บทที่ 2
ทฤษฎีและหลักการ

โครงงานวิจัยนี้ออกแบบระบบบันทึกข้อมูลชนิดไร้สาย กล่าวคือ การสร้างโปรแกรมคอมพิวเตอร์ซึ่งทำหน้าที่บันทึกและแสดงผลข้อมูลในรูปแบบของกราฟ โปรแกรมที่สร้างขึ้นนี้สามารถที่จะบันทึกข้อมูลแบบไร้สายผ่านการสื่อสารชนิด Wi-Fi และติดตั้งบนระบบปฏิบัติการ Windows เพื่อให้โครงงานวิจัยชิ้นนี้มีประโยชน์สูงสุด คณะผู้วิจัยได้ทำการสร้างระบบควบคุมอุณหภูมิขึ้นมาทดสอบ โปรแกรมบันทึกข้อมูลแบบไร้สาย ซึ่งเป็นระบบง่ายๆ ไม่มีความสลับซับซ้อน สร้างขึ้นด้วยอุปกรณ์ที่หาได้โดยง่ายและราคาถูก ตัวระบบทำมาจากท่อ PVC ด้านหนึ่งมีอุปกรณ์กำเนิดแหล่งความร้อนซึ่งใช้ไดเป่าผมทั่วๆไป อีกด่านหนึ่งนั้นมีอุปกรณ์เซนเซอร์วัดค่าอุณหภูมิ สำหรับไดเป่าผมทั่ว ๆ ไปค่าอุณหภูมินี้จะเปลี่ยนไปถ้าปริมาณความเร็วลมของตัวไดเป่าผมเปลี่ยนไป นอกจากนี้ การเปลี่ยนอุณหภูมิภายในท่อสามารถทำได้อีกวิธีหนึ่ง คือการควบคุมความมากน้อยของแรงดันไฟฟ้าที่ป้อนให้กับขดลวดกำเนิดความร้อนของไดเป่าผม ดังนั้นโครงงานชิ้นนี้ได้ทำการสร้างวงจรไฟฟ้าเพื่อควบคุมแรงดันไฟฟ้าดังกล่าวด้วย ในส่วนของข้อมูลอุณหภูมิที่ได้จากอุปกรณ์เซนเซอร์จะถูกส่งไปบันทึกใช้บนคอมพิวเตอร์ ด้วยโปรแกรมบันทึกข้อมูลไร้สายที่สร้างขึ้น ผ่านการสื่อสารแบบ Wi-Fi

2.1 TCP/IP Protocol

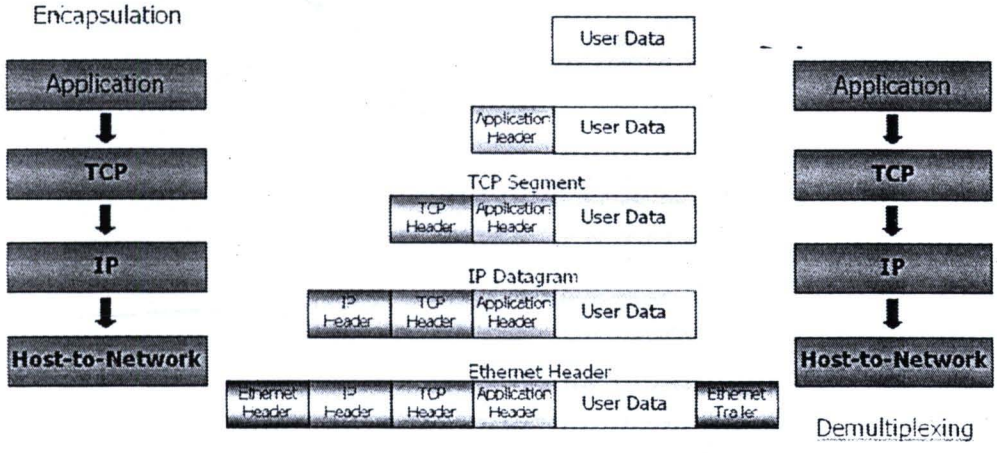
TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ตโดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหาโปรโตคอล(Protocol)ก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้ชุดโปรโตคอลนี้ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่ายARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ตทำให้TCP/IPเป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

ชุดของโปรโตคอลTCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐานสามประการสำหรับการติดต่อสื่อสารคือ

1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่ายเช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่แต่โนด(node)กลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาดกฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ

3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วนเช่นการจัดส่งแฟ้มข้อมูลและแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูลเช่นการสื่อสารแบบเรียลไทม์(Real-time)และทั้งการสื่อสารแบบเสียง(Voice) และข้อมูล(data)

การส่งข้อมูลผ่านในแต่ละระดับชั้น(Layer)แต่ละระดับชั้น จะทำการประกอบข้อมูลที่ได้รับมากับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่าเฮดเดอร์(Header)ภายในเฮดเดอร์จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการเอนแคปซูเลท(Encapsulate)เมื่อผู้รับได้รับข้อมูลก็จะเกิดกระบวนการทำงานย้อนกลับคือโปรโตคอล(Protocol) เดียวกันทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็นเฮดเดอร์ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไรซึ่งกระบวนการย้อนกลับนี้เรียกว่าดีมัลติเพลกซิ่ง(Demultiplexing)



