กรณีศึกษาความสัมพันธ์ของขนาดข้อมูลสูงสุด ขนาดของหน้าต่าง และอัตราความผิดพลาดของโปรโตคอล TCP

A Comparative Study of MTU, Windows Size and Error Rate in TCP

โดย

นาย เลิศวิทย์ วรพงศธร

รหัสนักศึกษา 40067290

อาจารย์ที่ปรึกษา

อาจารย์ อัครินทร์ คุณกิตติ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชาโครงการศึกษากรณีพิเศษ

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2541

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การศึกษาความสัมพันธ์ของขนาดข้อมูลสูงสุด ขนาดหน้าต่างและอัตราการผิดพลาดของโปรโตคอล TCP
นักศึกษา	นาย เลิศวิทย์ วรพงศธร
อาจารย์ที่ปรึกษา	อาจารย์ อัครินทร์ คุณกิตติ
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	การจัดการเทคโนโลยีสารสนเทศ
ปีการศึกษา	2541

บทคัดย่อ

วัตถุประสงค์ของโครงงานนี้ เพื่อทำการศึกษาคุณสมบัติเฉพาะของโปรโตคอล TCP (Transmission Control Protocol) และการทำงานของชุดโปรโตคอล TCP/IP (Transmission Control Protocol / Internet Protocol) ซึ่งมีลักษณะการทำงานที่สัมพันธ์กันของอัตราการส่งผ่านข้อมูล (MTU, Maximum Transfer Unit) ในปริมาณต่าง ๆ ในรูปความสามารถในการใช้ช่องสัญญาณ (Utilization) โดยพิจารณาจากค่าดำเนินการ (Overhead) ที่เกิดขึ้นจากการส่งข้อมูล และขนาดหน้าต่าง (Windows Size) ที่ใช้ รวมไปถึงความผิดพลาดที่เกิดขึ้น จากการศึกษาดังกล่าวพบว่าความสัมพันธ์ของตัวแปรทั้งสามมีความสัมพันธ์กันซึ่งความสามารถในการใช้ช่องสัญญาณของ โปรโตคอล TCP จะมีค่าสูงขึ้น ถ้าหากมีการส่งผ่านข้อมูลขนาดที่มีค่าเข้าใกล้ 65,535 ไบต์ แต่เมื่อมีการพิจารณาถึงค่าความผิดพลาดที่เกิดขึ้นพบว่าถ้ามีการส่งข้อมูลปริมาณสูงโอกาสในการเกิดความผิดพลาดย่อมสูงตาม ส่งผลให้ความสามารถในการใช้ช่องสัญญาณลดต่ำลงอย่างรวดเร็ว และเมื่อพิจารณาค่าขนาดหน้าต่างมาใช้โดยค่าที่เหมาะสมคือ ควรมีค่ามากกว่าหนึ่งหน้าต่าง

Title	A Comparative Study of MTU, Windows Size and Error Rate in TCP
Student	Mr. Lertvith Varapongsathorn
Advisor	Mr. Akharin Khunkitti
Level of Study	Master of Science in Information Technology
Major	Information Techonology Management
Academic Year	1998

ABSTRACT

The objective of project are to analyze the characteristics of TCP (Transmission Control Protocol) and to study the operation of the TCP/IP (Transmission Control Protocol / Internet Protocol). This project is emphasize on format for sending of packet how are utilization, involve occur overhead from communication also consideration in windows size use, compare on percent of error rate and any value of volumes based on MTU (Maximum Transfer Unit).

The result from study shows that the utilization of TCP will be improve if they communicate in data size nearly 65,535 byte and consideration the error value, that when transfer more data the error will be found then the utilization decrease. So the default windows size value should more than one packet along to Microsoft's recommend.

สารบัญ

สารบัญตาราง

สารบัญภาพ

ภาพที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความเป็นมา

ปัจจุบันในองค์กรต่าง ๆ ได้มีการนำเอาเทคโนโลยีทางด้านคอมพิวเตอร์มาประยุกต์ใช้งานในการติดต่อสื่อสาร หรือการแลกเปลี่ยนข้อมูลกัน โดยเฉพาะอย่างยิ่งเทคโนโลยีทางด้านการติดต่อสื่อสารมีการพัฒนาไปอย่างรวดเร็ว ทำให้มีการใช้งานสะดวกและรวดเร็ว ทำให้มีการทำงานเป็นระบบเครือข่าย สามารถใช้อุปกรณ์ต่าง ๆ ร่วมกันและสะดวกมาขึ้น ช่วยทำให้สามารถลดต้นทุนและเพิ่มประสิทธิภาพในการทำงาน โดยนำเครื่องคอมพิวเตอร์มาใช้งานในลักษณะที่เป็นเครือข่ายท้องถิ่น (Local Area Network, Lan) ที่มีการนำเอาโปรโตคอลการสื่อสารข้อมูลที่มีชื่อว่าโปรโตคอล TCP/IP (Transmission Control Protocol/ Internet Protocol) ซึ่งเป็นโปรโตคอลที่ถูกพัฒนาโดย US Department of Defense (DoD) เมื่อกลางปี ค.ศ. 1970 โดยโปรโตคอลชุดนี้ได้มีการนำมาใช้งานในการเชื่อมต่อระหว่างเครือข่ายคอมพิวเตอร์มากมายเพื่อให้เป็นการเข้าใจถึงหลักการทำงานของชุดโปรโตคอลผู้เขียนจึงได้ทำการศึกษาวิเคราะห์ชุดโปรโตคอล TCP ว่ามีลักษณะการทำงานเป็นอย่างไร โดยจะทำการศึกษารูปแบบการส่งข้อมูล โดยจะเน้นพิจารณาไปที่การศึกษาการส่งข้อมูลในปริมาณต่าง ๆ ว่ามี Format Utilization เป็นอย่างไร ซึ่งจะมีการพิจารณาจาก Overhead ที่เกิดขึ้นจากการส่งข้อมูล, Windows Size และค่าความผิดพลาดที่เกิดขึ้นว่าคิดเป็นเปอร์เซนต์ของการเกิดข้อผิดพลาดเป็นเท่าไรเมื่อมีการส่งข้อมูลที่ปริมาณ MTU ต่าง ๆ กัน เพื่อหาความสัมพันธ์ของตัวแปรทั้งสามว่ามีความสัมพันธ์กันอย่างไร

1.2 วัตถุประสงค์

1. เพื่อศึกษาลักษณะการทำงานของชุดโปรโตคอล TCP/IP โดยเน้นที่ TCP
2. เพื่อศึกษาหาความสัมพันธ์ของขนาด MTU (Maximum Transfer Unit) ขนาด Windows Size และค่าความผิดพลาดของการสื่อสารว่าเกี่ยวข้องกับการสื่อสารของโปรโตคอล TCP อย่างไร
3. เพื่อหาประสิทธิภาพหรือข้อจำกัดของการสื่อสารข้อมูลของโปรโตคอล TCP

1.3 แผนการดำเนินการศึกษา

1. ทำการศึกษาคุณลักษณะการทำงานของชุดโปรโตคอล TCP/IP และ TCP
2. ศึกษาถึงสมการการสื่อสารข้อมูลของ TCP แบบพิจารณาความผิดพลาดและแบบไม่พิจารณาความผิดพลาดของการสื่อสารว่ามีความแตกต่างกันอย่างไร
3. พิจารณาค่า MTU ค่า Windows Size และค่าความผิดพลาดจากการสื่อสารว่ามีความสัมพันธ์กันอย่างไร
4. ทำการเปรียบเทียบประสิทธิภาพของการส่งข้อมูลที่ปริมาณ MTU ต่าง ๆ รวมไปถึง Windows Size ที่แตกต่างกันว่าจะเกิดเปอร์เซ็นต์ความผิดพลาดมากน้อยเพียงใด

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้ความเข้าใจในหลักการทำงานของชุดโปรโตคอล TCP/IP
2. ทราบถึงประสิทธิภาพการสื่อสารข้อมูล TCP ว่ามีข้อจำกัดหรือเงื่อนไขการสื่อสารในด้านของขนาดปริมาณข้อมูลอย่างไร

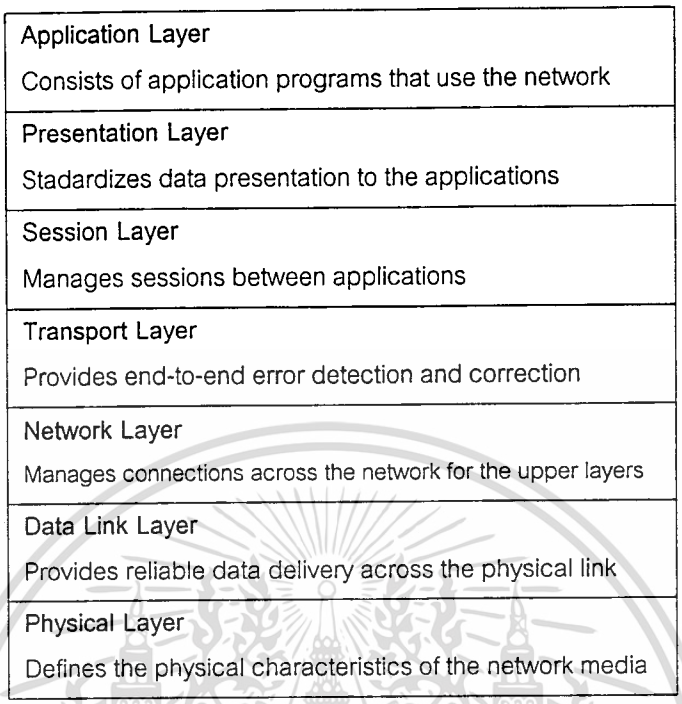
บทที่ 2

โปรโตคอล TCP/IP

2.1 ประวัติความเป็นมาของ TCP/IP

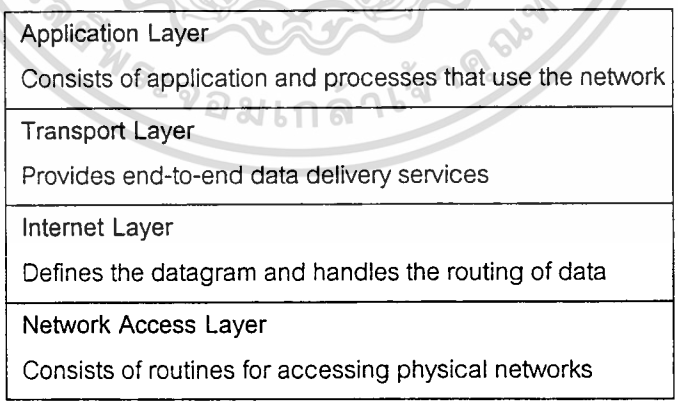
สืบเนื่องมาจากแนวความคิดของกระทรวงกลาโหมของประเทศสหรัฐอเมริกาที่ต้องการเชื่อมต่อเครื่องคอมพิวเตอร์ที่มีอยู่มากมายหลายยี่ห้อเข้าด้วยกันเป็นระบบเครือข่ายเดียว โดยมอบหมายให้ DARPA (Defense Advanced Research Project Agency, ซึ่งแต่เดิมเรียกว่า ARPA) เป็นผู้รับผิดชอบ ในช่วงปีฤดูใบไม้ร่วง ค.ศ. 1969 นับเป็นจุดเริ่มต้นของชุดโปรโตคอล TCP/IP เพราะมีการแสดงให้เห็นถึงการเชื่อมต่อเครื่องคอมพิวเตอร์จาก 4 แห่งเข้าหากัน ซึ่งประกอบไปด้วย University of California at LosAngeles (UCLA), Stanford Research Institute (SRI), University of California at Santa Barbara (UCSB) และ University of Utah โดยเรียกเครือข่ายนี้ว่า ARPANET และมีบริษัท Bolt Beranek and Newman (BBN) เป็นผู้ดูแลการเชื่อมโยงเข้าหากัน รูปแบบการสื่อสารข้อมูลใช้เทคนิคของการสวิตซ์กลุ่มข้อมูล (Packet Switching) ในปี ค.ศ. 1972 เป็นการสาธิตความสามารถของระบบเครือข่ายเป็นครั้งแรก โดยมีเครื่องคอมพิวเตอร์จำนวน 50 เครื่องต่อเข้ากับระบบที่มีอุปกรณ์สวิตซ์กลุ่มข้อมูลทำหน้าที่ส่งผ่านข้อมูลระหว่างเครื่องคอมพิวเตอร์ ตั้งแต่นั้นมาจนถึงปัจจุบันระบบสื่อสารข้อมูลโดยใช้ชุดโปรโตคอล TCP/IP ก็เป็นที่ยอมรับแก่คนทั่วไป จนกลายเป็นมาตรฐานแบบที่เรียกว่า "De Facto"

หลักการทำงานเริ่มต้นด้วยการศึกษาการทำงานของแต่ละชั้น (Layer) การทำงานทั้งหมดของโปรโตคอลจะประกอบไปด้วยหลาย ๆ ชั้น ซึ่งนำมาวางซ้อนทับกันได้ออกมาในรูปที่เรียกว่า Protocol Stack แต่ละชั้นจะมีหน้าที่การทำงานที่ชัดเจน และไม่เกี่ยวข้องกัน แต่ละชั้นจะรู้เพียงวิธีการส่งข้อมูลไปยังชั้นอื่น ๆ แต่ไม่รู้ถึงการทำงานข้างใน แต่ละ โปรโตคอล จะมีการแบ่งการทำงานออกเป็นจำนวนชั้นไม่เท่ากัน ทำให้เป็นการยากที่จะระบุว่า Network Protocol โดยรวมมีการทำงานกี่ชั้น แต่ก็มีมาตรฐานที่เป็นที่ยอมรับกันโดยทั่วไป เรียกว่า Open System Interconnection (OSI) Reference Model ซึ่งทำการแบ่งการทำงานของ Network Protocol ออกเป็น 7 ชั้น ดังรูปที่ 1



รูปที่ 1 แสดงลำดับโครงสร้างของ OSI Model

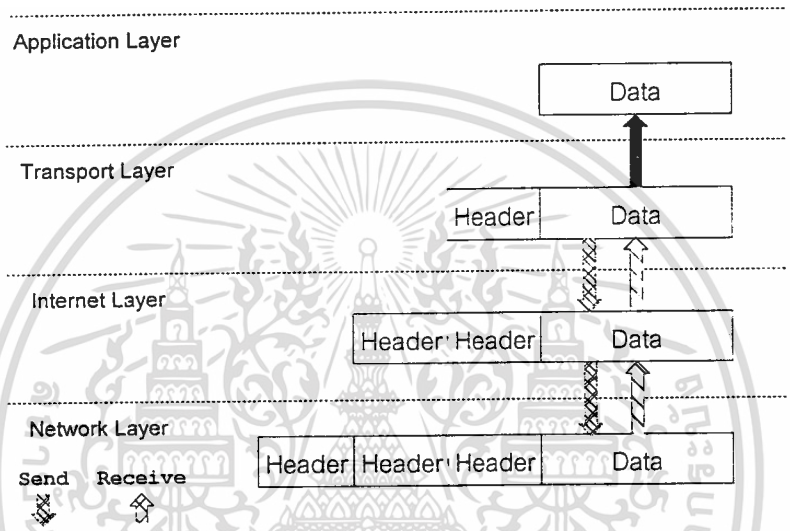
สำหรับการศึกษา TCP/IP ไม่อ้างอิง OSI Reference Model โดยจะทำการสร้างโครงสร้างขึ้นมาใหม่โดยแบ่งออกเป็น 4 ชั้น ดังรูปที่ 2



