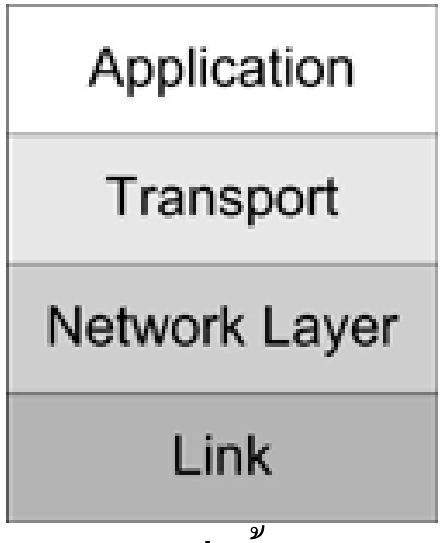
TCP/IP

TCP/IP (Transmitsion Control Protocol/InternetProtocol) เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ต ทำให้ TCP/IP เป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

การแบ่งชั้นของ TCP/IP

TCP/IP แบ่งออกเป็น 4 เลเยอร์ ดังรูปที่ 1.1



รูปที่ 1.1 การแบ่งชั้นของ TCP/IP

ในแต่ละเลเยอร์จะมีหน้าที่ดังนี้

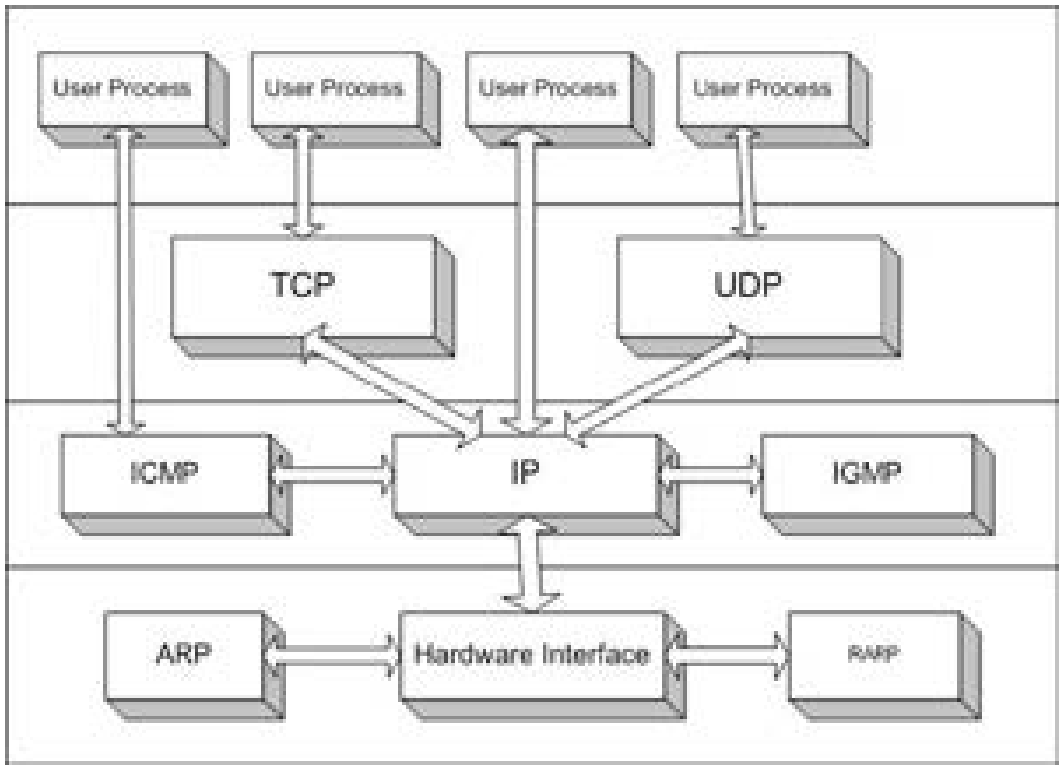
1. **Link Layer -** เลเยอร์นี้มีหน้าที่ควบคุมการรับส่งข้อมูลในระดับฮาร์ดแวร์ของเครือข่าย รับผิดชอบการรับส่งข้อมูลในระดับกายภาพ จนถึงการแปลความจะกสัญญาณไฟฟ้าเป็นข้อมูลทางคอมพิวเตอร์
2. **Network Layer -** ทำหน้าที่รับข้อมูลจากชั้น Transport Layer และค้นหาและเลือกเส้นทาง ระหว่างผู้รับและผู้ส่ง เทียบได้กับ Network Layer ของ OSI Model โปรโตคอลในเลเยอร์นี้ได้แก่ IP,ICMP,IGMP
3. **Transport Layer -** รับผิดชอบการรับส่งข้อมูลระหว่างปลายด้านส่งและด้านรับข้อมูล และส่งข้อมูลขึ้นไปให้ Application Layer นำไปใช้งาน ต่อ เทียบได้กับ Session Layer และ Transport Layer ของ OSI Model
4. **Application Layer -** เป็นเลเยอร์ที่แอพพลิเคชั่นเรียกโปรโตคอลระดับล่างๆลงไป เพื่อให้บริการต่างๆ เช่น FTP , SMTP , Telnet , HTTP , POP

โครงสร้างของโปรโตคอล TCP/IP

เนื่องจาก TCP/IP เป็นชุดของโปรโตคอลประกอบด้วยโปรโตคอลหลายตัวทำงานร่วมกันในเลเยอร์ต่างๆ และมีหน้าที่แตกต่างกันออกไป ได้แก่

- **TCP : (Tranmission Control Protocol)** - อยู่ใน Transport Layer ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล และมีกลไกความคุมการ รับส่งข้อมูลให้มีความถูกต้อง (reliable) และมีการสื่อสารอย่างเป็นกระบวนการ (connection-orient)
- **UDP : (User Datagram Protocol)** - อยู่ใน Transport Layer ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล แต่ไม่มีกลไกความคุมการรับ ส่งข้อมูลให้มีเสถียรภาพและเชื่อถือได้ (unreliable,connectionless) โดยปล่อยให้เป็นหน้าที่ของแอพพลิเคชั่นเลเยอร์ แต่ UDP มีข้อได้เปรียบในการส่งข้อมูลได้ทั้งแบบ unicast, multicast และ broadcast อีกทั้งยังทำการติดต่อสื่อสารได้เร็วกว่า TCP เนื่องจาก TCP ต้องเสีย overhead ให้กับขั้นตอนการสื่อสารที่ทำให้ TCP มีความน่าเชื่อถือในการรับส่งข้อมูลนั่นเอง

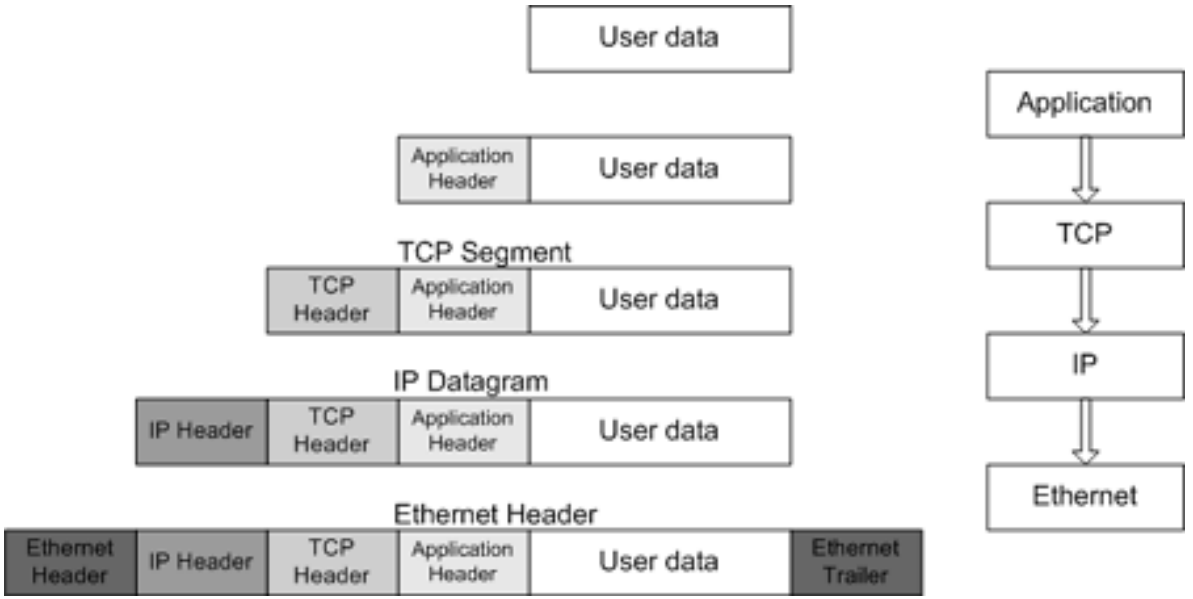
- **IP : (Internet Protocol)** - อยู่ใน Internetwork Layer เป็นโปรตคอลหลักในการสื่อสารข้อมูล มีหน้าที่ค้นหาเส้นทางระว่างผู้รับและผู้ส่ง โดยใช้ IP Address ซึ่งมีลักษณะเป็นเลขสี่ชุด แต่ละชุดมีค่าตั้งแต่ 0-255 เช่น 172.17.3.12 ในการอ้างอิงโฮสต์ต่างๆ และกลไกการ Route เพื่อส่งต่อข้อมูลไปจนถึงจุดหมายปลายทาง
- **ICMP : (Internet Control Message Protocol)** - อยู่ใน Internetwork Layer มีหน้าที่ส่งข่าวสารและแจ้งข้อผิดพลาดให้แก่ IP
- **IGMP : (Internet Group Management Protocol)** อยู่ในเน็ตเวิร์กเลเยอร์ ทำหน้าที่ในการส่ง UDP ดาต้าแกรมไปยัง กลุ่มของโฮสต์ หรือ โฮสต์หลายๆตัวพร้อมกัน
- **ARP : (Address Resolution Protocol)** - อยู่ใน Link Layer ทำหน้าที่เปลี่ยนระหว่าง IP แอดเดรส ให้เป็นแอดเดรสของ Network Interface เรียกว่า MAC Address ในการติดต่อระหว่างกัน MAC Address คือหมายเลขประจำของ Hardware Interface ซึ่งในโลกนี้จะไม่มี MAC Address ที่ซ้ำกัน มีลักษณะเป็นเลขฐาน 16 ยาว 6 ไบต์ เช่น 23:43:45:AF:3D:78 โดย 3 ไบต์แรกจะเป็นรหัสของผู้ผลิต และ 3 ไบต์หลังจะเป็นรหัสของผลิตภัณฑ์
- **RARP : (Reverse ARP)** - อยู่ในลิงค์เลเยอร์เช่นกัน แต่ทำหน้าที่กลับกันกับ ARP คือเปลี่ยนระหว่างแอดเดรสของ Network Interface ให้ เป็นแอดเดรสที่ใช้โดย IP Address



