

การศึกษาวิเคราะห์และเปรียบเทียบชุดโปรโตคอล TCP/IP และ IPX/SPX  
Analysis and Comparison of Protocol suites TCP/IP and IPX/SPX

โดย

เรืออากาศโท สามารถ หมดนุรักษ์

รหัส 38626008



\*H001491\*

อาจารย์ที่ปรึกษา

อาจารย์ อัครินทร์ คุณกิตติ

วัน เดือน ปี.....	07 S.A. 2549
เลขทะเบียน.....	01491
เลขเรียกหนังสือ.....	๑๗ ๑๒๕๓ ๒๕๔๐

"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นต้นฉบับของงานวิจัยที่จัดทำขึ้นไว้เพื่อใช้งาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2540

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ การศึกษาวิเคราะห์และเปรียบเทียบชุดโปรโตคอล TCP/IP และ IPX/SPX  
 นักศึกษา เรืออากาศโท สามารถ หมัดนุรักษ์  
 อาจารย์ที่ปรึกษา อาจารย์ อัครินทร์ คุณกิตติ  
 ระดับการศึกษา วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
 แขนงวิชา วิทยาการสารสนเทศ  
 พ.ศ. 2540

### บทคัดย่อ

วัตถุประสงค์ของโครงการพัฒนาระบบนี้ เพื่อทำการศึกษาถึงคุณสมบัติเฉพาะตัวของโปรโตคอลและการทำงานของชุดโปรโตคอล แล้วทำการวิเคราะห์และเปรียบเทียบระหว่างชุดโปรโตคอล TCP/IP (Transmission Control Protocol / Internet Protocol) และชุดโปรโตคอล IPX / SPX (Internetwork Packet Exchange / Sequence Packet Exchange) โดยจะเน้นที่การศึกษาในรูปแบบของเพิกเกิดของชุดโปรโตคอลทั้งสองแบบ และทำการหาประสิทธิภาพของการส่งข้อมูลในปริมาณต่างๆ รวมทั้งกำหนดการส่งข้อมูลที่ค่าของ MTU (Maximum Transfer Unit) ที่ค่าตั้งแต่ 128, 576, 1024, 1500 และ 4096 ไบต์

ผลจากการศึกษาและวิเคราะห์ ก็จะมีการจับคู่เปรียบเทียบโดย IP จะเปรียบเทียบกับ IPX ซึ่งเป็นโปรโตคอล ที่ทำงานในชั้น Network เหมือนกัน ส่วน TCP จะเปรียบเทียบกับ SPX ซึ่งเป็นโปรโตคอลแบบ Connection - Oriented และจากการศึกษาถึงรูปแบบข้อมูลและการแทนค่าในการส่งข้อมูลโดย MTU ต่าง ๆ ก็ปรากฏว่า IP มีประสิทธิภาพดีกว่า IPX และในส่วนของ TCP มีประสิทธิภาพดีกว่า SPX ตาม MTU ข้างต้น

Title Analysis and Comparison of Protocol suites TCP/IP and IPX/SPX  
Student Flg.Off.Samart Madnurak  
Advisor Mr. Akharin Khunkitti  
Level of Study Master of Science in Information Technology  
Major Information Science  
Year 1997

### ABSTRACT

The objective of this project are the analysis and comparison between TCP/IP Protocol suite and IPX / SPX by studying in characteristics of the protocols. The project is emphasize on format for sending of packet between TCP / IP and IPX / SPX and compare the effective for sending in any value of volumes based on MTU ( Maximum Transfer Unit ) at 128, 576, 1024, 1500 and 4096 Bytes.

This project will compare the Network Layer protocol between IP and IPX and the connection-oriented protocols of TCP and SPX in selected MTUs.

The result from study summaries that IP is better than IPX in the connectionless protocol type and TCP is better than SPX in the connection-oriented protocol .

## กิตติกรรมประกาศ

ผู้จัดทำขอขอบคุณ อาจารย์ อัครินทร์ คุณกิตติ อาจารย์ที่ปรึกษาซึ่ง เป็นผู้ให้คำแนะนำในการศึกษาในครั้งนี้รวมทั้งให้แนวทางในการศึกษาวิเคราะห์ครั้งนั้นนอกจากนี้ต้องขอขอบคุณเพื่อนๆ ที่แสวงหาที่คอยให้กำลังใจให้ทำงานได้สำเร็จในที่สุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	1
บทคัดย่อภาษาอังกฤษ	2
กิตติกรรมประกาศ	3
สารบัญ	4
สารบัญตาราง	6
สารบัญภาพ	7
บทที่	
1. บทนำ	8
1.1 ความเป็นมา	8
1.2 วัตถุประสงค์	9
1.3 ขั้นตอนและวิธีดำเนินการ	9
1.4 ประโยชน์ที่คาดว่าจะได้รับ	9
2. โพรโตคอล TCP/IP	10
2.1 TCP/IP OVERVIEW	10
2.2 อินเทอร์เน็ตแอดเดรส	11
2.3 โพรโตคอล IP	13
2.4 โพรโตคอล TCP	16
2.5 โพรโตคอล UDP	19
2.6 Establishing A TCP Connection	21
2.7 Closing a TCP Connection	23
2.8 TCP Connection Reset	23
3. โพรโตคอล IPX/SPX	24
3.1 โพรโตคอล IPX	24
3.2 IPX Packet Structure	24

## สารบัญ

บทที่	หน้า
3.4 โพรโตคอล SPX	27
3.5 SPX Packet Structure	27
4. การเปรียบเทียบชุดโปรโตคอล TCP/IP กับ IPX/SPX	30
4.1 เปรียบเทียบโปรโตคอล IP กับ IPX	30
4.2 เปรียบเทียบโปรโตคอล TCP กับ SPX	34
5. สรุปผลการวิเคราะห์เปรียบเทียบและข้อเสนอแนะ	54
5.1 สรุปผลการวิเคราะห์เปรียบเทียบ	54
5.2 ข้อเสนอแนะ	55
บรรณานุกรม	56
ประวัติผู้เขียน	57

## สารบัญตาราง

ตารางที่	หน้า
1. TCP port number	18
2. บิตของ Code field ใน TCP header	19
3. UDP port number	20
4. Format Utilization ของ IP และ IPX	31
5. Format Utilization ของ TCP และ SPX	35
6. การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 128	39
7. การส่งข้อมูลปริมาณต่างๆ โดย SPX ที่ MTU = 128	41
8. การส่งข้อมูลปริมาณต่างๆ โดย SPX ที่ MTU = 576	43
9. การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 576	44
10. การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 1024	46
11. การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 1500	48
12. การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 4096	50

## สารบัญภาพ

ภาพที่	หน้า
1. ลำดับชั้นของโปรโตคอล TCP/IP	11
2. รูปแบบอินเตอร์เน็ตแอดเดรส	12
3. รูปแบบของ IP ดาต้าแกรม	14
4. ส่วนย่อยของฟิลด์ SERVICE TYPE	14
5. รูปแบบของ TCP เชกเมนต์	17
6. รูปแบบของ UDP ดาต้าแกรม	19
7. รูปแบบของ pseudo header UDP Checksum	21
8. การสถาปนาการเชื่อมต่อแบบ three-way handshake	22
9. การยุติการเชื่อมต่อ	23
10. IPX Packet structure	25
11. SPX Packet structure	28
12. กราฟ Format Utilization ของ IP และ IPX	32
13. กราฟ Format Utilization ของ TCP และ SPX	36
14. กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 128	38
15. กราฟการส่งข้อมูลปริมาณต่างๆ โดย SPX ที่ MTU = 128	40
16. กราฟการส่งข้อมูลปริมาณต่างๆ โดย SPX ที่ MTU = 576	42
17. กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 576	45
18. กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 1024	47
19. กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 1500	49
20. กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 4096	51
21. กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP และ SPX ที่ MTU ต่างกัน	52

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมา

ปัจจุบันมีการใช้งานเครื่องคอมพิวเตอร์ในระบบเครือข่ายอย่างกว้างขวางซึ่งมีทั้งการใช้งานในลักษณะที่เป็นเครือข่ายท้องถิ่น (LAN) ที่มีการนำเอา Novell NetWare มาใช้เป็น Network Operating System (NOS) นอกจากนั้นยังมีการใช้งาน NOS. ตัวอื่นอีกด้วยเช่น การใช้ Window NT ซึ่งเป็น NOS ที่ใช้งานอย่างกว้างขวางเช่นกันการใช้งานจากคอมพิวเตอร์นอกจากจะมีการใช้งานกันภายในเครือข่ายแล้วยังมีการใช้งานแบบเชื่อมต่อกันระหว่าง เครือข่าย อีกด้วย ซึ่งลักษณะการใช้งานเชื่อมต่อกันระหว่างเครือข่ายนี้รู้จักดีในนามของ Internetworking ซึ่งในสถานการณ์ปัจจุบันนี้ด้วยแล้วได้มีการใช้งานของ Internet อย่างแพร่หลายและสิ่งที่น่าจะเป็นพื้นฐานในระบบเครือข่ายคอมพิวเตอร์อย่างหนึ่งก็คือ โพรโตคอลการสื่อสารข้อมูล ซึ่งมีการใช้งานกันอยู่หลายชนิดแต่ที่เป็นที่รู้จัก และคุ้นเคยกันดีทั้งในอดีตและปัจจุบันก็คือ ชุดของโพรโตคอล TCP/IP (Transmission Control Protocol / Internet Protocol) ซึ่งเป็นโพรโตคอล ที่ถูกพัฒนาโดย US Department of Defense (DoD) เมื่อกลางปี ค.ศ.1970 และโพรโตคอลนี้ ก็ได้มีการใช้งานในการเชื่อมต่อระหว่างเครือข่ายคอมพิวเตอร์มากมาย นอกจากชุดโพรโตคอล TCP/IP นี้แล้วยังมีชุดโพรโตคอลอื่นที่มีการใช้งานในระบบเครือข่ายอีกและเป็นที่ยอมรับกันดีอีกหนึ่งชุดโพรโตคอลก็คือ IPX/SPX (Internetwork Packet Exchange/Sequence Packet Exchange) ซึ่งเป็นชุดโพรโตคอลที่มีการใช้งานอยู่ใน ซอฟต์แวร์ Novell Netware ที่เป็น Network Operating System ที่มีการใช้งานอย่างกว้างขวางเช่นกัน จากที่กล่าวมาข้างต้น เพื่อให้ เป็นการเข้าใจถึงหลักการทำงานของชุดโพรโตคอลทั้งสองผู้เขียนจึงได้ศึกษาวิเคราะห์และ เปรียบเทียบการทำงานของชุดโพรโตคอลทั้งสองนี้เพื่อให้มีความรู้และความเข้าใจในหลักการส่งข้อมูลของชุดโพรโตคอล ทั้งสองข้างต้นรวมทั้งรู้ว่าชุดโพรโตคอลใดมีประสิทธิภาพในการส่งข้อมูลมากกว่ากัน

## 1.2 วัตถุประสงค์

เพื่อเป็นการศึกษาเปรียบเทียบชุด โพรโตคอล TCP/IP และ ชุดโพรโตคอล IPX/SPX ว่าชุดโพรโตคอลแต่ละแบบนั้นมี ลักษณะการทำงานเป็นอย่างไรซึ่งจะเป็นการศึกษาและจะเป็นการวิเคราะห์โดยจะทำการศึกษารูปแบบของแพ็กเก็ตข้อมูลของโพรโตคอลต่างๆ โดยจะเน้นไปที่การศึกษาถึงการส่งข้อมูลในปริมาณต่าง ๆ ว่ามี Utilization เป็นอย่างไร ซึ่งจะพิจารณา จาก Overhead ที่เกิดจากการส่งข้อมูล โดยในที่นี้จะพิจารณาในกรณีที่การส่งข้อมูลไม่มีข้อผิดพลาดเกิดขึ้นเท่านั้น

## 1.3 ขั้นตอนการศึกษา

- 1.3.1 ทำการศึกษาคูณลักษณะของ ชุด Protocol TCP/IP
- 1.3.2 ทำการศึกษาคูณลักษณะของชุด Protocol IPX/SPX
- 1.3.3 ทำการวิเคราะห์เปรียบเทียบระหว่างชุด Protocol ทั้งสอง

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 มีความรู้ความเข้าใจในชุดโพรโตคอล TCP/IP
- 1.4.2 มีความรู้ความเข้าใจในชุดโพรโตคอล IPX/SPX
- 1.4.3 ทราบถึงประสิทธิภาพการส่งข้อมูลที่ปริมาณ MTU ต่างๆ ของการส่งข้อมูลจำนวนต่าง ๆ กัน ของ TCP และ SPX ในกรณีที่การส่งข้อมูลไม่มีข้อผิดพลาดเกิดขึ้น

## บทที่ 2

# โปรโตคอล TCP/IP

### 2.1 TCP/IP OVERVIEW [1]

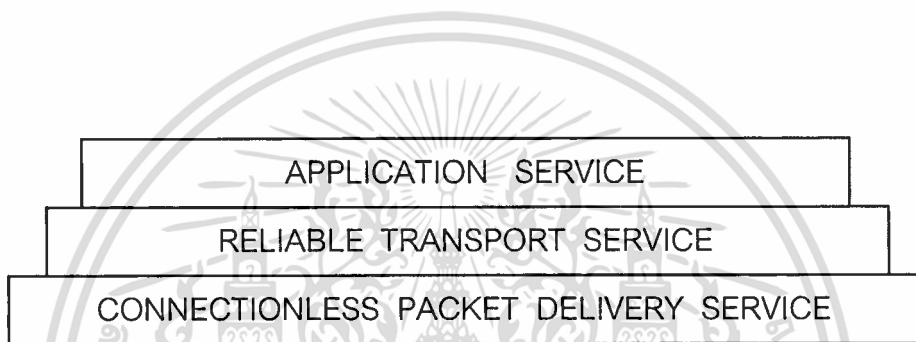
TCP/IP (Transmission Control protocol/Internet protocol) คล้ายคลึงกับ NetBIOS, IPX และ SPX ในหลายลักษณะ TCP/IP ได้รับการออกแบบโดย The Department of Defense เพื่อใช้ใน ARPANET ซึ่งเป็นเครือข่ายขนาดใหญ่ ที่เชื่อมสถานีต่าง ๆ ในโครงการของ DoD Advanced Research Projects Agency ตัว TCP/IP เป็นชุดโปรโตคอลมี IP ให้การสื่อสารแบบดาต้าแกรมระหว่างโหนดบนเครือข่าย TCP เหมือนกับ NetBIOS ในแง่ที่ให้การสื่อสารแบบจุดต่อจุด และรับประกันการได้รับข้อมูล มียูทิลิตี้มาตรฐานสำหรับการส่งไฟล์ (FTP) การเรียกโปรแกรมทำงานจากระยะไกล (TELNET) และการส่งไปรษณีย์อิเล็กทรอนิกส์ (SMTP) เนื่องจาก TCP/IP เป็นโปรโตคอลสาธารณะ จึงได้รับความนิยมนำไปใช้เป็นหลักสำหรับการเชื่อมต่อแลนต่างระบบ TCP/IP ทำงานตามแนวคิด การทำเครือข่ายแบบเท่าเทียมระบบคอมพิวเตอร์ทั้งหมดที่เชื่อมต่อ โดยเครือข่าย TCP/IP มีความเท่าเทียมกันในมุมมองของเครือข่ายแม้ว่าคอมพิวเตอร์ บางเครื่องจะมีการทำงาน และความสามารถมากกว่าเครื่องอื่นก็ตาม ผู้ออกแบบ TCP/IP วางรากฐานสถาปัตยกรรม ของมันบนลำดับชั้นของโปรโตคอล และอินเทอร์เน็ตแอดเดรส (Internet address) ซึ่งเป็นแผน การสำหรับจำแนกคอมพิวเตอร์

ในทางเทคนิค TCP/IP ประกอบด้วยโปรโตคอลสองตัว แม้ว่าบ่อยครั้งที่ผู้คนใช้ TCP/IP เป็นชื่อสามัญสำหรับกลุ่มของโปรโตคอล และโปรแกรมยูทิลิตี้

