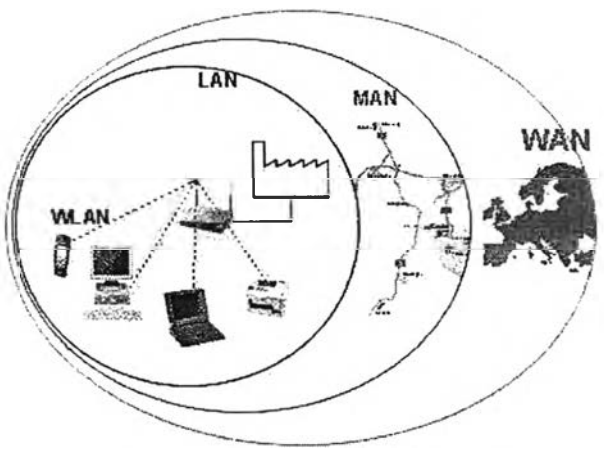
บทที่ 3
ระบบเครือข่าย

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) เกิดจากความต้องการแลกเปลี่ยนข้อมูลระหว่างคอมพิวเตอร์อย่างรวดเร็วและมีประสิทธิภาพ เนื่องจากวิธีแลกเปลี่ยนข้อมูลในแบบเดิม เช่น การใช้แผ่นดิสก์ (Floppy Disk) หรือการใช้กระดาษเป็นเอกสารในการส่ง มีความล่าช้าและต้องแปลงข้อมูลไปมา ทำให้ไม่สะดวกในการแลกเปลี่ยนข้อมูลระหว่างคอมพิวเตอร์ จึงมีการเริ่มส่งข้อมูลโดยตรงระหว่างคอมพิวเตอร์ขึ้น ระบบเครือข่ายสามารถแบ่งประเภทได้หลายแบบ เช่น แบ่งตามการติดตั้งทางภูมิศาสตร์ , แบ่งตามหน้าที่การทำงานของคอมพิวเตอร์ในระบบเครือข่าย เป็นต้น

ในปัจจุบันระบบเครือข่ายสามารถแบ่งตามการติดตั้งทางภูมิศาสตร์ ได้เป็น



รูปที่ 3.1 การแบ่งระบบเครือข่ายทางภูมิศาสตร์

1. ระบบเครือข่ายท้องถิ่น (Local Area Network : Lan) เป็นระบบเครือข่ายระยะใกล้ ครอบคลุมพื้นที่จำกัด เช่น ภายในอาคารเดียวกัน เป็นต้น

2. ระบบเครือข่ายระดับเมือง (Metropolitan Area Network : Man) เป็นระบบเครือข่ายขนาดกลาง ลักษณะคล้ายระบบเครือข่ายท้องถิ่น แต่ครอบคลุมพื้นที่กว้างขึ้น ใช้ภายในเมืองหรือจังหวัด สามารถรับส่งข้อมูลและโทรศัพท์พร้อมกัน ปัจจุบันเทคโนโลยีที่ใช้ในระบบเครือข่ายระดับเมืองเป็นเทคโนโลยีเดียวกับระบบเครือข่ายวงกว้าง ทำให้บางกรณีการแบ่งระบบเครือข่ายจะไม่มีระบบเครือข่ายระดับเมือง โดยถือว่าเป็นส่วนหนึ่งของระบบเครือข่ายวงกว้าง

3. ระบบเครือข่ายวงกว้าง (Wide Area Network : Wan) เป็นเครือข่ายขนาดใหญ่ สามารถเชื่อมต่อระบบเครือข่ายท้องถิ่นหลายกลุ่มเข้าด้วยกัน การส่งข้อมูลมีรูปแบบ เช่น ส่งผ่านดาวเทียม , สายเคเบิล , โมเดม (modem) หรือสายโทรศัพท์ เครือข่ายวงกว้างที่รู้จักดีคือ อินเตอร์เน็ต (Internet)

ปัจจุบันมีระบบเครือข่ายอีกตัวหนึ่งที่เป็นที่นิยม คือ ระบบเครือข่ายท้องถิ่นไร้สาย (Wireless Local Area Network) การเชื่อมต่อจะไม่ใช้สายสัญญาณ แต่จะใช้อากาศเป็นตัวกลางและใช้คลื่นวิทยุ (Radio Frequency) หรือแสงอินฟราเรดส่งข้อมูล ทำให้มีความคล่องตัวสูง

ในวิทยานิพนธ์บทนี้จะพูดถึงพื้นฐานของระบบเครือข่ายท้องถิ่นเป็นหลัก เนื่องจากระบบเครือข่ายท้องถิ่นมีความเร็วในการรับส่งข้อมูลสูงสุด อีกทั้งมีราคาและมีระยะทางในการส่งข้อมูลที่เหมาะสมกับการส่งข้อมูลสัญญาณภาพจากคอมพิวเตอร์ไปยังแผงแสดงภาพไดโอดเปล่งแสงที่อยู่ห่างกันไม่มากนัก

3.1 OSI Model [10,11]

องค์การมาตรฐานนานาชาติ (The International Organization for Standardization : ISO) กำหนดโครงสร้างมาตรฐาน OSI Model (Open System Interconnection) หรือ OSI 7-Layer Reference Model ขึ้น ในปี ค.ศ. 1984 เพื่อให้คอมพิวเตอร์สามารถติดต่อสื่อสารกันโดยไม่ขึ้นกับมาตรฐานเฉพาะของผู้ผลิต มาตรฐานนี้แบ่งการติดต่อสื่อสารออกเป็น 7 ชั้นย่อยๆ

การทำงานชั้นที่อยู่สูงกว่าจะเรียกใช้ชั้นที่อยู่ต่ำกว่าที่อยู่ติดกัน โดยไม่จำเป็นต้องทราบถึงวิธีการทำงานของชั้นต่ำกว่าที่ติดกันนั้น และจะไม่มีการทำงานข้ามชั้น ผู้ใช้จะติดต่อผ่านชั้นที่ 7 ซึ่งอยู่ด้านบนสุด ชั้นบน OSI Model สามารถแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ

