

# Table of Contents

<b>TCP/IP Model</b>	<b>3</b>
Encapsulation & Decapsulation	3
Layer 1 : Host-to-Network Layer	4
Layer 2 : The Internet Layer	4
IP (Internet Protocol)	4
IP Header	4
ICMP (Internet Control Message Protocol) - PING	5
ICMP Header	5
Layer 3 : Transport Layer	6
TCP (Transmission Control Protocol)	6
TCP Header	6
TCP Communication	7
TCP Three-Way Handshake	7
UDP (User Datagram Protocol)	8
UDP Header	8
Layer 4 : Application Layer	9
<b>OSI Model vs TCP/IP Model</b>	<b>9</b>
<b>OSI Model (OSI Layer)</b>	<b>10</b>
Concept	10
Encapsulation & Decapsulation	10
Layer 1 : Physical Layer	11
Data Unit	11
Concept	11
Responsibility	11
Protocol	11
Layer 2 : Data Link Layer	12
Data Unit	12
Concept	12
Responsibility	12
Protocol	13
Layer 3 : Network Layer	14
Data Unit	14
Concept	14
Responsibility	14
Protocol	14
Layer 4 : Transport Layer	15
Data Unit	15
Concept	15
Responsibility	15
Protocol	17

<b>IP Address (Internet Protocol Address)</b>	<b>18</b>
IPv4	18
Characteristic	18
Classfull Addressing	18
Subnets Mask	19
Characteristic	19
Classless Addressing	19
Classless InterDomain Routing (CIDR)	19
Variable Length Subnet Masks ( VLSM )	19
[การบ้าน] Let's try!!	20
IP Summarization	24
[การบ้าน] จากรูปด้านล่างจะทำ IP Summarization ...	24
Special IPv4 Address	25
Automatic Private Internet Protocol Addressing (APIPA)	25
Loopback Address / Localhost Address	25
Network ID / Subnet ID / Network Address / Subnet Address	25
Broadcast Address	26
Default Gateway Address	26
0.0.0.0	26
Address Mapping & Address Translation	26
Address Resolution Protocol (ARP)	26
Network Address Translation (NAT)	27
Dynamic Host Configuration Protocol (DHCP)	29
IPv6	29
Characteristic	30
Text Representation of IPv6 Addresses	31
Special IPv6 Address	31
DNS	31
Collision Domain & Broadcast Domain	33
Collision Domain	33
Broadcast Domain	33
[การบ้าน] Let's try!!	34
Packet Scenario	35
Let's try!!	35
[การบ้าน] ต้องการสื่อสารข้อมูลระหว่าง ...	37
[การบ้าน] จากรูปด้านล่างเป็นรูปแบบ ...	38

## TCP/IP Model

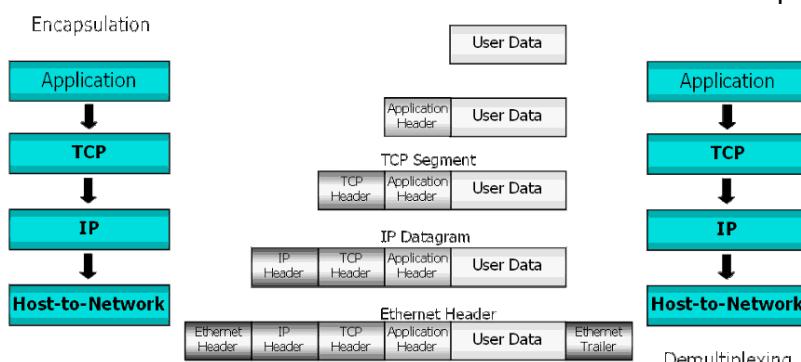
TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้โดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

ชุดโปรโตคอลนี้ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ต ทำให้โนเบล TCP/IP เป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน โดย TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐานสามประการคือ

- เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
- ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่ง และผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาเส้นทางเลือกอื่น เพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
- มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิด ทั้งแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ Real-Time รวมทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data) และแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล

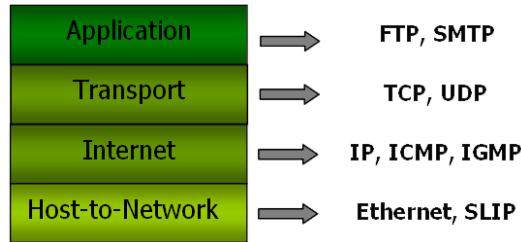
## Encapsulation & Decapsulation

การส่งข้อมูลผ่านในแต่ละレイเยอร์ แต่ละレイเยอร์จะทำการประกอบข้อมูลที่ได้รับมา กับข้อมูลส่วนควบคุมซึ่งถูกนำมายไว้ในส่วนหัวของข้อมูล เรียกว่า เ shedเดอร์ (Header) ภายในshedเดอร์จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการการทำงานย้อนกลับคือ โปรโตคอลเดียวกัน ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็นshedเดอร์ก่อน และนำไปประมวลจึงทำให้ทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing



ข้อมูลที่ผ่านการ Encapsulate ในแต่ละレイเยอร์มีชื่อเรียกแตกต่างกันโดยเมื่อข้อมูลที่มาจาก ผู้ใช้งาน หรือก็คือข้อมูลที่ผู้ใช้งานเป็นผู้ป้อนให้กับ Application เรียกว่า User Data

- เมื่อแอปพลิเคชันได้รับข้อมูลจากผู้ใช้งาน ก็จะนำมาประกอบกับshedเดอร์ของแอปพลิเคชัน เรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP
- เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมารวมกับshedเดอร์ของ โปรโตคอล TCP เรียกว่า TCP Segment และส่งต่อไปยังโปรโตคอล IP
- เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมารวมกับshedเดอร์ของ โปรโตคอล IP เรียกว่า IP Datagram และส่งต่อไปยังレイเยอร์ Host-to-Network Layer
- ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ Flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นลัญญาณไฟฟ้า ส่งผ่านสายลัญญาณที่เชื่อมโยงอยู่ต่อไป



## Layer 1 : Host-to-Network Layer

- ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ
- ทำหน้าที่รับข้อมูลจากชั้นสื่อสาร IP จากนั้นจึงส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูล
- ทางด้านผู้รับก็จะทำงานในทางกลับกัน

## Layer 2 : The Internet Layer

- ใช้ระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็คเก็ต (Packet-Switching Network) ในการสื่อสาร
- เป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless)
- หลักการทำงาน คือบล็อกให้ข้อมูลขนาดเล็กที่เรียกว่า แพ็คเก็ต (Packet) สามารถออกจากโหนดผู้ส่งไปตามโหนดต่าง ๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ
- เมื่อแพ็คเก็ตแต่ละตัวในชุดที่ถูกก็จะเป็นอิสระแก่กันและกัน ทำให้แพ็คเก็ตไปถึงปลายทางอาจจะไม่เป็นไปตามลำดับ

## IP (Internet Protocol)

- เป็นโปรโตคอลที่อยู่ใน Network Layer เมื่อเทียบกับโน้มเดล OSI
- เป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless)
  - ไม่ทราบถึงข้อมูลเดียวต่อเดียวที่ส่งก่อนหน้าหรือส่งตามมา เพราะเกิดเส้นทางการเชื่อมต่อในทุก ๆ ครั้งของการส่งข้อมูล 1 \data\agram
  - การส่งข้อมูลใน 1 \data\agram อาจจะเกิดการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อย ๆ (Fragmentation) และถูกนำไปรวมเป็น\data\agramเดิมเมื่อถึงปลายทาง
- ทำหน้าที่ต่าง ๆ ดังนี้
  - จัดการเกี่ยวกับแอดเดรสและข้อมูล
  - ควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของแพ็คเก็ต
  - หาเส้นทางที่ดีที่สุดในการส่งข้อมูล โดยสามารถเปลี่ยนแปลงเส้นทางในระหว่างการส่งได้
- มีระบบการแยกและประกอบ\data\agram (Datagram)
  - เพื่อรับการส่งข้อมูลระดับ Data Link ที่มีขนาด MTU (Maximum Transmission Unit) ที่แตกต่างกัน ทำให้สามารถนำ IP ไปใช้บนโปรโตคอลอื่นได้หลากหลาย เช่น Ethernet, Token Ring หรือ Apple Talk

## IP Header

4-bit Version	Header Length	8-bit Type of Service	16-bit Total Length in Byte	
16-bit Identification		3-bit Flag	16-bit Fragment Checksum	
8-bit Time to Live (TTL)	8-bit Protocol	16-bit Header Checksum		
32-bit Source IP Address				
32-bit Destination IP Address				
Option				
Data				

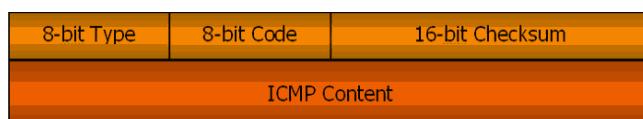
เซดเดอร์ของ IP โดยปกติจะมีขนาด 20 ไบต์ ยกเว้นในกรณีที่มีการเพิ่มส่วนเสริม (Option) บางอย่าง ฟลัดของเซดเดอร์ IP จะมีความหมายดังนี้

- **Version** : หมายเลขเวอร์ชันของโปรโตคอล ในปัจจุบันมีเวอร์ชัน 4 (IPv4) และเวอร์ชัน 6 (IPv6)
- **Header Length** : ความยาวของเซดเดอร์ โดยทั่วไปถ้าไม่มีส่วน Option จะมีค่าเป็น 5 ( $5 \times 32$  bit)
- **Type of Service (TOS)** : ใช้เป็นข้อมูลสำหรับเราเตอร์ในการตัดสินใจเลือกเส้นทางข้อมูลในแต่ละ\data\ต้าแกรม แต่ในปัจจุบันไม่ได้มีการนำ它ไปใช้งานแล้ว
- **Length** : ความยาวทั้งหมดเป็นจำนวนไบต์ของ\data\ต้าแกรม ซึ่งด้วยขนาด 16 บิตของฟลัดจะหมายถึงความยาวสูงสุดของ\data\ต้าแกรม คือ 65535 Byte (64k) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วน ๆ ตามขนาดของ MTU ที่กำหนดในลิงค์レイเยอร์ และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทาง แอพพลิเคชันส่วนใหญ่จะมีขนาดของ\data\ต้าแกรมไม่เกิน 512 Byte
- **Identification** : เป็นหมายเลขของ\data\ต้าแกรมในกรณีที่มีการแยก\data\ต้าแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี Identification เดียวกันมารวมกัน
- **Flag** : ใช้ในการกรณีที่มีการแยก\data\ต้าแกรม
- **Fragment Offset** : ใช้ในการกำหนดตำแหน่งของข้อมูลใน\data\ต้าแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อ กันได้อย่างถูกต้อง
- **Time to Live (TTL)** : กำหนดจำนวนครั้งที่มากที่สุดที่\data\ต้าแกรมจะถูกส่งระหว่าง Hop (การส่งผ่านข้อมูลระหว่างอุปกรณ์) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 Hop จะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และเราเตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมา�ังต้นทางว่ามีการเกิด Time Out ในระหว่างการส่งข้อมูล
- **Protocol** : ระบุโปรโตคอลที่ส่งใน\data\ต้าแกรม เช่น TCP, UDP หรือ ICMP
- **Header Checksum** : ใช้ในการตรวจสอบความถูกต้องของข้อมูลในเซดเดอร์
- **Source IP Address** : หมายเลข IP ของผู้ส่งข้อมูล
- **Destination IP Address** : หมายเลข IP ของผู้รับข้อมูล
- **Data** : ข้อมูลจากโปรโตคอลระดับบน

## ICMP (Internet Control Message Protocol) - PING

- เป็นโปรโตคอลที่ใช้ในการตรวจสอบ และรายงานสถานภาพของ\data\ต้าแกรม (Datagram) ในกรณีที่เกิดปัญหา กับ\data\ต้าแกรม เช่น เราเตอร์ไม่สามารถส่ง\data\ต้าแกรมไปถึงปลายทางได้ ICMP
- ไม่สามารถรับประทานได้ว่า ICMP Message ที่ส่งไปจะถูกผู้รับจริงหรือไม่
- หากมีการส่ง\data\ต้าแกรมออกไปแล้วไม่มี ICMP Message ฟ้อง Error กลับมา มีสองกรณีคือ
  - ข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อย
  - ICMP Message ที่ส่งกลับมา ก็มีปัญหาระหว่างทาง หรืออาจจะมีปัญหาในการสื่อสาร เช่น การส่ง\data\ต้าแกรม
- เป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (Unreliable) โดยจะใช้โปรโตคอลในระดับที่สูงกว่า เช่น ใน Network Layer สำหรับการจัดการให้การสื่อสารนั้น ๆ มีความน่าเชื่อถือ

## ICMP Header



ในส่วนของ ICMP Message จะประกอบด้วย Type ขนาด 8 บิต Checksum ขนาด 16 บิต และส่วนของ Content ซึ่งจะมีขนาดแตกต่างกันไปตาม Type และ Code ดังรูป

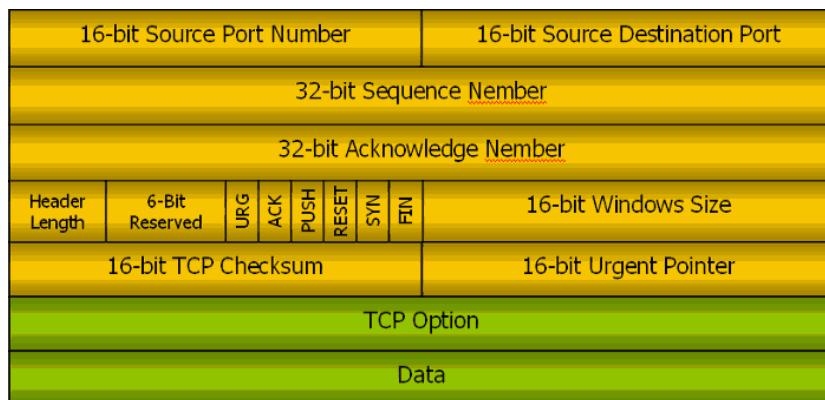
## Layer 3 : Transport Layer

สามารถแบ่งเป็นโปรโตคอล 2 ชนิดตามลักษณะ คือ Transmission Control Protocol (TCP) และ UDP (User Datagram Protocol)

### TCP (Transmission Control Protocol)

- เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI
- เป็นการติดต่อแบบมีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (Connection-Oriented)
- ทำหน้าที่จัดการ และควบคุมการรับส่งข้อมูลให้มีความถูกต้อง (Reliable)
- ส่งข้อมูลเป็นแบบ Byte Stream โดยไม่มีข้อผิดพลาด
- ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็ก ๆ เรียกว่า Message โดยจะถูกส่งไปยังผู้รับผ่านทางขั้นสื่อสารของอินเทอร์เน็ต
- ฝ่ายผู้รับจะนำ Message มาเรียงต่อกันตามลำดับเป็นข้อมูลเดิม TCP และมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทัน

### TCP Header



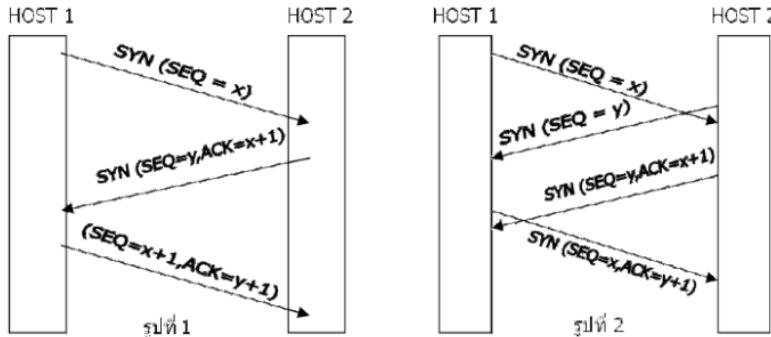
- **Source Port Number** : หมายเลขพอร์ตต้นทางที่ส่งdataframeนี้
- **Destination Port Number** : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับdataframe
- **Sequence Number** : ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมารับลำดับได้ถูกต้อง
- **Acknowledgment Number** : ทำหน้าที่เหมือนกับ Sequence Number แต่จะใช้ในการตอบรับ
- **Header Length** : โดยปกติความยาวของส่วนหัว TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ Option แต่ต้องไม่เกิน 60 ไบต์
- **Flag** : เป็นข้อมูลระดับบิตที่อยู่ในส่วนหัว TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้น ๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย ซึ่ง Flag มีอยู่ทั้งหมด 6 บิต แบ่งได้ดังนี้