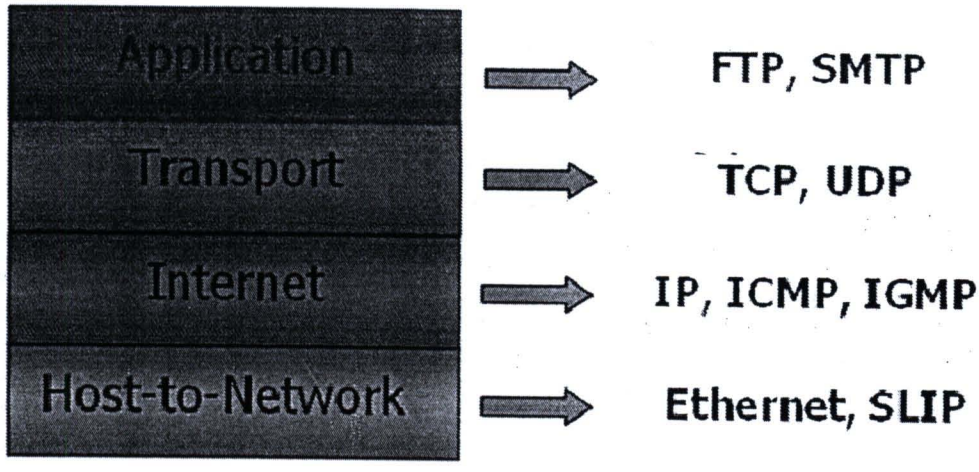
รูปที่ 2.1 ขั้นตอนการ Encapsulation และDemultiplexing

ข้อมูลที่ผ่านการเอนแคปซูเลท(Encapsulate)ในแต่ละระดับชัน (Layer) มีชื่อเรียกแตกต่างกันดังนี้

- ข้อมูลที่มาจากผู้ใช้(User)หรือก็คือข้อมูลที่ผู้ใช้(User)เป็นผู้ป้อนให้กับแอปพลิเคชั่น(Application)เรียกว่ายูสเซอร์ดาต้า(User Data)
- เมื่อแอพพลิเคชั่น(Applicaion)ได้รับข้อมูลจากผู้ใช้(User)ก็จะนำมาประกอบกับส่วนหัวของแอพพลิเคชั่นเรียกว่าแอปพลิเคชั่นดาต้า(Application Data)และส่งต่อไปยังโปรโตคอลทีซีพี(Protocol TCP)
- เมื่อโปรโตคอลทีซีพีได้รับแอปพลิเคชั่นดาต้า(Application Data)ก็จะนำมารวมกับเฮดเดอร์ของโปรโตคอลทีซีพีเรียกว่าทีซีพีเซกเมนต์(TCP Segment)และส่งต่อไปยังโปรโตคอลไอพี(Protocol IP)

- เมื่อโปรโตคอล(Protocol) IP ได้รับทีซีพีเซกเมนต์ก็จะนำมารวมกับเฮดเดอร์ของโปรโตคอลไอพีเรียกว่า ไอพีดาต้าแกรม(IPDatagram)และส่งจากโฮสต์ไปยังเน็ทเวิร์คเลเยอร์(Host-to-Network Layer)
- ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วนError Correction และ flag เรียกว่าEthernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้าส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป

ในแต่ละเลเยอร์ของโครงสร้าง TCP/IP สามารถอธิบายได้ดังนี้



รูปที่2.2โครงสร้าง TCP/IP

2.1.1 ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)

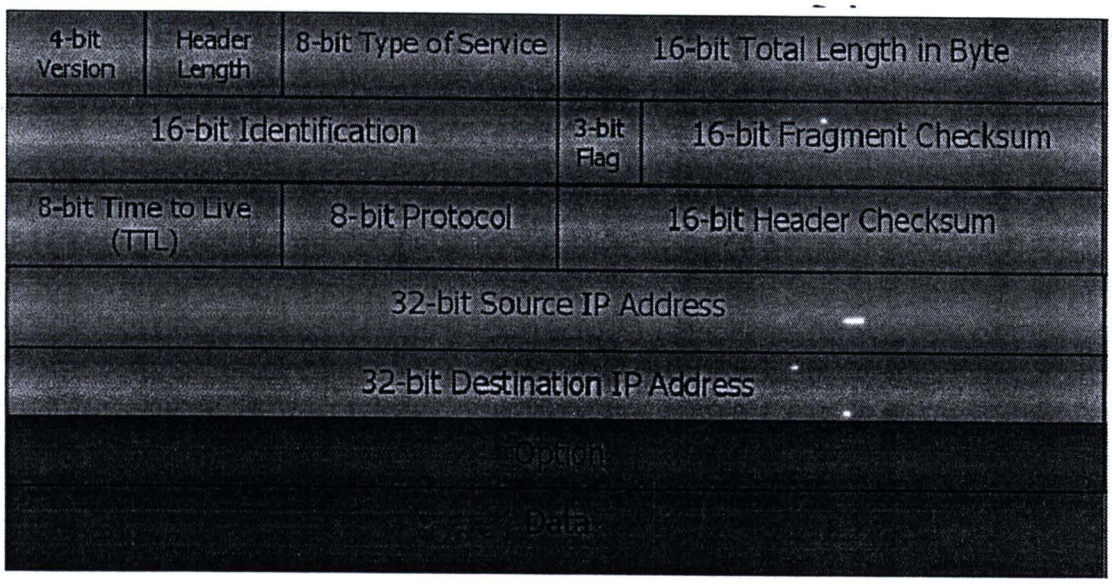
โพรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการหน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสาร IP มาแล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูลทางด้านผู้รับก็จะทำงานในทางกลับกันคือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสาร

2.1.2 ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer)

ใช้ประเภทของระบบการสื่อสารที่เรียกว่าระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็กเก็ต (PacketSwitchingNetwork) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่าแพ็กเก็ต (Packet) สามารถไหลจากโหนดผู้ส่งไปตามโหนดต่างๆในระบบจนถึงจุดหมายปลายทางได้โดยอิสระหากว่ามีการส่งแพ็กเก็ตออกมาเป็นชุดโดยมีจุดหมายปลายทางเดียวกันในระหว่างการเดินทางในเครือข่ายแพ็กเก็ตแต่ละตัวในชุดนี้ก็จะเป็นอิสระแก่กันและกันดังนั้นแพ็กเก็ตที่ส่งไปถึงปลายทางอาจจะไม่เป็นไปตามลำดับก็ได้

