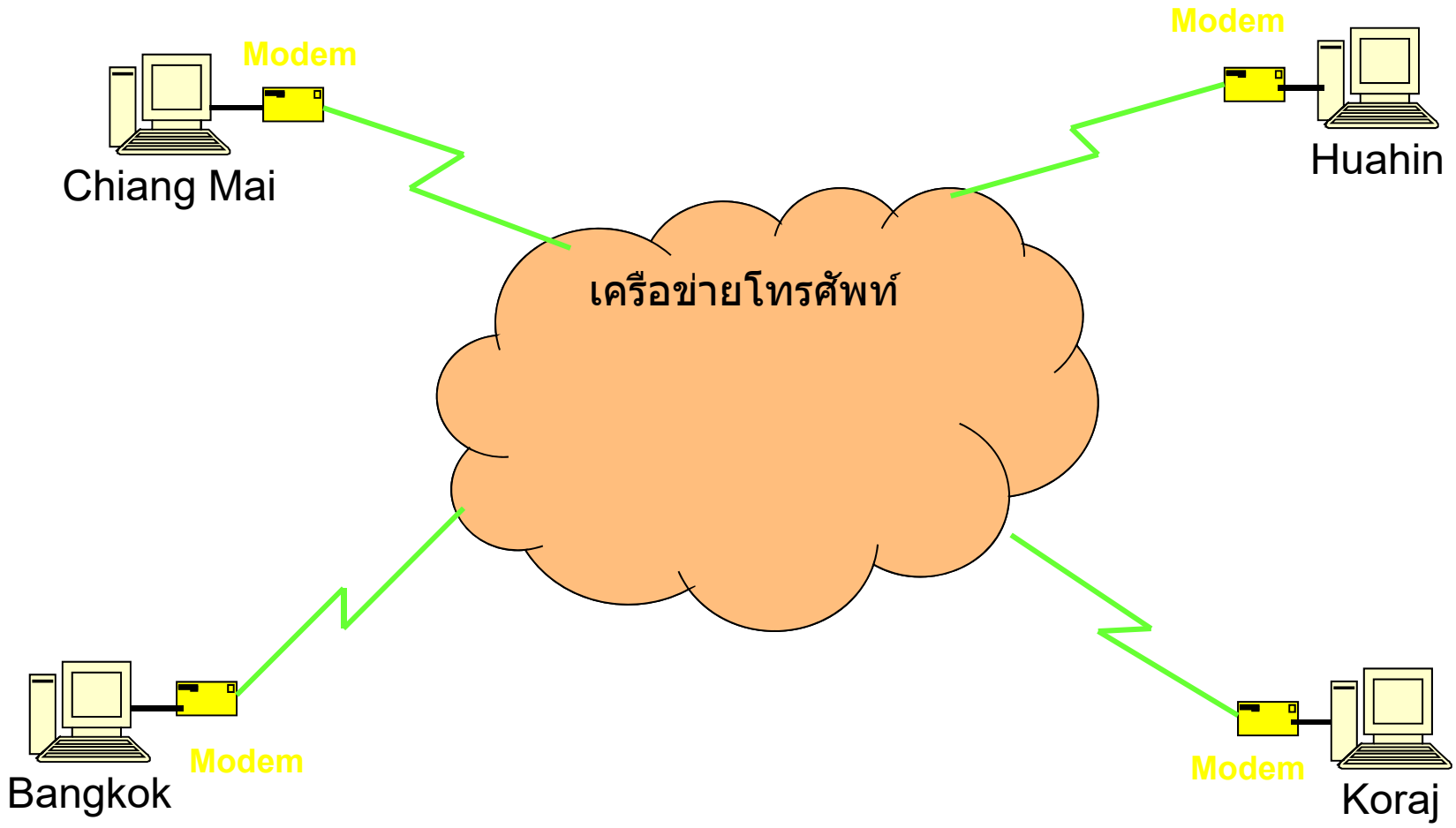
Simple Switched Network



Circuit Switch

- **เมื่อนำมาส่ง Data**
  - ผ่านอุปกรณ์ MODEM = Modulator/Demodulator
- **พฤติกรรมการส่งข้อมูลปกติจะเป็น Burst คือ นานๆจะส่ง แต่เมื่อส่งจะส่งข้อมูลทีละมากๆ**
  - การดู WEB Page เรา Load Web ในช่วงเวลาสั้นๆ ถ้า Page มีขนาดใหญ่ ข้อมูลจำนวนมากจะถูกส่งในเวลาสั้นๆนั้น
  - เราอ่าน Web Page เราไม่ได้ใช้ Network Bandwidth ปกติเราจะใช้เวลาอ่านนานกว่าการ Load
  - Circuit ที่จองไว้ เวลาส่วนใหญ่จะเสียไป ไม่ได้ใช้งาน แต่คนอื่นใช้ไม่ได้
  - ประสิทธิภาพจะต่ำ

Example


Modem
Chiang Mai
Modem
Huahin
เครือข่ายโทรศัพท์
Bangkok
Modem
Modem
Koraj

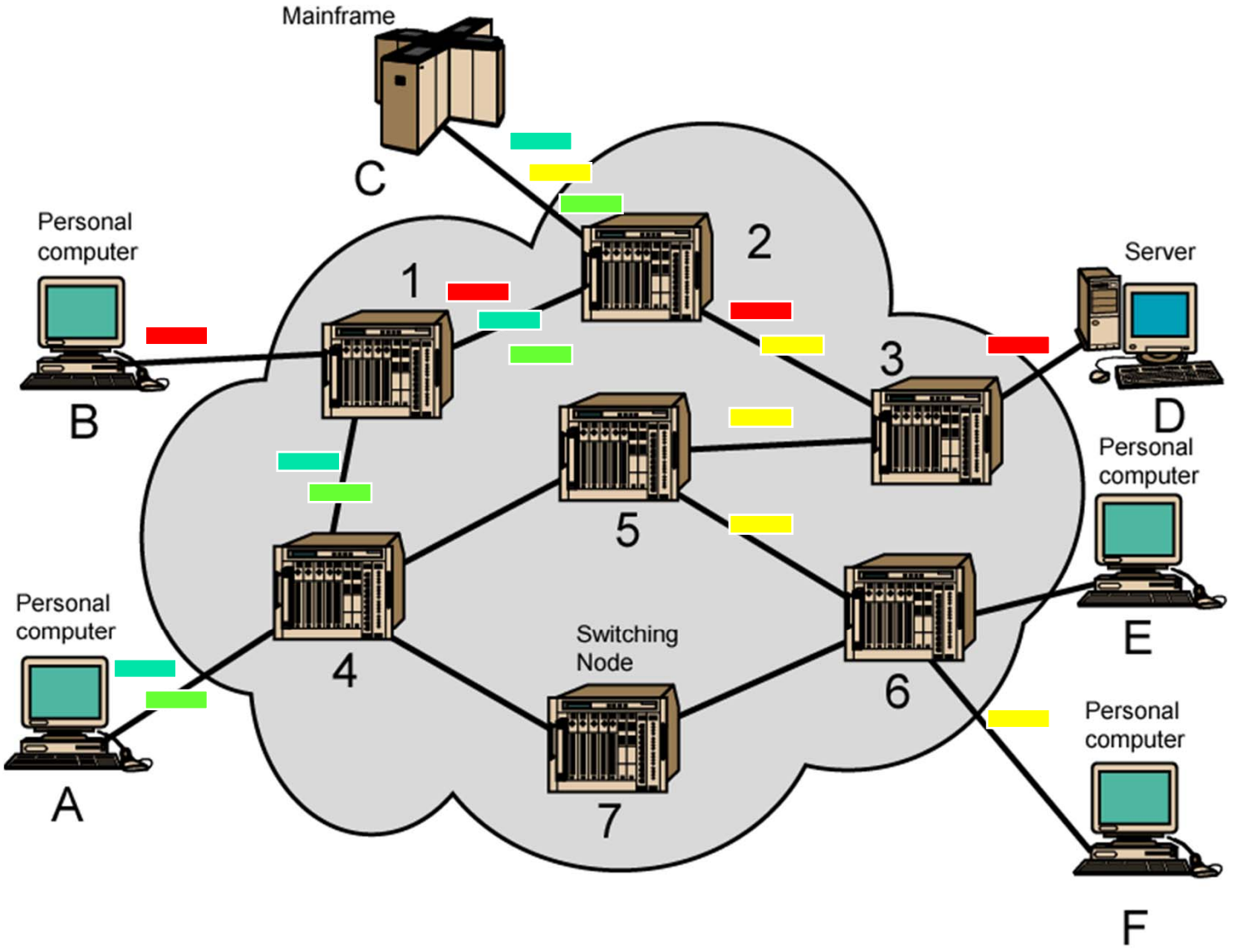
Packet Switching Network

- **Circuit Switching ไม่เหมาะสมสำหรับการส่งข้อมูล**
- **เราใช้ Packet Switching**
  - ข้อมูลจะถูกตัดเป็น Packet ส่งออกไป
  - ในหนึ่ง Circuit สามารถแชร์กันได้หลายคน ทำให้ประสิทธิภาพสูงกว่า
    - Online จะใช้สามารถใช้ได้ทันที ไม่ถูก Block
  - ประสิทธิภาพสูงกว่า ถ้าเรา Share กันเพียง Circuit เดียว เมื่อผู้ใดไม่ส่ง คนอื่นส่งได้
    - ถ้าส่งพร้อมกันหลายคนก็ทำได้ แต่ละคนจะใช้เวลาในการส่งมากขึ้น เรียกว่าเกิด Delay
    - หลายข้อมูล ของหลายคนใช้ Circuit เดียว
  - แต่ละคนคิดว่าตัวเองเป็นเจ้าของ Circuit คนเดียว = Transparency
  - อย่างไรก็ตามข้อมูลจริงๆ วิ่งอยู่บน Circuit เดียวกัน ต้องมีวิธีบ่งบอก
    - Address ผู้ส่งและผู้รับ แปะที่ส่วน Header ของข้อมูล
    - หรือ ใช้ Virtual Circuit Number สำหรับแต่ละคน และแปะที่ส่วนหัวข้อมูล

Packet Switching Network

- เนื่องจากต้อง Share วงจรกัน เพื่อป้องกันไม่ให้ผู้ใดผู้หนึ่งผูกขาดการใช้งาน ต้องกำหนดขนาดข้อมูลสูงสุดที่ผู้ส่ง ส่งได้ในแต่ละครั้ง = MTU, Maximum Transfer Unit
- ถ้าข้อมูลใหญ่กว่านั้น ต้องแบ่ง หรือตัดข้อมูลเป็น Packet ย่อยๆ แต่ละ Packet มีส่วนหัว นอกเหนือจาก Address/VC แล้ว จะต้องมี Sequence Number บ่งบอกลำดับของข้อมูล
- กระทำโดย Protocol ผู้ใช้ (Application) ไม่ต้องทำ
- นี่คือ Packet Switching Network

Packet Switched Network



Advantages

- **Line efficiency**
  - แต่ละ Link สามารถจะ Share กันได้
  - Packets ที่เข้ามาแต่ละ Node จะถูกเข้า Queue เพื่อส่งออกไป
- **Data rate conversion**
  - Each station เชื่อมต่อกับ Local Node ด้วยความเร็วที่ตัวเองกำหนด
- **Packets are accepted even when network is busy โดยเก็บไว้ใน Queue**
  - Delivery may slow down = Delay
- **Priorities can be used**

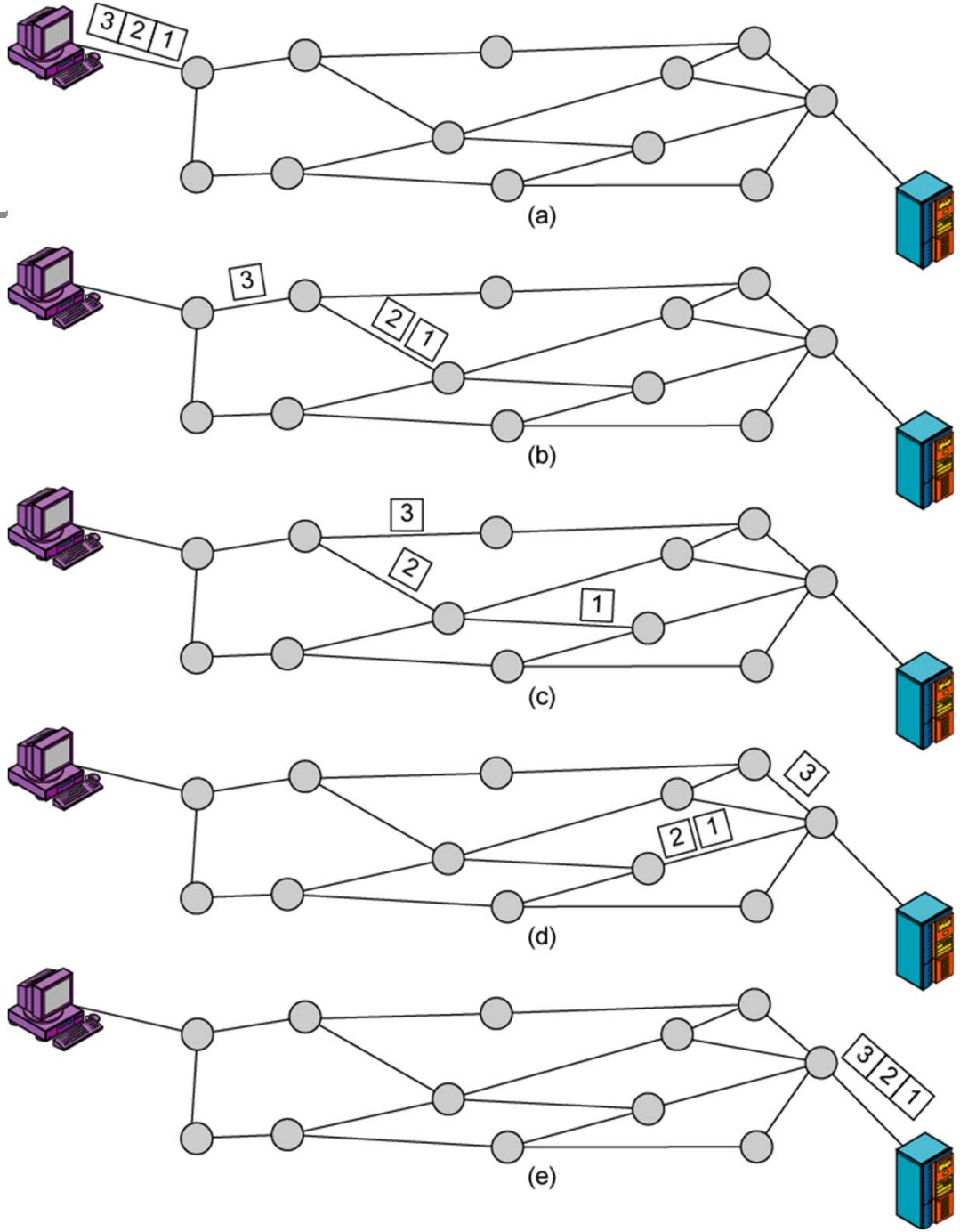
สรุป Packet Switching Network

- **2 Concepts กำหนดการทำงานใน Network(L3)**
  - Datagram
  - Virtual Circuit
- **2 Concepts กำหนดการเชื่อมต่อกับผู้ใช้ภายนอก (ปกติจะอยู่ใน L4)**
  - Connection Oriented
  - Connectionless

การทำงานของ Datagram

- Each packet treated independently
- Packets can take any practical route
- Packets may arrive out of order
- Packets may go missing
- Up to receiver(ปลายทาง) to re-order packets and recover from missing packets
- สรุปแล้ว การทำงานของ Network ประเภทนี้จะไม่ Guarantee การส่งข้อมูล

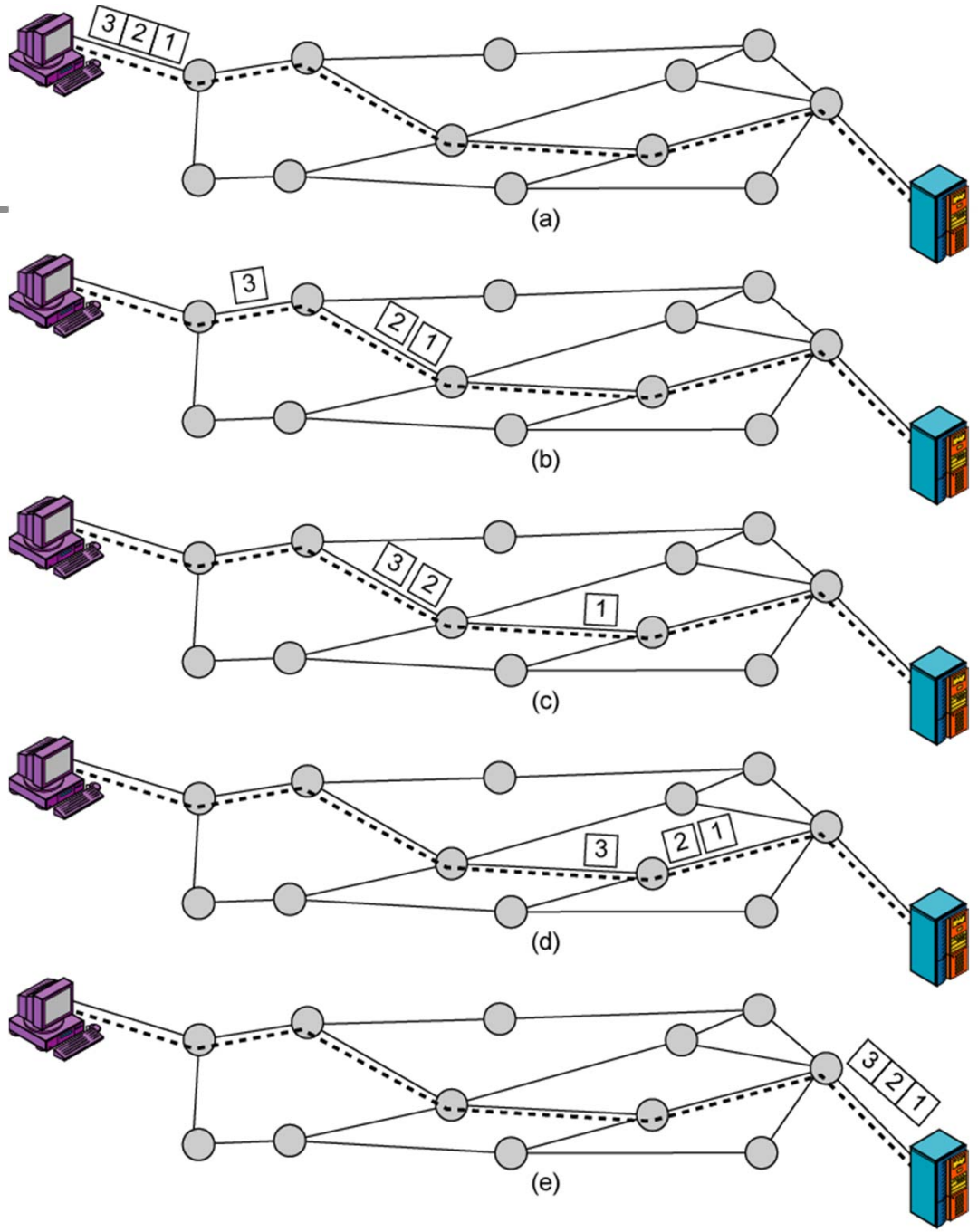
Datagram Diagram



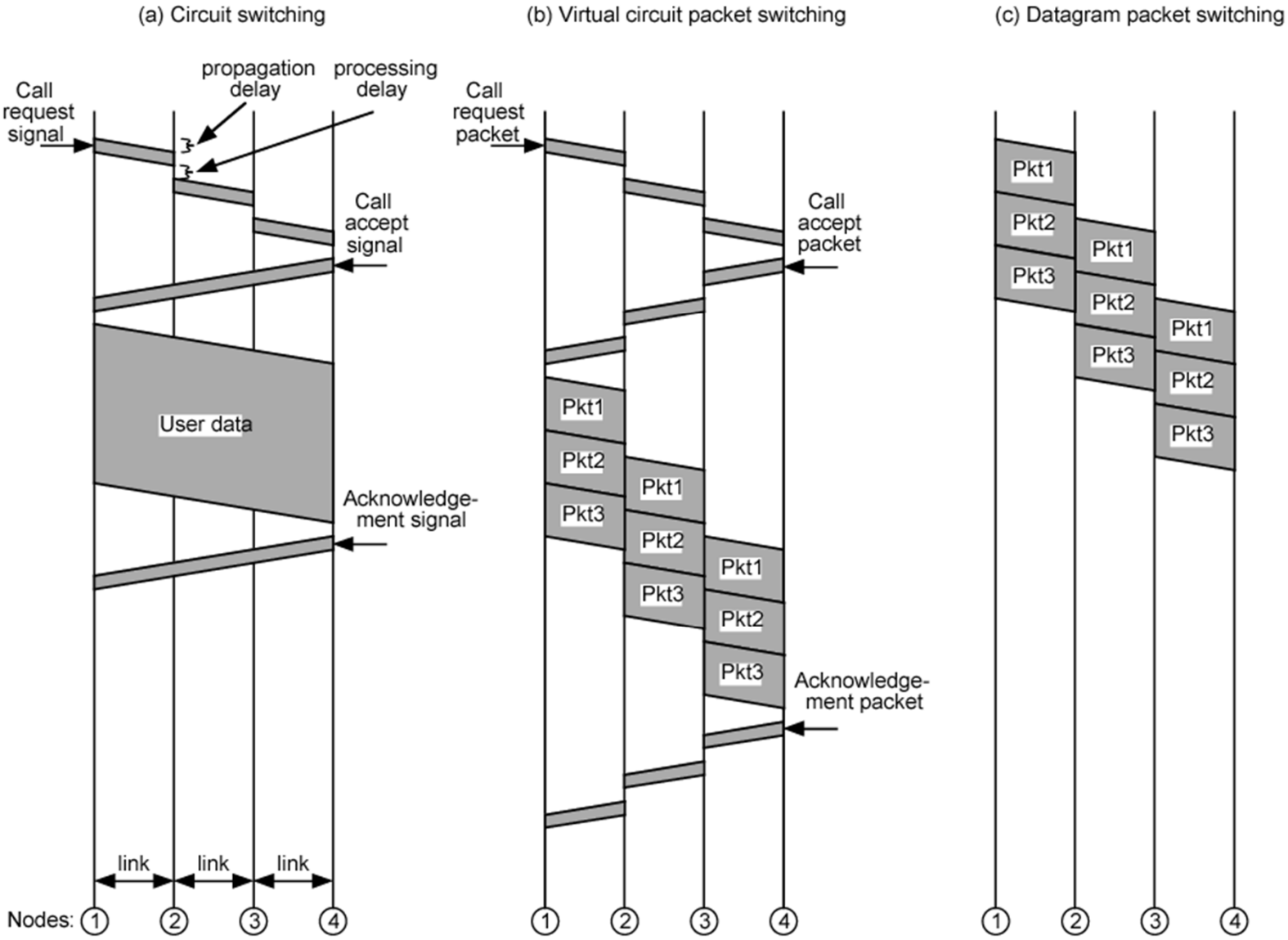
Virtual Circuit

- Preplanned route established before any packets sent เส้นทางจะถูกกำหนดในช่วงการ Connection
- Call request and call accept packets establish connection (handshake) กำหนด Connection ด้วยตัวเลข คือ VC Number
- Each packet contains a virtual circuit identifier instead of destination address
- No routing decisions required for each packet ดูจาก VC # ก็เพียงพอ
- Clear request to drop circuit เมื่อจบ
- Not a dedicated path แต่มองจากผู้ใช้เหมือน Circuit Switching

Virtual Circuit Diagram



Event Timing เปรียบเทียบ 3 NW



End of Review Part I

- **End of Review Part I**
- **Next Week**
  - LAN and LAN Technologies
- **Internet Concept**
- **ยังไม่มีการบ้าน**

CPE 426 Computer Networks

Chapter 2: Review Part II

LAN Technologies

Course Outlines

- ดูใน Sheet
- สามารถ Download ได้
  - http://cpe.rsu.ac.th/ut

Review Part II

- LAN and LAN Technologies
- เน้นที่ Ethernet Technologies

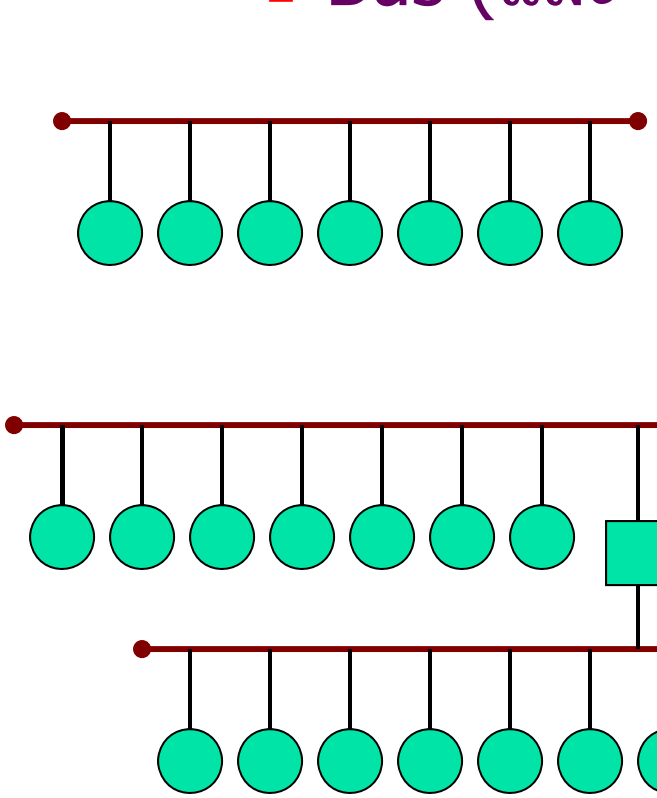
Network types

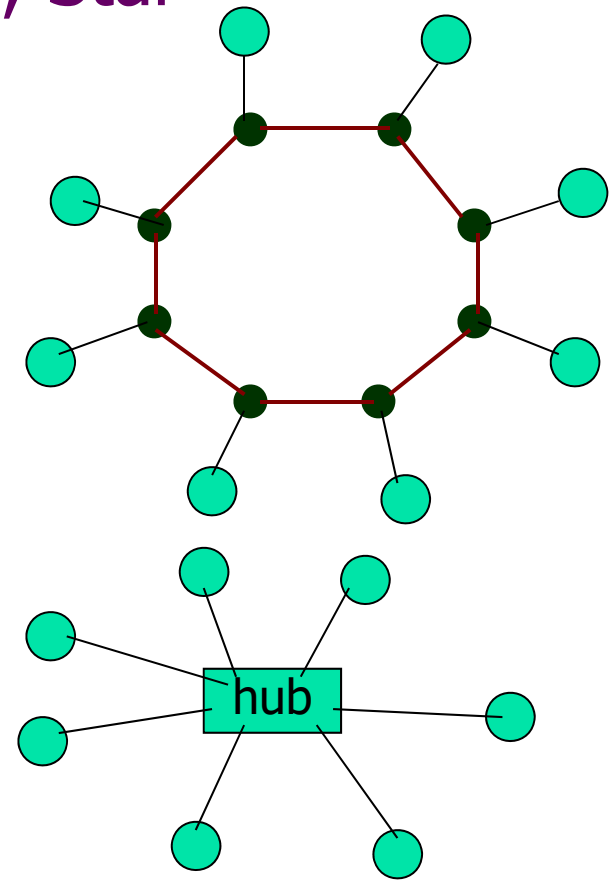
ปัจจุบัน Technology ของ LAN ผ่าน Fiber Optic สามารถส่งได้ไกลในระดับ MAN

	Range	*Bandwidth (Mbps)*	*Latency (ms)*
LAN	1-2 kms	10-1000	1-10
WAN	worldwide	0.010-600	100-500
MAN	2-50 kms	1-150	10
Wireless LAN	0.15-1.5 km	2-11	5-20
Wireless WAN	worldwide	0.010-2	100-500
Internet	worldwide	0.010-2	100-500

วิธีแก้คือ Share Medium และทำ Multiple Access Control

- ใน LAN จะใช้ Topology 3 แบบที่สำคัญ
  - Bus (และ Tree), Ring, Star





การ Share Medium

- **ต้องมีการควบคุม = Medium Access Control**
- **End Node จะต้องมีการกำหนดชื่อหรือ Address สำหรับอ้างอิง หรือกำหนด Circuit Number**
- **Intermediate Node จะใช้หมายเลขอ้างอิงดังกล่าวในการตัดสินใจส่งข้อมูลต่อออกไป(Forwarding)**
- **ดังนั้น**
  - 1. Data ที่ส่งจะต้องแปะส่วนหัว (Header) ด้วยข้อมูลต่างๆของ Address และการ Control เราเรียกว่าเป็นการทำ Encapsulation ผลลัพธ์ที่ได้เรียกว่า Frame
  - 2. ที่ส่วนท้ายของ Frame จะมีการต่อด้วยข้อมูลช่วยตรวจจับความผิดพลาด (Error Detection) มักจะเป็น CRC Code เรียก Frame Check Sequence(FCS)
  - 3. ก่อนหน้าส่วน Header และหลัง FCS อาจจะมีการเติมบิตสำหรับช่วยตรวจจับหัวและท้ายของ Frame (Frame Delimiter: Pre-amble/Post-amble)
  - 4. สำคัญที่สุดต้องมีการกำหนดกฎเกณฑ์ต่างๆเหล่านี้ให้เป็นมาตรฐาน คือกำหนดเป็น Protocol ของการสื่อสาร

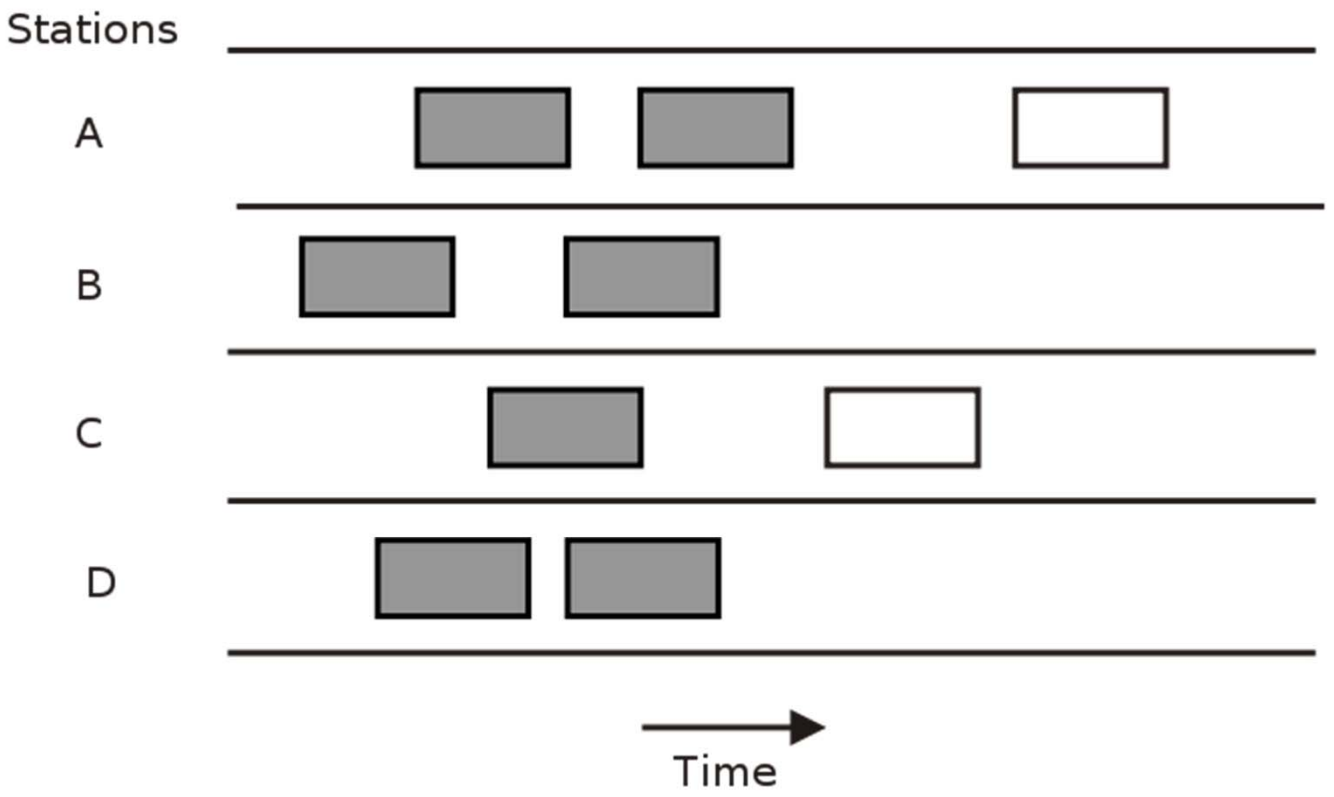
LAN vs WAN Technologies

- **LAN มักจะใช้การ Share Medium แบบ Contention ดังนั้นจะต้องมีขบวนการควบคุมการทำ Multiple Access**
  - Topology ที่เหมาะสมคือ Bus, Ring, Star
- **WAN จะ Share Medium เช่นกัน แต่มักจะใช้วิธีของ Synchronous Multiplexing (TDM) ใน Circuit Switching Networkหรือ Statistical Multiplexing (ใช้ใน Packet Switching Network)**
  - Topology ที่เหมาะสมคือ Mesh Network และมักจะเป็น Partial Mesh
- **Internetworking Technologies มักจะถูกใช้ในการเชื่อมต่อระหว่าง LAN ผ่าน WAN Network**
  - ที่นิยมคือ Internet (IP Network)

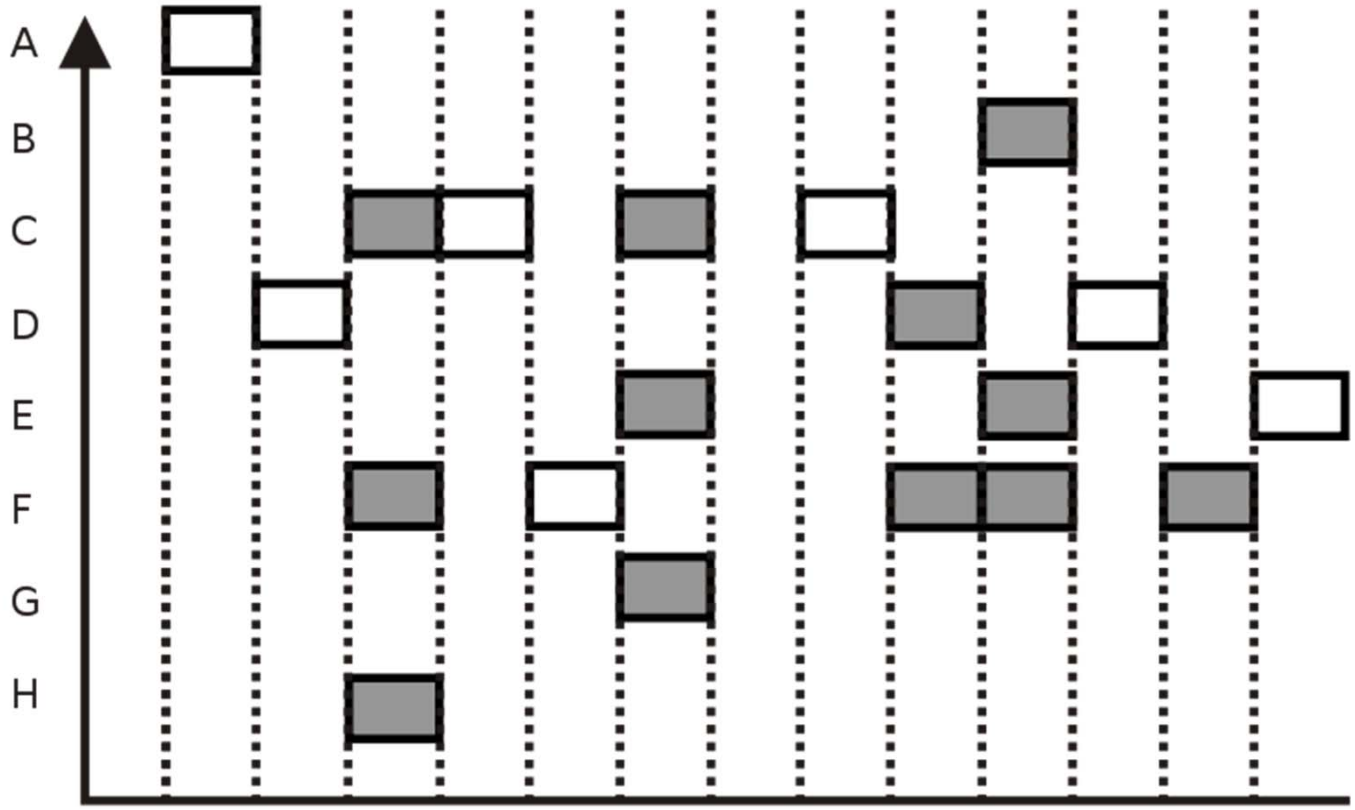
ALOHA System

- เป็นระบบที่ใช้ทดลองการทำ Multiple Access(Random Access) ของ Packet Radio System
  - 1970 University of Hawaii
- จากการวิจัยพบว่า Efficiency ของระบบ มีได้สูงสุด 18%
  - ถ้าใช้ Slotted ALOHA จะได้ถึง 36%
- ค่านี้เป็นค่าสูงสุดในทางทฤษฎี
- การศึกษาวิจัย ALOHA นำไปสู่การพัฒนา CSMA ซึ่งถูกนำไปใช้เป็นครั้งแรกใน Ethernet
  - ให้ฟังก่อนที่จะส่ง

Pure ALOHA



Slotted ALOHA



Slotted ALOHA protocol (shaded slots indicate collision)

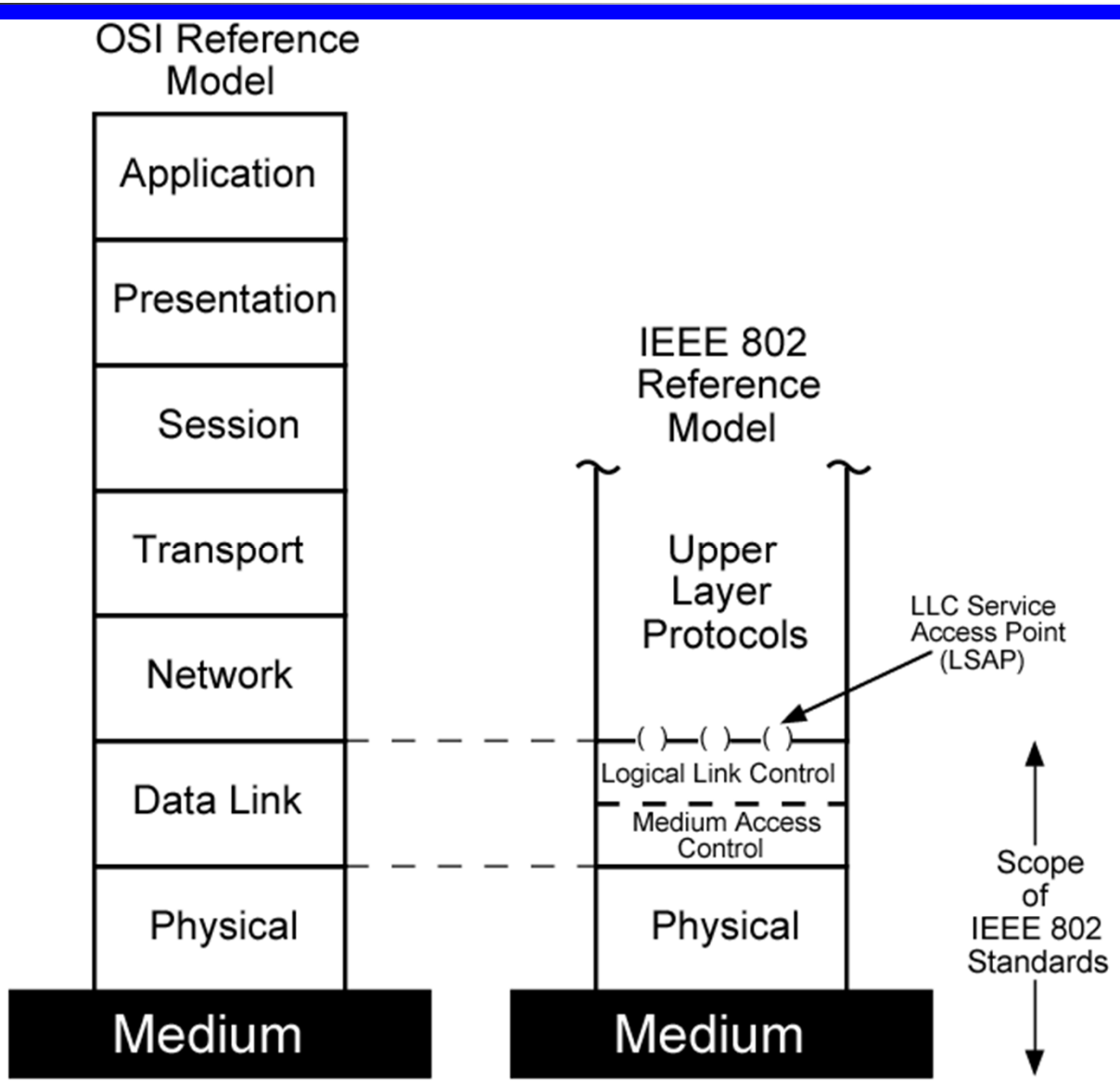
Local Area Networks

- **Smaller scope**
  - Building or small campus
- **Usually owned by same organization as attached devices**
- **Data rates much higher**
- **Usually broadcast systems**
- **Now some switched systems and ATM are being introduced**

LAN Configurations

- **Switched**
  - Switched Ethernet
    - May be single or multiple switches
  - ATM LAN
  - Fibre Channel
- **Wireless**
  - Mobility
  - Ease of installation

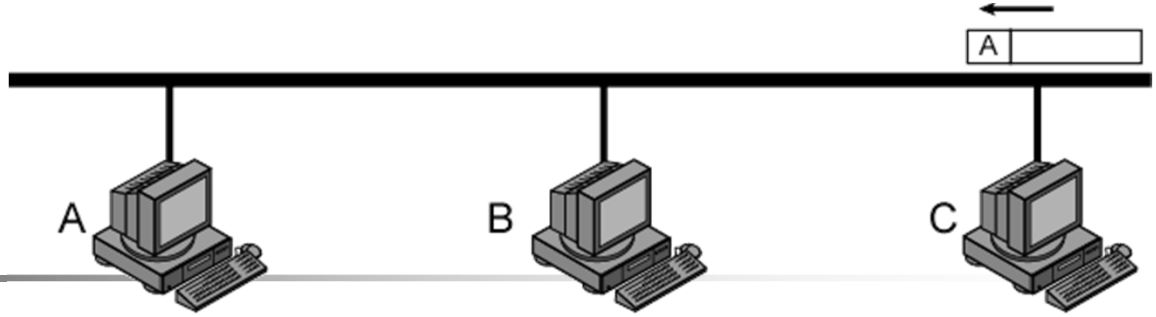
IEEE 802 v OSI



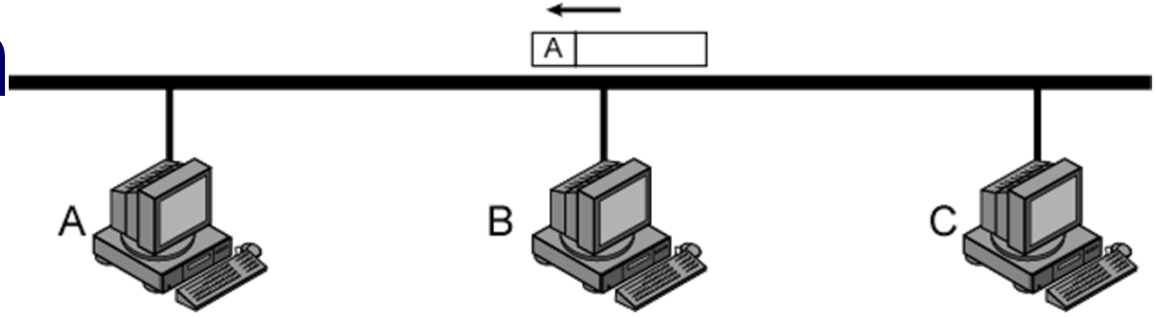
Protocol Stack

- **Application**
- **Data link**
  - Logical Link Control(LLC)
  - Medium Access Control(MAC)
- **Physical**

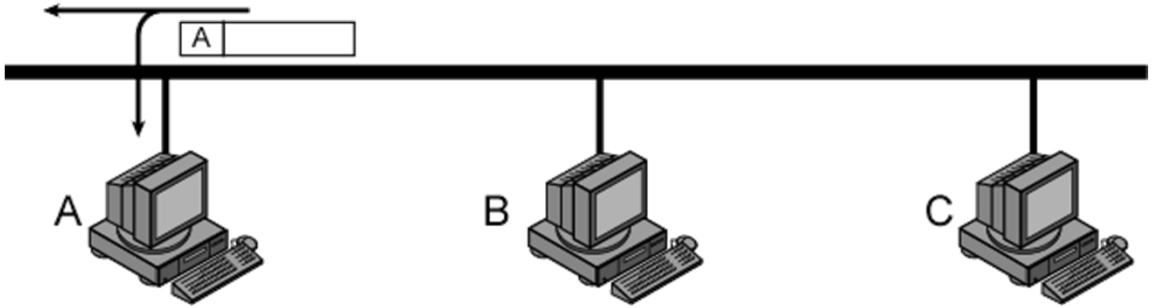
Frame Transmission on Bus LAN



C transmits frame addressed to A



Frame is not addressed to B; B ignores it



A copies frame as it goes by

CSMA ใน BUS Ethernet

- **Carrier Sense Multiple Access**
- **1. ก่อนส่งข้อมูล ให้ฟังก่อนว่ามีใครกำลังใช้ Channel หรือไม่ ถ้าไม่มีให้ส่งได้ มิฉะนั้นให้รอจนกว่า Channel จะว่าง ถึงส่งได้**
- **ปัญหา**
  - ถ้ามีผู้รอส่งมากกว่าหนึ่งคน เมื่อสายว่าง คนเหล่านั้นจะส่งข้อมูลออกมา และชนกัน
  - เราต้องการกลไกเพิ่มเติม ในการตรวจจับการชนกันและจัดการ เรียก Collision Detection(CD)

CD or Collision Detection

- **2. ในขณะที่กำลังส่งข้อมูลให้ทำการฟังด้วย ถ้าข้อมูลที่ฟังได้ไม่เหมือนกับที่ส่ง แสดงว่าเกิดการชนกัน ให้หยุดส่งทันที พร้อมทั้งส่งสัญญาณบอกสถานีอื่นว่าได้มีการชนกันเกิดขึ้น (Jamming Signal)**
- **3. หยุดรอเป็นระยะเวลา Random และลองใหม่ (กลับไปยังข้อ 1)**
- **4. ถ้ามีการชนกันติดต่อกัน แต่ละครั้งที่หยุดรอ ให้จับเลข Random ที่มีค่า Standard Deviation เป็นสองเท่า**
  - เรียก Binary Exponential Back-Off
- **5. ถ้าจำนวนครั้งที่ชนกัน ติดต่อกันเกินกำหนด ให้เลิกล้มการส่งข้อมูลและ Report ไปยังผู้ส่ง**
- **ขบวนการรวมเรียก CSMA/CD**

Notes on CSMA/CD

- **เพื่อที่จะให้ CD สามารถทำงานได้ ข้อมูลต้องส่งเป็นจำนวนมากพอ**
  - ใน Ethernet กำหนดให้ขนาดของ Frame ที่ส่งอย่างต่ำต้องมีความยาว 64 Octet(512 Bit)
- **เพื่อป้องกันไม่ให้ผู้ใดผู้หนึ่งใช้ Channel นานเกินไป จะต้องมีการกำหนดค่า MTU (Maximum Transfer Unit)**
  - Ethernet กำหนดขนาด Frame สูงสุดคือ 1518 Octet โดยส่วน Payload จะมีขนาดสูงสุดคือ 1500 Octet
- **เมื่อคนหนึ่งส่งไปหนึ่ง Frame แล้ว จะส่งอีก Frame ต่อเลยไม่ได้ ต้องรอว่ามีใครต้องการส่งหรือเปล่า (Inter-Frame Gap) ถ้าไม่มีจึงส่ง Frame ต่อไปได้**

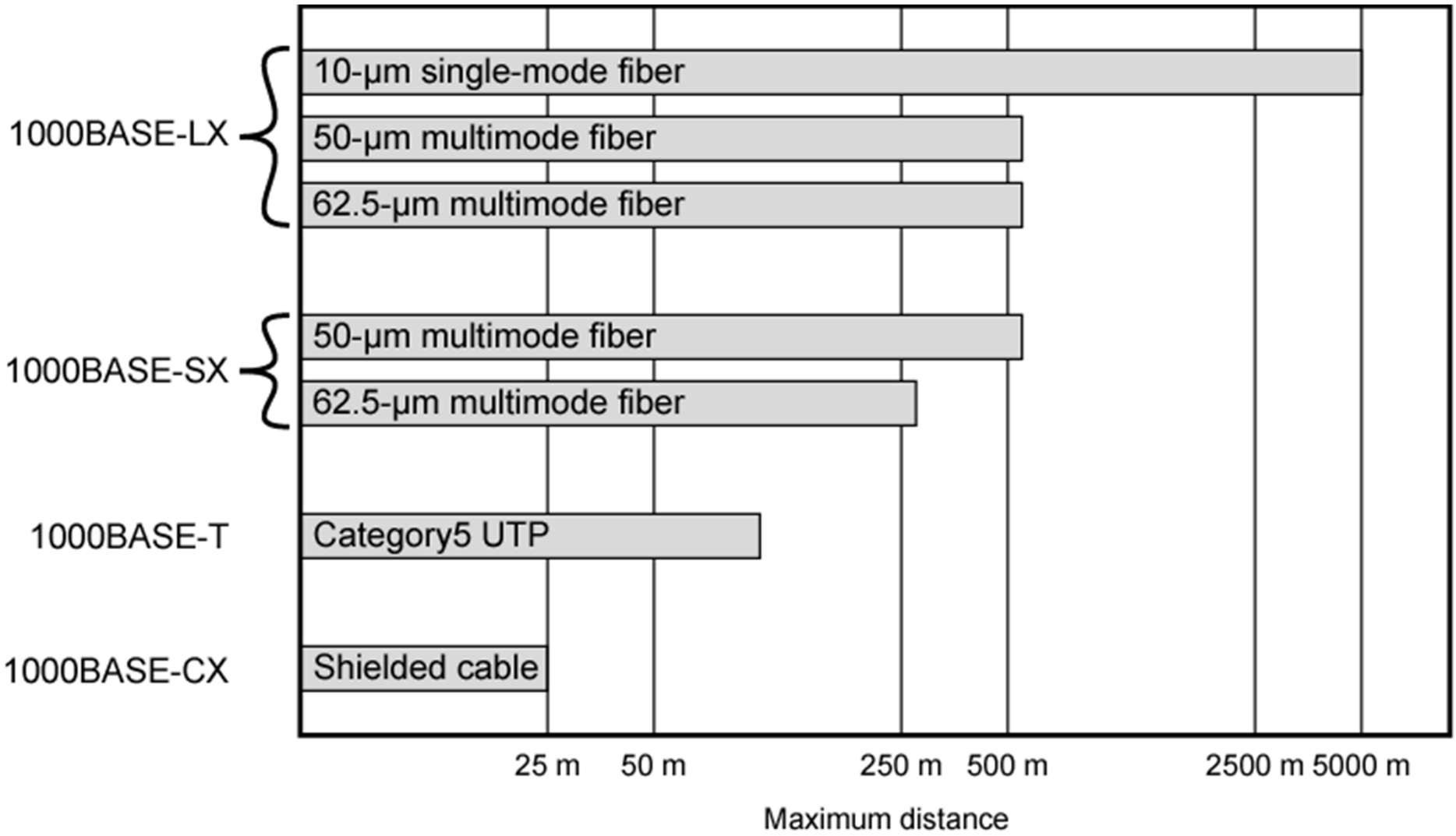
Review Ethernet Technologies

- **IEEE 802**
  - IEEE 802.1 Management
    - IEEE 802.1D Spanning Tree
    - IEEE 802.1Q VLAN Tag
    - IEEE 802.1X อื่นๆ
  - IEEE 802.2 LLC
  - IEEE 802.3 Ethernet
  - IEEE 802.11 WLAN

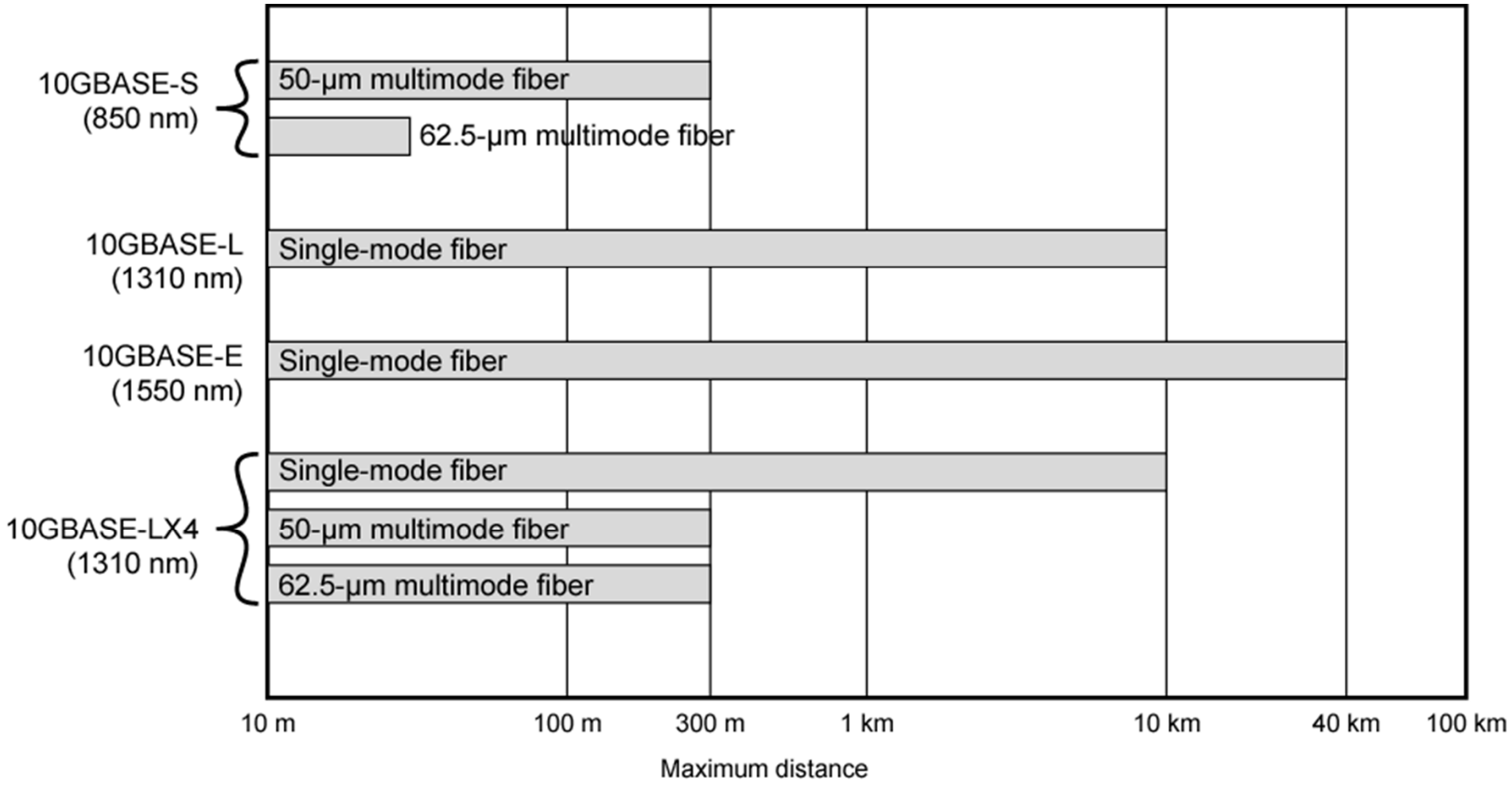
LAN Technologies(Ethernet)

- **Wired**
  - 10M(Ethernet), 100M(Fast Ethernet), 1000M(Gigabit Ethernet), 10G(10 Gigabit), 40G
  - Coaxial, UTP, STP, Fiber Optics(62.5/125,50/125,8-10/125 with 850 nm, 1300/1310 nm, 1550nm)
- **Wireless**
  - 802.11a
  - 802.11b
  - 802.11g
  - Infrared
  - Laser
  - Radio, Microwave

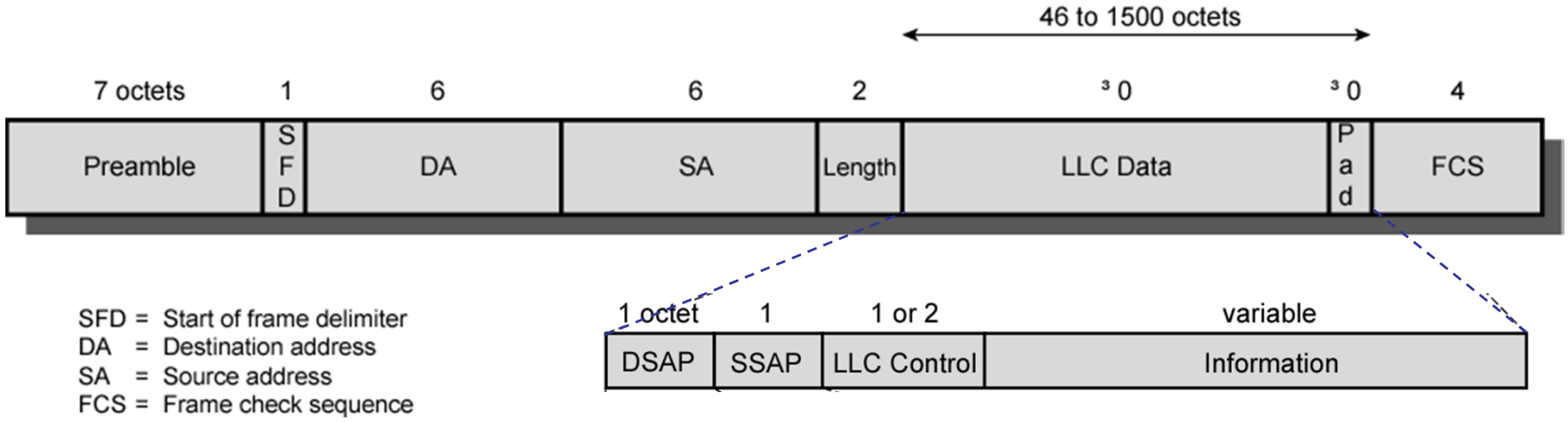
Gbit Ethernet Medium Options (log scale)



10Gbps Ethernet Distance Options (log scale)



IEEE 802.3 Frame Format



General Ethernet Frame Format

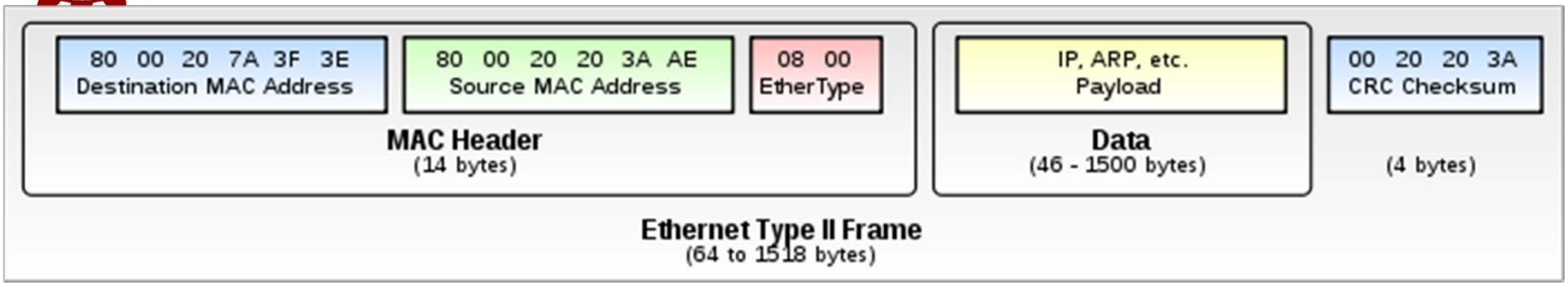
802.3 Ethernet frame structure

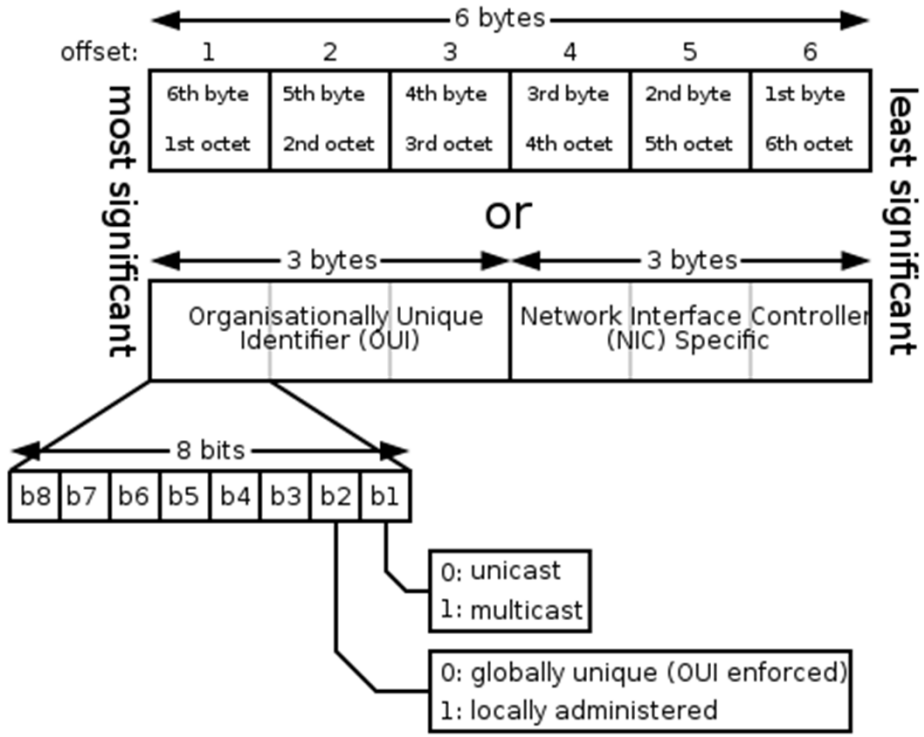
Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype or length	Payload	Cyclic redundancy check	Interframe gap
7 octets of 10101010	1 octet of 10101011	6 octets	6 octets	(4 octets)	2 octets	46–1500 octets	4 octets	12 octets
		64–1522 octets						
72–1530 octets								
84–1542 octets								

IP over Ethernet

- **ปัญหาในการบรรจุ IP Packet ลงใน Ethernet IEEE 802.3**
  - เมื่อ IP ถูกประกอบส่วนหัวผ่าน LLC Layer โดยส่วนหัวนี้จะประกอบด้วย DSAP และ SSAP และส่วน Control ที่บ่งบอกว่า LLC Frame มี Payload ของ Protocol อะไรอยู่ จากนั้นจึงส่งให้ MAC Layer เพื่อประกอบ MAC Frame
  - มาตรฐานของ LLC มี Code กำหนด IP Packet แต่ไม่มี Code กำหนด ARP Protocol ทำให้เกิดปัญหาในการประกอบ MAC Frame
    - ARP จะต้องใช้ในการประกอบ MAC Frame เพื่อใช้หา MAC Address ของปลาย Link ที่ Match กับ IP Address (ของเครื่องปลายทางหรือของ Gateway)
    - ดังนั้น เมื่อมีการ Run ARP มันจะประกอบ LLC Frame ไม่ได้
  - วิธีแก้
    - ปรับปรุงมาตรฐาน LLC ใหม่ให้สามารถรองรับ ARP ได้
      - มีวิธีการของ SNAP แต่ไม่เคยมีการใช้งาน
    - ย้อนกลับไปใช้ Ethernet Frame Type II ซึ่งเป็นมาตรฐานเก่า ไม่มี LLC
  - ปกติ เรามักจะใช้ TCP/IP วางบน Ethernet สำหรับ Network ในองค์กร (คือ Technology ของ Intranet) ดังนั้นจะพบว่า Ethernet Frame ส่วนใหญ่ที่วิ่งใน LAN จะเป็น Ethernet Type II
    - Switch ปัจจุบัน รับ Ethernet Frame ได้ทั้งสองแบบ เพราะ Switch จะดูแค่ MAC Address ในส่วนหัวของ MAC Frame

Ether Type II (DIX Frame)





MAC-48 Address
In Transmission Order
01-23-45-67-89-ab,
01:23:45:67:89:ab,
0123.4567.89ab

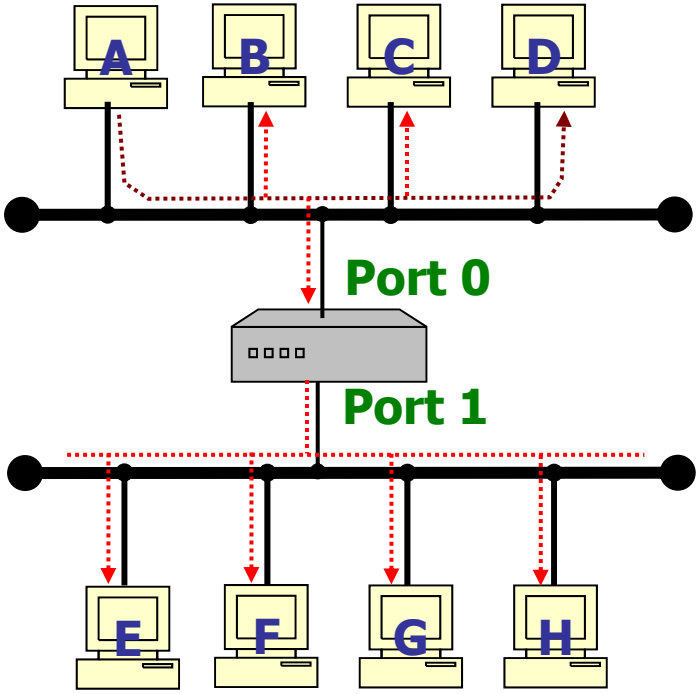
802.3/.4 Send LSBit First
(Canonical Format)
10000000 11000100 10100010 …

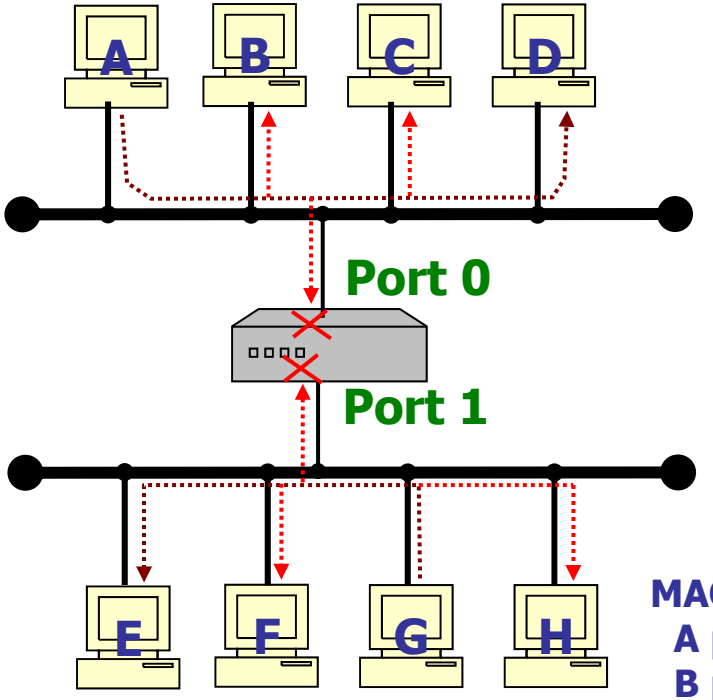
802.5/.6 Send MSBit First
(Bit-Reverse/Non-canonical)
00000001 00100011 01000101 …

Bridge Concept

- **Repeater แค่ทำการสร้างสัญญาณใหม่ ส่งไปยังอีกฝั่งหนึ่ง ไม่สนใจว่าสัญญาณนั้นต้องการจะส่งไปยังที่ใด**
- **Bridge จะสร้างสัญญาณใหม่ และส่งไปยังอีกฝั่งก็ต่อเมื่อ MAC address ปลายทางของ Frame ไม่ได้อยู่ในตาราง MAC address table ของด้านที่ส่งมา**
  - การเรียนรู้ตาราง MAC Address จะกระทำโดยอัตโนมัติ เมื่อมี Frame เข้ามา โดยดูจาก MAC address ต้นทางของ Frame และจะ Update ตลอดเวลา

Bridge vs Repeater

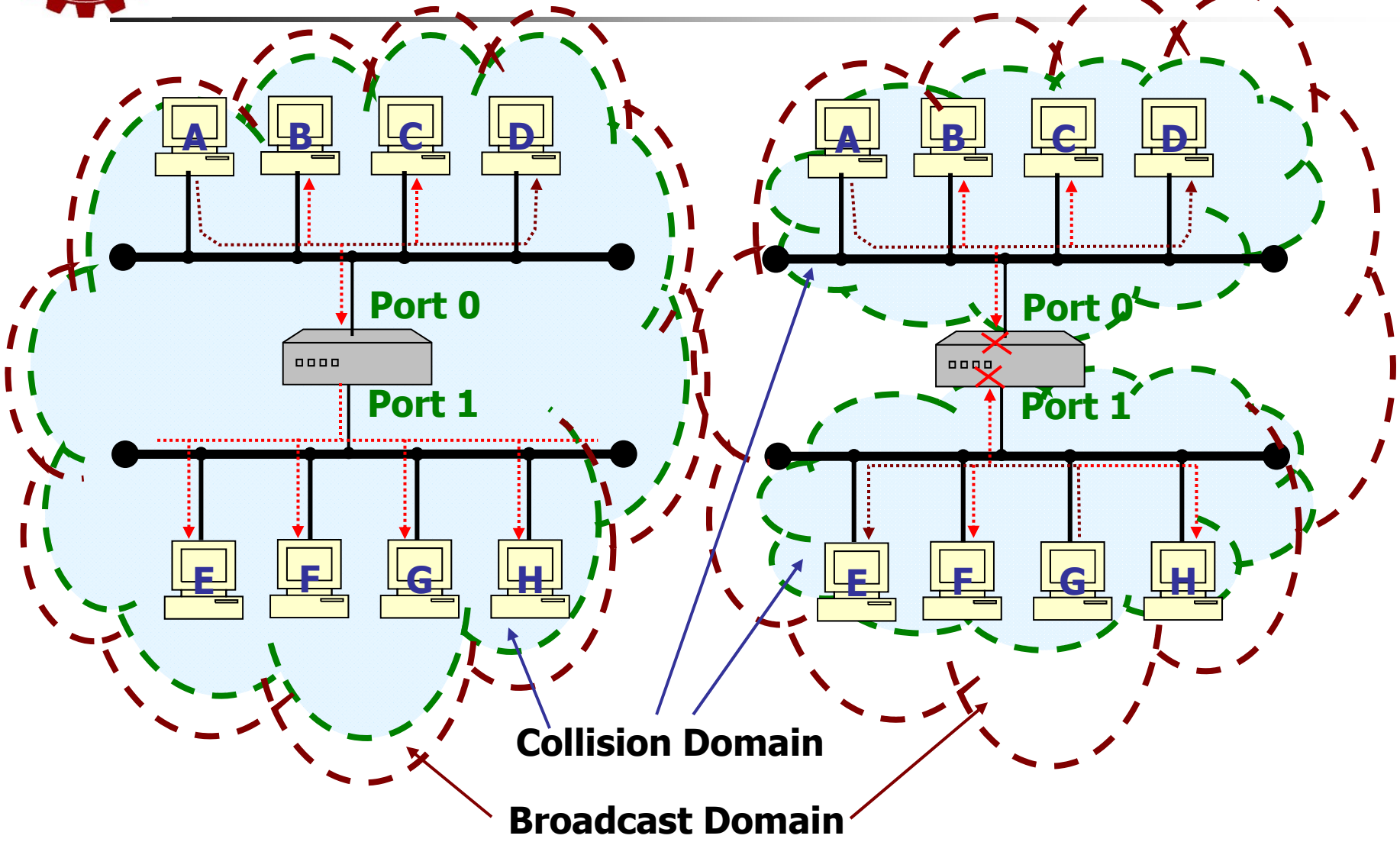




1. ถ้า A ส่งข้อมูลให้ C เกิดอะไรขึ้น ?
2. ถ้า A ส่ง Broadcast Traffic เกิดอะไรขึ้น ?

MAC Table
A port 0
B port 0
D port 0
E port 1
F port 1
G port 1
H port 1

Bridge vs Repeater: Collision Domain vs Broadcast Domain



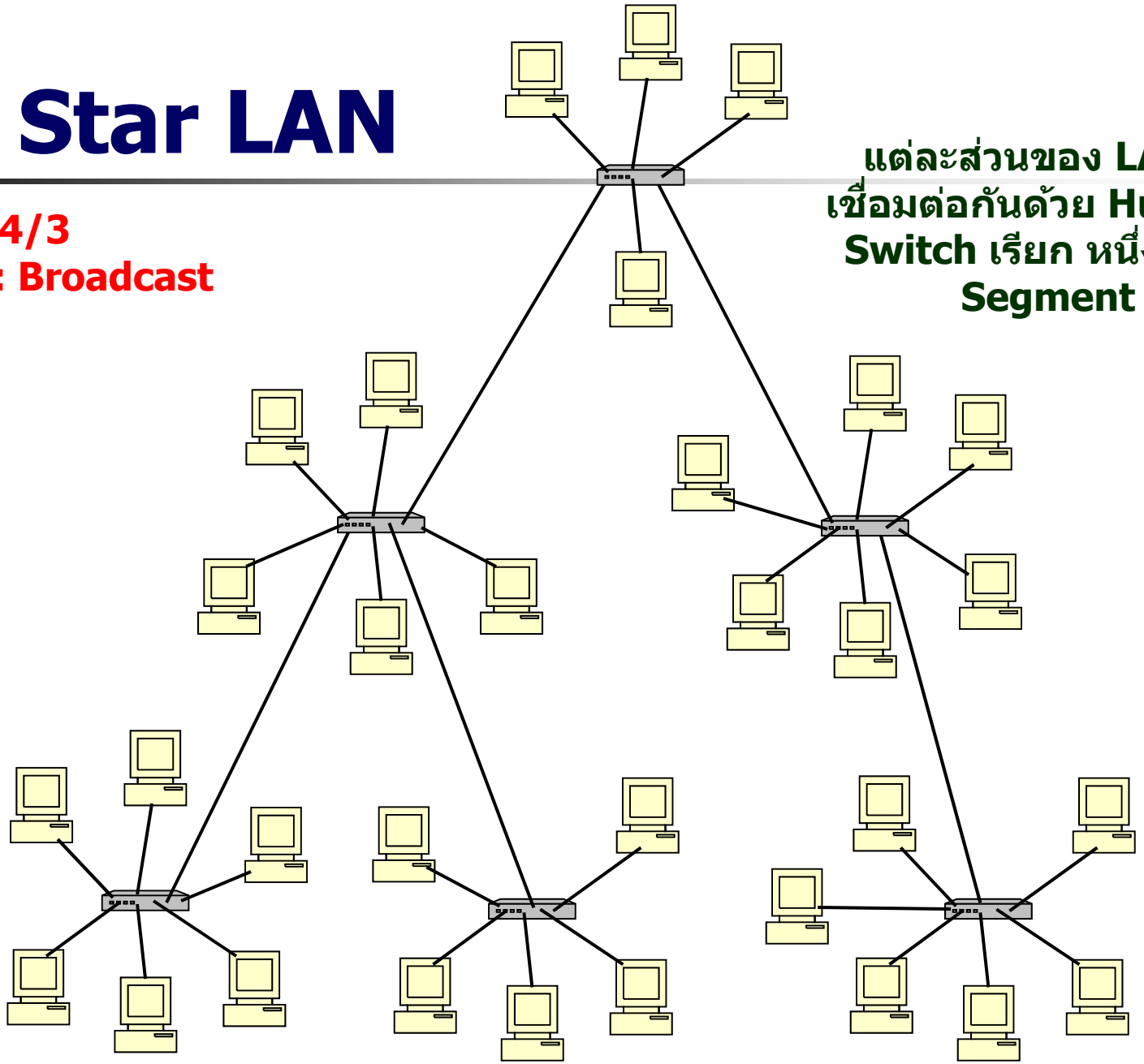
Repeater and Bridge in Star LAN

- **ใน Topology แบบ LAN ตัว Hub จะเป็น Repeater ที่มีหลาย Port ทำงานใน Layer 1**
  - ทุกๆอุปกรณ์ที่ต่อผ่าน Hub จะเป็น Collision Domain เดียวกัน
- **ต่อมาเราเปลี่ยนเป็น Switch ซึ่งมีการทำงานแบบ Bridge ที่มีหลาย Port ทำงานใน L 2 (MAC Layer)**
  - แต่ละอุปกรณ์ที่ต่อผ่าน Switch ถือว่าแต่ละ Port จะเป็นหนึ่ง Collision Domain
- **อย่างไรก็ตาม ไม่ว่าเราจะเชื่อมต่อด้วย Hub หรือ Switch ยังคงจัดว่าเป็น Network เดียวกัน (มี Network ID หรือ Subnetwork ID เดียวกัน) คือเป็น LAN วงเดียวกัน**
  - Broadcast Traffic จะกระจายทั้ง Network และทั้งหมดนี้ จัดได้ว่าเป็น Broadcast Domain เดียวกัน
  - การเชื่อมต่อระหว่างสอง Network ต้องใช้อุปกรณ์ Layer 3 คือ Router หรือ L3 Switch
    - Port ต่างๆของ Switch จะต่อคนละ Network และคนละ Broadcast Domain

Star LAN

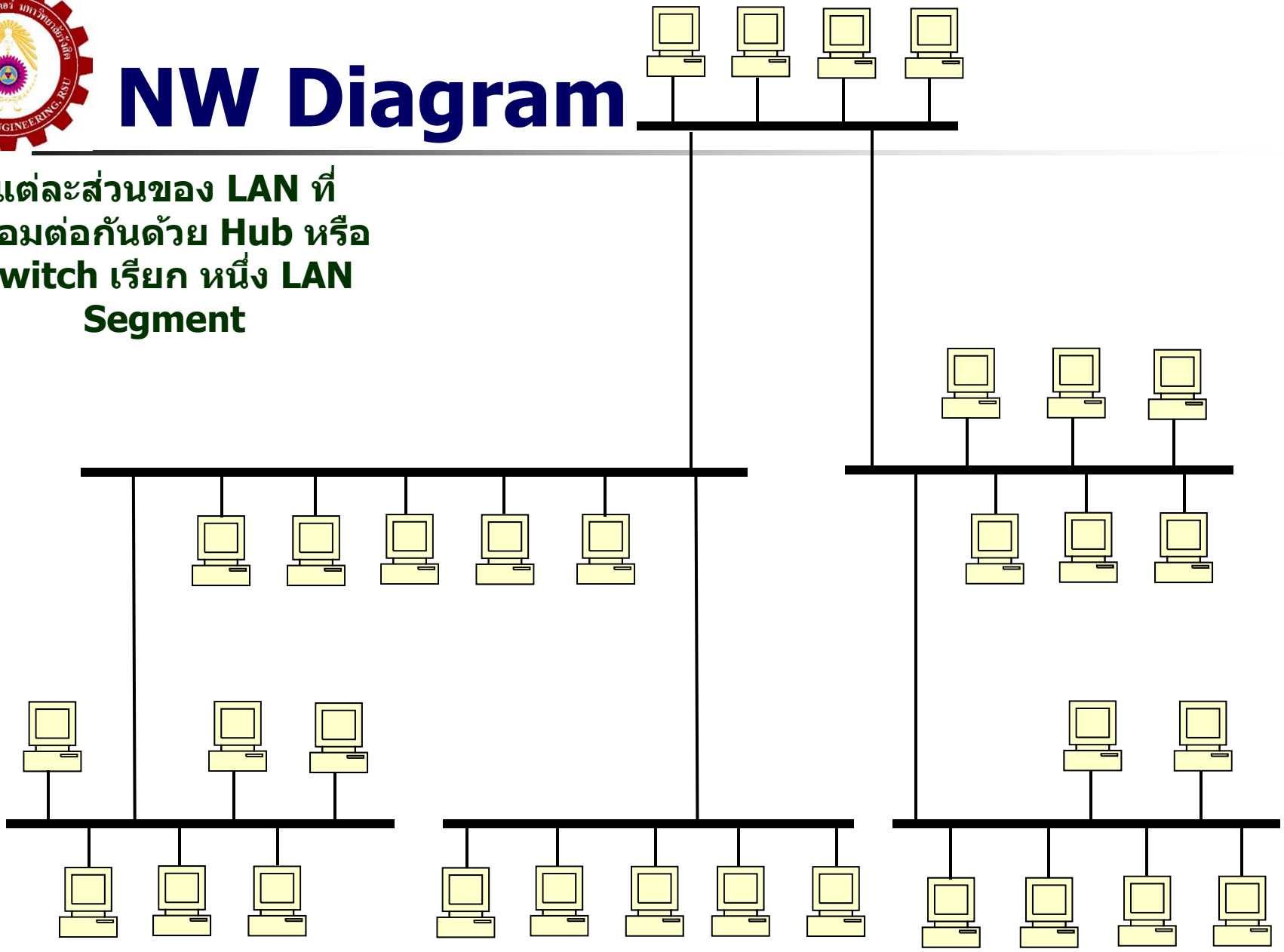
HUB: 5/4/3
SWITCH: Broadcast Traffic

แต่ละส่วนของ LAN ที่เชื่อมต่อกันด้วย Hub หรือ Switch เรียก หนึ่ง LAN Segment

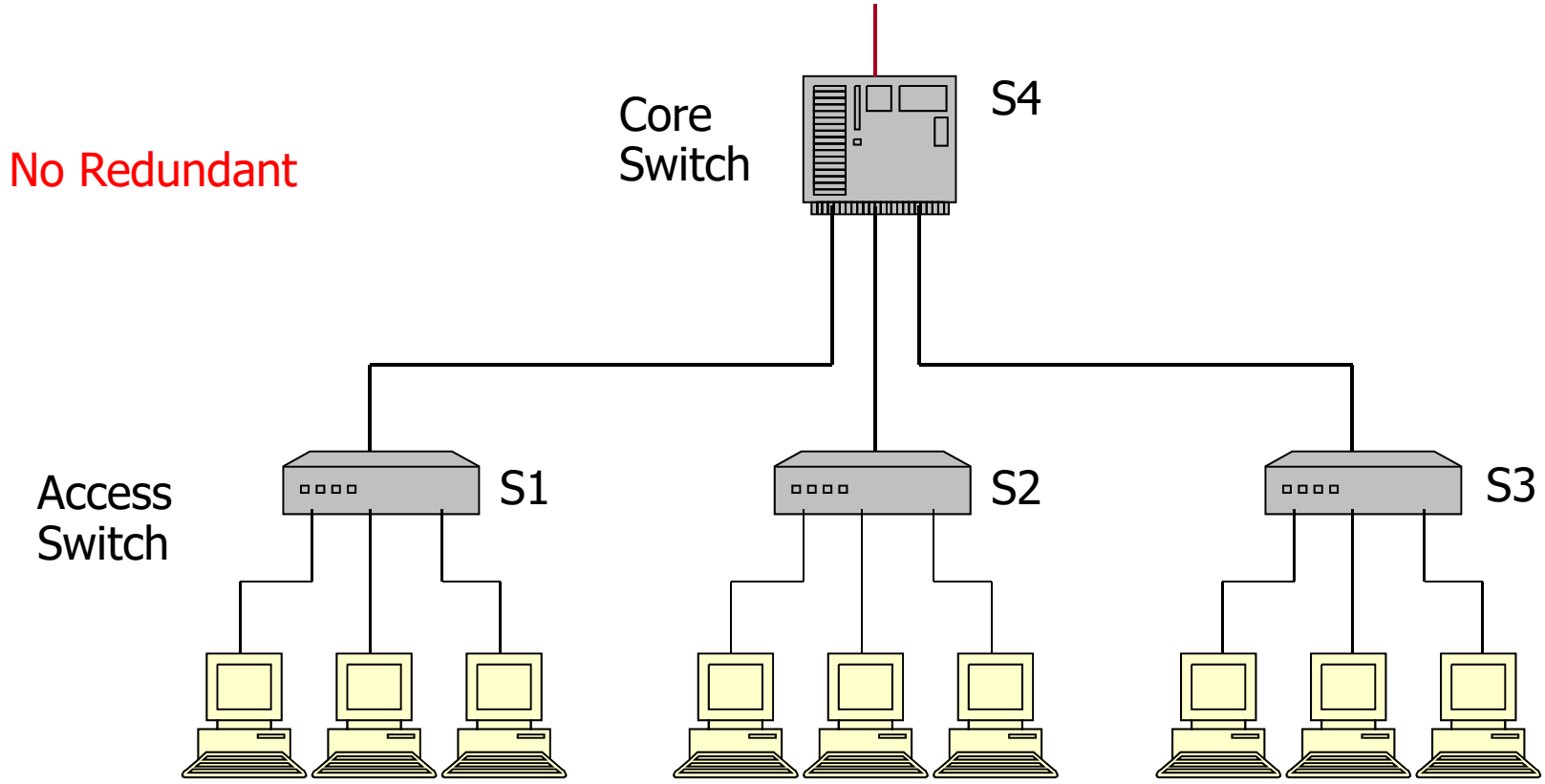


NW Diagram

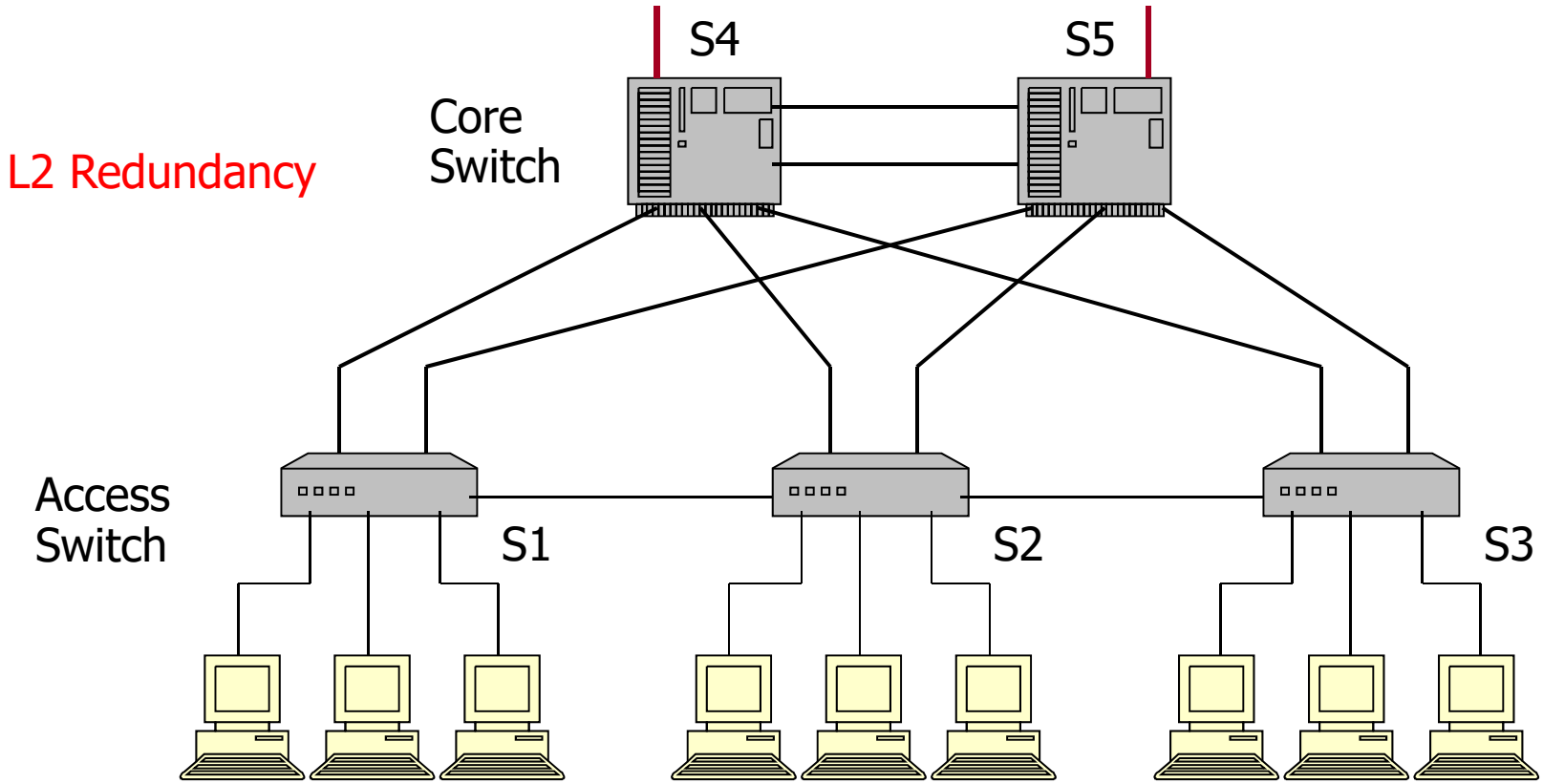
แต่ละส่วนของ LAN ที่เชื่อมต่อกันด้วย Hub หรือ Switch เรียก หนึ่ง LAN Segment



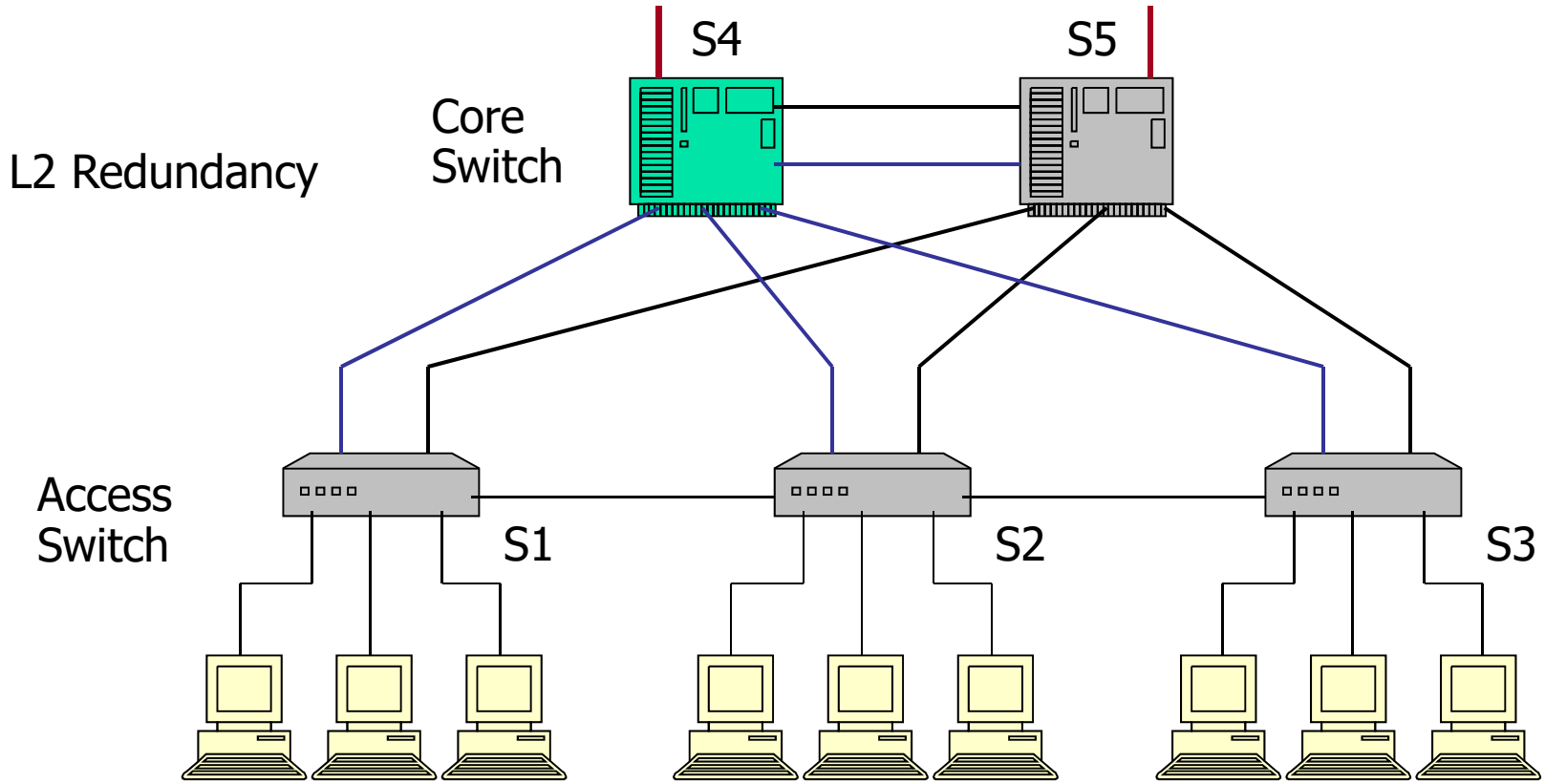
Layer 2: LAN No Redundancy



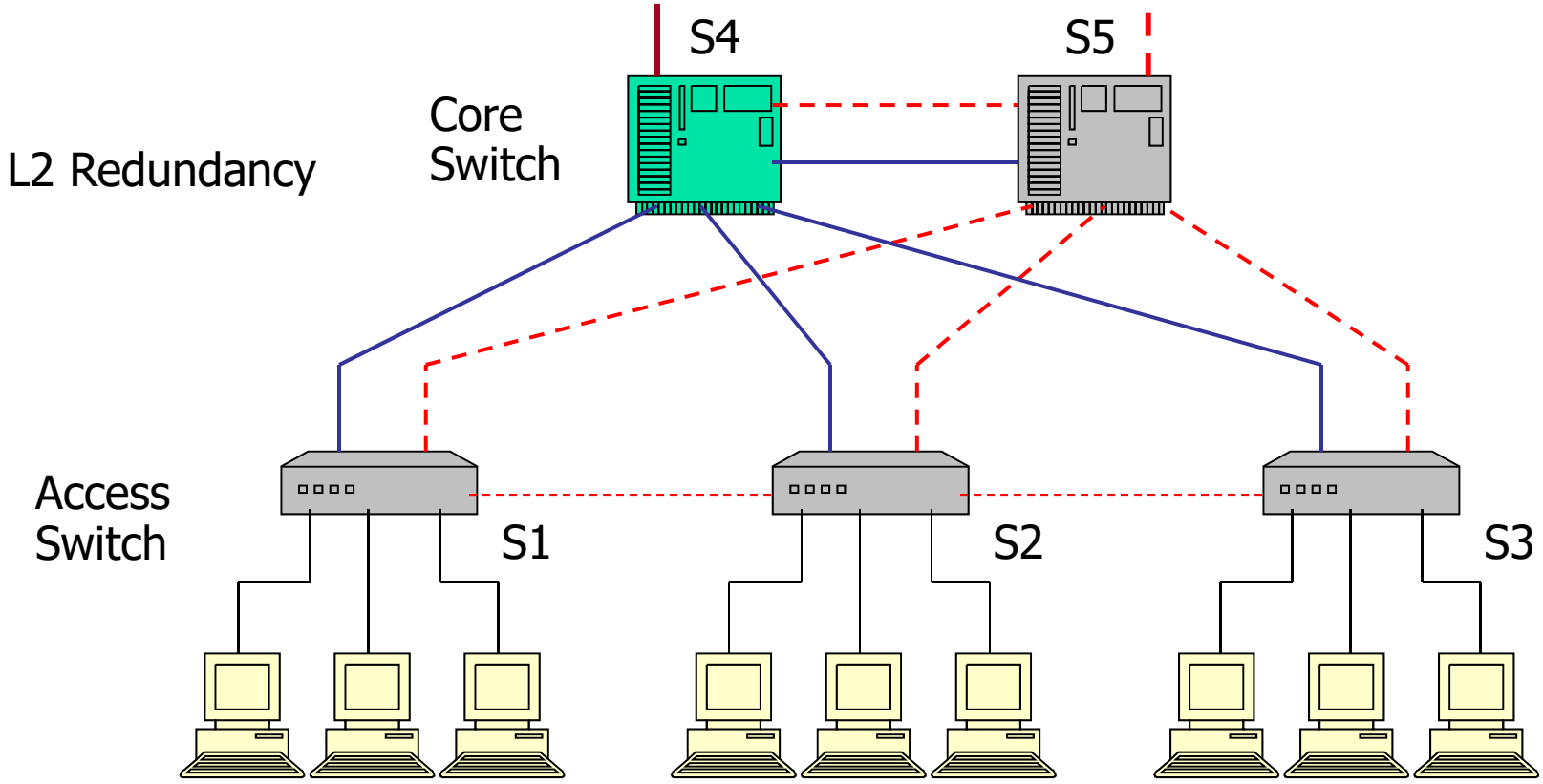
Layer 2: LAN with Link & Core Redundancy



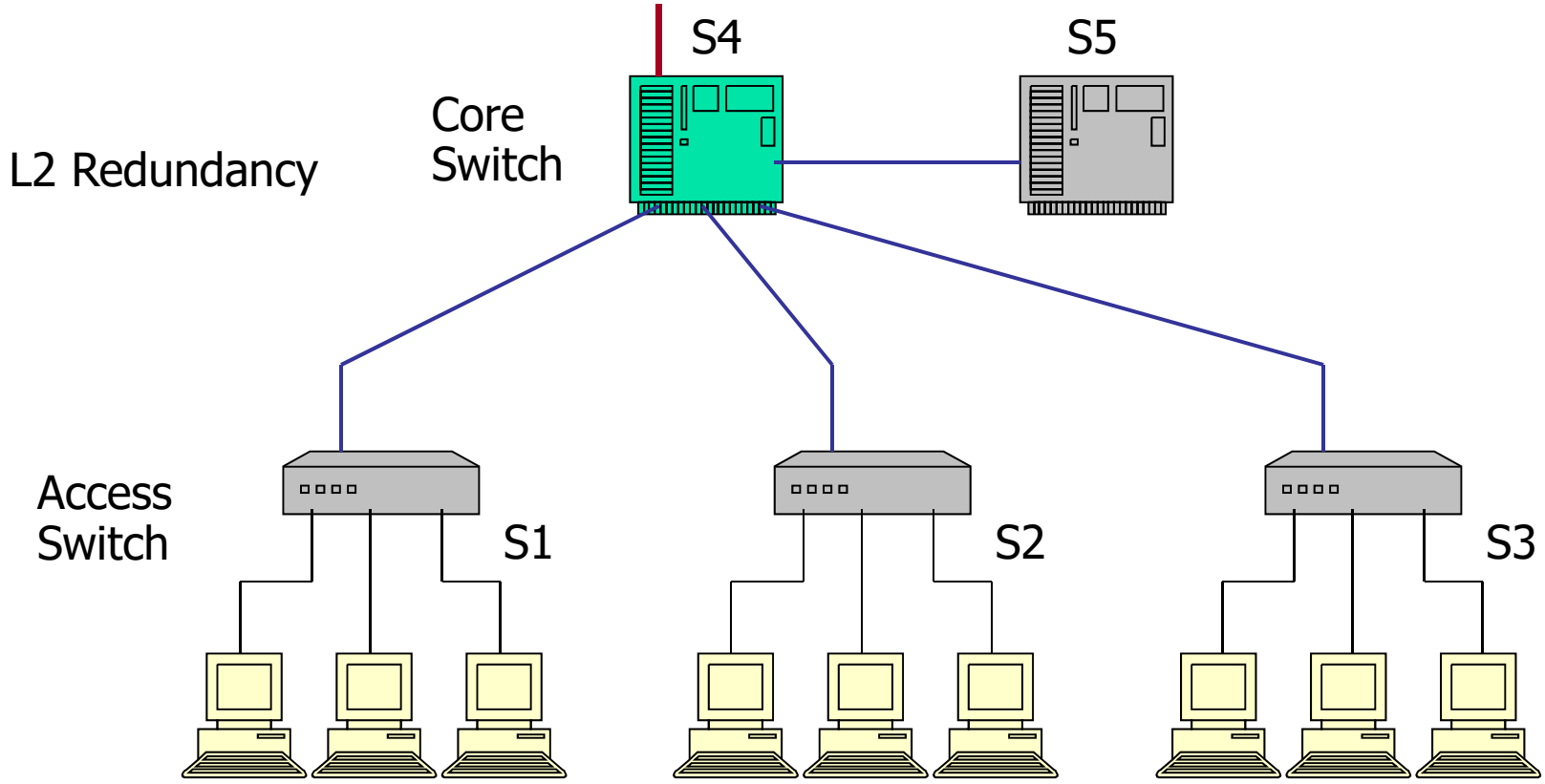
Layer 2 Redundancy: Active Link



Layer 2 Redundancy: SPT



Layer 2 Redundancy: SPT



Spanning Tree

- L2 Protocol
- LAN มี Loop ไม่ได้
- แต่เราต้องการสร้าง Redundancy
- ปิด Port ไม่ให้เกิด Loop
- เปิด Port เพื่อเปิดเส้นทาง เมื่อเส้นทางเก่ามีปัญหา
- IEEE 802.1D
- IEEE 802.1W
- IEEE 802.1S

Spanning Tree

- Transparent
- ทำงานโดยอัตโนมัติ
- บางครั้ง Tree ที่ได้อาจจะไม่เป็นที่เราต้องการ
- อาจต้องมีการ Configure
- ปกติเป็นการกำหนด Root Bridge จาก Bridge Priority

Steps 1: Root Bridge Selection

- เลือก Root Bridge โดยทุก Switch ส่ง BPDU ออกทุก Port และใส่ค่า Bridge ID
- Bridge ID = Bridge Priority(2 Octet) + MAC Address(6 Octet)
- Switch ที่มี Bridge ID ต่ำสุดจะเป็น Root
- Default Bridge Priority = 32768
- ถ้าไม่มีการ Configure ดังนั้น Switch ที่มี MAC Address ต่ำสุดจะได้รับเลือก

Steps 2: Minimum Cost Tree

- สร้าง Minimum Cost Tree โดยจาก Root ส่ง BPDU ที่มี Cost = 0 ออกทุกๆ Port ที่มันต่อ ซึ่งถูกจัดว่าเป็น Designated Port
- เมื่อ Switch ได้รับ BPDU มันจะบวกค่า Cost กับ Cost ของ Link ที่เข้ามา และส่งต่อ
- ถ้ามันได้รับมากกว่า 1 BPDU แสดงว่ามีมากกว่าหนึ่งเส้นทางไปยัง Root (Loop)
- เลือกเส้นทางที่ Cost ต่ำกว่า เป็น Root Port
- ถ้ามีมากกว่าหนึ่งเส้นทางและ Cost เท่ากัน เลือก Port ไปยัง Bridge ID ต่ำกว่า
- ถ้ายังเท่ากันเลือก Port Priority ต่ำกว่า

Steps 3: เลือก Designated Port และ Port Blocking

- เส้นทางที่ไม่ได้ถูกเลือกจะถูกปิด
- การปิด ทำโดย Blocking Port
- Port จะถูกปิดด้านเดียว
  - ปิด Port ที่มี Cost สูงกว่าไปยัง Root ถ้าเท่ากัน
  - ปิด Port Switch ที่มี Bridge ID สูงกว่า ถ้าเท่ากัน
  - ปิด Port ที่มี Port ID สูงกว่า
    - Port ID = Port Priority(1 Byte, Default = 128) + Port Number
- Port ที่เปิดเรียก Designated Port

Cost Table

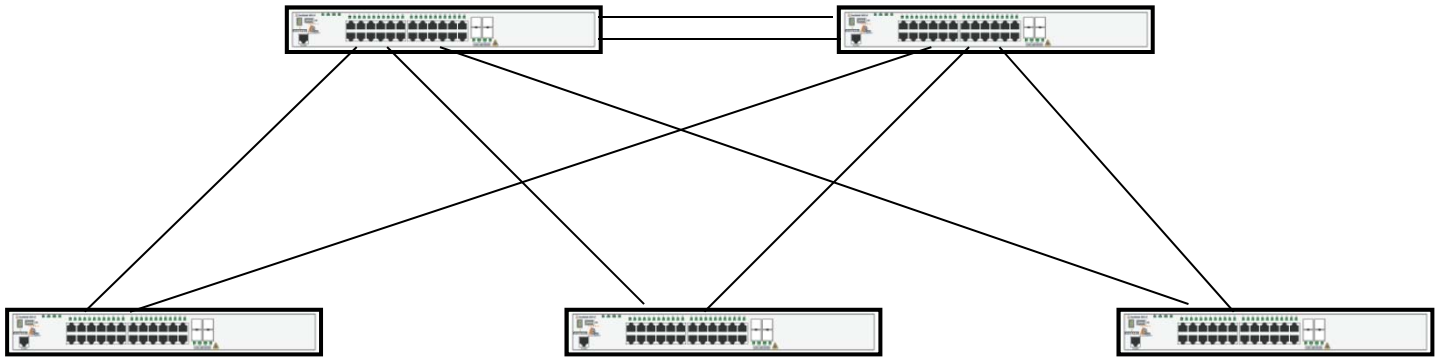
Link Bandwidth	Path Cost (Old Version)	Path Cost (New Version)
4 Mbps	**250**	**250**
10 Mbps	**100**	**100**
16 Mbps	**63**	**62**
45 Mbps	**22**	**39**
100 Mbps	**10**	**19**
155 Mbps	**6**	**14**
622 Mbps	**2**	**6**
1 Gbps	**1**	**4**
10 Gbps	**0**	**2**

Spanning Tree Protocol

- **STP เป็น Protocol และ Algorithm ที่จะแก้ไขปัญหา Loop ที่เกิดใน LAN(เชื่อมต่อด้วยLayer 2 Switch)**
- **มาตรฐานคือ IEEE 802.1D**
- **ประกอบด้วย Root Bridge และ Set ของ Port บน Switch ที่มี Cost ต่ำสุด ที่จะส่ง Traffic มายัง Root**
  - Root Bridge เป็น SW ที่มี Bridge ID ต่ำสุด
- **Switch Port ที่ไม่ได้เป็นส่วนหนึ่งของ Tree จะถูก Disable ดังนั้นจะมีเพียง Path เดียวระหว่าง 2 Station**
- **แต่ละ Switch จะส่ง Bridge Protocol Data Unit (BPDU) ให้แก่กันเพื่อรักษา Spanning Tree**
- **BPDU จะถูกส่งเมื่อ State ของ Port เปลี่ยน**
- **นอกจากนี้ BPDU จะถูกส่งทุกๆ 2 วินาที**
- **Root Bridge ควรจะเลือก Manually จาก Switch กึ่งกลางที่ความเร็วสูง**

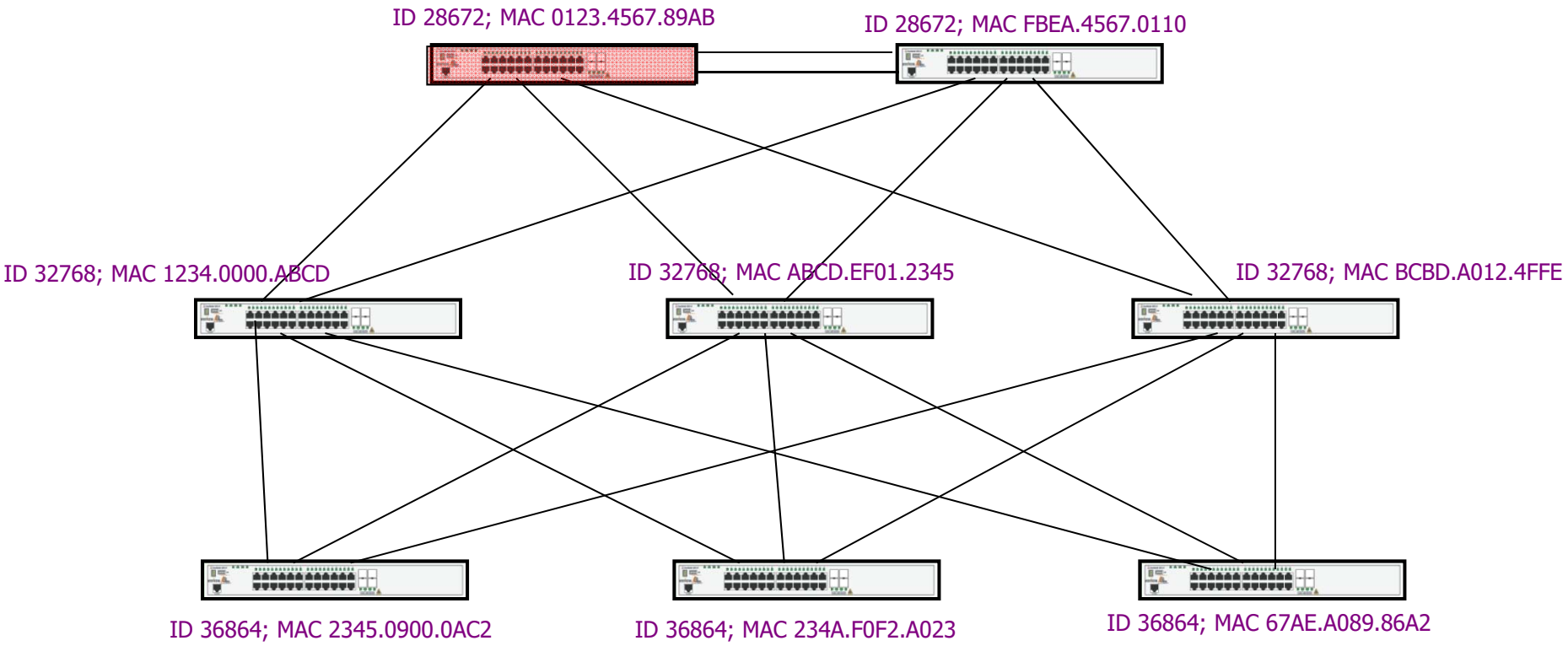
Order of Precedence

- 1. Lowest Root Bridge ID
- 2. Best Root Path Cost
- 3. Lowest Bridge ID that Send BPDU
- 4. Port ID



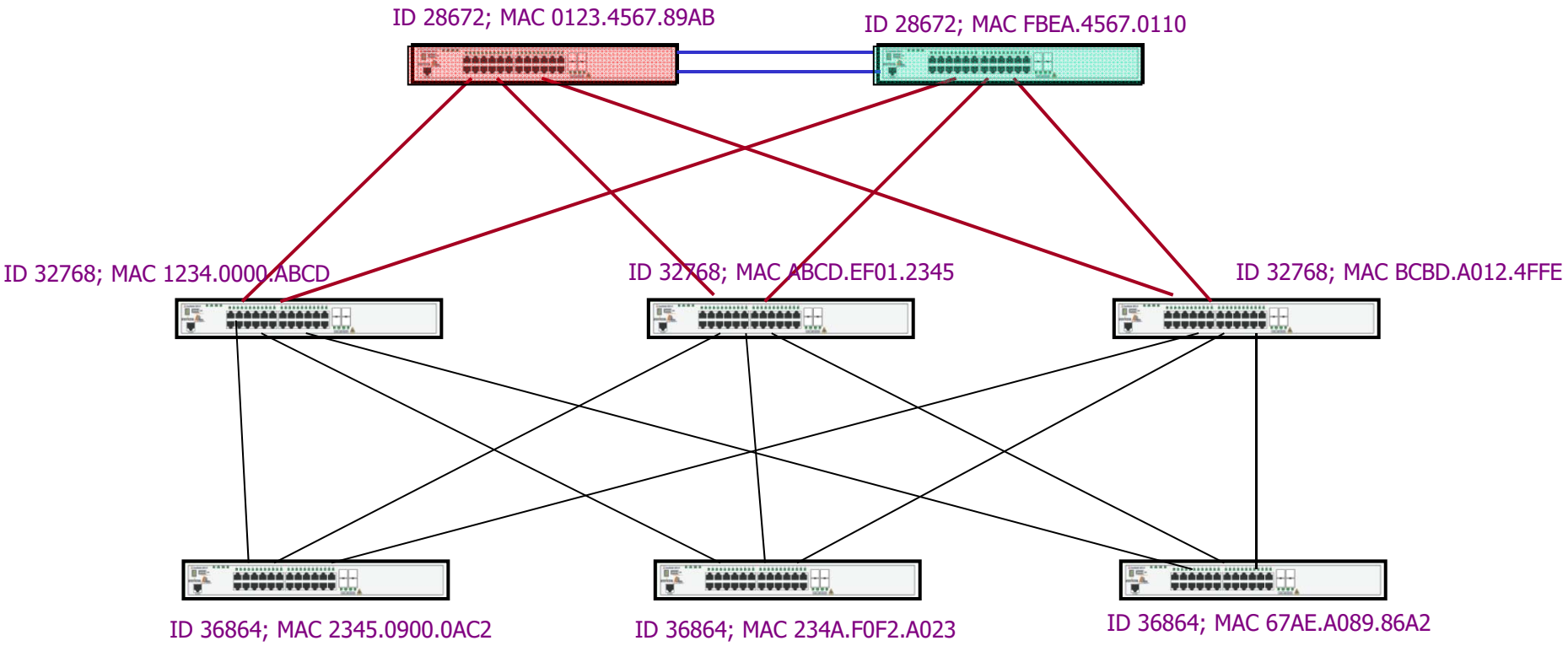
Order of Precedence

เลือก Root Bridge



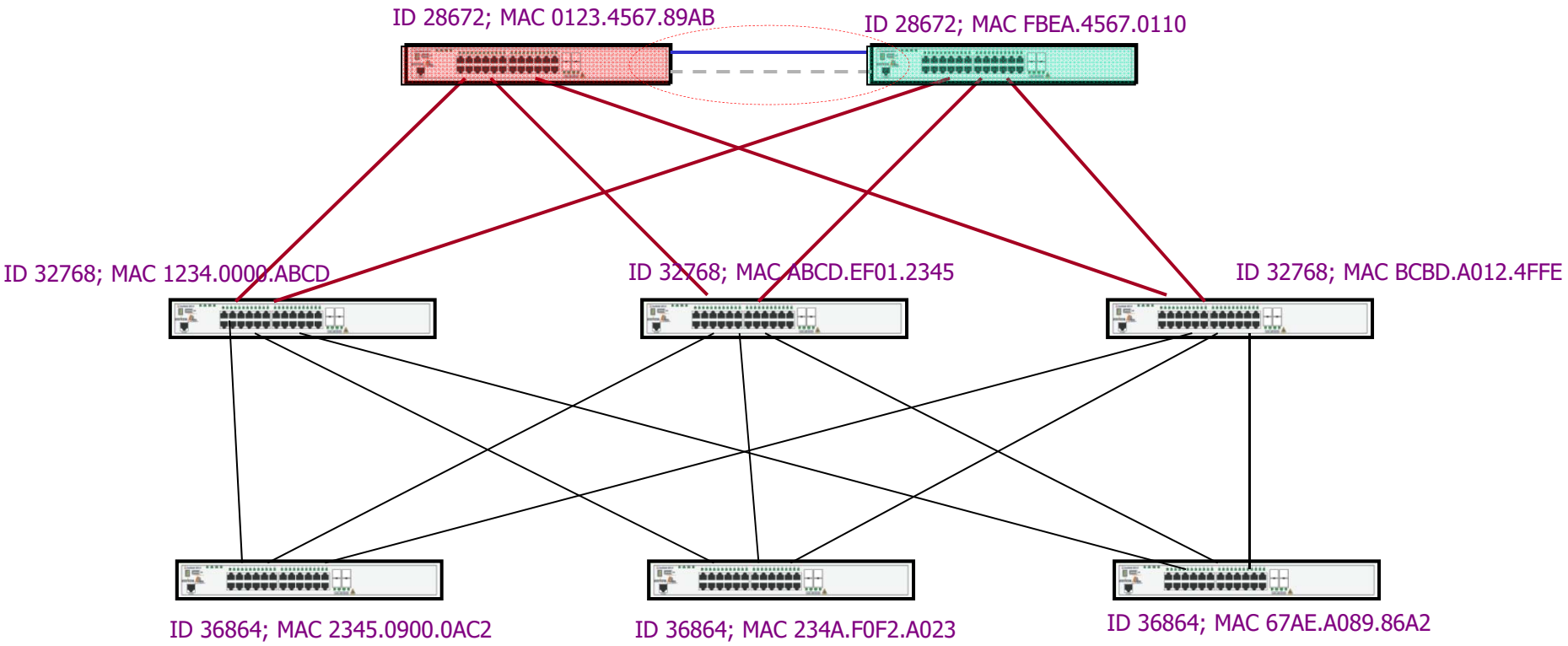
Order of Precedence

กำหนด Designated Port และส่ง BPDU จาก Root



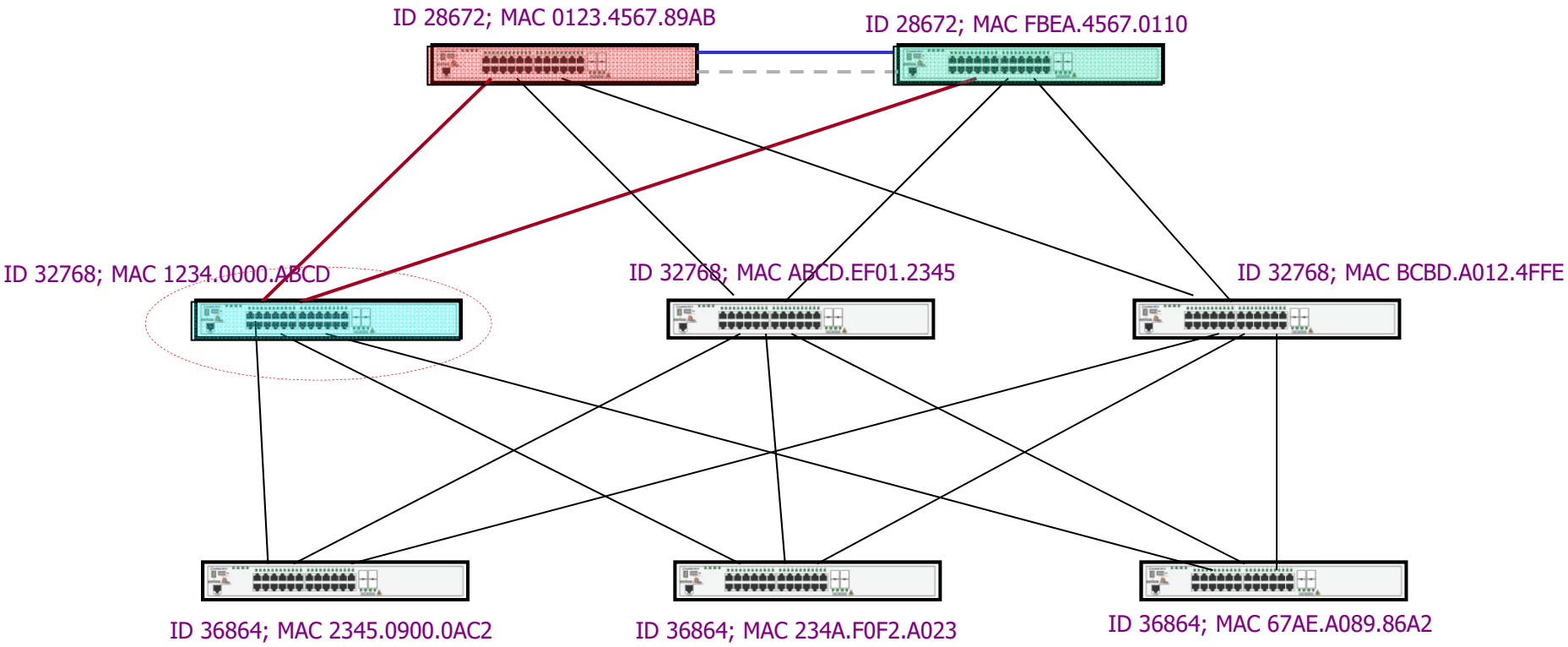
Order of Precedence

สร้าง SPT และ Block Link ที่ไม่เป็นส่วนของ Tree ตามกฎ Precedence Rule

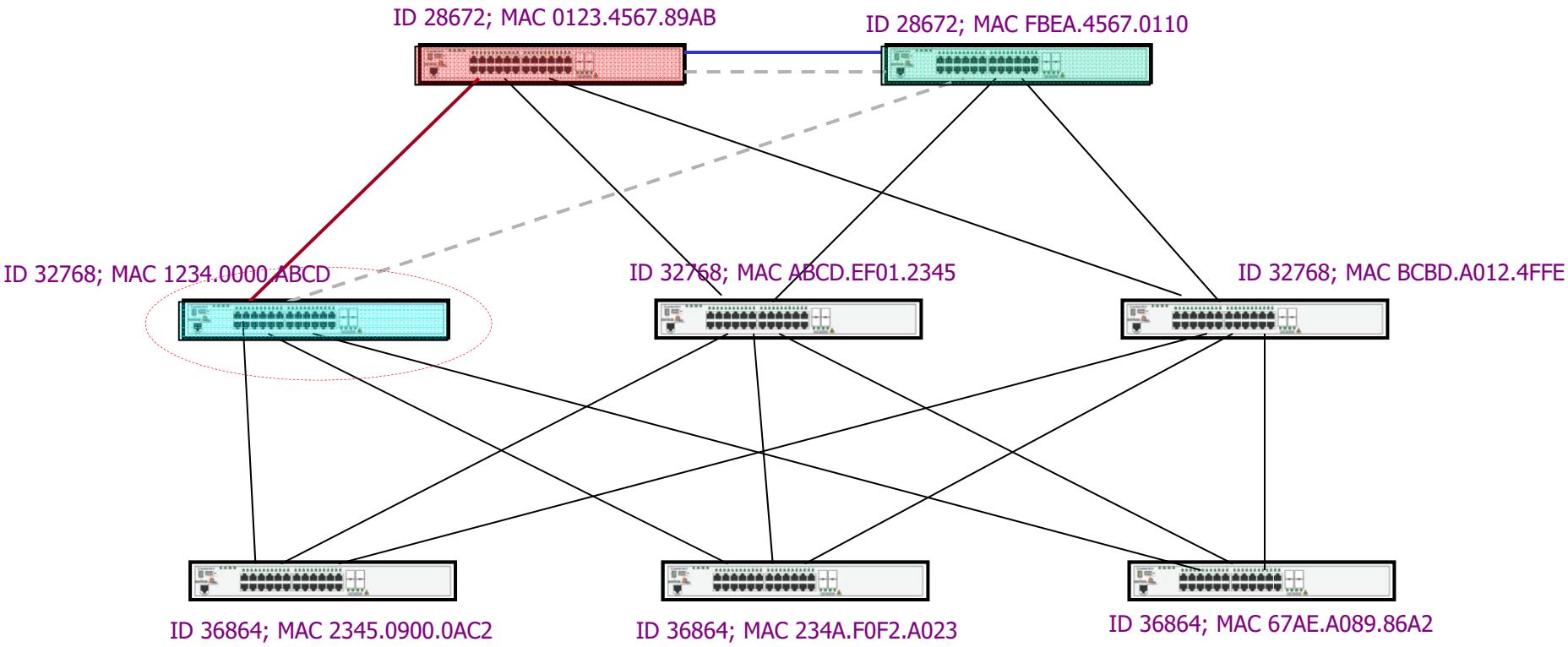


Order of Precedence

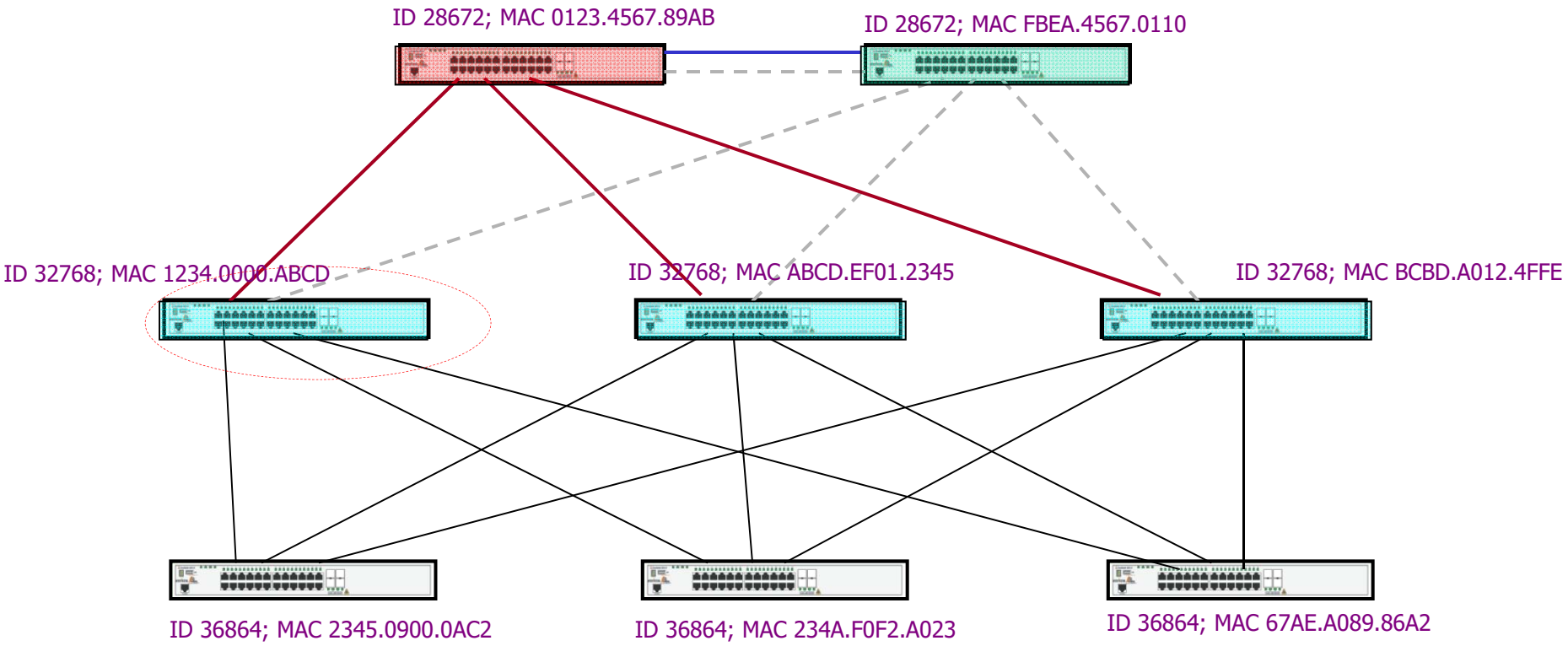
สร้าง SPT และ Block Link ที่ไม่เป็นส่วนของ Tree ตามกฎ Precedence Rule