รูปที่ 1.2 แสดงให้เห็นถึงความสัมพันธ์ระหว่างโปรโตคอลต่างๆใน TCP/IP

Encapsulation/Demultiplexing

เวลาส่งข้อมูล เมื่อข้อมูลถูกส่งผ่านในแต่ละเลเยอร์ แต่ละเลเยอร์จะทำการประกอบข้อมูลที่ได้รับมา กับส่วนควบคุมซึ่งอยู่ส่วนหัวของข้อมูลเรียกว่า Header ภายใน Header จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับ ได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับคือ โปรโตคอลเดียวกัน ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing



รูปที่ 1.3 ขั้นตอนการ encapsulation เมื่อข้อมูลถูกส่งผ่านโปรโตคอลต่างๆ

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละระดับมีชื่อเรียกแตกต่างกัน ข้อมูลที่มาจาก User หรือก็คือข้อมูลที่ User เป็นผู้ป้อนให้กับ Application เรียกว่า **User Data** เมื่อ Application ได้รับข้อมูลจาก user ก็จะนำมาประกอบกับส่วนหัวของ Appliction เรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP

เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมารวมกับ Header ของ โปรโตคอล TCP เรียกว่า **TCP Segment** และส่งต่อไปยังโปรโตคอล IP

เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมารวมกับ Header ของ โปรโตคอล IP เรียกว่า **IP Datagram** และส่งต่อไปยังเลเยอร์ Datalink Layer

ในระดับ Datalink จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป

Internet Protocol

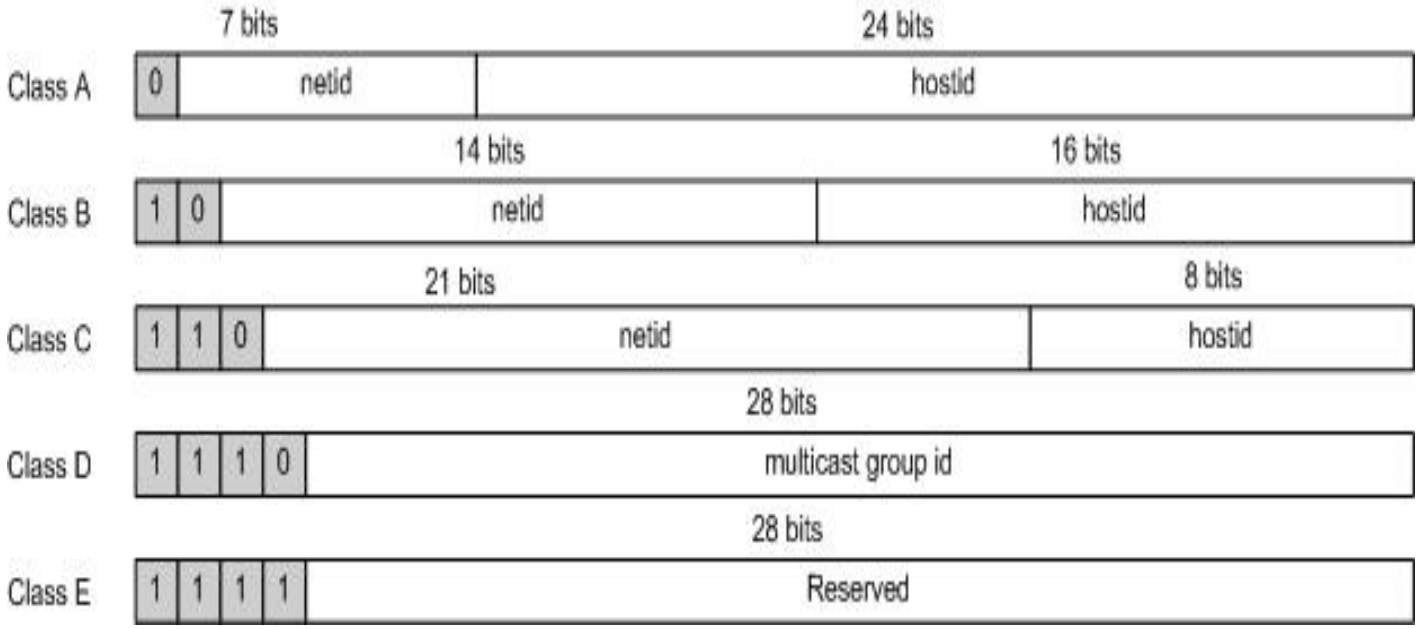
ด้วยเหตุที่ IP เป็นโปรโตคอลหลักในการสื่อสารข้อมูล และถือได้ว่าเป็นหัวใจสำคัญของโปรโตคอล TCP/IP ผมจะขอยกมาอธิบายก่อน เพื่อให้ง่ายต่อการอธิบายโปรตคอลตัวอื่นๆ ต่อไป ในบทนี้ท่านจะได้เรียนรู้เกี่ยวกับหน้าที่และลักษณะของโปรโตคอล IP , Internet Address , รูปร่างของ IP Header , การ Routing และ การจัดสรร IP ด้วย Subnet

IP เป็นโปรโตคอลที่ทำหน้าที่รับภาระในการนำข้อมูลไปส่งยังผู้รับ ที่เชื่อมต่ออยู่ในระบบ network ซึ่งทั้งสองฝั่งอาจอยู่คนละเน็ตเวิร์คกันก็ได้ โปรโตคอลอื่นๆ ในระดับ network Layer ขึ้นไปทั้งTCP , UDP ,ICMP ต่างก็ต้องอาศัยโปรโตคอล IP ในการรับส่งข้อมูลทั้งสิ้น

โปรโตคอล IP มีความสามารถในการค้นหาเส้นทางจากผู้รับไปยังผู้ส่ง มีกลไกที่ชาญฉลาดในการค้นหาเส้นทาง สามารถค้นหาเส้นทางได้ไปถึงผู้รับได้เอง หากมีเส้นทางที่สามารถไปได้ แต่ไม่ได้ติดต่อระหว่างผู้รับกับผู้ส่งโดยตรง และไม่มีการยืนยันว่า ข้อมูลถึงผู้รับจริงหรือไม่ ทั้งนี้อาจเกิดจากหลายสาเหตุ เช่น ที่อยู่ของผู้รับไม่มีการเชื่อมต่ออยู่ในระบบ Internet กล่าวได้ว่า โปรโตคอล IP มีหน้าที่ในการค้นหาเส้นทางเท่านั้น ไม่มีการยืนยันผลสำเร็จในการส่งข้อมูล หากเกิดข้อผิดพลาดในการส่งข้อมูล แม้ว่าจะมีการส่ง icmp massage กลับมารายงานข้อผิดพลาด แต่ก็รับประกันไม่ได้อยู่ดีว่า icmp message จะกลับมาถึงเรียบร้อยหรือไม่ ด้วยเหตุนี้ จึงถือว่า IP เป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (reliable)

IP Addressing

ทุกอินเตอร์เฟซที่ต่ออยู่บนอินเตอร์เน็ตจะต้องมีหมายเลขประจำตัวเพื่อใช้ในการสื่อสารข้อมูล เรียกว่า Internet Address หรือเรียกย่อๆว่า IP Address โดยค่า IP Address นี้จะเป็นหมายเลขจำนวน 32 บิต แต่แทนที่จะกำหนดให้เลขทั้ง 32 บิตนั้นถูกนับต่อเนื่องกันไป ก็จะใช้วิธีการแบ่งหมายเลขดังกล่าวออกเป็นกลุ่มของเลขขนาด 8 บิตจำนวน 4 ชุด และคั่นแต่ละชุดด้วยจุด ตัวอย่างเช่น 172.17.3.12 นอกจากนี้ใน IP Address นั้นยังถูกแบ่งออกเป็น 2 ส่วนคือ ส่วนที่ เป็นแอดเดรสของเน็ตเวิร์ก (Network ID) และส่วนที่เป็นแอดเดรสของโฮสต์ (Host ID) ซึ่งข้อมูลในส่วนนี้จะถูกใช้สำหรับ ค้นหาเส้นทางของ IP ในการที่จะขนส่งข้อมูลจากต้นทางให้ถึงปลายทางอย่างถูกต้อง เพื่อเป็นการกำหนดขนาดของเน็ตเวิร์ก สำหรับ IP Address ต่างๆดังนั้นจึงมีการจัด IP Address ในแต่ละช่วงออกเป็นคลาส (class) ต่างๆกันจาก A ถึง E เพื่อจะได้ทำการจัดสรร IP Address ได้อย่างเหมาะสมกับขนาดของเน็ตเวิร์ก



รูปที่ 2.1 การกำหนด IP Address ในคลาสต่างๆ

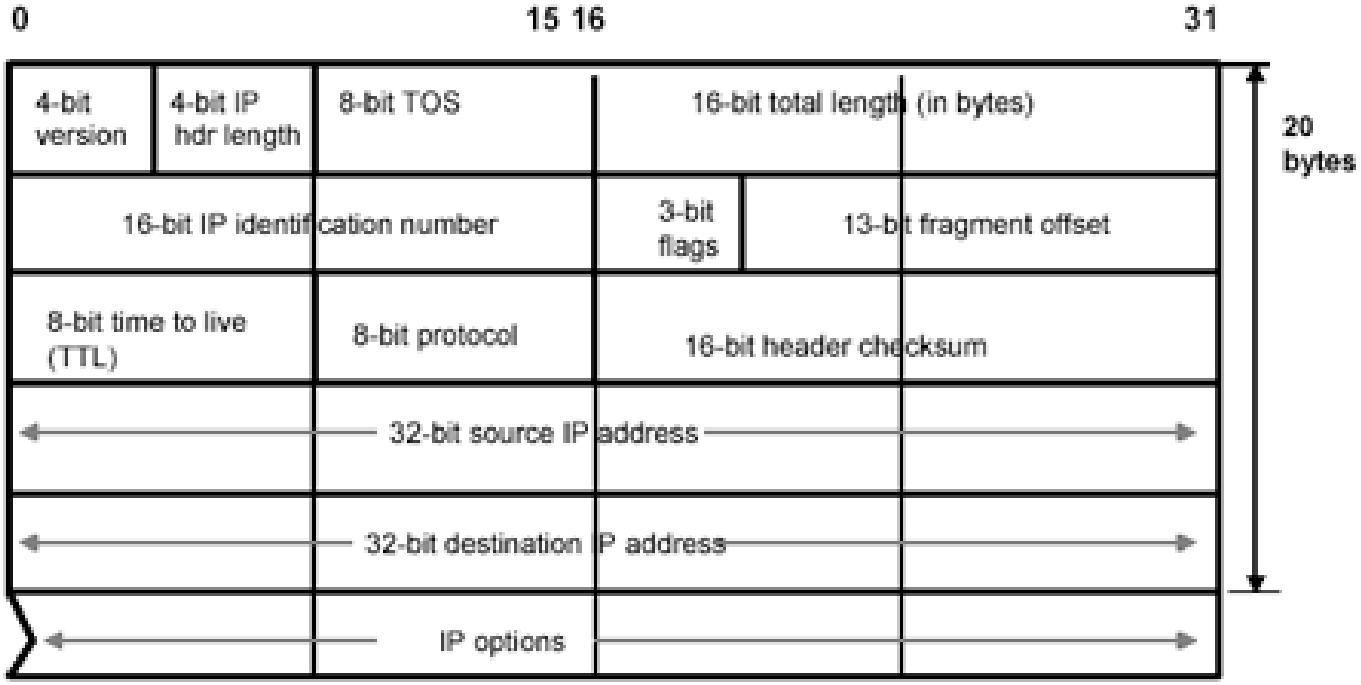
จากข้อกำหนดในการแบ่งคลาสของ IP Address หากลองนำบิตที่อยู่ในตอนต้นของ ip address ในแต่ละคลาสมาแปลงเป็น ip address ในเลขฐานสิบ จะเห็นว่าแต่ละคลาสครอบคลุม ip address ช่วงต่างๆ ดังตารางที่ 2.2