รูปที่ 2 แสดงลำดับโครงสร้างของ TCP/IP Model

ลักษณะการทำงานคือ ข้อมูลจะถูกส่งลงมาจากชั้นบนลงมาข้างล่าง ขณะที่ข้อมูลถูกส่งผ่านในแต่ละชั้นจะทำการเพิ่มข้อมูลควบคุมเข้าไปเพื่อให้การส่งข้อมูลถูกต้อง และเป็นการส่ง

พารามิเตอร์ที่จำเป็นไปให้กับชั้นของมันในเครื่องปลายทาง ข้อมูลเหล่านี้เรียกว่า Header แต่ละชั้นจะมี Header ที่มีรูปแบบเป็นของตัวเอง การเพิ่ม Header เข้าไปรวมกับข้อมูลเรียกว่า Data Encapsulation แสดงดังรูปที่ 3 โดยกระทำเช่นนี้ไปเรื่อย ๆ จนกระทั่งส่งออกไปยังสายสื่อสาร เมื่อเครื่องปลายทางได้รับเฟรมข้อมูลก็จะถอดรหัสส่วนหัวออก แล้วนำข้อมูลให้กับ Layer ชั้นบนต่อไป เป็นเช่นนี้ไปเรื่อย ๆ จนถึงชั้นโปรแกรม Application



รูปที่ 3 แสดงถึงลักษณะ Data Encapsulation

2.2 ลักษณะทำงานแต่ละชั้น

2.2.1 Network Access Layer

ในชั้นนี้จะทำหน้าที่จัดส่งข้อมูลผ่านสายสัญญาณชนิดต่าง ๆ โดยการทำงานจะเป็นเช่นไรขึ้นอยู่กับชนิดของสายสัญญาณที่ใช้ สายแต่ละชนิดก็จะมี โปรโตคอล ที่ใช้ควบคุมการทำงานที่แตกต่างกัน เช่น อีเทอร์เนต, FDDI, Frame Relay เป็นต้น

ลักษณะงานที่เกิดขึ้นในชั้นนี้ ได้แก่

- การเพิ่ม Header เข้าไปใน Datagram เพื่อให้กลายเป็น Frame แล้วส่งไปตาม Network
- การเปลี่ยนเลข IP ไปเป็น Physical address เป็นต้น

2.2.2 Internet Layer

ในชั้นนี้มีโปรโตคอล ที่ทำงานอยู่ 2 ชนิดคือ IP (Internet Protocol) และ ICMP (Internet Control Message Protocol)

2.2.2.1 Internet Protocol (IP)

เป็นโปรโตคอลที่ให้บริการแก่โปรโตคอลชั้นที่สี่ ในแบบที่เรียกว่าเป็นโปรโตคอลชนิด Connectionless คือ จะไม่ทำการ Handshake กับสถานีปลายทางก่อนแต่จะทำการส่งข้อมูลไปเลย โดยไม่คำนึงถึงว่ามันจะไปถึงปลายทางหรือไม่ นอกนั้นยังไม่มีการตรวจสอบความถูกต้องของข้อมูลที่ส่งไปด้วย กล่าวคือเพียงแต่ทำการส่งข้อมูลไปให้ถึงปลายทางเท่านั้นโดยมีสนใจว่าข้อมูลนั้นเหมือนกับข้อมูลต้นทางหรือไม่ โปรโตคอลนี้มีหน้าที่สำคัญคือ

- สร้าง Datagram
- หาเส้นทางเพื่อทำการส่ง Datagram
- แบ่งและประกอบ Datagram

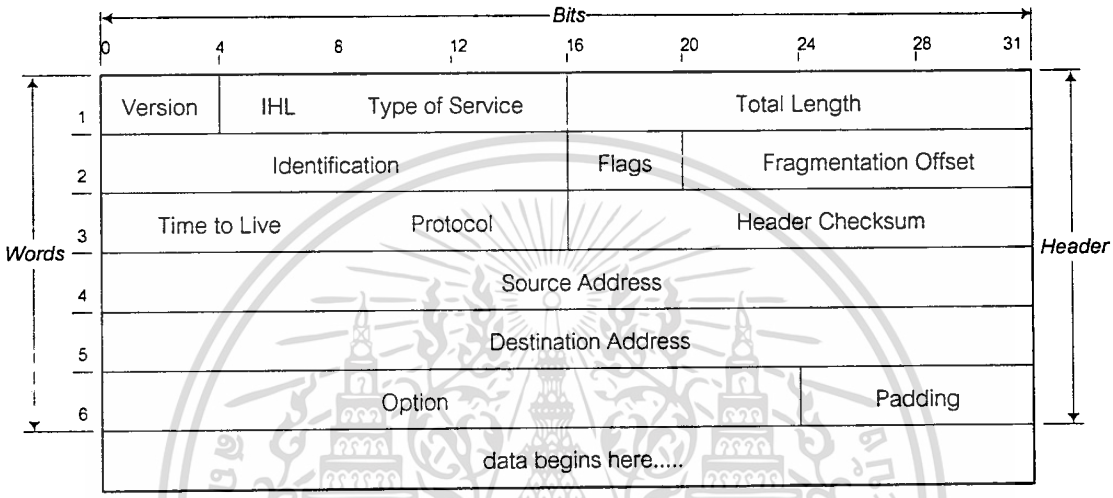
Datagram

Datagram เป็น Basic Unit ของข้อมูลที่ IP ทำการส่ง ซึ่งถูกออกแบบมาให้ทำงานกับเครือข่ายแบบ Packet Switch ซึ่งข้อมูลของผู้ใช้มักถูกแบ่งออกเป็นหลาย ๆ Datagram โดยแต่ละตัวจะมี Header ที่เก็บรายละเอียดเกี่ยวกับตัวมันเอง และปลายทางที่ต้องส่งไป

Header ของ Datagram ได้แสดงไว้ในรูปที่ 4 โดยประกอบไปด้วย 6-Word แต่ละ Word มีขนาด 32 Bit ปกติข้อมูลต่าง ๆ ที่จำเป็นจะถูกเก็บไว้ใน 5-Word แรก ส่วน Word ที่ 6 นั้นอาจจะมีหรือไม่มีก็ได้ ซึ่งทำให้ขนาดของ Header ไม่แน่นอน ดังนั้นใน Header จะมี Field หนึ่งที่ชื่อ Internet Header Length (IHL) คอยบอกความยาวของ Header นั้น ๆ สำหรับรายละเอียดต่างมีดังนี้

■ **Version (Ver)**

มีขนาด 4 บิต แสดงถึงรุ่นของโปรโตคอล IP เนื่องจากการตีความในเขตต่าง ๆ ของโปรโตคอล IP ใช้ซอฟต์แวร์เป็นตัวจัดการ ดังนั้นจึงจำเป็นต้องมีการบ่งบอกถึงรุ่นของโปรโตคอล IP สำหรับในกรณีที่รุ่นไม่ตรงกันจะไม่มีการตีความเกิดขึ้น



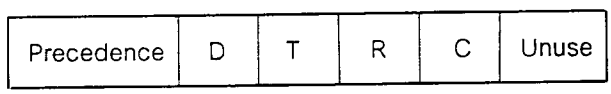
รูปที่ 4 แสดงโครงสร้างต่าง ๆ ของ IP แพคเก็ต

■ **Header (Header Length)**

มีขนาด 4 บิต บอกถึงความยาวเฉพาะส่วนหัวของ IP แพคเก็ต (ส่วนหัวของ IP แพคเก็ตเริ่มนับจากเขต Version ไปจนถึงไบต์สุดท้ายก่อนที่จะถึงเขต Data) หน่วยนับเป็นจำนวนเท่าของ 4 ไบต์ เช่น ถ้า Header มีค่าเท่ากับ 5 หมายถึงส่วนหัวมีขนาด 20 ไบต์

■ **Type of Service (ToS)**

มีขนาด 8 บิต แบ่งออกเป็น 6 เขตย่อยดังแสดงในรูปที่ 5



รูปที่ 5 แสดงเขตย่อยต่าง ๆ ภายใต้เขต TOS

■ **Precedence**

มีขนาด 3 บิต ใช้ในการกำหนดลำดับความสำคัญของ IP แพคเก็ต ในกรณีที่ปริมาณของแพคเก็ตมีอยู่มากในระบบ อุปกรณ์หาเส้นทางจะไม่ทำการหาเส้นทางให้กับแพคเก็ตที่มีลำดับความสำคัญต่ำสุด หรืออาจกล่าวได้ว่าคือ ตัดแพคเก็ตนั้นออกจากระบบ

- D บิต มีขนาด 1 บิต ใช้บอกถึงความต้องการการให้บริการจากเครือข่าย ถ้าบิตนี้มีค่าเป็น 1 หมายถึงต้องการเส้นทางที่มีการหน่วงเวลา (Delay Time) ต่ำที่สุดเท่าที่ระบบจะหาให้ได้

- T บิต มีขนาด 1 บิต ถ้าบิตนี้มีค่าเป็น 1 หมายถึงต้องการเส้นทางที่มีความสามารถส่งผ่านข้อมูลได้ปริมาณมาก ๆ ในหนึ่งหน่วงเวลา

- R บิต มีขนาด 1 บิต ถ้าบิตนี้มีค่าเป็น 1 หมายถึงต้องการเส้นทางหรือบริการที่มีความเชื่อถือได้สูง

- C บิต มีขนาด 1 บิต ถ้าบิตนี้มีค่าเป็น 1 หมายถึงต้องการเส้นทางที่มีค่าพารามิเตอร์ของ Cost ต่ำ หมายถึงเป็นเส้นทางที่ดีที่สุด

- Unused มีขนาด 1 บิต ยังไม่มีการนำบิตนี้มาใช้งาน

■ **Total Length**

มีขนาด 16 บิต ใช้บอกถึงความยาวของ IP แพคเก็ตทั้งหมด (ส่วนหัวและส่วนข้อมูล) หน่วยนับเป็นจำนวนเท่าของไบต์ ดังนั้น IP แพคเก็ตมีความยาวสูงสุดเท่ากับ 2^{16} -1 หรือ 65.535 ไบต์

■ **Identification**

มีขนาด 16 บิต ใช้บอกถึงหมายเลขของ IP แพคเก็ต เขตนี้ถูกกำหนดขึ้นมาเพื่อไม่ให้สถานีปลายทางสับสนว่าตัวมันเองได้รับแพคเก็ตที่สมควรได้รับแล้วหรือไม่ เพราะการส่งผ่านข้อมูลในลักษณะ Connectionless มีโอกาสที่ข้อมูลจะสูญหายหรือซ้ำได้ ดังนั้นการกำหนดหมายเลขนี้เพื่อป้องกันการสับสนที่อาจจะเกิดขึ้นได้

■ **Flags**

มีขนาด 3 บิต ใช้ในการควบคุม Fragmentation และ Reassemble โดยที่บิตแรกจะถูกตั้งค่าเป็น 0 ซึ่งอีก 2 Bit จะเป็นบิต D และ M จะเป็นการกำหนดว่า Datagram นี้จะทำการ Fragmentation หรือไม่ และหากมี Fragment แล้ว Fragment นี้จะเป็น Fragment สุดท้ายหรือไม่

ค่า	บิต D	บิต M
0	Fragmentation Allow	Last Fragment
1	Fragmentation Not Allow	More Fragments Exits

ตารางที่ 1 แสดงค่าบิต D และ M ของฟิลด์ Flags

- **Fragmentation Offset**

มีขนาด 13 บิต ใช้ในการพิจารณาร่วมกับเขต Identification และ Total Length เพื่อเรียงลำดับและหาขนาดความยาวของแพคเก็ตที่แท้จริงที่ถูกส่งออกมาจากสถานีต้นทาง

- **Time to Live (TTL)**

มีขนาด 8 บิต เป็นเขตที่ใช้ในการกำหนดอายุของ IP แพคเก็ต (วินาที) โดยปกติอายุของแพคเก็ตเท่ากับ 255 (2^8-1) ทุกครั้งที่แพคเก็ตผ่านอุปกรณ์หาเส้นทาง ค่าของ TTL จะลดลงหนึ่งเสมอ ถ้าค่าของ TTL เป็นศูนย์ก็หมายความว่าหมดอายุงาน เมื่ออุปกรณ์หาเส้นทางตัวใดพบสภาพเช่นนี้ก็จะนำแพคเก็ตนั้น ๆ ออกจากระบบทันที การกำหนดให้มีเขตนี้เพื่อป้องกันการเกิดแพคเก็ตวนอยู่ในระบบตลอดเวลา ซึ่งก่อให้เกิดความเสียหายแก่ระบบโดยรวมเช่นเป็นสาเหตุให้มีปริมาณแพคเก็ตในระบบมาก สถานีปลายทางอาจจะรับแพคเก็ตผิดได้ เป็นต้น

- **Protocol**

มีขนาด 8 บิต เป็นเขตที่บอกชนิดของโปรโตคอลที่ถูกห่อหุ้ม (Encapsulate) โดยส่วนหัวของ IP แพคเก็ตซึ่งอาจจะเป็นโปรโตคอลชั้นที่สามหรือที่สี่ก็ได้ดังแสดงในตารางที่ 2

โปรโตคอล	ชนิดของโปรโตคอล
1	ICMP
6	TCP
16	CHAOS
17	UDP
89	OSPF

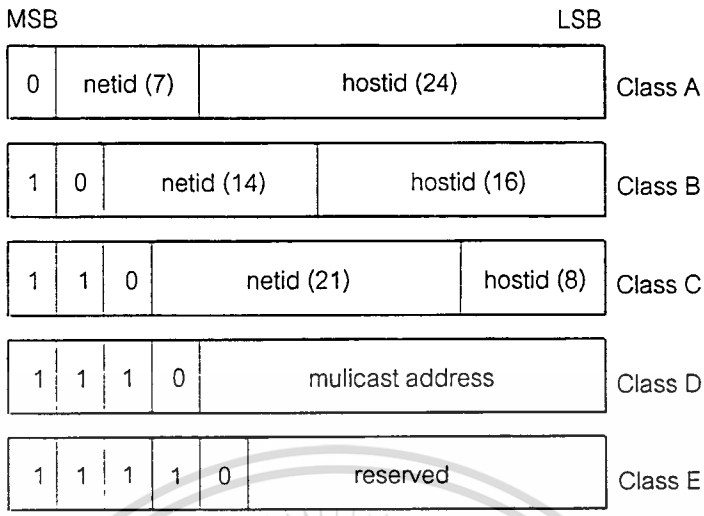
ตารางที่ 2 แสดงตัวอย่างหมายเลขที่แสดงถึงชนิดของโปรโตคอล