2.1.3 IP (Internet Protocol)

IP เป็นโปรโตคอลในระดับเน็ตเวิร์คเลเยอร์ทำหน้าที่จัดการเกี่ยวกับแอดเดรสและข้อมูลและควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของแพ็กเก็ตซึ่งกลไกในการหาเส้นทางของ IP จะมีความสามารถในการหาเส้นทางที่ดีที่สุดและสามารถเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูลและมีระบบการแยกและประกอบดาต้าแกรม (datagram) เพื่อรองรับการส่งข้อมูลระดับ data link ที่มีขนาด MTU(MaximumTransmission Unit) ที่แตกต่างกันทำให้สามารถนำ IP ไปใช้บนโปรโตคอลอื่นได้หลากหลายเช่น Ethernet ,Token Ring หรือApple Talkการเชื่อมต่อของ IP เพื่อทำการส่งข้อมูลจะเป็นแบบ Connectionless หรือเกิดเส้นทางการเชื่อมต่อในทุกๆครั้งของการส่งข้อมูล 1 ดาต้าแกรมโดยจะไม่ทราบถึงข้อมูลดาต้าแกรมที่ส่งก่อนหน้าหรือส่งตามมาแต่การส่งข้อมูลใน 1 ดาต้าแกรมอาจจะเกิดการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ (Fragmentation) และถูกนำไปรวมเป็นดาต้าแกรมเดิมเมื่อถึงปลายทาง



รูปที่2.3 IP Header

เฮดเดอร์ของ IP โดยปกติจะมีขนาด 20 bytes ยกเว้นในกรณีที่มีการเพิ่ม Option บางอย่าง ฟิลด์ของเฮดเดอร์ IP จะมีความหมายดังนี้

Version:หมายเลขเวอร์ชันของโปรโตคอลที่ใช้งานในปัจจุบันคือเวอร์ชัน 4 (IPv4) และเวอร์ชัน 6 (IPv6)

Header Length:ความยาวของข้อมูล เฮดเดอร์โดยทั่วไปถ้าไม่มีส่วนข้อมูล optionจะมีค่าเป็น 5 (5*32 bit)

Type of Service (TOS):ใช้เป็นข้อมูลสำหรับเราเตอร์(Router)ในการตัดสินใจเลือกการเราต์ข้อมูลในแต่ละดาต้าแกรมแต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว

Length:ความยาวทั้งหมดเป็นจำนวนไบต์ของดาต้าแกรมซึ่งด้วยขนาด16 บิตของฟิลด์จะหมายถึงความยาวสูงสุดของดาต้าแกรมคือ 65535byte (64k) แต่ในการส่งข้อมูลจริงข้อมูลจะถูกแยกเป็นส่วนๆตามขนาดของ MTU ที่กำหนดในลิงค์เลเยอร์และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทางแอพพลิเคชั่นส่วนใหญ่จะมีขนาดของดาต้าแกรมไม่เกิน 512 byte

Identification:เป็นหมายเลขของดาต้าแกรมในกรณีที่มีการแยกดาต้าแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี Identification เดียวกันมารวมกัน

Flag:ใช้ในกรณีที่มีการแยกดาต้าแกรม

Fragment offset:ใช้ในการกำหนดตำแหน่งข้อมูลในดาต้าแกรมที่มีการแยกส่วนเพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง

Time to live (TTL):กำหนดจำนวนครั้งที่มากที่สุดที่ดาต้าแกรมจะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุดโดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่าTTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทางข้อมูลนั้นจะถูกยกเลิกและเราเตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิด time out ในระหว่างการส่งข้อมูล

Protocol:ระบุโปรโตคอลที่ส่งในดาต้าแกรมเช่นTCP ,UDPหรือICMP

Header checksum:ใช้ในการตรวจสอบความถูกต้องของข้อมูลในเฮดเดอร์

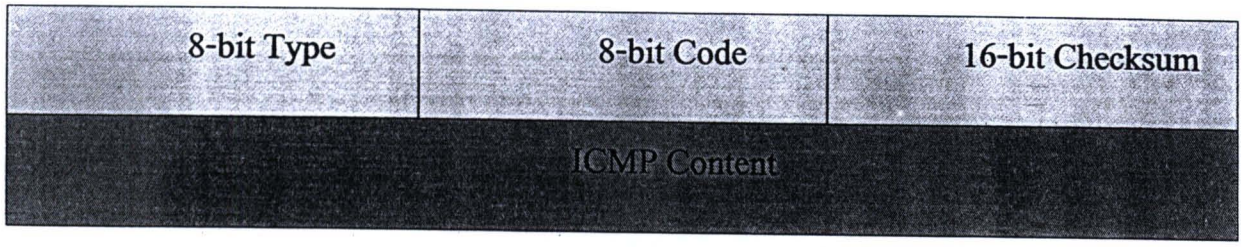
Source IP address:หมายเลข IP ของผู้ส่งข้อมูล

Destination IP address:หมายเลข IP ของผู้รับข้อมูล

Data:ข้อมูลจากโปรโตคอลระดับบน

2.1.4 ICMP โปรโตคอล

ICMP (Internet Control Message Protocol)เป็นโปรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานภาพของดาต้าแกรม(Datagram) ในกรณีที่เกิดปัญหากับดาต้าแกรมเช่นเราเตอร์ไม่สามารถส่งดาต้าแกรมไปถึงปลายทางได้ ICMP จะถูกส่งออกไปยังโฮสต้นทางเพื่อรายงานข้อผิดพลาดที่เกิดขึ้นอย่างไรก็ดีไม่มีอะไรรับประกันได้ว่า ICMP Message ที่ส่งไปจะถึงผู้รับจริงหรือไม่หากมีการส่งดาต้าแกรมออกไปแล้วไม่มี ICMP Message ฟ้อง Error กลับมาก็แปลความหมายได้สองกรณีคือข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อยหรืออาจจะมีปัญหาในการสื่อสารทั้งการส่งดาต้าแกรมและ ICMP Message ที่ส่งกลับมาก็มีปัญหาระว่างทางก็ได้ ICMP จึงเป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (Unreliable) ซึ่งจะเป็นหน้าที่ของโปรโตคอลในระดับสูงกว่า NetworkLayerในการจัดการให้การสื่อสารนั้นๆมีความน่าเชื่อถือในส่วนของ ICMP Message จะประกอบด้วย Type ขนาด 8 บิต Checksum ขนาด 16 บิตและส่วนของ Content ซึ่งจะมีขนาดแตกต่างกันไปตาม Type และ Code ดังรูป