Type	Description
URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent Pointer)
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรจะส่งข้อมูล Segment นี้ไปยัง Application ที่กำลังรออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ (Reset) เนื่องจากในกรณีที่เกิดการสับสนขึ้นด้วยเหตุการณ์ต่าง ๆ เช่น โ伊斯ต์มีปัญหา ให้เริ่มสื่อสารใหม่
SYN	ใช้ในการเริ่มต้นข้อติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่าหยุดการติดต่อ

- การทำงานแต่ละอย่างจะมีการใช้งานฟิลต์ที่ไม่เหมือนกัน โดยจะใช้ Flag เป็นตัวกำหนดการทำงานของ TCP Segment ว่าให้ใช้งานฟิลต์ใด เช่น ฟิลต์ Acknowledgment Number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ แต่จะมีข้อมูลในฟิลต์ ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆ หากไม่มี Flag เป็นตัวกำหนดก็อาจจะมีการนำข้อมูลมาใช้ และก่อให้เกิดความผิดพลาดได้

## TCP Communication

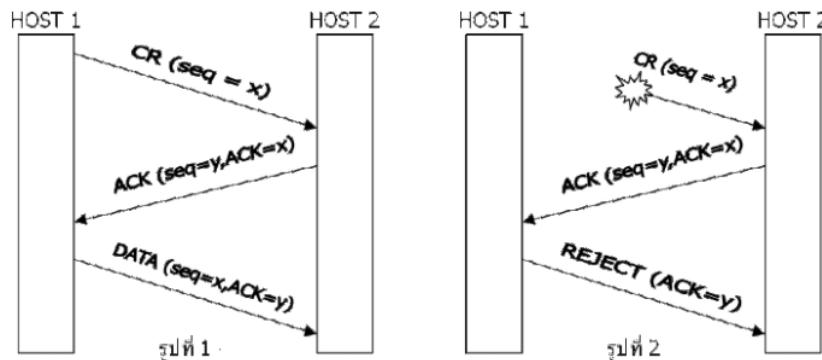
การเชื่อมต่อเริ่มต้นจากสถานะ CLOSED เมื่อเรียกใช้บริการ LISTEN หรือ CONNECT ก็จะมีการเปลี่ยนสถานะไปจากเดิม และถ้าอีกฝ่ายต้องการเชื่อมต่อด้วย การเชื่อมต่อ ก็จะเกิดขึ้น และย้ายไปอยู่ในสถานะ ESTABLISHED คือการเชื่อต่อสมบูรณ์ และถ้าหากยกเลิกการติดต่องจะกลับไปสู่สถานะ CLOSED อีกครั้งเดิม



- เชกเม้นต์ CONNECT (SYN = "1" และ ACK = "0") เดินทางมาถึง Entity TCP ที่โไฮสต์ป्लา yer ทาง
  - ค้นหาพอร์ตเซสตามหมายเลขพอร์ตที่กำหนดในเขตข้อมูล Destination port
  - ↪ ถ้าหากไม่พบ ก็จะตอบปฎิเสธด้วยเชกเม้นต์ที่มี RST = "1" กลับไปยังผู้ส่ง
  - ↪ เชกเม้นต์ CONNECT ของผู้ส่งจะถูกส่งต่อไปยังพอร์ตที่ระบุโดย
    - ถ้าหากพอร์ตเซ็นน์ต้องการสื่อสารด้วย ก็จะส่งเชกเม้นต์ตอบรับกลับไป ดังรูปที่ 1
    - ถ้าหากพอร์ตเซ็นน์ไม่ต้องการสื่อสารด้วย จะตอบปฎิเสธ
- ในกรณีที่โไฮสต์สองแห่งพยายามสร้างการเชื่อมต่อระหว่างชื่อค์เก็ตคู่เดียวกัน จะเกิดการเชื่อมต่อขึ้นเพียงช่องทางเดียว ดังรูปที่ 2
  - เนื่องจากการเชื่อมต่อในแต่ละช่องทางจะถูกกำหนดด้วยหมายเลขชื่อค์เก็ตผู้ส่ง และผู้รับ
  - ถ้าการเชื่อมต่อลำดับแรกสำเร็จ ก็จะถูกบันทึกไว้ในตารางการสื่อสาร เช่น (x, y)
  - ถ้าการเชื่อมต่อลำดับที่สองสำเร็จในเวลาต่อมา ข้อมูลนี้ก็จะถูกบันทึกไว้ที่เดียวกันคือ (x, y)

## TCP Three-Way Handshake

- เป็นวิธีการส่งแพ็กเก็ตที่ช่วยแก้ปัญหาในเรื่องแพ็กเก็ตช้าช้อนได้ดี
- ใช้ร่วมกับวิธีการจัดจังหวะการทำงานให้พร้อมกัน (Synchronization)
- ไม่บังคับให้ผู้ส่งและผู้รับข้อมูลกำหนดค่าเริ่มต้นของหมายเลขลำดับเป็นเลขเดียวกัน
- ต้องสร้างช่องสื่อสารให้ได้ก่อนที่จะเริ่มรับ-ส่งข้อมูล
- แพ็กเก็ตควบคุมที่ใช้ในการต่อรองค่าด้วยแพร์สำหรับการสื่อสารต่าง ๆ อาจเกิดการตกค้างอยู่ในระบบ
  - ทำให้การกำหนดค่าหมายเลขลำดับมีปัญหาไปด้วย เช่น การสร้างช่องสื่อสารระหว่างโไฮสต์ 1 และ โไฮสต์ 2
    - ↪ โไฮสต์ 1 ขอเริ่มการเชื่อมต่อด้วยการส่งแพ็กเก็ต CR (Connection Request) ไปยังโไฮสต์ 2 ซึ่งมีค่าด้วยเบอร์ต่าง ๆ สำหรับการสื่อสารรวมทั้งหมายเลขลำดับ และหมายเลขช่องสื่อสาร
    - ↪ โไฮสต์ 2 (ผู้รับ) ก็จะส่ง ACK (Acknowledge) กลับมายังโไฮสต์ 1 แต่ถ้าแพ็กเก็ตจากผู้ส่งเกิดสูญหายระหว่างทาง และสำเนาแพ็กเก็ตที่ยังตกค้างอยู่ระบบเกิดเดินทางไปถึงผู้รับในภายหลัง ก็จะทำให้การสร้างช่องสื่อสารใช้การไม่ได้เนื่องจากมีค่าด้วยเบอร์ต่าง ๆ ไม่ตรงกัน

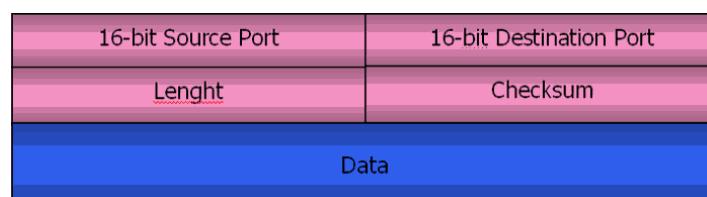


- จากรูปที่ 1 แสดงขั้นตอนการเริ่มต้นการทำงานจากไฮสต์ 1 ไปยังไฮสต์ 2
  - ไฮสต์ 1 เลือกหมายเลขลำดับเป็น “x” และส่งแพ็คเก็ต Connection Request “ไปยังไฮสต์ 2
  - ไฮสต์ 2 ตอบรับด้วยแพ็คเก็ต Connection Accepted ซึ่งจะยอมรับหมายเลขลำดับ “x” พร้อมกับประกาศหมายเลขลำดับ “y” ที่เป็นของตนเอง
  - ไฮสต์ 1 ก็จะตอบรับค่าตัวเลือกของไฮสต์ 2 ผ่านทางเขตข้อมูลสำหรับการควบคุมในแพ็คเก็ตข้อมูลแรกที่ส่งมา
- จากรูปที่ 2 แสดงการเกิดปัญหาการสูญหายของแพ็คเก็ตในขณะที่สำเนาแพ็คเก็ตที่ค้างในระบบเดินทางไปถึงผู้รับ
  - แพ็คเก็ต TPDU (ตัวแรกในรูป) เป็นสำเนาแพ็คเก็ตเดิมที่พึ่งจะเดินทางไปถึงไฮสต์ 2 โดยที่ไฮสต์ 1 ไม่ทราบ (Delay)
  - ไฮสต์ 2 จะตอบรับด้วยการส่งแพ็คเก็ต Connection Accepted TPDU กลับมา ที่ไฮสต์ 1
  - ไฮสต์ 1 จะตรวจสอบว่าหมายเลขลำดับไฮสต์ 2 ตอบกลับมานั้นเป็นหมายเลขลำดับที่ได้เลิกใช้ไปแล้ว จึงมีการส่งแพ็คเก็ต REJECT กลับมายังไฮสต์ 2 เพื่อบอกยกเลิกการทำงาน
  - จะเห็นว่าวิธีการนี้อาศัยการสื่อสารผ่านแพ็คเก็ต 3 ตัวซึ่งเป็นที่มาของคำว่า “การจับมือร่วมสามขั้นตอน” ผลสุดท้าย ทั้งไฮสต์ 1 และไฮสต์ 2 ก็จะไม่มีการสร้างช่องสื่อสารขึ้นมาจากข้อมูลในสำเนาแพ็คเก็ตเดิมแต่อย่างใด

## UDP (User Datagram Protocol)

- เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI
- เป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless)
- ข้อดี คือมีความรวดเร็วในการส่งข้อมูล จึงนิยมใช้กับผู้ให้และผู้ใช้บริการ (Client - Server System) เนื่องจากมีการสื่อสารแบบถาม/ตอบ (Request/Reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหว หรือการส่งเสียง (Voice) ทางอินเทอร์เน็ต
- เป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP Datagram (ซึ่งจะไม่มีความสัมพันธ์กับ Datagram)
- มีกลไกการตรวจสอบโดยคือ UDP Checksum
  - เพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง
  - ปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้น หากพบ Checksum Error ผู้รับจะทำการทิ้งข้อมูลนั้น
  - ไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล จึงถือเป็นการตรวจสอบเพียงฝ่ายเดียว เพราะไม่มีการแจ้งกลับไปยังผู้ส่ง
- หากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึง, หมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message

## UDP Header



- **Source Port Number** : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้
- **Destination Port Number** : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม
- **UDP Length** : ความยาวของดาต้าแกรม หัวส่วนเซดเดอร์ และ Data นั้นหมายความว่า ค่าที่น้อยที่สุดในฟลัต์นี้ คือ 8 ชีบ เป็นขนาดของเซดเดอร์
- **Checksum** : เป็นตัวตรวจสอบความถูกต้องของ UDP Datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

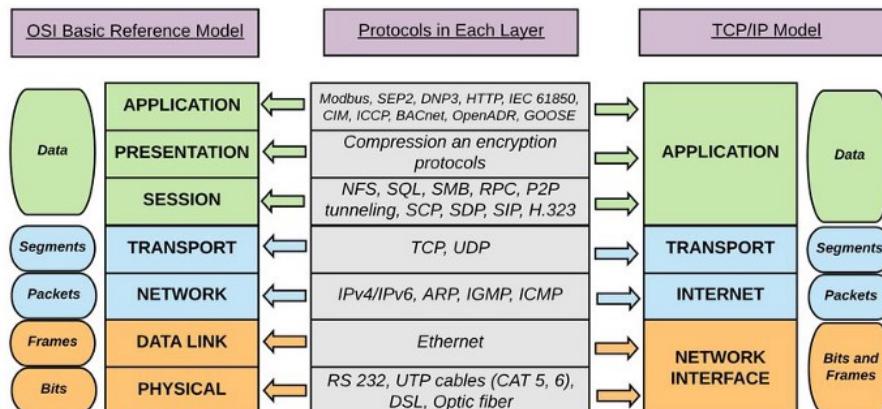
## Layer 4 : Application Layer

- มีโปรโตคอลสำหรับสร้างจ่อเทอร์มินัลเสมือน เรียกว่า TELNET ซึ่งจะช่วยให้ผู้ใช้สามารถติดตอกับเครื่องไฮสต์ทอยู่ไกลออกไปโดยผ่านอินเทอร์เน็ต และสามารถทำงานได้เสมือนกับกำลังนั่งทำงานอยู่ที่เครื่องไฮสต์ตนั้น
- มีโปรโตคอลสำหรับการจัดการแฟ้มข้อมูล เรียกว่า FTP ซึ่งจะช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่าย หรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องใด ๆ ก็ได้
- มีโปรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า SMTP ซึ่งจะช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

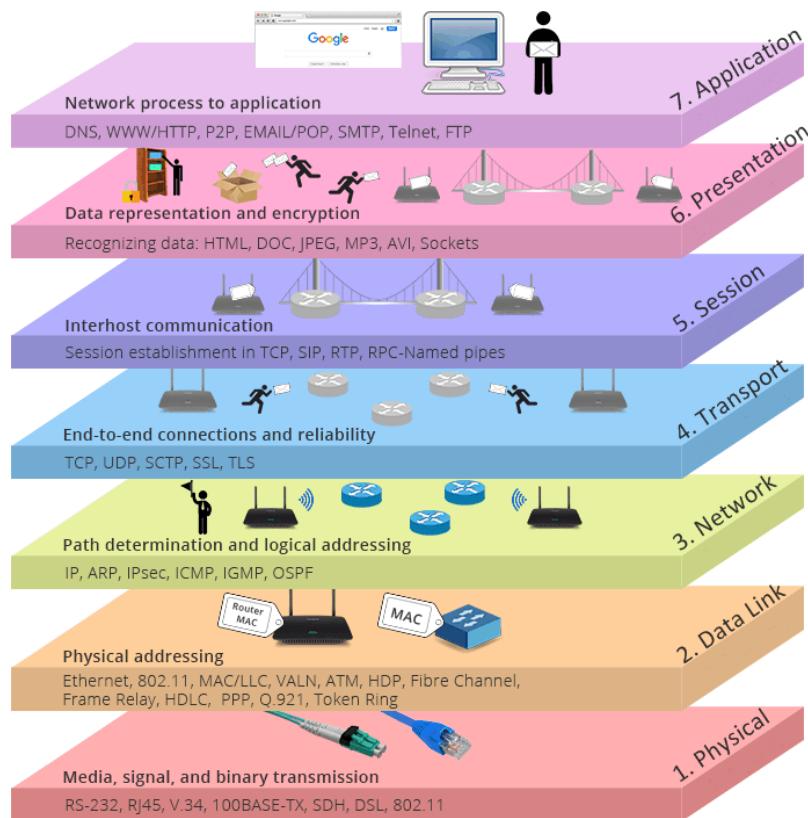
## OSI Model vs TCP/IP Model

1. TCP / IP เป็นรูปแบบไคลเอนต์ - เชิร์ฟเวอร์ (เมื่อไคลเอนต์ร้องขอบริการที่มีให้โดยเชิร์ฟเวอร์) ในขณะที่ OSI เป็นแบบจำลองแนวคิดจับต้องไม่ได้
2. TCP / IP เป็นโปรโตคอลมาตรฐานที่ใช้สำหรับทุกเครือข่ายรวมถึงอินเทอร์เน็ต ในขณะที่ OSI ไม่ได้เป็นโปรโตคอล แต่เป็นรูปแบบการอ้างอิงใช้สำหรับการทำความเข้าใจ และออกแบบสถาปัตยกรรมของระบบ
3. TCP / IP เป็นรูปแบบสื้นในขณะที่ OSI มี 7 Layer

TCP/IP	OSI Model	Protocols
Application Layer	Application Layer	DNS, DHCP, FTP, HTTPS, IMAP, LDAP, NTP, POP3, RTP, RTSP, SSH, SIP, SMTP, SNMP, Telnet, TFTP
	Presentation Layer	JPEG, MIDI, MPEG, PICT, TIFF
	Session Layer	NetBIOS, NFS, PAP, SCP, SQL, ZIP
Transport Layer	Transport Layer	TCP, UDP
Internet Layer	Network Layer	ICMP, IGMP, IPsec, IPv4, IPv6, IPX, RIP
Link Layer	Data Link Layer	ARP, ATM, CDP, FDDI, Frame Relay, HDLC, MPLS, PPP, STP, Token Ring
	Physical Layer	Bluetooth, Ethernet, DSL, ISDN, 802.11 Wi-Fi



## OSI Model (OSI Layer)

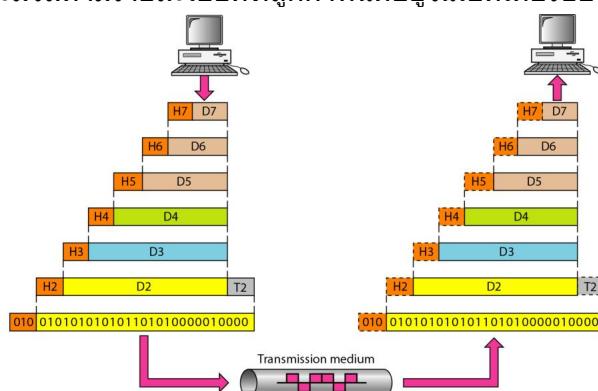


## Concept

- ISO (International Standards Organization) เริ่มร่างขึ้นในปี 1970 และปล่อยให้ใช้จริงปี 1984
- ให้ระบบที่แตกต่างกันสามารถสื่อสารกันได้ โดยรับผิดชอบหน้าที่ของตัวเองให้มีประสิทธิภาพที่สุด
- มีการแบ่งแยกกันรับผิดชอบหน้าที่ต่าง ๆ แยกกันอย่างชัดเจน (ผู้ผลิตแต่ละรายไม่สามารถสร้างองค์ประกอบทุกอย่างได้เองทั้งหมด) โดยแบ่งเป็น 7 Layers คือ Physical, Data Link, Network, Transport, Session, Presentation และ Application