- Application-oriented Layer เป็นชั้นที่รับส่งข้อมูลระหว่างผู้ใช้กับโปรแกรมต่างๆ ได้แก่ชั้น 7 , 6 และ 5
- Network-dependent Layer เป็นชั้นที่รับส่งข้อมูลผ่านทางสายส่ง และควบคุมการรับส่งข้อมูล อีกทั้งตรวจสอบความผิดพลาดในการส่ง ได้แก่ชั้น 4 , 3 , 2 และ 1

ชั้นการทำงานของ OSI Model จะชื่อเรียกและการทำงานคร่าวๆดังนี้

ชั้นที่ 7 Application Layer ทำหน้าที่รับคำสั่งจากผู้ใช้ส่งให้คอมพิวเตอร์แปลความหมายและทำงานตามคำสั่งที่ได้รับ

ชั้นที่ 6 Presentation Layer รับผิดชอบในรูปแบบการรับส่งข้อมูลผ่านระบบเครือข่าย ทั้งในเรื่อง ขั้นตอน ข้อบังคับ และการเข้ารหัสต่างๆ

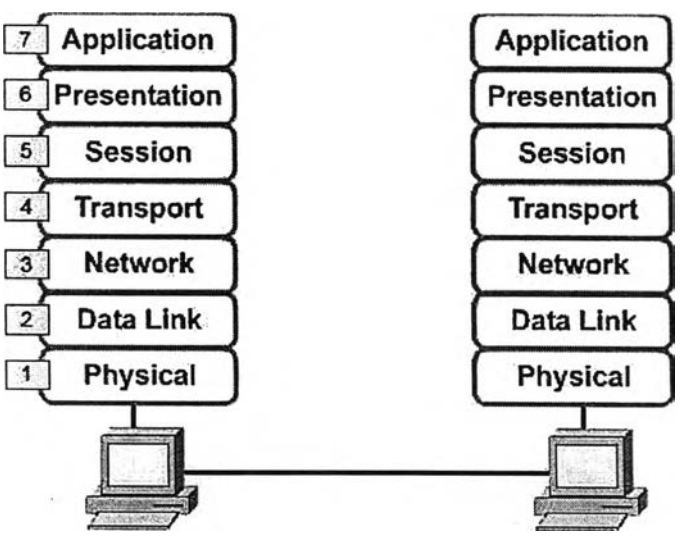
ชั้นที่ 5 Session Layer ควบคุมการแลกเปลี่ยนข้อมูล ควบคุมจังหวะการรับส่งข้อมูล (Synchronization) กำหนดวิธีรับส่งข้อมูลว่าเป็นแบบทางเดียว (Half Duplex / Unidirectional) หรือแบบ 2 ทาง (Full Duplex / Bi-directtional)

ชั้นที่ 4 Transport Layer แปลงข้อมูลในระดับสูงให้อยู่ในระดับฮาร์ดแวร์ พร้อมทั้งควบคุมการรับส่งจากด้านส่งไปถึงด้านรับตามจังหวะควบคุมจากชั้น 5

ชั้นที่ 3 Network Layer ทำหน้าที่จัดเส้นทางข้อมูลระหว่างด้านรับและส่ง

ชั้นที่ 2 Datalink Layer ทำหน้าที่รับส่งและตรวจสอบความผิดพลาดของข้อมูล ข้อมูลในชั้นนี้จะอยู่ในรูปของเฟรม (Frame) ซึ่งรูปร่างของเฟรมจะต่างกันตามลักษณะที่ใช้ เช่น อีเธอร์เน็ต (Ethernet) โทเคนริง (Token Ring) เป็นต้น

ชั้นที่ 1 Physical Layer รับผิดชอบการส่งข้อมูลในระดับฮาร์ดแวร์ที่เป็นข้อมูลบิต



รูปที่ 3.2 OSI Model

3.2 **องค์ประกอบของระบบเครือข่ายท้องถิ่น** [10,11]

คอมพิวเตอร์จะติดต่อกันผ่านระบบเครือข่ายได้ จำเป็นต้องมีองค์ประกอบต่างๆ ซึ่งแบ่งออกเป็นส่วนใหญ่ๆ ดังนี้

1. โปรโตคอล (Protocol)

โปรโตคอล หรือสถาปัตยกรรมเครือข่าย คือมาตรฐานสำหรับสื่อสารระหว่างคอมพิวเตอร์และอุปกรณ์ที่ใช้งานในระบบเครือข่าย เพื่อให้อุปกรณ์ต่างๆกัน สามารถสื่อสารกันได้ โปรโตคอลที่นิยมใช้งานในปัจจุบันได้แก่ ชุดโปรโตคอล TCP/IP (Transmission Control Protocol / Internet Protocol) , ชุดโปรโตคอล IPX/SPX (Internet Packet Exchange / Sequenced Packet Exchange) , ชุดโปรโตคอล NetBEUI (NetBIOS Extended User Interface โดย NetBIOS ย่อมาจาก Network Basic Input/Output System) และ ชุดโปรโตคอล AppleTalk

2. การเชื่อมต่อเครือข่ายท้องถิ่น

การเชื่อมต่อเครือข่ายท้องถิ่นจะอยู่ในชั้น Data Link และ Physical ของ OSI model รูปแบบและอุปกรณ์ในการเชื่อมต่อ จะแตกต่างกันไปตามบริษัท ในแต่ละแบบมีข้อกำหนดในเรื่องความเร็วในการรับส่ง , จำนวนเครื่องในระบบเครือข่าย , รูปแบบโทโปโลยี , ชนิดสายสัญญาณ ฯ แตกต่างกันไป การเชื่อมต่อเครือข่ายที่พบเห็นในปัจจุบันมีๆ ดังนี้ อีเธอร์เน็ต (Ethernet) , โทเคนริง (Token Ring) , ATM (Asynchronous Transfer Mode) , FDDI (Fiber Distribution Data Interface) และ CDDI (Copper Distribution Data Interface)

