

โปรแกรมสำรวจเครือข่าย
Network Discovery Tool



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2546

ได้รับหนังสือ/เอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เลขที่ 55139

วันที่ 8 เม.ย. 2548

วันเดือนปี 8 เม.ย. 2548

บ.
ก.

โปรแกรมสำรวจเครือข่าย
Network Discovery Tool



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมสำรวจเครือข่าย

ผู้จัดทำ

1. นายณพพร

รัตนวิไล

43010117

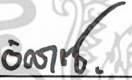
2. นายทองศักดิ์

เปรมผาตินาถ

43010144



อาจารย์ที่ปรึกษา
(อาจารย์ธนา หงษ์สุวรรณ)



อาจารย์ที่ปรึกษา
(อาจารย์อัครเดช วัชรพงษ์)



อาจารย์ที่ปรึกษา
(อาจารย์ธัญชัย ตีรภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสำรวจเครือข่าย

นายณพพร	รัตนวิไล	43010117
นายทงศักดิ์	เปรมผาตินาถ	43010144
อาจารย์ธนา	หงษ์สุวรรณ	อาจารย์ที่ปรึกษา
อาจารย์อัศรเดช	วัชรภพพงษ์	อาจารย์ที่ปรึกษา
อาจารย์ธนัญชัย	ตรีภาค	อาจารย์ที่ปรึกษา
ปีการศึกษา 2546		

บทคัดย่อ

ในปัจจุบันนี้มีการใช้งานเครือข่ายการสื่อสารข้อมูลภายในองค์กรต่างๆ เป็นไปอย่างแพร่หลายทั้งในด้านการสื่อสารข้อมูลภายในองค์กรและการสื่อสารข้อมูลระหว่างองค์กร ดังนั้นก็จะเป็นเรื่องยากสำหรับผู้ดูแลระบบ หรือผู้ที่ต้องการศึกษาระบบเครือข่ายนั้นๆ ว่ามีการวางระบบหรือมีการเชื่อมต่ออย่างไร ภายในองค์กรหรือระหว่างองค์กร

วิทยานิพนธ์นี้เป็นการนำเสนอ โครงการพัฒนาโปรแกรมสำรวจเครือข่าย (Network Discovery Tool) โดยตัวโปรแกรมนี้จะมองภาพรวมของระบบเครือข่าย จึงช่วยให้ผู้ใช้งานโปรแกรมมองเห็นภาพของระบบเครือข่ายได้ง่ายขึ้น และโครงการนี้จะนำเสนอเทคนิคต่างๆ ที่ใช้ในการรวบรวมข้อมูลของระบบเครือข่าย(Ping Sweep, Trace Route, NetStat, SNMP, Whois) ส่วนที่ใช้ในการจัดเก็บข้อมูลและส่วนที่ใช้ในการจัดการเพื่อนำเสนอผลบน โปรแกรม รวมถึงรายละเอียดในการพัฒนาโปรแกรม เพื่อให้ผู้ที่สนใจสามารถนำไปเป็นแนวทางในการพัฒนาเครื่องมือที่ใช้อำนวยความสะดวกในการจัดการเครือข่ายอื่นๆ เพื่อลดการนำเข้าโปรแกรมต่างๆจากต่างประเทศ

Network Discovery Tool

Mr. Naphachara	Rattanawilai	
Mr. Tanongsak	Prempatinart	
Mr. Thana	Hongsuwan	Advisor
Mr. Akkradach	Watcharapupong	Advisor
Mr. Thanunchai	Threepak	Advisor

ABSTRACT

Presently, Communication Network is applied widely in many organizations including both internal communication and external communication. Therefore, It is hard for administrator or anyone who wants study about network how to design the system or how to connect it?

The thesis describes the Network Discovery Tool that this program will look overview of network and it can help users to clarify the topology including describes the use of enumeration technique for gathering network information(Ping Sweep, Trace Route, NetStat, SNMP, Whois), storing database and information processing in order to display. Eventually, other developers will be able to continue this work efficiently in many different ways.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่สำเร็จลุล่วงไปด้วยดี หากปราศจากคำแนะนำและการให้คำปรึกษาจาก อาจารย์ ธนา หงษ์สุวรรณ, อาจารย์ อัครเดช วัชรระภูงษ์ และ อาจารย์ ธนัญชัย ศรีภาค ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอขอบคุณเป็นอย่างสูง

ขอขอบคุณห้องปฏิบัติการ ISAG ที่อำนวยความสะดวกเกี่ยวกับสถานที่ทำงาน และยังเป็นสถานที่พบปะเพื่อน ๆ เพื่อให้มีกำลังใจในการทำงานต่อไปด้วย

ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ จนทุก ๆ คนในครอบครัวที่ทำให้กำลังใจทำวิทยานิพนธ์ชิ้นนี้จนสำเร็จได้ด้วยดี

ขอขอบคุณตัวข้าพเจ้าทั้งสองที่ไม่ท้อแท้ไปเสียก่อนที่จะประสบความสำเร็จ ขอขอบคุณที่มีชีวิตมาถึงทุกวันนี้ และโอกาสดีๆ ที่ได้รับมา

และต้องขอขอบคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างดี และยังให้กำลังใจ เอาใจใส่เสมอมาในทุกๆ ด้านอันหาที่เปรียบมิได้ ขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

ณ พชร รัตนวิไล
ทนางศักดิ์ เปรมผาตินาถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญประกอบภาพ	VII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ประโยชน์ที่ได้รับ	2
1.4 ขอบเขตของโครงการ	3
1.5 แผนการดำเนินงาน	5
บทที่ 2 เทคโนโลยีเครือข่ายคอมพิวเตอร์	6
2.1 แลตามข้อกำหนดของ IEEE	6
2.2 โปรโตคอลจุดต่อจุด	12
2.3 ไอพีและเทคโนโลยีเครือข่ายอื่น	13
2.4 ลูปแบ็คอินเทอร์เฟซ	14
2.5 เอ็มทียู	14
2.6 อุปกรณ์เครือข่าย	14
บทที่ 3 โพรโทคอลที่ซีพี/ไอพี	18
3.1 ความเป็นมาของโพรโทคอลที่ซีพี/ไอพี	18
3.2 การเชื่อมต่อของโพรโทคอลที่ซีพี/ไอพี (TCP/IP Linking)	18
3.3 โพรโทคอลที่ซีพี (TCP: Transmission Control Protocol)	20
3.4 โพรโทคอลยูดีพี (UDP: User Datagram Protocol)	23
3.5 โพรโทคอลไอพี (IP: Internet Protocol)	23
บทที่ 4 ไอพีแอดเดรส (IP Address)	27
4.1 ไอพีแอดเดรส	27
4.2 การแบ่งเครือข่ายย่อย	31
4.3 ไอพีแอดเดรสเฉพาะกรณี	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
4.4 ชั้นเน็ตชนิดแปรเปลี่ยนได้	42
4.5 ปัญหาการจัดสรร ไอพีแอดเดรส	44
บทที่ 5 โพรโทคอลเลือกเส้นทาง (Routing Protocol)	49
5.1 การเลือกเส้นทาง (Routing)	49
5.2 ตารางเส้นทาง(Routing Table)	50
5.3 ประเภทของการเลือกเส้นทาง	50
5.4 อาร์ไอพี(RIP)	54
5.4.1 การทำงานของอาร์ไอพี	54
5.4.2 การปรับค่าเมื่อเครือข่ายเปลี่ยน	57
5.4.3 เวลาผู้เข้า	59
5.4.4 สปลิตซอไรซัน(Split Horizon)	59
5.4.5 สปลิตซอไรซันแบบพอยซันรีเวอร์ส (Split Horizon with Poisoned Reverse)	60
5.4.6 ทริกเกอร์อัปเดต	61
5.4.7 เฟรมอาร์ไอพี	62
5.4.8 เฟรมอาร์ไอพี 2	64
5.4.9 การพิสูจน์ตัวจริง	65
5.5 โอเอสพีเอฟ (OSPF : Open Shortest Path First)	66
5.5.1 หลักการพื้นฐานของโพรโทคอลแบบโอเอสพีเอฟ	66
5.5.2 การจัดองค์ประกอบเครือข่าย	71
5.5.3 เราเตอร์ตัวแทน	73
5.5.4 การเลือกเส้นทาง	74
5.5.5 เมตริกในโอเอสพีเอฟ	75
5.5.6 รูปแบบเฟรมโพรโทคอลโอเอสพีเอฟ	76
บทที่ 6 SNMP Service	78
6.1 พื้นฐานการบริหารเครือข่าย	78
6.2 เอสเอ็นเอ็มพีเอเจนต์	79
6.3 โพรโทคอล	80
6.4 โครงสร้างเอ็มไอบี	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
บทที่ 7 Scan	85
7.1 Network Ping Sweeps	85
7.2 ICMP Queries	87
7.3 Port Scanning	87
7.4 การตรวจหาประเภทของระบบปฏิบัติการ	89
7.5 ตัวอย่างเครื่องมือที่ใช้ในการสำรวจระบบเครือข่าย	92
บทที่ 8 การออกแบบและพัฒนาโปรแกรม	93
8.1 รายละเอียดของการพัฒนา	93
8.2 การออกแบบโครงสร้างของโปรแกรม	94
8.3 ตัวอย่างส่วนติดต่อกับผู้ใช้	97
บทที่ 9 การทำงานของโปรแกรมสำรวจเครือข่าย	99
9.1 ข้อมูลที่ได้จากการใช้วิธีต่างๆ	99
Ping (Ping Sweep), Trace Route, Whois, RIP (Listener), SNMP	
9.2 การสำรวจเครือข่าย	103
9.3 การตรวจสอบเครื่อง	104
บทที่ 10 วิเคราะห์ผลการทดลองและสรุป	106
10.1 วิเคราะห์ผลการทดลอง	106
10.2 การทดลองด้วยวิธีการ Ping และ Trace Route	106
10.3 การทดลองด้วยวิธีการทั้งหมด (Typical)	108
10.4 สรุปผล	109
10.5 แนวทางในการพัฒนาสำหรับผู้สนใจในอนาคต	109
บรรณานุกรม	110