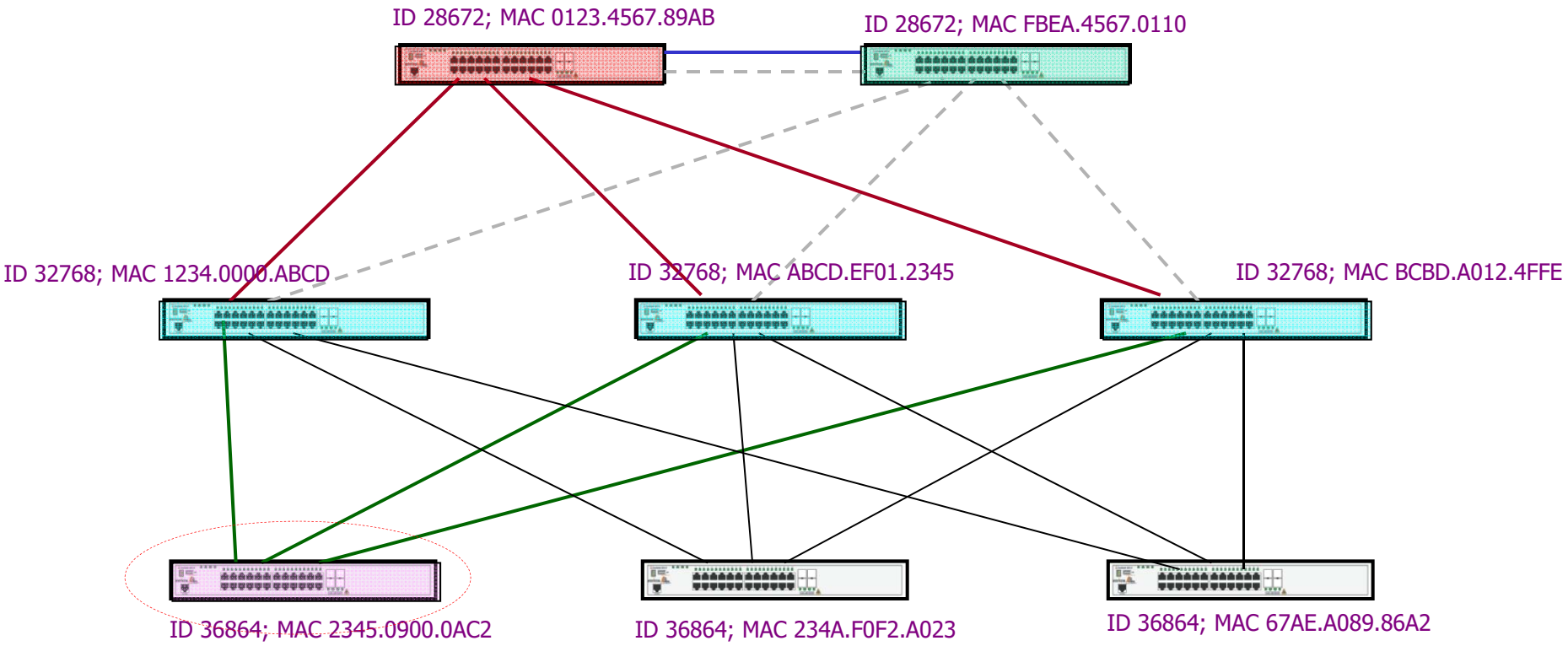
Order of Precedence



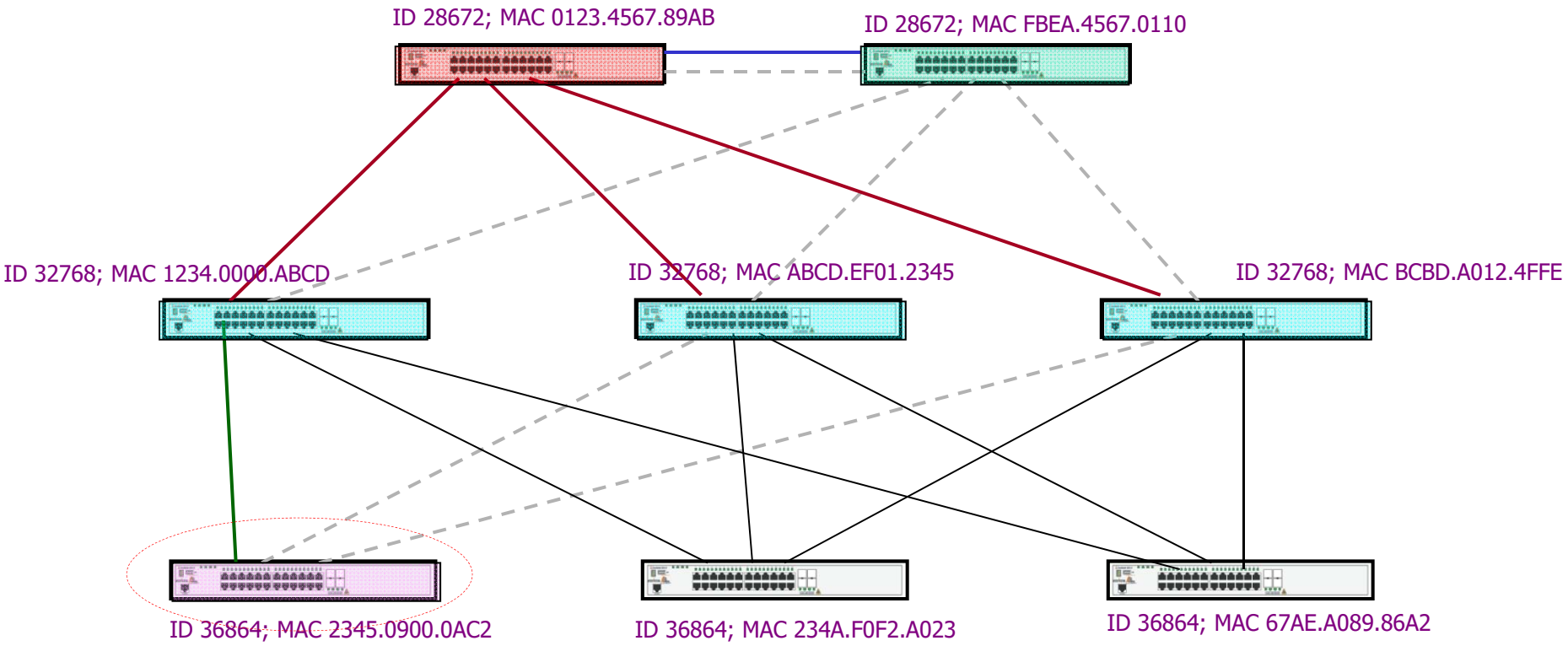
Order of Precedence



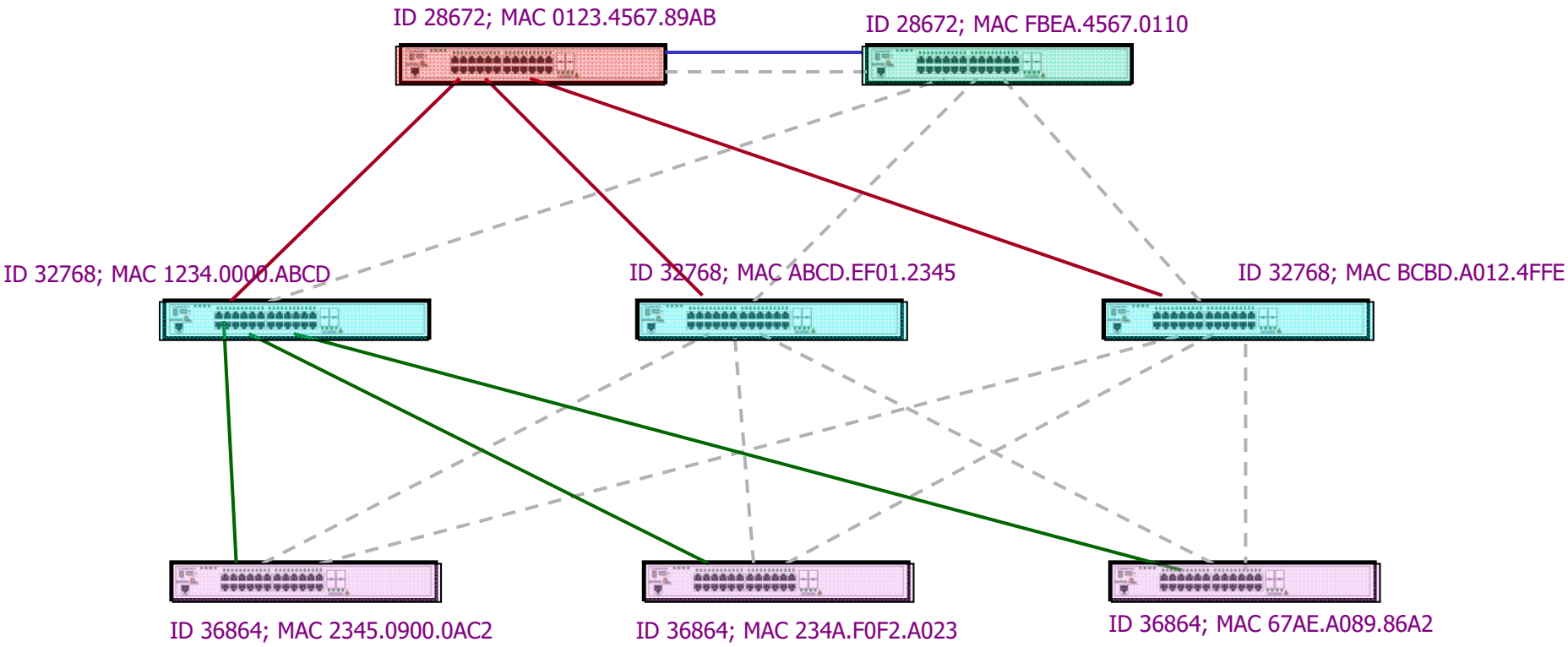
Order of Precedence



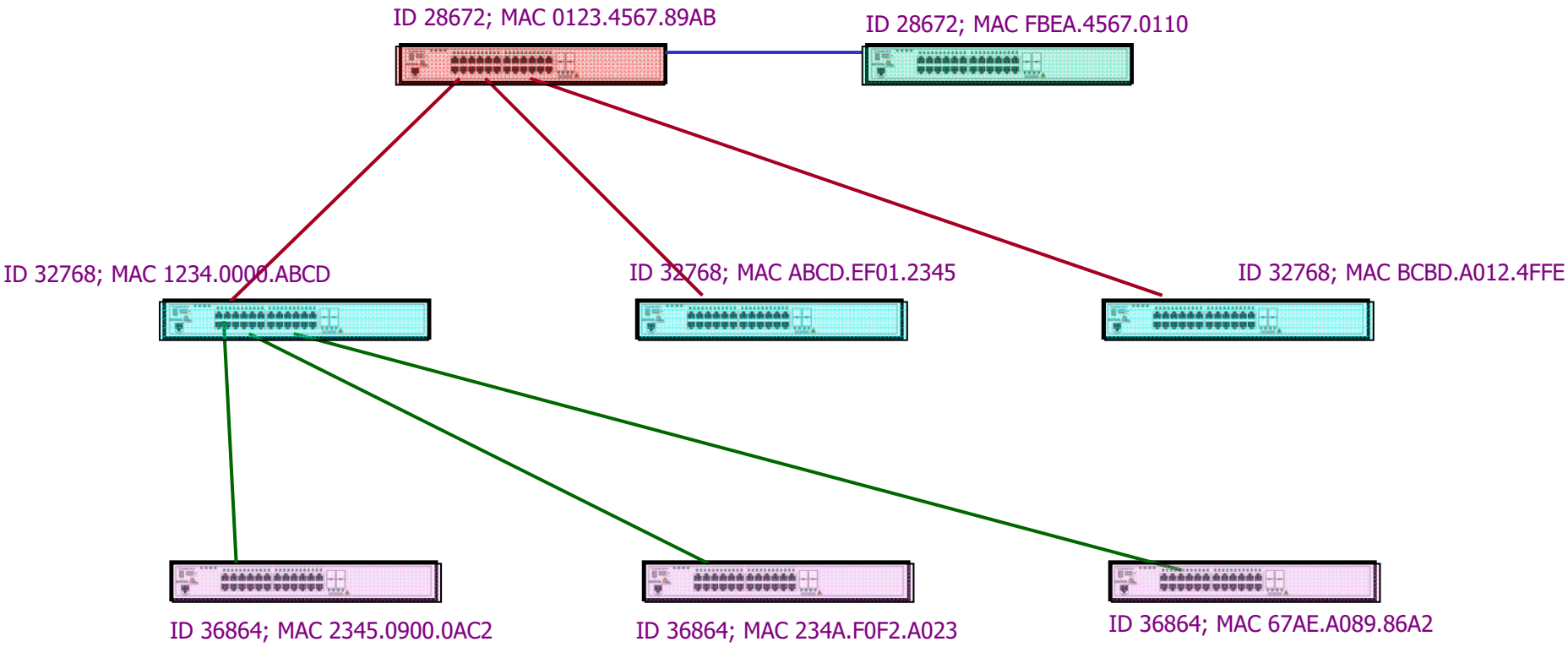
Order of Precedence



Order of Precedence



Order of Precedence



VLAN

- **แยก Broadcast Domain ออกภายใน Switch ตัวเดียว**
- **L2 Protocol**
- **เหมือนกับมีหลาย Switch ที่ไม่เชื่อมต่อกันในตัวเดียว**
- **สามารถทำการ Configure ได้ว่าจะแยกอย่างไร**
  - VLAN by Port (Static) กำหนดแต่ละ Port ตายตัวว่าเป็นของ VLAN อะไร
  - Dynamic VLAN : ตาม MAC, IP, Protocol หรืออื่นๆ กรณีนี้แต่ละ Port จะเปลี่ยน VLAN ตาม Condition ที่กำหนด เราเรียกว่าเป็น Mobile Port

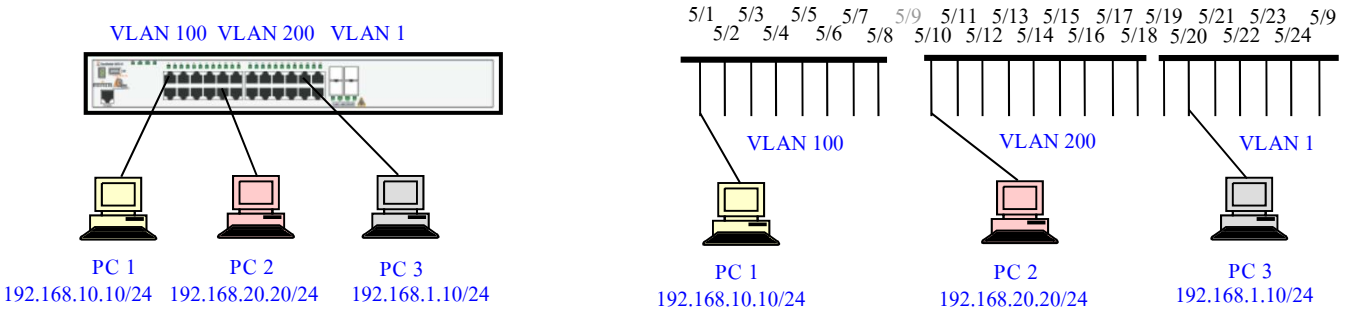
VLAN

- VLAN 1 คือ Default VLAN ลบและสร้างไม่ได้
- ทุก Port ถ้าไม่มีการกำหนดจะอยู่ใน VLAN 1
- VLAN Number = 12 Bit แต่ปกติการสร้าง จะให้หมายเลขระหว่าง VLAN 2 – VLAN 4094
- การเชื่อมต่อสอง VLAN ด้วยกันต้องใช้ความสามารถของ L3
- VLAN สามารถแยก Physical NW ออกจาก Logical NW

VLAN แบ่ง Switch เป็นหลายส่วน

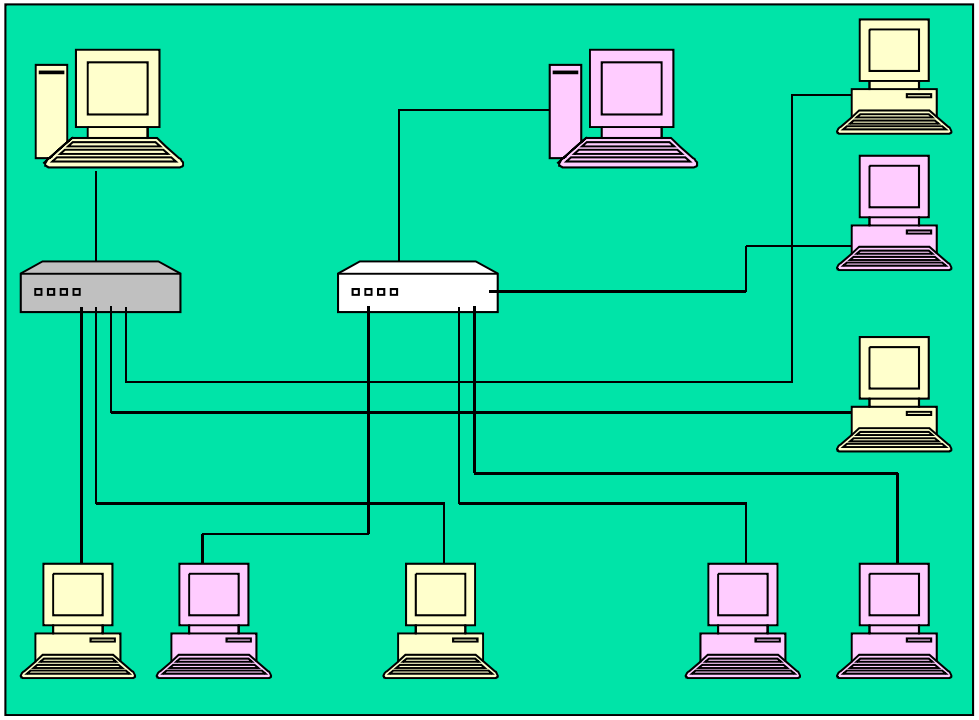
Switch ปกติเมื่อไม่แบ่ง VLAN หรือไม่ใช่ Managed Switch

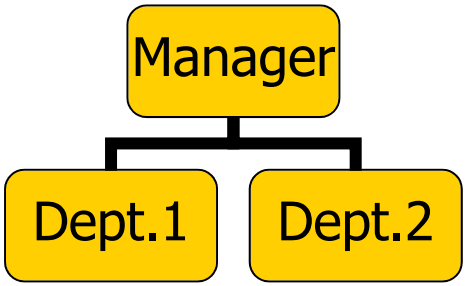
Switch ตัวเดียว ถูกแบ่งเป็น 3 VLAN



แต่ละ VLAN ถูกแยกออกจากกัน เสมือนอยู่คนละ Switch
จัดว่าอยู่คนละ Sub-network/Broadcast Domain
ต้องใช้อุปกรณ์ Layer 3(Router) มาเชื่อมต่อ

VLAN

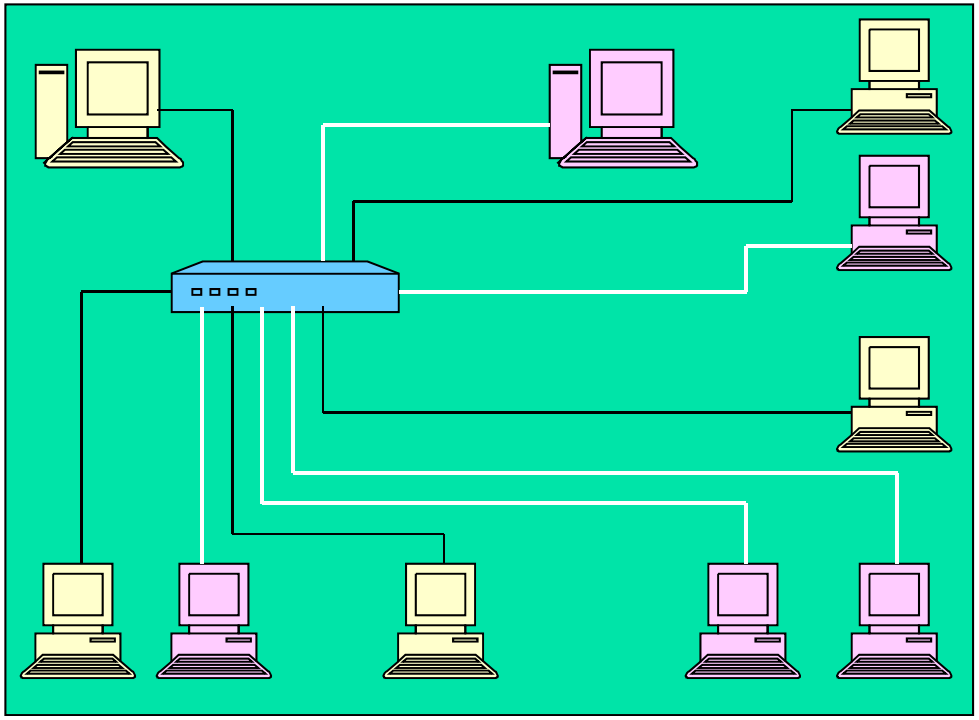


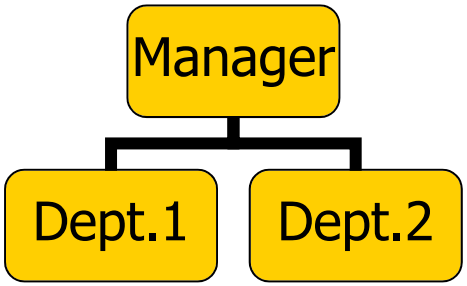


สอง Network สอง Server
ต้องการแยกออกจากกัน

ลงทุน สอง Switch
ปัญหาในการย้ายสถานที่
ต้องวางสายใหม่สำหรับ
Network ของตัวเอง

VLAN

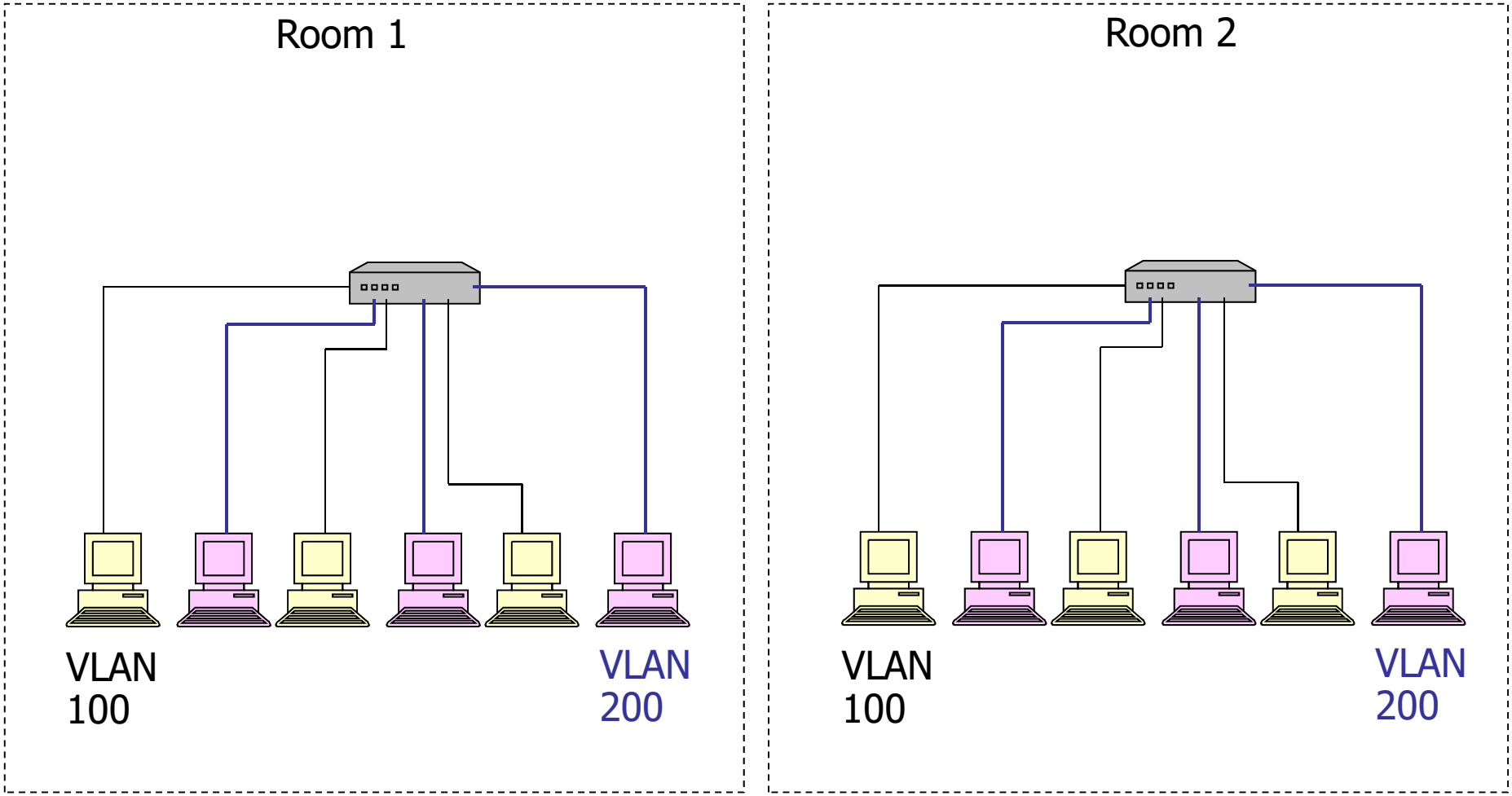




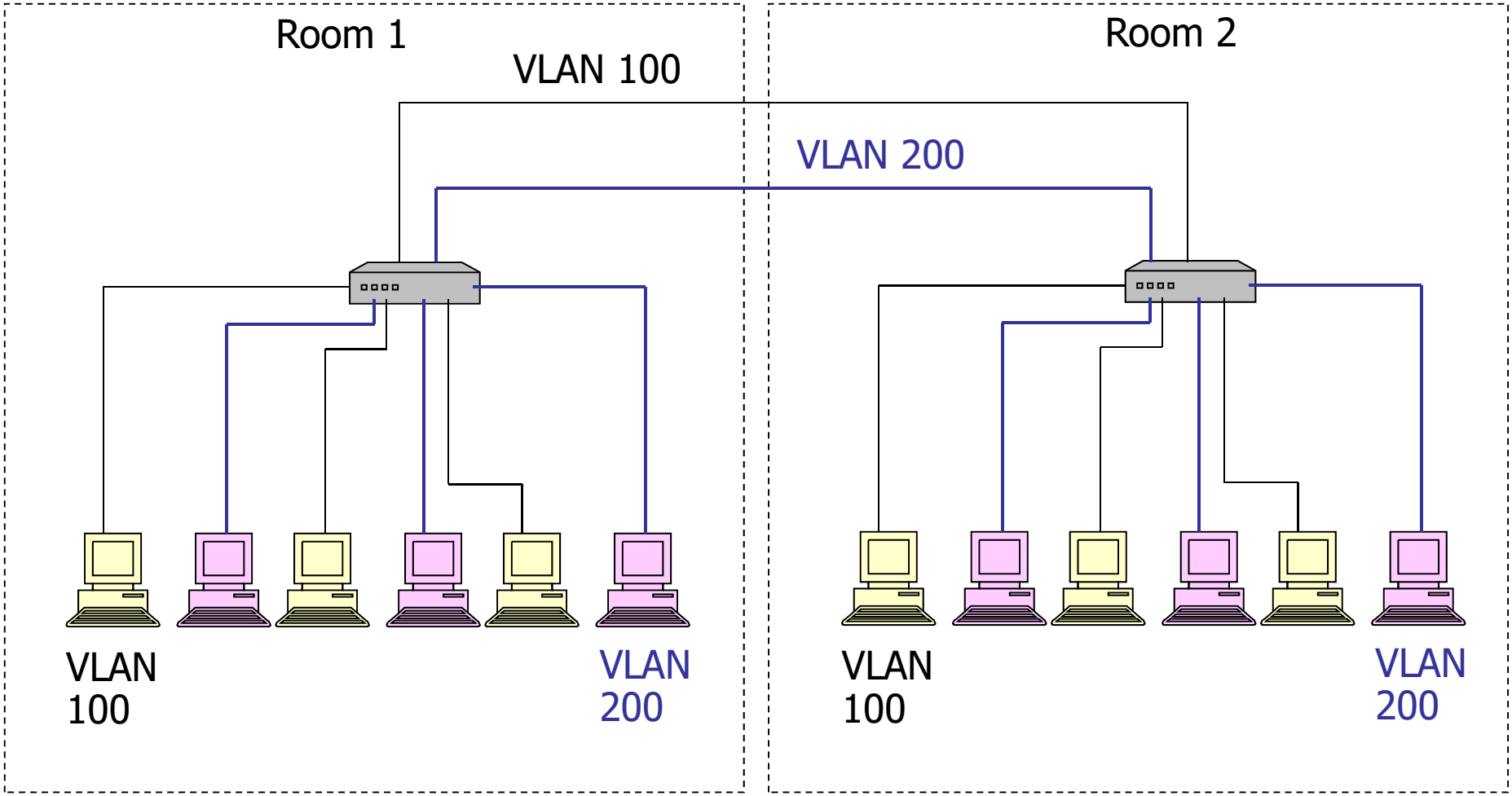
สอง Network สอง Server
ต้องการแยกออกจากกัน

ใช้ VLAN แก้ปัญหา
ย้ายที่ เชื่อมต่อกับ Port
ไหนของ Switch ก็ได้
แค่ Configure Port ให้ถูก VLAN

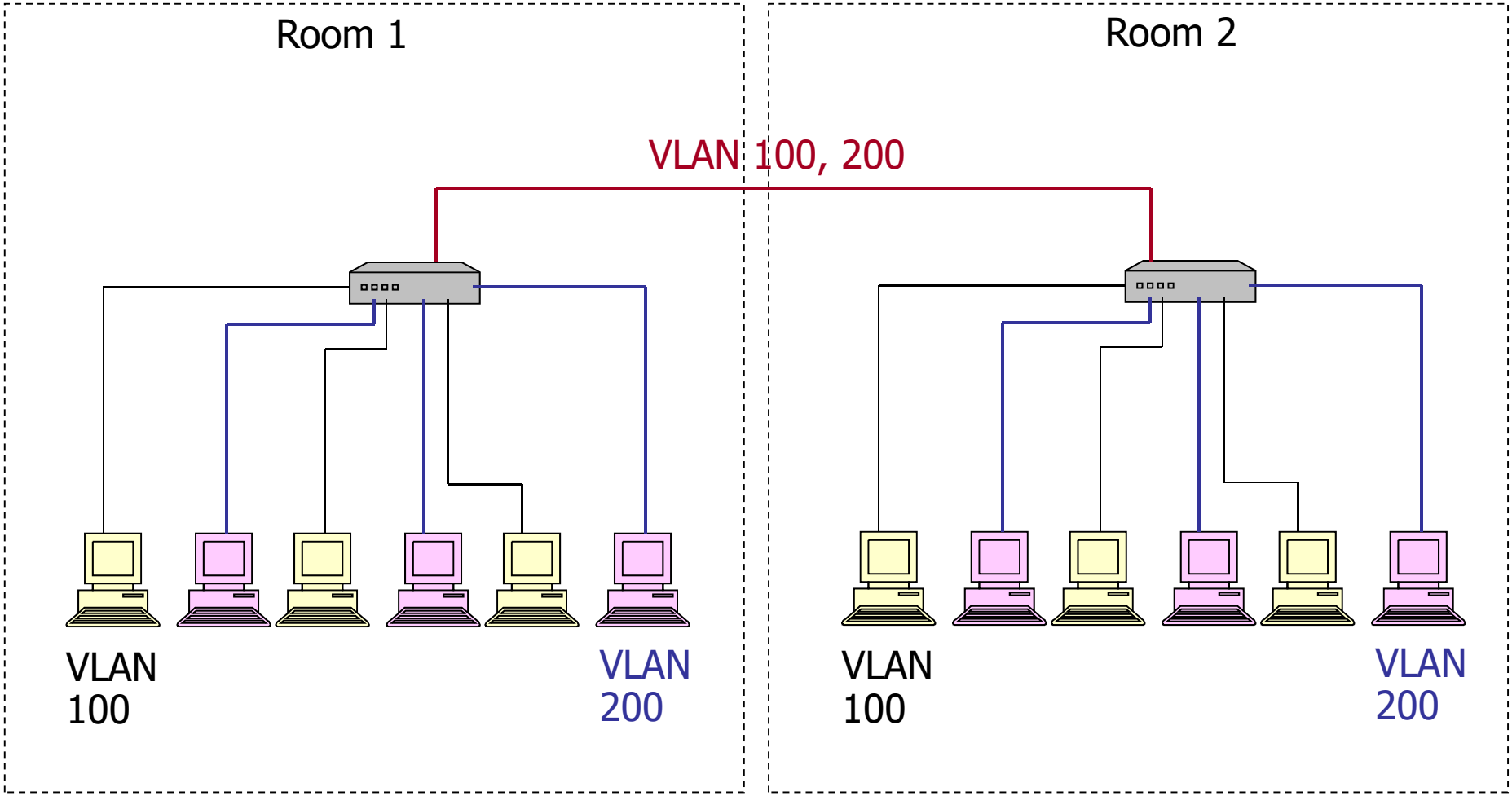
VLAN สามารถขยายผ่านมากกว่า 1 Switch



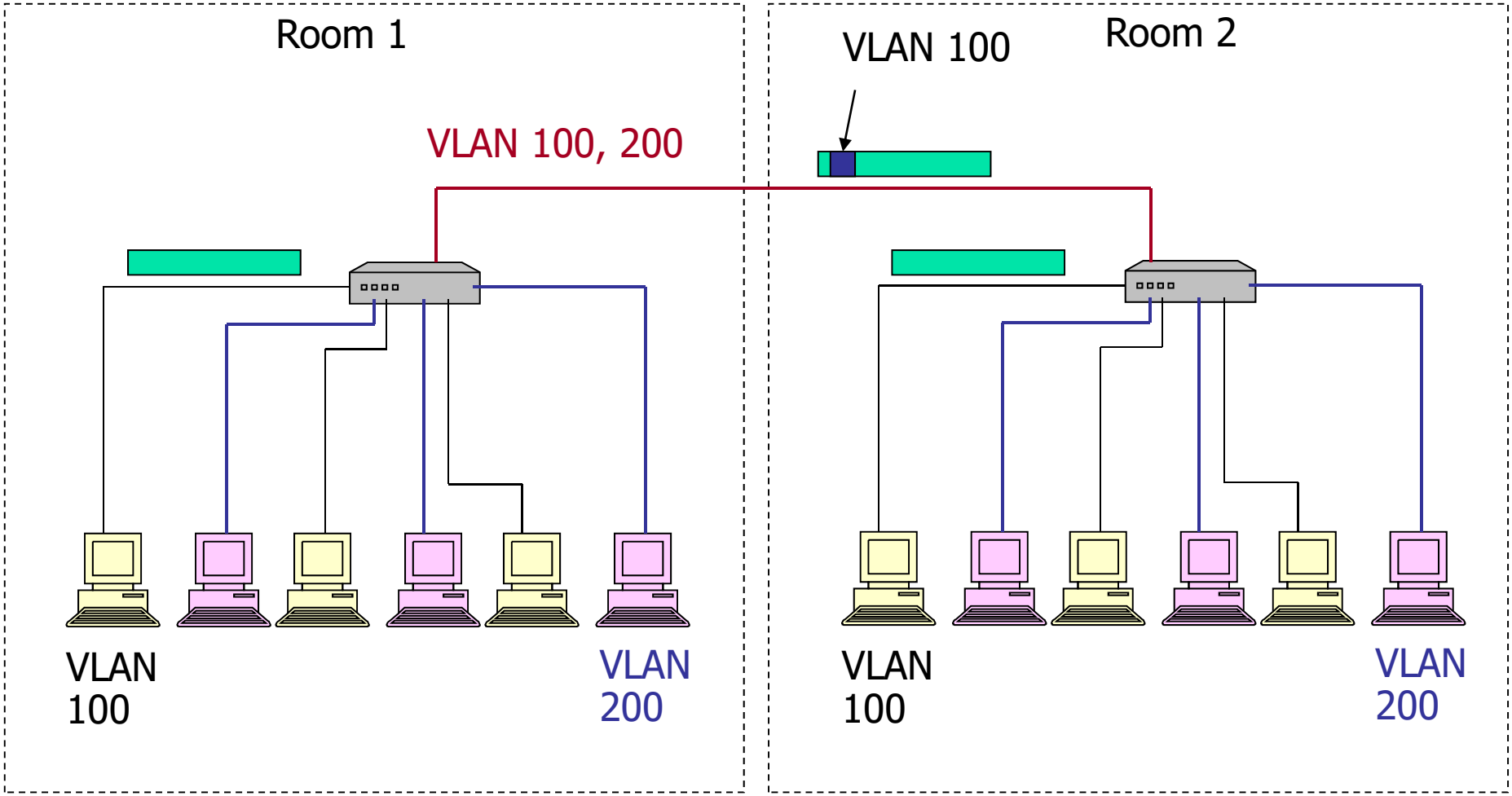
VLAN สามารถขยายผ่านมากกว่า 1 Switch



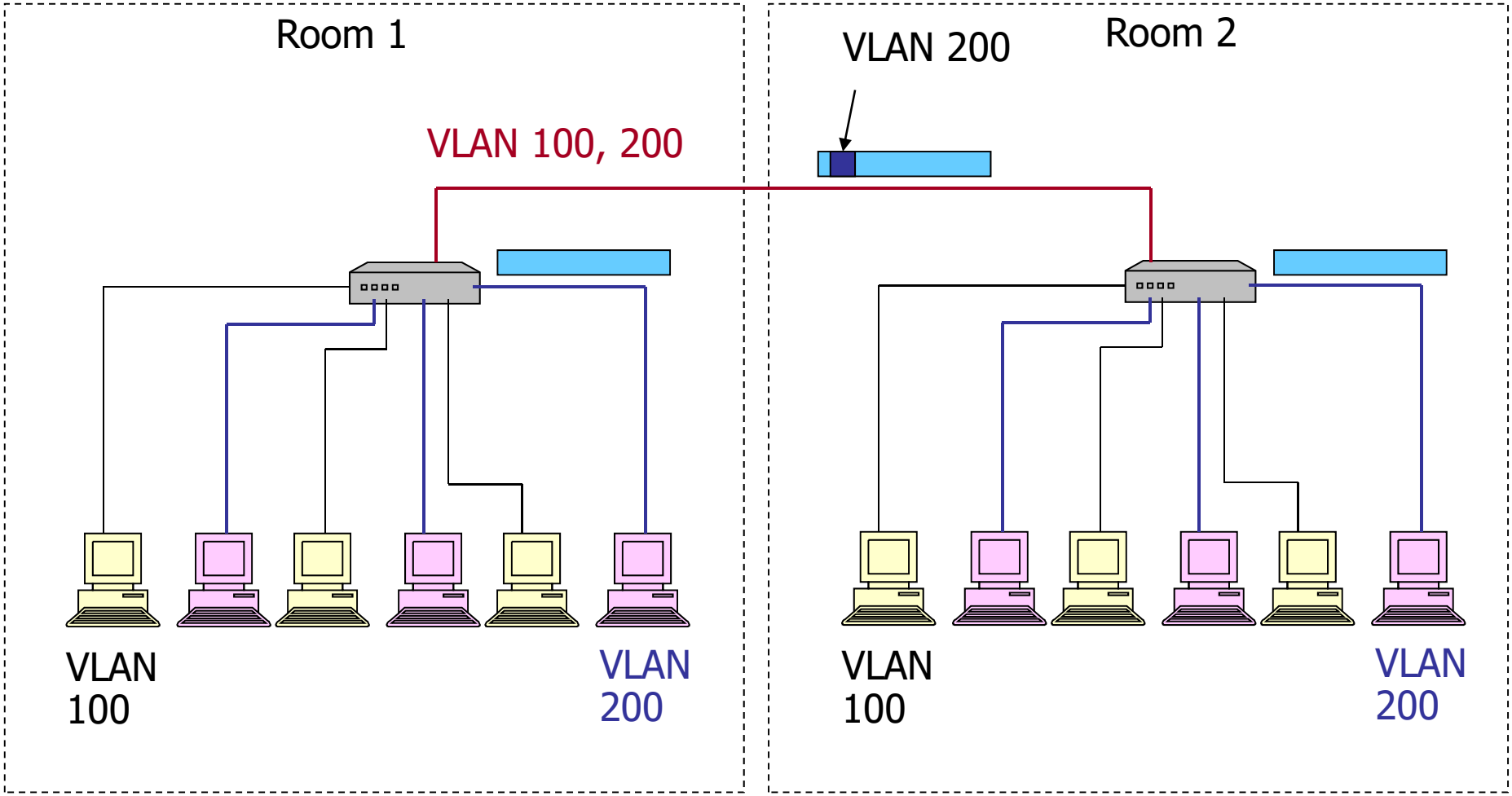
VLAN สามารถขยายผ่านมากกว่า 1 Switch



VLAN สามารถขยายผ่านมากกว่า 1 Switch



VLAN สามารถขยายผ่านมากกว่า 1 Switch



VLAN TAGGING

- **IEEE 802.1Q Standard**
  - 4 Byte เพิ่มในส่วนของ Header
  - 12 Bit เป็น VLAN Number
- **ISL(Cisco)**
  - Encapsulation

VLAN Tagging (IEEE 802.1Q)

- **Port ของ Switch จะต้องถูกกำหนดเป็น Tag Port**
- **เมื่อข้อมูลถูกส่งออกไปยัง Tag Port จะมีการใส่ Tag กำหนด VLAN**
- **เมื่อข้อมูลมาถึง Tag Port จะถูกส่งไปยัง VLAN ตาม Tag และตัว Tag จะถูกนำออก**
- **VLAN Default ของ Port นั้นจะไม่ถูกใส่ Tag**
  - VLAN Number จะเป็น Local ยกเว้นทำ Tagging
  - อุปกรณ์บางยี่ห้อจะมี Protocol สื่อสารระหว่าง SW (Interswitch Protocol)

Communication Between VLAN

- Connect Through Router (L3)
- Using L3 Switch ดีกว่า

VLAN Static vs Dynamic

- **เมื่อ VLAN ถูกกำหนดโดย Port ของ Switch เราเรียก Static VLAN**
  - อุปกรณ์ที่เชื่อมต่อกับ Port ดังกล่าวจะถูกจับไปอยู่ใน VLAN ที่กำหนด
- **แต่ถ้าเรากำหนดให้อุปกรณ์ที่มาเชื่อมกับ Port ไปอยู่ใน VLAN ตามคุณสมบัติของอุปกรณ์ เช่น ตาม IP Address, MAC Address หรือตามการ Authentication เราเรียก Dynamic VLAN**
  - Port ดังกล่าวจะเป็น “Mobile Port” และต้องกำหนด VLAN Rule ให้

การกำหนด VLAN

- **หนึ่ง Subnet ให้เป็น หนึ่ง VLAN**
- **เมื่อเรากำหนด Topology เราได้**
  - Subnet ของแต่ละ Network
  - กำหนด IP Address ให้กับแต่ละ Subnet
  - กำหนด VLAN ให้กับแต่ละ Subnet
    - ดังนั้นแต่ละ Subnet สามารถอยู่ร่วมกันบน Switch เดียวกันได้
    - แต่ละ Subnet สามารถกระจาย ครอบคลุมหลาย Switch ได้
    - กล่าวคือ Logical Network(Diagram) และ Physical Network(Wiring Diagram) สามารถแยกจากกัน
      - Network จะประกอบด้วยสอง Diagram

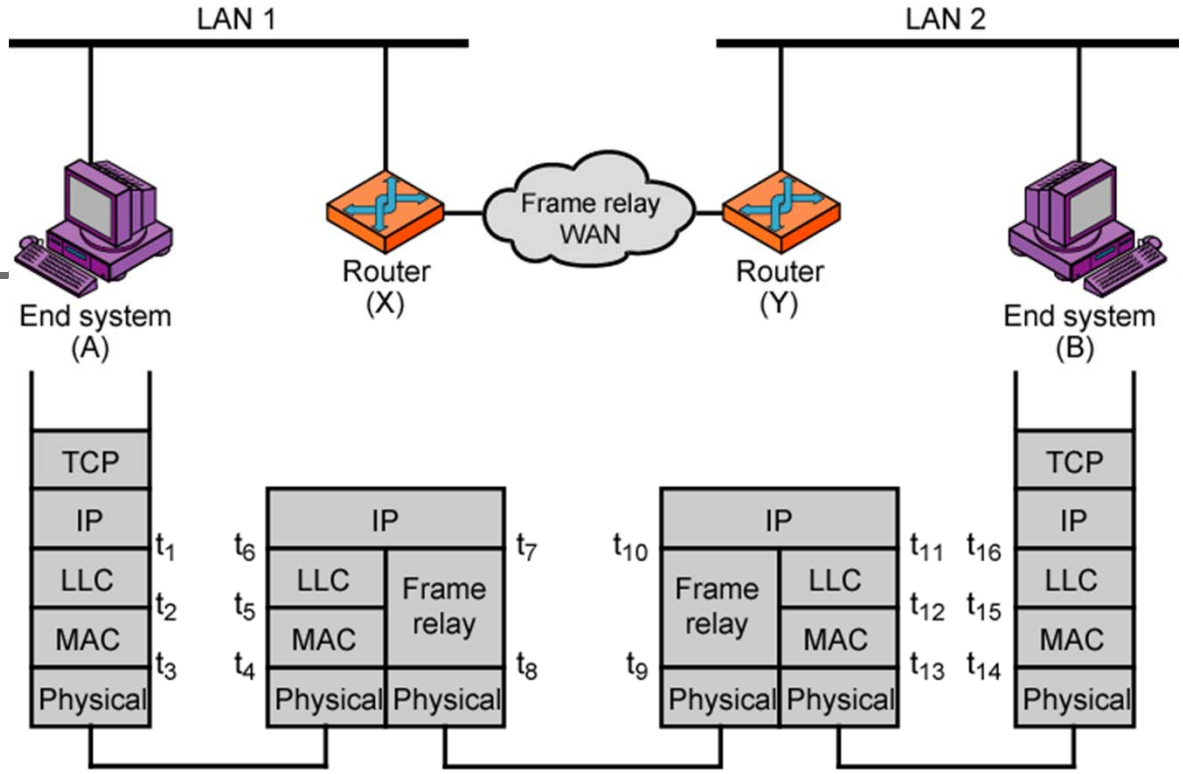
Spanning Tree and VLAN

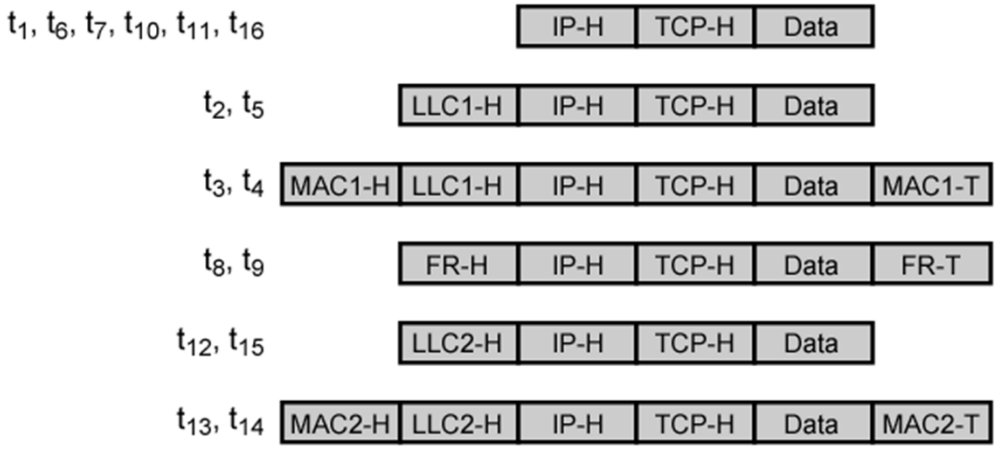
- เนื่องจากมาตรฐานของ Spanning Tree(802.1D) นั้นได้ตั้งขึ้นมาก่อน VLAN ดังนั้นการทำ VLAN ใน Network จะมีมากกว่า 1 Spanning Tree ไม่ได้ นั่นหมายถึงทุกๆ VLAN จะต้องมี Spanning Tree เดียว ซึ่งถ้าทำ VLAN แบบ ง่ายๆจะไม่มีปัญหา แต่บางครั้งถ้าเรามีการทำ Filter ของ Trunk Port อาจจะทำให้บาง VLAN หลุดจาก Spanning Tree ได้
- Cisco ได้เพิ่มส่วนของ Protocol ของ Spanning Tree ที่ทำให้สามารถมี Spanning Tree แยกสำหรับแต่ละ VLAN ได้ แต่ก็ใช้ได้กับ Switch ของ Cisco เท่านั้น อย่างไรก็ตามมาตรฐานใหม่ของ IEEE คือ IEEE 802.1s ซึ่งเป็นมาตรฐานสำหรับ Multiple Spanning Tree(MST) จะยอมให้มีหลาย Spanning Tree ได้

WAN Technologies

- **ในการเชื่อมต่อระยะไกล, Ethernet Technologies ไม่สามารถนำมาใช้ได้**
- **IP เป็น WAN แต่อยู่ใน Layer 3 ดังนั้นต้องการ Layer 2 และ Layer 1 เป็นตัวนำ IP Packet**
  - IP บน Ethernet ใช้ได้ใน LAN เท่านั้น
  - ในการส่งไกลกว่านั้น ต้องหา WAN Technologies มานำ IP Packet
    - IP บรรจุใน WAN Layer 2 ส่งผ่าน Layer 1 (HDLC, FR, SDH, MPLS, ATM ผ่าน Modem, Fiber, ...)
    - IP บรรจุใน Layer 3 WAN Frame เช่นใน X.25

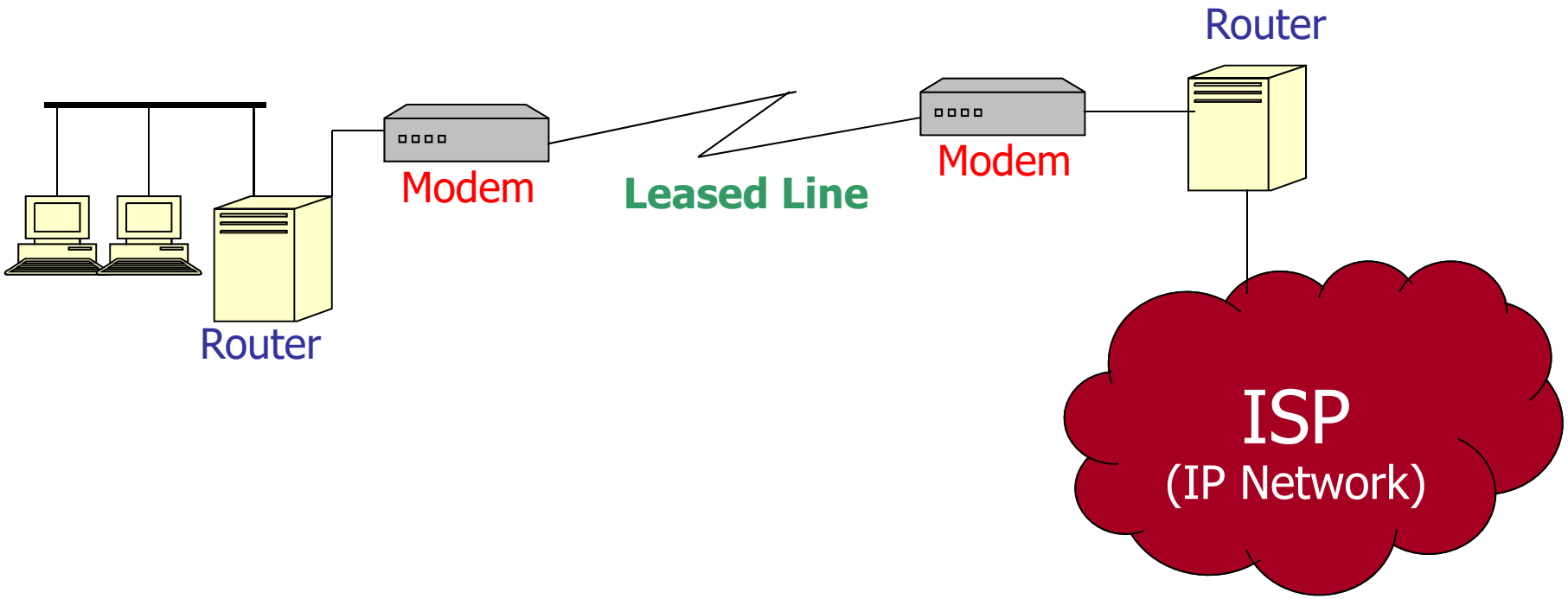
WAN Connection





TCP-H = TCP header
IP-H = IP header
LLCi-H = LLC header
MACi-H = MAC header
MACi-T = MAC trailer
FR-H = Frame relay header
FR-T = Frame relay trailer

Connect to ISP



Note: ปัจจุบัน Technology ของ Ethernet สามารถส่งได้ไกลขึ้น
ทำให้เราขยาย LAN ได้ในระยะทางหลายสิบ กม. แต่เราไม่สามารถเดินสายได้เอง
ยังคงต้องพึ่ง Public Network

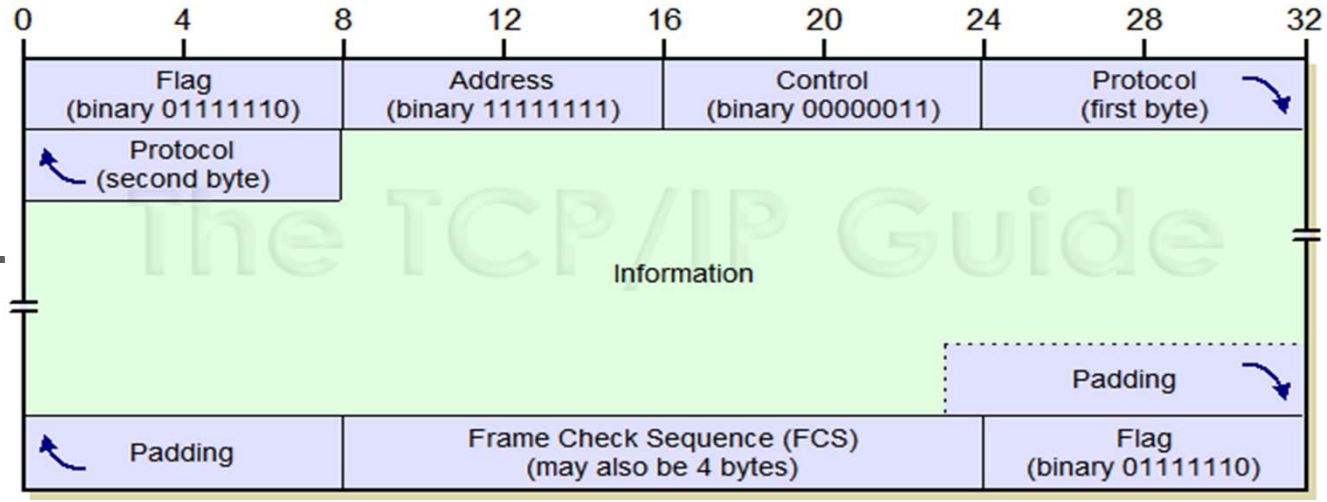
WAN Technologies

Option:	Description	Advantages	Disadvantages	Bandwidth range	Sample protocols used
Leased line	Point-to-Point connection between two computers or Local Area Networks (LANs)	Most secure	Expensive		PPP, HDLC, SDLC, HNAS
Circuit switching	A dedicated circuit path is created between end points. Best example is dialup connections	Less Expensive	Call Setup	28 kbit/s - 144 kbit/s	PPP, ISDN
Packet switching	Devices transport packets via a shared single point-to-point or point-to-multipoint link across a carrier internetwork. Variable length packets are transmitted over Permanent Virtual Circuits (PVC) or Switched Virtual Circuits (SVC)		Shared media across link		X.25 Frame-Relay
Cell relay	Similar to packet switching, but uses fixed length cells instead of variable length packets. Data is divided into fixed-length cells and then transported across virtual circuits	best for simultaneous use of Voice and data	Overhead can be considerable		ATM

PPP (Point-to-Point Protocol)

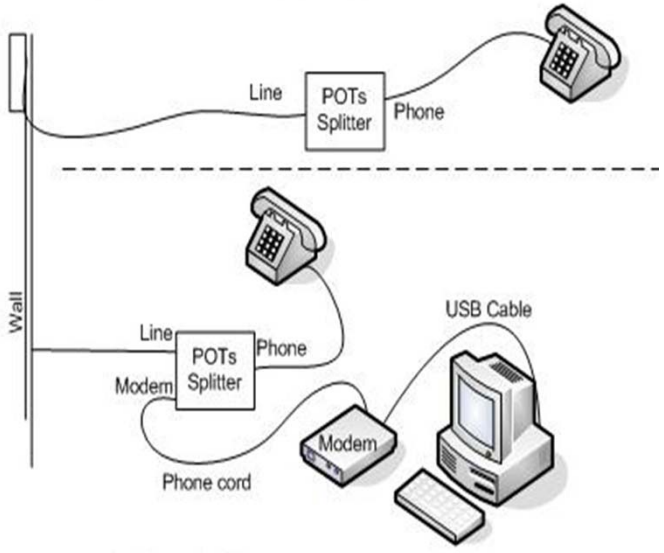
- นิยมใช้ในปัจจุบัน สำหรับเป็น Data Link Protocol ในการเชื่อมต่อโดยตรงระหว่าง Node (Point-to-Point)
- ใช้ได้ผ่าน Physical Link หลายแบบ เช่น Serial Cable, Phone Line, Cell Phone, SONET โดยที่ ISP ส่วนใหญ่จะใช้สำหรับลูกค้าที่จะ Dial-Up Access กับ Internet
- มาแทนที่ Protocol เก่าได้แก่
  - SLIP (Serial Line Internet Protocol)
  - LAPB ใน X.25
- ถูกออกแบบมาให้ใช้กับ Network Layer ต่างๆ รวมถึง IP
- ยังถูกใช้เป็น Protocol ในการเชื่อมต่อ Broadband ด้วย ใน PPPoE และ PPPoA

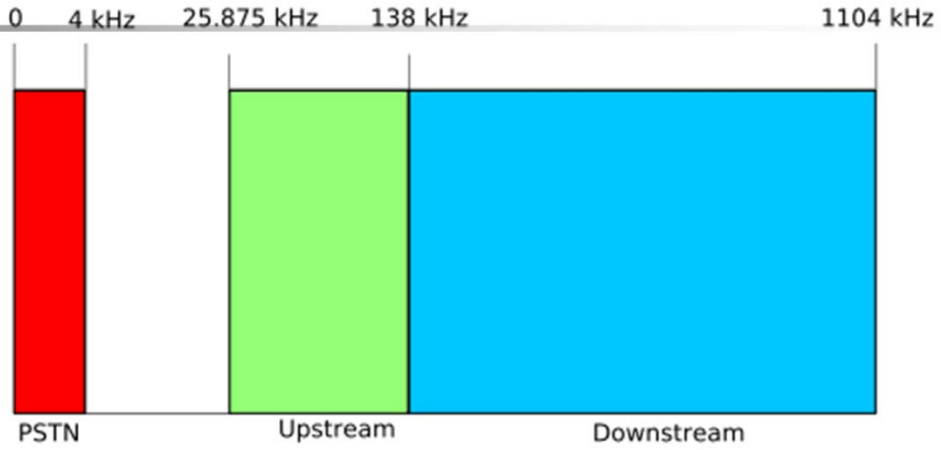
PPP Frame

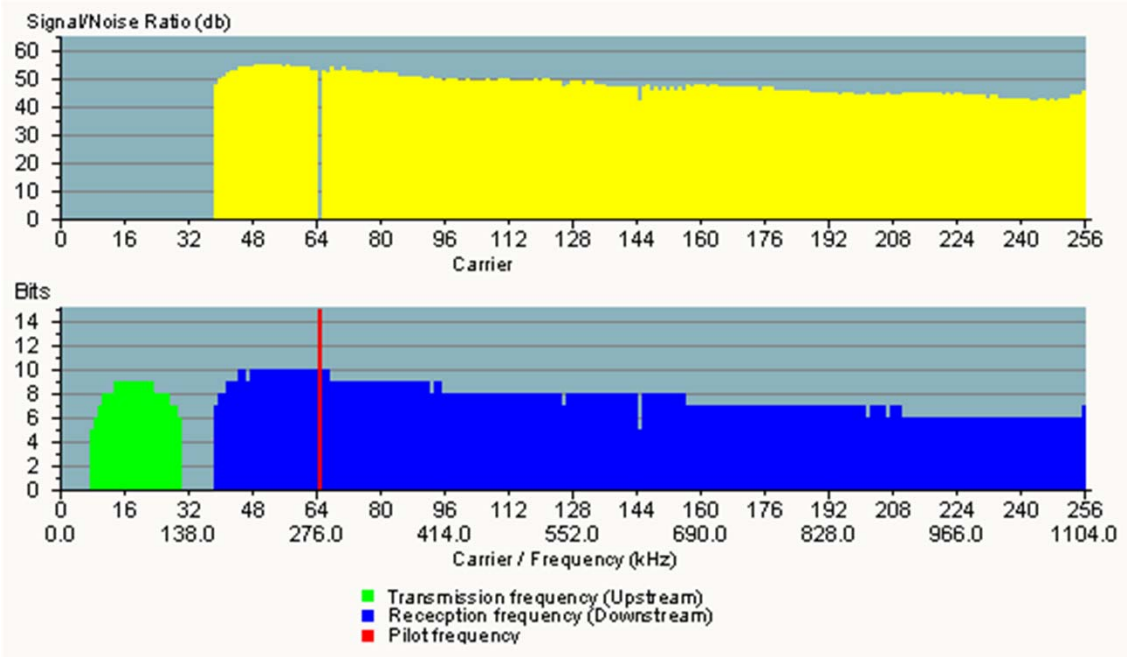


Field Name	Size (bytes)	Description
Flag	**1**	*Flag:* **Indicates the start of a PPP frame. Always has the value "01111110" binary (0x7E hexadecimal, or 126 decimal).**
Address	**1**	*Address:* **In HDLC this is the address of the destination of the frame. But in PPP we are dealing with a direct link between two devices, so this field has no real meaning. It is thus always set to "11111111" (0xFF or 255 decimal), which is equivalent to a broadcast (it means "all stations").**
Control	**1**	*Control:* **This field is used in HDLC for various control purposes, but in PPP it is set to "00000011" (3 decimal).**
Protocol	**2**	*Protocol:* **Identifies the protocol of the datagram encapsulated in the Information field of the frame. See below for more information on the *Protocol* field.**
Information	**Variable**	*Information:* **Zero or more bytes of payload that contains either data or control information, depending on the frame type. For regular PPP data frames the network-layer datagram is encapsulated here. For control frames, the control information fields are placed here instead.**
Padding	**Variable**	*Padding:* **In some cases, additional dummy bytes may be added to pad out the size of the PPP frame.**
FCS	**2 (or 4)**	*Frame Check Sequence (FCS):* **A checksum computed over the frame to provide basic protection against errors in transmission. This is a CRC code similar to the one used for other layer two protocol error protection schemes such as the one used in Ethernet. It can be either 16 bits or 32 bits in size (default is 16 bits).** **The FCS is calculated over the *Address*, *Control*, *Protocol*, *Information* and *Padding* fields.**
Flag	**1**	*Flag:* **Indicates the end of a PPP frame. Always has the value "01111110" binary (0x7E hexadecimal, or 126 decimal).**

Broad-band (ADSL)

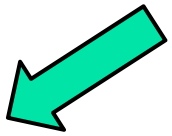




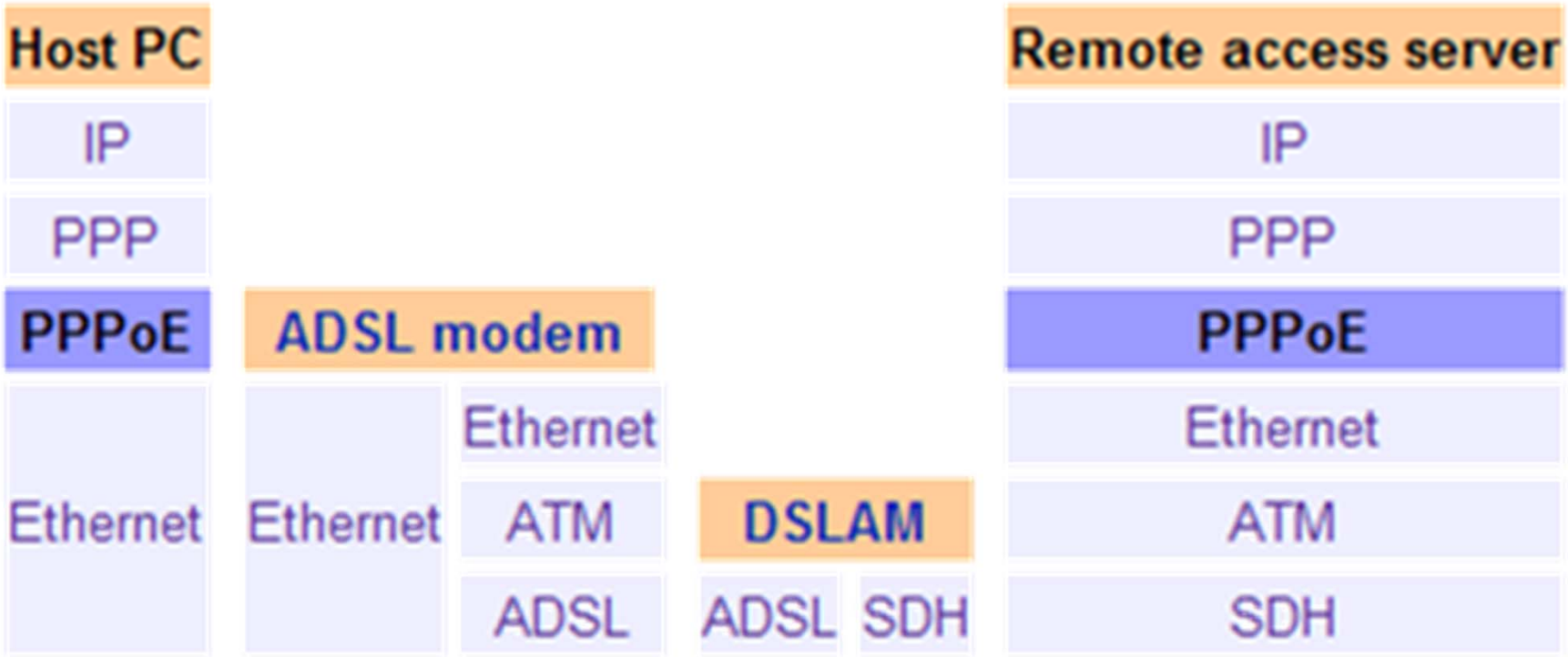


PPPoE

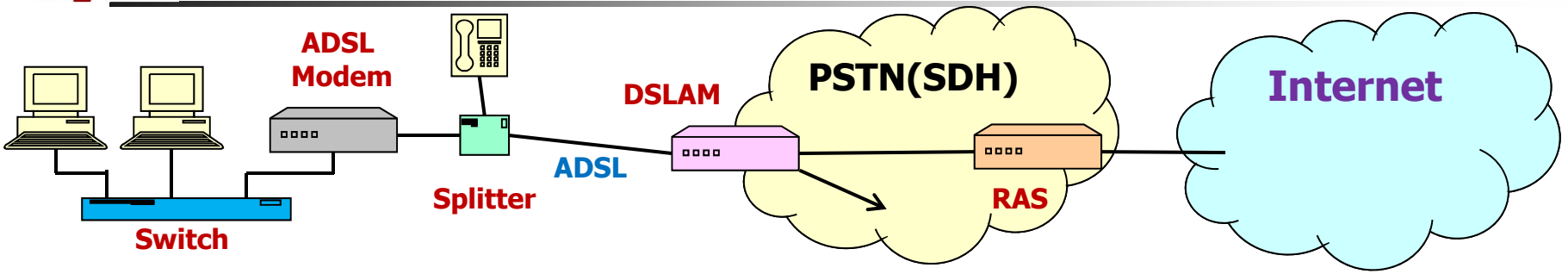
Application	FTP	SMTP	HTTP	…	DNS	…
Transport	TCP				UDP	
Network	IP				IPv6	
Network access	PPP					
	PPPoE					
	Ethernet					



ADSL internet access architecture



PPPoE



Host PC				Remote access server
IP				IP
PPP				PPP
PPPoE	ADSL modem			PPPoE
Ethernet	Ethernet	Ethernet		Ethernet
		ATM	DSLAM	ATM
		ADSL	ADSL SDH	SDH

Importance Ethernet Standards

Ethernet Standard	Date	Description
Experimental Ethernet	1973[1]	2.94 Mbit/s (367 kB/s) over coaxial cable (coax) bus
Ethernet II (DIX v2.0)	1982	10 Mbit/s (1.25 MB/s) over thick coax. Frames have a Type field. This frame format is used on all forms of Ethernet by protocols in the Internet protocol suite.
IEEE 802.3 standard	1983	10BASE5 10 Mbit/s (1.25 MB/s) over thick coax. Same as Ethernet II (above) except Type field is replaced by Length, and an 802.2 LLC header follows the 802.3 header. Based on the CSMA/CD Process.
802.3a	1985	10BASE2 10 Mbit/s (1.25 MB/s) over thin Coax (a.k.a. thinnet or cheapernet)
802.3i	1990	10BASE-T 10 Mbit/s (1.25 MB/s) over twisted pair
802.3j	1993	10BASE-F 10 Mbit/s (1.25 MB/s) over Fiber-Optic

Importance Ethernet Standards

Ethernet Standard	Date	Description
802.3u	1995	100BASE-TX, 100BASE-T4, 100BASE-FX Fast Ethernet at 100 Mbit/s (12.5 MB/s) w/autonegotiation
802.3x	1997	Full Duplex and flow control; also incorporates DIX framing, so there's no longer a DIX/802.3 split
802.3ab	1999	1000BASE-T Gbit/s Ethernet over twisted pair at 1 Gbit/s (125 MB/s)
802.3ad	2000	Link aggregation for parallel links, since moved to IEEE 802.1AX
802.3ae	2002	10 Gigabit Ethernet over fiber; 10GBASE-SR, 10GBASE-LR, 10GBASE-ER, 10GBASE-SW, 10GBASE-LW, 10GBASE-EW
802.3af	2003	Power over Ethernet (15.4 W)
802.3an	2006	10GBASE-T 10 Gbit/s (1,250 MB/s) Ethernet over unshielded twisted pair (UTP)
802.3at	2009	Power over Ethernet enhancements (25.5 W)

Ethernet Standard	Date	Description
802.3ba	2010	40 Gbit/s and 100 Gbit/s Ethernet. 40 Gbit/s over 1m backplane, 10 m Cu cable assembly (4x25 Gbit or 10x10 Gbit lanes) and 100 m of MMF and 100 Gbit/s up to 10 m of Cu cable assembly, 100 m of MMF or 40 km of SMF respectively
802.3.1	2011	MIB definitions for Ethernet. It consolidates the Ethernet related MIBs present in Annex 30A&B, various IETF RFCs, and 802.1AB annex F into one master document with a machine readable extract. (workgroup name was P802.3be)
802.3bm	2015	100G/40G Ethernet for optical fiber
802.3bq	~Feb 2016	40GBASE-T for 4-pair balanced twisted-pair cabling with 2 connectors over 30 m distances
802.3bs	~ 2017	400 Gbit/s Ethernet over optical fiber using multiple 25G/50G lanes
802.3by	~Sep 2016	25G Ethernet
802.3bz	TBD	2.5 Gigabit and 5 Gigabit Ethernet over twisted pair - 2.5GBASE-T and 5GBASE-T

HW1 Due Next Week

ให้นักศึกษา Download HW 1, Week 2
ทำการพิมพ์คำถามบนกระดาษ A4 จากนั้นให้ทำการบ้าน
ลงในกระดาษที่พิมพ์ ด้วยการเขียนเท่านั้น(ห้ามพิมพ์)
และส่งต้นชั่วโมง สัปดาห์ถัดไป

อย่าลืมใส่ชื่อ รหัส และ Section ในหน้าแรก

End of Review Part I+II

- **Next Week**
  - Internet Concept
  - IP Address

CPE 426 Computer Networks

Chapter 3:

Review 3: SPT & VLAN

Textbook Chapter 20: Internet Concept

Textbook Chapter 21: IP Address

TOPICS

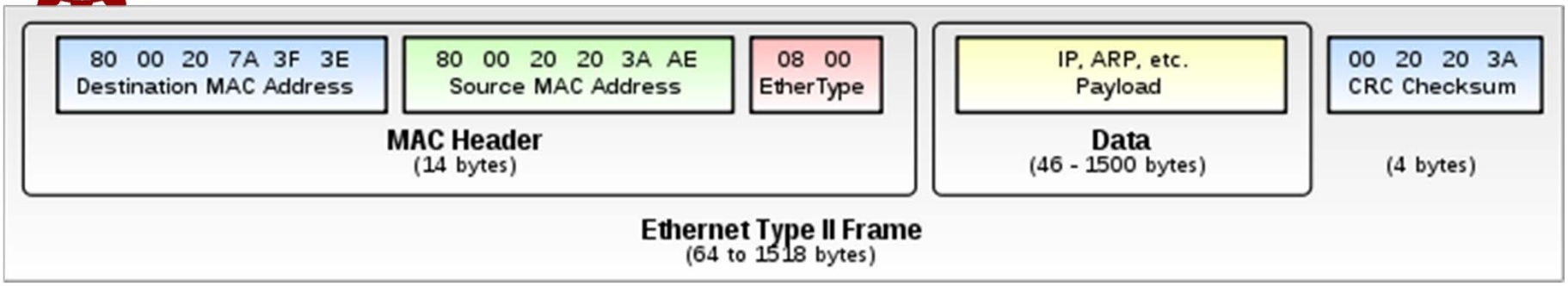
- 1. Repeater/Bridges
  - Chapter 17: 17.1-17.6
- 2. SPT
  - Chapter 17: 17.7-17.8
- **3. VLAN**
  - Chapter 17: 17.9-17.11
- **4. X.25/FR/ATM/MPLS/ISDN**
  - Chapter 19: 19.1-19.4

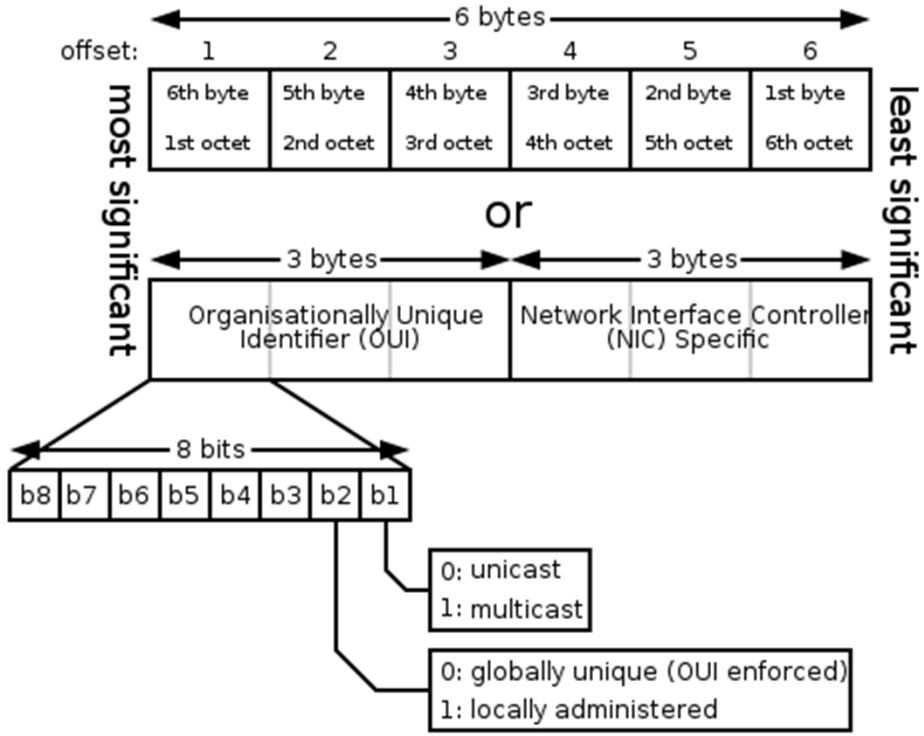
ALSO Reference From CPE 326 (Stalling Book)

TOPICS

- **BREAK**
- **Chapter 20: Internetworking**
  - Motivation: Sec. 20.1-20.2
  - Concept: Sec. 20.3-20.4
  - Internetworking: Sec. 20.5-20.8
  - Protocol Architecture: Sec. 20.9-20.11
  - Routers: Sec. 20.12
- **Chapter 21: IP Address (Not Finish)**
  - Addressing Scheme/Prefix&Suffix: 21.1-21.8
  - Subnetting and Mask: 21.9-21.13

Ether Type II (DIX Frame)



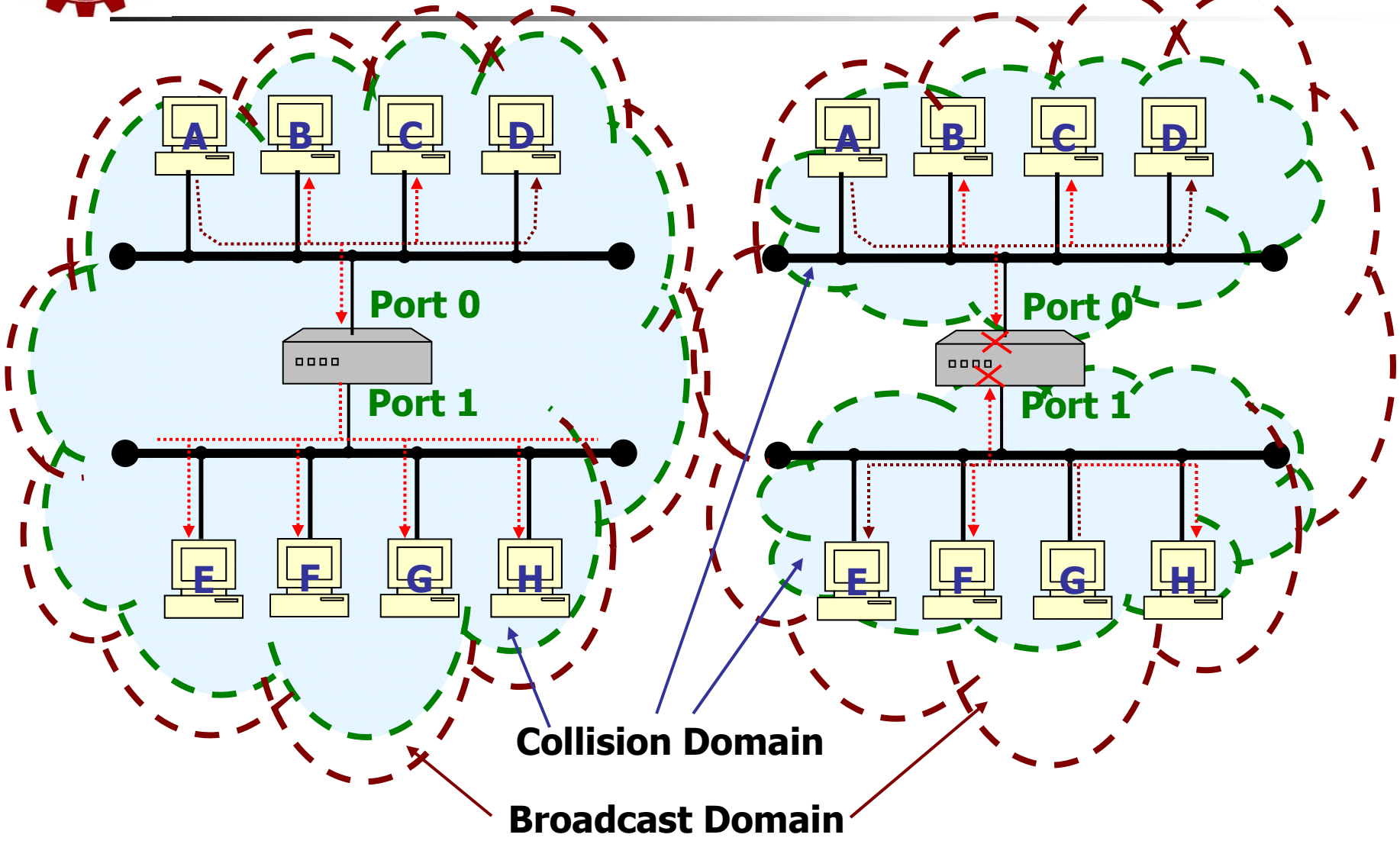


MAC-48 Address
In Transmission Order
01-23-45-67-89-ab,
01:23:45:67:89:ab,
0123.4567.89ab

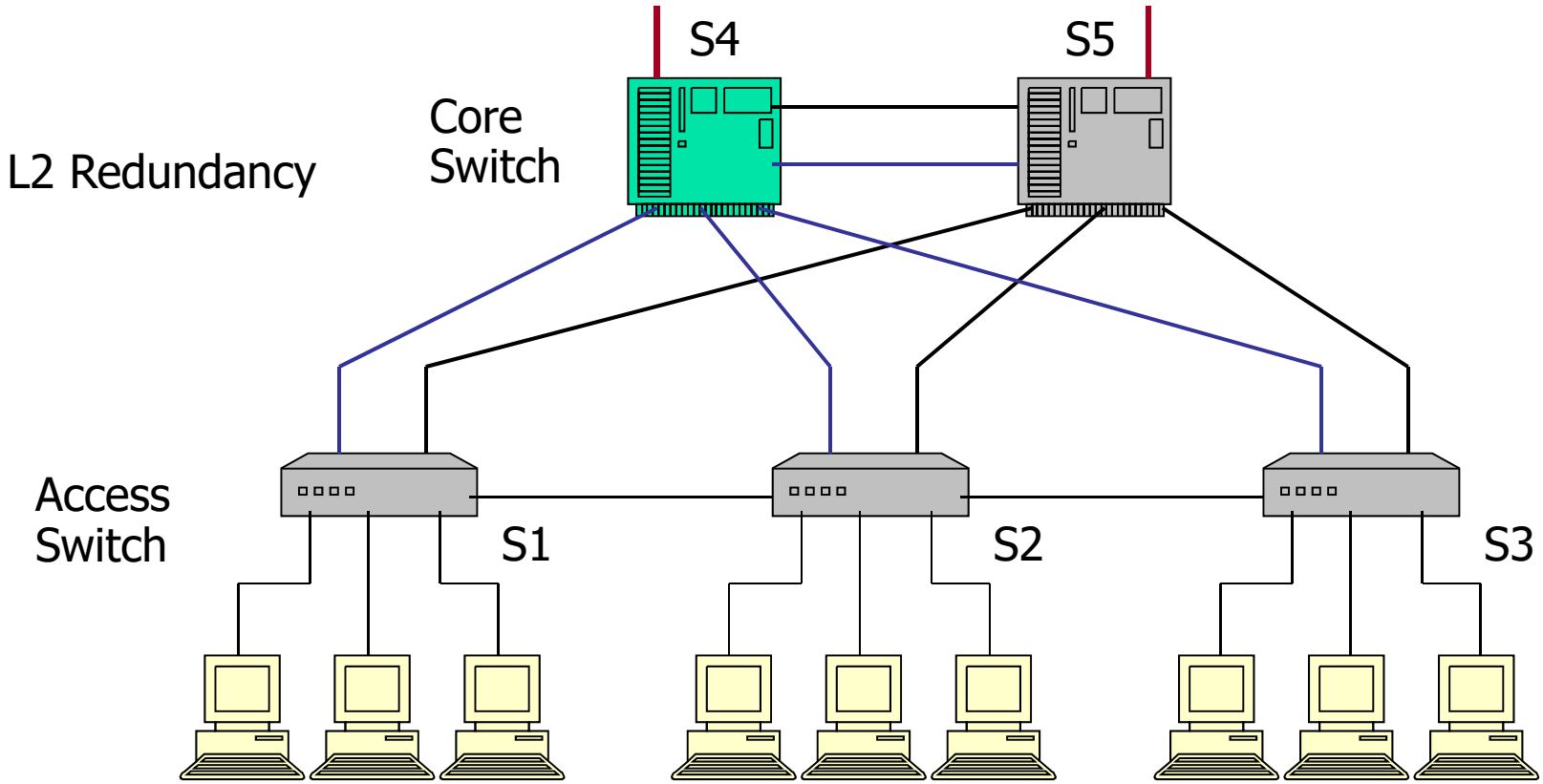
802.3/.4 Send LSBit First
(Canonical Format)
10000000 11000100 10100010 …

802.5/.6 Send MSBit First
(Bit-Reverse/Non-canonical)
00000001 00100011 01000101 …

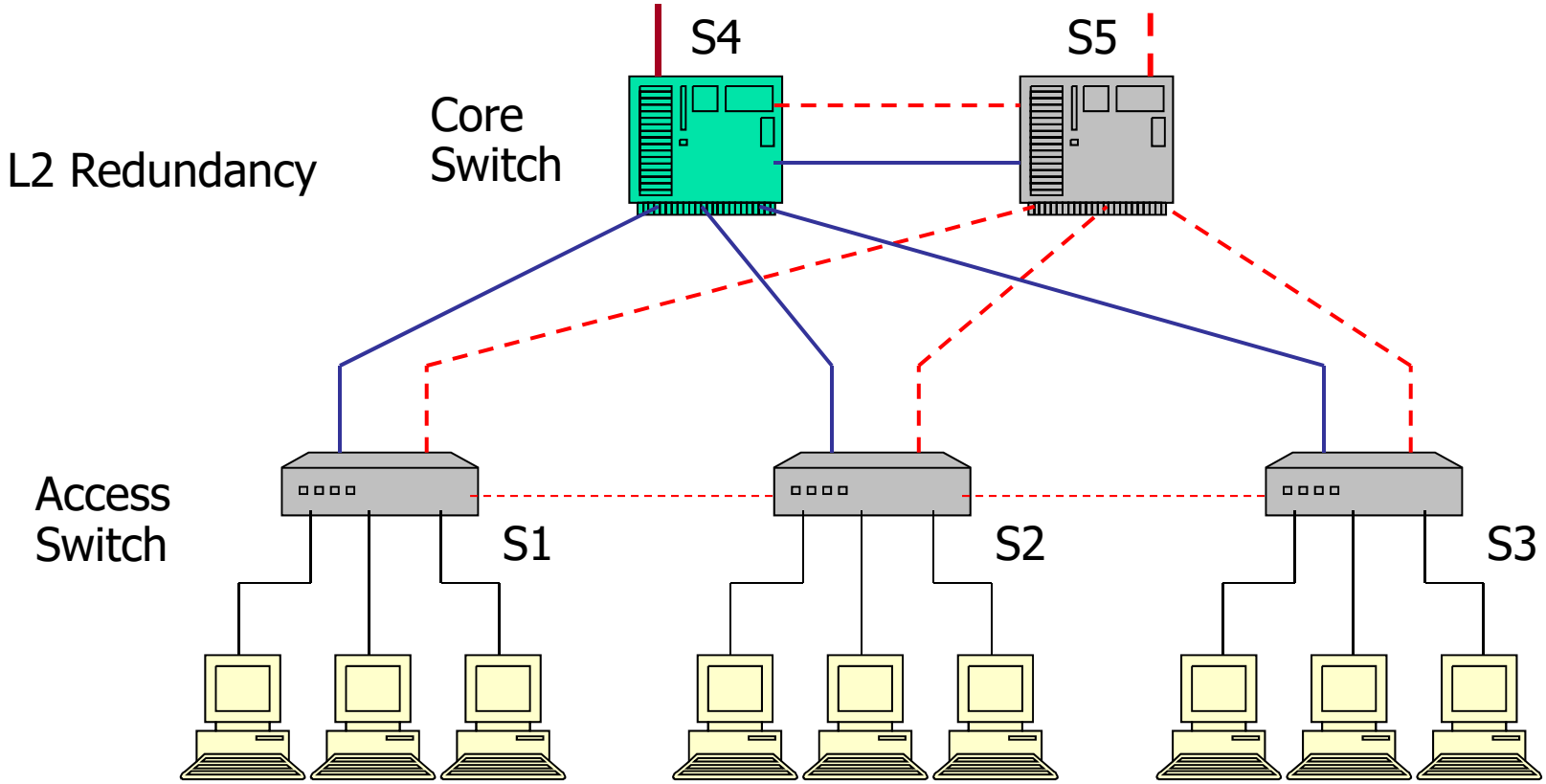
Bridge vs Repeater: Collision Domain vs Broadcast Domain



Layer 2 Redundancy: Active Link



Layer 2 Redundancy: SPT



Steps 1: Root Bridge Selection

- เลือก Root Bridge โดยทุก Switch ส่ง BPDU ออกทุก Port และใส่ค่า Bridge ID
- Bridge ID = Bridge Priority(2 Octet) + MAC Address(6 Octet)
- Switch ที่มี Bridge ID ต่ำสุดจะเป็น Root
- Default Bridge Priority = 32768
- ถ้าไม่มีการ Configure ดังนั้น Switch ที่มี MAC Address ต่ำสุดจะได้รับเลือก

Steps 2: Minimum Cost Tree

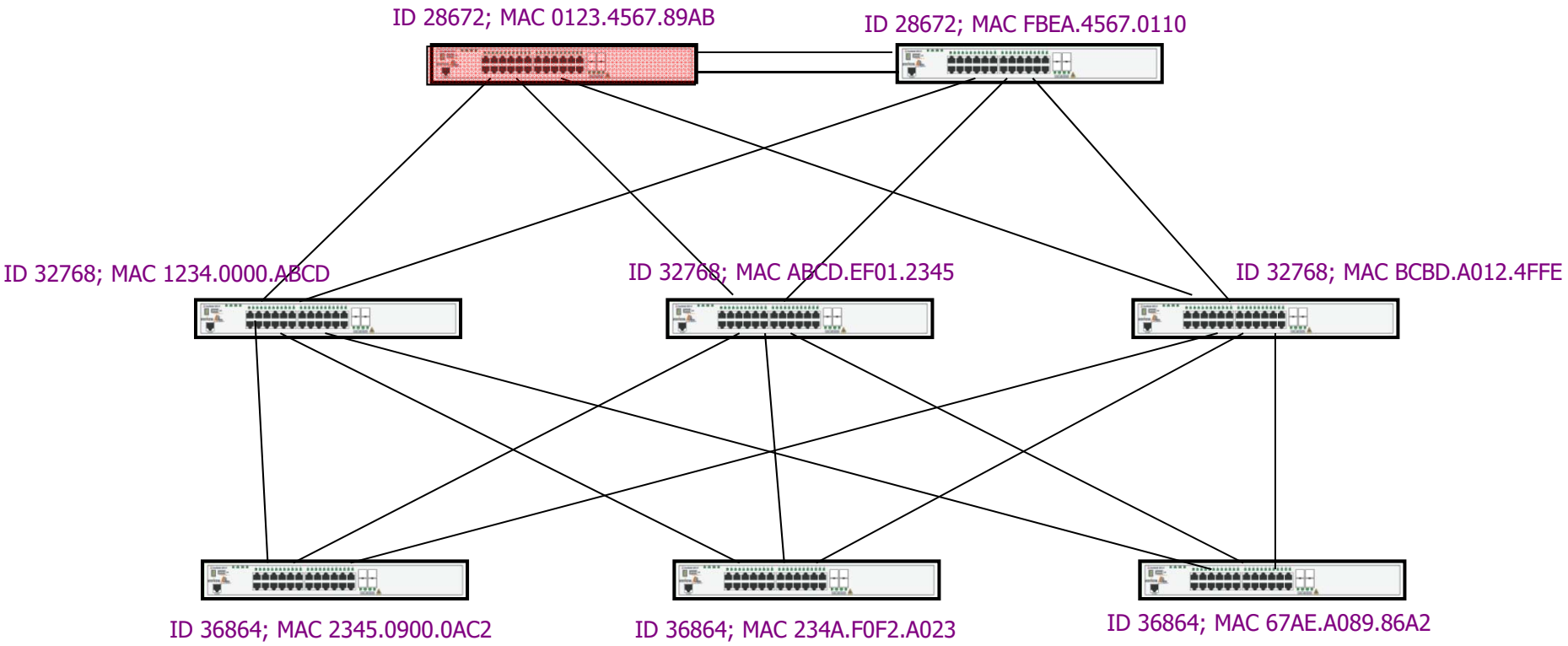
- สร้าง Minimum Cost Tree โดยจาก Root ส่ง BPDU ที่มี Cost = 0 ออกทุกๆ Port ที่มันต่อ ซึ่งถูกจัดว่าเป็น Designated Port
- เมื่อ Switch ได้รับ BPDU มันจะบวกค่า Cost กับ Cost ของ Link ที่เข้ามา และส่งต่อ
- ถ้ามันได้รับมากกว่า 1 BPDU แสดงว่ามีมากกว่าหนึ่งเส้นทางไปยัง Root (Loop)
- เลือกเส้นทางที่ Cost ต่ำกว่า เป็น Root Port
- ถ้ามีมากกว่าหนึ่งเส้นทางและ Cost เท่ากัน เลือก Port ไปยัง Bridge ID ต่ำกว่า
- ถ้ายังเท่ากันเลือก Port Priority ต่ำกว่า

Steps 3: เลือก Designated Port และ Port Blocking

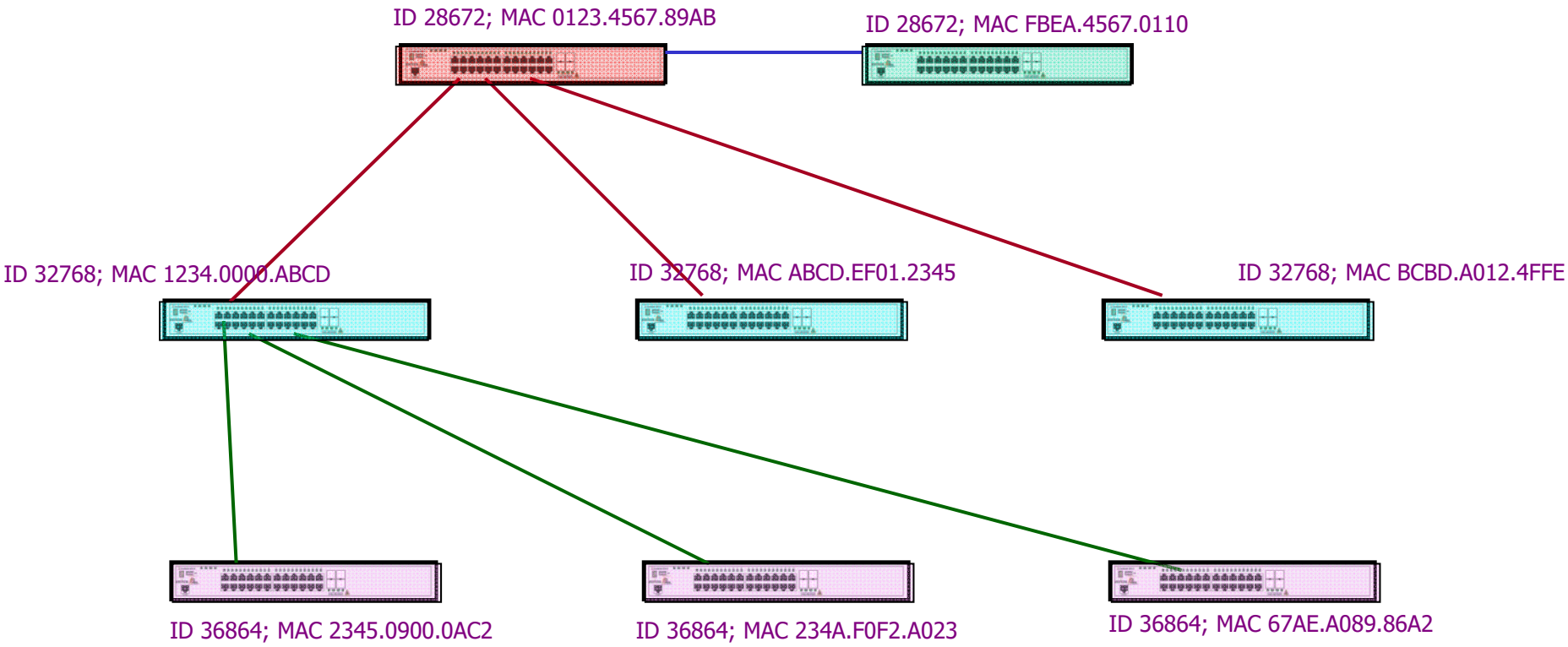
- **เส้นทางที่ไม่ได้ถูกเลือกจะถูกปิด**
- **การปิด ทำโดย Blocking Port**
- **Port จะถูกปิดด้านเดียว**
  - ปิด Port ที่มี Cost สูงกว่าไปยัง Root ถ้าเท่ากัน
  - ปิด Port Switch ที่มี Bridge ID สูงกว่า ถ้าเท่ากัน
  - ปิด Port ที่มี Port ID สูงกว่า
    - Port ID = Port Priority(1 Byte, Default = 128) + Port Number
- **Port ที่เปิดเรียก Designated Port**

Order of Precedence

เลือก Root Bridge



Order of Precedence



VLAN

- **แยก Broadcast Domain ออกภายใน Switch ตัวเดียว**
- **L2 Protocol**
- **เหมือนกับมีหลาย Switch ที่ไม่เชื่อมต่อกันในตัวเดียว**
- **สามารถทำการ Configure ได้ว่าจะแยกอย่างไร**
  - VLAN by Port (Static) กำหนดแต่ละ Port ตายตัวว่าเป็นของ VLAN อะไร
  - Dynamic VLAN : ตาม MAC, IP, Protocol หรืออื่นๆ กรณีนี้แต่ละ Port จะเปลี่ยน VLAN ตาม Condition ที่กำหนด เราเรียกว่าเป็น Mobile Port

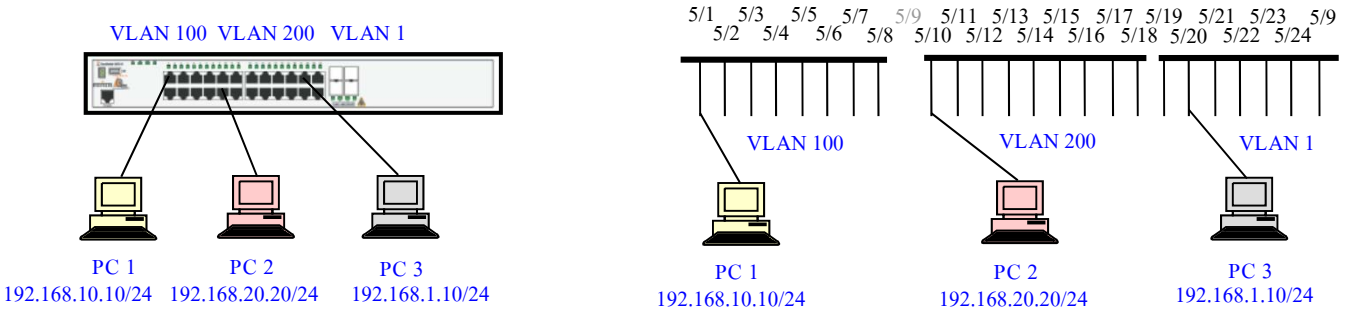
VLAN

- VLAN 1 คือ Default VLAN ลบและสร้างไม่ได้
- ทุก Port ถ้าไม่มีการกำหนดจะอยู่ใน VLAN 1
- VLAN Number = 12 Bit แต่ปกติการสร้าง จะให้หมายเลขระหว่าง VLAN 2 – VLAN 4094
- การเชื่อมต่อสอง VLAN ด้วยกันต้องใช้ความสามารถของ L3
- VLAN สามารถแยก Physical NW ออกจาก Logical NW

VLAN แบ่ง Switch เป็นหลายส่วน

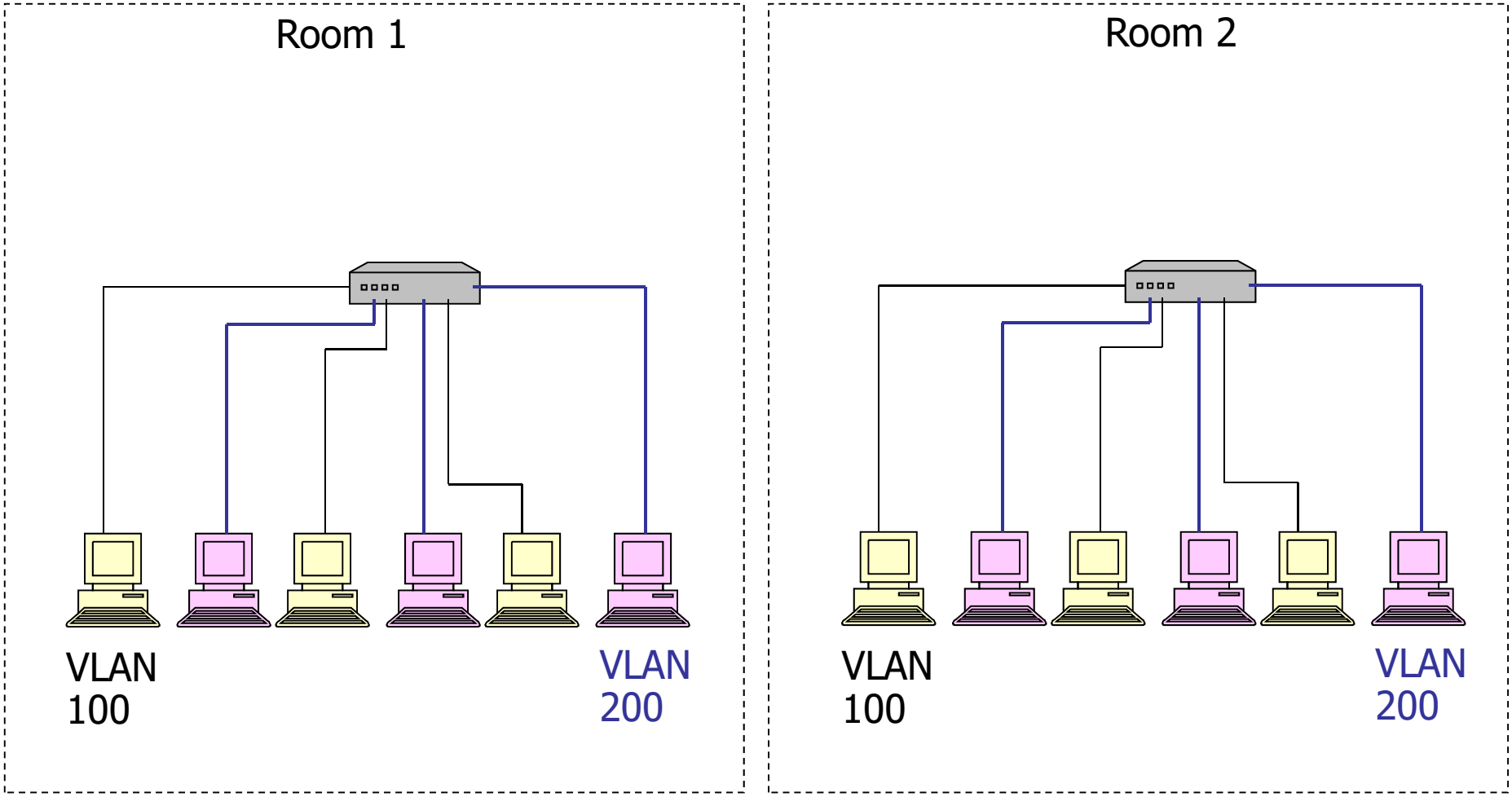
Switch ปกติเมื่อไม่แบ่ง VLAN หรือไม่ใช่ Managed Switch

Switch ตัวเดียว ถูกแบ่งเป็น 3 VLAN

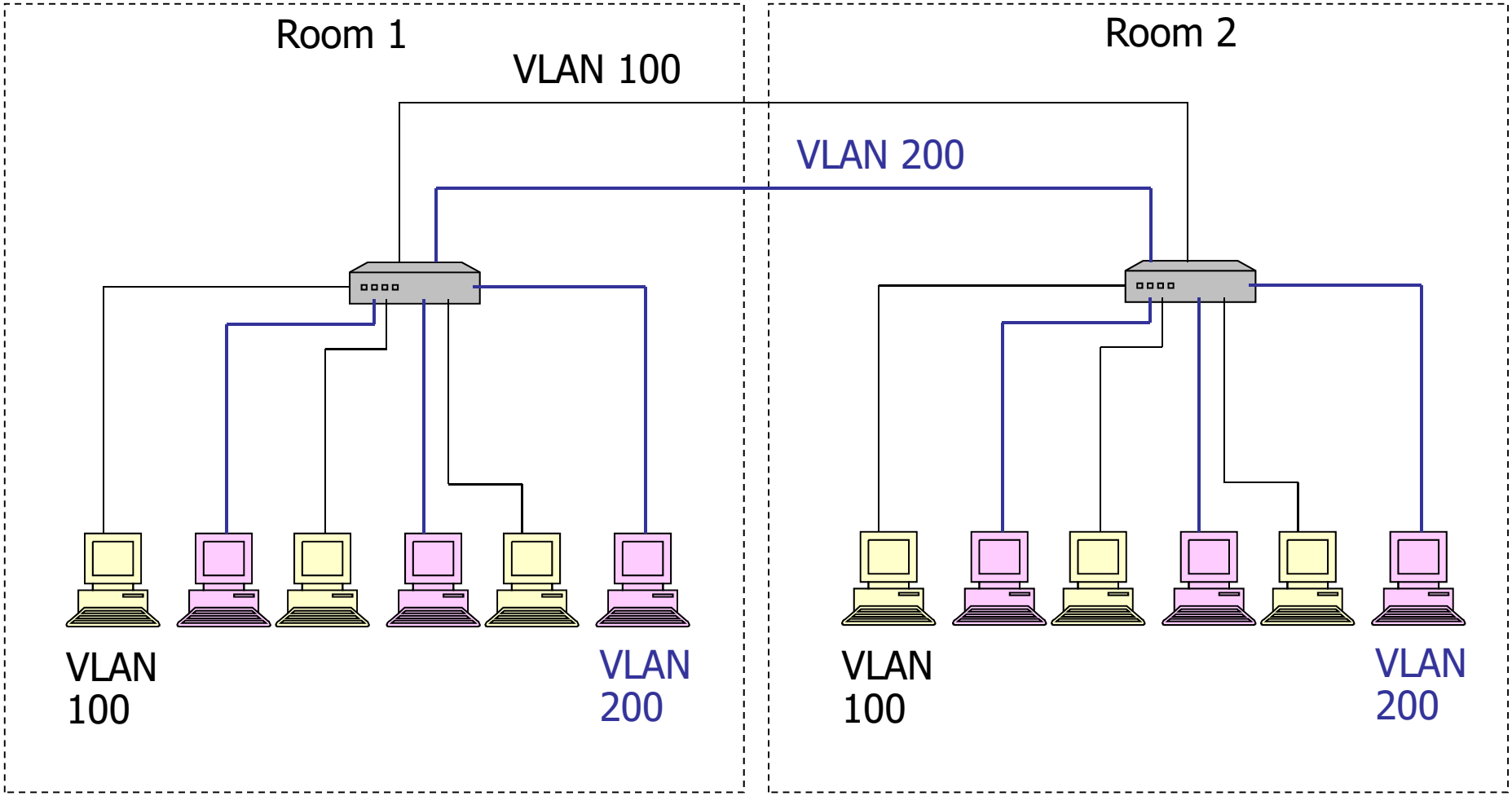


แต่ละ VLAN ถูกแยกออกจากกัน เสมือนอยู่คนละ Switch
จัดว่าอยู่คนละ Sub-network/Broadcast Domain
ต้องใช้อุปกรณ์ Layer 3(Router) มาเชื่อมต่อ

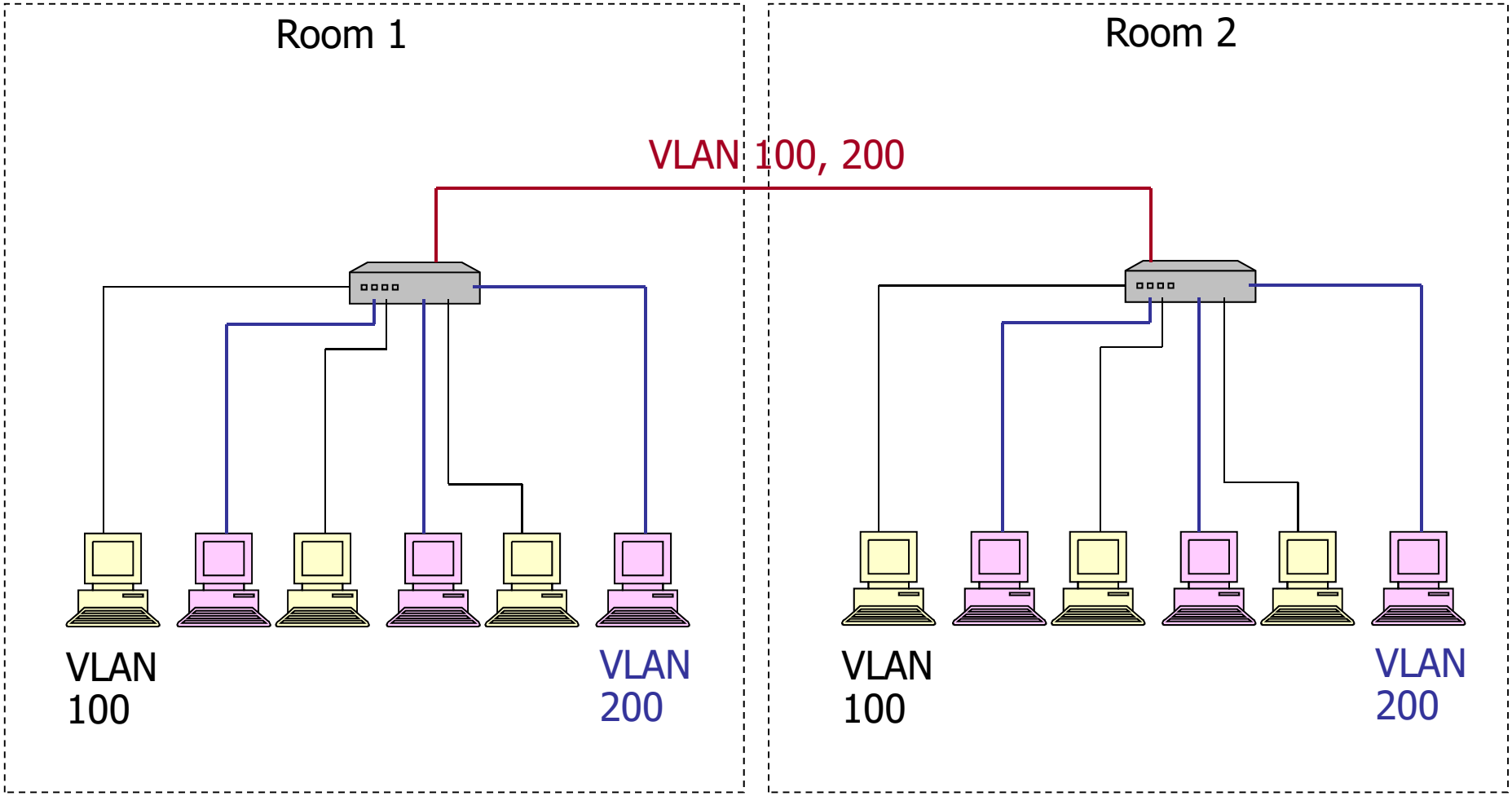
VLAN สามารถขยายผ่านมากกว่า 1 Switch



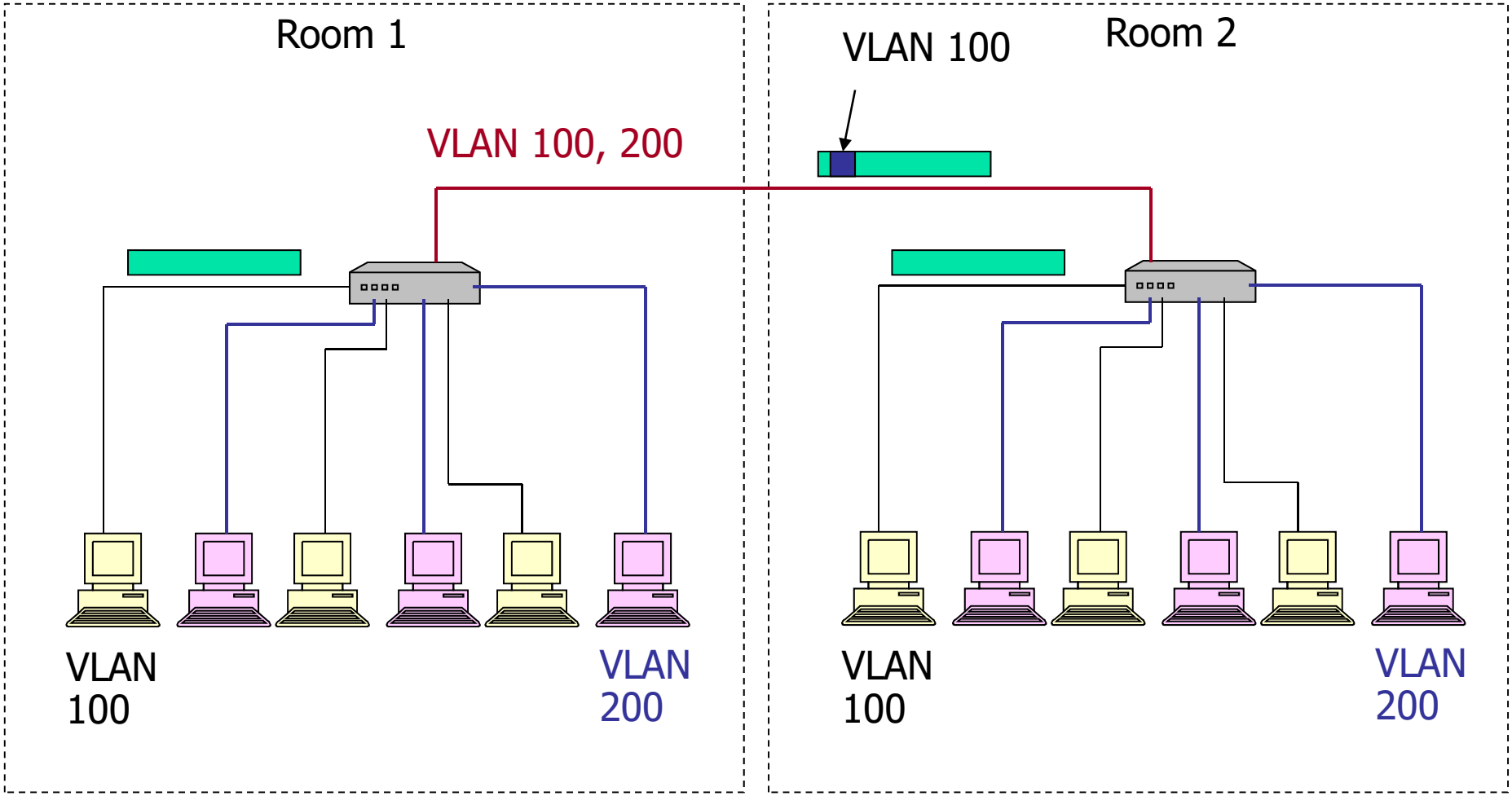
VLAN สามารถขยายผ่านมากกว่า 1 Switch



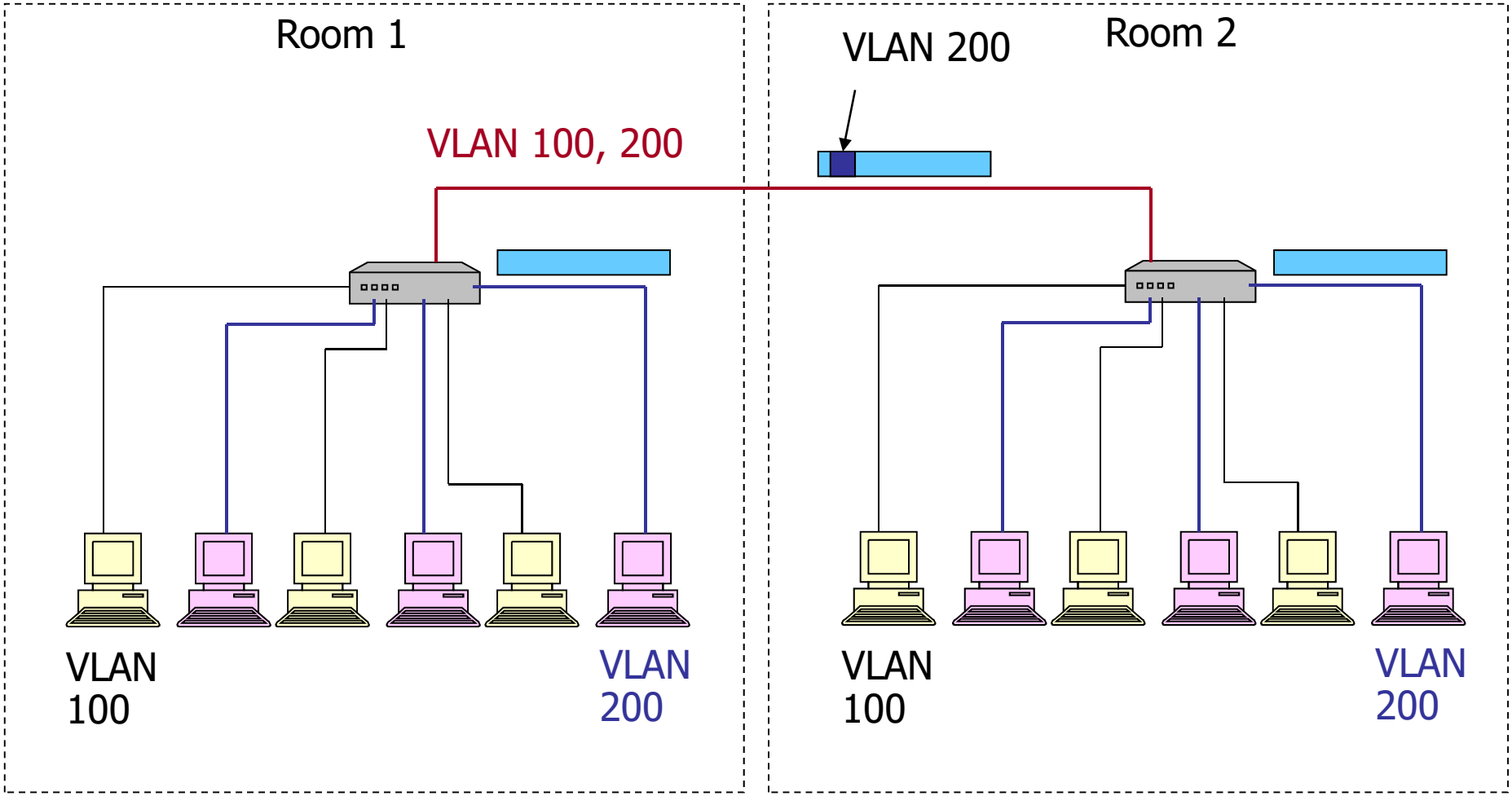
VLAN สามารถขยายผ่านมากกว่า 1 Switch



VLAN สามารถขยายผ่านมากกว่า 1 Switch



VLAN สามารถขยายผ่านมากกว่า 1 Switch



VLAN TAGGING

- **IEEE 802.1Q Standard**
  - 4 Byte เพิ่มในส่วนของ Header
  - 12 Bit เป็น VLAN Number
- **ISL(Cisco)**
  - Encapsulation

VLAN Tagging (IEEE 802.1Q)

- **Port ของ Switch จะต้องถูกกำหนดเป็น Tag Port**
- **เมื่อข้อมูลถูกส่งออกไปยัง Tag Port จะมีการใส่ Tag กำหนด VLAN**
- **เมื่อข้อมูลมาถึง Tag Port จะถูกส่งไปยัง VLAN ตาม Tag และตัว Tag จะถูกนำออก**
- **VLAN Default ของ Port นั้นจะไม่ถูกใส่ Tag**
  - VLAN Number จะเป็น Local ยกเว้นทำ Tagging
  - อุปกรณ์บางยี่ห้อจะมี Protocol สื่อสารระหว่าง SW (Interswitch Protocol)

Communication Between VLAN

- Connect Through Router (L3)
- Using L3 Switch ดีกว่า

VLAN Static vs Dynamic

- **เมื่อ VLAN ถูกกำหนดโดย Port ของ Switch เราเรียก Static VLAN**
  - อุปกรณ์ที่เชื่อมต่อกับ Port ดังกล่าวจะถูกจับไปอยู่ใน VLAN ที่กำหนด
- **แต่ถ้าเรากำหนดให้อุปกรณ์ที่มาเชื่อมกับ Port ไปอยู่ใน VLAN ตามคุณสมบัติของอุปกรณ์ เช่น ตาม IP Address, MAC Address หรือตามการ Authentication เราเรียก Dynamic VLAN**
  - Port ดังกล่าวจะเป็น “Mobile Port” และต้องกำหนด VLAN Rule ให้

การกำหนด VLAN

- **หนึ่ง Subnet ให้เป็น หนึ่ง VLAN**
- **เมื่อเรากำหนด Topology เราได้**
  - Subnet ของแต่ละ Network
  - กำหนด IP Address ให้กับแต่ละ Subnet
  - กำหนด VLAN ให้กับแต่ละ Subnet
    - ดังนั้นแต่ละ Subnet สามารถอยู่ร่วมกันบน Switch เดียวกันได้
    - แต่ละ Subnet สามารถกระจาย ครอบคลุมหลาย Switch ได้
    - กล่าวคือ Logical Network(Diagram) และ Physical Network(Wiring Diagram) สามารถแยกจากกัน
      - Network จะประกอบด้วยสอง Diagram

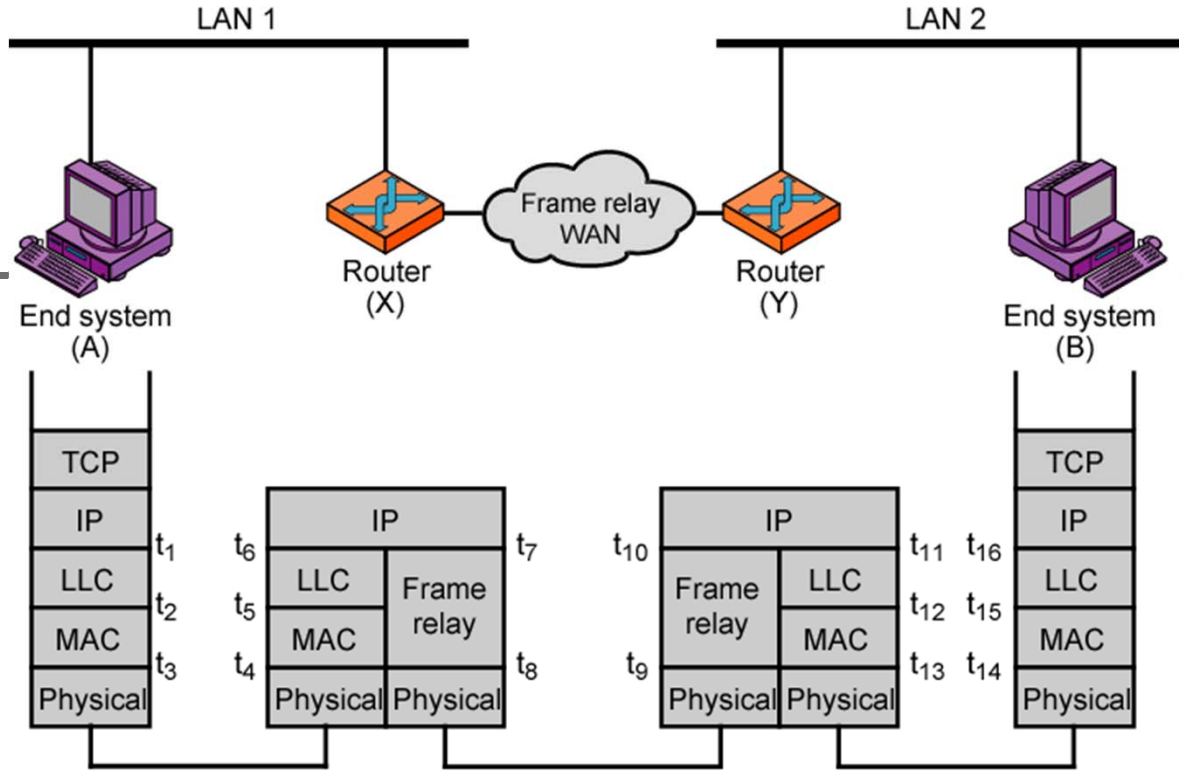
Spanning Tree and VLAN

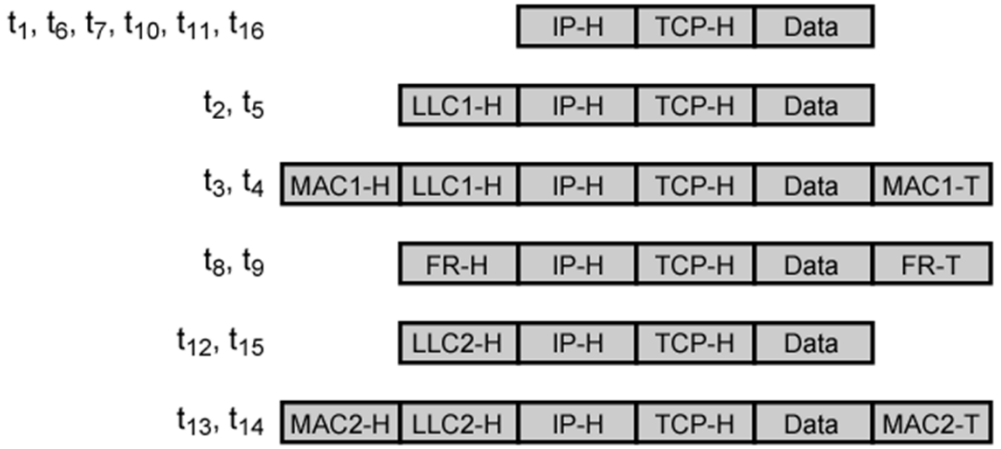
- เนื่องจากมาตรฐานของ Spanning Tree(802.1D) นั้นได้ตั้งขึ้นมาก่อน VLAN ดังนั้นการทำ VLAN ใน Network จะมีมากกว่า 1 Spanning Tree ไม่ได้ นั่นหมายถึงทุกๆ VLAN จะต้องมี Spanning Tree เดียว ซึ่งถ้าทำ VLAN แบบ ง่ายๆจะไม่มีปัญหา แต่บางครั้งถ้าเรามีการทำ Filter ของ Trunk Port อาจจะทำให้บาง VLAN หลุดจาก Spanning Tree ได้
- Cisco ได้เพิ่มส่วนของ Protocol ของ Spanning Tree ที่ทำให้สามารถมี Spanning Tree แยกสำหรับแต่ละ VLAN ได้ แต่ก็ใช้ได้กับ Switch ของ Cisco เท่านั้น อย่างไรก็ตามมาตรฐานใหม่ของ IEEE คือ IEEE 802.1s ซึ่งเป็นมาตรฐานสำหรับ Multiple Spanning Tree(MST) จะยอมให้มีหลาย Spanning Tree ได้

WAN Technologies

- **ในการเชื่อมต่อระยะไกล, Ethernet Technologies ไม่สามารถนำมาใช้ได้**
- **IP เป็น WAN แต่อยู่ใน Layer 3 ดังนั้นต้องการ Layer 2 และ Layer 1 เป็นตัวนำ IP Packet**
  - IP บน Ethernet ใช้ได้ใน LAN เท่านั้น
  - ในการส่งไกลกว่านั้น ต้องหา WAN Technologies มานำ IP Packet
    - IP บรรจุใน WAN Layer 2 ส่งผ่าน Layer 1 (HDLC, FR, SDH, MPLS, ATM ผ่าน Modem, Fiber, ...)
    - IP บรรจุใน Layer 3 WAN Frame เช่นใน X.25

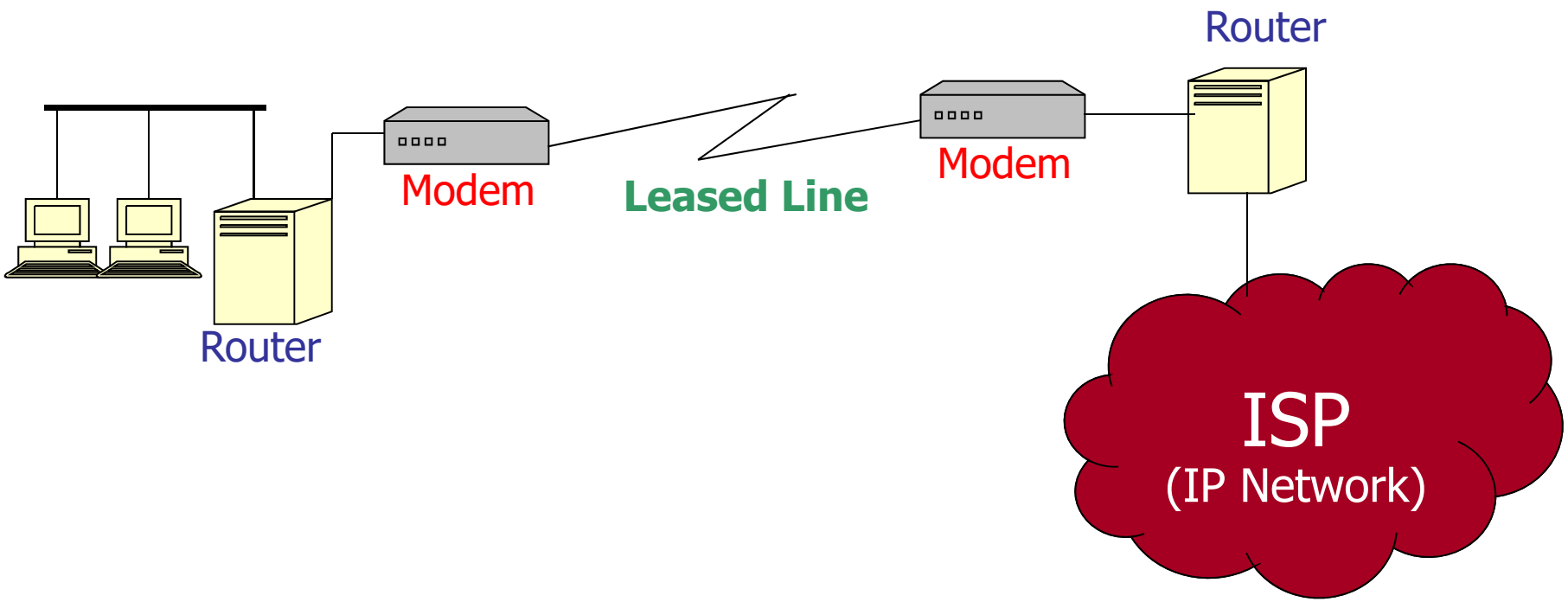
WAN Connection





TCP-H = TCP header
IP-H = IP header
LLCi-H = LLC header
MACi-H = MAC header
MACi-T = MAC trailer
FR-H = Frame relay header
FR-T = Frame relay trailer

Connect to ISP



Note: ปัจจุบัน Technology ของ Ethernet สามารถส่งได้ไกลขึ้น
ทำให้เราขยาย LAN ได้ในระยะทางหลายสิบ กม. แต่เราไม่สามารถเดินสายได้เอง
ยังคงต้องพึ่ง Public Network