การเชื่อมต่อเครือข่าย IEEE ได้กำหนดมาตรฐาน 802 เพื่อใช้เป็นมาตรฐานของส่วนประกอบในระบบเครือข่าย ที่ทำงานในชั้น Physical Layer และ Data-Link Layer ของ OSI model มาตรฐาน IEEE 802 แบ่งออกเป็นมาตรฐานย่อยดังแสดงในตารางที่ 3.1 โดยในแต่ละมาตรฐานจะมีการแบ่งย่อยลงไปอีก

มาตรฐาน IEEE	รายละเอียด
802.1	การบริหารจัดการระบบเครือข่าย
802.2	มาตรฐานควบคุมของ Data Link Layer ได้แก่ LLC และ MMC
802.3	มาตรฐานอีเธอร์เน็ต
802.4	มาตรฐานโทเคนบัส*
802.5	มาตรฐานโทเคนริง
802.6	มาตรฐาน MAN*
802.7	ข้อกำหนดการส่งสัญญาณแบบ Broadband*
802.8	ข้อกำหนดการส่งสัญญาณใยแก้วนำแสง (Fiber Optic TAG)
802.9	มาตรฐาน Isochronous Ethernet (IsoEnet)*
802.1	มาตรฐานความปลอดภัยบนระบบเครือข่าย*
802.11	มาตรฐานระบบเครือข่ายท้องถิ่นไร้สาย
802.12	กำหนดลำดับความสำคัญของความต้องการใช้งานบนระบบเครือข่าย
802.13	ไม่ใช้งาน
802.14	มาตรฐานสาย Modem*
802.15	กำหนดพื้นที่เครือข่ายไร้สายส่วนบุคคล (Wireless Personal Area Network)
802.16	มาตรฐาน Boardband ไร้สาย

* ยกเลิกการใช้งาน

ตารางที่ 3.1 มาตรฐาน IEEE 802

3.3 องค์ประกอบของเครือข่ายท้องถิ่นที่เลือกใช้ในงานวิจัย

วิทยานิพนธ์นี้เลือกใช้อีเธอร์เน็ต ในส่วนการเชื่อมต่อเครือข่ายท้องถิ่น เนื่องจากอีเธอร์เน็ตรองรับความเร็วในการส่งข้อมูลได้สูง และมีอุปกรณ์ราคาถูก ในส่วนโปรโตคอลรับส่งข้อมูลเลือกใช้โปรโตคอล TCP/IP ที่มีความปลอดภัยในการรับส่ง และไม่เสียค่าลิขสิทธิ์ โดยอีเธอร์เน็ตและ โปรโตคอล TCP/IP มีรายละเอียดดังนี้

1. **อีเธอร์เน็ต** (Ethernet)

อีเธอร์เน็ต เป็นการเชื่อมต่อระบบเครือข่ายท้องถิ่นที่ได้รับความนิยมมากที่สุด อีเธอร์เน็ตถูกพัฒนาและปรับปรุงโดย IEEE เดิมอีเธอร์เน็ตเป็นลิขสิทธิ์ของบริษัทซีร็อกซ์ (Xerox) ต่อมา IEEE ได้ประกาศมาตรฐาน IEEE 802.3 ออกมาในปี 1985 ภายหลัง ISO ได้ยอมรับมาตรฐาน IEEE 802.3 เป็นมาตรฐานอีเธอร์เน็ตนานาชาติ ทำให้การผลิตอุปกรณ์อีเธอร์เน็ตไม่ต้องเสียค่าลิขสิทธิ์ หลังจากนั้นการใช้งานอีเธอร์เน็ตก็มีการใช้งานกันอย่างแพร่หลาย

IEEE ได้กำหนดมาตรฐานอีเธอร์เน็ต IEEE 802.3 แยกออกมาอีกหลายประเภท โดยมีที่สำคัญดังนี้

มาตรฐาน IEEE 802.3	รายละเอียด
802.3	มาตรฐานอีเธอร์เน็ตความเร็ว 10Mb/s โดยใช้สายโคแอ็กซ์แบบบาง
802.3a	มาตรฐานอีเธอร์เน็ตความเร็ว 10Mb/s โดยใช้สายโคแอ็กซ์แบบหนา
802.3c	ข้อกำหนดอุปกรณ์ทวนสัญญาณ (repeater) ความเร็ว 10Mb/s
802.3d	Fiber-Optic Inter Repeater Link
802.3i	มาตรฐานอีเธอร์เน็ตความเร็ว 10Mb/s โดยใช้สายคู่บิดเกลียว
802.3j	มาตรฐานอีเธอร์เน็ตความเร็ว 10Mb/s โดยใช้สายใยแก้วนำแสง
802.3u	มาตรฐานอีเธอร์เน็ตความเร็ว 100Mb/s
802.3x	มาตรฐานการควบคุมฟลูดูเพล็กซ์ (Full-Duplex)
802.3z	มาตรฐานอีเธอร์เน็ตความเร็ว 1Gb/s โดยใช้สายใยแก้วนำแสง
802.3ab	มาตรฐานอีเธอร์เน็ตความเร็ว 1Gb/s โดยใช้สายคู่บิดเกลียว
802.3ad	ข้อกำหนดการเพิ่มแบนด์วิธด้วยการเพิ่ม link
802.3ae	มาตรฐานอีเธอร์เน็ตความเร็ว 10Gb/s โดยใช้สายใยแก้วนำแสง

ตารางที่ 3.2 มาตรฐาน IEEE 802.3 ประเภทต่างๆ

ในปัจจุบันความเร็วของอีเธอร์เน็ตที่นิยมใช้งานอยู่ที่ 100 Mbps (Fast Ethernet) แต่ก็มีอีเธอร์เน็ตที่มีความเร็วสูงกว่าออกมาเพื่อรองรับความต้องการของผู้ใช้งาน ได้แก่ Gigabit Ethernet (มาตรฐาน IEEE 802.3z , IEEE 802.3ab) และ 10 Gigabit Ethernet (มาตรฐาน IEEE 802.3ae หรือ มาตรฐาน IEEE 802.3an ที่เป็นมาตรฐานอีเธอร์เน็ตความเร็ว 10Gb/s สำหรับสายคู่บิดเกลียว ที่กำลังพัฒนาอยู่) เนื่องจากความเร็วการรับส่งข้อมูลที่สูง และสามารถเพิ่มความเร็วการรับส่งเพิ่มขึ้นได้ในอนาคต วิทยานิพนธ์นี้จึงได้เลือกใช้งานอีเธอร์เน็ตในส่วนการเชื่อมต่อระบบเครือข่าย