## Encapsulation & Decapsulation

- เมื่อข้อมูลถูกส่งจากผู้ใช้งาน ข้อมูลจะถูกประมวลผลตามลำดับต่อไป โดยจะมีการเพิ่มข้อมูลรายละเอียดการทำงานของแต่ละเลเยอร์ลงในเขตเดอร์ (ยกเว้นแล耶อร์ที่ 2 ที่มีการเพิ่มส่วนหางที่เป็นส่วนเช็คความถูกต้องเพิ่มเข้ามาด้วย) ซึ่งการกระทำนี้เรียกว่า “Encapsulation”
- ในทางกลับกันผู้รับข้อมูล เมื่อได้ข้อมูลก็จะทำการ “Decapsulation” คือการถอดเขตเดอร์ออกตามแต่ละเลเยอร์ เพื่อประมวลผลรายละเอียดที่ถูกกำหนดโดยในเขตเดอร์ของแต่ละเลเยอร์



# Layer 1 : Physical Layer

## Data Unit

- Bit

## Concept

- รับผิดชอบการรับ-ส่งข้อมูลผ่านตัวกลาง (Physical Medium) ในรูปแบบของสายข้อมูลเป็นบิต (Bit Streams)
- มีอุปกรณ์ที่เกี่ยวข้อง เช่น Hub, Repeater เป็นต้น

## Responsibility

- กำหนดมาตรฐานตัวกลาง หรืออินเตอร์เฟซต่าง ๆ ดังนี้
  - สายสัญญาณต่าง ๆ เช่น
    - ↪ สายโทรศัพท์ (ADSL/VDSL)
    - ↪ สายคู่บิดเกลียว (Twisted Pair, LAN)
    - ↪ สายเคเบิล (Coaxial cable, DOCSIS)
    - ↪ สายไฟเบอร์ (Fiber Optic)
  - หัวต่าง ๆ ของสายสัญญาณ เช่น
    - ↪ หัวสายโทรศัพท์ (RJ11)
    - ↪ หัวสายแลน (RJ45)
    - ↪ หัวสายเคเบิล (เช่น RG, BNC)
    - ↪ หัวสายไฟเบอร์ (เช่น FC, ST, SC, MT-RJ)
  - คลื่นสัญญาณต่าง ๆ เช่น
    - ↪ คลื่นวิทยุ (AM-FM Radio)
    - ↪ NFC
    - ↪ Bluetooth
    - ↪ Microwave
    - ↪ Infrared
- เกี่ยวข้องกับหัวข้อต่อไปนี้ (สามารถศึกษาเพิ่มเติมทีหลังเองได้)
  - Signal and Data Rate (Digital signal, Analog signal)
  - Signal Transmission (e.g. PCM, Line coding, Encode/Modulation)
  - Bandwidth Utilization and Multiplexing
  - Transmission Mode (Simple, Half-Duplex, Full-duplex)
  - Bit Synchronization
  - Line configuration & Topology

## Protocol

จะมีโปรโตคอลที่ใช้ควบคุมการรับ-ส่งข้อมูลในレイเยอร์นี้ยกตัวอย่าง เช่น Ethernet Physical Layer (IEEE802.3), LoRa เป็นต้น

## Layer 2 : Data Link Layer

### Data Unit

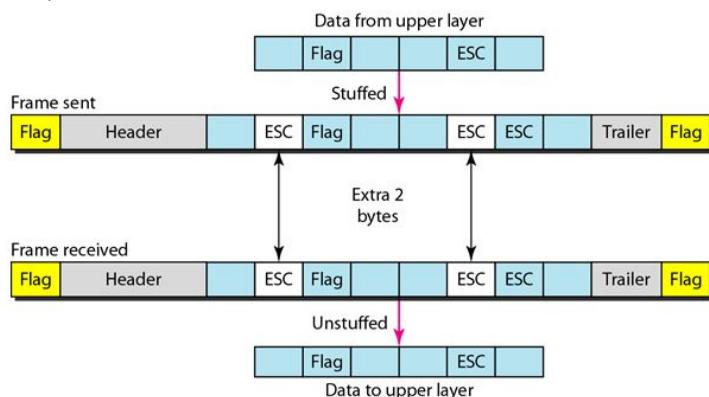
- Frame

### Concept

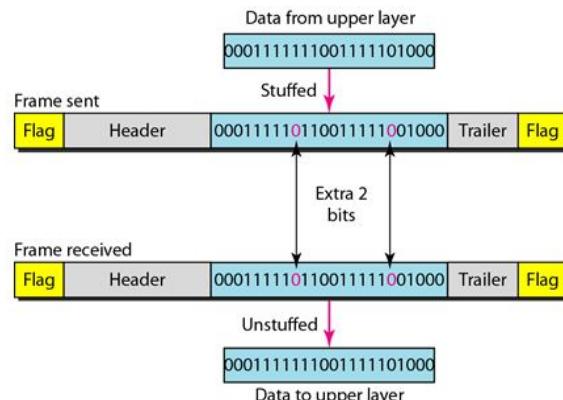
- รับผิดชอบการรับ-ส่งข้อมูลข้อมูลแบบ Hop-to-Hop (อุปกรณ์ที่ใช้เชื่อมต่อระหว่าง 2 Physically Nodes) มีรับการรับประกัน Node-to-Node Delivery (Frame Error Free) ทั้งในเรื่องความถูกต้อง เพราะมีการทำ Error Checking และในเรื่องการส่งถึงปลายทางแน่นอน (โหนดปลายทาง)
- นอกเหนือไปยังเบอร์อย่างขนาดแพ็คเกตที่ได้รับจากเลเยอร์ 3 ให้ได้ขนาดที่เหมาะสม
- มีอุปกรณ์ที่เกี่ยวข้อง เช่น Bridge, Switch, Network Interface Card (NIC) เป็นต้น

### Responsibility

- จัดการเฟรมข้อมูล เช่น
  - การแบ่งขนาดให้เหมาะสม
    - ↪ Fixed-Size Framing : แบ่งขนาดเท่ากันทุกเฟรม
    - ↪ Variable-Size Framing
      - Character-oriented Protocol : มีการใส่ Flag ที่หัวและท้ายเฟรม เพื่อที่จะได้ว่าใน 1 เฟรมยาวเท่าไร โดยจะมีการเพิ่มข้อมูล (ESC) ขนาด 1 ไบต์หากข้อมูลตรงกับ Flag และหากข้อมูลในเฟรมตรงกับ ESC อีก็จะทำการเพิ่ม ESC ข้าไปอีก ซึ่งขั้นตอนเหล่านี้เรียกว่า Stuffed ในทางกลับกันถ้าเจอ Flag หรือ ESC 1 ตัวก็จะทำการถอดทิ้ง (Unstuffed) ถ้าเจอ 2 ตัว จะทำการถอดให้เหลือ 1 ตัว



- Bit-oriented Protocol : ทำการกำหนด Flag เช่น ให้บิต 1 จำนวน 6 บิตเป็น Flag เมื่อเจอบิต 1 ติดกัน 5 บิต จะทำการเติม 0 แทรก (Stuffed) ทำให้ประยัดขนาดข้อมูลมากกว่าการเติมข้อมูลทั้งในตัว



- การเพิ่ม Header และ Trailer

- มี Physical Addressing คือ MAC Address
  - เป็นเลขฐาน 16 จำนวน 12 หลัก โดยปกติมักจะคั่นด้วย : หรือ - ทุก ๆ 2 หลัก นอกจากนั้นยังคั่นด้วย . ทุก ๆ 4 หลักได้ เช่น 08:0B:F0:AF:DC:09, 080B.F0AF.DC09
  - ในทางทฤษฎี MAC Address ถูกกำหนดโดยผู้ผลิต เปเลี่ยนไม่ได้
  - ประกอบด้วย 2 ส่วน คือส่วนแรกที่ OUI กำหนดให้ผู้ผลิตแต่ละราย และส่วนที่ผู้ผลิตสูงให้กับอุปกรณ์แต่ละชิ้นที่ผลิตขึ้นมา
- ทำ Flow Control เพื่อป้องกันเกิดการแออัดของข้อมูลในฝั่ง End-user ที่ทำให้บัฟเฟอร์เต็ม เช่น
  - Simplest (Error Free and No limit buffer)
  - Stop and Wait
  - Stop and Wait ARQ
  - Go Back N ARQ
  - Selective Repeat ARQ
- ทำ Error Control
  - Error Detection : ตรวจสอบข้อมูลว่ามีความผิดพลาดหรือไม่ ถ้ามีจะทำการร้องขอให้ส่งข้อมูลใหม่อีกรอบ (Automatic Repeat Request, ARQ) เช่น
    - ↪ Checksum
    - ↪ Block Coding Techniques
    - ↪ Cyclic Code & Analysis
      - Polynomials
      - Cyclic Redundancy Check
      - Hardware Implementation (Divisor, Augmented Dataword, Remainder)
    - ↪ Simple Parity-Check Code (Even or Odd)
    - ↪ Two-Dimensional Parity-Check Code (Even or Odd)
  - Error Correction : ตรวจสอบข้อมูลว่ามีความผิดพลาด ถ้าหากมีความผิดพลาดจะทำการซ่อมแซมให้ถูกต้อง และไม่สามารถแก้ความผิดพลาดของข้อมูลได้ทั้งหมด ซึ่งถือเป็นข้อจำกัดของการทำ Error Correction เช่น
    - ↪ Linear Block Codes เช่น Hamming codes
  - การทำ Block Code คือการแบ่งข้อมูลเป็นบล็อก (Block) แต่ละบล็อกจะเรียกว่า Dataword หลังจากนั้นจะเพิ่มตัวเช็คความถูกต้องเข้าไปซึ่งจะเรียกว่า Redundant
- ทำ Access Control เช่น Multipoint Connection
  - Random Access Protocol (ALOHA, CSMA, CSMA/CD, CSMA/CA) : ข้อมูลมีโอกาสชน
  - Controlled Access Protocol (Reservation, Polling, Token Passing) : ข้อมูลไม่ชนแน่ ๆ
  - Channelization Protocol (FDMA, TDMA, CDMA) : แบ่งช่องสัญญาณมาใช้ร่วมกัน

## Protocol

จะมีโปรโตคอลที่ใช้ควบคุมการรับ-ส่งข้อมูลในแล耶อร์นี้ยกตัวอย่าง เช่น HDLC, CSMA/CD, Asynchronous Transfer Mode (ATM), Fiber Distributed Data Interface (FDDI), PPP เป็นต้น

## Layer 3 : Network Layer

### Data Unit

- Datagram หรือ Packet

### Concept

- รับผิดชอบการรับ-ส่งข้อมูลข้อมูลแบบ End-to-End (ระหว่าง Host-to-Host) และมีการรับประกันว่าได้ส่งแพ็คเกตออกไปแน่นอน ("ไม่ได้รับประกันว่าจะถึงปลายทาง")
- รวมไปถึงการทำ Addressing, Routing และ Traffic control ซึ่งอยู่ในความรับผิดชอบของเลเยอร์นี้
- มีอุปกรณ์ที่เกี่ยวข้อง เช่น Router, Brouters, 3-Layer Switches, Firewall, Gateway เป็นต้น

### Responsibility

- มี Logical Addressing คือ IP Address (Internet Protocol Address) [ที่งอธิบายในหัวข้อต่อไป](#)
- ทำการ Forwarding เพื่อส่งต่อข้อมูลจากขาเข้าไปยังเส้นทางขาออกที่เหมาะสมตาม Forwarding Table
- ทำ Routing เพื่อเลือกเส้นทางที่ดีที่สุดสำหรับการส่งข้อมูล
  - Static Route
  - Dynamic Route
    - ↪ Interior Gateway Protocols (IGPs)
      - Link State
        - Open ShortestPath First v1 v2 v3 (OSPF)
        - Intermediate System to Intermediate System (IS-IS) : ทำงานตาม OSI ใน เลเยอร์ที่ 2
      - Distance Vector
        - Routing Information Protocol v1 v2 (RIP)
        - Interior Gateway Routing Protocol (IGRP)
        - Enhanced Interior Gateway Routing Protocol (EIGRP)
    - ↪ EGP (Exterior Gateway Protocol)
      - Exterior Gateway Protocol (EGP)
      - Border Gateway Protocol (BGP)
      - The ISO's InterDomain Routing Protocol (IDRP)
- ทำ Connection Setup (มีแค่บาง Network เท่านั้น) เช่น Frame Relay, X.25, Asynchronous Transfer Mode (ATM) เป็นต้น โดยจะใช้หลักการของ Virtual Circuits (VC) ซึ่งประกอบด้วย 3 ขั้นตอน คือ
  - Setup Phase
  - Data Transfer Phase
  - Teardown Phase
- Virtual Circuits (VC) ถือเป็นการเชื่อมต่อแบบ Connection Service และ Datagram Network ถือเป็นการเชื่อมต่อแบบ Connection-Less Service
- นอกจากรับ-ส่งข้อมูลแล้วยังรับผิดชอบการทำ Encapsulation และการทำ Fragmentation

### Protocol

จะมีโปรโตคอลที่ใช้ควบคุมการรับ-ส่งข้อมูลในเลเยอร์นี้ยกตัวอย่าง เช่น EGP, EIGRP, ICMP, IGMP, IPsec, IPX, OSPF, PIM, RIP, WireGuard, Internet Protocol (IP) เป็นต้น

## Layer 4 : Transport Layer

### Data Unit

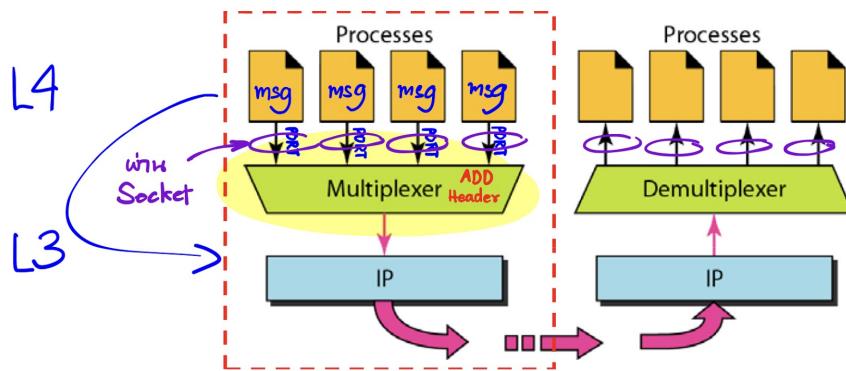
- Segment

### Concept

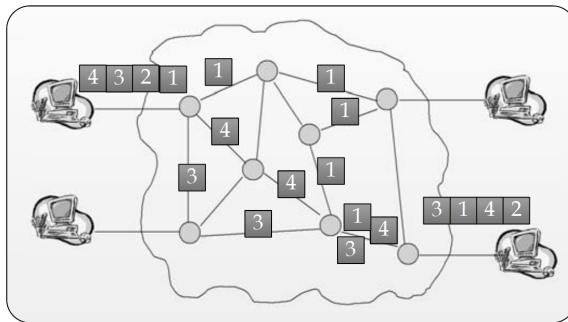
- รับผิดชอบการรับ-ส่งข้อมูลข้อมูลจากโปรเซส (Process) หนึ่งสู่โปรเซสอื่น ๆ และมีรับประกันการส่งข้อความ (Message) ถึงยังปลายทางแน่นอน
- รวมไปถึงการทำ Segmentation, Acknowledgement และ Multiplexing ซึ่งอยู่ในความรับผิดชอบของเลเยอร์นี้
- มีอุปกรณ์ที่เกี่ยวข้อง เช่น Firewall เป็นต้น