รูปที่2.4 ICMP Header

2.1.5 ชั้นสื่อสารนำส่งข้อมูล

แบ่งเป็นโพรโตคอล 2 ชนิดตามลักษณะลักษณะแรกเรียกว่า Transmission Control Protocol (TCP)เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (Connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นแบบ Byte stream ที่ไว้ใจได้โดยไม่มีข้อผิดพลาดข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆเรียกว่า Message ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ตทางฝ่ายผู้รับจะนำ Message มาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่งส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วยโปรโตคอลการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) มีการตรวจสอบความถูกต้องของข้อมูลแต่จะไม่มีการแจ้งกลับไปยังผู้ส่งจึงถือได้ว่าไม่มีการตรวจสอบความถูกต้องของข้อมูลอย่างไรก็ตามวิธีการนี้มีข้อดีในด้านความรวดเร็วในการส่งข้อมูลจึงนิยมใช้ในระบบผู้ให้และผู้ใช้บริการ (client/server system) ซึ่งมีการสื่อสารแบบถาม/ตอบ(request/reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการส่งเสียง (voice) ทางอินเทอร์เน็ต

2.1.6 UDP โปรโตคอล

เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของUDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูลเรียกว่า UDP datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูลกลไกการตรวจสอบโดย checksum ของ UDP นั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไขหรือมีความผิดพลาดระหว่างการส่งและหากเกิดเหตุการณ์ดังกล่าวปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้นแต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้นโดยในข้อกำหนดของ UDPหากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้นแต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใดการรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่นส่งไม่ถึง,หมดเวลาผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้องแต่เกิดข้อผิดพลาดในส่วนของ UDP เองจะไม่มีการยืนยันหรือแจ้งให้ผู้ส่งทราบแต่อย่างใด

16-bit Source Port	16-bit Destination Port
Lenght	Checksum
Data	

รูปที่2.5 UDP Header

Source Port Number:หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้

Destination Port Number:หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม

UDP Length:ความยาวของดาต้าแกรมทั้งส่วน Header และ data นั่นหมายความว่าค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header

Checksum:เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

2.1.7 TCP โปรโตคอล

อยู่ใน Transport Layer เช่นเดียวกับ UDP ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูลซึ่งมีความสามารถและรายละเอียดมากกว่า UDP โดยดาต้าแกรมของ TCP จะมีความสัมพันธ์ต่อเนื่องกันและมีกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง (Reliable) และมีการสื่อสารอย่างเป็นกระบวนการ (Connection - oriented)

16-bit Source Port Number								16-bit Source Destination Port
32-bit Sequence Number								
32-bit Acknowledge Number								
Header Length	6-Bit Reserved	URG	ACK	PUSH	RESET	SYN	FIN	16-bit Windows Size
16-bit TCP Checksum								16-bit Urgent Pointer
TCP Option								
Data								

รูปที่2.6 TCP Header

Source Port Number:หมายเลขพอร์ต(Port) ต้นทางที่ส่งดาต้าแกรมนี้

Destination Port Number:หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม

Sequence Number:ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้งเพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใดและนำมาจัดลำดับได้ถูกต้อง

Acknowledgment Number:ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ

Header Length:โดยปกติความยาวของเฮดเดอร์ TCP จะมีความยาว 20 ไบต์แต่อาจจะมากกว่านั้นถ้ามีข้อมูลในฟิลด์ Option แต่ต้องไม่เกิน 60 ไบต์

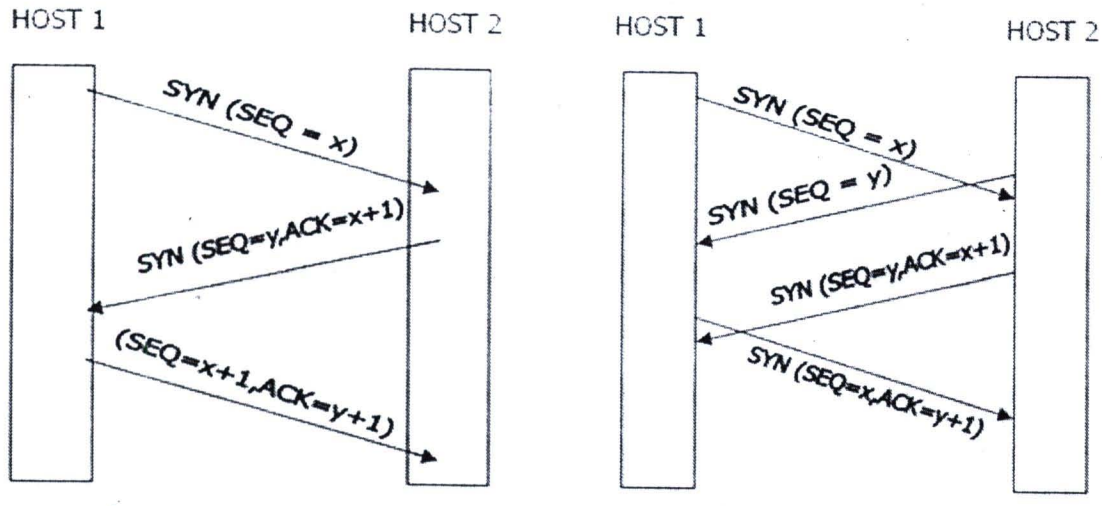
Flag:เป็นข้อมูลระดับบิตที่อยู่ในเฮดเดอร์ TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้นๆและใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วยซึ่ง Flag มีอยู่ทั้งหมด 6 บิตแบ่งได้ดังนี้