■ **Header Checksum**

มีขนาด 16 บิต เป็นเขตที่ใช้ในการตรวจสอบความผิดพลาดเฉพาะส่วนหัวของ IPแพคเก็ต การคำนวณจะใช้วิธีบอกเลข 16 บิตแบบ 1's Complement เมื่อได้ผลลัพธ์แล้วจะทำการผกผัน (Inverse) ผลลัพธ์อีกครั้งหนึ่งจึงจะได้ Checksum ที่แท้จริง การตรวจสอบกระทำเฉพาะส่วนหัวของข้อมูล เพราะส่วนข้อมูลของ IP แพคเก็ตจะเป็นโปรโตคอลชั้นที่สี่ซึ่งมี Checksum เป็นของตัวเองอยู่แล้ว อนึ่ง Checksum ของ IP แพคเก็ตอาจจะไม่ถูกนำมาใช้งานก็ได้ถ้า IPแพคเก็ตถูกห่อหุ้มโดยโปรโตคอลชั้นที่สองที่ใช้ Checksum ที่มีความเชื่อถือได้ เช่น อีเทอร์เนต ในกรณีที่ Checksum ของ IP แพคเก็ตไม่ถูกใช้งานค่าของมันจะเป็นศูนย์ขนาด 16 บิต

■ **Source/Destination Address**

มีขนาดอย่างละ 32 บิต แบ่งออกได้เป็น 5 ประเภทดังแสดงในรูปที่ 6 คือประเภท A, B, C, D และ E แอดเดรสที่ใช้งานทั่วไป (กำหนดเรียกแอดเดรสของสถานี (Host)) คือ 3 ประเภทแรก สำหรับประเภทที่ 4 ใช้ในกรณี พิเศษสำหรับประเภทสุดท้ายสำรองไว้ใช้ในอนาคต



รูปที่ 6 แสดง IP แอดเดรสทั้ง 5 ประเภท

IP แอดเดรสแต่ละประเภทมีหมายเลขที่ไม่ซ้ำกัน ถ้ากำหนดสถานีใด ๆ ด้วย IP แอดเดรสหมายเลขของสถานีก็ไม่ซ้ำกันด้วยเช่นกัน เมื่อดูแค่ 3 ประเภทแรกจะพบว่าภายในแอดเดรสขนาด 32 บิตแบ่งออกเป็น 2 ส่วนย่อยคือ หมายเลขเครือข่าย (Network Identification, Netid) และหมายเลขสถานี (Host Identification, hostid) แต่ละประเภทมีขนาด Netid และ Hostid ไม่เท่ากัน การกำหนดแอดเดรสในการใช้งานมีข้อปลีกย่อยที่กล่าวถึงมีดังนี้

- IP แอดเดรสที่เป็น 0 หมดทุกบิตไม่มีการนำไปใช้งานทั่วไป เว้นแต่จะใช้ในอุปกรณ์หาเส้นทาง (Router) เพื่อกำหนดเป็น "Default Route" เท่านั้น
- ในส่วนของ Netid และ Hostid จะเป็น 0 หรือ 1 หมดทุกบิตไม่ได้ เช่น แอดเดรสประเภท B สามารถกำหนดเป็นอะไรก็ได้ตั้งแต่ 128.0.0.0 ~ 191.254.0.0 แต่ไม่สามารถที่จะมี 128.0.0.0 และ 191.255.0.0 โดยเด็ดขาด

หมายเลขของ Netid ที่ถูกนำไปใช้งานได้ถ้าส่วนของ Hostid เป็น 0 หมดทุกบิต หมายถึงหมายเลขของเครือข่ายเท่านั้น หมายเลขนี้มีประโยชน์ในการใช้งานร่วมกับอุปกรณ์หาเส้นทางหรือใช้บอกให้สถานีทราบว่าตัวมันเองอยู่สังกัดเครือข่ายใด

- หมายเลขของ Netid ที่ถูกนำไปใช้งานได้ถ้าส่วนของ Hostid เป็น 1 หมดทุกบิต หมายถึงแอดเดรสที่ใช้ในการกระจายข่าย (Broadcast Address) ภายในหมายเลขเครือข่ายนั้น ๆ

- IP แอดเดรสที่มีส่วนของ Netid และ Hostid เป็น 1 ทุกบิต หมายถึงแอดเดรสที่ใช้ในการกระจายข่าวเช่นกัน แต่ในกรณีนี้สามารถสรุปได้ว่า สถานีที่ปล่อยแอดเดรสเช่นนี้ลงไปในระบบไม่ทราบว่าตัวเองอยู่ใน Netid หมายเลขเท่าใด
- เฉพาะ IP แอดเดรสประเภท A ที่มีหมายเลข Netid เป็น 127.0.0.0 ถูกสงวนไว้ใช้งานเฉพาะอย่างเท่านั้น เช่น IPC (Inter-Process Communication)

	Class A	Class B	Class C
Netid	1.0.0.0~126.0.0.0	128.1.0.0 ~ 191.254.0.0	192.0.1.0 ~ 223.255.254.0
Hostid	x.0.0.1~x.255.255.254	x.x.0.0 ~ x.x.255.254	x.x.x.x1 ~ x.x.x.254
จำนวน Host	2^{24} - 2	2^{16} - 2	2^{8} - 2

ตารางที่ 3 แสดง IP แอดเดรสและจำนวน Host ที่ใช้งานได้

■ **Option**

ในส่วนนี้อาจจะมีหรือไม่มี หรือมีขนาดเท่าใดขึ้นอยู่กับชนิดของ IP แพคเก็ต

■ **Padding**

มีขนาด 0 - 3 ไบต์ ใช้เป็นส่วนที่ทำให้ขนาดของ Option เป็นจำนวนเท่าของ 32 บิต

■ **Data**

เป็นส่วนของข้อมูลหรือโปรโตคอลที่อยู่ในชั้นที่สูงกว่าหรือเท่ากับIP แพคเก็ต

Routing Datagrams

แต่ละ Datagram จะมี IP Address ของปลายทางที่จะไป ถ้าหากว่า IP นั้นอยู่ใน Network เดียวกัน Internet Protocol ก็จะทำการส่งมันไปยังเครื่องปลายทางทันที แต่ถ้าหากว่าปลายทางนั้นอยู่คนละ Network ก็จะทำการส่งผ่าน Gateway ของ Network นั้นโดย Gateway จะทำหน้าที่ส่งต่อ Datagram ไปยัง Networkอื่น โดยพิจารณาจาก IP Address ของปลายทางที่ต้องการจะไป การเดินทางของ Datagramไปยังจุดหมายนั้นอาจต้องผ่าน Gateway หลายตัว การเดินทางจึงมีลักษณะเหมือนการกระโดดจากที่หนึ่งไปยังอีกที่หนึ่ง จนกระทั่งถึงที่หมายการกระโดดแต่ละครั้งเราเรียกว่า HOP

ในการเดินทางของ Datagrams อาจที่จะต้องผ่าน Network หลาย ๆ ชนิด เช่น ผ่าน Ethernet LAN, Leased Line, Fiber Optics หรืออาจจะต้องผ่านสื่อสัญญาณดาวเทียม ซึ่งแต่ละช่วงจะใช้

โปรโตคอล ใน Network Access Layer ไม่เหมือนกัน เช่น Ethernet ใช้ CSMA/CD, Leased-Line อาจใช้ Frame Relay, Fiber Optics อาจใช้ FDDI เป็นต้น แต่ละโปรโตคอลมักมีขนาดของ Unit Data หรือที่เรียกว่า Maximum Transmission Unit (MTU) ที่แตกต่างกัน ทำให้ต้องมีการแบ่ง Datagram ให้มีขนาดเล็กลงเพื่อให้สามารถเดินทางไปได้ เมื่อทำการแบ่ง Datagram แล้ว Word ที่ 2 ใน Header จะเป็นต้นบอกรายละเอียดของการแบ่ง เพื่อให้ปลายทางสามารถนำมันมาประกอบกันใหม่ได้อย่างถูกต้อง

เมื่อ Internet Protocol ในเครื่องปลายทางรับ Datagram เข้ามาแล้วมันจะต้องทำการส่งส่วนของ Data ขึ้นไปยัง Transport Layer ซึ่งใน Layer นี้จะมี โปรโตคอล หลายตัว การที่มันจะรู้ได้ว่าต้องส่งไปให้โปรโตคอลใด จะพิจารณาจากค่าใน Word ที่ 3 ซึ่งจะเก็บ Protocol Number ไว้

2.2.2.2 Internet Control Message Protocol (ICMP)

ทำหน้าที่ในการส่งสัญญาณควบคุมต่าง ๆ ไปยังเครื่องปลายทาง โดย โปรโตคอลจะอาศัย IP Datagram ในการส่งสัญญาณควบคุมเหล่านั้น คำสั่งควบคุมที่สำคัญของ ICMP ได้แก่

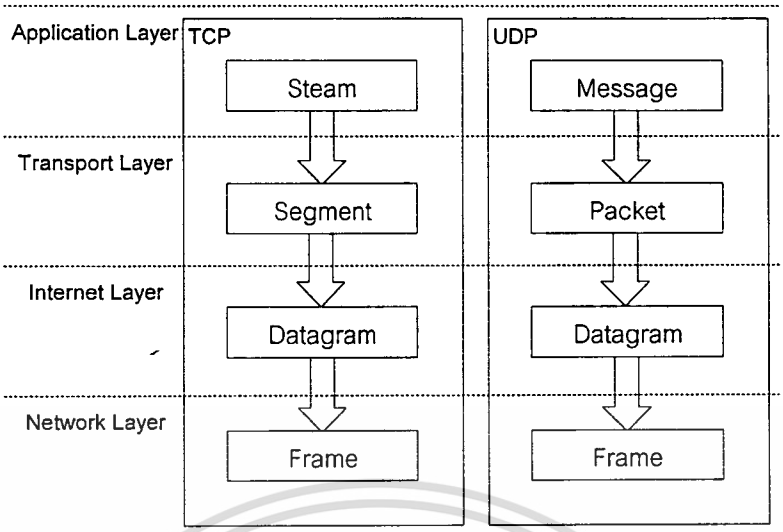
- Flow Control
- Detecting Unreachable Destinations
- Redirecting Routes
- Checking Remote Hosts

2.2.3 Transport Layer

ในชั้นนี้มีโปรโตคอลที่สำคัญอยู่ 2 ชนิด คือ

- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)

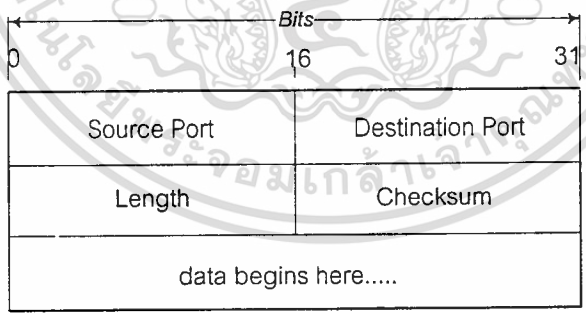
ทั้ง 2 โปรโตคอลทำหน้าที่เหมือนกัน คือ เป็นตัวเชื่อมต่อการส่งข้อมูลระหว่าง Application Layer และ Internet Layer



รูปที่ 7 แสดงรูปแบบ Data Structure ที่แตกต่างกันของ TCP และ UDP

2.2.3.1 User Datagrams Protocol (UDP)

เป็นโปรโตคอลแบบ Connectionless คือ ไม่มีการ Handshake กับเครื่องปลายทาง ไม่มีการตรวจสอบความถูกต้องของข้อมูลที่รับมา ทำให้มีความเร็วในการทำงานสูง เหมาะที่จะใช้กับงานที่ส่งข้อมูลขนาดเล็กและส่งบ่อย ๆ โดยมีรายละเอียดดังแสดงในรูปที่ 8



รูปที่ 8 แสดงรูปแบบของ UDP Datagram

- Source & Destination Port

มีขนาด 32 บิต โดยแบ่งออกเป็น Source Port 16 บิตและ Destination Port 16 บิต แสดงถึงการเข้าถึงแอพพลิเคชั่นของชุดโปรโตคอลโดยให้หมายเลขพอร์ตเป็นตัวกำหนด ดังนั้น Source Port หมายถึงหมายเลขพอร์ตของสถานีต้นทาง แสดงดังตารางที่ 4

Decimal	Keyword	UNIX Keyword	Description
0			Reverved
7	ECHO	Echo	Echo
9	DISCARD	Discard	Discard
11	USERS	Svstat	Active Users
13	DAYTIME	Daytime	Daytime
15	-	Netstat	Who is up or NETSTAT
17	QUOTE	Qotd	Quote of the Day
19	CHARGEN	Chargen	Character Generator
37	TIME	Time	Time
42	NAMESERV	Name	Host Name Server
43	NICKNAME	Whois	Who is
53	DOMAIN	Nameserver	Domain Name Server
67	BOOTPS	Bootps	Bootstrap Protocol Server
68	BOOTPC	Bootpc	Bootstrap Protocol Client
69	TFTP	Tftp	Trivial File Transfer
111	SUNRPC	Sunrpc	SUN Remote Procedure Call
123	NTP	Ntp	Network Time Protocol
161	-	Snmp	SNMP Net Monitor
162	-	Snmp-trap	SNMP Traps
512	-	Biff	UNIX Comsat
513	-	Who	Unix Rwho Daemon
514	-	Svslog	System Log
525	-	Timed	Time Daemon

ตารางที่ 4 แสดงถึงความหมายของ UDP พอร์ตต่าง ๆ

- Length

มีขนาด 16 บิต ใช้บอกความยาวของ UDP เซกเมนต์ นับเป็นจำนวนไบต์

- Checksum

มีขนาด 16 บิต ใช้ในการตรวจสอบความผิดพลาดโดยมีลักษณะการทำงานเช่นเดียวกับ IP

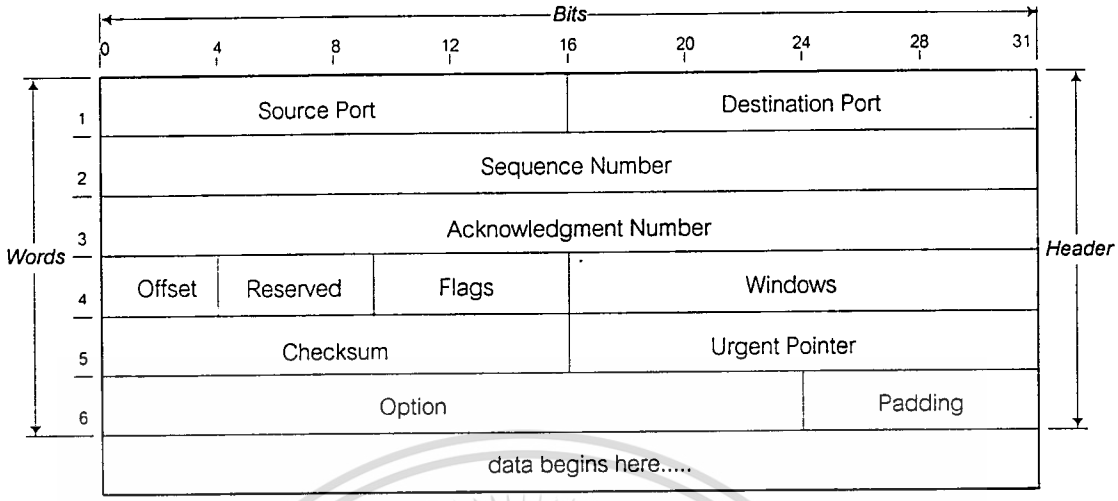
2.2.3.2 Transmission Control Protocol (TCP)