โปรโตคอลทางกายภาพภายใต้ TCP/IP เป็นได้ทั้ง EtherNet, Token Ring, การเชื่อมโยงแบบอนุกรม (โมเด็ม) หรือสื่อกลางอื่นสำหรับการส่งและการรับแพ็คเกจข้อมูล TCP/IP ไม่ได้กำหนดตัวกลาง หรือโปรโตคอลในระดับ Physical โปรโตคอลที่ใช้กัน ทั่วไปมากที่สุด คือ EtherNet และการเชื่อมต่อแบบอนุกรม (โมเด็ม) แต่ก็สามารถใช้ตัวกลาง ใดๆ สำหรับการส่งข้อมูลทราบเท่าที่อุปกรณ์ที่ ทำการสื่อสารกันใช้ตัวกลางเดียวกัน เกตเวย์สามารถ ใช้เพื่อเชื่อมต่อเครือข่ายที่ใช้โปรโตคอลในระดับ Physical ต่างกัน ทั้งเครือข่ายบริเวณเฉพาะที่ และเครือข่ายบริเวณกว้างสามารถใช้ TCP/IP เพื่อแลกเปลี่ยนสารสนเทศกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Internet Protocol (IP) เป็นโปรโตคอลระดับล่างสุดของ TCP/IP Transmission Control Protocol (TCP) และ User Datagram Protocol (UDP) เป็นโปรโตคอลสองตัวที่ใช้ IP โปรโตคอลระดับกลางตัวอื่นที่ใช้ IP คือ Internet Control Message Protocol (ICMP) ICMP ใช้เพื่อควบคุมการแลกเปลี่ยนข่าวสารและข้อผิดพลาดระหว่างแม่ข่าย ลำดับชั้น Application อยู่บนสุดของ TCP/IP และประกอบด้วยโปรแกรมยูทิลิตี้และแอปพลิเคชันซอฟต์แวร์ตามภาพแสดงถึงความสัมพันธ์ของโปรโตคอลเหล่านี้



ภาพที่ 1 ลำดับชั้นของโปรโตคอล TCP/IP [1]

## 2.2 อินเทอร์เน็ตแอดเดรส

หัวใจของ TCP/IP ในการจัดเส้นทางของข่าวสาร อยู่ที่การกำหนดแอดเดรส ซึ่งเรียกว่า อินเทอร์เน็ตแอดเดรสแม่ข่าย TCP/IP และสถานีงานทุกตัวจะมีอินเทอร์เน็ตแอดเดรส โดยเฉพาะและไม่ซ้ำกัน ประกอบด้วยรหัสประจำเครือข่ายซึ่งกำหนดจากศูนย์กลางและแอดเดรสของแม่ข่าย ซึ่งถูกบริหารโดยเครือข่ายท้องถิ่น แผนการนี้ทำให้การจัดเส้นทางของข่าวสารระหว่างเครือข่ายบริเวณเฉพาะที่หรือภายในเครือข่ายเดียวกันเป็นไปได้ส่วนที่ทำหน้าที่แยกแยะอินเทอร์เน็ตแอดเดรส คือ Address Resolution Protocol (ARP)

อินเทอร์เน็ตแอดเดรส มีอยู่ 32 บิตปกติถูกแสดงในรูปแบบ AAA.BBB.CCC.DDD ในแต่ละส่วนหรือฟิลด์ ซึ่งเรียกว่า Dotted Decimal มักจะใช้ตัวเลขฐานสิบ โดยมีจุด แบ่งแยก แต่ละฟิลด์ คลาส (class) ของ อินเทอร์เน็ตแอดเดรส คือ A ถ้าเป็นฟิลด์แรกมีค่า 0 - 127 คลาส B เป็น 128 - 191 หรือคลาส C ถ้าฟิลด์แรกเป็น 192 - 223 ฟิลด์แรกมีค่าไม่เกิน 255 และในความเป็นจริงแล้วอินเทอร์เน็ตแอดเดรสมี 5 ประเภท ซึ่งคลาส E มีไว้สำหรับ ใช้ในอนาคต หรือเพื่อการทดลอง ตัวอย่างอินเทอร์เน็ตแอดเดรสในคลาส A เช่น 89.1.10.2

## Class A

0	Network (7 bits)	Local address (24 bits)
---	------------------	-------------------------

## Class B

10	Network (14 bits)	Local address (16 bits)
----	-------------------	-------------------------

## Class C

110	Network (21 bits)	Local address (8 bits)
-----	-------------------	------------------------

## Class D

1110	Multicast address (28 bits)
------	-----------------------------

## Class E Future format

11110	Future use
-------	------------

## ภาพที่ 2 รูปแบบอินเตอร์เน็ตแอดเดรส

การแปลความหมายของอินเตอร์เน็ตแอดเดรสจะขึ้นอยู่กับคลาสของมัน สำหรับแอดเดรสในคลาส A พิลด์แรกจะเป็นส่วนของเครือข่าย (network portion) ในคลาส B ส่วนของเครือข่ายคือสองพิลด์แรก สำหรับคลาส C จะใช้สามพิลด์แรก ระบบคอมพิวเตอร์ บนเครือข่ายหนึ่ง ควรใช้คลาสอินเตอร์เน็ตแอดเดรสเดียวกันทั้งหมด ส่วนของแอดเดรสแม่ข่าย (host address) ประกอบด้วยพิลด์ที่เหลือต่อจากส่วนของเครือข่ายแต่ละพิลด์ในแอดเดรส แม่ข่าย มีค่าน้อยกว่า 256 การกำหนดไม่ควรใช้ค่า 0 ในทุกพิลด์ของแอดเดรสแม่ข่าย โดยปกติซอฟต์แวร์ของ TCP/IP จะรับรู้ว่าจะต้องมีอย่างน้อยหนึ่งพิลด์ที่ไม่เป็น 0 ขนาดและช่วงของแอดเดรส แม่ข่ายขึ้น อยู่กับคลาส

192.10.100.1

192.10.100.2

192.10.100.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์เน็ตแอดเดรสทั้งสามตัวนี้อยู่ในคลาส C เพราะว่าฟิลด์แรกเป็น 192 ส่วนของเครือข่าย ของแอดเดรส ในตัวอย่างคือ 192.10.100 และส่วนของแอดเดรสแม่ข่ายคือ 1,2 และ 3 ตัวเลข เหล่านี้ยากแก่การจดจำและใช้เพื่อติดต่อกับสถานี่งานแต่ละเครื่องบนเครือข่าย ดังนั้น การกำหนด ตัวเลขนี้ จึงยอมให้กำหนดชื่อของแม่ข่าย (host name) ให้กับ อินเทอร์เน็ต แอดเดรส ชื่อของแม่ข่ายนี้จำได้ง่ายกว่า วิธีการกำหนดชื่อในระดับสูงขึ้นไปอีก เรียกว่า การกำหนดชื่อโดเมน (Domain Naming)

Network Information Center ซึ่งเป็นส่วนหนึ่งของ SRI International เป็นผู้กำหนดแอดเดรสของเครือข่ายอินเทอร์เน็ต และชื่อโดเมนเพื่อรับประกันว่าการจัดเส้นทาง ของข่าวสาร TCP/IP ระหว่างบริษัท และองค์กรมากมายจะไม่ซ้ำซ้อนกัน

### 2.3 โพรโตคอล IP

Internet Protocol จะเป็นการส่งแบบ Connectionless IP จะมีการนิยาม 3 ส่วนที่สำคัญคือ ลำดับแรก IP โพรโตคอล หน่วยพื้นฐานของการส่งข้อมูลผ่าน TCP/IP ซึ่งจะกำหนดรูปแบบของข้อมูลที่จะส่ง และลำดับที่สอง IP ซอฟต์แวร์ จะทำหน้าที่ routing คือเลือกเส้นทางของข้อมูลที่จะส่ง และลำดับที่สามจะเพิ่มความถูกต้องของทั้งรูปแบบของ data และ routing ซึ่งเป็นไปตามกฎและลักษณะที่ host และ gateways จะกระทำกับ packets

ในระดับ physical network หน่วยที่ใช้ในการส่งผ่านข้อมูลเราเรียกว่า frame ซึ่งประกอบด้วย header และ data ซึ่ง header จะให้ข้อมูลเช่น source address และ destination address ใน ส่วนของ internet protocol เรียกหน่วยการโอนย้ายข้อมูลว่า Internet datagram หรือบางครั้ง เรียกว่า IP datagram หรือเรียกง่าย ๆ ว่า datagram ซึ่งคล้ายกับ physical network frame คือ datagram จะแบ่งออกเป็นส่วนของ header และส่วนของ data ส่วนของ datagram header จะประกอบไปด้วย source address และ destination address และชนิดของ field ที่ประกอบ อยู่ใน datagram ข้อแตกต่างกันคือ datagram header จะบรรจุ IP address ส่วนที่ frame header จะเป็น Physical address รายละเอียดของรูปแบบ datagram แสดงดังภาพที่ 3

0	4	8	16	19	24	31	
VERS		HLEN		SERVICETYPE		TOTAL LENGTH	
IDENTIFICATION				FLAGS		FRAGMENT OFFSET	
TIME TO LIVE			PROTOCOL		HEADER CHECKSUM		
SOURCE IP ADDRESS							
DESTINATION IP ADDRESS							
IP OPTION(IF ANY)						PADDING	
DATA							
....							

ภาพที่ 3 รูปแบบของ IP ดาต้าแกรม

VERS จะเป็น 4 bits แรกของดาต้าแกรม 필ด์นี้จะระบุ version ของ IP โปรโตคอล ที่ใช้ในการสร้าง ดาต้าแกรม มันใช้สำหรับการตรวจสอบ สถานีส่ง,สถานีรับ และ gateway ซึ่ง IP ซอฟต์แวร์ ต้องตรวจสอบ version 필ด์นี้ก่อนจะ Process datagram เพื่อให้แน่ใจว่า ซอฟต์แวร์สามารถใช้งาน ดาต้าแกรม นี้ได้ ปัจจุบันเราใช้ IP โปรโตคอล เวอร์ชัน 4

HLEN (header length) 필ด์นี้จะมี 4 bits ซึ่งจะเป็นส่วนที่กำหนดความยาวของ ดาต้าแกรม โดยจะบอกจำนวนเป็น 32-bit words ซึ่งแต่ละฟิลด์ในส่วนของ header นี้ จะมีค่า ความยาวคงที่ยกเว้น IP OPTIONS และ PADDING 필ด์ ซึ่งปกติ header ที่ไม่มีฟิลด์ IP OPTIONS และ PADDING จะมีขนาด 20 bytes ซึ่ง header length จะมีค่าเท่ากับ 5

TOTAL LENGTH 필ด์นี้จะมีขนาด 16 bits ใช้สำหรับระบุความยาวของ IP datagram บอกจำนวนเป็น bytes จะรวมทั้ง header และ data ซึ่งขนาดของส่วนของ data สามารถ คำนวณได้จากการลบขนาดของ header ออกจาก TOTAL LENGTH จากที่ TOTAL LENGTH มีขนาด 16 บิต ทำให้จำนวนสูงสุดของ IP datagram คือ  $2^{16} - 1$  หรือ 65,535 bytes

SERVICE TYPE 필ด์นี้มีขนาด 8 บิต ซึ่งแบ่งเป็น 5 ส่วนย่อยดังภาพที่ 4

0	1	2	3	4	5	6	7
PERCEDENCE		D	T	R	UNUSED		

ภาพที่ 4 ส่วนย่อยของ 필ด์ SERVICE TYPE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRECEDENCE เป็น 3 บิต ที่ใช้กำหนด datagram precedence ซึ่งจะมีค่าตั้งแต่ 0 (normal precedence) ถึง 7 (network control)

Routine	000
Priority	001
Immediate	010
Flash	011
Critical	100
Internetwork Control	110
Network Control	111

บิต D, T และ R ใช้กำหนดชนิดของการขนส่ง datagram ตามความต้องการ คือถ้า บิต D ถูกตั้งค่าก็จะหมายถึงต้องการ low delay ถ้าบิต T ถูกตั้งค่าหมายถึงต้องการ high throughput และถ้าบิต R ถูกตั้งค่าจะหมายถึงต้องการ high reliability

IDENTIFICATION ฟิลด์นี้ใช้ในการควบคุม fragmentation และ reassembly ของ datagram โดยฟิลด์นี้จะกำหนดหมายเลขเฉพาะให้กับ datagram ซึ่ง fragments ของ datagram นั้นจะมีค่า IDENTIFICATION NUMBER เดียวกัน

FRAGMENT OFFSET ฟิลด์นี้ ใช้ในการช่วย fragmented datagram และ reassembly ซึ่งค่าของ FRAGMENT OFFSET นี้จะมีขนาด 13 บิต โดยค่าที่เป็นไปได้มันจะต้องเป็นจำนวนทวีคูณของ 8 bytes ( 64 bits) โดยที่ offset แรกของ segment นั้นจะเริ่มต้นด้วย 0 เสมอ

FLAGS ฟิลด์นี้มีขนาด 3 บิต ใช้ในการควบคุม fragmentation และ reassembly โดยที่บิตแรกจะถูกตั้งค่าเป็น 0 ซึ่งอีก 2 บิต จะเป็นบิต D และ M จะเป็นการกำหนดว่า datagram นี้จะมีการทำ fragmentation หรือไม่ และหากมี fragment แล้ว fragment นี้เป็น fragment สุดท้ายหรือเปล่า

D 0 = Fragmentation allowed;	1 = Fragmentation not allowed
M 0 = Last fragment ;	1 = More fragments exist

TIME TO LIVE ฟิลด์นี้จะใช้กำหนดเวลา(วินาที) ในการที่จะอนุญาต ให้ datagram นี้จะอยู่ในเครือข่าย แต่ละ IP gateway เมื่อ datagram ผ่านมา มันก็จะลดค่าที่ ฟิลด์นี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยจำนวนเวลา เช่นใช้เวลา 1 วินาที มันก็จะทำการลบฟิลด์นี้ด้วย 1 จนกระทั่งฟิลด์นี้ เป็น 0 มันก็จะกำจัด datagram นี้ไป

PROTOCOL ฟิลด์นี้จะเป็นการระบุ โพรโตคอลในระดับสูงที่ใช้ IP ในการส่งข้อมูลใน datagram นี้

HEADER CHECKSUM ฟิลด์นี้จะใช้ checksum ของ header เท่านั้นไม่ ได้รวมข้อมูล checksum นี้จะคำนวณโดย 16-bit one's complement ของ one's complement ของผลรวมของ header สำหรับการคำนวณ checksum นี้ ฟิลด์นี้จะตั้งค่าไว้เป็น 0

SOURCE IP ADDRESS เป็น 32-bit IP address ของผู้ส่ง datagram

DESTINATION ADDRESS เป็น 32-bit IP address ของสถานีรับ datagram

OPTIONS ฟิลด์นี้จะมีคความยาวที่ไม่คงที่ ซึ่ง IP datagram โดยทั่วไปแล้วจะ ไม่ต้องมีค่า ฟิลด์นี้จะมีเพียง datagram ที่ต้องการฟิลด์ OPTIONS นี้

## 2.4 โพรโตคอล TCP

โพรโตคอล TCP เป็นโพรโตคอลที่อยู่ในชั้นที่สี่ มีการให้บริการในแบบ “connection-oriented:CO” ซึ่งหมายความว่าก่อนที่จะมีการรับส่งข้อมูลระหว่างต้นทางกับปลายทางจะต้องมีการสถาปนาการเชื่อมต่อขึ้นมาเสียก่อน และเมื่อสิ้นสุดการส่งผ่านข้อมูลก็จะยกเลิกการสถาปนา นั้นๆ ท่านผู้อ่านคงจะสงสัยว่าทำไมต้องทำให้บริการของ TCP แตกต่างไปจาก IP ผู้เขียนใคร่ขอสร้างความกระจ่างที่ประเด็นนี้ก่อนแล้วจึงจะอธิบาย Field ต่าง ๆ ของรูปแบบ โพรโตคอล TCP เนื่องจาก IP เป็น connectionless(CL) เพราะคณะผู้ออกแบบชุดโพรโตคอล TCP/IP เชื่อมั่นว่าระบบเครือข่ายที่ให้บริการในการส่งผ่านข้อมูลย่อมจะมีการผิดพลาดเกิดขึ้น ได้เสมอ และการทำให้แต่ละโหนดสถาปนาวงจรขึ้นมาทุกครั้งที่มีการรับส่งข้อมูลจาก ต้นทาง ไปยังปลายทางทำให้เกิดความยุ่งยากอย่างมากในระบบเครือข่าย ด้วยสองสาเหตุหลักนี้ทำให้การออกแบบโพรโตคอล เลือกที่จะใช้ CL สำหรับ IP แพคเก็ต เมื่อโพรโตคอลในระดับล่าง ให้บริการในลักษณะเช่นนี้ ผู้ออกแบบต้องยอมรับสภาพการมีแพคเก็ตผิดพลาด แพคเก็ตซ้ำหรือ แพคเก็ตสูญหายเพื่อแก้ไขปัญหาเช่นนี้ ก็ไม่มีอะไรดีไปกว่าการเลือกใช้ CO สำหรับโพรโตคอล ในชั้นที่อยู่เหนือกว่า ซึ่งโพรโตคอล TCP ก็เลือกใช้วิธีนี้ แต่คู่แฝดของ TCP คือ UDP นั้นมีการให้บริการในลักษณะ CL ซึ่งรายละเอียดจะขอกล่าวในภายหลัง คำศัพท์ที่ใช้เรียก TCP แพคเก็ตคือเซกเมนต์ (segment) สำหรับรายละเอียด Field ต่าง ๆ ของเซกเมนต์ TCP แสดงในภาพที่ 5 ซึ่งอธิบาย ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0	4	10	16	24	31
SOURCE PORT			DESTINATION PORT		
SEQUENCE NUMBER					
ACKNOWLEDGEMENT NUMBER					
HLEN	RESERVED	CODE BITS	WINDOW		
CHECKSUM			URGENT POINTER		
OPTION (IF ANY)				PADDING	
DATA					
...					