Type	Description
URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent Pointer)
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรจะส่งข้อมูล Segment นี้ไปยัง Application ที่กำลังรออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ(reset)เนื่องจากในกรณีที่เกิดการสับสนขึ้นด้วยเหตุผลต่างๆ เช่น โฮสต์มีปัญหา ให้เริ่มต้นสื่อสารกันใหม่
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

ตารางที่ 2.1 รายละเอียดของ Flag

Flag ในเฮดเดอร์ของ TCP มีความสำคัญในการกำหนดการทำงานของ TCP segment เนื่องจากข้อมูลในเฮดเดอร์ของ TCP จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูลซึ่งในการทำงานแต่ละอย่างจะมีการใช้งานฟิลด์ไม่เหมือนกัน Flag จะเป็นตัวกำหนดว่าให้ใช้งานฟิลด์ไหน เช่นฟิลด์ Acknowledgment numberจะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อแต่จะมีข้อมูลในฟิลด์ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆซึ่งถ้าไม่มี flag เป็นตัวกำหนดก็อาจจะมีการนำข้อมูลมาใช้และก่อให้เกิดความผิดพลาดได้

2.1.7.1 การสื่อสารของ TCP

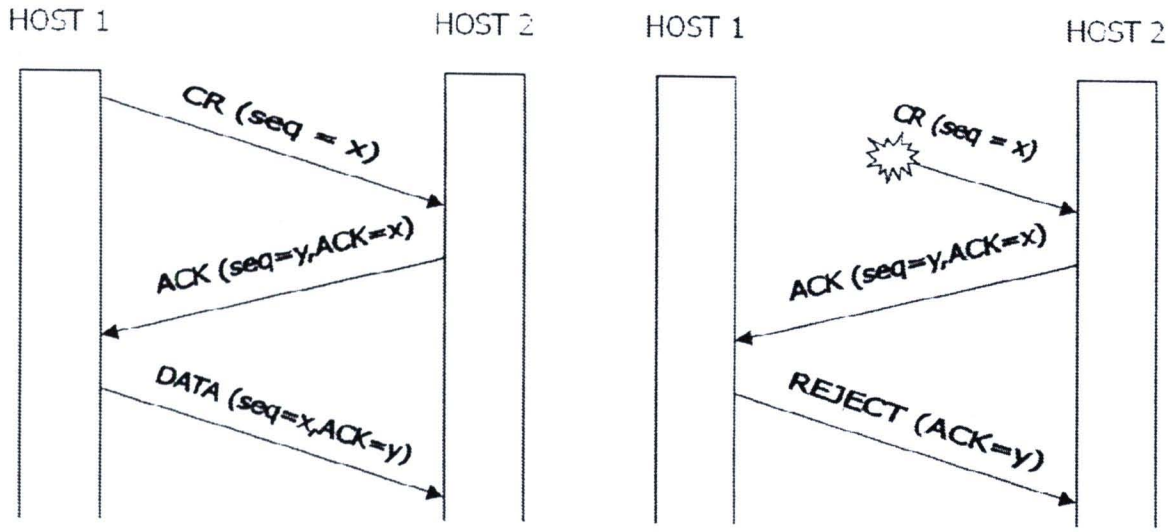


รูปที่ 2.7การสื่อสารของ TCP

เมื่อเซกเมนต์ CONNECT (SYN = "1" และ ACK = "0") เดินทางมาถึง Entity TCP ที่โฮสต์ปลายทางจะค้นหาโพรเซส(Process) ตามหมายเลขพอร์ตที่กำหนดในเขตข้อมูล Destination port ซึ่งถ้าหากไม่พบก็จะตอบปฏิเสธด้วยเซกเมนต์ที่มี RST = "1" กลับไปยังผู้ส่งเซกเมนต์ CONNECT ของผู้ส่งจะถูกส่งต่อไปยังโพรเซสตามพอร์ตที่ระบุซึ่งอาจจะตอบรับหรือตอบปฏิเสธก็ได้ถ้าโพรเซสนั้นต้องการสื่อสารด้วยก็จะส่งเซกเมนต์ตอบรับกลับไปรูปที่ 2.7แสดงลำดับขั้นตอนการส่ง TCP เซกเมนต์ในการสร้างการเชื่อมต่อในสภาวะปกติระหว่างผู้ส่งและผู้รับในกรณีที่โฮสต์สองแห่งพยายามสร้างการเชื่อมต่อระหว่างซ็อคเก็ต(Socket) คู่เดียวกันผลสุดท้ายจะมีการเชื่อมต่อเกิดขึ้นเพียงหนึ่งช่องทางเท่านั้นเนื่องจากการเชื่อมต่อในแต่ละช่องทางจะถูกกำหนดขึ้นโดยใช้หมายเลขซ็อคเก็ตผู้ส่งและผู้รับถ้าการเชื่อมต่อลำดับแรกสำเร็จก็จะถูกบันทึกไว้ในตารางการสื่อสาร เช่น (x, y) ถ้าการเชื่อมต่อลำดับที่สองสำเร็จในเวลาต่อมาข้อมูลนี้ก็จะถูกบันทึกไว้ที่เดียวกันคือ (x, y)ขั้นตอนในการสร้างการเชื่อมต่อและการยกเลิกสามารถเขียนอธิบายด้วยไฟไนท์สเตทแมชชีน (Finite State Machine)ที่มีการทำงาน 11 สถานะในแต่ละสถานะจะมีเหตุการณ์บางอย่างที่เป็นไปได้ซึ่งจะได้รับการตอบสนองด้วยการกระทำที่เหมาะสมในทางตรงกันข้ามเหตุการณ์ที่เป็นไปไม่ได้จะกลายเป็นข้อผิดพลาดที่จะต้องรายงานให้ทราบการเชื่อมต่อเริ่มต้นจากสถานะ CLOSED เมื่อเรียกใช้บริการ LISTEN หรือ CONNECT ก็จะมีการเปลี่ยนสถานะไปจากเดิมและถ้าอีกฝ่ายตองการเชื่อมต่อด้วยการเชื่อมต่อก็จะเกิดขึ้นและย้ายไปอยู่ในสถานะESTABLISHED คือการเชื่อต่อสมบูรณ์และเมื่อยกเลิกการติดต่อก็จะกลับไปสู่สถานะ CLOSED อย่างเดิม