Class	IP Range
A	0.0.0.0 - 127.255.255.255
B	128.0.0.0 - 191.255.255.255
C	192.0.0.0 - 223.255.255.255
D	224.0.0.0 - 239.255.255.255
E	240.0.0.0 - 255.255.255.255

ตารางที่ 2.2 แสดงช่วงของ IP Adress ในแต่ละคลาส

IP Header

เมื่อข้อมูลถูกส่งลงมาจากชั้น Transport Layer สู่ชั้นNetwork Layer กระบวนการ Encapsulate ของ IP Protocol จะทำหารเพิ่มส่วน Header ลงไป Header ของ IP datagram มีขนาด 20-32 ไบต์ มีส่วนประกอบต่างๆ ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 IP Header

<table>
<tr><th>ตำแหน่ง</th><th>ชื่อ</th><th>อธิบาย</th></tr>
<tr><td>0-3</td><td>Version</td><td>มีขนาด 4 บิตเป็นเวอร์ชั่นของ IP ปัจจุบันค่านี้ถูกกำหนดให้เป็น 4</td></tr>
<tr><td>4-7</td><td>Lenght</td><td>มีขนาด 4 บิตเป็นค่าความยาวของ Header นี้ โดยปกติจะเป็น 5 หมายความว่า 5*32 บิต = 20 ไบต์</td></tr>
<tr><td>8-15</td><td>Type of Service</td><td>เป็นข้อมูลขนาด 8 บิต ปัจจุบันไม่ได้ใช้งานแล้ว</td></tr>
<tr><td>16-31</td><td>Total length</td><td>เป็นฟิลด์ที่บอกจำนวนไบต์ทั้งหมดของ IP Datagram ด้วยขนาด 16 บิตทำให้ Datagram มีขนาดสูงสุดไม่เกิน 65535 ไบต์ และมีขนาดเล็กสุดไม่ต่ำกว่า 512 ไบต์</td></tr>
<tr><td>32-47</td><td>Identification</td><td>ใช้ในกรณีที่มีการแบ่งดาต้าแกรมออกเป็นแฟรกเมนต์ เมื่อนำกลับมารวมกันใหม่จะได้รู้ว่ามาจากดาต้าแกรมเดียวกัน</td></tr>
<tr><td>48-50</td><td>Flag</td><td>ใช้ในกรณีที่มีการแบ่งข้อมูลออกเป็นแฟรกเมนต์ มีความหมายดังนี้
<table>
<tr><td>บิต 0 : reserved</td><td>เป็น 0 เสมอ</td></tr>
<tr><td>บิต 1 (DF)</td><td>0 = May Fragment, 1 = Don't Fragment</td></tr>
<tr><td>บิต 2 (MF)</td><td>0 = Last Fragment, 1 = More Fragments.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>51-63</td><td>fragment offset</td><td>เป็นส่วนระบุข้อมูลที่ใช้แยกรวมข้อมูล เพื่อให้ข้มูลที่ถูกแยกออกเป็นแฟรกเมนต์กลับมารวมกันได้อย่างถูกต้องตามลำดับ</td></tr>
<tr><td>64-71</td><td>Time to Live (TTL)</td><td>เป็นจำนวนครั้งสูงสุดที่ดาต้าแกรมนี้จะถูกส่งผ่านครือข่ายไปยังปลายทางได้ เพื่อ ป้องกันไม่ให้ดาต้าแกรมถูกเราต์ไปเรื่อยๆอย่างไม่สิ้นสุด ปกติค่านี้จะเริ่มต้นที่ 32 และจะถูกลดค่าลงทีละ 1 เมื่อมีการเราต์ จนค่านี้มีค่าเป็น 0 ก็จะไม่ถูกเราต์อีกต่อไป</td></tr>
<tr><td>72-79</td><td>Protocol</td><td>เป็นข้อมูลที่ระบุโปรโตคอลที่ส่งดาต้าแกรมนี้มา ตัวอย่างโปรโตคอลที่ใช้บ่อยๆ ได้แก่
<table>
<tr><td>โปรโตคอล</td><td>ค่าในฟิลด์ Protocol</td><td>อธิบาย</td></tr>
<tr><td>ICMP</td><td>1</td><td>Internet Comtrol Message Protocol</td></tr>
<tr><td>TCP</td><td>6</td><td>Transmission Control Protocol</td></tr>
<tr><td>UDP</td><td>17</td><td>User Datagram Protocol</td></tr>
</table></td></tr>
<tr><td>80-95</td><td>Header Checksum</td><td>เป็นส่วนตรวจสอบความถูกต้องของข้อมลใน Header โดยไม่เกี่ยวกับส่วนข้อมูลที่อยู่ภายใน payload ค่านี้จะถูกคำนวนใหม่ทุกครั้งที่มีการเปลี่ยนแปลงข้อมูลใน Header (เช่น TTL ที่มีการเปลี่ยนแปลงทุกครั้งที่ IP datagram ถูกส่งผ่านเราเตอร์)</td></tr>
</table>

86-127	Source IP Address	คือ IP Address ของผู้ส่งดาต้าแกรม
128-163	Destination IP Address	คือ IP Address ของผู้รับดาต้าแกรม
ไม่แน่นอน	Option	มีขนาดข้อมูลไม่แน่นอน ใช้สำหรับกำหนดค่าพารามิเตอร์ปลีกย่อย ซึ่งส่วนใหญ่ไม่มีการนำไปใช้งาน
ขึ้นอยู่กับ Option	Padding	มีข้อมูลว่างเปล่า ใช้เป็นส่วนเติมเต็มของฟิลด์ Option ให้ครบ 32 ไบต์

IP Routing

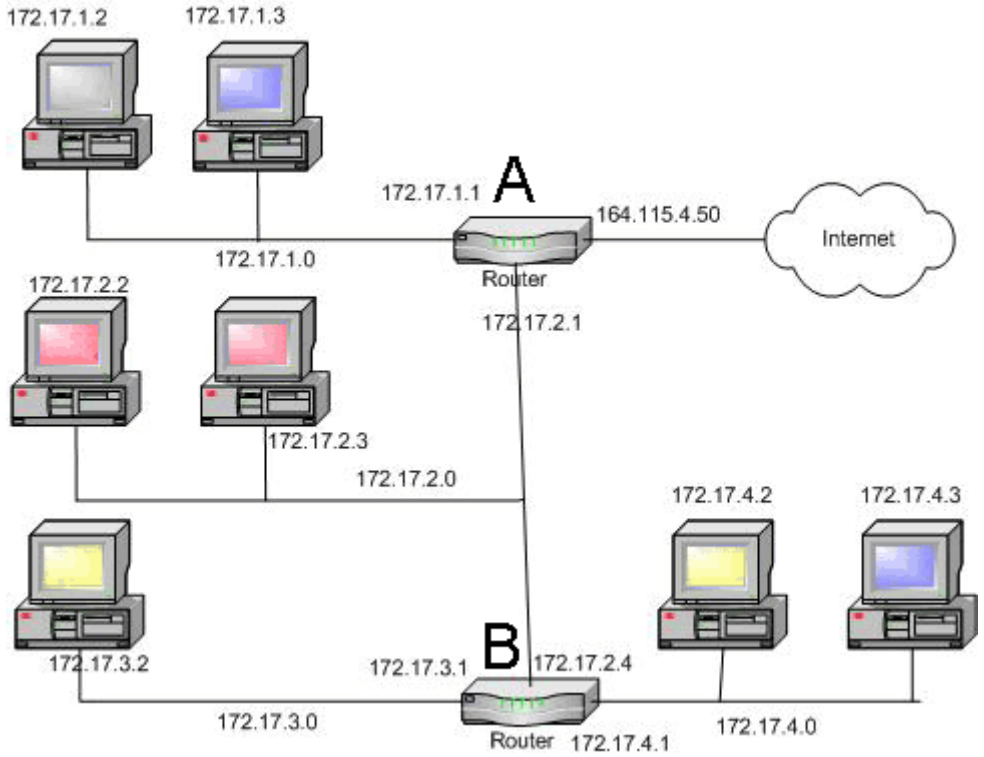
IP Routing เป็นกระบวนการค้นหาเส้นทางในการส่งผ่านข้อมูลจากต้นทางไปยังปลายทางโดยผ่านการส่งต่อข้อมูลไปจนกว่าจะถึงปลายทาง นับเป็นกลไกสำคัญที่ทำให้ IP เป็นโปรโตคอลที่สามารถส่งข้อมูลจากโฮสต์หนึ่งไปอีกโฮสต์หนึ่งได้แม้ว่าจะอยู่ไกลแสนไกล ขอเริ่มต้นทฤษฎีของ IP Routing ด้วยการทำความรู้จักกับส่วนประกอบต่างๆ ของเน็ตเวิร์ค ในแง่ของ IP Routing กันก่อน

Host โฮสต์เป็นอุปกรณ์ที่ทำหน้าที่ให้กำเนิดข้อมูลในกรณีเป็นผู้ส่ง หรือทำหน้าที่รับข้อมูลไปใช้งานในกรณีเป็นผู้รับ การสื่อสาร ข้อมูลใดๆจะต้องเป็นการสื่อสารจากโฮสต์ไปยังโฮสต์เสมอ สำหรับ IP Packet แล้วข้อมูลในเฮดเดอร์ที่ปรากฏอยู่ในฟิลด์ Source Address และ Destination Address ซึ่งเรียกว่า IP Address จะเป็นหมายเลขระบุตำแหน่งของโฮสต์ต้นทางและโฮสต์ ปลายทางเท่านั้น