ภาพที่ 5 รูปแบบของ TCP เซกเมนต์

Source port มีขนาด 16 บิต การเข้าถึงแอฟพลิเคชันของชุดโปรโตคอล TCP/IP ให้หมายเลขพอร์ตเป็นตัวกำหนดดังนั้น Source port หมายถึงหมายเลขพอร์ตของสถานีต้นทาง Destination port มีขนาด 16 บิต เป็นหมายเลขพอร์ตของสถานีปลายทาง ที่ต้องการจะติดต่อด้วย แอปพลิเคชันมาตรฐานจะมีหมายเลขพอร์ตคงที่ เช่น Telnet ใช้พอร์ตหมายเลข 23, FTP (File Transfer Protocol) ใช้พอร์ตหมายเลข 20, 21 เป็นต้น ตารางที่ 1 แสดงถึงบางส่วนของหมายเลขพอร์ตที่มีการใช้งานสำหรับพอร์ตมาตรฐาน ถูกกำหนด ไว้ที่หมายเลข 0 - 255 ในส่วนที่นอกเหนือจากนี้ผู้เขียนแอฟพลิเคชันสามารถกำหนดเองได้

Sequence number มีขนาด 32 บิต TCP เซกเมนต์ใช้หมายเลขลำดับนี้ ในการบอกถึงตำแหน่งไบต์แรกของข้อมูลที่ปรากฏอยู่ในส่วนของเขต data หมายความว่าถ้าข้อมูลและส่วนหัวของ TCP เซกเมนต์ที่จะส่งมีความยาวเกินกว่าเครือข่ายได้

Acknowledgement Number 필ด์นี้จะกำหนดหมายเลข ของไบต์ที่คาดว่าจะได้รับถัดไป

HLEN 필ด์นี้จะประกอบด้วย integer ที่ระบุความยาวของ Segment header ซึ่งจะเขียนอยู่ในรูปเท่าของ 32 บิต ซึ่งมันจะมีความจำเป็นเพราะในฟิลด์ Options จะมีความยาวไม่แน่นอน ซึ่งขนาดของ TCP header จึงมีค่าเปลี่ยนแปลงได้ตามขนาดของฟิลด์ Options

Reserved 필ด์ นี้จะถูกสงวนไว้สำหรับอนาคตจะถูกตั้งค่าไว้เป็น 0

Code bits จะถูกใช้โดยซอฟต์แวร์ TCP เพื่อกำหนดวัตถุประสงค์และสิ่งที่อยู่ในเซ็กเมนต์ มีทั้งหมด 6 บิตมีความหมายดังตารางที่ 2

Checksum มีขนาด 16 บิต ใช้วิธีการเช่นเดียวกับ IP แพคเก็ต แต่จะมีส่วนที่นำมาใช้ในการคำนวณเพิ่มเติมนอกเหนือไปจากส่วนหัวของ TCP เซกเมนต์ซึ่งจะขอกกล่าวในภายหลัง

Window ฟิลด์นี้จะเป็นการใช้งานโดย ซอฟต์แวร์ TCP ว่าขณะเวลานั้นสามารถจะรับข้อมูลได้จำนวนเท่าไรซึ่งสามารถกำหนด buffer size ใน ฟิลด์ Window ซึ่งฟิลด์นี้จะประกอบไปด้วย 32 บิต

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 Users
13	DAYTIME	davtime	Daytime
15	-	netstat	Network status program
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chargen	Character Generator
20	FTP-DATA	ftd-data	File Transfer Protocol(data)
21	FTP	ftp	File Transfer Protocol
23	TELNET	telnet	Terminal Connection
25	SMTP	smtp	Simple Mail Transfer Protocol
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	hostnames	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 Service
117	UUCP-PATH	uucp-path	UUCP Path Service
119	NNTP	nntp	USENET News Transfer Protocol
129	PWDGEN	-	Password Generator Protocol
139	NETBIOS-SSN	-	NETBIOS Session Service
160-223	Reserved		

### ตารางที่ 1 TCP port number

Urgent pointer มีขนาด 16 บิต เปรียบเสมือนเป็นค่า offset ของ sequence number โดยจะบอกถึงตำแหน่งของไบต์สุดท้ายภายใน TCP เซกเมนต์ที่เป็นข้อมูลเร่งด่วน (ข้อมูลภายใน TCP เซกเมนต์หนึ่ง ๆ อาจจะไม่มีความเร่งด่วนทั้งหมดก็ได้) ที่ต้องการให้ แอปพลิเคชันพิจารณาทันที หนึ่งค่าที่ถูกบรรจุในเซตนี้จะมีความหมายก็ต่อเมื่อ เซตย่อย urgent ถูกกำหนด

ค่าเป็น "1" เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Option ส่วนของ option นี้ อาจจะมีหรือไม่มี หรือมีขนาดเท่าใดขึ้นอยู่กับชนิดของ TCP เชกเมนต์

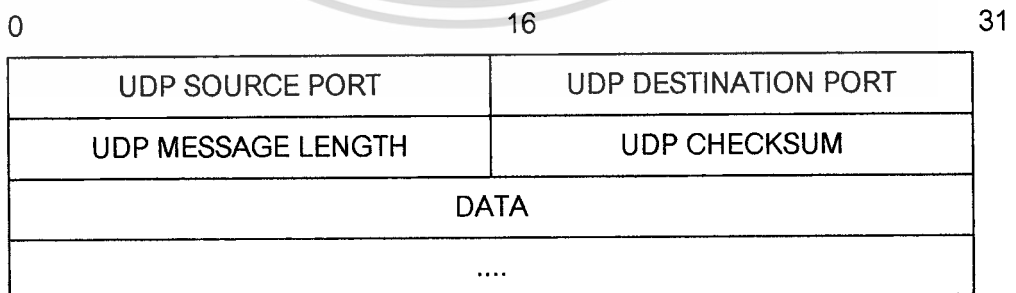
Padding มีขนาด 0-3 ไบต์ ใช้เป็นส่วนที่ทำให้ขนาดของ option เป็นจำนวนเท่าของ 32 บิต

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 2 บิตของ Code field ใน TCP header

## 2.5 โพรโตคอล UDP

โพรโตคอล UDP เป็นคู่แข่งของ TCP แต่มีการให้บริการแบบ connectionless (CL) รูปแบบของโพรโตคอลแสดงในภาพที่ 6 โดยมีรายละเอียดดังนี้



ภาพที่ 6 รูปแบบของ UDP ดาต้าแกรม

Decimal	Keyword	UNIX Keyword	Description
0			Reserved
7	ECHO	echo	Echo
9	DISCARD	discard	Discard
11	USERS	systat	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	NAMESERVER	name	Host Name Server
43	NICNAME	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	-	syslog	system log
525	-	timed	Time daemon

### ตารางที่ 3 UDP port number

Source & Destination port แสดงในตารางที่ 3 มีความหมาย เช่นเดียวกับที่กล่าวในหัวข้อโปรโตคอล TCP ถ้าผู้อ่านสังเกตพอร์ตหมายเลข 53 จะพบว่า ทั้ง TCP และ UDP ใช้พอร์ตหมายเลขเดียวกัน แต่ทำไมสถานีปลายทางจึงแยกแยะได้ว่าเป็น TCP เซกเมนต์หรือ UDP เซกเมนต์ สำหรับคำตอบก็ขอให้ย้อนกลับไปดูที่หัวข้อโปรโตคอล IP ในส่วนของเซต protocol

length มีขนาด 16 บิต ใช้บอกความยาวของ UDP เซกเมนต์(นับเป็นจำนวนไบต์)

Checksum มีขนาด 16 บิต มีวิธีการในการคำนวณเช่นเดียวกับ TCP

ในการคำนวณ Checksum ของทั้ง TCP และ UDP จะมีการเพิ่ม "pseudo header" เข้ามาในการคำนวณ ดังแสดงในภาพที่ 7 แต่ส่วนที่เพิ่มมานี้ไม่ถูกผนวก เข้าเป็น ส่วน หัวหรือข้อมูลของ TCP และ UDP ซึ่งก็หมายความว่าในการส่งผ่าน TCP หรือ UDP เซกเมนต์ จะไม่มี pseudo header ปรากฏอยู่เลย pseudo header ประกอบด้วยเซตต่าง ๆ ดังนี้

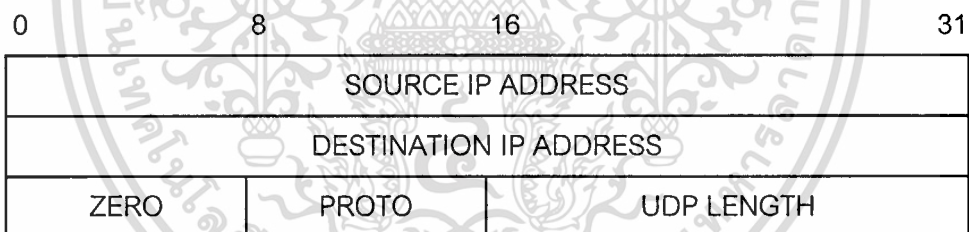
Source&destination address มีขนาด 32 บิต คือ IP แอดเดรส

Zero มีขนาด 8 บิต เป็นศูนย์หมดทุกบิต

เอกสารนี้เป็นเอกสารที่ Protocol มีขนาด 8 บิต มีความหมายเช่นเดียวกับที่กล่าวในหัวข้อโปรโตคอล IP การคำนวณว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Protocol มีขนาด 8 บิต มีความหมายเช่นเดียวกับที่กล่าวในหัวข้อโปรโตคอล IP Length มีขนาด 16 บิต เป็นความยาวของ TCP หรือ UDP เชกเมนต์ซึ่งไม่รวม pseudo header

อาจจะเกิดการสงสัยว่าทำไมต้องมี pseudo header ด้วย การสื่อสารข้อมูลโดยใช้ชุดโปรโตคอล TCP/IP นั้น อาศัยทั้ง IP แอดเดรสและหมายเลขพอร์ต (ซึ่งรวมเรียกว่า หมายเลขซ็อกเก็ต (Socket number) ) ในการสถาปนาการเชื่อมต่อและแยกแยะ แอปพลิเคชัน ออกจากกัน เพื่อป้องกันการสับสนระหว่างสถานีต้นทางและปลายทางการใช้ pseudo header เป็นส่วนหนึ่งของการคำนวณ checksum ก็เพื่อให้ปลายทางมีความมั่นใจ ต่อแพ็คเกจหรือเชกเมนต์ที่ได้รับ เพราะใน IP แพ็คเกจและ TCP หรือ UDP เชกเมนต์จะมีส่วนของ IP แอดเดรสและ หมายเลขพอร์ตตามลำดับเมื่อนำทั้งสองส่วนนี้มารวมกันให้มีรูปแบบเหมือน pseudo header ที่ต้นทางและทำการคำนวณโดยวิธีเดียวกันผลลัพธ์ย่อมเป็น checksum ตัวเดียวกัน ถ้า แพ็คเกจ และเชกเมนต์ที่ได้รับนั้นถูกต้อง แต่ถ้ามีการผิดพลาดของข้อมูลเกิดขึ้นค่าที่ได้จากการคำนวณ checksum ย่อมแตกต่างไปจากเดิมด้วย



ภาพที่ 7 รูปแบบของ pseudo header UDP Checksum

## 2.6 Establishing A TCP Connection

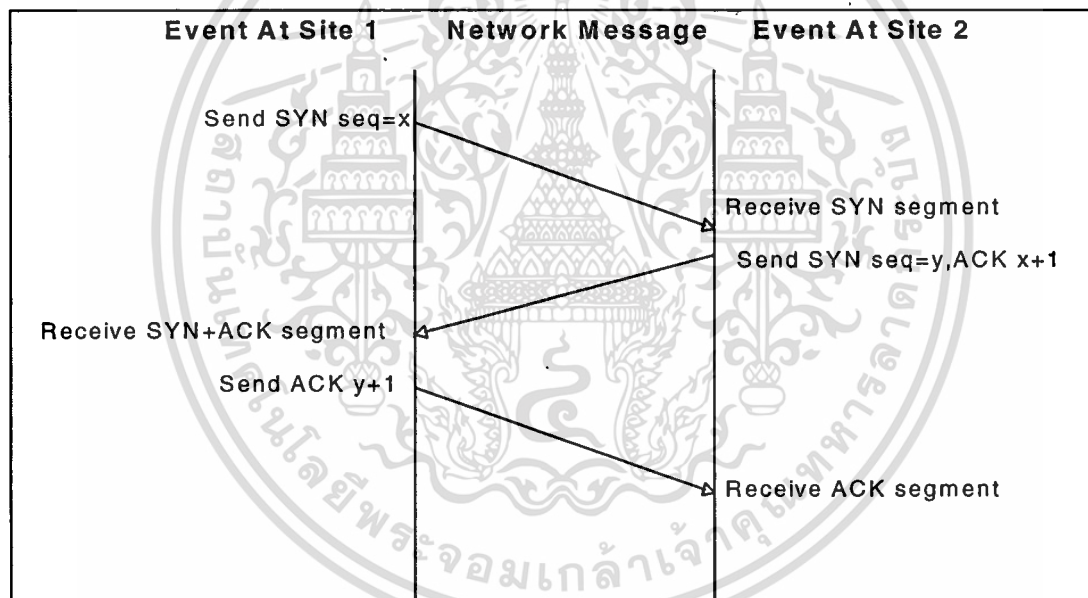
การสถาปนาการเชื่อมต่อของ TCP ใช้ three-way handshake ส่วน segment แรกของ handshake สามารถกำหนดได้โดย SYN bit ถูกกำหนดใน CODE field message ที่ 2 จะมีทั้ง SYN บิต และ ACK บิต ถูก set ซึ่งเป็นการ acknowledges ของ SYN Segment แรก และ handshake message สุดท้ายจะมีเพียง acknowledgement เพื่อเป็นการแจ้งปลายทางซึ่งทั้ง 2 ด้านจะเห็นพ้องกันถึงการเชื่อมต่อได้สถาปนาขึ้นแล้ว

ซอฟต์แวร์ TCP บนเครื่องใดเครื่องหนึ่งจะคอยรับสัญญาณ handshake จากซอฟต์แวร์

TCP จากเครื่องอื่น อย่างไรก็ตาม handshake จะถูกกำหนดอย่างรอบคอบที่ทำงานถ้าหากว่าพบเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่อง 2 เครื่องสร้างการเชื่อมต่อกันดังนั้นการเชื่อมต่อจะถูกสร้างจากด้านใดด้านหนึ่ง Data สามารถส่งได้ทั้ง 2 ทิศทาง

3-way handshake จำเป็นและเพียงพอสำหรับการ synchronization ระหว่างการเชื่อมต่อของทั้ง 2 ด้าน เพื่อจะเข้าใจว่าทำไม เราต้องเข้าใจว่า TCP นั้น สร้างขึ้นบน Unreliable packet delivery Service ดังนั้น Message มีโอกาสที่จะหายไปหรือถูกหน่วงเวลาหรือ ช้าหรือไม่เรียงลำดับ ดังนั้นโปรโตคอล มักจะใช้กลไกเวลาและการร้องขอการส่งซ้ำ ปัญหาที่เกิดขึ้น ถ้าการส่งซ้ำและการเริ่มต้นร้องขอมาถึงขณะที่การเชื่อมโยงกำลังถูกสถาปนาขึ้นหรือการร้องขอถูกทำให้ช้าลงจนกระทั่งหลังจากการเชื่อมต่อได้ถูกสร้างขึ้นแล้วหรือจบลง 3-way handshake ก็จะไม่แก้ปัญหาเหล่านี้ได้