2.1.7.2 การบันทึกเวลาแบบ Three-way Handshake

Three-way Handshake เป็นวิธีการส่งแพ็กเก็ตที่สามารถช่วยแก้ปัญหาในเรื่องแพ็กเก็ตซ้ำซ้อนได้ดีแต่วิธีนี้จำเป็นจะต้องสร้างช่องสื่อสารให้ได้ก่อนที่จะเริ่มรับ-ส่งข้อมูลอย่างไรก็ตามแพ็กเก็ตควบคุมที่ใช้ในการต่อรองค่าตัวแปรสำหรับการสื่อสารต่างๆอาจเกิดการตกค้างอยู่ในระบบได้ทำให้การกำหนดค่าหมายเลขลำดับมีปัญหาไปด้วยเช่นการสร้างช่องสื่อสารระหว่างโฮสต์1 และโฮสต์2 เริ่มจากโฮสต์1 ขอเริ่มการเชื่อต่อด้วยการส่งแพ็กเก็ต CR (Connection Request) ไปยังโฮสต์ 2 ซึ่งจะมีค่าตัวแปรต่างๆสำหรับการสื่อสารรวมทั้งหมายเลขลำดับและหมายเลขช่องสื่อสารไปด้วย ผู้รับคือโฮสต์2 ก็จะส่ง ACK (Acknowledge)กลับมายังโฮสต์1 แต่ถ้าแพ็กเก็ตจากผู้ส่งเกิดสูญหายระหว่างทางและสำเนาแพ็กเก็ตที่ยังตกค้างอยู่ระบบเกิดเดินทางไปถึงผู้รับในภายหลังก็จะทำให้การสร้างช่องสื่อสารใช้การไม่ได้เนื่องจากมีค่าตัวแปรต่างๆไม่ตรงกันการใช้ Three-way handshake เป็นการไม่บังคับให้ผู้ส่งและผู้รับข้อมูลจะต้องกำหนดค่าเริ่มต้นของหมายเลขลำดับเป็นเลขเดียวกัน ทำให้สามารถนำวิธีนี้มาใช้ร่วมกับวิธีการจัดจังหวะการทำงานให้พร้อมกัน(Synchronization) แบบต่างๆได้แทนที่จะเป็นการใช้วิธีการบันทึกเวลาดังรูปที่ 7-1 แสดงขั้นตอนการเริ่มต้นการทำงานจากโฮสต์ 1 ไปยังโฮสต์ 2 สมมุติให้โฮสต์ 1 เลือกหมายเลขลำดับเป็น “x” และส่งแพ็กเก็ต CONNECTION REQUEST ไปยังโฮสต์ 2 โฮสต์ 2 ตอบรับด้วยแพ็กเก็ต CONNECTION ACCEPTED ซึ่งจะยอมรับหมายเลขลำดับ “x” พร้อมกับประกาศหมายเลขลำดับ “y” ที่เป็นของตนเองจากนั้นโฮสต์ 1 ก็จะตอบรับค่าตัวเลือกของโฮสต์ 2 ผ่านทางเขตข้อมูลสำหรับการควบคุมในแพ็กเก็ตข้อมูลแรกที่ส่งมา



รูปที่ 2.8ขั้นตอนการทำงานจากโฮสต์1 ไปยังโฮสต์2

สมมติว่าได้เกิดปัญหาการสูญหายของแพ็กเก็ตในขณะที่สำเนาแพ็กเก็ตที่ค้างในระบบเดินทางไปถึงผู้รับแทนรูปที่7-2 แสดงเหตุการณ์ที่แพ็กเก็ตTPDU (ตัวแรกในรูป) เป็นสำเนาแพ็กเก็ตเก่าที่พึ่งจะเดินทางไปถึงโฮสต์ 2 โดยที่โฮสต์ 1 ไม่ทราบโฮสต์ 2 ก็จะทำงานตามปกติคือจะตอบรับด้วยการส่งแพ็กเก็ตCONNECTION ACCEPTED TPDU กลับมาที่โฮสต์ 1 ซึ่งโฮสต์1 จะสามารถตรวจสอบได้ว่าหมายเลขลำดับโฮสต์2 ตอบกลับมานั้นเป็นหมายเลขลำดับที่ได้เลิกใช้ไปแล้วจึงมีการส่งแพ็กเก็ต REJECTกลับมายังโฮสต์ 2 เพื่อบอกยกเลิกการทำงานจะเห็นว่าวิธีการนี้อาศัยการสื่อสารผ่านแพ็กเก็ต 3 ตัวซึ่งเป็นที่มาของคำว่า "การจับมือร่วมสามขั้นตอน" ผลสุดท้ายทั้งโฮสต์ 1 และโฮสต์ 2 ก็จะไม่มีการสร้างช่องสื่อสารขึ้นมาจากข้อมูลในสำเนาแพ็กเก็ตเก่าแต่อย่างใด