### Responsibility

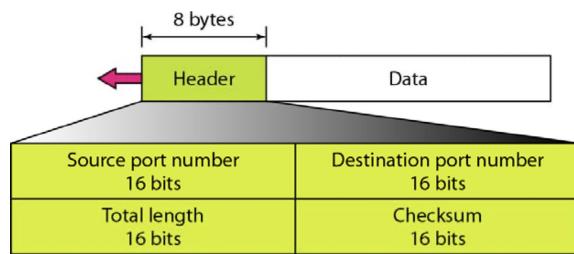
- รับผิดชอบเกี่ยวกับ Port Address โดยทุกแอพพลิเคชันจะมีเลข Port ที่ระบุเฉพาะเจาะจง
  - Port Number
    - ↪ เป็นเลขจำนวนเต็ม 16 บิต อยู่ในช่วง 0 - 65,535
    - ↪ Client จะกำหนดพอร์ตโดยการสุ่ม เรียกว่า Dynamic Ports / Random Ports / Ephemeral Ports / Temporary Ports
    - ↪ Server จะกำหนดเป็นพอร์ตที่แน่นอน ไม่เปลี่ยนแปลงบ่อย ๆ
    - ↪ พอร์ตแบ่งประเภทได้ 3 ประเภท
      - Well-known Ports อยู่ในช่วง 0 - 1,023 กำหนดโดย IANA เช่น
        - 20(FTP), 21(FTP)
        - 23(Telnet)
        - 25(SMTP)
        - 53(DNS)
        - 80(HTTP)
        - 110(POP3)
        - 137-139(NetBIOS)
      - Registered Ports อยู่ในช่วง 1,024 - 49,151 สามารถลงทะเบียนกับ IANA เพื่อป้องกันการใช้งานซ้ำ เช่น
        - 1863(MSN Messenger)
        - 3389(Remote Desktop Port)
        - 5000(Universal Plug and Play, UPnP)
      - Dynamic Ports อยู่ในช่วง 49,152 - 65,535 เป็นพอร์ตที่คระจะใช้ก็ได้ ไม่ต้องลงทะเบียน ซึ่งอาจจะเกิดการใช้งานซ้ำกันได้
    - Socket Address
      - ↪ เป็นการนำ IP Address และ Port Number ผสมกันโดยใช้ ‘:’ เช่น 192.168.1.1:8080
      - ↪ Client Socket Address และ Server Socket Address เป็นส่วนประกอบของ Transport Layer Protocol Header ที่มี Port Number อยู่ด้านใน และ IP Header ที่มี IP Address อยู่ด้านใน
    - ทำ Segmentation และ Reassembly
      - Segmentation คือการแบ่งข้อมูล (Message) ที่ได้มาจากการเลเยอร์ 5 ออกเป็น Segment ที่เล็กลงในฝั่งต้นทาง โดยทำ Multiplexing และเมื่อทำเสร็จเรียบร้อยแล้ว Datagram จะต้องมี 1 Segment เท่านั้น (มี 1 Source Port และ 1 Destination Port)
      - ทำ Reassembly คือการรวมแต่ละเซกเมนต์ (Segment) ให้เป็นข้อมูลในฝั่งปลายทาง โดยทำ Demultiplexing



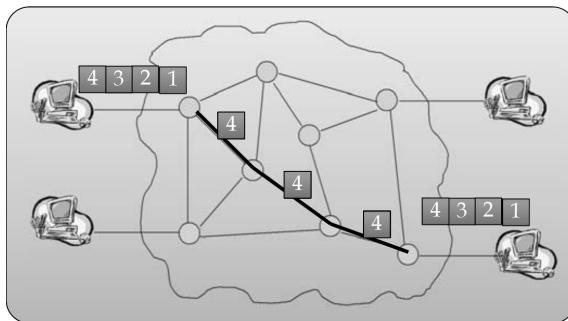
- ทำ Connection Control
- Connectionless



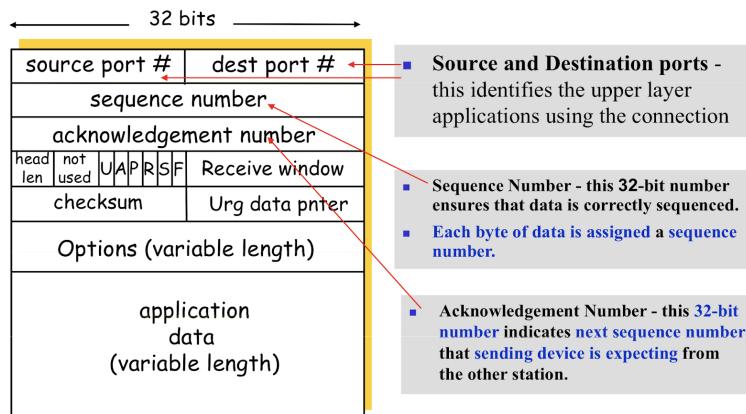
- ↪ ข้อมูลถึงปลายทางแบบไม่เรียงลำดับ
- ↪ เป็นการเชื่อมต่อที่ปลายทางไม่ต้องพร้อมใช้งาน เช่น E-mail (จะเปิดดูตอนไหนก็ได้)
- ↪ เช่น โปรโตคอล UDP ใน Header จะเก็บแค่ Destination IP Address และ Destination Port เท่านั้น ดังนั้นจึงอาจมีมากกว่า 1 Client ที่ใช้ Socket เดียวกัน เพราะปลายทางเดียวกัน โดยภาพด้านล่างแสดงถึง Header ของ UDP



- Connection-oriented



- ↪ ข้อมูลถึงปลายทางแบบเรียงลำดับ
- ↪ เป็นการเชื่อมต่อที่ปลายทางต้องพร้อมใช้งาน เช่น Voice Calling
- ↪ เช่น โปรโตคอล TCP ใน Header จะเก็บ Destination IP Address, Destination Port, Source IP Address และ Source Port ดังนั้นแต่ละ Socket จะมี Client เดียว เพราะ Source IP Address ต่างกัน



- ทำ Error Control โดยการทำ Error Detection และ Error Correction ทั้งข้อความ เช่น การทำ UDP Checksum

153.18.8.105		
171.2.14.10		
All 0s	17	15
1087	13	
15	All 0s	
T	E	S
I	N	G
All 0s		

10011001 00010010	→ 153.18
00001000 01101001	→ 8.105
10101011 00000010	→ 171.2
00001110 00001010	→ 14.10
00000000 00010001	→ 0 and 17
00000000 00001111	→ 15
00000100 00111111	→ 1087
00000000 00001101	→ 13
00000000 00001111	→ 15
00000000 00000000	→ 0 (checksum)
01010100 01000101	→ T and E
01010011 01010100	→ S and T
01001001 01001110	→ I and N
01000111 00000000	→ G and 0 (padding)
10010110 11101011	→ Sum
01101001 00010100	→ Checksum

- ทำ Flow Control โดย

- ควบคุมความเร็วการรับ-ส่งข้อมูล จากบัฟเฟอร์ (Receive Window ในแต่ละ Segment)
- ควบคุมความคับคั่งในเครือข่าย (Congestion Control) เช่น ABR Congestion Control, ATM, TCP Congestion Control เป็นต้น

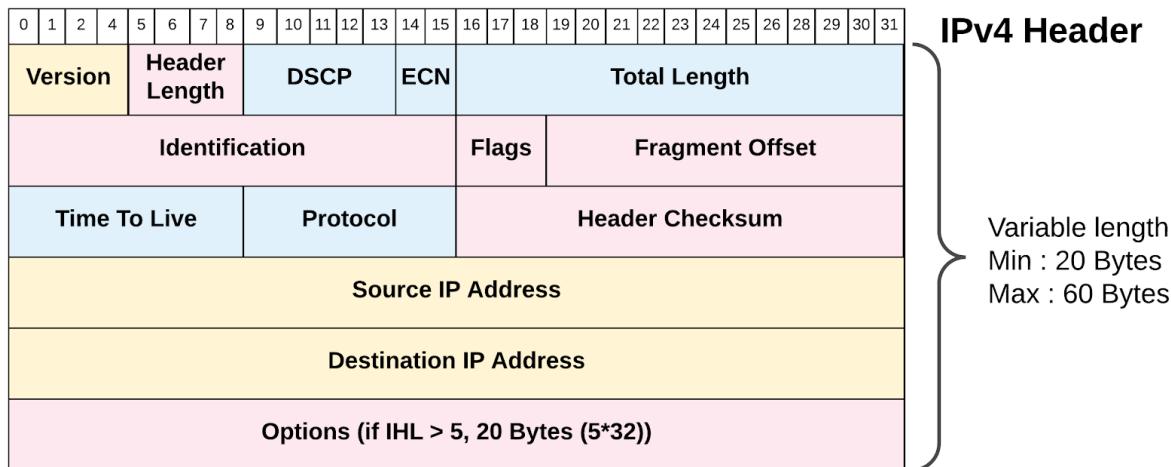
## Protocol

จะมีโปรโตคอลที่ใช้ควบคุมการรับ-ส่งข้อมูลในเลเยอร์นี้ยกตัวอย่าง เช่น MPTCP, UDP, TCP, RDP, RUDP เป็นต้น

## IP Address (Internet Protocol Address)

IP คือเลขที่ระบุตัวตนของอุปกรณ์ใด ๆ ที่เชื่อมต่ออยู่บนเครือข่าย ถ้าเปรียบเทียบกับชีวิตประจำวัน IP จะเปรียบเสมือนเป็นเลขที่บ้าน ใช้สำหรับใช้ในการติดต่อในโลกอินเตอร์เน็ต ตั้งนั้นอุปกรณ์ที่เชื่อมต่อกันในอินเตอร์เน็ตจะได้ต้อง IP Address ที่ไม่ซ้ำกัน โดย IP มี 2 เวอร์ชัน คือ IPv4 และ IPv6

### IPv4



### Characteristic

- IPv4 เป็นตัวเลขฐานสอง 4 ชุด ขนาด 32 บิต แต่ละชุดประกอบด้วยเลขฐานสอง 8 บิต ถูกคั่นด้วย . นิยมเขียนในรูปแบบเลขฐานสิบ เช่น 10.10.10.10, 192.168.1.1 เป็นต้น
- IPv4 มีส่วนประกอบสองส่วน คือ
  - Network(N) คือส่วนของเลขเครือข่าย ถ้าเปรียบเทียบกับชีวิตประจำวัน IP จะเปรียบเสมือน เป็นหมู่บ้าน
  - Host(H) คือส่วนของเครื่องที่เชื่อมต่ออยู่ในเครือข่ายนั้น ๆ ถ้าเปรียบเทียบกับชีวิตประจำวัน IP จะเปรียบเสมือนเป็นบ้านแต่ละบ้านที่อยู่ในหมู่บ้าน (อยู่ใน Network)

### Classfull Addressing

- IPv4 โดยทั่วไปจะแบ่งตามคลาส (Class-full Addressing) ได้ดังตารางด้านล่างนี้

Class	1st Octet (Dec)	1st Octet bits	Network(N) and Host(H)	Default Subnet Mask	Number of Network and Host
A	<u>1-127</u>	<u>00000000-</u> <u>01111111</u>	<u>N.H.H.H</u>	<u>255.0.0.0</u>	<u>128 Nets (2<sup>7</sup>)</u> 16,777,214 Hosts (2 <sup>24</sup> -2)
B	<u>128-191</u>	<u>10000000-</u> <u>10111111</u>	<u>N.N.H.H</u>	<u>255.255.0.0</u>	<u>16,384 Nets (2<sup>14</sup>)</u> 65,534 Hosts (2 <sup>16</sup> -2)
C	<u>192-223</u>	<u>11000000-</u> <u>11011111</u>	<u>N.N.N.H</u>	<u>255.255.255.0</u>	<u>2,097,150 Nets (2<sup>21</sup>)</u> 254 Hosts (2 <sup>8</sup> -2)
D	<u>224-239</u>	<u>11100000-</u> <u>11101111</u>	#NA (Multicast)		
E	<u>240-255</u>	<u>11110000-</u> <u>11111111</u>	#NA (Reserved)		

- บางอุปกรณ์ที่ใช้งานภายใน และไม่จำเป็นต้องเชื่อมต่ออินเตอร์เน็ต เช่น เชิร์ฟเวอร์ภายในบริษัท ทำให้ไม่จำเป็นต้องมี IP Address บนโลกอินเตอร์เน็ตได้ จึงได้มีการกำหนด IPv4 Private Address สำหรับเครือข่ายภายใน (Private Network) ดังตารางด้านล่างนี้

Class	IP Address Range	Number of Addresses	Host ID size	Mask bits	Largest CIDR block (subnet mask)
A	10.0.0.0 – 10.255.255.255	16,777,216	24 bits	8 bits	10.0.0.0/8 (255.0.0.0)
B	172.16.0.0 – 172.31.255.255	1,048,576	20 bits	12 bits	172.16.0.0/12 (255.240.0.0)
C	192.168.0.0 – 192.168.255.255	65,536	16 bits	16 bits	192.168.0.0/16 (255.255.0.0)

## Subnets Mask

### Characteristic

- IPv4 เป็นตัวเลขฐานสอง 4 ชุด ขนาด 32 บิต แต่ละชุดประกอบด้วยเลขฐานสอง 8 บิต ถูกคั่นด้วย . นิยมเขียนในรูปแบบเลขฐานสิบ เช่น 255.0.0.0, 255.255.0.0, 255.255.255.0 เป็นต้น
- ทำหน้าที่ระบุบิตที่เป็น Network(N) ให้เป็นบิต 1 และระบุบิตที่เป็น Host(H) ให้เป็นบิต 0 โดย
  - เลขฐานสองทุกบิตที่อยู่ฝั่งซ้าย หรือ Network(N) ต้องมีค่าเป็น 1 เสมอ (มี 0 แทรกไม่ได้)
  - เลขฐานสองทุกบิตที่อยู่ฝ่ายขวา หรือ Host(H) ต้องมีค่าเป็น 0 เสมอ (มี 1 แทรกไม่ได้)

## Classless Addressing

- การแบ่งไอพีแอดเดรสแบบ Classfull Addressing มีข้อจำกัดอยู่มาก
  - อาจทำให้เกิดการใช้งานไอพีแอดเดรสอย่างไม่มีประสิทธิภาพ เพราะเกิดการสูญเสียไอพีแอดเดรสที่ไม่ได้ใช้งาน
  - เช่น แผนกหนึ่งในบริษัทด้วยการให้อุปกรณ์เชื่อมต่อกับอินเตอร์เน็ต 10 เครื่อง หากแบ่งตาม Class A, B หรือ C ก็จะเกิดไอพีแอดเดรสที่ไม่ได้ใช้งานจำนวนมาก และไม่สามารถนำไอพีแอดเดรสที่ไม่ได้ใช้งานไปใช้กับแผนกอื่นได้
- เป็นการทำชั้นเน็ต (Subnet) โดยยึดบิตบางส่วนของ Host(H) มาเป็นบิตของ Network(N) เพื่อสร้างชั้นเน็ตที่ยืดหยุ่นกว่า (สามารถปรับจำนวน Host ที่อยู่ในแต่ละชั้นเน็ตให้น้อยลงได้)

## Classless InterDomain Routing (CIDR)

- นิยมใช้ควบคู่กับการแบ่งไอพีแอดเดรสแบบ Classless Addressing
- เป็นวิธีการในการจัดการ และระบุตำแหน่งบิตที่เป็น Network(N) ซึ่งยืดหยุ่นมากกว่าระบบเดิมที่ใช้เลข Subnets Mask
- ใช้แทน Subnets Mask ด้วยการเพิ่มสัญลักษณ์ '/' และตามด้วยขนาดของ Network(N) Mask เช่น 128.10.0.0/16 จะมี 16 บิตแรกเป็น Network(Prefix) และมี Host(Suffix) คือ 16 บิตหลัง

## Variable Length Subnet Masks ( VLSM )

- จากหลักการเครือข่ายที่ใช้งาน ไม่จำเป็นจะต้องมีขนาดเท่ากันเสมอไป (ไม่จำเป็นต้องมีตัว Mask เท่ากัน) เช่น
  - การเชื่อมต่อแบบจุดต่อจุด (Point-to-Point) ต้องการแค่ 2 IP ก็เพียงพอ ดังนั้นควรใช้ Subnet Mask เป็น 255.255.255.252 (Mask 30 bit หรือ /30)
  - การเชื่อมต่อภายใน LAN ที่มีเครื่องเพียง 20 เครื่อง ดังนั้นควรใช้ Subnet Mask เป็น 255.255.255.224 (Mask 27 bit หรือ /27)
- ประโยชน์ของการใช้ VLSM มีดังนี้
  - VLSM จะยอมให้มีการแบ่ง Subnet ได้มากกว่า 1 ครั้งสำหรับแต่ละชุด IP เพื่อให้ได้ขนาดไอพีแอดเดรสตามที่ต้องการ
  - VLSM จะช่วยลดจำนวนการจัดสรรและเรสลง เป็นการใช้งานแอดเดรสอย่างมีประสิทธิภาพ
  - VLSM ยังช่วยให้เราเตอร์ทำงานได้เร็วขึ้น เนื่องจากขนาดของ Routing Table เล็กลง