เป็นโปรโตคอลแบบ Connection-Oriented และมีการตรวจสอบความถูกต้องของข้อมูลที่ได้รับด้วย ทำให้การส่งเป็นไปอย่างน่าเชื่อถือ การส่งข้อมูลของ TCP จะถูกส่งทีละ Segment โดยเครื่องปลายทางจะส่งสัญญาณตอบรับกลับมายังเครื่องต้นทางสำหรับทุก ๆ Segment ที่มันได้รับและตรวจสอบแล้วไม่พบข้อผิดพลาด ถ้าหากว่าเครื่องต้นทางไม่ได้รับสัญญาณตอบรับกลับมา ก็จะสันนิษฐานว่า Segment เกิดปัญหาและทำการส่ง Segment นั้นไปอีกครั้ง วิธีการนี้เราเรียกว่า Positive Acknowledgment with Re-transmission (PAR) สำหรับข้อแตกต่างระหว่างการให้บริการแบบ Connection-Oriented และ Connectionless แสดงดังตารางที่ 5

ประเด็น	Connection-Oriented	Connectionless
ต้องเริ่มตั้งค่าก่อน	ต้องการ	ไม่ต้องการ
แอดเดรสปลายทาง	ต้องการขณะเริ่มต้นเท่านั้น	บนทุกแพคเก็ต
เรียงลำดับแพคเก็ต	รับประกัน	ไม่มี
ควบคุมความผิดพลาด	กระทำที่ Network Layer	กระทำที่ Transport Layer
ควบคุมข้อมูลไหล	กระทำโดย Network Layer	Network Layer ไม่กระทำให้
มีทางเลือกต่อรองได้	มี	ไม่มี
ต้องการตัวระบุคอนเนคชั่น	มี	ไม่มี

ตารางที่ 5 แสดงถึงความแตกต่างระหว่างการให้บริการแบบ Connection-Oriented และ Connectionless

การที่บอกว่า TCP เป็นการส่งข้อมูลแบบ Connection-Oriented นั้นแสดงว่ามันจะต้องมีการส่งสัญญาณ Handshake ระหว่างเครื่องต้นทางและปลายทางเพื่อให้แน่ใจว่าสามารถติดต่อถึงกันได้แล้วเริ่มส่งข้อมูล วิธีการ Handshake ที่ TCP ใช้นั้นเราเรียกว่า Three-Way Handshake โดยเมื่อเครื่องต้นทางต้องการส่งข้อมูล มันจะส่งสัญญาณ Synchronize (SYN) ไปยังเครื่องปลายทาง เมื่อเครื่องปลายทางได้รับสัญญาณ SYN แล้วมันก็จะส่งสัญญาณ ACK กลับมา เมื่อสัญญาณ ACK วิ่งกลับไปยังเครื่องต้นทาง ก็ถือว่าเส้นทางเชื่อมต่อได้เกิดขึ้นแล้ว เครื่องต้นทางจะส่ง ACK กลับไปให้เครื่องปลายทางพร้อมกับเริ่มต้นการส่งข้อมูล



รูปที่ 9 แสดงรูปแบบของ TCP Datagram

สำหรับรายละเอียดฟิลด์ต่าง ๆ ของเซกเมนต์ TCP แสดงในรูปที่ 9 สามารถอธิบายได้ดังนี้

- Source Port

มีขนาด 16 บิต แสดงถึงการเข้าถึงแอพพลิเคชันโดยใช้หมายเลขพอร์ตเป็นตัวกำหนดหรืออาจจะกล่าวได้ว่า Source Port หมายถึงหมายเลขพอร์ตของสถานีต้นทาง

- Destination Port

มีขนาด 16 บิต แสดงถึงหมายเลขพอร์ตของสถานีปลายทาง ที่ต้องการทำการติดต่อด้วยแอพพลิเคชันมาตรฐานโดยที่หมายเลขพอร์ตคงที่ เช่น Telnet ใช้พอร์ตหมายเลข 23, File Transfer Protocol (FTP) ใช้หมายเลขพอร์ต 20, 21 เป็นต้น โดยตารางที่ 5 แสดงถึงบางส่วนของหมายเลขพอร์ตที่มีการใช้งานสำหรับพอร์ตมาตรฐานที่ถูกกำหนดไว้ที่หมายเลข 0-255 ในส่วนที่นอกเหนือจากนี้ผู้ใช้งานแอพพลิเคชันสามารถกำหนดเองได้

- Sequence Number

มีขนาด 32 บิต TCP เซกเมนต์ใช้หมายเลขลำดับนี้ ในการบอกถึงตำแหน่งไบต์แรกของข้อมูลที่ปรากฏอยู่ในส่วนของเขตข้อมูล

- Acknowledgement Number

มีขนาด 32 บิต แสดงถึงไบต์ที่คาดว่าจะได้รับถัดไป

- HLEN (Header Length)

มีขนาด 32 บิต โดยประกอบด้วย Interger ที่ระบุความยาวของ Segment Header ซึ่งจะเขียนอยู่ในรูปเท่าของ 32 บิต ซึ่งมันมีความจำเป็นเพราะในฟิลด์ Option จะมีความยาวไม่แน่นอน ซึ่งขนาดของ TCP Header จึงมีค่าเปลี่ยนแปลงได้ตามขนาดของฟิลด์ Option

- Reserved

ฟิลด์นี้ถูกสงวนไว้สำหรับอนาคตโดยจะถูกตั้งค่าเป็นศูนย์

- Code Bits

จะถูกใช้โดยซอฟต์แวร์ TCP เพื่อกำหนดวัตถุประสงค์และสิ่งที่อยู่ในเซกเมนต์ โดยมีขนาดทั้งหมด 6 บิต มีความหมายดังตารางที่ 6

Decimal	Keyword	UNIX Keyword	Description
0			Reserved
1	TCPMUX	-	TCP Multiplexor
5	RJE	-	Remote Job Entry
7	ECHO	echo	Echo
9	DISCARD	discard	Discard
11	USERS	svstat	Active User
13	DAYTIME	daytime	Daytime
15	-	netstat	Network Status Program
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chargen	Character Generator
20	FTP-DATA	ftp-data	File Transfer Protocol (data)
21	FTP	ftp	File Transfer Protocol
23	TELNET	telent	Terminal Connection
25	SMTP	smtp	Simple Mail Transfer
37	TIME	time	Time
42	NAMESERVER	name	Host Name Server
43	NICNAME	whois	Who Is
53	DOMAIN	nameserver	Domain Name Server
77	-	rie	Any Private RJE Service
79	FINGER	finger	Finger
93	DCP	-	Device Control Protocol
95	SUPDUP	supdup	SUPDUP Protocol
101	HOSTNAME	hostmanes	NIC Host Name Server
102	ISO-TSAP	iso-tsap	ISO-TSAP
103	X400	x400	X.400 Mail Service
104	X400-SND	x400-snd	X.400 Mail Sending
111	SUNRPC	sunrpc	SUN Remote Procedure Call
113	AUTH	auth	Authentication Server
117	UUCP-PATH	uucp-path	UUCP Path Service
119	NNTP	nntp	USENET News Transfer
129	PWDGEN	-	Protocol
139	NETBIOS-SSN	-	Password Generator Protocol
160~223	RESERVEED		NETBIOS Session Service

ตารางที่ 6 แสดงถึงความหมายของ TCP พอร์ตต่าง ๆ

- Checksum

มีขนาด 16 บิต ใช้วิธีการเดียวกันกับ IP แพกเก็ต

- Windows

มีขนาด 32 บิต จะเป็นการใช้งานโดยซอฟต์แวร์ว่าขณะเวลานั้นสามารถจะรับข้อมูลได้จำนวนเท่าไรซึ่งสามารถกำหนดขนาดบัพเฟอร์ในฟิลด์นี้

- Urgent Pointer

มีขนาด 16 บิต เปรียบเสมือนเป็นค่า Offset ของ Sequence Number โดยจะบอกถึงตำแหน่งของไบต์สุดท้ายภายใน TCP เซกเมนต์ที่เป็นข้อมูลเร่งด่วน (ข้อมูลภายใน TCP เซกเมนต์หนึ่ง ๆ อาจจะไม่มีความเร่งด่วนทั้งหมดก็ได้) ที่ต้องการให้ แอพพลิเคชันพิจารณาทันที อนึ่งคำที่ถูกบรรจุในเขตนี้จะมีความหมายก็ต่อเมื่อเขตย่อย Urgent ถูกกำหนดค่าเป็นหนึ่ง

- Option

ส่วนของ Option นี้อาจจะมีหรือไม่มี หรือมีขนาดเท่าใดขึ้นอยู่กับชนิดของ TCP Segment

- Padding

มีขนาด 0 ~ 3 ไบต์ ใช้เป็นส่วนที่ทำให้ขนาดของ Option เป็นจำนวนเท่าของ 32 บิต

Bit (Left to Right)	Meaning if bit set to 1
URG	Urgent pointer field is valid
ACK	Acknowledgement field is valid
PSH	This segment requests a push
RST	Reset the connection
SYN	Synchronize sequences number
FIN	Sender has reached end of it byte stream

ตารางที่ 7 แสดงถึงบิตของ Code Field ใน TCP Header

2.2.4 Application Layer

เป็น Layer บนสุด ซึ่งประกอบไปด้วยหลาย โปรโตคอล (หรืออาจจะเรียกว่าโปรแกรมมากกว่า) ได้แก่

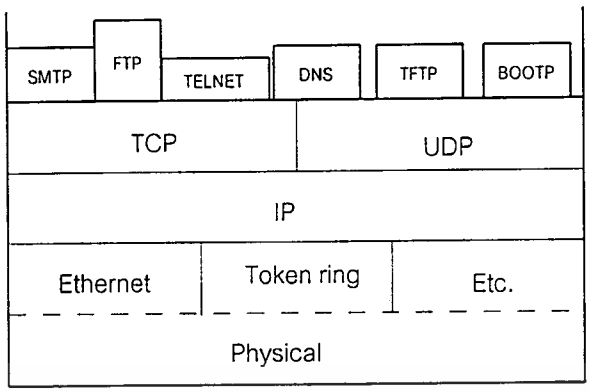
- Network Terminal Protocol (Telnet)
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)

ทั้ง 3 โปรโตคอลทำงานโดยใช้ TCP ใน Transport Layer ส่วน โปรโตคอล ที่ใช้ UDP ได้แก่

- Domain Name Service (DNS)
- Routing Information Protocol (RIP)
- Network File System (NFS)

2.3 ความสัมพันธ์และการทำงานร่วมกันของโปรโตคอลแต่ละชั้น

ตัวอย่างลำดับชั้นของชุดโปรโตคอล TCP/IP และแอพพลิเคชัน แสดงในรูปที่ 9 ซึ่งจากรูปจะเห็นได้ว่าแอพพลิเคชันแต่ละประเภทเลือกใช้โปรโตคอลชั้นที่ 4 แตกต่างกัน เช่น SMTP เลือกใช้ TCP ในขณะที่ BOOTP เลือกใช้ UDP อย่างไรก็ตาม มีแอพพลิเคชั่นบางประเภทเลือกใช้ทั้ง TCP และ UDP สำหรับ TCP และ UDP เลือกใช้โปรโตคอล IP และถัดจากโปรโตคอลชั้นที่ 3 ลงมาก็ขึ้นอยู่กับว่าผู้ใช้งานเลือกใช้เครือข่ายประเภทใดในการส่งผ่านชุดโปรโตคอล TCP/IP ถ้าเป็นเครือข่ายท้องถิ่น (LAN) ก็อาจจะเลือกใช้อีเธอร์เนตหรือ LAN แบบอื่น ๆ ที่เห็นว่าสมควร แต่ถ้าเป็นเครือข่ายชนิดกว้าง (WAN) ก็อาจจะใช้ PPP หรือ HDLC เป็นต้น สำหรับชั้น Physical นั้นขึ้นอยู่กับว่าเลือกใช้เครือข่ายชนิดใด เพราะในแต่ละเครือข่ายจะมีอุปกรณ์เชื่อมต่อแตกต่างกัน เช่น RS232C, CCITT V.35 เป็นต้น



รูปที่ 10 แสดงลำดับชั้นของแอพพลิเคชันและชุดโปรโตคอล TCP/IP เมื่อเปรียบเทียบกับ OSI Layer

ในการส่งข้อมูลของ Application ต่าง ๆ ที่ทำงานอยู่ไปยังเครื่องปลายทางนั้นมันอาศัย TCP หรือไม่ก็ UDP ในการส่งข้อมูลลงไปให้ IP เพื่อทำการส่งส่งไป นั่นคือข้อมูลของทุก ๆ โปรแกรมจะต้องส่งผ่าน IP ทั้งนั้น เราเรียกลักษณะเช่นนี้ว่าการ Multiplexing ส่วนกระบวนการย้อนกลับนั้นเราเรียกว่า Demultiplexing คือ ข้อมูลที่รับเข้ามาจะเข้าไปยัง IP แล้ว IP ก็จะต้องแจกจ่ายข้อมูลเหล่านั้นขึ้นไปยัง Application ที่เป็นเจ้าของให้ถูกต้อง

ข้อมูลที่มาถึงจะต้องผ่าน Layer ชั้นล่างขึ้นไปถึงชั้นบน ใน Internet Layer นั้น Internet Protocol จะใช้หมายเลข Protocol ในการส่งข้อมูลขึ้นไปยัง Transport Layer และในทำนองเดียวกัน Transport Layer ก็จะใช้หมายเลข Port ในการส่งข้อมูลขึ้นไปยัง Application Layer ในระบบ UNIX หมายเลข Protocol และ Port นี้จะถูกกำหนดไว้ในไฟล์ /etc/protocols และ /etc/services ตามลำดับ Port และ Protocol ที่ใช้กันบ่อยมักถูกกำหนดหมายเลขให้เป็นค่าที่แน่นอนตายตัวสำหรับทุก ๆ เครื่องใน Internet เช่น Protocol TCP และ UDP มีหมายเลขเป็น 6 และ 17 (ดูค่าในตารางที่ 2) และหมายเลข Port ของโปรแกรม Telnet และ FTP จะเท่ากับ 23 และ 21 ตามลำดับ (ดูค่าในตารางที่ 6) เป็นต้น

จากหลักการใช้ IP Address, Protocol และ Port Number ข้างต้นจะเห็นได้ว่าเราต้องการให้ทุก ๆ Datagram ที่ส่งไปมาใน Internet นั้นมีปลายทางที่ Unique คือไม่ซ้ำกัน แต่ละ Datagram จะมีปลายทางชี้ไปที่ Application ที่แน่นอนในเครื่องปลายทาง แต่วิธีการข้างต้นนี้ยังไม่เพียงพอ เนื่องจากเครื่องคอมพิวเตอร์เครื่องหนึ่งสามารถที่จะ Run โปรแกรมเดียวกันพร้อม ๆ กันได้ เช่น ในเครื่อง UNIX Server ตัวหนึ่งอาจมีผู้ใช้หลายคนกำลังใช้โปรแกรม Talk อยู่ ในกรณีอย่างนี้เราจะทราบได้อย่างไรว่า Datagram ที่เข้ามาเป็นของโปรแกรม Talk ตัวใด เพราะหมายเลข IP Address,