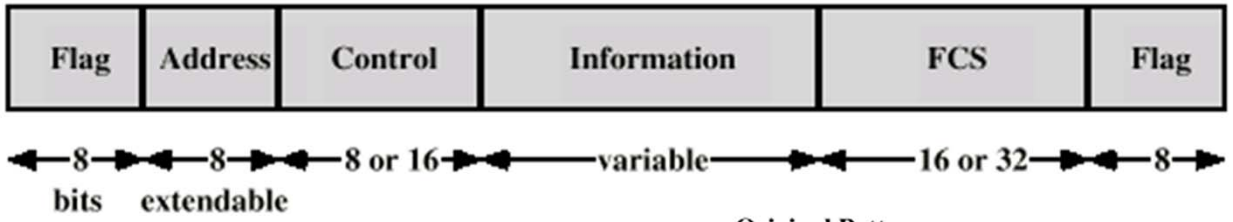
WAN Technologies

Option:	Description	Advantages	Disadvantages	Bandwidth range	Sample protocols used
Leased line	Point-to-Point connection between two computers or Local Area Networks (LANs)	Most secure	Expensive		PPP, HDLC, SDLC, HNAS
Circuit switching	A dedicated circuit path is created between end points. Best example is dialup connections	Less Expensive	Call Setup	28 kbit/s - 144 kbit/s	PPP, ISDN
Packet switching	Devices transport packets via a shared single point-to-point or point-to-multipoint link across a carrier internetwork. Variable length packets are transmitted over Permanent Virtual Circuits (PVC) or Switched Virtual Circuits (SVC)		Shared media across link		X.25 Frame-Relay
Cell relay	Similar to packet switching, but uses fixed length cells instead of variable length packets. Data is divided into fixed-length cells and then transported across virtual circuits	best for simultaneous use of Voice and data	Overhead can be considerable		ATM

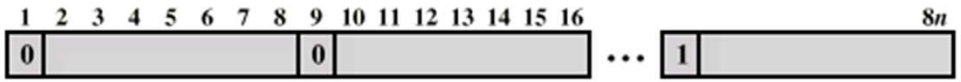
HDLC

- **High Level Data Link Control Protocol**
- **ISO Standard**
  - Current Standard = ISO 13239
- **Connection Oriented and Connectionless**
- **Most common mode = point-to-point using ABM (Asynchronous Balanced Mode)**
- **Transmission Mode/Station Type/Flow**
  - ดูใน CPE326 (Stalling Book)

HDLC Frame Format



(a) Frame format



(b) Extended Address Field

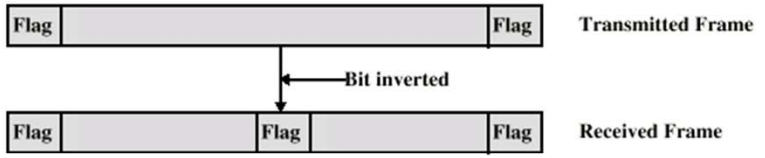
Original Pattern:

```
11111111111101111101111110
```

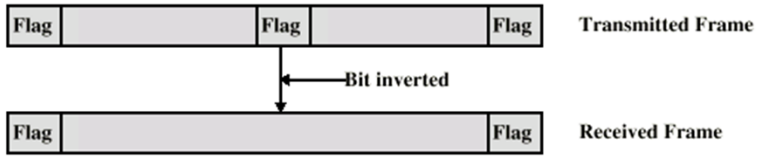
After bit-stuffing

```
11111011111011011111010111111010
```

(a) Example



(b) An inverted bit splits a frame in two

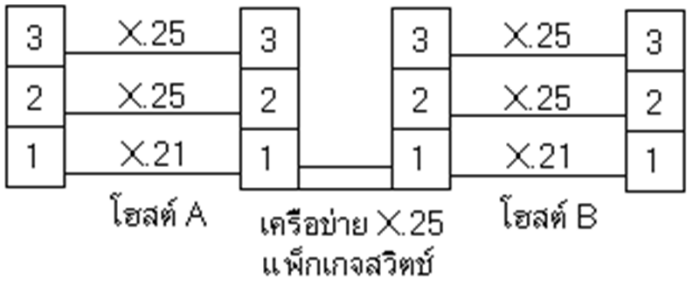


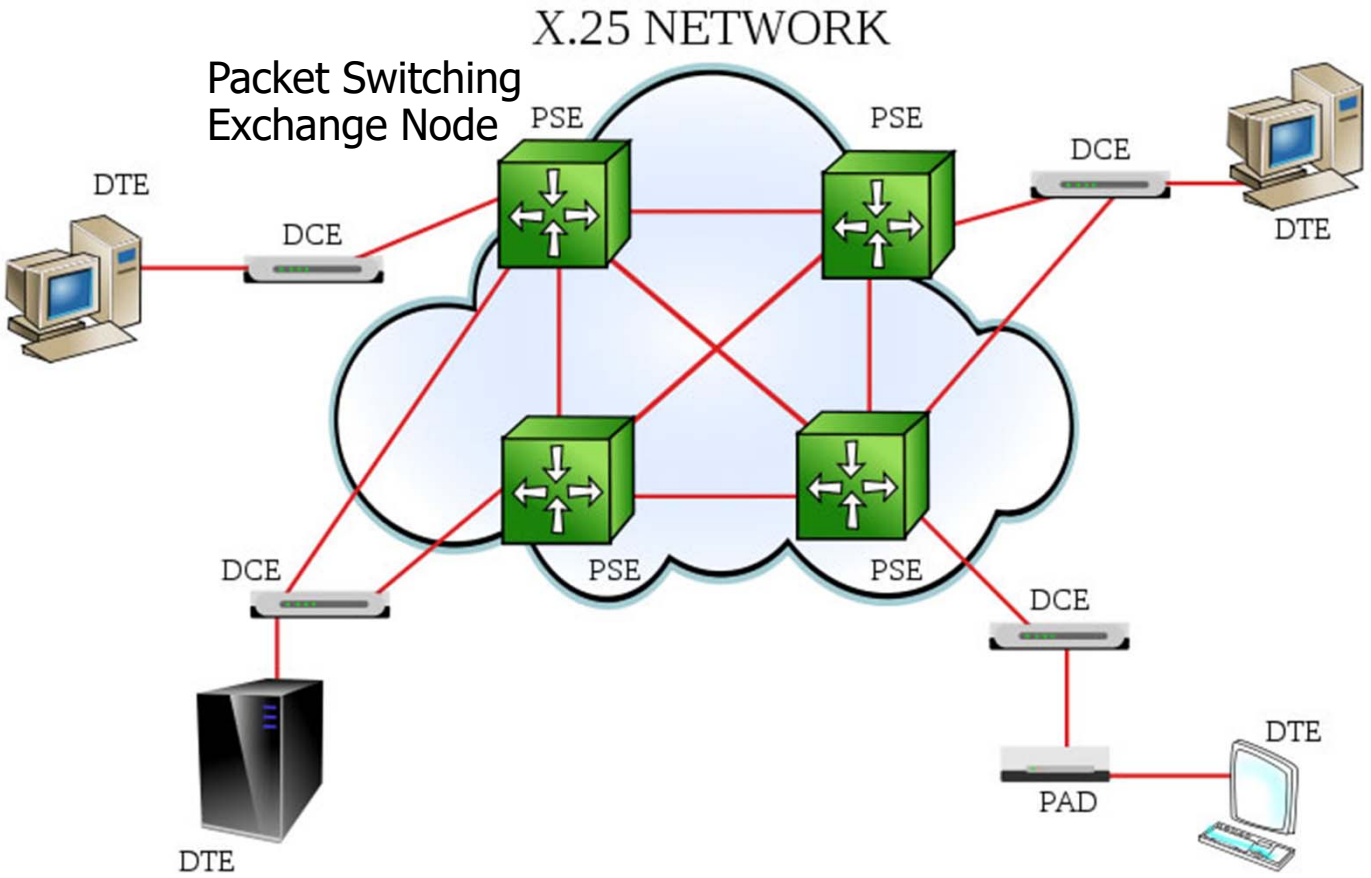
(c) An inverted bit merges two frames

HDLC เป็นต้นกำเนิดของ Frame Format และ L2 Protocol อื่นๆ

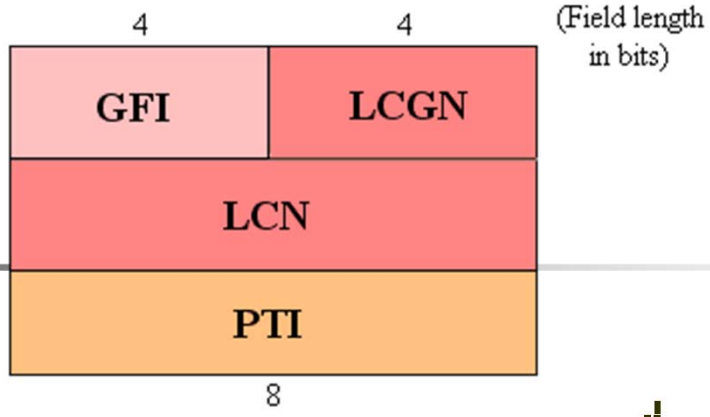
- -LLC
- -MAC
- -PPP
- -LAPB
- -LAPD
- -LAPF
- -ฯลฯ

X.25


3
2
1
X.25
X.25
X.21
3
2
1
โฮสต์ A
เครือข่าย X.25
แพ็กเกจสวิตช์
โฮสต์ B


X.25 NETWORK
Packet Switching
Exchange Node
PSE
PSE
PSE
PSE
DTE
DCE
DCE
DTE
DCE
DCE
DTE
PAD
DTE

X.25

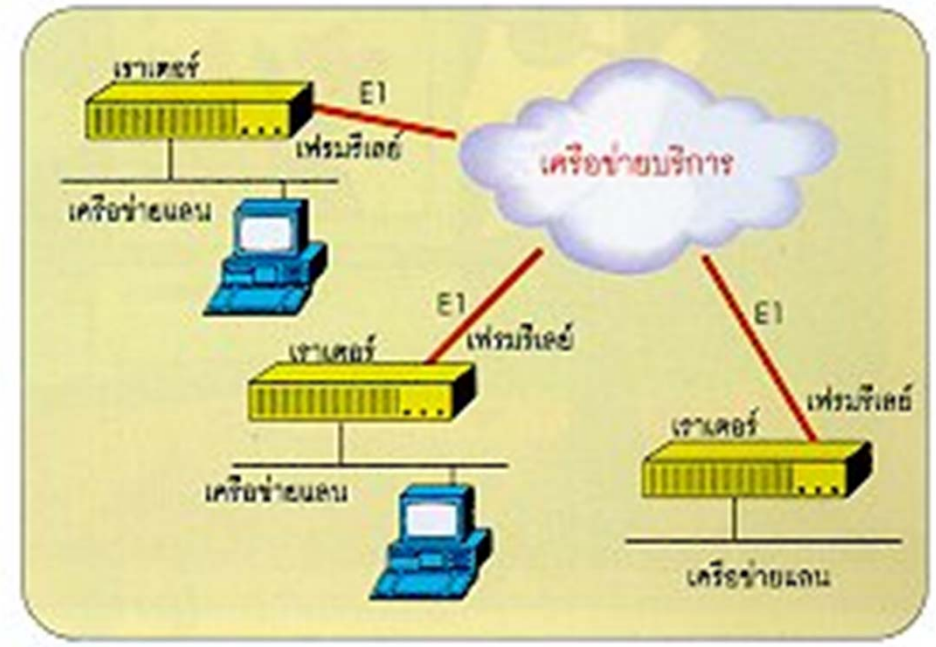


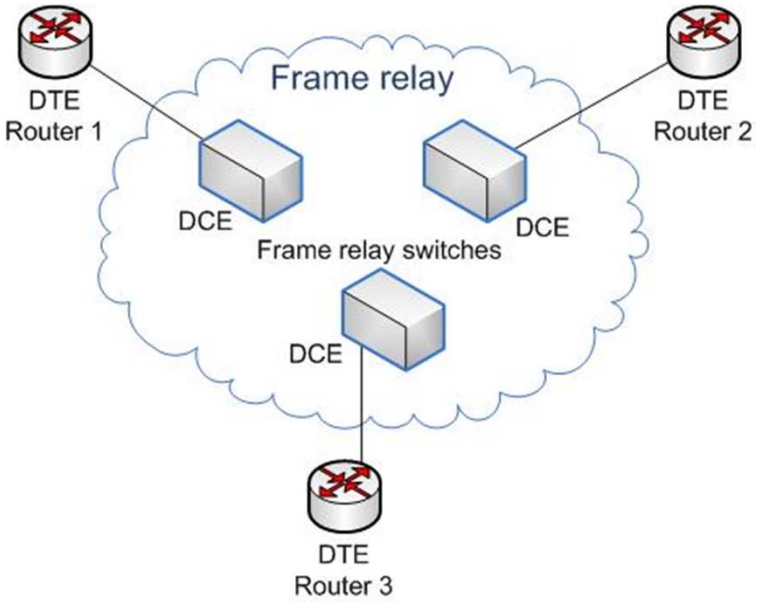
- **Physical Layer: กำหนดการเชื่อมต่อทางไฟฟ้า ระหว่าง DTE/DCE จะอยู่ใน X.21 หรือจะใช้ EIA-232, EIA-449 หรือ Serial Protocol อื่น**
- **Data Link Layer: กำหนดขบวนการใช้ Link สำหรับการส่งข้อมูลระหว่าง DTE/DCE จะใช้ LAPB**

Flag 01111110 (8bits)	Address (8bits)	Control (8bits)	Data (Variable)	Checksum (16 bits)	Flag 01111110 (8bits)

- **Packet Layer กำหนด Protocol ในระดับ Packet ในการแลกเปลี่ยน Control และ Data กับ PSN ผ่าน Virtual Circuit**

Frame Relay





พัฒนาต่อจาก X.25 ใช้ LAP-D ในการส่ง Data, ตัดส่วน Flow Control ออก และ Switch ใน L2 ทำให้ส่งข้อมูลได้เร็วและเป็น Stream มากขึ้น

Diagram of the UNI ATM Cell

ATM (Cell Switching)

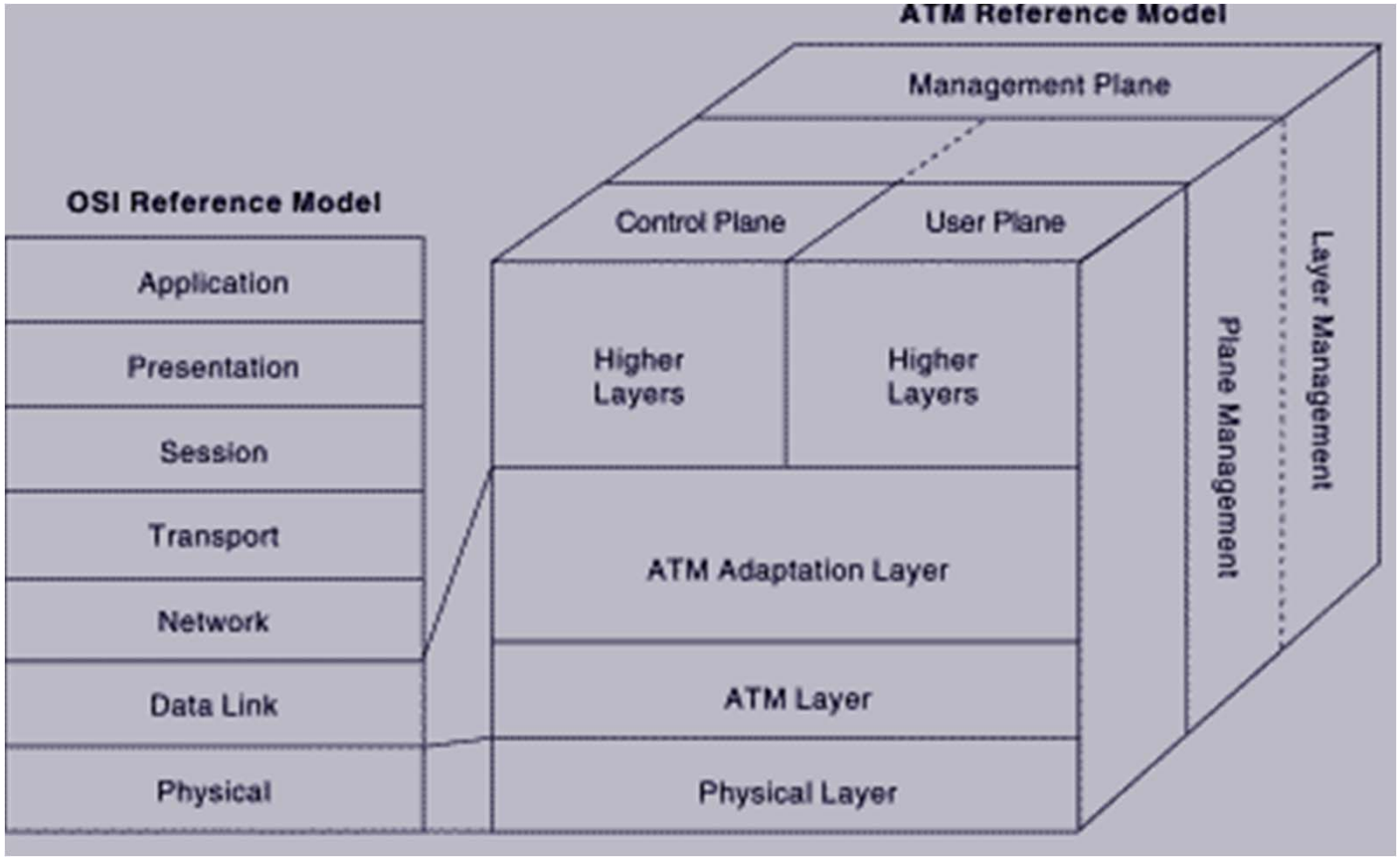
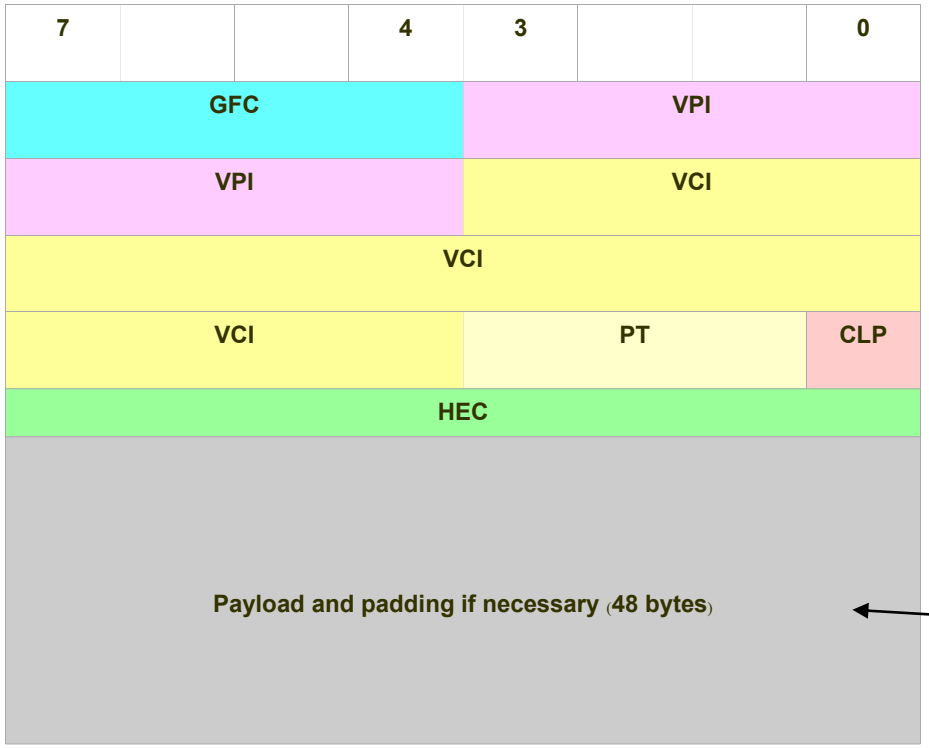


Diagram of the UNI ATM Cell

ATM (Cell Switching)

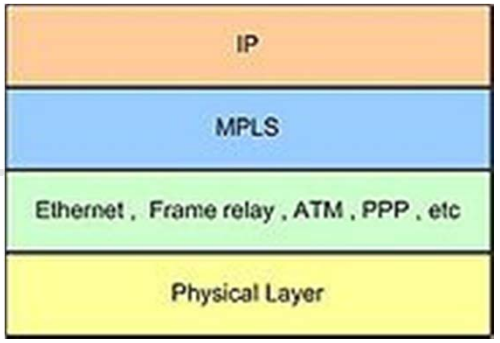


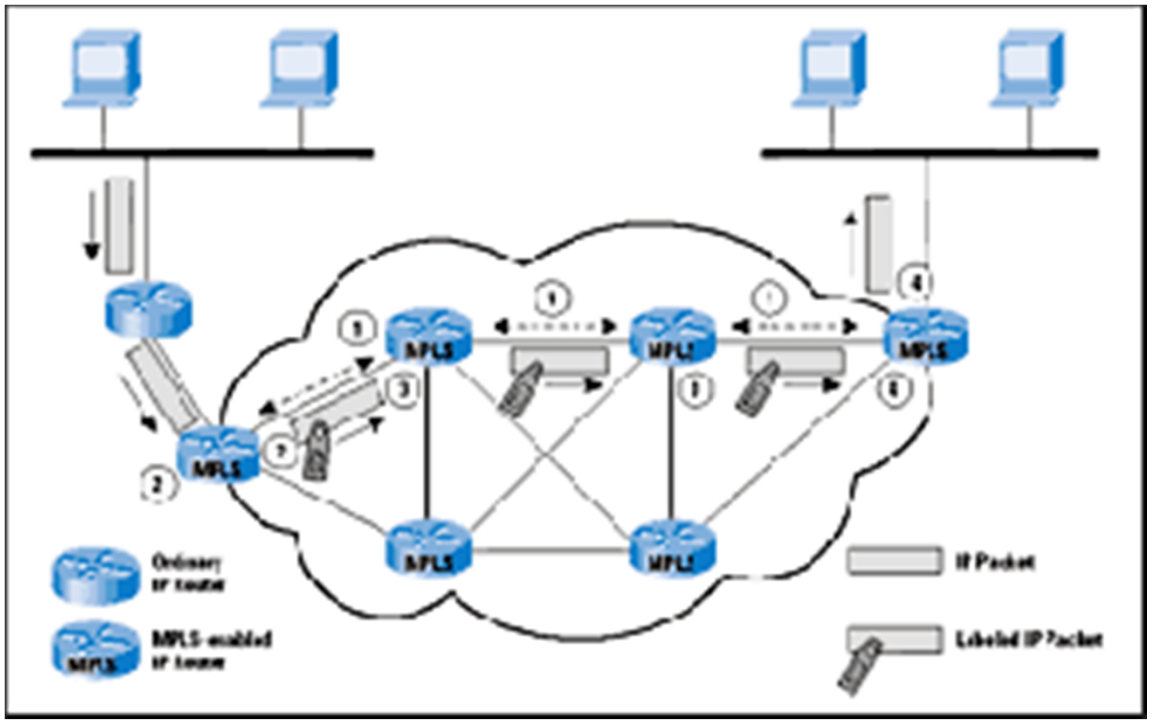
ปัจจุบัน SDH ถูกใช้ในการ Transport ATM

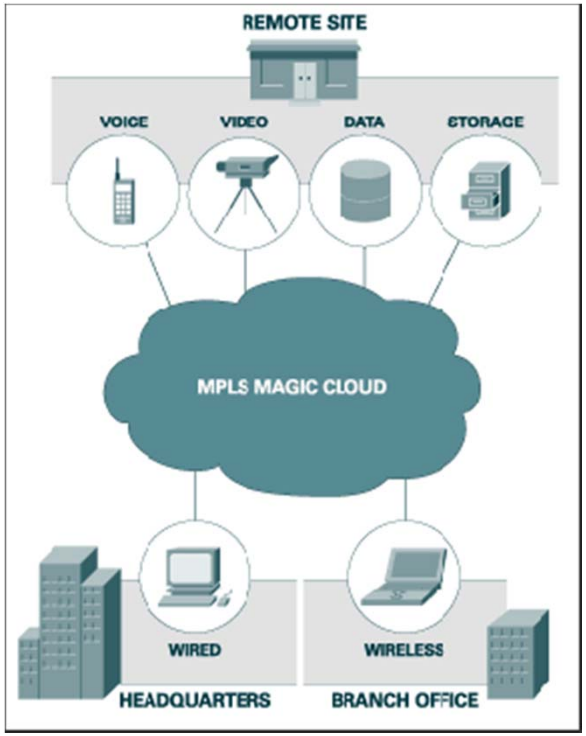
- CBR
- VBR
- ABR
- UBR

AAL Type 1-5

MPLS (Multiprotocol Label Switching)



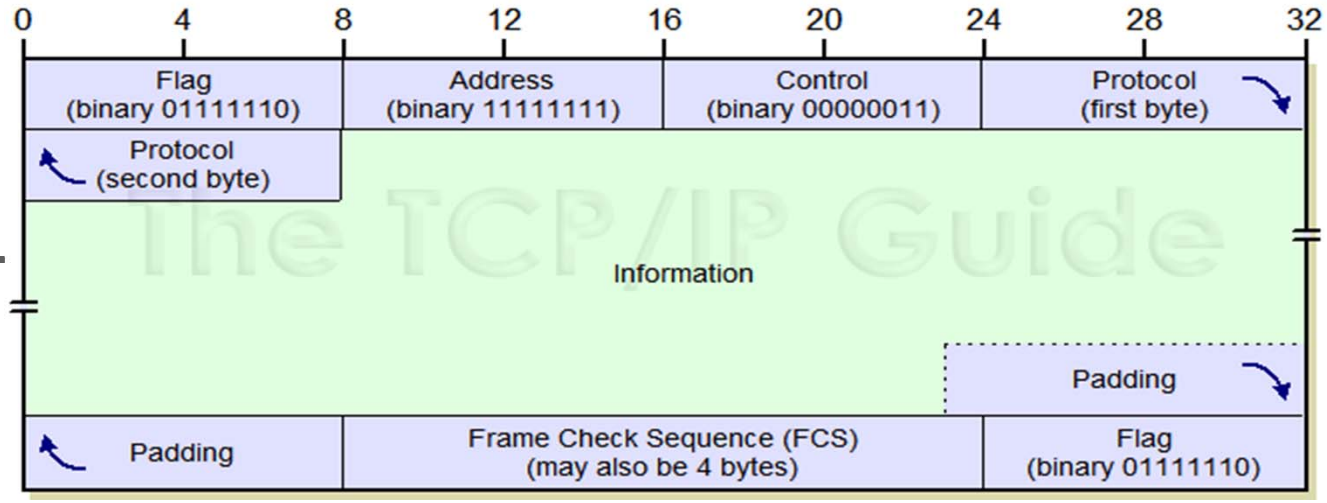




PPP (Point-to-Point Protocol)

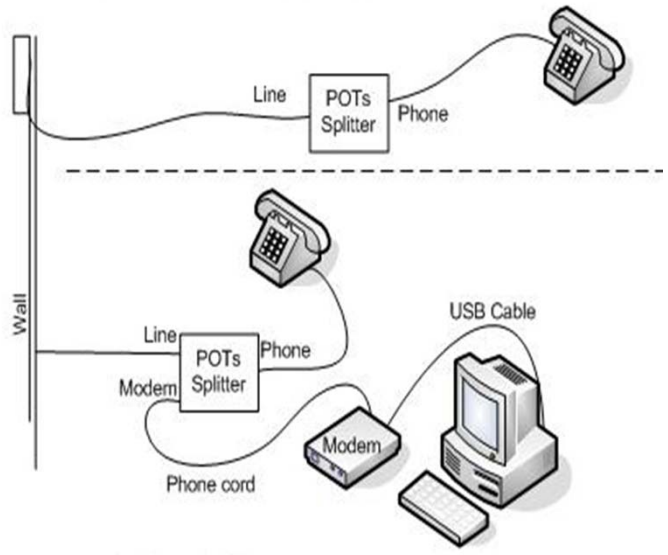
- นิยมใช้ในปัจจุบัน สำหรับเป็น Data Link Protocol ในการเชื่อมต่อโดยตรงระหว่าง Node (Point-to-Point)
- ใช้ได้ผ่าน Physical Link หลายแบบ เช่น Serial Cable, Phone Line, Cell Phone, SONET โดยที่ ISP ส่วนใหญ่จะใช้สำหรับลูกค้าที่จะ Dial-Up Access กับ Internet
- มาแทนที่ Protocol เก่าได้แก่
  - SLIP (Serial Line Internet Protocol)
  - LAPB ใน X.25
- ถูกออกแบบมาให้ใช้กับ Network Layer ต่างๆ รวมถึง IP
- ยังถูกใช้เป็น Protocol ในการเชื่อมต่อ Broadband ด้วย ใน PPPoE และ PPPoA

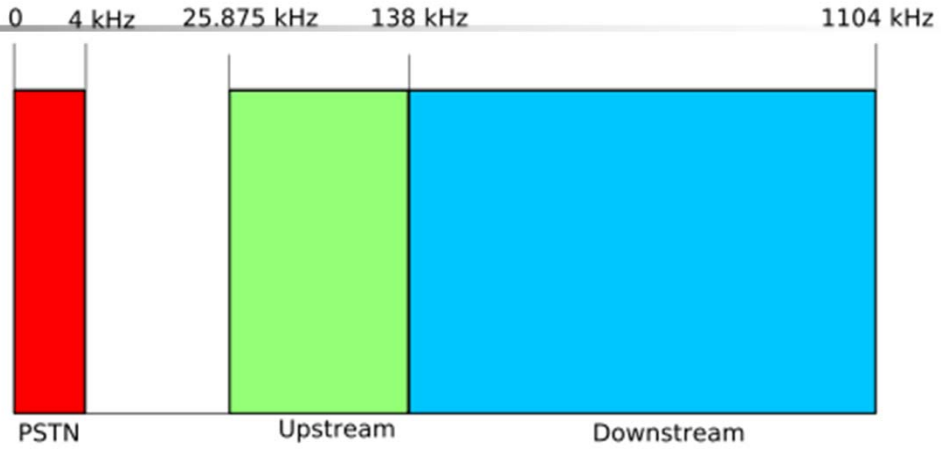
PPP Frame

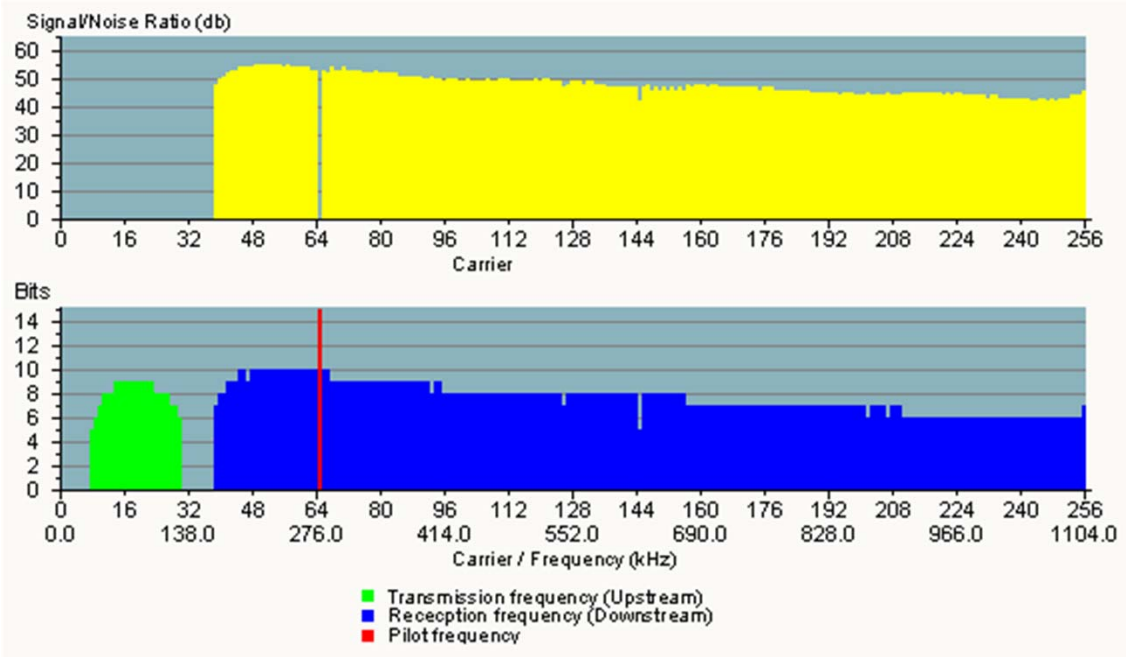


Field Name	Size (bytes)	Description
Flag	**1**	*Flag:* **Indicates the start of a PPP frame. Always has the value "01111110" binary (0x7E hexadecimal, or 126 decimal).**
Address	**1**	*Address:* **In HDLC this is the address of the destination of the frame. But in PPP we are dealing with a direct link between two devices, so this field has no real meaning. It is thus always set to "11111111" (0xFF or 255 decimal), which is equivalent to a broadcast (it means "all stations").**
Control	**1**	*Control:* **This field is used in HDLC for various control purposes, but in PPP it is set to "00000011" (3 decimal).**
Protocol	**2**	*Protocol:* **Identifies the protocol of the datagram encapsulated in the Information field of the frame. See below for more information on the *Protocol* field.**
Information	**Variable**	*Information:* **Zero or more bytes of payload that contains either data or control information, depending on the frame type. For regular PPP data frames the network-layer datagram is encapsulated here. For control frames, the control information fields are placed here instead.**
Padding	**Variable**	*Padding:* **In some cases, additional dummy bytes may be added to pad out the size of the PPP frame.**
FCS	**2 (or 4)**	*Frame Check Sequence (FCS):* **A checksum computed over the frame to provide basic protection against errors in transmission. This is a CRC code similar to the one used for other layer two protocol error protection schemes such as the one used in Ethernet. It can be either 16 bits or 32 bits in size (default is 16 bits).** **The FCS is calculated over the *Address*, *Control*, *Protocol*, *Information* and *Padding* fields.**
Flag	**1**	*Flag:* **Indicates the end of a PPP frame. Always has the value "01111110" binary (0x7E hexadecimal, or 126 decimal).**

Broad-band (ADSL)



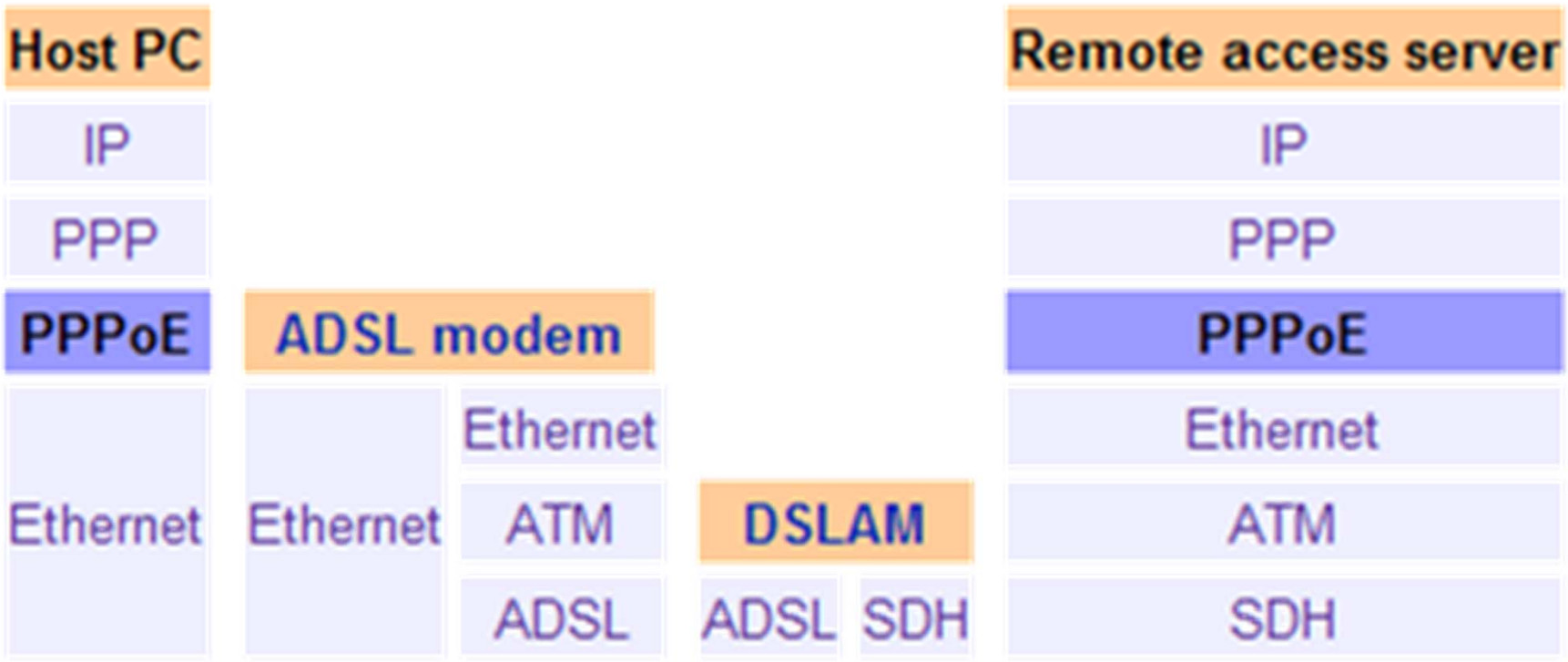




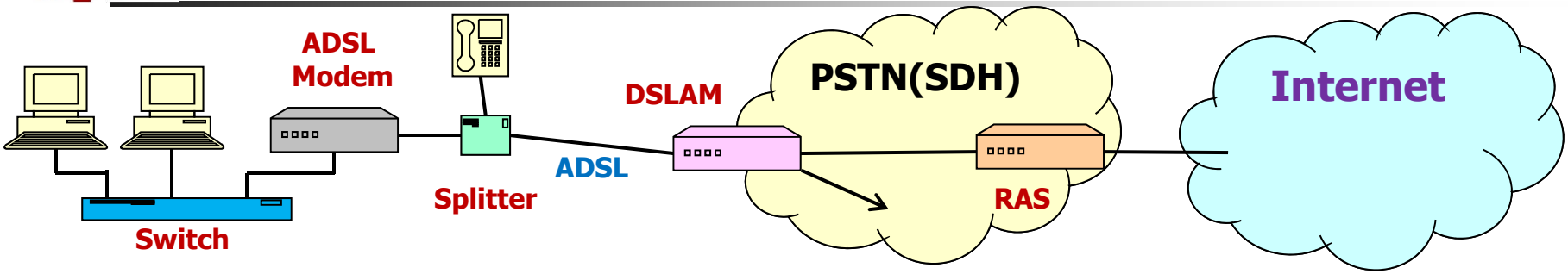
PPPoE

Application	FTP	SMTP	HTTP	…	DNS	…
Transport	TCP				UDP	
Network	IP				IPv6	
Network access	PPP					
	PPPoE					
	Ethernet					

ADSL internet access architecture



PPPoE



Host PC				Remote access server
IP				IP
PPP				PPP
PPPoE	ADSL modem			PPPoE
Ethernet	Ethernet	Ethernet		Ethernet
		ATM	DSLAM	ATM
		ADSL	ADSL SDH	SDH

Importance Ethernet Standards

Ethernet Standard	Date	Description
Experimental Ethernet	1973[1]	2.94 Mbit/s (367 kB/s) over coaxial cable (coax) bus
Ethernet II (DIX v2.0)	1982	10 Mbit/s (1.25 MB/s) over thick coax. Frames have a Type field. This frame format is used on all forms of Ethernet by protocols in the Internet protocol suite.
IEEE 802.3 standard	1983	10BASE5 10 Mbit/s (1.25 MB/s) over thick coax. Same as Ethernet II (above) except Type field is replaced by Length, and an 802.2 LLC header follows the 802.3 header. Based on the CSMA/CD Process.
802.3a	1985	10BASE2 10 Mbit/s (1.25 MB/s) over thin Coax (a.k.a. thinnet or cheapernet)
802.3i	1990	10BASE-T 10 Mbit/s (1.25 MB/s) over twisted pair
802.3j	1993	10BASE-F 10 Mbit/s (1.25 MB/s) over Fiber-Optic

Importance Ethernet Standards

Ethernet Standard	Date	Description
802.3u	1995	100BASE-TX, 100BASE-T4, 100BASE-FX Fast Ethernet at 100 Mbit/s (12.5 MB/s) w/autonegotiation
802.3x	1997	Full Duplex and flow control; also incorporates DIX framing, so there's no longer a DIX/802.3 split
802.3ab	1999	1000BASE-T Gbit/s Ethernet over twisted pair at 1 Gbit/s (125 MB/s)
802.3ad	2000	Link aggregation for parallel links, since moved to IEEE 802.1AX
802.3ae	2002	10 Gigabit Ethernet over fiber; 10GBASE-SR, 10GBASE-LR, 10GBASE-ER, 10GBASE-SW, 10GBASE-LW, 10GBASE-EW
802.3af	2003	Power over Ethernet (15.4 W)
802.3an	2006	10GBASE-T 10 Gbit/s (1,250 MB/s) Ethernet over unshielded twisted pair (UTP)
802.3at	2009	Power over Ethernet enhancements (25.5 W)

Ethernet Standard	Date	Description
802.3ba	2010	40 Gbit/s and 100 Gbit/s Ethernet. 40 Gbit/s over 1m backplane, 10 m Cu cable assembly (4x25 Gbit or 10x10 Gbit lanes) and 100 m of MMF and 100 Gbit/s up to 10 m of Cu cable assembly, 100 m of MMF or 40 km of SMF respectively
802.3.1	2011	MIB definitions for Ethernet. It consolidates the Ethernet related MIBs present in Annex 30A&B, various IETF RFCs, and 802.1AB annex F into one master document with a machine readable extract. (workgroup name was P802.3be)
802.3bm	2015	100G/40G Ethernet for optical fiber
802.3bq	~Feb 2016	40GBASE-T for 4-pair balanced twisted-pair cabling with 2 connectors over 30 m distances
802.3bs	~ 2017	400 Gbit/s Ethernet over optical fiber using multiple 25G/50G lanes
802.3by	~Sep 2016	25G Ethernet
802.3bz	TBD	2.5 Gigabit and 5 Gigabit Ethernet over twisted pair - 2.5GBASE-T and 5GBASE-T

- **END OF REVIEW**
- **BREAK**
- **CHAPTER 20: TCP/IP Concept**

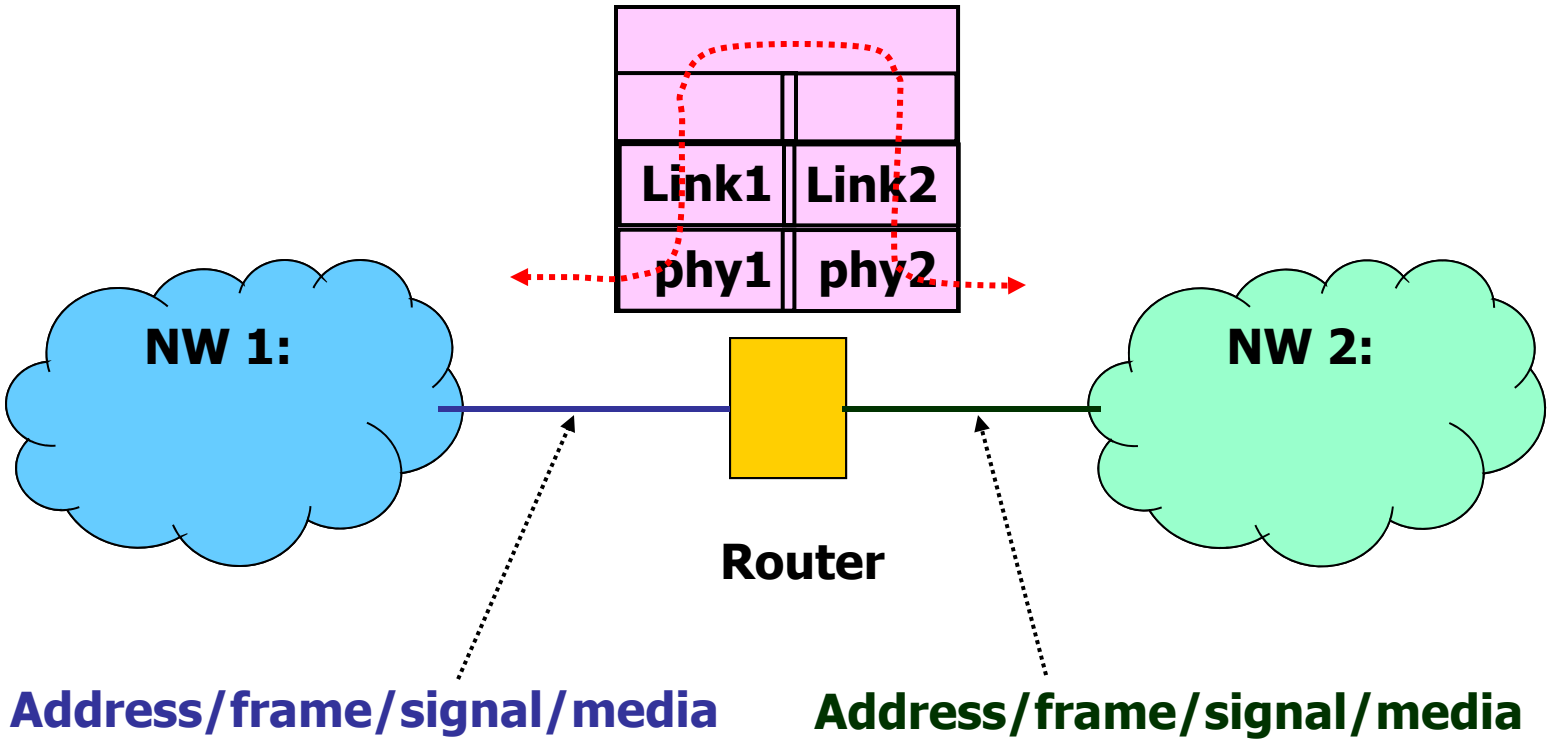
Chapter 20: TCP/IP Concept/ ที่มาของปัญหา

- **สมัยก่อน การนำ Network มาใช้งาน จะขึ้นอยู่กับงานที่ต้องการทำให้ NW นั้นมีหลากหลาย**
  - แต่ละองค์กร มี NW ที่ใช้ Technology ต่างๆกัน
- **การที่จะเชื่อม NW ต่างๆเหล่านี้ด้วยกัน เพื่อ Share ข้อมูลไม่สามารถทำได้ เพราะ Protocol ที่ใช้ไม่เหมือนกัน**
  - คอมพิวเตอร์ใน NW หนึ่งสื่อสารได้กับคอมพิวเตอร์ใน NW เดียวกันเท่านั้น
  - ถ้าองค์กรมีหลาย NW เพื่อที่จะทำงานหลายๆอย่าง ผู้ใช้งานต้องมีคอมพิวเตอร์หลายตัว เพื่อเชื่อมต่อแต่ละ NW สำหรับงานแต่ละประเภท
- **สรุปแล้ว NW สมัยก่อนเป็น Heterogeneous**
- **เราต้องการหา Technology ที่สามารถจะเชื่อมต่อ NW เหล่านี้เข้าด้วยกัน เพื่อจะสื่อสารและแลกเปลี่ยนข้อมูลกันได้**
  - ปัญหาที่ต้องแก้มีมากมาย เช่น
    - เรื่องของสัญญาณและ Hardware ที่ต่างกัน
    - Frame Format ที่ไม่เหมือนกัน
    - ระบบการใช้ Address ที่ไม่เหมือนกัน

Chapter 20: TCP/IP ที่มาของปัญหา –> Solution: internetworking

- **เทคโนโลยีที่จะทำให้หลายๆ Network สามารถเชื่อมต่อกันได้เรียก**
  - Internetworking หรือ internet
  - (กรณีนี้ ยังไม่เจาะจงว่าเป็น IP Network ดังนั้นเราจะใช้คำว่า 'a internet' ไม่ใช่ 'the Internet'
- **รูปแบบการเชื่อมต่อ Network เข้าด้วยกัน จะใช้การต่อผ่านอุปกรณ์ที่ชื่อ 'Router'**
  - Router จะเชื่อมต่อ NW สองด้าน(หรือมากกว่า) เข้าด้วยกัน
    - หมายความว่า Port หนึ่งของ Router อาจจะ Run Protocol หนึ่ง ในขณะที่อีก Port หนึ่งจะ Run Protocol ที่ต่างกัน
      - แต่ละ Port ของ Router เราเรียก "Interface"
    - NW แต่ละด้าน อาจจะเป็นคนละ Technology รวมถึงใช้ Frame Format/Media/Address/Signal ที่แตกต่างกัน

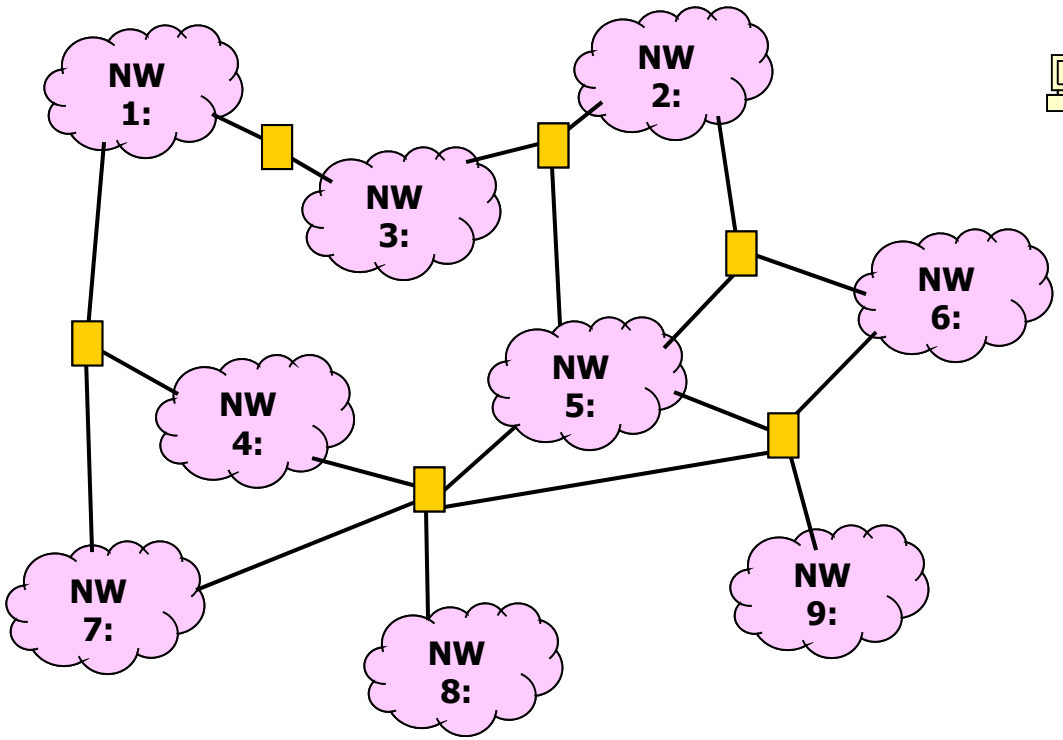
Chapter 20: Internetworking with Router

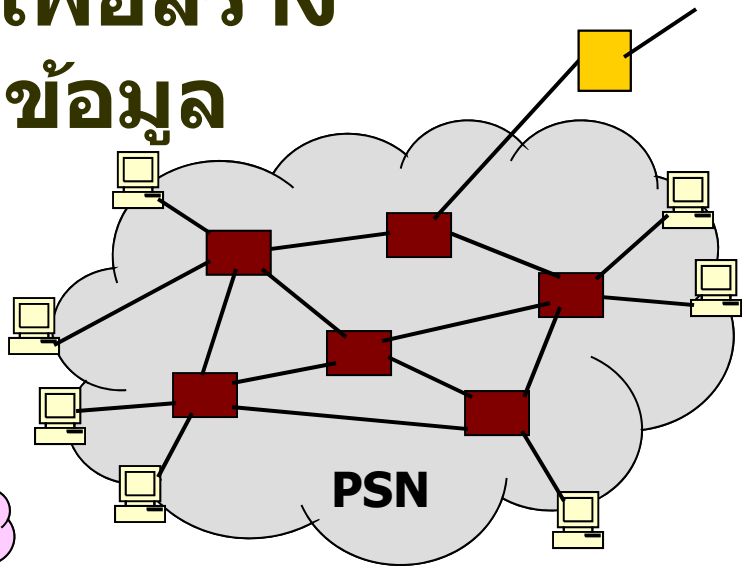


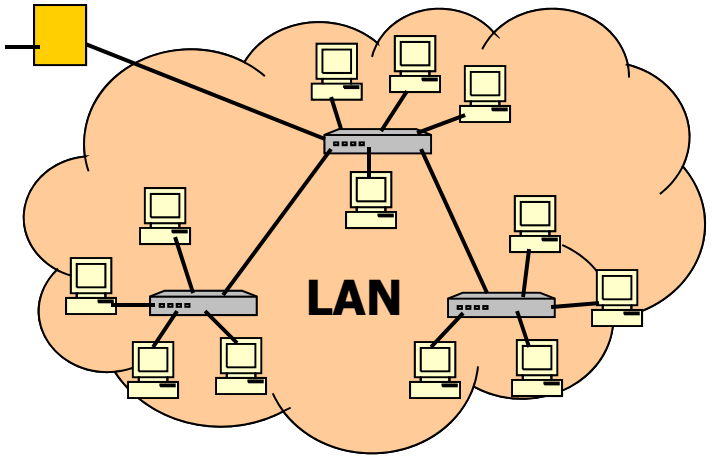
An internet ประกอบด้วย NW หลายๆตัว เชื่อมต่อกันผ่าน Router

Chapter 20: Internet Architecture

- มักจะใช้ Partial Mesh เพื่อสร้าง Redundancy ในการส่งข้อมูล





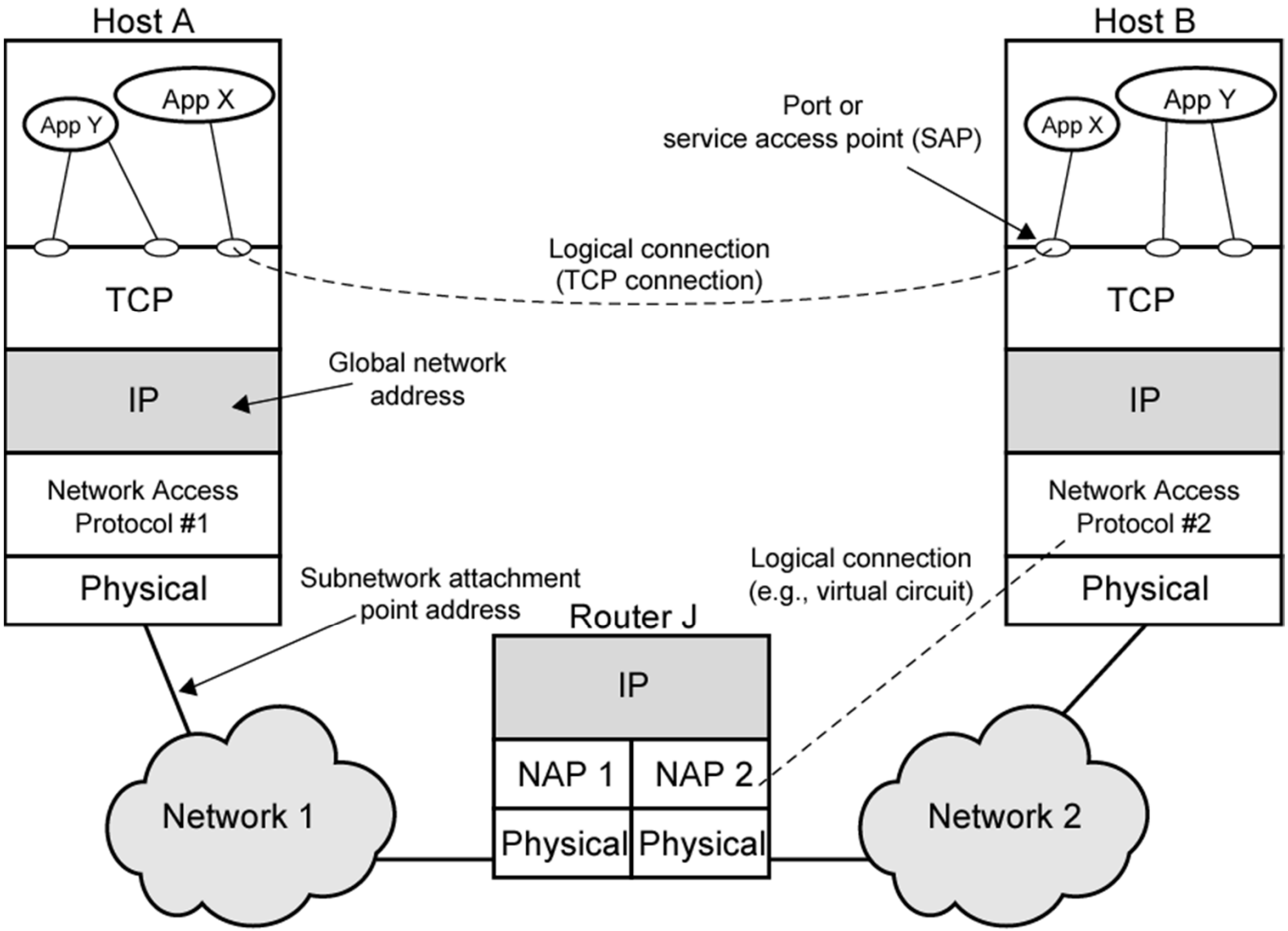


Internet Universal Service

- **แม้ว่า Router จะมาช่วยแก้ปัญหาในการเชื่อมต่อระหว่าง Network**
  - ปัญหายังคงอยู่ในการส่งข้อมูลผ่านระหว่าง Network เนื่องจาก Address ที่ใช้ในแต่ละ Network ต่างกัน
    - เราสามารถทำ Network Address Translation แต่จะยุ่งยาก
  - เราต้องการ Protocol กลางที่จะสามารถเชื่อมต่อคอมพิวเตอร์ต้นทาง ผ่าน Router ไปยังคอมพิวเตอร์ปลายทางที่ต่าง Network
    - Software Protocol นี้ต้อง Run ที่คอมพิวเตอร์ต้นทางและปลายทาง รวมถึงที่ Router ด้วย
    - Protocol ดังกล่าวจะใช้ Network Address ที่เป็น Universal สามารถบ่งบอกตำแหน่งอุปกรณ์ทั้งต้นทางและปลายทาง
  - Packet และรับข้อมูลส่งให้ลำดับชั้นบนของ Internet Protocol จนถึง Application Layer

Internet Universal Service

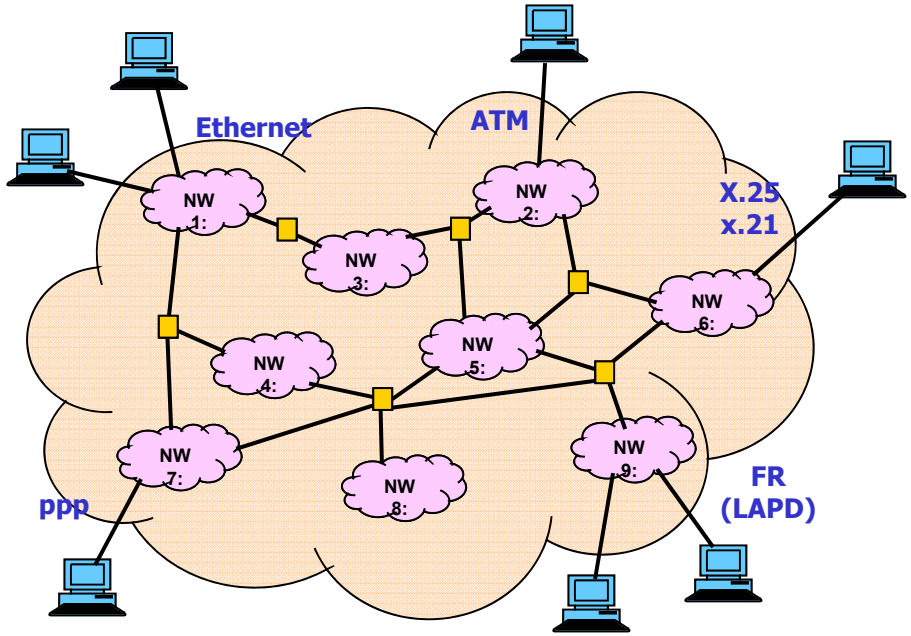
- **แม้ว่า Router จะมาช่วยแก้ปัญหาในการเชื่อมต่อระหว่าง Network**
  - การทำงาน
    - ที่ PC ต้นทาง จะ Run Internet Protocol ที่กล่าว จากนั้นเมื่อส่งข้อมูลผ่าน Network1 จะส่ง Packet ของ Protocol นั้น ผ่าน Network Protocol ที่มันเชื่อมต่ออยู่ไปยัง Router (บางทีเราเรียกว่าเป็นการทำ Tunneling)
    - ที่ Router เมื่อได้รับ Packet จะปลอก Network Protocol ที่เชื่อมต่อออก จนเหลือ Internet Protocol จากนั้นจะใช้ Address ของ Internet Protocol ทำการหาเส้นทางเพื่อส่งข้อมูลต่อออกไปยัง Interface ที่ถูกต้อง ซึ่งจะ Run Network Protocol ที่อาจจะแตกต่างกัน ดังนั้นมันจะจับ Internet Packet ใส่ลงใน Protocol ของ Interface ที่เชื่อมต่อนั้นส่งออกไป
    - ข้อมูลอาจจะต้องส่งผ่านหลาย Network และหลาย Router แต่ละช่วงเรียกว่า Hop จะมีการปลอก Packet จนถึง Internet Protocol และประกอบด้วย Protocol ใหม่ตาม Interface ขาออกที่ต้องส่งออกไป
    - เมื่อข้อมูลถึงคอมพิวเตอร์ปลายทาง คอมพิวเตอร์ปลายทางจะปลอก Packet จนถึง Internet Protocol จากนั้นจะตรวจสอบ Internet Packet และรับข้อมูลส่งให้ลำดับชั้นบนของ Internet Protocl จนถึง Application Layer

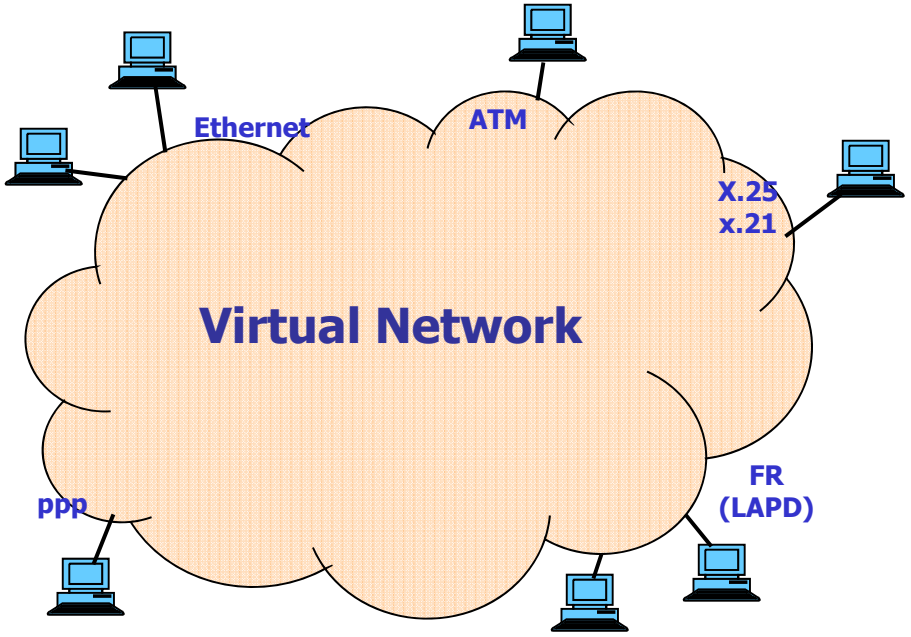

Host A
App Y
App X
TCP
IP
Network Access Protocol #1
Physical
Host B
App X
App Y
TCP
IP
Network Access Protocol #2
Physical
Port or service access point (SAP)
Logical connection (TCP connection)
Global network address
Subnetwork attachment point address
Logical connection (e.g., virtual circuit)
Router J
IP
NAP 1
NAP 2
Physical
Physical
Network 1
Network 2

Virtual Network

- **ลักษณะของ Internet Software ได้ถูกออกแบบมาเพื่อซ่อน Layer ที่อยู่ด้านล่าง**
  - Layer ล่างจะเป็นอะไรก็ได้ (ปกติจะเป็น Protocol L2 หรือ L3) ผู้ใช้จะมองไม่เห็นและไม่ต้องสนใจ เพราะ Protocol เหล่านี้ใช้ในการทำ Network Access ในมุมมองของ Internet
    - Ethernet, HDLC, ppp, X.25, FR, ATM, ฯลฯ
  - ผู้ใช้แค่กำหนด Application ที่จะ Run และกำหนด Internet Address ปลายทางที่จะส่ง
    - การต่อกับ Internet ผู้ใช้เลือก Protocol ที่เหมาะสมที่จะไปเชื่อมต่อกับ Network ที่ต่ออยู่แล้วภายใน Internet
      - L1+L2+(L3)
  - นี่คือ Concept ของ Virtual Network

Virtual Network



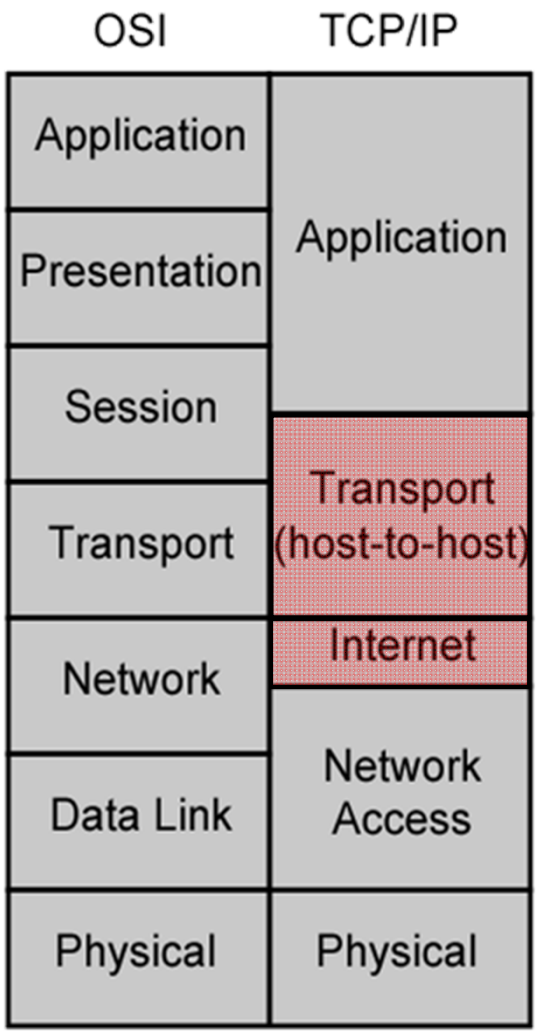


Concept ของ The Internet คือถ้าเราต้องการต่อกับ Internet เราเลือก Network Access Protocol ที่เหมาะสม ทำการเชื่อมต่อกับ Network ที่ต่ออยู่แล้วใน Internet ในทางปฏิบัติ คงไม่มีใครที่อยากให้เราต่อพ่วงด้วย ดังนั้นจึงเกิด ISP (Internet Service Provider) ที่จะรับการเชื่อมต่อดังกล่าว โดยมีการคิดค่าบริการ

Internetworking Protocol

- **แม้ว่ามีหลาย Protocol ที่ถูกเสนอขึ้นมา แต่มีเพียงชุดของ Protocol เดียวที่ได้รับความนิยมอย่างกว้างขวาง นั่นคือ ชุดของ Protocol TCP/IP**
  - พัฒนาเริ่มจากปี 1970s ซึ่งเป็นเวลาเดียวกันกับที่ Ethernet ถูกพัฒนาขึ้นมา
  - ต่อมาในปี 1990s Protocol นี้ได้ถูกปล่อยและนำมาใช้ใน Commercial

TCP/IP Protocol Stack



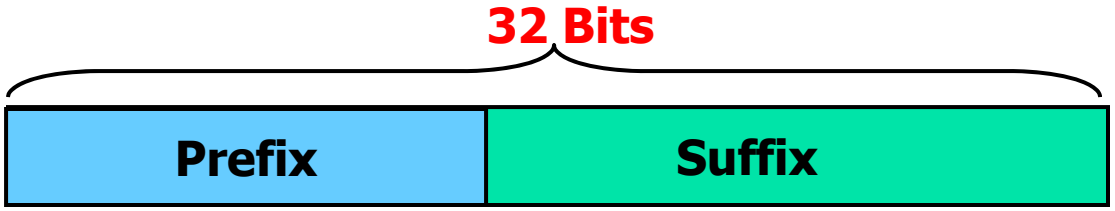
- ที่สำคัญคือ **Layer 3: Internet Protocol(IP)** ทำหน้าที่กำหนดรูปแบบของ **Packet** ที่จะส่งผ่านตลอดทั้ง **Network** โดยที่ส่วน **Address** ของ **Layer** นี้ (**IP Address**) จะถูกใช้โดย **Router** เพื่อที่จะทำการส่งข้อมูลผ่าน **Router** ที่เชื่อมต่อกัน จนกระทั่งถึงเครื่อง (**Host**) ปลายทาง ดังนั้น **Protocol** นี้จะต้องถูก **Run** ใน **Router** ทุกตัว รวมทั้ง **Host**
- **Layer 4: Transport Layer (Host-to-Host)** ที่สำคัญคือ **TCP** จะถูกใช้เพื่อที่จะให้แน่ใจว่าข้อมูลที่ส่งจากต้นทาง(**Host**) ไปถึงปลายทาง (**Host**) ได้อย่างถูกต้อง ซึ่ง **Protocol** นี้จะถูก **Run** ที่ **Host** ต้นทางและปลายทางเท่านั้น

Chapter 21: IP Addressing

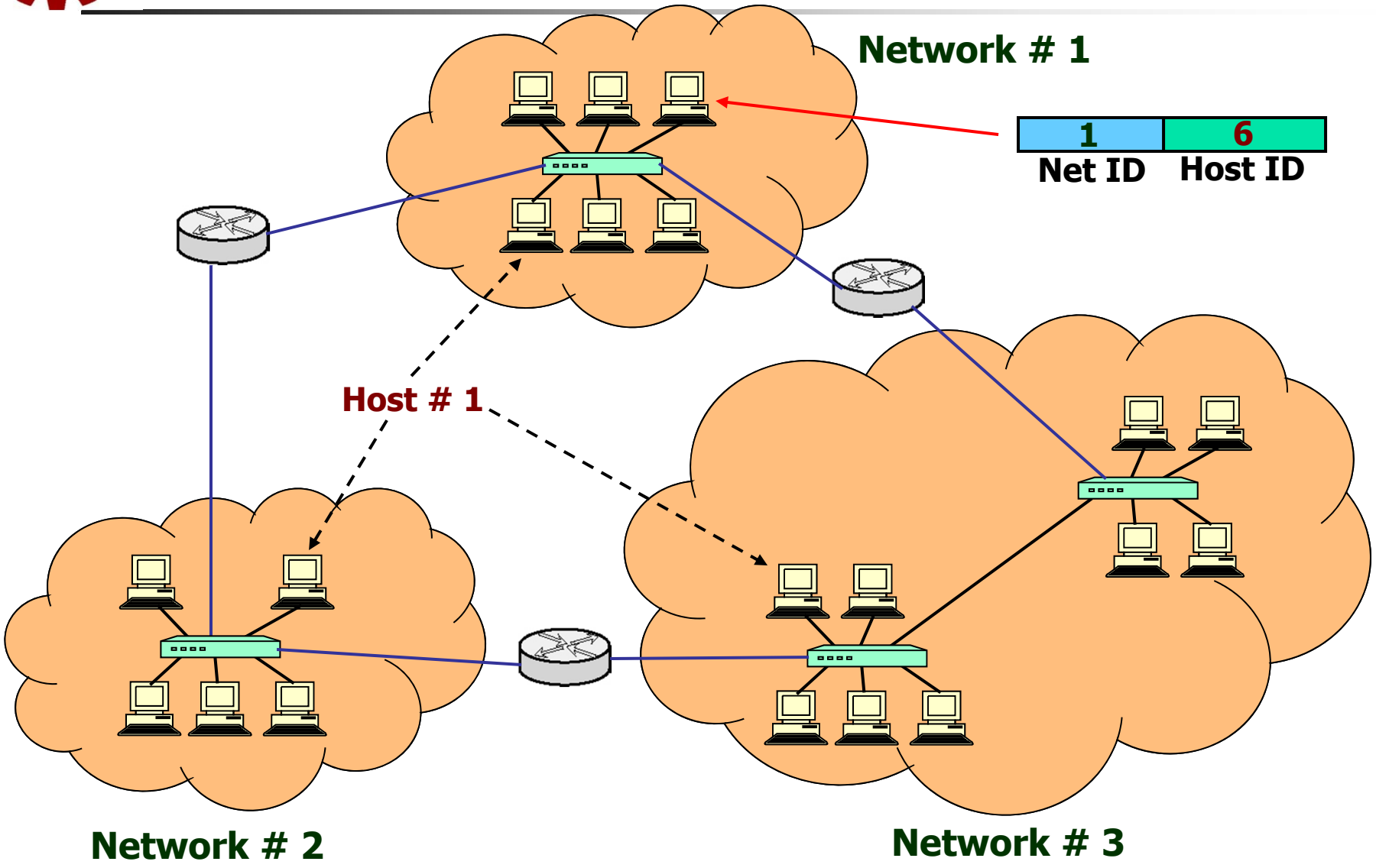
- **IP Address เป็นตัวกำหนด End System (Host) ตลอดทั้งเครือข่าย Internet**
  - ดังนั้นมันจะต้องเป็น Global Address ขณะที่ MAC Address จะหมดอายุเมื่อออกจาก LAN
    - เป็นแค่ Local Address ใน LAN
  - เครื่องสองเครื่องในเครือข่าย จะมีหมายเลขเดียวกันไม่ได้
    - เครื่องที่ต่อใน LAN ออก Internet จะมีทั้ง MAC Address และ IP Address ที่ Match กัน
      - Protocol ARP จะกล่าวในบทที่ 23
- **ปัจจุบันใช้ตามมาตรฐานของ IPv4**
  - Address ขนาด 32 บิต
    - เรียก IP Address, Internet Address หรือ Internet Protocol Address
  - กำหนดหมายเลขเครื่องได้โดยไม่เกี่ยวข้องกับ MAC Address

Ch. 21: 21.4 IP Address Hierarchy

- **แต่ละ IP Address ขนาด 32 บิต จะถูกแบ่งออกเป็นสองส่วน**
  - ส่วนต้น เรียก Prefix เป็นตัวกำหนดหมายเลข Network (Network ID)
  - ส่วนที่เหลือ เรียก Suffix เป็นตัวกำหนดหมายเลข Host ใน Network นั้นๆ
    - หมายเลข Host ใน Network เดียวกันจะซ้ำกันไม่ได้
    - หมายเลข Host ที่อยู่คนละ Network สามารถซ้ำกันได้ เพราะส่วน Prefix (Net ID) นั้นไม่เหมือนกัน



Ch. 21: 21.4 IP Address Hierarchy



Ch. 21: 21.5 Original Classful IP Addressing

- **เราควรกำหนดกี่บิตเป็น Prefix และ Suffix**
  - จำนวน = $2^{\text{bit ที่ใช้กำหนด}}$ ทั้ง Prefix และ Suffix
  - Prefix มากเกินไป ทำให้จำนวน Suffix Bit มีน้อย ไม่พอรองรับจำนวน Host ใน Network ขนาดใหญ่
  - Prefix น้อยไป จำนวน Network ใน Internet มีได้น้อย ไม่เพียงพอในการใช้งาน
  - เนื่องจาก Network มีขนาดหลากหลาย จึงกำหนด 4 บิตแรก บ่งบอก Class ของ IP Address ซึ่งจะเป็นการกำหนดจำนวนบิตที่เป็นทั้ง Prefix และ Suffix
    - Class A-C ใช้ปกติ
    - Class D สำหรับทำ Multicast (ไม่มี Prefix และ Suffix)
    - Class E Reserved

Ch. 21: 21.5 Original Classful IP Addressing

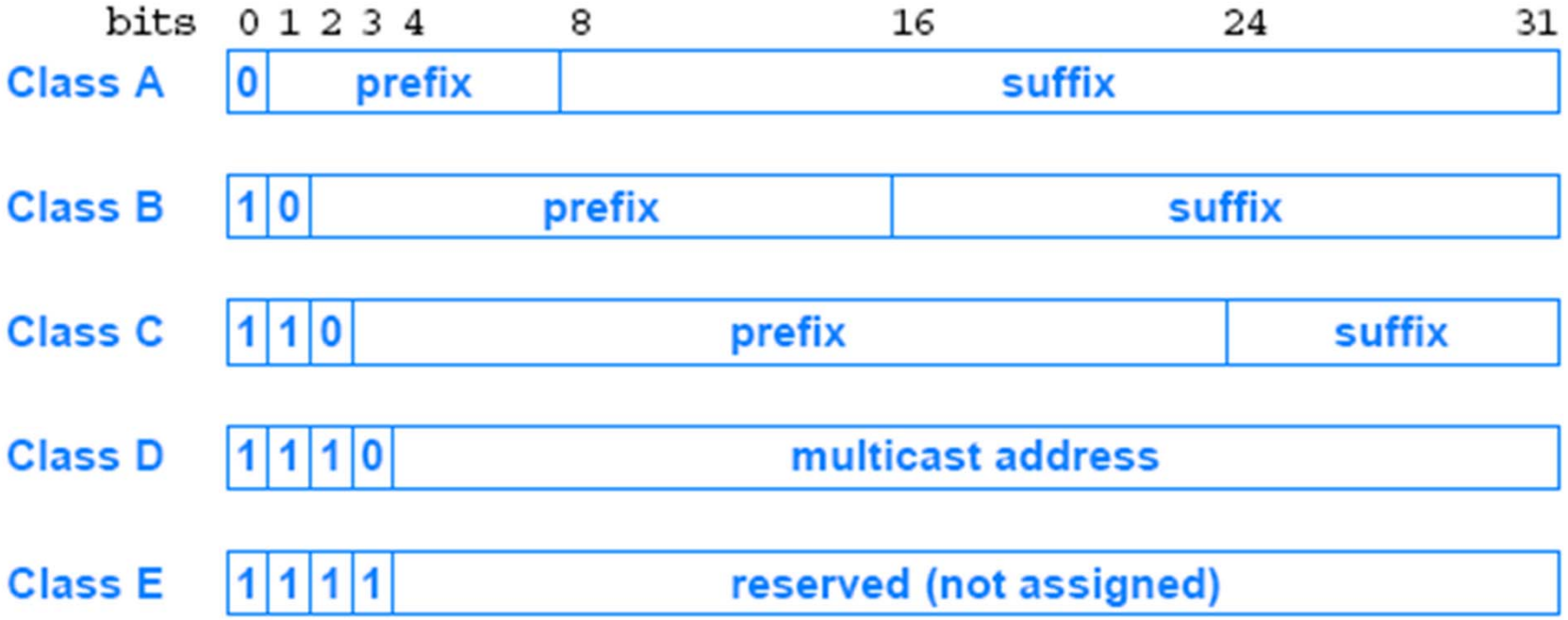


Figure 21.1 The five classes of IP addresses in the original classful scheme.

Ch. 21: 21.6 Dotted Decimal Notation

- **ใน 32 บิต IP Address จะถูกแบ่งเป็นสี่ส่วน ส่วนละ 8 บิต**
  - แต่ละส่วนจะเขียนเป็นเลขฐาน 10 มีค่าได้ระหว่าง 0 – 255
  - แต่ละส่วนจะเขียนต่อกัน ขั้นด้วย “จุด”
- **เรียก Dotted Decimal Notation**
  - Address ต่ำสุดคือ 0.0.0.0
  - Address สูงสุดคือ 255.255.255.255
  - แต่ละ Class สามารถสังเกตุได้จาก Octet แรกของ IP Address

Ch. 21: 21.6 Dotted Decimal Notation

32-bit Binary Number	Equivalent Dotted Decimal
10000001 00110100 00000110 00000000	129.52.6.0
11000000 00000101 00110000 00000011	192.5.48.3
00001010 00000010 00000000 00100101	10.2.0.37
10000000 00001010 00000010 00000011	128.10.2.3
10000000 10000000 11111111 00000000	128.128.255.0

Figure 21.2 Examples of 32-bit binary numbers and their equivalent in dotted decimal notation.

Ch. 21: 21.7 Division of Address Space

- สังเกตุว่า แม้ว่า Class A จะมีแค่ 128 Network แต่มันประกอบด้วยครึ่งหนึ่งของ Address Space ทั้งหมด

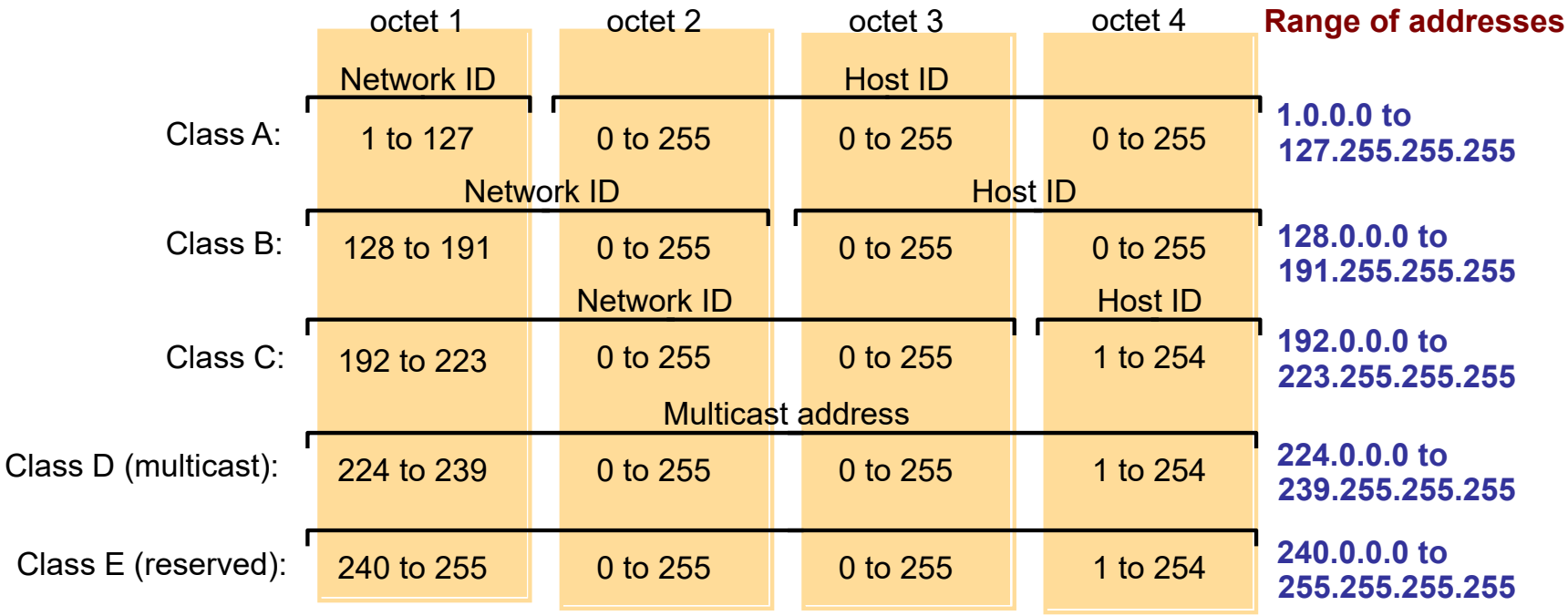
Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Figure 21.3 The number of networks and hosts per network in each of the original three primary IP address classes.

Ch. 21: 21.8 Authority for Address

- **องค์กรที่ดูแลจัดการเรื่อง IP Address คือ ICANN**
  - Internet Corporation for Assigned Names and Numbers
- **ปกติ ICANN จะกำหนดให้ Registrar เป็นผู้จัดสรรในแต่ละภูมิภาคอีกทีหนึ่ง**
- **Registrar จะจัดสรร Block ของ IP Address ให้แก่ ISP แต่ละราย**
  - ผู้ใช้งานจะได้รับ IP Address จาก ISP อีกที

Summary of IP Classful



- **END of Week 3**
- **Download HW 2: WK 3**
- **พิมพ์ลงบนกระดาษ A4**
- **ทำการบ้านลงในช่องที่กำหนด**
- **ส่งสัปดาห์หน้า ต้นชั่วโมง**

CPE 426 Computer Networks

Chapter 4:

Text Chapter 21: IP Address

Text Chapter 22: IP Datagram

TOPICS

- **Chapter 21: IP Address**
  - Addressing Scheme/Prefix&Suffix: 21.1-21.8
  - Subnetting and Mask: 21.9-21.13
  - Special IP Address: 21.14
  - Router IP Addressing/Multi-Homed Host: 21.17-21.18
- **BREAK**
- **Chapter 22: IP Datagram**
  - Header Format: 22.1-22.5
  - Datagram Forwarding: 22.6-22.10
  - Encapsulation: 22.11-22.12
  - MTU and Fragmentation: 22.13-22.17

สรุปบทที่ 20

- **การเชื่อมต่อระหว่าง Network เข้าด้วยกัน จะใช้อุปกรณ์ชื่อ Router ซึ่งทำงานใน Layer 3**
  - แต่ละ Interface ของ Router จะเชื่อมต่อกับแต่ละ Network และต้อง Run Network Protocol ตาม Network ที่มันไปเชื่อมต่ออยู่
- **เราจะต้องมี Global Address ที่ไปขึ้นกับ Address ของแต่ละ Network ที่จะบ่งบอกหมายเลข Host ในการส่งข้อมูลระหว่างสอง Host ที่อยู่คนละ Network**
  - Global Address นี้ต้องสามารถ Mapping กับ Network Address ที่กำหนด Host ในแต่ละ Network ได้ (ต้องมีขบวนการแปลง Address)
  - อย่างไรก็ตาม ในการส่งข้อมูลภายใน Network ยังคงอ้างอิงกับ Address ตาม Protocol ของ Network นั้นๆ ซึ่ง Address นี้จะมี Scope เพียงเฉพาะใน Network หนึ่งๆ เท่านั้น
- **เมื่อมีระบบ Address ใหม่ จำเป็นต้องมี Internet Protocol ที่จะกำหนด Global Address นี้**
  - Protocol นี้ต้อง Run ที่ทั้ง Host และ Router

สรุปบทที่ 20

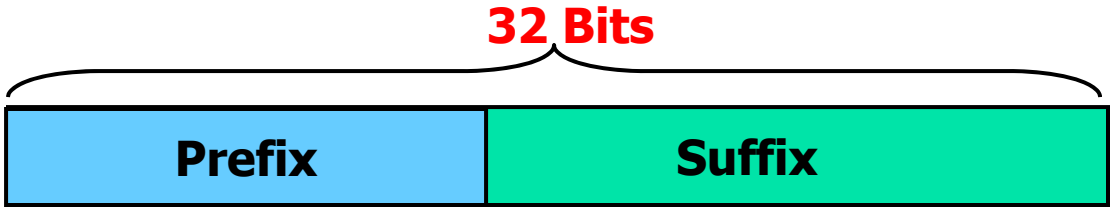
- **Internetworking Protocol ที่นิยมที่สุดคือ IP หรือ Internet Protocol (ใช้ The Internet)**
- **Global Address ที่กำหนดหมายเลข Host คือ IP Address**
- **IP อยู่ใน Layer 3 และต้อง Run ที่ทั้ง Host และ Router**
  - การส่งข้อมูลให้ถึง Host ที่หมาย จะกำหนดด้วย IP Address ปลายทาง แม้ว่าจะมีการส่งข้อมูลผ่านหลาย NW เราไม่จำเป็นต้องรู้ NW Address ของแต่ละ Network ที่ผ่าน เพราะจะมีขบวนการ Mapping โดยอัตโนมัติ (ARP)
  - เรามองเห็นแค่ IP Layer ส่วน Layer ล่างจะ Transparent กับ User เสมือนเราเชื่อมต่อกับ Internet อย่างเดียว
    - เป็น Virtual Network
- **ในการส่งข้อมูลจะแบ่งเป็นสอง Step**
  - 1. ส่ง Packet ไปให้ถึง NW ปลายทางก่อน โดย Router จะดูจากส่วน Network ID ของ IP Address
  - 2. เมื่อ Packet ไปถึง Network ปลายทาง จะเป็นหน้าที่ของ Network Protocol นั้นๆทำการส่งไปให้ถึง Host ใน Network นั้น ในกรณีนี้ ส่วนของ Host ID ใน IP Address จะถูก Map เข้ากับระบบ Address ของ Network นั้น (ใน LAN จะ MAP กับ MAC Address, โดยใช้ ARP)
    - Network Protocol ใน Network นั้นๆจะรับผิดชอบส่ง Packet จาก Router ไปยัง Host ปลายทาง

Chapter 21: IP Addressing

- **IP Address เป็นตัวกำหนด End System (Host) ตลอดทั้งเครือข่าย Internet**
  - ดังนั้นมันจะต้องเป็น Global Address ขณะที่ MAC Address จะหมดอายุเมื่อออกจาก LAN
    - เป็นแค่ Local Address ใน LAN
  - เครื่องสองเครื่องในเครือข่าย จะมีหมายเลขเดียวกันไม่ได้
    - เครื่องที่ต่อใน LAN ออก Internet จะมีทั้ง MAC Address และ IP Address ที่ Match กัน
      - Protocol ARP จะกล่าวในบทที่ 23
- **ปัจจุบันใช้ตามมาตรฐานของ IPv4**
  - Address ขนาด 32 บิต
    - เรียก IP Address, Internet Address หรือ Internet Protocol Address
  - กำหนดหมายเลขเครื่องได้โดยไม่เกี่ยวข้องกับ MAC Address

Ch. 21: 21.4 IP Address Hierarchy

- **แต่ละ IP Address ขนาด 32 บิต จะถูกแบ่งออกเป็นสองส่วน**
  - ส่วนต้น เรียก Prefix เป็นตัวกำหนดหมายเลข Network (Network ID)
  - ส่วนที่เหลือ เรียก Suffix เป็นตัวกำหนดหมายเลข Host ใน Network นั้นๆ
    - หมายเลข Host ใน Network เดียวกันจะซ้ำกันไม่ได้
    - หมายเลข Host ที่อยู่คนละ Network สามารถซ้ำกันได้ เพราะส่วน Prefix (Net ID) นั้นไม่เหมือนกัน



Ch. 21: 21.5 Original Classful IP Addressing

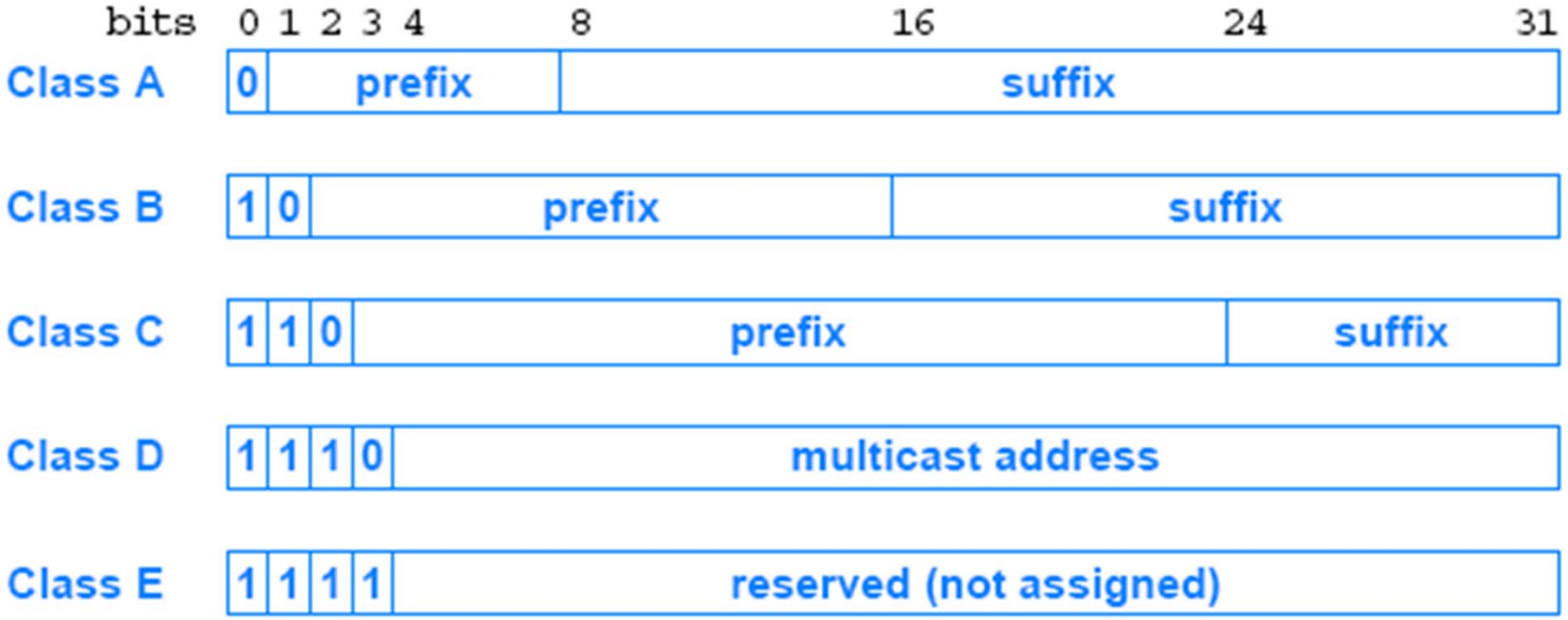


Figure 21.1 The five classes of IP addresses in the original classful scheme.

Ch. 21: 21.6 Dotted Decimal Notation

- **ใน 32 บิต IP Address จะถูกแบ่งเป็นสี่ส่วน ส่วนละ 8 บิต**
  - แต่ละส่วนจะเขียนเป็นเลขฐาน 10 มีค่าได้ระหว่าง 0 – 255
  - แต่ละส่วนจะเขียนต่อกัน ขั้นด้วย “จุด”
- **เรียก Dotted Decimal Notation**
  - Address ต่ำสุดคือ 0.0.0.0
  - Address สูงสุดคือ 255.255.255.255
  - แต่ละ Class สามารถสังเกตุได้จาก Octet แรกของ IP Address

Ch. 21: 21.7 Division of Address Space

- สังเกตุว่า แม้ว่า Class A จะมีแค่ 128 Network แต่มันประกอบด้วยครึ่งหนึ่งของ Address Space ทั้งหมด

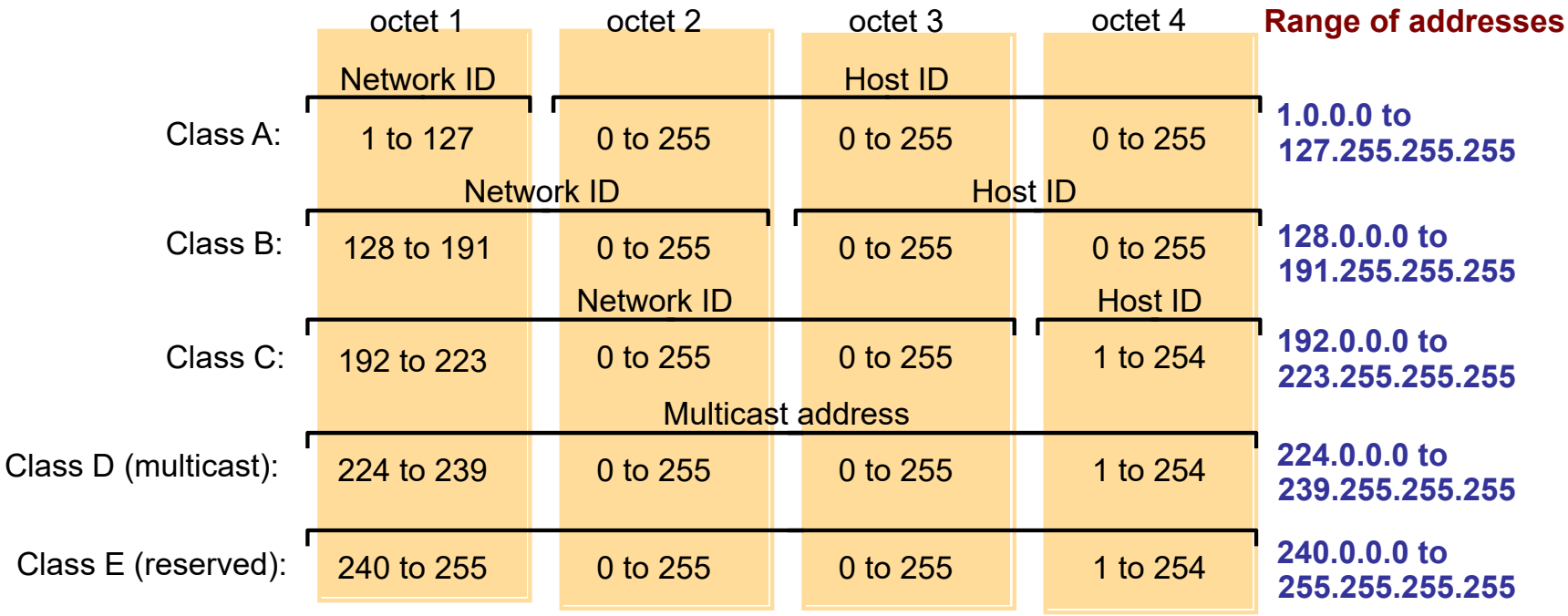
Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Figure 21.3 The number of networks and hosts per network in each of the original three primary IP address classes.

Ch. 21: 21.8 Authority for Address

- **องค์กรที่ดูแลจัดการเรื่อง IP Address คือ ICANN**
  - Internet Corporation for Assigned Names and Numbers
- **ปกติ ICANN จะกำหนดให้ Registrar เป็นผู้จัดสรรในแต่ละภูมิภาคอีกทีหนึ่ง**
- **Registrar จะจัดสรร Block ของ IP Address ให้แก่ ISP แต่ละราย**
  - ผู้ใช้งานจะได้รับ IP Address จาก ISP อีกที

Summary of IP Classful

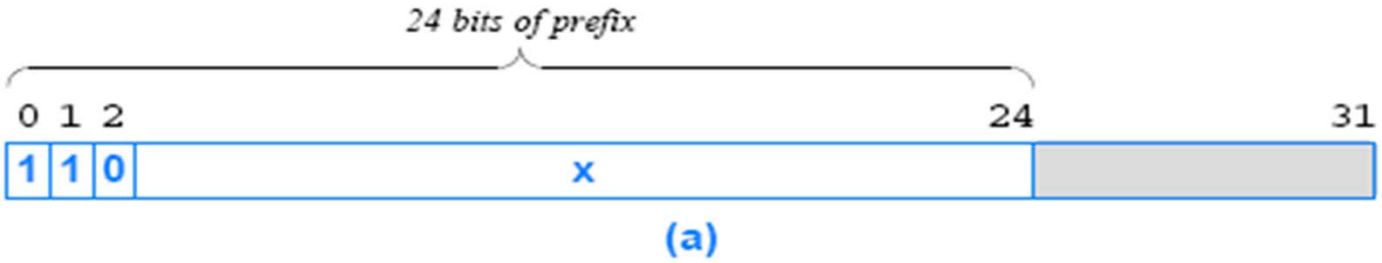


Ch. 21: 21.9 Subnet and Classless Addressing

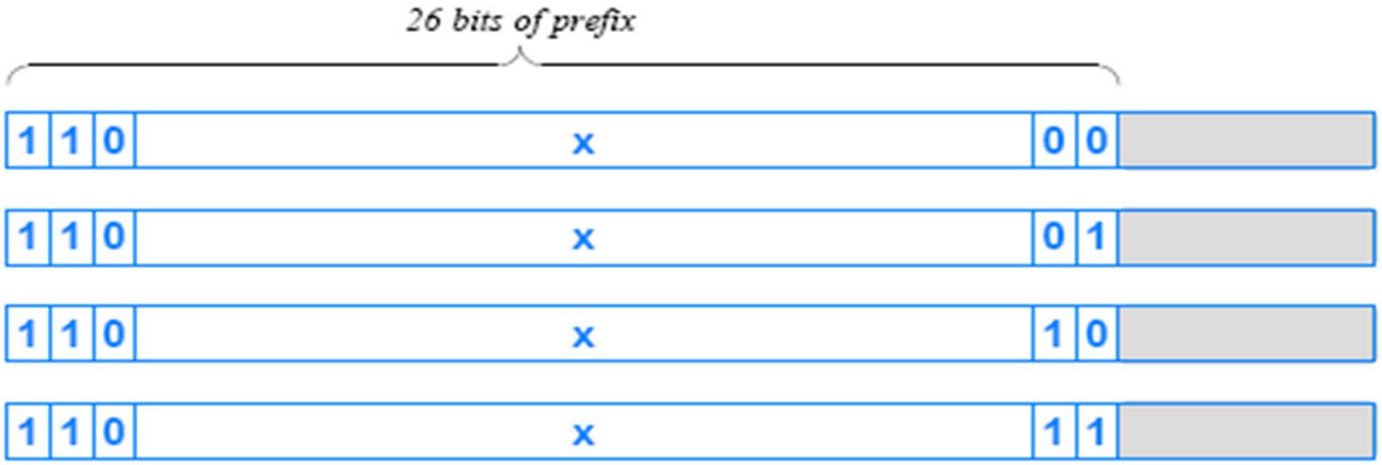
- **การแบ่ง Class ทำให้การใช้งาน IP Address ขาดประสิทธิภาพ**
  - เรามีคอมพิวเตอร์ 15 เครื่อง ดังนั้นต้องใช้ Class C ซึ่งรองรับได้ 256 (-2) เครื่อง แต่ใช้จริงแค่ 15 ที่เหลือ คนอื่นใช้ไม่ได้ เพราะเป็นคนละ Network กับเรา
- **ปัจจุบัน IP Class A และ B ได้ถูกใช้งานหมดแล้ว**
- **เหลือ Class C จำนวนเล็กน้อย และขอได้ยากมาก**
- **เพื่อแก้ปัญหา IP Address ไม่พอใช้ มีสองวิธีที่คิดขึ้นและคล้ายกัน คือ**
  - Subnet Addressing : การแบ่ง Network เป็น NW ย่อย
  - Classless Addressing : การแบ่ง NW โดยกำหนด Prefix เอง
  - กล่าวคือ ไม่มีการกำหนด Class แต่สามารถกำหนดบิตของ Prefix และ Suffix ได้ตามต้องการ

Ch. 21: 21.9 Subnet and Classless Addressing

- ตัวอย่าง Class C เดิม (24 Bit Prefix) แบ่งเป็นสี่ส่วน โดยเพิ่ม Prefix อีก 2 บิต (Suffix ถูกลดลง 2 บิต เหลือ 6 บิต)



(a)



(b)

Ch. 21: 21.10 Address Masks

- **การที่เรากำหนด Prefix เองทำให้ IP Address ไม่มี Class อีกต่อไป**
  - เรียก Classless
- **ในการนี้ เราจะต้องบ่งบอกเองว่าส่วนไหนเป็น Prefix และส่วนใหนเป็น Suffix**
- **กำหนดจาก ค่า 32 Bit เช่นกันเรียก Address Mask (เมื่อก่อนเรียก Subnet Mask)**
  - ส่วนต้น ที่กำหนด Prefix ให้เป็นบิต '1'
    - ต้องเขียนติดต่อกัน
  - ส่วนที่เหลือเป็น Bit '0' กำหนด Suffix เขียนติดกัน
- **การหา Network ID(Prefix) ทำได้โดยทำ Bitwise AND ระหว่าง IP Address กับ Mask**
  - N == (D & M)

Ch. 21: 21.10 Address Masks

Example:

- **Consider the following 32-bit network prefix:**

  10000000 00001010 00000000 00000000 = 128.10.0.0

- **Consider a 32-bit mask:**

  11111111 11111111 00000000 00000000 = 255.255.0.0

- **Consider a 32-bit destination address, which has a**

  10000000 00001010 00000010 00000011 = 128.10.2.3

- **A logical and between the destination address and the address mask extracts the high-order 16-bits**

  10000000 00001010 00000000 00000000 = 128.10.0.0

Ch. 21: 21.11 CIDR Notations

- ปกติ Address Mask เขียนในลักษณะ Dotted Decimal Format เช่นเดียวกันกับ IP Address
  - เช่น 1111 1111 1111 1111 1111 1111 1100 0000 จะเขียนเป็น 255.255.255.192
- เพื่อให้ง่ายในการอ่าน แทนที่เราจะเขียน IP Address คู่กับ Address Mask เราเขียน IP Address ตามด้วยจำนวนบิตที่กำหนด Suffix ในรูป
  - ddd.ddd.ddd.ddd / m
  - เช่น 192.5.48.69 / 26 มีความหมายเดียวกับ IP Address 192.5.48.69 และ Mask 255.255.255.192
- เรียก CIDR Notation
  - Classless Inter-Domain Routing

CIDR

Length (CIDR)	Address Mask	Notes
/0	0 . 0 . 0 . 0	All 0s (equivalent to no mask)
/1	128 . 0 . 0 . 0	
/2	192 . 0 . 0 . 0	
/3	224 . 0 . 0 . 0	
/4	240 . 0 . 0 . 0	
/5	248 . 0 . 0 . 0	
/6	252 . 0 . 0 . 0	
/7	254 . 0 . 0 . 0	
/8	255 . 0 . 0 . 0	Original Class A mask
/9	255 . 128 . 0 . 0	
/10	255 . 192 . 0 . 0	
/11	255 . 224 . 0 . 0	
/12	255 . 240 . 0 . 0	
/13	255 . 248 . 0 . 0	
/14	255 . 252 . 0 . 0	
/15	255 . 254 . 0 . 0	
/16	255 . 255 . 0 . 0	Original Class B mask
/17	255 . 255 . 128 . 0	
/18	255 . 255 . 192 . 0	
/19	255 . 255 . 224 . 0	
/20	255 . 255 . 240 . 0	
/21	255 . 255 . 248 . 0	
/22	255 . 255 . 252 . 0	
/23	255 . 255 . 254 . 0	
/24	255 . 255 . 255 . 0	Original Class C mask
/25	255 . 255 . 255 . 128	
/26	255 . 255 . 255 . 192	
/27	255 . 255 . 255 . 224	
/28	255 . 255 . 255 . 240	
/29	255 . 255 . 255 . 248	
/30	255 . 255 . 255 . 252	
/31	255 . 255 . 255 . 254	
/32	255 . 255 . 255 . 255	All 1s (host specific mask)

Ch. 21: 21.12 CIDR Example

- สมมุติ ISP มี Block ของ IP Address
  - 128.211.0.0 / 16
- ถ้า ISP มีลูกค้าสองราย แต่ละรายต้องการ 12 และ 9 IP Address ซึ่ง ISP สามารถกำหนด Prefix ให้แก่ลูกค้าทั้งสองดังนี้
  - 128.211.0.16 / 28
  - 128.211.0.32 /28
- แม้ว่าลูกค้าทั้งสอง จะใช้ขนาดของ Mask เท่ากัน แต่ Prefix จะต่างกันคือ
  - 10000000 11010011 00000000 0001 0000
  - 10000000 11010011 00000000 0010 0000

Ch. 21: 21.13 CIDR Host Address

- **ในแต่ละละ Block ของ CIDR เราสามารถกำหนด Host Address จากส่วน Suffix**
  - จำนวน Host ในแต่ละ Block เท่ากับ $2^{\text{suffix bit}} - 2$
  - เหตุผลที่ต้องลบสอง คือ Address แรก และ Address สุดท้าย ของ Host จะไม่ใช้
    - Address แรก คือเมื่อ Bit ของ Host เป็นศูนย์ทั้งหมด เนื่องจาก IP Address ของ Host นี้จะเหมือนกับ Network Address และป้องกันปัญหาใน Berkeley Broadcast Address Form
    - Address สุดท้าย คือเมื่อ Bit ของ Host เป็นหนึ่งทั้งหมด จะถูกกันไว้เป็น Address พิเศษสำหรับ Broadcast ใน Network นั้น

Ch. 21: 21.13 CIDR Host Address

- **Example: 28 Bit Prefix**

Ch. 21: 21.14 Special IP Address

- **Network Address**
  - Address ที่ประกอบด้วย Host Bit เป็น '0' ทั้งหมดจะถูก Reserved ไว้สำหรับ Network Address
    - 128.211.0.16/28 หมายถึง Network ไม่ใช่ IP Address
- **Directed Broadcast Address**
  - Address ที่ประกอบด้วย Host Bit เป็น '1' ทั้งหมดจะถูก Reserved ไว้สำหรับส่ง Packet ไปให้กับทุก Host ใน Physical Network นั้น
    - เมื่อ Packet นี้ถูกส่ง จะส่งไป Copy เดียว ผ่าน Internet จนกระทั่งถึง Network ปลายทาง จากนั้น Packet จะถูกส่งต่อไปให้ทุกๆ Host ใน NW ปลายทาง

Ch. 21: 21.14 Special IP Address

- **Limited Broadcast Address**
  - เป็นการ Broadcast กับเครื่องที่ต่อเฉพาะกับ Link นั้น จะใช้ในตอนที่ระบบ Start และยังไม่มี IP Address เป็นของตัวเอง
  - Address คือ 255.255.255.255 (All '1')
  - IP จะ Broadcast Packet ที่ใช้ Address นี้ไปตลอดทั่ว Local Network
- **This Computer Address**
  - กำหนดโดย Address ที่เป็น '0' ทั้งหมด
  - คือ 0.0.0.0
  - ใช้ในการกำหนด Source IP Address สำหรับเครื่องที่ยังไม่รู้ IP Address ของตนเอง
    - ใช้ใน Protocol DHCP เมื่อคอมพิวเตอร์เริ่ม Boot และขอ IP Address จาก DHCP Server

Ch. 21: 21.14 Special IP Address

- **Loopback Address**
  - ใช้สำหรับทดสอบการสื่อสารของสอง Network Application บนเครื่องเดียวกัน โดยไม่ต้องผ่าน Network
  - IP Reserver Network Prefix 127/8 สำหรับใช้กับ Loopback
    - ปกติจะใช้ Host number 1: 127.0.0.1
  - Packet Loopback จะไม่ออกมานอกเครื่อง

Ch. 21: 21.15 Summary Special IP Address

Prefix	Suffix	Type Of Address	Purpose
all-0s	all-0s	this computer	used during bootstrap
network	all-0s	network	identifies a network
network	all-1s	directed broadcast	broadcast on specified net
all-1s	all-1s	limited broadcast	broadcast on local net
127 / 8	any	loopback	testing

Figure 21.7 Summary of the special IP address forms.

Ch. 21: 21.16 The Berkeley Broadcast Address Form

- **ในการพัฒนา TCP/IP บน UNIX ยุคแรกๆ โดย University of California at Berkeley (BSD UNIX) ได้ใช้ Directed Broadcast Address ที่ไม่ได้เป็นมาตรฐาน**
  - กล่าวคือใช้ Host Bit '0' ทั้งหมด กำหนดเป็น Directed Broadcast Address
    - รู้จักกันในนาม 'Berkeley Broadcast'
  - ยังมีอุปกรณ์รุ่นเก่าที่ยังใช้งานอยู่ที่ใช้ Broadcast Address นี้
    - Network Manager จะต้องระวัง และเลือกใช้

Ch. 21: 21.17 Router and IP Address

- **นอกเหนือจากการกำหนด IP Address ให้แก่ Host แล้ว อุปกรณ์ Router ต้องกำหนด IP Address ด้วย**
  - Router ต่อเข้ากับหลาย Physical Network
  - แต่ละ Interface ของ Router ต้องใช้ Prefix ตาม Network ที่มาต่อ
    - แต่ละ Interface ของ Router จะใช้ Host ID (Suffix) ที่ไม่ซ้ำกับ Network ที่มาต่อนั้น

Ch. 21: 21.17 Router and IP Address

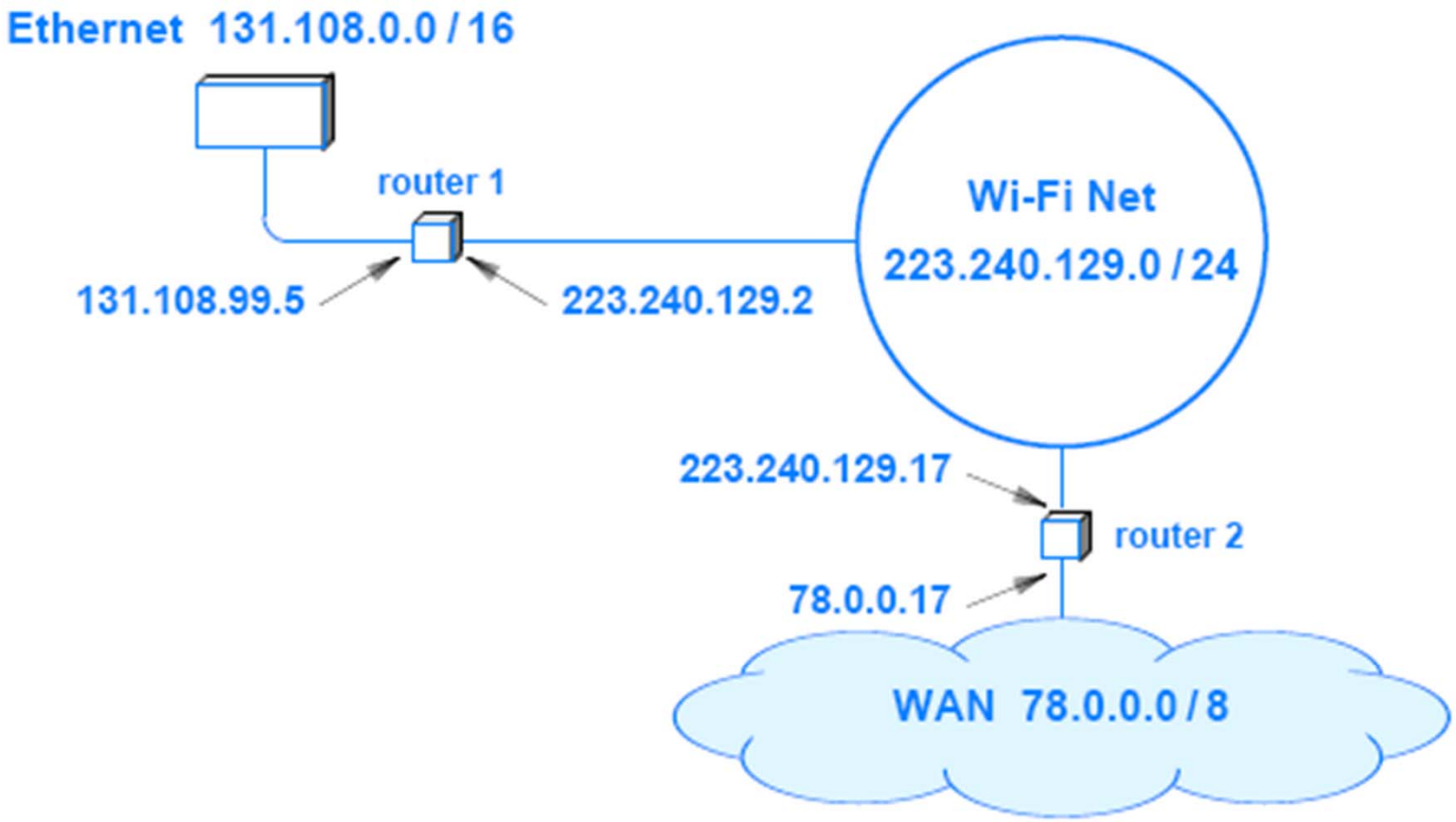
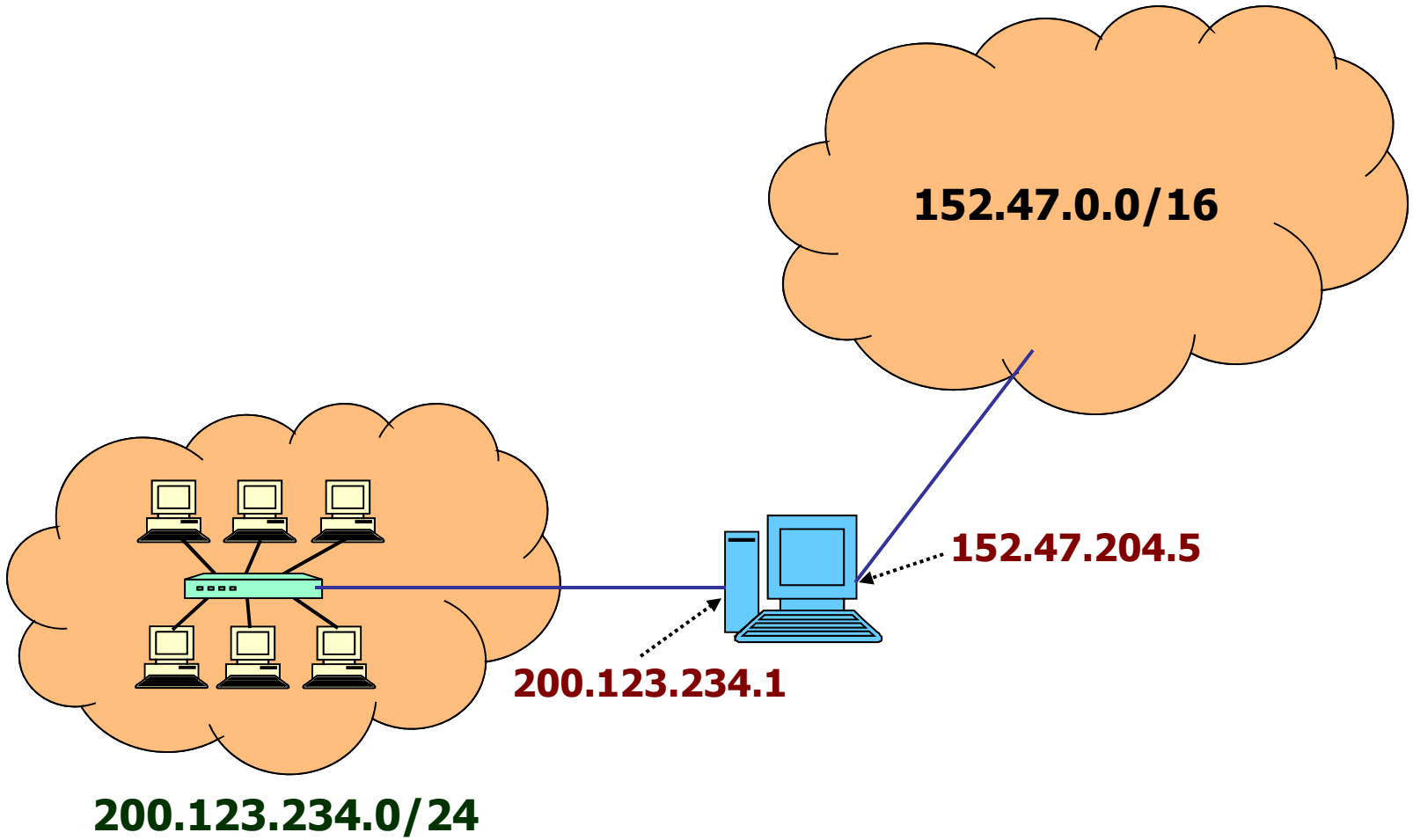


Figure 21.8 An example of IP addresses assigned to two routers.