ระดับชั้นการทำงานของอีเธอร์เน็ต หากเปรียบเทียบมาตรฐาน IEEE 802.3 กับ OSI model การทำงานของอีเธอร์เน็ตจะอยู่ในส่วนของชั้น Data Link Layer และ Physical Layer โดยจะแบ่งการทำงานออกเป็นชั้นๆ ซึ่งในแต่ละขั้นของ OSI model จะแบ่งออกเป็นชั้นย่อยดังนี้

Data Link Layer

IEEE ได้แบ่งชั้น Data Link Layer ใน OSI model ออกเป็น 2 ส่วนหลัก คือ Logical Link Control (LLC) และ Media Access Control (MAC) โดยแต่ละชั้นมีหน้าที่ดังต่อไปนี้

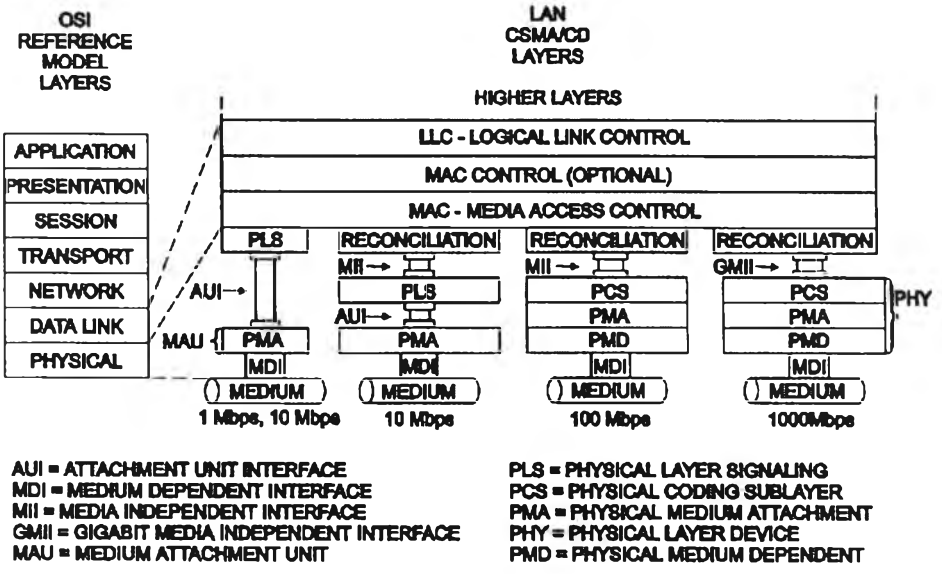
- LLC เป็นชั้นบนของ Data Link Layer เชื่อมต่อกับโปรโตคอลด้านบนให้สามารถใช้อุปกรณ์ได้ LLC จะทำหน้าที่ปรับเฟรมข้อมูลให้ส่งไปในสายสัญญาณได้

- MAC เป็นชั้นล่างใน Data Link Layer เชื่อมต่อกับ Physical Layer รับผิดชอบการรับส่งข้อมูล มีหน้าที่ดังนี้

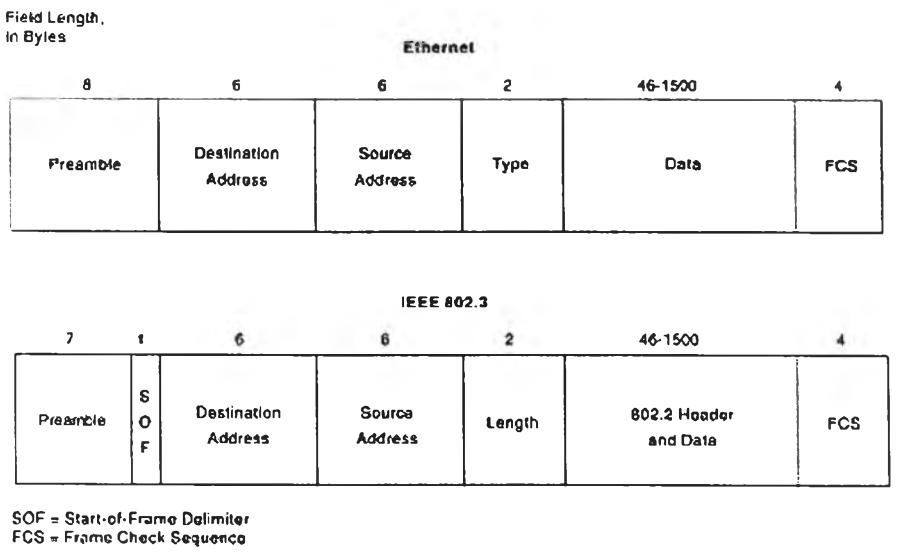
    1. รับข้อมูลจาก LLC มาเพิ่มที่อยู่ (Address) และข้อมูลที่จำเป็นต่อการส่งข้อมูลให้ถึงผู้รับปลายทาง แล้วมาทำให้อยู่ในรูปเฟรมข้อมูล
    2. รับผิดชอบการตรวจสอบข้อผิดพลาดของข้อมูลในเฟรม
    3. ตรวจสอบ Physical Layer ว่าช่องสัญญาณพร้อมสำหรับการส่งข้อมูลหรือไม่
    4. ตรวจสอบว่ามีการชนกันของข้อมูลที่ส่งไปหรือไม่ ถ้าพบการชนกัน ก็จะหยุดการส่งข้อมูล และรอเวลาเพื่อส่งใหม่อีกครั้ง

Physical Layer

ชั้นนี้ถูกแบ่งย่อยออกเป็นชั้นย่อย เพื่อให้สะดวกต่อการปรับเปลี่ยนระบบ เมื่อมีเทคโนโลยีใหม่ ทำให้การปรับเปลี่ยนระบบทำในบางชั้นเท่านั้น ไม่ต้องทำใหม่ทั้งระบบ