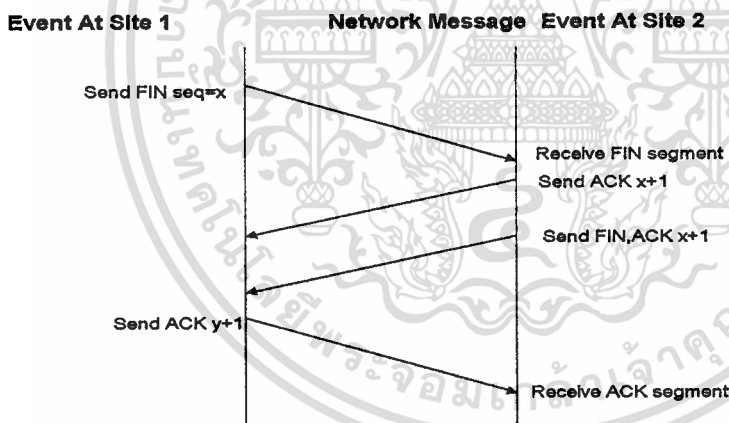
ภาพที่ 8 การสถาปนาการเชื่อมต่อแบบ three-way handshake

Three-Way handshake ทำให้สำเร็จ 2 ฟังก์ชันที่สำคัญ มันจะ Guarantees ทั้ง 2 ด้าน และมันจะอนุญาตให้ทั้งสอง ด้านเริ่มต้น Sequence Number Sequence Number ถูกส่งระหว่าง handshake เครื่องแต่ละเครื่องจะเลือกการเริ่มต้น Sequence Number แบบสุ่ม มันจะใช้ สำหรับกำหนด bytes ของชุดที่จะส่ง sequence Number ไม่สามารถเริ่มต้นด้วยค่าที่เหมือนกัน แน่แน่นอนมันเป็นสิ่งสำคัญทั้งสองด้าน จะเห็นฟองกัน กับจำนวนเริ่มต้น ดังนั้นจำนวน bytes ใช้ใน acknowledgements ซึ่งถูกใช้ใน Data segments

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 Closing a TCP Connection

สองโปรแกรมที่ใช้ TCP เพื่อการติดต่อกันสามารถยุติการติดต่ออย่างงดงามด้วย close operation TCP ใช้การดัดแปลง three-way handshake สำหรับการยุติการติดต่อ TCP จะเป็นการสื่อสารแบบ full duplex ซึ่งสามารถทำการส่งข้อมูลได้ทั้ง 2 ด้านเป็นอิสระต่อกัน เมื่อโปรแกรมประยุกต์บอก TCP ว่าไม่มี data ที่จะส่ง TCP จะปิดการติดต่อในด้านหนึ่งก่อน โดยจะส่ง finishes transmitting ไปและรอรับ Acknowledge และส่ง Segment พร้อมทั้ง ตั้งค่า FIN bit ทางด้านรับจะรับ TCP acknowledges ที่มี FIN บิตถูกตั้งค่า ระหว่างที่จะยุติ การเชื่อมต่อ TCP จะไม่ยอมรับข้อมูลอื่นอีก หลังจากที่ ด้านปลายทางที่ได้รับ FIN segment เริ่มต้นมันก็จะสร้าง FIN segment ที่สองขึ้นมาทันที, TCP จะส่ง acknowledgement ไปแจ้ง ทางด้านแรก จากนั้นก็จะส่ง acknowledgement ของ FIN segment แรกและส่ง FIN segment ที่สอง ไปด้านเริ่มต้นและด้านที่เริ่มต้นก็จะตอบรับด้วยการส่ง ACK ของการได้รับ FIN segment ที่สองก็เป็นการยุติการเชื่อมโยง



ภาพที่ 9 การยุติการเชื่อมต่อ

## 2.8 TCP Connection Reset

ปกติโปรแกรมประยุกต์จะใช้ Close operation สำหรับการยุติการเชื่อมต่อเมื่อใช้งานเสร็จ แต่บางครั้งก็ต้องการจบการใช้งานอย่างไม่ปกติซึ่ง TCP ก็เตรียม reset ไว้อำนวยความสะดวก สำหรับการสิ้นสุดการเชื่อมต่อแบบไม่ปกติ reset connection จะเริ่มต้นด้วยด้าน หนึ่งจะส่ง segment ที่มี RST บิตถูก set ไว้ซึ่งอยู่ในฟิลด์ CODE อีกด้านหนึ่งก็จะตอบรับทันทีด้วยการยกเลิกการเชื่อมต่อทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### โปรโตคอล IPX/SPX

NetWare IPX และ SPX เป็น โปรโตคอลพื้นฐาน โดยที่ IPX เป็นโปรโตคอล แบบ Connectionless และ SPX เป็นโปรโตคอล แบบ Connection-Oriented ซึ่งจากนี้ไปจะ แสดงให้เห็นถึงข้อเด่นข้อด้อยโปรโตคอลแต่ละชนิด [4]

#### 3.1 โปรโตคอล IPX

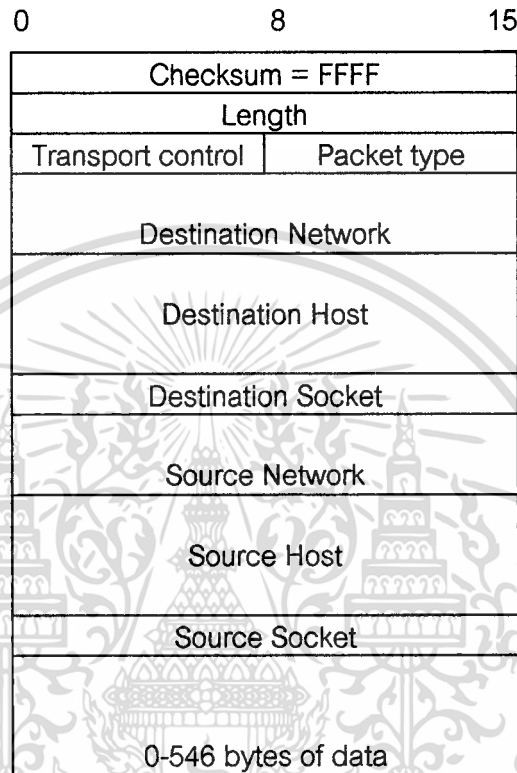
NetWare IPX เป็น Connectionless หรือ Datagram protocol ซึ่งหมายความว่า เมื่อ Workstation application ใช้ IPX เพื่อติดต่อกับ Workstation application อื่น จะไม่มี การสร้างเส้นทางการเชื่อมต่อสำหรับ 2 สถานีนี้ ดังนั้น IPX packets Data จะประกอบด้วย แอดเดรส ที่จะส่งไปยังปลายทาง ซึ่งไม่มีการรับประกันความถูกต้องของการส่งข้อมูล IPX จะทำงานอยู่ในชั้นที่ 3 ของ OSI model ซึ่งก็คือชั้นของ Network นั่นเองซึ่งจะทำงานรวมถึงการ กำหนดแอดเดรส, การกำหนดเส้นทางและการ Switching ของ Information packets IPX ทำงานได้เร็วกว่า SPX แต่อย่างไรก็ตาม IPX ก็ไม่สามารถรับประกันความถูกต้องได้ดังกับชั้นของ Transport Layer ในการส่ง Message สามารถถูกส่งโดยใช้ IPX โดยการรวมเอา Message ไว้ใน ส่วน Data ของ IPX Packet เหมือนกับการใส่จดหมายในซองจดหมาย โดยที่ IPX packet header จะมีการบรรจุ destination network , node, และ socket number ซึ่ง IPX จะทำการส่งแต่ละ packet ไปบนเส้นทางที่ สะดวก จนกระทั่ง packet นั้นไปถึงปลายทางเมื่อ packet มาถึง

#### 3.2 IPX Packet Structure

IPX packet มีลักษณะเช่นเดียวกับ Xerox IDP (Internetwork Datagram packet Protocol) แพคเกจจะประกอบด้วย 2 ส่วนเป็น ส่วน header 30 bytes และส่วนของ data ซึ่งจะมีได้ตั้งแต่ 0 ถึง 546 bytes จำนวนความยาว ของ packet ที่น้อยที่สุดคือ 30 bytes ซึ่งก็คือส่วนของ header นั่นเองและขนาดที่มากที่สุดของ IPX packet คือ 576 bytes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Checksum : Field นี้จะสอดคล้องกับของ Xerox packet header มันจะถูก Set ให้เป็น 0xFFFF โดย IPX เสมอ



ภาพที่ 10 IPX Packet Structure

Length : field นี้จะประกอบไปด้วยความยาวทั้งหมดของ packet ซึ่งจะเป็นความยาวของ header บวกกับความยาวของ data ค่าที่น้อยที่สุดคือ 30 bytes (header) และมีความยาวที่สุด 576 bytes (ส่วนของ header บวกกับ Data ) IPX จะกำหนดค่าให้ field นี้

Transport Control : field นี้ถูกใช้โดย Netware internetwork bridge IPX จะ set field นี้ให้เป็น 0 เสมอก่อนส่ง packet

Packet Type : field นี้จะทำให้รู้ถึงชนิดของ service offered IPX จะ set packet นี้ เป็น 0 หรือ 4

- 0 Unknown Packet Type
- 1 Routing Information Packet
- 2 Echo Packet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3 Error Packet
- 4 Packet Exchange Packet
- 5 Sequenced Packet Protocol Packet
- 16-31 Experimental Protocols
- 17 Netware Core Protocol

IPX user จะ set Packet type 0 หรือ 4 SPX จะ set Packet type นี้ เป็น 5

Destination Network : field นี้จะบรรจุ network number ของ networks ขึ้นอยู่กับโหมด  
ปลายทาง ใน netware networks ใน internetwork ถูกกำหนดเฉพาะ 4 byte

เมื่อ field นี้เป็น 0 Netware จะสันนิษฐานว่าโหมดปลายทางอยู่บน network เดียวกัน

Destination Node : Field นี้จะบรรจุ physical address ของ destination

ถ้า Node address FF FF FF FF FF FF จะเป็น broadcasts

Destination Socket : Field นี้จะประกอบด้วย socket address

1	Routing Information Packet
2	Echo Protocol Packet
3	Error Handler Packet
20h - 3Fh	Experimental
1h-BB8h	Registered with Xerox
BB9h-and higher	Dynamically assignable
451h	File Service Packet
452h	Service Advertising Packet
453h	Routing Information Packet
455h	NetBIOS Packet
456h	Diagnostic Packet

Source Network : Field นี้จะประกอบไปด้วย network number ซึ่งขึ้นอยู่กับ Source  
Node ถ้าเป็นภายใน Network เดียวกันจะกำหนดโดย 4 byte ไม่ซ้ำกัน

Field นี้อาจเป็น 0 หมายถึง Physical network ไม่ทราบ source

Source Node : IPX Set field นี้เป็น physical address ของ source node

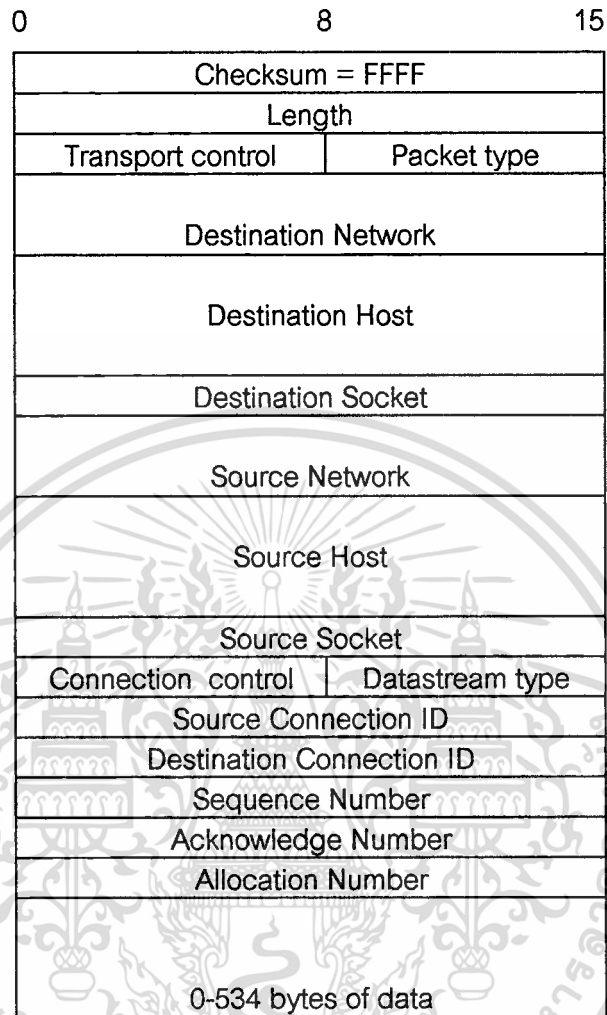
Source Socket : IPX จะ set field นี้เป็น socket address ของ โปรเซสที่ทำการรับส่ง แพคเกจ โปรเซสการสื่อสารแบบ peer-to-peer จะไม่ส่งและรับบน Socket หมายเลขเดียวกัน ในรูปแบบของ client/Server เครื่อง Server จะรอการร้องขอจาก Socket

### 3.3 โปรโตคอล SPX

SPX ก็มีลักษณะคล้ายกับ IPX แต่มันจะเพิ่ม Overhead ในส่วนที่ทำงานใน Transport Layer ของ OSI model เข้ามาซึ่งเป็นการทำงานเหนือ Network layer 1 layer โดยมีการเพิ่ม Header เข้ามาทำให้ SPX เป็นการทำงานแบบ Connection-Oriented นั่นก็หมายความว่า ก่อนที่จะมีการส่งข้อมูลของ SPX packet จะมีการสร้างเส้นทางการสื่อสารระหว่างสถานีส่งและ สถานีรับก่อน SPX จะมีการรับประกันการส่งข้อมูลที่จะส่งเป็นลำดับและจะมีการตรวจสอบข้อผิดพลาด และการซ้ำกันของ packet SPX จะรับประกันการส่งข้อมูล จะมีการสถาปนาเส้นทางของ สถานีรับและส่งก่อนที่จะมีการส่งข้อมูล และมีการตรวจสอบการส่งสถานีรับกลับมาสถานีส่ง แต่หาก สถานีรับไม่ได้รับข้อมูล ด้านสถานีส่งก็ไม่สามารถจะดำเนินการส่ง Packet ข้อมูลต่อไปได้ การวางกระจายข้อมูลไปยังหลายสถานีรับทำได้ยากเนื่องจากต้องมีสถาปนาเส้นทางการส่งข้อมูลก่อน ลำดับของ Packets จะมีการรับประกันความถูกต้อง (Guaranteed sequencing) นอกจากนี้ SPX จะกำจัด การซ้ำกันของ Packet (Duplicate packet suppression) โดยจะรับ ข้อมูลไว้แค่ เพียง 1 ชุดเท่านั้นหากมีการซ้ำกันของข้อมูล

### 3.4 SPX Packet Structure

SPX packet ก็คล้ายกับ IPX packet เพียงแต่มีการเพิ่ม header เข้าไปอีก 12 bytes ซึ่ง มันจะประกอบไปด้วย 2 ส่วนคือ 42 bytes ที่เป็น header และส่วนที่เป็น Data จะมี ข้อมูล ได้ ตั้งแต่ 0 ถึง 534 bytes จำนวน packet น้อยที่สุดที่มีได้คือ 42 bytes ซึ่งก็เป็น header ของ SPX นั่นเอง และจำนวน packet ที่มีได้สูงสุดก็คือ 576 bytes field ของ Packet SPX ตั้งแต่ checksum จนถึง field source Socket จะมีความหมายเหมือนกันกับ Field ของ IPX packet ซึ่งจะมีส่วนที่ต่างกันดังนี้



ภาพที่ 11 SPX Packet Structure

Packet Type Field จะถูก Set เป็น 5 โดย SPX

Destination Node field อาจจะไม่บรรจุ broadcast address เฉพาะการเชื่อมต่อโดย SPX นี้จะทำการเชื่อมต่อโดย SPX นี้จะทำการเชื่อมต่อกันเป็นการสื่อสารการส่งข้อมูลแบบ Broadcasting ไม่มีใน SPX

Connection Control : field นี้จะประกอบไปด้วย 4 บิต เป็น flags โดย SPX

1h-8h (Undefined) - ค่านี้จะไม่ถูกกำหนด

10h (End-of-Message) -Client จะ Set flag นี้เพื่อเป็นการบอกว่าเป็นการสิ้นสุดของการเชื่อมต่อ