Ch. 21: 21.18 Multi-Homed Host

- **คอมพิวเตอร์ที่มีมากกว่าหนึ่ง Network Card (LAN Card หรือ Network Card อื่น) สามารถเชื่อมต่อได้มากกว่าหนึ่ง Network และสามารถมี IP Address ได้มากกว่าหนึ่งเบอร์**
  - โดยแต่ละเบอร์มี Prefix ตาม Network ที่มาต่อ
  - เรียก Multi-Homed
  - ปกติจะใช้ในการเพิ่มความเชื่อถือได้ของอุปกรณ์ในกรณีที่ Network หนึ่ง Down
    - ยังเป็นการเพิ่มประสิทธิภาพด้วย
  - สามารถใช้งานเป็น Gateway หรือทำ Firewall หรือทำเป็น Router ได้ ถ้ามี Software ที่ถูกต้อง

Ch. 21: 21.18 Multi-Homed Host



TOPICS

- **Chapter 22: IP Datagram**
  - Header Format: 22.1-22.5
  - Datagram Forwarding: 22.6-22.10
  - Encapsulation: 22.11-22.12
  - MTU and Fragmentation: 22.13-22.17

Chapter 22: IP Datagram

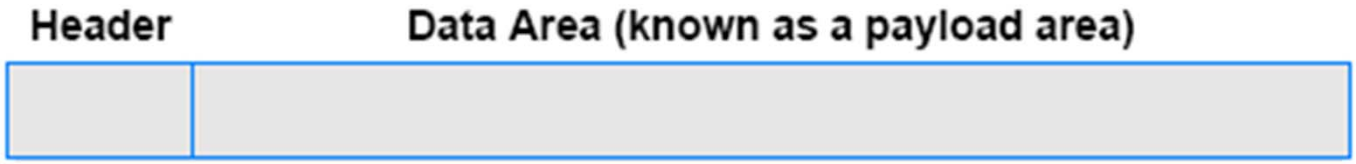
- จุดประสงค์ของ **Internetworking** เพื่อที่จะให้ **Program** ที่ **Run** อยู่บนคอมพิวเตอร์หนึ่ง สามารถส่ง **Packet** สื่อสารได้กับอีก **Program** หนึ่งที่อยู่ต่างคอมพิวเตอร์กัน
- การอกแบบ **Internetworking** ที่ดี การสื่อสารดังกล่าวจะเป็น **Transparent** และไม่ต้องคำนึงถึง **Hardware** เชื่อมต่อที่แตกต่างกัน
- **Internet** ควรจะให้ **Service** แบบใดแก่ **Application Program** เหล่านี้ : **Connectionless** หรือ **Connection-Oriented**
- ผู้ออกแบบ **TCP/IP** ตกลงเลือก **Service** ทั้งสองประเภท
  - Connectionless: User Datagram Protocol(UDP)
  - Connection Oriented: Transport Control Protocol(TCP)
- อย่างไรก็ตาม วิธีการส่ง **Packet** จะใช้แบบ **Connectionless** หรือ **Datagram** โดยถ้าผู้ใช้ต้องการความเชื่อถือในการส่งข้อมูล สามารถผนวกวิธีการของ **Connection Oriented(TCP)** เข้ากับการส่ง **Packet** ที่เป็นแบบ **Datagram (IP)** ได้

Ch22: 22.3 Virtual Packet

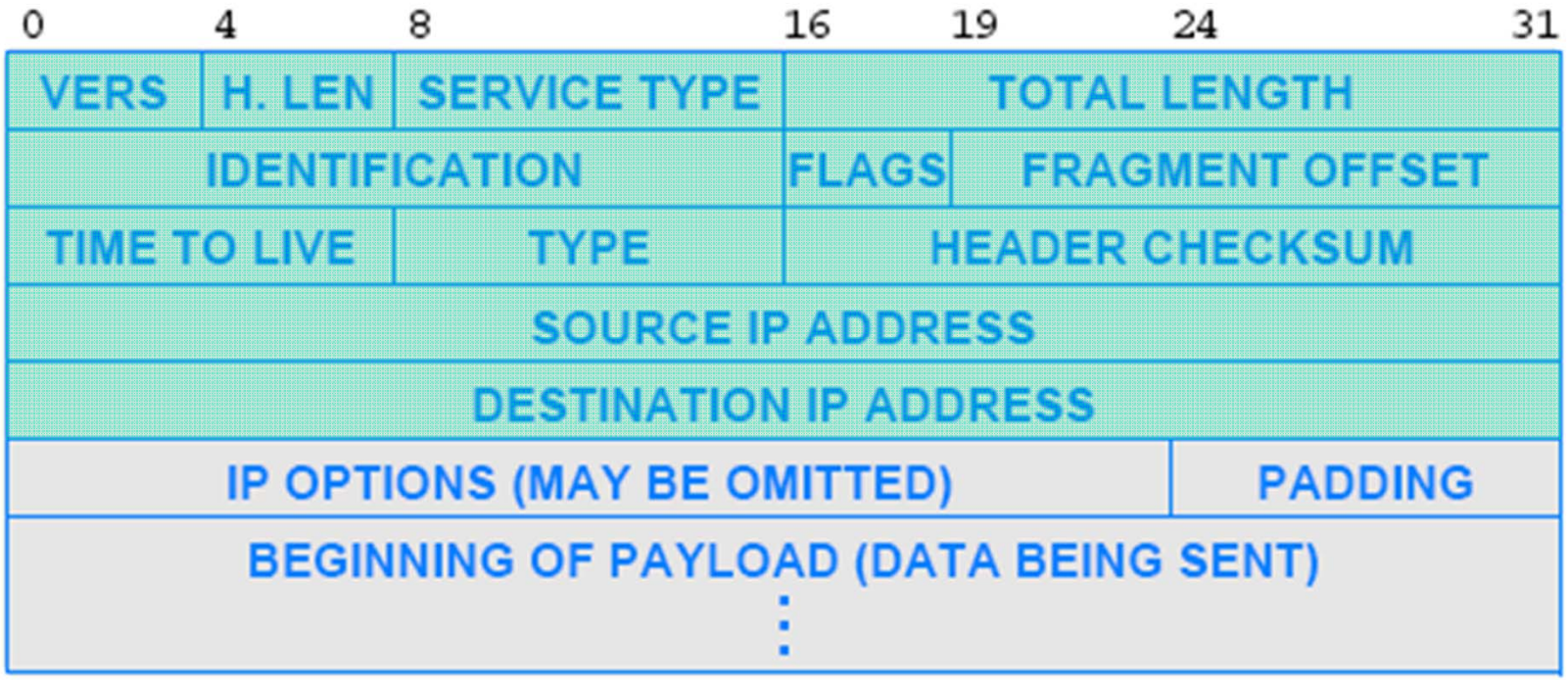
- **การส่งข้อมูลที่เป็นแบบ Datagram หรือ Connectionless ภายใน Network เป็นการปรับปรุงมาจาก Packet Switching Network**
  - โดยผู้ส่งสามารถส่งแต่ละ Packet ผ่าน Internet และแต่ละ Packet จะเดินทางโดยไม่ขึ้นต่อกัน
  - ตัวการที่สำคัญในการส่งผ่าน Packet คือ Router โดยที่ Router จะดู Address ปลายทางและตัดสินใจว่าควรจะส่ง Packet ให้ Router ตัวถัดไปตัวใด
    - Router รับผิดชอบเพียงเท่านี้ คือส่ง Packet ให้ Router ตัวถัดไป (หรือส่งไปให้ Destination ถ้าเป็น Router ตัวสุดท้ายของเส้นทาง)
  - เนื่องจาก Router อาจจะเชื่อมต่อ Network ที่ต่างกันและ Address ของแต่ละ Network ต่างกัน ซึ่ง Router ไม่สามารถที่จะเปลี่ยน Address Format ได้ทุกครั้งที่ผ่านแต่ละ Network เพราะผู้รับปลายทางอาจจะใช้ Address Format ที่ต่างออกไปอีก
    - ในการนี้ Internet จะต้องออกแบบ Packet Format ใหม่ และใช้ Address ที่เป็นหนึ่งเดียว (คือ IP Address) กล่าวคือ Internet Protocol จะต้องวางบน Network Layer ของแต่ละ Network อีกทีหนึ่ง
    - ผลลัพธ์คือ Universal Virtual Packet ที่สามารถส่งผ่าน Network อะไรก็ได้

Ch22: 22.4 IP Datagram

- ใน **TCP/IP** จะใช้คำว่า **IP Datagram** ในการอ้างถึง **Packet** ของ **Internet** ซึ่งประกอบด้วย **Payload** ที่ได้มาจาก **Layer** บน และส่วนหัวที่เรียก **IP Header**
- ขนาดของ **Payload** ไม่ได้กำหนดตายตัว ขึ้นอยู่กับ **Application** ที่ใช้
  - IPv4 สามารถมี Payload ได้ตั้งแต่ 1 Octet จนถึงสูงสุด 64K Octet(65535: รวมส่วนหัวด้วย)
  - ขนาดของ Header ปกติจะคงที่คือ 20 Octet แต่สามารถขยายได้ (ส่วนของ Option)

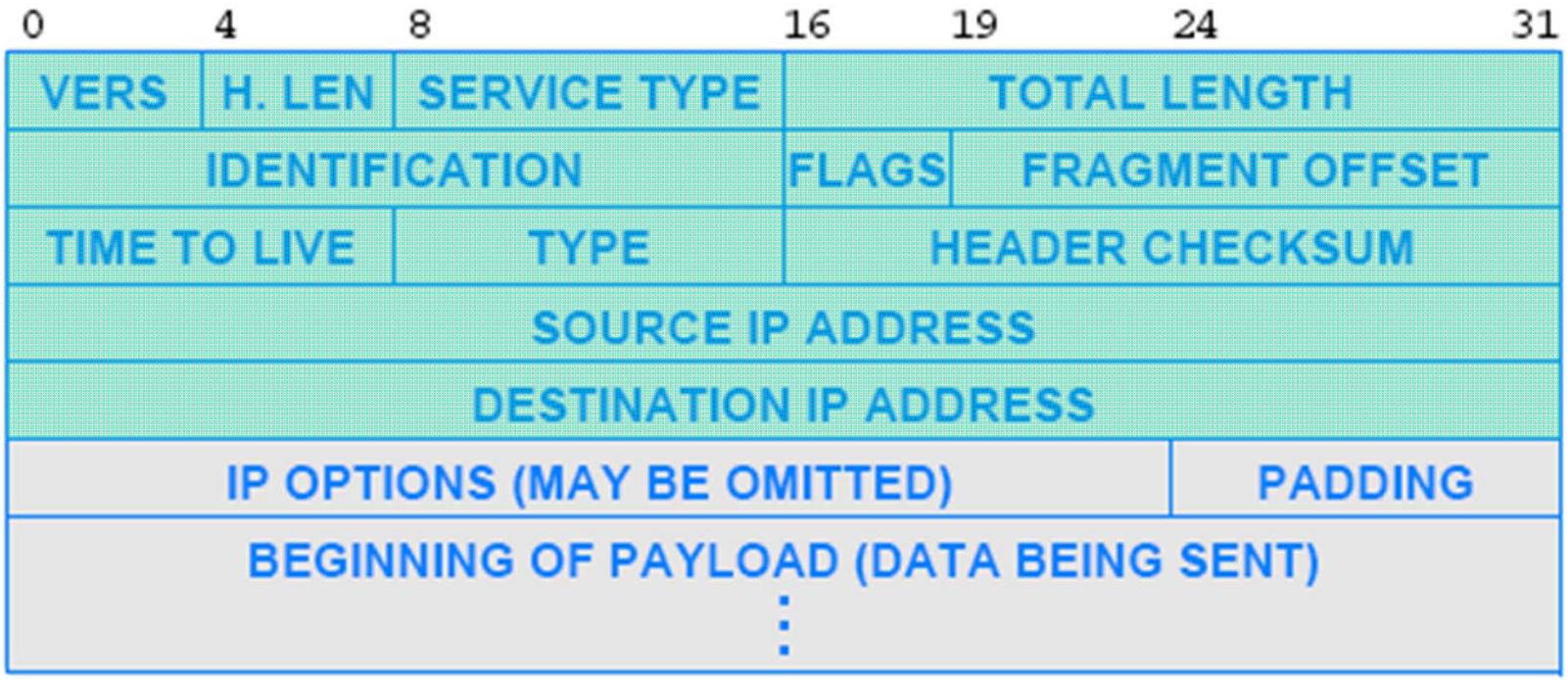


Ch22: 22.5 IP Datagram Header Format (IPv4)



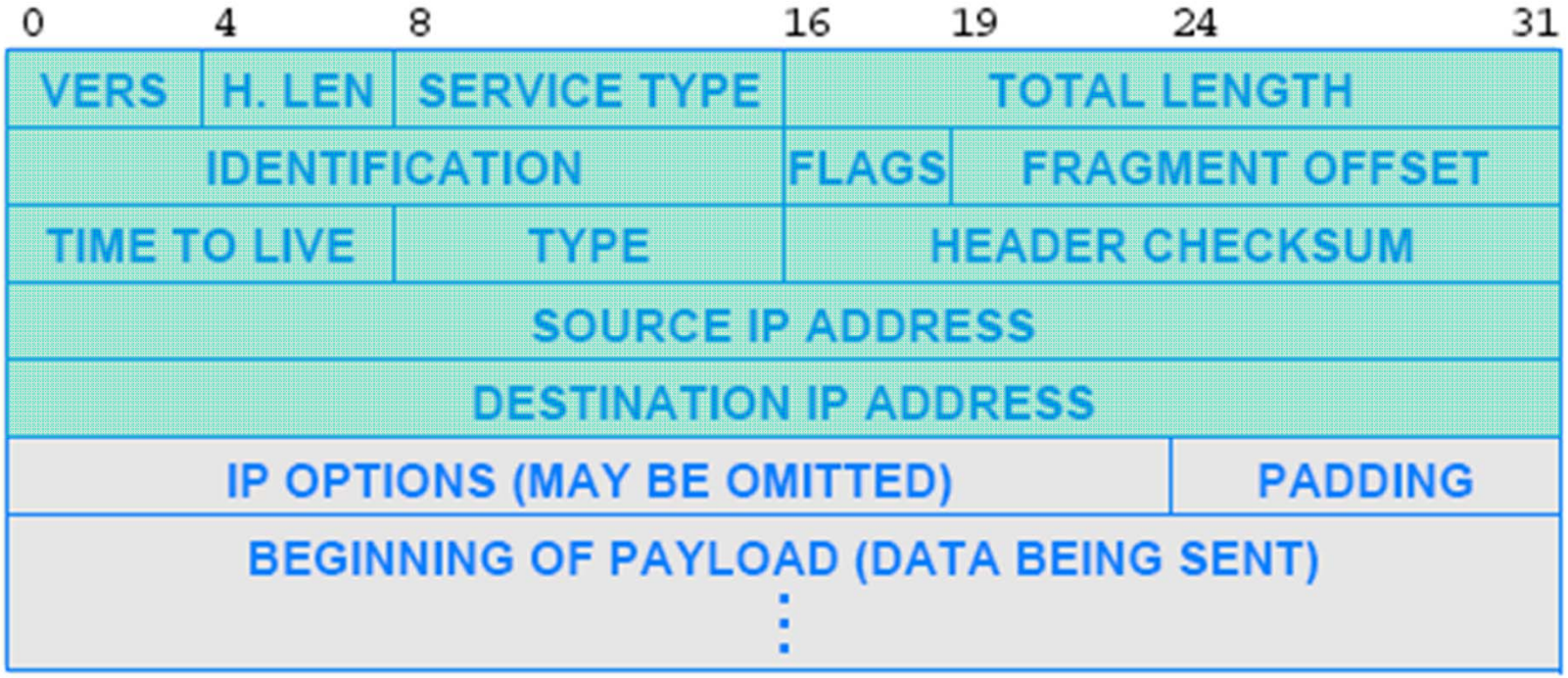
- **VERS.** กำหนด **Version** ในที่นี้คือ **0100 B**
- **H.LEN.** กำหนดความยาวของ **Header** มีหน่วยเป็น **32 Bit Word** ถ้าไม่มี **Option** ค่านี้คือ **0101 B**

Ch22: 22.5 IP Datagram Header Format (IPv4)



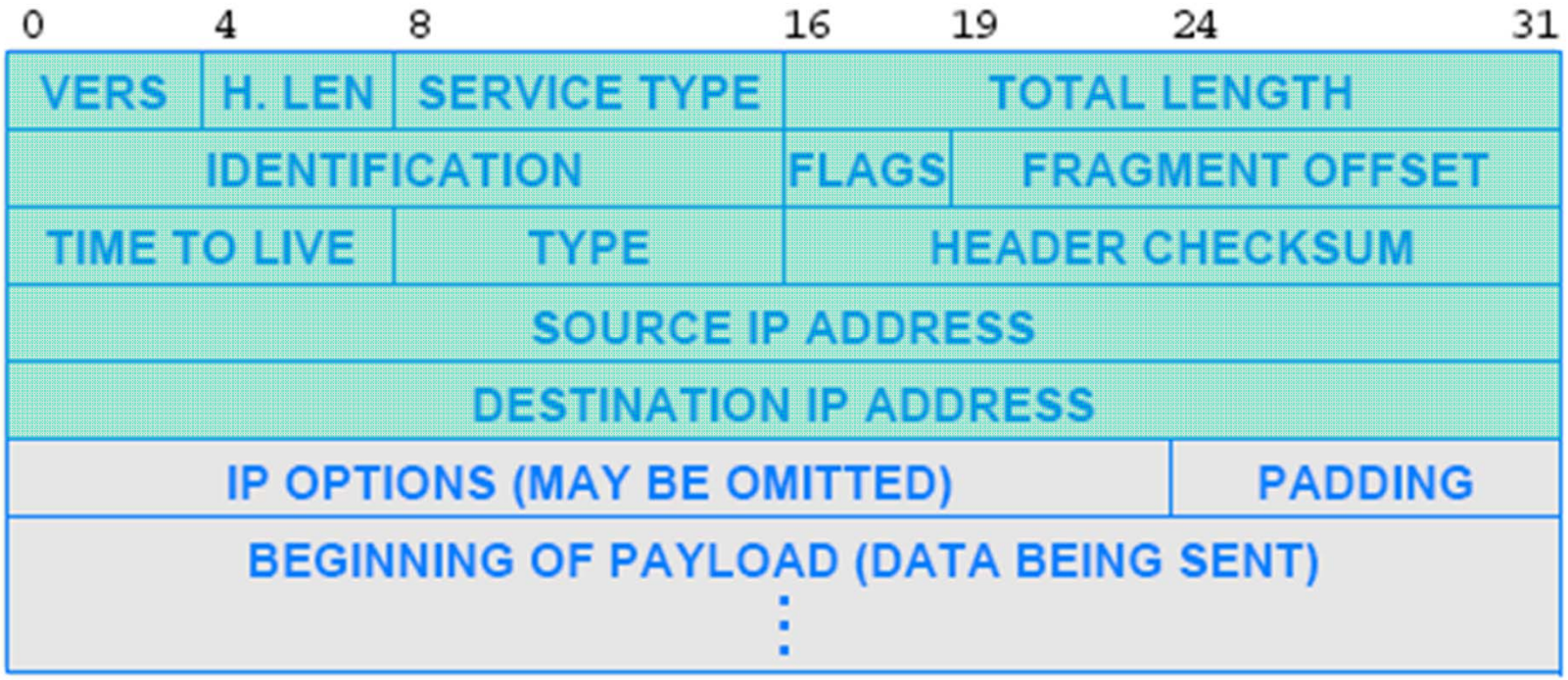
- **SERVICE TYPE.** กำหนด Class ของ Service ปกติไม่ใช้ (จะพูดในเรื่อง QoS)
- **TOTAL LENGTH.** ความยาวเป็น Byte รวมทั้งส่วนหัว

Ch22: 22.5 IP Datagram Header Format (IPv4)



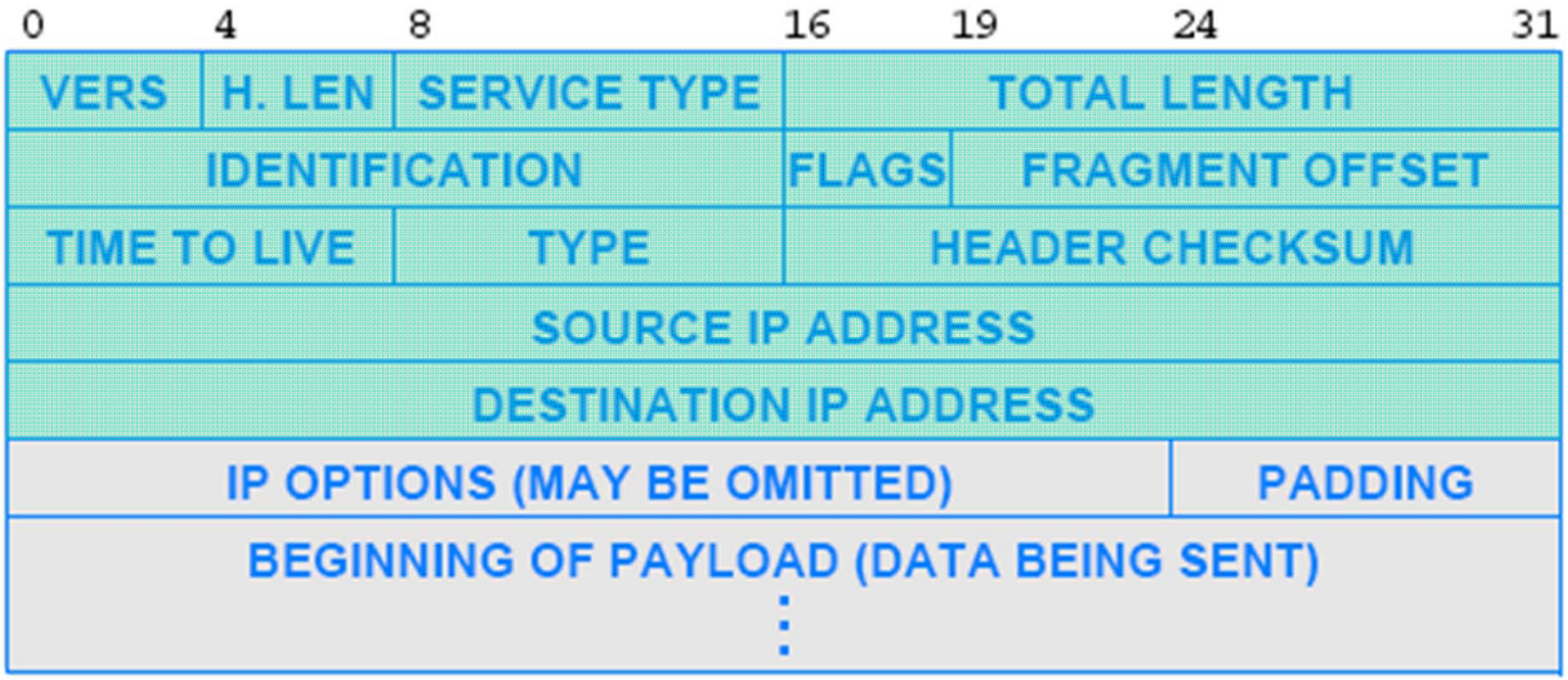
- **IDENTIFICATION.** กำหนด **Sequence Number** ใช้ในกรณีที่มีการ **Fragmentation**
- **FLAGS.** เป็นตัวกำหนดว่าจะยอมให้ทำ **Fragment** ได้หรือไม่ และกำหนด**Fragment** สุดท้าย

Ch22: 22.5 IP Datagram Header Format (IPv4)



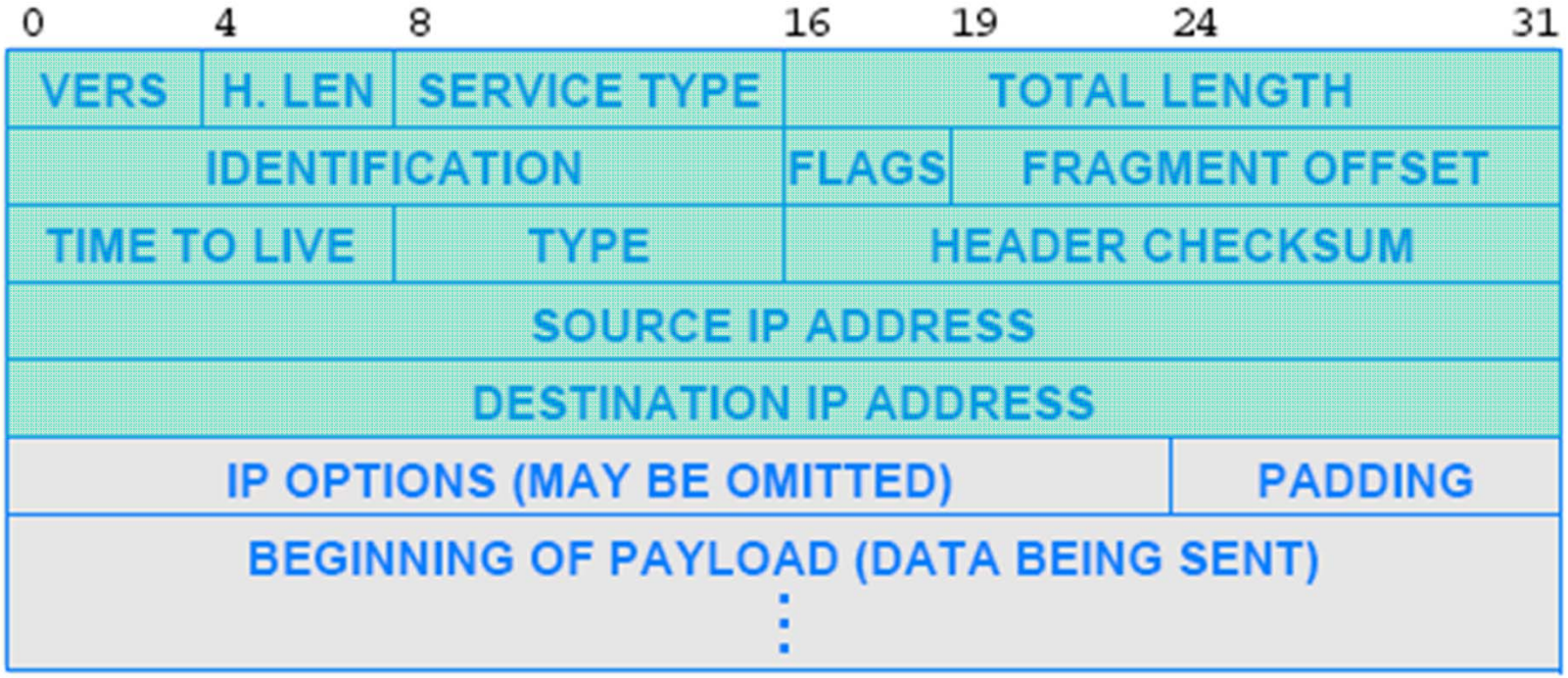
- **FRAGMENT OFFSET.** ค่าจะถูกคูณด้วย 8 เพื่อบ่งบ่งตำแหน่งของ Datagramนี้ใน Datagram ก่อน Fragment
- **TIME TO LIVE.** กำหนดโดยผู้ส่ง โดยจะเป็นตัวกำจัด Packet ในกรณีที่มี Loop ซึ่งค่านี้จะถูกลดลงหนึ่งทุกครั้งที่ผ่าน Router เมื่อค่านี้เป็นศูนย์ Datagram นี้จะถูกโยนทิ้ง และมีการส่ง Error Message กลับไปยังผู้ส่ง

Ch22: 22.5 IP Datagram Header Format (IPv4)



- **TYPE. กำหนดชนิด(Protocol) ของ Payload ที่อยู่ข้างใน (ที่อยู่ Layer บนถัดไป)**
- **HEADER CHECKSUM. ผลบวกของ 16 Bit Word และ Complement**

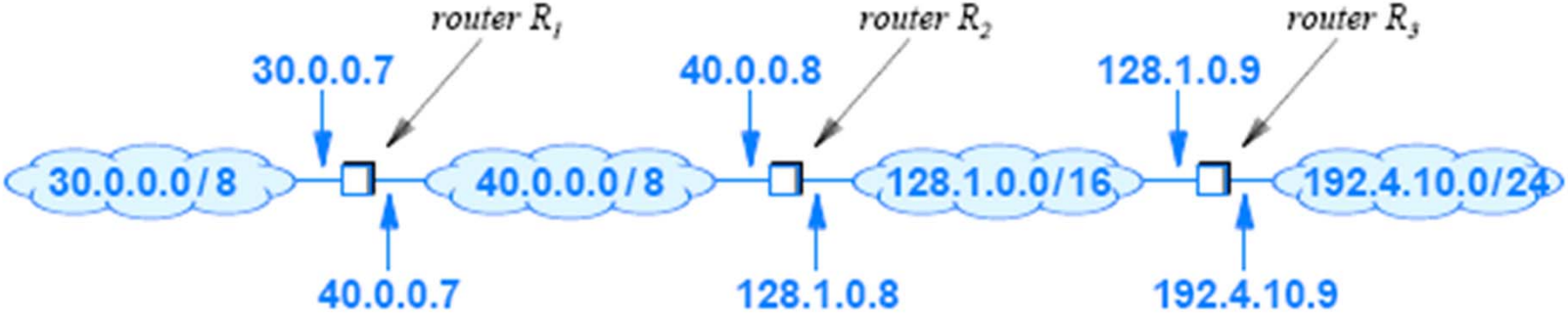
Ch22: 22.5 IP Datagram Header Format (IPv4)



- **SOURCE & DESTINATION IP ADDRESS.** กำหนดเครื่องต้นทางและปลายทาง ไม่ใช่ **Address** ของ **Router**
- **IP OPTIONS.** ใช้ในการควบคุมการทำ **Routing** ปกติจะไม่ใช้
- **PADDING.** ด้วยศูนย์เพื่อให้ครบ **32 Bit Word**

Ch22: 22.6 IP Datagram Forwarding

- **IP Router จะใช้ Forwarding Table (Routing Table) ซึ่งสามารถจะ Update ได้เป็นระยะ**
  - จะเป็นลักษณะของ Next-hop Forwarding



(a)

Router 2

Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

(b)

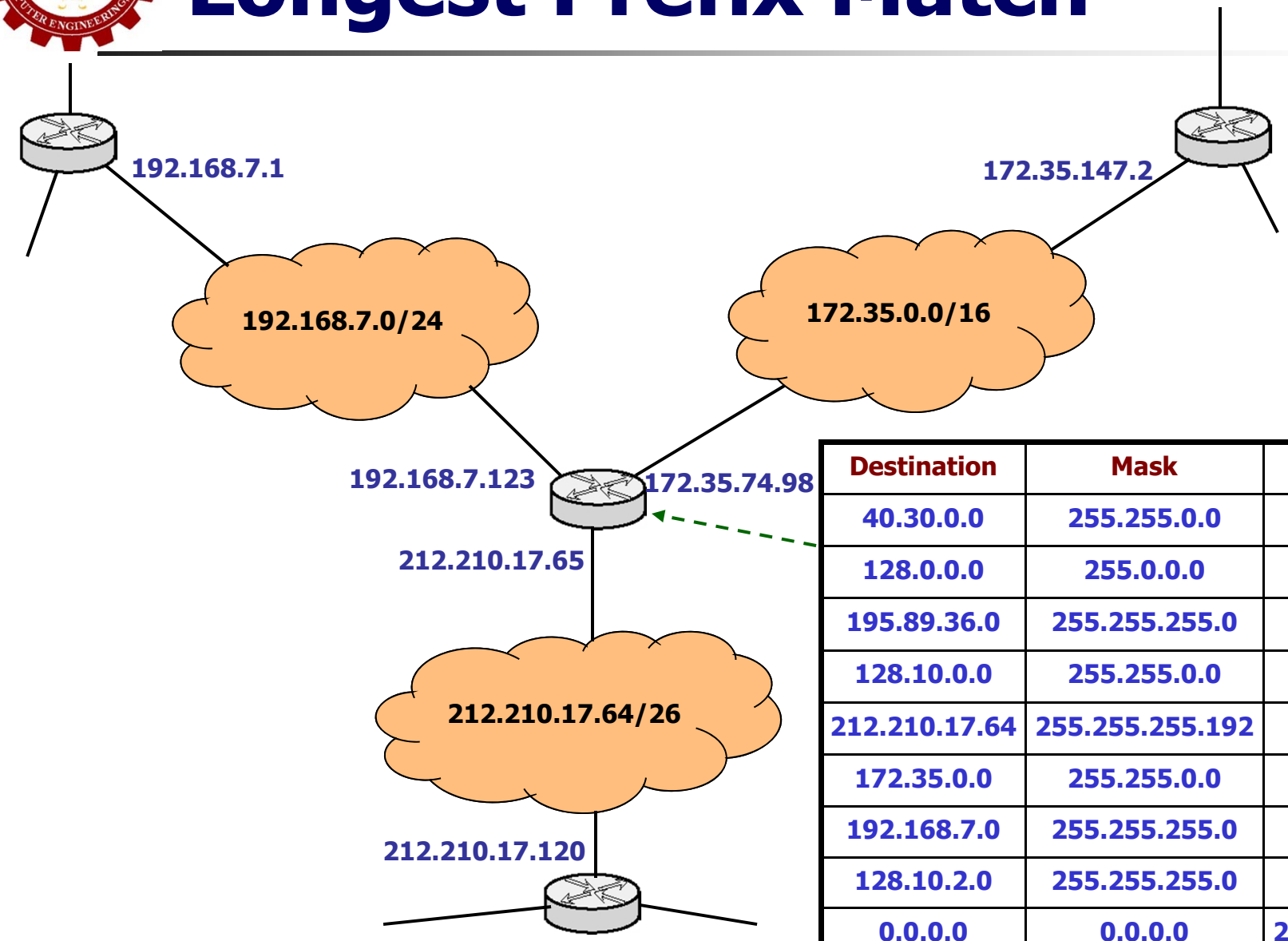
Ch22: 22.7 Network Prefix Extraction

- **จากที่กล่าวแล้ว Router จะดูแค่ส่วน Prefix ของ IP Address(NW ID) จากนั้นจะเปรียบเทียบกับตาราง Routing Table เพื่อหาเส้นทางในการส่งข้อมูลต่อไป ดังนี้**
  - ในแต่ละแถวของตาราง Routing Table ตัว Router จะนำ Mask จากแถวนั้นมาทำการ Bitwise AND กับ IP Address ปลายทางที่ได้จากส่วน Header ของ IP Datagram และเปรียบเทียบกับค่าในตาราง ถ้าค่า Prefix ที่ได้ตรงกันกับค่าจากแถวนั้นในตาราง ถือว่า Match และข้อมูลจะถูกส่งไปในทิศทางนั้น
    - ยกตัวอย่าง มี Datagram ต้องการส่งไปที่ 192.4.10.3 ส่งมาที่ Router R2
      - ในแถวแรกของตาราง เมื่อทำการ AND เราได้ 255.0.0.0 & 192.4.10.3 = 192.0.0.0 ซึ่งไม่ตรงกับ 30.0.0.0
      - แถวที่สองจะได้ 192.0.0.0 เช่นกันและไม่ตรงกับ 40.0.0.0
      - แถวที่สามจะได้ 192.4.0.0 และไม่ตรงกับ 128.1.0.0
      - แถวที่สี่จะได้ 192.4.10.0 ซึ่งตรงหรือ Match ดังนั้นข้อมูลจุถูกส่งออกไปที่ 128.1.0.9 คือ IP Address ของ Interface ของ R3 ที่ต่อกัน เรียกว่า Next Hop Address
    - การส่งให้ Router ตัวถัดไปนั้น จะไม่มีการเปลี่ยนแปลง IP Address ของ Datagram แต่จะส่งโดยใช้ Local Network Address แทน เนื่องจาก Interface ของ Router ที่ต่อตัวถัดไป จะอยู่ใน NW เดียวกันกับ Router ตัวที่ส่ง
    - ถ้าเป็น Direct Connect ตัว Router สามารถส่งข้อมูลให้กับ Host ได้โดยตรง ไม่ต้องส่งให้กับ Router ตัวถัดไป

Ch22: 22.8 Longest Prefix Match

- **ในทางปฏิบัติ ตาราง Routing Table อาจจะมีขนาดใหญ่ และ Routing Algorithm ค่อนข้างสลับซับซ้อน การส่งผ่านข้อมูลจะมีข้อกำหนดเพิ่มเติมที่สำคัญดังนี้**
  - ถ้าไม่มี Prefix ใด Match เลยในตาราง ข้อมูลจะส่งไม่ได้ ดังนั้นแถวสุดท้ายมักจะกำหนด 'Default Route' โดยใช้ Address 0.0.0.0 และ Mask เป็น 0.0.0.0 (Always Match)
  - ผู้ส่งสามารถกำหนด Route เองได้ โดยกำหนดเส้นทางส่งข้อมูลผ่าน Interface ต่างๆในส่วน Option ของ Header
  - ในกรณีที่มีการ Match Prefix มากกว่าหนึ่งอัน ตัว Router จะส่งข้อมูลในทิศทางที่มีการ Match Prefix ที่ยาวที่สุด
    - ดังนั้นในตาราง Routing Table, Software จะทำการเรียงตารางใหม่เริ่มจากแถวที่มี Prefix มากที่สุดก่อน
    - เช่นในตารางมีสอง Prefix คือ 128.10.0.0/16 และ 128.10.2.0/24 ถ้ามี Datagram ที่ต้องการส่งไปที่ 128.10.2.3 เข้ามา มันจะถูกส่งไปตามแถวที่กำหนดจาก 128.10.2.0/24 เพราะมีการ Match ยาวกว่า

Longest Prefix Match



Destination	Mask	Next Hop
40.30.0.0	255.255.0.0	172.35.147.2
128.0.0.0	255.0.0.0	192.168.7.1
195.89.36.0	255.255.255.0	192.168.7.1
128.10.0.0	255.255.0.0	172.35.147.2
212.210.17.64	255.255.255.192	Directed
172.35.0.0	255.255.0.0	Directed
192.168.7.0	255.255.255.0	Directed
128.10.2.0	255.255.255.0	192.168.7.1
0.0.0.0	0.0.0.0	212.210.17.120

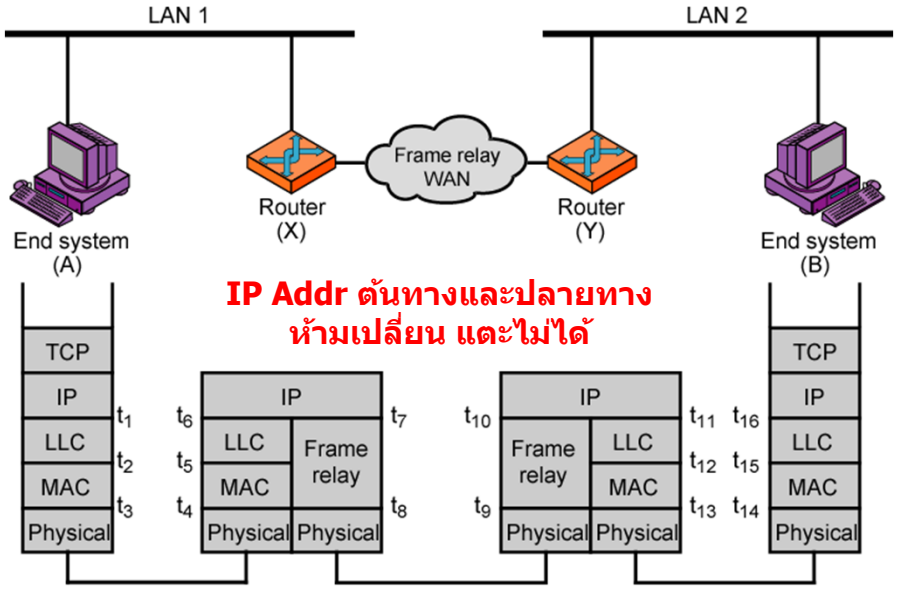
Ch22: 22.10 Best Effort Delivery

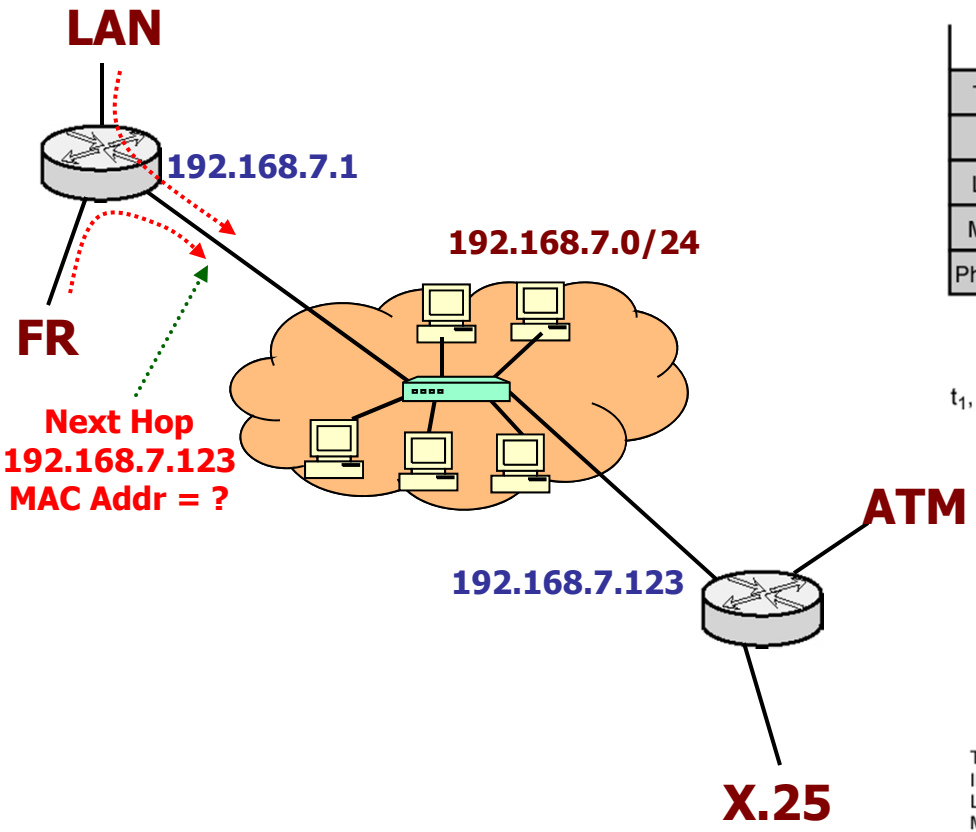
- **การส่ง Datagram ใน IP Network จะเป็นลักษณะ Best-Effort Delivery คือส่งเท่าที่จะสามารถส่งได้ อย่างไรก็ตามสิ่งต่างๆเหล่านี้สามารถเกิดขึ้นได้**
  - Datagram Duplication
  - Delayed หรือ Out-of-Order Delivery
  - Data Corruption
  - Datagram Loss
- **กล่าวคือ จะมี Error เกิดขึ้นได้เสมอในการส่งข้อมูลผ่าน Internet**
- **ในการนี้ มาตรฐาน TCP/IP ได้ออกแบบ Protocol อีก Layer หนึ่งเพื่อที่จะมาจัดการกับเรื่องเหล่านี้ (TCP)**
  - ขึ้นอยู่กับผู้ใช้งาน จะเลือกแค่ Best-Effort Delivery หรือต้องการ Protocol เสริมที่จะช่วยจัดการกับ Error
    - UDP/IP หรือ TCP/IP

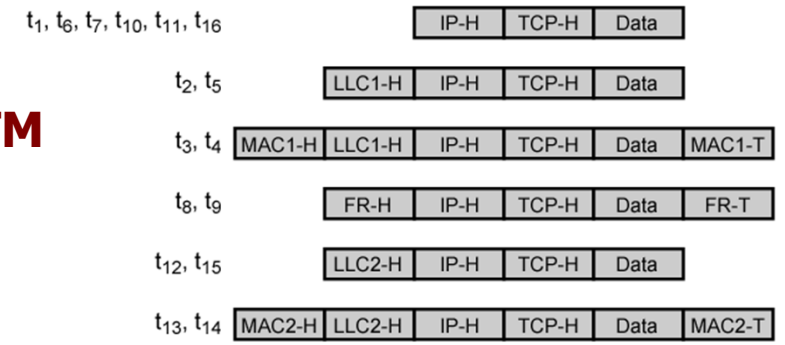
Ch22: 22.11 IP Encapsulation

- **ในการส่ง Datagram ผ่านแต่ละ Network จำเป็นที่จะต้องบรรจุ Datagram ลงใน Network Protocol ที่กำลังผ่าน**
  - เรียกว่าการทำ Encapsulation
- **เมื่อ Encapsulated Datagram ผ่านจาก Network หนึ่งไปยังอีก Network หนึ่ง (ผ่าน Router) มันจะถูก Decapsulated และถูก Encapsulate ใหม่ตาม Protocol ของ Network ถัดไป**
  - การ Decapsulate นั้น จะต้องรู้ว่ามันเป็น IP Datagram Encapsulate มิฉะนั้นแล้วการทำงานจะผิดพลาด เนื่องจากแต่ละ Network อาจจะมีการ Encapsulate หลาย Protocol
  - ดังนั้นที่ส่วนหัวของ Network Protocol ที่ทำการ Encapsulate จะต้องมี Code กำกับบอกชนิดของ Payload
    - ใน Ethernet Frame ถ้าเป็น IP Datagram บรรจุอยู่ใน MAC Frame จะใช้ Ethertype 0x0800
  - นอกจากนี้แล้ว การ Encapsulate ยังต้องการการกำหนด Network Address ของ Router ที่ Datagram จะถูกส่งออกไปยัง NW ถัดไป (หรือ NW Address ของ Host ในกรณีที่ Datagram ไปถึงที่หมาย)
    - อย่าลืมว่า Router กำหนดแค่ IP Address ของ Next-Hop Router ไม่ได้กำหนด Network Address ของ Next-Hop Router
    - เราจะใช้ Protocol ชื่อ ARP มาทำการหา Network Address จาก IP Address ของ Next-Hop Router (Interface)
    - กรณีของ Ethernet Network, ARP จะใช้ในการหา MAC Address จาก IP Address ที่กำหนด รายละเอียดจะอยู่ในบทถัดไป

Ch22: 22.11 IP Encapsulation







TCP-H = TCP header
IP-H = IP header
LLCi-H = LLC header
MACi-H = MAC header
MACi-T = MAC trailer
FR-H = Frame relay header
FR-T = Frame relay trailer

Ch22: 22.12 Transmission Across An Internet

- การส่ง Datagram ใน Internet จะถูก Encapsulate-Decapsulate เป็นทอดๆ ตาม Protocol ในแต่ละ Network ที่ผ่าน
  - มีการกำหนด Code ของ Payload ที่ส่วนหัวของแต่ละ Network Protocol
  - มีการหา Network Address ของ Router ตัวถัดไป ที่ตรงกับ IP Address โดยใช้ ARP

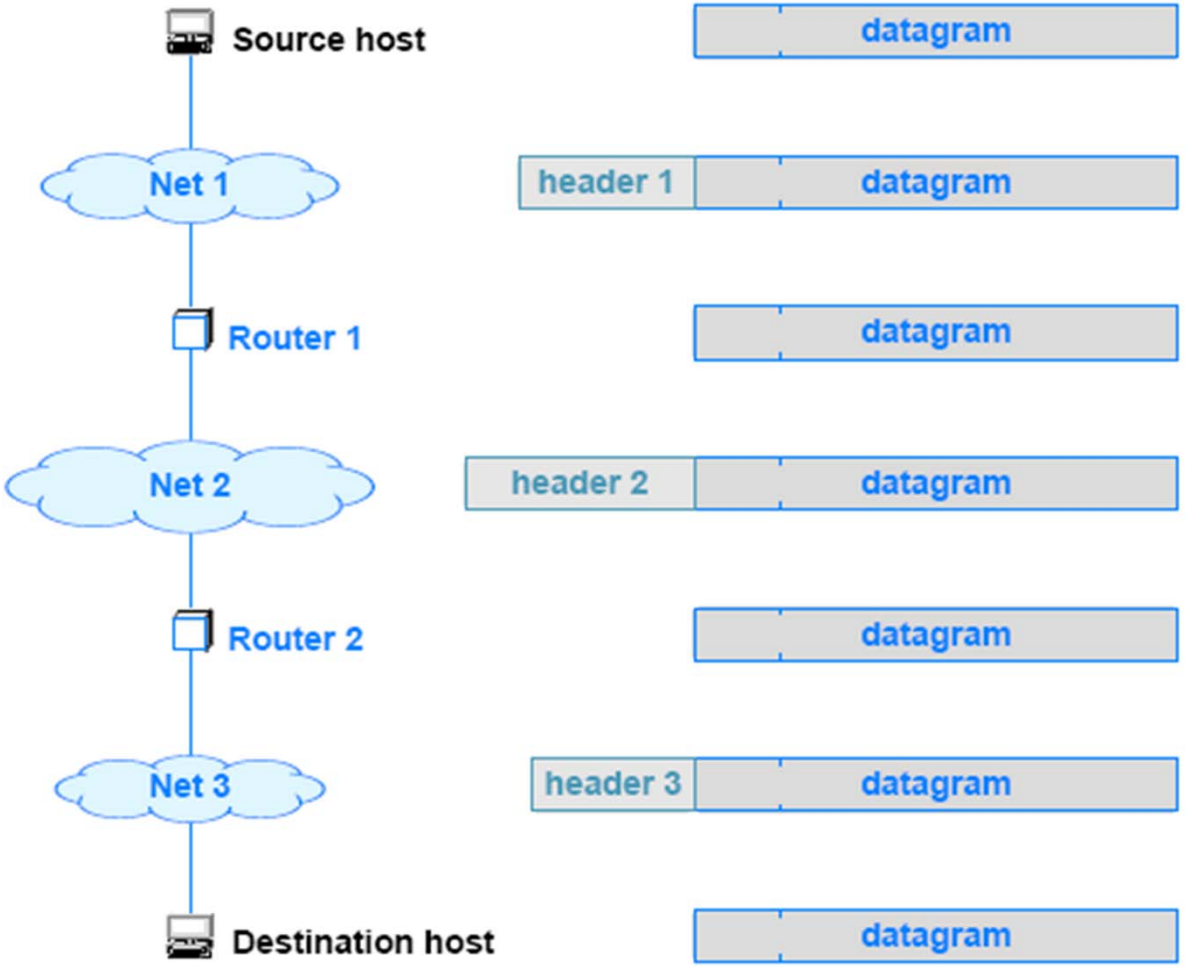


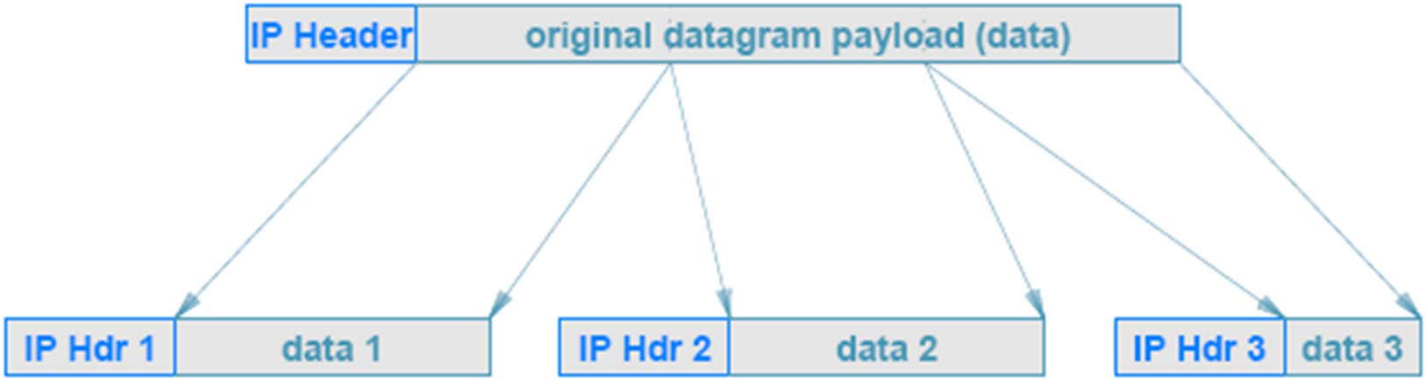
Figure 22.5 An IP datagram as it travels across the Internet.

Ch22: 22.13-14 MTU and Datagram Fragmentation/Reassembly

- **ขนาดของ Data ที่จะใส่ใน Frame สูงสุดที่สามารถส่งผ่านได้เรียก MTU: Maximum Transfer Unit**
  - แต่ละ Network จะกำหนดค่า MTU ต่างกัน
  - เป็นไปได้ว่า Datagram ที่ส่ง อาจจะมีขนาดใหญ่กว่าค่า MTU ของ NW นั้น
    - จะต้องมีการแตก Datagram เป็นส่วนสั้นๆ เรียก Fragment เพื่อจะสามารถส่งผ่านไปได้ ขบวนการนี้เรียก Fragmentation

Figure 22.6 Illustration of a router that connects two networks with different MTUs.

Ch22: 22.13-14 MTU and Datagram Fragmentation/Reassembly



- **ในแต่ละ Fragment ของ Datagram จะมี Format เหมือน Datagram ปกติ ส่วน Header ของ Fragment จะกำหนด IP ต้นทางและปลายทางเหมือน Header เดิม ในส่วน Field 'FLAG' จะบ่งบอกว่านี่คือ Fragment ไม่ใช่ IP Datagram เต็มๆ และส่วนใน 'FRAGMENT OFFSET' จะบ่งบอกว่า Fragment นี้อยู่ที่ส่วนใหนของ Datagram เดิม**
  - การประกอบคืน (Reassembly) กำหนดให้กระทำที่ Host ปลายทาง
  - เพราะ Host ปลายทางสามารถรู้ได้ว่า Fragment ได้ครบหรือไม่
  - ปกติ Router จะไม่สนใจว่า Datagram ที่ส่งเป็น Fragment หรือไม่

IP Header Flags

- **จากที่กล่าวมาแล้ว ส่วนของ Header จะประกอบด้วย FLAG ขนาด 3 บิต ใช้ในกรณีที่ทำ Fragment ดังนี้**
  - Bit 0: จะถูก Reserve และจะต้องมีค่าเป็น 0
  - Bit 1: Don't Fragment(DF) ถ้ามีการ Set จะไม่มีการทำ Fragment ที่ Datagram นี้
    - เมื่อ Datagram ต้องผ่าน NW ที่กำหนดค่า MTU น้อยกว่าขนาดของ Datagram จะส่งผ่านไม่ได้ และจะต้องโยนทิ้ง
  - Bit 2: More Fragment(MF) จะ Set ถ้า Datagram นี้เป็น Fragment ยกเว้น Fragment สุดท้าย
    - ถ้าไม่มีการทำ Fragment ส่วน Bit 2 นี้จะไม่ถูก Set เช่นกัน ซึ่งมีความหมายเดียวกับเป็น Fragment สุดท้าย

Ch22: 22.15 Collecting The Datagram Fragments

- **IP จะไม่ Guarantee การส่งข้อมูล**
  - Fragment อาจหาย มี Error หรือถูกส่งไปในทิศทางที่ต่างกัน และมาถึงปลายทางอย่างไม่เป็นลำดับ
- **เราจะรู้ได้อย่างไรว่า Fragment ของแต่ละ Datagram ได้รับครบแล้ว พร้อมจะประกอบเป็น Datagram เดิมคืนมา**
  - Bit ในส่วนของ 'FLAG' Field จะบอกว่านี่เป็น Fragment ให้รอ Fragment อื่นจนครบ และประกอบเป็น Datagram เดิมก่อนที่จะส่งให้ Layer บนต่อไป
  - ปลายทางจะใช้ส่วนของ IDENTIFICATION Field รวมกับ Source IP Address เป็นตัวกำหนดว่า Fragment ที่ได้อยู่ใน Datagram เดิมอะไร โดยที่ Fragment ของ Datagram เดียวกันจะมีค่า 'IDENTIFICATION' เหมือนกัน ส่วน Source IP Address จะบ่งบอกอีกทีให้แน่ใจ เพราะแต่ละผู้ส่งอาจจะใช้ Identification เหมือนกัน
  - ส่วนของ 'FRAGMENT OFFSET' Field จะบ่งบอกว่า Fragment นี้อยู่ส่วนไหนของ Datagram เดิม

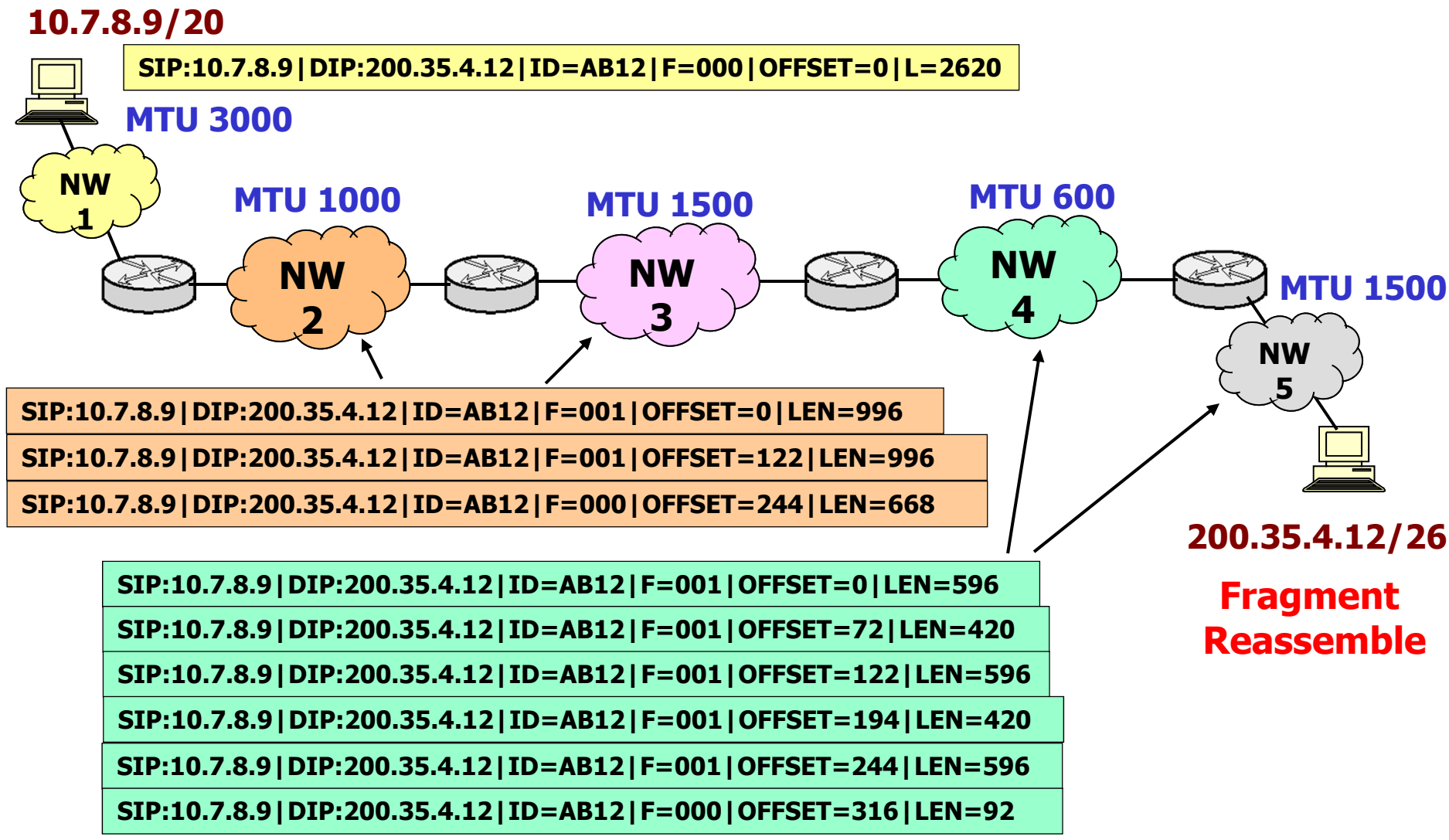
Ch22: 22.16 Fragment Loss

- **Fragment ต้องมาครบจึงจะประกอบได้ ดังนั้นถ้ามี Fragment Loss ปลายทางจะต้องเก็บ Fragment ที่เหลือไว้จนกว่าจะครบ ซึ่งจะเกิด Delay**
- **การเก็บ Fragment จะใช้ Memory และไม่สามารถเก็บได้ตลอดไป มิฉะนั้น Resource ของ Host ปลายทางจะไม่เหลือให้ทำอย่างอื่น**
- **IP กำหนด Timer ชื่อ 'Reassembly Timer' ซึ่งถ้า Timer Expire(หมดอายุ) ทุกๆ Fragment ในชุดของ Datagram นั้นจะถูกโยนทิ้ง**
  - IP ไม่มี Mechanism สำหรับ Fragment Retransmission เนื่องจาก Fragment กระทำที่กลางทาง
    - ที่จริงจะกระทำไม่ได้ เพราะผู้ส่งไม่รู้ว่า Fragment มันแบ่งอย่างไร ที่ส่วนใหนของ Datagram เดิม
  - ดังนั้นถ้า Fragment Loss จะทำให้ Timer Expire และจะลงเอยด้วยการส่งทั้ง Datagram นั้นมาใหม่ (All-or-Nothing)

Ch22: 22.17 Fragmenting A Fragment

- **เป็นไปได้ที่ Router ตัวหนึ่งทำการ Fragment Datagram ส่งไปใน Internet และ Fragment นั้นต้องผ่าน Network ที่มีค่า MTU ต่ำกว่าขนาดของ Fragment**
  - เราสามารถทำ Fragment ของ Fragment กี่ระดับก็ได้
  - Router จะมองไม่ออกว่านี่เป็น Fragment ของ Fragment
  - การประกอบจะกระทำครั้งเดียวที่ปลายทาง
    - ไม่มีการประกอบจาก Sub-fragment เป็น Fragment ก่อน แล้วจึงค่อยประกอบ Datagram

IP Fragmentation



End of Chapter 21-22

- **Homework II Due Next Week**
  - Download HW2 (Week 4)

End of Week 4

- **Next, Week V: Chapter 23**
  - Chapter 23: Supporting Protocol and Technologies
    - ARP
    - ICMP
    - DHCP
    - NAT

CPE 426 Computer Networks

Chapter 5:

Text Chapter 23: Support Protocols

TOPICS

- **สรุปเรื่อง IP Address Subnetting**
- **Chapter 23: Supporting Protocols**
  - ARP: 23.1-23.7
    - ใช้สำหรับหา HW Address(MAC Address)
  - ICMP: 23.8-23.9
    - ใช้ในการส่ง Message ใน Internet
  - DHCP: 23.10-23.14
    - ใช้สำหรับกำหนด IP Address ให้กับ Host
  - NAT: 23.15-23.19
    - ใช้เพื่อแก้ไขปัญหา IP Address ไม่พอใช้งาน

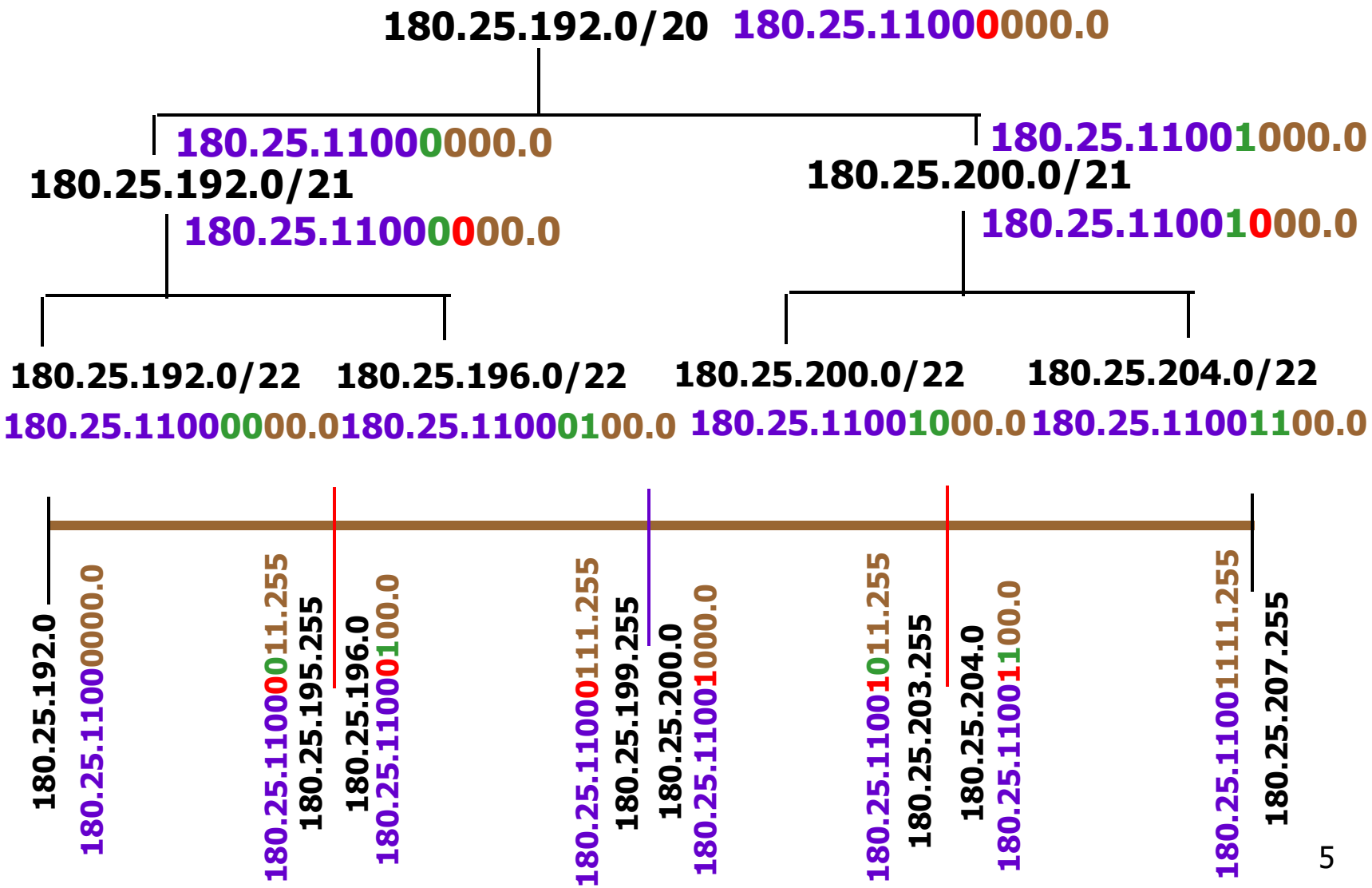
สรุปการอ่าน IP Address

- กำหนด **IP Address** และ **Net Mask**
  - ถ้าไม่กำหนด Net Mask ถือว่าเป็น Classful ให้ใช้ Default Net Mask
- **Prefix** หาได้จากการทำ **Bit-wise Logical 'AND'** ระหว่าง **IP Address** และ **Netmask**
- **Suffix** ได้จากการทำ **'AND'** ระหว่าง **IP Address** และ **1's Complement** ของ **Netmask**
  - หรือได้จากการนำ IP Address ลบออกด้วย Prefix
- **Host range** เริ่มจากนำส่วน **Prefix** มาและกำหนดให้ส่วน **Suffix** มีค่าเท่ากับ **'1'** (ไม่ใช่ **'0'**=ยกเว้น) จนถึง **Host ID** สุดท้ายคือเมื่อส่วน **Suffix** มี **Bit** เป็นหนึ่งทั้งหมด ยกเว้นบิตสุดท้ายจะเป็นศูนย์
- **Broadcast Address** ได้จากการนำส่วน **Prefix** มาและกำหนดให้ส่วน **Suffix** มีบิตเป็นหนึ่งทั้งหมด

สรุปการแบ่ง Subnet

- เริ่มจากการนำ Network เดิม ที่ประกอบด้วย Net ID(Prefix) และ Net Mask เดิม
- จำนวน Subnet ที่แบ่งได้จะเท่ากับ 2^B
  - B คือจำนวนบิตที่เพิ่มจาก Net mask เดิม
- Prefix ใหม่ ทั้ง 2^B ตัว (NW ID ใหม่ของ Subnet ที่ได้ 2^B Subnet) หาได้จากการนำ Prefix เดิม ต่อด้วย Bit ที่เพิ่ม ซึ่ง Pattern ของ Bit ที่เพิ่มจะมี ได้ 2^B แบบ
- การหา Address Range และ Broadcast Address ของแต่ละ Subnet ใช้หลักการเช่นเดียวกับที่กล่าวก่อนหน้า

Example


180.25.192.0/20 180.25.11000000.0
180.25.11000000.0
180.25.192.0/21
180.25.11001000.0
180.25.200.0/21
180.25.11000000.0
180.25.11001000.0
180.25.192.0/22
180.25.196.0/22
180.25.200.0/22
180.25.204.0/22
180.25.11000000.0
180.25.11000100.0
180.25.11001000.0
180.25.11001100.0
180.25.192.0
180.25.11000000.0
180.25.195.255
180.25.11000011.255
180.25.196.0
180.25.11000100.0
180.25.199.255
180.25.11000111.255
180.25.200.0
180.25.11001000.0
180.25.203.255
180.25.11001011.255
180.25.204.0
180.25.11001100.0
180.25.207.255
180.25.11001111.255

Suggestion

- **ถ้ายังไม่เข้าใจเรื่อง IP Address ให้ Download Program ชื่อ 'subnet10' จาก Bosun Software มาทดลองเล่นดู**
  - http://www.filewatcher.com/m/subnet10.zip.62032.0.0.html
  - Run บน XP ถ้าจะ Run บน Window 7
    - ตั้ง XP SP1-3Compatible Mode
    - Install 'MSVBVM50.DLL'

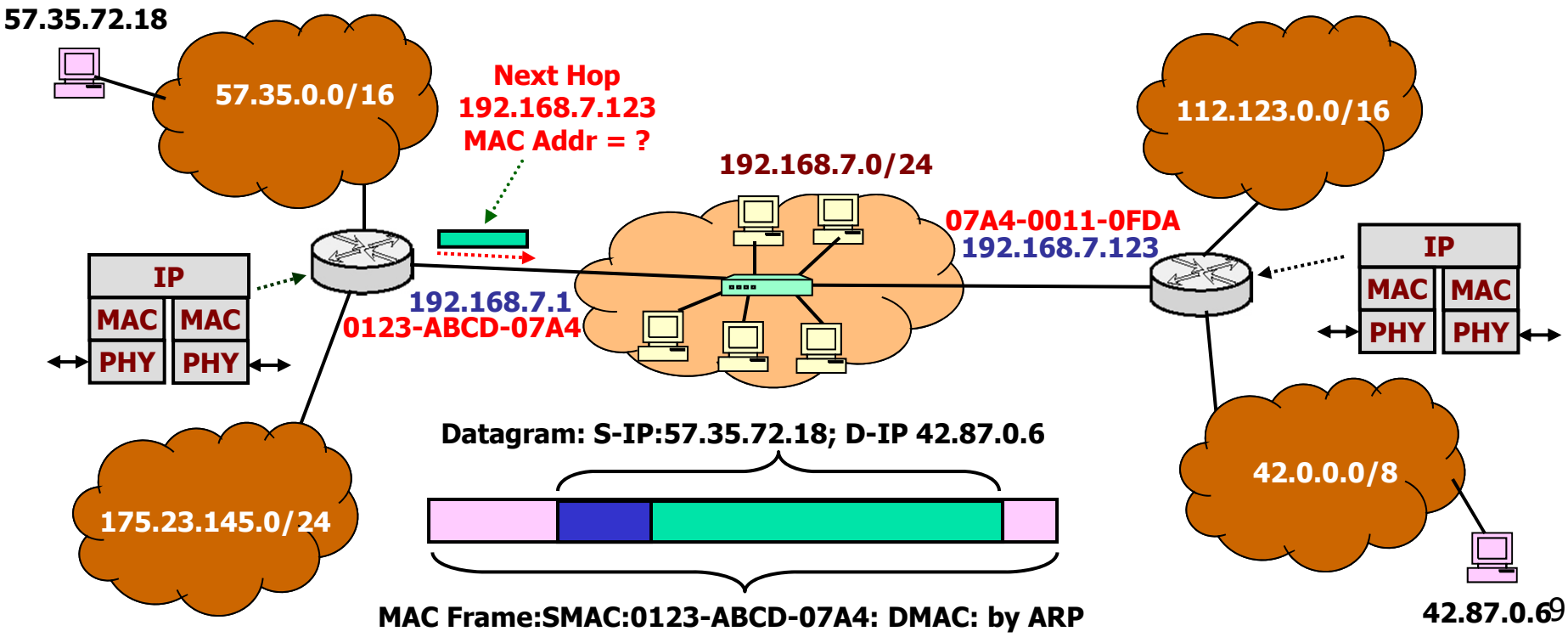
- **Chapter 23: Supporting Protocols and Technologies**
  - ARP: 23.1-23.7
    - ใช้สำหรับหา HW Address(MAC Address)
  - ICMP: 23.8-23.9
    - ใช้ในการส่ง Message ใน Internet
  - DHCP: 23.10-23.14
    - ใช้สำหรับกำหนด IP Address ให้กับ Host
  - NAT: 23.15-23.19
    - ใช้เพื่อแก้ไขปัญหา IP Address ไม่พอใช้งาน

Ch.23: 23.2 Address Resolution

- **จากที่กล่าวมาแล้ว การส่ง IP Datagram ผ่าน Internet จะส่งผ่าน Router ที่เชื่อมต่อระหว่าง Network ที่แตกต่างกัน**
  - Router จะส่งข้อมูลให้ Router ตัวถัดไป กำหนดจาก IP Address ของ Interface ของ Router ตัวถัดไป
  - Datagram จะต้องถูกบรรจุใน Network Protocol ที่ต้องส่งผ่าน โดยกำหนด Network Address ต้นทางคือ Interface ของ Router ผู้ส่ง และ Network Address ปลายทางคือ Interface ของ Router ที่เป็น Next Hop
  - แต่ตาราง Routing Table จะกำหนดแค่ IP Address ของ Interface ของ Next Hop เท่านั้น
    - การประกอบ Network Packet จะต้องใช้ Network Address ด้วย
  - ดังนั้นเราต้องทำ Address Resolution กล่าวคือหา Network Address จาก IP Address ที่กำหนด (Mapping)
    - เราอาศัย Protocol ที่ชื่อ Address Resolution Protocol(ARP)
  - Address Resolution จะทำงานใน Local Network เท่านั้น

Ch.23: 23.2 Address Resolution

- ในบทนี้ เราจะเน้นเฉพาะกรณีที่ IP Datagram ต้องผ่าน LAN และจำเป็นที่จะต้องบรรจุ ARP ภายใน MAC Frame(Ethertype II)
  - ARP จะทำหน้าที่หา MAC Address จาก IP ที่กำหนด
  - ARP เป็น Protocol ที่อยู่บน Layer 2 (ไม่บรรจุใน Datagram)



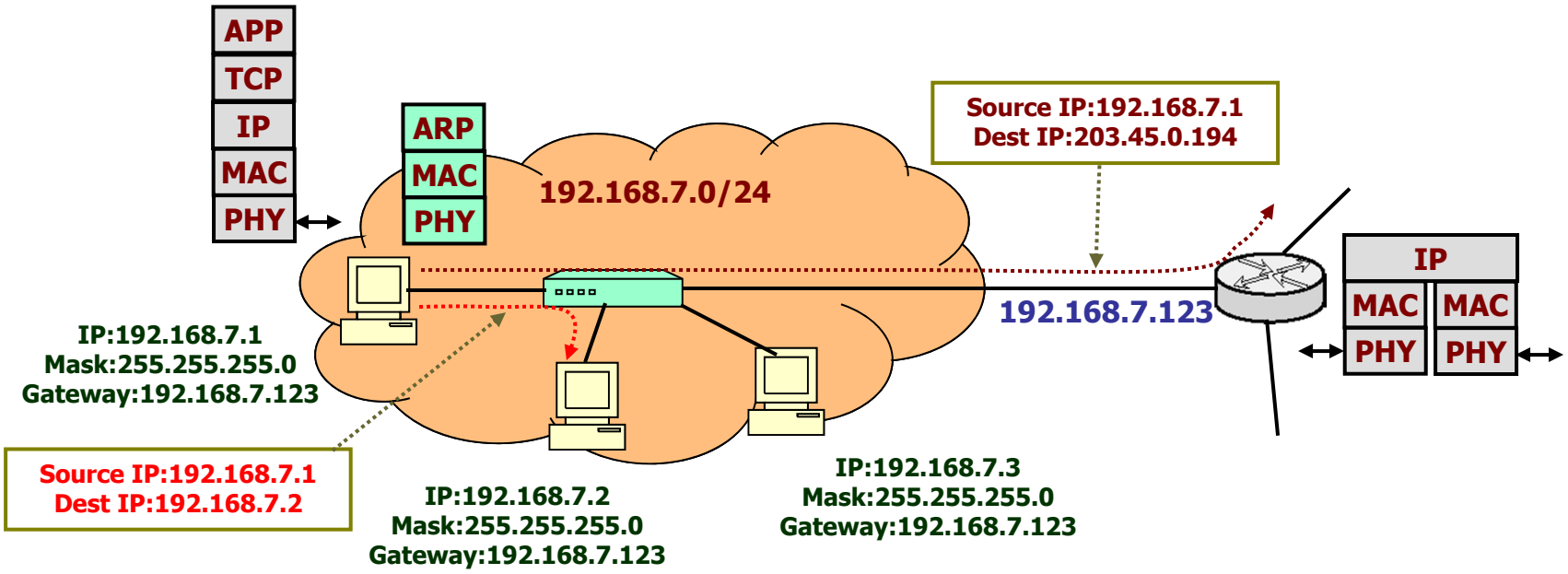
Ch.23: 23.3 Address Resolution Protocol (ARP)

- **เนื่องจาก Ethernet เป็น Broadcast Network**
  - ARP ใน Ethernet จะใช้การ Broadcast ถามว่าใครเป็นเจ้าของ IP Address ที่ต้องการ Resolve
    - ทุกๆ Host และ Interface ของ Router จะต้อง Run ARP
    - เมื่อ Host หรือ Interface ของ Router ได้รับ ARP Broadcast ถ้ามันเป็นเจ้าของ IP Address นี้ มันจะตอบกลับด้วย Hardware Address(คือ MAC Address ของมัน)
- **แม้แต่การส่งข้อมูลใน LAN วงเดียวกัน ไม่ผ่าน Router**
  - ถ้าเรา Run Application ของ TCP/IP ซึ่งการสื่อสารกำหนดจาก IP Address เราต้องใช้ ARP เพื่อที่จะประกอบ MAC Frame

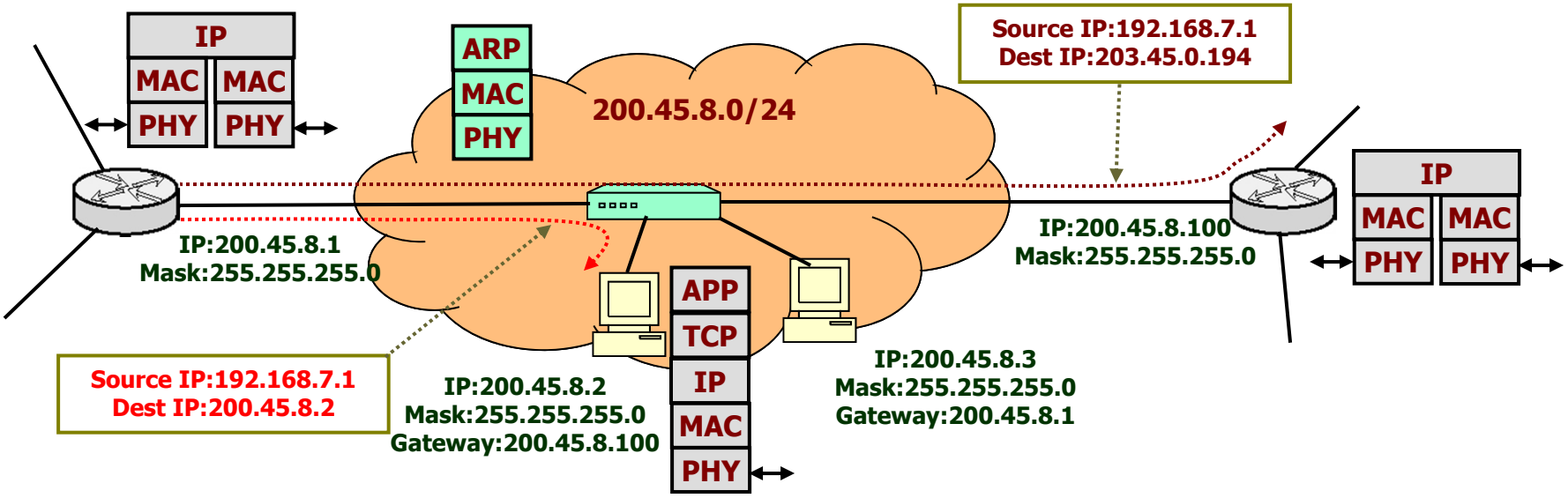
Ch.23: 23.3 Address Resolution Protocol (ARP)

- **TCP/IP Application ต้องการส่งข้อมูลให้แก่ Host กำหนดโดย IP Address ปลายทาง ส่วนของ OS จะทำการหาว่า IP ปลายทางนั้นอยู่ใน LAN วงเดียวกันหรือไม่ โดยการ Match ค่า Prefix ของ IP ปลายทาง กับ Prefix ของเครื่อง**
  - กรณีที่ 1: อยู่ใน LAN วงเดียวกัน ARP จะถูกใช้ในการหา MAC Address ของ IP ปลายทาง
  - กรณีที่ 2: อยู่นอก Network ดังนั้น Datagram จะต้องถูกส่งให้ Interface ของ Router ที่กำหนดโดยค่า IP ของ Gateway และ ARP จะถูกนำมาใช้ในการหา MAC Address ของ Interface นั้น
- **ในกรณีของ Router เมื่อต้องการจะส่งข้อมูลผ่าน LAN**
  - กรณีที่ 1: Datagram ส่งให้ IP ของ Host อยู่ใน NW ที่ต่อโดยตรงกับ Router ARP จะถูกใช้ในการหา MAC Address ของ Host นั้น
  - กรณีที่ 2: Datagram ส่งให้ IP ที่เป็น Interface ของ Router ตัวถัดไป ARP จะถูกนำมาใช้ในการหา MAC Address ของ Interface ของ Router ที่เป็น Next Hop

Host ARP Mechanism



Router ARP Mechanism



Ch.23: 23.4-5 ARP Message Format/Encapsulation

- **ARP เกือบทั้งหมดใช้ในการ Resolve MAC Address จาก IP Address ที่กำหนด**
  - แต่ Message Format ออกแบบมาให้เป็น Generic คือใช้กับ Protocol อื่นๆได้
- **ARP จะถูกบรรจุใน Hardware Frame(ปกติคือ L2 Frame) ของ Network นั้น ใน LAN จะถูกบรรจุใน MAC Frame**
  - Ethertype Field จะกำหนดว่าเป็น ARP Message = 0x806 ทั้ง ARP Request และ ARP Reply

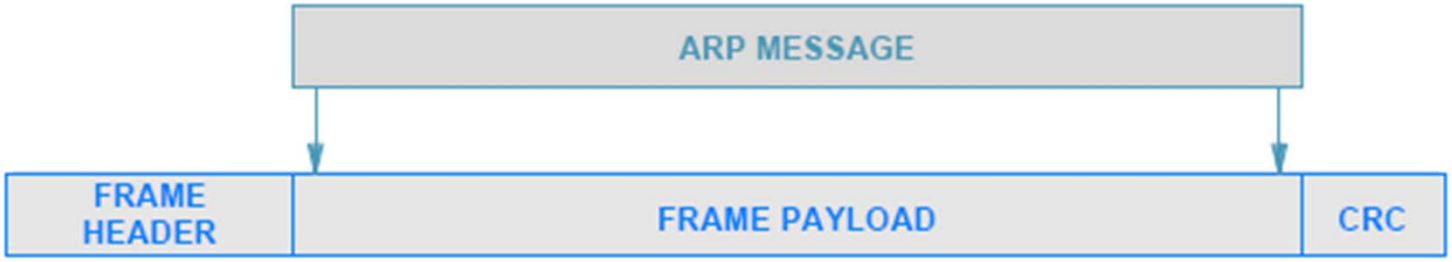
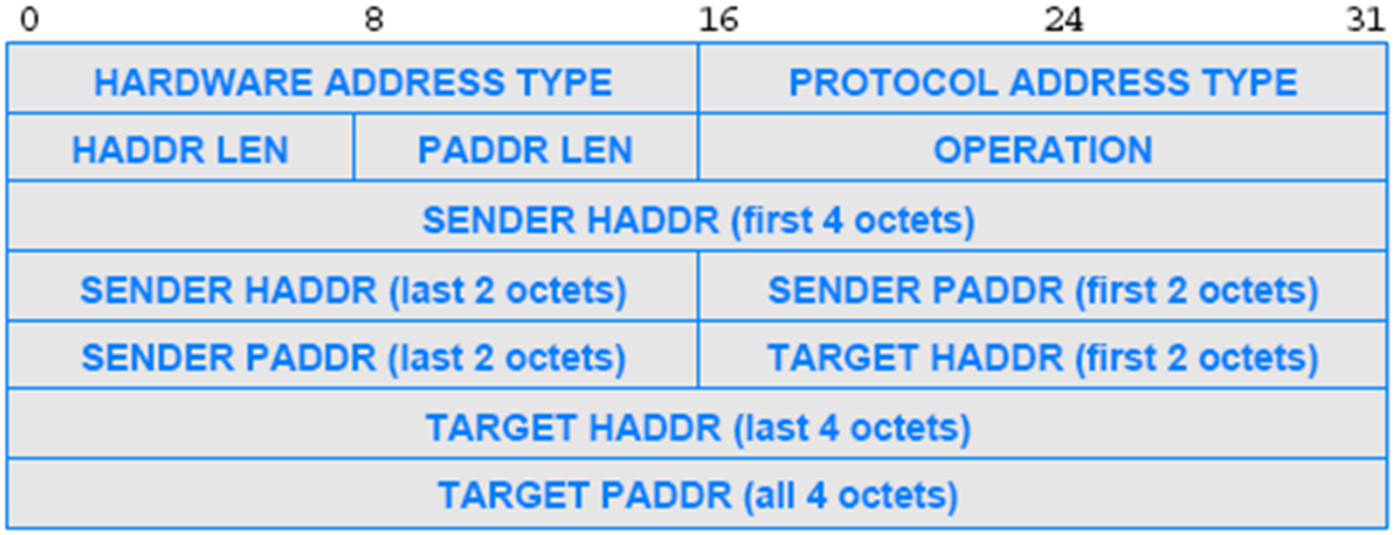


Figure 23.4 Illustration of ARP encapsulation in an Ethernet frame.

Ch.23: 23.4-5 ARP Message Format/Encapsulation



- Hardware Address Type: 1 = Ethernet
- Protocol Address Type: 0x0800 = IPv4
- HADDR LEN: Size of HW Address(Bytes)
- PADDR LEN: Size of Protocol Address(Bytes)
- Operation: Request = 1, Response = 2
- Sender HADDR; Sender PADDR; Target HADDR; Target PADDR

Ch.23: 23.6 ARP Caching and Message Processing

- **ถ้าทุกครั้งที่มีการส่ง Datagram ต้องทำ ARP Request และรอ ARP Reply การสื่อสารจะขาดประสิทธิภาพ**
  - ข้อมูลของ ARP จะถูก Cache ไว้ (ใน Window ดูจาก Command Line: 'arp –a')
  - Cache จะเป็นตารางไม่ใหญ่
    - ข้อมูลที่เก่าที่สุดจะถูกแทนที่ด้วยข้อมูลใหม่
    - ข้อมูลจะถูกลบทิ้ง เมื่อถึงเวลาหมดอายุ
    - เมื่อต้องการจะ Resolve Address ส่วน Cache จะถูกตรวจดูก่อนว่ามีอยู่หรือไม่ ถ้าไม่มีค่อยทำ ARP Request
    - Cache จะ Update เมื่อได้รับทั้ง ARP Request และ ARP Reply

Ch.23: 23.7 Conceptual Address Boundary

- **ARP เป็น Function ที่เกี่ยวข้องกับ Network Interface Layer ใน 5 Layer ของ TCP/IP Model**
- **ARP จะช่วยซ่อนรายละเอียดของ Hardware Address ทำให้การสื่อสารใน Application ใช้เพียงแค่ IP Address**
  - ARP ทำหน้าที่เป็น Conceptual Address Boundary

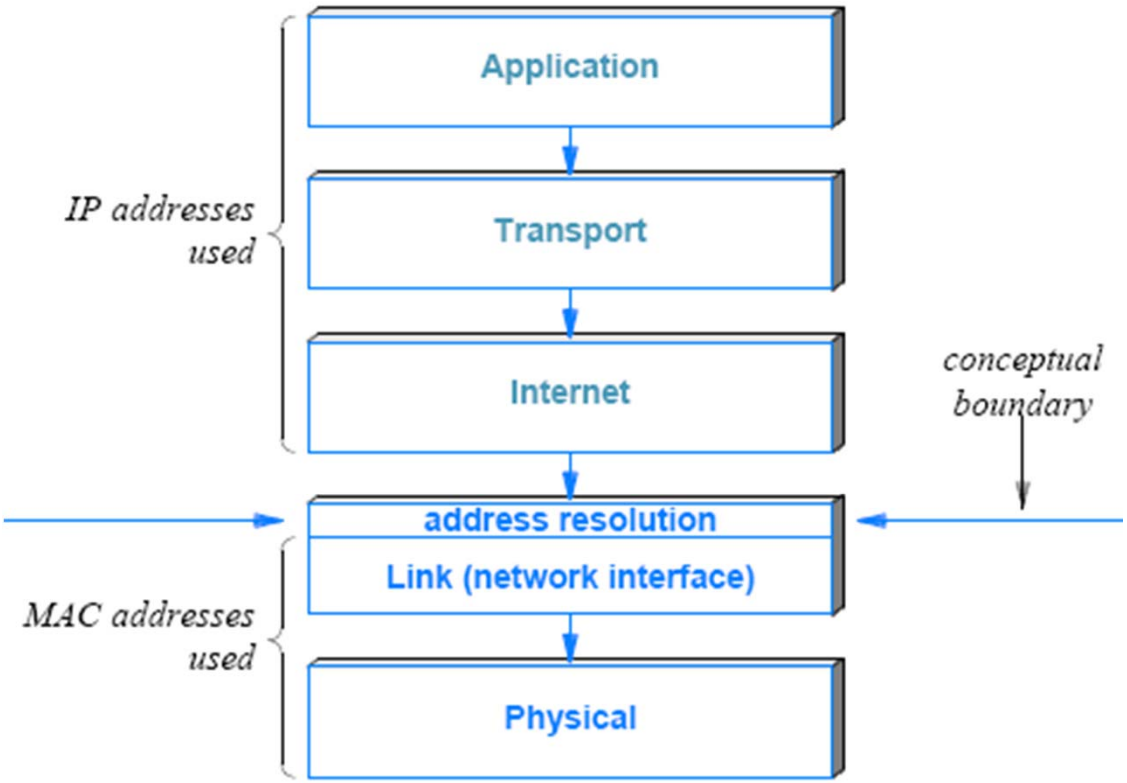


Figure 23.5 Illustration of the boundary between the use of IP addresses and MAC addresses.

สรุป ARP

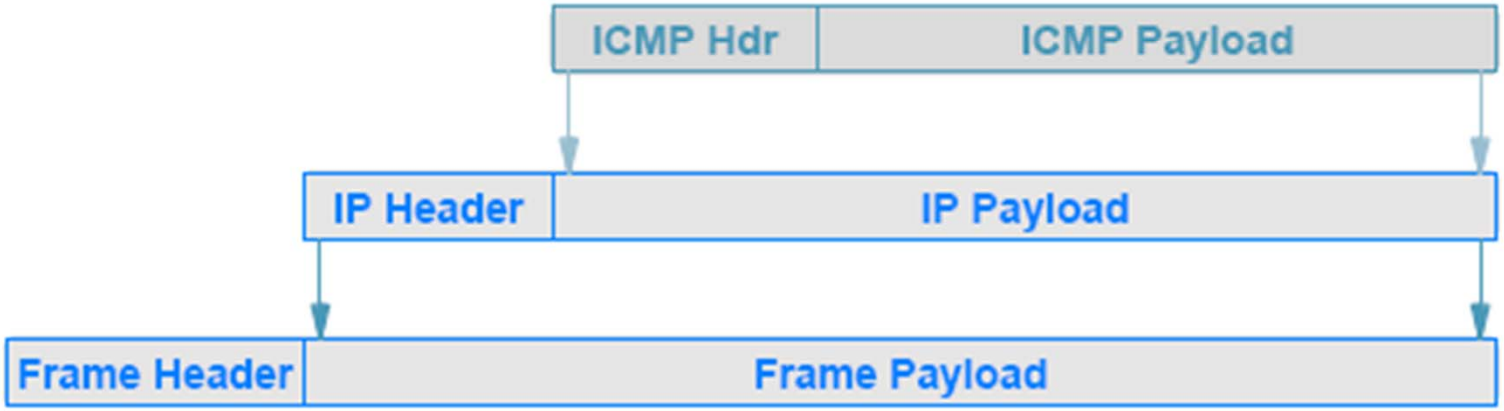
- ใช้ในการหา Hardware Address(MAC Address) จาก IP Address(Network Protocol Address) ที่กำหนด
- เป็น Ethertype II Message, Code 0x0806
- ใน Ethernet จะอาศัยการ Broadcast
- Run ที่ทั้ง Host และ Interface ของ Router
- จุดประสงค์เพื่อหา Destination MAC Address ในการประกอบ Frame
- เป็น Transparent กับ User ช่วยซ่อนรายละเอียดของ Network ที่อยู่ชั้นล่างของ IP
- ตาราง MAC Address จาก ARP จะมีการ Cache ในช่วงเวลาสั้นๆ

Ch.23: 23.8-9 Internet Control Message Protocol (ICMP)

- **เนื่องจาก IP มีการทำงานแบบ Best-Effort การส่ง Datagram จะมี Error เกิดขึ้นได้เสมอ เช่น Lost, Delay, Duplicate หรือ Out of Order**
- **IP Header มี Mechanism ในการช่วยจัดการกับ Error**
  - Header Checksum
  - Time to Live
- **อย่างไรก็ตาม เมื่อมี Error เกิดขึ้น IP มี Protocol ชื่อ ICMP=Internet Control Message Protocol ใช้สำหรับการรายงาน Error นั้นกลับมายังผู้ส่ง**
  - ICMP จะถูกบรรจุอยู่ใน IP Datagram และส่ง ดังนั้น ICMP จะเป็น Protocol ที่วางอยู่บน IP Layer
  - ICMP จะนำทั้ง Error Message และ Information อื่นๆด้วย
    - ที่สำคัญแสดงดังตาราง

Ch.23: 23.8-9 Internet Control Message Protocol (ICMP)

Number	Type	Purpose
0	Echo Reply	Used by the ping program
3	Dest. Unreachable	Datagram could not be delivered
5	Redirect	Host must change a route
8	Echo	Used by the ping program
11	Time Exceeded	TTL expired or fragments timed out
12	Parameter Problem	IP header is incorrect
30	Traceroute	Used by the traceroute program



Ch.23: 23.10 Host Configuration

- **Host Configuration ปกติจะประกอบไปด้วยสองขั้นตอน เรียก Bootstraping**
  - เมื่อเรา Boot Computer ตัว OS จะกำหนดค่า Configuration พื้นฐานในการเชื่อมต่อกับ Local Network
  - ต่อมา Protocol Software จะรับผิดชอบในการเติมข้อมูลในส่วนที่เหลือ เช่น IP Address, Mask และ IP ของ DNS Server

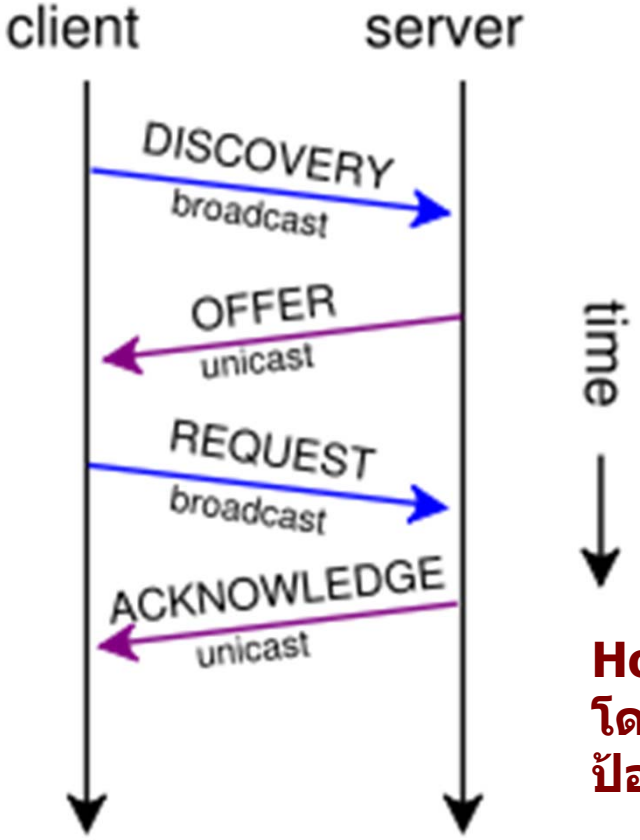
Ch.23: 23.11 Dynamic Host Configuration Protocol (DHCP)

- **สมัยก่อนจะใช้ RARP (Reverse Address Resolution Protocol) ในการที่ Host จะได้รับ IP Address จาก Server โดยกำหนด MAC Address ของ Host (การทำงานจะกลับกับ ARP)**
  - เราสามารถส่ง ICMP Message 'address mask request' และ 'router discovery' เพื่อจะได้ Address Mask และ IP ของ Gateway
  - ลักษณะการทำงานจะอาศัยการ Broadcast เป็นหลัก
- **ต่อมา Protocol ที่ชื่อ 'Bootstrap Protocol'(BOOTP) ได้ถูกพัฒนา**
  - Host สื่อสารกับ BOOTP Server ผ่านการ Broadcast โดยใช้ IP 255.255.255.255(ปลายทาง) และ 0.0.0.0(ต้นทาง)
  - BOOTP Server จะใช้ MAC Address ของผู้ส่งในการส่งข้อมูลกลับแบบ Unicast
  - BOOTP จะส่ง IP ให้ตามตาราง ดูจาก ID(MAC Address) ของ Host
    - ตารางนี้ จะต้องจัดตั้งโดย Network Administrator แบบ Manual
- **พัฒนาล่าสุดคือ DHCP แก้ปัญหาเรื่อง Manual Configuration**

Ch.23: 23.11 Dynamic Host Configuration Protocol (DHCP)

- **DHCP พัฒนาต่อจาก BootP (Booth Protocol) มีความสามารถทำ Lease และส่ง Options อื่นๆเช่น Default Gateway, Subnet Mask, DNS ได้**
  - เมื่อเปิดคอมพิวเตอร์ คอมพิวเตอร์จะส่ง DHCP Discover ผ่านการ Broadcast
  - DHCP Server จะส่ง DHCP offer ผ่าน DHCP Reply
    - Static Permanent Address สำหรับ Server
    - Dynamic Address จาก IP Pool ที่ Configure ไว้
    - การจ่าย IP จะเป็นลักษณะ Lease (ให้ยืม) ตามเวลา ถ้า Host ยังต้องการใช้ต่อต้องทำการ Renew
      - Renew จะกระทำเมื่อถึงครึ่งของเวลา Lease
  - ปกติ DHCP Server อาจจะมีได้มากกว่าหนึ่งตัว ดังนั้น Host จะต้องเลือกว่าจะใช้ IP ที่เสนอ จาก Server ใดโดยส่ง Message DHCP Request ไปยัง Server
  - Server ตัวที่ได้รับเลือกจะส่ง DHCP Acknowledge กลับมายัง Host เพื่อยืนยัน
    - Server ที่ไม่ได้รับเลือก สามารถนำ IP ที่เสนอ ไปให้คนอื่นได้

23.12 Mechanism ของ DHCP



1. DHCP ใช้ Port เดียวกับ BootP
2. 67/UDP Server Side
3. 68/UDP Client Side
4. 4 Basic Phases
    1. IP Lease Request
    2. IP Lease Offer
    3. IP Lease Selection
    4. IP Lease Acknowledgement

Host จะ Retransmit Request จนกว่าจะได้คำตอบ โดยจะจับ Random Number ก่อน Retransmit เพื่อป้องกันการ Flooding ของ DHCP

Ch.23: 23.13 DHCP Message Format

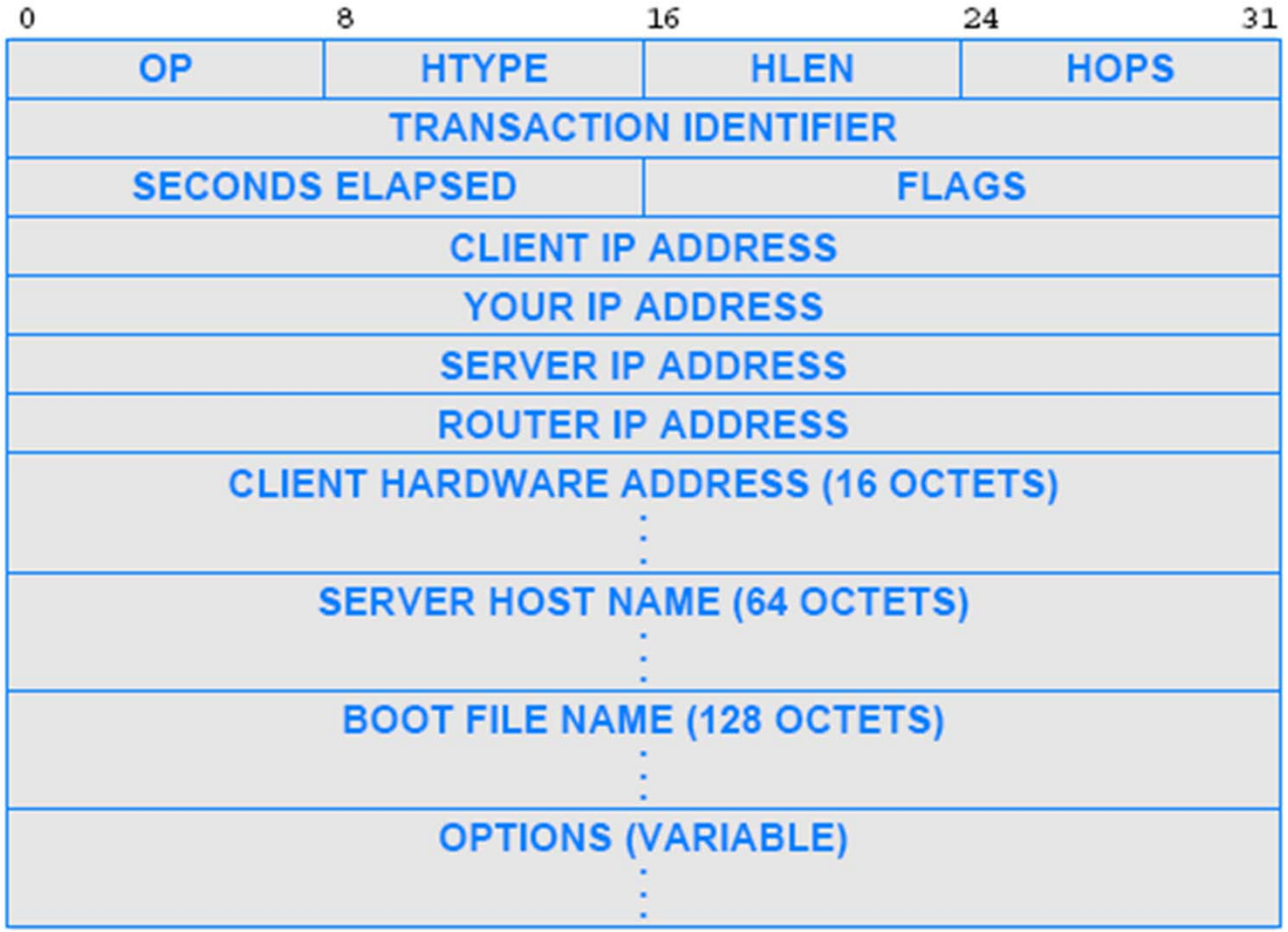


Figure 23.8 The DHCP message format.

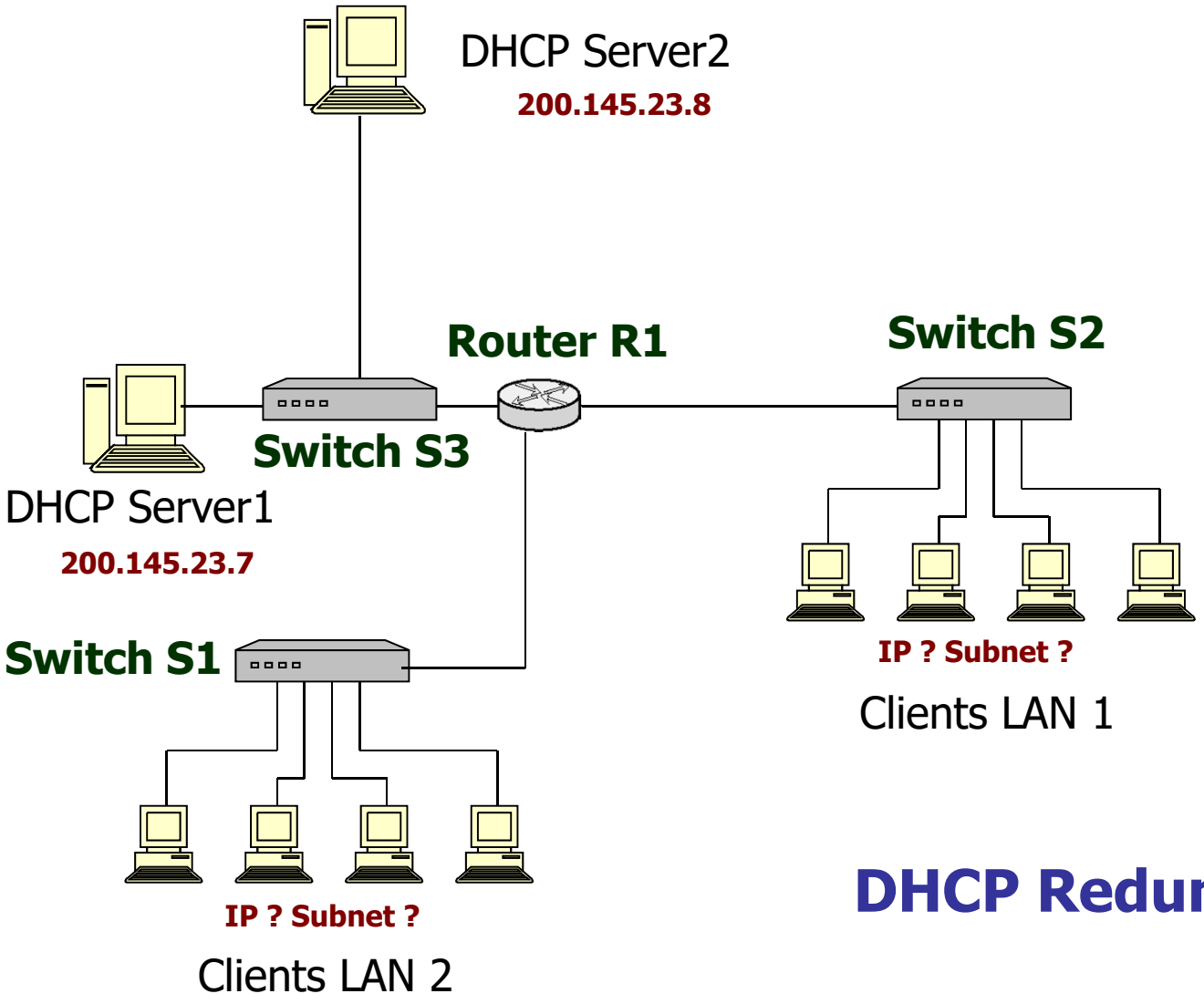
Ch.23: 23.13 DHCP Message Format

- OP บ่งบอกว่าเป็น Request หรือ Response
- HTYPE & HLEN กำหนดชนิดและความยาว Address ของ Hardware
- FLAG กำหนดว่า Reply จะเป็น Broadcast หรือ Direct
- HOPS กำหนดช่วงของ Message Forward
- TRANSACTION IDENTIFIER กำหนด ID ของ Request
- SECONDS ELAPSED จำนวนวินาทีที่ Client ได้ Boot แล้ว
- CLIENT IP ADDRESS ใช้กรณีที่ Client รู้ IP ของตนเองแล้ว
- Field ที่เหลือใช้สำหรับ Server ใส่ข้อมูลส่งกลับมาให้ Client
- ส่วน Boot File Name ใช้ในการส่ง Boot Image Filename ซึ่ง Client สามารถ Download File นี้ผ่าน FTP

Ch.23: 23.14 Indirect DHCP Server Access Through Relay

- การ Broadcast ของ Host เพื่อหา Server จะกระจายในเฉพาะ Local Network
- อย่างไรก็ตาม เราไม่จำเป็นต้องตั้ง DHCP Server ในทุกๆ Local Network
- ที่ตัว Router เราสามารถ Configure ให้ทำ DHCP Relay
  - โดยตั้ง DHCP Relay Agent ที่ Interface ของ Router

Ch.23: 23.14 Indirect DHCP Server Access Through Relay



DHCP Redundancy

Ch.23: 23.15 Network Address Translation (NAT)

- **ในการแก้ปัญหา IP Address ไม่พอใช้ บทที่แล้วเราพูดถึงกลไกสองวิธีที่จะจัดการกับปัญหาดังกล่าว**
  - (1)Subnetting และ (2) Classless Addressing (CIDR)
- **ในส่วนนี้ เราจะมาพูดถึงวิธีที่ 3 คือการทำ NAT**
  - คอมพิวเตอร์หลายๆเครื่องในแต่ละองค์กรและแต่ละ Network สามารถใช้ IP Address ที่ซ้ำกันได้
  - แต่ IP ที่ใช้ภายในองค์กรนี้ไม่สามารถต่อออก Internet โดยตรงได้ (Non-Routable IP)
  - การต่อออก Internet จะต้องเปลี่ยนเป็น IP ที่สามารถต่อออกโดยตรง (Routable IP) ซึ่งเป็น IP Address ที่ต้องไม่ซ้ำกับใคร
    - เรียกการทำ Network Address Translation (NAT)
  - ขบวนการแปลง IP นั้นจะกระทำที่ Gateway ทางออก Internet
    - การสื่อสารภายในองค์กร ไม่จำเป็นต้องแปลง
  - ข้อมูลที่จะออกนอก Internet จะถูกเปลี่ยน IP Address ต้นทางเป็น Routable IP
  - ข้อมูลที่กลับมาจาก Internet จะถูกเปลี่ยน IP Address ปลายทางเป็น IP ที่ใช้ภายใน
  - การเปลี่ยนจะอาศัยตารางเพื่อ แปลง IP Address
  - NAT อาจจะฝังตัวอยู่ในอุปกรณ์ที่ทำหน้าที่เป็น Gateway ต่างๆ เช่น Router หรือ WIFI Access Point

Ch.23: 23.15 Network Address Translation (NAT)

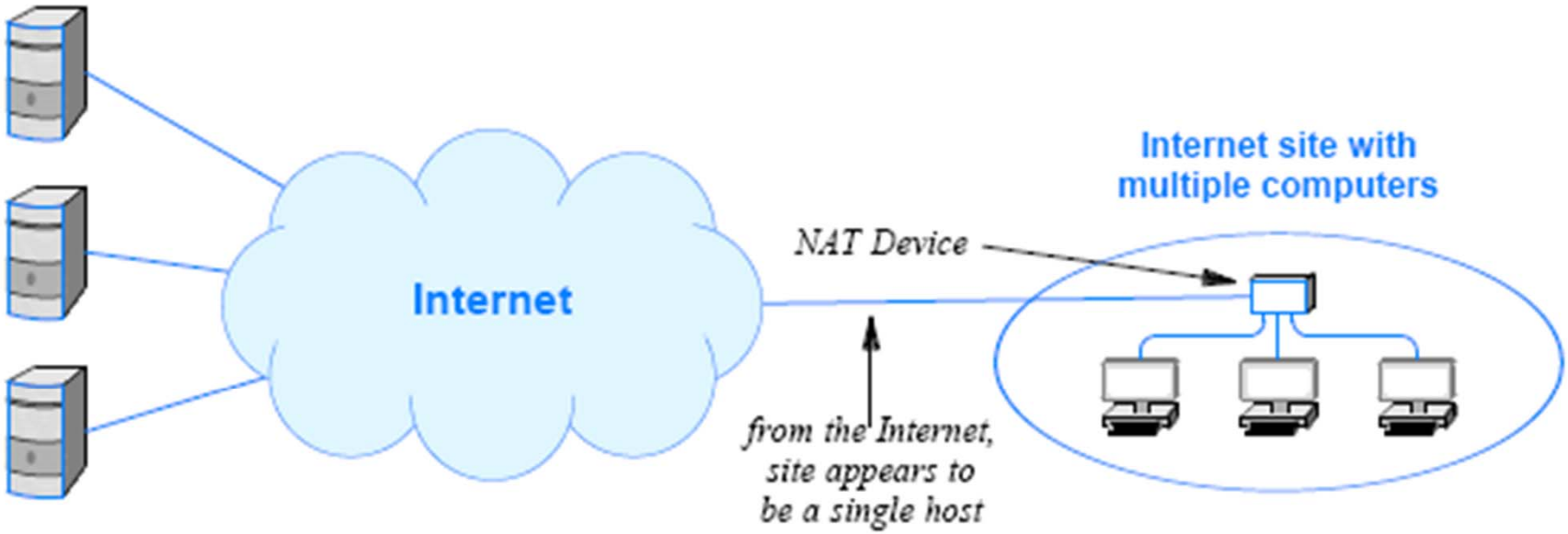


Figure 23.9 The conceptual architecture used with NAT.

Ch.23: 23.16 NAT Operation and Private Addresses

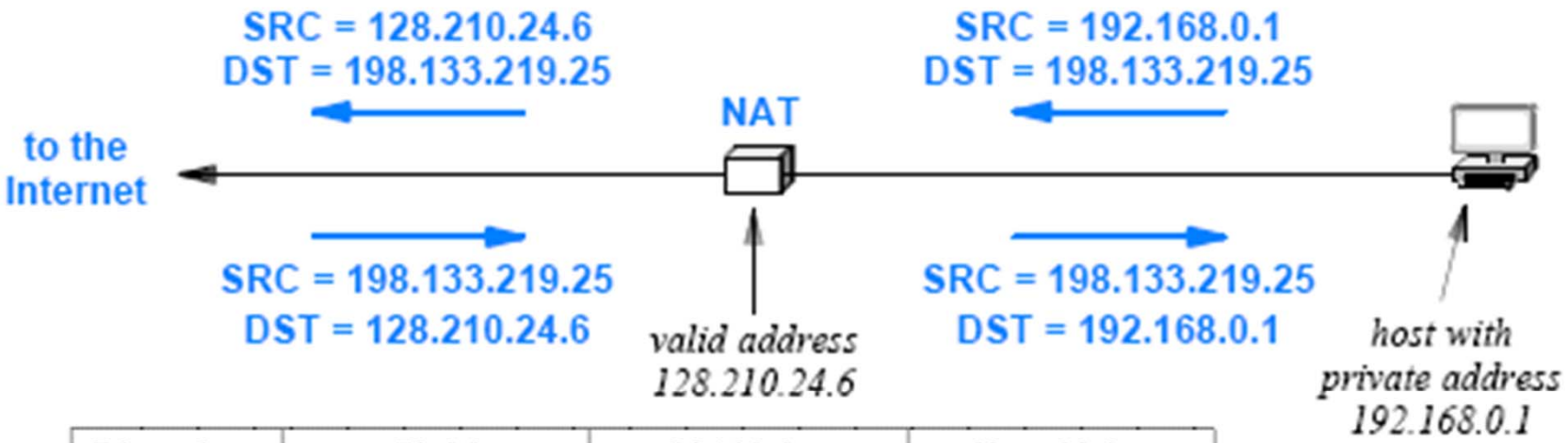
- **เป้าหมายของ NAT คือการสร้างภาพลวงตา**
  - มองจาก Internet ภายนอก จะมองเห็นแค่คอมพิวเตอร์เครื่องเดียว (หรือ Subnet เดียว) ที่มี IP Address ที่ถูกต้อง เนื่องจาก Datagram ที่ส่งออกมา ล้วนแต่ใช้ IP Address ต้นทางชุดเดียวกัน และ Datagram ที่ส่งกลับจะใช้ IP Address ปลายทางชุดนั้น
  - เมื่อมองจาก Host ภายใน คือการเชื่อมต่อโดยใช้ Private IP ที่สามารถ Route ภายใน Network ได้
    - Private IP ที่ใช้ เสมือนกับเราใช้สื่อสารภายใน Network นั้นเท่านั้น ไม่เกี่ยวกับภายนอก ดังนั้นองค์กรอื่นสามารถใช้ Private IP ชุดเดียวกันได้
    - แม้ว่าจะเป็น Private IP ที่ใช้และรู้จักเฉพาะใน Network ขององค์กร แต่อย่าลืมว่า Host ภายใน Network ขององค์กรจะต้องมี IP ไม่ซ้ำกัน ตามหลักการของ Prefix-Suffix ที่กล่าวในบทก่อน
    - Private IP นี้จะมองไม่เห็นจากภายนอก ดังนั้นใครจะใช้อย่างไร หรือใช้ซ้ำกันได้ ถ้าต่างองค์กรกัน
  - มาตรฐานของ Internet กำหนดชุดของ Private Address หรือ Nonroutable Address แสดงดังตาราง

Ch.23: 23.16 NAT Operation and Private Addresses

Block	Description
10.0.0.0/8	Class A private address block
169.254.0.0/16	Class B private address block
172.16.0.0/12	16 contiguous Class B blocks
192.168.0.0/16	256 contiguous Class C blocks

- **Private Address ไม่สามารถใช้ได้ ในการเชื่อมต่อกับ Internet ภายนอก**
- **เมื่อต้องการเชื่อมกับภายนอกต้องต่อผ่าน NAT**
  - Translate Source IP ใน Datagram ขาออก
  - Translate Destination IP ใน Datagram ขาเข้า

Ch.23: 23.16 NAT Operation and Private Addresses

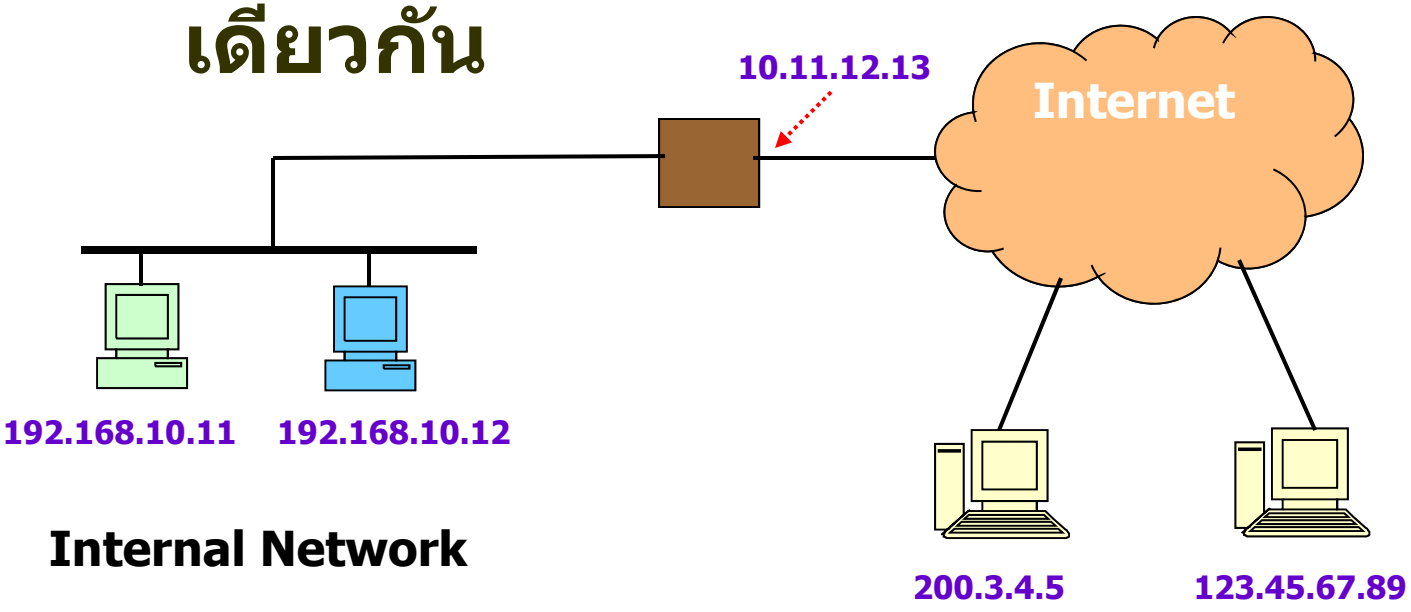


Direction	Field	Old Value	New Value
out	IP Source	192.168.0.1	128.210.24.6
	IP Destination	198.133.219.25	-- no change --
in	IP Source	198.133.219.25	-- no change --
	IP Destination	128.210.24.6	192.168.0.1

- NAT จะใช้ Translation Table
- ปกติ NAT จะทำงานโดยอัตโนมัติเมื่อมีการส่ง Packet เข้า-ออก

ปัญหาของ NAT(1)

- สองคอมพิวเตอร์ ต่อเข้า Server ตัวเดียวกัน



192.168.10.11 และ 192.168.10.12 เข้า Server 200.3.4.5 พร้อมกัน
Server ส่งข้อมูลกลับมาที่ 10.11.12.13 แต่ NAT ไม่รู้จะแปลงกลับไปไหน

Ch.23: 23.17 Transport Layer NAT (NAPT)

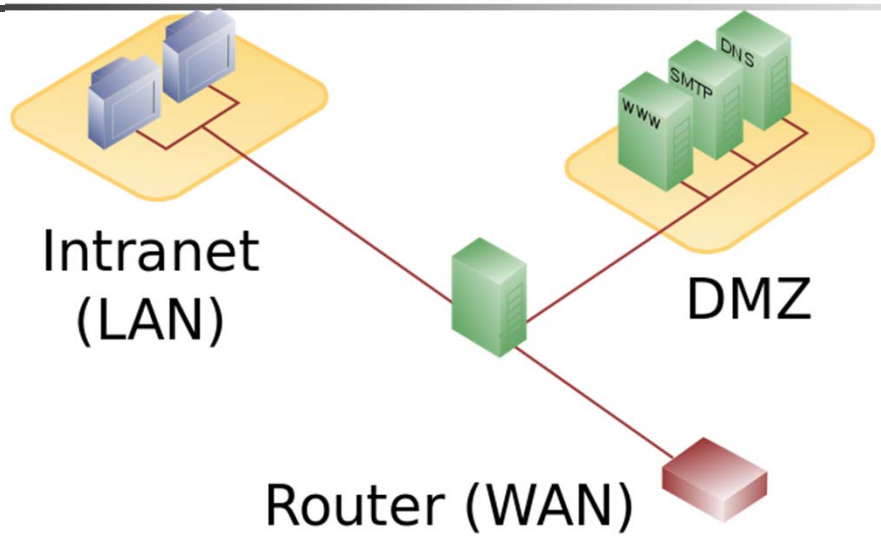
- ถ้าเรากำหนด **IP** เพียงเบอร์เดียวในการ **Translate** จะเกิดปัญหาเมื่อสองคอมพิวเตอร์พยายามติดต่อกับ **Server** ภายนอกตัวเดียวกัน เพราะ **NAT** ไม่สามารถแยก **Datagram** ทั้งสองออกได้

- วิธีแก้ปัญหาคือใช้ **Network Address and Port Translation (NAPT)**
  - นอกจากจะ Translate IP Address แล้ว จะทำการ Translate Port Number ด้วย

Dir.	Fields	Old Value	New Value
out	IP SRC:TCP SRC	192.168.0.1:30000	128.10.24.6:40001
out	IP SRC:TCP SRC	192.168.0.2:30000	128.10.24.6:40002
in	IP DEST:TCP DEST	128.10.19.20:40001	192.168.0.1:30000
in	IP DEST:TCP DEST	128.10.19.20:40002	192.168.0.2:30000

Ch.23: 23.18 NAT and Servers

- การสร้างตารางแบบอัตโนมัติใน NAT จะใช้งานได้ดีกรณีที่คอมพิวเตอร์ภายในทำการเริ่มต้นการสื่อสารเพื่อเชื่อมกับภายนอก (ออก Net)
- ในกรณีที่องค์กรตั้ง Web Server หลายตัว และยอมให้ภายนอกติดต่อเข้ามาจะทำให้ NAT ไม่สามารถรู้ได้ว่าจะต้องต่อกับ Server ตัวใด (Web Server อยู่ในองค์กร)
- วิธีแก้คือใช้ Twice NAT
  - Twice NAT จะทำงานร่วมกับ DNS Server
  - เมื่อคอมพิวเตอร์ภายนอก สอบถาม IP Address จาก DNS Server ขององค์กร DNS Server จะตอบกลับด้วย IP Address ของ NAT ที่กำหนดและจะเพิ่มตาราง Translation Table ที่ตัว NAT เพื่อต่อกับ Server ภายในที่มี Domain Name ตามที่ร้องขอ
  - อย่างไรก็ตาม Twice NAT จะทำงานไม่ได้ถ้าผู้ใช้เลือกติดต่อโดยใช้ IP Address โดยตรง
- วิธีที่สองคือใช้ IP จริงสำหรับ Web Server(ไม่ใช้ NAT) โดยแยกส่วน DMZ Zone สำหรับ Server



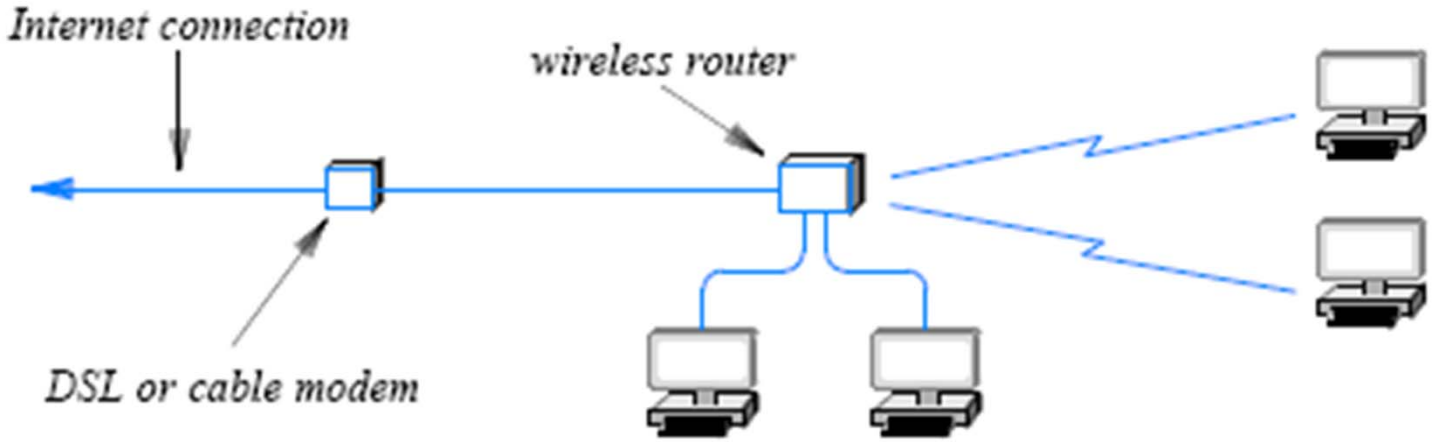
ในส่วน **DMZ** จะประกอบด้วย **Server** เพื่อให้ **Service** สำหรับภายนอก **Network** ขององค์กร ยกตัวอย่างเช่น

- **Web servers**
- **Mail servers**
- **FTP servers**
- **VoIP servers**
- **DNS Serer**

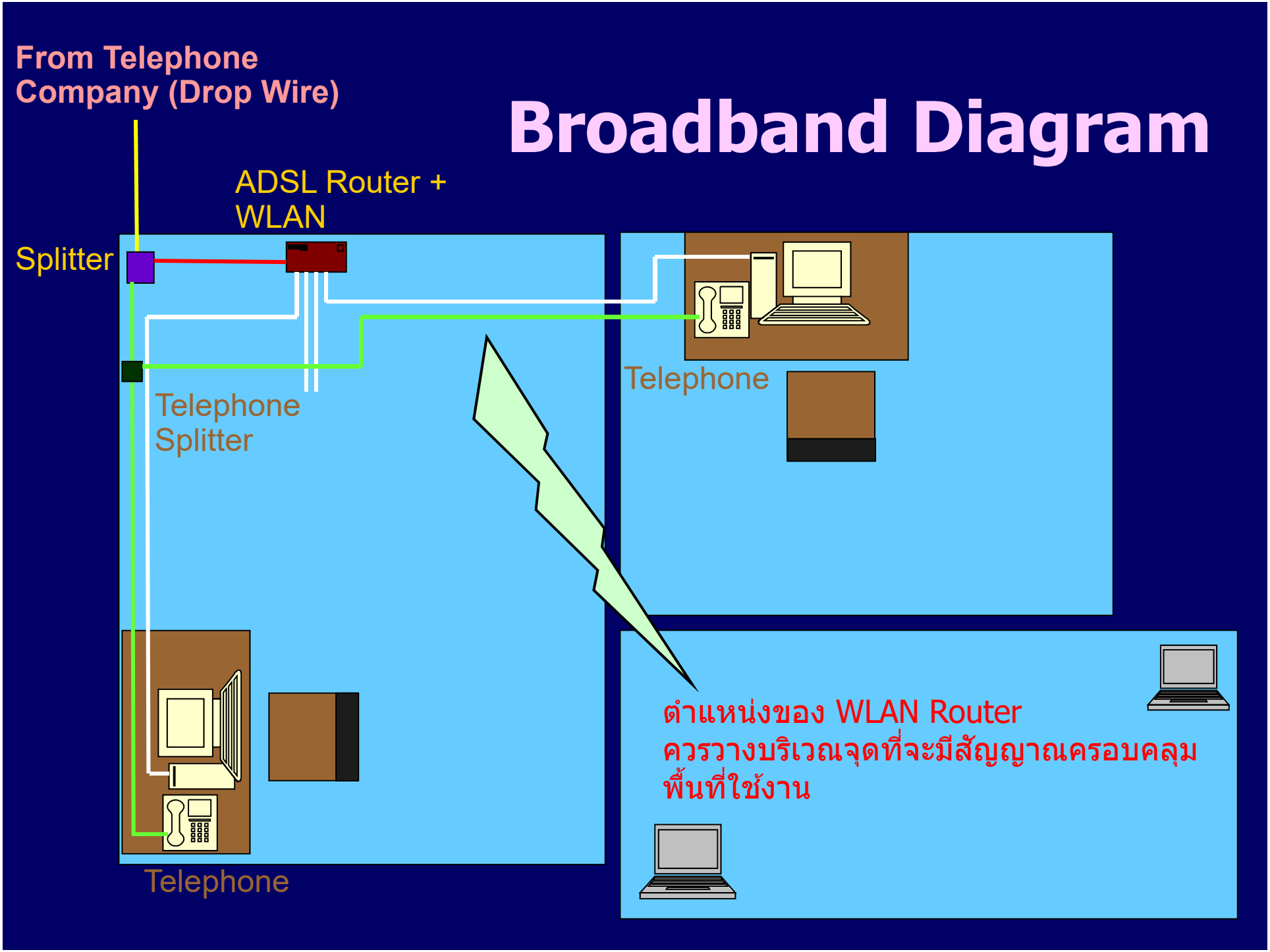
- ปกติจะวาง **Firewall** เพื่อป้องกันไม่ให้ภายนอกเข้ามายัง **Network** ภายในขององค์กรได้โดยตรง

Ch.23: 23.19 NAT Software and Systems For Use AT Home

- NAT จะมีประโยชน์ในการเชื่อมต่อ Internet ตามบ้านหรือองค์กรขนาดเล็ก ผ่าน Broadband (ADSL) ที่มีราคาถูก โดยที่เราไม่ต้องเสียค่าเช่าหมายเลข IP
- เราสามารถใช้ NAT Software Run บน PC หรือใช้ NAT Hardware ซึ่งสามารถ Implement ได้ในราคาถูก อย่างเช่นใน ADSL Router



Broadband Diagram



HW Week 5

- **Download HW4 Week 5**

CPE 426 Computer Networks

Chapter 6:
Text Chapter 24: IPv6

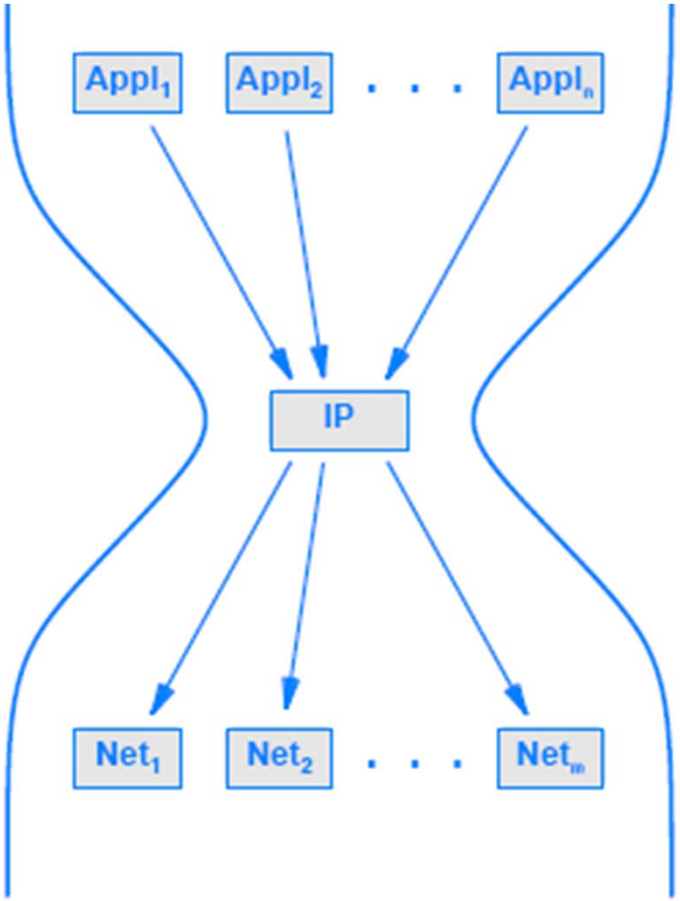
TOPICS

- **Chapter 24: IPv6**
  - Motivation
  - Features
  - Header Format
  - Addressing

Chapter 24:24.3 IPv6 Motivation

- **IP Protocol ประสพผลสำเร็จอย่างมาก ก่อให้เกิดการเปลี่ยนแปลงทั้งด้าน Hardware และ Software ทำให้การใช้งาน IP Address ขนาด 32 บิต ไม่พอ**
  - แม้ว่าจะมีวิธีการอื่นมาช่วย เช่นการทำ CIDR หรือ NAT แต่ยังคงมีข้อจำกัดการใช้งาน
- **นอกจากนี้แล้ว การออกแบบที่เป็น Best Effort ทำให้เกิดปัญหาใน Application สมัยใหม่ บางอย่าง เช่นพวก Real-Time Service**
- **ความต้องการอื่นๆที่ IP เดิมไม่สามารถให้ได้ เช่นเรื่องของการทำ Routing และ Addressing ที่สลับซับซ้อนขึ้นที่จะจัดการกับ Replicate Service หรือ Collaboration Group**

Chapter 24: 24.4 Difficulties



- **IP version ใหม่เริ่มคิดในปี 1993 แต่จนปัจจุบันนี้ยังไม่ถูกนำมาใช้อย่างกว้างขวาง และ Version เดิมยังคงมีการใช้งานไปอีกนาน**
- **เนื่องจาก IP จะต้องมีอยู่ในทุกอุปกรณ์ต้นทาง ปลายทาง และ Router**
  - หมายความว่า ถ้าจะเปลี่ยนเป็น Version ใหม่ จะต้องรื้อทั้ง Network
  - นอกจากนี้จะต้องลงทุนเขียน Application ใหม่ด้วย

Chapter 24: 24.5 Name and Version Number

- IP Version ใหม่ เรียกอีกอย่างว่า 'IP The Next Generation'
  - บางทีเรียก IPng
- สำหรับ Version คือ Version 6 เนื่องจากชื่อ Version 5 นั้นได้ถูกใช้ไปแล้ว(เป็น Experiment Protocol)
- ดังนั้น IP ใหม่นี้จะถูกเรียกอีกอย่างว่า IPv6

Chapter 24: 24.6 IPv6 Features

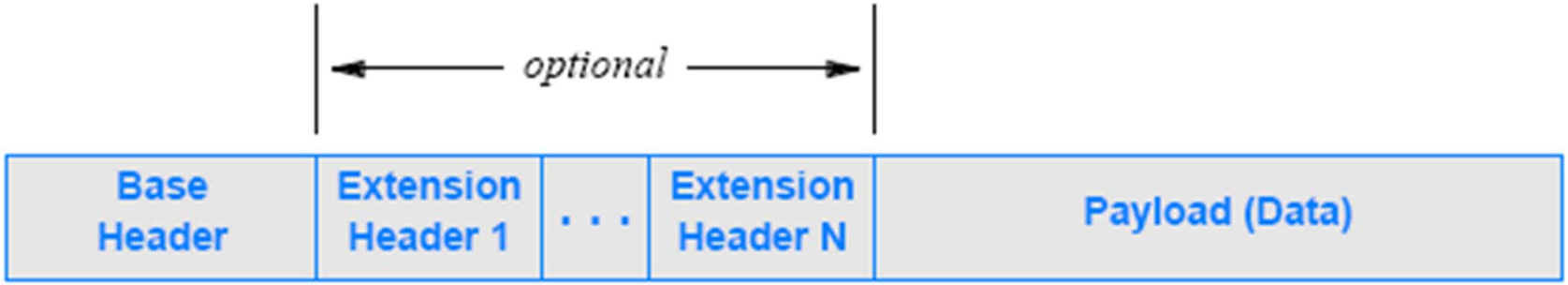
- **IPv6 ยังคงไว้ในคุณสมบัติของ IPv4 ที่ทำให้มันประสพผลสำเร็จขึ้นมากล่าวคือ**
  - ยังคงเป็น Datagram(Connectionless)
    - การส่ง Datagram จะมีการ Route ที่ไม่ขึ้นต่อกัน
  - ส่วนหัวยังมีการกำหนดจำนวน Hop สูงสุด
- **ลักษณะส่วนใหญ่ของ IPv4 แม้ว่าจะคงไว้ แต่จะมีรายละเอียดที่แตกต่างกัน**
- **Header จะมีขนาดคงที่ แต่ขยายได้โดยส่วนขยายถือเป็นอีก Header หนึ่ง(Extension) แยกออกจากกัน**
  - ผิดกับ IPv4 ที่ส่วนขยายของ Header จะผนวกกับ Header เดิม(Option) ทำให้ขนาดของ Header เพิ่มขึ้น

Chapter 24: 24.6 IPv6 Features

- **ลักษณะที่เพิ่มมาใน IPv6 สรุปได้ดังนี้**
  - Address Size
    - เพิ่มจาก 32 บิต เป็น 128 บิต
  - Header Format
    - จะแตกต่างจากรูปแบบเดิม โดย Field จะไม่เหมือนเดิม (ดูรายละเอียดต่อไป)
  - Extension Header
    - แต่ละข้อมูลของ IPv6 จะถูกใส่ใน Header ที่แยกจากกัน โดยที่ Datagram จะประกอบด้วย Base Header ตามด้วย Extension Header (ถ้ามี) ได้อีกหลายอัน
  - Support for Real-Time Traffic
    - จะมีกลไกในการกำหนดเส้นทางพิเศษสำหรับส่งข้อมูลเหล่านี้ (คือทำ QoS)
  - Extensible Protocol
    - IPv6 ยอมให้มีการเพิ่มคุณลักษณะของ Datagram ได้ในภายหลัง ทำให้มีความยืดหยุ่นในการใช้งาน

Ch. 24: 24.7 IPv6 Datagram Format

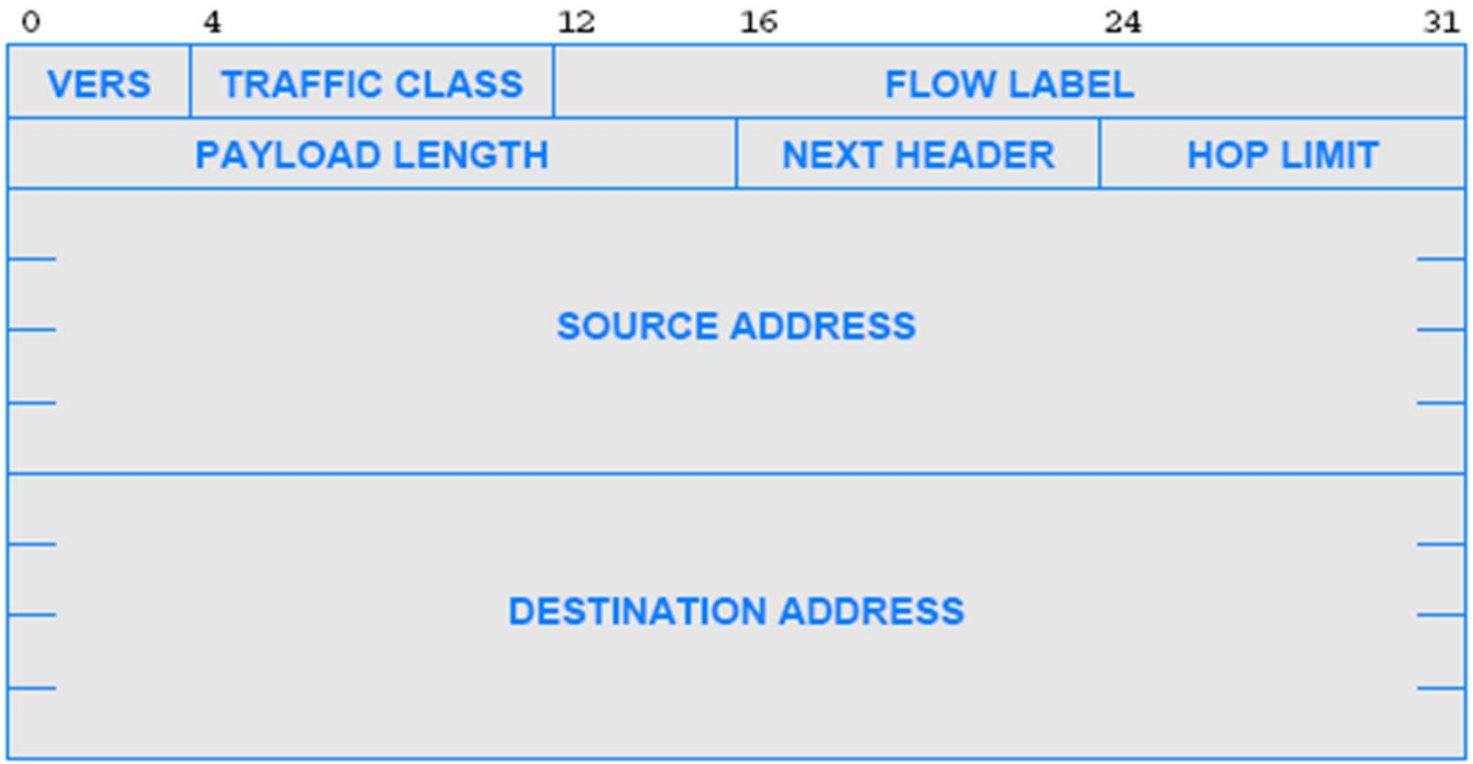
- **IPv6 Datagram ประกอบด้วย Header ต่างๆ ที่ต่อกัน เริ่มจาก Base Header ตามด้วย ศูนย์หรือมากกว่าของ Extension Header**
  - Base Header จะมีขนาดคงที่
  - Extension Header จะมีขนาดตามชนิดของ Extension



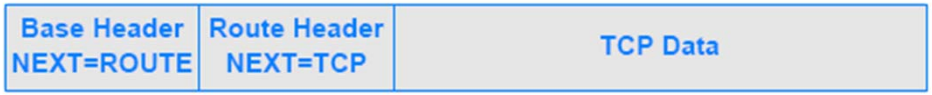
Ch. 24: 24.8 IPv6 Base Header Format

- มีขนาด **40 Octet(10 32-Bit Words)**
  - สองเท่าของ IPv4
- แต่ละ **Field** มีดังนี้
  - Vers 4 bit จะมีค่า 0x0110
  - Traffic Class กำหนดชนิดของ Traffic ที่ส่ง รู้จักกันในนาม Differentiated Service
    - เราสามารถแยกชนิดของข้อมูลเพื่อให้ Network ทำการส่งข้อมูลตามความเหมาะสม เช่น Interactive Traffic, Real-Time Traffic เป็นต้น จะกล่าวอีกครั้งในบทหลัง
  - Payload Length กำหนดความยาวของ Payload จะต่างจาก IPv4 คือจะเป็นเฉพาะความยาวของข้อมูล ไม่รวมส่วนหัว
  - Time-to-Live กำหนด Hop Limit
  - Flow Label กำหนดเส้นทางในการส่งผ่านแต่ละ Network คล้าย Virtual Circuit ปกติไม่ได้ใช่งาน
  - IP Address ต้นทางและปลายทางขนาด 128 บิต
  - Next Header กำหนดว่า Extension Header ต่อไปเป็นอะไร ถ้าไม่มีจะกำหนดชนิดของ Data(Payload) ที่อยู่ถัดไป (ดูรูปประกอบ)

Ch. 24: 24.8 IPv6 Base Header Format



(a)



(b)

Ch. 24: 24.9 Implicit and Explicit Header Size

- กรณีที่ Extension Header เป็นแบบที่มีขนาดคงที่ การตีความจะไม่กำกวม เราสามารถหาได้ว่าส่วนใหน เป็น Extension Header และส่วนใหนเป็น Payload ไม่ต้องแสดงขนาดของ Header
  - เป็น Implicit Header Size
- แต่ Extension Header บางประเภทจะมีขนาดไม่คงที่ ดังนั้น Header พวกนี้ต้องมีการบ่งบอกขนาดของมันว่าจะสิ้นสุดที่ใด
  - กำหนด Explicit Header Size

Ch. 24: 24.10 Fragmentation, Reassembly and Path MTU

- การทำ **Fragmentation** ของ **IPv6** จะคล้ายกับ **IPv4** คือมีการ **Copy** ส่วน **Header** ที่สำคัญ และมีการเปลี่ยนแปลงขนาดของ **Payload** ในแต่ละ **Fragment**
- แต่ **Base Header** ไม่มี **Field** ของ **Fragment Offset** และ **Flag** ดังนั้นข้อมูลของ **Fragment** ในแต่ละ **Fragment** จะถูกบรรจุใน **Extension Header** ที่เพิ่มขึ้นมา
- ที่ต่างจาก **IPv4** คือการทำ **Fragment** จะไม่กระทำที่ **Router** แต่ **Host** ที่ต้นทางจะทำ **Fragment**
  - ถ้า Fragment มีขนาดใหญ่กว่า MTU ของ Network ตัว Router จะทิ้ง Fragment นั้น และส่ง ICMP กลับมายังผู้ส่ง
    - ผู้ส่งต้องลดขนาด Fragment ลง

Ch. 24: 24.10 Fragmentation, Reassembly and Path MTU

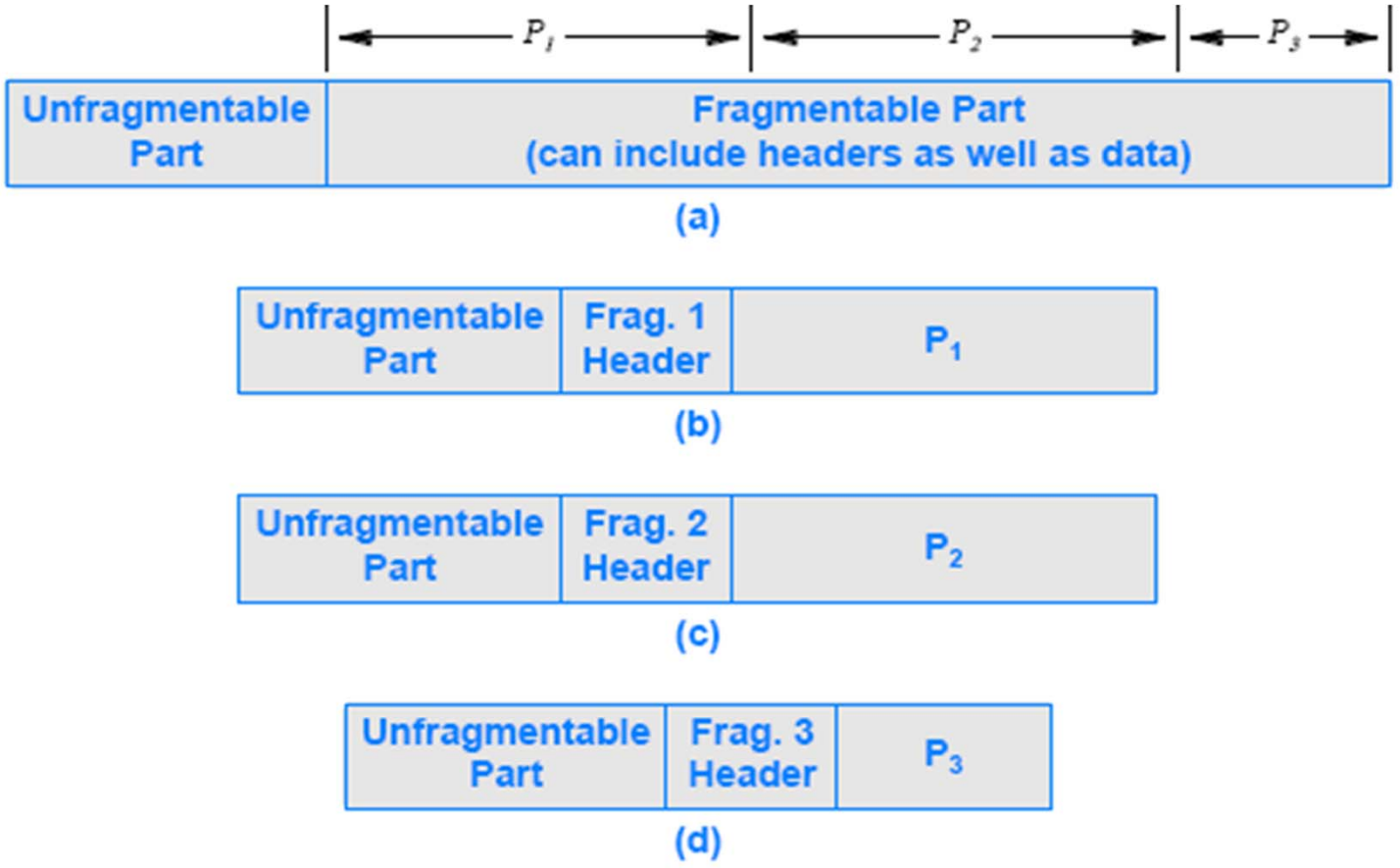


Figure 24.6 Illustration of IPv6 fragmentation with a datagram (a) divided into fragments (b) through (d).