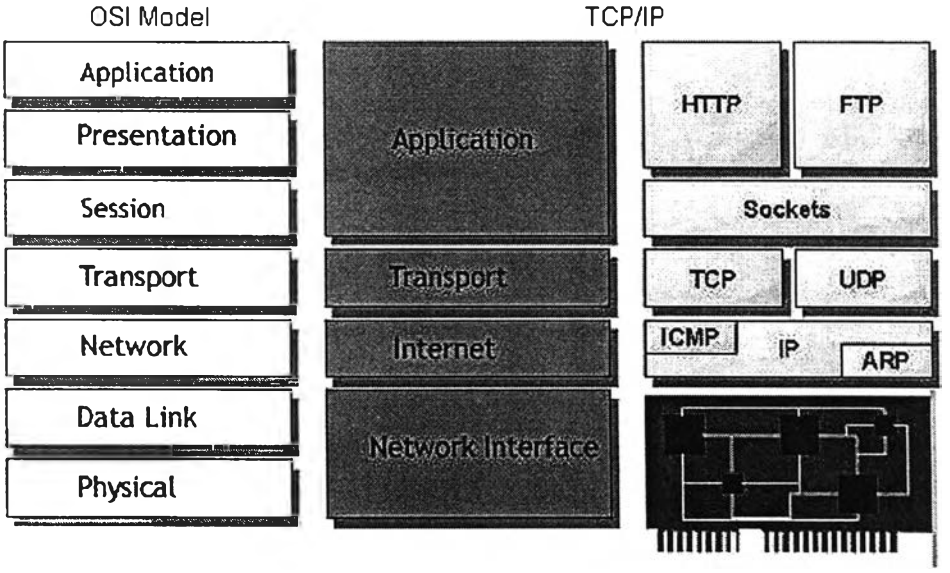
รูปที่ 3.3 มาตรฐาน IEEE 802.3 เปรียบเทียบกับ OSI model



รูปที่ 3.4 รูปแบบเฟรมข้อมูลของอีเธอร์เน็ต

2. **โปรโตคอล** TCP/IP [12]

โปรโตคอล TCP/IP เป็นโปรโตคอลที่ประกอบจากโปรโตคอลย่อยหลายตัว สามารถใช้งานได้โดยไม่ต้องจ่ายค่าลิขสิทธิ์ มีการใช้งานแพร่หลายก่อนมีการกำหนด OSI model ขึ้น มีการแบ่งชั้นการทำงานเป็น 4 ชั้นคล้าย OSI model ซึ่งแต่ละชั้นสามารถเปรียบเทียบกับ OSI model ได้ดังรูปที่ 3.5 ชั้นต่างๆของ TCP/IP มีการทำงานดังนี้

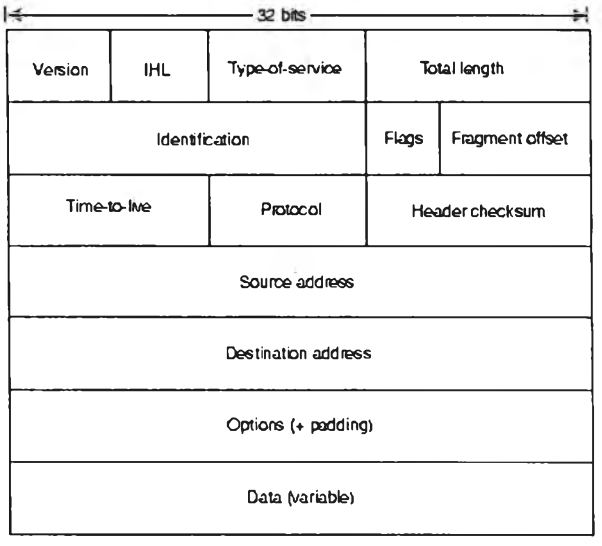


รูปที่ 3.5 การเปรียบเทียบระหว่าง OSI model และ TCP/IP

Internet Layer

เทียบได้กับชั้น 3 ของ OSI model ทำหน้าที่เชื่อมต่อและเลือกเส้นทางการรับส่งข้อมูลจากผู้ส่งไปยังผู้รับ โปรโตคอลในชั้นนี้ได้แก่ IP , ARP , RARP , ICMP , และ IGMP โดยมีโปรโตคอลหลักคือ IP แต่ละโปรโตคอลจะมีหน้าที่ดังนี้

- Internet Protocol (IP) ทำหน้าที่จัดการการรับส่งชุดข้อมูล (Packet) ที่รับมาจากโปรโตคอลด้านบน เช่น TCP , UDP เป็นการให้บริการแบบไม่มีการเชื่อมต่อ (Connectionless) การส่งข้อมูลจะไม่ติดต่อกับปลายทาง แต่จะส่งชุดข้อมูลออกไปทันที ซึ่งอาจทำให้ชุดข้อมูลสูญหาย , ชุดข้อมูลอาจเดินทางถึงปลายทางไม่เรียงลำดับ และอาจมีการส่งข้อมูลซ้ำได้ ซึ่งโปรโตคอลที่อยู่สูงกว่าจะรับผิดชอบแก้ปัญหานี้



รูปที่ 3.6 รูปแบบของชุดข้อมูลจาก IP

- Address Resolution Protocol (ARP) การสื่อสารในระบบเครือข่ายจำเป็นต้องทราบหมายเลขของเน็ตเวิร์คการ์ด (Network Card) หรือแม็กแอดเดรส (MAC Address) โปรโตคอลนี้จะค้นหาหมายเลขแม็กแอดเดรสของเครื่องที่มีหมายเลข IP ที่ต้องการ โดยจะส่งข้อมูลไปยังคอมพิวเตอร์ในระบบเครือข่ายทุกเครื่อง หากมีเครื่องที่มีหมายเลข IP ที่ต้องการส่งข้อมูล เครื่องนั้นจะส่งหมายเลขแม็กแอดเดรสกลับมาให้ เมื่อได้หมายเลขแม็กแอดเดรสแล้วก็สามารถสื่อสารกับปลายทางได้โดยตรงโดยใช้หมายเลขแม็กแอดเดรส