20h (Attention) -Client จะ Set Flag นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 40h (Acknowledgement Required) -SPX จะ Set bit นี้หากต้องการ Acknowledgement packet
- 80h (System Packet)- SPX จะ Set bit นี้ หาก Packet นี้เป็น System packet Clients จะไม่กำหนด Acknowledgement, System bits เพราะจะถูกใช้โดย SPX
- Datastream Type Field นี้จะเป็น 1-byte flag จะบ่งบอกชนิดของ Data ที่พบใน packet ซึ่งค่าที่เป็นไปได้มีดังนี้
- 0h - FDh (Client Defined) - SPX ไม่สนใจ field นี้
  - FEh (End-of-Connection) -เมื่อ client ต้องการจะยุติการเชื่อมต่อ SPX จะสร้าง End-of-Connection packet Packet นี้จะถูกส่งไปยังคู่ของการเชื่อมต่อเป็น messageสุดท้ายของการเชื่อมต่อ
  - FFh (End-of-Connection-Acknowledgment) -SPX จะสร้าง End-of-Packet-Acknowledgment โดยอัตโนมัติซึ่งมันจะเป็น system packet
  - Source Connection ID Field นี้จะประกอบด้วยการกำหนดหมายเลขโดยเชื่อมต่อถูกกำหนดโดย SPX
  - Destination Connection ID Field นี้จะประกอบด้วยการกำหนดหมายเลขการเชื่อมต่อถูกกำหนดโดย SPX นี้ใช้เมื่อ Demultiplexer Packets จากหลายการเชื่อมต่อที่มาถึงบน socket เดียวกัน
  - Sequence Number Field นี้จะเก็บจำนวนนับของ Packets ที่เปลี่ยนแปลงใน 1 ทิศทาง การเชื่อมต่อแต่ละด้านของการเชื่อมต่อจะเก็บจำนวน Packet ของมันและจำนวนนี้จะเปลี่ยนเป็น 0h หลังจากค่ามันถึง FFFFh SPX จัดการ field นี้
  - Acknowledge Number Field นี้ทำให้รู้ถึง Sequence Number ของ packet ถัดไปของ SPX ที่คาดว่าจะได้รับ packet ใด ๆ ที่มี Sequence Number น้อยกว่าที่ระบุไว้ใน Acknowledge Number field นี้จะถูกจัดการโดย SPX
  - Allocation Number Field นี้จะบ่งบอกถึงจำนวนของ buffer ของสถานีรับสถานีส่ง อาจ จะส่ง packet จนกระทั่งเท่ากับ Allocation Numbers

## บทที่ 4

### การเปรียบเทียบชุดโพรโตคอล TCP/IP กับ IPX/SPX

ในบทนี้จะทำการวิเคราะห์เปรียบเทียบชุดโพรโตคอล TCP/IP กับ IPX/SPX โดยสามารถจับคู่และเปรียบเทียบกันได้ในลักษณะต่างๆ ดังนี้

#### 4.1 เปรียบเทียบโพรโตคอล IP กับ IPX

ในส่วนของ TCP/IP นั้นจะมี IP เป็นโพรโตคอลในระดับ Network Layer เมื่อเทียบกับ OSI MODEL ส่วนทางด้าน IPX/SPX นั้นจะมี IPX ที่ทำงานในระดับ Network Layer เมื่อเทียบกับ OSI MODEL เช่นเดียวกันซึ่งทั้ง IP และ IPX นั้นจะมีความแตกต่างกันในส่วนของรูปแบบของแพคเกจ ข้อมูล ที่ใช้ส่งซึ่งในส่วนของ IP นั้นจะเรียกว่า Internet datagram โดยในส่วนของ IP HEADER นี้จะมีความยาวคงที่หากไม่ รวมส่วนของ IP OPTIONS จะมีขนาด 20 ไบต์ และจำนวนขนาดของ IP datagram นี้มีขนาด ได้สูงสุดรวมทั้งส่วนของข้อมูลและเฮดเดอร์ ด้วยนั้นมีขนาด 65,535 ไบต์ ในส่วนของ IPX แพคเกจ นั้นจะมีส่วนที่เป็นส่วนของเฮดเดอร์ มีขนาด 30 ไบต์ และในส่วนที่เป็นข้อมูลโดยเฉพาะ จะมีค่า ตั้งแต่ 0 ถึง 546 ไบต์ จากที่กล่าวมาข้างต้นพอจะเห็นแล้วว่าขนาดของแพคเกจ ของ IP นั้นถ้า ไม่รวม IP OPTIONS แล้วนั้นจะมีจำนวน เฮดเดอร์ ที่มีปริมาณน้อยกว่า IPX และที่สำคัญอีก ประการหนึ่งก็คือในส่วนของ แพคเกจ IP ที่เรียกว่า IP datagram นั้น จะมีส่วนของ ข้อมูลได้ มากกว่าของ IPX กล่าวคือจะมีข้อมูลได้ตั้งแต่ 0 ไบต์ จนกระทั่งถึง 65,515 ไบต์ ในกรณีที่ IP datagram ที่ไม่มี IP OPTIONS ในส่วนของ IPX นั้น ส่วนของข้อมูลจะมีได้ตั้งแต่ 0 ไบต์ ถึง 546 ไบต์เท่านั้นซึ่งนั่นก็หมายความว่า IP datagram นั้น สามารถมีขนาด แพคเกจสูงสุดเท่ากับ 65,535 ไบต์ ส่วนแพคเกจของ IPX นั้นจะมีขนาดของแพคเกจได้สูงสุด 576 ไบต์นั่นเอง

จากที่กล่าวมาข้างต้นสามารถนำมาหาค่า ของ format Utilization ได้ดังนี้

$$\text{IP format utilization} = \text{data size} / (\text{data size} + \text{header})$$

$$\text{ที่ DATA 0 ไบต์ utilization} = 0 / 20 = 0$$

$$\text{ที่ DATA 65515 ไบต์ utilization} = 65515 / 65535 = 0.99969$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนข้อมูล	Utilization	
	IPX	IP
0 byte	0.00000	0.00000
1 byte	0.03226	0.04762
3 bytes	0.09091	0.13043
5 bytes	0.14286	0.20000
10 bytes	0.25000	0.33333
20 bytes	0.40000	0.50000
30 bytes	0.50000	0.60000
50 bytes	0.62500	0.71429
100 bytes	0.76923	0.83333
150 bytes	0.83333	0.88235
200 bytes	0.86957	0.90909
250 bytes	0.89286	0.92593
300 bytes	0.90909	0.93750
350 bytes	0.92105	0.94595
400 bytes	0.93023	0.95238
546 bytes	0.94792	0.96466
1024 bytes		0.98084
2048 bytes		0.99033
3072 bytes		0.99353
4096 bytes		0.99514
8192 bytes		0.99756
10240 bytes		0.99805
20480 bytes		0.99902
40960 bytes		0.99951
65515 bytes		0.99969

ตารางที่ 4 Format Utilization ของ IP และ IPX

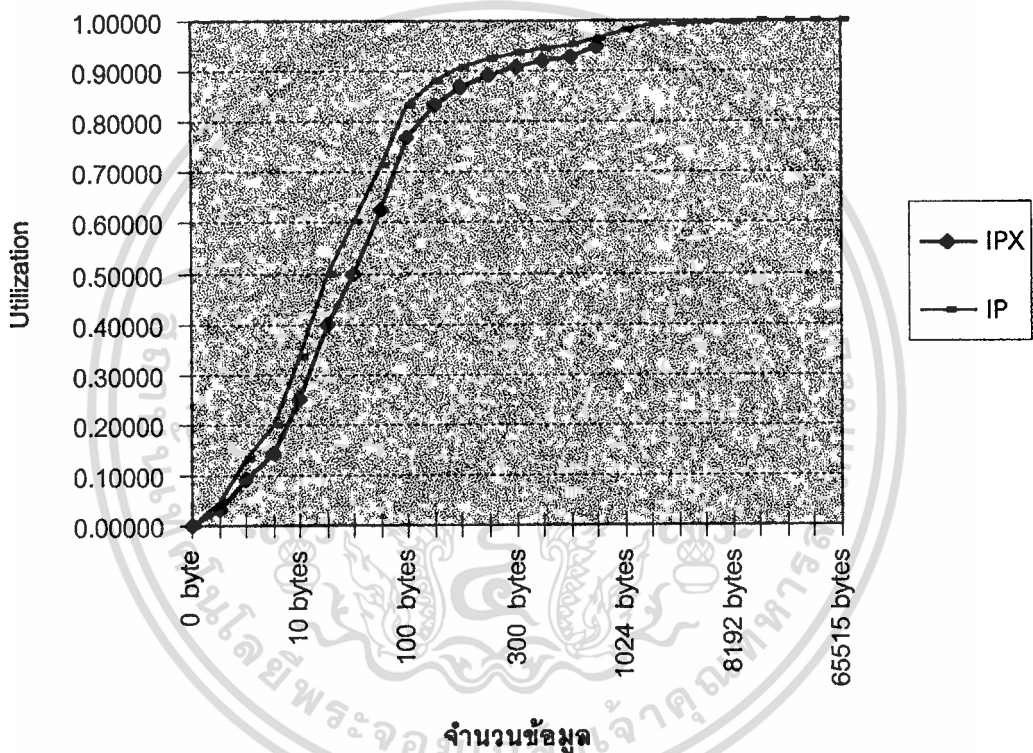
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IPX format utilization = data size / (data size + header)

ที่ DATA 0 ไบต์ utilization = 0/30 = 0

ที่ DATA 546 ไบต์ utilization = 546/576 = 0.94792

จากการคำนวณค่าของ format utilization ของทั้ง IP และ IPX ที่ค่าของข้อมูลที่ไม่เกินจำนวน 1 แพคเกจ ของข้อมูลที่เป็นไปได้แล้วนั้นจะมีค่าของจำนวนข้อมูลที่ส่ง ตามตารางที่ 4 และนำมาสร้างเป็นกราฟได้ดังภาพที่ 12



ภาพที่ 12 กราฟ Format Utilization ของ IP และ IPX

จากการพินิจภาพที่ 12 จะแสดงถึง Utilization ของ IP และ IPX นั้นค่า Utilization ที่ต่ำที่สุดของทั้ง IP และ IPX นั้นมีค่าเท่ากับศูนย์ ส่วนค่าที่สูงที่สุดของ IPX นั้นจะมีค่าเท่ากับ 0.94792 ทั้งนี้เนื่องจากจำนวนข้อมูลที่ส่งได้สูงสุดแค่ 546 ไบต์สำหรับในส่วนของ IP นั้นค่า Utilization สูงสุดจะมีค่าเท่ากับ 0.99969 เพราะในหนึ่งแพคเกจของ IP นั้นจะมีข้อมูลได้สูงสุด 65,515 ไบต์ ซึ่งอัตราส่วนระหว่างจำนวนข้อมูลที่ส่งได้สูงสุด ในหนึ่งแพคเกจจะหารด้วยจำนวนสูงสุดของแพคเกจ ซึ่งจะมีค่าใกล้เคียงกันมากค่าที่ได้จะมีค่าเข้าใกล้ 1 มากขึ้น ส่วนค่า Utilization ที่การส่งข้อมูลจำนวนเท่ากันนั้น IP จะมีค่า Utilization ที่สูงกว่า

ข้อพิจารณาอีกข้อหนึ่งที่สำคัญของการเปรียบเทียบ IP กับ IPX นี้ก็คือทั้ง IP และ IPX นี้ ต่างก็เป็นโปรโตคอล ที่ต้องอาศัยโปรโตคอลที่ใช้ในการเราท์แพคเกจข้อมูล ซึ่งในที่นี้ก็คือ RIP (routing information protocol) ซึ่งจะเป็นการเราท์แพคเกจข้อมูล ที่ใช้ หลักการของเวกเตอร์ คือมีทั้งทิศทางและระยะทาง ซึ่งเราท์เตอร์ที่ใช้โปรโตคอล RIP จะเก็บ ตารางของระยะทางซึ่งก็คือ จำนวน hop ที่วิ่งผ่านไปยังเราท์เตอร์อื่นตามที่ได้รับรายงานจาก เราท์เตอร์ ที่อยู่ใกล้เคียง การทำงานของ RIP ในระบบของ Novell คือที่ ใช้กับ IPX ตัวเราท์เตอร์จะ broadcast ตารางที่อัปเดต ใหม่ออกไปบนเน็ตเวิร์กทุกๆ 30 วินาที ไม่ว่าจะมีการ เปลี่ยนแปลงทางโทโพโลยีของเน็ตเวิร์ก หรือไม่ก็ตาม การอัปเดตข้อมูลนี้เป็นการกินแบนด์วิธ ของระบบ WAN เป็นอย่างมากและ เราท์เตอร์ RIP ก็ตอบสนองต่อการเปลี่ยนแปลงโทโพโลยี ได้ช้ามาก ที่แย่ไปกว่านั้นก็คือถ้าการ เชื่อมต่อไปยังเราท์เตอร์เกิดผิดพลาด เราท์เตอร์ที่มีเส้นทาง ไปยังส่วนที่ผิดพลาดได้หลายทางก็จะ สับสนและเริ่มที่จะเพิ่มช่วง hop อย่างผิดๆดังนั้น Novell จึงมีการจำกัดรัศมีของ IPX บนระบบ เน็ตเวิร์กได้แค่ 15 hop เท่านั้นในส่วนของ IP นั้นนอกจากจะมี RIP ที่เป็นโปรโตคอลในการ เราท์แพคเกจแล้วยังมีโปรโตคอลในการเราท์แพคเกจ OSPF (open shortest path first) ซึ่งมัน จะใช้เทคโนโลยีของสถานะการเชื่อมต่อที่ เก่งกว่าระบบเวกเตอร์ ทั้งนี้เพราะเราท์เตอร์ที่ใช้สถานะ การเชื่อมต่อจะส่งแผนที่สมบูรณ์ของอุปกรณ์ใกล้เคียงออกไปตามช่วงเวลาที่กำหนดและจะเก็บ แผนที่ ที่มันจะได้รับจากเราท์เตอร์ตัวอื่นเอาไว้ มันจะรู้จัก ระบบเน็ตเวิร์กมากกว่าเราท์เตอร์ที่ใช้ เวกเตอร์ และสามารถปรับตัวให้เข้ากับการเปลี่ยนแปลง ได้เร็ว กว่าทั้งยังคงหาเส้นทางสำรองได้ เมื่อมีการเชื่อมต่อหลักมีปัญหาเนื่องจากการอัปเดตข้อมูลจะเกิดขึ้นเมื่อมีการเปลี่ยนแปลงเท่านั้น โอเวอร์เฮดที่เกิดจากการส่งอัปเดตแผนที่จะลดลงไปมาก เราท์เตอร์ที่ใช้สถานะการเชื่อมต่อจะ เห็นภาพรวมของระบบไม่ใช่รับรู้เพียงรายงานโทโพโลยีที่ได้ จากเราท์เตอร์ที่อยู่ข้างเคียงเท่านั้น ปัญหาการนับจนถึงค่าอนันต์จะไม่เกิดขึ้น

ข้อแตกต่างอีกประการหนึ่งก็คือการอ้างแอดเดรสของทั้งสองโปรโตคอลที่แตกต่างกัน คือ IP นั้นจะระบุแอดเดรสต้นทางและแอดเดรสปลายทางเป็นลักษณะของลจิคคอลแอดเดรสตาม ที่ ได้อธิบายไว้แล้วในเรื่อง IP แอดเดรส ซึ่งในการที่จะติดต่อกับแอดเดรสปลายทางได้จริงนั้นต้อง อาศัยส่วนที่ทำหน้าที่ในการแยกแยะอินเตอร์เน็ตแอดเดรสซึ่งก็คือ ARP (Address Resolution Protocol) แต่ในส่วนของ IPX นั้นจะใช้หมายเลขที่อยู่ทีการ์ด อีเทอร์เน็ต หรือ โทกเคนริง ได้เลย ซึ่งไม่มีโอกาสที่ซ้ำกันอยู่แล้วเป็นแอดเดรสของมันเลย ในระบบ IPX นี้การกำหนดแอดเดรสของ เครื่องแต่ละเครื่องจะไม่มีปัญหาเลย คุณสมบัติที่ดีอีกอย่างหนึ่งของ IPX ก็คือมันจะมีขนาด แอดเดรส ที่กว้างใหญ่กว่าของ IP แอดเดรส คือแอดเดรสของ IPX ประกอบด้วยหมายเลข เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เน็ตเวิร์ก 4 ไบต์ กับหมายเลขโหนดอีก 6 ไบต์ ในขณะที่ IP นั้นจะมีหมายเลขเน็ตเวิร์ก และหมายเลขโหนดอยู่รวมกันเพียง 4 ไบต์ ตามที่ได้อธิบายไปแล้ว

## 4.2 เปรียบเทียบโปรโตคอล TCP กับ SPX

ในการทำงานของโปรโตคอล TCP/IP และ IPX/SPX ต่างก็มีส่วนที่เป็นโปรโตคอลในรูปแบบของ Connection-Oriented ด้วยกันทั้งคู่คือทางชุดโปรโตคอล TCP/IP จะมีโปรโตคอล TCP ที่ทำงานในชั้นของ Transport Layer เมื่อเทียบกับ OSI MODEL ส่วนทางด้าน IPX/SPX ก็จะมี SPX ซึ่งทำงานในชั้นของ Transport Layer เมื่อเทียบกับ OSI MODEL เช่นเดียวกันซึ่งในการที่จะทำการส่งข้อมูลก่อนซึ่งการทำงานในลักษณะนี้จะสามารถรับประกันความถูกต้องของข้อมูล และมีความน่าเชื่อถือและขณะส่งข้อมูลก็จะมีกลไก ในการตรวจสอบความถูกต้องตลอดเวลา ที่นี้เราลองพิจารณาโปรโตคอลทั้งสองในส่วนต่างๆดังนี้