สารบัญรูปภาพ

	หน้าที่
รูปที่ 1-1 ภาพผลลัพธ์ของโปรแกรมตัวอย่างที่ทำงานในลักษณะการค้นหาเครือข่าย	2
รูปที่ 1-2 แสดงการค้นหาเครือข่ายจากเครือข่ายผู้ใช้งานขยายออกสู่ภายนอก	3
รูปที่ 1-3 แสดงการค้นหาเครือข่ายภายนอกเครือข่ายผู้ใช้งาน	4
รูปที่ 1-4 แสดงแผนการดำเนินงาน	5
รูปที่ 2-1 แบบจำลองเครือข่าย IEEE 802.3, 802.4 และ 802.5	6
รูปที่ 2-2 อีเทอร์เน็ตประเภทต่างๆ	8
รูปที่ 2-3 การทำงานโทเค็นริง	9
รูปที่ 2-4 โทโปโลยีเอฟดีไอ	10
รูปที่ 2-5 เปรียบเทียบการเอ็นแคปซูลเอเทอร์เน็ตแบบเส้นคู่และแบบดั้งเดิม	11
รูปที่ 2-6 การเชื่อมโยงด้วยสลิป	12
รูปที่ 2-7 การใช้ฮับในโทโปโลยีแบบดาว	15
รูปที่ 2-8 แบบจำลองการทำงานของฮับ	15
รูปที่ 2-9 แบบจำลองการทำงานของบริดจ์	16
รูปที่ 2-10 แบบจำลองการทำงานของเราเตอร์	17
รูปที่ 3-1 แสดงการเปรียบเทียบเลขอร์ของไอเอสไอกับเลขอร์ของทีซีพี/ไอพี	19
รูปที่ 3-2 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี	20
รูปที่ 3-3 แสดงการทำ 3-Way Handshake	21
รูปที่ 3-4 แสดงแพ็กเก็ตทีซีพี	22
รูปที่ 3-5 แสดงแพ็กเก็ตยูดีพี	23
รูปที่ 3-6 แสดงการทำแฟร็กเมนต์ชัน	24
รูปที่ 3-7 แสดงการรีแอสเซมเบิล	24
รูปที่ 3-8 แสดงแพ็กเก็ตไอพี	26
รูปที่ 4-1 รูปแบบของไอพีแอดเดรส	28
รูปที่ 4-2 เราเตอร์เชื่อมโยงเครือข่ายที่มีเลขเครือข่ายต่างกัน	28
รูปที่ 4-3 การแบ่งคลาสเครือข่าย	29
รูปที่ 4-4 การแบ่งคลาส D และ E	29
รูปที่ 4-5 ตัวอย่างการแบ่งเครือข่ายย่อยของ 161.246	32
รูปที่ 4-6 การตรวจสอบหาแอดเดรสซบเน็ตเพื่อเลือกเส้นทาง	34
	หน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-7 ตัวอย่างการจัดแบ่งชั้นเน็ต 194.1.1.0 แบบต่างๆ	37
รูปที่ 4-8 ไอพีแอดเดรสเฉพาะกรณี	38
รูปที่ 4-9 เครือข่ายที่ใช้ชั้นเน็ตความยาวแปรค่า	43
รูปที่ 4-10 การแบ่งชั้นเน็ตความยาวแบบรีเคอร์ซีฟ	44
รูปที่ 5-1 เครือข่ายการแสดงการเลือกเส้นทางแบบสเตติก	51
รูปที่ 5-2 เครือข่ายสถิติการทำงานของอาร์ไอพี	54
รูปที่ 5-3 การประกาศค่าและปรับค่าและตารางของอาร์ไอพี	57
รูปที่ 5-4 การนับเข้าสู่อนันต์	58
รูปที่ 5-5 การประกาศเส้นทางด้วยวิธีสปลิตฮอปไรชัน	59
รูปที่ 5-6 การประกาศเส้นทางด้วยวิธีสปลิตฮอปไรชันแบบพอยชันรีเวอร์ส	60
รูปที่ 5-7 เครือข่ายที่จำเป็นต้องใช้โหนดความรู้ร่วมกับทริกเกอร์อัปเดต	61
รูปที่ 5-8 ฟอรัมเมตของเฟรมอาร์ไอพี	63
รูปที่ 5-9 ฟอรัมเมตของเฟรมอาร์ไอพีเวอร์ชัน 2	65
รูปที่ 5-10 เฟรมอาร์ไอพีเวอร์ชัน 2 เมื่อใช้การพิสูจน์ตัวจริง	65
รูปที่ 5-11 แผนที่ทางหลวง	66
รูปที่ 5-12 เราเตอร์ A ส่งแพ็กเก็ตตกท้าย	67
รูปที่ 5-13 การคำนวณหาเส้นทางสั้นที่สุดตามขั้นตอนวิธีของไดจ์สตรา	69
รูปที่ 5-14 ปัญหาเส้นทางขัดข้อง	70
รูปที่ 5-15 เส้นทางใหม่ที่คำนวณได้	70
รูปที่ 5-16 เราเตอร์ขอบปลายของระบบบอโตโนมัส	71
รูปที่ 5-17 ตัวอย่างการจัดพื้นที่ในระบบบอโตโนมัสของโอเอสพีเอฟ	72
รูปที่ 5-18 การใช้เราเตอร์ตัวแทน	73
รูปที่ 5-19 เซกเตอร์โอเอสพีเอฟแพ็กเก็ต	76
รูปที่ 6-1 องค์ประกอบในระบบจัดการเครือข่าย	78
รูปที่ 6-2 เอสเอ็นเอ็มพีเอเจนต์	79
รูปที่ 6-3 โครงสร้างของเอเจนต์	80
รูปที่ 6-4 การเอนแคปซูเลตเอสเอ็นเอ็มพี	81
รูปที่ 6-5 โครงสร้างพิดิยูของ get, get-next และ get-response	81
รูปที่ 6-6 โครงสร้างพิดิยูของคำสั่ง trap	82
รูปที่ 6-7 อ็อบเจ็กต์ไอดีเอ็นดีไฟเบอร์ในโครงสร้างฐานข้อมูลสารสนเทศ	83
รูปที่ 7-1 แสดงการทำงานของ ping sweep	85

รูปที่ 7-2 แสดงการทำงานของ ICMP QUERY	87
รูปที่ 7-3 TCP 's 3-way handshake	88
รูปที่ 8-1 แสดง Use Case ของโปรแกรม	94
รูปที่ 8-2 แสดงโครงสร้างโปรแกรม	95
รูปที่ 8-3 ตัวอย่างต้นแบบหน้าจอที่ใช้ในการใส่ Input ของโปรแกรม	97
รูปที่ 8-4 ตัวอย่างต้นแบบหน้าจอแผนภาพที่ได้จากการสำรวจ	98
รูปที่ 9-1 เป็นผลลัพธ์ที่ได้จากการ Ping Sweep (ของตัวโปรแกรม)	99
รูปที่ 9-2 เป็นผลลัพธ์ที่ได้จาก Trace Route ไปยัง Host	100
รูปที่ 9-3 เป็นผลลัพธ์ที่ได้จาก Trace Route ไปยัง Router หรือ Gateway	100
รูปที่ 9-4 เป็นการสอบถามข้อมูลไอพี 209.15.15.15 จาก Whois.internic.net	101
รูปที่ 9-5 เรดิ่งเทเบิลภายในเครื่อง จากการ Netstat -nr	102
รูปที่ 9-6 ipNetToMediaTable	102
รูปที่ 9-7 ipRoutingTable	103
รูปที่ 9-8 ipAddrTable	103
รูปที่ 9-9 แสดงการเลือก New Scan	103
รูปที่ 9-10 แสดงเมนูการที่มีในการคลิกขวาที่แต่ละไอพีแบบที่เป็นขยาย	104
รูปที่ 9-11 แสดงเมนูการที่มีในการคลิกขวาที่แต่ละไอพีแบบที่ย่อ	105
รูปที่ 9-12 แสดง System อย่างคร่าวๆ	104
รูปที่ 9-13 แสดงว่าเครื่องยังมีการทำงานอยู่	106
รูปที่ 10-1 ภาพตัวอย่างเครือข่ายที่ต้องการที่จะสำรวจ	106
รูปที่ 10-2 ภาพผลลัพธ์บางส่วน of โปรแกรมจากการสำรวจโดยเลือกใช้วิธีการ Ping + Trace Route	107
รูปที่ 10-3 ภาพจากการนำผลลัพธ์ที่ได้มาวาดเป็นแผนภาพ (Ping + Trace Route)	107
รูปที่ 10-4 ภาพผลลัพธ์บางส่วน of โปรแกรมจากการสำรวจโดยเลือกใช้วิธีการทั้งหมด (Typical)	108
รูปที่ 10-5 ภาพจากการนำผลลัพธ์ที่ได้มาวาดเป็นแผนภาพ (Typical)	108

สารบัญตาราง

	หน้าที่
ตารางที่ 3-1 การทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี	20
ตารางที่ 4-1 การจัดแบ่งเครือข่าย 161.246 ด้วยซับเน็ต 8 บิต	32
ตารางที่ 4-2 ค่าดีฟอลต์ซับเน็ตมาสก์	33
ตารางที่ 4-3 ซับเน็ตมาสก์ในเครือข่ายคลาส B	35
ตารางที่ 4-4 ซับเน็ตมาสก์ในเครือข่ายคลาส C	36
ตารางที่ 5-1 ตารางเส้นทางที่เราเตอร์ A	67
ตารางที่ 5-2 ฐานข้อมูลลิงสเทตประจำแต่ละเราเตอร์	68
ตารางที่ 6-1 รหัสผิพลาในเอสเอ็นเอ็มพี	82
ตารางที่ 6-2 กลุ่มย่อยภายใต้ mgmt	84
ตารางที่ 8-1 แสดงคำอธิบาย Use Case Diagram	94



บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

ในปัจจุบันนี้ระบบเครือข่ายคอมพิวเตอร์กำลังเป็นที่นิยมอย่างสูง ทำให้จำนวนผู้ใช้ในระบบเพิ่มขึ้นอย่างรวดเร็วและมีผลให้ระบบเครือข่ายคอมพิวเตอร์มีขนาด และมีความซับซ้อนมากขึ้น การดูแลระบบเครือข่ายก็ทำได้ยากขึ้นตามไปด้วย โดยการดูแลระบบเครือข่ายเบื้องต้นก็จำเป็น ต้องทราบแผนภาพเครือข่าย ว่าสถานะปัจจุบันของเครือข่ายนั้นเป็นเช่นไร เช่น ในองค์กรขนาดใหญ่จะมีเครือข่ายย่อย (LAN) เชื่อมต่อกันอย่างไร มีการเชื่อมต่อจากไหนไปไหนบ้าง เพื่อไว้สำหรับศึกษาและแก้ไขระบบของเครือข่าย ได้ว่ามีปัญหาหรือว่าต้องมีการแก้ไขปรับปรุงอย่างไร