Ch. 24: 24.10 Fragmentation, Reassembly and Path MTU

- ถ้า Fragment มีขนาดใหญ่กว่า MTU ของ Network ตัว Router จะทิ้ง Fragment นั้น และส่ง ICMP กลับมายังผู้ส่ง
  - ผู้ส่งต้องลดขนาด Fragment ลง เพราะ Router จะไม่ทำ
- **Host ผู้ส่งจะรู้ได้อย่างไรว่าควรจะส่ง Fragment ขนาดเท่าไร**
  - Host ที่ส่งจะต้องเรียนรู้ค่า MTU ของแต่ละ Network ที่ Datagram ต้องผ่าน และเลือกขนาด Fragment ที่เท่ากับค่า MTU ต่ำสุด
    - ค่านี้เรียก Path MTU
    - เรียนรู้ผ่าน Iterative Process ชื่อ “Path MTU Discovery”
      - ทำโดยการทดลองส่ง Fragment ที่มีค่าต่างๆกัน จนกว่าจะไม่มี Error

Ch. 24: 24.11 จุดประสงค์ของ Multiple Header

- มีเหตุผลสองประการ
  - Economy
    - การ Partition ข้อมูลของ Header เป็นส่วนต่างๆทำให้ประหยัดเนื้อที่ โดยตัดข้อมูลที่ไม่ต้องการออก เช่นการทำ Fragmentation ใน IPv4 แม้ว่า Datagram จะไม่มีการทำ Fragmentation แต่ส่วนหัวยังคงมี Field ที่เราจะต้องใส่ข้อมูลลงไปอยู่ สำหรับ IPv6 ถ้าไม่มีการทำ Fragment เราก็ไม่ต้องมี Fragment Extension Header
  - Extensibility
    - เราสามารถเพิ่ม Feature ของ IPv6 ในตอนหลังได้ โดยข้อมูลของ Feature ที่เพิ่มสามารถใส่ลงใน Extension Header โดยกำหนด Extension Header ชนิดใหม่ขึ้นมา ทำให้ตัว Protocol มีความยืดหยุ่น สามารถขยายได้
    - เราสามารถทดลองส่วนขยายนี้ก่อนที่จะกำหนดเป็น Protocol ก็ได้โดยยังม่ต้องปรับเปลี่ยนอุปกรณ์ Network ตราบใดที่ Extension Header ทดลองนี้อยู่ข้างหลัง Routing Header

Ch. 24: 24.12 IPv6 Addressing

- คล้ายกับ **IPv4** คือมีการกำหนดแต่ละ **Address** ในแต่ละ **Network Connection**
- มีการแบ่งส่วน **Address** เป็น **Prefix** และ **Suffix** เช่นกัน
- แต่รายละเอียดของการกำหนด **Address** จะต่างจาก **IPv4**
  - ใช้ CIDR คือ Prefix จะกำหนดอย่างไรก็ได้
  - แต่การแบ่ง Prefix จะสามารถกำหนดได้หลายลำดับชั้น (Hierarchy)
- **IPv6 กำหนด Special Address ที่ต่างจาก IPv4**
  - ไม่มี Special Address สำหรับการทำ Broadcasting ในแต่ละ Network (จะใช้การทำ Multicast แทน)
  - มีการเพิ่ม Anycast Address ขึ้นมา
    - เรียกอีกอย่างหนึ่งว่า Cluster Addressing
    - ข้อมูลจะส่งเพียงหนึ่ง Copy ให้กับใครก็ได้ใน Anycast Group
    - ต่างจาก Multicast ที่ข้อมูลจะส่งให้ทุกคนใน Group
    - ให้ในการกำหนด Replicate Service

Ch. 24: 24.12 IPv6 Addressing

Type	Purpose
unicast	The address corresponds to a single computer. A datagram sent to the address is routed along a shortest path to the computer.
multicast	The address corresponds to a set of computers, and membership in the set can change at any time. IPv6 delivers one copy of the datagram to each member of the set.
anycast	The address corresponds to a set of computers that share a common prefix. A datagram sent to the address is delivered to exactly one of the computers (e.g., the computer closest to the sender).

Figure 24.7 The three types of IPv6 addresses.

Ch. 24: 24.13 IPv6 Colon Hexadecimal Notation

- การใช้ **Dotted Decimal Notation** สำหรับ **128** บิต จะยาวเกินไป

  105.220.136.100.255.255.255.255.0.0.18.128.140.10.255.255

- ดังนั้นการเขียน **IPv6 Address** จะใช้เลขฐาน **16** กลุ่มละ **16** บิต คั่นด้วย **Colon**
  - Colon Hexadecimal Notation

  69DC : 8864 : FFFF : FFFF : 0 : 1280 : 8C0A : FFFF

Ch. 24: 24.13 IPv6 Colon Hexadecimal Notation

- **การเขียน IPv6 Address จะใช้เลขฐาน 16 กลุ่มละ 16 บิต คั่นด้วย Colon**
  - Colon Hexadecimal Notation
  - ในกรณีที่เป็นศูนย์ข้างหน้าเลขแต่ละกลุ่ม สามารถตัดทิ้งได้
  - *69DC : 8864 : FFFF : FFFF : 0 : 1280 : 8C0A : FFFF*
- **ในกรณีที่มีศูนย์หลายกลุ่มติดกัน สามารถทำ Zero Compression ได้**
  - ยุบศูนย์หลายๆตัว แทนด้วยสอง Colon(ทำได้ที่เดียว)
  - *เช่น FF0C:0:0:0:0:0:0:B1 → FF0C : : B1*
  - *Zero Compression ทำได้ที่ตำแหน่งเดียว*
- ***IPv6 Address ที่เริ่มด้วยศูนย์ 96 ตัว คือAddress ที่ Transition จาก IPv4 Protocol***
  - การ Transition จาก IPv4 → IPv6 ต้องใช้เวลา เราจะเห็นทั้งสอง Protocol นี้อยู่ร่วมกันระยะหนึ่ง ก่อนที่การ Transition จะสมบูรณ์
    - ไม่มีใครบอกได้ว่าเมื่อไร

Unicast Address Type

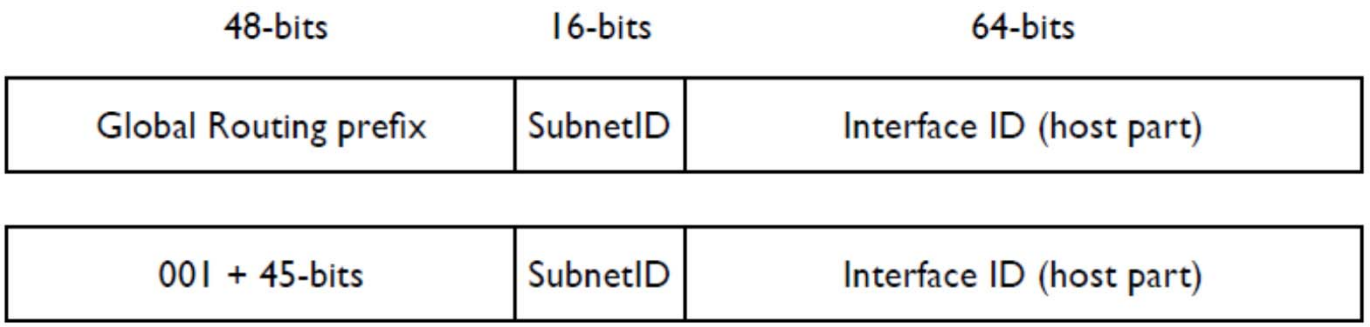
- **Global Unicast Address**
  - Static Address, Stateless Autoconfiguration, DHCP Assigned
  - Tunneled Address
  - Others
- **Link Local Address (FE80::/10)**
- **Unique Local Address(FC00::/7)**
- **Loopback (::1)**
- **Unspecified (::)**

Link Local Address

- ทุก IPv6 Network Interface จะมี Link Local Address
- เป็น Address ที่ใช้สำหรับการสื่อสารบน Local Subnet
- กำหนดจาก Auto-configuration โดยนำ FE80::/64 (รวมถึง Address ที่เริ่มจาก FE80,FE90,FEA0 และ FEB0, กล่าวคือ FE80::/10)มาต่อกับ Mac Address ที่ดัดแปลงในรูป EUI-64 (จะกล่าวต่อไป)
- สามารถใช้หมายเลขเดียวกันถ้ามีหลาย Interface และปกติจะแยกด้วย Scope-ID

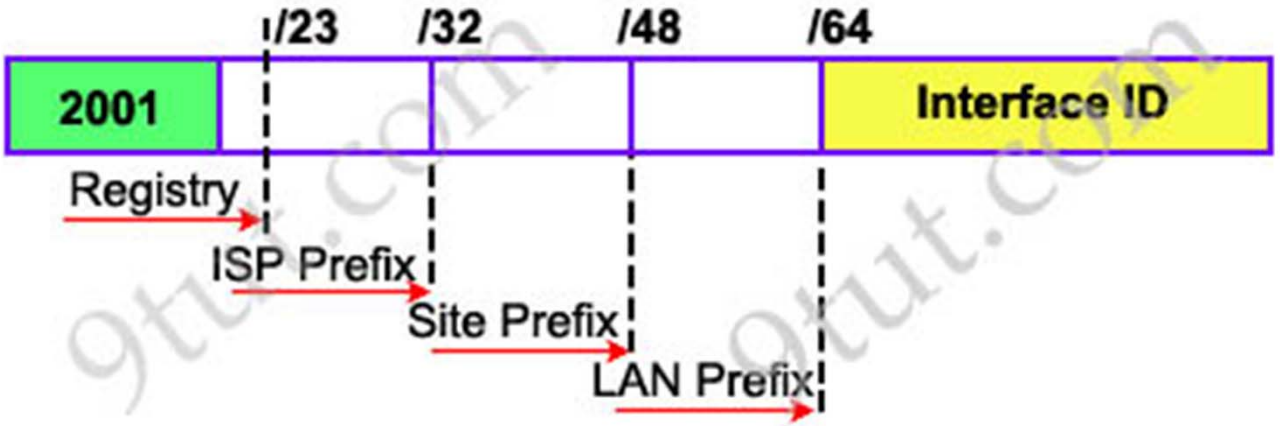
Global IPv6 Address

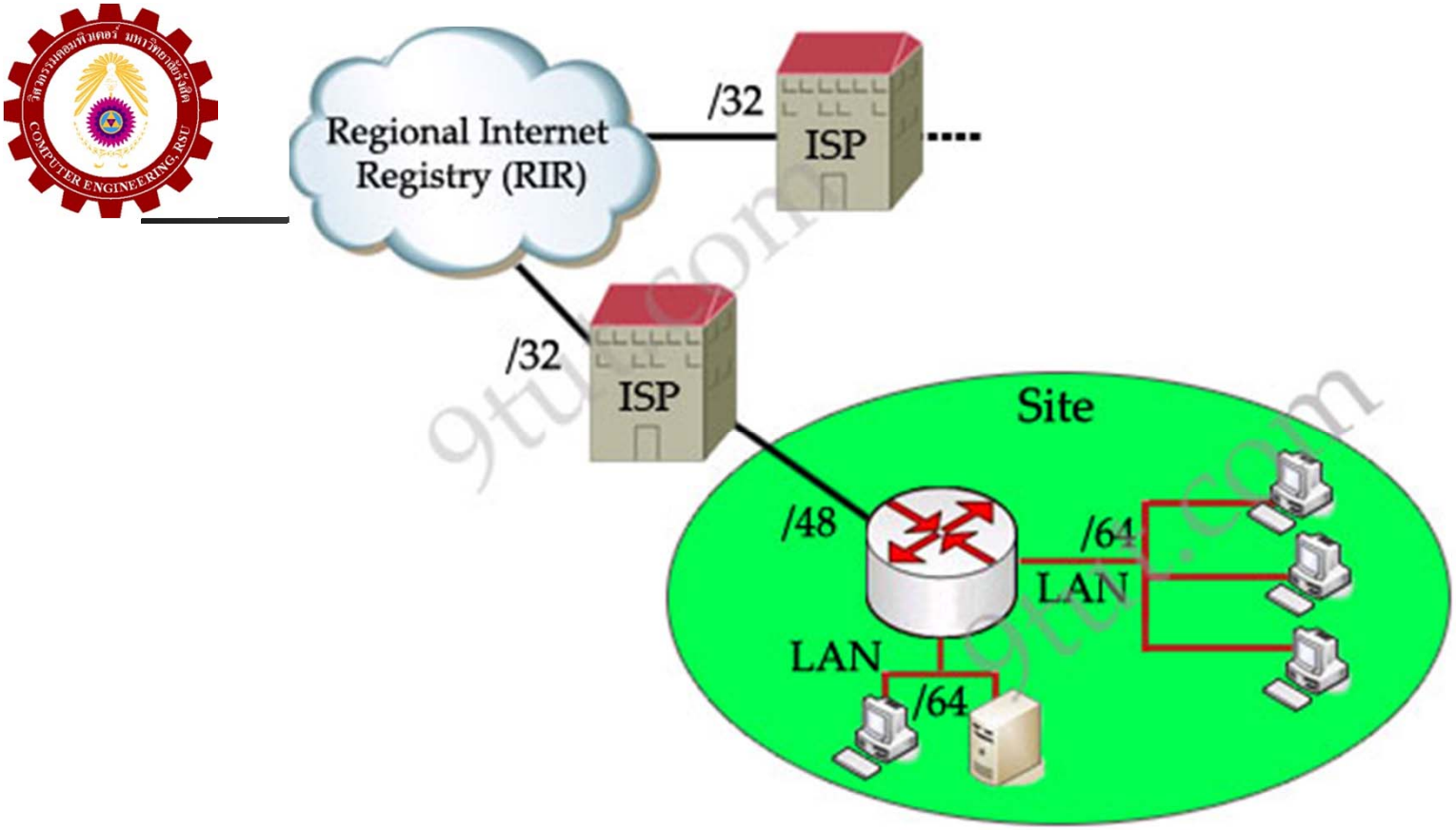
- Prefix 2000::/3 คือเริ่มจากบิต '001'
- ถัดมาอีก 45 บิตจะกำหนด Global Routing Prefix (IANA→RIR→ISP)
- ถัดมาคือ 16 บิตกำหนด Subnet ID
- 64 บิตสุดท้ายคือ Interface ID คือหมายเลข Host ใน Subnet (Host ID จะใช้ 64 บิต)
- Global Routing Information จะกำหนดจาก 64 บิต Prefix ไม่มากกว่านั้น (มียกเว้นกรณีพิเศษที่ใช้ /127)

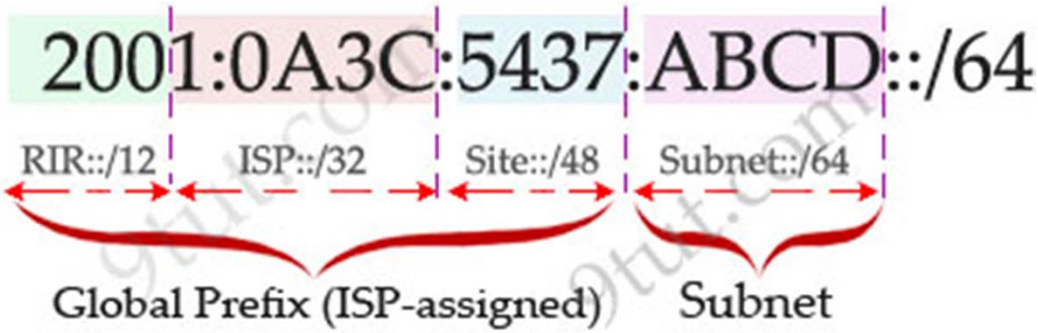


Global Address

- ICANN กำหนดช่วง IP Address สำหรับ Regional Internet Registry(RIR) ด้วย Prefix /12 (2000::/12 ถึง 200F:FFFF:FFFF:FFFF::/64)
- แต่ละ ISP จะได้รับ /32 และสามารถจ่าย /48 ให้กับแต่ละ Site
- แต่ละ Site สามารถใช้ /64 สำหรับแต่ละ LAN
- แต่ละ LAN จะมีได้ 2^{64} Interface

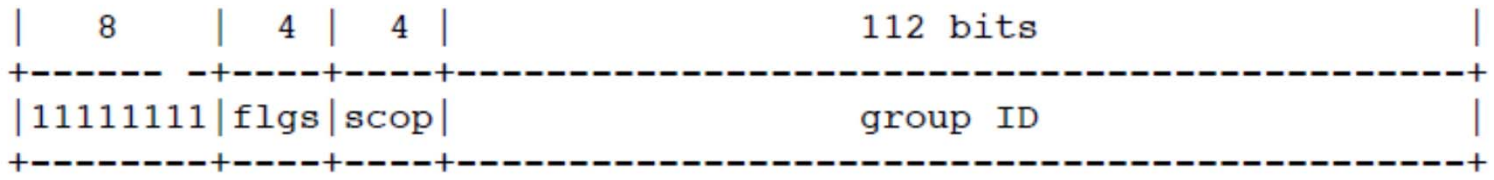



Regional Internet Registry (RIR)
/32
ISP
/32
ISP
Site
/48
/64
LAN
LAN
/64


2001:0A3C:5437:ABCD::/64
RIR::/12
ISP::/32
Site::/48
Subnet::/64
Global Prefix (ISP-assigned)
Subnet

Multicast Address

- ใช้ Prefix FF00::/8
- สามารถมี Scope ของการทำ Multicast ได้หลายรูปแบบ โดยดูจากส่วนของ Bit ใน Multicast Group
  - Link Local, Site, Global Scope
- ใช้ MLD (Multicast Listener Discovery) เทียบเท่า IGMP ใน IPv4

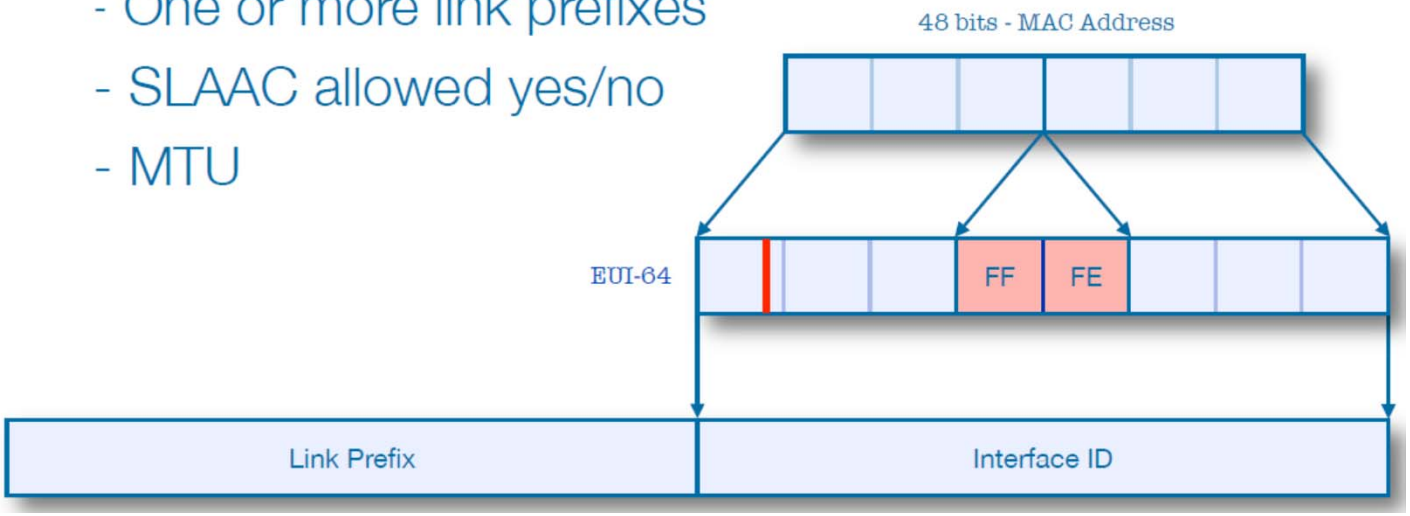


IPv6 บน LAN

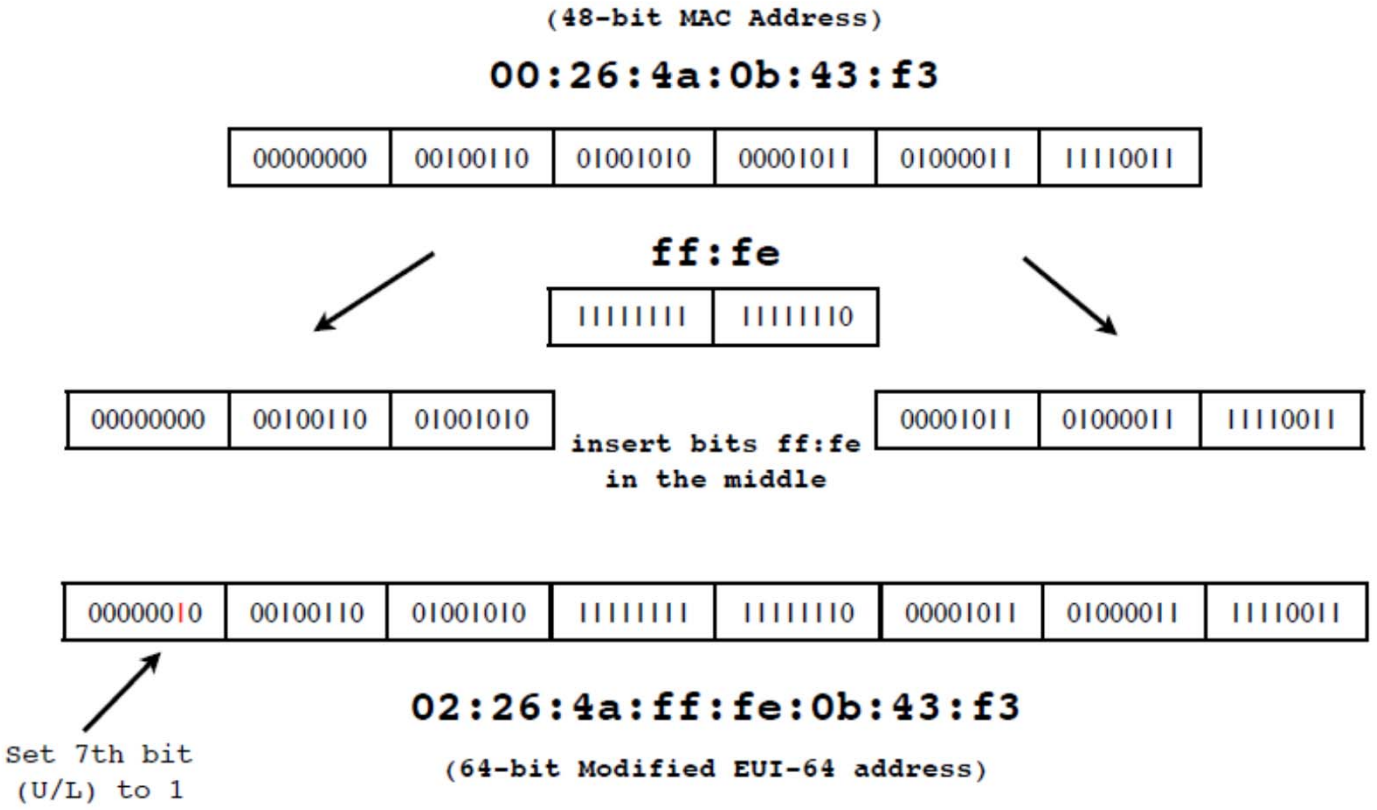
- สามารถทำ **Automatic Configuration**
- ใช้ **ICMPv6** นำ **Message** โดยอาศัยการทำ **Multicast**
- แบ่งเป็น **Stateless Address Autoconfiguration** ไม่ใช้ **DHCP**
  - Host สามารถกำหนด Address ให้แก่ตนเอง
- และ **Stateful** ใช้ **DHCP6**

Stateless Address Auto-configuration

- ฟังจาก **Router Advertisement** เพื่อที่จะรู้ **Prefix / 64** ของ **Subnet** ของตนเอง
- จากนั้นนำ **Prefix** ที่ได้ รวมกับ **Interface Address** ได้จาก **MAC Address** ในรูป **EUI-64**
  - Host will automatically start looking for a router
  - Response will contain:
    - Router's address
    - One or more link prefixes
    - SLAAC allowed yes/no
    - MTU



Modified EUI-64



Example

- RFC 4862: Stateless Address Autoconfiguration (SLAAC)
- Host listens to Router Advertisements (RA) on local subnet
- Obtains 64-bit subnet prefix from RA (and perhaps other parameters)
- Computes modified EUI-64 from its MAC address and concatenates it to 64-bit subnet prefix to form IPv6 address

```
Link prefix from RA: 2001:db8:abcd:1234::/64
Host MAC address: 00:1b:63:94:9d:73
EUI-64 address:   021b:63ff:fe94:9d73
Resulting IPv6 address:
   2001:db8:abcd:1234:021b:63ff:fe94:9d73
```

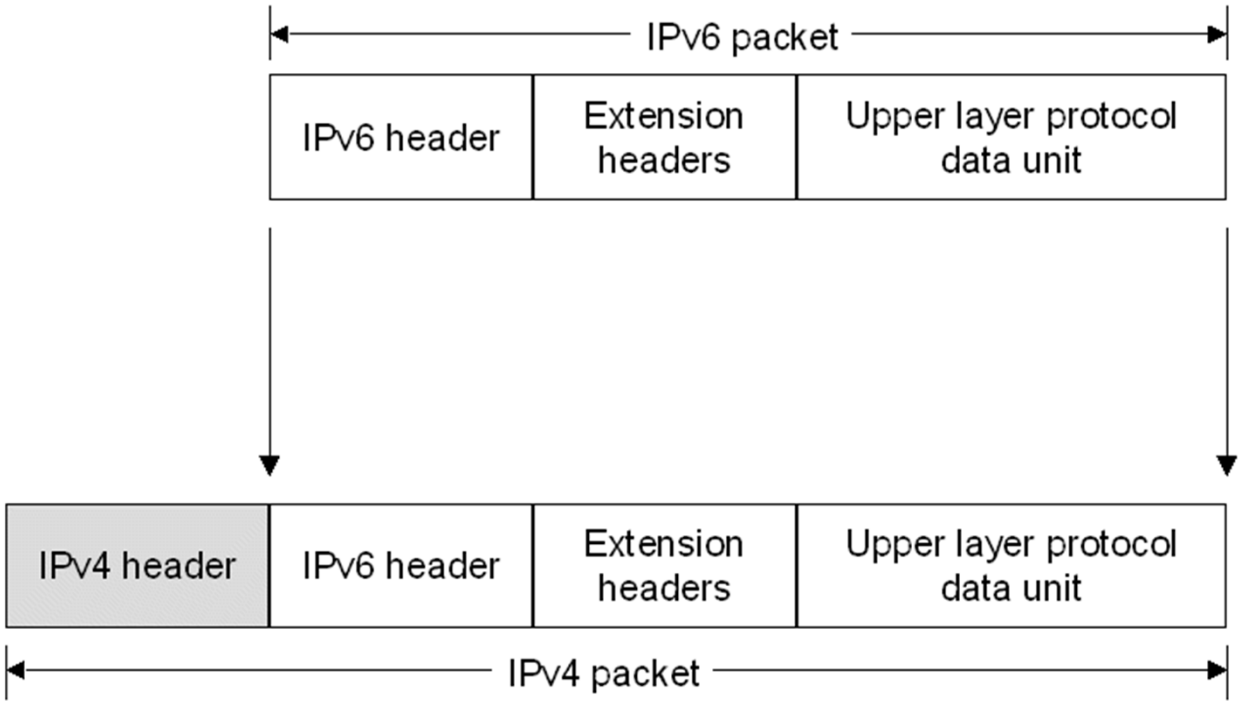
Transition Mechanism

- **Dual Stack Implementation**
  - Run ทั้ง IPv4 และ IPv6 บน Network เดียวกัน ดังนั้น Network เดียวจะ Support ทั้งสอง Protocol
  - ทำไม่ได้เสมอไป อุปกรณ์บางตัวอาจจะไม่ Support IPv6 หรืออาจจะต้องประสพปัญหายุ่งยากในการ Update Software/Firmware
- **Tunneling**
  - ใช้ในการเชื่อมต่อระหว่าง Network IPv6 กับ IPv4
  - ใช้ IPv4 Network ส่งผ่าน IPv6 Packet โดยการบรรจุ IPv6 Packet ลงใน IPv4 Packet (IP Protocol 41)
  - นิยมใช้การทำ Automatic Tunneling โดยที่เราสามารถทำ Tunnel ได้โดยไม่ต้องรู้ Tunneling End Point (จะหาเองโดยอัตโนมัติ)
    - บรรจุ IPv4 ลงใน IPv6 จากนั้นส่งไปที่ Relay Router ที่ต่อกับ IPv6 Network

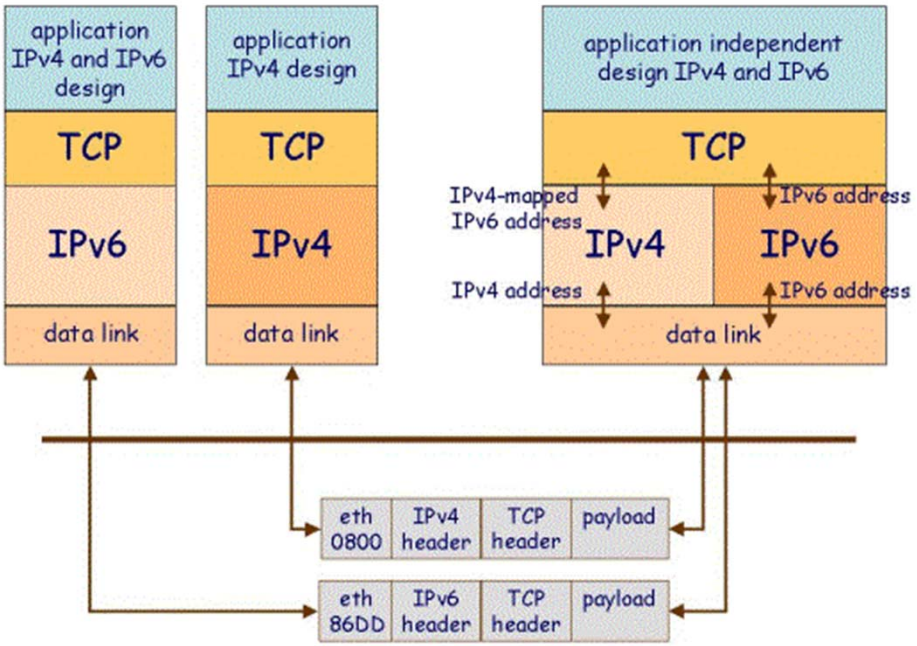
Automatic Tunneling

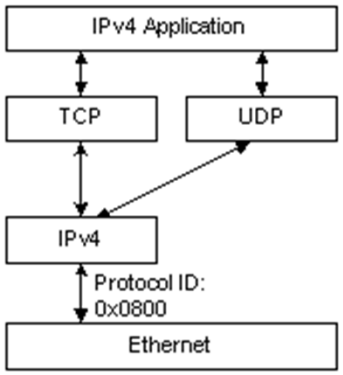
- **มีมาตรฐานหลายตัว สำหรับวิธีการ Transition จาก IPv4 เป็น IPv6**
  - จุดประสงค์เพื่อที่ IPv6 Host สามารถส่งข้อมูลถึง IPv6 Host หรือ IPv6 Network ผ่าน IPv4
    - Host จะเชื่อมต่อกับ IPv4
    - ดังนั้นจะ Run Dual Stack ที่ Host
      - Dual Stack สามารถทำได้ตั้งแต่ Window XP ขึ้นไป
- **เช่น4in6 ·6in4 ·6over4 ·DS-Lite ·6rd ·6to4 ·ISATAP ·NAT64 / DNS64 ·-Teredo ·SIIT**
  - ที่นิยมมี 2 แบบ: 6to4 และ Teredo ทั้งสอง Enable โดย Default ใน Window 7/Vista

IPv6 over IPv4 Tunneling

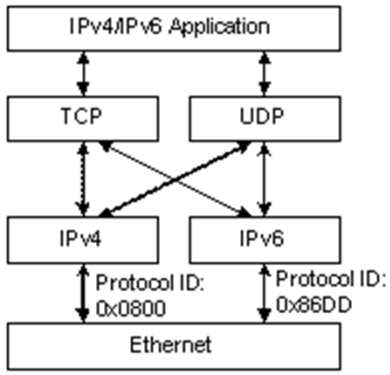


Dual Stack Host



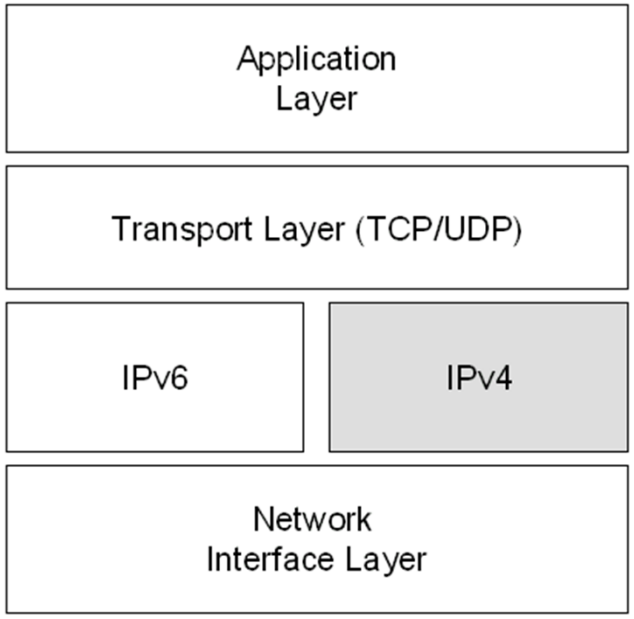


IPv4 Stack

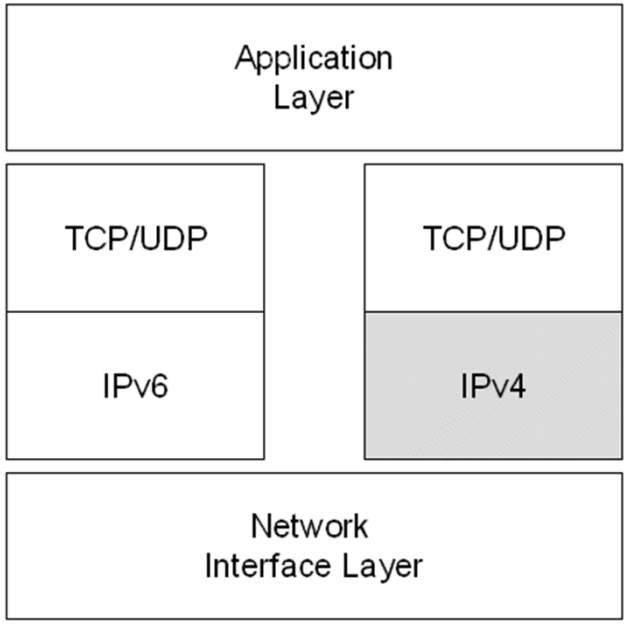


Dual Stack

Window Implementation



Dual IP Layer Architecture (Window Vista/2008)

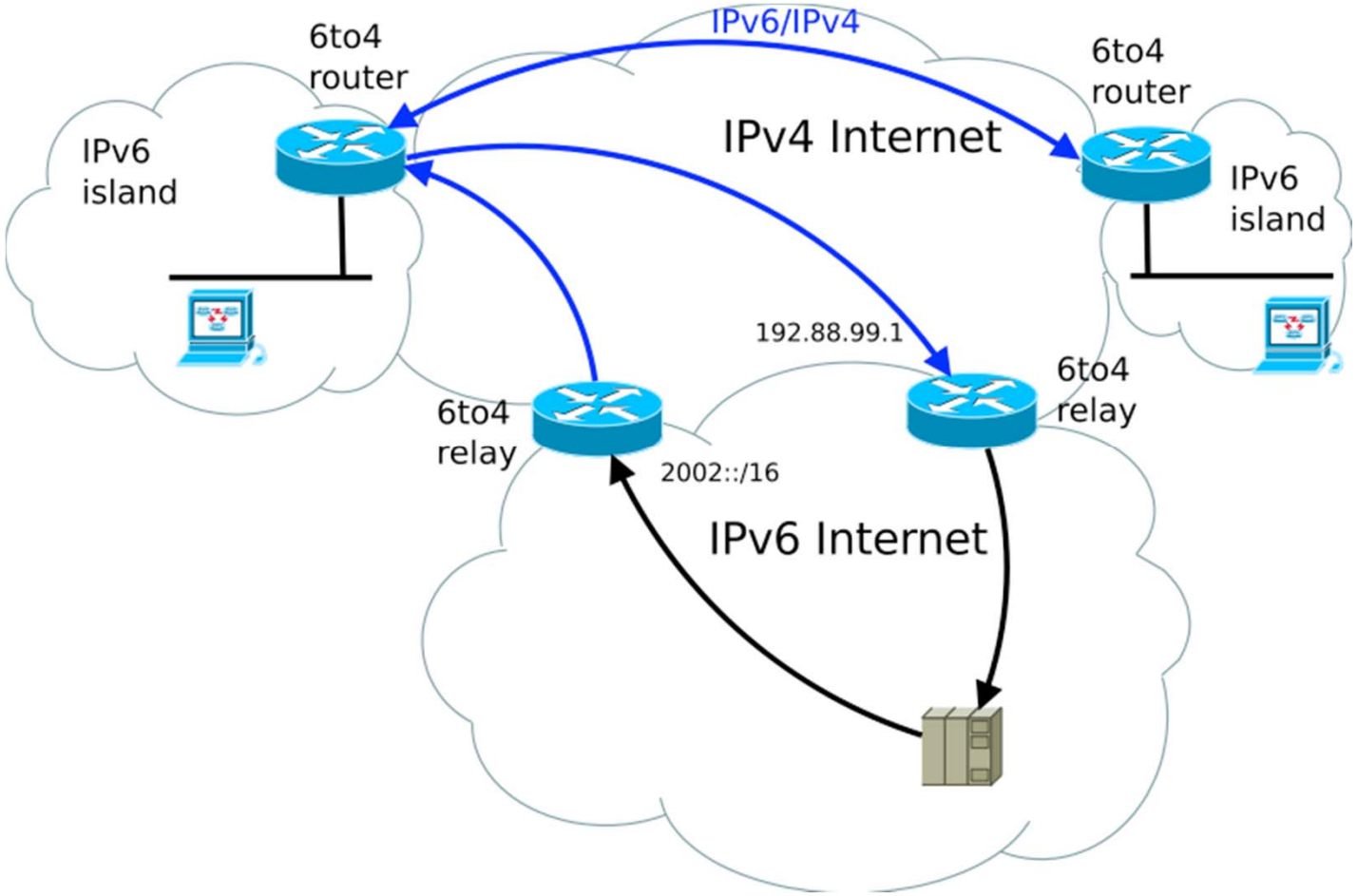


Dual Stack Architecture (Window XP, 2003)

Automatic Tunneling

- **4in6 ·6in4 ·6over4 ·DS-Lite ·6rd ·6to4 ·ISATAP ·NAT64 / DNS64 ·Teredo ·SIIT**
- **ที่นิยมมี 2 แบบ: 6to4 และ Teredo ทั้งสอง Enable โดย Default ใน Window 7/Vista**
- **6to4 (RFC3056)**
  - ใช้มากที่สุด โดยใช้ Protocol 41 ทำการ Encapsulate IPv6 ลงใน IPv4
  - 6to4 Host และ Network จะใช้ 2002::/16 Prefix ต่อผ่าน 6to4 Router, Host ถ้ามีเครื่องเดียวสามารถเป็น 6to4 Router ได้
  - 6to4 Relay Router จะอยู่ที่ขอบของ IPv4 Network และเชื่อมกับ IPv6 Network
    - Relay Router มักจะใช้ 6to4 Anycast Address
    - 192.88.99.1 และ 2002:C058:6301::
  - Site จะสร้าง /48 IPv6 Prefix โดยนำ IPv4 Address ของ 6to4 Router ไปต่อกับ 2002::/16 จากนั้นส่งผ่าน Tunnel จาก 6to4 Router ไปยัง 6to4 Relay

6to4 Tunneling



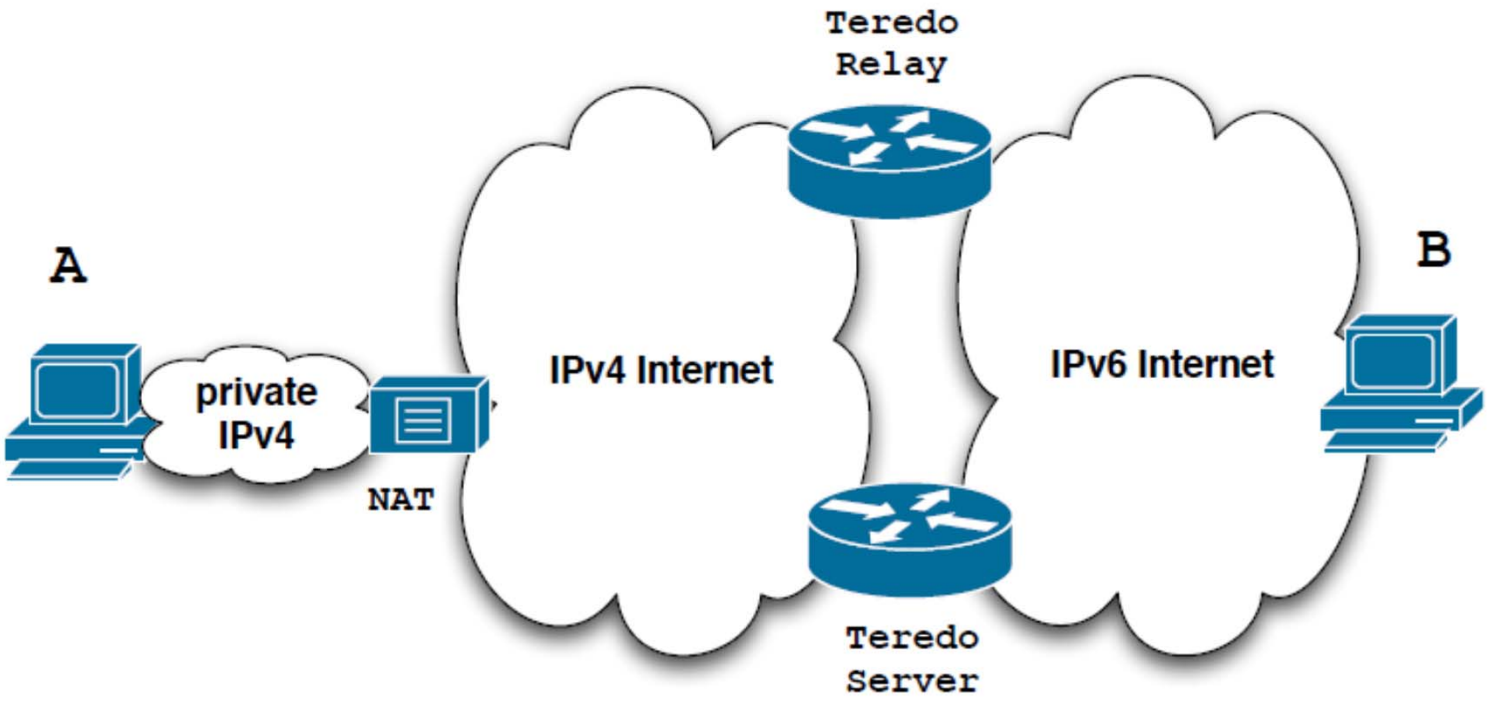
6to4 Tunneling

- Example of a single computer acting as a 6to4 router.
  - **IPv4 address:** 203.0.113.5 **(in hex: cb 00 71 05)**
  - **6to4 network prefix is:** 2002:cb00:7105::/48 **(2002::/16 + 32-bit IPv4)**
- Configure my IPv6 address as (subnet 1, interface-id 1)
  - **My IPv6 address:** 2002:cb00:7105:1::1
- 6to4 relay anycast IPv4 address: **192.88.99.1**
- 6to4 relay anycast IPv6 address: **2002:c058:6301::**
- To send a packet to 2001:db8:ab:cd::3, the computer encapsulates the IPv6 packet inside an IPv4 packet that is sent to the 6to4 relay IPv4 address:
  - **IPv4 src = 203.0.113.5 IPv4 dst = 192.88.99.1**
  - **IPv6 src = 2002:cb00:7105:1::1 IPv6 dst = 2001:db8:ab:cd::3**

Teredo Tunneling

- พัฒนาโดย Microsoft แต่ไม่ Support ใน Unix/Linux ส่วนใหญ่ (RFC4380)
- ทำงานโดยนำเอา IPv6 Packet บรรจุใน UDP จากนั้นจึงบรรจุลงใน IPv4 Packet
  - ข้อดีคือสามารถส่งผ่าน NAT ได้
- ใช้ Special IPv6 Prefix 2001::/32
  - ส่งข้อมูลผ่าน Teredo Relays และดูแลโดย Teredo Server

Toredo Diagram



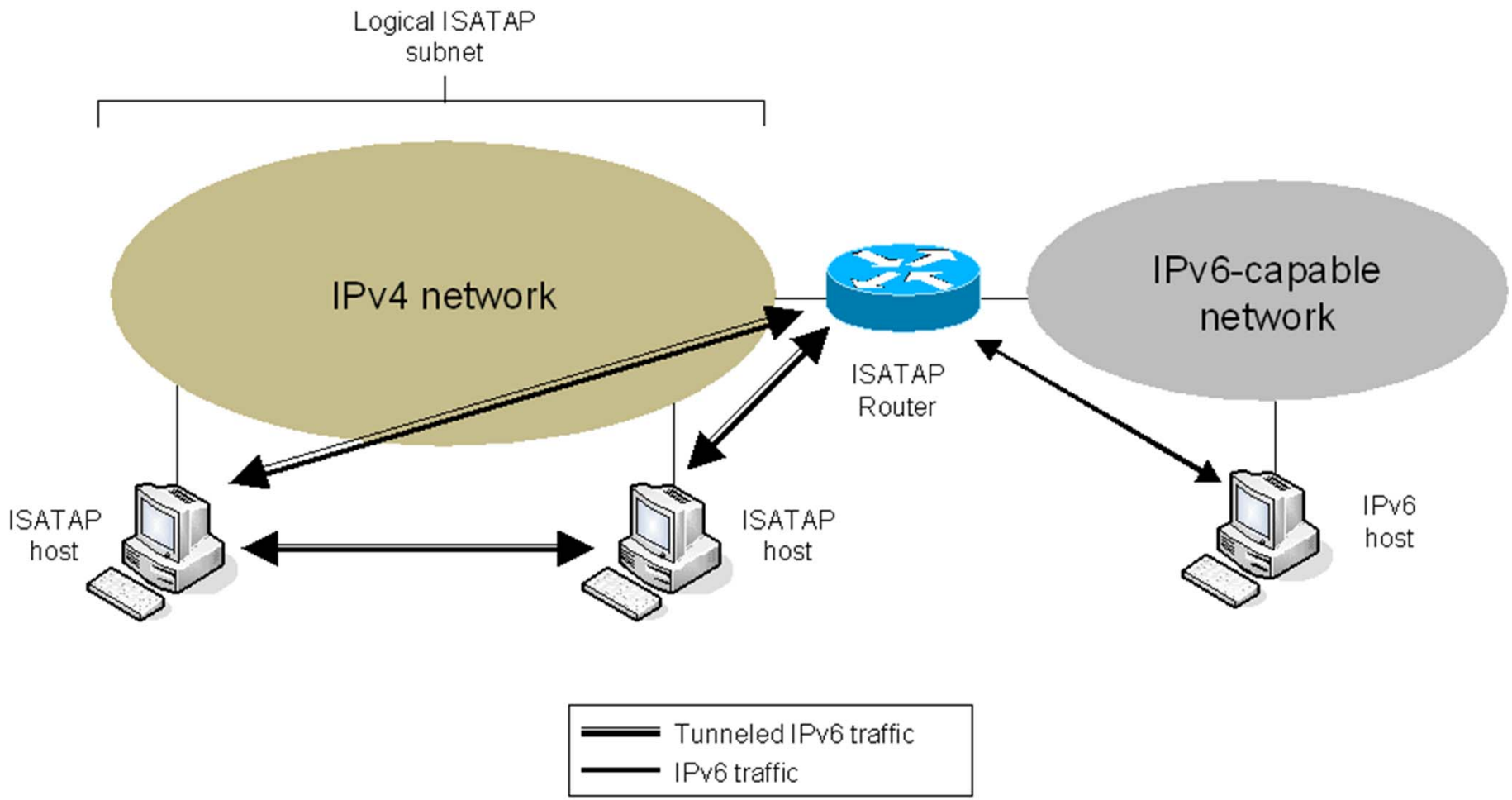
6to4 vs Teredo

- **6to4 ใช้ Prefix 2002::/16 ส่วน Teredo ใช้ 2001::/32**
- **ทั้งสองวิธีใช้การ Encapsulating IPv6 โดย**
  - 6to4 Encapsulate IPv6 ลงใน Payload IPv4 โดยตรง
  - Teredo จะ Encapsulate IPv6 ใน UDP ภายใน IPv4
- **6to4 ใช้ Well-Known anycast relay router (192.88.99.0/24) ส่วน Teredo ไม่ได้กำหนด**

ISATAP

- **เป็นการทำ Tunnel อีกแบบหนึ่งของ IPv6**
  - ใช้สำหรับรับ-ส่ง ข้อมูลระหว่าง IPv6 Host ผ่าน IPv4 Network
  - จะใช้ IPv4 ในลักษณะที่เป็น Non-broadcast หรือ NBMA (ไม่อาศัย ICMP6) ดังนั้นNeighbor Discovery ไม่จำเป็นต้องใช้
    - Router หาได้จาก PRL; Potential Router List และทำ Unicast-Only Autoconfiguration
  - Intra-Site Automatic Tunnel Addressing Protocol
  - บรรจุ IPv6 ลงใน IPv4 Packet โดยตรง
  - RFC 4214
- **ใช้การสร้าง Link-Local IPv6 Address จาก IPv4 Address**
  - FE80:0:0:0:200:5efe + IPv4 Address
  - EX: Host 192.0.2.143 จะมี Link-Local IPv6 Address เป็น
    - fe80:0:0:0:200:5efe:192.0.2.143 หรือ
    - fe80:0:0:0:200:5efe:c000:28f

ISATAP


Logical ISATAP subnet
IPv4 network
IPv6-capable network
ISATAP Router
ISATAP host
ISATAP host
IPv6 host
Tunneled IPv6 traffic
IPv6 traffic

End of Chapter 24

- **Homework 5 (Week 6) Download**
  - ส่งวันอังคาร ก่อนเที่ยง
  - ใส่กล่อง ที่ห้องสาขาวิชาเท่านั้น
  - กล่องอยู่ที่ Counter เลขาสาขา พี่หนึ่ง
  - ไม่รับการบ้านนอกเหนือจากนี้
  - เฉลยจะขึ้นบน Web ภายในวันศุกร์
- **สัปดาห์หน้า ตรงกับมาฆบูชา**
  - ไม่มีการเรียนการสอน

End of Week 6

- **No Class Week 7** (มาฆบูชา)
  - Week 8-9 Midterm
- **Week 10** หลัง **Midterm**
  - UDP+TCP
- **Prepare for MT**
- **MT Exam 35%**
  - F 4 March 2016; 13.30 – 16.00, 2-1/2 hr. ไม่มีการ Makeup Exam
  - 6 ข้อ ข้อละ 10 คะแนน รวม 60 คะแนน เก็บ 35%
  - ห้ามใช้เครื่องคิดเลข

MT Exam Preparation

- **6 ข้อ 60 คะแนน จะแยกเป็นเรื่องไป ดังนี้**
  - 1. LAN/LAN Technology อยู่ใน Slide Week 1-2
  - 2-3. IP Concept, IP Address/Subneting อยู่ใน Slide Week 3-4
  - 4. IP Forwarding และ IP Datagram อยู่ใน Slide Wk 4
  - 5. Supporting Protocols อยู่ใน Slide Week 5
  - 6. IPv6 อยู่ใน Slide Week 6
- **จะมีข้อย่อย ให้คำนวณหรืออธิบายสั้นๆ หรือเติมคำ**
  - ถ้าเป็นอธิบาย ต้องเขียนด้วยลายมือที่อ่านได้ จะไม่มีการเดาในการตรวจ ถ้าอ่านไม่ออกคือไม่ได้คะแนน
  - นักศึกษาควรจะได้ 15-20% จาก 35% เพื่อจะผ่านวิชานี้ ทั้งนี้ขึ้นกับการสอบ Final ด้วย

CPE 426 Computer Networks

Chapter 7:
Text Chapter 25: UDP

TOPICS

- **Chapter 25: UDP** (User Datagram Protocol)
  - Connectionless and Protocol
  - Communication Semantics and Port Numbers
  - UDP Format and Encapsulation
- **Chapter 26: TCP (Introduction)**

Chapter 25: UDP; Datagram Transport Service

- **ในกรณีที่ Application ต้องการใช้ Service ของ Connectionless โดยที่ไม่ต้องการ Service ใดๆ เพิ่มเติม จะสามารถใช้ IP ได้โดยตรง แต่ยังมีปัญหาสองประการ**
  - Application เป็น Layer 5 แต่ IP เป็น Layer 3 นำมาวางโดยตรงไม่ได้
  - แต่ละ Host อาจจะมีหลาย Application ที่ต้องการ Service จาก IP เราต้องมีการกำหนดหมายเลข Application
- **กล่าวคือ เราต้องการ Layer 4 Protocol บางๆ ไม่ทำหน้าที่อะไรมาก แค่เป็นการเชื่อมต่อระหว่าง Application Layer กับ IP Layer และมีการกำหนดหมายเลข Application**
  - หมายเลขที่กำหนด เราเรียก Port Number
- **Protocol นี้คือ User Datagram Protocol (UDP)**

Ch. 25: 25.2 Transport Protocol and End-to-end Communication

- **IP ไม่มี Mechanism ในการแยกแยะระหว่างหลายๆ Application ที่ส่งในระดับ End-to-end**
  - Field ใน IP Header กำหนดแค่ IP Address คือหมายเลขของ Host เท่านั้น
  - Host คือ End Point ในมุมมองของ IP
- **เราต้องการ Transport Layer Protocol**
  - End Point ของ Transport Protocol คือ Application
  - ดังนั้น End-to-End Protocol ที่แท้จริงใน TCP/IP จะอยู่ใน Transport Protocol แยกต่างจาก IP เป็น Layer 4

Ch. 25: 25.3 The User Datagram Protocol

- **TCP/IP จะมีสอง Transport Protocol**
  - User Datagram Protocol(UDP) (Protocol 17; 0x11)
    - ไม่มีความสลับซับซ้อน และง่าย
    - แต่จะไม่มี Service อะไรมากนักให้กับผู้ใช้
  - Transport Control Protocol(TCP)
- **UDP จัดว่าเป็น**
  - End-to-end สามารถแยกแต่ละ Application ได้
  - Connectionless เชื่อมต่อโดยไม่ต้องทำ Connection
  - Message-Oriented แต่ละข้อมูลที่ Application ส่งจะถือว่าเป็นหนึ่ง Message ไม่มีการทำ Segmentation
  - Best-Effort ใช้ IP ในการส่งโดยไม่มี Mechanism เพิ่มเติม
  - Arbitrary Interaction แต่ละ Application สามารถส่งให้กับหลาย และรับได้จากหลาย Application
  - OS Independent

Ch. 25: 25.4 Connectionless Paradigm

- **UDP ถือว่าเป็น Thin Protocol Layer เชื่อมต่อระหว่างหลาย Application โดยการส่งข้อมูลผ่าน IP**
- **ใช้วิธีการเชื่อมต่อกับ Network แบบ Connectionless**
  - Application ไม่ต้องทำ Connection ก่อนจะส่ง Message และเมื่อส่งจบ ไม่ต้องบอก Network ในการจบการสื่อสาร อยากจะหยุดเมื่อไร ก็หยุดได้เลย
  - ไม่มีการบันทึก State ของการสื่อสาร
    - ไม่มีการส่ง Control Message
  - ดังนั้นจะมี Overhead ต่ำมาก

Ch. 25: 25.5 Message Oriented Interface

- **Application ส่ง Block ของ Data โดยที่ UDP จะมองว่าเป็นหนึ่ง Message และจะถูกส่งไปทั้ง Message โดยประกอบใน Packet เดียว ไม่มีการแบ่ง หรือรวม**
- **ดังนั้นไม่ต้องกังวลเรื่องการประกอบ Data กลับคืนมา หรือหาขอบเขตของข้อมูล**
  - แต่ถ้า Message มีขนาดเล็ก จะมี Overhead จาก Header ต่างๆสูง
  - กลับกัน ถ้า Message มีขนาดใหญ่กว่า MTU จะเกิดการทำ Fragmentation ในชั้น IP
    - Fragment อาจจะมีการกระทำตั้งแต่ที่ Host ทำให้ Efficiency ลดลง
    - ในทางปฏิบัติ ผู้เขียน Program ควรกำหนดขนาด Message ไม่เกิน 1400 หรือ 1450 ในการส่งผ่าน Ethernet

Ch. 25: 25.6 UDP Communication Semantics

- **เนื่องจาก UDP จะใช้ IP โดยไม่มี Service อะไรเพิ่มเติม ดังนั้นมันจะส่ง Message แบบ Best-Effort เช่นเดียวกับ IP นั่นหมายถึงว่า Message ที่ส่ง อาจจะเกิด (เรียกว่า Communication Semantics)**
  - สูญหาย
  - Duplicate
  - Delayed
  - Delivery Out-of-Order
  - Corrupted
- **UDP จะไม่มีการตรวจสอบ หรือพยายามแก้ไขปัญหาต่างๆเหล่านี้**
- **หมายความว่า ผู้เขียน Application ที่ต้องการใช้ UDP จะต้องเขียน Mechanism ที่จะจัดการกับสิ่งเหล่านี้เองใน Software ถ้าต้องการ**
  - ถ้าเป็นพวก Real-time Audio อาจจะไม่ต้องทำ แต่ถ้าเป็นข้อมูลสำคัญ ต้องทำเอง หรือหันไปใช้ TCP ซึ่งจะกล่าวในบทหน้า

Ch. 25: 25.7 Mode of Interaction and Broadcast Delivery

- **UDP จะยอมให้มีการส่งข้อมูลได้ 4 แบบ**
  - 1-to-1
  - 1-to-Many
    - หนึ่ง Application สามารถส่ง Message ให้หลาย Application
    - UDP ยอมให้ Application ส่ง Message ผ่าน Multicast หรือ Broadcast
  - Many-to-1
    - หนึ่ง Application สามารถรับ Message จากหลายผู้ส่ง
  - Many-to-Many
    - กลุ่มของ Application สามารถแลกเปลี่ยนข้อมูลกับกลุ่มของ Application

Ch. 25: 25.8 End Point Identification with Protocol Port

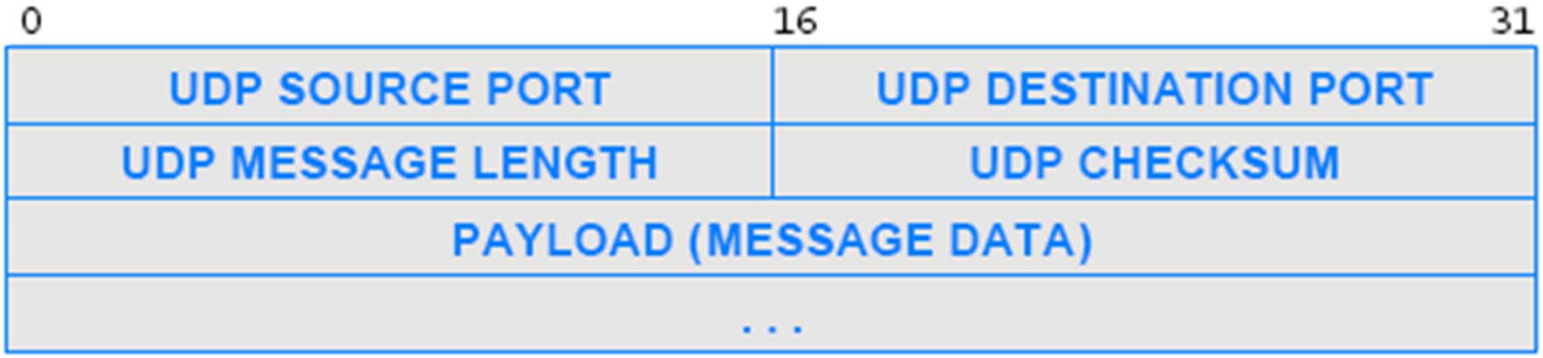
- **UDP จะต้องเป็น Independent กับ Operating System**
  - แต่ละ OS กำหนด Application ด้วย PID(Process ID)
  - แต่ละ OS มีวิธีการกำหนด PID ต่างกัน
  - UDP จะใช้ PID เป็นตัวกำหนด Application ไปได้ เพราะแต่ละเครื่องจะใช้แตกต่างกัน แม้ว่าจะเป็น Application เดียวกันก็ตาม
- **UDP จะกำหนด Application ด้วย Port Number ซึ่งจะ Map เข้ากับ PID ของแต่ละ Host อีกทีหนึ่ง**
  - เช่น Echo Service ใช้ Port 7 หรือ Timeserver Service ใช้ Port 36
  - Port Number จะ Independent กับ OS ของ Host
- **การสื่อสารกระทำโดย Application จะขอ Service ผ่าน Socket โดยกำหนด Address และ Port Number ให้กับ Socket**
  - 1-to-1 Application กำหนด Local Port, Remote IP Address, Remote Port
  - Many-to-1 Application กำหนด Local Port แต่จะบอก UDP ว่า Message จะมาจากที่ใดก็ได้ ดังนั้นทุกๆ Message ที่มีปลายทางที่ Port นี้ จะถูกส่งมายัง Application นี้

Ch. 25: 25.8 End Point Identification with Protocol Port

- **UDP Port 16 Bit จาก 0 – 65535**
- **กำหนดการใช้งานโดย IANA(Internet Assigned Numbers Authority, ปัจจุบันเป็น Department ของ ICANN, Internet Corporation for Assigned Names and Numbers)**
  - Internet Assigned Numbers Authority
  - Port 0 Reserved แต่ใช้ได้สำหรับเป็น Port ส่งและไม่ต้องการรับ
  - Port 0 – 1023 สำหรับ Common, Well-Known Service
  - Port 1024 – 49151 เป็น Registered Port สำหรับ Application ที่ขึ้นทะเบียนไว้กับ IANA
  - Port 49152 – 65535 เป็น Dynamic Port ที่สามารถนำไปใช้ได้

Ch. 25: 25.9 UDP Datagram Format

- แต่ละ **UDP Message** เรียกว่า **User Datagram** ซึ่งจะถูกแบ่งเป็นสองส่วนคือ **Header** ขนาดสั้นๆ**(8 Octet)** และ **Payload**
  - UDP Message Length จะกำหนดขนาดของ Message รวม Header เป็น Byte



Ch. 25: 25.10 UDP Checksum และ Pseudo Header

- **ส่วน Checksum ของ UDP เป็น Option ถ้าไม่ใช้จะใส่ศูนย์ สำหรับ IPv4**
  - ถ้าเป็น IPv6 จะต้องมี Checksum
- **UDP Header จะไม่มีข้อมูลของ IP Address เพราะมันต้องบรรจุใน IP ที่บ่งบอก Address อยู่แล้ว**
  - แต่ถ้า IP Layer เกิดมีปัญหา ส่ง Message UDP มาผิดที่ตัว UDP จะไม่สามารถตรวจสอบได้ว่า Message ตั้งใจจะส่งมาให้ตัวเองหรือไม่
  - ดังนั้น Header Checksum จะคำนวนไม่ใช่เฉพาะส่วนหัวของ UDP แต่จะคำนวนจากส่วนหัวของมันบวกกับข้อมูลจาก IP Header ด้วย เรียก Pseudo Header

Ch. 25: 25.10 UDP Checksum และ Pseudo Header

- **ดังนั้น ถ้ามีการใช้ Checksum มันจะคำนวนจากส่วนหัว บวกด้วย IP Address, Protocol(Type) และ UDP Length จาก IP Header (Pseudo Header) และ Data**
  - การคำนวณจะนำแต่ละ 16 Bit Word มาทำ 1's Complement Sum ผลลัพธ์ที่ได้จะทำ 1's Complement อีกทีหนึ่ง ถ้าผลเป็นศูนย์จะทำ 1's Complement อีกครั้ง

bits	0 – 7	8 – 15	16 – 23	24 – 31
0	Source address			
32	Destination address			
64	Zeros	Protocol	UDP length	
96	Source Port		Destination Port	
128	Length		Checksum	
160	Data			

Pseudo Header

Pseudo Header สำหรับ IPv6

- **Next Header = UDP(0x11, 17)**

Offsets	Octet	0	1	2	3
Octet	Bit	0 – 7	8 – 15	16 – 23	24 – 31
0	0	Source address			
4	32				
8	64				
12	96				
16	128	Destination address			
20	160				
24	192				
28	224				
32	256	UDP length			
36	288	Zeros			Next Header
40	320	Source Port		Destination Port	
44	352	Length		Checksum	
48+	384+	Data			

Pseudo Header

Ch. 25: 25.11 UDP Encapsulation

- UDP บรรจุใน IP โดยใช้ Protocol เบอร์ 17 (0x11)

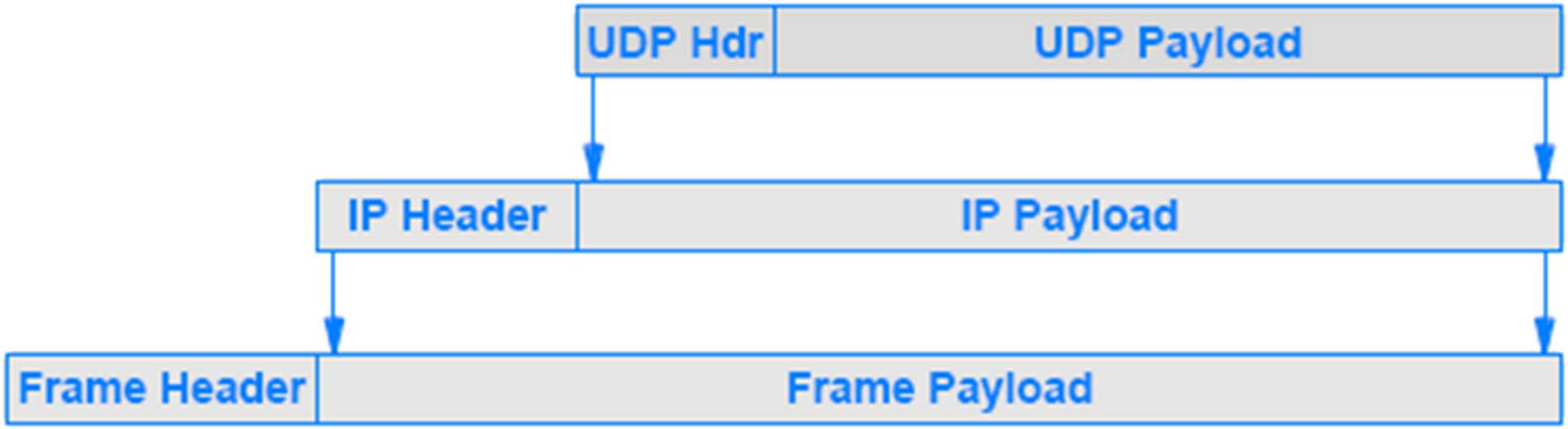


Figure 25.3 The encapsulation of a UDP message in an IP datagram.

การใช้งาน UDP

- **UDP** จะใช้กรณีที่ไม่เน้นในเรื่องความเชื่อถือได้ของการส่งข้อมูล แต่จะเน้นที่ **Overhead** ต่ำ และความรวดเร็ว
  - DNS (Domain Name Service)
  - DHCP (Dynamic Host Configuration Protocol)
  - SNMP (Simple Network Management Protocol)
  - RIP (Routing Information Protocol)
  - VoIP (Voice over IP)
  - Streaming Media (Real-time Video/Audio)
  - Online Games (MMORPG, **Massively multiplayer online role-playing games**)

Chapter 26

- TCP
- Transport Control Protocol
- Reliable Protocol for the Internet
- TCP/IP (TCP over IP)

Chapter 26 TCP: Reliable Transport Service

- **ในกรณีที่เราต้องการความมั่นใจในการส่งข้อมูลได้อย่างถูกต้อง ผ่าน IP ที่มีการทำงานแบบ Datagram เราต้องใส่ Mechanism เพื่อให้ความมั่นใจดังกล่าวลงใน Protocol ของ Layer 4 (Host-to-Host หรือ Transport)**
  - เราจะต้องเลือกใช้ TCP (Transmission Control Protocol)
  - เรียกรวมๆว่า TCP/IP
  - การใช้ TCP+IP ทำให้ผู้เขียน Application ไม่ต้องกังวลเรื่องของการส่งข้อมูลอีกต่อไป
    - เพียงแต่นำข้อมูลส่งให้ พร้อมกำหนด IP Address และ Port Number จากนั้น ระบบ TCP/IP จะจัดการที่เหลือให้
    - ผิดกับกรณีที่ใช้ UDP/IP ในบทที่แล้ว
    - กล่าวคือ TCP จะให้ Reliability ในการสื่อสาร

Chapter 26 TCP: 26.3 TCP Service

- **TCP ให้บริการของ 7 ส่วนใหญ่ๆ ดังนี้**
  - Connection Orientation
    - TCP ให้บริการแบบ Connection-Oriented ซึ่ง Application จะต้องร้องขอการเชื่อมต่อก่อนที่จะมีการส่งข้อมูล
  - Point-to-Point Communication
    - แต่ละ Connection ของ TCP จะมีสอง End Points เท่านั้น (กำหนด Port ต้นทาง-ปลายทาง และ IP ต้นทาง-ปลายทาง)
  - Complete Reliability
    - TCP จะ Guarantee ว่าข้อมูลที่ส่ง จะไปถึงที่หมายได้อย่างถูกต้อง สมบูรณ์ และเป็นลำดับ
  - Full Duplex Communication
    - แต่ละ TCP Connection จะยอมให้ข้อมูลส่งได้สองทิศทางตลอดเวลา

Chapter 26 TCP: 26.3 TCP Service(2)

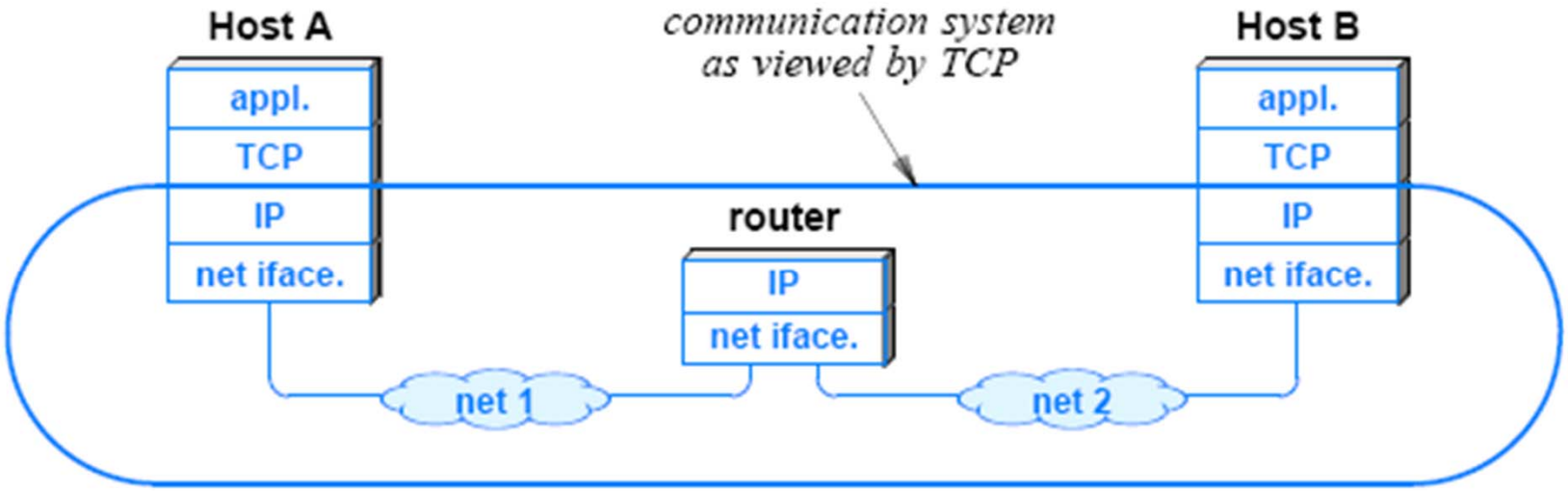
- **TCP ให้บริการของ 7 ส่วนใหญ่ๆ ดังนี้(ต่อ)**
  - Stream Interface
    - Application สามารถส่งข้อมูลได้อย่างต่อเนื่อง Octet ต่อ Octet ผ่าน TCP Connection โดย TCP จะไม่มีการรวมกลุ่มของ Data ให้เป็น Record หรือ Message และจะไม่ Guarantee ว่าแต่ละส่วนของข้อมูลที่ไปถึงมีขนาดเท่ากับที่ Application ส่งให้(มีการทำ Segmentation)
  - Reliable Connection Startup
    - TCP ให้สอง Application สามารถเริ่มต้นการสื่อสารได้อย่างมั่นใจ
  - Graceful Connection Shutdown
    - ก่อนที่จะจบ Connection นั้น TCP จะให้ความมั่นใจว่าข้อมูลได้ถูกส่งอย่างครบถ้วนทั้งสองฝั่ง และทั้งสองฝั่งยินยอมให้มีการจบการเชื่อมต่อ

Chapter 26 TCP: 26.4 End-to-End Service and Virtual Connection

- **TCP จัดว่าเป็น End-to-End Protocol เหมือน UDP**
  - เนื่องจากมันให้บริการการสื่อสารระหว่าง Application ของสอง Computer
- **แต่มันเป็น Connection-Oriented**
  - เนื่องจากต้องมีการทำ Connection ก่อนส่งข้อมูล
  - และต้องมีการทำ Disconnection
- **Connection ของ TCP จัดว่าเป็น Virtual Connection**
  - เนื่องจากกระทำผ่าน Software เพราะตัว Network (IP) นั้นเป็น Connectionless(Datagram)

Chapter 26 TCP: 26.4 End-to-End Service and Virtual Connection

- **แต่ละ TCP Message(Segment) จะถูกบรรจุใน IP Datagram และส่งผ่าน Internet เมื่อถึงปลายทาง IP จะปลอก Datagram Payload และส่งต่อไปยัง TCP**
  - IP จะมอง TCP เป็นแค่ Data ที่จะต้องส่ง
  - TCP จะมอง IP เป็นพาหนะสำหรับส่งข้อมูล ไปยัง TCP อีกฝั่งหนึ่ง



Chapter 26 TCP: 26.5 Transport Protocol Techniques

- **ปัญหาที่ End-to-End Transport Protocol จะต้องเจอ ในการได้มาซึ่ง Reliable Service**
  - Unreliable Communication
    - Message อาจจะสูญหาย ซ้ำซ้อน ผิดพลาด ถูกหน่วงเวลา หรือส่งมาไม่เป็นลำดับ (Lost, Duplicated, Corrupted, Delayed, Out of Order)
  - End System Reboot
    - ถ้า Host เกิด Crash หรือมีการ Reboot ข้อมูลที่ค้างอยู่ จะต้องไม่สับสนกับ Session ที่ถูกสร้างขึ้นมาใหม่
  - Heterogeneous End System
    - แต่ละ Host ที่เชื่อมต่อใน Internet มีความสามารถรับ-ส่งข้อมูลได้ไม่เท่ากัน
  - Congestion in the Internet
    - เมื่อมีการส่งข้อมูลมากเกินไป จะเกิดความคับคั่งภายใน Network ได้
- **ปกติแล้ว Transport Protocol จะใช้หลายๆวิธีร่วมกันในการจัดการกับปัญหาเหล่านี้ เราจะกล่าวพื้นฐานแต่ละอันต่อไป**

Chapter 26 TCP: 26.5.1 Sequencing to Handle Duplicates and Out-of-Order Delivery

- **Transport Protocol จะทำ Sequencing (กำหนดหมายเลขลำดับของข้อมูล) ในการจัดการกับปัญหาเรื่อง Duplicate Data และข้อมูลที่ไปถึงอย่างไม่เป็นลำดับ**
  - คล้ายกับวิธีที่กระทำใน Layer 2
  - หมายเลขข้อมูลทำให้เรารู้ว่าข้อมูลนั้นมาซ้ำกันหรือไม่
  - ทำให้รู้ว่าข้อมูลมาเป็นลำดับหรือไม่

Chapter 26 TCP: 26.5.2 Retransmission to Handle Lost Packet

- **ในการจัดการกับ Packet Lost นั้น Transport Protocol จะใช้ Positive Acknowledgement ร่วมกับการ Retransmission**
  - คล้ายกับ Mechanism ใน Layer 2 เช่นกัน
  - เมื่อ Frame มาถึงอย่างถูกต้อง จะมีการส่ง ACK Message กลับไป
  - เมื่อผู้ส่ง ส่งข้อมูลแต่ละ Packet จะมีการจับเวลาโดยใช้ Timer
    - ถ้า Timer Expire (คือไม่ได้รับ ACK ในเวลาที่กำหนด) ผู้ส่งจะทำการ Retransmission ข้อมูลนั้นไปใหม่
  - ถ้า Packet มีการ Delay มาก อาจจะทำให้เกิด Retransmission ยังผลให้เกิด Duplicate Packet

Chapter 26 TCP: 26.5.3 Techniques to Avoid Replay

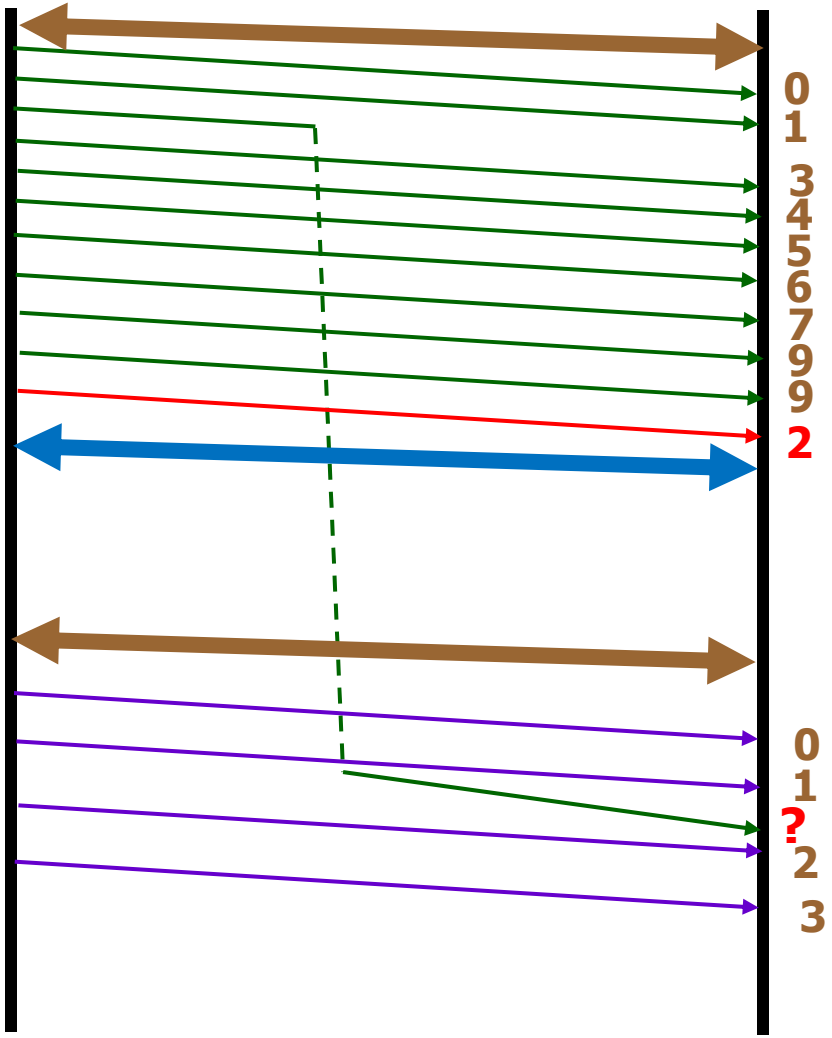
- **ในกรณีที่ Network มี Delay สูง อาจจะทำให้เกิด 'Replay Error' โดยที่ Packet ที่ Delay นั้นจะส่งผลต่อการสื่อสารในตอนหลัง**
  ยกตัวอย่าง
  - สองคอมพิวเตอร์ตกลงจะสื่อสารกันเมื่อเวลา 13.00 น.
  - คอมพิวเตอร์เครื่องหนึ่ง ส่ง 10 Packet ติดต่อกันไปยังฝั่งตรงข้าม
  - ปัญหาด้าน Hardware ทำให้ Packet 3 ถูก Delay ไป
  - เส้นทางการส่งข้อมูลถูกเปลี่ยน เพื่อใช้เส้นทางที่ไม่ผ่านอุปกรณ์ที่มีปัญหา
  - Protocol ที่คอมพิวเตอร์ต้นทางส่ง Packet 3 ไปใหม่ จากนั้นส่ง Packet อื่นๆตาม จากนั้นจบการสื่อสาร

Chapter 26 TCP: 26.5.3 Techniques to Avoid Replay

- สองคอมพิวเตอร์ตกลงจะสื่อสารกันเมื่อเวลา 13.00 น.
- คอมพิวเตอร์เครื่องหนึ่ง ส่ง 10 Packet ติดต่อกันไปยังฝั่งตรงข้าม
- ปัญหาด้าน Hardware ทำให้ Packet 3 ถูก Delay ไป
- เส้นทางการส่งข้อมูลถูกเปลี่ยน เพื่อใช้เส้นทางที่ไม่ผ่านอุปกรณ์ที่มีปัญหา
- Protocol ที่คอมพิวเตอร์ต้นทางส่ง Packet 3 ไปใหม่ จากนั้นส่ง Packet อื่นๆตาม จากนั้นจบการสื่อสาร
- เวลา 13.05 น. คอมพิวเตอร์ทั้งสองตกลงที่จะสื่อสารกันใหม่
- หลังจากผู้ส่งคนเดิม ส่งไปได้สอง Packet แล้ว ตัว Packet 3 จากการสื่อสารครั้งแรกที่ถูก Delay มาถึงยังผู้รับ
- ต่อจากนั้น Packet ที่ 3 จากการสื่อสารครั้งที่สองมาถึงยังผู้รับ

- **Transport Protocol ต้องออกแบบเพื่อจัดการกับเรื่องนี้ มิฉะนั้นผู้รับจะได้ Packet 3 ที่ผิดพลาด ในขณะที่โยนทิ้ง Packet ที่ถูกต้อง**
  - Protocol จะต้อง Mark แต่ละ Session โดยใช้ ID ที่เฉพาะ ไม่ซ้ำกัน โดยการนำ ID กลับมาใช้ใหม่ต้องให้แน่ใจว่าจะไม่เกิด Replay (เวลาต้องห่างกัน)

Replay (Selective Reject)



Chapter 26 TCP: 26.5.3 Techniques to Avoid Replay

- **Replay ยังสามารถเกิดกับ Control Packet ได้เช่นกัน**
  - เช่น Connection Closing Packet ถูก Delay และไปถึงหลังจากมีการทำ Connection ใหม่ครั้งที่สอง

Chapter 26 TCP: 26.5.4 Flow Control to Prevent Data Overrun

- **ในการจัดการเรื่อง Heterogeneous End System ที่มีความสามารถไม่เท่ากัน ตัว Transport Protocol จะใช้วิธีการของ Flow Control เพื่อควบคุม**
  - Stop-and-go Protocol มักจะไม่ใช้ เนื่องจากมีประสิทธิภาพต่ำ
  - ปกติจะใช้ Sliding Window Flowcontrol
  - เช่นเดียวกัน จะคล้ายกับ Flow Control ใน Layer 2
    - การคำนวณ Efficiency จะคล้ายกับที่กล่าวมาแล้วใน Layer 2
    - ให้ดูใน Slide ก่อนหน้า หรือดูจาก CPE 326 Course Notes

Chapter 26 TCP: 26.5.4 Flow Control to Prevent Data Overrun

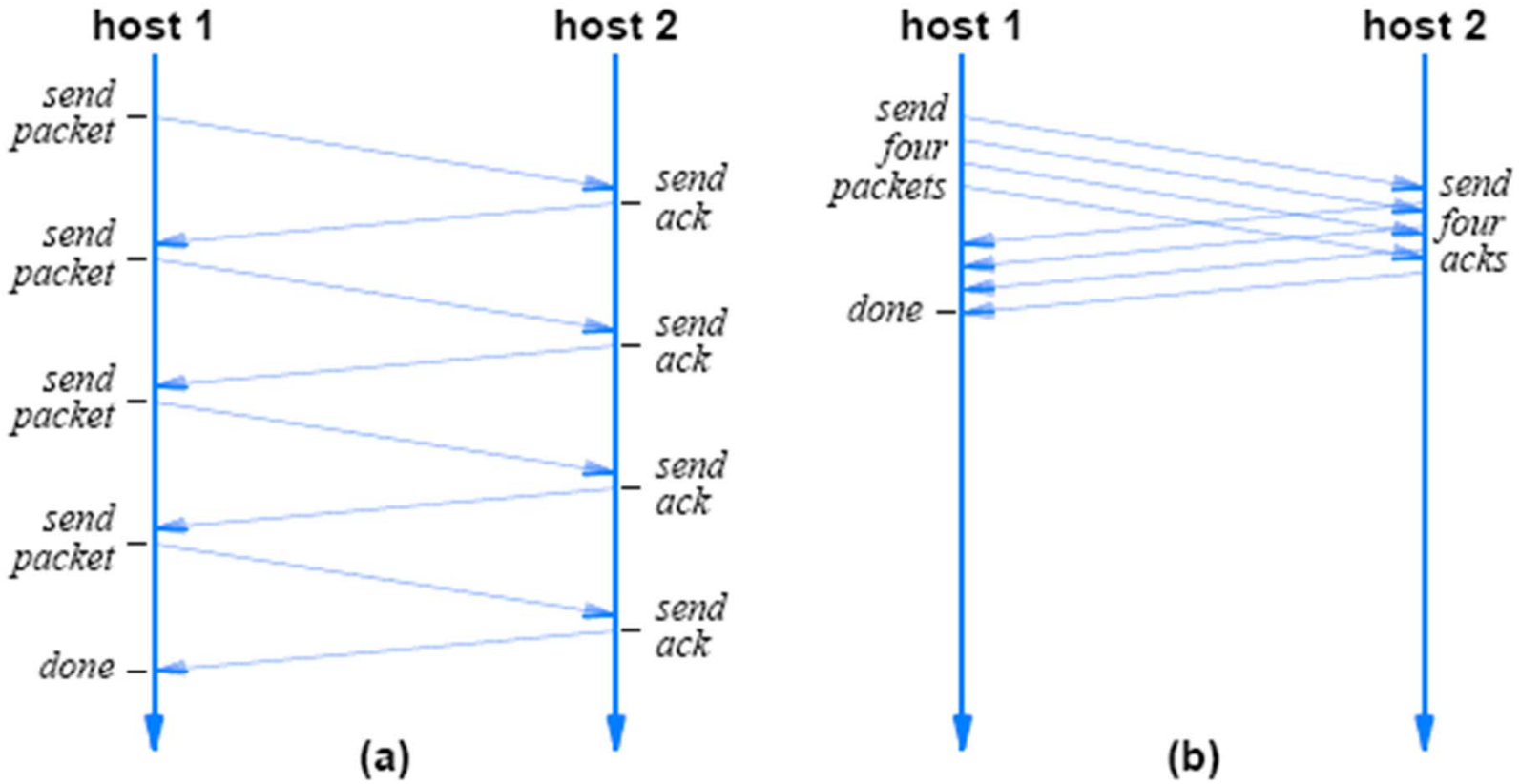
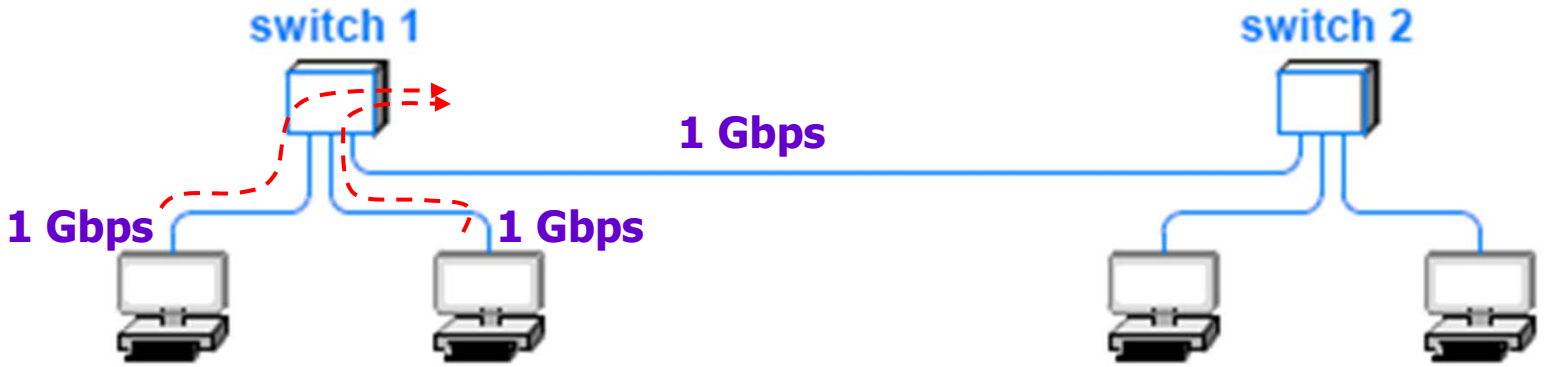


Figure 26.3 Comparison of transmission using (a) stop-and-go, and (b) sliding window.

Chapter 26 TCP: 26.6 Techniques To Avoid Congestion

- **Congestion เกิดเมื่อมีข้อมูลที่จะส่งมากเกินกว่าที่ Network จะรับได้**
  - อธิบายได้โดยใช้ Queuing Theory
  - Packet จะถูก Delay ไปมาก ในลักษณะ Exponential
  - ถ้า Queue ล้น Packet จะสูญหาย



Chapter 26 TCP: 26.6 Techniques To Avoid Congestion

- **Retransmission เนื่องจาก Queue ล้น จะทำให้เกิด Congestion Collapse**
  - Packet ที่ Retransmit จะถูกโยนทิ้งอีก เพราะ Queue ล้น
- **Congestion จะต้องตรวจจับได้อย่างทันท่วงที และทำการแก้ไข มิฉะนั้นจะเกิดผลเสียหายต่อทั้ง Network ซึ่งการตรวจจับของ Transport Protocol จะใช้พื้นฐานจากสองวิธี**
  - Intermediate System(Router) จะต้องแจ้งไปยังผู้ส่ง เมื่อมี Congestion
  - ผู้ส่งตรวจจับการเกิด Congestion เอง โดยดูจากค่า Delay หรือจำนวน Loss ของ Packet
    - ใน Internet จะใช้วิธีการนี้ เนื่องจาก Delay และ Loss ส่วนใหญ่ (ถ้าไม่ใช่เกิดจาก Hardware Failure) จะเป็นผลมาจาก Congestion
    - การแก้คือลดการส่งลง ทำได้โดยการควบคุมที่ Flow Control (ลดขนาดของ Window ลง)

Homework

- ยังไม่มีการบ้านในสัปดาห์นี้

CPE 426 Computer Networks

Chapter 8:
Text Chapter 26: TCP

TOPICS

- **Chapter 26: TCP** (Transmission Control Protocol)
  - **Transport Services**
  - **Transport Protocol Techniques**
  - **Congestion Control/Avoidance Techniques**
  - **Packet Loss and Adaptive Retransmission**
  - **Buffers, Flow Control and Windows**
  - **TCP Three-Way Handshake**
  - **TCP Congestion Control**
  - **TCP Segment Format**

Chapter 26

- TCP
- Transport Control Protocol
- Reliable Protocol for the Internet
- TCP/IP (TCP over IP)

Chapter 26 TCP: Reliable Transport Service

- **ในกรณีที่เราต้องการความมั่นใจในการส่งข้อมูลได้อย่างถูกต้อง ผ่าน IP ที่มีการทำงานแบบ Datagram เราต้องใส่ Mechanism เพื่อให้ความมั่นใจดังกล่าวลงใน Protocol ของ Layer 4 (Host-to-Host หรือ Transport)**
  - เราจะต้องเลือกใช้ TCP (Transmission Control Protocol)
  - เรียกรวมๆว่า TCP/IP
  - การใช้ TCP+IP ทำให้ผู้เขียน Application ไม่ต้องกังวลเรื่องของการส่งข้อมูลอีกต่อไป
    - เพียงแต่นำข้อมูลส่งให้ พร้อมกำหนด IP Address และ Port Number จากนั้น ระบบ TCP/IP จะจัดการที่เหลือให้
    - ผิดกับกรณีที่ใช้ UDP/IP ในบทที่แล้ว
    - กล่าวคือ TCP จะให้ Reliability ในการสื่อสาร

Chapter 26 TCP: 26.3 TCP Service

- **TCP ให้บริการของ 7 ส่วนใหญ่ๆ ดังนี้**
  - Connection Orientation
    - TCP ให้บริการแบบ Connection-Oriented ซึ่ง Application จะต้องร้องขอการเชื่อมต่อก่อนที่จะมีการส่งข้อมูล
  - Point-to-Point Communication
    - แต่ละ Connection ของ TCP จะมีสอง End Points เท่านั้น (กำหนด Port ต้นทาง-ปลายทาง และ IP ต้นทาง-ปลายทาง)
  - Complete Reliability
    - TCP จะ Guarantee ว่าข้อมูลที่ส่ง จะไปถึงที่หมายได้อย่างถูกต้อง สมบูรณ์ และเป็นลำดับ
  - Full Duplex Communication
    - แต่ละ TCP Connection จะยอมให้ข้อมูลส่งได้สองทิศทางตลอดเวลา

Chapter 26 TCP: 26.3 TCP Service(2)

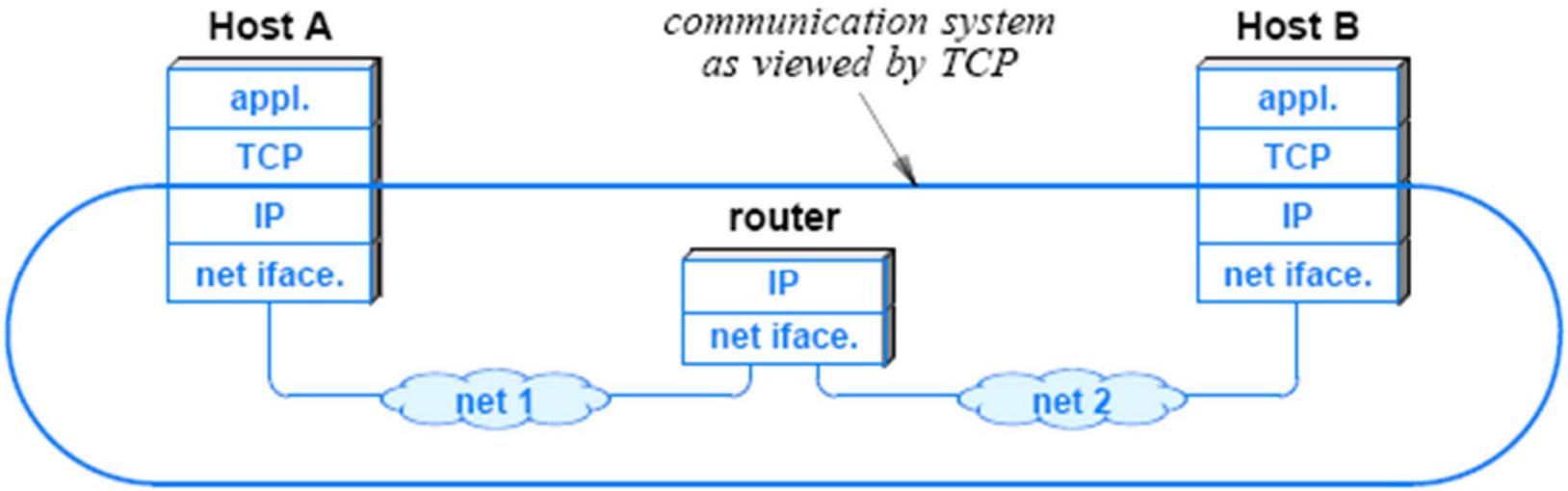
- **TCP ให้บริการของ 7 ส่วนใหญ่ๆ ดังนี้(ต่อ)**
  - Stream Interface
    - Application สามารถส่งข้อมูลได้อย่างต่อเนื่อง Octet ต่อ Octet ผ่าน TCP Connection โดย TCP จะไม่มีการรวมกลุ่มของ Data ให้เป็น Record หรือ Message และจะไม่ Guarantee ว่าแต่ละส่วนของข้อมูลที่ไปถึงมีขนาดเท่ากับที่ Application ส่งให้(มีการทำ Segmentation)
  - Reliable Connection Startup
    - TCP ให้สอง Application สามารถเริ่มต้นการสื่อสารได้อย่างมั่นใจ
  - Graceful Connection Shutdown
    - ก่อนที่จะจบ Connection นั้น TCP จะให้ความมั่นใจว่าข้อมูลได้ถูกส่งอย่างครบถ้วนทั้งสองฝั่ง และทั้งสองฝั่งยินยอมให้มีการจบการเชื่อมต่อ

Chapter 26 TCP: 26.4 End-to-End Service and Virtual Connection

- **TCP จัดว่าเป็น End-to-End Protocol เหมือน UDP**
  - เนื่องจากมันให้บริการการสื่อสารระหว่าง Application ของสอง Computer
- **แต่มันเป็น Connection-Oriented**
  - เนื่องจากต้องมีการทำ Connection ก่อนส่งข้อมูล
  - และต้องมีการทำ Disconnection
- **Connection ของ TCP จัดว่าเป็น Virtual Connection**
  - เนื่องจากกระทำผ่าน Software เพราะตัว Network (IP) นั้นเป็น Connectionless(Datagram)

Chapter 26 TCP: 26.4 End-to-End Service and Virtual Connection

- **แต่ละ TCP Message(Segment) จะถูกบรรจุใน IP Datagram และส่งผ่าน Internet เมื่อถึงปลายทาง IP จะปลอก Datagram Payload และส่งต่อไปยัง TCP**
  - IP จะมอง TCP เป็นแค่ Data ที่จะต้องส่ง
  - TCP จะมอง IP เป็นพาหนะสำหรับส่งข้อมูล ไปยัง TCP อีกฝั่งหนึ่ง



Chapter 26 TCP: 26.5 Transport Protocol Techniques

- **ปัญหาที่ End-to-End Transport Protocol จะต้องเจอ ในการได้มาซึ่ง Reliable Service**
  - Unreliable Communication
    - Message อาจจะสูญหาย ซ้ำซ้อน ผิดพลาด ถูกหน่วงเวลา หรือส่งมาไม่เป็นลำดับ (Lost, Duplicated, Corrupted, Delayed, Out of Order)
  - End System Reboot
    - ถ้า Host เกิด Crash หรือมีการ Reboot ข้อมูลที่ค้างอยู่ จะต้องไม่สับสนกับ Session ที่ถูกสร้างขึ้นมาใหม่
  - Heterogeneous End System
    - แต่ละ Host ที่เชื่อมต่อใน Internet มีความสามารถรับ-ส่งข้อมูลได้ไม่เท่ากัน
  - Congestion in the Internet
    - เมื่อมีการส่งข้อมูลมากเกินไป จะเกิดความคับคั่งภายใน Network ได้
- **ปกติแล้ว Transport Protocol จะใช้หลายๆวิธีร่วมกันในการจัดการกับปัญหาเหล่านี้ เราจะกล่าวพื้นฐานแต่ละอันต่อไป**

Chapter 26 TCP: 26.5.1 Sequencing to Handle Duplicates and Out-of-Order Delivery

- **Transport Protocol จะทำ Sequencing (กำหนดหมายเลขลำดับของข้อมูล) ในการจัดการกับปัญหาเรื่อง Duplicate Data และข้อมูลที่ไปถึงอย่างไม่เป็นลำดับ**
  - คล้ายกับวิธีที่กระทำใน Layer 2
  - หมายเลขข้อมูลทำให้เรารู้ว่าข้อมูลนั้นมาซ้ำกันหรือไม่
  - ทำให้รู้ว่าข้อมูลมาเป็นลำดับหรือไม่

Chapter 26 TCP: 26.5.2 Retransmission to Handle Lost Packet

- **ในการจัดการกับ Packet Lost นั้น Transport Protocol จะใช้ Positive Acknowledgement ร่วมกับการ Retransmission**
  - คล้ายกับ Mechanism ใน Layer 2 เช่นกัน
  - เมื่อ Frame มาถึงอย่างถูกต้อง จะมีการส่ง ACK Message กลับไป
  - เมื่อผู้ส่ง ส่งข้อมูลแต่ละ Packet จะมีการจับเวลาโดยใช้ Timer
    - ถ้า Timer Expire (คือไม่ได้รับ ACK ในเวลาที่กำหนด) ผู้ส่งจะทำการ Retransmission ข้อมูลนั้นไปใหม่
  - ถ้า Packet มีการ Delay มาก อาจจะทำให้เกิด Retransmission ยังผลให้เกิด Duplicate Packet

Chapter 26 TCP: 26.5.3 Techniques to Avoid Replay

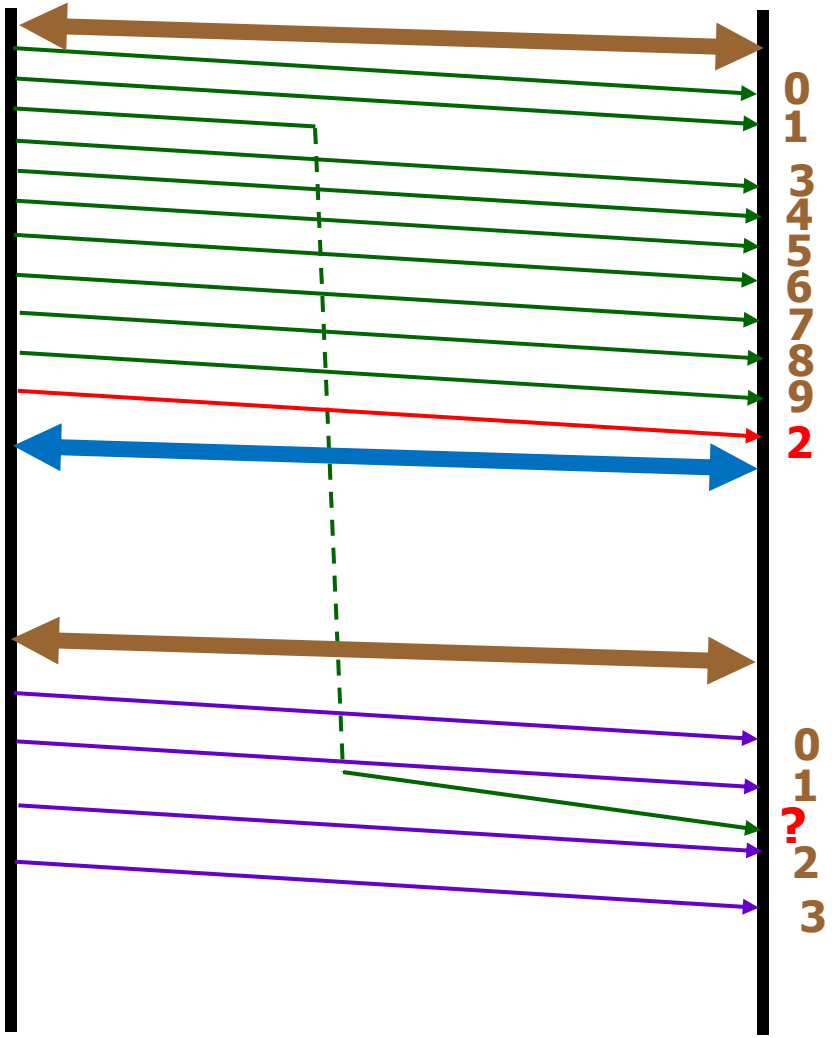
- **ในกรณีที่ Network มี Delay สูง อาจจะทำให้เกิด 'Replay Error' โดยที่ Packet ที่ Delay นั้นจะส่งผลต่อการสื่อสารในตอนหลัง**
  ยกตัวอย่าง
  - สองคอมพิวเตอร์ตกลงจะสื่อสารกันเมื่อเวลา 13.00 น.
  - คอมพิวเตอร์เครื่องหนึ่ง ส่ง 10 Packet ติดต่อกันไปยังฝั่งตรงข้าม
  - ปัญหาด้าน Hardware ทำให้ Packet 3 ถูก Delay ไป
  - เส้นทางการส่งข้อมูลถูกเปลี่ยน เพื่อใช้เส้นทางที่ไม่ผ่านอุปกรณ์ที่มีปัญหา
  - Protocol ที่คอมพิวเตอร์ต้นทางส่ง Packet 3 ไปใหม่ จากนั้นส่ง Packet อื่นๆตาม จากนั้นจบการสื่อสาร

Chapter 26 TCP: 26.5.3 Techniques to Avoid Replay

- สองคอมพิวเตอร์ตกลงจะสื่อสารกันเมื่อเวลา 13.00 น.
- คอมพิวเตอร์เครื่องหนึ่ง ส่ง 10 Packet ติดต่อกันไปยังฝั่งตรงข้าม
- ปัญหาด้าน Hardware ทำให้ Packet 3 ถูก Delay ไป
- เส้นทางการส่งข้อมูลถูกเปลี่ยน เพื่อใช้เส้นทางที่ไม่ผ่านอุปกรณ์ที่มีปัญหา
- Protocol ที่คอมพิวเตอร์ต้นทางส่ง Packet 3 ไปใหม่ จากนั้นส่ง Packet อื่นๆตาม จากนั้นจบการสื่อสาร
- เวลา 13.05 น. คอมพิวเตอร์ทั้งสองตกลงที่จะสื่อสารกันใหม่
- หลังจากผู้ส่งคนเดิม ส่งไปได้สอง Packet แล้ว ตัว Packet 3 จากการสื่อสารครั้งแรกที่ถูก Delay มาถึงยังผู้รับ
- ต่อจากนั้น Packet ที่ 3 จากการสื่อสารครั้งที่สองมาถึงยังผู้รับ

- **Transport Protocol ต้องออกแบบเพื่อจัดการกับเรื่องนี้ มิฉะนั้นผู้รับจะได้ Packet 3 ที่ผิดพลาด ในขณะที่โยนทิ้ง Packet ที่ถูกต้อง**
  - Protocol จะต้อง Mark แต่ละ Session โดยใช้ ID ที่เฉพาะ ไม่ซ้ำกัน โดยการนำ ID กลับมาใช้ใหม่ต้องให้แน่ใจว่าจะไม่เกิด Replay (เวลาต้องห่างกัน)

Replay (Selective Reject)



Chapter 26 TCP: 26.5.3 Techniques to Avoid Replay

- **Replay ยังสามารถเกิดกับ Control Packet ได้เช่นกัน**
  - เช่น Connection Closing Packet ถูก Delay และไปถึงหลังจากมีการทำ Connection ใหม่ครั้งที่สอง

Chapter 26 TCP: 26.5.4 Flow Control to Prevent Data Overrun

- **ในการจัดการเรื่อง Heterogeneous End System ที่มีความสามารถไม่เท่ากัน ตัว Transport Protocol จะใช้วิธีการของ Flow Control เพื่อควบคุม**
  - Stop-and-go Protocol มักจะไม่ใช้ เนื่องจากมีประสิทธิภาพต่ำ
  - ปกติจะใช้ Sliding Window Flowcontrol
  - เช่นเดียวกัน จะคล้ายกับ Flow Control ใน Layer 2
    - การคำนวณ Efficiency จะคล้ายกับที่กล่าวมาแล้วใน Layer 2
    - ให้ดูใน Slide ก่อนหน้า หรือดูจาก CPE 326 Course Notes

Chapter 26 TCP: 26.5.4 Flow Control to Prevent Data Overrun

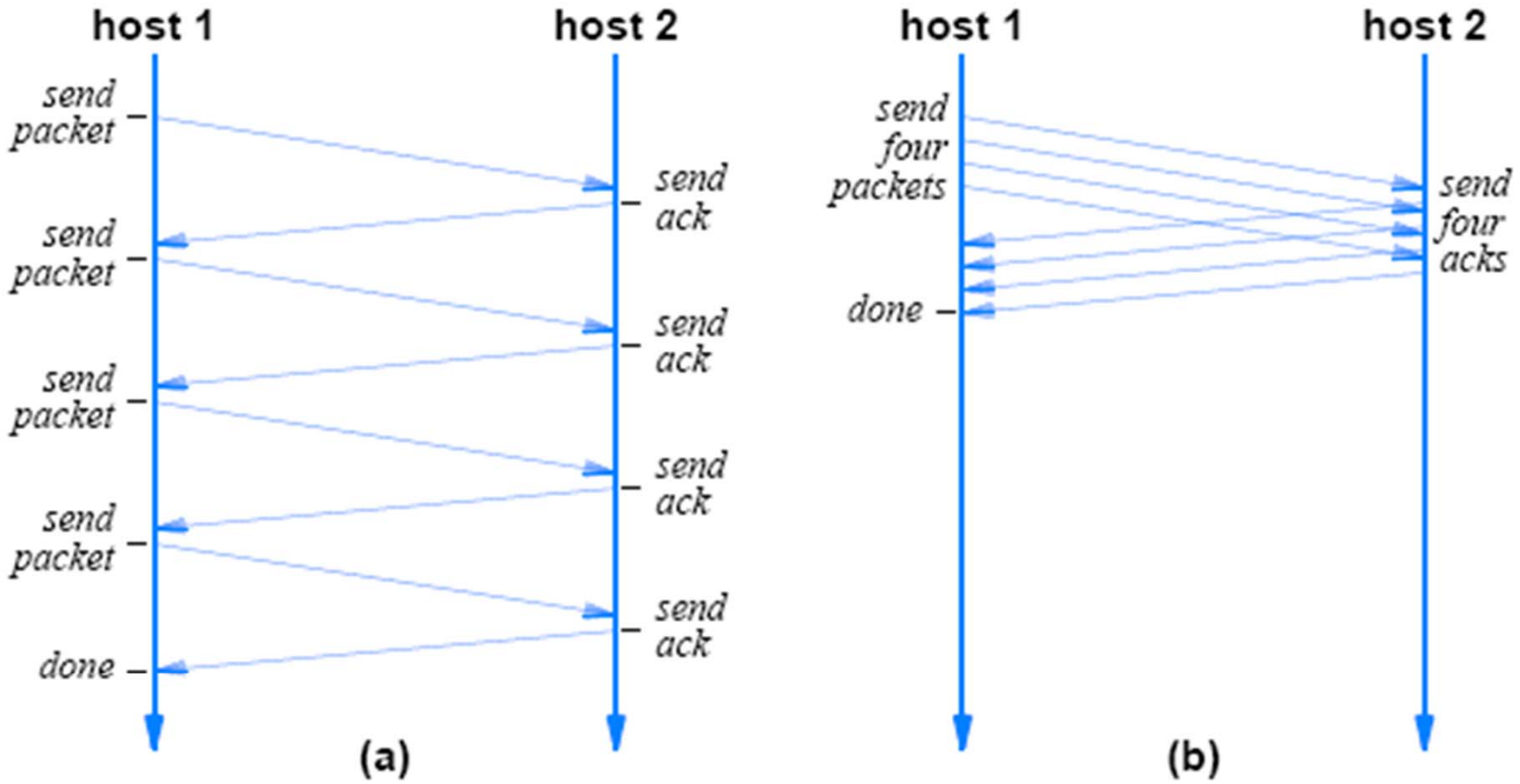
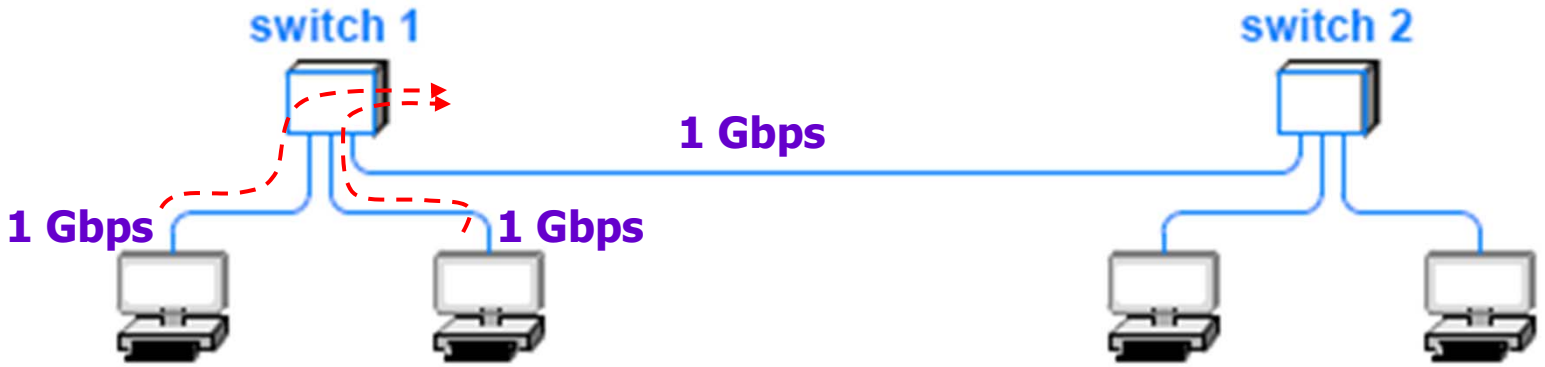


Figure 26.3 Comparison of transmission using (a) stop-and-go, and (b) sliding window.