Protocol และ Port เหมือนกันหมด ปัญหานี้สามารถแก้ไขได้โดยใช้วิธีที่เรียกว่า Dynamically Allocated Port คือระบบจะไม่กำหนดเลข Port ไว้ตายตัว แต่จะทำการกำหนดเลข Port ให้กับ Process ในขณะที่มันเริ่มต้นทำงาน ซึ่งเลขนี้จะต้องไม่ซ้ำกับเลข Port ของ Process อื่นยกตัวอย่าง เช่น ถ้าเครื่องปลายทางเครื่องหนึ่งมีผู้ใช้ A และ B กำลังใช้โปรแกรม Telnet อยู่ เครื่องต้นทาง (ซึ่งไม่จำเป็นจะต้องเป็นเครื่องเดียวกัน) จะกำหนดหมายเลข Port ต้นทางให้เป็นค่าสุ่มค่าหนึ่งและเลข Port ปลายทางให้เป็นหมายเลข Port มาตราฐาน ในตัวอย่างนี้ Telnet จะเท่ากับ 23 ด้วยวิธีการนี้ Datagram ของ A และ B จะมีเลขคู่ที่แตกต่างกัน เช่น ของ A อาจเป็น 3044,23 ส่วน B อาจจะเป็น 1027,23 ทำให้ Datagram สามารถเดินทางไปยัง Application ที่ถูกต้องได้

บทที่ 3
การสื่อสารชนิด TCP

3.1 การเชื่อมต่อแบบ TCP (Establishing TCP Connection)

การเริ่มต้นการเชื่อมต่อของโปรโตคอล TCP จะกระทำชนิด Three-Way Handshake ส่วนเซกเมนต์แรกของ Handshake สามารถกำหนดได้โดย SYN Bit ที่อยู่ใน Code Field Message ที่ 2 จะมีทั้งบิต SYN และบิต ACK ซึ่งถูกกำหนดให้เป็นการ Acknowledges ของ SYN Segment แรก และ Handshake Message สุดท้ายจะมีเพียง Acknowledgement เพื่อเป็นการแจ้งปลายทางซึ่งทั้งสองด้านจะเห็นพ้องกันถึงการเชื่อมต่อได้เกิดขึ้นแล้ว และซอฟต์แวร์ TCP บนเครื่องใดเครื่องหนึ่งจะคอยรับสัญญาณ Handshake จากซอฟต์แวร์ TCP เครื่องอื่น ๆ อย่างไรก็ตามสัญญาณ Handshake จะถูกกำหนดอย่างรอบคอบ หากพบว่าเครื่องสองเครื่องสร้างการเชื่อมต่อกันดังนั้นการเชื่อมต่อจะถูกสร้างจากด้านใดด้านหนึ่งทำให้ข้อมูลสามารถส่งได้ทั้งสองทิศทาง

เพื่อที่จะอธิบายลักษณะการทำงานของ TCP เราจะพิจารณาเฟสของการเชื่อมต่อของ TCPระหว่างสองโหนดคือ โหนดแอดเดรส 128.1.0.1 และโหนดแอดเดรส 128.1.0.9 โดยที่โหนด 128.1.0.1 จะเริ่มทำการติดต่อก่อน โดยแสดงในรูปที่ 11 สามารถอธิบายได้ว่า

1. โหนด 128.1.0.1 ตัว TCP จะส่งชุดเซกเมนต์ของการเริ่มต้นสื่อสารซึ่งประกอบไปด้วยบิต SYN และ ACK โดยที่ค่าในบิต SYN จะมีค่าเป็น Sequence Number, SEQ (ในรูปคือค่า 921) สำหรับค่าของบิต ACK ยังไม่ได้ถูกกำหนด ในขั้นตอนนี้ตัวเซ็กเมนต์จะผ่านเข้าไปยัง IP Layer ดังนั้นถ้า IP Layer ยังไม่เคยสื่อสารกับ IP Address 128.1.0.9 กล่าวคือไม่มีค่า MAC (Meduim Access Control Address) ที่สัมพันธ์กับแอดเดรสดังกล่าวอยู่ในหน่วยความจำ ARP (ARP Cache, Address Resolution Protocol) ตัว ARP ก็จะทำการส่งดาต้าแกรม ARP Request ไปยังการ์ดเครือข่าย (Lan Card, Network Interface Card) ทุก ๆ การ์ดผ่านวิธี MAC Broadcast Address (0xFFFF_FFFF_FFFF) โดยที่ ARP ใช้อีเทอร์เนต (Ethernet) ชนิด 0x0806 สำหรับการร้องขอครั้งนี้ เมื่อมีการตอบกลับคืนมาจากการ์ดเครือข่าย ค่า MAC ที่สัมพันธ์กันกับ IP Address ตัว ARP ก็จะทำการเก็บค่านี้ไว้ในหน่วยความจำ (ARP Cache) เพื่อไว้ใช้ต่อไป

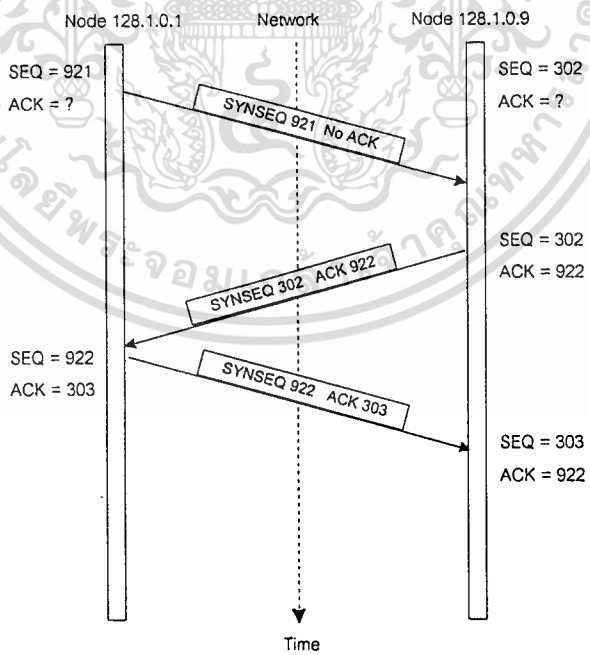
2. โหนด 128.1.0.9 จะทำการตอบกลับคืนมาด้วยเซกเมนต์ที่คล้ายคลึงเมื่อได้รับเซกเมนต์จาก โหนด 128.1.0.1 คือจะตอบกลับด้วยบิต SYN และบิต ACK โดยในบิต ACK จะมีค่ามากกว่า

ค่า SEQ ที่ถูกส่งมาจากโหนดต้นทางอยู่หนึ่ง ดังนั้นค่า ACK ที่ตอบกลับจากโหนด 128.1.0.9 จะมีค่าเท่ากับ 922

3. เมื่อโหนด 128.1.0.1 ได้รับการตอบสนองคืนจากโหนด 128.1.0.9 ก็จะทำการส่งเซกเมนต์ในครั้งที่สองโดยทำการส่งค่า ACK และค่า SEQ กลับคืน

เมื่อสิ้นสุดเหตุการณ์ทั้งสามข้อแล้วแสดงว่าการเชื่อมต่อได้เกิดขึ้นแล้วอย่างสมบูรณ์เพื่อเตรียมพร้อมสำหรับส่งข้อมูลโดยทำการส่งเซกเมนต์ Keep-alive เพื่อทำการเช็กว่ายังมีการเชื่อมต่ออยู่เป็นปกติอยู่ยังไม่มีการขาดหรือสิ้นสุดการสื่อสารเพียงแค่เป็นการรอข้อมูลอยู่เท่านั้น

3-Way Handshake จำเป็นและเพียงพอสำหรับการ Synchronization ระหว่างการเชื่อมต่อของทั้งสองด้าน เพื่อที่จะเข้าใจว่าทำไม เราต้องทำความเข้าใจว่า TCP นั้น สร้างขึ้นบน Unreliable Packet Delivery Service ดังนั้นแมสเซสมีโอกาสที่จะหายไปได้หรือถูกหน่วยเวลาหรือเกิดซ้ำหรือเกิดการไม่เรียงลำดับ ดังนั้นโปรโตคอลมักจะใช้กลไกเวลาและการร้องขอการส่งซ้ำ ปัญหาก็เกิดขึ้นถ้าการส่งซ้ำและการเริ่มต้นร้องขอมาถึงขณะที่การเชื่อมโยงกำลังถูกสถาปนาขึ้นหรือการร้องขอถูกทำให้ช้าลงกระทั่งหลังจากการเชื่อมต่อได้ถูกสร้างขึ้นแล้วหรือจบ

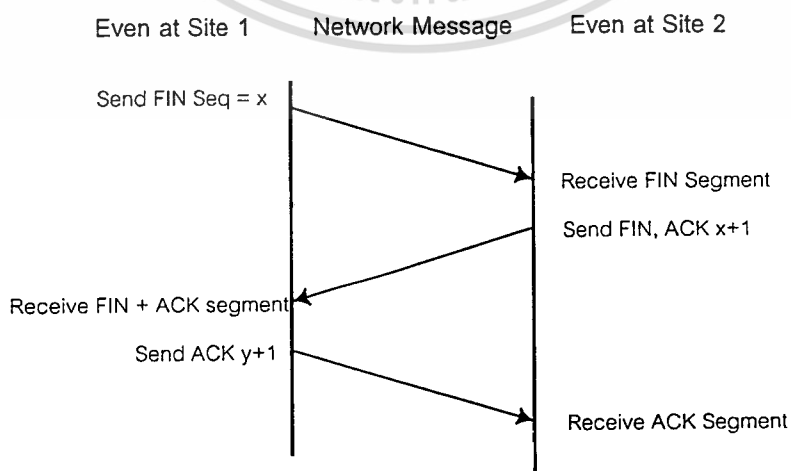


รูปที่ 11 แสดงถึงการเริ่มการเชื่อมต่อของ TCP

การสื่อสารแบบ 3-Way Handshake จะรับประกันการส่งทั้งสองด้านและจะอนุญาตให้ทั้งสองด้านเริ่มต้น Sequence Number โดยที่ Sequence Number จะถูกส่งระหว่าง Handshake เครื่องแต่ละเครื่องการเริ่มต้น Sequence Number แบบสุ่มจะใช้สำหรับกำหนดไบต์ของชุดที่จะส่ง Sequence Number ที่ไม่สามารถเริ่มต้นด้วยค่าที่เหมือนกัน แน่นอนว่ามันเป็นสิ่งสำคัญทั้งสองด้านที่จะเห็นพ้องต้องกันกับจำนวนเริ่มต้น ดังนั้นจำนวนไบต์ที่ใช้ใน Acknowledgements ซึ่งถูกใช้ใน Data Segments

3.2 การยกเลิกการติดต่อของ TCP (Closing TCP Connection)

ในขั้นตอนสิ้นสุดการสื่อสารเกิดขึ้นเมื่อฝั่งใดเสร็จสิ้นการส่งข้อมูล TCP สามารถยุติการติดต่อสื่อสารได้นั้นโดยใช้การดัดแปลง Three-way Handshake สำหรับการยุติการติดต่อ TCP จะเป็นการสื่อสารแบบ Full Duplex ซึ่งสามารถทำการส่งข้อมูลได้ทั้ง 2 ด้านเป็นอิสระต่อกัน เมื่อโปรแกรมประยุกต์บอก TCP ว่าไม่มีข้อมูลที่จะส่งแล้ว TCP จะปิดการติดต่อในด้านในด้านหนึ่งก่อน โดยจะส่งค่า Finishing Transmitting (FIN) ไปและรอรับ Acknowledge และส่งค่า Segment พร้อมทั้งตั้งค่าบิต FIN ทางด้านรับจะรับ TCP Acknowledge ที่มีบิต FIN ถูกตั้งค่า ระหว่างที่จะยุติการเชื่อมต่อ TCP จะไม่ยอมรับข้อมูลอื่นอีก หลังจากที่ด้านปลายทางที่ได้รับ FIN Segment เริ่มต้นมันก็จะสร้าง FIN Segment ที่สองขึ้นมาทันที TCP จะส่ง Acknowledge ไปแจ้งทางด้านแรกจากนั้นก็จะส่ง Acknowledge ของ FIN Segment แรกและส่ง FIN Segment ที่สองไป ด้านเริ่มต้นและด้านที่เริ่มต้นก็จะตอบรับด้วยการส่ง ACK ของการได้รับ FIN Segment ที่สองก็ถือเป็นการยุติการเชื่อมโยง



รูปที่ 12 แสดงถึงการยุติการเชื่อมต่อของ TCP

3.3 TCP Connection Reset

ปกติโปรแกรมประยุกต์จะใช้ Close Operation สำหรับการยุติการเชื่อมต่อเมื่อใช้งานเสร็จ แต่ในบางกรณีก็ต้องมีการจบการใช้งานอย่างไม่ปกติซึ่ง TCP ก็เตรียม การ Reset ไว้เพื่ออำนวยความสะดวก สำหรับการสิ้นสุดการเชื่อมต่อแบบไม่ปกติ Reset Connection จะเริ่มต้นด้วยด้านใดด้านหนึ่ง โดยจะส่ง Segment ที่มีบิต RST ที่ถูกกำหนดค่าไว้ในฟิลด์ Code อีกด้านหนึ่งก็จะตอบรับทันทีด้วยที่มีการขอยกเลิกการเชื่อมต่อทันที

3.4 Flow Control

กล่าวถึงกลไก 2 ชนิดในการทำงานของ TCP คือ Windows Size และ Acknowledgement (ACK)

1. **กลไกการทำงาน Acknowledgement (ACK)**

   กลไก ACK เป็นหัวใจในการทำงานของ TCP กล่าวคือ เมื่อข้อมูลมาถึงผู้รับตัวโปรโตคอลต้องทำการส่ง Acknowledge กลับคืนไปสู่ผู้ส่ง โดยกำหนดว่าไบต์ของข้อมูลเป็นตัวเลขเรียงลำดับ (Sequence Number) ดังนั้นผู้รับจะสามารถรับรู้ข้อมูล Acknowledge โดยไบต์ที่มีค่ามากที่สุดที่ได้รับแล้วทำการทยอยส่ง Acknowledge ไบต์ก่อน ๆ โดยใน โปรโตคอล นี้จะสามารถแค่ยืนยันว่าข้อมูลได้ถูกรับแล้ว แต่ไม่ได้ชี้ชัดว่าเร็วแค่ไหนที่ Acknowledge ต้องถูกส่งกลับคืน หรือขนาด Acknowledge ที่ถูกส่งกลับในแต่ละ Acknowledge ที่แยกกัน