ในการที่จะสร้างรูปแผนภาพจำลองระบบเครือข่ายคอมพิวเตอร์ จำเป็นต้องทราบข้อมูล เบื้องต้นของระบบเครือข่ายที่ต้องการจะสร้างก่อน โดยโปรแกรมนี้จะทำการรวบรวมข้อมูลเพื่อทำการสร้างเครือข่ายของคอมพิวเตอร์ โดยจะเก็บข้อมูลเครือข่ายตามขอบเขตที่ผู้ใช้ได้เลือกไว้ และนำข้อมูลเหล่านั้น มาสร้างภาพเครือข่ายอย่างง่าย เพื่อที่จะให้ผู้ใช้งานนำข้อมูลเหล่านี้ไปศึกษาต่อ ซึ่งในขั้นแรกจะทำการส่งแพ็กเก็ต ไปตรวจสอบก่อนว่ามีเครื่องภายในเครือข่ายนั้นทำงานอยู่หรือไม่ เมื่อตรวจสอบเสร็จแล้ว ถ้ามีเครื่องเหล่านั้นอยู่ก็จะทำการค้นหาเส้นทางที่จะไปยังเครื่องเหล่านั้น แล้วก็จะนำข้อมูลเหล่านี้มาสร้างเป็นแผนภาพจำลองระบบเครือข่ายขึ้น โดยจะนำมาแสดงในรูปแบบของการเชื่อมต่อจริงของเครือข่ายนั้นๆ

โดยการรวบรวมจะหาข้อมูลที่เป็นให้มากที่สุด เมื่อได้ข้อมูลต่างๆมาแล้วก็จะนำข้อมูลเหล่านี้ มาสร้างเป็นแผนภาพจำลองระบบเครือข่ายขึ้น โดยจะนำเสนอในรูปแบบที่ง่ายต่อการแสดงผล และมีการเปลี่ยนแปลงสถานะหากในระบบเครือข่ายเปลี่ยนแปลง เพื่อสะดวกในการดูแลระบบเครือข่ายแบบตลอดเวลา

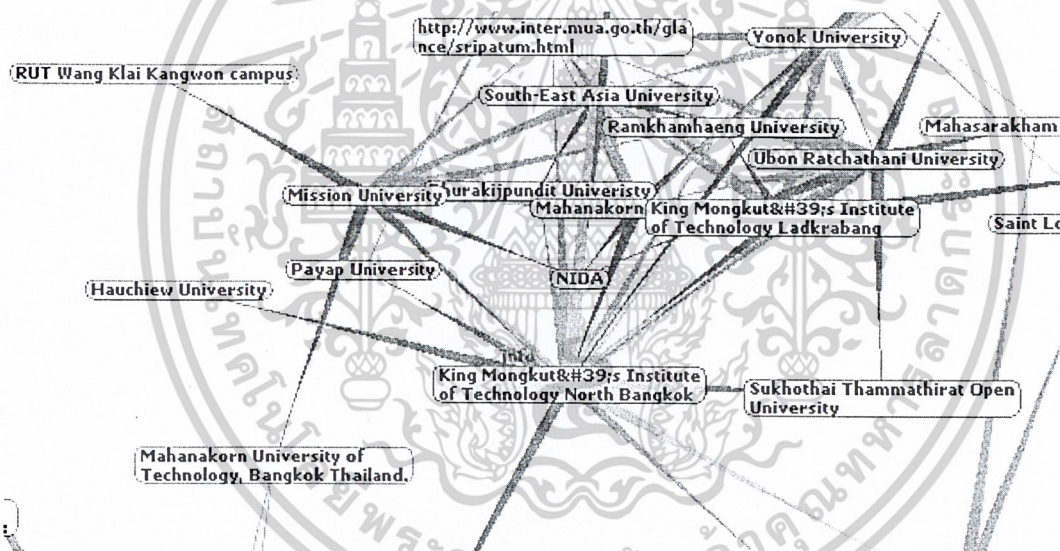
1.2 วัตถุประสงค์ของโครงการ

1. เพื่อพัฒนาโปรแกรมที่อำนวยความสะดวกในการสำรวจระบบเครือข่ายคอมพิวเตอร์
2. เพื่อพัฒนาต้นแบบของ โปรแกรมที่ใช้ในการสำรวจระบบเครือข่ายคอมพิวเตอร์ เพื่อนำไปใช้ทดแทนการนำเข้าโปรแกรมจากต่างประเทศ

1.3 ประโยชน์ที่ได้รับ

- โปรแกรมต้นแบบเพื่อใช้ในการค้นหาระบบเครือข่าย เพื่ออำนวยความสะดวกแก่ผู้ใช้ต่างๆ เช่น
 - 1) สำหรับผู้ที่ต้องการศึกษาเครือข่ายใดๆว่าภายในเครือข่ายนั้นๆมีการเชื่อมต่ออย่างไร เพื่อที่จะนำไปเรียนรู้ หรือพัฒนาเครือข่ายนั้นๆต่อไป
 - 2) กรณีสำหรับผู้ดูแลระบบ สามารถช่วยในการตรวจสอบว่าลิงค์ใดเสียหาย หรือเกิดความผิดพลาดที่จุดใด
- ได้รับความรู้จากการศึกษา เกี่ยวกับเรื่องการเดินทางของข้อมูล ในระบบเครือข่ายและการเขียนโปรแกรมทางด้านเครือข่าย
- ช่วยลดการนำเข้าโปรแกรมที่ใช้ในการค้นหาระบบเครือข่ายจากต่างประเทศ ซึ่งยังไม่มีโปรแกรมที่ทำงานในลักษณะนี้ที่เป็นแบบ Open Source

(ต่อไปเป็นลักษณะของโปรแกรมที่มีผลลัพธ์คล้ายกันกับตัวโปรแกรมนี้)



รูปที่ 1-1 ภาพผลลัพธ์ของโปรแกรมตัวอย่างที่ทำงานในลักษณะการค้นหาเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของโครงการ

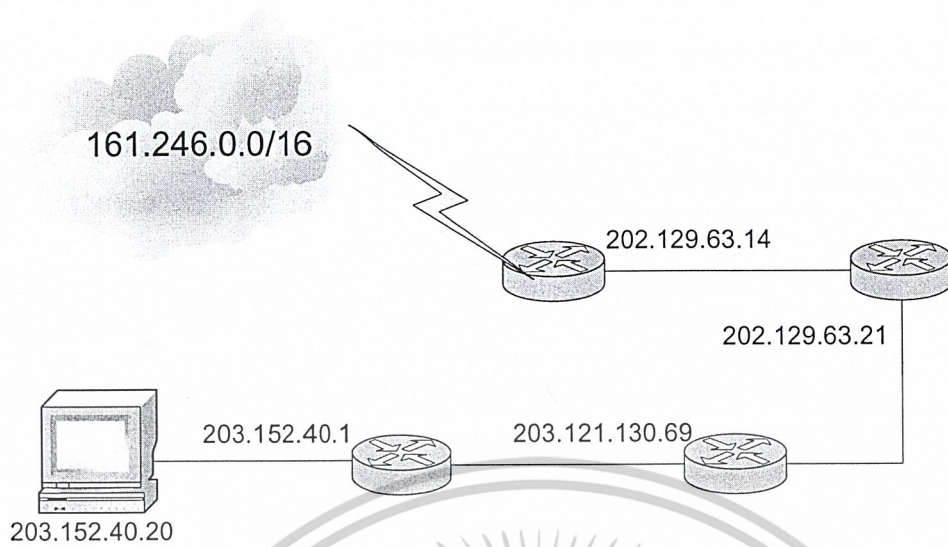
(โดยโปรแกรมจะมีคุณสมบัติพื้นฐานดังนี้)

1. สามารถตรวจสอบได้ว่าเครื่องที่เราสนใจขณะนั้น ได้เปิดใช้งานหรือไม่ (โดยปราศจาก Firewall และ การ Block ICMP)
2. สามารถแสดงเส้นทางไปยังปลายทางได้ด้วย Trace Route
3. สามารถอ่านค่า SNMP Service ที่สนใจ ในการบอกลักษณะของคอมพิวเตอร์ในระบบเครือข่ายนั้นๆได้
4. สามารถทำงานเป็น RIP Listener โดยใช้ Routing Information Protocol (RIP) ช่วยในการเก็บข้อมูลของเครือข่ายนั้นๆได้ ในกรณีที่เครือข่ายที่เราสนใจไม่ใหญ่มากนัก และเป็น การค้นหาเครือข่ายของผู้ใช้เอง
5. สามารถหาว่า Network นั้นมี Subnet อย่างไรโดยใช้ Whois
6. สามารถนำข้อมูลที่รวบรวมได้มาสร้างเป็น แผนภาพระบบเครือข่ายคอมพิวเตอร์ทาง Logical

(โดยตัวโปรแกรมจะให้ผลลัพธ์ของการทำงานใน 2 ลักษณะ ดังรูปที่ 1-2 และ รูปที่ 1-3)



รูปที่ 1-2 แสดงการค้นหาเครือข่ายจากเครือข่ายผู้ใช้งานขยายออกสู่ภายนอก



รูปที่ 1-3 แสดงการค้นหาเครือข่ายภายนอกเครือข่ายผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 แผนการดำเนินงาน

ภาคการศึกษาที่ 1

1. ศึกษาความเป็นไปได้ของโครงการ ที่จะสร้างและนำเครื่องมือนี้มาใช้
2. กำหนดขอบเขตของโครงการ
3. ศึกษาแนวทางการและเทคนิคต่างๆที่จะใช้ในการรวบรวมข้อมูลเครือข่ายที่จะนำมาใช้ในการสร้างแผนภาพ
4. ศึกษาและหา Library หรือ Open Source Code ที่จำเป็น เช่น Library Ping , Trace Route , SNMP , RIP , Whois , TCP Ping , Subnet Topology
5. ทดลองเขียน โปรแกรมย่อยในบางส่วน เช่น Ping
6. Design Interface

ภาคการเรียนที่ 2

1. ทดลองการเขียนโปรแกรมย่อยทั้งหมด
2. นำโปรแกรมย่อยส่วนต่างๆ ที่เขียนไว้มารวมกันเป็นโปรแกรมเดียว
3. เขียนโปรแกรมติดต่อระหว่างโปรแกรมย่อยต่างๆกับฐานข้อมูล เพื่อเก็บข้อมูลลงในฐานข้อมูล
4. ศึกษา Algorithm ที่จะนำมาจัดการกับข้อมูลที่รวบรวมมาได้ เพื่อที่จะเขียน โปรแกรมสร้างแผนภาพระบบเครือข่ายคอมพิวเตอร์