- Reverse Address Resolution Protocol (RARP) ทำงานตรงข้ามกับโปรโตคอล ARP โดยจะหาหมายเลข IP ในกรณีที่รู้หมายเลยแม็กแอดเดรส อยู่แล้ว

- Internet Control Message Protocol (ICMP) ทำหน้าที่รายงานข้อผิดพลาดที่เกิดขึ้นในการส่งข้อมูลในระบบเครือข่าย โดยการส่งของ ICMP จะเป็นแบบไม่มีการเชื่อมต่อ โปรโตคอลนี้จะมีประโยชน์ในการวิเคราะห์และค้นหาจุดเสียของระบบเครือข่าย

- Internet Group Management Protocol (IGMP) ทำหน้าที่แจ้งเราเตอร์ (Router) ให้ทราบกลุ่มหมายเลข IP ที่เป็นมัลติคาสต์ (Multicast) เพื่อให้เครือข่ายสามารถรองรับการส่งข้อมูลกลุ่มนี้ได้ โดยการส่งข้อมูลของ IGMP เป็นแบบไม่มีการเชื่อมต่อ

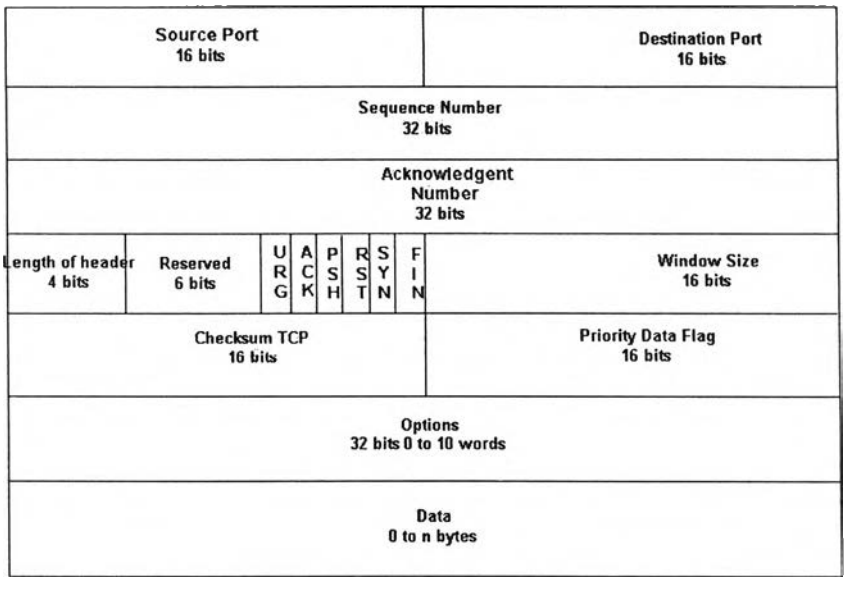
Transport Layer

เทียบได้กับชั้น 4 ของ OSI model ทำหน้าที่ควบคุมการรับส่งข้อมูลและแบ่งข้อมูลให้เหมาะกับระบบเครือข่าย โปรโตคอลในชั้นนี้จะประกอบด้วย TCP และ UDP ซึ่งสามารถเลือกใช้งานได้ ขึ้นกับรูปแบบในการส่งข้อมูล แต่ละโปรโตคอล มีรายละเอียดดังนี้

TCP (Transmission Control Protocol) เป็นโปรโตคอลแบบมีการเชื่อมต่อ (Connection Oriented) จะส่งข้อมูลทั้งหมดจนสำเร็จ ถ้าข้อมูลที่ส่งมีขนาดใหญ่จะมีการแบ่งย่อยออกเป็นชุดเล็กๆ และใส่ลำดับหมายเลขไว้ (Sequence Number) เพื่อใช้รวมข้อมูลให้กลับเป็นเหมือนเดิมได้ที่เครื่องรับ การส่งข้อมูลจะมีขั้นตอนดังนี้

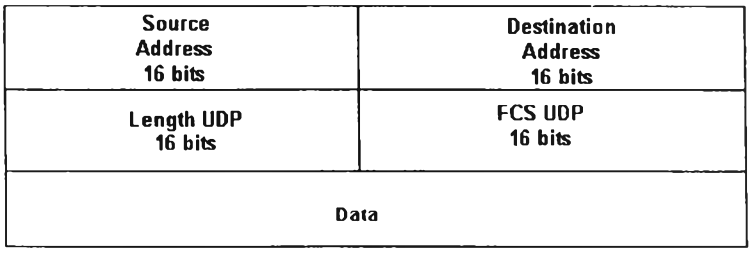
- เครื่องส่งจะส่งข้อมูลไปยังเครื่องรับเพื่อแจ้งให้ทราบว่าต้องการส่งข้อมูล
- เครื่องรับจะตอบตกลงกลับมาพร้อมรหัสในการส่งข้อมูล
- เครื่องส่งจะส่งข้อมูลพร้อมรหัสที่รับมาเพื่อเป็นการยืนยันการเชื่อมต่อ

หลังจากรับข้อมูลแล้ว เครื่องรับจะตรวจสอบความถูกต้องของข้อมูลที่รับมา และแจ้งผลการตรวจสอบนั้นกลับไปยังเครื่องส่งในเวลาที่กำหนด หากเครื่องส่งไม่ได้รับการตอบรับ เครื่องส่งจะส่งข้อมูลชุดนั้นไปใหม่อีกครั้ง วิธีนี้จะสิ้นเปลืองแบนด์วิธในการส่ง เนื่องจากมีการส่งข้อมูลไปกลับเพื่อตรวจสอบหลายครั้ง แต่จะมีความน่าเชื่อถือสูง