Chapter 26 TCP: 26.6 Techniques To Avoid Congestion

- **Congestion เกิดเมื่อมีข้อมูลที่จะส่งมากเกินกว่าที่ Network จะรับได้**
  - อธิบายได้โดยใช้ Queuing Theory
  - Packet จะถูก Delay ไปมาก ในลักษณะ Exponential
  - ถ้า Queue ล้น Packet จะสูญหาย



Chapter 26 TCP: 26.6 Techniques To Avoid Congestion

- **Retransmission เนื่องจาก Queue ล้น จะทำให้เกิด Congestion Collapse**
  - Packet ที่ Retransmit จะถูกโยนทิ้งอีก เพราะ Queue ล้น
- **Congestion จะต้องตรวจจับได้อย่างทันท่วงที และทำการแก้ไข มิฉะนั้นจะเกิดผลเสียหายต่อทั้ง Network ซึ่งการตรวจจับของ Transport Protocol จะใช้พื้นฐานจากสองวิธี**
  - Intermediate System(Router) จะต้องแจ้งไปยังผู้ส่ง เมื่อมี Congestion
  - ผู้ส่งตรวจจับการเกิด Congestion เอง โดยดูจากค่า Delay หรือจำนวน Loss ของ Packet
    - ใน Internet จะใช้วิธีการนี้ เนื่องจาก Delay และ Loss ส่วนใหญ่ (ถ้าไม่ใช่เกิดจาก Hardware Failure) จะเป็นผลมาจาก Congestion
    - การแก้คือลดการส่งลง ทำได้โดยการควบคุมที่ Flow Control (ลดขนาดของ Window ลง)

Chapter 26 TCP: 26.7 The Art of Protocol Design

- แต่ละ **Technique** ที่กล่าวมาจะต้องถูกเลือกใช้อย่าระวัง และส่งผลต่อการออกแบบ **Protocol**
- **Protocol Header** จะต้องมี **Field** ที่จะรองรับข้อมูลต่างๆเพื่อควบคุมการทำงาน
- จะต้องแก้ปัญหาเรื่อง **Computer Reboot** ด้วย เพื่อป้องกัน **Duplicate Connection** หรือการทำ **Connection** ที่ไม่ถูกต้อง
- ในหัวข้อที่จะกล่าวต่อไป จะเป็นรายละเอียดเฉพาะที่ **TCP** เลือกใช้

Chapter 26 TCP: 26.8 Techniques Used In TCP To Handle Packet Loss

- **TCP ใช้การ Retransmission ในกรณีที่ Packet Loss**
  - ผ่านการทำ Positive Acknowledge ในการสื่อสารทั้งสองทิศทาง
  - มีการใช้ Transmission Timer
    - Transmission Timer ควรตั้งไว้เท่าไร
    - ต่ำเกินไป จะมีผลให้มี Duplicate Data ใน Network ที่ช้าเช่น Satellite Network
    - สูงเกินไป จะทำให้ประสิทธิภาพลดลง เช่นใน LAN
    - Congestion ที่เกิดขึ้นจะทำให้เกิด Delay และมีผลต่อการตั้ง Retransmission Timer
      - Ack อาจจะมี Delay เพิ่มในระดับ Magnitude ถ้ามี Congestion
  - Retransmission Timer ใน TCP จะต้องมีความสามารถปรับให้เหมาะกับ Delay ใน Network
    - ไม่เหมือนกับ Layer 2 Retransmission Timer ซึ่งเป็นระดับ Point-to-Point (Link) ที่สามารถคำนวณและกำหนดล่วงหน้าได้

Chapter 26 TCP: 26.8 Techniques Used In TCP To Handle Packet Loss

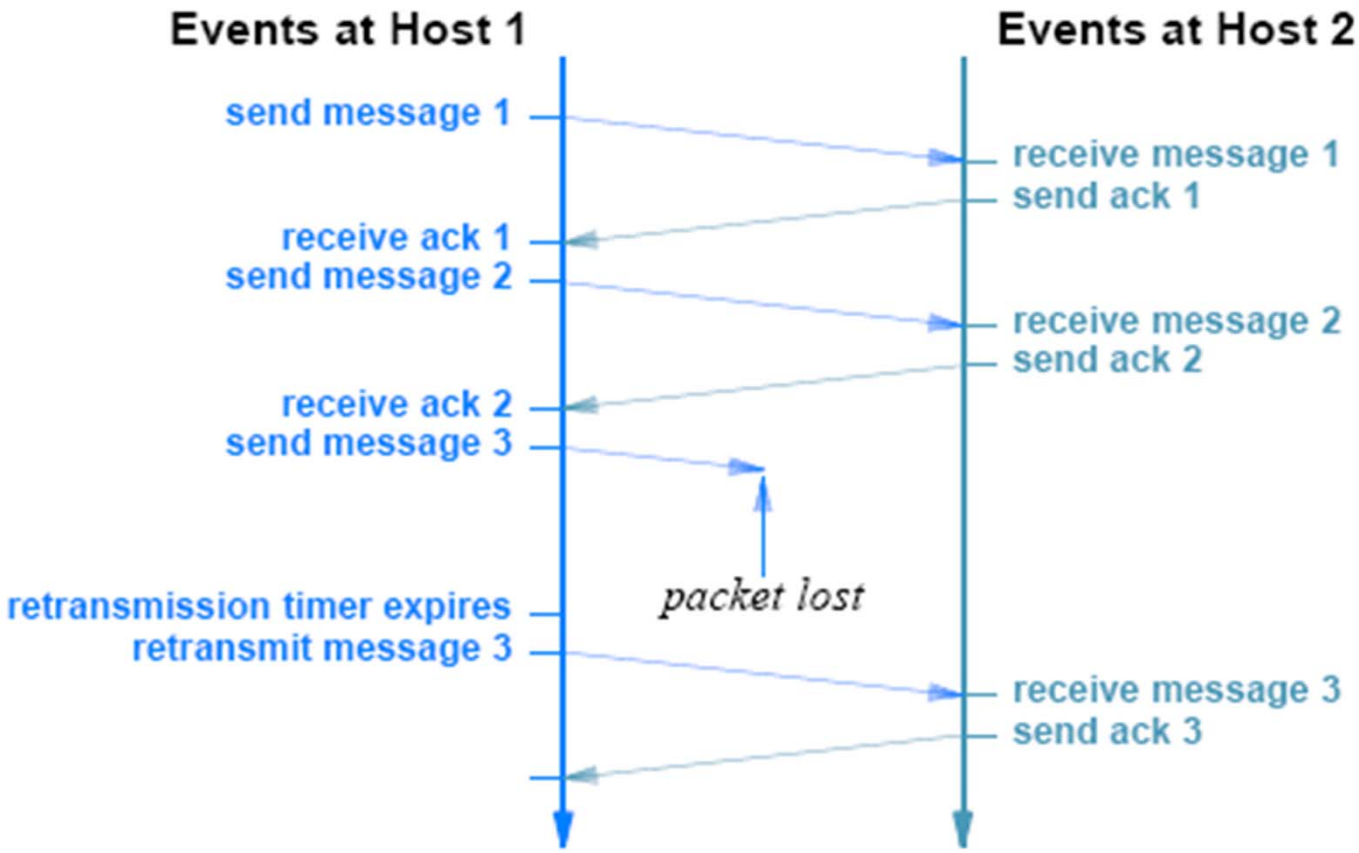


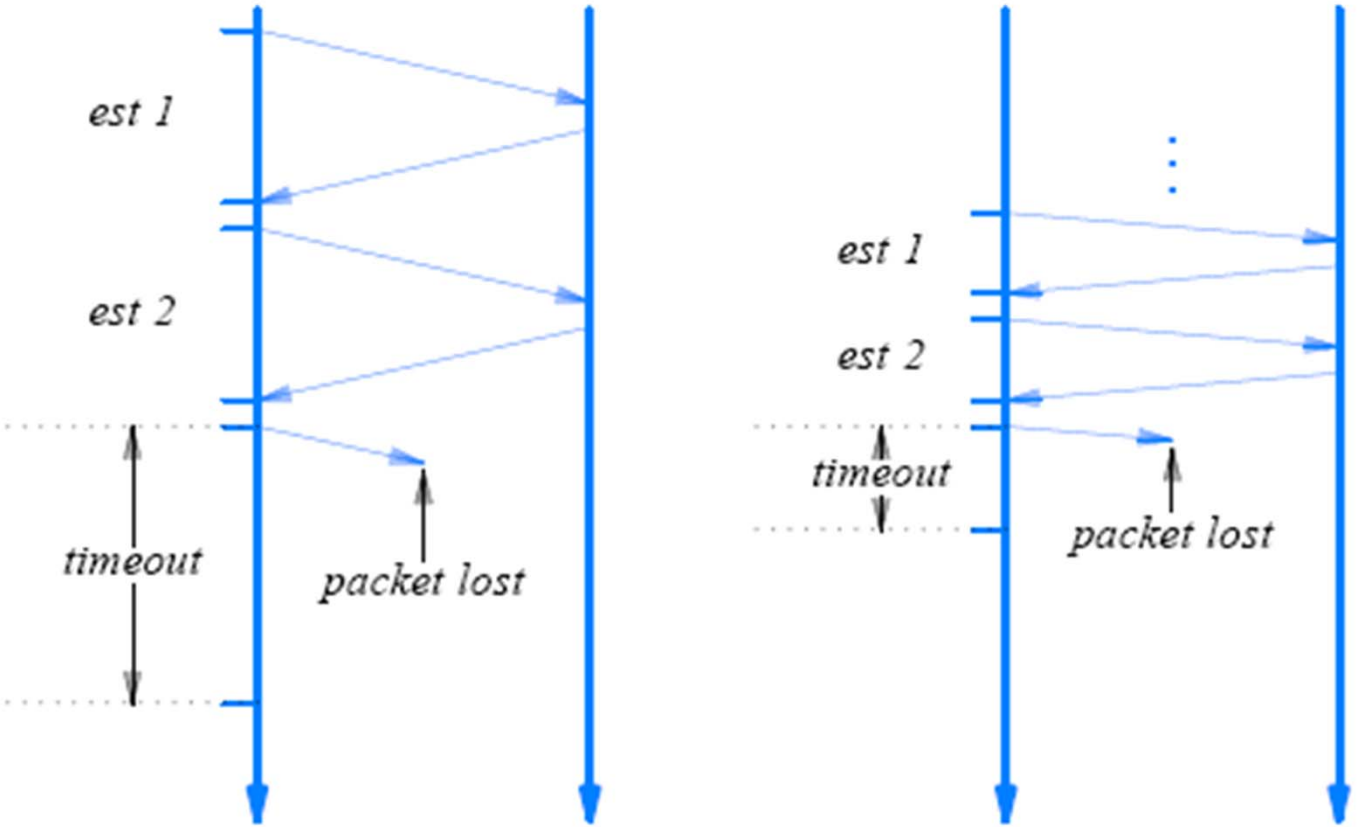
Figure 26.5 Illustration of TCP retransmission after a packet loss.

Chapter 26 TCP: 26.9 Adaptive Retransmission

- **ผู้ออกแบบ TCP เล็งเห็นว่า การตั้ง Retransmission Timer ที่คงที่จะมีผลต่อประสิทธิภาพที่ลดลงของ Internet**
  - เนื่องจาก Internet ประกอบด้วย Network หลายๆ Technology ที่มีค่า Delay ต่างกัน
- **TCP จะใช้ Adaptive Retransmission Timer ที่สามารถปรับค่าตามค่า Delay ของ Network**
  - TCP จะประมาณค่า Round-trip Delay จากเวลาที่ส่ง Packet ออกไปและได้รับคำตอบกลับมา เช่นตอนทำ Connection หรือช่วงการส่งข้อมูลและได้รับ ACK
    - มันจะคำนวณค่า Round-Trip Delay โดยใช้ Weighted Average ร่วมกับค่า Variance
    - ปกติ มันจะตั้งค่า Timer สูงกว่า Round Trip เล็กน้อย
    - เมื่อ Delay เริ่มมีการเปลี่ยนแปลง มันจะตั้งค่า Timer เพิ่มขึ้นที่สูงกว่าค่า Mean ที่คำนวณได้ ร่วมกับค่า Variance
    - การใช้ Weight Average จะทำให้สามารถ Reset Timer เมื่อ Delay กลับเข้าสู่ค่าปกติ

Chapter 26 TCP: 26.10 Comparison of Retransmission Timer

- พิจารณาจากการเกิด Packet Loss ของสอง Network ที่มี Delay ต่างกัน
  - ถ้า Delay มีค่าสูง TCP จะตั้งค่า Timer ที่สูงตาม



Chapter 26 TCP: 26.11 Buffers, Flow Control and Windows

- **TCP ใช้ Window Mechanism ในการควบคุมการไหลข้อมูล**
  - ต่างจาก Window Flow Control ใน Layer 2 ที่นับเป็น Frame แต่ TCP จะกำหนด Window เป็น Byte ของ Data ที่ส่งได้
  - หลังจากทำ Connection คอมพิวเตอร์ทั้งสองฝั่งจะกำหนดค่า Buffer (Memory) เพื่อใช้ในการรับ-ส่งข้อมูล
    - จากนั้นมันจะส่งขนาดของ Buffer ที่เหลือไปยังฝั่งตรงข้าม
    - เมื่อได้รับข้อมูล มันจะ Acknowledge ด้วยค่าของ Buffer ที่ยังสามารถรับได้ (Buffer ที่เหลือ)
    - ดังนั้นคำว่า Window ในความหมายของ TCP คือ จำนวน Buffer เป็น Byte ที่ยังเหลืออยู่ที่จะรับข้อมูลได้
      - เรียก Window Advertisement
    - ถ้าผู้ส่ง ส่งข้อมูลเร็วเกินไป ผู้รับไม่สามารถ Process ทัน มันจะ Acknowledge กลับมาด้วย Zero Window
      - ผู้ส่งต้องหยุดส่ง

Chapter 26 TCP: 26.11 Buffers, Flow Control and Windows(2)

- **ในตัวอย่าง ผู้ส่งใช้ขนาดของ Segment สูงสุด 1000 Bytes โดยกำหนด Initial Window Size 2500 Bytes**
  - ผู้ส่งส่งสาม Segment ขนาด 1000, 1000 และ 500 Bytes
  - ผู้รับ Process ไม่ทัน ส่ง Zero Window Advertisement
  - ผู้ส่งหยุดส่ง
  - ต่อมาผู้รัยทำการ Process 2000 Byte ของ Data จากนั้นทำการ Advertise ว่ารับได้อีก 2000 Byte เหนือจากที่รับแล้ว 2500 Byte
    - Window Size จะวัดต่อจาก Data ที่ทำการ Acknowledge เสมอ
  - ผู้ส่งส่งอีกสอง Segment แต่ละอันมีขนาด 1000 Byte
  - Window Size ลดลงเหลือศูนย์อีก ทำให้ผู้ส่งหยุดส่ง

Chapter 26 TCP: 26.11 Buffers, Flow Control and Windows(3)

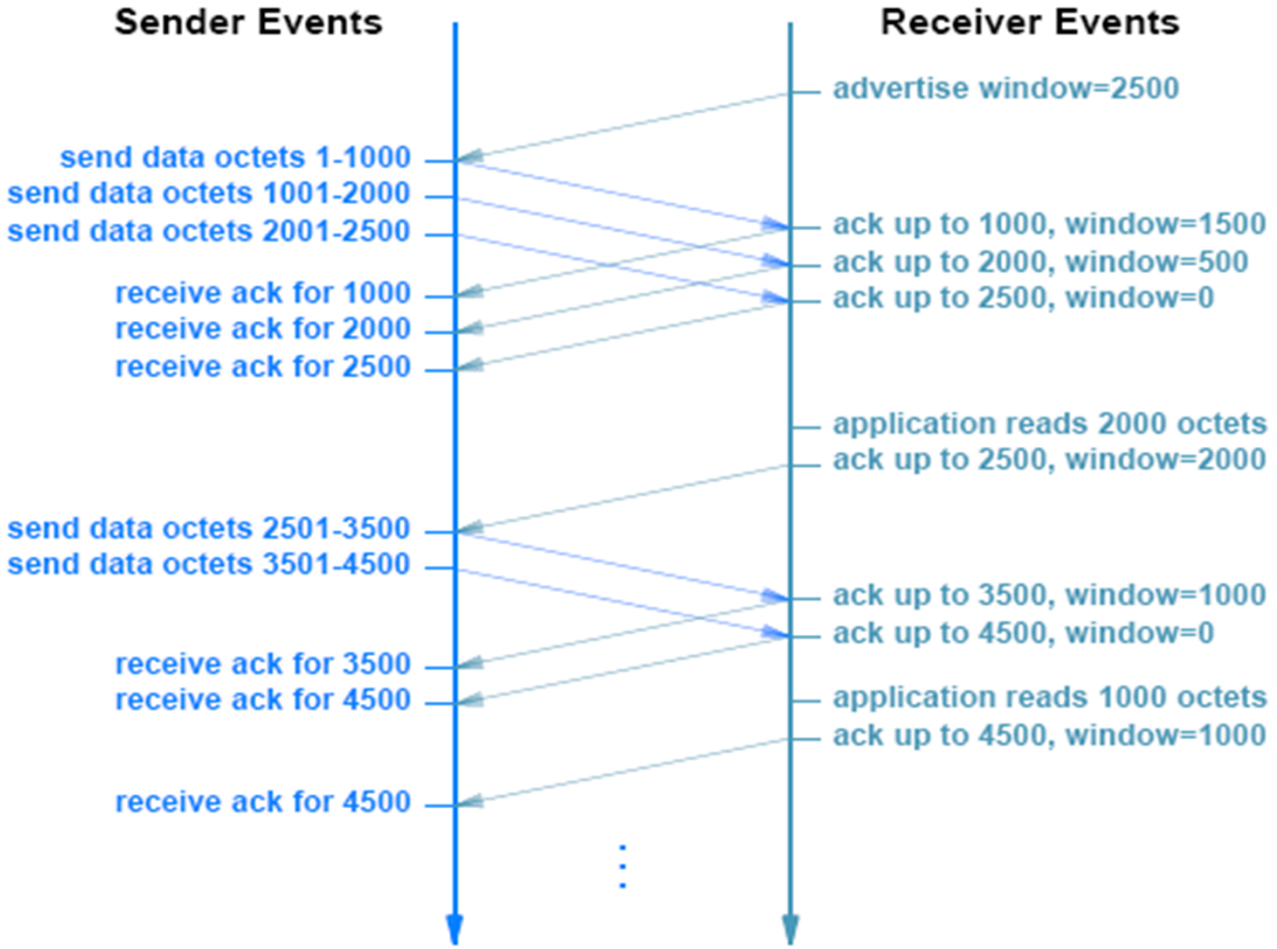


Figure 26.7 A sequence of messages that illustrates TCP flow for a maximum segment size of *1000* bytes.

Chapter 26 TCP: 26.12 TCP Three-Way Handshake

- **ในการที่จะให้แน่ใจว่าการทำ Connection และการ Terminate Connection ไม่ผิดพลาด TCP จะใช้วิธีการของ Three-Way Handshake**
  - ระหว่างการทำ Three-way Handshake จะมีการแลกเปลี่ยน Control Message และกำหนดขนาดของ Buffer
  - Three-Way Handshake จะสามารถจัดการกับกรณีที่เกิด Packet Loss, Duplicate, Delay และ Replay ได้ และจะ Guarantee ว่า Connection จะสร้างและจบลง เมื่อทั้งสองฝ่ายตกลงกัน
- **Control Message ที่ใช้ระหว่างทำ Three-Way Handshake Connection เรียก Synchronization Segment (SYN Segment) และเรียก Finish Segment(FIN Segment) สำหรับ Control Message ตอนทำ Termination**
- **ตอนทำ Three-Way Handshake จะมีการเลือก Sequence Number**
  - แต่ละด้านจะเลือกเลข 32 Bit แบบ Random เป็น Sequence เริ่มต้น ของ Data
  - ถ้าเกิดการ Reboot และ Application พยายามจะทำ Connection ใหม่ โอกาสที่จะได้ Sequence เดิมจะน้อยมาก ดังนั้นปัญหา Replay จะเลี่ยงได้
- **ในกรณีที่ทำ Close Connection จะมีการส่ง FIN ร่วมกับ Acknowledge เพื่อให้แน่ใจว่า Data ได้รับครบถ้วนก่อนจะปิด Connection**

Chapter 26 TCP: 26.12 TCP Three-Way Handshake(Open Connection)

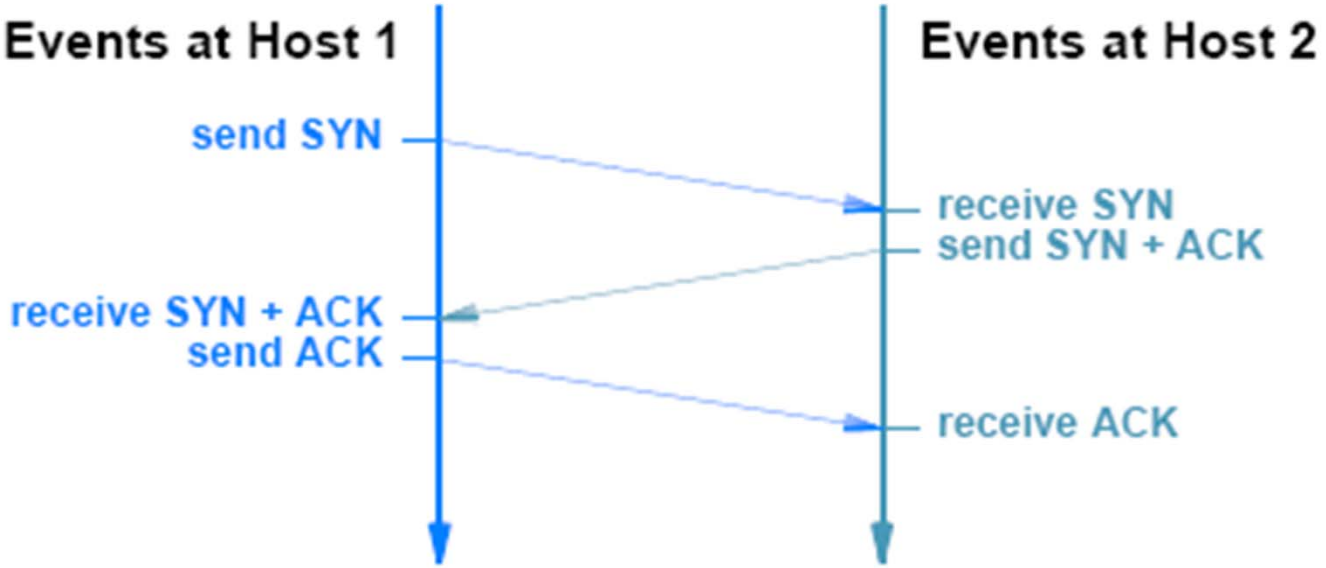


Figure 26.8 The 3-way handshake used to create a TCP connection.

Chapter 26 TCP: 26.12 TCP Three-Way Handshake(Close Connection)

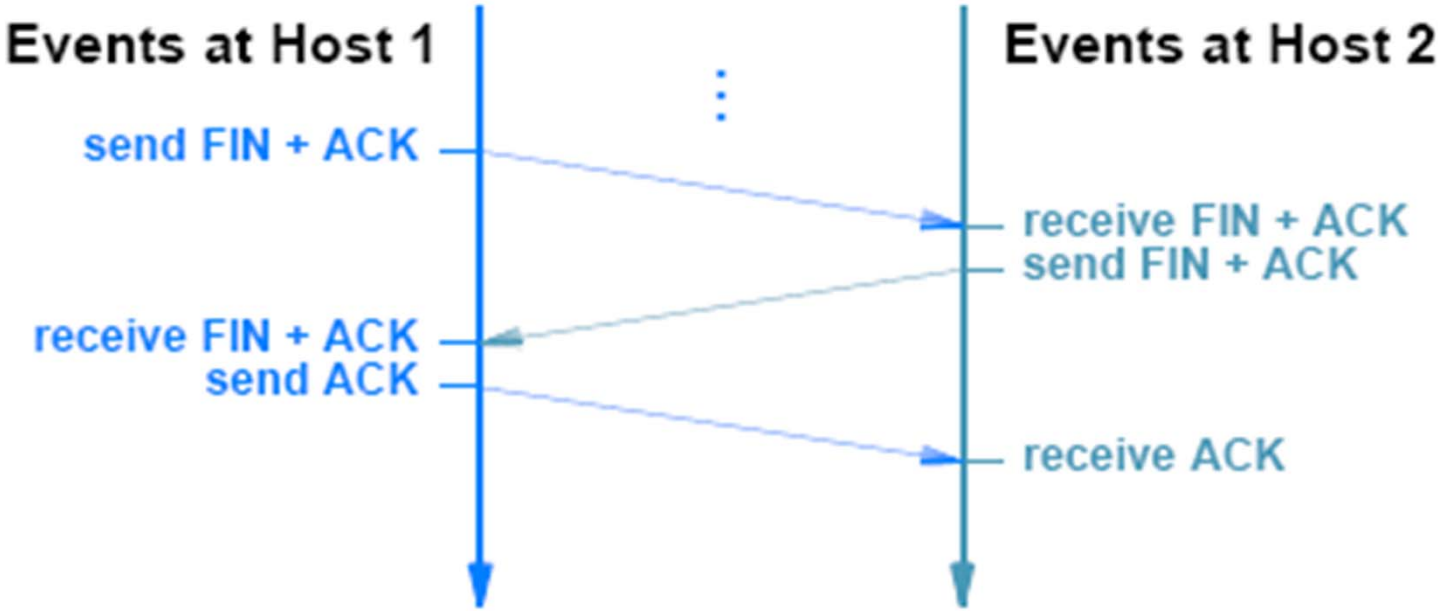


Figure 26.9 The 3-way handshake used to close a connection.

Chapter 26 TCP: 26.13 TCP Congestion Control

- **TCP มี Congestion Control ที่ดูเหมือนประหลาดกว่าทั่วไป**
  - เนื่องจาก Congestion ทำให้เกิด Delay ส่งผลให้เกิด Retransmission ส่งผลกลับมาให้ Congestion หนักขึ้น คือเกิด Congestion Collapse
  - แม้ว่าการจัดการกับ Congestion ทั่วไปจะใช้วิธีลดอัตราการส่ง แต่ TCP ไม่สามารถคำนวณอัตราการส่งได้ เพราะวัดการส่งจาก Buffer ที่ได้รับจากอีกฝั่งหนึ่ง
  - ดังนั้นการควบคุมอัตราการส่งใน TCP จะควบคุมจากการเปลี่ยนแปลงขนาดของ Window แทน
    - การลดขนาด Window ลงชั่วคราว จะเป็นการลดอัตราการส่งข้อมูลไปในตัว
    - เมื่อมี Data Loss (คือ Congestion ในมุมมอง TCP) ขนาด Window จะถูกลดลง

Chapter 26 TCP: 26.13 TCP Congestion Control(2)

- **TCP มี Congestion Control หลาย Algorithm ที่นิยมคือวิธีของ 'Slow Start'**
  - เมื่อเริ่ม Connection ใหม่ หรือกรณีที่มี Message Loss มันจะเริ่มจากการส่งหนึ่ง Message
    - ถ้ามันได้รับ ACK มันจะส่งสอง Message
    - ถ้าได้รับ ACK ทั้งสอง มันจะส่ง 4 Messgae
    - และจะส่งเป็นสองเท่าตราบใดที่ได้รับ ACK ครบ เรื่อยไปจนถึงส่งเท่ากับครึ่งหนึ่งของ Window Size ที่ Advertise จากผู้รับ
      - จากนั้นมันจะเพิ่มการส่งแบบ Linear ตราบเท่าที่ไม่มีการ Congestion
    - ถ้ามี Congestion จะเริ่มส่งทีละ Message ใหม่
      - ดังนั้นทุกคนจะหยุดส่งถ้าเกิด Congestion และป้องกันการเกิด Congestion Collapse
  - Congestion Control จะใช้ร่วมกับ Congestion Avoidance ซึ่งมี Algorithm มากมาย

Chapter 26 TCP: 26.14 TCP Segment Format

- มี **Format** เดียว ใช้กับ **Message, Acknowledge, SYN, FIN**
  - แต่ละ Message ของ TCP เรียก Segment
  - แต่ละ Segment สามารถส่ง Data, Acknowledge ของ Data ที่ได้รับ, Window Advertisement
    - Acknowledgement Number และ Window Field หมายถึง Data ที่ได้รับ คือ Sequence Number เริ่มต้นของ Data ถัดไปที่ต้องการ และ Buffer ที่เหลือ ถ้า Data ที่ได้ไม่เป็น Order มันจะส่ง Ack Number ของ Data ที่คาดหวังซ้ำๆจนกว่าจะได้รับ
    - Sequence Number หมายถึง Sequence Number ของ Byte แรกของ Data ที่ส่ง
    - Checksum จะเป็น Checksum ของ Header และ Data
    - Code Bit บ่งบอกว่าเป็น Segment ชนิดใหน (Data, SYN, FIN)

Chapter 26 TCP: 26.14 TCP Segment Format

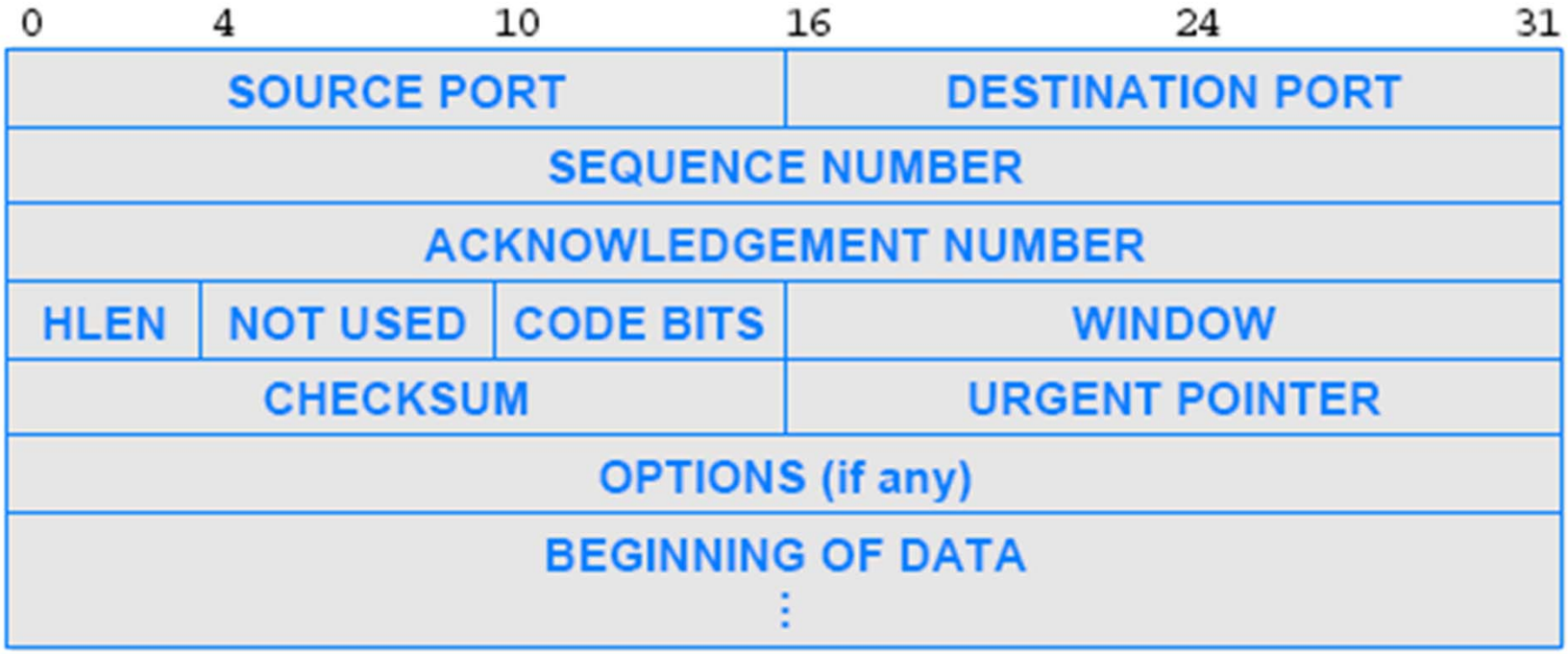


Figure 26.10 The TCP segment format used for both data and control messages.

TCP Port

- **เช่นเดียวกันกับ UDP Port**
  - Well-Known Port: 0-1023
    - FTP: 20/21 (Data/Control)
    - Telnet: 23
    - SMTP: 25
    - DNS: 53 (UDP/TCP)
    - BOOTP: 67/68 (Server/Client) (Mostly UDP)
    - HTTP: 80
  - Registered Port: 1024-49151
  - Dynamic Port: 49152-65535

End of Week 11

- **HW 6: UDP และ TCP**
  - ให้ Download Question จาก Web
  - ส่ง สัปดาห์หน้า
- **Next Week, Week 12**
  - Routing Part I: Routing Mechanism

CPE 426 Computer Networks

Chapter 9:

Text Chapter 18&27: Internet Routing Part I:

TOPICS

- Chapter 27: Internet Routing and Routing Protocols
  - 27.1 Introduction
  - 27.2 Static vs Dynamic Routing
  - Extra: Router Configuration in Network
  - 27.3 Static Routing and Default Route
  - Extra: Examples of Static Routing
- BREAK
  - 27.4 Dynamic Routing and Router
  - 27.5 Routing in Global Internet
  - 27.6 Autonomous System Concept
  - 27.7 Two Types of Routing Protocol
  - 27.8 Routes and Data Traffic
  - Extra: Bellman-Ford Algorithm Review
  - Extra: Dijkstra Algorithm Review

Chapter 27:Internet Routing and Routing Protocol

- ลักษณะการส่ง Packet ใน IP Network จะส่งทีละ Hop จาก Network หนึ่ง ไปยังอีก Network หนึ่ง
- Router จะทำหน้าที่ดังกล่าว เนื่องจาก Router เป็นตัวเชื่อมระหว่าง Network
- การส่งจะดูที่ส่วน Prefix ของ IP Address ดังนั้น Router จะต้อง Run IP Protocol คือ ทำงานในระดับ Layer 3
- ที่ Router จะมีตารางชื่อ Routing Table ที่กำหนด IP Address ของ Next Hop

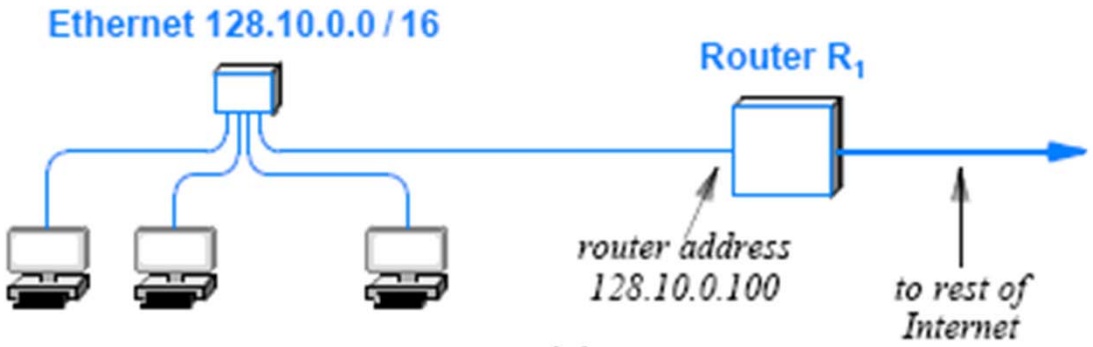
Chapter 27: 27.2 Static vs Dynamic Routing

- **การทำ Routing ทำได้สองแบบ**
  - Static Routing: หมายถึงตาราง Routing Table ของ Router แต่ละตัวจะไม่เปลี่ยน ปกติตารางนี้จะถูกกำหนดจาก Network Administration
    - คือตารางนี้จะได้จากการทำ Configuration ของ Router
  - Dynamic Routing: หมายถึงตาราง Routing Table สามารถเปลี่ยนได้ ตามสภาวะความคับคั่งของ Network ขณะนั้น โดยมันจะมีการ Update ตลอดเวลา
    - ตารางนี้ได้จากการกำหนดให้ Router ทำการ Run Routing Protocol
      - ในแต่ละ Routing Protocol ตัว Router จะทำการแลกเปลี่ยนข้อมูลกันเอง โดยผู้ดูแล Network ไม่ต้องมาเกี่ยวข้อง จากนั้น Router แต่ละตัวจะสร้างตาราง Routing Table จากข้อมูลที่มันรวบรวมได้ เมื่อข้อมูลที่ได้รับเปลี่ยน เนื่องจาก Network เปลี่ยน มันจะคำนวณตารางใหม่ และปรับให้เข้ากับสภาพของ Network ที่เปลี่ยนไป

Chapter 27: 27.3 Static Routing in Host and Default Route

- ข้อดีของ Static Routing คือง่าย และไม่ต้องใช้ Routing Software(Router จะทำงานน้อยลง) นอกจากนี้จะไม่ใช้ Resource ของ Network เพราะ Router ไม่ต้องแลกเปลี่ยนข้อมูลกัน
- แต่ข้อเสียคือ ตารางเปลี่ยนไม่ได้ ถ้ามี Link Down หรือ Router Down เส้นทางนั้นจะใช้ไม่ได้ ทำให้การส่งข้อมูลที่กำหนดเส้นทางนั้นหยุดชะงัก
  - นอกจากนี้แล้ว Static Routing จะจำกัดที่ Network ขนาดเล็ก เช่น ระหว่าง LAN ขององค์กร การเชื่อมต่อกับ Internet จะไม่ใช้ Static Routing
  - หรือใช้กำหนดสำหรับ Host ที่เชื่อมต่อกับ Network ที่มีทางออกผ่าน Router ตัวเดียว(คือค่า Gateway)
- การใช้ Static Route ควรมีการกำหนด Default Route เสมอ
- การทำ Static Route ต้องตรวจสอบให้ดีว่าจะไม่มี Route Loop

Chapter 27: 27.3 Static Routing in Host and Default Route



(a)

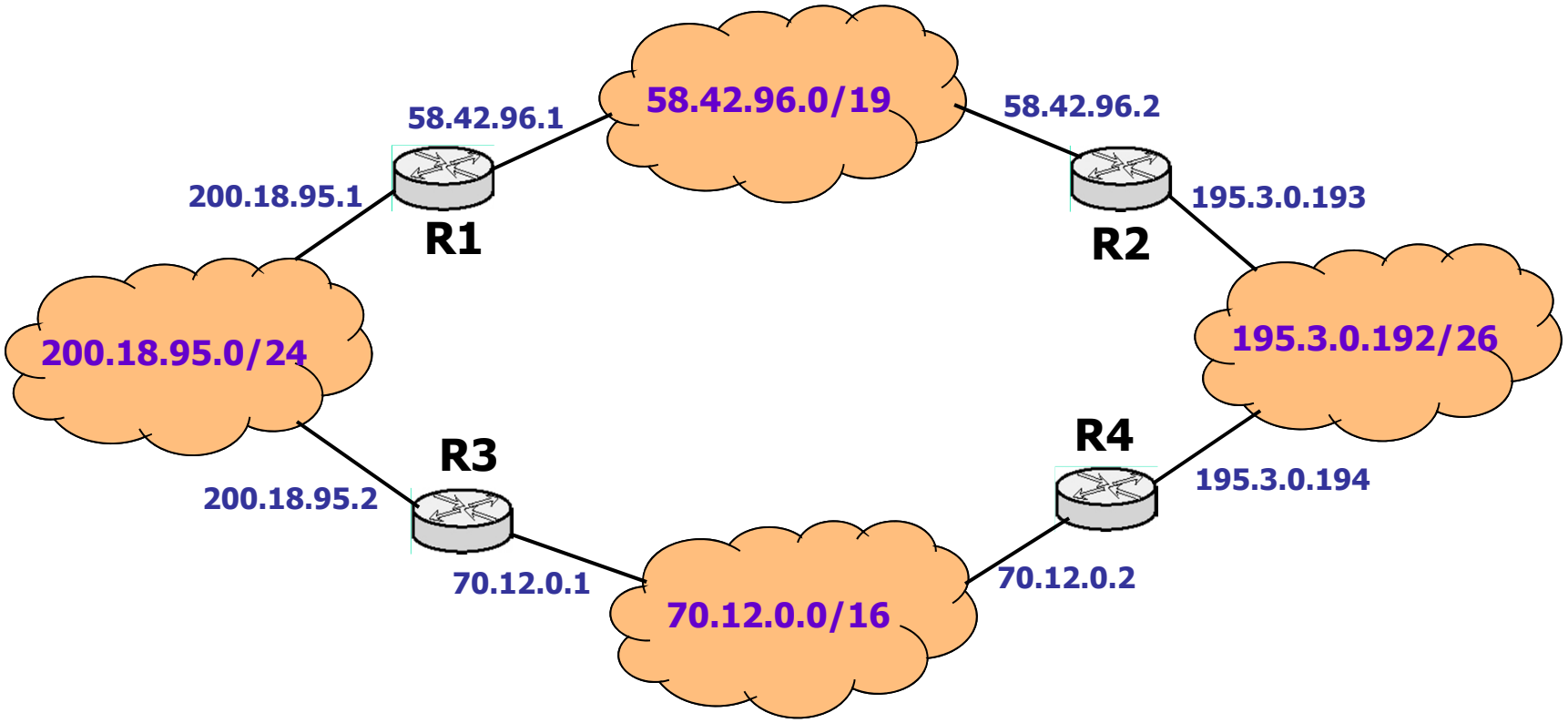
Net	Mask	Next hop
128.10.0.0	255.255.0.0	direct
default	0.0.0.0	128.10.0.100

(b)

Figure 27.1 (a) A typical connection to the Internet, and (b) the static forwarding table used in each host.

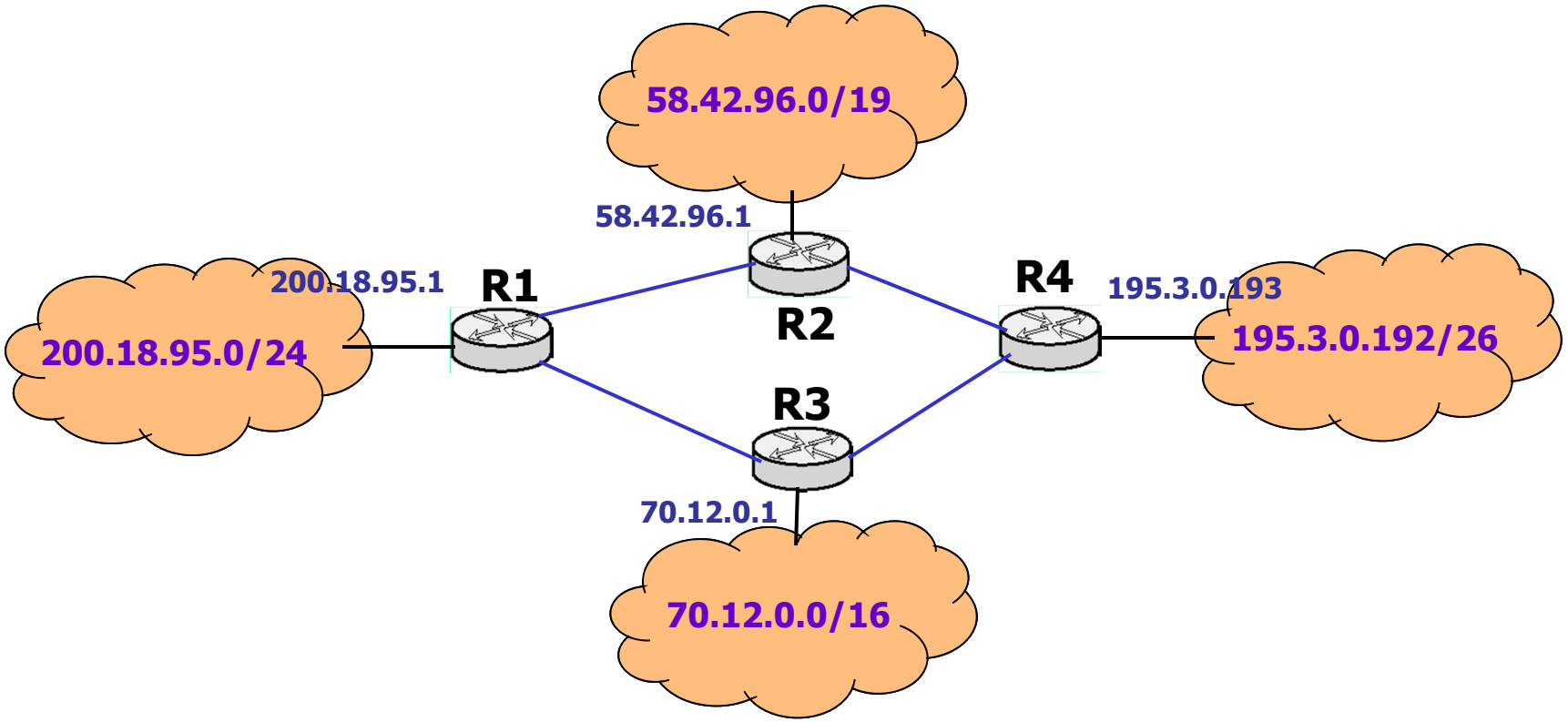
การกำหนด Interface สำหรับ Router

- แต่ละ Interface ของ Router จะต้องมี IP Address ที่ตรงกับแต่ละ Network ที่ Interface เชื่อมต่อ (Prefix เหมือนกัน แต่ Suffix ต่างกัน)



การกำหนด Interface สำหรับ Router

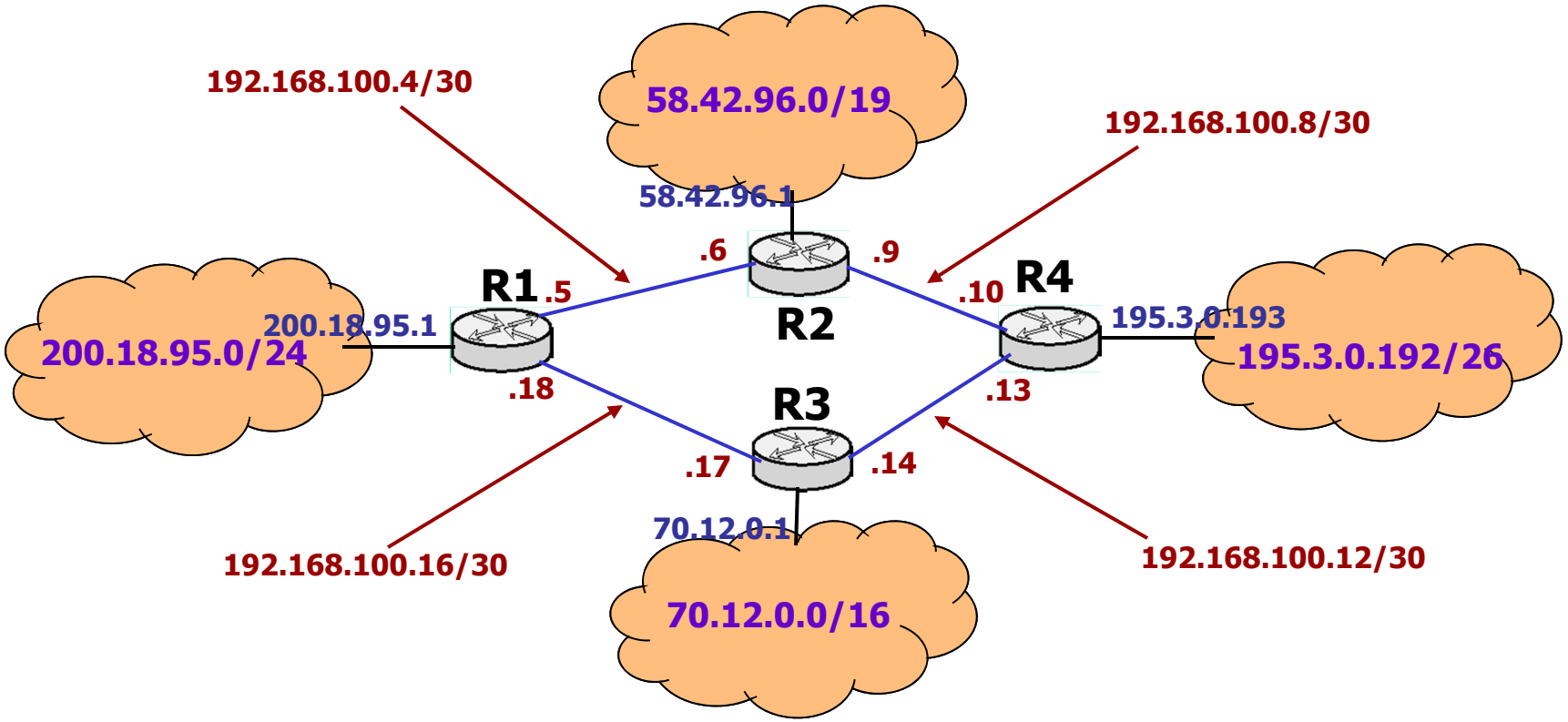
- สายที่เชื่อมต่อโดยตรงระหว่าง Router จะต้องถือเป็นหนึ่ง Network เช่นกัน แต่ Network นี้เป็นแค่ทางผ่านของข้อมูล ไม่จำเป็นต้องกำหนดใน Routing Table



การกำหนด Interface สำหรับ Router

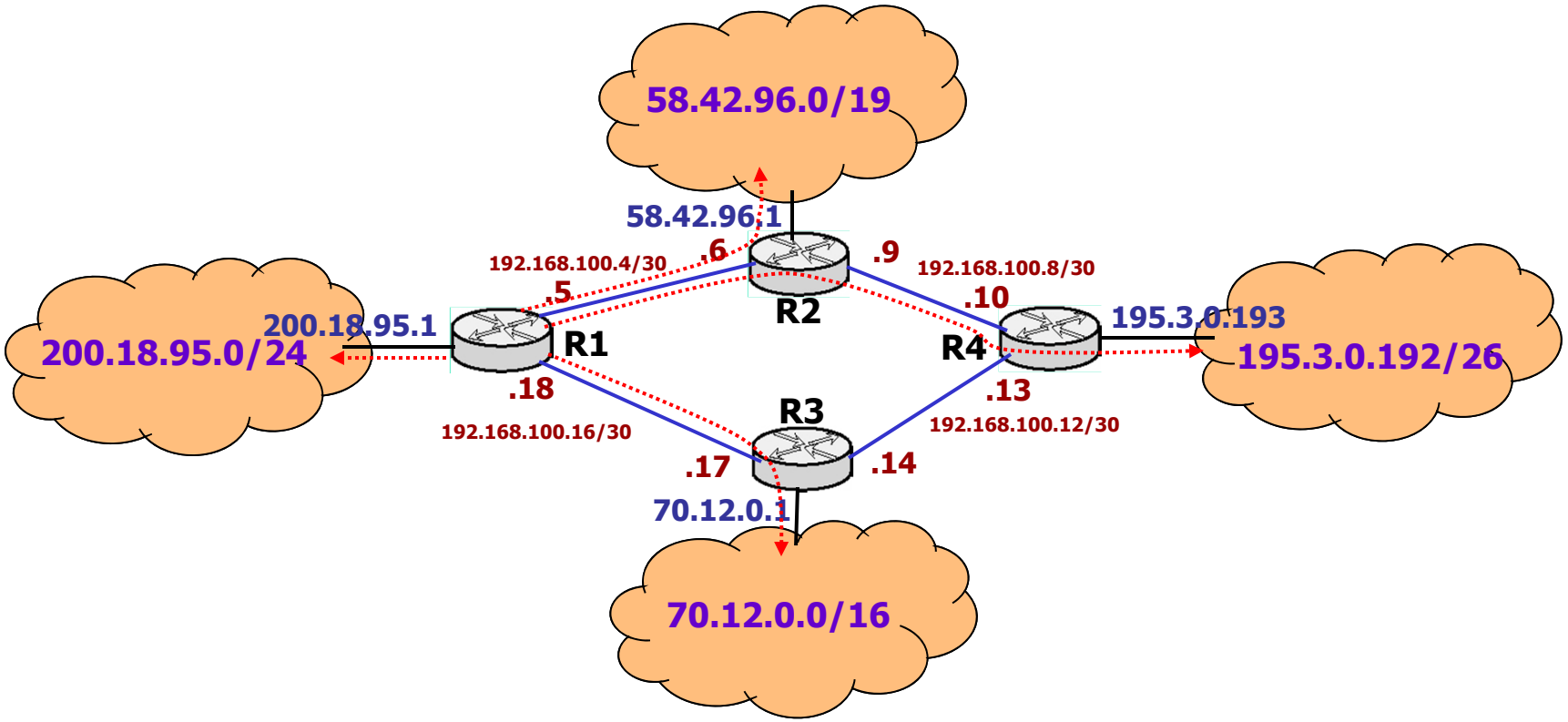
- สายระหว่าง Router สามารถ Subnet จาก Private IP ได้ และต้องการเพียงสอง Host Address ซึ่งปกติจะใช้ /30 ก็พอ เช่น Sub จาก 192.168.100.0/24 จะได้ 64 Subnet

Subnet 192.168.100.4/30 : Host Range 192.168.100.5-192.168.100.6; Broadcast 192.168.100.7



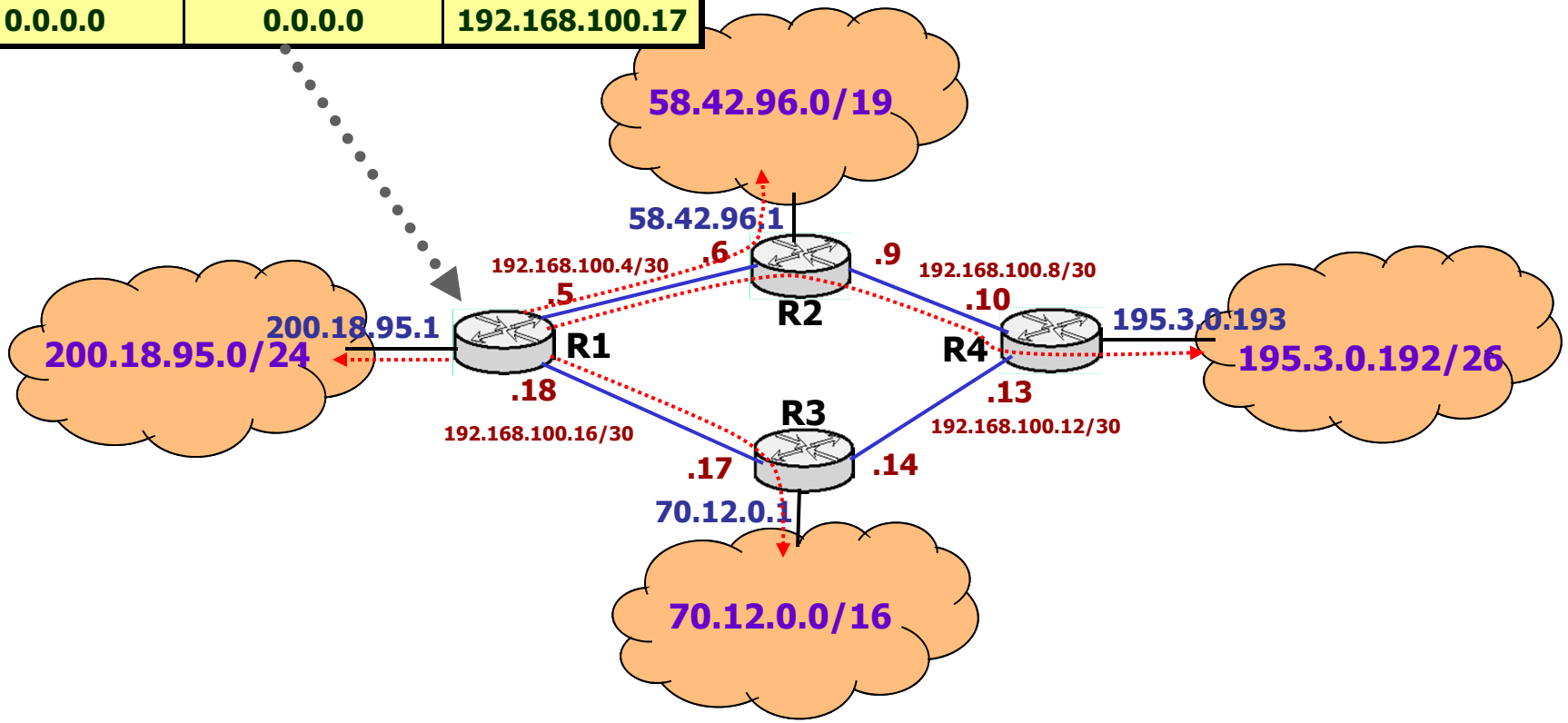
การกำหนด Interface สำหรับ Router

- การกำหนด Static Routing Table ให้กับ Router แต่ละตัวทำได้จากการกำหนดเส้นทางส่งข้อมูลจาก Router ไปยังทุกๆ Network (Network ที่เป็น Transit ไม่ต้องกำหนด เพราะไม่มี Host ปลายทาง) จากนั้นกำหนดเส้นทางหนึ่งให้เป็น Default Route เช่น R1



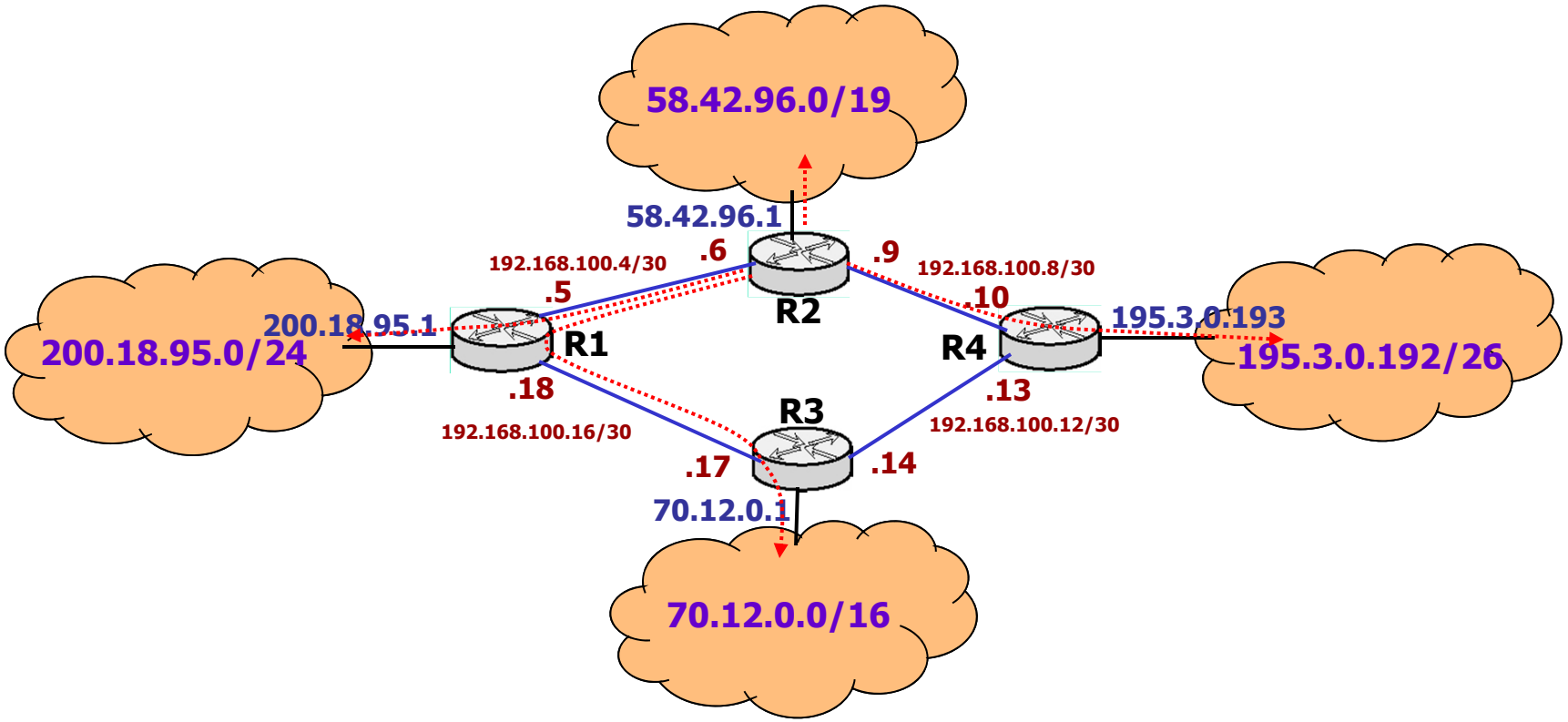
การกำหนด Interface สำหรับ Router

NW	Mask	Next Hop
200.18.95.0	255.255.255.0	Direct
70.12.0.0	255.255.0.0	192.168.100.17
58.42.96.0	255.255.224.0	192.168.100.6
195.3.0.192	255.255.255.192	192.168.100.6
0.0.0.0	0.0.0.0	192.168.100.17



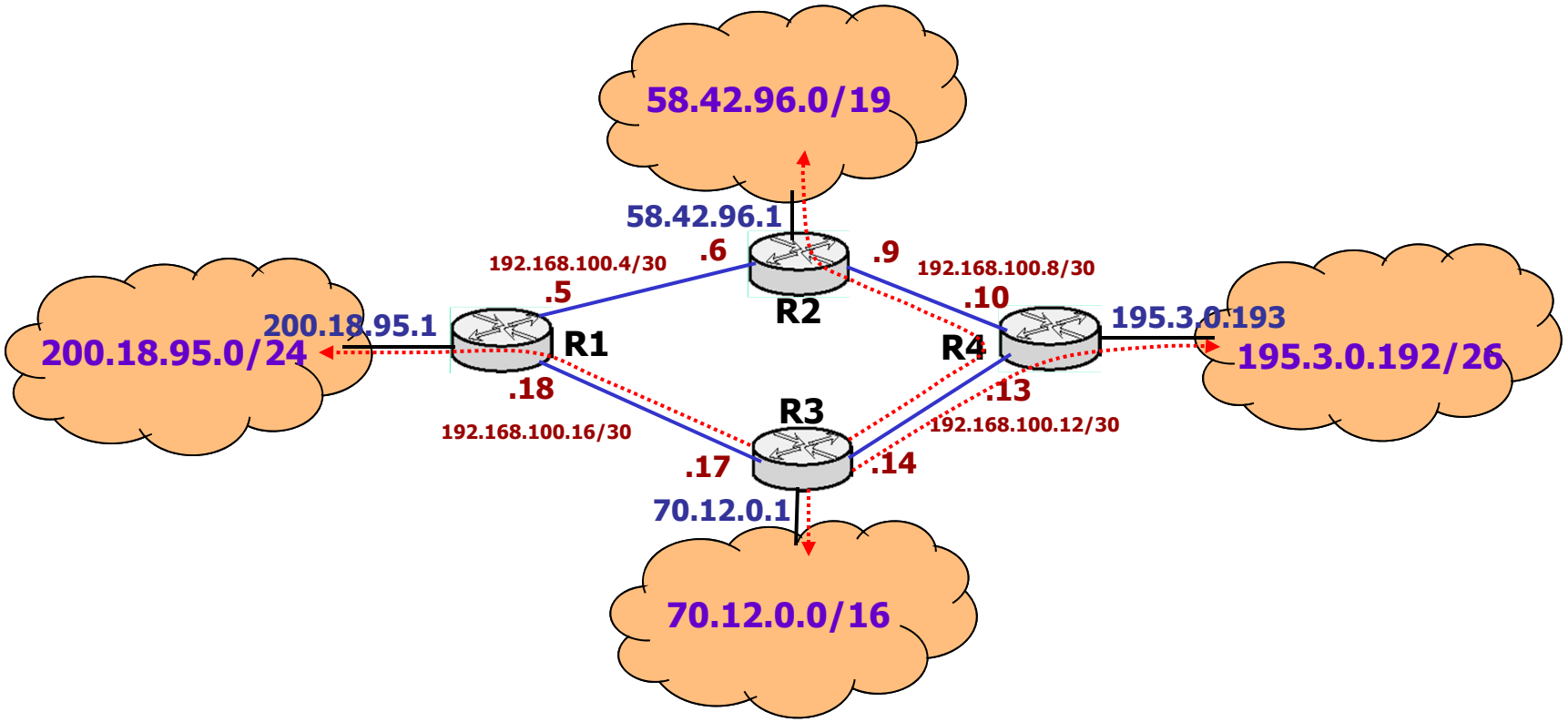
การกำหนด Interface สำหรับ Router

R2 Routing Table



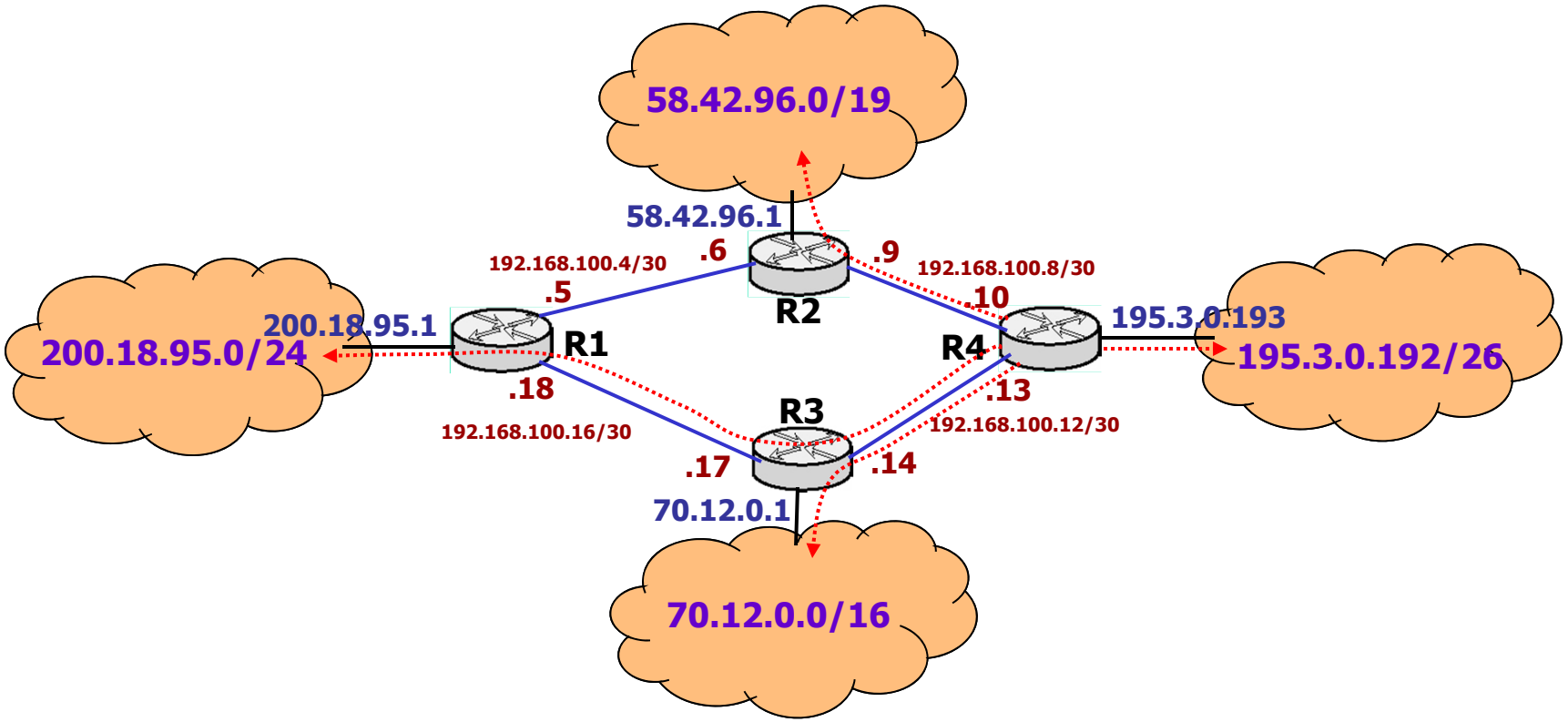
การกำหนด Interface สำหรับ Router

R3 Routing Table



การกำหนด Interface สำหรับ Router

R4 Routing Table



Chapter 27: 27.4 Dynamic Routing and Router

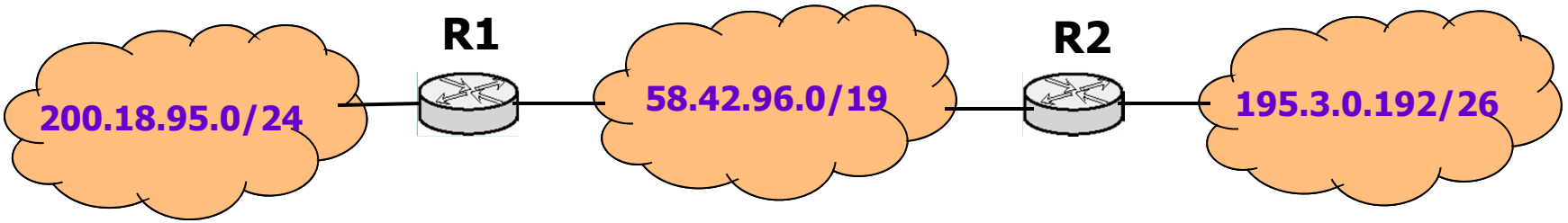
- ปกติ Router ใน Internet จะใช้ Dynamic Routing
  - เพื่อให้มีการแลกเปลี่ยน Routing Information ระหว่างกัน
- Static Routing อาจจะถูกใช้ในกรณีที่ Customer เชื่อมต่อกับ ISP ผ่าน Router ซึ่งในกรณีนี้มีทางออก Internet เพียงทางเดียว และไม่จำเป็นต้องใช้ Dynamic Routing
- หรือ Static Routing อาจจะใช้ภายในองค์กรเพื่อเชื่อมต่อ LAN หลายๆวงภายในอาคาร

Chapter 27: 27.4 Dynamic Routing and Router

- Router จะรู้จัก Network ที่เป็น Direct Connect เท่านั้น
- การที่ Router จะรู้จัก Network อื่น มันจะต้องเรียนรู้มาจาก Router ตัวอื่นที่ต่อโดยตรงกับ Network นั้น
- ใน Static Routing จะไม่มีวิธีที่ Router จะเรียนรู้ได้
- ดังนั้นการเรียนรู้ต้องมีการกำหนดจาก Software ใน Routing Protocol ใน Dynamic Routing
- ด้วยการเรียนรู้นี้เอง ทำให้ Router สามารถปรับตารางของตนได้อย่างเหมาะสม ตามสภาพ Network เป็นผลให้ตาราง Routing เป็น Dynamic

Chapter 27: 27.4 Dynamic Routing and Router

- จากรูป R1 รู้จัก 200.18.95.0/24 และ 58.42.96.0/19 แต่ไม่รู้จัก 195.3.0.192/26
- R2 รู้จัก 58.42.96.0/19 และ 195.3.0.192/26 แต่ไม่รู้จัก 200.18.95.0/24
- R1 และ R2 แลกเปลี่ยนข้อมูลกันผ่าน 58.42.96.0/19 ทำให้ Router แต่ละตัวรู้จัก Network อื่นๆ
- ถ้า 195.3.0.192 เกิดล่ม R2 จะรู้ และสามารถบอกต่อไปยัง R1 ได้ ว่า 195.3.0.192/26 Unreachable
- ถ้า R2 ล่ม ทำให้ R1 ไม่สามารถติดต่อได้ ดังนั้น R1 จะ Mark ว่า 195.3.0.192/26 เป็น Unreachable เช่นกัน



Chapter 27: 27.5 Routing in Global Internet

- **Internet ประกอบด้วย Router มากมาย ถ้าจะให้ Router ทุกตัวแลกเปลี่ยนข้อมูลกับ Router ทุกตัว จะทำให้ Routing Traffic มีมหาศาล**
- **เพื่อจำกัดจำนวน Traffic ใน Internet จะใช้การทำ Routing แบบเป็นลำดับชั้น (Hierarchy) โดยมีการแบ่งกลุ่มของ Router และมีการแลกเปลี่ยนข้อมูลภายในกลุ่ม จากนั้นจะมี Router ที่เป็นตัวแทนของกลุ่มทำการแลกเปลี่ยนข้อมูลกับภายนอก**
- **Routing Protocol ภายในกลุ่ม จะแตกต่างจาก Routing Protocol ระหว่างกลุ่ม**
  - ขนาดของกลุ่มจะไม่จำกัด ขึ้นอยู่กับขนาดขององค์กร
  - แต่ละองค์กรที่เป็นเจ้าของกลุ่ม มีสิทธิจะเลือก Routing Protocol อย่างไรก็ได้ ที่อยู่ภายในกลุ่ม
  - แต่ Routing Protocol ที่ใช้แลกเปลี่ยนข้อมูลระหว่างกลุ่มจะต้องเป็น Protocol เดียวกัน

Chapter 27: 27.6 Autonomous System Concept

- **แต่ละกลุ่มของ Router ที่ดูแลจัดการโดยองค์กรเดียว เราเรียกว่า Autonomous System (AS)**
  - อาจจะเป็นหนึ่งองค์กร หรือ หนึ่ง ISP
  - องค์กรหนึ่งอาจจะแบ่งกลุ่มของ Router เป็นหลาย AS ก็ได้
  - Router ภายใน AS จะมีการแลกเปลี่ยน Routing Information กัน

Chapter 27: 27.7 Two Type of Internet Routing Protocol

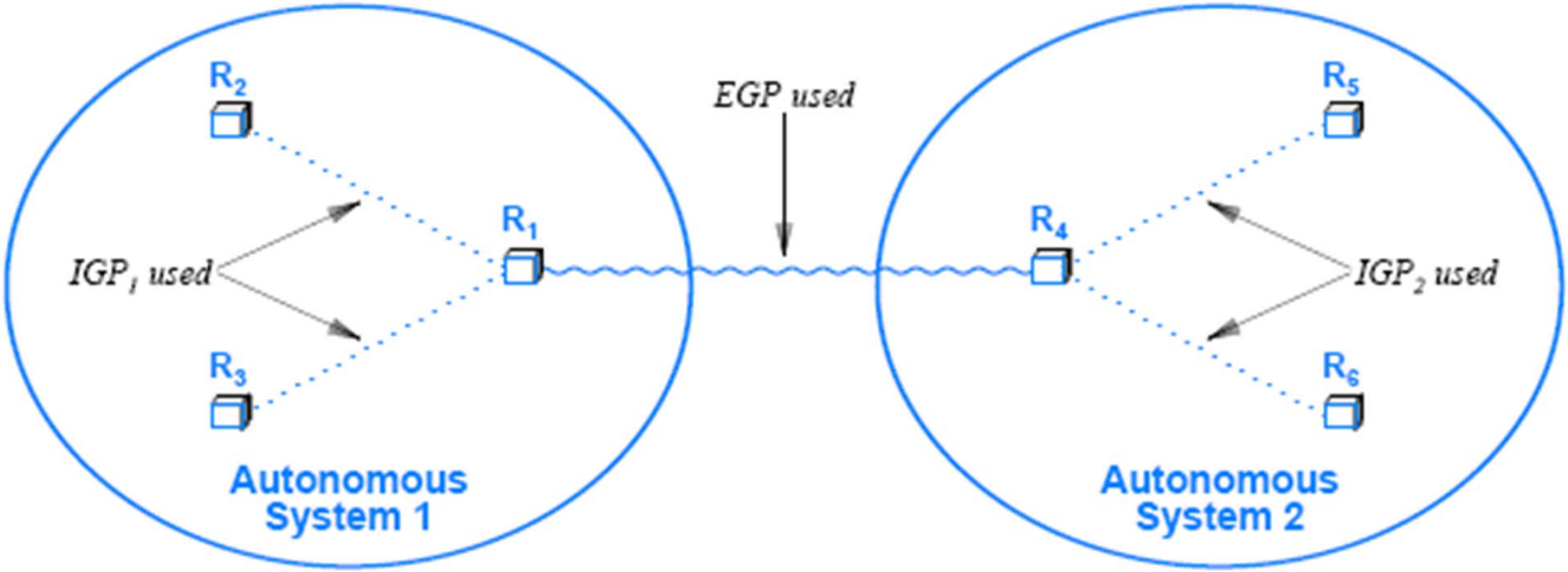
- **27.7.1 Interior Gateway Protocols(IGP)**
  - Router ภายใน AS จะใช้ Interior Gateway Protocol (IGP) ในการแลกเปลี่ยน Routing Information
  - แต่ละ AS มีสิทธิ์ที่จะเลือก IGP อันใดก็ได้
    - RIP
    - OSPF
    - IGRP (Cisco)
    - EIGRP (Cisco)

Chapter 27: 27.7 Two Type of Internet Routing Protocol

- **27.7.2 Exterior Gateway Protocols(EGP)**
  - Router ในแต่ละ AS จะใช้ EGP ในการแลกเปลี่ยน Routing Information กับ Router ในอีก AS หนึ่ง
  - EGP ปกติจะซับซ้อนกว่า IGP แต่ว่าการใช้งานจะยืดหยุ่นกว่า และมี Overhead ต่ำกว่า
  - EGP จะสรุป Routing Information ในแต่ละ AS ก่อนที่จะส่งไปให้อีก AS หนึ่ง
    - การส่ง Routing Information ออกนอก AS สามารถกำหนดว่าข้อมูลใดให้ส่ง หรือไม่ให้ส่งได้

Chapter 27: 27.7 Two Type of Internet Routing Protocol

- **27.7.3 ตัวอย่างการใช้งาน IGP และ EGP**
  - Router R1 จะ Run ทั้ง IGP1 และ EGP
  - Router R4 จะ Run ทั้ง IGP2 และ EGP

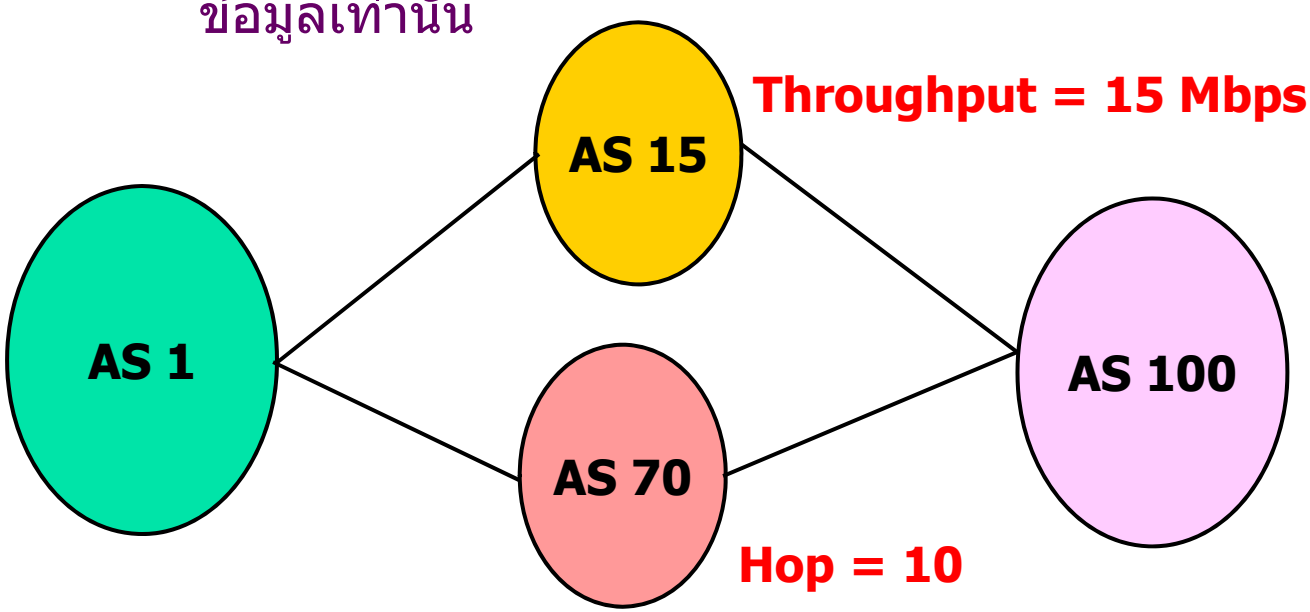


Chapter 27: 27.7 Two Type of Internet Routing Protocol

- **27.7.4 Optimum Routes, Routing Metrics and IGP**
  - ปกติเส้นทางส่งข้อมูลใน Internet จะมีหลายเส้นทาง
  - Router จะเลือกเส้นทางที่ดีที่สุด (Optimal Routes)
    - Remote Application อาจจะเป็นเส้นทางที่ Delay ต่ำสุด
    - Web Application อาจเป็นเส้นทางที่ Throughput สูงสุด
    - Webcast หรือ Real-Time อาจเป็นเส้นทางที่ Jitter ต่ำสุด
  - เราใช้คำว่า Routing Metric เป็นตัววัดราคาของการส่งในแต่ละเส้นทาง
    - อาจจะวัดจาก Delay, Throughput หรือ Jitter หรือผสมกัน
  - แต่ปกติใน Internet จะใช้ Metric สองตัวร่วมกัน
    - Hop Count (จำนวน Network หรือ Router ที่ผ่าน)
    - Administrative Cost (กำหนดเองจาก Administrator) เพื่อควบคุมให้เส้นทางส่งข้อมูลเป็นไปตามต้องการ
      - เช่นกำหนดเส้นทาง 4 Hop ให้มี Administrative Cost ต่ำกว่าเส้นทางสอง Hop เพื่อบังคับให้ข้อมูลส่งไปทางนี้

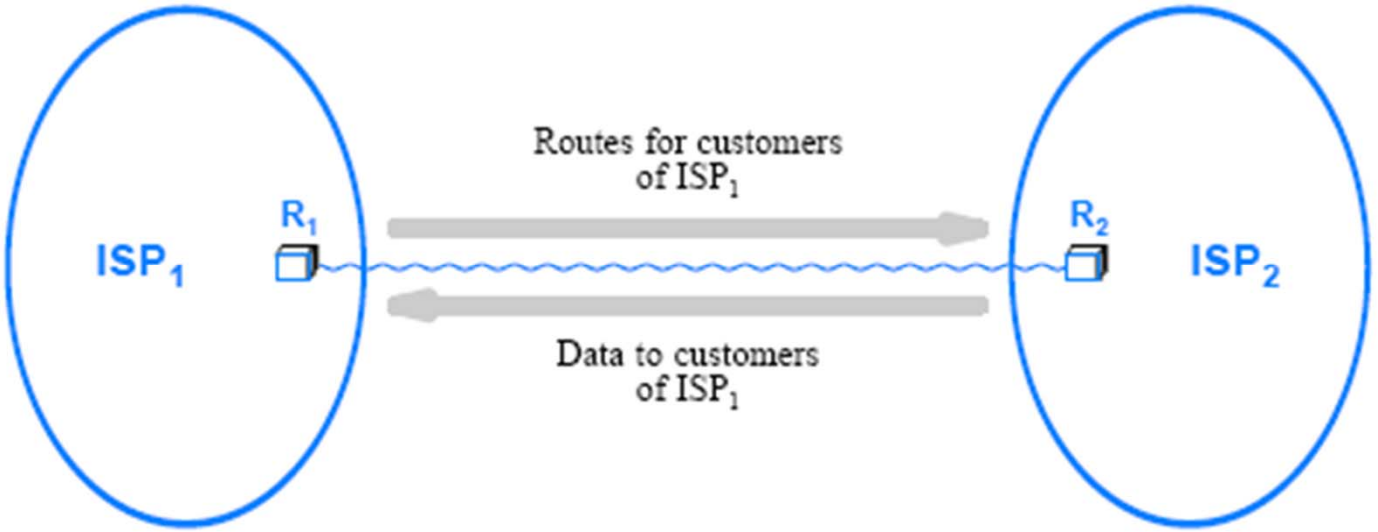
Chapter 27: 27.7 Two Type of Internet Routing Protocol

- **27.7.4 Optimum Routes, Routing Metrics and IGP**
  - การหาเส้นทางของ IGP จะใช้ Routing Metric
  - EGP จะไม่ใช้
    - เนื่องจากแต่ละ AS ใช้ IGP ต่างกัน และใช้ Metric ต่างกัน ไม่สามารถเปรียบเทียบได้
  - ดังนั้น EGP จะไม่พยายามหา Optimal Path มันเพียงหาเส้นทางส่งข้อมูลเท่านั้น



Chapter 27: 27.8 Routes and Data Traffic

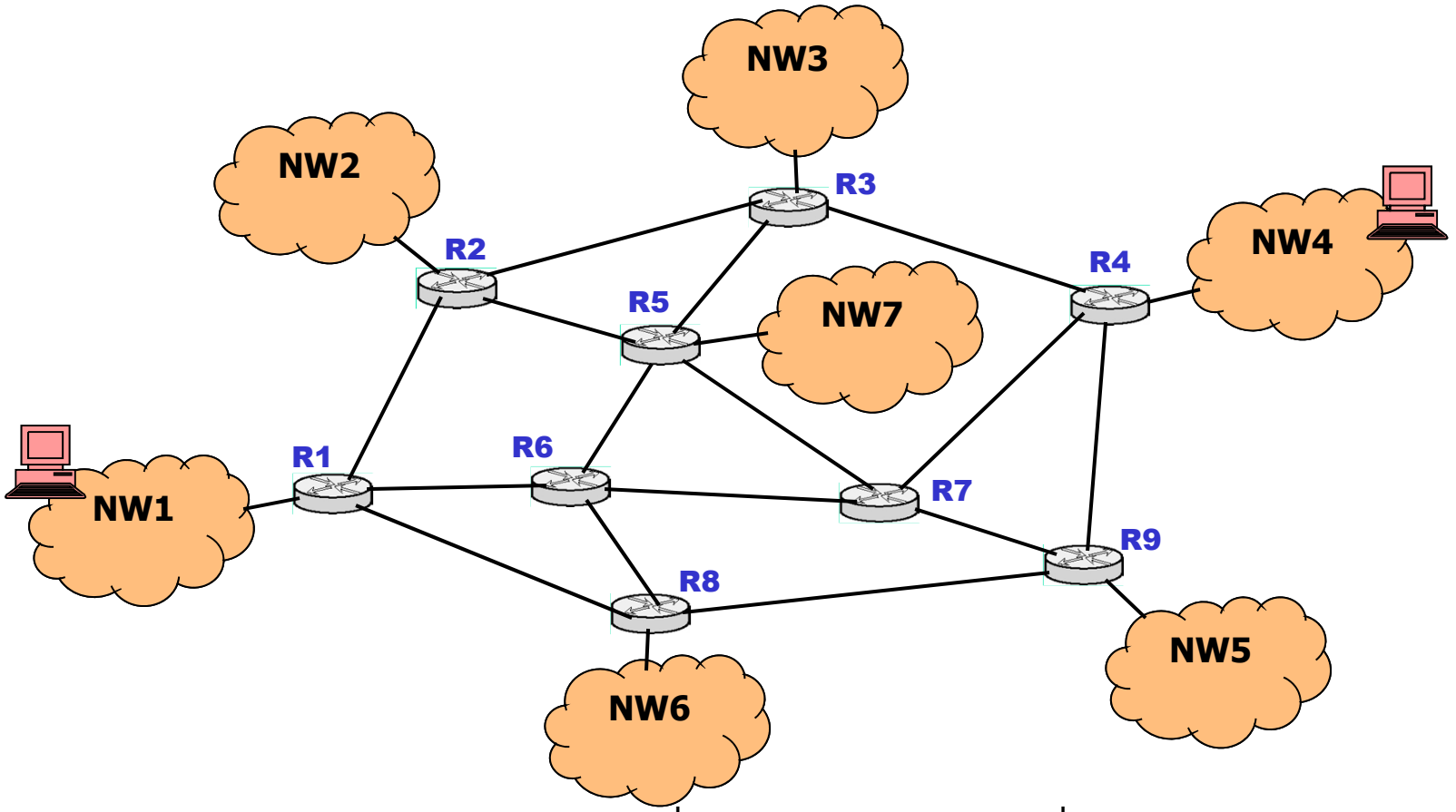
- **Data Traffic จะมีทิศทางการไหลสวนทางกับ Routing Traffic**



Least Cost Path Algorithms

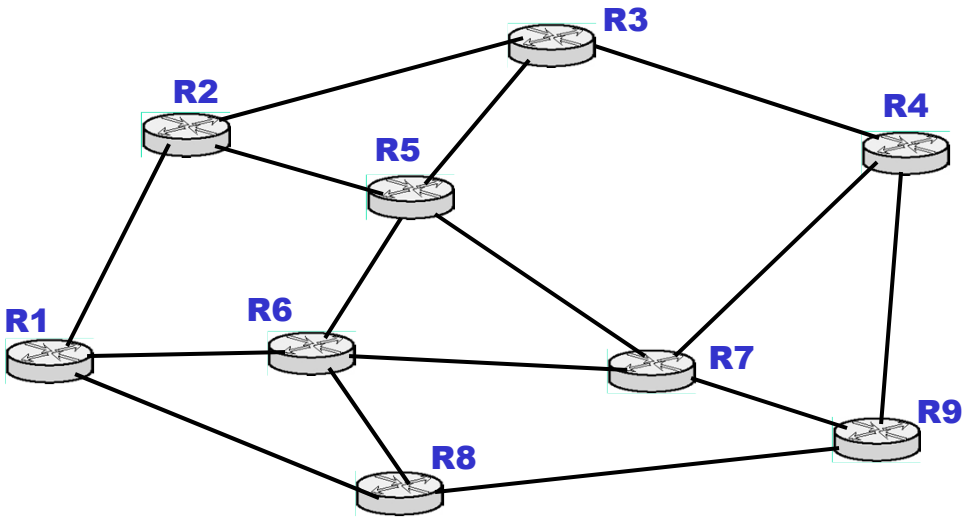
- ดูรายละเอียดบทที่ 18 18.12-18.13
- ดูจาก Course Notes วิชา CPE 326
- ดูจาก Course Notes วิชา CPE 231
  - เรื่อง Graph
- มีสอง Algorithm ที่ให้คำตอบเหมือนกัน แต่ใช้วิธีคำนวณต่างกัน
  - ดังนั้นจะใช้ข้อมูลจาก Routing Information ต่างกัน
  - กำเนิดเป็นวิธีการ Routing สองแบบ
    - Distance Vector Routing จะใช้ Bellman-Ford Algorithm
      - ในการนี้ Router จะแลกเปลี่ยนตาราง Routing Table เฉพาะกับเพื่อนบ้านเท่านั้น ข้อมูลจะ Propagate ทีละ Router เมื่อถึงเวลาแลกตาราง เช่น RIP(Routing Information Protocol)
    - Link-State Routing จะใช้ Dijkstra Algorithm
      - ในการนี้ Router จะส่ง Link State ของตนเอง (เฉพาะ Direct Connect) ไปให้กับทุกๆ Router แต่ละ Router จะรวบรวม Link State Database และสร้าง Topology ของทั้ง Network จากนั้นมันจะสร้าง Shortest Path First Tree โดยตัวมันเป็น Root ไปยัง Router ทุกๆตัว

Least Cost Path Algorithms



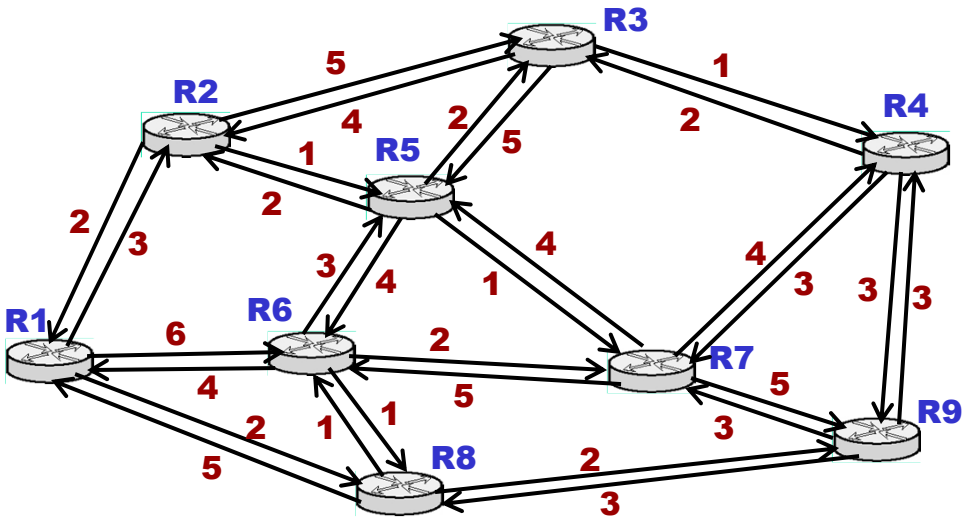
1. การส่งข้อมูลจาก Host หนึ่ง ไปยังอีก Host หนึ่ง กระทำผ่าน Router ถ้าอยู่คนละ Network การหาเส้นทางคือหาเส้นทางจาก Router หนึ่งไปยังอีก Router หนึ่ง ที่ Network นั้นเชื่อมต่ออยู่

Least Cost Path Algorithms



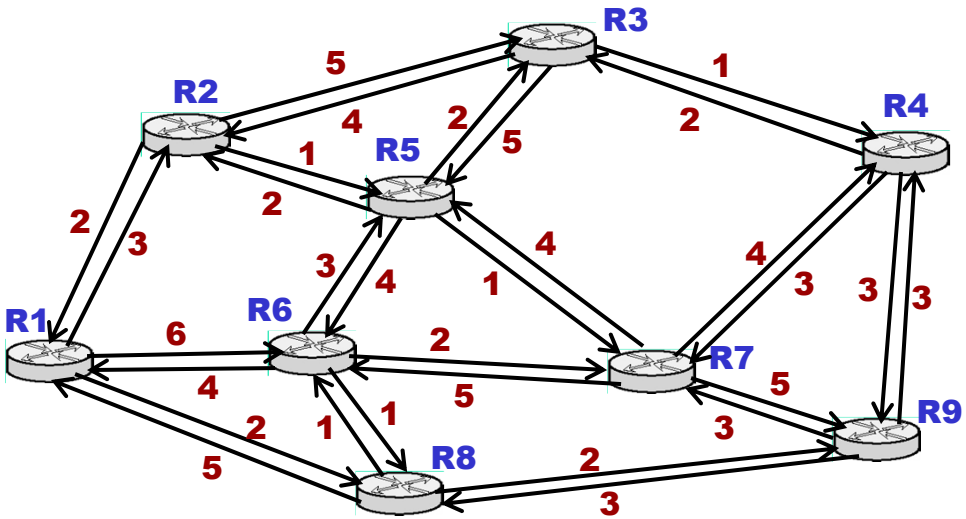
2. Cost ที่ส่งระหว่าง Router ไปและกลับไม่จำเป็นต้องเท่ากัน เพราะขึ้นอยู่กับ Queue ที่ Interface ของ Router ต้นทาง ไม่ใช่ปลายทาง เราแสดงค่า Cost ระหว่าง Router ที่มีเส้นเชื่อมต่อ โดยใช้ Cost Matrix

Least Cost Path Algorithms



2. Cost ที่ส่งระหว่าง Router ไปและกลับไม่จำเป็นต้องเท่ากัน เพราะขึ้นอยู่กับ Queue ที่ Interface ของ Router ต้นทาง ไม่ใช่ปลายทาง เราแสดงค่า Cost ระหว่าง Router ที่มีเส้นเชื่อมต่อ โดยใช้ Cost Matrix (W)

Least Cost Path Algorithms



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

2. Cost ที่ส่งระหว่าง Router ไปและกลับไม่จำเป็นต้องเท่ากัน เพราะขึ้นอยู่กับ Queue ที่ Interface ของ Router ต้นทาง ไม่ใช่ปลายทาง เราแสดงค่า Cost ระหว่าง Router ที่มีเส้นเชื่อมต่อ โดยใช้ Cost Matrix (W)

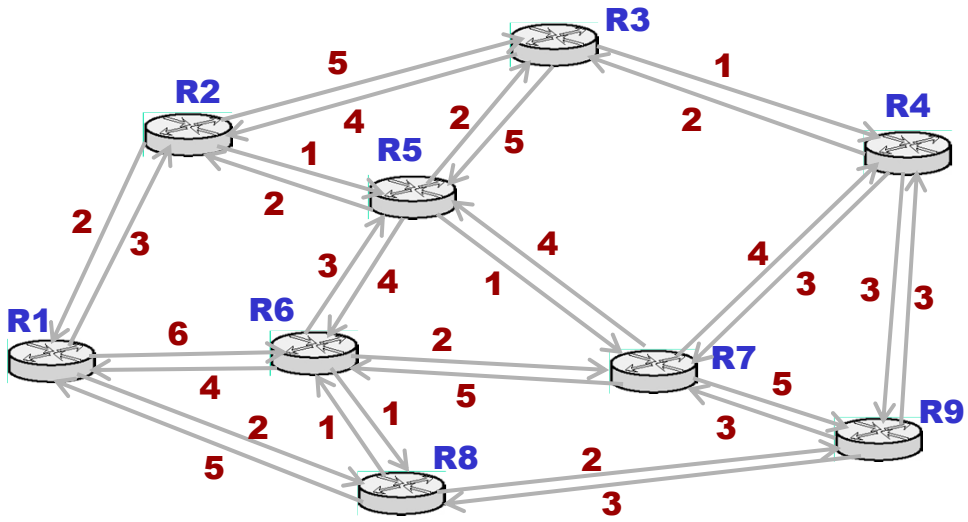
Bellman-Ford Algorithm Definitions

- **Find shortest paths from given node subject to constraint that paths contain at most one link**
- **Find the shortest paths with a constraint of paths of at most two links**
- **And so on**
- **s = source node**
- **w(i, j) = link cost from node i to node j**
  - w(i, i) = 0
  - w(i, j) = ∞ if the two nodes are not directly connected
  - w(i, j) ≥ 0 if the two nodes are directly connected
- **h = maximum number of links in path at current stage of the algorithm**
- **$L_h(n)$ = cost of least-cost path from s to n under constraint of no more than h links**

Bellman-Ford Algorithm Method

- **Step 1 [Initialization]**
  - $L_0(n) = \infty$, for all $n \neq s$
  - $L_h(s) = 0$, for all h
- **Step 2 [Update]**
- **For each successive $h \geq 0$**
  - For each $n \neq s$, compute
  - $L_{h+1}(n) = \min_j [L_h(j) + w(j,n)]$; $\forall$j
- **Connect n with predecessor node j that achieves minimum**
- **Eliminate any connection of n with different predecessor node formed during an earlier iteration**
- **Path from s to n terminates with link from j to n**

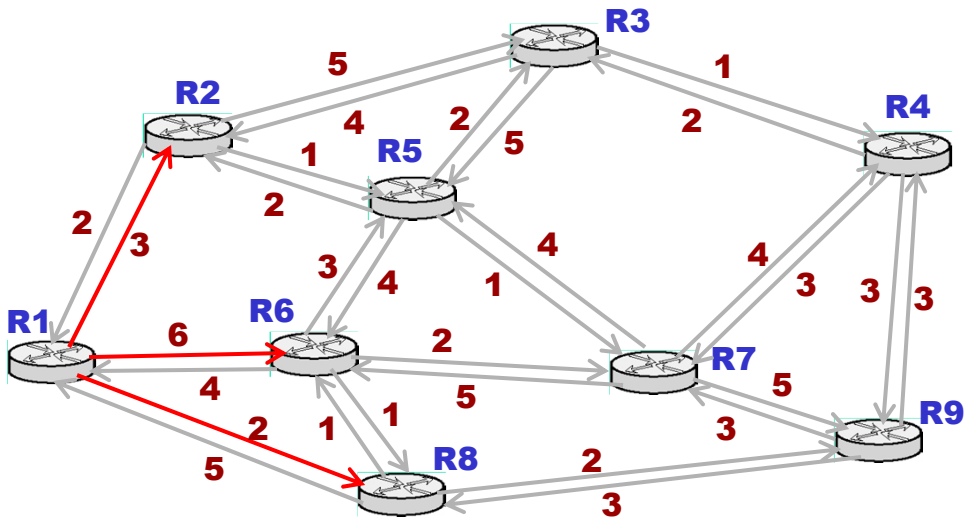
Example: Bellman Ford จาก Node 1 (h=0: Initialization: s=1)



Cost R1 ไปทุกๆ Node = Infinity
$L_0(n)=\infty$; n = 2,..,9: $L_0(1)=0$

	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Example: Bellman Ford จาก Node 1 (h=1: Initialization: s=1)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไปทุกๆ Node = Infinity

$L_0(n)=\infty$; n = 2,..,9: $L_0(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_1(2)=\min_j[L_0(j)+w(j,2)]$; j = 1,..,9

$=\min[L_0(1)+w(1,2),L_0(2)+w(2,2),L_0(3)+w(3,2),\ldots,L_0(9)+w(9,2)]$

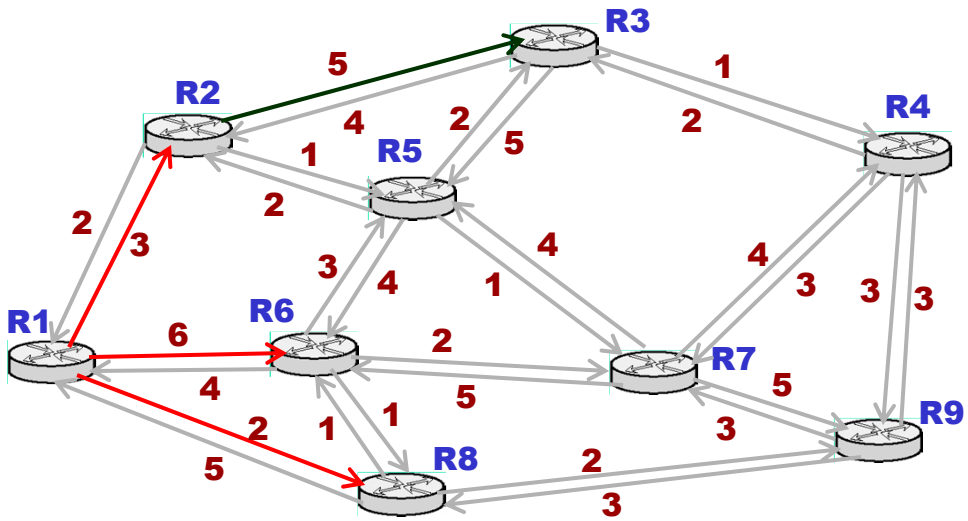
=min j = 1, path = 1-2, cost = 3

$L_1(3)=\min j[L_0(j)+w(j,3)]$; j = 1,..,9

$=\min[L_0(1)+w(1,3),L_0(2)+w(2,3),L_0(3)+w(3,3),\ldots,L_0(9)+w(9,3)]$

=all infinity

Example: Bellman Ford จาก Node 1 (h=2: n=2,3)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไป Node 3,4,5,7,9= Infinity

$L_1(2)=3$; $L_1(6)=6$; $L_1(8)=2$; : $L_1(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_2(2)=\min_j[L_1(j)+w(j,2)]$; j = 1,..,9

$=\min[L_1(1)+w(1,2),L_1(2)+w(2,2),L_1(3)+w(3,2),\ldots,L_1(9)+w(9,2)]$

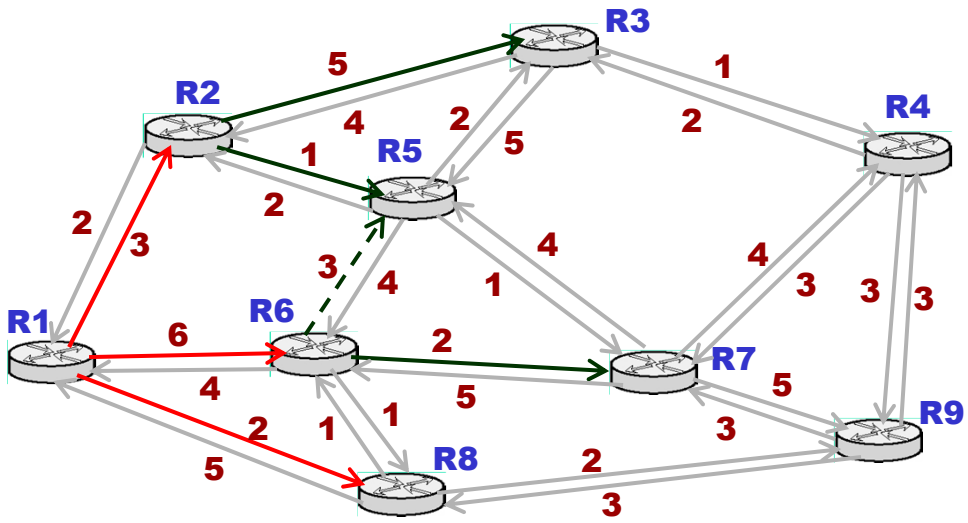
=min j = 1, path = 1-2, cost = 3

$L_2(3)=\min j[L_1(j)+w(j,3)]$; j = 1,..,9

$=\min[L_1(1)+w(1,3),L_1(2)+w(2,3),L_1(3)+w(3,3),\ldots,L_1(9)+w(9,3)]$

=min j=2, path = 1-2 plus 2-3 = 1-2-3, cost = 3+5=8

Example: Bellman Ford จาก Node 1 (h=2: n=5,7)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไป Node 3,4,5,7,9= Infinity

$L_1(2)=3$; $L_1(6)=6$; $L_1(8)=2$; : $L_1(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_2(5)=\min_j[L_1(j)+w(j,5)]$; j = 1,..,9

$=\min[L_1(1)+w(1,5),L_1(2)+w(2,5),L_1(3)+w(3,5),\ldots,L_1(6)+w(6,5),..,L_1(9)+w(9,5)]$

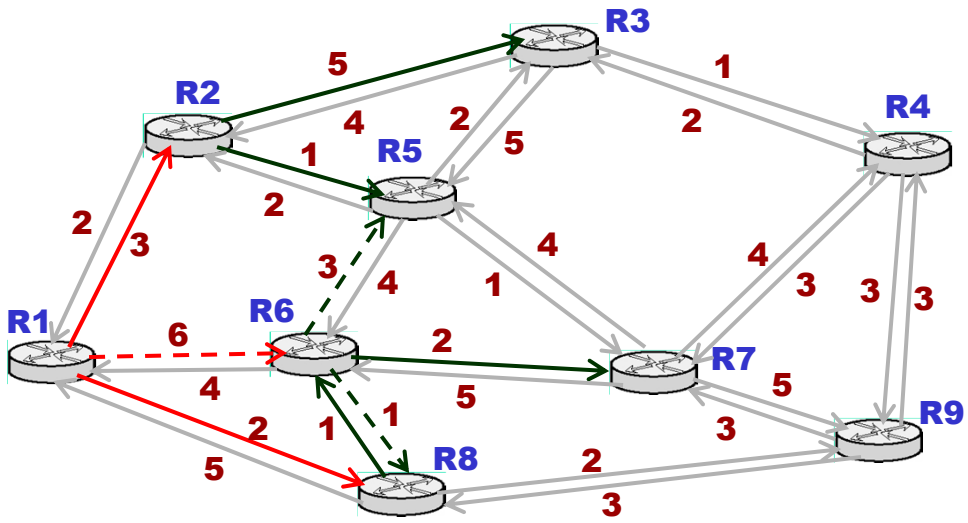
$=\min(3+1,6+3)=4$ (j=2, path = 1-2-5, cost = 4)

$L_2(7)=\min j[L_1(j)+w(j,7)]$; j = 1,..,9

$=\min[L_1(1)+w(1,7),L_1(2)+w(2,7),\ldots,L_1(6)+w(6,7),\ldots,L_1(9)+w(9,7)]$

=min j=6, path = 1-6 plus 6-7 = 1-6-7, cost = 6+2=8

Example: Bellman Ford จาก Node 1 (h=2: n=6)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไปทุกๆ Node = Infinity

$L_1(2)=3$; $L_1(6)=6$; $L_1(8)=2$; : $L_1(1)=0$

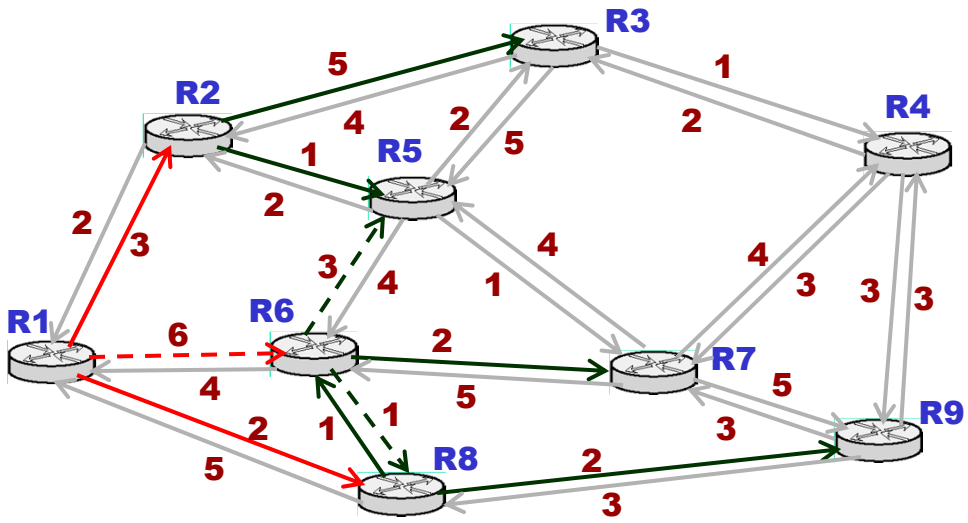
คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_2(6)=\min_j[L_1(j)+w(j,6)]$; j = 1,..,9

$=\min[L_1(1)+w(1,6),\ldots, L_1(8)+w(8,6),..,L_1(9)+w(9,6)]$

= j=8; Path 1-8+8-6 ถูกกว่า Path 1-6

Example: Bellman Ford จาก Node 1 (h=2: n=8)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไปทุกๆ Node = Infinity

$L_1(2)=3$; $L_1(6)=6$; $L_1(8)=2$; : $L_1(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

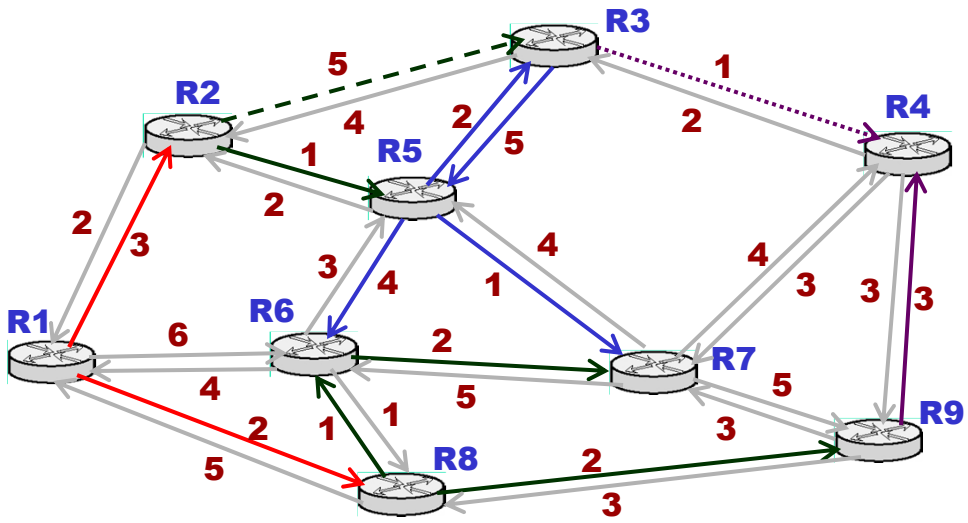
$L_2(8)=\min_j[L_1(j)+w(j,8)]$; j = 1,..,9

$=\min[L_1(1)+w(1,8),\ldots, L_1(6)+w(6,8),..,L_1(9)+w(9,8)]$

=Path ไม่เปลี่ยน

กรณี n=9 พบว่า Path 1-8-9 มีอันเดียวที่มี 2 Hop

Example: Bellman Ford จาก Node 1 (h=3)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไป Node 4 = Infinity

$L_2(2)=3$; $L_2(3)=7$; $L_2(5)=4$; $L_2(6)=6$
$L_2(7)=8$; L2(8)=2; $L_2(9)=4$: $L_2(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_3(5)=\min_j[L_2(j)+w(j,5)]$; j = 1,..,9

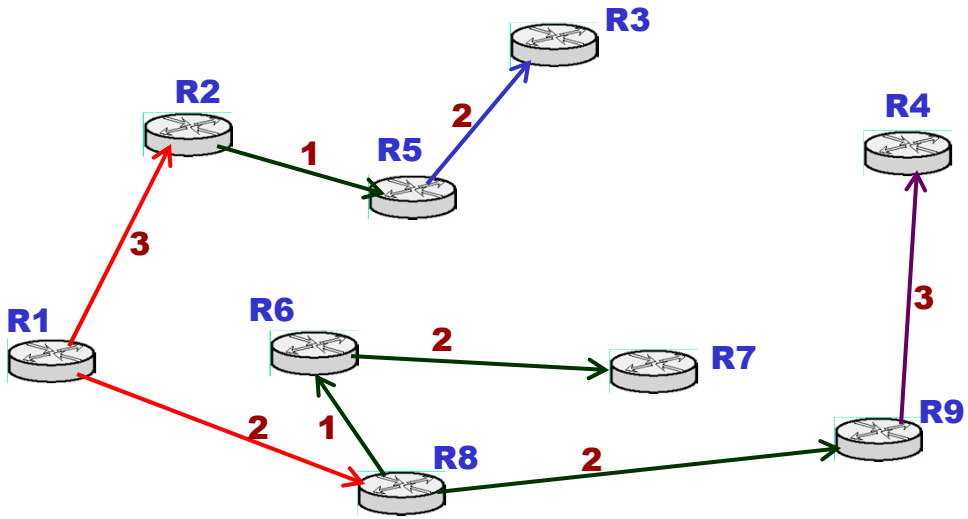
$=\min[L_2(1)+w(1,5),L_2(2)+w(2,5),L_2(3)+w(3,5),..,L_1(9)+w(9,5)]$

=Path ไม่เปลี่ยน

Path ไป R7 ไม่เปลี่ยน แม้ว่า Path 1-2-5-7 จะเท่ากัน

Path ไป R3 เปลี่ยน จาก 1-2-3 เป็น 1-2-5-3, Path R4 = min(1-2-3-4,1-8-9-4)

Example: Bellman Ford จาก Node 1 (h=4)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Algorithm จะหยุดเมื่อถึงจำนวน Hop สูงสุดของ NW
Algorithm สามารถคำนวณได้แม้ว่าจะไม่รู้ Topology โดยการรับข้อมูลจาก Node ข้างเคียง แล้วมา Update ตารางของตัวเอง โดยใช้ค่า Cost ที่ต่ำกว่า

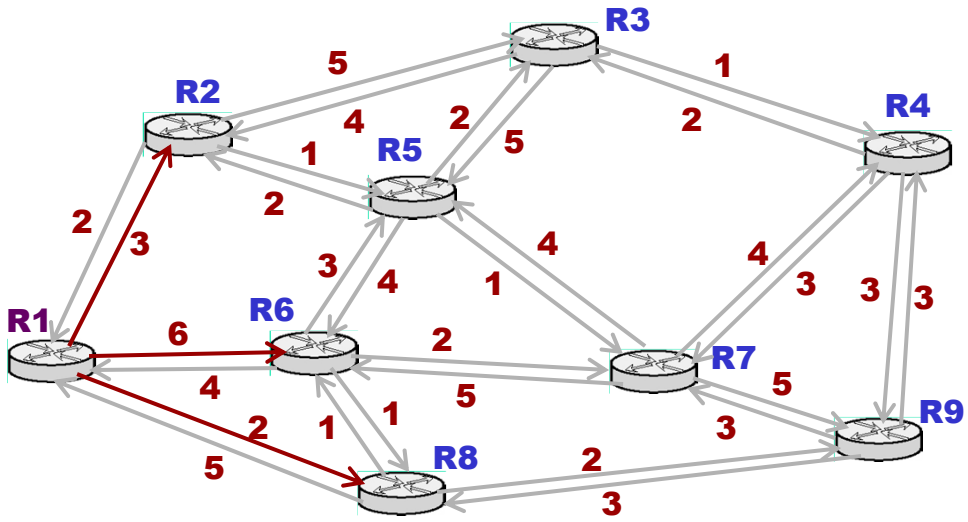
Dijkstra's Algorithm Definitions

- **Find shortest paths from given source node to all other nodes, by developing paths in order of increasing path length**
- **N = set of nodes in the network**
- **s = source node**
- **T = set of nodes so far incorporated by the algorithm**
- **w(i, j) = link cost from node i to node j**
  - w(i, i) = 0
  - w(i, j) = ∞ if the two nodes are not directly connected
  - w(i, j) ≥ 0 if the two nodes are directly connected
- **L(n) = cost of least-cost path from node s to node n currently known**
  - At termination, L(n) is cost of least-cost path from s to n

Dijkstra's Algorithm Method

- **Step 1 [Initialization]**
  - T = {s} Set of nodes so far incorporated consists of only source node
  - L(n) = w(s, n) for n ≠ s
  - Initial path costs to neighboring nodes are simply link costs
- **Step 2 [Get Next Node]**
  - Find neighboring node not in T with least-cost path from s
  - Incorporate node into T
  - Also incorporate the edge that is incident on that node and a node in T that contributes to the path
- **Step 3 [Update Least-Cost Paths]**
  - L(n) = min[L(n), L(x) + w(x, n)] for all n ∉ T
  - If latter term is minimum, path from s to n is path from s to x concatenated with edge from x to n
- **Algorithm terminates when all nodes have been added to T**

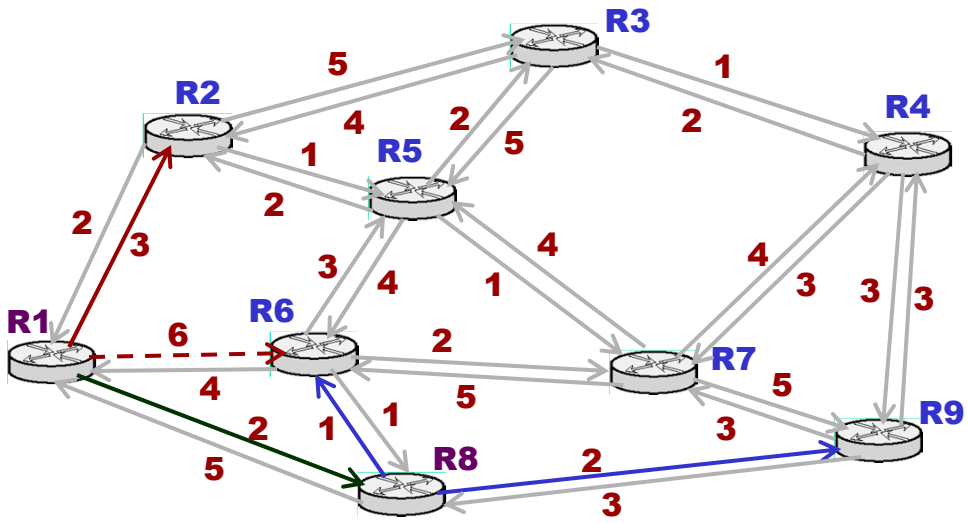
Example: Dijkstra จาก Node 1 (T={1})



T={1}, L(2)=3,L(3)=inf,L(4)=inf, L(5)=inf,L(6)=6,L(7)=inf,L(8)=2, L(9)=inf

	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Example: Dijkstra จาก Node 1 (T={1})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

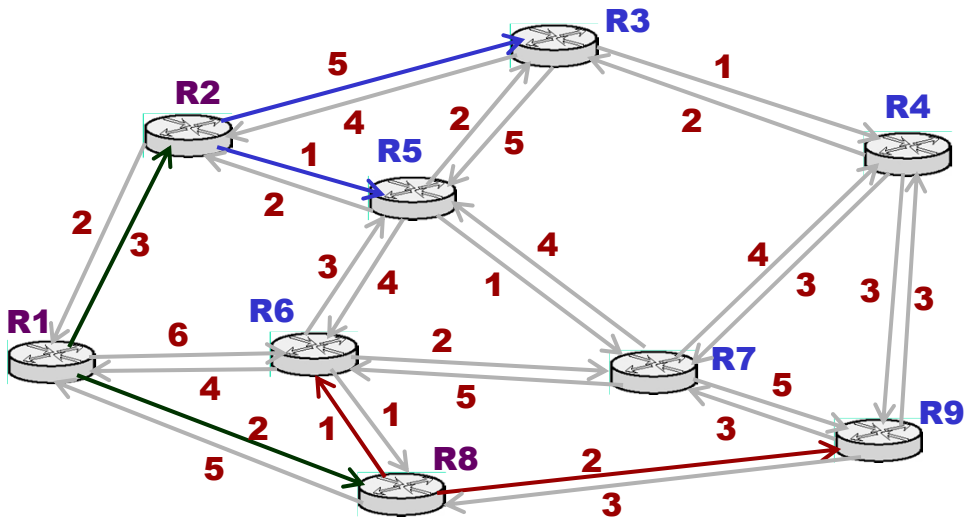
T={1}, L(2)=3,L(3)=inf,L(4)=inf, L(5)=inf,L(6)=6,L(7)=inf, L(8)=2, L(9)=inf

Min = L(8) ดังนั้นนำ Node 8 ใส่ใน T; T={1,8}

L(n) = min[L(n), L(8) + w(8, n)] for all n not in T

T={1,8}, L(2)=3,L(3)=inf,L(4)=inf, L(5)=inf, L(6)=3, L(7)=inf,L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,8})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

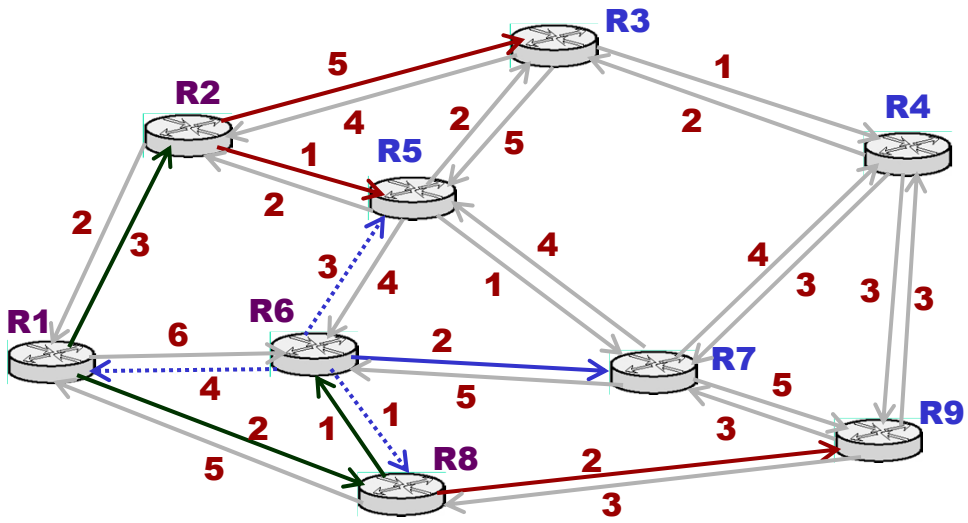
T={1,8}, L(2)=3, L(3)=inf,L(4)=inf, L(5)=inf,L(6)=3,L(7)=inf, L(8)=2, L(9)=4

Min = L(2) ดังนั้นนำ Node 2 ใส่ใน T; T={1,2,8}

L(n) = min[L(n), L(2) + w(2, n)] for all n not in T

T={1,2,8}, L(2)=3, L(3)=8, L(4)=inf, L(5)=4, L(6)=3, L(7)=inf, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,8})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

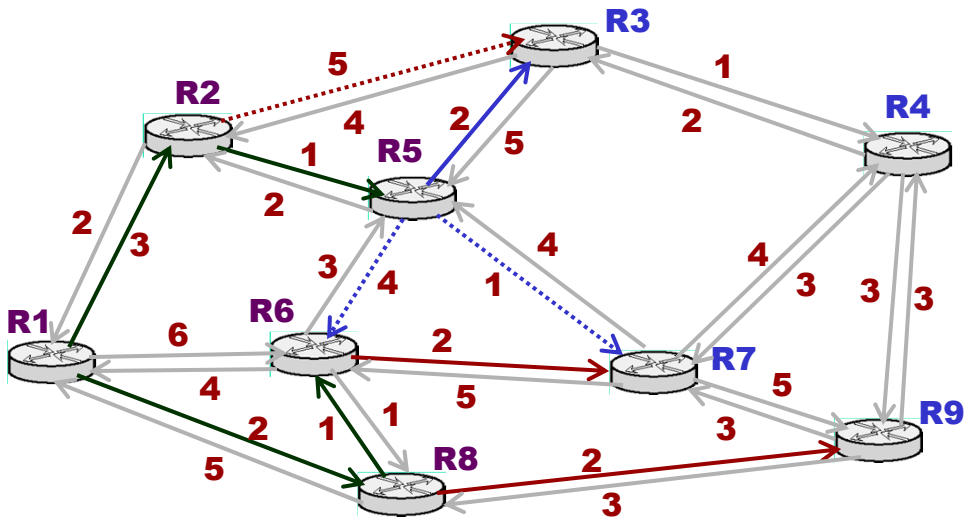
T={1,2,8}, L(2)=3, L(3)=8,L(4)=inf, L(5)=4, L(6)=3, L(7)=inf, L(8)=2, L(9)=4

Min = L(6) ดังนั้นนำ Node 6 ใส่ใน T; T={1,2,6,8}

L(n) = min[L(n), L(6) + w(6, n)] for all n not in T

T={1,2,6,8}, L(2)=3, L(3)=8, L(4)=inf, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,6,8})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

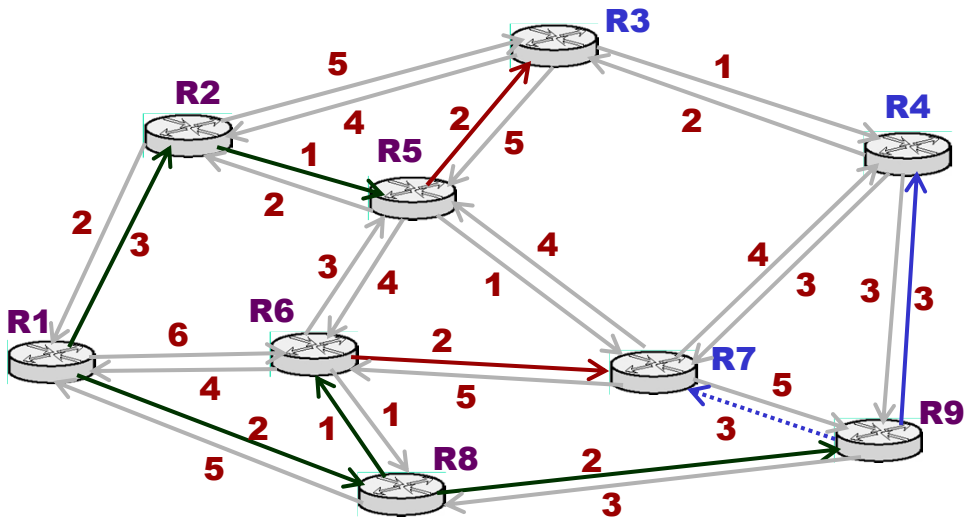
T={1,2,6,8}, L(2)=3, L(3)=8,L(4)=inf, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(5) หรือ L(9) ก็ได้ ดังนั้นนำ Node 5 ใส่ใน T; T={1,2,5,6,8}

L(n) = min[L(n), L(5) + w(5, n)] for all n not in T

T={1,2,5,6,8}, L(2)=3, L(3)=6, L(4)=inf, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,5,6,8})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

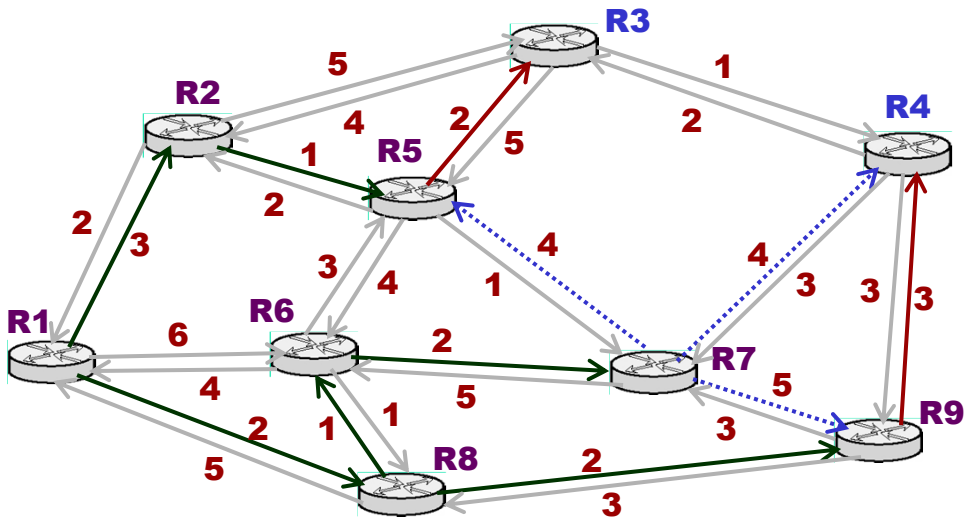
T={1,2,5,6,8}, L(2)=3, L(3)=6, L(4)=inf, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(9) ดังนั้นนำ Node 9 ใส่ใน T; T={1,2,5,6,8,9}

L(n) = min[L(n), L(9) + w(9, n)] for all n not in T

T={1,2,5,6,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,5,6,8,9})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

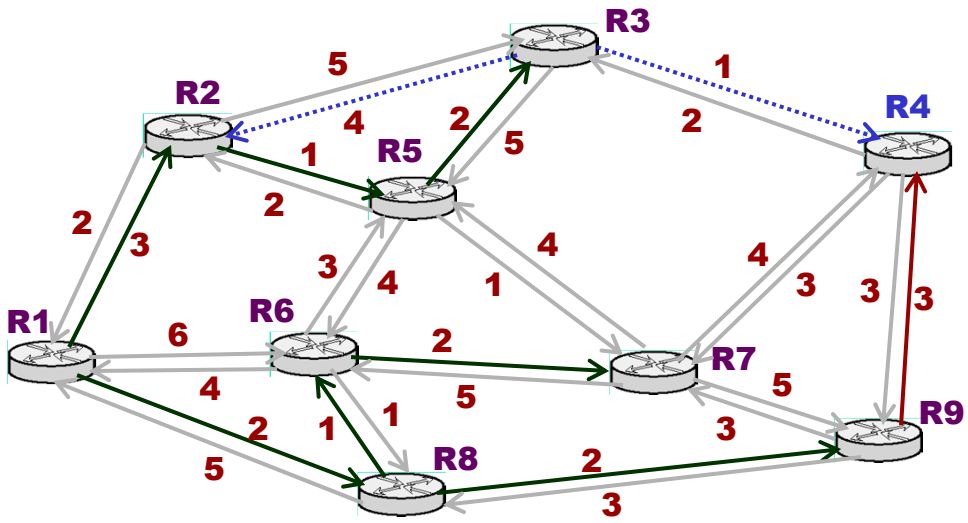
T={1,2,5,6,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(7) ดังนั้นนำ Node 7 ใส่ใน T; T={1,2,5,6,7,8,9}

L(n) = min[L(n), L(7) + w(7, n)] for all n not in T

T={1,2,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,5,6,7,8,9})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

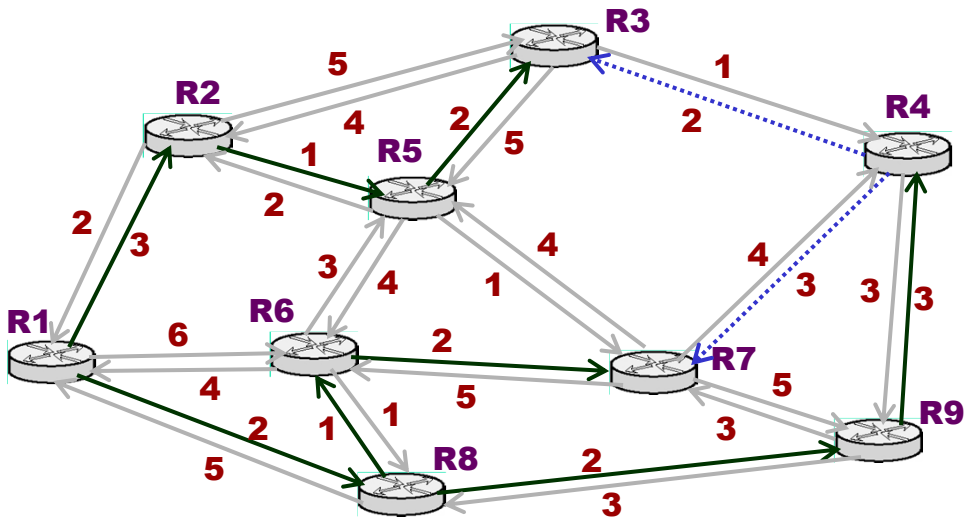
T={1,2,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(3) ดังนั้นนำ Node 3 ใส่ใน T; T={1,2,3,5,6,7,8,9}

L(n) = min[L(n), L(3) + w(3, n)] for all n not in T

T={1,2,3,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,3,5,6,7,8,9})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