Task Name	June			July			August			September			October			November			December			January		
	B	M	E	B	M	E	B	M	E	B	M	E	B	M	E	B	M	E	B	M	E	B	M	E
1 ศึกษาความเป็นไปได้ของโครงการ																								
2 กำหนดขอบเขตของโครงการ																								
3 ศึกษา SNMP และ ICMP และ Protocol อื่นๆ																								
4 ศึกษา Source Code และ Utility ต่างๆ																								
5 ดำเนินการทดลองเขียนโปรแกรมย่อยต่างๆ																								
6 ศึกษาว่าจะใช้ Algorithm อะไร																								
7 หา Source Code และศึกษา Source Code Algorithm ต่างๆ																								
8 ดำเนินการเขียนโปรแกรมย่อยต่างๆ																								
9 รวมแต่ละโปรแกรมย่อยต่างๆ																								

■ งานที่ทำไว้จริง

□ งานที่วางไว้ว่าจะทำได้

รูปที่ 1-4 แสดงแผนการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

เทคโนโลยีเครือข่ายคอมพิวเตอร์

ทีซีพี/ไอพีผ่านการออกแบบมาให้สามารถทำงานกับระบบการสื่อสารระดับล่างโดยไม่จำกัดประเภท ในปัจจุบันมีฮาร์ดแวร์เครือข่ายจำนวนมากทั้งในกลุ่มของแลนและแวนที่รองรับการทำงานร่วมกับทีซีพี/ไอพี เนื้อหาในบทนี้จะกล่าวถึงความรู้เบื้องต้นเกี่ยวกับแลนและโปรโตคอลที่ใช้ในแวน รวมทั้งความรู้พื้นฐานเกี่ยวกับอุปกรณ์เครือข่าย หัวข้อที่จะกล่าวในบทนี้ประกอบด้วย

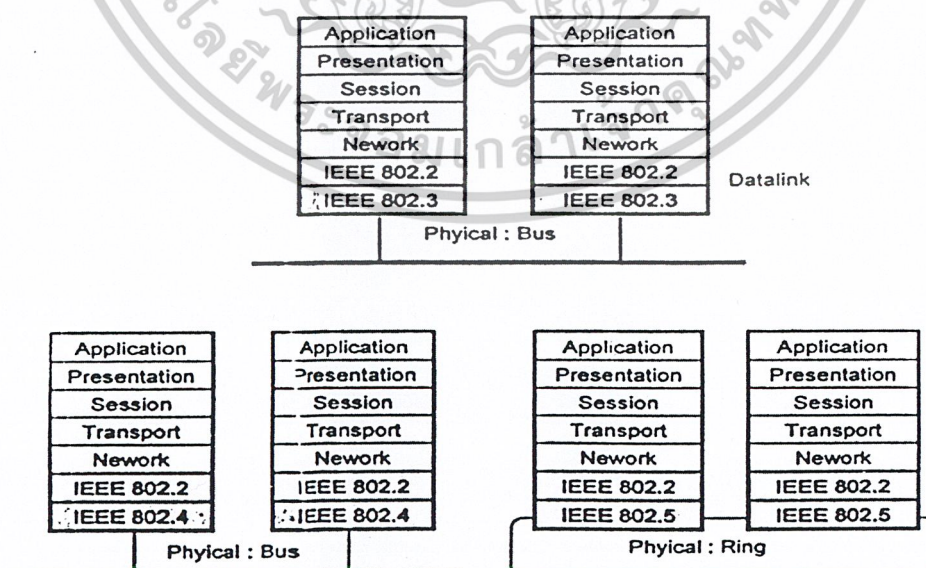
- เครือข่ายแลน ได้แก่ อีเทอร์เน็ตและโทเคนริง
- โปรโตคอลแบบจุดต่อจุด ได้แก่ สลิปและพีพีพี
- อุปกรณ์เครือข่ายพื้นฐาน ได้แก่ ฮับ บริดจ์ และเราเตอร์

2.1 แลนตามข้อกำหนดของ IEEE

เครือข่ายเฉพาะที่หรือแลนมีลักษณะโดยทั่วไปคือ มีอัตราส่วนส่งข้อมูลสูงและมักมีรัศมีเครือข่ายครอบคลุมระยะทางประมาณ 5 กิโลเมตร ในหัวข้อนี้จะกล่าวถึงแลนตามมาตรฐานที่กำหนดโดย IEEE (Institute of Electrical and Electronics Engineering)

IEEE กำหนดเครือข่ายเฉพาะที่โดยใช้ตัวเลข 802 ตามตัวเลขย่อยเป็นรหัสประจำแต่ละมาตรฐาน รูปที่ 2-1 เป็นแบบจำลองบางส่วนของมาตรฐานซึ่งเป็นที่รู้จักแพร่หลายได้แก่

- IEEE 802.3 หรืออีเทอร์เน็ต ใช้โปรโตคอลซีเอสเอ็มเอ/ซีดีในโทโปโลยีแบบบัส
- IEEE 802.4 หรือโทเคนบัส ใช้โปรโตคอลส่งผ่านโทเคนในโทโปโลยีแบบบัส
- IEEE 802.5 หรือโทเคนริง ใช้โปรโตคอลส่งผ่านโทเคนในโทโปโลยีวงแหวน



รูปที่ 2-1 แบบจำลองเครือข่าย IEEE 802.3, 802.4 และ 802.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดเพิ่มเติมที่ IEEE สร้างขึ้นในทุกมาตรฐาน 802 คือแยกระดับชั้นดาตาลงคือออกเป็น 2 ส่วนย่อย โดยใช้มาตรฐาน 802.2 ซึ่งเรียกว่า แอลแอลซี (LLC : Logical Link Control) เป็นส่วนเชื่อมต่อกับชั้นเน็ตเวิร์ก อินเทอร์เน็ตของแอลแอลซีในเครือข่ายแต่ละชนิด (802.3, 802.4 และ 802.5) จะมีรูปแบบเชื่อมต่อกับชั้นเน็ตเวิร์กเช่นเดียวกันหมด

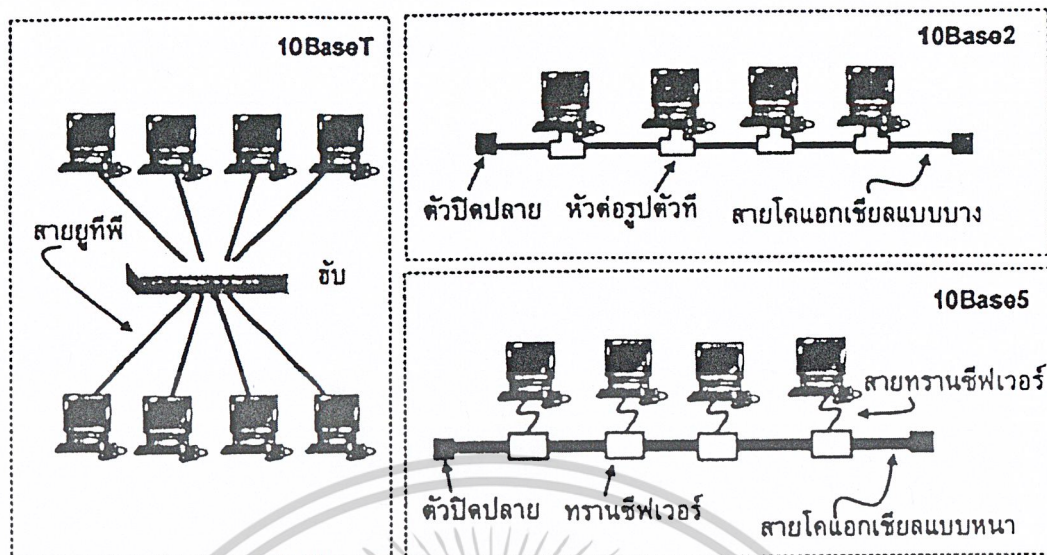
2.1.1 IEEE 802.3

IEEE 802.3 หรือ อีเทอร์เน็ต (Ethernet) เป็นเครือข่ายที่มีความเร็วในการส่งข้อมูล 10 เมกะบิตต่อวินาที สถานีในเครือข่ายอาจมีโทโปโลยีแบบบัสหรือแบบดาว IEEE ได้กำหนดมาตรฐานอีเทอร์เน็ตซึ่งทำงานที่ความเร็ว 10 เมกะบิตต่อวินาทีไว้หลายประเภทตามชนิดสายสัญญาณ เช่น

- 10Base5 อีเทอร์เน็ตโทโปโลยีแบบบัสซึ่งใช้สายโคแอกเชียลแบบหนา (Thick Ethernet) ความยาวสายในเซกเมนต์หนึ่งๆ ไม่เกิน 500 เมตร
- 10Base2 อีเทอร์เน็ตโทโปโลยีแบบบัสซึ่งใช้สายโคแอกเชียลแบบบาง (Thin Ethernet) ความยาวของสายในเซกเมนต์หนึ่งๆ ไม่เกิน 185 เมตร
- 10BaseT อีเทอร์เน็ตโทโปโลยีแบบดาวซึ่งใช้สับเป็นศูนย์กลาง สถานีและเชื่อมด้วยสายยูทีพี (Unshield Twisted Pair) ด้วยความยาวไม่เกิน 100 เมตร

รูปที่ 2-2 แสดงถึงลักษณะเครือข่ายอีเทอร์เน็ตแยกตามประเภทของสายสัญญาณ รหัสขึ้นต้นด้วย 10 หมายถึงความเร็วสายสัญญาณ 10 เมกะบิตต่อวินาที คำว่า “Base” หมายถึงสัญญาณชนิด “Baseband” รหัสถัดมาหากเป็นตัวเลขหมายถึงความยาวสายต่อเซกเมนต์ ในหน่วยหนึ่งร้อยเมตร (5=500, 2แทนค่า 185) หากเป็นอักษรจะหมายถึงชนิดของสาย เช่น T คือ Twisted Pair หรือ F คือ Fiber Optics

ส่วนมาตรฐานอีเทอร์เน็ตความเร็ว 100 เมกะบิตต่อวินาทีที่นิยมใช้ในปัจจุบันได้แก่ 100BaseTX และ 100BaseFX สำหรับอีเทอร์เน็ตความเร็วสูงแบบกิกะบิตอีเทอร์เน็ตเริ่มแพร่หลายมากขึ้น ตัวอย่างของมาตรฐานกิกะบิตอีเทอร์เน็ตในปัจจุบันได้แก่ 1000BaseT, 1000BaseLX, 1000BaseSX เป็นต้น