Network เน็ตเวิร์คเป็นเครือข่ายที่มีการเชื่อมต่อกันของโฮส 2 ตัวขึ้นไป โฮสต์แต่ละตัวในเน็ตเวิร์คเดียวกันสามารถเชื่อมต่อถึงกันได้โดยตรง

Router เราเตอร์ทำหน้าที่ในการ ส่งผ่านข้อมูลจากเน็ตเวิร์กหนึ่งไปยังอีกเน็ตเวิร์กหนึ่ง ตำแหน่งของเราเตอร์จะอยู่ในจุดที่เชื่อมต่อระหว่างสองเน็ตเวิร์กเข้าด้วยกัน ด้วยข้อกำหนดของ IP ข้อมูลจะส่งไปถึงกันโดยตรงข้ามเน็ตเวิร์กไม่ได้ จะต้องอาศัยเราเตอร์เป็นผู้ทำหน้าที่ส่งผ่านข้อมูลไปให้ ใน Router จะมี Routing Table สำหรับเก็บข้อมูล เพื่อใช้ในการพิจารณาเลือกเส้นทางในการส่งดาต้าแกรม ซึ่งจะอธิบายกลไกการทำงานในหัวข้อถัดไป

ในการอธิบายกระบวนการ Routing ให้เป็นที่เข้าใจในเบื้องต้น ผมจะขออธิบายจากเน็ตเวิร์คตัวอย่างที่แสดงในรูปที่ 2.4



รูปที่ 2.4 network ตัวอย่าง

การ Routing จะเป็นไปตามขั้นตอนดังนี้

1. ถ้าโฮสต์ต้นทางและปลายทางต่อเชื่อมร่วมอยู่ในเน็ตเวิร์กเดียวกัน มีการเชื่อมต่อถึงกันโดยตรง เช่น อีเธอร์เน็ต หรือโทเค็นริง ดังแสดงในภาพที่ 2.3 เป็นการติดต่อระหว่าง 172.17.2.2 และ 172.17.2.3 (เครื่องสีแดง) IP ดาต้าแกรมก็จะถูกส่งไปยังโฮสต์ปลายทางโดยตรง

2. หากโฮสต์ต้นทางและปลายทางไม่ได้อยู่ในเน็ตเวิร์กเดียวกัน IP ดาต้าแกรมจะถูกส่งไปยังดีฟอลต์เราเตอร์ 3. เมื่อเราเตอร์ได้รับ IP ดาต้าแกรมจากข้อ 2 แล้วตรวจสอบดู หากพบว่าโฮสต์ปลายทางต่อร่วมอยู่บนเน็ตเวิร์กเดียวกันกับเราเตอร์ ให้ทำการส่งดาต้าแกรมไปที่โฮสต์นั้น เช่น หาก 172.17.3.2 ต้องการส่งดาต้าแกรมไปยัง 172.17.4.2 (เครื่องสีเหลือง) จะต้องส่งดาต้าแกรมไปที่ Router B Router B จะส่งดาต้าแกรมต่อไปยังโฮสปลายทาง 4. หากไม่ได้ต่อร่วมกันก็ส่งดาต้าแกรมไปที่เราเตอร์ตัวต่อไป โดย Router จะเป็นผู้เลือกเส้นทาง ซึ่งมีอยู่ 2 กรณีคือ

1. ถ้ามีข้อมูลของโฮสปลายทางอยู่ใน Routing Table Router จะส่งดาต้าแกรมไปยัง router ตัวที่ระบุไว้ใน routing table
2. ถ้าไม่มีข้อมูลของโฮสปลายทางอยู่ใน Routing Table Router จะส่งดาต้าแกรมไปยัง default router

และกลับไปที่ขั้นตอนในข้อ 3 ใหม่ จนกว่า IP ดาต้าแกรมจะเดินทางถึงปลายทางหรือหมดเวลาในการส่ง (TTL=0)

สมมติว่าเครื่อง 172.17.1.3 ต้องการติดต่อกับ 172.17.4.3 จะต้องส่ง ip datagram ไปยัง Router A หาก Router A มีข้อมูลเกี่ยวกับ 172.17.4.3 อยู่ ก็จะรู้ว่าต้องส่งดาต้าแกรมไปยัง Router B คือ 172.17.2.4 และ Router B ก็จะส่ง ip datagram ไปยังโฮสปลายทางได้สำเร็จ

Subnet Adressing / Subnet Mask

ในการใช้งาน โปรโตคอล TCP/IP ใน internet นั้นการแบ่ง IP Address ออกเป็นแอดเดรสของเน็ตเวิร์ก (netid) และแอดเดรสของ โฮสต์ ตามที่ระบุของแต่ละคลาสค่อนข้างจะขาดประสิทธิภาพ คือในเน็ตเวิร์กคลาส A และ B แต่ละเน็ตเวิร์กนั้น สามารถมีจำนวนโฮสต์ได้มาก ซึ่งการที่จะนำ IP Address มาใช้อย่างทั่วถึงนั้นมีโอกาสเป็นไปได้ยากมากทั้งคลาส A และคลาส B เพราะมีโอกาสน้อยมากที่จะมีเน็ตเวิร์ก ใดในโลกมีจำนวนโฮสต์มากมายขนาดนั้นอยู่ภายในเน็ตเวิร์กเดียว ดังนั้น IP Address ที่จัดสรรให้ไปในแต่ละเน็ตเวิร์กของ คลาสเหล่านี้จึงถูกใช้ไม่หมดและไม่สามารถนำไปใช้ประโยชน์ที่อื่นได้เลย ดังนั้นเพื่อให้การจัดสรร ip เป็นไปอย่างมีประสิทธิภาพ จึงมีการนำส่วนของ hostid มาแบ่งย่อยเป็นสองส่วนคือ subnet id และ host id ทำให้ได้เน็ตเวิร์คย่อยหลายๆเน็ตเวิร์ค โดยในแต่ละเน็ตเวิร์ค มีจำนวนโฮสไม่มากเกินไปและเพียงพอต่อการใช้งาน

การแบ่ง Subnet ใช้เทคนิคที่เรียกว่า Subnet Mask ซึ่งเป็นตัวเลขมีความยาว 32 บิต แบ่งออกเป็นสี่ชุดเช่นเดียวกับ ip แต่ค่าของ subnet mask จะขึ้นอยู่กับความต้องการในการแบ่ง subnet ว่าต้องการจำนวน subnet เท่าใดและมีจำนวนโฮสเท่าใด หากนำ subnet mask มาเขียนเป็นเลขฐานสอง จะมีลักษณะพิเศษคือ ขึ้นต้นด้วยเลข 1 มีจำนวนกี่ตัวก็ได้ ตามแต่ความต้องการในการแบ่ง subnet และตำแหน่งที่เหลือจะมีค่าเป็น 0 ความสัมพันธ์ระหว่าง ip address , subnet mask , host id , net id , จำนวน subnet และ จำนวน host จะเป็นดังนี้

- host id = (ip address) AND ~(subnet mask)
- net id = (ip address) AND (subnet mask)
- จำนวน host = ((2^จำนวนบิตที่เป็น 0 ของ subnet mask)-2) เนื่องจาก ip แรกของ subnet ถูกใช้เป็น net ip และ ip สุดท้ายของ subnet ถูกใช้เป็น broadcast id
- จำนวน subnet = (2^(จำนวนบิตที่เป็น 1 ของ subnet mask ในตำแหน่งที่เป็น host id ของ ip address))

ARP : Ethernet Address Resolution Protocol

ในการสื่อสารใดๆ ก็ตาม จำเป็นต้องมีการสื่อสารผ่านตัวกลางในระดับ Physical เสมอ ซึ่งเป็นระดับล่างสุดในการสื่อสาร สำหรับในโปรโตคอล TCP/IP ถือว่าเป็นชั้น Link Layer นั่นเอง การสื่อสารในระดับนี้เป็นการสื่อสารระหว่าง Hardware Interface ในเน็ตเวิร์คเดียวกัน ซึ่งมองข้อมูลเป็น Ethernet Frame เท่านั้น จะไม่สนใจว่าข้อมูลภายในเป็นอย่างไร มีต้นทางอยู่ที่ไหน หรือปลายทางไปหาใคร แต่สิ่งที่โปรโตคอลในชั้นนี้สนใจก็คือ ข้อมูลที่ Network Layer ส่งมาให้นั้น จะต้องส่งไปยัง Hardware Interface ไหน ซึ่งการระบุ Hardware Interface จะใช้เป็น MAC Address มีลักษณะเป็นเลขฐาน 16 ยาว 6 ไบต์ เช่น 23:43:AA:5B:32:2C ซึ่งจะไม่มีอุปกรณ์ที่มีหมายเลขนี้ซ้ำกันเด็ดขาด

ทำไมต้องมี ARP ?

ในกรณีที่มีการส่งข้อมูลจาก interface หนึ่ง ทุกๆ interface ที่อยู่ในเน็ตเวิร์คเดียวกันจะได้รับข้อมูล แต่มีเพียงอินเทอร์เฟสที่มี MAC Address ตรงกับ MAC Address ของผู้รับที่ระบุในเฟรมข้อมูลเท่านั้น ที่จะนำข้อมูลนั้นไปประมวลผล ดังนั้นในการส่งข้อมูลจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง ฝั่งผู้ส่งจะต้องระบุ MAC Address ของผู้รับให้ถูกต้อง จึงจะสามารถส่งข้อมูลไปได้ สมมติว่า เครื่องคอมพิวเตอร์ ip 172.17.3.12 ต้องการติดต่อกับ 161.246.10.21 การทำงานในระดับ IP จะสั่งให้ ส่งข้อมูลไปยัง 172.17.3.1 ซึ่งเป็น default Router แต่ 172.17.2.12 จะรู้ได้อย่างไรว่า 172.17.2.12 มี MAC Address คืออะไร ?