## [การบ้าน] Let's try!!

0. สถานที่ได้รับไอพีแอดเดรสในช่วง 161.246.0.0/16 มาใช้งาน หากต้องการนำไปใช้กับตึก ECC ดังนี้

- 0.1 [126 IP] ขั้น 5 ห้อง
- 0.2 [50 IP] ห้อง 701
- 0.3 [64 IP] ขั้น 8 ห้อง
- 0.4 [6 IP] ห้องธุรกิจ

จะต้องกำหนดบิตของ Host(H) ให้เหมาะสมกับแต่ละสถานที่ โดยหลักสากลให้คำนวนกลุ่มใหญ่ที่ต้องการใช้ IP มาก ใช้ IP เลขน้อยก่อน และวิจัยเรียงลงมาเรื่อยๆ

→ ขั้น 5 ห้อง จะกำหนดทั้งหมด Host(H) ทั้งหมด 7 บิต

- Host(H) 7 บิต สามารถมี IP ได้  $2^7 = 128$  IP (หากใช้ 6 บิต  $2^6 = 64$  IP >> "ไม่พอใช้")
- IP ที่ใช้งานจริงได้มีทั้งหมด  $2^7 - 2 = 126$  IP (ตัด NetworkID และ Broadcast Address)
- เนื่องจากมีบิต Host(H) 7 บิต ดังนั้นจะมีบิต Network(N) =  $32 - 7 = 25$  บิต
- สามารถเขียน NetworkID ได้เป็น 161.246.0.0/25 (IP กลุ่มใหญ่ ใช้ IP เลขน้อยก่อน)
  - ↪ NetworkID : 161.246.0.0, Subnet Mask : 255.255.255.128
  - ↪ Broadcast Address : 161.246.0.127
  - ↪ Available Host Address (126) : 161.246.0.1 (First) - 161.246.0.126 (Last)

→ ขั้น 8 ห้อง จะกำหนดทั้งหมด Host(H) ทั้งหมด 7 บิต

- หากใช้บิต Host(H) 6 บิต  $2^6 = 64$  IP จะไม่พอใช้ เนื่องจากยังไม่รวม NetworkID และ Broadcast Address
- สามารถเขียน NetworkID ได้เป็น 161.246.0.128/25 (IP กลุ่มใหญ่รองลงมา ใช้ IP ต่อจากกลุ่มแรก)
  - ↪ NetworkID : 161.246.0.128, Subnet Mask : 255.255.255.128
  - ↪ Broadcast Address : 161.246.0.255
  - ↪ Available Host Address (126) : 161.246.0.129 (First) - 161.246.0.254 (Last)

→ ห้อง 701 จะกำหนดทั้งหมด Host(H) ทั้งหมด 6 บิต

- Host(H) 6 บิต สามารถมี IP ได้  $2^6 = 64$  IP (หากใช้ 5 บิต  $2^5 = 32$  IP >> "ไม่พอใช้")
- IP ที่ใช้งานจริงได้มีทั้งหมด  $2^6 - 2 = 62$  IP (ตัด NetworkID และ Broadcast Address)
- เนื่องจากมีบิต Host(H) 6 บิต ดังนั้นจะมีบิต Network(N)  $32 - 6 = 26$  บิต
- สามารถเขียน NetworkID ได้เป็น 161.246.1.0/26
  - ↪ NetworkID : 161.246.1.0, Subnet Mask : 255.255.255.192
  - ↪ Broadcast Address : 161.246.1.63
  - ↪ Available Host Address (62) : 161.246.1.1 (First) - 161.246.1.62 (Last)

→ ห้องธุรกิจ จะกำหนดทั้งหมด Host(H) ทั้งหมด 3 บิต

- สามารถเขียน NetworkID ได้เป็น 161.246.1.64/29
  - ↪ NetworkID : 161.246.1.64, Subnet Mask : 255.255.255.248
  - ↪ Broadcast Address : 161.246.1.71
  - ↪ Available Host Address (6) : 161.246.1.65 (First) - 161.246.1.70 (Last)



2. หากได้รับไอพีแอดเดรสในช่วง 192.168.4.0 - 192.168.7.255 มาแบ่งใช้งานตามห้อง โดยแต่ละห้องมีเงื่อนไขตามด้านล่างนี้ จงแบ่งไอพีแอดเดรสให้เพียงพอต่อการใช้งาน โดยตอบค่าแอดเดรส ต่าง ๆ ดังนี้

2.1 [130 IP] ห้อง A

2.1.1 NetworkID = \_\_\_\_\_ (CIDR Format)  
 2.1.2 NetworkID = \_\_\_\_\_  
 2.1.3 Subnet Mask = \_\_\_\_\_  
 2.1.4 Broadcast Address = \_\_\_\_\_  
 2.1.5 First Host Address = \_\_\_\_\_  
 2.1.6 Last Host Address = \_\_\_\_\_  
 2.1.7 Available Host = \_\_\_\_\_

2.2 [99 IP] ห้อง B

2.2.1 NetworkID = \_\_\_\_\_ (CIDR Format)  
 2.2.2 NetworkID = \_\_\_\_\_  
 2.2.3 Subnet Mask = \_\_\_\_\_  
 2.2.4 Broadcast Address = \_\_\_\_\_  
 2.2.5 First Host Address = \_\_\_\_\_  
 2.2.6 Last Host Address = \_\_\_\_\_  
 2.2.7 Available Host = \_\_\_\_\_

2.3 [270 IP] ห้อง C

2.3.1 NetworkID = \_\_\_\_\_ (CIDR Format)  
 2.3.2 NetworkID = \_\_\_\_\_  
 2.3.3 Subnet Mask = \_\_\_\_\_  
 2.3.4 Broadcast Address = \_\_\_\_\_  
 2.3.5 First Host Address = \_\_\_\_\_  
 2.3.6 Last Host Address = \_\_\_\_\_  
 2.3.7 Available Host = \_\_\_\_\_

2.4 [2 IP] ห้อง D

2.4.1 NetworkID = \_\_\_\_\_ (CIDR Format)  
 2.4.2 NetworkID = \_\_\_\_\_  
 2.4.3 Subnet Mask = \_\_\_\_\_  
 2.4.4 Broadcast Address = \_\_\_\_\_  
 2.4.5 First Host Address = \_\_\_\_\_  
 2.4.6 Last Host Address = \_\_\_\_\_  
 2.4.7 Available Host = \_\_\_\_\_

2.5 [40 IP] ห้อง E

2.5.1 NetworkID = \_\_\_\_\_ (CIDR Format)  
 2.5.2 NetworkID = \_\_\_\_\_  
 2.5.3 Subnet Mask = \_\_\_\_\_  
 2.5.4 Broadcast Address = \_\_\_\_\_  
 2.5.5 First Host Address = \_\_\_\_\_  
 2.5.6 Last Host Address = \_\_\_\_\_  
 2.5.7 Available Host = \_\_\_\_\_

3. ไอพีแอดเดรส 20 หมายเลขดังนี้ จงหาว่ามีทั้งหมดกี่ชั้นเน็ต (Subnet) หรือมีทั้งหมดกี่เครือข่าย (Network กี่วง) และในแต่ละเครือข่ายมีไอพีแอดเดรสในข้อใดอยู่บ้าง

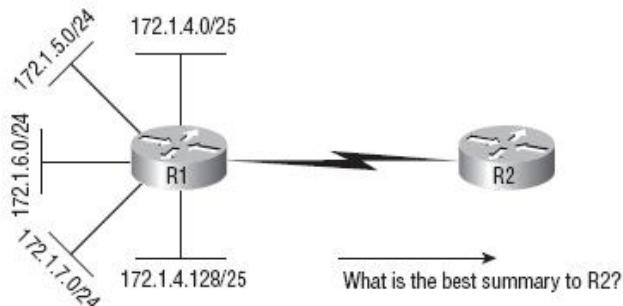
1. 1.1.1.1
  2. 10.10.10.10
  3. 10.10.10.10 (Subnet Mask : 255.0.0.0)
  4. 10.10.10.10/24
  5. 161.246.5.44
  6. 161.246.5.26 (Subnet Mask : 255.255.255.0)
  7. 172.0.0.1
  8. 172.15.100.100
  9. 172.16.100.100
  10. 172.16.32.111/18
  11. 172.16.0.123/17
  12. 172.17.17.17/15
  13. 172.24.25.26/11
  14. 172.31.32.255/12
  15. 172.32.123.1/10
  16. 192.168.1.1
  17. 192.168.1.128 (Subnet Mask : 255.255.255.0)
  18. 192.168.1.128 (Subnet Mask : 255.255.255.128)
  19. 192.168.1.128/25
  20. 192.168.1.200

Total \_\_\_\_\_ Subnet(s)

Subnet 1 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 2 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 3 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 4 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 5 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 6 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 7 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 8 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 9 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 10 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 11 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 12 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 13 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	
Subnet 14 =	>> ประกอบไปด้วยไอพีแอดเดรสข้อ	

## IP Summarization

คือ การรวมรวมไอพีแอดเดรสหลาย ๆ Networks ให้เหลือเพียง Network เดียว โดย Network นั้นจะต้องครอบคลุมไอพีแอดเดรสทั้งหมดที่อยู่ใน Networks ที่ถูกรวมนั้นด้วย เช่น



การทำ Summarization ทำได้โดยการ

- นำ Subnet ทั้งหมดแบ่งเป็นเลขฐานสองทั้งหมด
- ดูเฉพาะตัวเลขที่เหมือนกันไม่จากบิต Network(N) ซ้ายสุดไปทางขวาเรื่อย ๆ หากสิ้นสุดที่ไหน ก็จะทำการ Mask ตรงนั้น

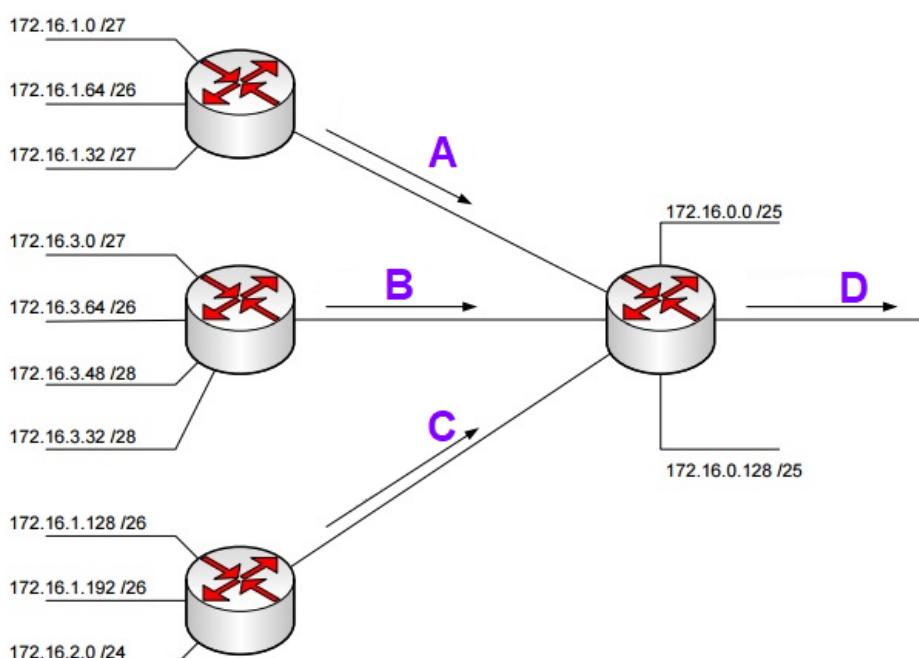
172.1.4.0/25	<b>10101100.00000001.00000100.00000000</b>
172.1.5.0/24	<b>10101100.00000001.00000101.00000000</b>
172.1.6.0/24	<b>10101100.00000001.00000110.00000000</b>
172.1.7.0/24	<b>10101100.00000001.00000111.00000000</b>
<b>172.1.4.128/25</b>	<b>10101100.00000001.00000100.10000000</b>

**Summarization**      **172.1.4.0/22**      **10101100.00000001.00000100.00000000**

ดังนั้นจะได้ Summarization เป็น 172.1.4.0/22 จาก Networks ทั้งหมด เพราะตัวเลขที่เหมือนกันจะสิ้นสุดที่บิตที่ 22 ดังนั้นจะ Mask 22bits (/22)

[การบ้าน] จากรูปด้านล่าง จงทำ IP Summarization ของเครือข่ายที่จุด A, B, C และ D

A = \_\_\_\_\_  
 B = \_\_\_\_\_  
 C = \_\_\_\_\_  
 D = \_\_\_\_\_



## Special IPv4 Address

### Automatic Private Internet Protocol Addressing (APIPA)

- เป็นไอพีที่อยู่ในช่วง 169.254.0.1 - 169.254.255.254
- เป็นไอพีที่ไม่ได้ถูกแจกล่ายโดยเราเตอร์ ซึ่งไม่มีเส้นทาง (Route) ออกไปยังอินเตอร์เน็ต และไม่ได้ทำการเซ็ตที่อยู่เกตเวย์ (Gateway Address)
- ถ้าหาก DHCP ไม่สามารถแจกล่ายไอพีให้อุปกรณ์ได้จะมีการแจก APIPA

### Loopback Address / Localhost Address

- เป็นไอพีที่อยู่ในช่วง 127.0.0.1/8 หรือ 127.0.0.1 - 127.255.255.254 (หมายเลขสุดท้ายของไอพี Class A)
- เป็นไอพีที่ส่วนใหญ่ใช้ Loopback Function คือหากมีการส่งข้อมูลมาจากแล耶อร์ที่สูงกว่า -many IP ในกลุ่มนี้ จะไม่มีการส่งข้อมูลออกไปจากเครื่องนั้น ๆ โดยเด็ดขาด
- เนื่องจากระบบเครือข่ายในอดีตมีราคาแพงมาก ดังนั้นการทดสอบบนอุปกรณ์จริงจึงเป็นไปได้ยาก จึงต้องจำลอง หรือสร้างระบบเพื่อให้ผู้ผลิต ผู้วิจัย หรือผู้ที่ทำงานในแล耶อร์ 4-7 สามารถทำงานได้โดยไม่จำเป็นต้องมีอุปกรณ์จริง

### Network ID / Subnet ID / Network Address / Subnet Address

- เป็น IP Address แรกก่อนของ Subnet ที่ระบุว่าอุปกรณ์อยู่ในเครือข่ายใด (อยู่เน็ตเวิร์คไหน) โดย Network ID จะมีบิตของ Host เป็น 0 หมดทุกบิต **ถ้าเปรียบเทียบกับชีวิตประจำวัน Network ID จะเปรียบเสมือนเป็นชื่อหมู่บ้าน**
- เป็นส่วนที่เราเตอร์ใช้สำหรับหาเส้นทาง โดยถ้าเป็น
  - Network ID ของเราเตอร์นั้น ข้อมูลจะถูกส่งต่อไปยังอุปกรณ์ปลายทางที่อยู่ภายใต้เราเตอร์
  - Network ID ของเราเตอร์อื่น ข้อมูลจะถูกส่งต่อไปยังเราเตอร์ปลายทาง จากนั้นข้อมูลจึงจะถูกส่งต่อไปยังอุปกรณ์ปลายทางต่อไป
- การหา Network ID ให้นำไอพีแล้วเดรสมำทำกำร And(&) ทีละบิตกับ Subnet Mask เช่น
  - **10.10.10.10/8 >> IP : 10.10.10.10, Subnet Mask : 255.0.0.0**  
00001010.00001010.00001010.00001010 (IP)  
&  
11111111.00000000.00000000.00000000 (Subnet Mask)  
00001010.00000000.00000000.00000000 (Network ID) >> **10.0.0.0 (Class A)**
  - **172.3.4.56/16 >> IP : 172.3.4.56, Subnet Mask : 255.255.0.0**  
10101100.00000011.00000100.00111000 (IP)  
&  
11111111.11111111.00000000.00000000 (Subnet Mask)  
10101100.00000011.00000000.00000000 (Network ID) >> **172.3.0.0 (Class B)**
  - **192.168.1.2/24 >> IP : 192.168.1.2/24, Subnet Mask : 255.255.255.0**  
11000000.10101000.00000001.00000010 (IP)  
&  
11111111.11111111.11111111.00000000 (Subnet Mask)  
11000000.10101000.00000001.00000000 (Network ID) >> **192.168.1.0 (Class C)**
  - **172.3.4.56/24 >> IP : 172.3.4.56, Subnet Mask : 255.255.255.0**  
10101100.00000011.00000100.00111000 (IP)  
&  
11111111.11111111.11111111.00000000 (Subnet Mask)  
10101100.00000011.00000100.00000000 (Network ID) >> **172.3.4.0**
  - **192.168.1.222/25 >> IP : 192.168.1.222, Subnet Mask : 255.255.255.128**  
11000000.10101000.00000001.11011100 (IP)  
&  
11111111.11111111.11111111.10000000 (Subnet Mask)  
11000000.10101000.00000001.10000000 (Network ID) >> **192.168.1.128**