รูปที่ 2-2 อีเทอร์เน็ตประเภทต่างๆ

อีเทอร์เน็ตใช้โปรโตคอล ซีเอสเอ็มเอ/ซีดี (CSMA/CD : Carrier Sense Multiple Access with Collision Detection) เป็นตัวกำหนดขั้นตอนให้สถานีเข้าครอบครองสายสัญญาณ ในขณะเวลาหนึ่งจะมีเพียงสถานีเดียวที่เข้าครอบครองสายสัญญาณเพื่อส่งข้อมูล

สถานีที่ต้องการส่งข้อมูลต้องตรวจสอบสายสัญญาณว่ามีสถานีอื่นใช้สายอยู่หรือไม่ ถ้าสายสัญญาณว่างก็ส่งข้อมูลได้ทันที หากไม่ว่างก็ต้องคอยจนกว่าสายสัญญาณว่างจึงจะส่งข้อมูลได้ ขณะที่สถานีหนึ่งกำลังส่งข้อมูลก็ต้องตรวจสอบสายสัญญาณไปพร้อมกันด้วยเพื่อตรวจสอบ ในจังหวะเวลาที่ใกล้เคียงกันนั้นมีสถานีอื่นซึ่งพบว่าสายสัญญาณว่างและส่งข้อมูลมาหรือไม่ หากเกิดเช่นนี้ขึ้นแล้ว ข้อมูลจากทั้งสองสถานีจะผสมกันหรือเรียกว่า การชนกัน (Collision) และนำไปใช้ไม่ได้ สถานีจะต้องหยุดส่งและสุ่มเวลาเพื่อเข้าใช้สายสัญญาณใหม่

ในเครือข่ายอีเทอร์เน็ตที่มีสถานีจำนวนมากมักพบว่าการทำงานจะล่าช้าเพราะแต่ละสถานีพยายามยึดครองสัญญาณเพื่อส่งข้อมูลและเกิดการชนกันเกือบตลอดเวลาโดยไม่สามารถกำหนดได้ว่าสถานีใดจะได้ใช้สายสัญญาณเมื่อเวลาใด อีเทอร์เน็ตจึงไม่เหมาะกับการใช้งานในระบบเวลาจริง

2.1.2 IEEE 802.4

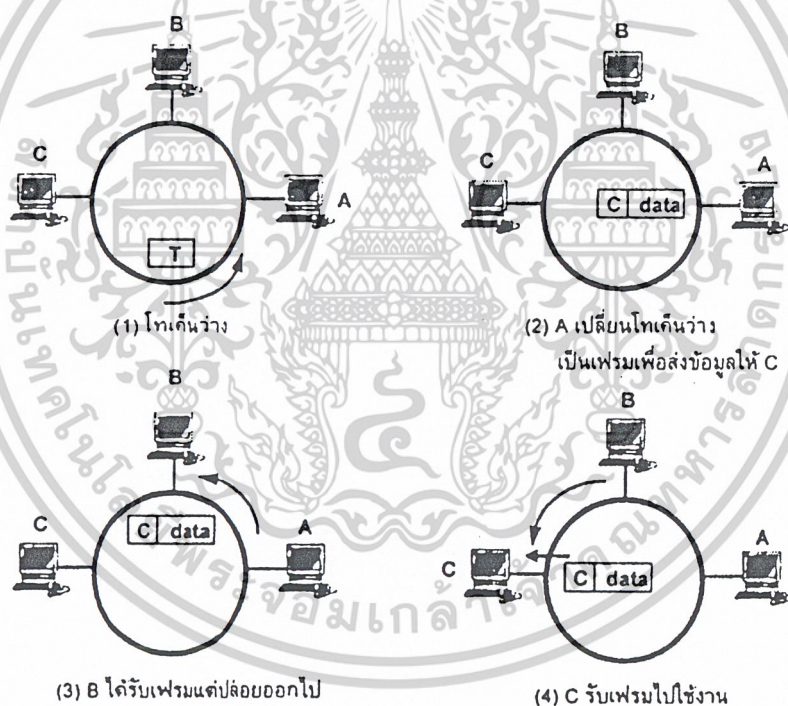
IEEE 802.4 หรือ โทเค็นบัส (Token Bus) มีโทโปโลยีแบบบัสเช่นเดียวกับ IEEE 802.3 แต่มีข้อกำหนดการเข้าใช้สายสื่อสารโดยใช้โทเค็นพิเศษซึ่งทำหน้าที่เป็นแฟรมสัญญาณ กำหนดจังหวะให้สถานีเข้าใช้สายสื่อสาร โทเค็นจะถูกนำส่งจากสถานีหนึ่งไปยังสถานีหนึ่งและวนกลับที่เดิมเป็นวงรอบ สถานีที่ได้รับโทเค็นจะมีสิทธิ์ใช้สายสื่อสารเพื่อส่งข้อมูลได้

สายสื่อสารในโทเค็นบัสมักใช้สายโคแอกเชียล และมีอัตราเร็วหลายระดับคือ 1, 5 หรือ 10 เมกะบิตต่อวินาที การใช้โทเค็นช่วยให้สถานีไม่ต้องแย่งชิงช่องสัญญาณเหมือนใน IEEE 802.3 หากแต่ความซับซ้อนของโปรโตคอลทำให้ IEEE 802.4 ไม่เป็นที่นิยมใช้

2.1.3 IEEE 802.5

IEEE 802.5 หรือ โทเค็นริง (Token Ring) หรือมักเรียกว่าไอบีเอ็มโทเค็นริง จัดเป็นเครือข่ายที่ใช้โทโพลีแบบวงแหวนด้วยสายคู่ตีเกลียวหรือเส้นใยนำแสง อัตราการส่งข้อมูลของโทเค็นริงที่ใช้โดยทั่วไปคือ 4 และ 16 เมกะบิตต่อวินาที

รูปที่ 2-3 แสดงการทำงานของโทเค็นริง โดยมีเฟรมพิเศษเรียกว่า โทเค็นว่าง (free token) วิ่งวนอยู่ สถานีที่ต้องการส่งข้อมูลและรอให้โทเค็นว่างเดินทางมาถึงแล้วรับโทเค็นว่างมาเปลี่ยนเป็น เฟรมข้อมูล (data frame) โดยใส่แฟล็กแสดงเฟรมข้อมูลและบรรจุแอดเดรสของสถานีต้นทางและปลายทางตลอดจนข้อมูลอื่นๆ จากนั้นสถานีจึงปล่อยเฟรมนี้ออกไป



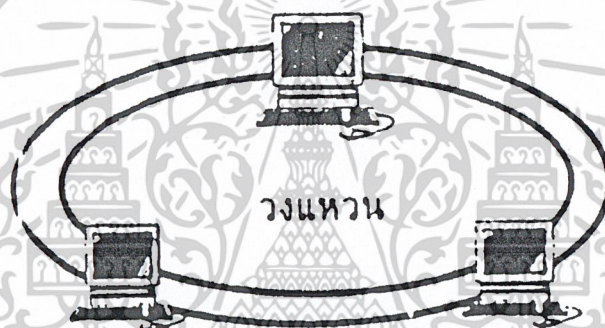
รูปที่ 2-3 การทำงานโทเค็นริง

เมื่อสถานีปลายทางได้รับเฟรมจะสำเนาข้อมูลไว้และปล่อยเฟรมให้วนกลับมายังสถานีส่ง สถานีส่งจะตรวจสอบเฟรมและปล่อยโทเค็นว่างคืนสู่เครือข่ายให้สถานีอื่นมีโอกาสดำเนินการต่อไป กลไกแบบส่งผ่านโทเค็นจัดอยู่ในประเภทประเมินเวลาได้ กล่าวคือสามารถคำนวณเวลาสูงสุดที่สถานีมีสิทธิจับโทเค็นเพื่อส่งข้อมูลได้ โทเค็นจึงเหมาะกับระบบที่ต้องการความแน่นอนทางเวลาหรืองานแบบเวลาจริง

2.1.4 เอฟดีดีไอ

เอฟดีดีไอ (FFDI : Fiber Distributed Data Interface) มีอัตราส่งข้อมูล 100 เมกะบิตต่อวินาที จึงมักใช้เป็น แกนหลัก (Backbone) ซึ่งเป็นส่วนของเครือข่ายที่จะต้องรับภาระการสื่อสารในปริมาณมาก และนิยมใช้กับระบบงานเวลาจริงเช่นเดียวกับโทเค็นริงเนื่องจากใช้หลักการของโทเค็นเช่นเดียวกัน

โทโปโลยีเอฟดีดีไอ เป็นแบบวงแหวนคู่ดังรูปที่ 2-4 ในขณะที่โทเค็นริงมีเพียงวงเดียวเท่านั้น วงแหวนประกอบด้วยวงแหวนหลักเรียกว่า วงแหวนปฐมภูมิ (primary ring) และวงแหวนรองเรียกว่า วงแหวนทุติยภูมิ (secondary ring) การออกแบบให้มีวงแหวนสองวงเพื่อมีเส้นทางสำรอง หากวงแหวนหลักเกิดชำรุด ระบบจะเปลี่ยนไปใช้วงแหวนสำรองแทน



รูปที่ 2-4 โทโปโลยีเอฟดีดีไอ

สายสัญญาณที่ใช้กับเอฟดีดีไอมี 2 ประเภทคือเส้นใยนำแสงและสายคู่ตีเกลียว เส้นใยนำแสงที่ใช้งานมีทั้งแบบ หลายภาวะ (Multi mode) และภาวะเดี่ยว (Single mode) เส้นใยนำแสงแบบหลายภาวะสามารถเชื่อมสถานีที่อยู่ห่างกันได้ 2 กิโลเมตร ขณะที่แบบภาวะเดี่ยวสามารถเชื่อมสถานีที่อยู่ห่างกันได้ระยะทางราว 40 กิโลเมตร ส่วนสายคู่ตีเกลียวจะใช้สายประเภท 5 (Category 5) เอฟดีดีไอที่ใช้สายคู่ตีเกลียวมีชื่อเรียกเฉพาะว่า ซีดีดีไอ (CDDI : Copper Distributed Data Interface)