จุดนี้เองที่จะต้องมีการใช้ ARP ในการสอบถาม MAC Address จากเครื่องที่เราต้องการส่งข้อมูล เมื่อได้รับ MAC Address ของผู้รับมาแล้วจึงสามารถเชื่อมต่อกับ เครื่องอีกฝั่ง เพื่อการสื่อสารในระดับสูงขึ้นไปได้

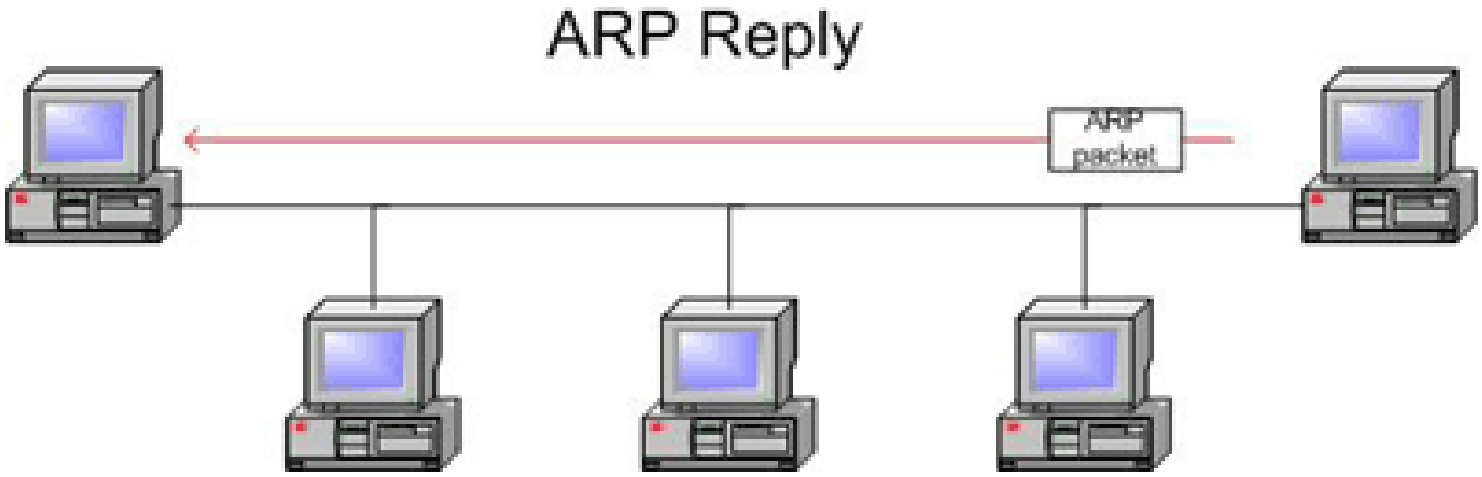
กลไกการทำงานของ ARP

การทำงานของ ARP เป็นเรื่องไม่ซับซ้อน มีเพียง 2 ขั้นตอนเท่านั้นคือ

1. เครื่องที่ต้องการสอบถาม MAC Address ส่ง ARP packet เรียกว่า **ARP Request** ซึ่งบรรจุ IP , MAC Address ของตนเอง และ IP Address ของเครื่องที่ต้องการทราบ MAC Address ส่วน MAC Address ปลายทางนั้น จะถูกกำหนดเป็น FF:FF:FF:FF:FF:FF ซึ่งเป็น Broardcast Address เพื่อให้ ARP packet ถูกส่งไปยังเครื่องทุกเครื่องที่อยู่ในเน็ตเวิร์คเดียวกัน

รูปที่ 3.1 ARP Request จะถูกส่งไปยังเครื่องทุกเครื่องในเน็ตเวิร์ค

2. เฉพาะเครื่องที่มี IP Address ตรงกับที่ระบุใน ARP Packet จะตอบกลับมาด้วย ARP Packet เช่นกัน โดยใส่ MAC Address และ IP Address ของตนเองเป็นผู้ส่ง และใส่ MAC Address และ IP Address ของเครื่องที่ส่งมาเป็นผู้รับ packet ที่ตอบกลับนี้เรียกว่า **ARP Reply**



รูปที่ 3.2 ARP Reply จะถูกตอบกลับมาจากเครื่องที่มี IP Address เพื่อบอก MAC Address ของตนเอง

ARP Cache

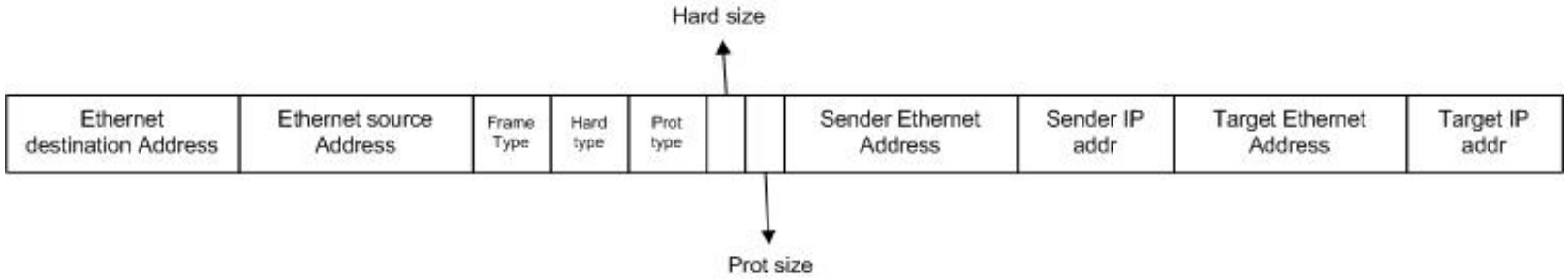
จากที่กล่าวมา จะเห็นว่ากระบวนการ ARP จะเกิดขึ้นทุกครั้งที่มีการส่ง IP datagram และกระบวการ ARP ก็กินเวลารับส่งข้อมูลและทรัพยากรในเน็ตเวิร์คพอสมควร โดยเฉพาะในจุดที่ต้องมีการ Broadcast ARP Request ซึ่งหากเป็นเช่นนั้นแบนวิธ์อันมีค่าของเน็ตเวิร์คคงหมดไปกับ ARP Packet ที่วิ่งพล่านในสายเคเบิ้ลแน่ๆ จึงมีการออกแบบบัฟเฟอร์เป็นตารางจับคู่ ระหว่าง ARP กับ IP Address เพื่อไม่ต้องส่ง ARP Request / Reply ทุกครั้งที่จะทำการส่ง IP datagram แต่ IP Address นั้นเป็นสิ่งที่ผู้ใช้กำหนดขึ้น เป็น Logical ซึ่งสามารถเปลี่ยนแปลงได้ ดังนั้น ข้อมูลในตารางนี้จึงต้องมีอายุการใช้งานโดยทั่วไป กำหนดให้เป็นเวลา 20 นาที เมื่อหมดเวลาแล้ว หากจะส่ง IP Datagram ครั้งต่อไป จะต้องทำการส่ง ARP Request ใหม่ ท่านสามารถเรียกดู ARP cache ในระบบปฏิบัติการ Linux ได้ด้วยคำสั่ง

#ARP [Enter]

```
[root@Sandy root]# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.17.0.1               ether   00:09:E9:95:D9:40   C                     eth0
172.17.3.28              ether   00:01:03:41:52:D5   C                     eth0
[root@Sandy root]#
```

รูปที่ 3.3 ผลของคำสั่ง arp แสดงตาราง arp cache สำหรับระบบปฏิบัติการ Linux

ARP Packet Format



รูปที่ 3.4 ARP Packet Format

ตำแหน่ง	ชื่อ	อธิบาย
ไบต์ 0-5	Ethernet Destination Address	ส่วนนี้อยู่ใน Header ของ Ethernet Frame ทั่วไป มีความหมายเป็น Address ปลายทาง ในกรณีของ ARP Request ข้อมูลในฟิลด์นี้จะเป็น FF:FF:FF:FF:FF:FF
ไบต์ 6-11	Ethernet Source Address	ส่วนนี้อยู่ใน Header ของ Ethernet Frame ทั่วไป มีความหมายเป็น Address ต้นทาง
ไบต์ 12-13	Ethernet Frame Type	ระบุถึงโปรโตคอลที่ Encapsulate อยู่ใน Ethernet Frame นี้ สำหรับ ARP จะต้องเป็น 0x0806

<table>
<tr><td>ไบต์ 14-15</td><td>Hard Type</td><td>ระบุถึงประเภทของ Hardware ที่โปรโตคอล ARP ถามอยู่ สำหรับกรณีนี้คือ Ethernet Address จะต้องเป็น 1</td></tr>
<tr><td>ไบต์ 16-17</td><td>Prot Type</td><td>ระบุชนิดของโปรโตคอลที่ถาม ว่าเป็น Hardware Address ของโปรโตคอลอะไร ในที่นี้คือ IP</td></tr>
<tr><td>ไบต์ 18</td><td>HArd Size</td><td></td></tr>
<tr><td>ไบต์ 19</td><td></td><td>ระบุขนาดของแอดเดรสในโปรโตคอลที่ถาม ซึ่งมีค่าเป็น 4 สำหรับ IP</td></tr>
<tr><td>ไบต์ 20-21</td><td>OP Field</td><td>เป็นการระบุว่าเป็น ARP ชนิดใดซึ่งมีค่าตามตาราง
<table>
<tr><td>ค่าในฟิลด์ OP</td><td>คำอธิบาย</td></tr>
<tr><td>1</td><td>ARP Request</td></tr>
<tr><td>2</td><td>ARP Reply</td></tr>
<tr><td>3</td><td>RARP Request</td></tr>
<tr><td>4</td><td>RARP Reply</td></tr>
</table></td></tr>
<tr><td>ไบต์ 22-27</td><td>Sender Ethernet Address</td><td>Ethernet Address ของผู้ส่ง มีค่าซ้ำกับในไบต์ที่ 6-11</td></tr>
<tr><td>ไบต์ 28-31</td><td>Sender IP Address</td><td>IP Address ของผู้ส่ง</td></tr>
<tr><td>ไบต์ 32-37</td><td>Target Ethernet Address</td><td>Ethernet Address ของผู้รับ ค่านี้จะว่างไว้ในกรณีของ ARP Request</td></tr>
<tr><td>ไบต์ 38-41</td><td>Target IP Address</td><td>IP Address ของผู้รับ</td></tr>
</table>

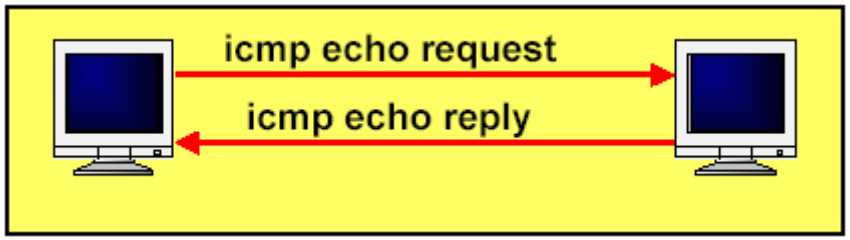
ICMP : Internet Control Message Protocol