รูปที่ 3.7 รูปแบบของชุดข้อมูลจาก TCP

UDP (User Datagram Protocol) เป็นโปรโตคอลที่ส่งข้อมูลแบบไม่มีการเชื่อมต่อ หรือดาต้าแกรม (Datagram) การส่งข้อมูลไม่มีการแจ้งเครื่องฝ่ายรับก่อน และไม่มีการตอบกลับว่าได้รับข้อมูลถูกต้องหรือไม่ มีความน่าเชื่อถือต่ำ แต่มีข้อดีคือไม่ต้องรอการตอบกลับจากฝ่ายรับ ทำให้ใช้แบนด์วิธน้อย เหมาะสำหรับการส่งข้อมูลแบบบรอดคาสต์ (Broadcast) และมัลติคาสต์ (Multicast)



รูปที่ 3.8 รูปแบบของชุดข้อมูลจาก UDP

Application Layer

ทำหน้าที่เชื่อมต่อกับผู้ใช้งานและการให้บริการต่างๆ เช่น HTTP ,FTP เป็นต้น โปรโตคอลในชั้นนี้ได้แก่ Hyper Text Transfer Protocol (HTTP) , Simple Mail Transfer Protocol (SMTP) , Post Office Protocol (POP) , InternetMessage Access Protocol (IMAP) , File Transfer Protocol (FTP) และ Telnet โดยแต่ละโปรโตคอลจะมีลักษณะการใช้งานแตกต่างกันออกไป

Network Interface

ในชั้นล่างสุดนี้ ทำหน้าที่ควบคุมฮาร์ดแวร์เพื่อรับส่งข้อมูล ในชั้นนี้เทียบได้กับชั้น 1 และ 2 ของ OSI model โปรโตคอล TCP/IP ไม่ได้จำกัดรูปแบบไว้ ทำให้สามารถใช้งานกับเทคโนโลยีระบบเครือข่ายต่างๆได้ ไม่ว่าจะเป็น อีเธอร์เน็ต , โทเคนริง ,อินทราเน็ต ฯลฯ

3.4 การเขียนโปรแกรมเชื่อมต่อในระบบเครือข่ายผ่านโปรโตคอล TCP/IP

ในการรับส่งข้อมูลในระบบเครือข่ายท้องถิ่น ระหว่างไคลเอนท์ กับเซิร์ฟเวอร์ สามารถเลือกรูปแบบการส่งได้ 2 ลักษณะ คือ ใช้โปรโตคอล TCP ส่งข้อมูล หรือ ใช้โปรโตคอล UDP โดยการส่งข้อมูลทั้ง 2 แบบจำเป็นต้องระบุหมายเลย IP ของเครื่องปลายทาง และหมายเลขพอร์ตที่ใช้ส่งข้อมูล

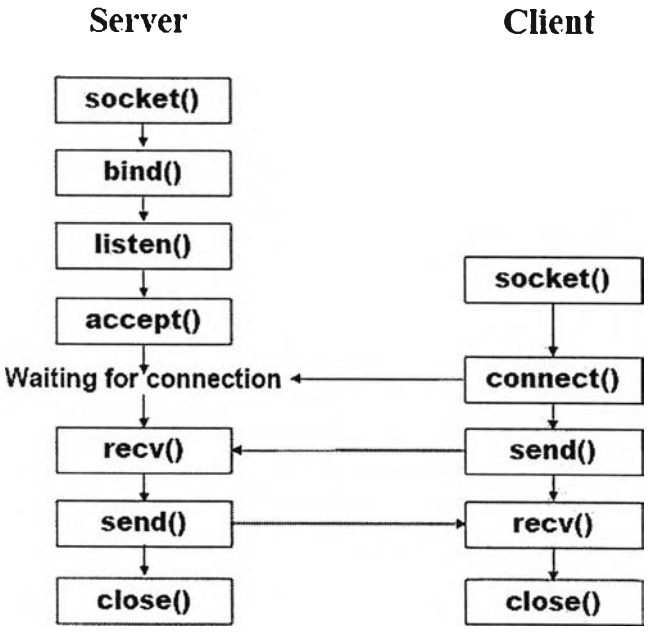
- <u>หมายเลข IP</u> เป็นหมายเลขอ้างอิง ใช้บอกตำแหน่งของอุปกรณ์ในระบบเครือข่าย เพื่อความสะดวกในการสื่อสาร หมายเลข IP ของแต่ละอุปกรณ์จะไม่ซ้ำกัน อุปกรณ์หนึ่งสามารถมี หมายเลข IP มากกว่า 1 ได้ หมายเลข IP สามารถเปลี่ยนแปลงได้ ต่างจาก MAC Address (Media Access Control Address) ที่ถูกกำหนดจากผู้ผลิตอุปกรณ์ หมายเลข IP ที่นิยมใช้งานในปัจจุบันจะมีขนาด 32 บิต แบ่งออกเป็น 4 กลุ่ม กลุ่มละ 8 บิต นิยมเขียนในรูปแบบด็อตเดซิมอล ทำให้มี หมายเลข IP อยู่ในช่วง 0.0.0.0 ถึง 255.255.255.255 เรียกหมายเลข IP แบบนี้ว่า IPV4 ซึ่งมีแนวโน้มว่าไม่เพียงพอต่อการใช้งานในอนาคต จึงมีการกำหนด IPV6 ที่มีขนาด 40 บิต ซึ่งมีหมายเลข IP อยู่ในช่วง 0.0.0.0.0.0 จนถึง 255.255.255.255.255.255
- <u>หมายเลขพอร์ต</u> บนคอมพิวเตอร์อาจมีการเรียกใช้งานโปรโตคอล TCP/IP พร้อมกันครั้งละหลายตัว เพื่อให้สามารถรองรับการใช้งานได้ จึงมีการใช้งานพอร์ตและซ็อกเก็ต (Port and Socket) โดยพอร์ตจะมีตั้งแต่หมายเลข 0 จนถึง 65,536

การส่งข้อมูลโดยใช้โปรโตคอล TCP และ UDP จะมีรูปแบบการเชื่อมต่อที่ต่างจากกัน โดยมีรายละเอียดต่างๆดังนี้