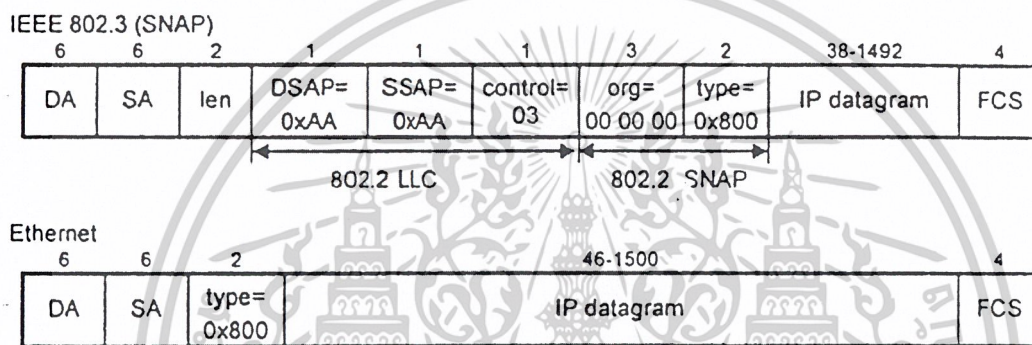
2.1.5 แสนิป

ค่า DSAP/SSAP ใน 802.2 LLC ถูกกำหนดโดยไอเอสไอเพื่อใช้กับโปรโตคอลตามรูปแบบ ไอเอสไอ เช่นกำหนดให้ใช้หมายเลข 6 สำหรับไอพี ทั้งนี้เพื่อให้ไอพีทำงานร่วมกับโปรโตคอลของ ไอเอสไอแลนได้ แต่ในเครือข่ายทั่วไปแล้วจะใช้ไอพีอย่างอิสระโดยไม่ขึ้นกับไอเอสไอ การเอ็นแคปซูล ไอพีกับ IEEE 802.3 จึงมีรูปแบบที่กำหนดขึ้นมาโดยเฉพาะและเรียกว่าเฟรม แสนิป (SNAP :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Subnetwork Access Protocol) ด้วยเหตุนี้เราจึงพบเฟรมในเครือข่ายอีเทอร์เน็ตที่บรรจุข้อมูลไอพีสองรูปแบบคือแบบเส้นที่ซึ่งใช้เฟรม IEEE 802.3 (RFC 1042) และแบบอีเทอร์เน็ตดั้งเดิม (RFC 894) รูปที่ 2-5 แสดงการเ็นแคปซูลเฟรมทั้งสองแบบ โปรดสังเกตว่าฟิลด์ DSAP และ SSAP ในเส้นที่จะมีค่าเท่ากับ 0xAA และฟิลด์ control มีค่าเท่ากับ 3 ส่วนของฟิลด์เส้นที่จะมีเฮดเดอร์เพิ่มอีก 2 ฟิลด์คือ

- org (organization code) ขนาด 3 ไบต์ : ระบุรหัสขององค์กรซึ่งทำหน้าที่กำหนดเลขโปรโตคอล โดยปกติแล้วจะมีค่าเป็น 00 (ซึ่งหมายถึงบริษัทซีร็อกซ์)
- type ขนาด 2 ไบต์ : ชนิดโปรโตคอลระดับบน หากเป็นไอพีจะมีค่า 0x0800



รูปที่ 2-5 เปรียบเทียบการเ็นแคปซูลอีเทอร์เน็ตแบบเส้นที่และแบบดั้งเดิม

ชื่อที่ใช้เรียกรูปแบบการเ็นแคปซูลเฟรมของอีเทอร์เน็ตในปัจจุบันมีหลายชื่อและมักทำให้ผู้ใช้งานสับสนได้ง่าย ในที่นี้จะสรุปชื่อที่มีใช้อยู่ดังต่อไปนี้

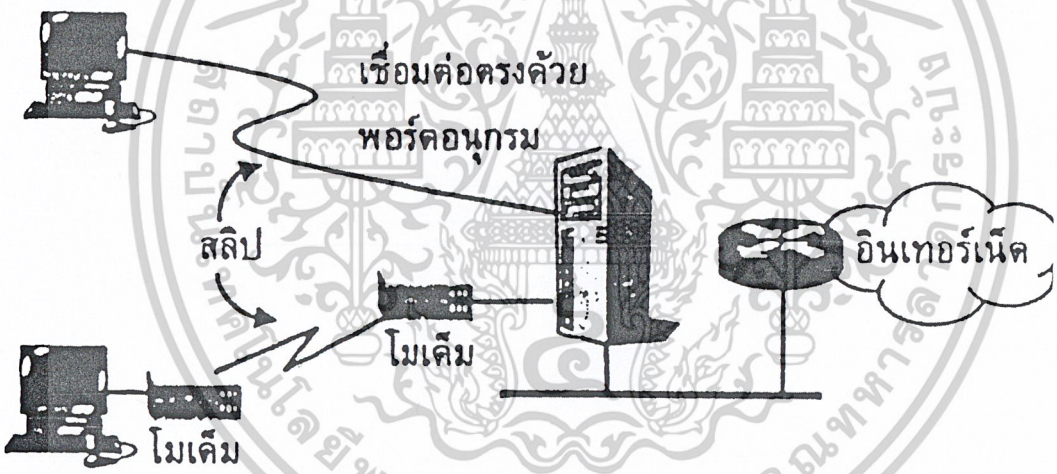
- เฟรมที่เ็นแคปซูลตามอีเทอร์เน็ตดั้งเดิม เรียกว่า Ethernet
- เฟรมที่เ็นแคปซูลโดยใช้ 802.2 LLC เรียกว่า IEEE 802.3 แต่ในระบบปฏิบัติการเน็ตแวร์จะเรียกการเ็นแคปซูลนี้ตามศัพท์เทคนิคของตนเองว่า Ethernet 802.2
- เฟรมที่เ็นแคปซูลโดยใช้ 802.2 LLC แบบเส้นที่เรียกว่า Ethernet SNAP หรือ 802.2 SNAP
- ในบางระบบปฏิบัติการเช่นเน็ตแวร์จะใช้เฟรมตามแบบ IEEE 802.3 แต่จะตัดฟิลด์ 802.2 LLC ออกไป กล่าวคือตัดจากฟิลด์ len แล้วจะต่อด้วยเฮดเดอร์ของโปรโตคอลไอพีเอ็กซ์ของเน็ตแวร์ทันที การเ็นแคปซูลแบบนี้เรียกว่า raw 802.3

2.2 โปรโตคอลจุดต่อจุด

การเชื่อมโยงเป็นเครือข่ายบางสภาพมีความจำเป็นต้องการใช้การสื่อสารอนุกรมแบบอะซิงโครนัสเนื่องจากสาเหตุหลายประการเช่น ข้อจำกัดด้านการลงทุน หรือต้องใช้เครือข่ายผ่านโมเด็ม หรือใช้สายต่อเชื่อมโดยตรงระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ ทีซีพี/ไอพีสามารถรองรับการทำงานลักษณะนี้โดยใช้โปรโตคอลระดับเตทาลิงค์แบบจุดต่อจุดที่ออกแบบมาเพื่อการสื่อสารแบบอนุกรม โปรโตคอลแบบจุดต่อจุดที่ใช้แพร่หลายในปัจจุบันได้แก่ สลิป (SLIP : Serial Line Protocol) และ พีพีพี (PPP : Point-to-Point Protocol)

2.2.1 สลิป

สลิปเป็นโปรโตคอลที่ออกแบบเพื่อการเชื่อมโยงแบบจุดต่อจุดระหว่างอุปกรณ์แบบอะซิงโครนัส อุปกรณ์ที่เชื่อมโยงอาจเป็นคอมพิวเตอร์เชื่อมกันโดยตรงหรือคอมพิวเตอร์เชื่อมกับเราเตอร์ หรือเราเตอร์กับเราเตอร์ การเชื่อมโยงอาจเป็นการต่อเชื่อมตรงหากมีระยะทางสั้นๆ หรือใช้โมเด็มในกรณีที่อุปกรณ์ตั้งอยู่ห่างกันดังรูปที่ 2-6



รูปที่ 2-6 การเชื่อมโยงด้วยสลิป

หลักการทำงานของสลิป

สลิปเป็นโปรโตคอลที่เรียบง่ายและไม่มีกลไกใดรองรับปัญหาความผิดพลาดข้อมูล สลิปปล่อยหน้าที่ส่วนใหญ่ให้เป็นภาระของโปรโตคอลระดับบน แต่เนื่องจากโปรโตคอลระดับประยุกต์บางประเภทซึ่งใช้บริการยูทิลิตี้จะถือว่าปัญหาข้อผิดพลาดการส่งข้อมูลเป็นภาระของโปรโตคอลในระดับชั้นเตทาลิงค์ สลิปจึงไม่เหมาะกับโปรโตคอลประยุกต์ในกลุ่มนี้

การเชื่อมโยงระหว่างต้นทางและปลายทางต้องใช้ไอพีแอดเดรส 2 ค่าที่อยู่ในเครือข่ายเดียวกัน สลิปไม่มีกลไกตรวจสอบแอดเดรสของอีกด้านหนึ่ง ผู้ใช้จึงต้องกำหนดแอดเดรสด้วยตนเอง การใช้

งานสลับในกรณีที่ต้องเชื่อมโยงไปยังเครือข่ายปลายทางที่มีหลายแห่งจึงไม่สะดวกเพราะต้องกำหนดไอพีแอดเดรสให้เข้ากับปลายทางเสมอ

2.2.2 พีพีพี

พีพีพีเป็นโปรโตคอลระดับชั้นเดทาลิงก์แบบจุดต่อจุดอีกโปรโตคอลหนึ่งที่สามารถทำงานร่วมกับไอพีในการสื่อสารแบบอนุกรมทั้งแบบอะซิงโครนัส พีพีพีมีข้อเด่นที่ชัดเจนข้อบกพร่องที่พบในสลิป กล่าวคือรองรับการทำงานกับโปรโตคอลระดับชั้นเน็ตเวิร์กได้หลายโปรโตคอลเช่น เดคเน็ต ไอพีเอ็กซ์ หรือ ไอพี

ภายในพีพีพีมีโปรโตคอลย่อยที่สนับสนุนการติดตั้ง ปรับแต่ง ทดสอบการเชื่อมโยงระหว่างต้นทางกับปลายทาง รวมทั้งมีกลไกตรวจสอบความถูกต้องของข้อมูลที่ได้รับ พีพีพีมีกระบวนการตรวจสอบเพื่อกำหนดไอพีแอดเดรสระหว่างต้นทางและปลายทางได้โดยไม่ต้องตั้งค่าโดยตรง ด้วยขีดความสามารถเหล่านี้พีพีพีจึงเป็นโปรโตคอลที่นิยมใช้มากกว่าสลิป

โปรโตคอลประกอบพีพีพี

พีพีพีมีโปรโตคอล แอลซีพี (LCP : Link Control Protocol) ทำหน้าที่สถาปนากำหนดรูปแบบและทดสอบการเชื่อมโยง และอาศัยโปรโตคอลซึ่งรวมเรียกว่า เอ็นซีพี (NCP : Network Control Protocol) เพื่อสถาปนาและติดตั้งรูปแบบการเชื่อมโยงตามโปรโตคอลระดับชั้นเน็ตเวิร์กแต่ละชนิดที่ใช้ งาน ตัวอย่างเช่นไอพีซีพี(IPCP : IP Control Protocol) ทำหน้าที่ร่วมกับไอพี เอ็นซีพีช่วยให้พีพีพีสนับสนุนการทำงานได้หลายโปรโตคอลในเวลาเดียวกัน การติดตั้งและเลือกใช้พารามิเตอร์กำหนดเฟรมข้อมูลเป็นหน้าที่ของแอลซีพี