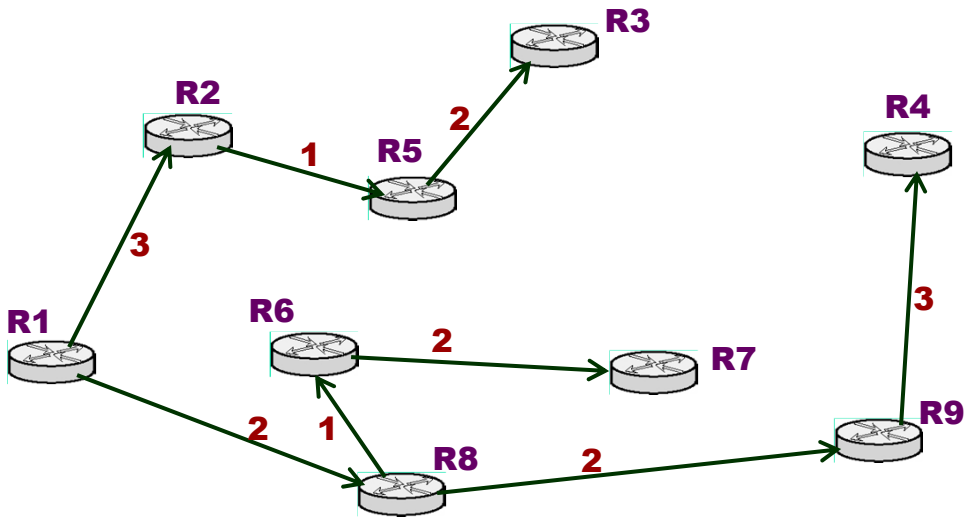
T={1,2,3,5,6,7,8,9}, L(2)=3, L(3)=6, **L(4)=7,** L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(4) ดังนั้นนำ Node 4 ใส่ใน T; T={1,2,3,4,5,6,7,8,9}

L(n) = min[L(n), L(4) + w(4, n)] for all n not in T

T={1,2,3,4,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,3,4,5,6,7,8,9})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

T={1,2,3,4,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Algorithm Terminates

Dijkstra จะคำนวณได้ต่อเมื่อรู้ Topology ของ Network

HW # 7

- **Download และส่งสัปดาห์หน้า**

End of Week 12

- **Week 13 IP Routing II: BGP/RIP/OSPF and Multicast Protocols**
  - Chapter 27:27.9-27.16 + Extra Notes
    - BGP
    - RIP
    - OSPF
    - Subnet and VLAN
    - Switch Layer 3

CPE 426 Computer Networks

Chapter 10:

Text Chapter 27: Internet Routing

Part II:BGP, RIP & OSPF

TOPICS

- **Chapter 27: Internet Routing and Routing Protocols**
  - **27.9 Border Gateway Protocol (BGP)**
  - **27.10 Routing Information Protocol (RIP)**
  - **27.11 RIP Packet Format**
  - **27.12 The Open Shortest Path First Protocol (OSPF)**
  - **27.13 OSPF Graph**
  - **27.14 OSPF Area**
  - **27.15 IS-IS**
  - **BREAK**
  - **27.16 Multicast Routing**
  - **Extra Subnet and VLAN**
  - **Extra Switch Layer 3 vs Router**
  - **Extra Organization Network**

27.9 The Border Gateway Protocol(BGP)

- เป็น Exterior Gateway Protocol ที่ใช้กันมากที่สุด
- ปัจจุบันที่ใช้คือ Version 4 หรือ BGP-4
- ISP จะใช้ BGP ในการแลกเปลี่ยน Routing Information ซึ่งกันและกันและกับส่วน Center ของ Internet
- ปกติจะไม่พบการใช้งานภายในองค์กร เพราะภายในองค์กรใช้ IGP จะดีกว่า

27.9 The Border Gateway Protocol (BGP)

- **คุณสมบัติของ BGP**
  - สามารถทำการ Routing ระหว่าง AS
    - BGP ได้ถูกออกแบบมาให้เป็น EGP ดังนั้นมันจะส่งข้อมูล Routing Information ในระดับ AS ต่อ AS แต่ละเส้นทางที่ส่ง จะบ่งบอกว่าจะผ่าน AS อะไรบ้าง เช่นเส้นทางไปถึงที่หมายต้องผ่าน AS 17, 2, 56 และ 12 และจะไม่มีการใช้ Routing Matrix และ BGP จะไม่สามารถส่งรายละเอียดเกี่ยวกับ Router ใน AS
  - สามารถกำหนดนโยบาย (Policy) ในการทำ Routing
    - ผู้ดูแลระบบ สามารถกำหนดว่าจะประกาศเส้นทางใดบ้าง ผ่าน BGP บอกยัง AS อื่นๆ

27.9 The Border Gateway Protocol (BGP)

- **คุณสมบัติของ BGP**
  - สามารถกำหนดการทำ Transit Routing
    - ถ้า AS ใดเป็นแค่ทางผ่านเพื่อจะไปยัง AS อื่น BGP จะจัด AS นั้นว่าเป็น Transit System
    - ถ้าข้อมูลส่งจาก AS หรือ ส่งไปที่ AS ใด BGP จะจัด AS นั้นเป็น Stub System
    - เราสามารถประกาศให้ AS ของเราเป็น Stub System ได้ แม้ว่าเราจะมีทางต่อออกนอก Internet ได้มากกว่าหนึ่งทาง (Multi-Homed)
  - BGP ใช้การส่งข้อมูลแบบเชื่อถือได้
    - BGP จะเป็น Protocol วางอยู่บน TCP หมายถึงว่าการส่ง Routing Information จะต้องมีการทำ Connection และข้อมูลที่ส่งสามารถเชื่อถือได้ว่าส่งไปอย่างถูกต้อง

27.10 RIP (Routing Information Protocol)

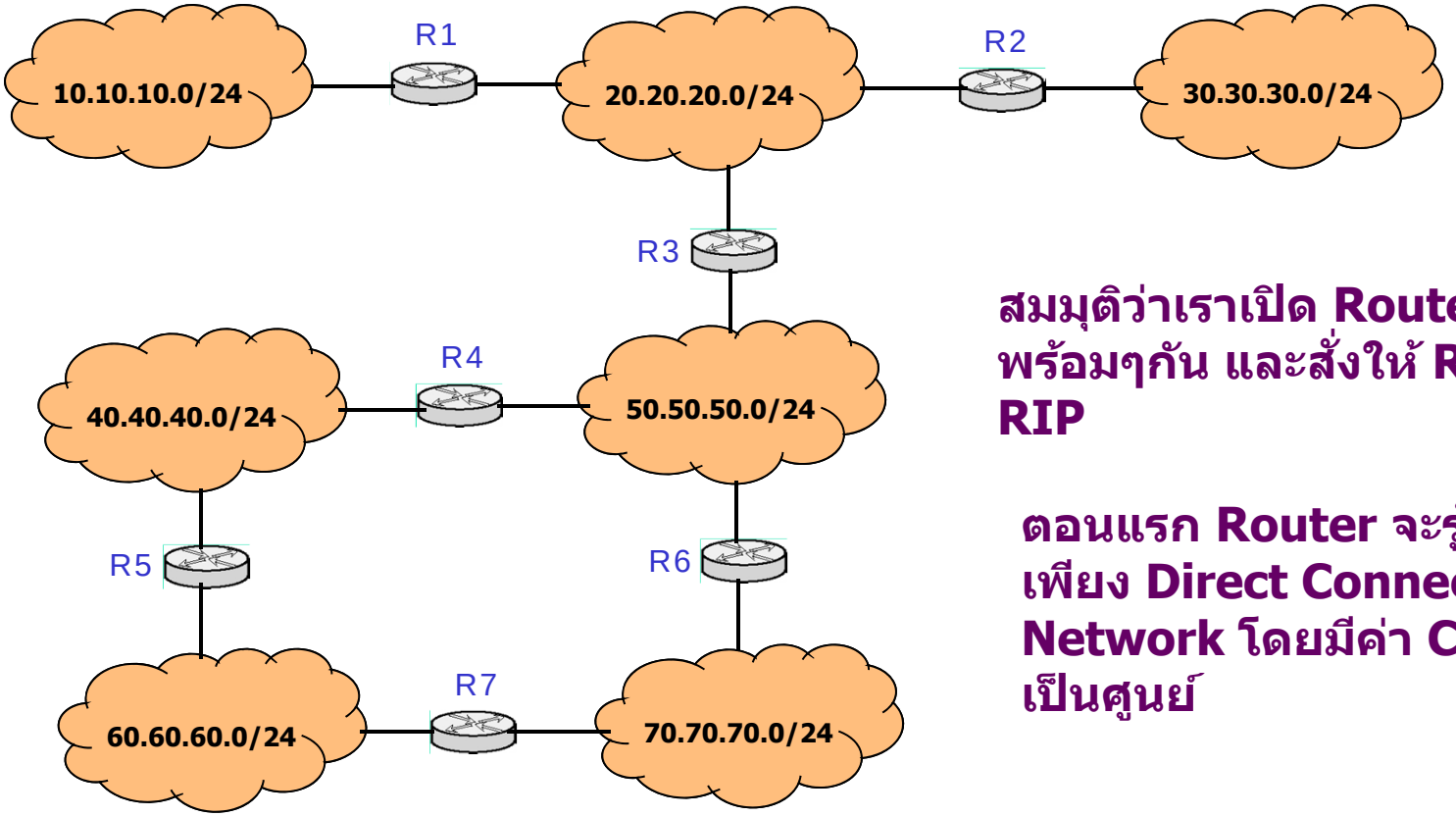
- **RIP เป็น IGP Protocol แรกๆที่ถูกนำมาใช้ใน Internet โดยมีคุณสมบัติดังนี้**
  - RIP จะทำงานภายใน Autonomous System เดียวกัน โดยถูกออกแบบมาให้เป็น IGP
  - RIP ใช้ Hop Count เป็น Routing Metric ค่า Cost คือ จำนวน Network ที่จะต้องส่งผ่าน
  - RIP ใช้ UDP ในการส่ง Routing Information ซึ่งเป็น Unreliable Transport
  - การส่ง Routing Information ของ RIP ใช้วิธี Broadcast (Version 1) หรือ Multicast (V.1/V.2) โดย RIP ได้ออกแบบมาให้ใช้กับ LAN (Ethernet) ซึ่งสนับสนุนการสื่อสารแบบ Broadcast และ Multicast
    - 224.0.0.9

27.10 RIP (Routing Information Protocol)

- **RIP เป็น IGP Protocol แรกๆที่ถูกนำมาใช้ใน Internet โดยมีคุณสมบัติดังนี้**
  - RIP สนับสนุนการทำ CIDR และ Subnetting (Version 2) โดย V.2 จะมีการส่ง Address Mask ด้วย
  - RIP สนับสนุนการส่งค่า Default Route ไปกับตาราง Routing Table ใน Routing Information ด้วย
    - เรากำหนด Default Route ให้แก่ Router ตัวเดียวก็พอ เช่นตัวที่ต่อกับ ISP
  - RIP ใช้ Distance Vector Algorithm โดย Router ที่เป็นเพื่อนบ้านกันจะแลกเปลี่ยนตาราง Routing Table และ Router แต่ละตัวจะทำการ Update ตารางของตนเองถ้าพบเส้นทางที่มีราคาต่ำกว่า
  - RIP สามารถกำหนด Passive Mode เพื่อใช้กับ Host ได้ โดย Host ที่ Run Passive RIP จะฟังอย่างเดียว และสามารถนำข้อมูลที่ฟังมา Update ตารางได้ (Router เท่านั้นที่สามารถส่งข้อมูล RIP ได้)
- **ข้อดีของ RIP คือใช้งานง่าย เพียงแค่ Start RIP ที่ตัว Router ก็สามารถทำงานได้**

27.10 RIP (Distance Vector) Example

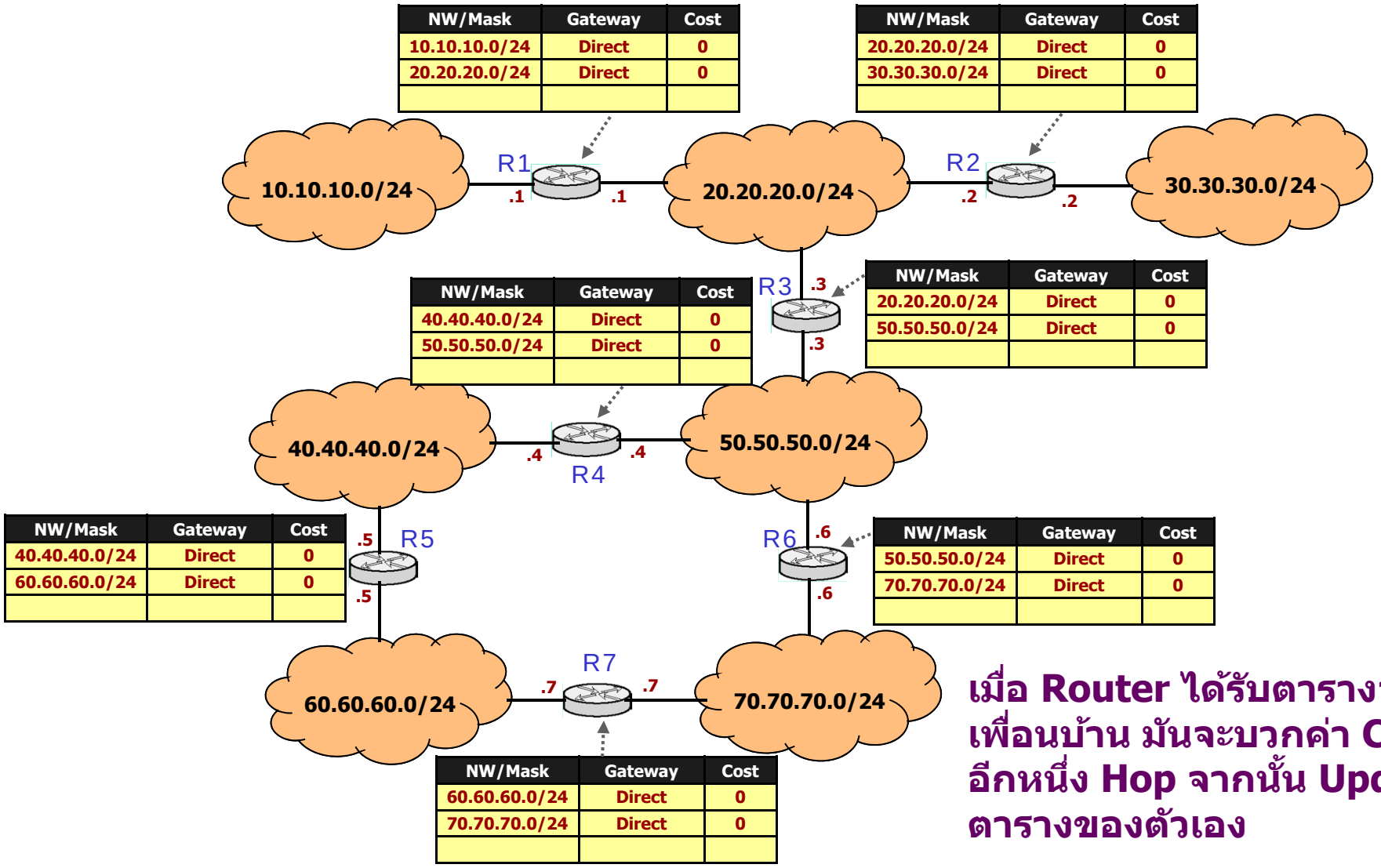
- ตัวอย่างการทำงานของ Distance Vector



สมมุติว่าเราเปิด Router พร้อมๆกัน และสั่งให้ Run RIP

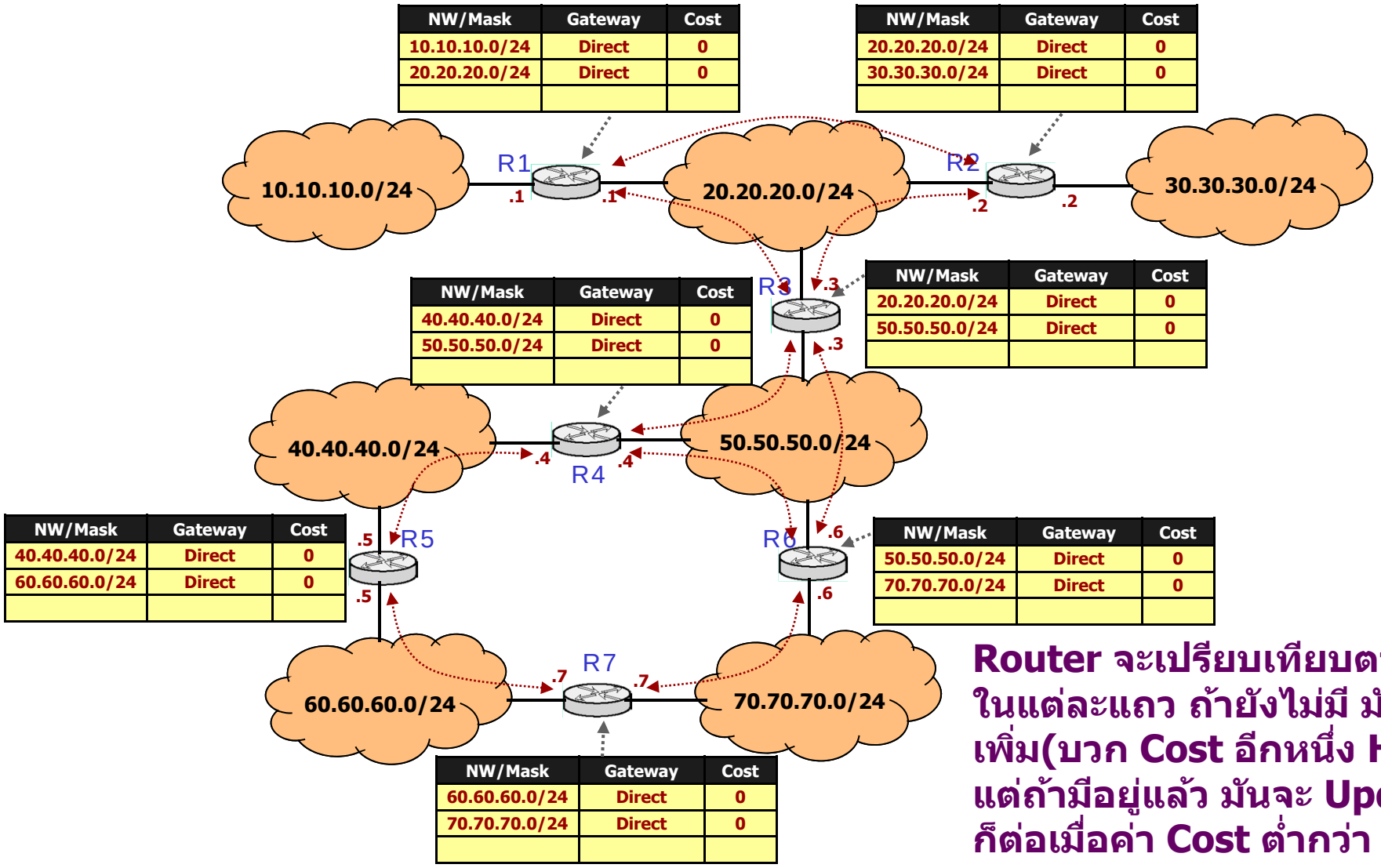
ตอนแรก Router จะรู้จักเพียง Direct Connect Network โดยมีค่า Cost เป็นศูนย์

27.10 RIP (Distance Vector) Example



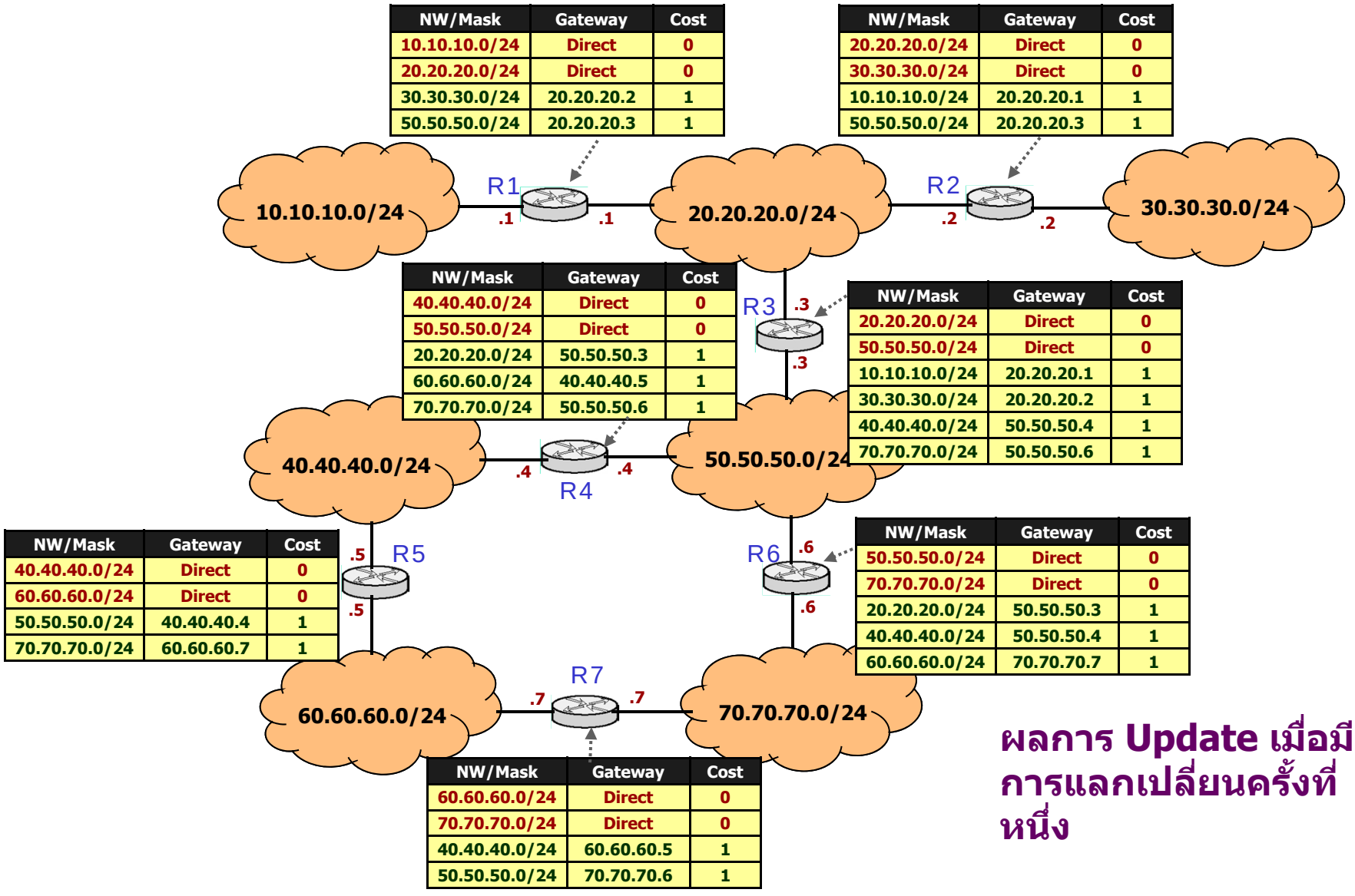
เมื่อ Router ได้รับตารางจากเพื่อนบ้าน มันจะบวกค่า Cost อีกหนึ่ง Hop จากนั้น Update ตารางของตัวเอง

27.10 RIP (Distance Vector) Example



Router จะเปรียบเทียบตารางในแต่ละแถว ถ้ายังไม่มี มันจะเพิ่ม(บวก Cost อีกหนึ่ง Hop) แต่ถ้ามีอยู่แล้ว มันจะ Update ก็ต่อเมื่อค่า Cost ต่ำกว่า

27.10 RIP (Distance Vector) Example



NW/Mask	Gateway	Cost
10.10.10.0/24	Direct	0
20.20.20.0/24	Direct	0
30.30.30.0/24	20.20.20.2	1
50.50.50.0/24	20.20.20.3	1

NW/Mask	Gateway	Cost
20.20.20.0/24	Direct	0
30.30.30.0/24	Direct	0
10.10.10.0/24	20.20.20.1	1
50.50.50.0/24	20.20.20.3	1

NW/Mask	Gateway	Cost
40.40.40.0/24	Direct	0
50.50.50.0/24	Direct	0
20.20.20.0/24	50.50.50.3	1
60.60.60.0/24	40.40.40.5	1
70.70.70.0/24	50.50.50.6	1

NW/Mask	Gateway	Cost
20.20.20.0/24	Direct	0
50.50.50.0/24	Direct	0
10.10.10.0/24	20.20.20.1	1
30.30.30.0/24	20.20.20.2	1
40.40.40.0/24	50.50.50.4	1
70.70.70.0/24	50.50.50.6	1

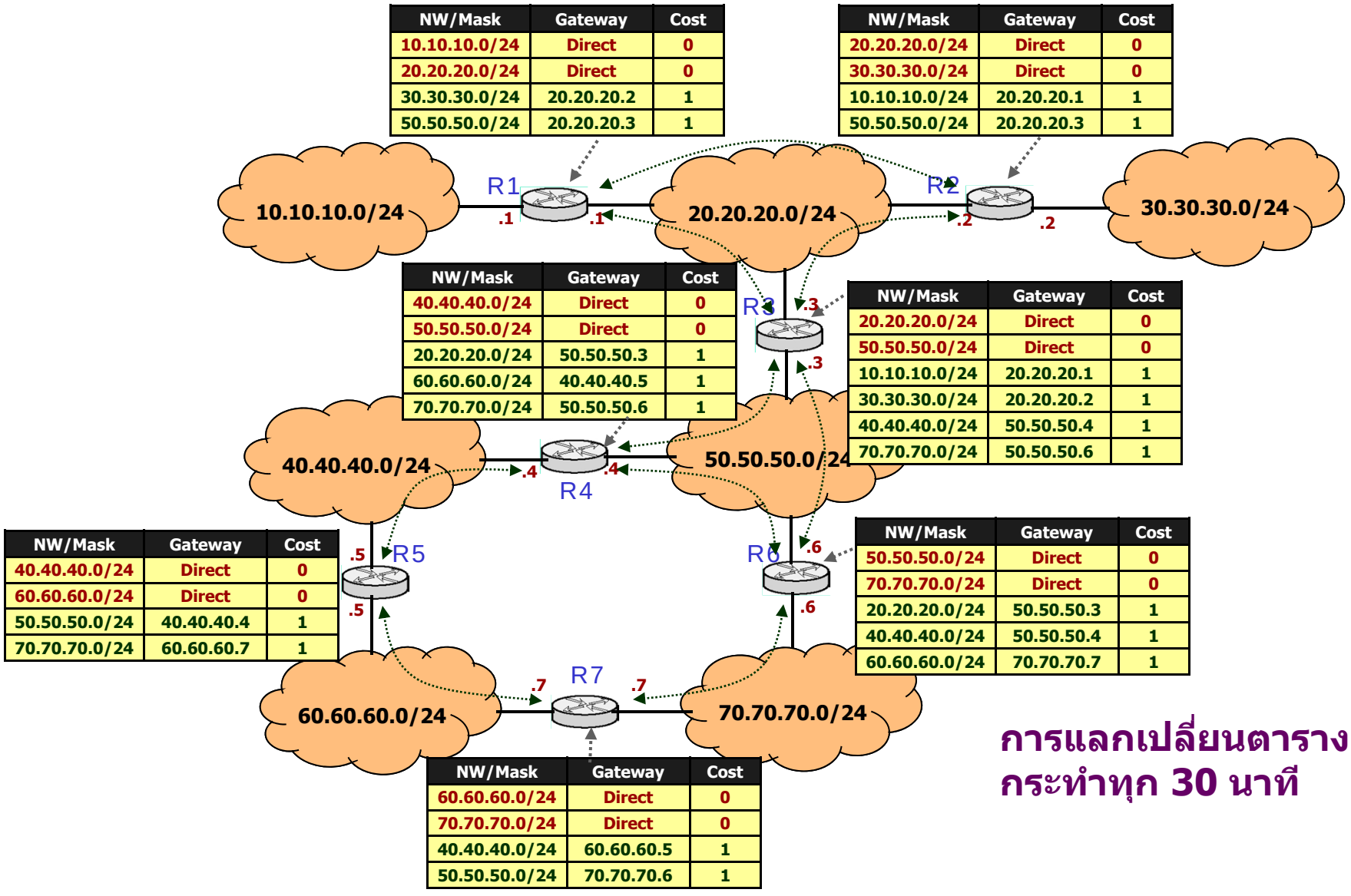
NW/Mask	Gateway	Cost
40.40.40.0/24	Direct	0
60.60.60.0/24	Direct	0
50.50.50.0/24	40.40.40.4	1
70.70.70.0/24	60.60.60.7	1

NW/Mask	Gateway	Cost
50.50.50.0/24	Direct	0
70.70.70.0/24	Direct	0
20.20.20.0/24	50.50.50.3	1
40.40.40.0/24	50.50.50.4	1
60.60.60.0/24	70.70.70.7	1

NW/Mask	Gateway	Cost
60.60.60.0/24	Direct	0
70.70.70.0/24	Direct	0
40.40.40.0/24	60.60.60.5	1
50.50.50.0/24	70.70.70.6	1

ผลการ Update เมื่อมีการแลกเปลี่ยนครั้งที่หนึ่ง

27.10 RIP (Distance Vector) Example



NW/Mask	Gateway	Cost
10.10.10.0/24	Direct	0
20.20.20.0/24	Direct	0
30.30.30.0/24	20.20.20.2	1
50.50.50.0/24	20.20.20.3	1

NW/Mask	Gateway	Cost
20.20.20.0/24	Direct	0
30.30.30.0/24	Direct	0
10.10.10.0/24	20.20.20.1	1
50.50.50.0/24	20.20.20.3	1

NW/Mask	Gateway	Cost
40.40.40.0/24	Direct	0
50.50.50.0/24	Direct	0
20.20.20.0/24	50.50.50.3	1
60.60.60.0/24	40.40.40.5	1
70.70.70.0/24	50.50.50.6	1

NW/Mask	Gateway	Cost
20.20.20.0/24	Direct	0
50.50.50.0/24	Direct	0
10.10.10.0/24	20.20.20.1	1
30.30.30.0/24	20.20.20.2	1
40.40.40.0/24	50.50.50.4	1
70.70.70.0/24	50.50.50.6	1

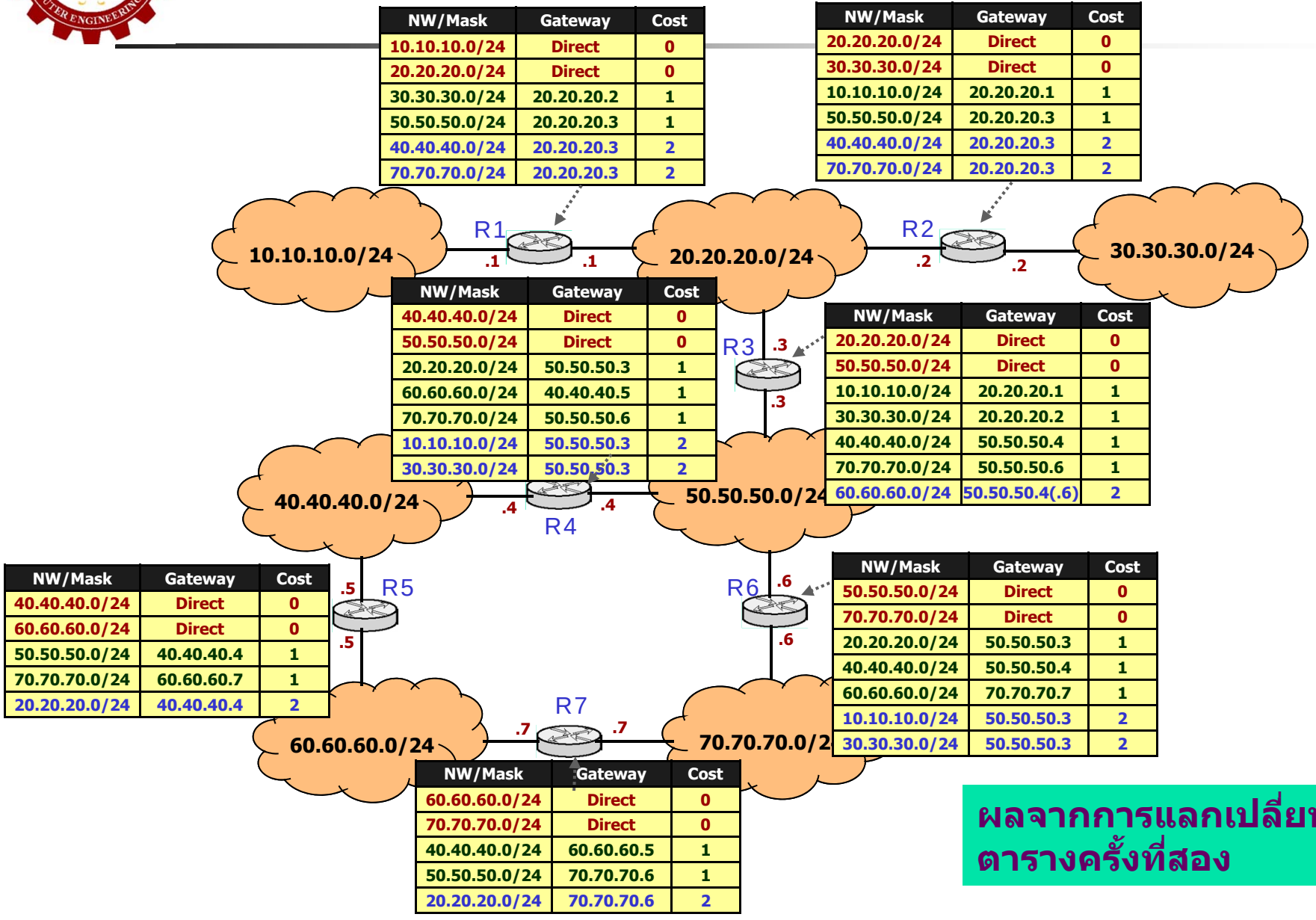
NW/Mask	Gateway	Cost
40.40.40.0/24	Direct	0
60.60.60.0/24	Direct	0
50.50.50.0/24	40.40.40.4	1
70.70.70.0/24	60.60.60.7	1

NW/Mask	Gateway	Cost
50.50.50.0/24	Direct	0
70.70.70.0/24	Direct	0
20.20.20.0/24	50.50.50.3	1
40.40.40.0/24	50.50.50.4	1
60.60.60.0/24	70.70.70.7	1

NW/Mask	Gateway	Cost
60.60.60.0/24	Direct	0
70.70.70.0/24	Direct	0
40.40.40.0/24	60.60.60.5	1
50.50.50.0/24	70.70.70.6	1

การแลกเปลี่ยนตารางกระทำทุก 30 นาที

27.10 RIP (Distance Vector) Example



NW/Mask	Gateway	Cost
10.10.10.0/24	Direct	0
20.20.20.0/24	Direct	0
30.30.30.0/24	20.20.20.2	1
50.50.50.0/24	20.20.20.3	1
40.40.40.0/24	20.20.20.3	2
70.70.70.0/24	20.20.20.3	2

NW/Mask	Gateway	Cost
20.20.20.0/24	Direct	0
30.30.30.0/24	Direct	0
10.10.10.0/24	20.20.20.1	1
50.50.50.0/24	20.20.20.3	1
40.40.40.0/24	20.20.20.3	2
70.70.70.0/24	20.20.20.3	2

NW/Mask	Gateway	Cost
40.40.40.0/24	Direct	0
50.50.50.0/24	Direct	0
20.20.20.0/24	50.50.50.3	1
60.60.60.0/24	40.40.40.5	1
70.70.70.0/24	50.50.50.6	1
10.10.10.0/24	50.50.50.3	2
30.30.30.0/24	50.50.50.3	2

NW/Mask	Gateway	Cost
20.20.20.0/24	Direct	0
50.50.50.0/24	Direct	0
10.10.10.0/24	20.20.20.1	1
30.30.30.0/24	20.20.20.2	1
40.40.40.0/24	50.50.50.4	1
70.70.70.0/24	50.50.50.6	1
60.60.60.0/24	50.50.50.4(.6)	2

NW/Mask	Gateway	Cost
40.40.40.0/24	Direct	0
60.60.60.0/24	Direct	0
50.50.50.0/24	40.40.40.4	1
70.70.70.0/24	60.60.60.7	1
20.20.20.0/24	40.40.40.4	2

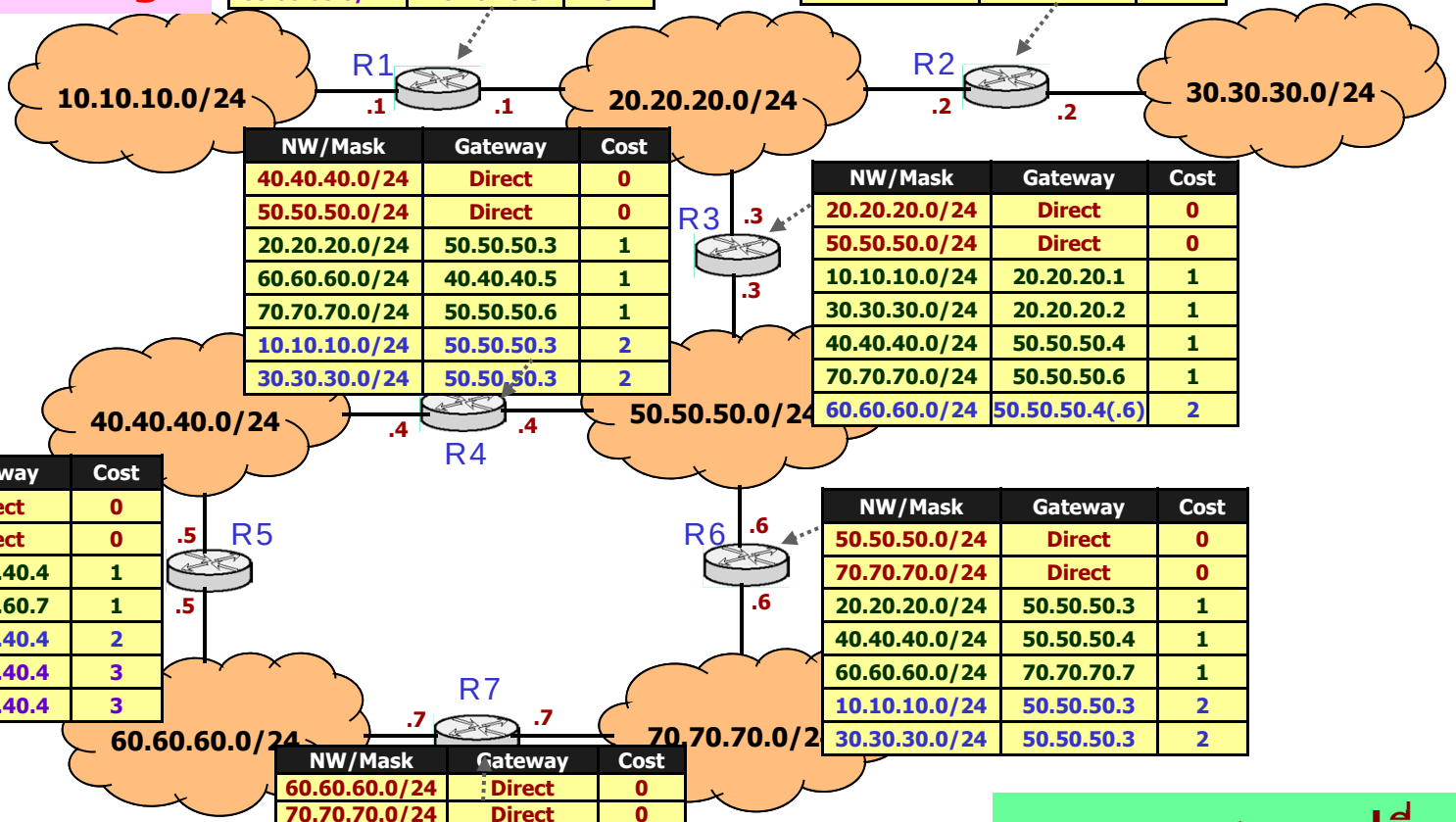
NW/Mask	Gateway	Cost
50.50.50.0/24	Direct	0
70.70.70.0/24	Direct	0
20.20.20.0/24	50.50.50.3	1
40.40.40.0/24	50.50.50.4	1
60.60.60.0/24	70.70.70.7	1
10.10.10.0/24	50.50.50.3	2
30.30.30.0/24	50.50.50.3	2

NW/Mask	Gateway	Cost
60.60.60.0/24	Direct	0
70.70.70.0/24	Direct	0
40.40.40.0/24	60.60.60.5	1
50.50.50.0/24	70.70.70.6	1
20.20.20.0/24	70.70.70.6	2

ผลจากการแลกเปลี่ยนตารางครั้งที่สอง

27.10 RIP (Distance Vector) Example

สังเกตุว่า การหาเส้นทางทีละ Hop คือ Bellman-Ford Alg.



NW/Mask	Gateway	Cost
10.10.10.0/24	Direct	0
20.20.20.0/24	Direct	0
30.30.30.0/24	20.20.20.2	1
50.50.50.0/24	20.20.20.3	1
40.40.40.0/24	20.20.20.3	2
70.70.70.0/24	20.20.20.3	2
60.60.60.0/24	20.20.20.3	3

NW/Mask	Gateway	Cost
20.20.20.0/24	Direct	0
30.30.30.0/24	Direct	0
10.10.10.0/24	20.20.20.1	1
50.50.50.0/24	20.20.20.3	1
40.40.40.0/24	20.20.20.3	2
70.70.70.0/24	20.20.20.3	2
60.60.60.0/24	20.20.20.3	3

NW/Mask	Gateway	Cost
40.40.40.0/24	Direct	0
50.50.50.0/24	Direct	0
20.20.20.0/24	50.50.50.3	1
60.60.60.0/24	40.40.40.5	1
70.70.70.0/24	50.50.50.6	1
10.10.10.0/24	50.50.50.3	2
30.30.30.0/24	50.50.50.3	2

NW/Mask	Gateway	Cost
20.20.20.0/24	Direct	0
50.50.50.0/24	Direct	0
10.10.10.0/24	20.20.20.1	1
30.30.30.0/24	20.20.20.2	1
40.40.40.0/24	50.50.50.4	1
70.70.70.0/24	50.50.50.6	1
60.60.60.0/24	50.50.50.4(.6)	2

NW/Mask	Gateway	Cost
40.40.40.0/24	Direct	0
60.60.60.0/24	Direct	0
50.50.50.0/24	40.40.40.4	1
70.70.70.0/24	60.60.60.7	1
20.20.20.0/24	40.40.40.4	2
10.10.10.0/24	40.40.40.4	3
30.30.30.0/24	40.40.40.4	3

NW/Mask	Gateway	Cost
50.50.50.0/24	Direct	0
70.70.70.0/24	Direct	0
20.20.20.0/24	50.50.50.3	1
40.40.40.0/24	50.50.50.4	1
60.60.60.0/24	70.70.70.7	1
10.10.10.0/24	50.50.50.3	2
30.30.30.0/24	50.50.50.3	2

NW/Mask	Gateway	Cost
60.60.60.0/24	Direct	0
70.70.70.0/24	Direct	0
40.40.40.0/24	60.60.60.5	1
50.50.50.0/24	70.70.70.6	1
20.20.20.0/24	70.70.70.6	2
10.10.10.0/24	70.70.70.6	3
30.30.30.0/24	70.70.70.6	3

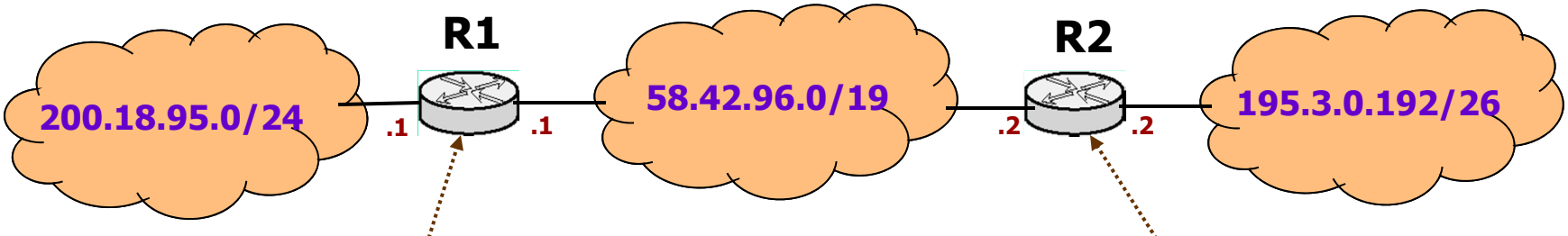
ผลจากการแลกเปลี่ยนตารางครั้งที่สาม

ปัญหาของ RIP (และ Distance Vector)

- เนื่องจาก Router แลกเปลี่ยนข้อมูลกับเพื่อนบ้านเท่านั้น เมื่อเกิดปัญหาที่จุดใด ตัว Router ที่อยู่ไกลจะรู้และปรับตารางช้ากว่า Router ที่อยู่ใกล้ ทำให้การ Converge ช้า ทำให้เหมาะสมสำหรับ Network ขนาดเล็กเท่านั้น
- ขนาดของ Message ที่ Router แต่ละตัวส่งจะสัมพันธ์กับจำนวนของ Network ถ้า Network มีจำนวนมาก ข้อมูลที่ส่งจะมีมาก
- และด้วยการที่ Update ไม่พร้อมกัน จะทำให้เกิด Route Loop ได้
  - สำคัญ ต้องแก้ปัญหานี้

การเกิด Loop ใน Distance Vector

- สมมุติ มีสาม Network เชื่อมต่อผ่านสอง Router และทำการ Update ตารางเรียบร้อย

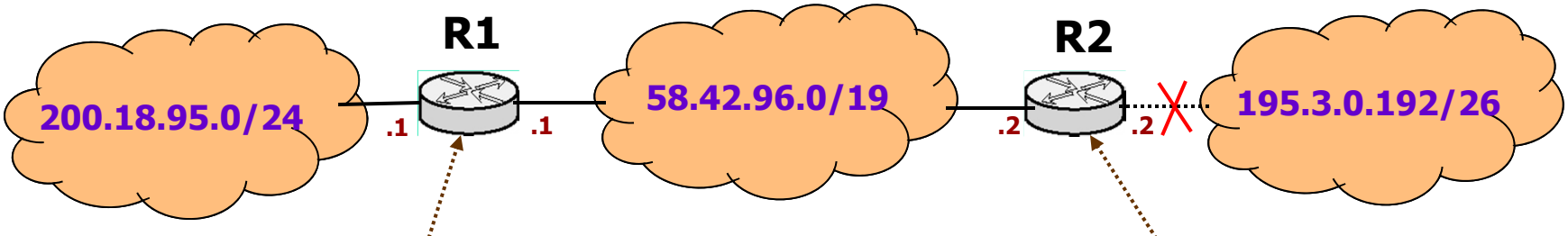


Network	Net Mask	Gateway	Cost
200.18.95.0	255.255.255.0	Direct	0
58.42.96.0	255.255.224.0	Direct	0
195.3.0.192	255.255.255.192	58.42.96.2	1

Network	Net Mask	Gateway	Cost
195.3.0.192	255.255.255.192	Direct	0
58.42.96.0	255.255.224.0	Direct	0
200.18.95.0	255.255.255.0	58.42.96.1	1

การเกิด Loop ใน Distance Vector

- สมมุติ ต่อว่า Network 195.3.0.192/26 เกิด Down เช่น Link ขาด ดังนั้น R2 จะตรวจจับได้ และ Mark ตารางของตนเป็น Unreachable (Infinity)

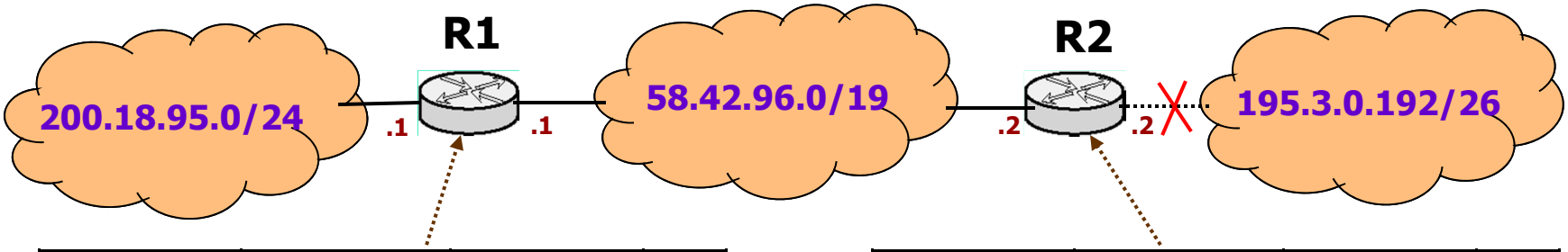


Network	Net Mask	Gateway	Cost
200.18.95.0	255.255.255.0	Direct	0
58.42.96.0	255.255.224.0	Direct	0
195.3.0.192	255.255.255.192	58.42.96.2	1

Network	Net Mask	Gateway	Cost
195.3.0.192	255.255.255.192	-	inf
58.42.96.0	255.255.224.0	Direct	0
200.18.95.0	255.255.255.0	58.42.96.1	1

การเกิด Loop ใน Distance Vector

- เมื่อถึงเวลา Update และมีการแลกเปลี่ยนตาราง R1 จะเรียนรู้แล้วว่า 195.3.0.192/26 นั้นเป็น Unreachable และปรับตารางตนเอง



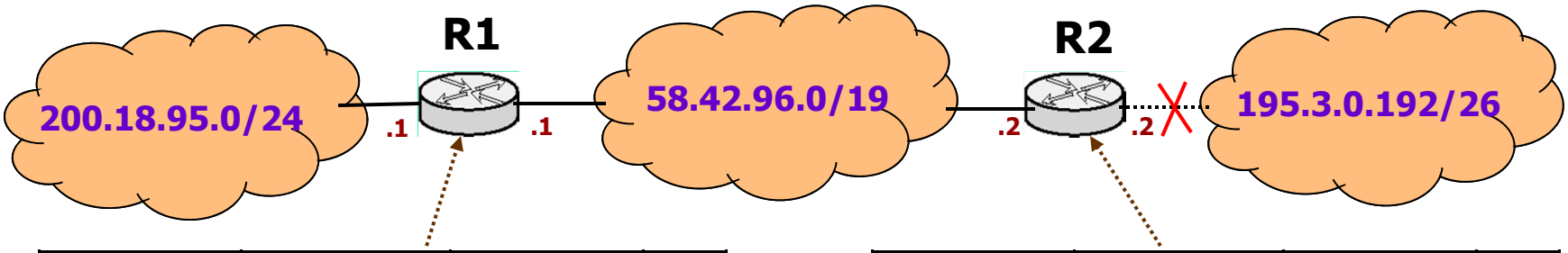
Network	Net Mask	Gateway	Cost
200.18.95.0	255.255.255.0	Direct	0
58.42.96.0	255.255.224.0	Direct	0
195.3.0.192	255.255.255.192	58.42.96.2	1

Network	Net Mask	Gateway	Cost
195.3.0.192	255.255.255.192	-	inf
58.42.96.0	255.255.224.0	Direct	0
200.18.95.0	255.255.255.0	58.42.96.1	1

- ในขณะเดียวกัน R2 ได้รับตารางจาก R1 และค้นพบว่า R1 มีทางไป 195.3.0.192/26 ด้วย Cost เท่ากับหนึ่ง มันจึง Update ตารางด้วย Cost = 2 โดยหารู้ไม่ว่าข้อมูลนั้น R1 ได้เรียนรู้จากตนเอง

การเกิด Loop ใน Distance Vector

- ผลที่ได้ จะทำให้ตารางผิดพลาด



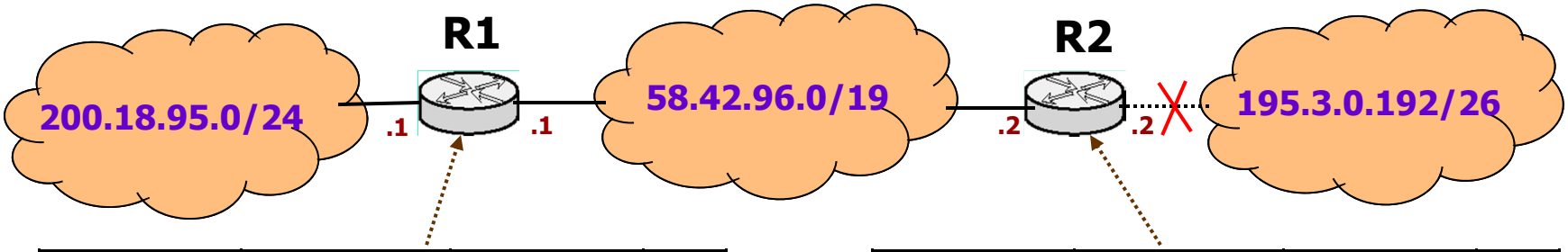
Network	Net Mask	Gateway	Cost
200.18.95.0	255.255.255.0	Direct	0
58.42.96.0	255.255.224.0	Direct	0
195.3.0.192	255.255.255.192	-	inf

Network	Net Mask	Gateway	Cost
195.3.0.192	255.255.255.192	58.42.96.1	2
58.42.96.0	255.255.224.0	Direct	0
200.18.95.0	255.255.255.0	58.42.96.1	1

- ถ้ามี Packet เข้ามาที่ R2 และต้องการไป 195.3.0.195 /26 มันจะถูกส่งไป R1 และเป็นไปได้ที่ ตาราง R1 จะยังไม่ Update และถูกส่งกลับไปกลับมาเป็น Loop

การเกิด Loop ใน Distance Vector

- ในการ Update ครั้งต่อไป R1 จะเรียนรู้ 195.3.0.192 /26 จาก R2 อีกครั้งด้วย Cost 2 และ Update ตารางตนเอง ขณะเดียวกัน R2 ทำการ Update ตารางตนเองเช่นกัน

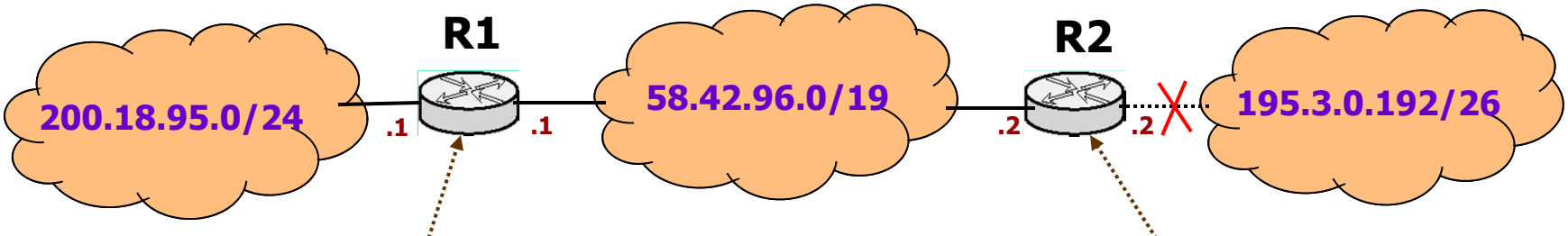


Network	Net Mask	Gateway	Cost
200.18.95.0	255.255.255.0	Direct	0
58.42.96.0	255.255.224.0	Direct	0
195.3.0.192	255.255.255.192	-	inf

Network	Net Mask	Gateway	Cost
195.3.0.192	255.255.255.192	58.42.96.1	2
58.42.96.0	255.255.224.0	Direct	0
200.18.95.0	255.255.255.0	58.42.96.1	1

การเกิด Loop ใน Distance Vector

- การ Update จะทำสลับไปมา และค่า Cost จะเพิ่มทีละหนึ่งจนเข้าสู่ Infinity เราเรียกเหตุการณ์นี้ว่า ‘Count to Infinity’



Network	Net Mask	Gateway	Cost
200.18.95.0	255.255.255.0	Direct	0
58.42.96.0	255.255.224.0	Direct	0
195.3.0.192	255.255.255.192	58.42.96.2	3

Network	Net Mask	Gateway	Cost
195.3.0.192	255.255.255.192	-	inf
58.42.96.0	255.255.224.0	Direct	0
200.18.95.0	255.255.255.0	58.42.96.1	1

- Packet จะถูกส่งวิ่งวนไปมาระหว่าง R1 และ R2

การแก้ปัญหา Loop ใน Distance Vector

- **Split Horizon:** กำหนดว่าการส่ง Routing Update ไปให้กับเพื่อนบ้านใดๆ ให้ตัดส่วน Subnet ที่ได้เรียนรู้มาจากเพื่อนบ้านนั้นออกไป
- **Maximum Hop Count:** มาตรฐานของ RIP จะมีค่าสูงสุดคือ 15 เลข 16 หมายถึงว่าเป็น Unreachable การกำหนดเช่นนี้ทำให้การ Count To Infinity หยุดเร็วขึ้น แต่จะทำให้ขนาดของ Network จำกัดอยู่ที่ 15 Hop ด้วยเช่นกัน
- **Triggerded Update:** กำหนดว่าถ้า Router ใดพบว่ามี Subnet Down ก็ให้ส่งสัญญาณบอก Router เพื่อนบ้านทุกตัวทันที โดยไม่ต้องรอให้ถึงเวลาของการ Update ปกติ(RIP จะมีเวลาการ Update ปกติคือทุกๆ 30 วินาที)
- **Route Poisoning:** จะใช้ร่วมกับ Triggered Update คือเมื่อ Router ใดพบว่ามี Subnet Down มันจะประกาศออกไปยัง Interface ทุกอันของมันว่า Subnet นี้ Unreachable โดยตั้งค่า Hop Count เท่ากับ Infinity(16 สำหรับ RIP)
- **Hold-Down Timer:** เมื่อ Router ได้รับ Triggered Update มันจะไม่สนใจเกี่ยวกับเส้นทางที่จะไปยัง Subnet ที่ Down ลงเป็นระยะเวลาหนึ่งเท่ากับ Hold-Down Timer(180 วินาที สำหรับ RIP) เพื่อป้องกันการเรียนรู้ที่ผิดพลาด จากนั้นมันถึงจะเริ่มเรียนรู้ Routing ใหม่ๆที่จะยัง Subnet นั้น
- ทั้งหมดนี้ ยังไม่ Guarantee ว่าจะกำจัด Loop ได้ โดยเฉพาะ Loop ที่เกิดระหว่างหลาย Router
  - **CISCO ได้คิด EIGRP ซึ่งมีเทคนิคในการแก้ปัญหาที่ดีกว่า แต่เป็น Protocol เฉพาะ**

27.11 RIP Packet Format(v2)

- **Command บ่งบอกว่าเป็น RIP Request หรือ Response**
- **Routing Information แต่ละชุดจะประกอบด้วย 5 Word (20 Bytes)**
  - Family(Address Family Identifier) โดย RIP สามารถจะส่ง Routing Information ได้กับหลาย Protocol ถ้าเป็น IP จะมีค่า 2, ถ้าเป็นการทำ Authentication จะใช้ค่า 0xffff
  - Route Tag กำหนดวิธีบ่งบอกความแตกต่างระหว่าง Internal Route (เรียนรู้จาก RIP) และ External Route (เรียนรู้จาก Protocol อื่น)
  - ที่เหลือคือ IP Address (Network ID), Subnet Mask และ Metric (จะมีค่าระหว่าง 0 และ 15), 16 หมายถึง Unreachable

27.11 RIP Packet Format(v2)

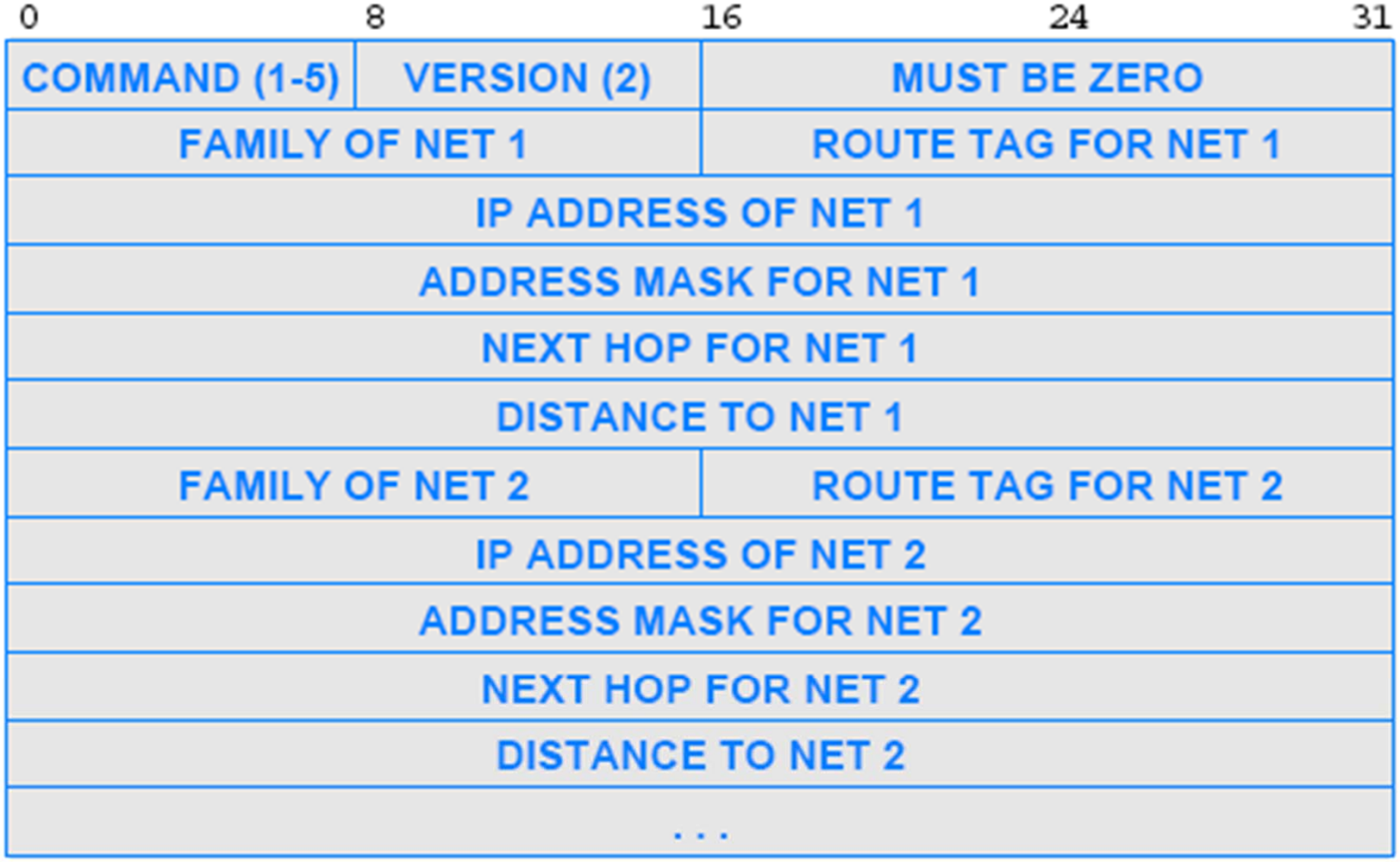


Figure 27.5 The format of a RIP version 2 update message.

27.12 The Open Shortest Path First Protocol (OSPF)

- **RIP ไม่เหมาะกับ Network ขององค์กรขนาดใหญ่ ซึ่ง IETF ได้ออกแบบ IGP ขึ้นมาอีกตัวหนึ่งชื่อ OPEN SHORTEST PATH FIRST PROTOCOL (OSPF) โดยใช้ Dijkstra Algorithm ที่ชื่อ SPF (Shortest Path First) ในการคำนวณเส้นทาง ซึ่ง OSPF จัดว่าเป็น Link-State Protocol**
  - การส่ง Routing Information จะส่งเฉพาะ Link-State ของตนเองซึ่งจะมีขนาดเท่าเดิมแม้ว่า Network จะมีการขยายตัว
  - ผิดกับ Distance Vector ที่ส่งทั้งตาราง (Distance Vector) ที่จะมีขนาดเพิ่มขึ้น เมื่อ Network มีการขยายตัว

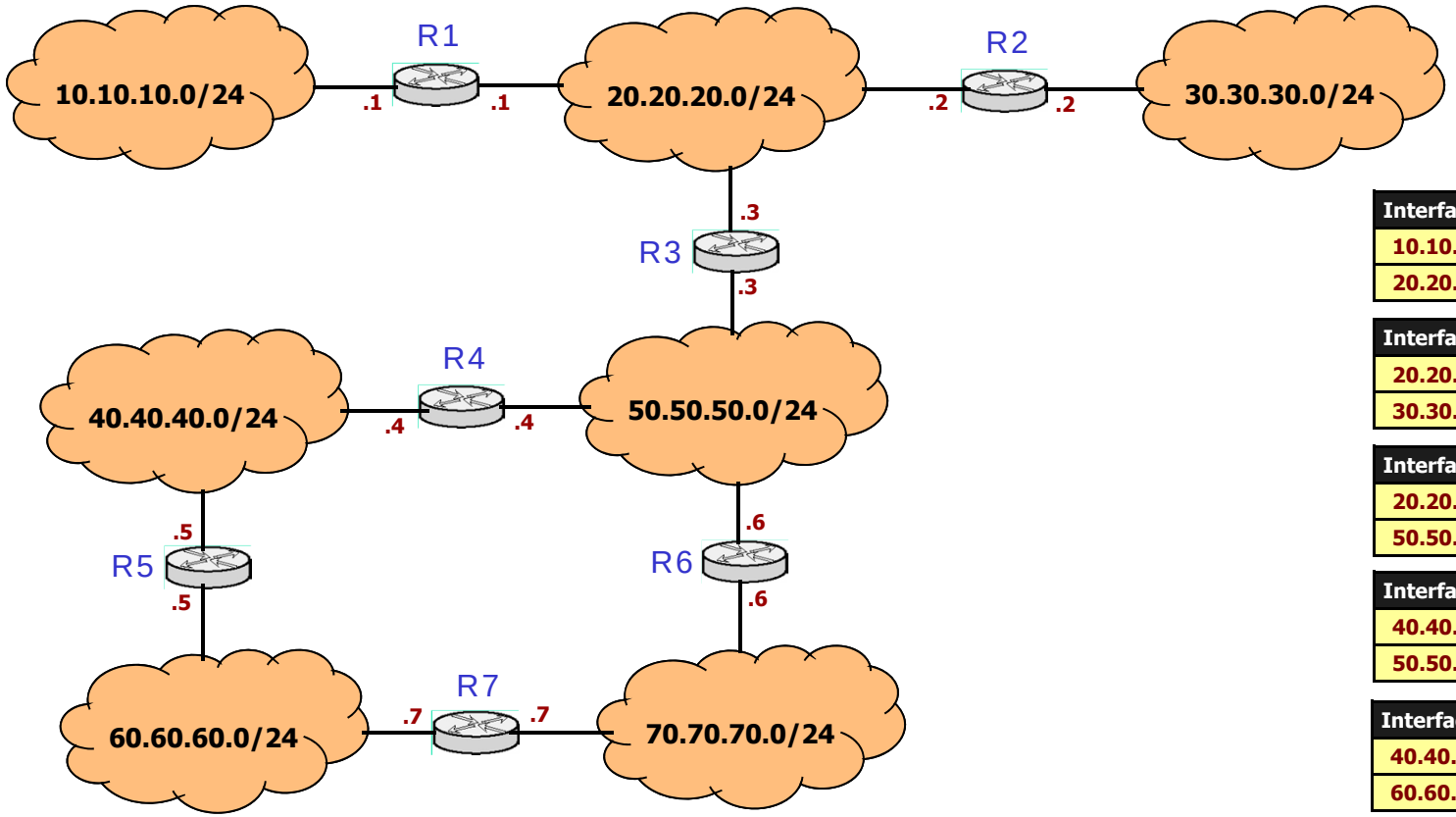
27.12 The Open Shortest Path First Protocol (OSPF)

- **OSPF มีคุณสมบัติดังนี้**
  - ออกแบบมาให้ทำงานภายใน AS คือเป็น IGP
  - สนับสนุนการทำ CIDR โดยมีการส่ง Address Mask ไปพร้อมกับ IP Address
  - สามารถทำ Authenticate ซึ่งกันและกันระหว่างสอง Router ได้
  - สามารถ Import Route ที่เรียนรู้โดยวิธีอื่นเข้ามาผนวกกับตารางของ OSPF ได้
  - ใช้วิธีการของ Link-State Routing
  - สนับสนุนการใช้ Route Metric หลายอย่าง และยอมใช้ผู้ดูแลระบบกำหนดค่า Cost ของแต่ละ Route ได้
  - สนับสนุนการทำงานของ Network ที่เป็น Multi-Access โดยมีวิธีการไม่ให้ Router ทุกตัวทำการ Broadcast Link-State ออกมา แต่จะกำหนดให้ Router เพียงหนึ่งตัวทำการ Broadcast
    - Router นี้ชื่อ Designated Router (DR)

27.13 ตัวอย่างการสร้าง OSPF Graph

- **Router แต่ละตัวจะส่ง Link-State Advertisement (LSA)**
- **Router แต่ละตัวจะรวบรวม LSA ของ Router ทุกตัวใน Network สร้างเป็น Link-State Database (LSB) ที่เหมือนกัน**
- **จาก LSB ตัว Router จะทำการคำนวน และสร้าง OSPF Graph**
- **จาก OSPF Graph ตัว Router จะคำนวณเส้นทางจากตัวมัน ไปยัง Router ตัวอื่นๆ โดยสร้าง SPF Tree ที่ตัวมันเป็น Root ด้วย Dijkstra Algorithm**
  - เนื่องจากมันสร้าง Tree ดังนั้นจะไม่เกิด Loop
- **ถ้า Topology เปลี่ยน ตัว Router ที่ Detect ได้จะส่ง Update Link State ใหม่**
  - Router ทุกตัวจะ Update LSB และคำนวณ SPF Tree ใหม่
  - Fast Convergence เพราะทุกตัว Update พร้อมกัน

OSPF Example: Link-State



LSB

R1 LSA

Interface /Mask	Cost
10.10.10.1/24	2
20.20.20.1/24	3

R2 LSA

Interface /Mask	Cost
20.20.20.2/24	1
30.30.30.2/24	2

R3 LSA

Interface /Mask	Cost
20.20.20.3/24	4
50.50.50.3/24	3

R4 LSA

Interface /Mask	Cost
40.40.40.4/24	5
50.50.50.4/24	2

R5 LSA

Interface /Mask	Cost
40.40.40.5/24	1
60.60.60.5/24	3

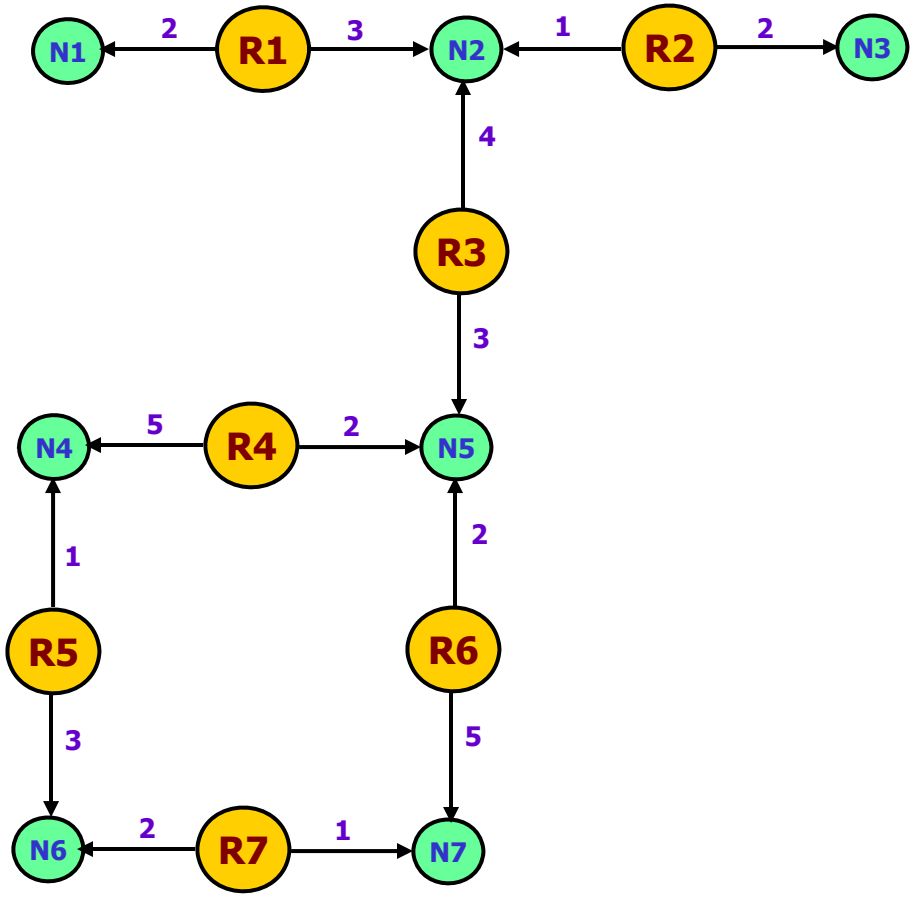
R6 LSA

Interface /Mask	Cost
50.50.50.6/24	2
70.70.70.6/24	5

R7 LSA

Interface /Mask	Cost
60.60.60.7/24	2
70.70.70.7/24	1

OSPF Example: Link-State



LSB

R1 LSA

	Interface /Mask	Cost
N1	10.10.10.1/24	2
N2	20.20.20.1/24	3

R2 LSA

	Interface /Mask	Cost
N2	20.20.20.2/24	1
N3	30.30.30.2/24	2

R3 LSA

	Interface /Mask	Cost
N2	20.20.20.3/24	4
N5	50.50.50.3/24	3

R4 LSA

	Interface /Mask	Cost
N4	40.40.40.4/24	5
N5	50.50.50.4/24	2

R5 LSA

	Interface /Mask	Cost
N4	40.40.40.5/24	1
N6	60.60.60.5/24	3

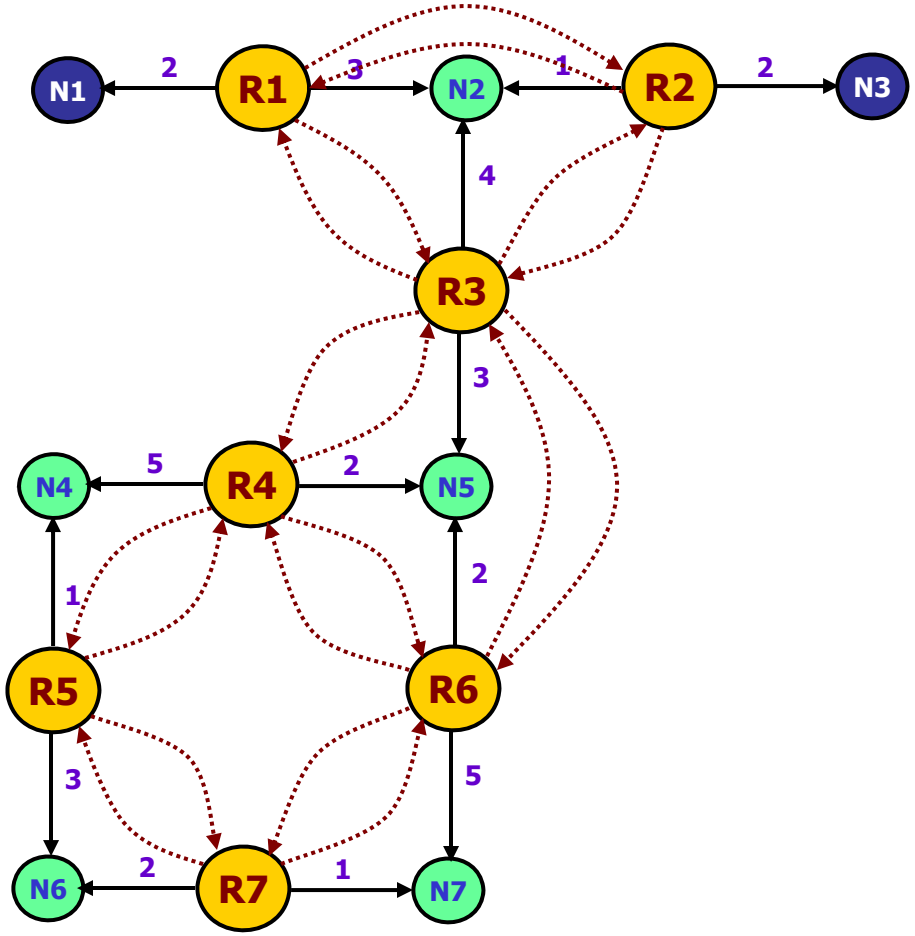
R6 LSA

	Interface /Mask	Cost
N5	50.50.50.6/24	2
N7	70.70.70.6/24	5

R7 LSA

	Interface /Mask	Cost
N6	60.60.60.7/24	2
N7	70.70.70.7/24	1

OSPF Example: Link-State



LSB

R1 LSA

	Interface /Mask	Cost
N1	10.10.10.1/24	2
N2	20.20.20.1/24	3

R2 LSA

	Interface /Mask	Cost
N2	20.20.20.2/24	1
N3	30.30.30.2/24	2

R3 LSA

	Interface /Mask	Cost
N2	20.20.20.3/24	4
N5	50.50.50.3/24	3

R4 LSA

	Interface /Mask	Cost
N4	40.40.40.4/24	5
N5	50.50.50.4/24	2

R5 LSA

	Interface /Mask	Cost
N4	40.40.40.5/24	1
N6	60.60.60.5/24	3

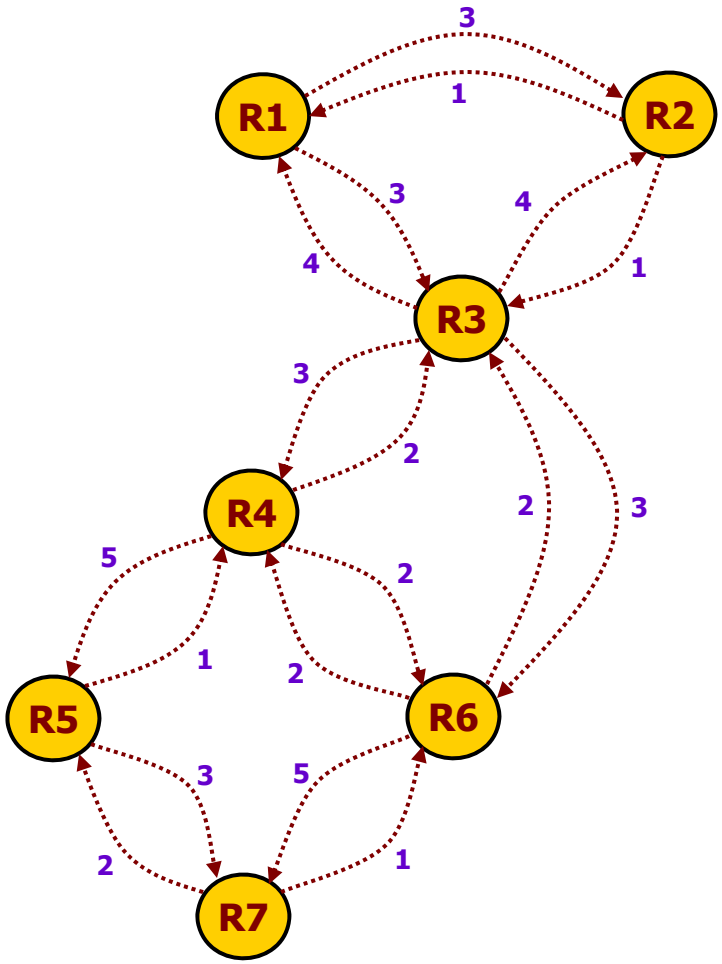
R6 LSA

	Interface /Mask	Cost
N5	50.50.50.6/24	2
N7	70.70.70.6/24	5

R7 LSA

	Interface /Mask	Cost
N6	60.60.60.7/24	2
N7	70.70.70.7/24	1

OSPF Example: Link-State



LSB

	Interface /Mask	Cost	
N1	10.10.10.1/24	2	R1 LSA
N2	20.20.20.1/24	3	

	Interface /Mask	Cost	
N2	20.20.20.2/24	1	R2 LSA
N3	30.30.30.2/24	2	

	Interface /Mask	Cost	
N2	20.20.20.3/24	4	R3 LSA
N5	50.50.50.3/24	3	

	Interface /Mask	Cost	
N4	40.40.40.4/24	5	R4 LSA
N5	50.50.50.4/24	2	

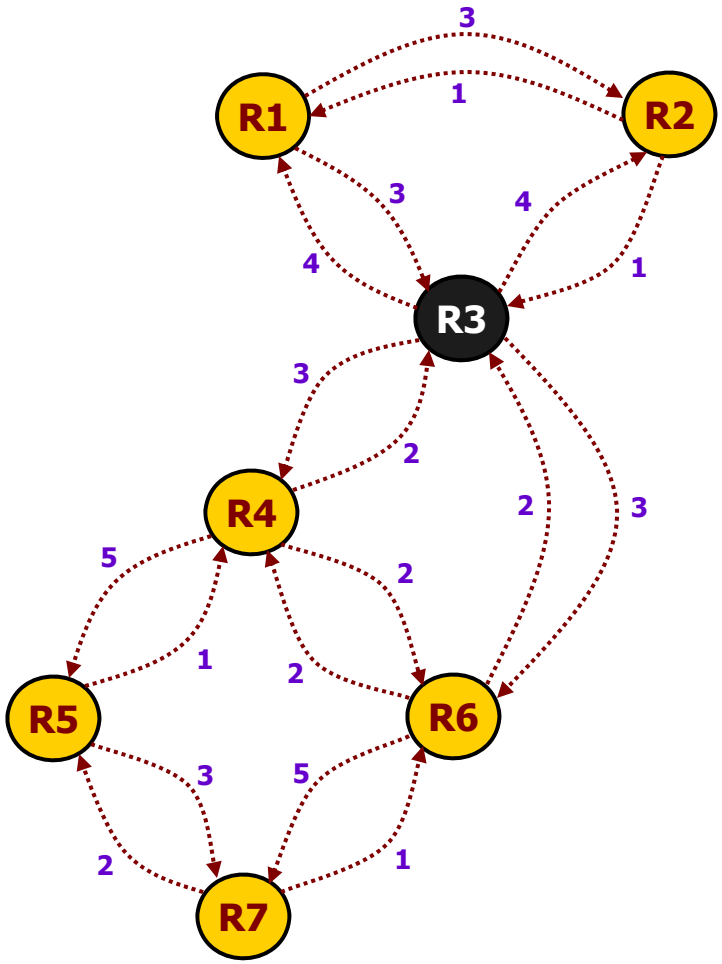
	Interface /Mask	Cost	
N4	40.40.40.5/24	1	R5 LSA
N6	60.60.60.5/24	3	

	Interface /Mask	Cost	
N5	50.50.50.6/24	2	R6 LSA
N7	70.70.70.6/24	5	

	Interface /Mask	Cost	
N6	60.60.60.7/24	2	R7 LSA
N7	70.70.70.7/24	1	

OSPF Example: Link-State

ตัวอย่าง R3 หา SPF Tree จาก Dijkstra



LSB

	Interface /Mask	Cost	
N1	10.10.10.1/24	2	R1 LSA
N2	20.20.20.1/24	3	

	Interface /Mask	Cost	
N2	20.20.20.2/24	1	R2 LSA
N3	30.30.30.2/24	2	

	Interface /Mask	Cost	
N2	20.20.20.3/24	4	R3 LSA
N5	50.50.50.3/24	3	

	Interface /Mask	Cost	
N4	40.40.40.4/24	5	R4 LSA
N5	50.50.50.4/24	2	

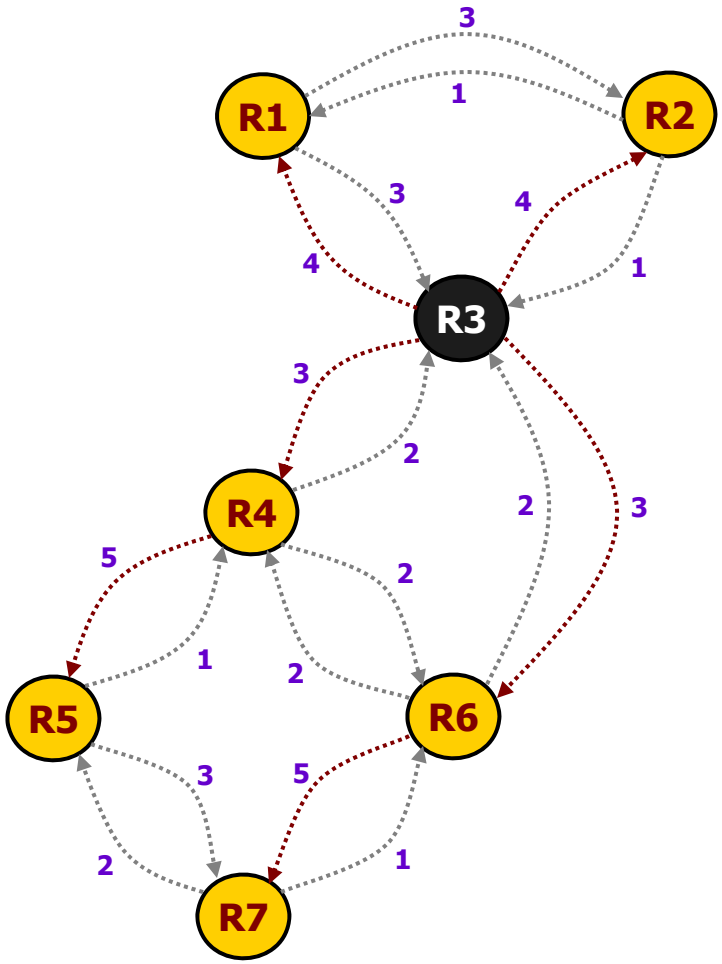
	Interface /Mask	Cost	
N4	40.40.40.5/24	1	R5 LSA
N6	60.60.60.5/24	3	

	Interface /Mask	Cost	
N5	50.50.50.6/24	2	R6 LSA
N7	70.70.70.6/24	5	

	Interface /Mask	Cost	
N6	60.60.60.7/24	2	R7 LSA
N7	70.70.70.7/24	1	

OSPF Example: Link-State

ตัวอย่าง R3 หา SPF Tree จาก Dijkstra



LSB

	Interface /Mask	Cost	
N1	10.10.10.1/24	2	R1 LSA
N2	20.20.20.1/24	3	

	Interface /Mask	Cost	
N2	20.20.20.2/24	1	R2 LSA
N3	30.30.30.2/24	2	

	Interface /Mask	Cost	
N2	20.20.20.3/24	4	R3 LSA
N5	50.50.50.3/24	3	

	Interface /Mask	Cost	
N4	40.40.40.4/24	5	R4 LSA
N5	50.50.50.4/24	2	

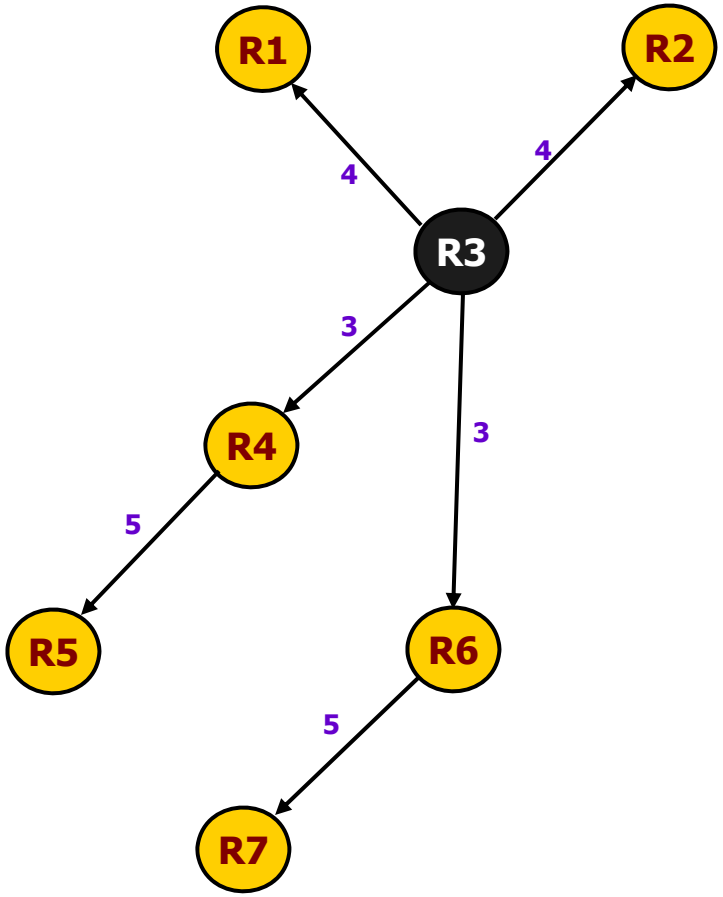
	Interface /Mask	Cost	
N4	40.40.40.5/24	1	R5 LSA
N6	60.60.60.5/24	3	

	Interface /Mask	Cost	
N5	50.50.50.6/24	2	R6 LSA
N7	70.70.70.6/24	5	

	Interface /Mask	Cost	
N6	60.60.60.7/24	2	R7 LSA
N7	70.70.70.7/24	1	

OSPF Example: Link-State

ตัวอย่าง R3 หา SPF Tree จาก Dijkstra



LSB

	Interface /Mask	Cost	
N1	10.10.10.1/24	2	R1 LSA
N2	20.20.20.1/24	3	

	Interface /Mask	Cost	
N2	20.20.20.2/24	1	R2 LSA
N3	30.30.30.2/24	2	

	Interface /Mask	Cost	
N2	20.20.20.3/24	4	R3 LSA
N5	50.50.50.3/24	3	

	Interface /Mask	Cost	
N4	40.40.40.4/24	5	R4 LSA
N5	50.50.50.4/24	2	

	Interface /Mask	Cost	
N4	40.40.40.5/24	1	R5 LSA
N6	60.60.60.5/24	3	

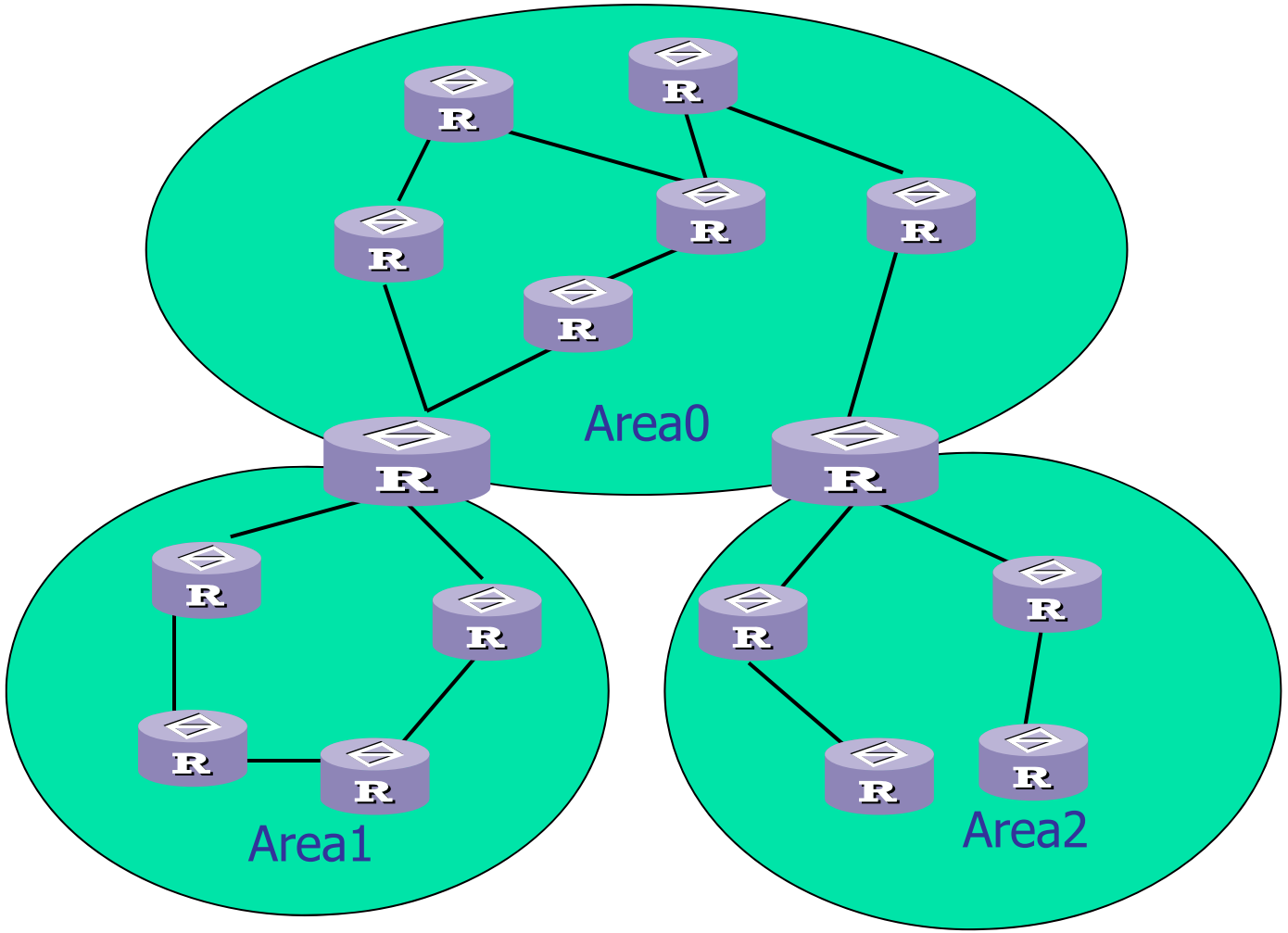
	Interface /Mask	Cost	
N5	50.50.50.6/24	2	R6 LSA
N7	70.70.70.6/24	5	

	Interface /Mask	Cost	
N6	60.60.60.7/24	2	R7 LSA
N7	70.70.70.7/24	1	

27.14 OSPF Area

- **อันหนึ่งที่ทำให้ OSPF มีความซับซ้อนมากกว่า Routing Protocol อื่น และเป็นตัวที่ทำให้มันเหนือกว่าตัวอื่นด้วยคือมันสามารถทำ Hierarchical Routing**
- **OSPF ยอมให้เราแบ่งกลุ่ม Router ภายใน AS ออกเป็น Area**
  - Router ภายใน Area จะมี LSB ที่เหมือนกัน
  - Routing Information ส่งภายใน Area จะเหมือนกับที่กล่าวมา
  - Routing Information ที่แลกเปลี่ยนระหว่าง Area จะถูกสรุป และส่งผ่าน Router ที่เชื่อมระหว่าง Area เรียก ABR (Area Border Router)
  - การทำเช่นนี้จะลด LSA ที่ Router จะต้องส่ง และลดขนาด LSB ลง
  - OSPF กำหนด Area กลางเรียก Area 0 (0.0.0.0) หรือ Backbone Area
  - Area อื่นๆจะต้องเชื่อมต่อกับ Backbone Area ผ่าน ABR
- **นอกจากนี้ OSPF ยังมีการกำหนด Router ที่จะสรุป Routing Information แลกเปลี่ยนกับ Routing Protocol อื่น หรือออกนอก AS ชื่อ ASBR (AS Border Router)**

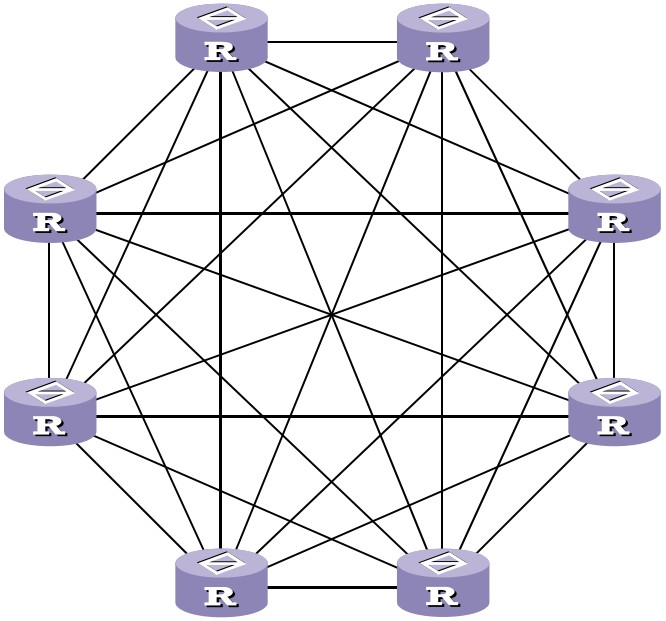
27.14 OSPF Area



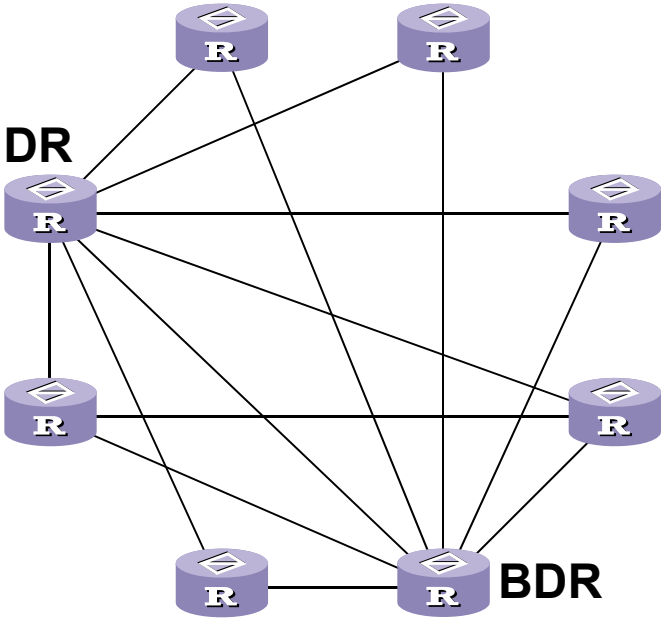
OSPF Designated Router และ Backup Designated Router

- **เพื่อลดจำนวนการส่ง LSA และการ Broadcasting (Multicast) รวมถึงการเชื่อมต่อกับ Router ทุกๆตัวใน Area**
  - OSPF กำหนดให้มี Router หนึ่งตัวทำหน้าที่รวบรวม LSA ของ Area จัดทำเป็น LSB และส่งให้กับ Router ทุกๆตัว เรียก Designated Router (DR)
    - DR จะเลือกจาก Router ที่มี Router ID ต่ำสุด
  - OSPF ยังกำหนด Backup DR(BDR) ที่จะทำงานในกรณีที่ DR เกิดมีปัญหา

OSPF Designated Router และ Backup Designated Router



M= n(n-1)/2
= 28

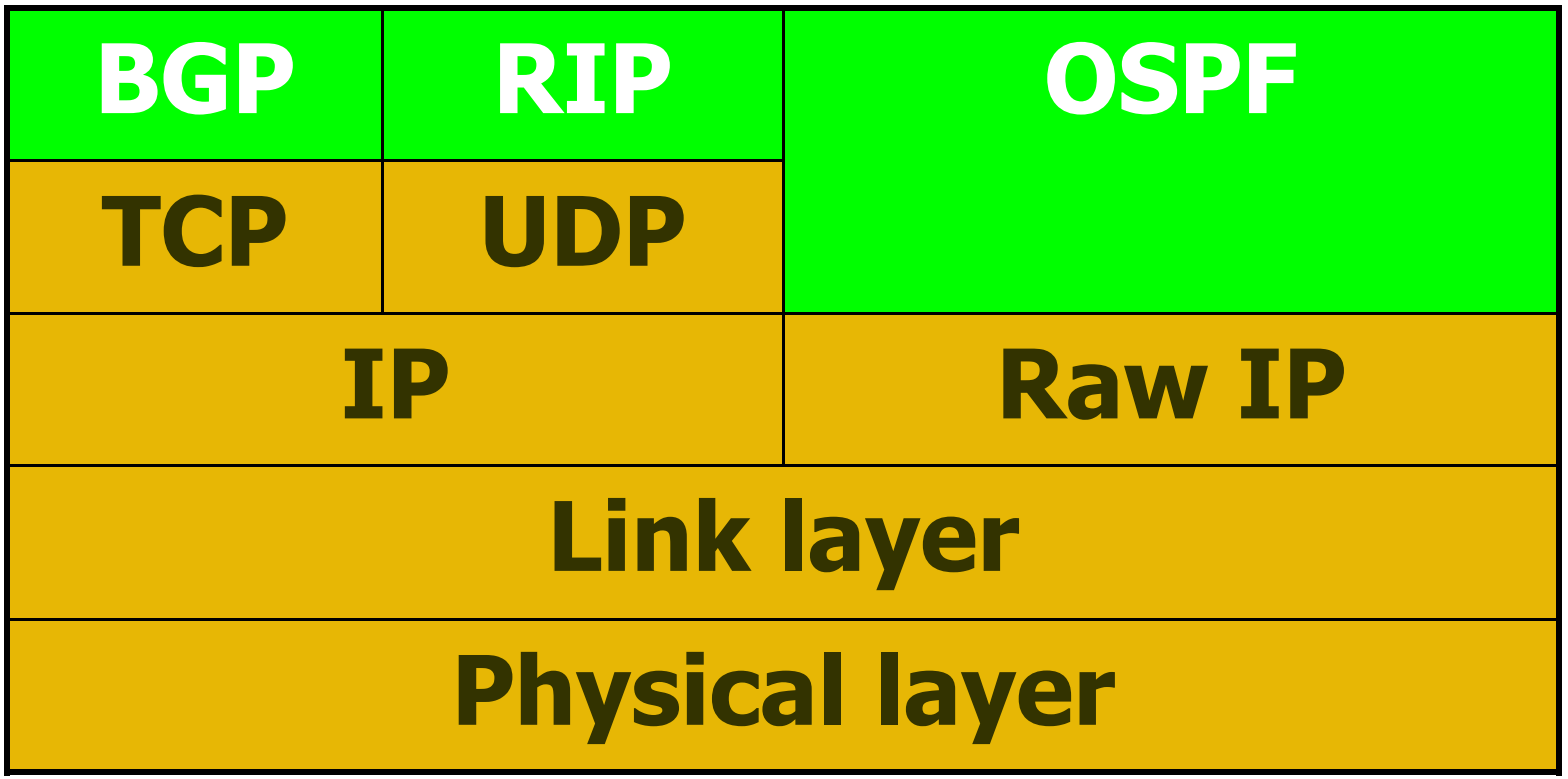


M= (n-2)×2+1
= 13

OSPF Protocol Layer

- **OSPF กำหนด Frame Format ที่จะบรรจุโดยตรงลงใน IP ไม่ผ่าน Transport Layer**
  - Packet Type ทั้งหมด 5 แบบ แต่ละแบบยังมีแยกออกไปอีก
    - OSPF Packet Type I = Hello Packet ส่งทุก 10 วินาที
    - OSPF Packet Type 4 = Link-State Update Packetจะส่ง LSA
      - LSA ที่ส่งมีหลายแบบ
- **Protocol Number ของ OSPF คือ 89**
- **การส่ง LSA จะส่งผ่าน Multicasting**
  - 224.0.0.5

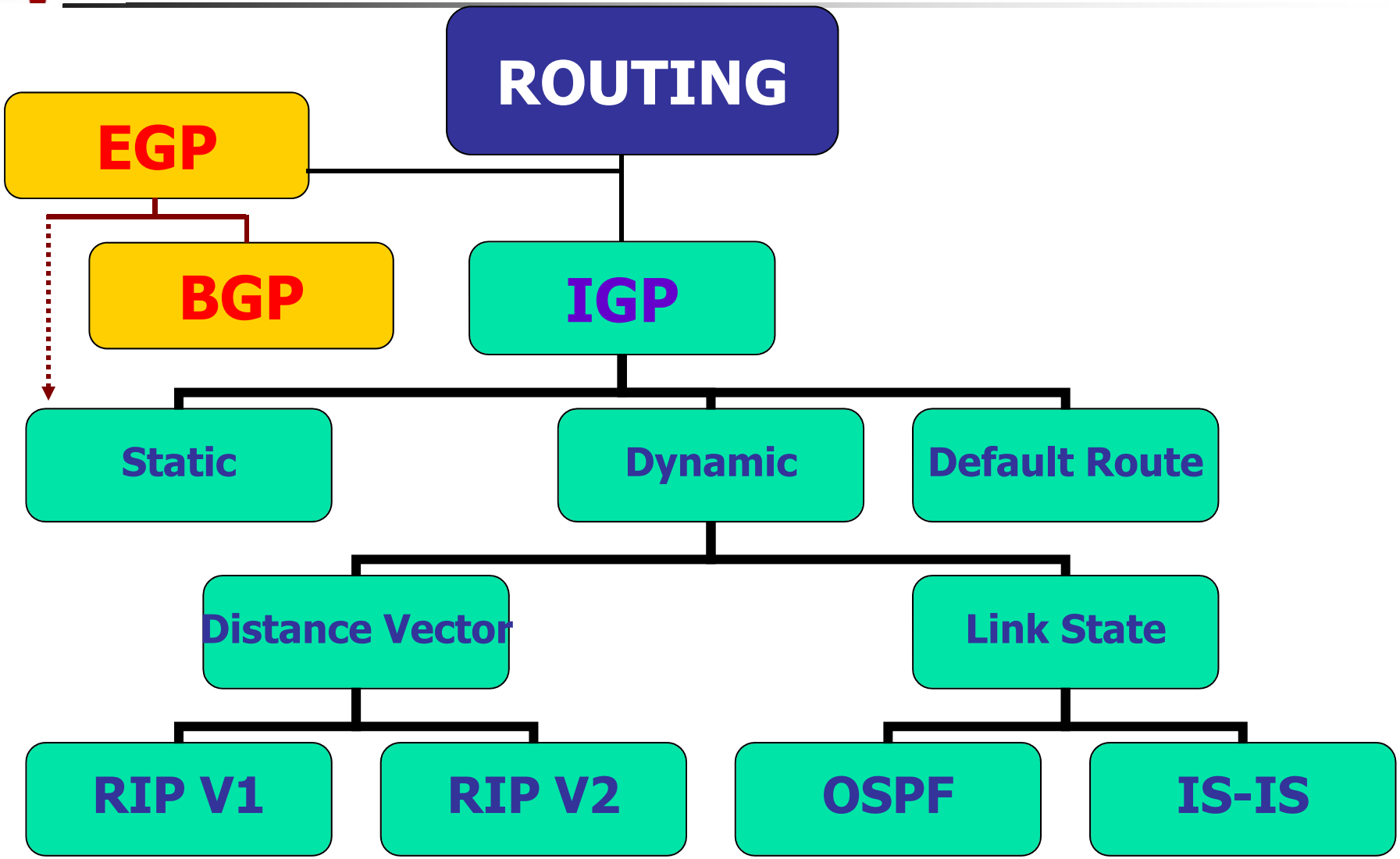
Position of the Dynamic Routing Protocols in the Protocol Stack



27.15 Intermediate System – Intermediate System (IS-IS)

- **ออกแบบโดย Digital Equipment Corporation ให้เป็นส่วนหนึ่งของ DECNET V ให้เป็น IGP**
- **ถูกสร้างขึ้นในเวลาเดียวกับ OSPF และทำงานคล้ายกัน โดยใช้วิธีของ Link-State และ Dijkstra Algorithm**
- **ข้อแตกต่างจาก OSPF**
  - IS-IS เป็น Proprietary ในตอนแรก, OSPF เป็น Open Standard
  - OSPF ถูกออกแบบให้ Run บน IP ส่วน IS-IS ถูกออกแบบให้ Run บน CLNS (Connectionless Network Service อยู่ในมาตรฐานของ OSI)
  - OSPF ออกแบบมาให้ส่งผ่าน Route ของ IPv4 ส่วน IS-IS จะส่งผ่าน Route สำหรับ OSI Protocol
  - ต่อมา OSPF ได้มีการปรับปรุง และใส่ความสามารถเพิ่มเติมลงไป ทำให้มี Overhead สูงกว่า IS-IS
- **IS-IS ถูกลืมไประยะหนึ่งและได้รับความนิยมขึ้นมา เนื่องจาก**
  - ปัจจุบัน DEC ถูกยุบไปแล้ว IS-IS ไม่ใช่ Proprietary Property
  - IS-IS มี Overhead ต่ำกว่า OSPF
  - IS-IS ได้ถูกออกแบบใหม่ให้ใช้กับ IP ได้
  - OSPF ไม่สามารถใช้กับ IPv6 ได้ จะต้องใช้ OSPF 6 ตัวใหม่
- **นิยมใช้ใน ISP เนื่องจาก ISP เป็น Network ขนาดใหญ่ การใช้ OSPF จะมี Overhead สูง**

IP Routing


ROUTING
EGP
BGP
IGP
Static
Dynamic
Default Route
Distance Vector
Link State
RIP V1
RIP V2
OSPF
IS-IS

27.16 Multicast Routing
27.16.1 Multicast Semantic

- **Multicast Routing** จะต่างจาก **Unicast Routing** เนื่องจาก **Internet** ยอมให้ **Multicast Group** เป็น **Dynamic** และไม่จำเป็นต้องมีการแสดงตัวผู้ส่ง
  - สมาชิกสามารถเป็นหรือบอกเลิกได้ตลอดเวลาตามความต้องการ
  - เมื่อเป็นสมาชิกของ Group จะได้รับทุกๆ Packet ที่มีการส่งให้กับ Group
    - ถ้ามีหลาย Application บน Host เดียวกันเป็นสมาชิกของ Group เดียวกัน จะได้รับ Packet เพียงหนึ่ง Copy จากนั้น Packet จะถูก Copy ให้กับแต่ละ Application
  - เมื่อ Application บอกเลิกการเป็นสมาชิกของ Group ตัว Host จะยังไม่แจ้งการบอกเลิกไปยัง Router ที่ต่ออยู่จนกระทั่ง Application สุดท้ายบอกเลิกเป็นสมาชิก Host จึงจะบอก Router ว่าไม่ต้องการเป็นสมาชิกแล้ว
- **IP Multicast Group** จะเป็นลักษณะ **Anonymous**
  - เราไม่สามารถรู้ผู้ส่งหรือผู้รับ หรือจำนวนสมาชิกในขณะนั้น
  - Router และ Host ไม่รู้ว่า Application ใดเป็นผู้ส่งข้อมูลให้แก่ Group
  - กลุ่มของ Multicast Group เพียงแค่กำหนดกลุ่มของผู้รับ
    - ผู้ส่งไม่จำเป็นต้องเป็นสมาชิกของ Group

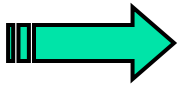
27.16 Multicast Routing
27.16.2 IGMP

- การที่ Host จะ Join หรือ Leave จาก Multicast Group จะกระทำผ่าน **Internet Group Multicast Protocol** (IGMP) กับ Router ที่เชื่อมต่อกับ Network ของ Host นั้น
- Protocol จะใช้เฉพาะสำหรับการสื่อสารระหว่าง Host และ Router
- Protocol จะกำหนด Host ไม่ใช่ Application ให้เป็นสมาชิกของ Multicast Group
  - ถ้ามีหลาย Application เป็นสมาชิกของ Group จะมองเหมือนเป็นสมาชิกเดียว และเป็นหน้าที่ของ Host ที่จะทำการ Copy ข้อมูลส่งให้แต่ละ Application
  - เมื่อ Application สุดท้ายบอกเลิกการเป็นสมาชิก Host จะส่ง IGMP ไปบอกยัง Router เพื่อบอกเลิก

27.16 Multicast Routing

27.16.3 การ Forward และการค้นหาสมาชิก

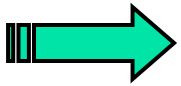
- **เมื่อ Router พบว่ามี Host บน Network ที่มันเชื่อมต่ออยู่เป็นสมาชิกของ Multicast Group มันจะต้องจัดสร้างเส้นทางไปยัง Group และทำการส่ง Datagram ที่มันได้รับสำหรับ Group ไปยัง Host สมาชิก**
  - ดังนั้น Router จะมีหน้าที่ในการกระจาย Multicast Routing Information
- **Multicast Routing จะซับซ้อนและยากกว่า Unicast Routing เนื่องจาก Group เป็น Dynamic และการใช้ Anonymous Sender**
  - ขนาดและ Topology ของ Group อาจจะเป็นภายใน Organization หรือทั่วโลก โดยมีสมาชิกไม่กี่คนจนถึงเป็นล้านคน (เช่น Webcast Application)



27.16 Multicast Routing

27.16.3 การ Forward และการค้นหาสมาชิก

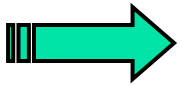
- Multicast Routing Protocol จะต้องทำงานได้รวดเร็วและต่อเนื่องเพื่อที่จะสามารถจัดการกับสมาชิกที่เป็น Dynamic
- เนื่องจาก User ใดก็ได้สามารถส่งข้อมูลให้ Group ดังนั้น Route Information จะต้องกินขอบเขตนอกเหนือจากสมาชิกของ Group
- Multicast Protocol จะใช้สามวิธีในการ Forward Datagram



27.16 Multicast Routing

27.16.3 การ Forward และการค้นหาสมาชิก

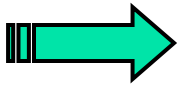
- **Multicast Protocol จะใช้สามวิธีในการ Forward Datagram**
  - Flood-and-Prune
  - Configuration-and-Tunneling
  - Core-Based Discovery
- **เราจะกล่าวรายละเอียดในแต่ละวิธีอย่างสังเขป**



27.16 Multicast Routing

27.16.3 การ Forward และการค้นหาสมาชิก

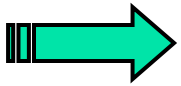
- **Flood-and-Prunes**
  - วิธีนี้จะดีที่สุดถ้าขนาดของกลุ่มเล็ก และสมาชิกทุกคนเชื่อมต่อกับ LAN ที่อยู่ติดกัน เช่น Network ขององค์กร
  - การทำงานจะเริ่มจาก Router จะ Forward แต่ละ Datagram ไปยังทุกๆ Network
    - เมื่อ Multicast Datagram มาถึง Router จะ Forward ไปยังทุกๆ Direct Connect LAN ผ่าน Hardware Multicast
  - เพื่อป้องกัน Loop วิธีการนี้จะใช้เทคนิคที่ชื่อ Reverse Path Broadcasting (RPB) ในการ Break Loop
  - ขณะที่ Router ทำการ Flood ข้อมูลไปยังทุก Network มันจะมีการแลกเปลี่ยนข้อมูลเกี่ยวกับสมาชิกของกลุ่ม
    - ถ้า Router เรียนรู้ว่า Network ใดไม่มีสมาชิกอยู่ มันจะหยุดการส่งข้อมูล Multicast ให้ Network นั้น นี่เป็นที่มาของคำว่า ‘Prunes”



27.16 Multicast Routing

27.16.3 การ Forward และการค้นหาสมาชิก

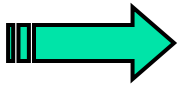
- **Configuration-and-Tunneling**
  - วิธีนี้จะใช้ได้ดีในกรณีที่สมาชิกกระจายตัว กินขอบเขตกว้าง โดยที่แต่ละ Site มีสมาชิกไม่กี่คน
  - Router ในแต่ละ Site จะถูก Configure ให้รู้จัก Site อื่นๆ
  - เมื่อมี Multicast Datagram มาถึง ตัว Router จะส่ง Datagram ไปยังทุก Site ที่เป็น LAN ที่เชื่อมต่อกับมันโดยตรง ผ่าน Hardware Multicast
  - จากนั้น Router จะมาดูที่ตาราง Configuration ว่า Datagram นี้จะต้องส่งไปยัง Remote Site ใด
    - การส่งจะเป็นการบรรจุ IP Multicast ลงใน IP Unicast Datagram เรียกว่าการทำ Tunneling (IP-in-IP Tunneling)
      - ดังนั้นการ Forward Multicast Datagram จะผ่านทาง Unicast Routing เนื่องจากมีการทำ Tunneling
  - คือการส่งภายใน Site ใช้ Multicast แต่การส่งให้ Site อื่นจะใช้ Unicast Tunneling



27.16 Multicast Routing

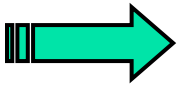
27.16.3 การ Forward และการค้นหาสมาชิก

- **Core-Based Discovery**
  - ในกรณีที่ขนาดกลุ่มและขอบเขตอยู่ระหว่างกลาง หรือมีการปรับเปลี่ยนไปมา สองวิธีแรกจะใช้ไม่ดี เราต้องการ Protocol ที่สามารถรองรับได้หลายๆ รูปแบบของกลุ่ม
  - วิธีนี้จะใช้การกำหนด Core Unicast Address สำหรับแต่ละ Multicast Group
  - เมื่อ Router R1 ได้รับ Multicast Datagram มันจะทำการ Encapsulate Datagram นั้นลงใน Unicast Datagram และส่งไปยัง Core Unicast Address ของ Group
    - เมื่อ Unicast Datagram นี้เดินทางผ่าน Internet ตัว Router ในทางผ่านแต่ละตัวจะดู Content ภายใน ถ้า Router เป็นส่วนหนึ่งของกลุ่ม มันจะ Process Multicast Message และส่ง Multicast Datagram ให้กับสมาชิกในส่วนของมัน
  - การเป็นสมาชิกของกลุ่มจะใช้วิธีการอย่างเดียวกัน
    - เมื่อ Router ได้รับการร้องขอการเป็นสมาชิก มันจะเพิ่มเส้นทางลงในตาราง Multicast Table ของมัน
  - ดังนั้นสมาชิกของ Multicast Group จะขยายตัวออกจาก Core และ Router จะสร้างเส้นทางเชื่อมต่อเป็น Multicast Tree



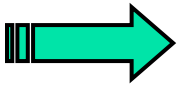
27.16 Multicast Routing
27.16.4 Multicast Protocols

- **มีหลาย Protocol ที่ถูกเสนอขึ้นมา แต่ยังไม่มีตัวใดที่สามารถใช้งานได้อย่างกว้างขวางตลอดทั้ง Internet (Internet-wide multicast routing)**
  - Distance Vector Multicast Routing Protocol (DVMRP)
    - เป็น Protocol ที่ถูกใช้โดย UNIX program 'mrouted' และใน Internet Multicast backBONE (MBONE)
    - DVMRP จะทำ Local Multicast โดยใช้ IP-in-IP encapsulation และส่ง Multicast Datagram จาก Site หนึ่งไปยังอีก Site หนึ่ง
    - รายละเอียด ดูได้จาก http://www.lbl.gov/web/Computers-and-Networks.html#MBONE
  - Core Based Tree (CBT)
    - ใช้วิธีการให้ Router สร้าง Delivery Tree จากจุดศูนย์กลางมายังแต่ละกลุ่ม โดย CBT จะอาศัย Unicast Routing ในการส่งข้อมูลมายังศูนย์กลาง



27.16 Multicast Routing
27.16.4 Multicast Protocols

- Protocol Independent Multicast-Sparse Mode (PIM-SM)
  - เป็น Protocol ที่ใช้วิธีการเช่นเดียวกันกับ CBT ในการสร้าง Multicast Routing Tree
  - การส่งข้อมูลระหว่าง Site ซึ่งใช้ Unicast ไม่ได้กำหนดว่าจะต้องใช้ Unicast Routing Protocol อะไร
- Protocol Independent Multicast-Dense Mode (PIM-DM)
  - เป็น Protocol ที่ออกแบบมาให้ใช้ภายในองค์กร
  - Router จะใช้วิธีการ Flooding (PIM-DM Broadcast) Packet ของ Multicast ไปยังทุกๆตำแหน่งของ Network ภายในองค์กร
    - ถ้า Router ใด ไม่มีสมาชิกของ Multicast อยู่ จะส่งข้อมูลกลับให้ทำการ Prune Multicast Tree (หยุดส่ง Packet)
  - วิธีการนี้จะใช้ได้ดี ถ้า Multicast Session มีอายุสั้น เพราะไม่ต้องการการ Setup ก่อนที่จะมีการส่งข้อมูล



27.16 Multicast Routing
27.16.4 Multicast Protocols

- Multicast Extensions to the Open Shortest Path First Protocol (MOSPF)
  - MOSPF ได้ถูกออกแบบเพื่อจะผ่าน Multicast Route ระหว่าง Router ภายในองค์กร
  - โดย MOSPF จะอาศัยการทำงานของ OSPF และทำงานร่วมกับ Link-State Routing
- **Multicast Routing เป็นเรื่องที่ยากมาก แม้ว่าจะมีการวิจัยค้นคว้ามานาน แต่ยังไม่มี Protocol ที่เป็น General-Purpose Internet Multicast ที่ประสพผลสำเร็จ**

Protocol	Type
DVMRP	Configuration-and-Tunneling
CBT	Core-Based-Discovery
PIM-SM	Core-Based-Discovery
PIM-DM	Flood-And-Prune
MOSPF	Link-State (within an organization)

Extra: Network Design Tips การใช้ VLAN and Subnet

- **การทำ VLAN คือการแบ่ง Switch ให้มีการทำงานเหมือนกับเป็น Switch หลายตัว**
  - Host ที่ต่อกับแต่ละ VLAN จะเหมือนกับว่าเป็น LAN คนละวง
  - แต่ละ VLAN จะ Broadcast กันภายใน คือเป็นหนึ่ง Broadcast Domain
- **ในการนำ TCP/IP มาใช้กับ LAN เราจะทำ Subnet ให้ LAN แต่ละวงเป็นหนึ่ง Network**
  - ดังนั้น หนึ่ง VLAN ในทางปฏิบัติคือหนึ่ง Subnet และ หนึ่ง Broadcast Domain
  - อย่างไรก็ตาม พึงเข้าใจว่า VLAN Number เป็นหมายเลขอ้างอิงเพื่อแบ่ง LAN และรู้จักเฉพาะ Switch นั้นๆ ในขณะที่ IP Number นั้นเป็น Global Address
  - เราสามารถใช้ VLAN เบอร์เดียวกัน สำหรับคนละ Subnet ที่อยู่ต่าง Switch กัน และไม่มีการเชื่อมกันในระดับ Layer 2 แต่ไม่มีเหตุผลอะไรที่จะทำเช่นนั้น เพราะ VLAN Number สามารถตั้งได้อย่างเหลือเฟือ
  - ดังนั้น แต่ละ Subnet ควรให้ VLAN Number ที่แตกต่างกัน

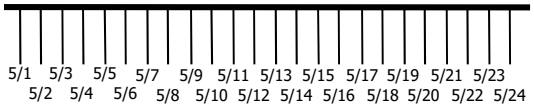
Extra: Network Design Tips การใช้ VLAN and Subnet

- การทำ VLAN Tag เป็นเสมือนการรวม LAN ที่อยู่ต่าง Switch กัน เป็น LAN เดียวกันและเป็น Subnet เดียวกัน (เพิ่มจำนวนของ LAN Port แต่ต่าง Switch)
- VLAN Tag เป็น Protocol ที่วางบน Layer 2
  - IP Packet ไม่มีข้อมูลของ VLAN Tag
- Tag จะใส่ลงใน Frame ที่ส่งผ่าน Port ที่ทำ Tag เท่านั้นและจะถูกนำออกเมื่อถึง Switch ปลายทาง
  - ดังนั้น VLAN Tag จะรู้จักกันระหว่าง Switch สองตัวเท่านั้น
  - Default VLAN ของ Port ที่ทำ Tagging จะไม่ใส่ VLAN Tag
- VLAN และ VLAN Tag ทำให้ Network ต้องมีสอง Diagram
  - Logical Diagram แสดงการเชื่อมต่อในการทำงานระหว่าง VLAN หรือ Subnet
  - Physical Diagram แสดงการเชื่อมต่อระหว่าง Switch ทาง Physical

Extra: Network Design Tips การใช้ VLAN and Subnet

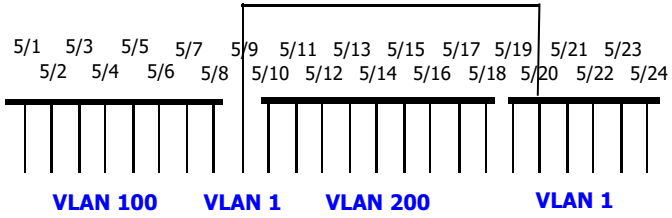
- ตัวอย่าง Switch 24 Port

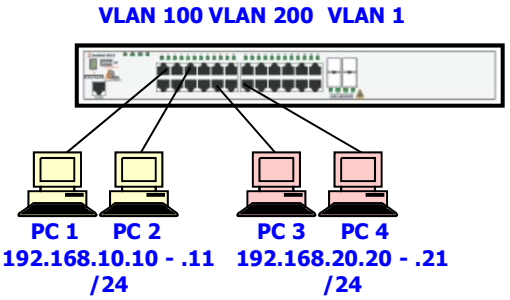
Physical Diagram

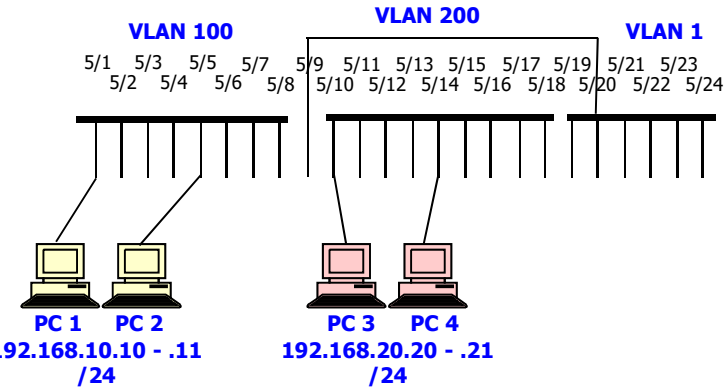


Logical Diagram

- เมื่อแบ่ง VLAN



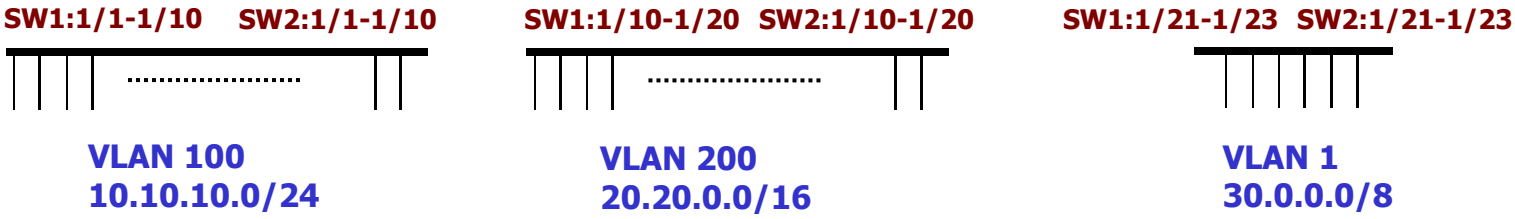




VLAN Tag Diagram

Physical Diagram

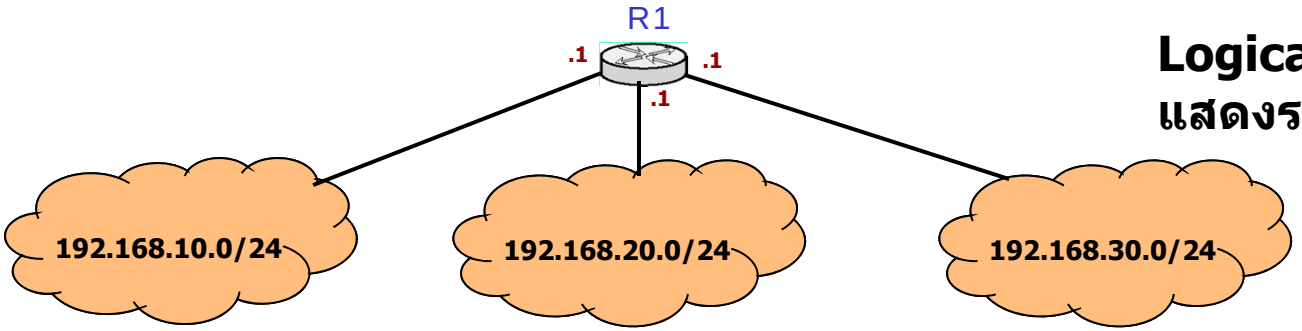
Logical Diagram



การเชื่อมต่อ VLAN

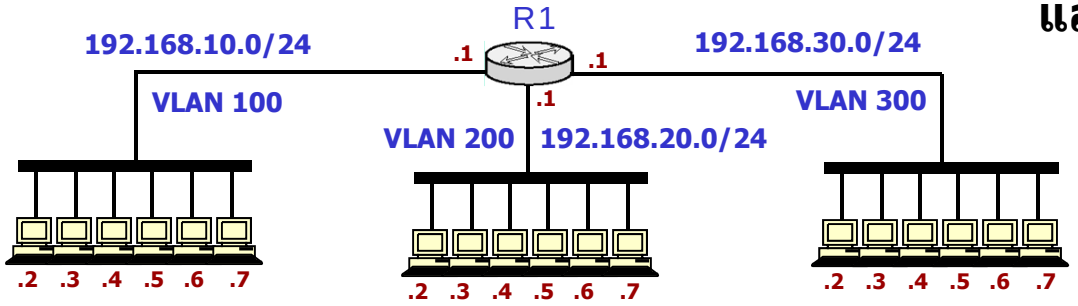
- คือการเชื่อมต่อ LAN คนละวงเข้าด้วยกัน
- คือการเชื่อมต่อแต่ละ Subnet หรือต่าง Network เข้าด้วยกัน
- ต้องใช้อุปกรณ์ Layer 3 คือ Router หรือ Switch L3
  - การส่งข้อมูลข้าม LAN จะส่งได้ในระดับ IP Packet เท่านั้น
  - Broadcast ปกติจะไม่ผ่าน

สมมุติเรามี 3 Network เชื่อมต่อผ่าน Router



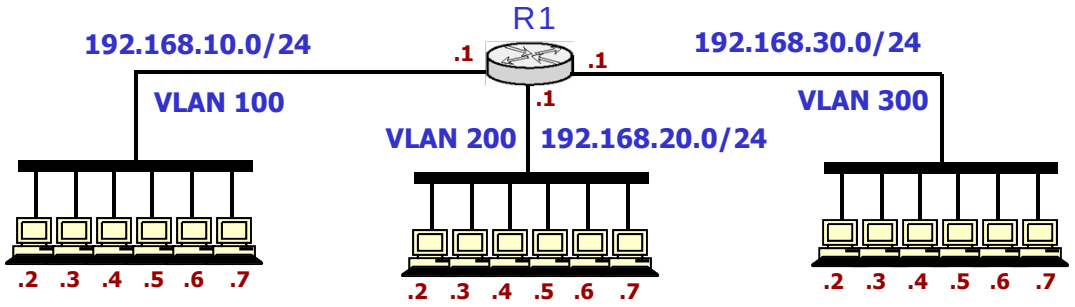
Logical Diagram
แสดงรายละเอียดระดับ L3

Logical Diagram
แสดงรายละเอียดระดับ L2
และ Host



ถ้าผู้ใช้งานแต่ละ Network มีน้อย เช่น 6 คน และอยู่บริเวณเดียวกัน เราสามารถใช้ Switch ตัวเดียว และ แบ่งเป็น 3 VLAN VLAN ละ 7 Port

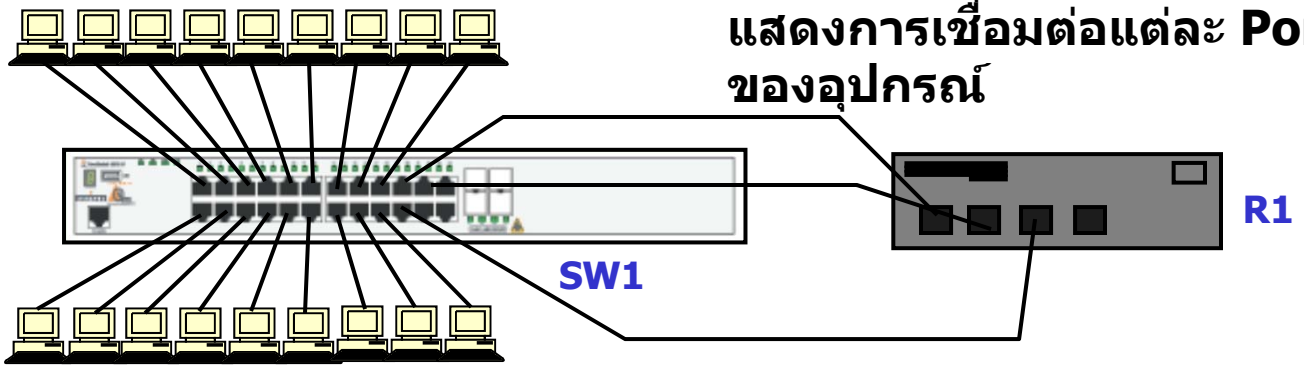
VLAN Diagram



Logical Diagram
แสดงรายละเอียดระดับ L2 และ Host

ถ้าผู้ใช้งานแต่ละ Network มีน้อย เช่น 6 คน และอยู่บริเวณเดียวกัน เราสามารถใช้ Switch ตัวเดียว และ แบ่งเป็น 3 VLAN VLAN ละ 7 Port

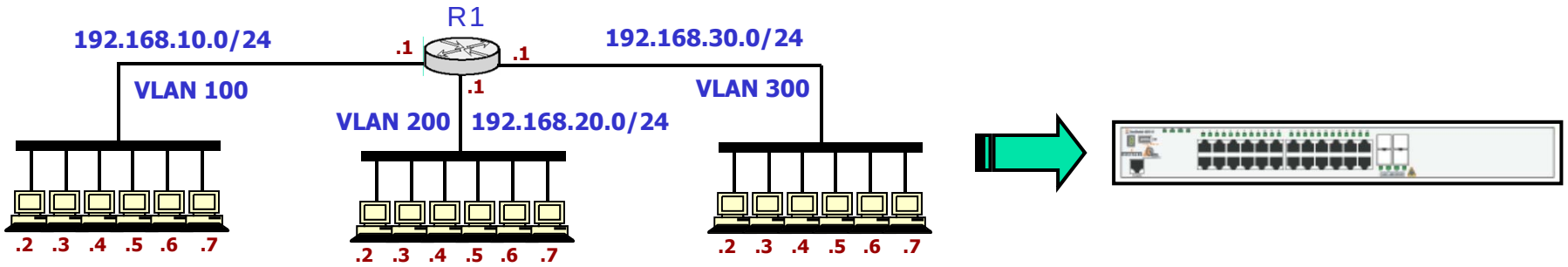
Physical Diagran
แสดงการเชื่อมต่อแต่ละ Port ของอุปกรณ์



VLAN100 Port 1/1,3,5,7,9,11,19 **VLAN300 Port 1/,13,14,15,16,17,18,21**
VLAN200 Port 1/2,4,6,8,10,12,20

Switch L3

- ในทางปฏิบัติ ถ้ามีการแบ่ง VLAN และเชื่อม VLAN รวมถึงการทำ Routing เราใช้ Switch L3 จะสะดวก ประหยัด และมีประสิทธิภาพสูงกว่า
- Switch L3 จะรวม Router Function อยู่ภายใน Switch
- นอกจากนี้ยังมี Switch L4 สามารถทำหน้าที่เป็น Firewall
- Switch L7 ทำ Security ได้ถึงระดับ Application
- เรียกรวมๆว่า Multilayer Switch



ขั้นตอนทั่วไปในการ Configure Switch L3 คือทำทีละ Layer

- **Layer 1:เชื่อมต่อสายก่อนและตรวจสอบ**
- **Layer 2:**
  - สร้าง VLAN
  - กำหนด Port ให้กับแต่ละ VLAN
  - ทำ VLAN Tagging (ถ้ามี)
- **Layer 3:**
  - กำหนด IP Interface ให้กับแต่ละ VLAN
    - เมื่อกำหนด IP Interface ตัว Switch จะทำงานใน Layer 3 โดยอัตโนมัติ
    - หนึ่ง VLAN คือหนึ่ง Subnet ดังนั้นระวังการกำหนด IP/Net Mask ให้แก่ Interface
    - เมื่อกำหนดแล้ว จะเป็นการกำหนดค่า Network ID (Prefix) ให้แก่ VLAN ไปในตัว
- **จากนั้นจึงค่อยทำ Routing**

End of Week 13

- **HW 8 Download**
- **No Class Week 14**
  - Songkran
- **Week 14: Advance Topic**
  - Textbook Part IV
  - QoS and IP Telephony

CPE 426 Computer Networks

Chapter 11:
PART IV in Textbook
Text Chapter 28: NW Performance
(QoS and DiffServ)
Text Chapter 29: IP Telephony

TOPICS

- Chapter 28 (Only Selected Topics)
  - 28.2 Performance Measurement
  - 28.3 Latency or Delay
  - 28.4 Throughput, Capacity and Goodput
  - 28.5 Understanding Throughput and Delay
  - 28.6 Jitter
  - 28.7 Relationship Between Throughput and Delay
  - 28.8 Measuring Delay, Throughput and Jitter
  - 28.9 Passive Measurement, Small Packet and Netflow
  - 28.10 QoS
  - 28.11 Fine-Grain and Coarse-Grain QoS
  - 28.12 Implementation of QoS
  - 28.13 Internet QoS Technologies
- Breaks
- Chapter 29 (Only Selected Topics)
  - Real-Time Transmission
  - Delay and Jitter
  - RTP
  - IP Telephony
  - Signaling
  - IP Telephone System Components
  - Protocol and Layering
  - H.323
  - SIP
  - Telephone Number Maping

28.2 Performance Measurements

- **การใช้คำว่า Low-Speed Network หรือ High-Speed Network ไม่เพียงพอต่อการบ่งบอกการทำงานของมัน เนื่องจาก Network Technologies เปลี่ยนเร็วมาก**
  - Low-Speed ปัจจุบัน เป็น High-Speed เมื่อ 2-3 ปีก่อน
- **ตัววัด Performance หลักๆที่ใช้มีสามตัว**
  - Latency(Delay) เป็นตัววัดเวลาที่ต้องใช้ในการส่งข้อมูลผ่าน Network
  - Throughput(Capacity) เป็นตัววัดจำนวนข้อมูลที่ส่งได้ในหนึ่งหน่วยเวลา
  - Jitter (Variability) เป็นค่าการเปลี่ยนแปลงของค่า Delay และระยะเวลาของการเปลี่ยนแปลง

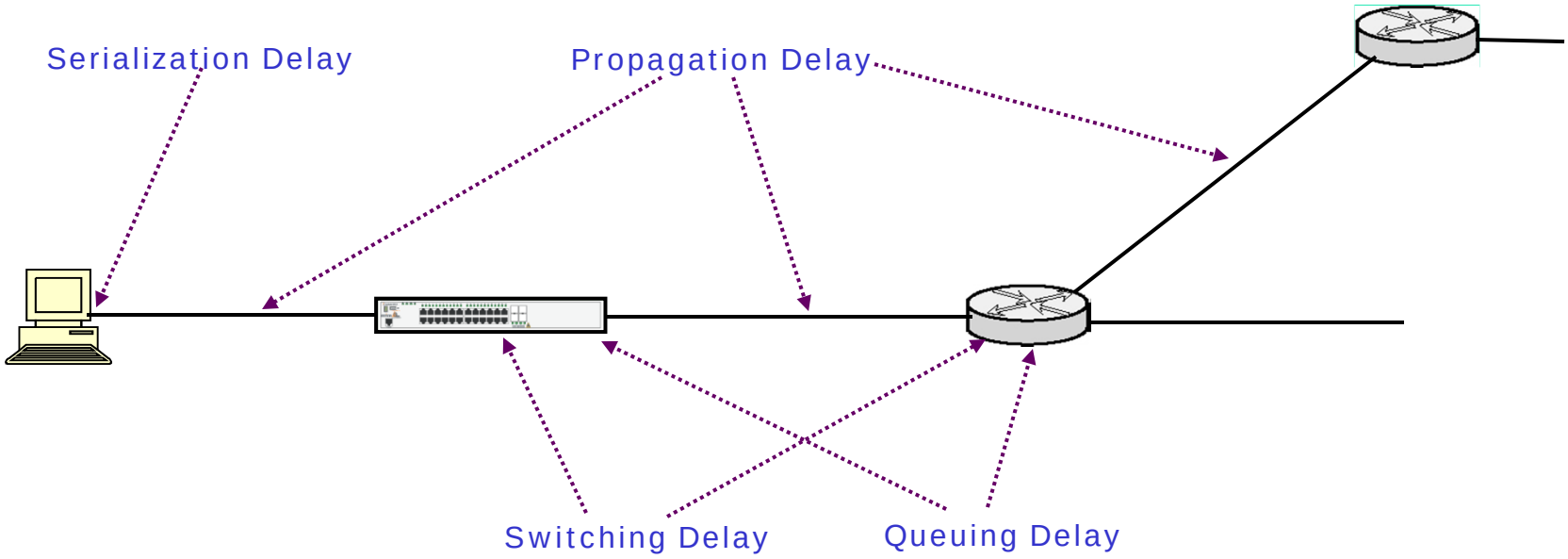
28.3 Latency or Delay

- **สามารถแบ่งออกได้เป็น**
  - Propagation Delay
    - เวลาที่สัญญาณต้องเดินทางผ่าน Transmission Medium ขึ้นอยู่กับความเร็วของสัญญาณและระยะทางที่ส่ง
  - Access Delay
    - เวลาที่ใช้ในการ Access Medium ยกตัวอย่างเช่นใน LAN ที่ใช้ CSMA/CD หรือใน WLAN ที่ใช้ CSMA/CA
  - Switching Delay
    - เวลาที่ใช้ในการส่งผ่าน Packet ในอุปกรณ์ Switch หรือ Router (เวลาที่อุปกรณ์ต้องใช้ในการ Process Packet)

Latency or Delay

- **สามารถแบ่งออกได้เป็น(2)**
  - Queuing Delay
    - เวลาที่ต้องรอใน Queue ก่อนที่จะถูกส่งออกไป
    - ขึ้นอยู่กับความยาวของ Queue
    - ค่านี้เป็นตัวกำหนดค่า Delay ใน Network
  - Server Delay
    - เวลาที่ Server ใช้ในการตรวจสอบการร้องขอและคำนวณการตอบสนอง
  - Serialization Delay
    - เวลาที่ต้องใช้ในการส่ง Packet ขึ้นกับ Data Rate และความยาวของ Packet

Latency


Serialization Delay
Propagation Delay
Switching Delay
Queuing Delay

28.4 Throughput, Capacity and Goodput

- **Capacity**
  - เป็นค่า Data Rate ของการส่งข้อมูล และแสดงจำนวน Data ที่สามารถส่งได้สูงสุด วัดเป็น bps
- **Throughput**
  - เป็นจำนวน Data ที่ส่งได้จริงผ่าน Network วัดเป็น bps (ไม่รวม Retransmission)
  - บางทีวัดเป็น Percent ของ Capacity เรียก Utilization
  - ค่านี้ปกติจะน้อยกว่า Capacity ขึ้นอยู่กับการทำ Flow Control
- **Goodput**
  - เป็นจำนวน Data จาก Application Layer ที่ส่งได้จริงๆ โดยตัด Overhead เช่นส่วน Header ของแต่ละ Layer ออก และไม่รวมส่วน Control Information, Support Protocol, Handshake, Congestion และ Retransmission
  - ต่างจากค่า Throughput ซึ่งจะรวมส่วนของ Overhead ด้วย

28.5 Throughput and Delay

- **Propagation Delay** เป็นตัวกำหนดเวลาที่แต่ละ **Bit** จะเดินทางใน **Network** จากต้นทางไปถึงปลายทาง
- **Throughput** กำหนดจำนวนบิตที่สามารถวิ่งผ่าน **Network** ได้ในแต่ละเวลา
- เราสามารถซื้อ **Throughput** เพิ่มได้ แต่ไม่สามารถจ่ายเงินเพิ่มเพื่อลดค่า **Delay** ได้
- ส่วน **Queuing Delay** เป็นเวลาที่ต้องรอใน **Queue** ของ **Switch/Router** ก่อนที่สามารถจะส่งข้อมูลได้

28.6 Jitter

- ค่านี้มีความสำคัญในกรณีที่เราต้องส่ง **Real-Time Voice** หรือ **Video**
- คือการเปลี่ยนแปลง หรือค่า **Variance** ของค่า **Delay**
- ปกติการส่ง **Real-Time Data** จะมีตัว **Buffer** ชื่อ **Jitter Buffer** ป้องกันการผันแปรของเวลาที่ข้อมูลมาถึง
  - ถ้าค่า Jitter สูงจะทำให้ Buffer ว่าง หรือ Overflow และเสียงจะขาดหายหรือภาพจะกระตุก
  - วิธีแก้คือใช้ Buffer ขนาดใหญ่ขึ้น แต่มีข้อเสียคือจะเกิด Delay ใน Buffer สูง และมันจะไม่เป็น Real-Time อย่างแท้จริง

Jitter

- **Real time Application เช่น Audio หรือ Video ต้องการรับข้อมูลอย่างต่อเนื่องและตามกำหนดเวลา**
  - Jitter ทำให้ข้อมูลบางส่วนมาถึงในเวลาที่ช้าเกินกำหนดที่จะต้องแสดงผล ทำให้คุณภาพลดลง
  - แม้ว่า Network จะมี Delay ต่ำ แต่ค่า Jitter สูง จะสู้ Network ที่มี Delay ปานกลาง แต่ Jitter ต่ำไม่ได้ เพราะ Delay แสดงเพียงค่าเฉลี่ยแต่ Jitter แสดง Variance (Standard Deviation)
- **การจัดการกับ Jitter**
  - ใช้ Isochronous Network (Circuit Switching Network)
  - ใช้ Protocol ในการช่วยจัดการกับ Jitter
    - Real-Time Protocol + Jitter Buffer

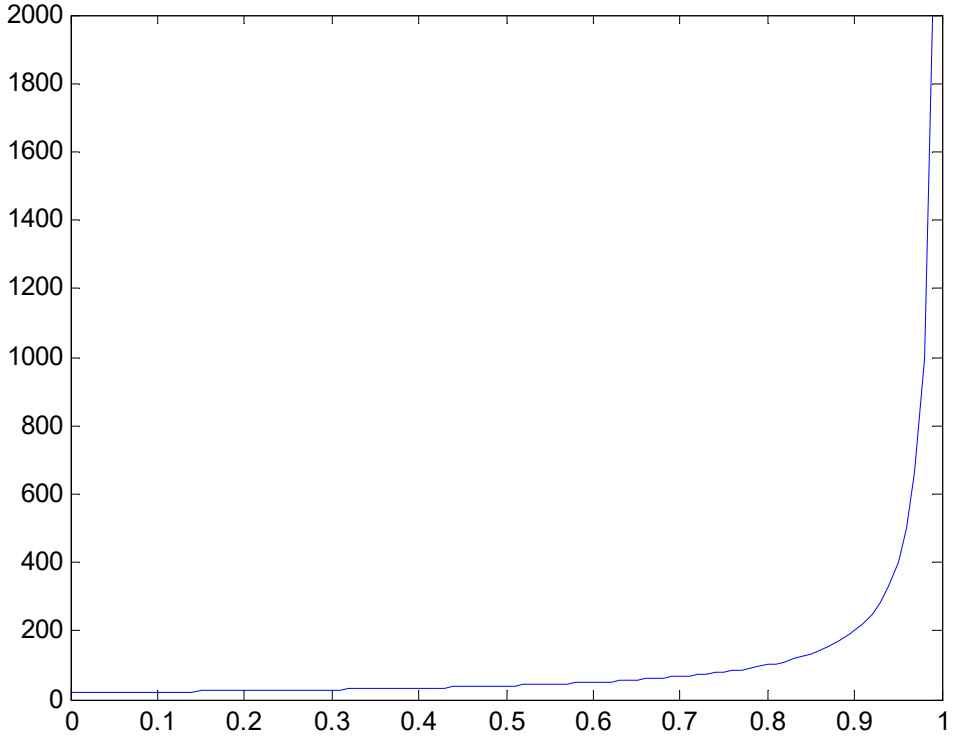
28.7 ความสัมพันธ์ระหว่าง Delay และ Throughput

- สามารถอธิบายโดย Queuing Theory (CPE332)
- ถ้าให้ u เป็นค่า Utilization เราจะได้ค่า Delay, D ดังนี้

$$D = \frac{D_0}{1-U}$$

$$U = 0.5 \Rightarrow D = 2D_0$$

D_0 คือ Delay ในกรณีที่ Network Idle



ความสัมพันธ์ระหว่าง Delay และ Throughput

- ค่า **Delay-Throughput Product**
- เป็นจำนวนของ **Bit** ที่วิ่งผ่าน **Network** ในเวลาใดเวลาหนึ่ง
  - Bits in Network = D x T (sec. x bit/sec.)