2. **กลไกหน้าต่าง Windows Size**

   กลไกหน้าต่าง คือเครื่องมือที่ใช้ในการควบคุมการไหลของข้อมูล กล่าวคือเมื่อข้อมูลของผู้รับได้ถูกส่งคืนกลับมายังผู้ส่งตัว ขนาดข้อมูลที่คืนกลับมาก็คือขนาดของบัพเฟอร์ที่ผู้รับต้องเผื่อไว้สำหรับข้อมูลที่เพิ่มเติมหรือเพิ่มขึ้นขนาดไบต์ ของข้อมูลดังกล่าวเราเรียกว่าหน้าต่าง ซึ่งมีค่าสูงสุด ณ จุดที่ผู้ส่งสามารถส่งจนกระทั่งผู้รับได้ส่งค่าหน้าต่างคืนกลับมา ในบางกรณีผู้รับอาจจะไม่มีขนาดของบัพเฟอร์ที่เพียงพอ ก็จะทำการส่งค่าหน้าต่างกลับมาเป็นค่าศูนย์ ภายใต้สภาวะดังกล่าว โปรโตคอลจะต้องกำหนดให้ผู้ส่งส่งข้อมูลให้มีขนาดเล็กลง แต่ถ้าหากว่ายังมีค่ากลับมายังผู้ส่งเป็นค่าศูนย์อยู่ โปรโตคอลจะสรุปว่าการส่งข้อมูลล้มเหลวจะทำการปิดการติดต่อ

กล่าวคือ ก่อนที่จะสามารถทำการส่งเซกเมนต์ออกไปได้ จะต้องมีการรอบิต ACK ก่อน จึงมีการแบ่งหรือย่อยขนาดปริมาณความจุของสื่อ (Media) หรือลิงก์ (Link) เพื่อเพิ่มค่าของ Round Trip Delay ในการใช้วิธีเช่นนี้เรียกว่า Sliding Windows โดยวิธีการปฏิบัติด้วยวิธีนี้จะมองว่าแต่ละเซกเมนต์เป็นอิสระไม่ขึ้นต่อกัน จึงไม่มีความจำเป็นที่จะต้องทำการรอบิต ACK เรียงต่อเนื่องกันก่อนทำการติดต่อกลับเป็นลำดับ ด้วยวิธีการนี้ทำให้ทรูพุตและประสิทธิภาพในการใช้แบนด์วิธ์ดีขึ้น โดยค่าขนาดมาตรฐานที่นิยมใช้จะอยู่ระหว่างค่า 1024 ถึง 4096

3.5 ความสัมพันธ์ของขนาดข้อมูลใน TCP, IP และ MTU

ในหัวข้อนี้จะกล่าวถึงขนาดข้อมูลสูงสุดของ TCP (TCP Maximum Segment Size, MSS) ค่าสูงสุดของดาต้าแกรมใน IP (IP Maximum Datagram Size)และขนาดสูงสุดของข้อมูลใน IP (Maximum Data Datagram Size, MDDS) โดยมีข้อกำหนดว่า โฮสต์จะต้องไม่ส่งดาต้าแกรมที่มีขนาดใหญ่กว่า 576 ไบต์ ดังนั้นค่านี้ก็เป็นค่าที่ฝั่งรับจะต้องกำหนดขนาดของบัฟเฟอร์เตรียมไว้ด้วย ยกเว้นในกรณีที่ฝั่งรับจะทราบก่อนว่าจะมีการส่งผ่านดาต้าแกรมที่มีขนาดเกิน 576 ไบต์มาให้ และข้อกำหนดอีกข้อคือ ขนาดเซ็กเมนต์สูงสุดของ TCP จะเท่ากับขนาดดาต้าแกรมสูงสุดของ IP ลบด้วยค่าสี่สิบ ดังนั้นถ้าค่าตกลงของขนาดดาต้าแกรมสูงสุดของ IP มีขนาด 576 ไบต์ ขนาดเซ็กเมนต์สูงสุดของ TCP คือ 536 ไบต์ โดยที่ค่า MSS จะนับเฉพาะขนาดของข้อมูลเท่านั้นไม่ได้รวมไปถึงขนาดส่วนหัวของ TCP และ IP กล่าวคือไม่ได้นับบิตควบคุม SYN และ FIN ถึงแม้ว่าบิตควบคุมทั้งสองจะเกี่ยวข้องอยู่ในค่า Sequence Number

นอกจากนี้ยังมีค่าอีกค่าหนึ่งที่มีความสัมพันธ์คือ ค่า MTU (Maximum Transmission Unit) หรือขนาดการส่งผ่านข้อมูลหนึ่งหน่วยโดยค่านี้จะเป็นค่าที่มากที่สุดหนึ่งหน่วยต่อการส่งที่สามารถผ่านเข้าไปในเครือข่ายที่ทำการติดต่ออยู่ ได้หรืออาจจะกล่าวได้ว่าคือ จำนวนข้อมูลสูงสุดที่สามารถจะส่งข้อมูลไปได้ในหนึ่งฟิสิคัลเฟรมโดยรวมขนาดเฮดเดอร์และขนาดข้อมูลที่ต้องการส่งค่า MTU นี้จะถูกกำหนดจากผู้ควบคุมเครือข่ายโดยพิจารณาจากชนิดของเครือข่ายที่ใช้อยู่ ดังตัวอย่างในตารางที่ 7 และความสัมพันธ์ระหว่างค่า MTU, MSS และค่า MDS ทั้งสามค่าจะอยู่ในรูปสมการว่า

MDDS = MTU - ขนาดเฮดเดอร์ของ IP

MSS = MTU - ขนาดเฮดเดอร์ของ TCP - ขนาดเฮดเดอร์ของ IP

หรือ = MDDS - ขนาดเฮดเดอร์ของ TCP

ชนิดของเครือข่าย / ชนิดของโปรโตคอล	ขนาดค่า MTU
ARPANET, MILNET	1007
Ethernet (10 Mb)	1500
Proteon PRONET	2046
PPP Default	1134
PPP Low Relay	1144
SLIP	1055

ตารางที่ 8 แสดงถึงค่า MTU ในเครือข่ายที่แตกต่างกัน

และ IP จะทำการเช็คขนาดของข้อมูลที่จะทำการส่งผ่านไปให้ TCP โดยที่ถ้าขนาดของข้อมูลมีค่าน้อยกว่าหรือเท่ากับค่า MDDS ตัว IP ก็จะทำการเพิ่ม IP Header เข้าไปติดกับข้อมูลแล้วทำการส่งผ่านไปให้ยังเลเยอร์ TCP ต่อไป แต่ถ้าหากว่าขนาดของข้อมูลมีค่ามากกว่า MDDS ก็จะทำการแบ่งย่อยขนาดของข้อมูล (Defragment) ก่อนที่จะทำการส่งผ่านไป ดังนั้นจะได้สมการว่า

TCP Maximum Segment Size = IP Maximum Datagram Size - 40

โดยที่ค่า 40 คือ ค่าเฮดเดอร์ที่น้อยที่สุดของ IP และ TCP คือทั้ง IP Header และ TCP Header มีค่าน้อยที่สุดคือ อย่างละ 20 ไบต์เท่ากัน

บทที่ 4

การวิเคราะห์การสื่อสาร TCP และผลการทดลอง

4.1 การวัดประสิทธิภาพของโปรโตคอลเมื่ออัตราการส่งข้อมูลเท่ากับขนาด MTU

ดังที่กล่าวมาแล้วข้างต้น รูปแบบของแพคเก็ตข้อมูลของ TCP นั้นจะมีส่วนเฮดเดอร์ซึ่งรวมกับเฮดเดอร์ของ IP ด้วยแล้วจะมีขนาดน้อยที่สุดที่เป็นไปได้คือ 40 ไบต์ และส่วนของข้อมูลจะมีได้ไม่เกิน 65,495 ไบต์ นั่นคือจะมีทั้งเฮดเดอร์และข้อมูลรวมกันได้ไม่เกิน 65,535 ไบต์ ต่อไปนี้จะแสดงให้เห็นถึงแนวโน้มประสิทธิภาพรูปแบบโครงสร้างกับจำนวนของ Overhead ของการส่งแพคเก็ตข้อมูล โดยกำหนดให้อัตราการส่งข้อมูลมีขนาดเท่ากับ MTU

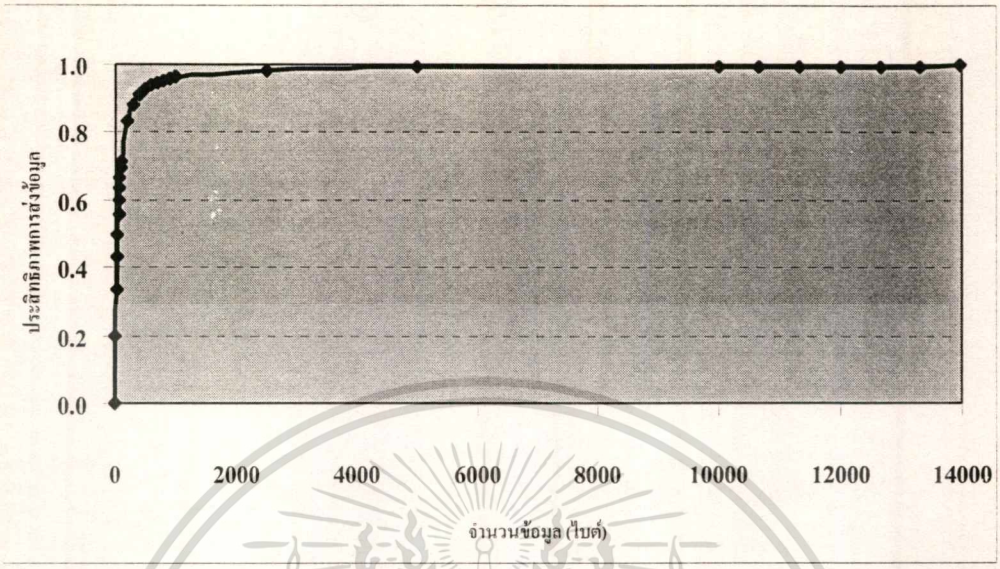
$$\text{TCP Format Utilization} = \frac{\text{MTU}}{(\text{MTU} + \text{Header})} \quad \text{.....(1)}$$

จำนวนข้อมูลที่ทำการส่ง (ไบต์)	TCP Format Utilization
0	0.00000
1	0.02439
2	0.04762
3	0.06977
5	0.11111
10	0.20000
20	0.33333
30	0.42857
40	0.50000
50	0.55556
100	0.71429
150	0.78947
200	0.83333

จำนวนข้อมูลที่ทำการส่ง (ไบต์)	TCP Format Utilization
250	0.86207
300	0.88235
400	0.90909
534	0.93031
1024	0.96241
2048	0.98084
3072	0.98715
4096	0.99033
8192	0.99514
10240	0.99611
20480	0.99805
40960	0.99902
65495	0.99939

ตารางที่ 9 แสดง Format Utilization ของ TCP

จากตารางที่ 9 จะพบว่าค่า Format Utilization ของ TCP นี้จะมีค่าสูงขึ้นในกรณีที่จำนวนที่ต้องการส่งมีค่าเพิ่มขึ้นมาก ซึ่ง ณ ค่าจำนวนข้อมูลสูงสุด 65,495 ไบต์ มีค่า Format Utilization ของ TCP ถึง 0.999 ซึ่งมีค่าใกล้หนึ่งมากทั้งนี้เนื่องจากจำนวนข้อมูลที่ต้องการส่งนั้นมีค่าใกล้เคียงกับจำนวนข้อมูลที่ส่งจริง โดยไม่มีการคิดค่าพารามิเตอร์อื่น ๆ มาเกี่ยวข้อง เช่น การเกิดความผิดพลาดหรืออัตราการหน่วงเวลา หรือเวลาที่เสียไป หรือคำนึงถึงสภาวะของเครือข่ายในขณะนั้น ค่า Format Utilization ของ TCP จากตารางที่ 8 สามารถนำมาแสดงได้ดังกราฟรูปที่ 13



รูปที่ 13 กราฟแสดง Format Utilization ของ TCP ในกรณีที่กำหนดให้จำนวนข้อมูลที่ทำการส่ง เท่ากับอัตราการส่งผ่านข้อมูลสูงสุด(MTU)

4.2 การคำนวณประสิทธิภาพของ TCP ในการส่งข้อมูลปริมาณต่าง ๆ

เนื่องจาก TCP เป็นโปรโตคอลที่มีลักษณะการเชื่อมต่อเป็นแบบ Connection Oriented คือ จะต้องมีการสถาปนาการเชื่อมต่อและเมื่อสิ้นสุดก็จะต้องมีการส่งสัญญาณยุติการเชื่อมต่อก่อนดังที่กล่าวมาแล้วในบทที่ 3 ทำให้เกิดแพคเก็ตข้อมูลเพิ่มขึ้นในแต่ละครั้งของการเชื่อมต่อ 4 แพคเก็ต คือ 2 แพคเก็ตสำหรับการเริ่มต้นและ อีก 2 แพคเก็ตสำหรับการยุติการสื่อสาร ดังนั้นเราจะมาพิจารณาถึงระดับของ MTU ที่ต่างค่ากันนั้น จะมีลักษณะแตกต่างกันอย่างไร โดยคำนวณเป็นเปอร์เซ็นต์ประสิทธิภาพการส่งเทียบกับจำนวนข้อมูลที่ต้องการส่ง เช่น

ถ้ากำหนดค่า MTU เป็น 128 ไบต์ แสดงว่าจะต้องเป็นข้อมูล 88 ไบต์ และเป็นส่วนหัว 40 ไบต์ และพิจารณาการส่งข้อมูลจำนวน 512 ไบต์ แสดงว่าเป็นจำนวนข้อมูลที่จะต้องส่งทั้งหมดเท่ากับ 6 แพคเก็ต (512 / 88)

แต่จำเป็นจะต้องมีแพคเก็ตที่ใช้ในการสื่อสารอีก 4 แพคเก็ต ดังนั้นจึงมีแพคเก็ตที่ต้องทำการส่งทั้งหมด 10 แพคเก็ต ดังนั้นจะต้องส่งข้อมูลทั้งหมด