### 4.2.1 เปรียบเทียบแพ็คเกจข้อมูล ระหว่าง TCP และ SPX

รูปแบบของแพ็คเกจข้อมูลในการส่งข้อมูลถ้าเป็นแพ็คเกจข้อมูลของ TCP นั้นเราจะเรียกแพ็คเกจของข้อมูลว่า Segment ซึ่งในหนึ่งแพ็คเกจข้อมูลของ TCP นั้นจะมีส่วนเฮดเดอร์ซึ่งรวมกับเฮดเดอร์ของ IP ด้วยแล้วจะมีขนาดน้อยที่สุดที่เป็นไปได้ก็คือ 40 ไบต์ และส่วนของข้อมูลจะมีได้ไม่เกิน 65,495 ไบต์ นั่นก็คือจะมีทั้งเฮดเดอร์และข้อมูลรวมกันได้ไม่เกิน 65,535 ไบต์ นั้นเอง ส่วนทางด้าน SPX นั้นแพ็คเกจของข้อมูลจะมีขนาดมากที่สุด 576 ไบต์ เช่นกันแต่จะเป็นส่วนของเฮดเดอร์ 42 ไบต์ ดังนั้นจึงเป็นส่วนของข้อมูลจริงๆ 534 ไบต์ ดังนั้นจึงเห็นได้ว่าทั้ง TCP และ SPX นั้น มีจำนวนขนาดของแพ็คเกจข้อมูลไม่เท่ากันและจะต่างกันที่ส่วนของเฮดเดอร์ด้วย ซึ่ง TCP นั้นมี 40 ไบต์ ส่วน SPX นั้นมี 42 ไบต์ และในส่วนของข้อมูลนั้น TCP มีได้ถึง 65,495 ไบต์ ในขณะที่ SPX มีข้อมูลสูงสุดได้เพียง 534 ไบต์ ซึ่งในหัวข้อถัดๆไปจากนี้จะแสดงให้เห็นถึงประสิทธิภาพของการส่งข้อมูลกับจำนวนของ Overhead ของการส่งแพ็คเกจข้อมูลของทั้ง TCP และ SPX ต่อไป ซึ่งก่อนที่จะถึงส่วนที่จะทดลองการจำลองการส่งข้อมูลในปริมาณต่างๆ จะขอหาจำนวนของ Format Utilization ของ TCP และ SPX โดยจะคิดค่าดังกล่าวดังนี้

$$\text{TCP format utilization} = \text{data size} / (\text{data size} + \text{header})$$

$$\text{ที่ DATA 0 ไบต์ utilization} = 0 / 40 = 0$$

$$\text{ที่ DATA 65495 ไบต์ utilization} = 65495 / 65535 = 0.99939$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนข้อมูล	Utilization	
	SPX	TCP
0 byte	0	0
1 byte	0.023256	0.02439
3 bytes	0.066667	0.069767
5 bytes	0.106383	0.111111
10 bytes	0.192308	0.2
20 bytes	0.322581	0.333333
30 bytes	0.416667	0.428571
50 bytes	0.543478	0.555556
100 bytes	0.704225	0.714286
150 bytes	0.78125	0.789474
200 bytes	0.826446	0.833333
250 bytes	0.856164	0.862069
300 bytes	0.877193	0.882353
350 bytes	0.892857	0.897436
400 bytes	0.904977	0.909091
534 bytes	0.927083	0.930314
1024 bytes		0.962406
2048 bytes		0.980843
3072 bytes		0.987147
4096 bytes		0.990329
8192 bytes		0.995141
10240 bytes		0.996109
20480 bytes		0.998051
40960 bytes		0.999024
65495 bytes		0.99939

ตารางที่ 5 Format Utilization ของ TCP และ SPX

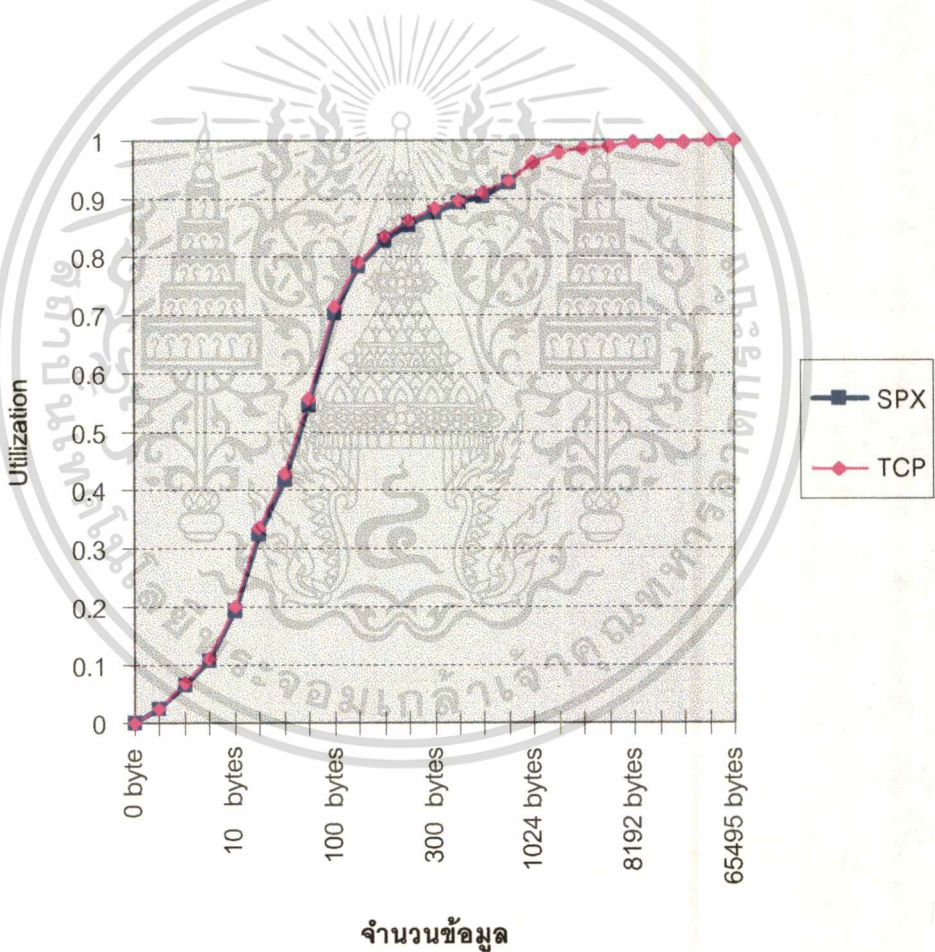
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPX format utilization = data size / (data size + header)

ที่ DATA 0 ไบต์ utilization = 0/42 = 0

ที่ DATA 534 ไบต์ utilization = 534/576 = 0.927083

จากการคำนวณค่าของ format utilization ของทั้ง TCP และ SPX ที่ค่าของข้อมูลที่ไม่เกินจำนวน 1 แพคเกจ ของข้อมูลที่เป็นไปได้แล้วนั้นจะมีค่าของจำนวนข้อมูลที่ส่ง ตามตารางที่ 5 และนำมาสร้างเป็นการฟีดดังภาพที่ 13



ภาพที่ 13 กราฟ Format Utilization ของ IP และ IPX

จากกราฟในภาพที่ 13 และข้อมูลในตารางที่ 5 จะเห็นได้ว่า Utilization ของแพคเกจ TCP และ SPX นั้นจะมีค่าใกล้เคียงกันมากซึ่งค่าของ Utilization นี้จะมีค่าสูงขึ้นในกรณีที่จำนวนข้อมูล เอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ต้องการส่งมีค่าเพิ่มมากขึ้นโดยในส่วนของ SPX นั้นจะมีส่วนของข้อมูลที่จะส่งได้สูงสุดในหนึ่งแพ็คเกจนั้นมีได้ 534 ไบต์ ทั้งนี้เนื่องจากขนาดแพ็คเกจของ SPX นั้นมีค่าได้สูงสุดเพียง 576 ไบต์เท่านั้น แต่สำหรับ TCP นั้นในหนึ่งแพ็คเกจสามารถส่งจำนวนข้อมูลได้สูงสุดถึง 65,495 ไบต์ และสามารถมีขนาดแพ็คเกจสูงสุดได้ 65,535 ไบต์ ซึ่งค่าของ Utilization ของ SPX นั้นมีค่าได้สูงสุด 0.947 แต่สำหรับ Utilization ของ TCP นั้นสามารถมีค่าได้สูงสุดถึง 0.999 ซึ่งมีค่าใกล้เคียงกันมากทั้งนี้เนื่องจากจำนวนข้อมูลที่ต้องการส่งนั้นมีค่าใกล้เคียงกับจำนวนข้อมูลที่ส่งจริง

#### 4.2.2 การเปรียบเทียบประสิทธิภาพของการส่งข้อมูลในปริมาณต่างๆของ TCP และ SPX

ทั้ง TCP และ SPX เป็นโปรโตคอลที่เป็น Connection-Oriented เหมือนกันในการส่งผ่านข้อมูลจะต้องมีการสถาปนากการเชื่อมต่อและเมื่อสิ้นสุดการส่งข้อมูลก็จะต้องมีการยุติการเชื่อมต่อเช่นเดียวกัน ซึ่งในส่วนของ TCP นี้ จะใช้หลักการของ three-way handshake(Douglas E. Comer 1991:194-195) ทั้งในส่วนของ การสถาปนากการเชื่อมต่อและการยุติการเชื่อมต่อ ซึ่งเงื่อนไขของการเปรียบเทียบครั้งนี้จะตั้งสมมุติฐานว่าในส่วนของ SPX นั้นก็จะใช้วิธีการสถาปนากการเชื่อมต่อและการยุติการเชื่อมต่อแบบ three-way handshake เช่นกัน ซึ่งทั้ง TCP และ SPX นั้นทำการติดต่อโดยด้านต้นทางที่ ทำการ active open นั้นมี แพ็คเกจข้อมูลเพิ่มขึ้นมาอีก 2 แพ็คเกจ และเมื่อยุติการเชื่อมโยงก็จะมีแพ็คเกจในการนี้เพิ่มอีก 2 แพ็คเกจ ดังนั้นจึงจะมีข้อมูลเพิ่มขึ้นมาอีกทั้งสิ้น 4 แพ็คเกจข้อมูลเพิ่มมาจากจำนวนข้อมูลที่จะ ส่งจริง จากนั้นเราจะมาดูจำนวนข้อมูลที่เราจะใช้ในการทดสอบและเปรียบเทียบจำนวนข้อมูลที่ ส่งจริงกับจำนวนไอบเวอร์เฮด ที่เกิดขึ้นที่ปริมาณข้อมูลต่าง ๆ ซึ่งจะทดสอบที่ระดับของ MTU(Maximum Transfer unit) ต่างๆ กัน ตั้งแต่ ค่า MTU 128 ไบต์ จนกระทั่งถึง 4096 ไบต์ โดยที่ MTU นั้นคือจำนวนข้อมูลสูงสุดที่สามารถจะส่งข้อมูลไปได้ในหนึ่งฟิสิคอลลเฟรม โดยรวมกันทั้งจำนวนเฮดเดอร์และจำนวนข้อมูลที่ต้องการส่ง ซึ่งขึ้นอยู่กับชนิดของฮาร์ดแวร์ที่ใช้ เช่น Ethernet สามารถส่งได้ถึง 1,500 ไบต์ ในการเปรียบเทียบครั้งนี้จะทำการเปรียบเทียบจำนวนข้อมูลของทั้ง TCP และ SPX ที่ MTU ต่าง ๆ กัน โดยที่ MTU ต่างกันนั้นจำนวนข้อมูลที่ส่งได้ในหนึ่งแพ็คเกจก็จะต่างกันในแต่ละ MTU จากนั้นจะนำจำนวนข้อมูลที่ต้องการส่งที่จำนวนต่าง ๆ กันนั้นมาคิดคำนวณหาจำนวนแพ็คเกจที่ต้องส่งจริงโดยคิดจากสูตร

แพ็คเกจที่ต้องการส่ง =  $\lfloor \text{จำนวนข้อมูลที่ต้องการส่ง} / \text{จำนวนข้อมูลสูงสุดที่ส่งได้ใน MTU ต่างๆ} \rfloor$

แพ็คเกจที่ต้องส่งจริง = แพ็คเกจที่ต้องการส่ง + 4

จากนั้นก็นำจำนวนแพ็คเกจที่ต้องส่งจริงมาคูณกับจำนวนของเฮดเดอร์ของ TCP หรือเฮดเดอร์ SPX ก็จะได้ออกมาเป็นจำนวน เฮดเดอร์ทั้งหมดที่ใช้ในการส่งข้อมูล ซึ่งหลังจากเราได้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนเฮดเดอร์ทั้งหมดมาแล้วเราก็นำจำนวนที่ได้มารวมกับจำนวนข้อมูลที่ต้องการส่ง ก็จะได้จำนวนข้อมูลที่ส่งจริงซึ่งจะนำมาคิดเปอร์เซ็นต์ประสิทธิภาพ ดังสูตรต่อไปนี้

$$\text{เปอร์เซ็นต์ประสิทธิภาพ} = (\text{ข้อมูลที่ต้องการส่ง} / \text{ข้อมูลที่ส่งจริง}) * 100$$

เปรียบเทียบ การส่งข้อมูลที่ MTU 128 bytes

TCP จะเป็นข้อมูล 88 bytes และเป็น header 40 bytes

พิจารณาการส่งข้อมูลที่ 512 bytes เป็นข้อมูลที่ส่ง =  $512 / 88 = 6$  Packet

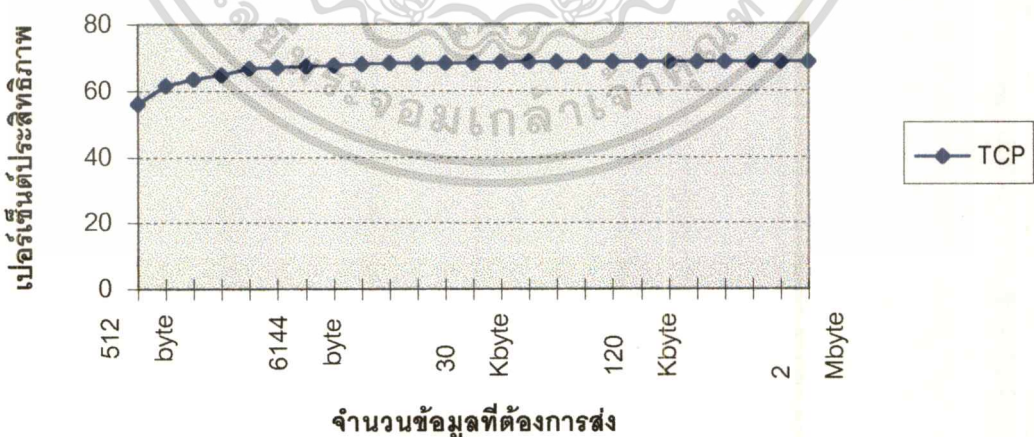
แต่ต้องใช้ในการเริ่มและเลิกการส่งข้อมูล 4 Packet ดังนั้น จะเป็นข้อมูลที่ส่งทั้งสิ้น 10 Packet

คิดเป็นจำนวน ข้อมูลที่ส่งจริง =  $400 + 512 = 912$  bytes

คิดเป็นเปอร์เซ็นต์ประสิทธิภาพ =  $(\text{ข้อมูลที่ต้องการส่ง} / \text{ข้อมูลที่ส่งจริง}) * 100$

$$= 512 / 912 * 100 = 56.14 \%$$

และใช้วิธีการคิดในลักษณะเดียวกันนี้กับข้อมูลที่ส่งจำนวนต่างกันจนถึงค่า 2 Mbyte จะได้ค่าของเปอร์เซ็นต์ประสิทธิภาพ ได้ดัง ตารางที่ 6 และได้กราฟดังภาพที่ 14