ICMP เป็นโปรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานภาพของดาต้าแกรม โปรโตคอลนี้ทำงานในระดับ Network Layer เช่นเดียวกับ IP ในกรณีที่เกิดปัญหากับดาต้าแกรม เช่น เราเตอร์ไม่สามารถส่งดาต้าแกรมไปถึงปลายทางได้ ICMP จะถูกส่งออกไปยังโฮสต้นทางเพื่อรายงานข้อผิดพลาดที่เกิดขึ้น อย่างไรก็ดี ไม่มีอะไรรับประกันได้ว่า ICMP Message ที่ส่งไปนั้น จะถึงผู้รับจริงหรือไม่ หากมีการส่งดาต้าแกรมออกไปแล้วไม่มี ICMP Message ฟ้อง Error กลับมา ก็แปลความหมายได้สองกรณีคือ ข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อย หรืออาจจะมีปัญหา ในการสื่อสารทั้งการส่งดาต้าแกรม และ ICMP Message ที่ส่งกลับมาก็มีปัญหาระว่างทางก็ได้ ICMP จึงเป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (unreliable) ซึ่งจะเป็นหน้าที่ของ โปรโตคอลในระดับสูงกว่า Network Layer ในการจัดการให้การสื่อสารนั้นๆ มีความน่าเชื่อถือ

การใช้งาน ICMP

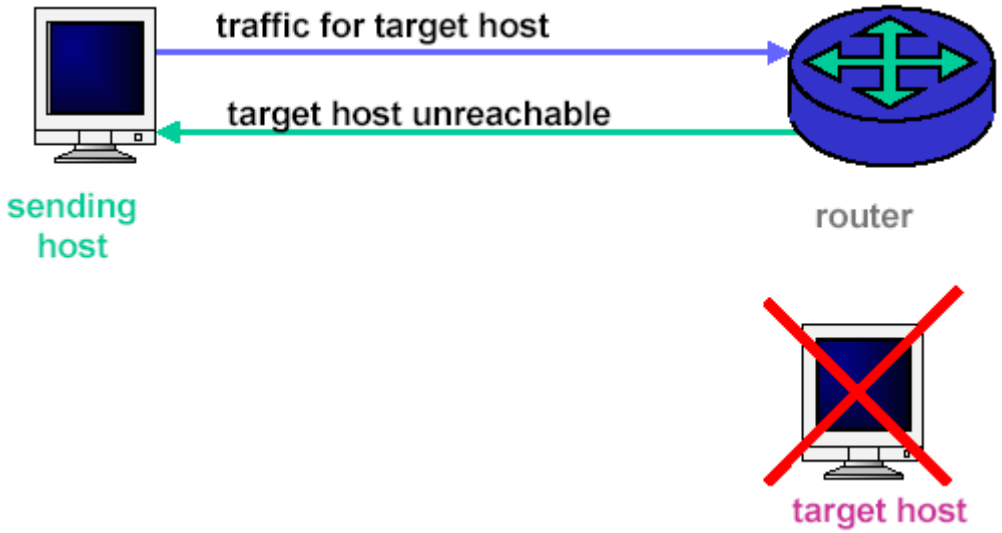
โดยทั่วไป ICMP มีการใช้งานในสองลักษณะคือ

1. **Query** ใช้สอบถามสถานะระหว่างกัน ในรูปที่ 4.1 เป็นการส่ง Echo request เพื่อถามสถานะของปลายทาง ซึ่งโฮสปลายทางอยู่ในสถานะปกติ สามารถทำการสื่อสารได้จะส่ง Echo Reply กลับมา



รูปที่ 4.1 การใช้งานโปรโตคอล ICMP เพื่อสอบถามสถานะระหว่างกัน

2. **Error Report** ใช้รายงานข้อผิดพลาดที่เกิดขึ้น เช่น หากไม่สามารถส่งดาต้าแกรมไปถึงปลายทางได้ เราเตอร์จะส่ง ICMP Message Host Unreachable กลับมารายงานโฮสต้นทาง (รูปที่ 4.2)



รูปที่ 4.2 การใช้งานโปรโตคอล ICMP เพื่อรายงานข้อผิดพลาดที่เกิดขึ้น

การรายงานข้อผิดพลาดของ ICMP นั้น จะ ไม่รายงานข้อผิดพลาดในบางกรณีคือ

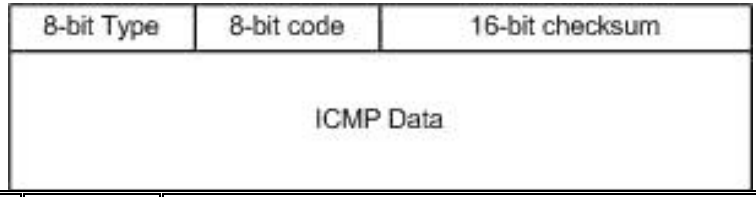
1. เกิดความผิดพลาดในการส่ง ICMP เสียเอง

2. การส่งข้อมูลเป็นการส่งข้อมูลแบบ Multicast หรือ Broadcast

3. มีการแบ่งดาต้าแกรมออกเป็นส่วนย่อยๆ เรียกว่าแฟรกเมนต์ ในกรณีนี้ จะส่ง ICMP Message สำหรับแฟรกเมนต์แรกเพียงครั้งเดียว ไม่ต้องแจ้งกลับทุกแฟรกเมนต์ เพราะถือว่าเป็นดาต้าแกรมเดียวกัน

4. Address ต้นทางไม่ได้เจาะจงโฮส เช่น Address ที่เป็น 0 ทั้งหมด Address ที่เป็น loopback หรือ Address ที่เป็นแบบ Multicast หรือ Broadcast

ICMP Message



ตำแหน่ง	ชื่อ	อธิบาย
บิต 0-7	Type	ประเภทของ ICMP Message
บิต 8-15	Code	รหัสของ ICMP Message
บิต 16-31	Checksum	เป็นตัวตรวจสอบความถูกต้องของ ICMP Massage ทั้งหมด
ไม่แน่นอน	Data	ข้อมูลภายใน ICMP Message ขึ้นอยู่กับ Type *

รูปที่ 4.3 แสดงรูปร่างของ ICMP Message

* ตำแหน่งต่างๆของข้อมูลภายใน Data จะขึ้นอยู่กับชนิดของ ICMP นั้นๆ ซึ่งมีรายละเอียดค่อนข้างมาก เพื่อไม่ให้เนื้อหายืดเยื้อจนเกินไป จะไม่นำมากล่าวถึงในเอกสารฉบับนี้ ท่านสามารถศึกษาโครงสร้างของ Data ของ ICMP แต่ละชนิดได้จาก RFC 792

ICMP Message Type

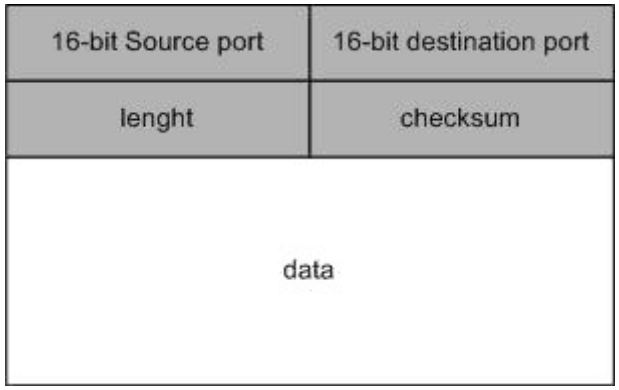
TYPES ของ ICMP มีความหมายดังนี้

TYPE	Description
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect Message
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request
14	Timestamp Reply
15	Information Request (No Longer Used)
16	Information Reply (No Longer Used)
17	Address Mask Request
18	Address Mask Reply

UDP : User Datagram Protocol

UDP เป็นโปรโตคอลที่ถูกออกแบบมาให้ทำหน้าที่รับส่งข้อมูลโดยมีขั้นตอนการทำงานไม่ซับซ้อนและทำงานได้รวดเร็ว แต่มีจุดด้อยคือไม่มีความน่าเชื่อถือ (unreliable) และเป็นการสื่อสารแบบไม่ต่อเนื่อง (connectionless) โปรโตคอล UDP ทำงานในชั้น Transport Layer ซึ่งจะต้องพึ่งพาโปรโตคอล IP ในการรับส่งข้อมูล

UDP Header



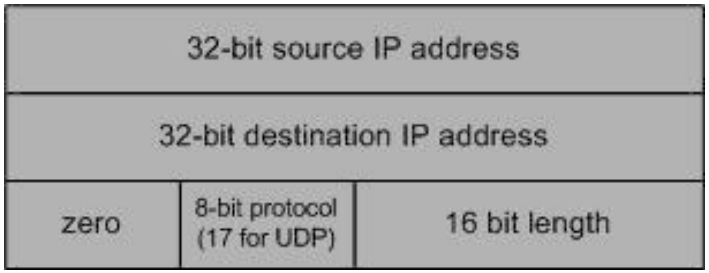
รูปที่ 5.1 UDP Header

ตำแหน่ง	ชื่อ	อธิบาย
บิต 0-15	Source port number	หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้ มีความยาว 16 บิต
บิต 16-31	destination port number	หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม มีความยาว 16 บิตเช่นกัน
บิต 32-47	UDP length	ความยาวของดาต้าแกรม ทั้งส่วน Header และ data นั่นหมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header
บิต 48-63	Checksum	เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวนด้วย

UDP Checksum

Checksum เป็น เลข 16 บิตถูกคำนวนด้วยวิธี one's complement โดยนำ Pseudo Header และข้อมูลทั้งหมดใน UDP Datagram มาคำนวน

Pseudo Header เป็นข้อมูลที่อยู่ในส่วนของ IP Header ประกอบด้วยฟิลด์ source ip address,destination ip address , zero , protocol , udp length ดังแสดงในรูปที่ 5.2



รูปที่ 5.2 Pseudo Header

หากค่า Checksum ที่คำนวนออกมาเป็น 0 ค่า checksum จะถูกเซ็ตเป็น 1 ทั้งหมดแทน (มีค่าเท่ากันในระบบ 1's complement) ทั้งนี้เพราะในบางแอพพลิเคชั่นที่ไม่ต้องการตรวจสอบค่า checksum ในระดับ UDP จะเซ็ตค่านี้เป็น 0 (disable checksum)