= 400 + 512 ไบต์

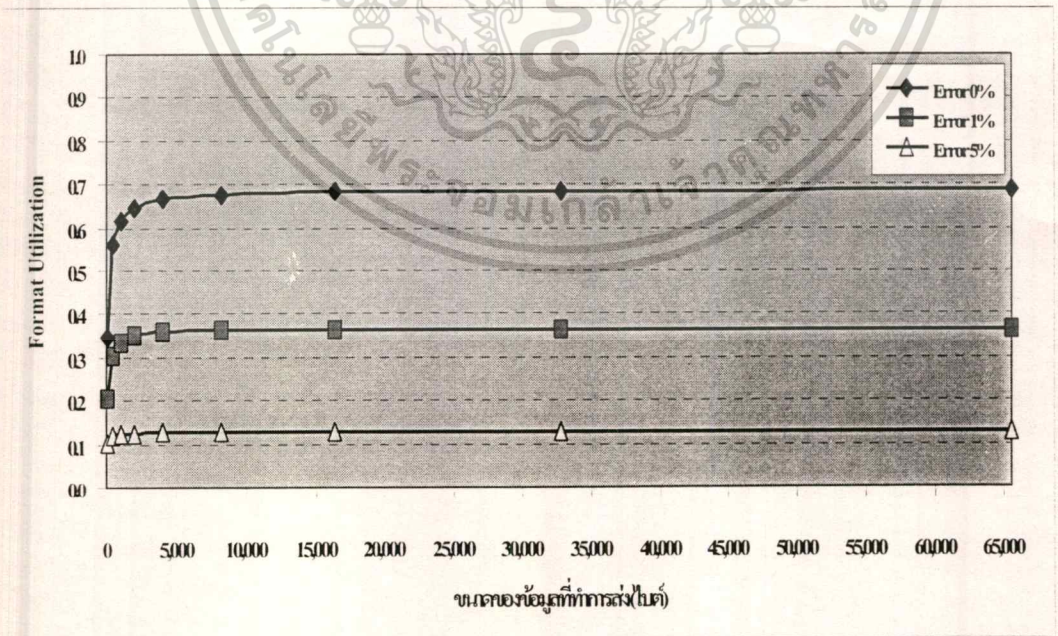
= 912 ไบต์

คิดเป็นเปอร์เซ็นต์ประสิทธิภาพ = (512/912) * 100

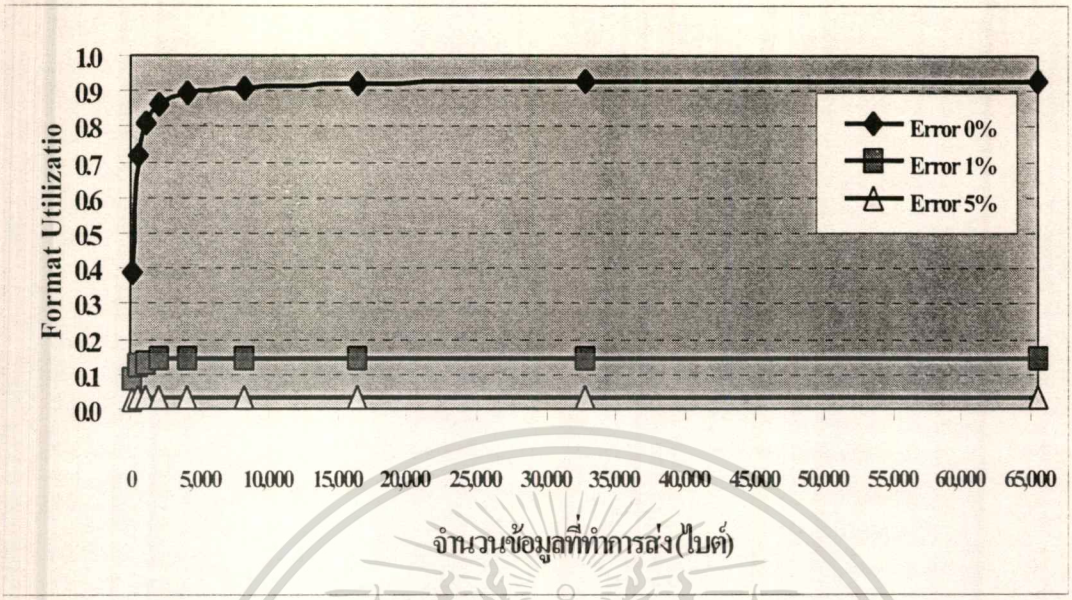
= 56.14 %

แต่ในลักษณะการคำนวณดังกล่าวไม่ได้นำค่าผิดพลาดที่เกิดขึ้นมาพิจารณา เพราะฉะนั้นถ้ามีการคิดคำนวณค่าผิดพลาดที่อาจจะเกิดขึ้นในการสื่อสารซึ่งค่าผิดพลาดนั้นอาจจะเกิดขึ้นเนื่องจากสัญญาณรบกวน (Noise) หรือ ความหนาแน่นของเครือข่ายที่สูงมาก ๆ ทำให้เกิดความผิดพลาดเกิดขึ้น ดังนั้นเพื่อให้ค่าใกล้เคียงกับความเป็นจริงแล้วจะต้องนำพจน์ความผิดพลาดมาใช้ในการคำนวณด้วย สำหรับในการสื่อสารโดยใช้โปรโตคอล TCP นั้น เมื่อเกิดการผิดพลาดเกิดขึ้นจะต้องทำการส่งข้อมูลใหม่อีกโดยพิจารณาจากเวลาที่กำหนดไว้ในการตอบกลับคืน (Timeout) โดยถ้าหากว่าไม่มีการตอบรับกลับคืนมาภายในเวลาที่กำหนดไว้ฝ่ายส่งจะต้องทำการส่งแพคเก็ตนั้นใหม่อีกครั้ง ดังนั้นความสัมพันธ์ในการพิจารณาในรูป TCP Format Utilization เท่ากับ

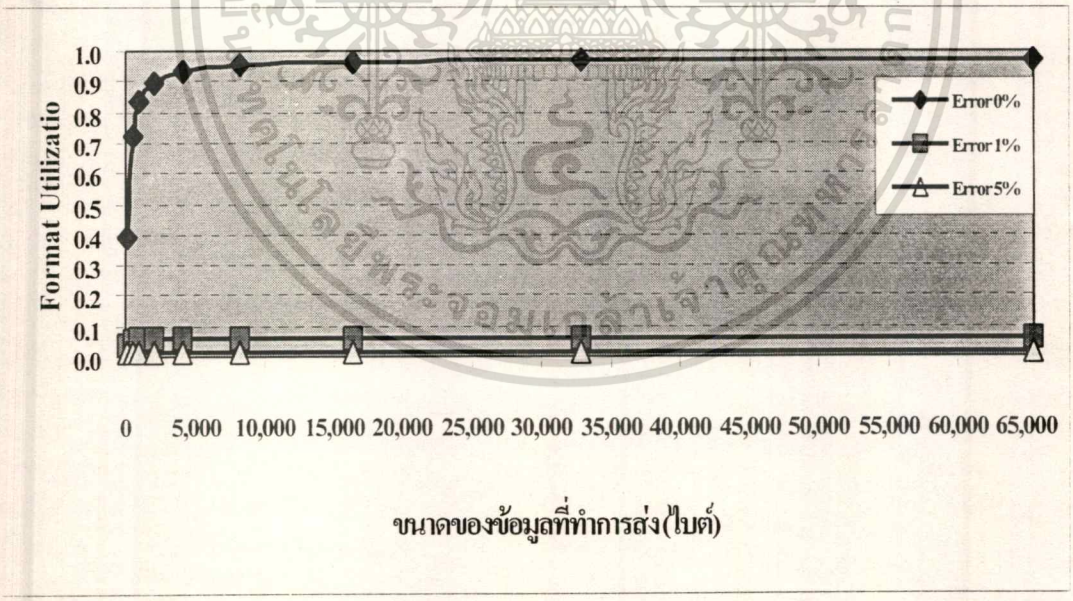
โดยกำหนดว่าทั้งค่า IP Header และ TCP Header มีค่าเท่ากับ 20 ไบต์ตามลำดับ และการเกิดความผิดพลาดเป็นเปอร์เซ็นต์นั้นหมายความว่าถ้าสมมติเกิดความผิดพลาดเกิดขึ้น 5 % หมายถึงในการส่งข้อมูล 100 ไบต์จะมีข้อมูลที่ผิดพลาดเกิดขึ้น 5 ไบต์ และข้อมูลที่ผิดพลาดที่เกิดขึ้นนั้นจะไม่อยู่ในแพคเก็ตเดียวกัน ดังนั้นจึงได้ทำการแทนค่าในความสัมพันธ์เพื่อหากราฟความสัมพันธ์ได้ออกมาดังรูปที่ 14, 15, 16 และ 17



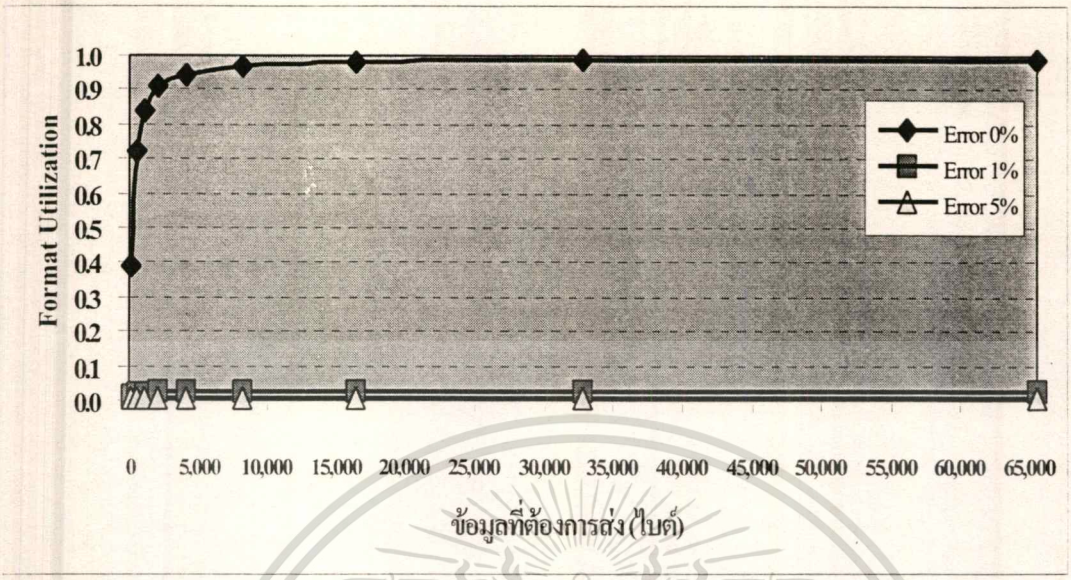
รูปที่ 14 แสดง Format Utilization ของ TCP ในการส่งข้อมูลปริมาณต่าง ๆ ที่ MTU = 128, Error 0%, 1% และ 5%



รูปที่ 15 แสดง Format Utilization ของ TCP ในการส่งข้อมูลปริมาณต่าง ๆ ที่ MTU = 576, Error 0%, 1% และ 5%



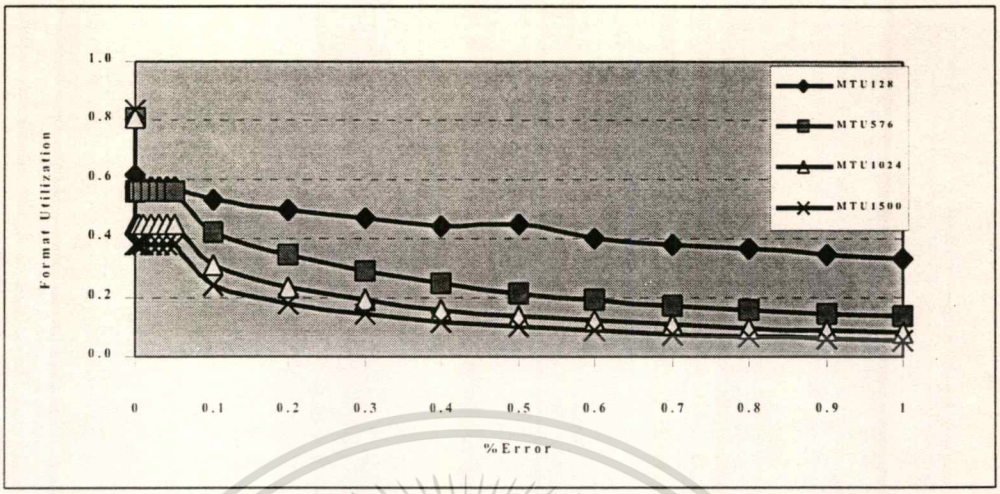
รูปที่ 16 แสดง Format Utilization ของ TCP ในการส่งข้อมูลปริมาณต่าง ๆ ที่ MTU = 1500, Error 0%, 1% และ 5%



รูปที่ 17 แสดง Format Utilization ของ TCP ในการส่งข้อมูลปริมาณต่าง ๆ ที่ MTU = 4096, Error 0%, 1% และ 5%

การวิเคราะจากกราฟของการส่งข้อมูลจำนวนต่าง ๆกันที่ใช้ MTU แตกต่างกันจะเห็นได้ว่าใน MTU ค่าต่ำการส่งข้อมูลในปริมาณน้อย ๆ จะมีประสิทธิภาพต่ำกว่าการส่งข้อมูลที่มีปริมาณมาก ทั้งนี้เนื่องจากที่ข้อมูลจำนวนน้อยนั้นจะมีข้อมูลที่ส่งมีค่าห่างจากจำนวนข้อมูลที่ส่งจริงมาก จึงทำให้มีค่าประสิทธิภาพที่ต่ำกว่าในส่วนของข้อมูลที่มีปริมาณมาก ค่าความแตกต่างของข้อมูลที่ส่งกับข้อมูลที่ส่งจริงจะแตกต่างกันน้อยลง คือมีค่าเข้าใกล้กันมากขึ้น สำหรับใน MTU ค่าสูงนั้นแนวโน้มของการส่งข้อมูลของ TCP จะมีประสิทธิภาพดีขึ้นเรื่อย ๆ ในกรณีที่มีข้อมูลที่ส่งเพิ่มมากขึ้น ทั้งนี้เนื่องจากข้อมูลที่ส่งมีค่าความแตกต่างจากข้อมูลที่ส่งจริงน้อยลง โดยจะเห็นได้ชัดในกรณีที่เป็นการส่งข้อมูลที่ปริมาณ MTU มากกว่า 576 ไบต์

สำหรับในทางปฏิบัติแล้วการที่จะเกิดความผิดพลาดหนึ่งถึงห้าเปอร์เซ็นต์เป็นไปได้น้อยมาก โดยเฉพาะอย่างยิ่งในเครือข่ายท้องถิ่น ซึ่งโดยปกติแล้วจะเกิดความผิดพลาดในระดับสิบยกกำลังลบหกถึงสิบยกกำลังลบสาม สำหรับในเครือข่ายระดับกว้างหรือ Wide Area Network (WAN) นั้นความผิดพลาดอาจจะมีค่าถึงหนึ่งเปอร์เซ็นต์ได้ แต่ถ้าเป็นความผิดพลาดระดับสามเปอร์เซ็นต์ขึ้นไปแล้วถึงเป็นความผิดพลาดที่ค่อนข้างสูงมาก ซึ่งอาจหมายถึงสัญญาณรบกวนที่สูงหรือเกิดสิ่งผิดปกติอย่างมาก ดังนั้นกราฟรูปที่ 18 จะเป็นการแสดงให้เห็นถึงความละเอียดมากยิ่งขึ้น



รูปที่ 18 แสดง Format Utilization ของ TCP เมื่อเทียบกับ % ค่าความผิดพลาดต่าง ๆ ที่ปริมาณการส่งข้อมูล 1024 ไบต์ ณ ค่า MTU แตกต่างกัน

สำหรับในการวิเคราะห์ต่อไปจะพิจารณาถึงขนาดหน้าต่าง (Windows Size) เข้ามาเกี่ยวข้อง โดยจะมีหน่วยเวลาที่ต้องเข้ามาพิจาราณารวมไปถึงขนาดความจุของช่องสัญญาณในสื่อสัญญาณ เพื่อพิจารณาถึงความสัมพันธ์ของขนาดหน้าต่างและเวลาที่ใช้ในการสื่อสารในการสื่อสารของ TCP โดยจะพิจารณาจากความสัมพันธ์ดังนี้