ภาพที่ 14 กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 128

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนข้อมูลที่ต้องการส่ง (bytes)	จำนวนแพคเกจ	จำนวนแพคเกจรวมที่ใช้จริง	จำนวน header	จำนวนข้อมูลที่ส่งจริง (bytes)	เปอร์เซ็นต์ประสิทธิภาพ
512 byte	6	10	400	912	56.14035
1024byte	12	16	640	1664	61.53846
1536 byte	18	22	880	2416	63.57616
2048 byte	24	28	1120	3168	64.64646
4096 byte	47	51	2040	6136	66.75359
5120 byte	59	63	2520	7640	67.01571
6144 byte	70	74	2960	9104	67.48682
8192 byte	94	98	3920	12112	67.6354
10 Kbyte	117	121	4840	15080	67.90451
15 Kbyte	175	179	7160	22520	68.20604
20 Kbyte	233	237	9480	29960	68.35781
25 Kbyte	291	295	11800	37400	68.4492
30 Kbyte	350	354	14160	44880	68.4492
40 Kbyte	466	470	18800	59760	68.54083
50 Kbyte	582	586	23440	74640	68.59593
60 Kbyte	699	703	28120	89560	68.60205
80 Kbyte	931	935	37400	119320	68.65572
100 Kbyte	1164	1168	46720	149120	68.66953
120 Kbyte	1397	1401	56040	178920	68.67874
150 Kbyte	1746	1750	70000	223600	68.6941
200 Kbyte	2328	2332	93280	298080	68.70639
400 Kbyte	4655	4659	186360	595960	68.72944
700 Kbyte	8146	8150	326000	1042800	68.73801
1 Mbyte	11916	11920	476800	1525376	68.74213
2 Mbyte	23832	23836	953440	3050592	68.74574

เอกสารนี้เป็นเอกสารที่ตารางที่ 6 การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 128 ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ MTU = 128 bytes SPX จะเป็นข้อมูล 86 bytes และเป็น header 42 bytes

พิจารณาการส่งข้อมูลที่ 512 bytes เป็นข้อมูลที่ส่ง =  $512 / 86 = 6$  Packet

แต่ต้องใช้ในการเริ่มและเลิกการส่งข้อมูล 4 Packet ดังนั้นจะเป็นข้อมูลที่ส่งทั้งสิ้น 10 Packet

คิดเป็นจำนวน ข้อมูลที่ส่งจริง =  $420 + 512 = 932$  bytes

คิดเป็นเปอร์เซ็นต์ประสิทธิภาพ =  $(\text{ข้อมูลที่ต้องการส่ง} / \text{ข้อมูลที่ส่งจริง}) * 100$

$$= 512 / 932 * 100 = 54.936 \%$$

และใช้วิธีการคิดในลักษณะเดียวกันนี้กับข้อมูลที่ส่งจำนวนต่างกันจนถึงค่า 2 Mbyte จะ

ได้ค่าของเปอร์เซ็นต์ประสิทธิภาพ ได้ดัง ตารางที่ 7 และได้กราฟดังภาพที่ 15



ภาพที่ 15 กราฟการส่งข้อมูลปริมาณต่างๆ โดย SPX ที่ MTU = 128

จำนวนข้อมูลที่ต้องการ ส่ง (bytes)	จำนวนแพคเกจ	จำนวนแพคเกจ รวมที่ใช้จริง	จำนวน header	จำนวนข้อมูลที่ส่ง จริง (bytes)	เปอร์เซ็นต์ ประสิทธิภาพ
512 byte	6	10	420	932	54.93562
1024byte	12	16	672	1696	60.37736
1536 byte	18	22	924	2460	62.43902
2048 byte	24	28	1176	3224	63.52357
4096 byte	48	52	2184	6280	65.22293
5120 byte	60	64	2688	7808	65.57377
6144 byte	72	76	3192	9336	65.80977
8192 byte	96	100	4200	12392	66.10717
10 Kbyte	120	124	5208	15448	66.2869
15 Kbyte	179	183	7686	23046	66.64931
20 Kbyte	239	243	10206	30686	66.74053
25 Kbyte	298	302	12684	38284	66.86867
30 Kbyte	358	362	15204	45924	66.89313
40 Kbyte	477	481	20202	61162	66.96969
50 Kbyte	596	600	25200	76400	67.01571
60 Kbyte	715	719	30198	91638	67.04642
80 Kbyte	953	957	40194	122114	67.08486
100 Kbyte	1191	1195	50190	152590	67.10794
120 Kbyte	1429	1433	60186	183066	67.12333
150 Kbyte	1787	1791	75222	228822	67.12641
200 Kbyte	2382	2386	100212	305012	67.1449
400 Kbyte	4763	4767	200214	609814	67.16802
700 Kbyte	8335	8339	350238	1067038	67.17661
1 Mbyte	12193	12197	512274	1560850	67.17981
2 Mbyte	24386	24390	1024380	3121532	67.18342

ตารางที่ 7 การส่งข้อมูลปริมาณต่างๆ โดย SPX ที่ MTU = 128

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ MTU เท่ากับ 576 ไบต์ จะทำการทดสอบข้อมูลโดยใช้ SPX ก่อน ซึ่งจะมีแพ็คเกจขนาด 576 ไบต์และจะเป็นส่วนของ ข้อมูลได้ 534 ไบต์

SPX จะเป็นข้อมูล 534 bytes และเป็น header 42 bytes

พิจารณาการส่งข้อมูลที่ 512 bytes เป็นข้อมูลที่ส่ง =  $512 / 534 = 1$  Packet

แต่ต้องใช้ในการเริ่มและเลิกการส่งข้อมูล 4 Packet ดังนั้นจะเป็นข้อมูลที่ส่งทั้งสิ้น 5 Packet

คิดเป็นจำนวน ข้อมูลที่ส่งจริง =  $210 + 512 = 722$  bytes

คิดเป็นเปอร์เซ็นต์ประสิทธิภาพ =  $(\text{ข้อมูลที่ต้องการส่ง} / \text{ข้อมูลที่ส่งจริง}) * 100$

$$= 512 / 722 * 100 = 70.914\%$$

และใช้วิธีการคิดในลักษณะเดียวกันนี้กับข้อมูลที่ส่งจำนวนต่างกันจนถึงค่า 2 Mbyte จะได้ค่าของเปอร์เซ็นต์ประสิทธิภาพ ได้ดัง ตารางที่ 8 และได้กราฟดังภาพที่ 16



ภาพที่ 16 กราฟการส่งข้อมูลปริมาณต่างๆ โดย SPX ที่ MTU = 576

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนข้อมูลที่ต้องการ ส่ง (bytes)	จำนวนแพคเก็ต	จำนวนแพคเก็ต รวมที่ใช้จริง	จำนวน header	จำนวนข้อมูลที่ ส่งจริง (bytes)	เปอร์เซ็นต์ ประสิทธิภาพ
512	1	5	210	722	70.91413
1024	2	6	252	1276	80.25078
1536	3	7	294	1830	83.93443
2048	4	8	336	2384	85.90604
4096	8	12	504	4600	89.04348
5120	10	14	588	5708	89.69867
6144	12	16	672	6816	90.14085
8192	16	20	840	9032	90.69973
10240	20	24	1008	11248	91.03841
15360	29	33	1386	16746	91.7234
20480	39	43	1806	22286	91.89626
25600	48	52	2184	27784	92.13936
30720	58	62	2604	33324	92.18581
40960	77	81	3402	44362	92.33127
51200	96	100	4200	55400	92.41877
61440	116	120	5040	66480	92.41877
81920	154	158	6636	88556	92.50644
102400	192	196	8232	110632	92.55911
122880	231	235	9870	132750	92.56497
153600	288	292	12264	165864	92.60599
204800	384	388	16296	221096	92.62945
409600	768	772	32424	442024	92.66465
716800	1343	1347	56574	773374	92.68478
1048576	1964	1968	82656	1131232	92.69328
2097152	3928	3932	165144	2262296	92.70016

ตารางที่ 8 การส่งข้อมูลปริมาณต่างๆ โดย SPX ที่ MTU = 576

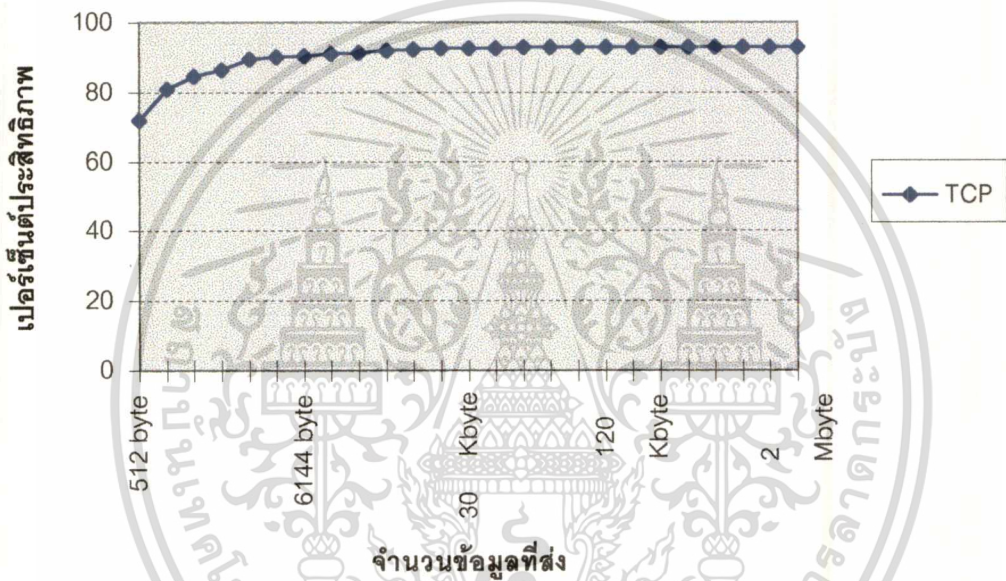
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนี้ไปจะเป็นการทดลองส่งข้อมูลปริมาณต่างๆโดยใช้ TCP ที่ MTU มีค่าเท่ากับ 576 ไบต์ จะเป็นจำนวนข้อมูล 536 ไบต์ต่อ 1 แพคเก็ต มีค่าปรากฏประสิทธิภาพดัง ตารางที่ 9 และ ภาพที่ 17

จำนวนข้อมูลที่ต้องการส่ง (bytes)	จำนวนแพคเก็ต	จำนวนแพคเก็ตรวมที่ใช้จริง	จำนวน header	จำนวนข้อมูลที่ส่งจริง (bytes)	เปอร์เซ็นต์ประสิทธิภาพ
512	1	5	200	712	71.91011
1024	2	6	240	1264	81.01266
1536	3	7	280	1816	84.5815
2048	4	8	320	2368	86.48649
4096	8	12	480	4576	89.51049
5120	10	14	560	5680	90.14085
6144	12	16	640	6784	90.56604
8192	16	20	800	8992	91.1032
10240	20	24	960	11200	91.42857
15360	29	33	1320	16680	92.08633
20480	39	43	1720	22200	92.25225
25600	48	52	2080	27680	92.48555
30720	58	62	2480	33200	92.53012
40960	77	81	3240	44200	92.66968
51200	96	100	4000	55200	92.75362
61440	115	119	4760	66200	92.80967
81920	153	157	6280	88200	92.87982
102400	192	196	7840	110240	92.88824
122880	230	234	9360	132240	92.92196
153600	287	291	11640	165240	92.9557
204800	383	387	15480	220280	92.97258
409600	765	769	30760	440360	93.01481
716800	1338	1342	53680	770480	93.03291
1048576	1957	1961	78440	1127016	93.04003
2097152	3913	3917	156680	2253832	93.04828

ตารางที่ 9 การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 576

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 17 กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 576

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนี้ไปจะเป็นการทดลองส่งข้อมูลปริมาณต่างๆโดยใช้ TCP ที่ MTU มีค่าเท่ากับ 1024 ไบต์ จะเป็นจำนวนข้อมูล 984 ไบต์ต่อ 1 แพคเกจ มีค่าปรากฏประสิทธิภาพดัง ตารางที่ 10 และ ภาพที่ 18

จำนวนข้อมูลที่ต้องการส่ง (bytes)	จำนวนแพคเกจ	จำนวนแพคเกจรวมที่ใช้จริง	จำนวน header	จำนวนข้อมูลที่ส่งจริง (bytes)	เปอร์เซ็นต์ประสิทธิภาพ
512	1	5	200	712	71.91011
1024	2	6	240	1264	81.01266
1536	2	6	240	1776	86.48649
2048	3	7	280	2328	87.97251
4096	5	9	360	4456	91.92101
5120	6	10	400	5520	92.75362
6144	7	11	440	6584	93.31713
8192	9	13	520	8712	94.03122
10240	11	15	600	10840	94.46494
15360	16	20	800	16160	95.0495
20480	21	25	1000	21480	95.34451
25600	27	31	1240	26840	95.38003
30720	32	36	1440	32160	95.52239
40960	42	46	1840	42800	95.70093
51200	53	57	2280	53480	95.73672
61440	63	67	2680	64120	95.82034
81920	84	88	3520	85440	95.88015
102400	105	109	4360	106760	95.91607
122880	125	129	5160	128040	95.97001
153600	157	161	6440	160040	95.97601
204800	209	213	8520	213320	96.006
409600	417	421	16840	426440	96.05103
716800	729	733	29320	746120	96.07034
1048576	1066	1070	42800	1091376	96.07835
2097152	2132	2136	85440	2182592	96.08539

ตารางที่ 10 การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 1024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 18 กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 1024

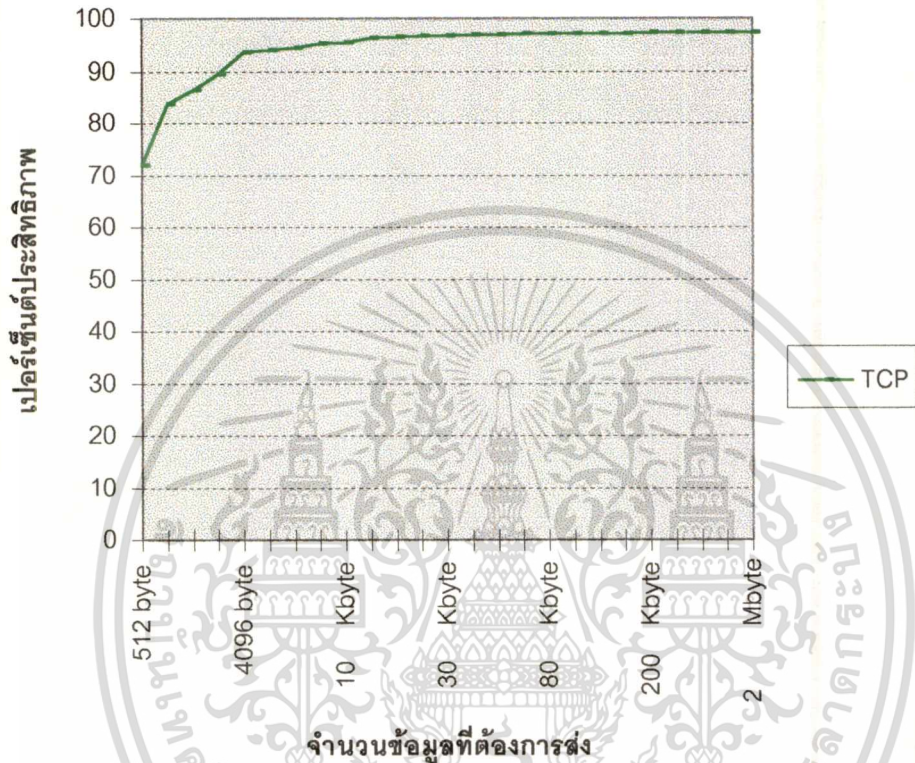
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนี้ไปจะเป็นการทดลองส่งข้อมูลปริมาณต่างๆโดยใช้ TCP ที่ MTU มีค่าเท่ากับ 1500 ไบต์ จะเป็นจำนวนข้อมูล 1460 ไบต์ต่อ 1 แพคเก็ต มีค่าปรากฏประสิทธิภาพดัง ตารางที่ 11 และภาพที่ 19

จำนวนข้อมูลที่ต้องการส่ง (bytes)	จำนวนแพคเก็ต	จำนวนแพคเก็ตรวมที่ใช้จริง	จำนวน header	จำนวนข้อมูลที่ส่งจริง (bytes)	เปอร์เซ็นต์ประสิทธิภาพ
512	1	5	200	712	71.91011
1024	1	5	200	1224	83.66013
1536	2	6	240	1776	86.48649
2048	2	6	240	2288	89.51049
4096	3	7	280	4376	93.60146
5120	4	8	320	5440	94.11765
6144	5	9	360	6504	94.46494
8192	6	10	400	8592	95.34451
10240	8	12	480	10720	95.52239
15360	11	15	600	15960	96.2406
20480	15	19	760	21240	96.42185
25600	18	22	880	26480	96.67674
30720	22	26	1040	31760	96.72544
40960	29	33	1320	42280	96.87796
51200	36	40	1600	52800	96.9697
61440	43	47	1880	63320	97.03095
81920	57	61	2440	84360	97.10763
102400	71	75	3000	105400	97.1537
122880	85	89	3560	126440	97.18444
153600	106	110	4400	158000	97.21519
204800	141	145	5800	210600	97.24596
409600	281	285	11400	421000	97.29216
716800	491	495	19800	736600	97.31197
1048576	719	723	28920	1077496	97.316
2097152	1437	1441	57640	2154792	97.32503