## Broadcast Address

- เป็นแอดเดรสที่มีเอาไว้ให้อุปกรณ์สื่อสารไปยังอุปกรณ์ทุก ๆ เครื่องที่อยู่ภายใต้เครือข่ายเดียวกันได้ เช่น ถ้าหากอุปกรณ์เลเยอร์ 2 ต้องการทำ Flooding (ส่งข้อมูลออกทุกพอร์ต ยกเว้นพอร์ตที่รับข้อมูลนั้น ๆ เช่นมา) ซึ่งการ Flooding ของ Broadcast Address จะไปสั่นสุดที่อุปกรณ์เลเยอร์ 3 นั้นก็คือเราเตอร์
- มีเพื่อให้อุปกรณ์สามารถสื่อสารผ่านโปรโตคอล กับอุปกรณ์ปลายทางได้ ๆ ได้แบบ 1 to all เช่น DHCP และ ARP เป็นต้น
- เลเยอร์ 2 คือ MAC Address หมายเลข FF-FF-FF-FF-FF-FF
- เลเยอร์ 3 แบ่งเป็น 2 ประเภท คือ
  - Local Broadcast Address คือ 255.255.255.255
  - Directed Broadcast Address คือ IP Address สุดท้ายก่อนจะเข้า Subnet ตัดไป หรือสามารถสั่งเกตได้จาก Network ID ที่เปลี่ยนบิตของ Host เป็น 1 หมดทุกบิต เช่น
    - ↪ Class A เช่น 1.255.255.255, 10.255.255.255, 100.255.255.255 เป็นต้น
    - ↪ Class.B เช่น 172.16.255.255, 172.17.255.255, 173.0.255.255 เป็นต้น
    - ↪ Class.C เช่น 192.168.1.255, 192.168.2.255, 195.195.0.255 เป็นต้น
    - ↪ Classless ยกตัวอย่าง Subnet ต่อไปนี้
      - 192.168.1.0/25 >> 192.168.1.0111 1111 >> 192.168.1.127
      - 192.168.1.128/25 >> 192.168.1.1111 1111 >> 192.168.1.255

## Default Gateway Address

- เป็น IP Address สำหรับให้อุปกรณ์ต่าง ๆ ในเครือข่ายหนึ่งสามารถสื่อสารกับอุปกรณ์ต่าง ๆ ในเครือข่ายอื่นได้
- ไม่จำเป็นต้องตั้ง Default Gateway Address สำหรับทุกเครือข่าย ถ้าหากเครือข่ายใดไม่ได้ตั้ง Default Gateway Address ก็จะไม่สามารถติดต่อกับเครือข่ายอื่น ๆ ได้

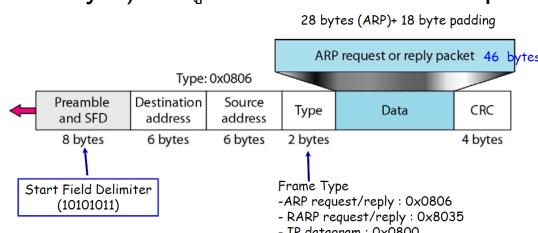
## 0.0.0.0

- ในเชิง Routing
  - คือ Default Route ใน IPv4 ซึ่งจะ Matches กับไอพีทุกตัว
  - สามารถเขียนในรูป CIDR Notation ได้เป็น 0.0.0.0/0
- ในเชิง IP Address มีได้หลายบริบทดังนี้
  - คือ IPv4 ทั้งหมด (นิยมใช้กับการตั้งค่าเซิร์ฟเวอร์ ซึ่งหมายถึงเครื่องทั้งหมด Local)
  - คือ IPv4 ที่จะถูกกำหนดให้มีอิเมร์ทำ DHCP (ขณะที่ยังไม่ได้รับการแจกล้ายไอพี)
  - คือ IPv4 ที่จะถูกกำหนดให้มีอิเมร์ทำ DHCP ไม่สำเร็จ (ในเครื่องสมัยใหม่จะใช้ APIPA แทน)
  - คือทางที่ใช้ในการกำหนดเป้าหมายที่ไม่พร้อมใช้งาน
  - คือทางที่ใช้ในการกำหนดเส้นทางที่ร้องขอ ไปยังเป้าหมายที่ไม่มีอยู่จริงแทนเป้าหมายเดิม (ซึ่งนิยมใช้กับพวก Adblock)

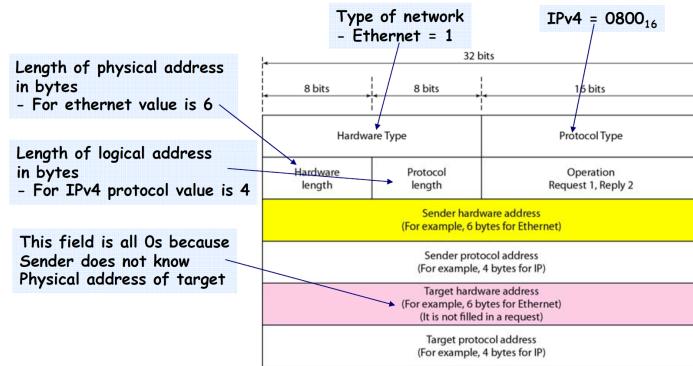
## Address Mapping & Address Translation

### Address Resolution Protocol (ARP)

ในการส่งแพ็คเกตถึงโฮสต์ หรือเราเตอร์จะใช้ Logical Address และ Physical Address จึงต้องมีการแปลง Logical Address ให้สัมพันธ์กับ Physical Address โดยใช้ ARP ในการ Mapping (ARP เป็นโปรโตคอลใน Data Link Layer) จากรูปด้านล่างเป็นการ Encapsulating ของ ARP Packet

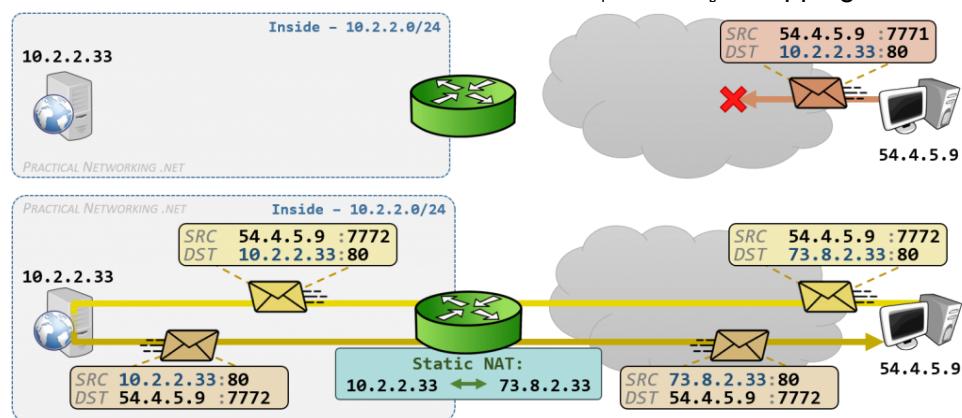


## ARP Packet

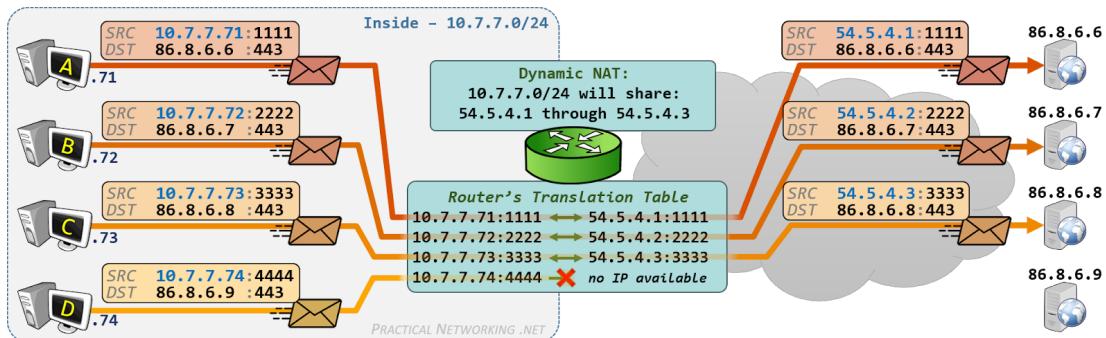


## Network Address Translation (NAT)

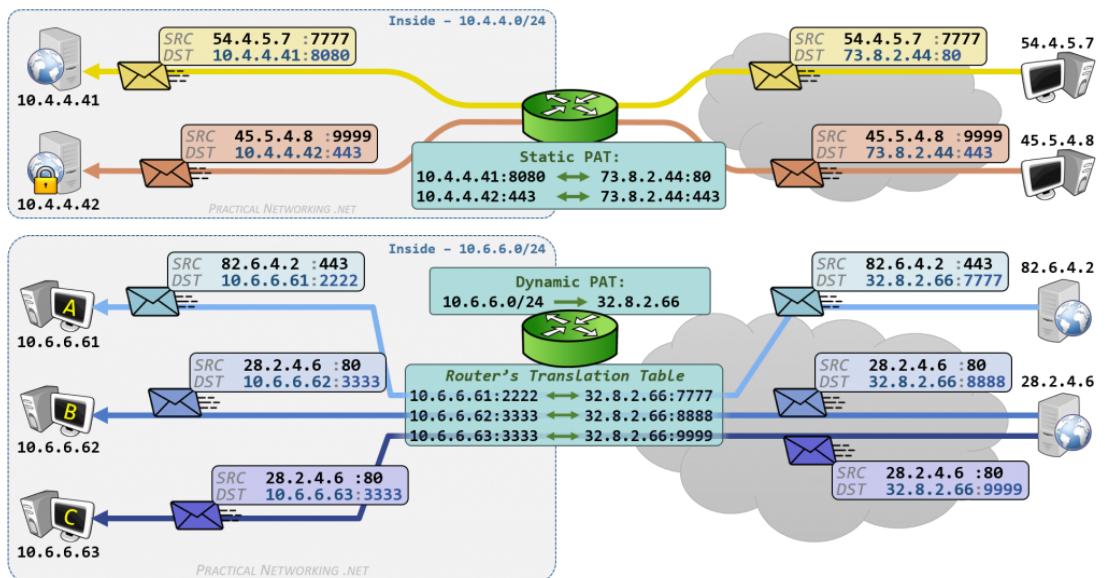
- เป็นวิธีการหนึ่งในการแปลง และแปลงไอพีแอดเดรสของอุปกรณ์ในระบบเครือข่ายภายในซึ่งในที่นี้จะเรียกว่า ไอพีส่วนตัว (Private IP Address) ให้เป็นไอพีแอดเดรสซึ่งเป็นที่ย้อมรับ และสื่อสารบนอินเตอร์เน็ต ซึ่งเรียกว่า ไอพีสาธารณะ (Public IP Address) โดยมี NAT Device ทำหน้าที่ในการแปลงไอพี เช่น เราระดับ เป็นต้น โดยการ NAT มีประโยชน์ดังนี้
  - ทำให้มี IPv4 เพียงพอต่อการใช้งาน
  - สามารถซ่อนไอพีแอดเดรสของอุปกรณ์ที่อยู่ภายในระบบเครือข่าย ไม่ให้ถูกค้นจากอินเตอร์เน็ตภายนอกได้ ซึ่งวิธีการนี้จะเรียกว่า IP Masquerading
  - IP Masquerading จะทำการซ่อน Private IP Address ที่อยู่ในระบบให้อยู่หลัง Public IP Address เดียว เพื่อทำให้มีความปลอดภัย
- NAT มีการแบ่งประเภทออกเป็น 3 ประเภทหลัก ๆ ดังนี้
  - Static NAT เป็นการทำ One-to-One Mapping ระหว่าง 1 Private IP Address กับ 1 Public IP Address โดย NAT ประเภทนี้จะมี
    - ↪ ข้อเสีย : มีเส้นทางที่ตายตัว (ไม่เปลี่ยนแปลง) ใช้แอดเดรสจำนวนมาก และต้องมีการดูแลจัดการการ Mapping
    - ↪ ข้อดี : สามารถจำกัดความต้องการใช้งาน และติดตามอุปกรณ์ที่ถูก Mapping ได้ง่าย



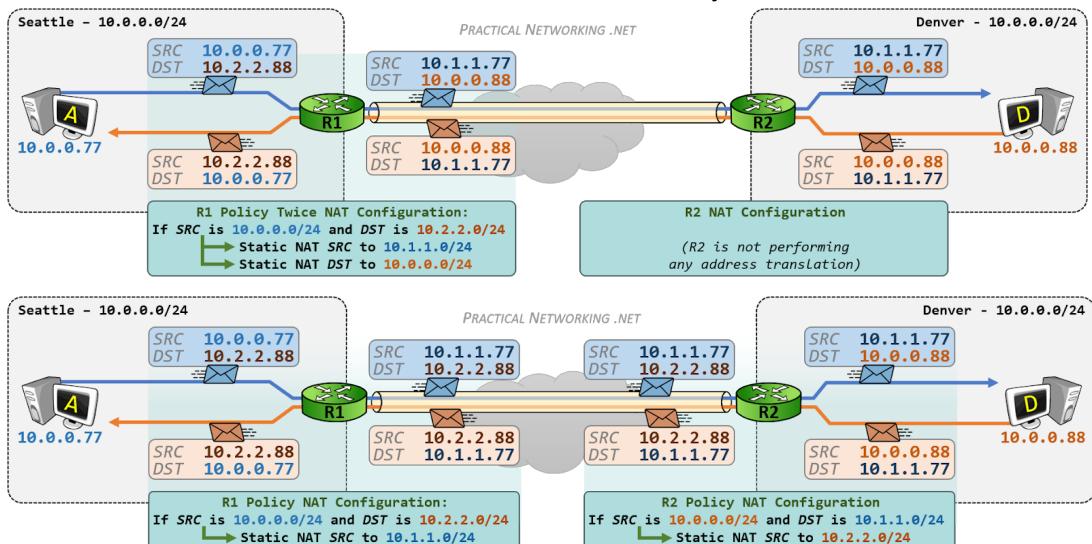
- Dynamic NAT เป็นการทำ One-to-One Mapping ระหว่าง 1 IP จากกลุ่มของ Public IP Address ที่ว่าง (เรียกว่า NAT Pool ซึ่งอาจจะมีมากกว่า 1 IP) กับ 1 Private IP Address ของเครื่องที่ต้องการติดต่อกับอินเตอร์เน็ตภายนอก เมื่อใช้งานเสร็จจะทำการคืน Public IP Address ลงใน NAT Pool โดย NAT ประเภทนี้จะมี
  - ↪ ข้อดี : สามารถใช้กับหลายเครื่องได้มากขึ้น
  - ↪ ข้อดี : ทำการตรวจสอบกลับหากขึ้น (ต้องตรวจสอบกับ Log ตามเวลา)



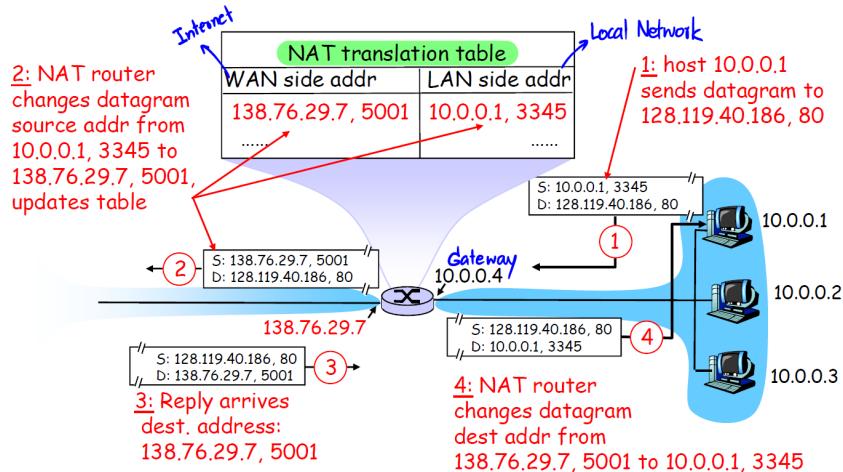
- **Overloading NAT หรือ PAT (Port Address Translation)** เป็นการ Mapping ระหว่างหลาย Private IP Address กับ 1 Public IP Address โดยใช้พอร์ตในการแยกแต่ละ Private IP Address ยกตัวอย่างการใช้งาน เช่น การเปิด Web Server ผ่านไอพีเดียว แต่ใช้หลายพอร์ต ( เช่น พอร์ต 3000 สำหรับ Front-end และพอร์ต 5900 สำหรับ Back-end )