2.1.7.3 ชั้นสื่อสารการประยุกต์

มีโพรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือนเรียกว่า TELNET โพรโตคอลสำหรับการจัดการแฟ้มข้อมูลเรียกว่า FTP และโพรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์เรียกว่า SMTP โดยโพรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่องโฮสต์ที่อยู่ไกลออกไปโดยผ่านอินเทอร์เน็ตและสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่เครื่องโฮสต์นั้นโพรโตคอลสำหรับการจัดการแฟ้มข้อมูลช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่ายหรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องใดๆก็ได้โพรโตคอลสำหรับให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบหรือรับข้อความที่มีผู้ส่งเข้ามา

2.2 พื้นฐานการใช้งาน GDI+

คำว่า GDI+ (อ่านว่า จีดีไอพลัส) ย่อมาจากคำว่า Graphics Device Interface เป็นคำที่ใช้เรียกกลุ่มออบเจ็กต์(Object)ที่รับผิดชอบด้านการแสดงผลในสถาปัตยกรรม .NET GDI+ แยกได้ 5 ลักษณะ คือ

1. ส่วนแสดงผลของคอนโทรลต่าง ๆ
2. การจัดการเกี่ยวกับระบบไฟล์รูปภาพ
3. ระบบการพิมพ์เอกสารไปยังเครื่องพิมพ์
4. การทำภาพเคลื่อนไหวแอนิเมชั่น(Animation)
5. ระบบฟอนต์และสี

2.2.1 พื้นฐานระบบกราฟิกใน GDI+

พื้นฐานการแสดงกราฟิกหน่วยที่เล็กที่สุดคือ "จุด" เมื่อนำจุดมาเรียงต่อกันจะได้เส้นตรงและเมื่อจุดหลายจุด หลายสีมาเรียงต่อกันก็จะได้ภาพขึ้นมาออบเจ็กต์ที่ทำหน้าที่รับผิดชอบหน่วยกราฟิกที่เล็กที่สุดมีอยู่ 2 ตัว คือ

1. **ออบเจ็กต์ Point** ทำหน้าที่ระบุพิกัดคู่ลำดับ (Co-ordinate) x,y โดยเทียบจากมุมซ้ายบนของพื้นที่แสดงผล
2. **ออบเจ็กต์ Size** ทำหน้าที่กำหนดพื้นที่จำลองขึ้นมา เพื่อสร้างงานด้านกราฟิกโดยทำงานร่วมกับออบเจ็กต์ Rectangle อาจจะใช้พื้นที่แสดงผลทั้งหมดของฟอร์ม หรืออยู่ในขอบเขตของพื้นที่จำลองก็ได้
3. พื้นที่แสดงผล หมายถึง พื้นที่แสดงผลทั้งหมดของฟอร์ม เรียกว่า ClientRectangle
4. พื้นที่จำลอง หมายถึง พื้นที่ที่คุณสร้างขึ้นมาด้วยออบเจ็กต์ Size เพื่อใช้เป็นเนื้อที่สำหรับแสดงงานด้านกราฟิก สามารถกำหนดขนาดพื้นที่จำลองได้ที่เราต้องการ ซึ่งอาจจะใช้พื้นที่บางส่วน หรือทั้งหมดของพื้นที่แสดงผลในฟอร์มก็ได้

2.2.2 ขั้นตอนการสร้างพื้นที่จำลองใน GDI+

1. ใช้ออบเจ็กต์ Point ระบุตำแหน่ง โดยวัดจากมุมซ้ายบนของฟอร์มกับมุมซ้ายบนของพื้นที่จำลอง
2. ใช้ออบเจ็กต์ Size กำหนดขนาดพื้นที่จำลองที่ต้องการสร้างขึ้นมา

2.2.3 ระบบสีใน GDI+

ก่อนที่เราจะเริ่มสร้างงานด้านกราฟิก VC# พื้นฐานแรกที่เราควรจะทราบก็คือ ระบบสี ในสถาปัตยกรรม GDI+ จะกำหนดให้ออบเจ็กต์ Color รับผิดชอบส่วนของการแสดงสีทั้งหมด โดยอ้างอิงระบบแม่สีแบบ RGB (แดง เขียว น้ำเงิน)

2.2.4 การใช้สีจากออบเจ็กต์ Color

สามารถระบุได้ 2 ลักษณะ

1. อาศัยเมธอด FromArgb() ทำหน้าที่ผสมแม่สี 3 สี คือ Red, Green, Blue
2. ระบุชื่อสีที่ออบเจ็กต์ Color สนับสนุน เช่น Color.White, Color.Red

2.2.5 การใช้งานเหตุการณ์ Paint() ของฟอร์ม

สำหรับวิธีการที่สามารถสั่งให้วาดงานด้านกราฟิกลงในพื้นที่แสดงผลของฟอร์ม สามารถทำได้ 2 วิธีคือ

1. อาศัยเหตุการณ์ Paint() ของฟอร์มเป็นวิธีพื้นฐาน โดยกำหนดให้เมื่อเกิดเหตุการณ์ Form_Paint() ขึ้นแล้ว สั่งให้วาดงานด้านกราฟิกลงในพื้นที่แสดงผล โดยเหตุการณ์ Form_Paint() เป็นเหตุการณ์ที่เกิดต่อจากเหตุการณ์ Form_Load() อัตโนมัติ
2. การทำ override เหตุการณ์ Form_Paint() ของฟอร์ม เป็นวิธีการที่เราสามารถจำลองหรือเลียนแบบเหตุการณ์ Form_Paint() โดยการใช้คำสั่ง override

2.2.6 การใช้งานออบเจ็กต์ Pen

ออบเจ็กต์ Pen ใช้สำหรับวาดเส้น โดยการกำหนดจุดเริ่มต้นและจุดสิ้นสุดด้วยออบเจ็กต์ Point ซึ่งเป็นออบเจ็กต์ที่มีลักษณะใกล้เคียงกับปากกาของจริง เพราะว่าในการเลือกปากกา สิ่งที่ต้องคำนึงถึงมี 2 อย่างคือ