ตารางที่ 11 การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 1500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 19 กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 1500

จากนี้ไปจะเป็นการทดลองส่งข้อมูลปริมาณต่างๆโดยใช้ TCP ที่ MTU มีค่าเท่ากับ 4096 ไบต์ จะเป็นจำนวนข้อมูล 4056 ไบต์ต่อ 1 แพคเก็ต มีค่าปรากฏประสิทธิภาพดัง ตารางที่ 12 และภาพที่ 20

จำนวนข้อมูลที่ต้องการส่ง (bytes)	จำนวนแพคเก็ต	จำนวนแพคเก็ตรวมที่ใช้จริง	จำนวน header	จำนวนข้อมูลที่ส่งจริง (bytes)	เปอร์เซ็นต์ประสิทธิภาพ
512	1	5	200	712	71.91011
1024	1	5	200	1224	83.66013
1536	1	5	200	1736	88.47926
2048	1	5	200	2248	91.1032
4096	2	6	240	4336	94.46494
5120	2	6	240	5360	95.52239
6144	2	6	240	6384	96.2406
8192	3	7	280	8472	96.695
10240	3	7	280	10520	97.3384
15360	4	8	320	15680	97.95918
20480	6	10	400	20880	98.08429
25600	7	11	440	26040	98.31029
30720	8	12	480	31200	98.46154
40960	11	15	600	41560	98.5563
51200	13	17	680	51880	98.68928
61440	16	20	800	62240	98.71465
81920	21	25	1000	82920	98.79402
102400	26	30	1200	103600	98.8417
122880	31	35	1400	124280	98.87351
153600	38	42	1680	155280	98.91808
204800	51	55	2200	207000	98.9372
409600	101	105	4200	413800	98.98502
716800	177	181	7240	724040	99.00006
1048576	259	263	10520	1059096	99.0067
2097152	518	522	20880	2118032	99.01418

ตารางที่ 12 การส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 4096

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

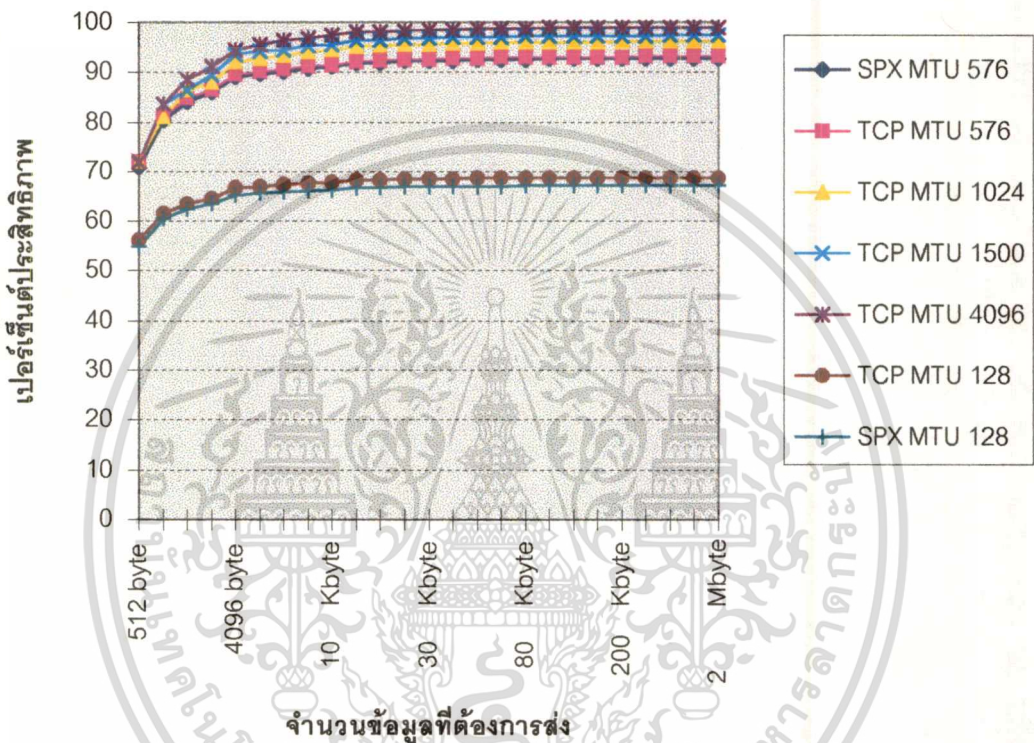
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 20 กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP ที่ MTU = 4096

จากการทดลองส่งข้อมูลในปริมาณต่างๆ โดยใช้ MTU ที่แตกต่างกันจากน้อยไปหามาก นั้นจะได้ผลออกมาตามตารางและภาพที่แสดงมาข้างต้น จากตารางที่ได้มานั้นนำค่าของ เปอร์เซ็นต์ประสิทธิภาพ มาทำการวาดออกมาเป็นกราฟจะได้ดัง ภาพที่ 21 ก็จะมีมองเห็นได้แล้วว่า โปรโตคอล TCP นั้นจะมีเปอร์เซ็นต์ของประสิทธิภาพที่เหนือกว่าโปรโตคอล SPX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 21 กราฟการส่งข้อมูลปริมาณต่างๆ โดย TCP และ SPX ที่ MTU ต่างกัน

การวิเคราะห์จากกราฟของการส่งข้อมูลจำนวนต่าง ๆ ที่ใช้ MTU แตกต่างกันนั้นจะเห็นได้ว่าในช่วงของ MTU 128 นั้น ในส่วนของการส่งข้อมูลทั้ง TCP และ SPX นั้น จะมีค่าของเปอร์เซ็นต์ประสิทธิภาพแตกต่างกันค่อนข้างเห็นได้ชัด ซึ่งในการส่งข้อมูลในปริมาณที่น้อย ๆ นั้น จะมีประสิทธิภาพต่ำกว่าการส่งข้อมูลที่มีปริมาณมาก ทั้งนี้เนื่องจากที่ข้อมูลจำนวนน้อยนั้นจะมีข้อมูลที่ส่งมีค่าห่างจากจำนวนข้อมูลที่ส่งจริงมาก จึงทำให้มีค่าประสิทธิภาพที่ต่ำกว่าในส่วนของการส่งข้อมูลที่มีปริมาณมากขึ้น ค่าความแตกต่างของข้อมูลที่ส่งกับข้อมูลที่ส่งจริงจะแตกต่างกันน้อยลง คือมีค่าเข้าใกล้กันมากขึ้น

ในส่วนการส่งข้อมูลที่เปลี่ยนค่า MTU ไปที่ค่าสูงขึ้น เช่นในกราฟ จะเห็นได้ว่าที่ MTU 576 นั้น ค่าความแตกต่างของการส่งข้อมูลของ TCP และ SPX นั้นจะแตกต่างกันน้อยลง มีค่าใกล้เคียงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้เผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันมาก และแนวโน้มของการส่งข้อมูลจะมีประสิทธิภาพดีขึ้นเรื่อย ๆ ในกรณีที่มีข้อมูลที่ส่งเพิ่มมากขึ้น และในส่วนของ TCP การส่งข้อมูลที่เปลี่ยนค่าของ MTU เป็นค่าที่มากขึ้นตามลำดับ จากกราฟจะเห็นได้ว่าประสิทธิภาพของการส่งข้อมูลที่ จำนวนต่าง ๆ จะมีค่าดีขึ้นเรื่อย ๆ ทั้งนี้เนื่องจากข้อมูลที่ส่งมีค่าความแตกต่างจากข้อมูลที่ส่งจริงน้อยลง ทั้งนี้เนื่องจากข้อมูลที่ส่งมีค่าความแตกต่างจากข้อมูลที่ส่งจริงน้อยลง ทั้งนี้เพราะว่าในส่วนของ TCP เองนั้นสามารถมีการปรับเปลี่ยนค่าของ Maximum Segment Size ได้ตาม MTU ที่มีค่าเพิ่มมากขึ้น ส่วน SPX นั้นจะมีค่า Maximum ของ MTU ได้แก่ 576 Byte เท่านั้น

เพราะฉะนั้น ในการส่งข้อมูลหากเป็นการส่งข้อมูลที่มีปริมาณ MTU น้อย ๆ การส่งข้อมูลพิจารณาการส่งข้อมูลโดยใช้ TCP จะมีประสิทธิภาพดีกว่า SPX หากเป็นการส่งข้อมูลที่ MTU 576 Byte แล้วนั้น อาจจะไม่เลือกให้การส่งข้อมูลแบบ TCP หรือ SPX ก็ได้เพราะมีประสิทธิภาพใกล้เคียงกัน แต่ถ้าหากเป็นการส่งข้อมูลที่มีปริมาณ MTU มากกว่า 576 Byte แล้วนั้น สมควรที่จะเลือกให้การส่งข้อมูลแบบ TCP จะดีกว่า

## บทที่ 5

### สรุปผลการวิเคราะห์เปรียบเทียบและข้อเสนอแนะ

#### 5.1 สรุปผลการวิเคราะห์เปรียบเทียบ

จากการศึกษาและวิเคราะห์เปรียบเทียบชุดโปรโตคอล TCP/IP และชุดโปรโตคอล IPX/SPX ซึ่งในที่นี่จะศึกษาถึงรูปแบบของแพคเกจข้อมูลของโปรโตคอลต่างๆ โดยการเปรียบเทียบนี้จะศึกษาทั้งโปรโตคอลที่ทำงาน แบบ Connectionless และ Connection-Oriented โดยในส่วนของ Connectionless นั้นก็คือโปรโตคอล IP และโปรโตคอล IPX รวมทั้งโปรโตคอล UDP ด้วย ในส่วนของ Connection-Oriented ก็คือโปรโตคอล TCP และโปรโตคอล SPX โดยในส่วนของ IP datagram จะแบ่งเป็นส่วนของ IP header ซึ่งถ้าขนาดไม่รวม IP OPTIONS จะมีขนาด 20 ไบต์ และตัว IP datagram ก็จะมีขนาดของข้อมูลได้ถึง 65,515 ไบต์และมีขนาดรวม IP header และ ส่วนของข้อมูลรวมกันได้ 65,535 ไบต์ ส่วน IPX แพคเกจนั้นจะมีส่วนของ header ขนาด 30 ไบต์ และส่วนของ ข้อมูลจะมีค่าตั้งแต่ 0 ถึง 546 ไบต์ และจากการเปรียบเทียบในส่วนของ format utilization ของทั้ง IP และ IPX แล้วนั้น IP จะมี format utilization ที่ดีกว่า แต่ถ้ามองในเรื่องการ ระบุแอดเดรสแล้ว IPX จะมีขนาดแอดเดรสกว้างใหญ่กว่า IP

ส่วนการทำงานในรูปแบบ Connection-Oriented นั้นจะมี TCP ทำการเปรียบเทียบกับ SPX โดยที่จำนวนของแพคเกจของ TCP นั้นจะมีขนาดได้สูงสุดถึง 65,535 ไบต์และมีส่วนของ header ของ IP เป็นขนาด 40 ไบต์ ส่วนของแพคเกจของ SPX นั้น จะมีขนาดมากที่สุด 576 ไบต์ และมีส่วนของเฮดเดอร์ 42 ไบต์ จึงทำให้ค่า Format Utilization หาดอกมาแล้ว TCP จะมีค่า Utilization ดีกว่า และเมื่อทำการพิจารณาการส่งข้อมูลขนาดต่างๆ โดยใช้ MTU ที่ค่าต่างๆ ตามที่ได้แสดงไว้ในบทที่ 4 แล้วนั้น จะเห็นได้ว่า ที่ปริมาณข้อมูลต่างๆ จนถึงข้อมูลปริมาณมาก ของการ ส่งข้อมูลของทั้ง TCP และ SPX จะเห็นได้ว่า TCP มีประสิทธิภาพดีกว่า SPX ทั้งนี้เนื่องจากใน ส่วนของ TCP นั้น ขนาดของแพคเกจสามารถเปลี่ยนแปลงได้ตาม MTU นั้นเอง

## 5.2 ข้อเสนอแนะ

จากการศึกษาและวิเคราะห์ชุดโปรโตคอลทั้งสองแบบ ซึ่งเป็นการศึกษาถึงรูปแบบของการส่งแพคเกจข้อมูล และวิธีการติดต่อสื่อสาร และการพิจารณาถึงการส่งข้อมูลที่ปริมาณต่างๆซึ่งเป็นการศึกษาในรูปแบบของกรณีที่ไม่มีข้อผิดพลาดในการส่งเกิดขึ้น ซึ่งทางผู้จัดทำยังไม่ได้ทำการวิเคราะห์ ในกรณีที่มีความผิดพลาดเกิดขึ้น ซึ่งหากมีการศึกษาในส่วนของ การส่งข้อมูลแล้ว เกิดความผิดพลาดขึ้นนั้น ก็จะทำให้เห็นได้ชัดเจนมากขึ้นไปอีกว่า โปรโตคอลแบบใดมีประสิทธิภาพดี กว่าในกรณีที่มีความผิดพลาดเกิดขึ้น



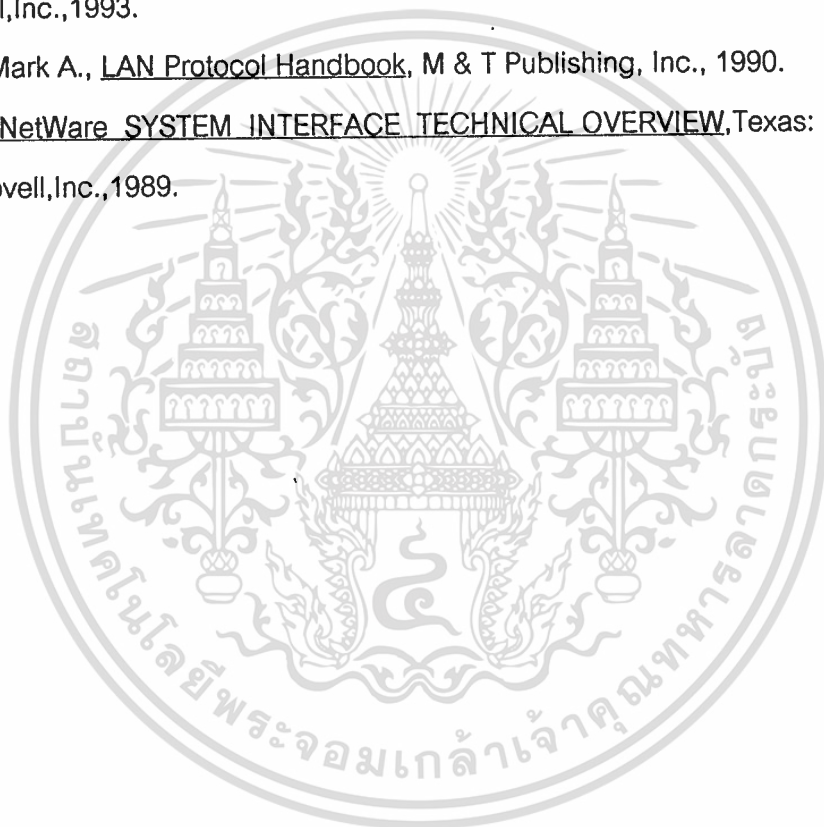
## บรรณานุกรม

Comer, Douglas E., INTERNETWORK WITH TCP/IP VOLUME 1, 2 ed. London: Prentice Hall International, 1991.

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

Miller, Mark A., LAN Protocol Handbook, M & T Publishing, Inc., 1990.

Novell, NetWare SYSTEM INTERFACE TECHNICAL OVERVIEW, Texas: Novell, Inc., 1989.



## ประวัติผู้เขียน

- ชื่อผู้เขียน เรืออากาศโท สามารถ หมดนุรักษ์
- วันเดือนปีเกิด 15 พฤษภาคม 2513
- สถานที่เกิด 98 ซอย วิภาวดี ถนน รามคำแหง แขวงสวนหลวง  
เขตสวนหลวง กรุงเทพมหานคร
- ประวัติการศึกษาปริญญาตรี สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
มหาวิทยาลัยรามคำแหง
- ประวัติการทำงาน พ.ศ. 2534 ทำงานที่ กรมการสนเทศทหาร กองบัญชาการทหารสูงสุด  
ตำแหน่ง นายทหารสื่อสารข้อมูล กองปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้