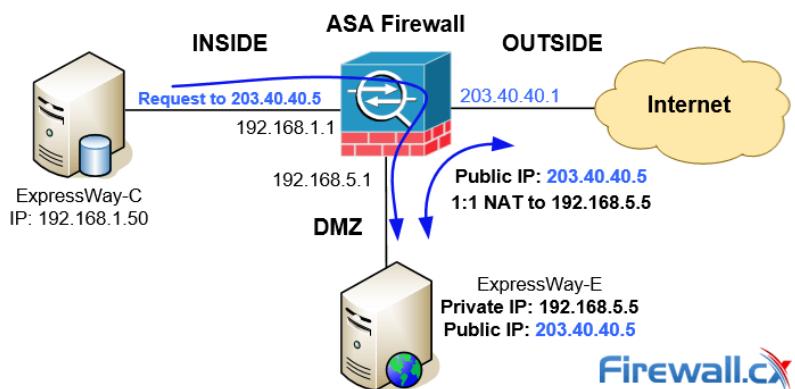
- **Overlapping NAT** เป็นการเปลี่ยนไอพีในเครือข่ายเดิมให้เป็นไอพีใหม่ที่ไม่ซ้ำกัน เช่น เมื่อเปลี่ยนเครือข่าย (เช่น เปลี่ยน ISP) ไอพีเดิมในระบบเครือข่ายเก่าก็สามารถใช้ได้จากเครือข่ายใหม่ ซึ่งสามารถใช้หลักการของ Static NAT หรือ Dynamic NAT ช่วยกันได้



- [เพิ่มเติม] Port Forwarding / Network Address Port Translation (NAPT) ถือเป็นการ Mapping ระหว่างไอพี และพอร์ตของอุปกรณ์ต้นทางกับปลายทาง โดยส่วนมากจะเป็นการ Mapping ระหว่างหลาย Private IP Address (หลาย Service) กับ 1 Public IP Address และในหลาย ๆ บทความ Port Forwarding ถือเป็น NAT แบบ PAT (Port Address Translation)

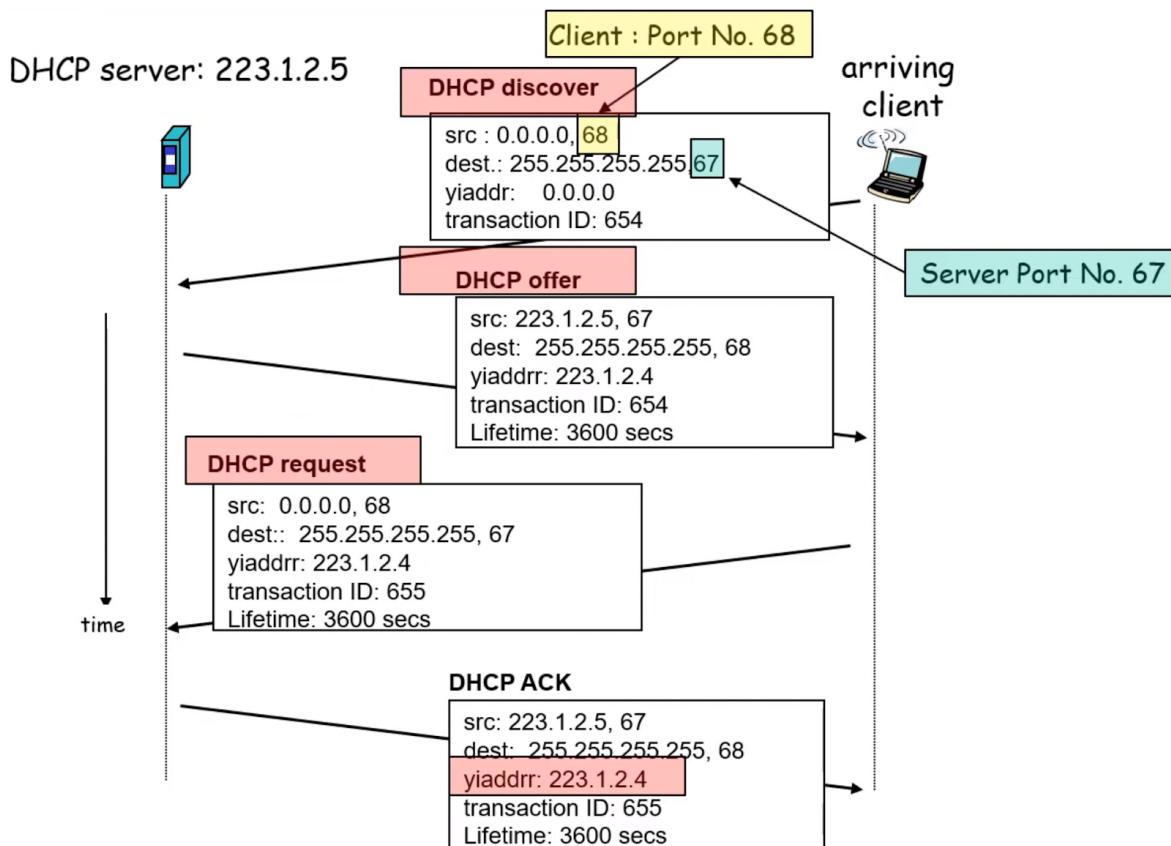


- [เพิ่มเติม] NAT Hairpinning / NAT Loopback / NAT Reflection มีสำหรับให้อุปกรณ์ภายในสามารถเรียกใช้เซอร์วิสบนอุปกรณ์อื่น ๆ ในเครือข่ายเดียวกัน (Local Network) ผ่าน Public IP Address เช่น จากรูปด้านล่าง
  - ↪ Destination IP Address 203.40.40.5 จะถูกเปลี่ยนเป็น 192.168.5.5 โดยการทำ Destination NAT (DNAT)
  - ↪ Source IP Address 192.168.1.50 จะถูกเปลี่ยนเป็น 192.168.5.1 โดยการทำ Source NAT (SNAT)



## Dynamic Host Configuration Protocol (DHCP)

- เป็นโปรโตคอลที่อยู่ในレイเยอร์ที่ 7 แต่มีความเกี่ยวข้องกับ IP
- เพื่อให้อุปกรณ์ที่เชื่อมต่ออยู่ในเครือข่ายได้รับไอพีและเดรสเมื่อเชื่อมต่อมายังเครือข่าย
- ใช้ UDP Protocol สำหรับเซิร์ฟเวอร์ใช้พอร์ต 67 และอุปกรณ์ต่าง ๆ ใช้พอร์ต 68
- สามารถช่วยในเรื่องของไอพีและเดรสไม่พอใช้ เนื่องจากสามารถนำไอพีและเดรสมาใช้ซ้ำได้
- มีหลักการทำงาน 4 ขั้นตอนดังนี้
  - Host broadcasts "DHCP Discover" message
  - DHCP Server responds " DHCP Offer" message
  - Host Request IP Address " DHCP Request" message
  - DHCP Server sends IP Address "DHCP Ack" message



## IPv6



## Characteristic

- IPv6 เป็นตัวเลขฐานสิบหก 6 ชุด แต่ละชุดประกอบด้วยเลขฐานสิบหก 4 ตัว ถูกคั้นด้วย : ขนาดรวม 128 มิติ เช่น
  - 3fee:085b:1f1f:0000:0000:0000:00a9:1234
  - 0000:0000:0000:0000:0000:0000:0001
  - 2001:0000:0000:34fe:0000:0000:00ff:0321
- IPv6 ประกอบด้วยสองส่วน คือ Subnet Prefix (64 bits) กำหนดโดย ISP และ Local Identifier (64 bits) ซึ่งกำหนดเอง
- IPv6 จะใช้หมายเลข IP Address แบ่งออกเป็น 3 ประเภท คือ Unicast, Multicast และ Anycast

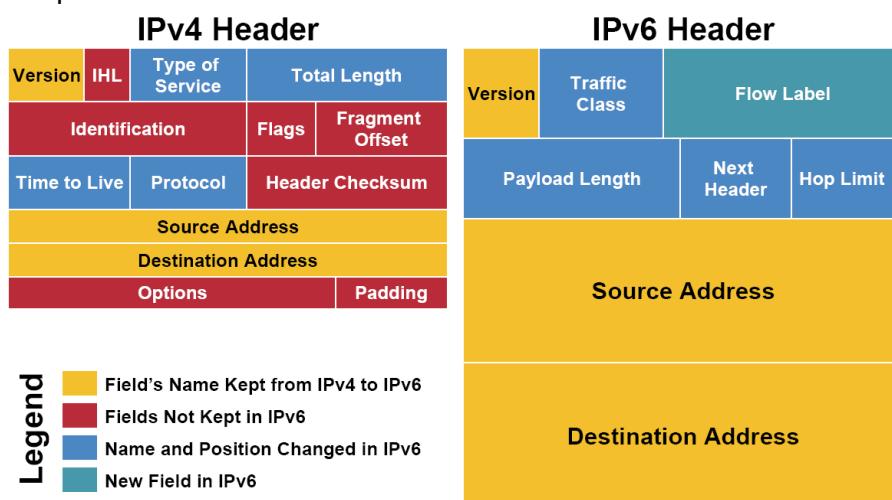
## Text Representation of IPv6 Addresses

จาก IPv6 ที่ยกตัวอย่างมาก่อนหน้านี้ สามารถย่อได้ตามหลักการต่อไปนี้

- Leading Zeros in a 16-Bit Field คือย่อโดยการลด 0 ที่นำหน้าออก (ศูนย์ข้างหน้าไม่มีความหมาย) และถ้าหากกลุ่มใดเป็นเลขศูนย์ทั้ง 4 ตัว (0000) สามารถเขียนแทนด้วย “0” ตัวเดียวจากตัวอย่างก่อนหน้านี้ จะได้หมายเลข IPv6 ดังนี้
  - 3fee:85b:1f1f:0:0:a9:1234
  - 0:0:0:0:0:0:1
  - 2001:0:0:34fe:0:0:ff:321
- Zero Compression คือย่อโดยการลด 0 ที่อยู่ติดกัน โดยลดเหลือ :: (สามารถทำได้ครั้งเดียว) โดยมีข้อกำหนดเพิ่มเติม ในกรณีที่มี 0 อยู่ติดกันมากกว่าหนึ่งที่ โดยให้ลด 0 ที่อยู่ติดกัน ที่มีจำนวนมากกว่าก่อน และ ในกรณีที่ 0 ที่อยู่ติดกันทั้งสองกลุ่ม มีจำนวนเท่ากัน ให้ย่อ 0 กลุ่มที่อยู่ข้างหน้าก่อนจากตัวอย่างก่อนหน้านี้ จะได้หมายเลข IPv6 ดังนี้
  - 3fee:85b:1f1f::a9:1234
  - ::1
  - 2001::34fe:0:0:ff:321

## Special IPv6 Address

- 2000::/3 Global Unicast
  - 2001::/32 สำหรับการทำ Teredo Tunneling
  - 2001:db8::/32 เอาไว้ใช้ในการทำเอกสาร เช่น ใช้ในการยกตัวอย่าง (RFC3849)
  - 2002::/16 สำหรับการทำ 6 to 4 Tunneling
- 3ffe:831f::/32 Teredo Tunneling แบบเก่า เช่นที่ Windows เคยใช้ (ยังมีเหลือให้เห็นอยู่)
- fe80::/10 Link Local Unicast สำหรับติดต่อ กันภายในลิงก์เดียวกัน (ทำงานเดียวกัน Subnet เดียวกัน)
- ff00::/8 Multicast ส่งไปที่ผู้รับหลาย ๆ เครื่อง
  - ff02::1 All Node Address ทุกเครื่องใน Subnet
  - ff02::2 All Routers Address ทุกเราเตอร์ใน Subnet
- ::/128 (ศูนย์หมด) Unspecified แทนไอพีที่ยังไม่ระบุ (ถ้าสั่ง netstat -a ดูพอร์ทที่เปิดรับ IPv6 จากเครื่องไหนก็ได้เอาไว้จะเจอไอพีนี้)
- ::1/128 Loopback

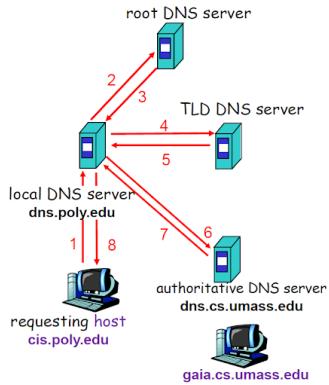


# DNS

Host at [cis.poly.edu](http://cis.poly.edu) wants IP address for [gaia.cs.umass.edu](http://gaia.cs.umass.edu)

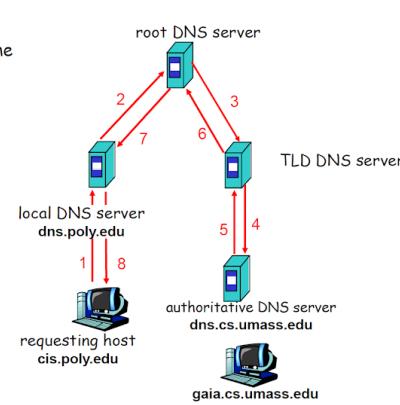
## Iterated Query:

- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"

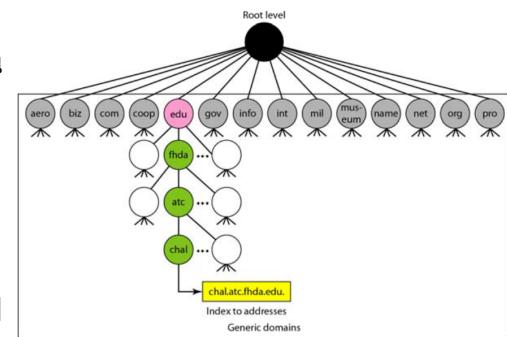


## Recursive Query:

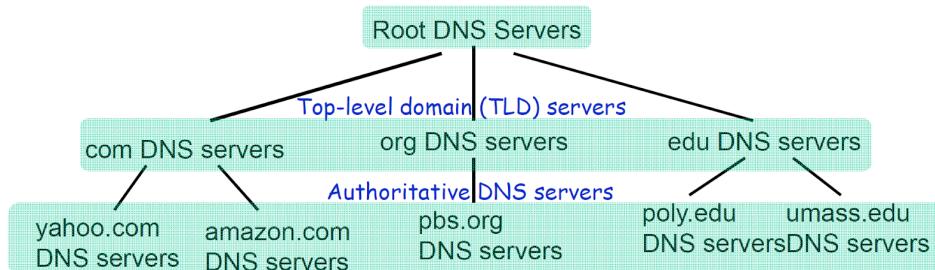
- puts burden of name resolution on contacted name server



- เป็นโปรโตคอลที่อยู่ในเลเยอร์ที่ 7 แต่มีความเกี่ยวข้องกับ IP และเห็นได้บ่อย จึงยกเนื้อหาเข้ามาในบทเรียน
- ระบบอินเตอร์เน็ตมีระเบียบวิธีสื่อสารระหว่างกันได้ด้วย Internet Protocol (IP) และในแต่ละอุปกรณ์บนอินเตอร์เน็ตจะได้รับเลขที่อยู่เฉพาะที่เรียกว่า IP Address เช่น 9.8.7.6 ซึ่งเป็นเรื่องยากที่จะจดจำเลข IP ของเว็บไซต์ จึงเกิดที่มาของ “ชื่อโดเมน”
- หลักการเดิมการตั้งชื่อ DNS คือภาษาอังกฤษ (ASCII Character Set) ตามหลัก RFC 1035 สามารถใช้สัญลักษณ์ ตัวอักษร a ถึง z (Case Insensitive), ตัวเลข 0 - 9 และเครื่องหมายยั่งค์ (-) แต่ปัจจุบันสามารถตั้งชื่อเป็นภาษาไทยได้เช่นกัน
- DNS จะมีส่วนการทำงานหลัก ๆ 3 ส่วน คือ
  - Name Resolvers ของ DNS คือหน้าที่หลักในการแปลงชื่อคอมพิวเตอร์ ให้เป็นหมายเลข IP เครื่องลูกข่ายที่ต้องการทราบหมายเลข IP ซึ่งจะเรียกว่า Resolver โดยจะเป็นซอฟต์แวร์ที่ถูกสร้างมากับเครื่องลูกข่าย หรือแอพพลิเคชัน หรือไลบรารีที่อยู่ในเครื่องนั้นจะเป็นส่วนที่รับข้อมูลไว้
  - Domain Name Space จะเป็นฐานข้อมูลระบบ DNS มีโครงสร้างเป็นแบบ Distributed / Hierarchical Database (Tree) โดยแต่ละ Domain Name Space จะมีชื่อเรียกและสามารถมีโดเมนย่อยหรือที่เรียกว่า Subdomain จะใช้จุด เป็นเครื่องหมายของแบ่งระหว่างโดเมนหลัก และโดเมนย่อย Domain Name Space (Tree) จะถูกแบ่งเป็น 3 ประเภทดังนี้
    - ↪ Generic Domain
      - ใช้สำหรับกำหนด Registered Hosts จนไปถึง Generic Behavior
      - แต่ละ Node จะประกอบไปด้วย 3 โดเมน
    - ↪ Country Domains
      - ใช้ตัวย่อชื่อประเทศ เป็นความยาว 2 ตัว
    - ↪ Inverse Domain
      - เป็นการแปลง IP Address ย้อนกลับไปเป็น Domain Name
- Name Servers คือเครื่องคอมพิวเตอร์แม่ข่ายที่มีหน้าที่จัดการฐานข้อมูลในระบบ DNS โดยจะใช้โปรแกรมตอบกลับการร้องขอที่ได้รับมา ด้วยการค้นหาข้อมูลในฐานข้อมูลตัวเอง หรือการทำงานสำหรับบางที่สามารถเขียนโปรแกรมให้ไปค้นหาข้อมูลที่ฐานข้อมูลอื่นใน Name Server อื่นได้ ถ้าพบข้อมูลที่ได้รับร้องขอ ก็อ้วว่าเป็นเจ้าของโดเมนนั้นจะเรียกว่า Authoritative แต่ถ้าไม่พบข้อมูลจะเรียกว่า Non-Authoritative โดย Name Servers จะแบ่งตามลำดับขั้นได้เป็น 3 ขั้นหลัก ๆ ดังนี้