TCP : Transmistion Control Protocol

TCP เป็นโปรโตคอลที่ใช้สื่อสารระหว่างโฮสที่มีความน่าเชื่อถือ จะเห็นได้ว่าโปรโตคอลในระดับ IP หรือแม้กระทั่ง UDP จะสนใจ ข้อมูลเพียง 1 ดาต้าแกรม กลไกของโปรโตคอลจะมีหน้าที่ตรวจสอบความถูกต้องเพียงเฉพาะดาต้าแกรมนั้น ๆ เมื่อจะทำ การส่งดาต้าแกรมใหม่ก็จะถือว่าเป็นข้อมูลชุดใหม่ที่ไม่มีความสัมพันธ์ใด ๆ กับข้อมูลดาต้าแกรมอื่น (การสื่อสาร 1 ครั้ง จึงใช้เพียง 1 ดาต้าแกรม) แต่สำหรับ TCP แล้วจะเห็นว่าข้อมูลนั้นเป็น stream คือมีความสัมพันธ์ต่อเนื่องกัน มีกลไกในการตรวจสอบทั้งด้านส่ง และด้านรับเพื่อให้แน่ใจว่าสามารถสื่อสารกันได้จริงจึงจะมีการส่งรับข้อมูลเกิดขึ้น ตลอดจนการยกเลิกการติดต่อก็มีกลไกสำหรับแจ้งให้อีกฝั่งทราบ ทำให้การสื่อสารด้วย TCP จึงเสมือนว่าทั้ง 2 ฝ่ายคือฝ่ายรับและฝ่ายส่งได้ทำการต่อสาย เน็ตเวิร์กถึงกัน (connected) ตลอดเวลาที่การรับส่งข้อมูลจนกระทั่งการสื่อสารทั้งหมดเสร็จสิ้นจึงจะทำการยกเลิกการเชื่อมต่อนั้นเสีย

จุดเด่นประการสำคัญของ TCP ที่กล่าวถึงอยู่เสมอคือ ความมีเสถียรภาพและความถูกต้องของการสื่อสารซึ่งมีความเชื่อถือได้สูง คุณสมบัติที่ทำให้ TCP มีข้อดีดังกล่าวคือ

1. ข้อมูลที่จะส่งผ่าน TCP จะถูกนำมาแตกย่อยออกเป็นส่วน ๆ ให้มีขนาดเหมาะสมสำหรับการส่งข้อมูล โดย TCP มีกลไกในการพิจารณาว่าขนาดเท่าใดจะทำให้การรับ - ส่งนั้นมีประสิทธิภาพและน่าเชื่อถือสูงสุด โดยข้อมูลแต่ละชุดที่แบ่งออกและ ทำการส่งโดย TCP แต่ละครั้งจะเรียกว่า TCP เซกเมนต์

2. ในการส่งข้อมูลแต่ละครั้ง TCP จะมีการจับเวลาไว้เสมอ เพื่อรอการตอบรับจากผู้รับว่าได้รับข้อมูลถูกต้อง หากหมดเวลาแล้วไม่มีการตอบรับ TCP จะถือว่าข้อมูลไปไม่ถึงและทำการแก้ปัญหาที่เกิดขึ้น เช่น ยกเลิกการติดต่อ , ส่งข้อมูลซ้ำ ทำให้ Application ทราบสถานะการส่งข้อมูลตลอดเวลา

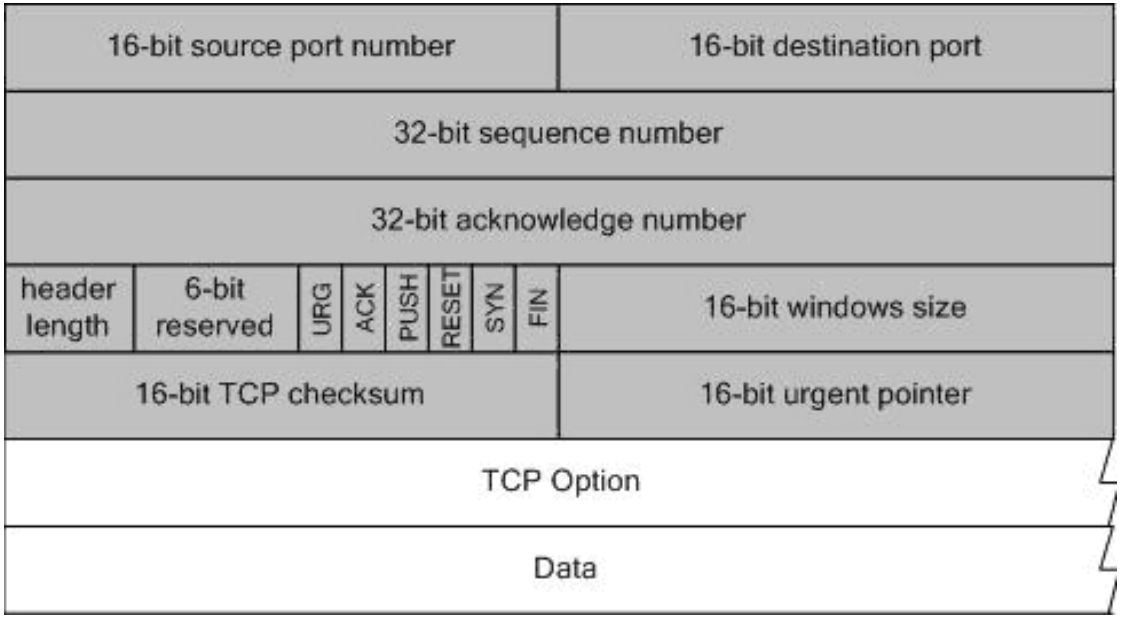
3.TCP มี checksum ซึ่งจะครอบคลุมทั้ง TCP Header และ TCP Data เพื่อเป็นการป้องกันและตรวจสอบว่าข้อมูล ที่ส่งมานั้นถูกต้อง และไม่ได้ถูกแก้ไขระหว่างทาง หาก TCP ได้รับข้อมูลที่ทำการตรวจสอบกับ checksum แล้วปรากฏว่า มีความผิดพลาดเกิดขึ้น TCP จะทิ้งข้อมูลที่ได้รับและจะไม่ทำการตอบรับข้อมูลนั้นกลับไปยังผู้ส่ง คือถือเสมือนว่าไม่ได้รับ ข้อมูลนั้น เพื่อให้ทางฝ่ายผู้ส่งทำการส่งใหม่หรือหาข้อบกพร่องและพยายามแก้ไขตามแต่แอพพลิเคชั่นทางฝ่ายผู้ส่งเห็น สมควร

4. เนื่องจาก TCP อาศัย IP ในการส่งข้อมูล ซึ่ง IP เองอาจจะถูกแฟรกเมนต์ได้ และทำให้ข้อมูลที่ถูกแฟรกเมนต์นั้นส่งถึง ปลายทางในลำดับที่ไม่ถูกต้องได้ หน้าที่ของ TCP เมื่อรับข้อมูลที่แฟรกเมนต์มานั้นจะต้องนำข้อมูลแต่ละส่วนมาประกอบ รวมกันให้ถูกต้องสมบูรณ์ก่อนจะส่งไปยัง Application Layer ต่อไป

5. การส่ง - รับข้อมูลด้วย IP อาจจะมีกรณีที่ IP Datagram นั้นถูกส่งซ้ำขึ้นได้ TCP ที่รับข้อมูลซ้ำดังกล่าวจะต้องทราบว่า เป็น IP Datagram ที่ซ้ำและไม่นำข้อมูลไปใช้งาน

6. TCP มีกลไกควบคุมการไหลของข้อมูล (Flow Control) โดยการควบคุมนี้จะต้องอาศัยลำดับของการรับส่งที่ถูกต้อง และสัมพันธ์กันทั้ง 2 ฝั่ง ในขณะเดียวกันข้อมูลที่ส่งนั้นจะต้องอาศัย IP หลายดาต้าแกรมจึงจะได้รับข้อมูลครบทั้งหมด ดังนั้นในการรับข้อมูลทางฝ่ายรับจึงต้องเตรียมบัฟเฟอร์ไว้จำนวนหนึ่งเพื่อรอรับข้อมูลและรวบรวมข้อมูลทั้งชุดให้อยู่ใน บัฟเฟอร์ก่อนที่จะทำการจัดเรียงข้อมูล ตรวจสอบความถูกต้องแล้วจึงส่งต่อไปยังแอพพลิเคชั่น ด้วยเหตุผลดังกล่าวจะเห็น ได้ว่าขนาดของข้อมูลมิได้ถูกจำกัดที่ขนาดของดาต้าแกรมใด ๆ ข้อมูลที่ส่งอาจจะมีขนาดใหญ่มากอยู่ในหลายดาต้าแกรม ก็เป็นได้ ดังนั้นเพื่อป้องกันการส่งข้อมูลขนาดใหญ่เร็วเกินไปจนทำให้ทางฝ่ายรับไม่มีหน่วยความจำเพียงพอที่จะเป็น บัฟเฟอร์ที่พักข้อมูล การส่งข้อมูลจึงถูกจำกัดโดยจะอนุญาตให้ทำการส่งข้อมูลได้เท่าที่ฝ่ายรับมีบัฟเฟอร์เพียงพอเท่านั้น