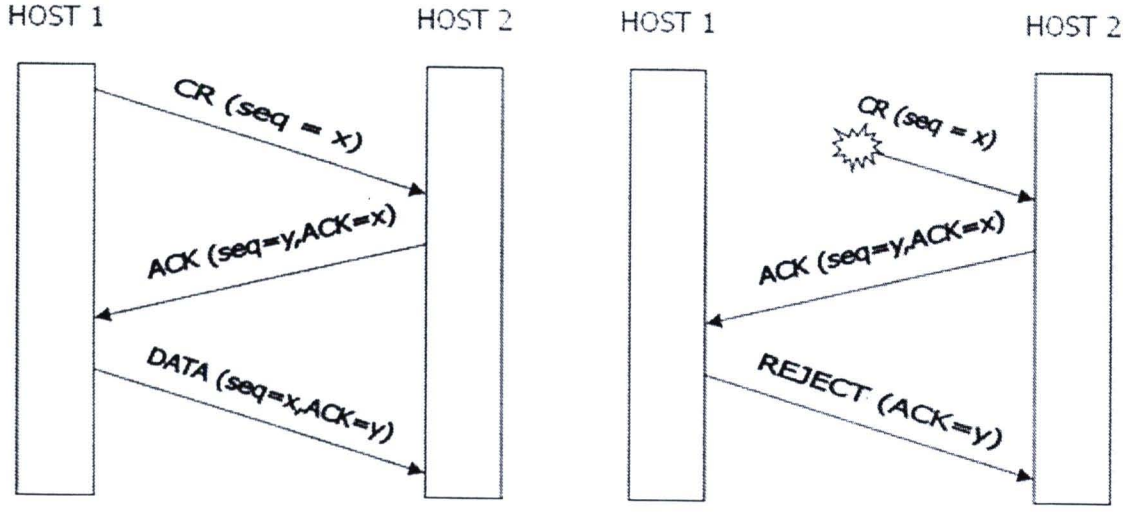
1. สีน้ำหมึกของปากกา เช่น หมึกสีแดง หมึกสีดำ ฯลฯ
2. ขนาดหัวปากกาที่ใช้ เช่น ขนาด 0.5, 0.3 ฯลฯ

นั่นคือ สิ่งที่คิดเมื่อใช้ปากกาในโลกของความเป็นจริง แต่ในโลกของ GDI+ แล้วปากกาสามารถกำหนดได้ 3 อย่างคือ

1. สีน้ำหมึก
2. ขนาดหัวปากกา
3. ลวดลายเส้นที่ออกจากหัวปากกา

2.3 การสื่อสารแบบ Wi-Fi

การสื่อสารแบบ Wi-Fi มีลักษณะเป็นเน็ทเวิร์ค ซึ่งประกอบด้วยอุปกรณ์กระจายสัญญาณ Router ทำหน้าที่เป็นตัวกลาง และควบคุมการสื่อสาร การสื่อสารแบบนี้โปรแกรมจะต้องทำหน้าที่เป็น Server และ Client กล่าวคือตัวบอร์ดวัดอุณหภูมิจะต้องมีโปรแกรมในส่วนของ Server หรือ Client ติดตั้งอยู่ ซึ่งในโครงงานวิจัยนี้ โปรแกรม Server ถูกติดตั้งเอาไว้ ดังนั้นโปรแกรม Client จะถูกติดตั้งไว้ในโปรแกรมบันทึกค่าและแสดงผล ซึ่งจะถูกติดตั้งไว้บนระบบปฏิบัติการ Windows โครงสร้างการสื่อสาร



รูปที่ 2.9 ขั้นตอนการทำงานจากโฮสต์1 ไปยังโฮสต์2

2.4 สรุป

เหตุผลข้อหนึ่งของการทำโครงงานวิจัยชิ้นนี้ก็เพื่อนำผลลัพธ์ที่ได้ไปใช้ให้เกิดประโยชน์กับการทำการทดลองของนักศึกษา ซึ่งปัจจุบันนักศึกษามีคอมพิวเตอร์เป็นของตนเองเกือบทุกคน และคอมพิวเตอร์เหล่านี้สามารถสื่อสารแบบ Wi-Fi ได้ทั้งสิ้น นั่นคือโปรแกรมเก็บข้อมูลไร้สายที่โครงงานวิจัยนี้สร้างขึ้น นำมาติดตั้งและใช้ร่วมกับบอร์ดทดลองที่สร้างขึ้นได้เป็นอย่างดี ดังนั้นถ้าใช้วิธีนี้ในการทำการทดลอง จะทำให้นักศึกษาทำการเก็บข้อมูลได้สะดวกและนำข้อมูลไปวิเคราะห์ได้ทันที การทำแบบนี้นอกจากจะสะดวกต่อการทำการทดลองแล้วยังประหยัดในส่วนของเครื่องมือวัดด้วย ซึ่งเครื่องมือวัดเหล่านี้มีข้อจำกัดในเรื่องของการดึงข้อมูลไปวิเคราะห์ต่อ ซึ่งส่วนใหญ่แล้วเครื่องมือวัดเหล่านี้จะเน้นไปทางแสดงผลเป็นหลัก

นอกเหนือจากประโยชน์หลักๆ ที่กล่าวไปแล้ว ความรู้ที่ได้ยังสามารถนำไปใช้ในด้านอื่นๆ เช่นเซนเซอร์ไร้สาย รวมถึงอุปกรณ์ควบคุมไร้สาย ทั้งนี้เนื่องจากระบบต่างๆ มีทิศทางไปในทิศเดียวกัน คือการรับส่งข้อมูลไร้สายนั่นเอง