2.3 ไอพีและเทคโนโลยีเครือข่ายอื่น

ที่ซีพี/ไอพีมีลักษณะสากลและทำงานร่วมกับโปรโตคอลอื่นได้ทั้งกลุ่มโปรโตคอลเครือข่ายเฉพาะที่ (เน็ตไบออส, ไอพีเอ็กซ์ และเอ็กซ์เอ็นเอส) และกลุ่มโปรโตคอลเครือข่ายระยะไกล (X.25, เฟรมรีเลย์ และเอสเอ็มดีเอส)

จากการขยายตัวของเทคโนโลยีเอทีเอ็มได้มีการพัฒนาโปรโตคอลเพื่อให้ไอพีสามารถทำงานร่วมกับเอทีเอ็มได้ ในปัจจุบันมีโปรโตคอลหลายรูปแบบเช่น IPOA (IP Over ATM), LANE (LAN Emulation) และ MPOA (Multiprotocol Over ATM) [Gins96]

ถึงแม้ว่าเอทีเอ็มจะให้เส้นทางออกหนึ่งสำหรับที่ซีพี/ไอพี ความเร็วสูง แต่เอทีเอ็มก็มีโอเวอร์เฮดสูงเนื่องจากเซลล์เอทีเอ็มมีขนาดเพียง 53 ไบต์และมีเฮดเดอร์ 5 ไบต์ แนวทางการใช้ไอพีทำงานบนโซเน็ต/เอสดีเอช (SONET/SDH) โดยตรงจึงเป็นอีกหัวข้อหนึ่งที่อยู่ในความสนใจของนักวิจัย [MADD98]

2.4 ลูปแบ็คอินเทอร์เฟซ

ลูปแบ็คอินเทอร์เฟซ (Loopback Interface) เป็นอินเทอร์เฟซทางซอฟต์แวร์สำหรับให้อุปกรณ์ในโฮสต์เดียวกันสามารถสื่อสารถึงกันโดยไม่ต้องส่งแพ็กเก็ตออกไปนอกโฮสต์แล้วย้อนกลับเข้ามาอีกที ลูปแบ็คอินเทอร์เฟซจะทำหน้าที่เป็นเสมือนเคทาลิงค์ประเภทหนึ่งที่เชื่อมต่อระดับชั้นเน็ตเวิร์ก

โปรโตคอลไอพีจะสงวนแอดเดสซึ่งขึ้นต้นด้วย 127 ไว้ใช้เป็นแอดเดรสลูปแบ็ค เช่นใช้ค่า 127.0.0.1 และมักกำหนดชื่อโฮสต์ประจำลูปแบ็คด้วยชื่อ localhost

2.5 เอ็มทียู

เฟรมของโปรโตคอลในระดับชั้นเคทาลิงค์สามารถบรรจุข้อมูลได้สูงสุดที่ค่าหนึ่งๆ เช่น อีเทอร์เน็ตบรรจุได้ 1500 ไบต์ และ 802.3 บรรจุได้ 1492 ไบต์ ขนาดข้อมูลนี้เป็นลักษณะสมบัติเฉพาะตัวของเคทาลิงค์แต่ละประเภทและเรียกว่า เอ็มทียู (MTU : Maximum Transfer Unit) หากไอพีจำเป็นต้องส่งข้อมูลเกินกว่าขนาดเอ็มทียูของเคทาลิงค์ ไอพีจะแยกข้อมูลเป็นหลายเฟรมเพื่อให้ข้อมูลไม่เกินค่าเอ็มทียู กระบวนการแยกเฟรมนี้เรียกว่า การแฟร็กเมนต์ (fragmentation)

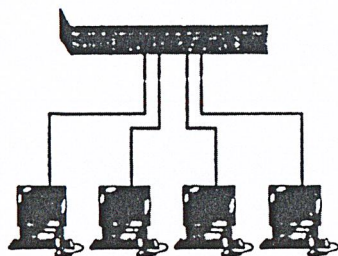
2.6 อุปกรณ์เครือข่าย

เครือข่ายขนาดเล็กมักประกอบด้วยสถานีงานและเซิร์ฟเวอร์ต่อเชื่อมกัน การเชื่อมเครือข่ายขนาดเล็กจำนวนหลายเครือข่ายเข้าด้วยกันเพื่อแลกเปลี่ยนข้อมูลระหว่างกันต้องอาศัยอุปกรณ์เครือข่ายที่มีลักษณะสมบัติแตกต่างกันไป อุปกรณ์เครือข่ายพื้นฐานที่พบโดยทั่วไปและจะอธิบายถึงในหัวข้อนี้ได้แก่ ฮับ บริดจ์ และเราเตอร์

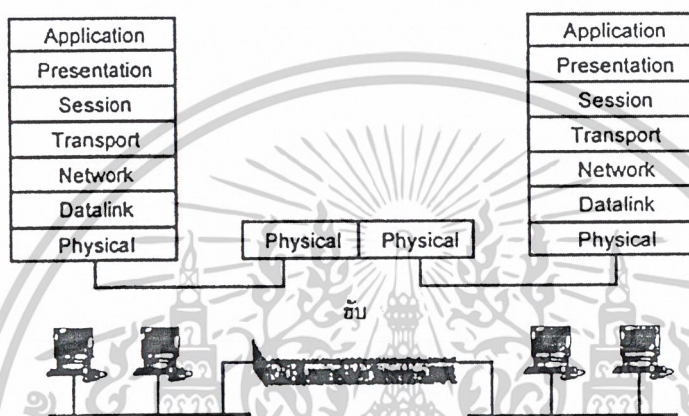
2.6.1 ฮับ

ฮับ (Hub) เป็นอุปกรณ์เชื่อมสถานีเครือข่ายที่ใช้โทโปโลยีแบบดาว เช่น อีเทอร์เน็ต 10BaseT ดังรูปที่ 2-7 ฮับมีพอร์ตได้หลายแบบเช่น พอร์ต RJ-45 ใช้กับสายคู่ตีเกลียวในเครือข่าย 10BaseT หรือพอร์ตไฟเบอร์ใช้กับเส้นใยนำแสงในเครือข่าย 10BaseF

ฮับเป็นอุปกรณ์ในระดับชั้นฟิสิคัลเช่นเดียวกับบริดจ์ดังรูปที่ 2-8 ด้วยเหตุนี้จึงมันเรียกฮับว่าเป็นมัลติพอร์ตรีพีตเตอร์ หน้าที่ของฮับคือขยายสัญญาณและกระจายแพ็กเก็ตไปทุกพอร์ต ฮับใช้เชื่อมต่อเครือข่ายประเภทเดียวกันเท่านั้น เครือข่ายที่เชื่อมด้วยฮับจะรวมเป็นเครือข่ายเดียวกันดังนั้นแพ็กเก็ตที่สร้างจากเครือข่ายหนึ่งจะผ่านฮับไปอีกเครือข่ายหนึ่ง



รูปที่ 2-7 การใช้ฮับในโทปโลยีแบบดาว

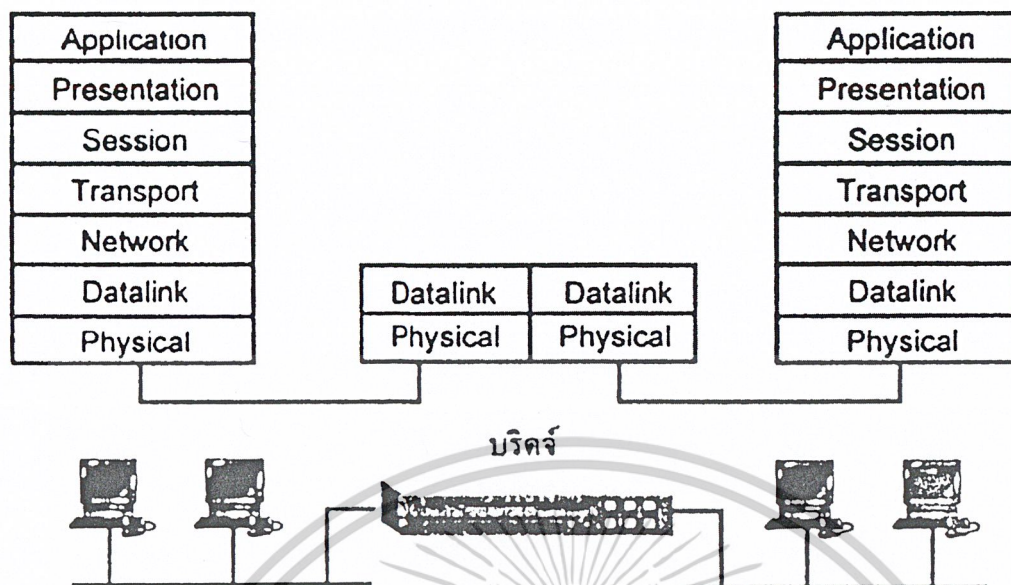


รูปที่ 2-8 แบบจำลองการทำงานของฮับ

2.6.2 บริดจ์

บริดจ์ (Bridge) ทำหน้าที่เชื่อมต่อเครือข่ายย่อยสองเครือข่ายเข้าด้วยกัน บริดจ์ทำงานในระดับชั้นเดทาลิงก์จึงสามารถใช้เชื่อมต่อเครือข่ายประเภทเดียวกันหรือต่างกันก็ได้ เช่นเชื่อมต่ออีเทอร์เน็ตเข้ากับโทเค็นริง แบบจำลองการทำงานของบริดจ์แสดงได้ดังรูปที่ 2-9

เมื่อใช้บริดจ์เชื่อมต่อเครือข่ายประเภทเดียวกัน บริดจ์จะตรวจสอบว่าจะนำส่งแพ็กเก็ตเกิดโดยตรวจสอบฮาร์ดแวร์แอดเดรสปลายทางกับตารางเลือกเส้นทางที่บริดจ์สร้างขึ้น หากมีแพ็กเก็ตที่ต้องข้ามบริดจ์จากเครือข่ายหนึ่งไปยังอีกเครือข่ายหนึ่ง บริดจ์จะค้นหาแพ็กเก็ตบิตต่อบิตข้ามไปยังอีกเครือข่ายหนึ่งโดยไม่เปลี่ยนแปลงเฮดเดอร์หรือข้อมูล หากสถานีต้นทางและสถานีปลายทางอยู่ในเครือข่ายเดียวกัน บริดจ์จะได้รับแพ็กเก็ตแต่ไม่ส่งแพ็กเก็ตข้ามไปยังเครือข่ายอีกด้านหนึ่ง