- ↪ Root Name Servers
  - จะไม่สามารถ Resolve Name ที่ถูกเรียกใช้งานโดย Local Name Server
  - ทำหน้าที่ติดต่อ Authoritative DNS Servers หากรู้จักชื่อที่ต้องการ Resolve
- ↪ Top-Level Domain Servers (TLD)
  - รับผิดชอบเกี่ยวกับ Top-level Country Domains เช่น uk, fr, ca, jp เป็นต้น รวมไปถึงโดเมน com, org, net, edu และอื่น ๆ
- ↪ Authoritative DNS Servers
  - เป็น Organization's DNS Servers สำหรับแปลง Authoritative Hostname เป็น IP สำหรับเครื่องในองค์กร
  - ถูกจัดการโดย Organization หรือ Internet Service Provider (ISP)
- ↪ Local Name Server
  - ไม่ถือว่าเป็นหนึ่งในลำดับชั้น ISP (Residential ISP, บริษัท, มหาลัย) โดยจะเรียกว่า "Default Name Server"
  - เมื่ออุปกรณ์ส่งคำร้อง DNS Query คำร้องจะถูกส่งต่อไปยัง Local DNS Server (เปรียบเสมือน Proxy ที่ส่งคำร้องต่อไปยังลำดับชั้นต่อ ๆ ไป)



## Collision Domain & Broadcast Domain

### Collision Domain

คือกลุ่ม หรือขอบเขตของการชนกันของสัญญาณ (Layer 1), เฟรม (Layer 2) และแพ็คเกต (Layer 3)

- Layer 1 - Hub : ไม่ว่า Hub จะมีกี่พอร์ตก็นับเป็น 1 Collision Domain เช่น
  - Hub 4 พอร์ต >> Collision Domain = 1
  - Hub 8 พอร์ต >> Collision Domain = 1
- Layer 2 - Switch : ทุกพอร์ตของ Switch **ที่ต่อใช้งานอยู่** จะนับเป็น 1 Collision Domain เช่น
  - Switch ต่อ กับ อุปกรณ์ 3 พอร์ต >> 3 Collision Domain
  - Switch ต่อ กับ อุปกรณ์ 10 พอร์ต >> 10 Collision Domain
- Layer 3 - Router : ทุกพอร์ตของ Router **ที่ต่อใช้งานอยู่** จะนับเป็น 1 Collision Domain เช่น
  - Router ต่อ กับ อุปกรณ์ 2 พอร์ต >> 2 Collision Domain
  - Router ต่อ กับ อุปกรณ์ 4 พอร์ต >> 4 Collision Domain

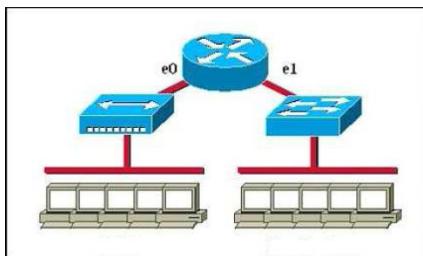
### Broadcast Domain

คือกลุ่มหรือขอบเขตของการ Broadcast โดย 1 Broadcast Domain สามารถถือได้ว่าเป็น 1 Network

- Layer 1 - Hub : ไม่ว่า Hub จะมีกี่พอร์ตก็นับเป็น 1 Broadcast Domain เช่น
  - Hub 4 พอร์ต >> Broadcast Domain = 1
  - Hub 8 พอร์ต >> Broadcast Domain = 1
- Layer 2 - Switch : ทุกพอร์ตของ Switch
  - ถ้าไม่ได้ทำการแบ่ง VLAN ไว้จะนับเป็น 1 Broadcast Domain เช่น

- ↪ Switch 16 พอร์ต >> 1 Broadcast Domain
- ↪ Switch 48 พอร์ต >> 1 Broadcast Domain
- ถ้าทำการแบ่ง VLAN จะนับตามจำนวน VLAN เช่น
  - ↪ Switch มีการทำ VLAN 3 VLAN >> 3 Broadcast Domain
  - ↪ Switch มีการทำ VLAN 5 VLAN >> 6 Broadcast Domain
- Layer 3 - Router : ทุกพอร์ตของเราระบบที่ต่อเข้ามาอยู่ใน 1 Broadcast Domain เช่น
  - Router ต่อกับอุปกรณ์ 2 พอร์ต >> 2 Broadcast Domain
  - Router ต่อกับอุปกรณ์ 4 พอร์ต >> 4 Broadcast Domain

## [การน้ำ necessità] Let's try!!



**Collision Domain = 7**

→ Switch 6 Ports = 6 Collision Domain

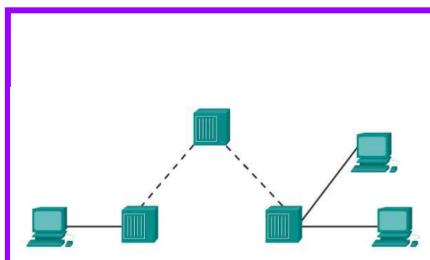
→ Hub = 1 Collision Domain

**Broadcast Domain = 2 (Switch no VLAN)**

→ Router 2 Ports = 2 Broadcast Domain

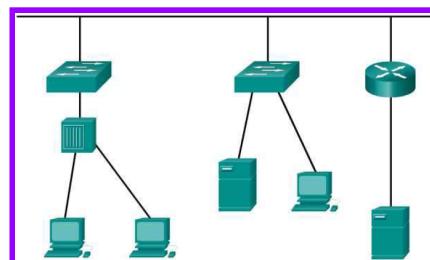
1. Broadcast Domain = \_\_\_\_\_

Collision Domain = \_\_\_\_\_



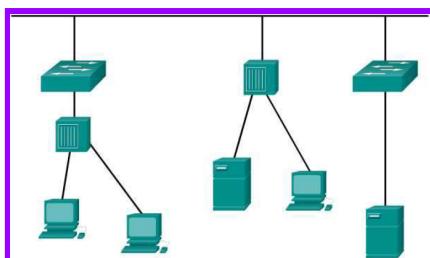
2. Broadcast Domain = \_\_\_\_\_

Collision Domain = \_\_\_\_\_



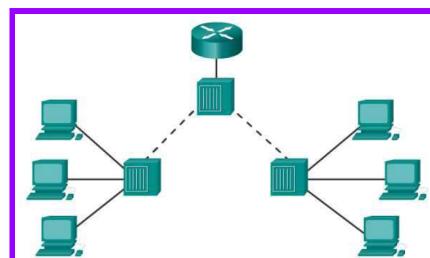
3. Broadcast Domain = \_\_\_\_\_

Collision Domain = \_\_\_\_\_



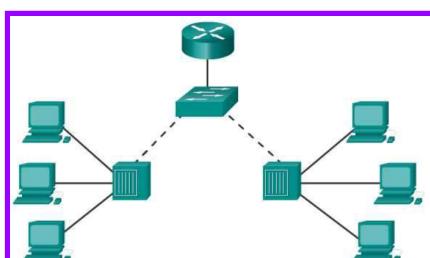
4. Broadcast Domain = \_\_\_\_\_

Collision Domain = \_\_\_\_\_



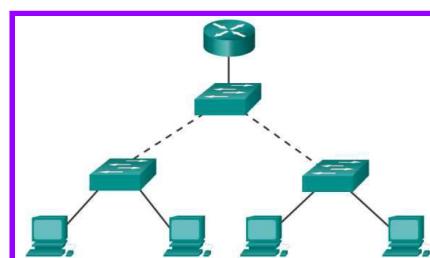
5. Broadcast Domain = \_\_\_\_\_

Collision Domain = \_\_\_\_\_



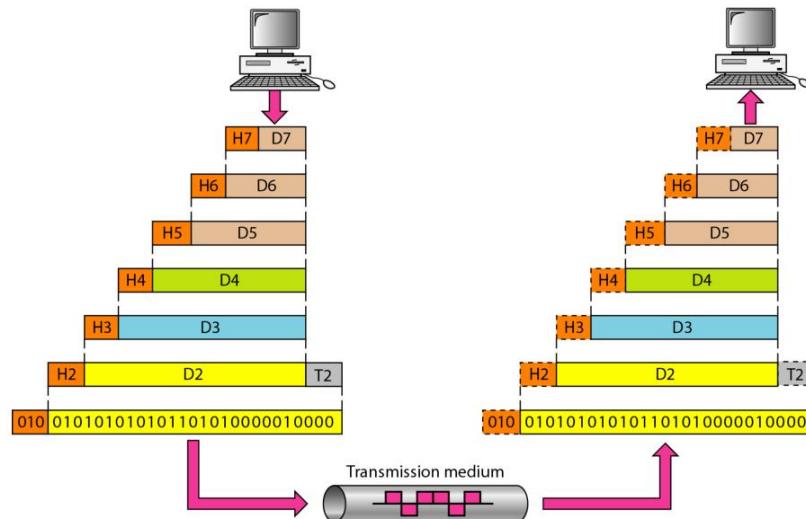
6. Broadcast Domain = \_\_\_\_\_

Collision Domain = \_\_\_\_\_



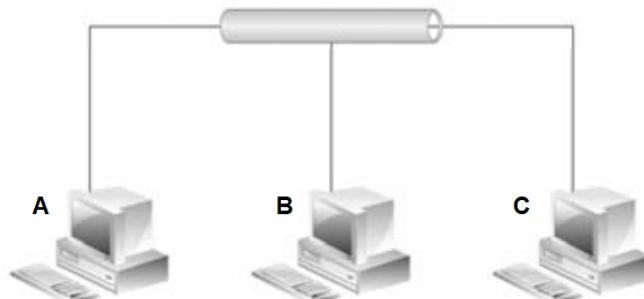
## Packet Scenario

จาก OSI Layer เมื่อทำการส่งข้อมูลต่าง ๆ ผ่านมาตรฐาน OSI จะสามารถจำลองรูปแบบคร่าว ๆ ของข้อมูลตามหลัก Encapsulation & Decapsulation ที่ OSI Layer ใช้ได้ดังนี้

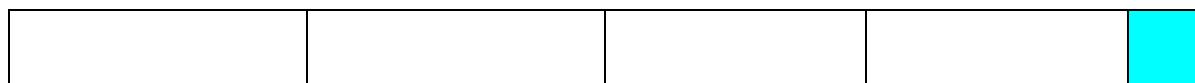


### Let's try!!

1. จากรูปด้านล่างเป็นการเชื่อมต่อคอมพิวเตอร์ระหว่าง คอมพิวเตอร์ 3 เครื่องด้วย Hub

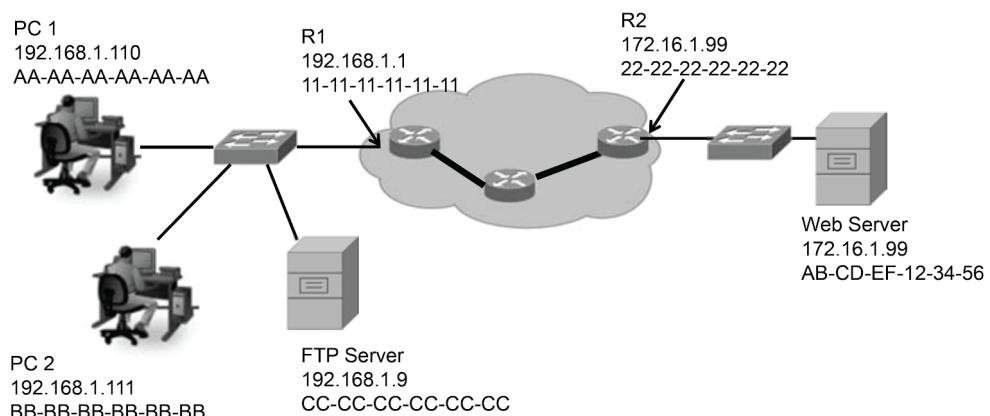


- 1.1. ต้องการสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์ A และ C การจำลองรูปแบบข้อมูลใน เลเยอร์ที่ 2 และ 3 เป็นดังนี้



- 1.2. ต้องการสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์ A และ C โดยใช้คำสั่ง PING การจำลองรูปแบบข้อมูลในเลเยอร์ที่ 2, 3 และ 4 เป็นดังนี้


2. จากรูปด้านล่างเป็นรูปแบบผังเครือข่ายในบริษัทแห่งหนึ่งดังนี้



- 2.1. ต้องการสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์ PC1 และ PC2 โดยใช้คำสั่ง PING การจำลองรูปแบบข้อมูลในเลเยอร์ที่ 2, 3 และ 4 เป็นดังนี้

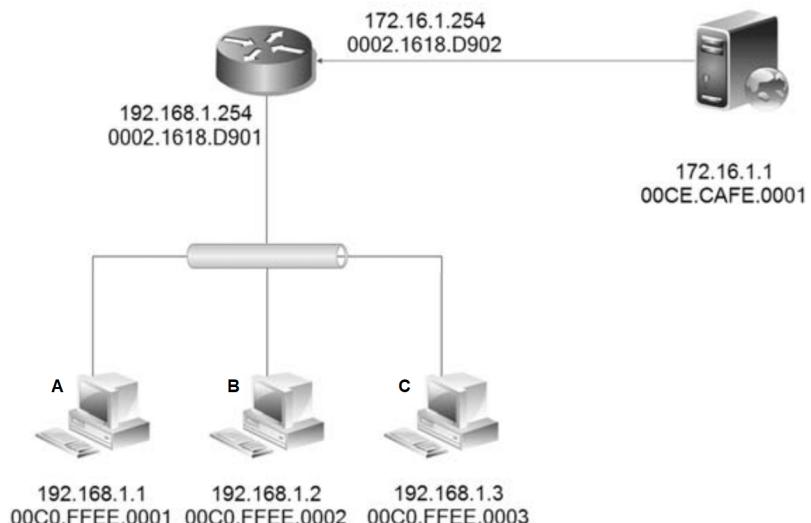

2.2. ต้องการสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์ PC2 กับ FTP Server การจำลองรูปแบบข้อมูลในเลเยอร์ที่ 2, 3 และ 4 เป็นดังนี้


2.3. ต้องการอัพเกรดเฟิร์มแวร์ของเราเตอร์ R2 โดยใช้เครื่องคอมพิวเตอร์ PC2 (กำหนดให้อัพเกรดผ่านพอร์ต 1234) การจำลองรูปแบบข้อมูลในเลเยอร์ที่ 2, 3 และ 4 เป็นดังนี้


2.4. [การบ้าน] ต้องการสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์ PC1 ด้วยพอร์ต 9999 กับ Web Server จงเขียนการจำลองรูปแบบข้อมูลในเลเยอร์ที่ 2, 3 และ 4



3. จากรูปด้านล่างเป็นรูปแบบผังเครือข่ายในบริษัทแห่งหนึ่ง หากต้องการสื่อสารข้อมูลระหว่าง เครื่องคอมพิวเตอร์ A กับ Web Server การจำลองรูปแบบข้อมูลในเลเยอร์ที่ 2, 3 และ 4 เป็น ดังนี้




4. [การบ้าน] จากรูปด้านล่างเป็นรูปแบบผังเครือข่ายในบริษัทแห่งนี้ ถ้าหากต้องการติดต่อ กับ [www.isageiei.com](http://www.isageiei.com) (Server-1) จะเขียนการจำลองรูปแบบข้อมูลในเลเยอร์ที่ 2, 3 และ 4 ดังแต่เริ่มติดต่อจนถึง Web Server ได้รับข้อมูล