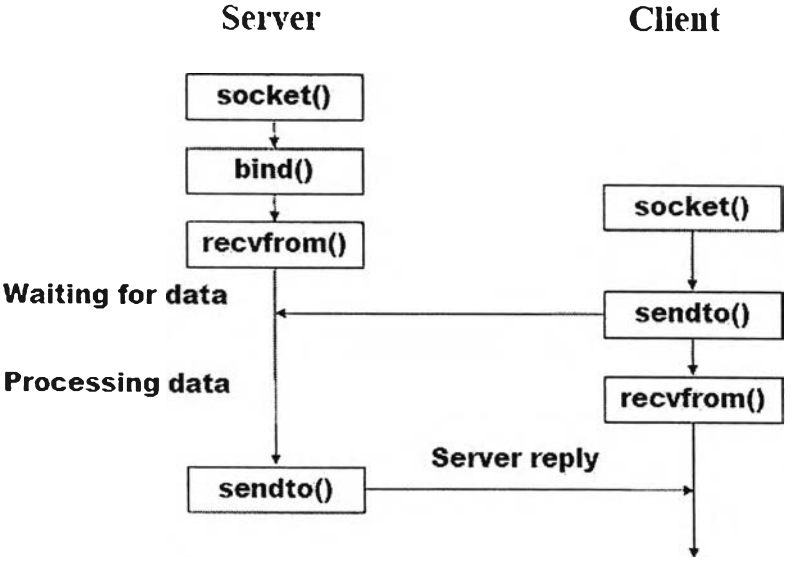
การส่งข้อมูลโดยใช้โปรโตคอล TCP

โปรโตคอล TCP มีการตรวจสอบข้อมูลในการส่งว่าถึงปลายทางถูกต้องหรือไม่ ทำให้มีความน่าเชื่อถือสูง ลำดับการเชื่อมต่อจะแสดงในรูปที่ 3.9 โดยในแต่ละคำสั่งมีหน้าที่ดังนี้

Socket	เป็นคำสั่งสร้างช่องทางการเชื่อมต่อของโปรแกรม ใช้งานทั้งในฝั่งไคลเอนท์และเซิร์ฟเวอร์
Bind	ใช้กำหนด IP address ของเครื่อง และพอร์ต (port) ที่จะใช้ในการติดต่อให้กับช่องทางที่สร้างขึ้น
Listen	เป็นคำสั่งในฝั่งเซิร์ฟเวอร์ เพื่อรอการเชื่อมต่อจากทางฝั่งไคลเอนท์
Connect	เป็นคำสั่งในฝั่งไคลเอนท์ ใช้เชื่อมต่อไปยังเซิร์ฟเวอร์ ในคำสั่งนี้มีส่วนกำหนด IP address ของเซิร์ฟเวอร์และพอร์ตที่ต้องการเชื่อมต่อ
Accept	เป็นคำสั่งในฝั่งเซิร์ฟเวอร์ ใช้รับการเชื่อมต่อจากฝั่งไคลเอนท์ การรับข้อมูลนี้ทำให้เซิร์ฟเวอร์รู้ที่อยู่ของไคลเอนท์ที่เชื่อมต่อเข้ามา เมื่อรับการเชื่อมต่อแล้วจะส่งข้อมูลกลับไปไคลเอนท์เพื่อแจ้งให้ทราบว่าการเชื่อมต่อสำเร็จ
Send	ใช้ส่งข้อมูลไปยังฝ่ายรับ ในช่องทางที่เชื่อมต่อแล้ว การส่งข้อมูลไม่จำเป็นต้องกำหนด IP address และหมายเลขพอร์ตอีก
Recv	ใช้รับข้อมูลจากฝ่ายส่ง ในช่องทางที่เชื่อมต่อแล้ว
Close	เป็นคำสั่งปิดช่องทางการเชื่อมต่อที่สร้างขึ้นจากคำสั่ง socket()



รูปที่ 3.9 ลำดับการเชื่อมต่อของโปรโตคอล TCP



รูปที่ 3.10 ลำดับการเชื่อมต่อของโปรโตคอล UDP

การส่งข้อมูลโดยใช้โปรโตคอล UDP

โปรโตคอล UDP มีความเร็วในการส่งข้อมูลสูงกว่าโปรโตคอล TCP เพราะไม่มีการตรวจสอบข้อมูลว่าถึงปลายทางหรือไม่ การเชื่อมต่อของโปรโตคอล UDP จะแสดงในรูปที่ 3.10 แต่ละคำสั่งมีหน้าที่ดังนี้

Socket	เป็นคำสั่งสร้างช่องทางการเชื่อมต่อของโปรแกรม ใช้งานทั้งในฝั่งไคลเอนท์และเซิร์ฟเวอร์
Bind	ใช้กำหนด IP address ของเครื่อง และพอร์ต (port) ที่จะใช้ในการติดต่อให้กับช่องทางที่สร้างขึ้น
Sendto	เป็นคำสั่งส่งข้อมูลในโปรโตคอล UDP ต่างจากคำสั่ง send ของโปรโตคอล TCP เนื่องจากไม่มีการเชื่อมต่อระหว่างไคลเอนท์กับเซิร์ฟเวอร์ ทำให้ต้องกำหนด IP address ปลายทาง และพอร์ตที่ใช้งานในคำสั่งทุกครั้งที่ส่งข้อมูล
Recvfrom	เป็นคำสั่งรับข้อมูลในโปรโตคอล UDP จากคำสั่ง recv ของโปรโตคอล TCP โดยภายในคำสั่งมีส่วนเก็บค่า IP address ของต้นทางที่ส่งมา และพอร์ตที่ใช้งาน
Close	เป็นคำสั่งปิดช่องทางการเชื่อมต่อที่สร้างขึ้นจากคำสั่ง socket()

ในวิทยานิพนธ์นี้ได้เลือกใชการเชื่อมต่ออีเธอร์เน็ต เพราะมีอุปกรณ์ราคาถูก หาง่าย มีความเร็วในการส่งข้อมูลสูง สามารถปรับเปลี่ยนเพิ่มความเร็วในการรับส่งเพิ่มขึ้นได้ในอนาคต และเลือกใช้โปรโตคอล TCP/IP โดยใช้งานโปรโตคอล TCP เนื่องจากข้อมูลจะถึงที่หมายได้ถูกต้อง และแม่นยำ แม้ว่าจะมีความเร็วการส่งข้อมูลต่ำกว่า UDP