28.8 การวัดค่า Delay, Throughput และ Jitter

- **การวัดค่า Throughput และค่า Jitter ทำได้ไม่ยาก โดยการส่ง Packet ติดต่อกัน และจับเวลาว่าส่งข้อมูลได้เท่าไร และแต่ละ Packet ที่ไปถึงห่างกันเท่าไร**
  - อาจจะใช้ Protocol Analyzer ช่วยในการคำนวณ เช่น Sniffer, Ethereal หรือ Wireshark
- **Delay จะวัดยากกว่า เพราะต้องจับเวลาที่ส่ง และที่รับ**
  - หมายถึงเครื่องส่งและเครื่องรับต้อง Synchronization กัน
  - Protocol Analyzer ต้องมีอุปกรณ์เสริมช่วย บางตัวจะทำไม่ได้
  - เราอาจจะวัดค่า Round Trip Time แทน เช่นใช้การ 'Ping'
- **ค่าที่วัดได้เหล่านี้ ค่อนข้างจะซับซ้อน**
  - Network เป็น Asymmetric
  - Network มีการเปลี่ยนสถานะค่อนข้างเร็ว
  - การวัดค่าจะ Load Network และทำให้ค่าเปลี่ยน
  - Traffic ใน Network ปกติจะมีลักษณะ Burst

28.9 Passive Measurement, Small Packet and Netflow

- การวัดแบบ **Active** เราต้อง **Inject Traffic** ลงใน **Network** อาจจะทำให้ค่าเปลี่ยน
- เราสามารถเลือกใช้วิธีการวัดแบบ **Passive** คือ **Monitor Network** และนับจำนวน **Packet** ที่ผ่าน
  - เทคนิคที่นิยมคือ 'Netflow'
  - Router จะสุ่มตัวอย่าง Packet จากนั้นจะวิเคราะห์ข้อมูลจากตัวอย่างนั้น

Quality of Service (QoS)

- **เป็นการออกแบบ Network ที่สามารถให้ระดับของการให้บริการแก่ผู้ใช้**
  - ผู้ใช้สามารถเลือกบริการตามความเหมาะสม
    - Real-Time Voice/Video
    - Web Service
    - E-Mail
- **Internet IPv4 เป็น Best Effort ไม่มี QoS เนื่องจาก IP Network เป็น Datagram**
  - QoS Mechanism ได้ถูกนำมาใช้กับ IPv6 ก่อน ต่อมาภายหลังเมื่อ IPv6 ยังไม่ได้รับความนิยม จึงมีการนำ Mechanism เหล่านี้มาใช้กับ IPv4

28.11 Fine-Grain and Coarse-Grain QoS

- **Fine-Grain QoS**
  - ผู้ให้บริการยอมให้ลูกค้ากำหนดความต้องการ QoS ที่เฉพาะเจาะจง เช่นกำหนด Maximum Delay, Minimum Data Rate หรือ อื่นๆ
- **Coarse-Grain QoS**
  - ผู้ให้บริการจัดแบ่ง Class ของการให้บริการ แต่ละ Class จะมีค่า QoS (BW, Delay, Jitter) ต่างกัน จากนั้นลูกค้าเลือกว่าจะใช้ Class ใดที่เหมาะสมกับตัวเอง

28.11.1 Fine-Grain QoS and Flow

- **Network แรกที่มีการให้ QoS แบบ Fine Grain ตั้งแต่เริ่มออกแบบ Protocol คือ ATM**
- **ปัจจุบัน ATM ตายไปแล้ว แต่คำศัพท์เกี่ยวกับ QoS ที่ ATM ให้บริการยังใช้อยู่**
  - CBR Constant Bit Rate สามารถส่งข้อมูลได้ในอัตราที่คงที่
  - VBR Variable Bit Rate สามารถส่งข้อมูลได้ในอัตราเฉลี่ย และอยู่ในช่วงที่กำหนด
    - กำหนด SBR (Sustain Bit Rate), PBR (Peak Bit Rate),
    - กำหนด SBS (Sustain Burst Size), PBS (Peak Burst Size)
  - ABR Available Bit Rate สามารถส่งข้อมูลได้ตาม Bandwidth ที่เหลือ
  - UBR Unspecified Bit Rate ไม่กำหนดอัตราการส่งข้อมูล ส่งได้แบบ Best-Effort
- **Mechanism ที่ Internet นำมาใช้ในการให้บริการแบบ Fine-Grain เรียก Integrated Service (IntServ)**

28.11.12 Coarse-Grain QoS and Class of Service

- ทางเลือกอีกทางหนึ่งในการทำ QoS คือใช้วิธีของ Coarse-Grain
  - Traffic จะถูกแบ่งออกเป็น Class และกำหนด QoS Parameter ให้กับแต่ละ Class
- เนื่องจากใน Internet นั้นการทำ Fine-Grain จะทำได้ยาก เพราะ Router จะต้องจดจำ Flow และ State ของ Flow ซึ่งที่ Core Router ประกอบด้วยหลายล้าน Flow
  - นอกจากนี้ User ปกติจะไม่เข้าใจ QoS เพียงพอในการเลือก Performance Parameter
- Coarse-Grain สามารถกระทำได้ในทางปฏิบัติ

28.12 Implementation of QoS

- **Router ที่สามารถทำ QoS จะประกอบไปด้วย 4 Step**

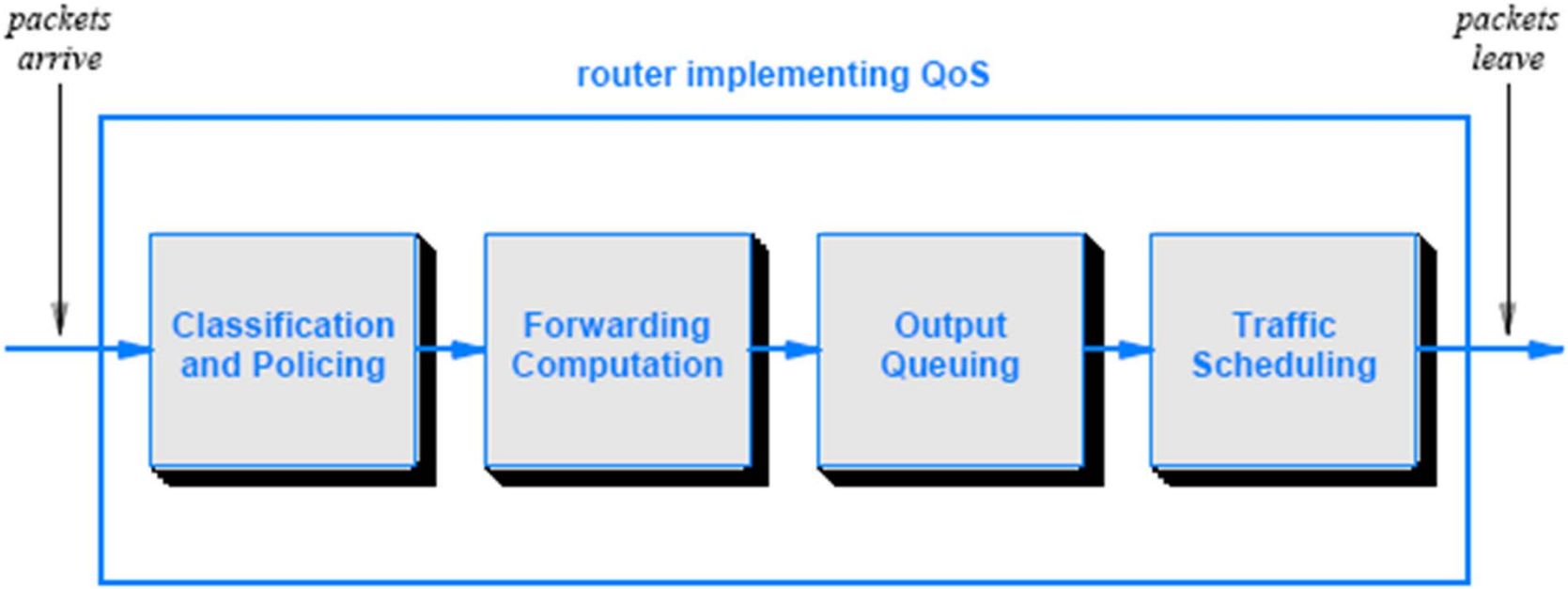


Figure 28.5 The four key steps used to implement QoS.

Implementation of QoS

- **Router ที่สามารถทำ QoS จะประกอบไปด้วย 4 Step**
  - 1. Classification and Policing
    - Router จะแบ่ง Class ของ Traffic โดยกำหนด Flow ID ให้แก่ Packet(Fine-Grain) หรือ กำหนด Class(Coarse-Grain) จากนั้น Router จะตรวจสอบ Policy ของ Class หรือ Flow นั้น ถ้าลูกค้าไม่กระทำตาม Policy ที่กำหนด จะมีการโยน Packet ทิ้งโดยใช้ค่า Probability

Implementation of QoS

- **Router ที่สามารถทำ QoS จะประกอบไปด้วย 4 Step**
  - 2. Forwarding Computation
    - คำนวณหา Next-Hop ของ Packet จากค่า Flow-ID ซึ่งอาจจะกำหนด Path ของการส่งข้อมูลสำหรับบาง Flow หรืออาจจะดูจาก IP ปลายทางและใช้ตาราง Routing Table
  - 3. Output Queuing
    - Router ที่ทำ QoS จะมีหลาย Output Queue (ผิดกับ Best-Effort Router ที่มี FIFO Queue เดียวและเป็น M/M/1) โดยส่วนนี้ Packet จะถูกนำมาแยกใส่ Queue ที่เหมาะสมสำหรับ Flow ของมัน
      - Coarse Grain ปกติจะใช้ 1 Queue ต่อหนึ่ง Class
      - Fine-Grain ปกติจะใช้ 1 Queue ต่อ Connection

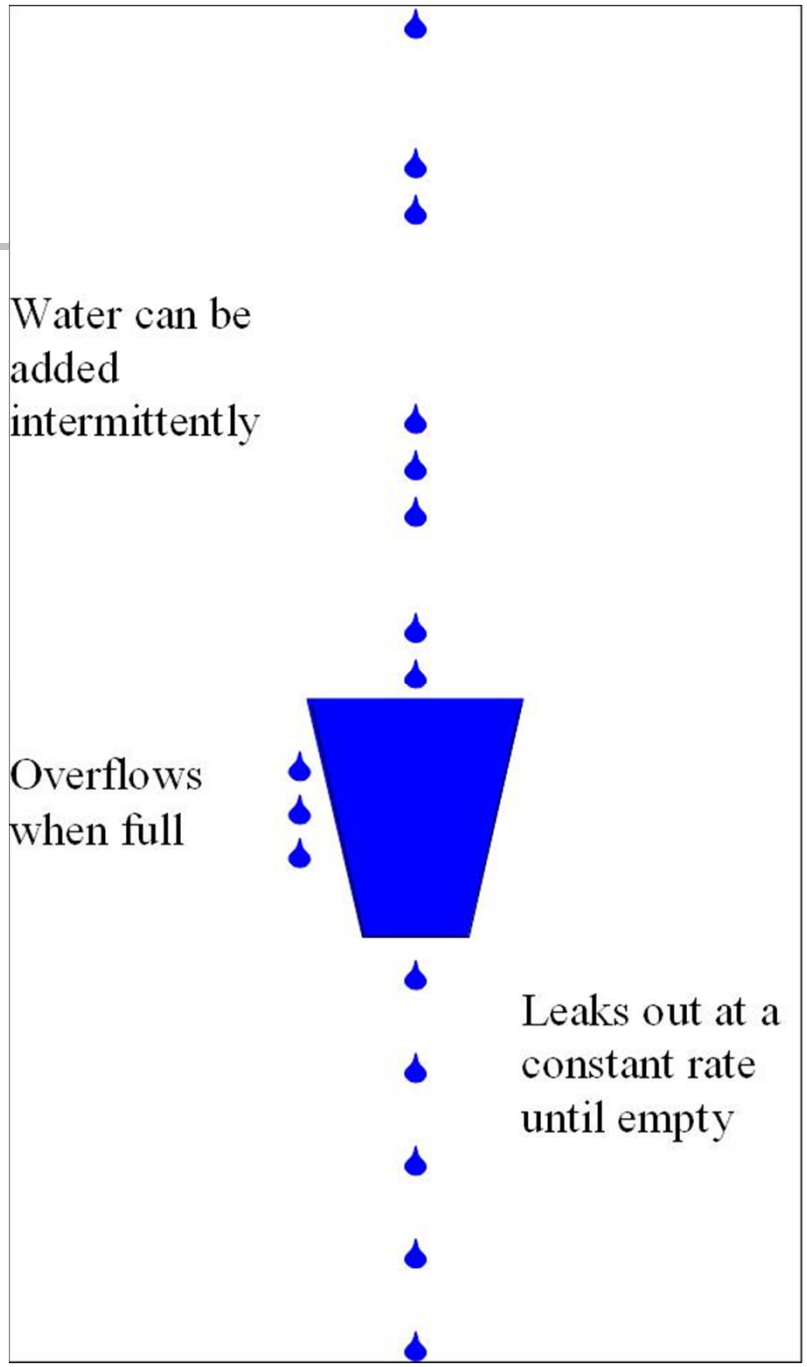
Implementation of QoS

- **Router ที่สามารถทำ QoS จะประกอบไปด้วย 4 Step**
  - 4. Traffic Scheduling
    - เป็นการเลือกว่าจะนำ Packet ของแต่ละ Queue ส่งไปอย่างไร Queue ใหนส่งก่อน และส่งได้ครั้งละกี่ Packet
    - Traffic Management มีหลายวิธี ที่สำคัญมีดังนี้
      - Leaky Bucket: ยอมให้แต่ละ Queue ส่งได้ในอัตราคงที่ โดยการนับและควบคุม Packet ที่ส่ง ถ้าส่งข้อมูลต่ำกว่าที่ยอมให้ในบางเวลา จะสามารถส่งข้อมูลเพิ่มได้บ้างในโอกาสต่อไป โดยใช้ Counter ในการควบคุมการส่ง
      - Token Bucket: ยอมให้ Queue ส่งข้อมูลได้คงที่เช่นกัน แต่จะนับเป็น Byte ที่ส่ง สามารถยอมให้ส่งข้อมูลแบบ Burst ได้บ้าง แต่ค่าเฉลี่ยต้องไม่เกินค่าที่กำหนด โดยมีการกำหนด Token ให้กับผู้ส่งเป็นระยะ

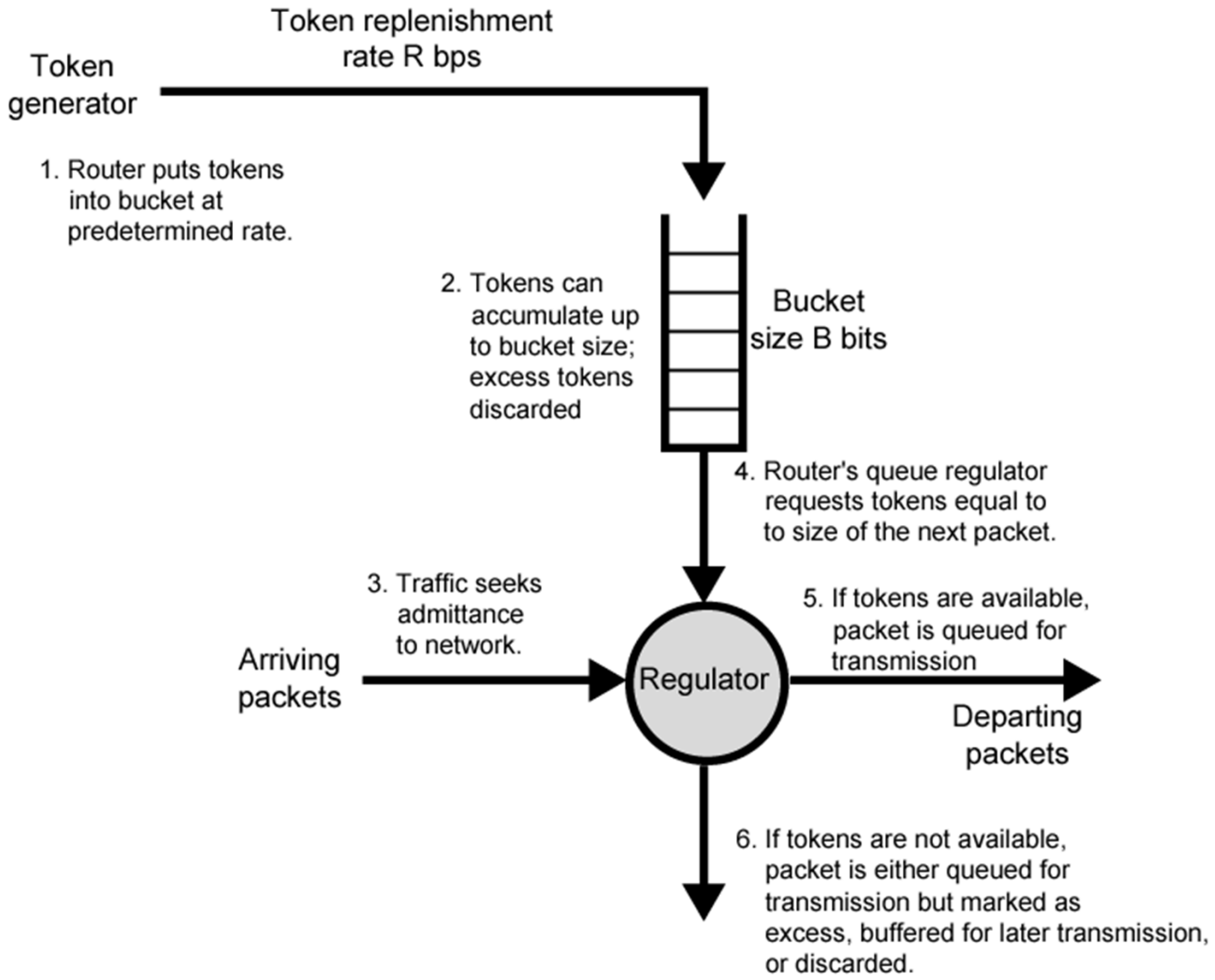
Implementation of QoS

- **ต่อ Traffic Management**
  - Traffic Management มีหลายวิธี ที่สำคัญมีดังนี้
    - Leaky Bucket
    - Token Bucket
    - WRR-Weight Round Robin: กำหนดจำนวน Packet ที่ส่งในแต่ละ Queue ในแต่ละครั้ง โดยการกำหนดน้ำหนักให้กับแต่ละ Queue และทำ Round Robin วน
    - DRR-Deficit Round Robin: กำหนดจำนวน Byte ที่แต่ละ Queue จะส่งได้ในแต่ละครั้ง และทำ Round Robin

Leaky Bucket



Token Bucket



Internet QoS Technologies

- **RSVP และ COPS**
  - RSVP: Resource Reservation Protocol เป็น Fine-Grain โดยต้องกำหนดในแต่ละ Session ของ TCP/UDP ซึ่ง Application จะส่ง Request มาก่อน และ Request จะถูกส่งผ่าน Router ซึ่งจะมีการจอง Resource เอาไว้ จนกระทั่งถึงปลายทาง ถ้าทุกๆ Hop ตอบรับในการให้ Request และมีการจอง Resource จึงจะมีการกำหนด Flow ID และส่งข้อมูลได้
  - COPS: Common Open Policy Service เป็น Protocol ที่ใช้ร่วมกับ RSVP ที่จะควบคุม Policy
  - RSVP มักจะไม่ค่อยเห็นใช้งาน เพราะเป็น Fine-Grain ซึ่งมีการทำ QoS ในระดับ Flow

Internet QoS Technologies

- **DiffServ: Differentiated Service**
  - เป็น Coarse-Grain QoS โดยมีการกำหนดการแบ่ง Class ในส่วนของ Field 'Type-of-Service' ใน IPv4 และ 'Traffic Class' ใน IPv6
  - ผู้ใช้ส่งข้อมูลโดยกำหนด Class ที่ตัวเองต้องการในส่วนนี้
  - ชนิดของ Class จะเป็นตัวกำหนด Traffic Management ที่ตัว Router อีกที
  - ยังไม่เป็นที่แพร่หลายมากนัก

Internet QoS Technologies

- **MPLS: MultiProtocol Label Switching**
  - เป็น Mechanism แบบ Connection-Oriented ที่สร้างขึ้นมาสวมบน IP อีกทีหนึ่ง
  - ในการใช้งาน ผู้ดูแลจะกำหนดเส้นทางส่งข้อมูลผ่าน Router ที่ทำ MPLS ได้
  - จากนั้นตัว Datagram ที่ส่งจะถูกแปะด้วย MPLS Header และส่งไปตามเส้นทาง เมื่อถึงปลายทาง ส่วน MPLS Header จะถูกนำออก
  - ในแต่ละเส้นทางที่ส่ง จะมีการกำหนดค่า QoS Parameter ต่างกัน ดังนั้น Datagram จะใช้เส้นทางตามความเหมาะสมที่จัดตั้งโดย ISP และใช้ Label ที่เหมาะสม
  - MPLS Packet จะถูก Switch ในระดับ Layer 2 และจะเร็วกว่า
  - เป็นวิธีที่นิยมสำหรับ ISP ที่จะให้บริการแก่ลูกค้าในปัจจุบัน

BREAKS

- **After Break: Chapter 29**
  - IP Telephony

Chapter 29: Multimedia and IP Telephony (VoIP)

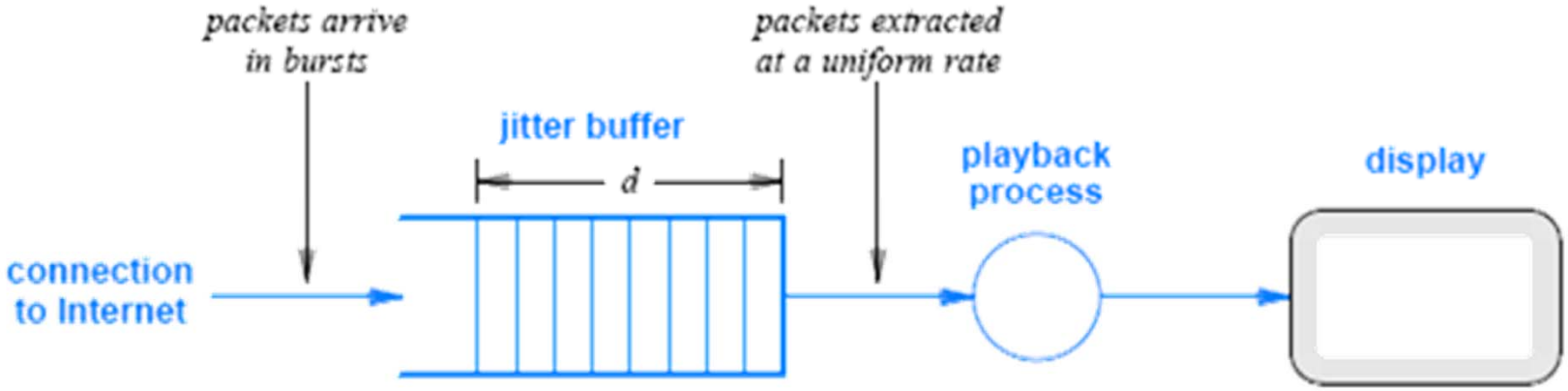
- **ในการส่งข้อมูลแบบ Real-time จะต้องคำนึงถึงค่า Jitter ใน Network**
  - อาจจะใช้ Isochronous Infrastructure
  - อาจจะใช้การทำงานของ Protocol
- **เราจะกล่าวถึงการส่งข้อมูลแบบ Multimedia ผ่าน Best-Effort Network ได้อย่างไร**
- **จากนั้นจะกล่าวถึง Technology ของ VoIP**
  - จะเน้นเฉพาะเรื่องของ SIP เพราะมีการใช้งานแพร่หลายมากกว่า

Real-Time Data Transmission and Best-Effort Delivery

- **Multimedia** หมายถึงข้อมูลที่ประกอบด้วยทั้ง **Voice** และ **Video**
- คำว่า **Real-Time Multimedia** หมายถึงข้อมูล **Multimedia** ที่จะต้องแสดงในอัตราที่เท่ากันกับอัตราของการส่งข้อมูล (หรืออัตราที่ข้อมูลถูกบันทึก)
- **Internet** เป็น **Best-Effort Delivery** ที่มีทั้ง **Lost, Delay, Out-of-Order** จะสามารถส่งข้อมูลพวกนี้ได้อย่างไร
  - การ Retransmission จะใช้ไม่ได้ เพราะมันจะไปถึงช้าเกิน
- ใน **Internet** จะกระทำการส่งโดยใช้ **Protocol Support**

Delayed Playback and Jitter Buffer

- **ในการจัดการกับ Jitter และเพื่อให้การแสดงผลราบเรียบ จะใช้สองเทคนิคดังนี้**
  - Timestamps: ผู้ส่งจะประทับเวลาสำหรับข้อมูลแต่ละชิ้นที่ส่ง ผู้รับจะใช้ค่า Timestamp นั้นจัดการกับเรื่อง Out-of-Order Packet และแสดงข้อมูลตามเวลาที่กำหนด
  - Jitter Buffer: ในการจัดการกับ Jitter จะนำข้อมูลที่ได้รับมาใส่ใน Buffer ก่อน และจะมีการหน่วงเวลาในการแสดงผล



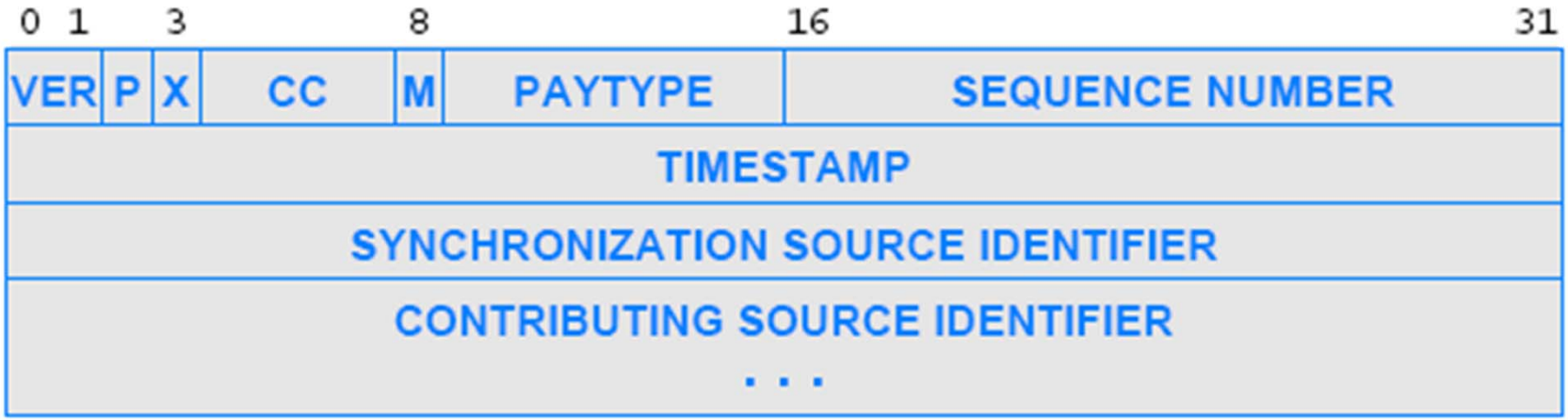
Real-Time Transport Protocol

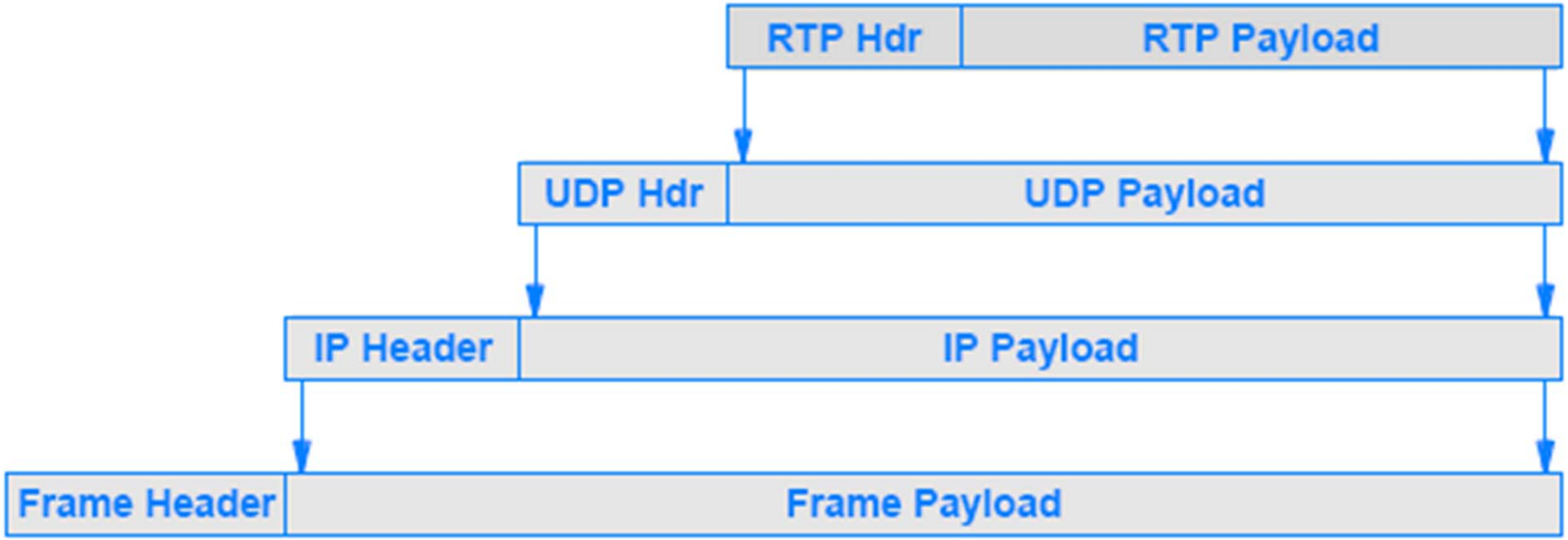
- **RTP เป็น Mechanism ที่ถูกใช้สำหรับส่ง Real-Time Data ผ่าน Internet**
  - จริงๆแล้วไม่ใช่ Transport Protocol เพราะมันจะวางอยู่บน Transport Protocol อีกทีหนึ่ง (ปกติจะเป็น UDP)
- **RTP ไม่ได้ Guarantee เรื่องการส่งข้อมูลที่เป็นไปตามเวลา และตัว Protocol ไม่มีการ Implement Jitter Buffer แต่มันช่วยให้ผู้รับสามารถสร้าง Jitter Buffer ได้ โดยให้ข้อมูลสามอย่าง**
  - Sequence Number: เพื่อตรวจจับ Packet ที่สูญหาย
  - Timestamp: เพื่อให้ผู้รับแสดงผลได้ตามเวลาที่ถูกต้อง
  - ชุดของ Source Identifier: บ่งบอก Source ของข้อมูลแก่ผู้รับ

Real-Time Transport Protocol

- **RTP Basic Header**
  - Ver ปัจจุบันคือ 2
  - P=มี Zero Padded in Payload หรือไม่, X=Extension Header Exist, CC = Source Count, M = Mark Bit สำหรับ Mark บาง Frame
  - PAYTYPE: Payload Type
  - Sequence ใช้ Random Number และจะ Increment ทีละหนึ่งในแต่ละ Packet
  - Timestamp จะเริ่มจาก Random Time เช่นกัน และจะไม่กำหนดหน่วยเวลา ขึ้นอยู่กับ Payload Type
  - Synchronization Source Identifier และ Contributing Source Identifier จะบ่งบอก Source ของข้อมูล
- **ปกติ RTP จะถูกบรรจุใน UDP และอาจจะถูกส่งแบบ Broadcast หรือ Multicast (หรือ Unicast)**

Real-Time Transport Protocol





IP Telephony

- IP Telephony หรือ Voice over IP (VoIP) เป็น Application ที่เกี่ยวข้องกับ Multimedia ที่สำคัญที่สุดตัวหนึ่ง
- ปัจจุบัน การสื่อสารโทรศัพท์เริ่มจะเปลี่ยนจากการใช้ Telephone Switch มาเป็น IP Router
- หลักการคือ เปลี่ยนเสียงพูดที่เป็น Analog ให้เป็น Digital จากนั้นส่ง Digital Stream ผ่าน IP Network และที่ปลายทางเปลี่ยนสัญญาณ Digital กลับเป็น Analog ตามเดิม

IP Telephony

- **รายละเอียดของ IP Telephony หรือ Voice over IP (VoIP) นั้นค่อนข้างจะซับซ้อน**
  - ปัญหา Delay และ Jitter ใน Network
  - การจัดการเกี่ยวกับ Call Setup
    - การแปลง Telephone Number เป็น IP Address
    - การค้นหาตำแหน่งของผู้รับ
    - สัญญาณควบคุมต่างๆ เช่นการเชื่อมต่อ การส่ง Ringing Signal การทำ Call Forwarding การบันทึกการใช้งาน หรือการจบการสื่อสาร
  - ส่วนที่ซับซ้อนที่สุดคือ ระบบ VoIP จะต้อง Backward Compatible กับระบบโทรศัพท์เดิม (Public Switching Telephone Network, PSTN)
    - โทรศัพท์ที่ต่อกับ IP ต้องสามารถโทรหรือรับสายจากโทรศัพท์ในเครือข่าย PSTN รวมถึงการรองรับ Feature อื่นๆ เช่น Call Forwarding, Call Waiting, Conference Call และ Caller ID
    - นอกจากนี้แล้ว องค์กรที่มีชุมสายเป็นของตัวเอง (PBX) อาจจะต้องการ IP Phone ที่ให้ Service เหมือนกับระบบที่ตนเองมีอยู่

Signaling and VoIP Signaling Standard

- **ITU (International Telecommunication Union) ผู้รับผิดชอบมาตรฐานโทรศัพท์ ได้ออกมาตรฐานของ IP Phone ที่สามารถใช้กับระบบโทรศัพท์ได้**
- **IETF (Internet Engineering Task Force) ผู้รับผิดชอบมาตรฐาน TCP/IP ได้ออกมาตรฐานออกมาเช่นกัน**
- **ทั้งสองมาตรฐานแตกต่างกัน แต่มีที่เหมือนกันคือ**
  - Audio จะใช้การ Encode ด้วย PCM (Pulse Coded Modulation)
  - RTP จะถูกใช้ในการส่ง Digitized Audio
- **ปัญหาที่ทำให้ VoIP ซับซ้อนคือเรื่องของการทำ Call Setup และ Call Management ที่เรียก Signaling**
  - Signaling มาตรฐานปัจจุบันคือ Signaling System 7 (SS7)
    - IETF ออก Session Initiation Protocol (SIP) และ Media Gateway Control Protocol (MGCP)
    - ITU ออก H.323 ออกมา
    - นอกจากนี้ยังมี H.248 (Megaco=Media Gateway Control Protocol) ซึ่งเป็น Protocol ร่วมของทั้งสองกลุ่ม

ส่วนประกอบของ IP Telephone System

- **IP Telephone**
  - ทำงานเหมือนกับโทรศัพท์ทั่วไป แต่เชื่อมต่อกับ Internet และส่ง Digitized Voice
- **Media Gateway Controller (Gatekeeper, Softswitch)**
  - ทำการควบคุมและประสานงานระหว่าง IP Telephone สำหรับ Service ต่างๆเช่น Call Setup, Call Termination, Call Forwarding และการหาตำแหน่ง
  - ควบคุมการทำงานของ Media Gateway และ Signaling Gateway
- **Media Gateway**
  - ทำหน้าที่เชื่อมต่อระหว่างสอง Network ที่ใช้การ Encoding ที่ไม่เหมือนกัน โดยการทำ Translation Audio Encoding ระหว่าง NW
- **Signaling Gateway**
  - ดูแลการเชื่อมต่อระหว่างสอง Network ที่ใช้ Signaling ต่างกัน และทำการ Translate Call Management

ส่วนประกอบของ IP Telephone System

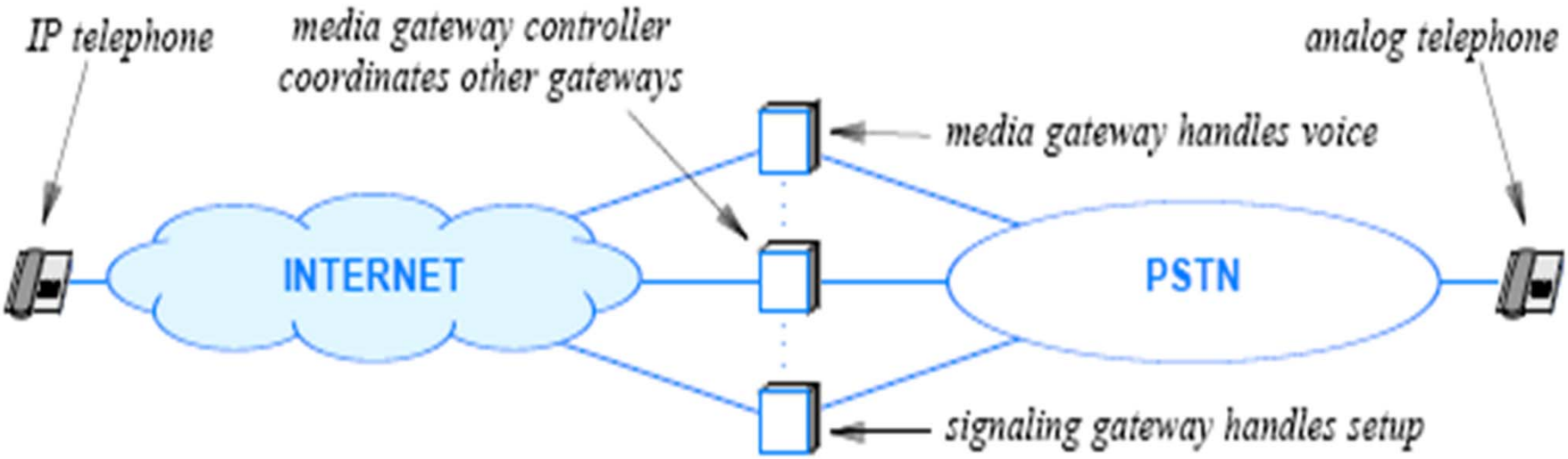


Figure 29.5 Connections among IP telephone components.

H.323 Terminology and Concept

- **มาตรฐานของ H.323 กว้างมาก ประกอบเป็นชุดของ Protocol ที่สามารถรองรับได้ทั้ง Voice และ Video**
  - Terminal
    - H.323 Terminal จะให้ Function ของ IP Telephone ซึ่งอาจจะรวมถึงอุปกรณ์ที่ใช้ในการส่ง Video
  - Gatekeeper
    - H.323 Gatekeeper จะทำหน้าที่หาตำแหน่ง และทำ Signaling และประสานงานกับ Gateway ที่เชื่อมต่อกับ PSTN
  - Gateway
    - H.323 ใช้เพียง Gateway เดียวซึ่งจะดูแลทั้ง Signaling และ Media Translation
  - Multipoint Control Unit (MCU)
    - ให้ Service เช่นใน Multipoint Conferencing (Video/Audio)

H.323 Layering

<table>
<tr><th>Layer</th><th>Signaling</th><th>Registration</th><th>Audio</th><th>Video</th><th>Data</th><th>Security</th></tr>
<tr><td rowspan="2">5</td><td rowspan="2">H.225.0-Q.931
H.250-Annex G
H.245
H.250</td><td rowspan="2">H.225.9-RAS</td><td>G.711
H.263
G.722
G.723
G.728</td><td>H.261
H.323</td><td rowspan="2">T.120</td><td rowspan="2">H.235</td></tr>
<tr><td colspan="2">RTP, RTCP</td></tr>
<tr><td>4</td><td>TCP, UDP</td><td colspan="2">UDP</td><td>TCP</td><td>TCP, UDP</td></tr>
<tr><td>3</td><td colspan="5">IP, RSVP, and IGMP</td></tr>
</table>

SIP Terminology and Concept

- **หลักการของ SIP คือพยายามใช้ Protocol ที่มีอยู่แล้วให้มากที่สุด เช่น การใช้ DNS ในการ MAP ระหว่างหมายเลขโทรศัพท์และ IP Address**
- **SIP กำหนด Element สามส่วน**
  - User Agent
    - หมายถึงอุปกรณ์ที่ Initiate หรือ Terminate Phone Call อาจจะเป็น IP Phone, Laptop, Computer หรือ PSTN Gateway
    - User Agent ประกอบด้วยสองส่วน
      - User Agent Client คือผู้ทำหน้าที่โทรออก
      - User Agent Server จะทำหน้าที่รับสายเข้า

SIP Terminology and Concept

- **SIP กำหนด Element ใน VoIP แบ่งเป็นสามส่วน**
  - User Agent
  - Location Server
    - ทำหน้าที่ในการจัดการฐานข้อมูลเกี่ยวกับผู้ใช้ เช่น IP Address, Service ต่างๆที่ผู้ใช้สมัคร โดยข้อมูลนี้จะถูกส่งเมื่อทำ Call Setup
  - Support Servers
    - Proxy Server; เป็น Proxy สำหรับการส่ง Request ของ User ไปยัง Location อื่น รวมถึง Optimum Routing ไปยัง Location นั้นๆ และกำดูแลเรื่อง Policy
    - Redirect Server; ดูแลจัดการเรื่องการทำ Call Forwarding และหมายเลข 1-800 โดยมันจะส่งค่า Alternate Location กลับไปให้ User
    - Registrar Server; ใช้ดูแลจัดการเรื่องการลงทะเบียนของผู้ใช้ รวมถึงการทำ Authentication และ Update ฐานข้อมูลของ Location Server

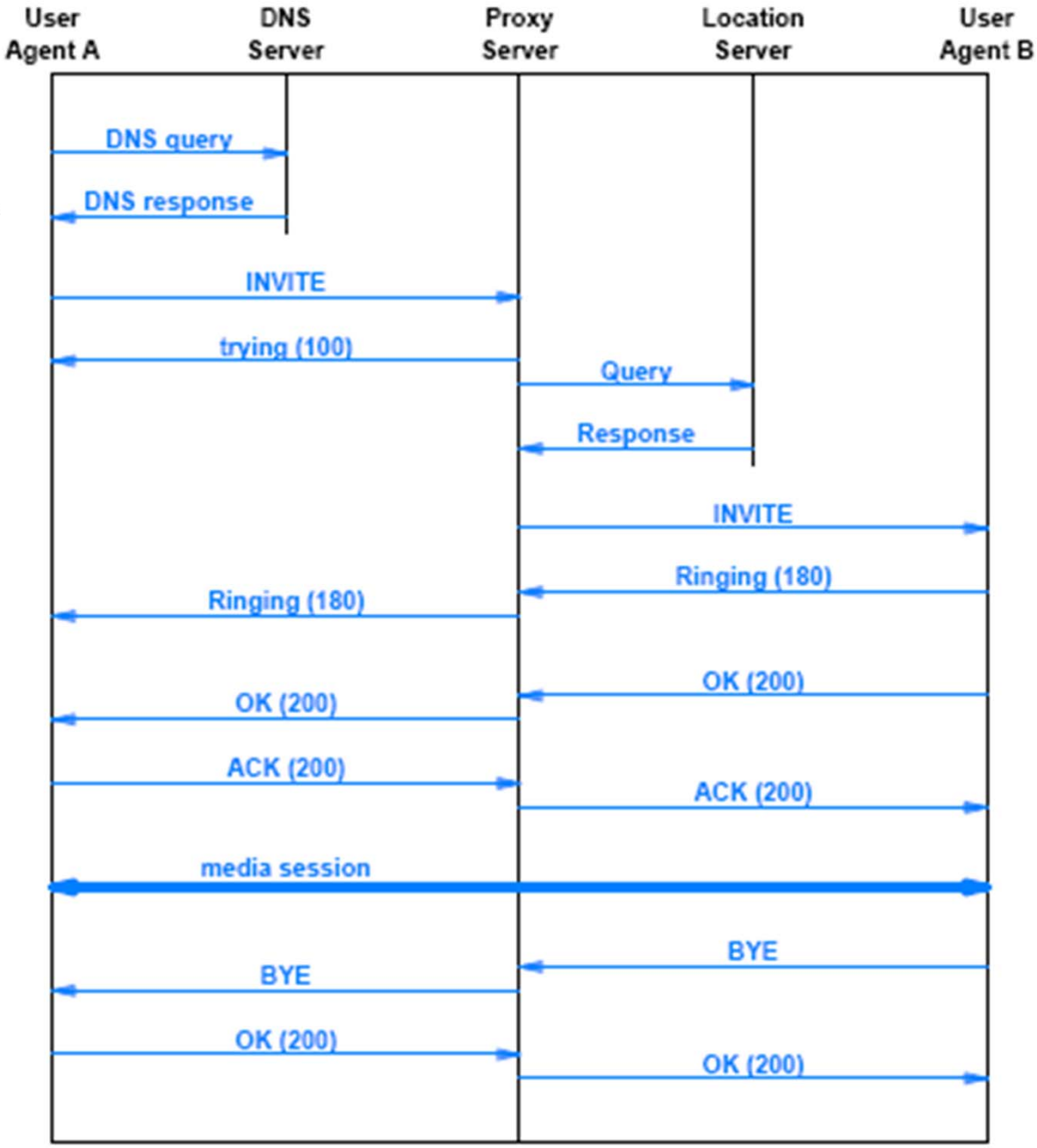
SIP Characteristics and Methods

- **คุณสมบัติของ SIP ประกอบด้วย**
  - จะ Run ในระดับ Application Layer
  - รวมหน้าที่ของ Signaling ต่างๆเข้าไว้ด้วยกัน
  - ให้บริการเสริมอื่นๆเช่น Call Forwarding
  - ใช้วิธีของ Multicasting ในการทำ Conference Call
  - ยอมให้ผู้ใช้ต้นทางและปลายทางสามารถ Negotiate และเลือก Parameter ในการเชื่อมต่อที่เหมาะสม
- **แต่ละ User จะอ้างถึงด้วย SIP URI (Uniform Resource Identification) ประกอบด้วยชื่อ และ Domain Name**
  - เช่น sip:smith@somecompany.com

SIP Basic Message Types

- **ประกอบด้วย 6 Basic Message Types**
  - Basic Message Type นี้เรียก Method
  - Invite: ใช้ในการสร้าง Session โดยทำส่ง Invite กับ End Point ให้เข้าร่วมใน Session
  - ACK: Acknowledge Response สำหรับ Invite
  - BYE: จบ Session และจบ Call
  - CANCEL: ในทำการ Cancel Request ที่ยังค้างอยู่
  - REGISTER: ใช้สำหรับการขึ้นทะเบียนตำแหน่งของผู้ใช้
  - OPTIONS: ใช้ร้องขอข้อมูลของอีกฝั่งหนึ่งว่ามีความสามารถอะไรบ้าง

ตัวอย่าง ของ SIP Session



Telephone Number Mapping and Routing

- เราจะกำหนดชื่อ และหาตำแหน่งIP Phone แต่ละเครื่องได้อย่างไร
- Telephone Number ใน PSTN จะใช้มาตรฐานของ E.164
- SIP จะใช้ IP Address
- IETF เสนอสอง Protocol ในการ Mapping
  - ENUM (E.164 NUMbers) ใช้เพื่อการ Convert หมายเลขโทรศัพท์ให้อยู่ในรูป URI
  - TRIP (Telephone Routing Over IP) เป็น Protocol ที่ใช้หาตำแหน่ง User ใน Network รวม (PSTN+IP)

Telephone Number Mapping and Routing

- **IETF เสนอสอง Protocol ในการ Mapping**
  - ENUM (E.164 NUMbers) โดยทำการ Convert หมายเลขโทรศัพท์ให้อยู่ในรูป URI
    - ENUM จะใช้ Domain Name System ในการเก็บ Mapping โดยใช้ Special Domain 'e164.arpa'
    - Conversion จะมองหมายเลขโทรศัพท์ว่าเป็น String จากนั้นจะทำการ Reverse String
      - เช่นหมายเลข 1-800-555-1234 จะมี URI เป็น
      - 4.3.2.1.5.5.5.0.0.8.1.e164.arpa
    - การ Map อาจจะเป็น 1-to-1 หรือ 1-to-Many
  - TRIP (Telephone Routing Over IP) เป็น Protocol ที่ใช้หาตำแหน่ง User ใน Network รวม (PSTN+IP)
    - ใช้สำหรับ Location Server หรืออุปกรณ์อื่นทำการ Advertise Route ที่ตัวเองรู้ออกไป
    - TRIP ใช้ Concept ของการแบ่งกลุ่มผู้ใช้ทั้งหมดออกเป็น ITAD (IP Telephone Administration Domain)

End of Chapter 28-29 (Week 15)

- **HW 9 Download**

CPE 426 Computer Networks

Chapter 12:
PART IV in Textbook
Text Chapter 30: NW Security
Text Chapter 31: NW Management

TOPICS

- Chapter 30 (Only Selected Topics)
  - Criminal Exploits and Attack
  - Security Policy
  - Security Technologies
  - Hashing and MAC
  - Access Control
  - Encryption
  - Authentication and Digital Signature
  - Key Authorities and Digital Certificates
  - Firewall
  - Intrusion Detection System
  - Deep Packet Inspection
  - VPN
  - Securities Technologies
- Chapter 31 NW Management
  - Intranet Management
  - FCAPS
  - NW Management Tools
  - SNMP
  - MIB

Ch.30; Major Internet Security Problems

- **Phishing**
  - เป็นการปลอมตัวเป็น Site ที่รู้จักกันดี เช่น Web ของธนาคาร เพื่อหลอกให้ผู้ใช้เข้าไปป้อนข้อมูล และทำการโขมยข้อมูลไป
- **Misrepresentation**
  - เป็นการเสนอสินค้าหรือบริการที่เกินความเป็นจริง หรือสินค้าที่มีมาตรฐานต่ำหรือสินค้าปลอม
- **Scams**
  - เป็นการหลอกลวงในรูปแบบต่างๆ เพื่อที่จะให้ผู้ใช้ที่ไม่รู้ เข้าไปลงทุน หรือกระทำความผิด
- **Denial of Service (DOS)**
  - เป็นการกีดกัน Internet บาง Site เพื่อไม่ให้ผู้ใช้เข้าไปใช้งานได้หรือใช้ได้อย่างสะดวก
- **Loss of Control**
  - เป็นการที่ผู้บุกรุกได้เข้าไปควบคุมระบบคอมพิวเตอร์ และใช้ระบบนั้นในการกระทำความผิด
- **Loss of Data**
  - เป็นการโขมยข้อมูลที่สำคัญหรือเป็นความลับขององค์กรอกไปภายนอก

Ch.30; Major Internet Security Problems

Problem	Description
Phishing	Masquerading as a well-known site such as a bank to obtain a user's personal information, typically an account number and access code
Misrepresentation	Making false or exaggerated claims about goods or services, or delivering fake or inferior products
Scams	Various forms of trickery intended to deceive naive users into investing money or abetting a crime
Denial of Service	Intentionally blocking a particular Internet site to prevent or hinder business activities and commerce
Loss of Control	An intruder gains control of a computer system and uses the system to perpetrate a crime
Loss of Data	Loss of intellectual property or other valuable proprietary business information

Ch.30; Techniques Used in Security Attacks

- **Wiretapping**
  - เป็นการดักและทำการ Copy Packet ข้อมูล ที่วิ่งอยู่ใน Network
  - จุดประสงค์ของ Wire Tapping เพื่อนำมาใช้ใน Replay Attack
  - สามารถทำได้ง่ายในกรณีของ Wireless LAN
- **Replay**
  - เป็นการส่ง Packet ที่ดักจับได้ก่อนหน้านี้ ซึ่งมีข้อมูลของ Password จากการ Login ครั้งก่อน นำไปใช้ในการ Login เข้าระบบ
- **Buffer Overflow**
  - เป็นการส่งข้อมูลจำนวนมากๆไปยังผู้รับ เพื่อที่จะให้ข้อมูลได้ถูกเก็บในพื้นที่นอกเหนือจาก Buffer เนื่องจากเกิด Buffer Overflow
- **Address Spoofing**
  - ทำการปลอม Source IP Address เพื่อหลอกให้มีการ Process Packet
  - IP และ MAC Address สามารถฟังได้จาก ARP Broadcast
  - อาจจะใช้วิธีการ Broadcast ARP ปลอม เพื่อให้ส่งข้อมูลให้ตนเอง
  - อาจจะปลอม Routing Protocol เพื่อส่งข้อมูลให้ตนเอง
  - หรือส่ง DNS Message ปลอม
- **Name Spoofing**
  - ใช้ชื่อที่เป็นที่รู้จักแต่เขียนให้ผิดเพียน หรือทำการเปลี่ยนแปลงข้อมูลใน Name Server เพื่อให้มีการเข้าใจผิด และส่ง IP ผิด

Ch.30; Techniques Used in Security Attacks

- **DoS and DDoS**
  - โจมตีโดยการป้อน Packet จำนวนมากๆ เพื่อไม่ให้ Site สามารถทำงานอย่างอื่นได้
  - DDOS จะ Attack ประสานกันโดยใช้หลายเครื่อง
- **SYN Flood**
  - ส่ง Random TCP SYN Segment เพื่อให้ผู้รับไม่มี TCP Connection เหลืออยู่
- **Key Breaking**
  - เป็นการคาดเดา Key หรือ Password แบบอัตโนมัติ เพื่อที่จะสามารถเข้าถึงระบบได้
- **Port Scanning**
  - พยายามที่จะเชื่อมต่อ โดยใช้ Protocol Port แบบต่างๆเพื่อที่จะหาจุดอ่อน
- **Packet Interception**
  - เป็นการดักจับ Packet ระหว่างทาง จากนั้นแทนที่ด้วย Packet ที่ต้องการ หรือเพื่อทำ Man-in-the-Middle Attack

Ch.30; Techniques Used in Security Attacks

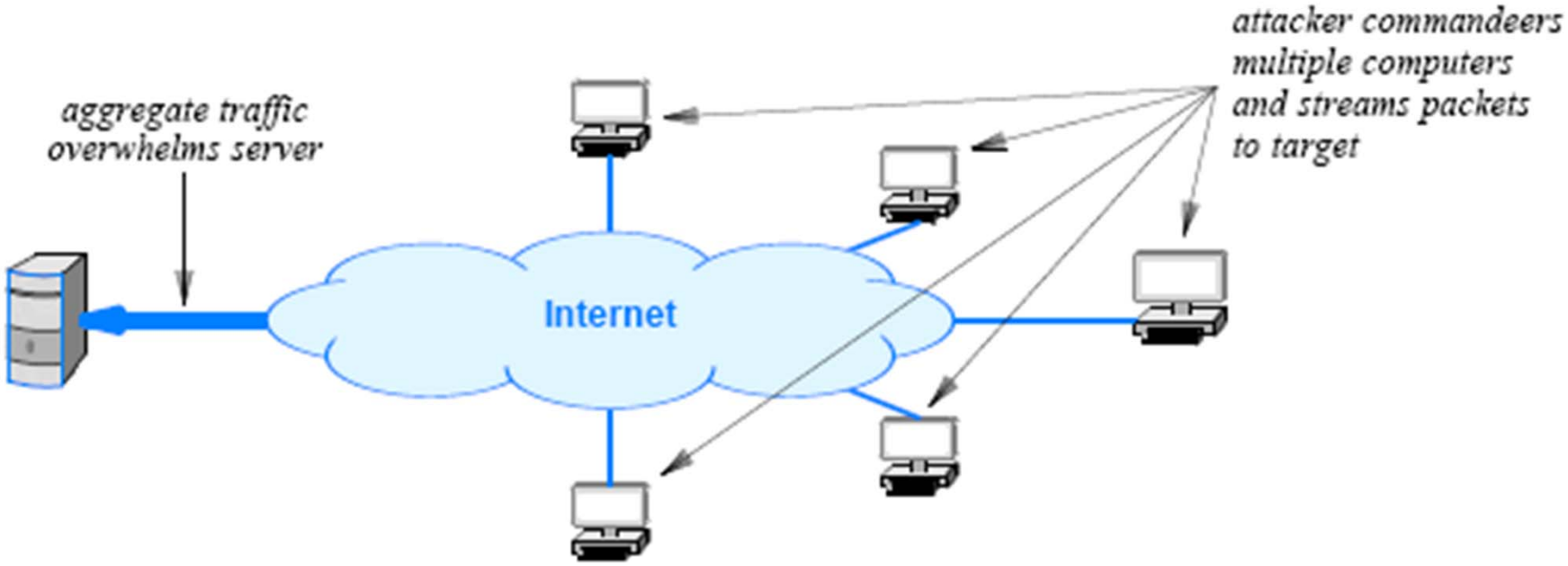


Figure 30.3 Illustration of a Distributed Denial of Service attack.

Ch.30; Techniques Used in Security Attacks

Figure 30.4 A man-in-the middle configuration and the attacks it permits.

Man-in-the middle Attack เป็นการ Intercept Packet จากนั้นส่ง Packet ที่ตัวเองต้องการออกไป อาจจะหลอกว่าเป็น Server ที่ Client ต้องการเชื่อมต่อ หรืออาจจะหลอกว่าเป็น Client เพื่อเชื่อมต่อกับ Server

Ch.30; Techniques Used in Security Attacks

Technique	Description
Wiretapping	Making a copy of packets as they traverse a network to obtain information
Replay	Sending packets captured from a previous session (e.g., a password packet from a previous login)
Buffer overflow	Sending more data than a receiver expects in order to store values in variables beyond the buffer
Address Spoofing	Faking the IP source address in a packet to trick a receiver into processing the packet
Name Spoofing	Using a misspelling of a well-known name or poisoning a name server with an incorrect binding
DoS and DDoS	Flooding a site with packets to prevent the site from successfully conducting normal business
SYN flood	Sending a stream of random TCP SYN segments to exhaust a receiver's set of TCP connections
Key Breaking	Automatically guessing a decryption key or a password to gain unauthorized access to data
Port Scanning	Attempting to connect to each possible protocol port on a host to find a vulnerability
Packet interception	Removing a packet from the Internet which allows substitution and man-in-the middle attacks

Security Policy

- **การกำหนด Security Policy ปกติจะสลับซับซ้อน**
  - เพราะจะต้องเกี่ยวข้องกับพฤติกรมของมนุษย์ และอุปกรณ์คอมพิวเตอร์และNetwork
  - องค์กรจะต้องตัดสินใจว่าต้องการ Security ในระดับใด และส่วนใหนของระบบเป็นส่วนที่สำคัญที่สุด และรองลงมา
  - อาจจะต้องมีการทำ Risk Analysis เพื่อหาอัตราส่วนระหว่างความสูญเสียและการลงทุน
- **ปกติการกำหนด Policy จะพิจารณาจาก**
  - Data Integrity: การป้องกันการเปลี่ยนแปลงข้อมูล
  - Data Availability: การป้องกันเพื่อให้ข้อมูลสามารถนำมาใช้งานได้
  - Data Confidentiality: การป้องกันความลับของข้อมูล
  - Privacy: การปกป้องผู้ส่ง หรือเจ้าของข้อมูล

Security Technologies

Technique	Purpose
Hashing	Data integrity
Encryption	Privacy
Digital Signatures	Message authentication
Digital Certificates	Sender authentication
Firewalls	Site integrity
Intrusion Detection Systems	Site integrity
Deep Packet Inspection & Content Scanning	Site integrity
Virtual Private Networks (VPNs)	Data privacy

Access Control and Password

- **เป็นการควบคุมผู้ใช้ หรือ Application Program ในการเข้าถึงข้อมูลและ Resource**
  - อาจจะอยู่ในรูป Password
  - หรืออาจจะใช้ Access Control List (ACL)
- **ในการขยายการทำ Access Control ผ่าน Network จะยุ่งยากกว่าระบบที่เป็น Standalone**
  - Wiretapping, Replay, Phishing, Spoofing etc.

AAA Protocol

- **เป็น Computer Security Protocol สำหรับ Authentication ผู้ใช้งานใน Network ที่จะกระทำ 3 อย่าง**
  - Authentication
    - คือขบวนการที่จะตรวจสอบเอกลักษณ์ของผู้ใช้ ว่าเป็นผู้ที่อ้างถึงนั้นจริงๆ เช่นการใช้ Password, Keycard, Biometrics
  - Authorization
    - เป็นการกำหนดสิทธิ์ให้กับแต่ละบุคคล ว่าจะสามารถทำอะไรได้บ้าง
  - Accounting
    - คือการบันทึกการใช้งานของแต่ละบุคคล

Hash and Message Authentication Code

- **Hash เป็นการสร้างตัวแทนของเอกสาร**
  - ได้จากการคำนวณค่า Hash ซึ่งจะมีจำนวน Bit ที่คงที่จากเอกสารที่มีความยาวไม่แน่นอน
  - Algorithm จะเป็น One-Way Function และมีการ Collision ต่ำ
  - สามารถใช้เป็นตัวตรวจสอบว่าเอกสารมีการแก้ไขหรือไม่
- **MAC ได้จากการนำ Data มารวมกับ Key และทำการสร้าง Hash**
  - จุดประสงค์เหมือนกับ Hash คือเพื่อจะใช้ตรวจสอบว่าเอกสารมีการแก้ไขหรือไม่
  - แต่ในการนี้ คนที่จะตรวจสอบได้จะต้องมี Key
- **HASH และ MAC ใช้ในการทำ Message Authentication**
  - MD5, SHA-1,

Encryption

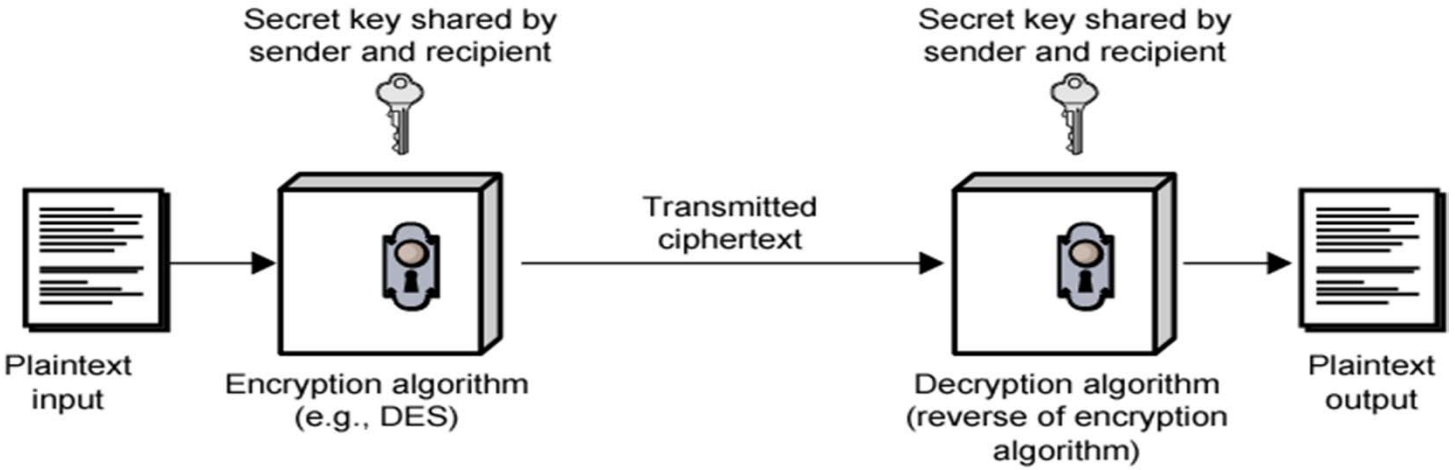
- ปกติ **Authentication** เป็นการพิสูจน์ตัวตน ใช้ป้องกันเรื่อง **Integrity**
- ในการป้องกันเรื่อง **Confidentiality** จะต้องใช้วิธีการ **Encryption**
- **Encryption** คือขบวนการเปลี่ยนแปลงเอกสารให้อยู่ในรูปที่ไม่สามารถอ่านได้
  - จะใช้ Encryption Algorithm และ Encryption Key
  - ขบวนการจะต้อง Inverse ได้
    - คือทำ Decryption ได้ โดยมี Decryption Algorithm ร่วมกับ Decryption Key

Encryption Terminology

- **Plaintext, M**
  - หมายถึงเอกสารที่ยังไม่ได้ถูกเข้ารหัส
- **Cyphertext, C**
  - เอกสารที่ถูกเข้ารหัสแล้ว
- **Encryption Key, K_1**
  - Bit String ที่ใช้สำหรับการเข้ารหัส
- **Decryption Key, K_2**
  - Bit String ที่ใช้สำหรับถอดรหัส
- **$C = encrypt(K_1, M)$**
- **$M = decrypt(K_2, C)$**
- **$M = decrypt(K_2, encrypt(K_1, M))$**

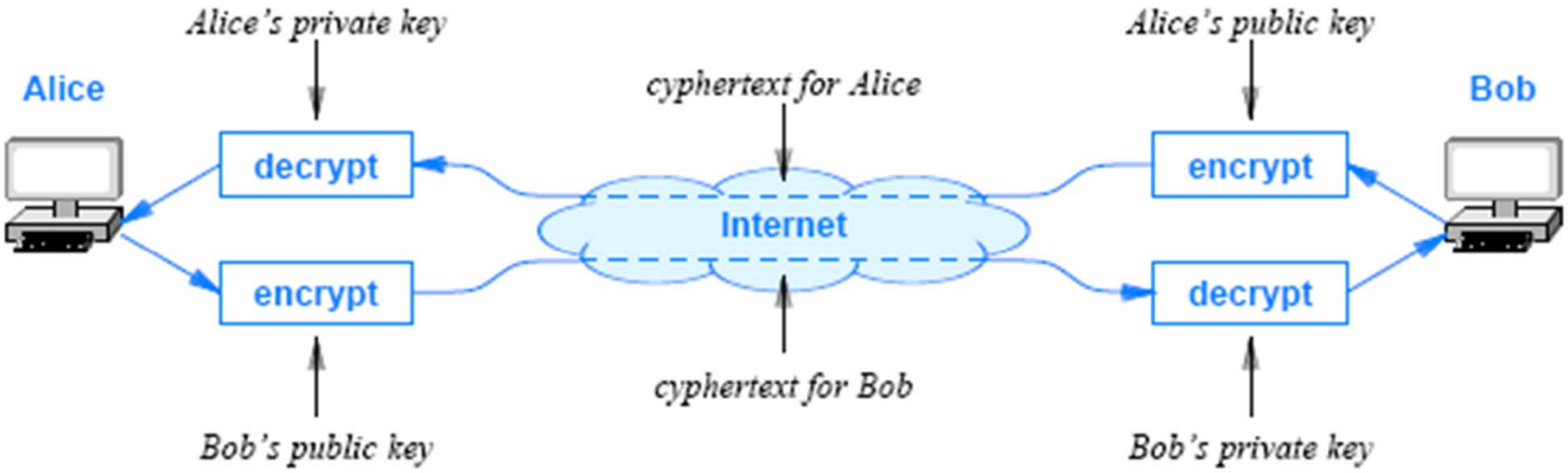
Private Key Encryption

- **Symmetric Cryptography**
- **Conventional Cryptography**
- **เป็นหนึ่งในสองเทคโนโลยีที่ใช้ในการเข้ารหัส**
- **กรณีนี้ Key ที่ใช้เข้ารหัสและถอดรหัสจะเป็นตัวเดียวกัน**
  - ผู้ส่งและผู้รับจะต้อง Share Key ร่วมกัน
  - DES, 3DES, AES



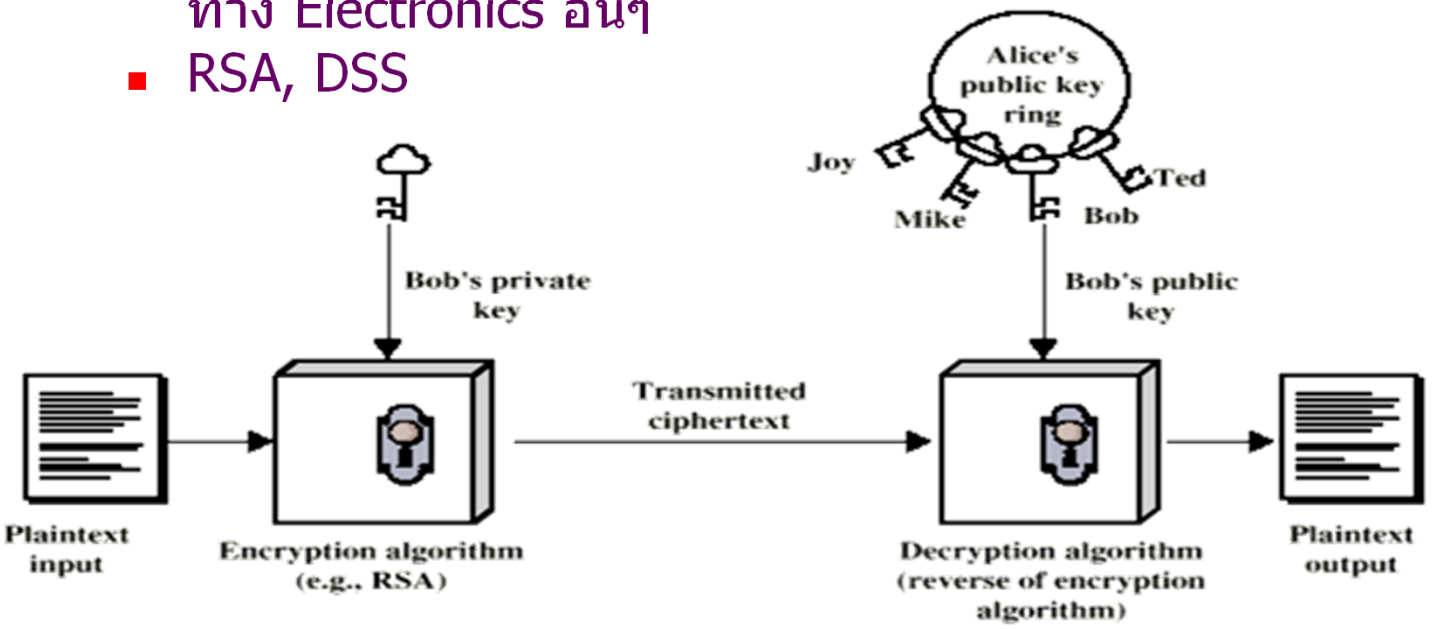
Public Key Encryption

- **Asymmetric Key Cryptography**
- **Key ที่ใช้เข้ารหัสเรียก Public Key สามารถแจกจ่ายได้**
- **Key ที่ใช้ถอดรหัส ต้องเป็นความลับ เรียก Private Key**
  - RSA, Diffie-Hellman, ECC
- **สามารถดัดแปลงมาใช้ทำ Digital Signature ได้**



Authentication With Digital Signature

- **ถ้าเรากลับ โดยเข้ารหัสด้วย Private Key แทนที่จะเป็น Public Key**
  - ดังนั้นจะมีเพียงเจ้าของ Key เท่านั้นที่สามารถสร้างเอกสารที่เป็น Ciphertext นี้ได้
  - เหมือนกับเป็นการเซ็นต์เอกสาร เพราะ Ciphertext ที่ได้เฉพาะเจาะจงว่าเป็นของผู้ใด
  - เรียกว่าเป็นการทำ Digital Signature
  - สามารถนำมาใช้ในการทำ Authentication รวมถึงการทำธุรกรรมทาง Electronics อื่นๆ
  - RSA, DSS



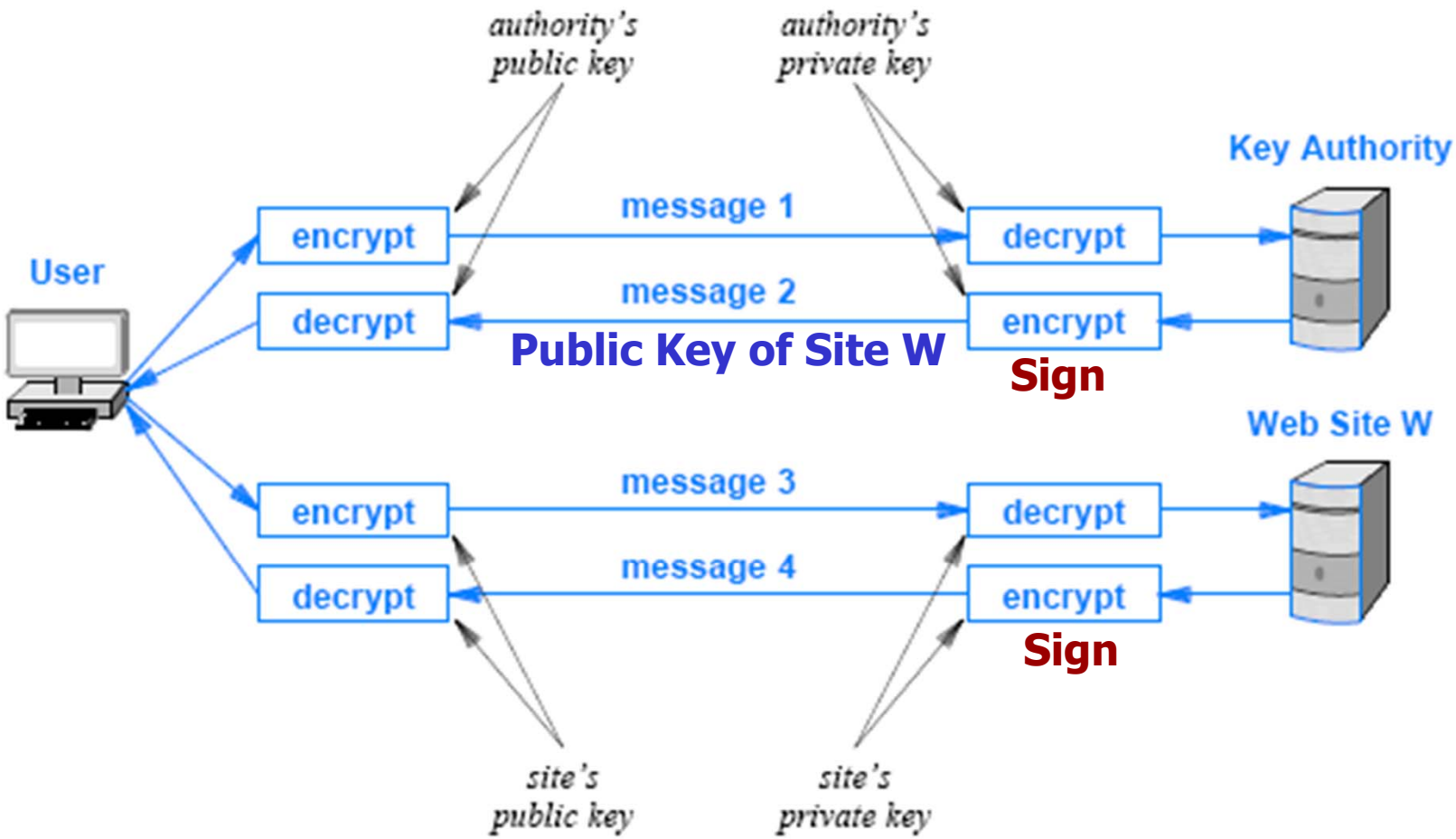
Key Authorities and Digital Certificates

- **ปัญหาของ Public Key คือการสร้างเอกสารส่งให้ใคร เราจะต้องได้ Public Key ของคนๆนั้น**
  - Public Key ไม่จำเป็นต้องเป็นความลับ สามารถส่งผ่านช่องทางการสื่อสารปกติ หรือประกาศออกมาได้
  - ปัญหาคือเรื่องของการ Authentication Public Key
    - เราจะรู้ได้อย่างไรว่า Public Key ที่ได้รับมานั้นเป็นของคนนั้นจริงๆ
    - Alice ต้องการส่งข้อมูลให้ Bob แบบปลอดภัย ต้องการ Public Key ของ Bob ซึ่ง Bob จะต้องส่งมาให้ แม้ว่าการส่งสามารถทำผ่านช่องทางปกติได้ และ Public Key ไม่ต้องเป็นความลับ แต่สามารถปลอมได้
      - ถ้าผู้ไม่หวังดี ส่ง Public Key ของตัวเองมา และหลอกว่าเป็น Public Key ของ Bob
      - เมื่อเราเข้ารหัสด้วย Key ปลอมนั้นส่งให้ Bob ตัว Bob ไม่สามารถนำ Private Key ของตัวเองมาถอดรหัสได้
      - ที่ร้ายกว่านั้นคือ ถ้าผู้ไม่หวังดีนั้น แอบดักฟังและ Copy (Wiretapped) เอกสาร เขาสามารถใช้ Private Key ที่เป็นคู่ของ Key ปลอม มาถอดรหัสและดูเอกสารได้
      - นี่คือ Man-in-the-middle attack
- **ปัญหานี้เราเรียก 'Key Distribution Problem'**
  - เราต้องการการ Authentication ของ Public Key

Key Authorities and Digital Certificates

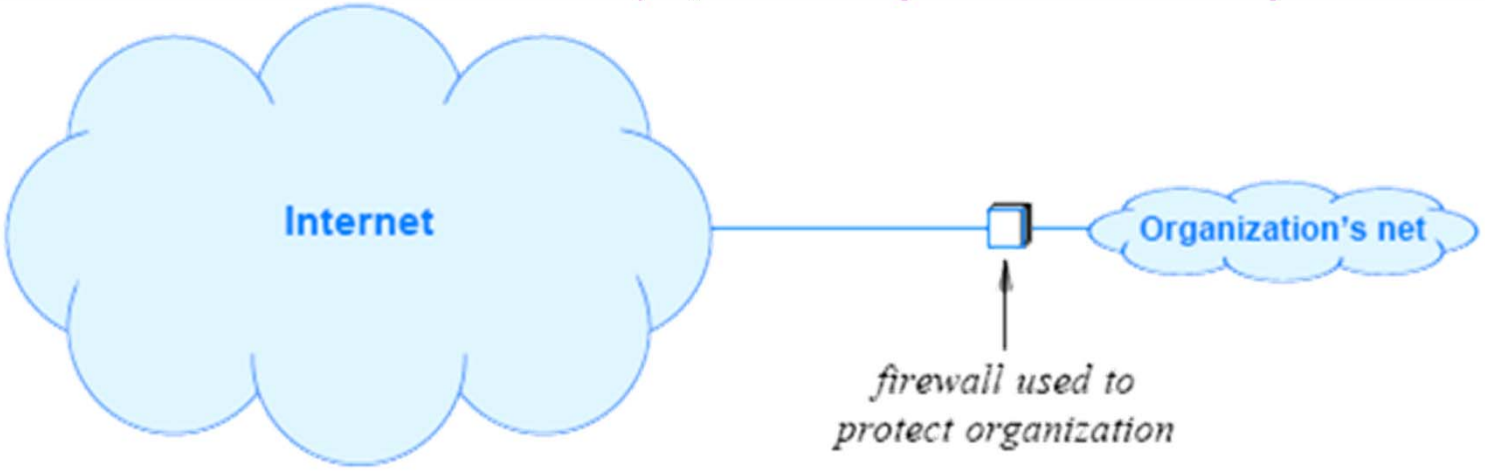
- **เราสามารถตั้ง Server กลาง ทำหน้าที่แจกจ่าย Key แบบอัตโนมัติ (และมีระบบรักษาความปลอดภัย)**
  - เรียก Key Distribution Center หรือ Key Authority
  - ทุกคนจะนำ Public Key ของตนเองไปฝากไว้
  - เมื่อต้องการส่งเอกสารให้ใคร ให้ขอ Public Key ของคนนั้นผ่าน Server นี้
- **ข้อเสียของ Key Authority คือ Server จะ Down ไม่ได้**
- **อีกวิธีหนึ่งคือออกเป็น Digital Certificate**
  - เป็นการนำ Public Key ของตนเอง ทำการรับรองโดยใช้ Digital Signature จาก Certification Authority
  - Certificate ไม่สามารถปลอมได้ สามารถตรวจสอบได้
  - User จะเก็บ Certificate ของตัวเองไว้ ถ้ามีใครต้องการส่งเอกสารให้ตน User จะทำการส่ง Certificate ให้กับคนนั้น
  - ดังนั้นการสื่อสารไม่ต้องใช้ Key Authority ที่ต้อง Online

Key Authorities and Digital Certificates



Firewalls

- เทคโนโลยีที่กล่าวข้างต้นใช้ในการปกป้องข้อมูลเป็นหลัก
- ในการปกป้อง Computer และ Network จาก Traffic ที่ไม่ต้องการ เราต้องใช้อีกเทคโนโลยีหนึ่ง คือ '**Firewall**'
- จุดประสงค์ของ Firewall คือเป็นตัวกั้น Traffic ที่ไม่ต้องการ ให้ผ่าน Network
  - ปกติจะวางกั้นระหว่างตำแหน่ง Network ขององค์กร และเส้นทางที่จะต่อออก Internet ทุกๆทางออก(ถ้ามีมากกว่าหนึ่ง)



Firewall

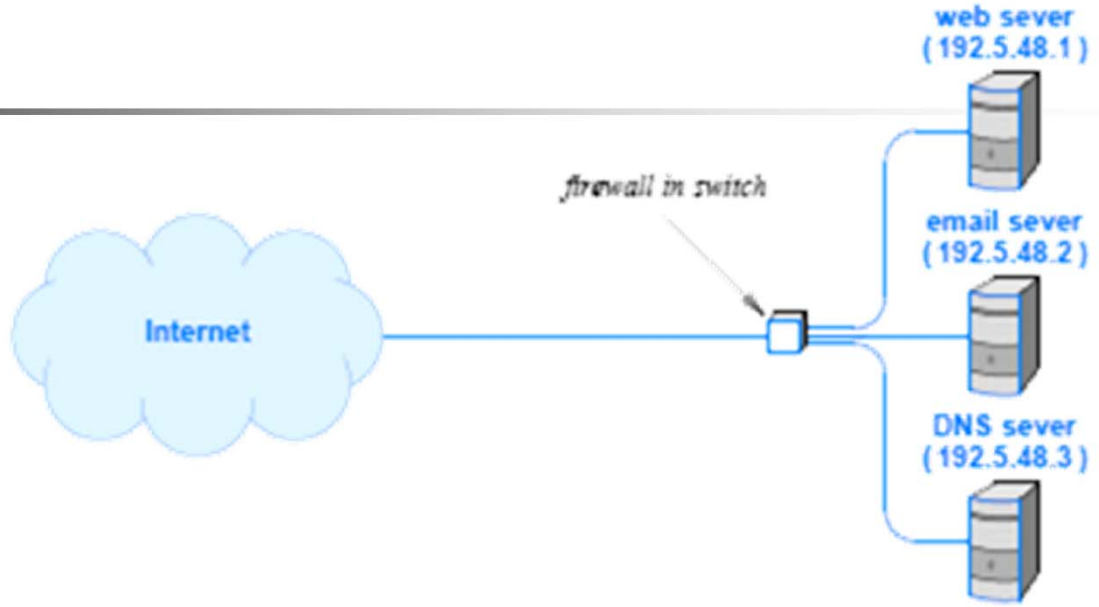
- **หลักในการควบคุม Traffic โดย Firewall**
  - ทุก Traffic ที่จะเข้ามาใน Network ขององค์กร ต้องผ่าน Firewall
  - ทุกๆ Traffic ที่จะออกจากองค์กร ต้องผ่าน Firewall
  - ที่ Firewall จะมีการกำหนด Security Policy ที่จะ Drop ทุกๆ Packet ที่ไม่เป็นไปตามข้อกำหนด
  - ตัว Firewall เองจะต้องคงทนต่อการ Attack
- **จุดที่วาง Firewall จะเป็นตัวกำหนด Secure Perimeter (ขอบเขต)**

ชนิดของ Firewall

- **Packet Filtering Router/Gateway**
  - ทำการกรอง Packet โดยดูจาก MAC Address, IP Address, Port Number, Protocol หรือข้อมูลในส่วน Header แต่จะไม่ดูภายใน Packet และไม่มีการบันทึกหรือจดจำ = Stateless
- **Application Level Gateway**
  - ปกติจะลงพวก Proxy Application ทำหน้าที่เป็นตัวแทนในการเชื่อมต่อกับ Application ภายนอก จะยอมให้เชื่อมต่อกับ Application ที่มี Proxy อยู่เท่านั้น ปกติสามารถทำเป็น Stateful สามารถตรวจสอบภายใน Packet และมี Function ของ Packet Filter อยู่ด้วย
- **Circuit Level Gateway**
  - เป็นตัว Relay สำหรับการทำ Connection ภายนอก ปกติจะทำงานร่วมกับ Proxy Gateway

Firewall Implementation with a Packet Filter

- สมมุติเรายอมให้เฉพาะ Traffic จาก Public Server เท่านั้นที่จะเข้า-ออก



Dir	Frame Type	IP Src	IP Dest	IP Type	Src Port	Dst Port
in	0800	*	192.5.48.1	TCP	*	80
in	0800	*	192.5.48.2	TCP	*	25
in	0800	*	192.5.48.3	TCP	*	53
in	0800	*	192.5.48.3	UDP	*	53
out	0800	192.5.48.1	*	TCP	80	*
out	0800	192.5.48.2	*	TCP	25	*
out	0800	192.5.48.3	*	TCP	53	*
out	0800	192.5.48.3	*	UDP	53	*

Intrusion Detection System

- IDS หรือ Intrusion Detection System จะตรวจสอบ Packet ทุกตัวที่เข้ามาใน Site และจะแจ้งแก่ผู้ดูแลระบบถ้ามี Packet ที่ไม่เป็นไปตามที่กำหนด
- เป็นการทำงานเสริมจาก Firewall แต่จะตรวจจับภายใน Network
- ปกติ IDS จะถูก Configure ให้ตรวจจับ Pattern เฉพาะของการ Attack เช่นการทำ Port Scanning
- บางครั้งจะทำงานร่วมกับ Firewall โดยแจ้งให้ Firewall ทำการ Block บาง Packet เช่นการทำ SYN Flood
- IDS จะเป็น Stateful มีการจดจำสถานะของการ Connection

Content Scanning and Deep Packet Inspection

- **ปกติ Firewall จะกัน Virus ไม่ได้ เพราะไม่สามารถตรวจสอบภายใน Packet**
  - เช่นการส่ง Virus ผ่าน E-mail Attachment
- **เราต้องการการทำ Content Analysis**
  - File Scanning ทำโดย Security Software บน PC
    - File Scanner จะนำ File ที่ได้รับ และทำการตรวจหา Pattern ที่น่าจะมีปัญหา เช่น String ของ Byte ที่เรียก 'Finger Print' อย่างไรก็ตาม เป็นไปได้ที่อาจจะเกิด 'False Positive' หรือ 'False Negative'
  - Deep Packet Inspection(DPI)
    - จะเป็นการตรวจสอบภายใน Packet แทนที่จะตรวจสอบ File ซึ่งในการนี้มันสามารถตรวจสอบ Packet ที่วิ่งเข้า-ออก ทั้งส่วน Header และ ภายใน Payload ด้วยเหตุนี้มันจะทำงานได้ค่อนข้างช้าเทียบกับ Firewall และไม่เหมาะกับ High-Speed Network

Virtual Private Network (VPN)

- **เป็น Technology ที่สำคัญที่สุดอันหนึ่งในการสร้าง Secure Access กับ Remote Site ผ่าน Internet**
  - สมัยก่อนจะใช้ Leased Circuit (Private Network) ในการเชื่อมต่อ ซึ่งจะมี Security สูง
  - ปัจจุบัน การต่อผ่าน Internet ราคาจะถูกกว่ามาก แต่มีปัญหาเรื่อง Security เพราะ Internet เป็น Public Network
  - วิธีแก้ คือการทำ Encryption ในข้อมูลที่ส่ง
    - อาจจะทำที่ Router ที่ทำหน้าที่เป็น Firewall ด้วย
    - อาจจะทำที่ Host
  - นี่คือ Technology ของ Virtual Private Network หรือ VPN คือทำ Internet ที่มีราคาถูกให้มีคุณภาพในแง่ Security เหมือน Private Network ที่ราคาแพง

Virtual Private Network (VPN)

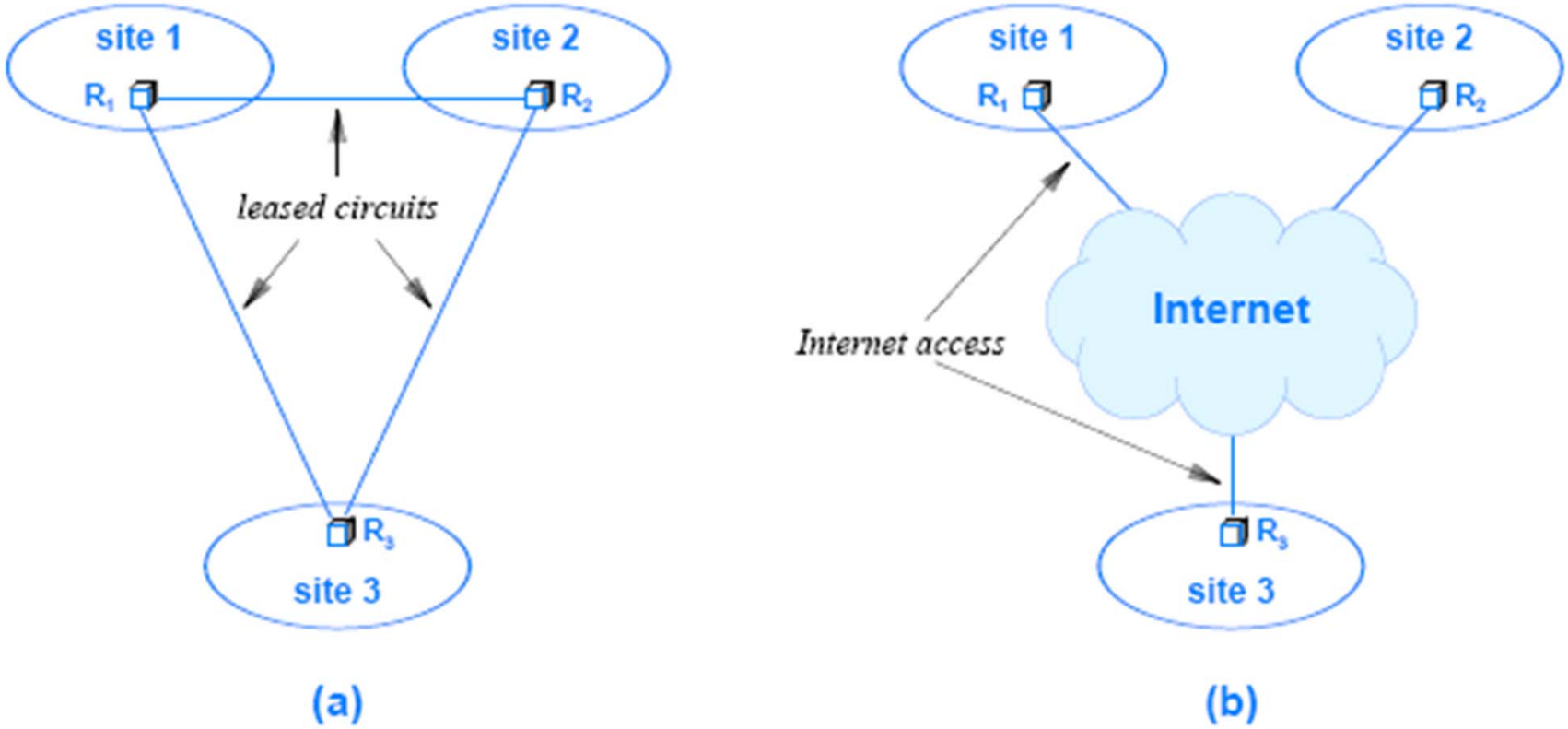


Figure 30.10 Sites connected by (a) leased circuits and (b) the Internet.

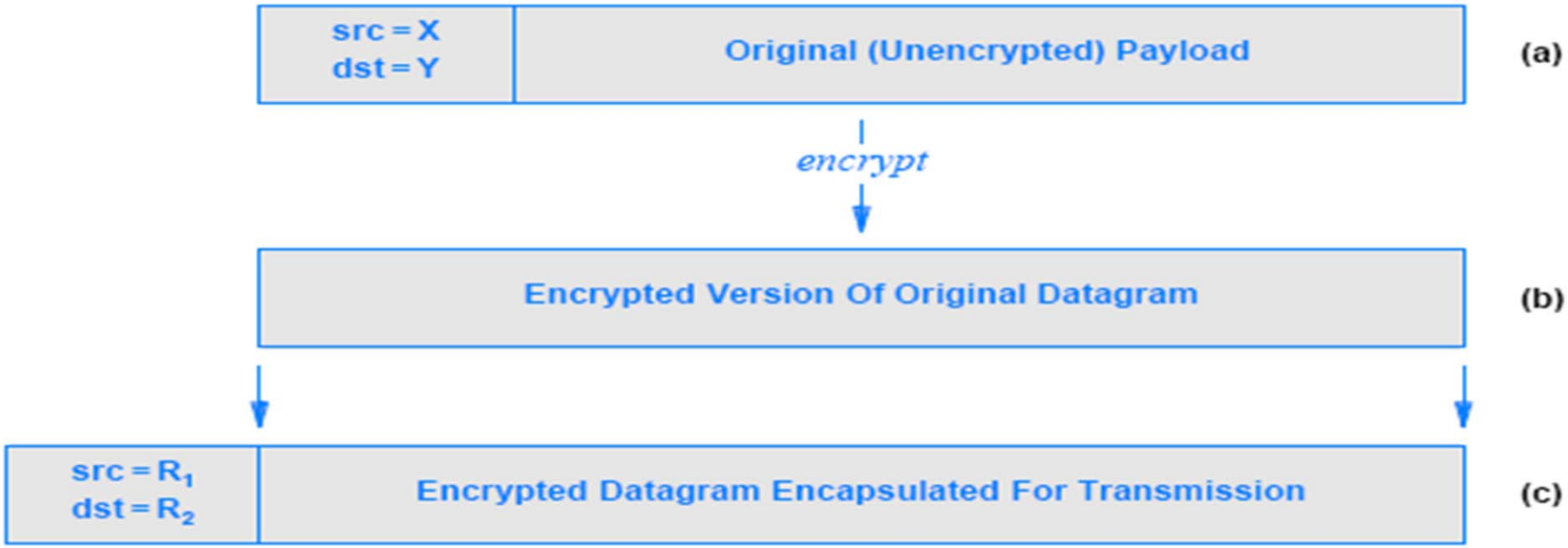
The Use of VPN Technology for Telecommuting

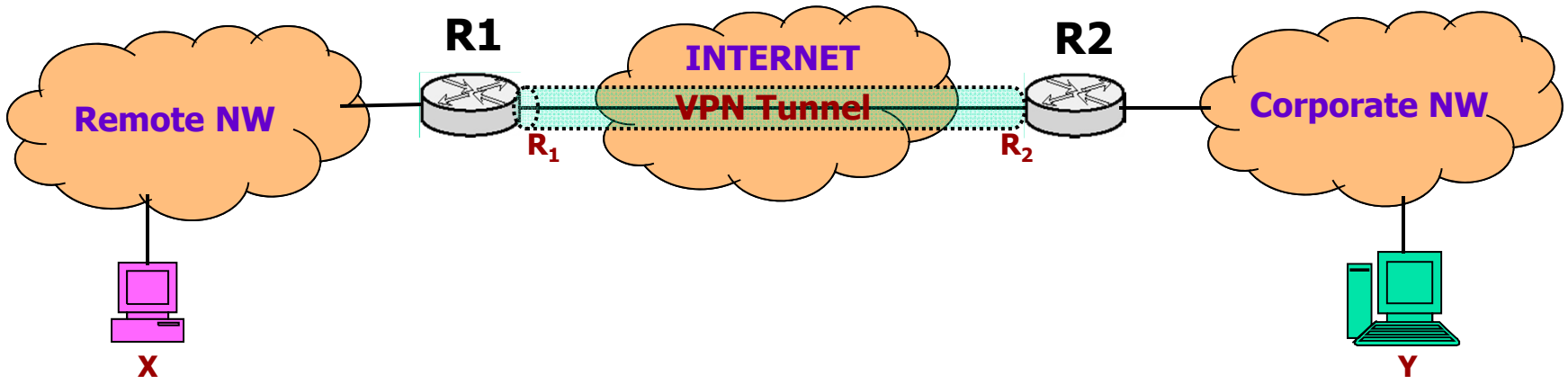
- **VPN มีการใช้งานสองรูปแบบ**
  - Stand-Alone Device
    - โดยการใช้อุปกรณ์ที่เรียก VPN Router ต่อกับ Internet โดยอุปกรณ์นี้จะสร้างช่องการสื่อสารที่ปลอดภัยกับ VPN Server ขององค์กร โดยจะส่ง Encrypt Packet และจะ Decrypt Packet ที่ได้รับ ผู้ใช้เมื่อต่อผ่าน Internet จะเสมือนว่าตัวเองได้ต่อตรงเข้ากับ LAN ขององค์กร โดยมีการจ่าย IP ผ่าน DHCP ขององค์กร
  - VPN Software
    - โดยการ Run VPN Software ที่ Host ของผู้ใช้ที่ต้องการเชื่อมต่อกับ Network ขององค์กรผ่าน Internet โดยที่ VPN Software จะดักจับ Packet ที่เข้าและออกจาก Host นั้น เพื่อทำการ Decryption และ Encryption ตามลำดับ

Packet Encryption vs. Tunneling

- **การทำ Encryption ของ VPN มีทางเลือกหลักสามทาง**
  - Payload Encryption
    - เฉพาะส่วน Payload ของ Datagram จะถูก Encrypt ดังนั้นผู้ดักฟังสามารถจะรู้ Address และ Port Number ของ Datagram ได้ เหมาะกับกรณีที่ส่วน Header ของ Datagram ไม่มีความสำคัญที่ต้องปกป้อง
  - IP-in-IP Tunneling
    - ในกรณีนี้ ทั้ง Datagram จะถูก Encrypt จากนั้น Packet ที่ถูก Encrypt แล้วจะถูกนำไปใส่ใน Datagram อันใหม่ ปกติ IP Address ใหม่ที่ใช้คือ IP ของ Router ที่ทำ Encryption ต้นทางและ Decryption ปลายทาง ผลลัพธ์คือทั้ง Datagram จะถูกปกป้อง แต่ Datagram ใหม่จะยาวมากกว่าเดิม

Packet Encryption vs. Tunneling





Packet Encryption vs. Tunneling

- **การทำ Encryption ของ VPN มีทางเลือกหลักสามทาง**
  - Payload Encryption
  - IP-in-IP Tunneling
  - IP-in-TCP Tunneling
    - จะใช้การบรรจุ IP Packet ที่ถูก Encrypt ลงใน TCP Segment จากนั้นจึงบรรจุ TCP ลงใน IP อีกทีหนึ่ง ข้อดีคือสามารถอาศัย TCP ในการส่งข้อมูลแบบเชื่อถือได้ ข้อเสียคือ ถ้า Packet ใดใน Connection นั้นสูญหาย ข้อมูลที่ได้รับต่อจากนั้นจะไม่สามารถส่งจากชั้น TCP ขึ้นไป Decrypt ที่ IP ชั้นบนได้

VPN Tunneling Performance

- **การใช้ VPN จะทำให้ Performance ของระบบลดลง ดังนี้**
  - Latency
    - การใช้ VPN ในการเชื่อมต่อกับ Site อื่น จะต้องผ่านท่อ VPN ไปยังองค์กร แล้วจึงต่อไปยัง Site นั้นได้ ทำให้เกิด Delay เพิ่มขึ้น
  - Throughput
    - แม้ว่าการเชื่อมผ่าน VPN จะเสมือนต่อกับ LAN ขององค์กร แต่ข้อมูลต้องวิ่งผ่าน Internet ทำให้ Throughput ต่ำกว่าการใช้งานของ LAN ปกติ
  - Overhead and Fragmentation
    - การทำ Tunnel จะเพิ่ม Overhead ในการส่ง Datagram และเป็นไปได้เมื่อทำแล้วทำให้ความยาวของ Packet สูงกว่าค่า MTU ซึ่งจะส่งผลให้เกิดการทำ Fragmentation

Security Technologies ที่สำคัญ

- **PGP (Pretty Good Privacy)**
  - เป็นระบบการเข้ารหัสข้อมูล ใช้สำหรับ Application เข้ารหัสข้อมูลก่อนที่จะส่งออกไป
- **SSH (Secure Shell)**
  - เป็น Application Layer Protocol สำหรับทำ Remote login เข้ามา ซึ่งจะมีการเข้ารหัสข้อมูลระหว่างการส่ง ผิดกับ 'Telnet' ที่จะส่งเป็น Plaintext
- **SSL (Secure Socket Layer)**
  - พัฒนาโดย Netscape Communication ใช้ทำ Authentication และปกป้องข้อมูล โดยจะเป็น Layer อยู่ระหว่าง Application และ Socket API (Transport Layer) ใช้สำหรับการสื่อสารผ่าน Web ในการทำ Financial Transaction
- **TLS (Transport Layer Security)**
  - เป็นมาตรฐานของ IETF เพื่อเป็นมาตรฐานแทน SSL โดยออกแบบจากพื้นฐายของ SSL v.3 ทั้ง SSL และ TLS สามารถนำมาใช้ร่วมกับ HTTPS

Security Technologies ที่สำคัญ

- **HTTPS (HTTP Security)**
  - เป็นเทคโนโลยีที่รวม HTTP และ SSL หรือ TLS เข้าด้วยกันกับการทำ Certificate เพื่อให้การสื่อสารผ่าน Web ได้อย่างปลอดภัย HTTPS จะใช้ TCP Port 443 แทนที่จะเป็น 80
- **IPsec (IP Security)**
  - เป็นเทคโนโลยีในการทำ Security กับ IP Datagram โดยสามารถเลือกทำ Authentication หรือ Confidentiality (Encryption)
- **RADIUS (Remote Authentication Dial-In User Service)**
  - เป็น Protocol ใช้สำหรับการทำ Authentication, Authorization และ Accounting (AAA) นิยมใช้ใน Dialup และ VPN สำหรับ Remote User
- **WEP (Wired Equivalent Privacy)**
  - เป็นส่วนหนึ่งของ Wi-Fi WLAN Standard ปัจจุบันถูกแทนที่ด้วย WPA (Wi-Fi Protection Access) เนื่องจากมีจุดอ่อน

Chapter 31: Network Management (SNMP)

- **NW Manager หรือ NW Administrator เป็นผู้ทำหน้าที่**
  - Planning
  - Installing
  - Operating
  - Monitoring
  - Controlling
- **NW Manager จะตรวจดู/แก้ไข ทั้ง HW และ SW เพื่อทำให้ NW ทำงานได้อย่างมีประสิทธิภาพ**

Chapter 31: 31.3 NW Management Standard Model

- **Industry Standard Model**
- **FCAPS**
  - Fault Detection and Correction
  - Configuration and Operation
  - Accounting and Billing
  - Performance Assessment and Optimization
  - Security Assurance and Protection

Chapter 31: 31.4 Network Elements

- หมายถึงอุปกรณ์ ระบบ หรือกลไก ที่สามารถจัดการได้ รวมถึง Service ต่างๆ ของ Network ด้วย

Manageable Network Elements	
Layer 2 Switch	IP router
VLAN Switch	Firewall
Wireless Access Point	Digital Circuit (CSU/DSU)
Head-End DSL Modem	DSLAM
DHCP Server	DNS Server
Web Server	Load Balancer

Chapter 31: 31.5 Network Management Tools

- **Physical Layer Testing**
  - One-Touch, DSP 4000, Cable Tester, RF Signal Meter
- **Reachability and Connectivity**
  - Ping
- **Packet Analysis**
  - Packet Analyzer (Protocol Analyzer) เช่น Ethereal, Wireshark, Sniffer
- **Network Discovery**
  - ใช้ในการสร้าง NW Map
- **Device Interrogation Tool**
  - ใช้ในการส่งคำสั่งไปยังอุปกรณ์ต้องการ

Chapter 31: 31.5 Network Management Tools

- **Event Monitoring**
  - เป็นตัวแสดงการทำงานของอุปกรณ์แต่ละตัวผ่านทางจอภาพ จะมีการส่ง Alert บนจอเมื่อมีสิ่งผิดปกติเกิดขึ้น
- **Performance Monitoring**
  - เป็นตัวตรวจจับและบันทึกประสิทธิภาพการทำงานของ NW
- **Flow Analysis**
  - เช่น NetFlow Analyzer แสดง Traffic ในแต่ละ Link และแต่ละ Application ที่วิ่งอยู่ใน NW

Chapter 31: 31.5 Network Management Tools

- **Routing and Traffic Engineering และ General Configuration Tools**
  - Routing จะควบคุมเส้นทางการไหลของข้อมูล และ ทำ Configuration Routing Protocol
  - Traffic Engineering จะตรวจจับและทำ Configuration ของเส้นทางของข้อมูลเพื่อให้เป็นไปตาม QoS ที่ต้องการ
  - General Configuration Tool ใช้สำหรับการทำ Configuration ทั่วไป

Chapter 31: 31.5 Network Management Tools

- **Security Enforcement**
  - ใช้เพื่อบังคับให้การทำงานของ NW เป็นไปตาม Security Policy
- **Network Planning**
  - จะเป็นตัวที่สลับซับซ้อนมากที่สุด ใช้สำหรับการวางแผนในการทำงานของ NW เช่นใช้ทำ NW Optimization สำหรับ NW Architecture หรือ Traffic Engineering

Chapter 31: 31.6 NW Management Application

- **ปกติเป็น Application Layer จะทำงานในลักษณะ Client-Server**
  - ส่วน Client จะทำงานบน PC เรียก Manager
  - ส่วน Server จะทำงานในอุปกรณ์ NW เรียก Agent
- **Manager จะร้องขอข้อมูลจาก Agent ที่อยู่บนอุปกรณ์ต่างๆ เพื่อมาเก็บไว้ในฐานข้อมูลและทำการวิเคราะห์**
  - การร้องขอและส่งข้อมูลจะกระทำผ่าน NW Management Protocol (Application Layer Protocol)

Chapt 31: 31.7 Simple Network Management Protocol (SNMP)

- เป็นมาตรฐานที่ใช้ใน Internet
  - ปัจจุบันคือ SNMPv3
- กำหนด Format ของข้อมูลที่ส่งระหว่าง Manager และ Agent
  - ข้อมูลที่ส่ง จะใช้รหัสแบบ ASN.1
    - Abstract Syntax Notation 1

Decimal Integer	Hexadecimal Equivalent	Length Byte	Bytes Of Value (in hex)
27	1B	01	1B
792	318	02	03 18
24,567	5FF7	02	5F F7
190,345	2E789	03	02 E7 89

Chapter 31: 31.8 SNMP's Fetch-Store Paradigm

- **SNMP มี Primitive Command เพียงไม่กี่ตัว**
- **ใช้ Fetch-Store Paradigm**
  - Fetch ใช้ในการดึงค่ามาจากตัวอุปกรณ์
  - Store ใช้ในการตั้งค่าให้กับอุปกรณ์
  - Operation จะกระทำกับ Object โดยกำหนด Object Name

Chapter 31: 31.9 SNMP MIB and Object Names

- **แต่ละ Object ที่จะสื่อสารผ่าน SNMP จะต้องมีชื่อเฉพาะเป็นของตนเอง**
  - ทั้ง Manager และ Agent จะต้องใช้ชื่อเดียวกันสำหรับ Object เดียวกัน
- **Set ของทุกๆ Object ที่ใช้ใน SNMP รู้จักกันในนาม MIB**
  - Management Information Base
- **มาตรฐาน MIB จะแยกออกจากมาตรฐานของ SNMP**
  - MIB ของอุปกรณ์แต่ละชนิด จะมีมาตรฐานแตกต่างกันออกไป

Chapter 31: 31.9 SNMP MIB and Object Names

- **Object ใน MIB จะถูกกำหนดโดยใช้ชื่อตามแบบของ ASN.1**
  - แต่ละ Object จะกำหนดด้วย Prefix ที่จะแน่ใจว่าชื่อจะไม่ซ้ำกัน
  - เช่น ชื่อ Object ที่ใช้นับจำนวนของ Datagram ที่อุปกรณ์ได้รับจะเป็น
    - Iso.org.dod.internet.mgmt.mib.ip.ipInReceives
- **แต่เมื่อชื่อ Object ถูกส่งไปใน Message ของ SNMP มันจะถูกแทนที่ด้วย Integer เช่นตัวอย่างข้างบนจะเป็น**
    - 1.3.6.1.2.1.4.3

Chapter 31: 31.10 ชนิดของ MIB Variables

- **SNMP ไม่ได้กำหนด Set ของ MIB ดังนั้น MIB สามารถออกแบบและกำหนดเป็นมาตรฐานได้อย่างอิสระ**
  - MIB ที่เป็นมาตรฐานเช่น UDP, TCP, IP, ARP และส่วนของ Ethernet
  - MIB สำหรับอุปกรณ์เช่น Switch, Router, Modem, Printer
  - เมื่อมี Protocol ใหม่เกิดขึ้น สามารถกำหนด MIB เพิ่มเติม โดยไม่ต้องแก้ไขส่วน SNMP
  - ผู้ผลิตอุปกรณ์สามารถเพิ่มเติมส่วน MIB Extension สำหรับอุปกรณ์ของตนเอง
- **RMON เป็น MIB Extension ที่สำคัญ สำหรับ Manage ใน Layer 2(LAN)**

Chapter 31: 31.11 MIB Variable สำหรับ Array

- นอกเหนือจาก **MIB Variable** ที่กำหนดเป็นค่า **Integer** แล้ว ยังมี **MIB** ที่กำหนดเป็น **Array** หรือ ตาราง
  - เช่น MIB สำหรับ Routing Table (Forwarding Table)
- อย่างไรก็ตาม ใน **ASN.1** ไม่มีการกำหนด **Index** ของ **Array** เราจะต้องรู้ตำแหน่งของ **Object** ในตาราง และกำหนด **Index** เอง ต่อจาก **Object Name** เช่น **IP Forwarding Table**
  - Standard MIB prefix.ip.ipRoutingTable
  - ถ้าเราต้องการเฉพาะบาง Field เราจะใช้
  - Standard MIB prefix.ip.ipRoutingTable.ipRouteEntry.<field>.IPdestaddr

End of Chapter 30-31 (Week 16)

- HW 10 Download: Last Homework
- Due Monday May 2 Before 12.00 Noon
  - เฉลยจะประกาศวันพุธ
- Course Ends: No Class Next Week
- Final Exam: Friday 13 May 13.30-16.00

Final Exam Preparations

- **Final Exam**
  - เน้นที่หลัง Midterm ก่อน MT จะออกเรื่อง IP และ IP Address
  - No Calculator
  - คะแนนเก็บ Final 50% (6 ข้อ 60 คะแนน)
    - การบ้าน 10 ครั้ง 15%
    - Midterm Exam 35%
  - เกณฑ์ผ่านคือคะแนนรวมต้องได้ 40%

Final Exam Preparations

- **Final Exam List**
- 1. IP Address and Subnetting
- 2. TCP/UDP
- 3. Routing General, Least Cost Algorithm
- 4. IP Routing, RIP, OSPF, BGP, Multicast
- 5. QoS and IP Telephony
- 6. NW Security และ NW Management
- เทอม 1/59 เปิด 1 Sec (จันทร์)