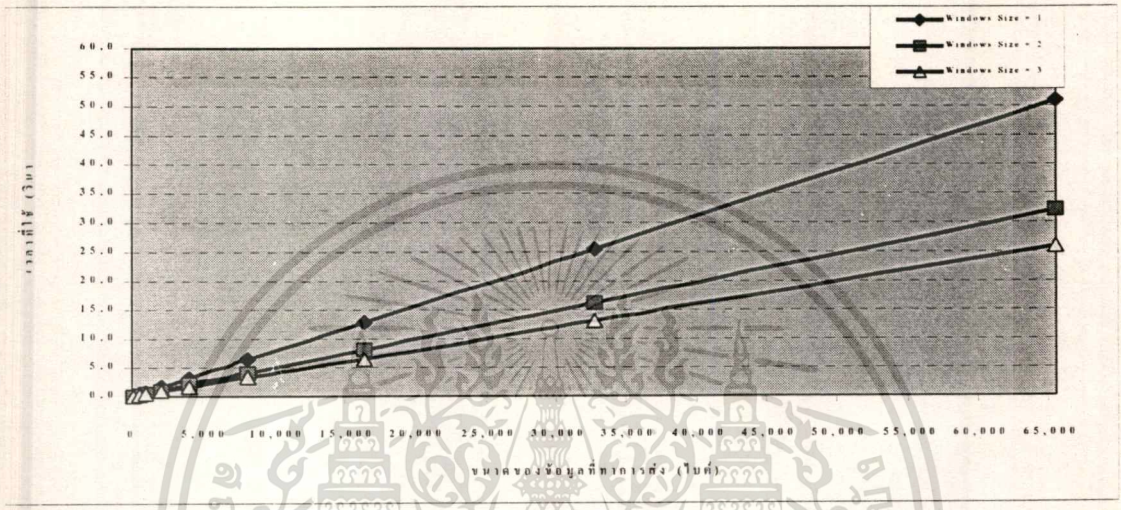
$$T = [(W*Ts) + Ti + Tack + (MTU*\%error) * (Tout + Ts + Tack)]*P \quad .....(3)$$

โดยที่ค่า

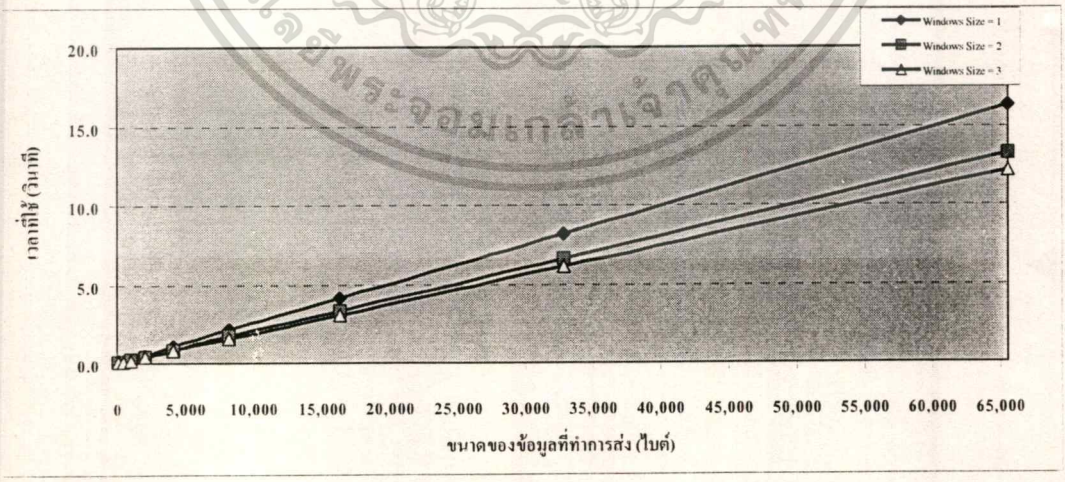
T	เวลาทั้งหมดในการสื่อสาร (วินาที)
W	จำนวนหน้าต่าง
Ts	เวลาที่ใช้ในการส่งแพคเก็ต (วินาที)
Ti	เวลาที่หน่วงไป + เวลาที่ใช้ในการตอบรับ (วินาที)
Tack	เวลาที่ใช้ในการตอบรับ
%Error	เปอร์เซ็นต์แพคเก็ตที่ผิดพลาดจากการสื่อสาร
Tout	เวลาที่ใช้รอการตอบกลับของแพคเก็ต (วินาที)
P	จำนวนแพคเก็ตทั้งหมดที่ใช้ในการสื่อสาร

สำหรับในการทดลองนี้กำหนดให้ค่าความจุของช่องสัญญาณ (C) มีค่าเท่ากับ 56,000 บิตต่อวินาที (7,000 ไบต์ต่อวินาที)และเวลาที่ใช้นั้นพิจารณาจากจำนวนข้อมูลที่ทำการส่งหารด้วย

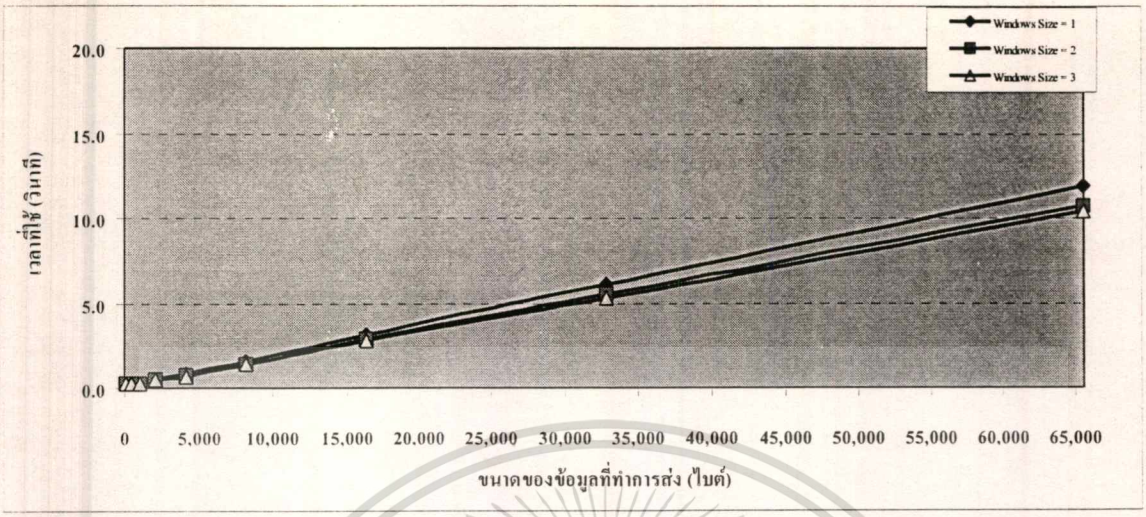
ขนาดความจุของช่องสัญญาณ โดยได้ทำการคิดถึงเวลาที่หน่วงไปเนื่องจากสาเหตุต่าง ๆ หลายประการที่ไม่สามารถควบคุมได้ เช่น ความแออัดในเครือข่าย, คุณลักษณะของแต่ละเครือข่าย รวมไปถึงสภาวะแวดล้อมขณะที่ทำการปฏิบัติ โดยกำหนดให้ค่า Ti มีค่าเท่ากับ 500 mSec จะได้ความสัมพันธ์ดังกราฟรูปที่ 19



รูปที่ 19 แสดงเวลาที่ใช้ ในการส่งข้อมูลปริมาณต่าง ๆ ที่ MTU = 128 เมื่อมีขนาดหน้าต่างแตกต่างกัน



รูปที่ 20 แสดง เวลาที่ใช้ ในการส่งข้อมูลปริมาณต่าง ๆ ที่ MTU = 576 เมื่อมีขนาดหน้าต่างแตกต่างกัน



รูปที่ 21 แสดงเวลาที่ใช้ในการส่งข้อมูลในปริมาณต่าง ๆ ที่ MTU = 1500 เมื่อมี ขนาดหน้าต่างแตกต่างกัน

รูปที่ 22 แสดงเวลาที่ใช้ในการส่งข้อมูลในปริมาณต่าง ๆ ที่ MTU = 4096 เมื่อมีขนาดหน้าต่างแตกต่างกัน

เพราะฉะนั้นจากรูปที่ 19 ถึง 22 จะพบว่าการส่งข้อมูลที่ขนาดหน้าต่างนั้นจะมีการให้ประสิทธิภาพที่แตกต่างกัน โดยการกำหนดขนาดของหน้าต่างที่จะใช้ในการส่งข้อมูลนั้นจะถูกกำหนดโดยซอฟต์แวร์ที่ถูกใช้งานอยู่ ซึ่งจะพบว่าในการส่งแต่ละครั้งนั้นถ้าคิดที่เปอร์เซ็นต์ความผิดพลาดค่าเดียวกันการส่งที่ใช้จำนวนหน้าต่างที่มากกว่าจะมีประสิทธิภาพที่ดีมากยิ่งขึ้น เพราะในการ

ส่งแพคเก็ตข้อมูลแต่ละครั้งถ้าจำนวนหน้าต่างมีค่าเท่ากับหนึ่ง จะต้องรอคอยการ ACK ทุกครั้งก่อนที่จะมีการส่งแพคเก็ตที่เหลือต่อไป แต่ถ้ามีการกำหนดขนาดหน้าต่างมากกว่าหนึ่ง ก็สามารถลดเวลาการรอคอยตรงจุดนี้ทิ้งไปได้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการศึกษาและวิเคราะห์การสื่อสารข้อมูลของ TCP ในส่วนความสัมพันธ์ของค่า MTU (Maximum Transmission Unit) ขนาดหน้าต่าง (Windows Size) และเปอร์เซ็นต์ความผิดพลาดที่เกิดขึ้นในการสื่อสาร จะพบว่าการสื่อสารข้อมูลของ TCP ที่มีการสื่อสารในลักษณะของ Connection-Oriented คือจะต้องมีการสถาปนาการเชื่อมต่อและเมื่อสิ้นสุดก็จะต้องมีการส่งสัญญาณยุติการเชื่อมต่อก่อนเพื่อให้แน่ใจว่าสามารถติดต่อถึงกันได้แล้วจึงจะเริ่มต้นการส่งข้อมูล โดยใน TCP จะใช้วิธีที่เรียกว่า Three-Way Handshake โดยจะต้องทำการ Synchonize สัญญาณกันก่อนทำการสื่อสารสิ่งต่าง ๆ เหล่านี้จัดเป็น Overhead ในการสื่อสารทั้งสิ้นกล่าวคือทำให้เกิดแพคเก็ตข้อมูลเพิ่มขึ้นในแต่ละครั้งของการเชื่อมต่อ 4 แพคเก็ต ส่งผลกระทบต่อประสิทธิภาพการสื่อสาร

ในการส่งข้อมูลของ TCP จะมีความสามารถในการส่งข้อมูลได้สูงสุด 65,535 ไบต์ โดยกำหนดค่ามาตรฐานของ TCP Maximum Segment Size และ IP Maximum Datagram Size ไว้มีขนาดมากที่สุดอย่างละ 20 ไบต์ จะพบว่าการสื่อสารแบบ TCP นั้นการสื่อสารข้อมูลขนาดของข้อมูลจำนวนสูงโดยมีค่าเข้าใกล้ 65,535 ไบต์ จะมี Format Utilization ดีกว่าการสื่อสารขนาดของข้อมูลจำนวนน้อย ๆ เพราะในการส่งแต่ละครั้งจะต้องเกิดการเพิ่มขนาดของข้อมูลในการส่งไปกับขนาดของ Header ของแต่ละโปรโตคอล ซึ่งสามารถที่จะทำการขยายขนาดได้ โดยเฉพาะในกรณีที่เกิดความผิดพลาดเกิดขึ้นแล้วจะต้องทำการส่งใหม่ถ้าหากเป็นการส่งขนาดแพคเก็ตที่ใหญ่ ก็จะทำให้เกิดโอเวอร์เฮดสูงขึ้นตามด้วยและจำนวนของหน้าต่างก็มีส่วนสัมพันธ์ด้วยเช่นกัน โดยถ้ามีการกำหนดขนาดของหน้าต่างเป็นจำนวนขนาด 2 หรือ 3 ก็จะทำให้ลดเวลาในการสื่อสารได้ เพราะเนื่องจากไม่ต้องเสียเวลารอคอยการตอบรับทุก ๆ ครั้งเมื่อมีการส่งแพคเก็ต จากผลการทดลองพบว่าขนาดของหน้าต่างต่าง ๆ กันส่งผลให้เกิดการเสียเวลาในการส่งที่แตกต่างกัน และจากการทดลองโดยเฉพาะค่าขนาดหน้าต่างที่ควรกำหนดไว้อย่างน้อยสุดคือ ขนาด 2 หน้าต่าง

5.2 ข้อเสนอแนะ

จากการศึกษาและวิเคราะห์การสื่อสารของ TCP ซึ่งเป็นการศึกษาถึงรูปแบบของการส่งแพคเก็ตข้อมูล และวิธีการติดต่อสื่อสาร และการพิจารณาการส่งข้อมูลที่ปริมาณต่าง ๆ กัน ซึ่งเป็นการศึกษาในกรณีที่คิดถึงองค์ประกอบต่าง ๆ เข้ามาเกี่ยวข้องเพื่อให้เกิดภาพพจน์ที่ตรงกับการใช้งานจริง แต่ในการทดลองครั้งนี้ไม่ได้พิจารณาถึงปัจจัยเวลาหน่วงที่เกิดจากสาเหตุต่าง ๆ เช่น สภาพสิ่งแวดล้อมที่เป็นอยู่, คุณลักษณะของเครือข่าย, ความแออัดจำนวนข้อมูลของเครือข่าย หรือ จำนวนงาน (Load) ที่ใช้ว่าเป็นอย่างไร ซึ่งหากมีการศึกษาลึกลงไปถึงเวลาที่เสียไปด้วยแล้วนั้น ก็จะสามารถทำให้เห็นได้ชัดเจนมากขึ้น

บรรณานุกรม

Commer, Douglas E., Interworking with TCP/IP Volume 1,2 Ed.London : Prentice Hall International, 1991.

Fert, Sidnie, TCP/IP Architecture Protocol Implementation, Singapore :McGraw Hill, 1993.

Jacobson, V., TCP Extensions for High Performance, RFC 1323, USC/Information Sciences Institute, May 1992.

Kevin, Washburn., TCP/IP Running a Successful Network, 2 Edition : Addiso-Wesley Longman, 1996.

Matthew, A., Inside TCP/IP, 2 Edition : New-Riders., 1995.

Miller, Mark A., Lan Protocol Handbook, M&T Publishing, Inc., 1991.

Postel, J., The TCP Maximum Segment Size and Related Topics, RFC 879, USC/Information Sciences Institute, November 1983.

Postel, J., Transmission Control Protocol, RFC 761. USC/Information Sciences Institute. January 1980.

ประวัติผู้จัดทำ

ชื่อ เลิศวิทย์ วรพงศธร

สถานที่เกิด กรุงเทพมหานคร

การศึกษา ระดับมัธยมศึกษา โรงเรียนเตรียมอุดมศึกษา

ระดับอุดมศึกษา สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์ Instrument Engineer / Contrologic Co.,Ltd.

Sale Engineer Electrical / F.E. Zuellig (Bangkok) Co.,Ltd.