TCP Header



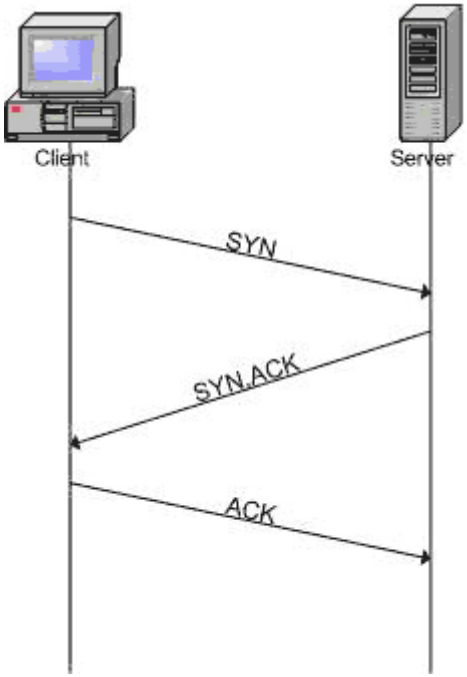
รูปที่ 6.1 TCP Header

ชื่อ	อธิบาย
Source Port Number	หมายถึงพอร์ตที่โฮสต์ต้นทางใช้ในการสื่อสารกันของเซสชั่นนี้ และ TCP/IP จะใช้พอร์ตนั้นไปตลอดตราบใดที่การสื่อสารในเซสชั่นนี้ยังไม่ยุติลง โดยทั่วไปพอร์ตนี้จะเรียกว่า "ไคลเอนต์พอร์ต" คือพอร์ตที่ไคลเอนต์เปิดขึ้น มาเพื่อรอการตอบรับจากเซิร์ฟเวอร์ (พิจารณาจากทิศทางของแพ็กเก็ตที่ส่งมาจากไคลเอนต์ไปยังเซิร์ฟเวอร์) ไคลเอนต์พอร์ต จะมีหมายเลขไม่แน่นอนและเปลี่ยนไปทุกครั้งที่มีการเริ่มการเชื่อมต่อใหม่ เป็นพอร์ตที่ถูกเปิดไว้ในระยะเวลาสั้น ๆ (ephenumeral port) ค่าที่เป็นไปได้ของพอร์ตนี้ขึ้นอยู่กับการจัดสรรของระบบปฏิบัติการ ในการกำหนดขอบเขตของพอร์ตเหล่านี้ส่วนใหญ่จะมีค่า อยู่ในช่วง 1024 - 5000
Destination Port Number	หมายถึงหมายเลขพอร์ตบนโฮสต์ปลายทางที่โฮสต์ต้นทางต้องการติดต่อด้วย โดยนัยแล้วจะหมายถึงแอพพลิเคชั่นที่ให้บริการอยู่พอร์ตนั้นที่โฮสต์ปลายทางนั่นเอง พอร์ตนั้นจะเรียกอีกอย่างหนึ่งว่า "เซิร์ฟเวอร์พอร์ต" หมายเลขพอร์ตที่เปิดไว้จะขึ้นอยู่กับแอพพลิเคชั่นที่ให้บริการ โดยทั่วไปแอพพลิเคชั่นแต่ละประเภทจะมีหมายเลขพอร์ต เป็นมาตรฐานสำหรับให้ำคลเอนต์ได้เรียกใช้บริการ
Sequence Number	เป็นฟิลด์ที่ระบุถึงหมายเลขลำดับที่ใช้อ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อให้ทั้ง 2 ฝ่าย จะได้รับทราบตรงกันว่าเป็นข้อมูลของชุดใด การนำไปใช้งานจะได้ไม่ปะปนกัน และมีลำดับที่ถูกต้อง เนื่องจากการสื่อสารข้อมูลผ่าน TCP นั้นจังหวะและลำดับเป็นส่วนสำคัญของโปรโตคอลไม่ยิ่งหย่อนไปกว่าข้อมูลใน TCP Header รวมไปถึง การที่ข้อมูลในแต่ละ TCP Segment อาจจะถูกทำการแฟรกเมนต์ในเลเยอร์ของ IP ถัดลงไป ทำให้ข้อมูลถูกแบ่งออกและส่งไปในลำดับที่ไม่เรียงกัน หากไม่มีจุดอ้างอิงของข้อมูลก็จะไม่สามารถอ่านข้อมูลกลับใหม่ได้อย่างสมบูรณ์และถูกต้อง การส่งข้อมูลและการตอบรับจะใช้ฟิลด์ นี้เป็นตัวยืนยันระหว่างกันเสมอ

<table>
<tr><td>Acknowledge Number</td><td>ทำหน้าที่เช่นเดียวกับ Sequence Number ต่างกันตรงที่เป็น Sequence Number ซึ่งในการตอบรับกล่าวคือ เนื่องจาก Sequence Number ที่ใช้ในการอ้างอิงนั้นผู้ที่เริ่มส่งข้อมูลจะเป็นผู้กำหนดเลขขึ้น มาและส่งไปพร้อมกับการสร้างการเชื่อมต่อครั้งใหม่แต่สำหรับฝ่ายที่ถูกติดต่อก็จำเป็นต้องกำหนดหมายเลขสำหรับใช้อ้างอิง ในการตอบรับเช่นกัน ค่าที่อยู่ใน Acknowledge Number ก็คือหมายเลขที่ใช้อ้างอิงในการตอบรับนี้</td></tr>
<tr><td>Header Length</td><td>โดยปกติความยาวของ TCP Header จะเท่ากับ 20 ไบต์ แต่ถ้าหากมีการใช้ Option อาจจะทำให้ขนาดของเฮดเดอร์ยาวขึ้นตามข้อมูลที่ต้องเพิ่มมาจาก Option นั้น แต่ทั้งหมดแล้วจะไม่เกิน 60 ไบต์</td></tr>
<tr><td></td><td>เป็นข้อมูลในระดับบิตที่ใช้เป็นตัวบอกคุณสมบัติของ TCP Segment ที่กำลังส่งอยู่นั้น และใช้เป็นตัวควบคุมจังหวะ การรับส่งข้อมูลด้วย ซึ่ง Flag ทั้งหมดมีอยู่ 6 บิต แต่ละบิตมีชื่อและมีความหมายดังนี้
<table>
<tr><td>Flag</td><td>อธิบาย</td></tr>
<tr><td>URG</td><td>ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent pointer)</td></tr>
<tr><td>ACK</td><td>แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้</td></tr>
<tr><td>DSH</td><td>เพื่อแจ้งให้ผู้รับข้อมูลทราบว่า ควรจะส่งข้อมูล Segment นี้ไปยังโพรเซสที่กำลังรออยู่ทันที</td></tr>
<tr><td>RST</td><td>ใช้ในการณีที่เกิดการสับสนขึ้นด้วยเหตุผลต่างๆ เช่น โฮสต์มีปัญหา ให้เริ่มต้นสื่อสารกันใหม่</td></tr>
<tr><td>SYN</td><td>ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง</td></tr>
<tr><td>FIN</td><td>ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ</td></tr>
</table>
</td></tr>
</table>

Window Size	เป็นขนาดของการรับ - ส่งข้อมูลในแต่ละครั้งที่ทางฝ่ายผู้รับจะสามารถรับได้ เนื่องจากในการรับข้อมูลนั้น ทางผู้รับจะต้องจัดเตรียมหน่วยความจำในการพักข้อมูลที่มากจาก TCP และทำการ Demultiplex ออกมา หากไม่มีการตกลง ถึงขนาดที่ทางฝ่ายรับสามารถรับได้ ก็จะทำให้การสื่อสารข้อมูลไม่สมดุล และฝ่ายรับอาจจะประมวลผลทัน ซึ่งจะส่งผลให้ต้องส่ง ข้อมูลซ้ำหลายครั้ง
Checksum	ฟิลด์ที่ใช้ในการตรวจสอบความถูกต้องของข้อมูลใน TCP เซกเมนต์ใช้ระบุหมายเลข Sequence Number ของ TCP เซกเมนต์ล่าสุดที่อยู่ในโหมด Urgent
Urgent Pointer	ข้อมูลเพิ่มเติมซึ่งจะอยู่ใน TCP Header เมื่อมีการตั้งค่า option บางอย่างที่ต้องการข้อมูลเพิ่มเติมซึ่งไม่มีใน TCP Header เช่น MSS, Strict Route

Connection Establishment

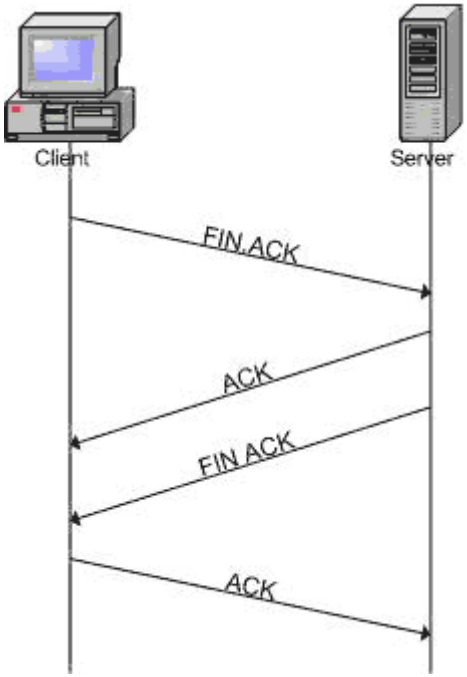


รูปที่ 6.2 3-way handshake

ก่อนที่จะเริ่มต้นการสื่อสาร จะต้องมีการส่งสัญญาณเพื่อบอกโฮสต์อีกฝั่งหนึ่งให้เตรียมตัวติดต่อ ซึ่งกระบวนการที่ใช้มีชื่อเรียกว่า 3-Way Hand Shake มีขั้นตอนคือ

1. เครื่องไคลเอนต์จะทำการส่งเซกเมนต์ โดยเปิด SYN Flag ระบุหมายเลขพอร์ตที่ต้องการติดต่อบนเซิร์ฟเวอร์และระบุหมายเลข ลำดับของข้อมูล (ISN - Initial Sequence Number)
2. เครื่องเซิร์ฟเวอร์เมื่อได้รับข้อมูลเซกเมนต์จากข้อ 1 ก็จะตอบกลับด้วยการเพิ่มค่า ISN ที่ได้รับขึ้นอีก 1 พร้อมทั้งระบุหมายเลขลำดับ (ISN) ของตนเอง และเปิด SYN กับ ACK Flag
3. ไคล์เอนต์เมื่อได้รับการตอบกลับจากเซิร์ฟเวอร์ตามข้อ 2 ก็จะทำการตอบรับกลับไป โดยการเพิ่มค่า ISN ของเซิร์ฟเวอร์ขึ้นอีก 1 และเปิด ACK Flag เมื่อผ่านการสร้าง connection ทั้ง 3 ขั้นตอนแล้ว ตอนนี้ทั้งไคล์เอนต์ และเซิร์ฟเวอร์เปรียบเสมือนมีการเชื่อมต่อถึงกันแล้ว สถานะของการเชื่อมต่อในขณะนี้เรียกว่า **Established**

Connection Termination



รูปที่ 6.1 TCP Header

เมื่อการสื่อสารของทั้งสองฝั่งจบลง และไม่ต้องการรับส่งข้อมูลอีกต่อไป จะต้องทำตามขั้นตอนการยุติการสื่อสารเพื่อให้การสื่อสารจบลงอย่างสมบูรณ์ ซึ่งมีอยู่ 4 ขั้นตอนคือ

1. ไคลเอนต์ทำการส่ง ISN พร้อมกับ FIN ACK Flag ไปยังเซิร์ฟเวอร์
2. เซิร์ฟเวอร์ทำการตอบรับ ISN และบวกค่า ISN อีก 1 พร้อม ACK Flag
3. เซิร์ฟเวอร์ทำการส่ง ISN พร้อมกับ FIN ACK Flag ไปยังไคลเอนต์
4. ไคลเอนต์ทำการตอบรับ ISN และบวกค่า ISN อีก 1 พร้อม ACK Flag

การยุติการเชื่อมต่อ โดยส่ง FIN ACK ออกไปมีความหมายคือ โฮสต์ที่ส่งไม่มีข้อมูลจะส่งไปอีก มิใช่ต้องการปิดการสื่อสารทั้งหมดในทันที ดังนั้นจึงต้องทำทั้งสองทาง การสื่อสารจึงจะยุติลงอย่างสมบูรณ์ ในการใช้งานจริง อาจมีการยุติการสื่อสารเพียงด้านเดียว คือหยุดส่งข้อมูล แต่ยังคงเปิดพอร์ตไว้รอรับข้อมูลจากอีกด้านหนึ่ง ทั้งนี้ขึ้นอยู่กับลักษณะการใช้งาน การปิดพอร์ตสื่อสารเพียงด้านเดียวเช่นนี้ เรียกว่า **Half-Close**