รูปที่ 2-9 แบบจำลองการทำงานของบริดจ์

บริดจ์นำส่งบรอดคาสต์แพ็กเก็ตข้ามเครือข่ายโดยไม่ป้องกัน สถานีอีเทอร์เน็ตหรือโทเค็นริงต้องดำเนินการกับแพ็กเก็ตที่สถานีอื่นบรอดคาสต์มาให้ บรอดคาสต์แพ็กเก็ตมันเกิดจากการทำงานของโปรโตคอลเออาร์พี, บูตพี, หรืออาร์ไอพี หากจำนวนบรอดคาสต์แพ็กเก็ตสูงจะส่งผลให้สมรรถนะการทำงานของสถานีในเครือข่ายลดลง

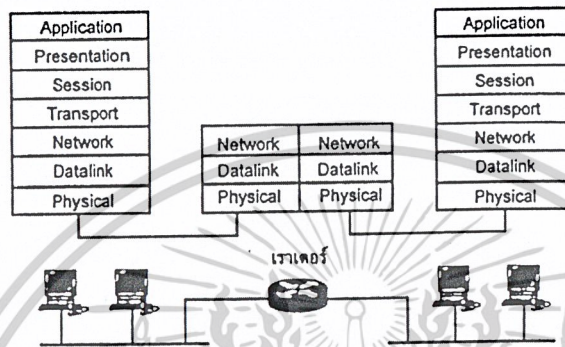
หากสถานีเครือข่ายสร้างบรอดคาสต์แพ็กเก็ตขึ้นพร้อมกันจำนวนมากหรือเรียกว่าพายุบรอดคาสต์ (Broadcast Storms) จะส่งผลกระทบต่อเครือข่าย การสื่อสารในเครือข่ายจะช้าจนสังเกตได้หากมีบรอดคาสต์แพ็กเก็ตตั้งแต่ 40 แพ็กเก็ตต่อวินาทีขึ้นไป หรือหากมีบรอดคาสต์แพ็กเก็ตในอัตรา 100 แพ็กเก็ตต่อวินาทีอาจทำให้การสื่อสารทั้งหมดหยุดชะงักได้

สวิตช์ทำหน้าที่เช่นเดียวกับบริดจ์ในการแบ่งเครือข่ายใหญ่ออกเป็นเครือข่ายย่อย และป้องกันไม่ให้มีการส่งเฟรมที่ไม่จำเป็นจากเครือข่ายหนึ่งข้ามไปยังอีกเครือข่ายหนึ่ง สวิตช์มักประกอบด้วยพอร์ตจำนวนมาก ต่างจากบริดจ์ที่มีพอร์ตเพียง 2 พอร์ต แต่ละพอร์ตของสวิตช์สามารถเชื่อมโยงระหว่างสวิตช์ด้วยกันหรือเชื่อมสถานีเข้าสู่สวิตช์โดยตรง สวิตช์บางรุ่นสามารถจัดแบ่งเครือข่ายย่อยตามแต่ละพอร์ตได้โดยใช้ วีแลน (VLAN : Virtual LAN) ตามมาตรฐาน IEEE 802.1q

คุณลักษณะสำคัญของสวิตช์คือมีอิเล็กทรอนิกส์สวิตช์ความเร็วสูงทำหน้าที่ส่งเฟรมจากพอร์ตต้นทางสู่ปลายทางโดยไม่รบกวนพอร์ตอื่น ลักษณะนี้ต่างจากบริดจ์หรือเราเตอร์ซึ่งเฟรมถูกส่งจากพอร์ตหนึ่งไปอีกพอร์ตหนึ่งโดยอาศัยการประมวลผลของไมโครโปรเซสเซอร์

2.6.3 เราเตอร์

เราเตอร์เป็นอุปกรณ์ที่ทำงานในระดับชั้นเน็ตเวิร์คตามรูปที่ 2-10 เราเตอร์ทำงานร่วมกับฮาร์ดแวร์ในระดับเคทาลิงค์ได้หลายรูปแบบ หน้าที่ของเราเตอร์คือจัดแบ่งเครือข่ายและเลือกเส้นทางที่เหมาะสมเพื่อนำส่งแพ็กเก็ต เราเตอร์จะป้องกันการบรอดคาสต์แพ็กเก็ตจากเครือข่ายหนึ่งไม่ให้ข้ามมายังอีกเครือข่ายหนึ่ง



รูปที่ 2-10 แบบจำลองการทำงานของเราเตอร์

ในอินเทอร์เน็ตมักเรียกเราเตอร์ว่า ไอพีเราเตอร์ (IP router) เนื่องจากเราเตอร์ทำงานตามข้อกำหนดของโปรโตคอลไอพี เราเตอร์ทำหน้าที่เลือกเส้นทางโดยสร้างแผนที่เครือข่ายและเก็บอยู่ในรูปตารางเส้นทาง เมื่อเราเตอร์ได้รับแพ็กเก็ตก็จะตรวจสอบแอดเดรสปลายทางและส่งแพ็กเก็ตไปยังอินเทอร์เน็ตที่เป็นช่องทางไปสู่เครือข่ายปลายทาง เราเตอร์ประกอบด้วยหลายอินเทอร์เน็ตเพื่อเชื่อมต่อกับเครือข่ายต่างชนิดเข้าด้วยกันได้ ตัวอย่างเช่น อินเทอร์เน็ตอีเทอร์เน็ต โทเค็นริง เอฟดีดีไอ เอทีเอ็ม หรือ อินเทอร์เน็ตแบบอนุกรม เป็นต้น

ซอฟต์แวร์ในเราเตอร์จะรับแพ็กเก็ตและเก็บเข้าบัฟเฟอร์ก่อนนำไปประมวลผล วิธีนี้ต่างจากสวิตช์ซึ่งฮาร์ดแวร์ตรวจสอบแอดเดรสปลายทางของแพ็กเก็ตและ “สวิตช์” แพ็กเก็ตนั้นจากพอร์ตหนึ่งไปสู่อีกพอร์ตหนึ่ง

บทที่ 3

โพรโตคอลทีซีพี/ไอพี

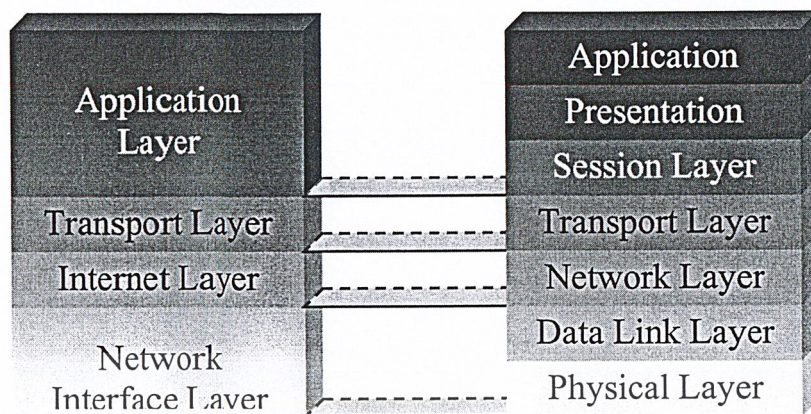
3.1 ความเป็นมาของโพรโตคอลทีซีพี/ไอพี

ทีซีพี/ไอพี เป็นโพรโตคอลมาตรฐานที่ใช้กันอยู่ในระบบปฏิบัติการแบบยูนิกซ์ เริ่มพัฒนาโดยกระทรวงกลาโหมของสหรัฐฯ ในปี ค.ศ. 1969 เพื่อเชื่อมโยงเครื่องคอมพิวเตอร์ทางทหารของแต่ละหน่วยที่อยู่ห่างไกลกัน โดยมีจุดประสงค์ คือสร้างระบบเครือข่ายให้เครื่องคอมพิวเตอร์สามารถรับส่งข้อมูลกันได้ แม้ว่าสายส่งข้อมูลบางส่วนจะถูกทำลายเสียหายไปก็ตาม เพื่อใช้งานในยามเกิดสงคราม โดยเครือข่ายที่จัดตั้งในระยะแรกชื่อว่า Advanced Research Projects Agency Network หรือ อาร์พานเน็ต (ARPANET)

ต่อมาได้พัฒนาเป็นเครือข่ายอินเทอร์เน็ต (INTERNET) โพรโตคอลนี้เหมาะสำหรับเชื่อมต่อคอมพิวเตอร์ทั้งใกล้ และไกลเข้าด้วยกัน และมีมาตรฐานรองรับทำให้ผู้ผลิตฮาร์ดแวร์ และซอฟต์แวร์สามารถสร้างอุปกรณ์ และโปรแกรมที่จะรองรับการทำงานของโพรโตคอลนี้ ทำให้เครื่องคอมพิวเตอร์สามารถรับส่งข้อมูลกันได้ไม่ว่าจะเป็นเครื่องขนาดเล็กหรือขนาดใหญ่ หรือใช้ระบบปฏิบัติการอะไรก็ตาม ทีซีพี/ไอพี (TCP/IP) เป็นชุดโพรโตคอลที่ประกอบด้วยโพรโตคอลต่างๆ หลายโพรโตคอล แต่ละโพรโตคอลมีคุณลักษณะ และมีความสามารถในการทำงานแตกต่างกัน โดยที่ในบทนี้ได้กล่าวถึงรายละเอียดและคุณสมบัติของโพรโตคอลที่สำคัญบางโพรโตคอล

3.2 การเชื่อมต่อของโพรโตคอลทีซีพี/ไอพี (TCP/IP Linking)

ทีซีพี/ไอพี (TCP/IP หรือ Transmission Control Protocol/Internet Protocol) เป็นโพรโตคอลในการสื่อสารในระบบอินเทอร์เน็ต และอินทราเน็ต มีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ของฝ่ายรับ และฝ่ายส่งให้ได้รับข้อมูลที่ถูกต้องครบถ้วน หากข้อมูลที่ส่งมาเกิดการสูญหายระหว่างทางจะมีการแจ้งให้ต้นทางส่งข้อมูลมาใหม่ การทำงานของทีซีพี/ไอพีสามารถเปรียบเทียบกับโมเดลอ้างอิงโอเอสไอ (Open System Interconnection Reference Model: OSI) ตามมาตรฐานไอเอสโอ (International Organization for Standardization: ISO) ได้ดังรูปที่ 3-1



รูปที่ 3-1 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพี/ไอพี

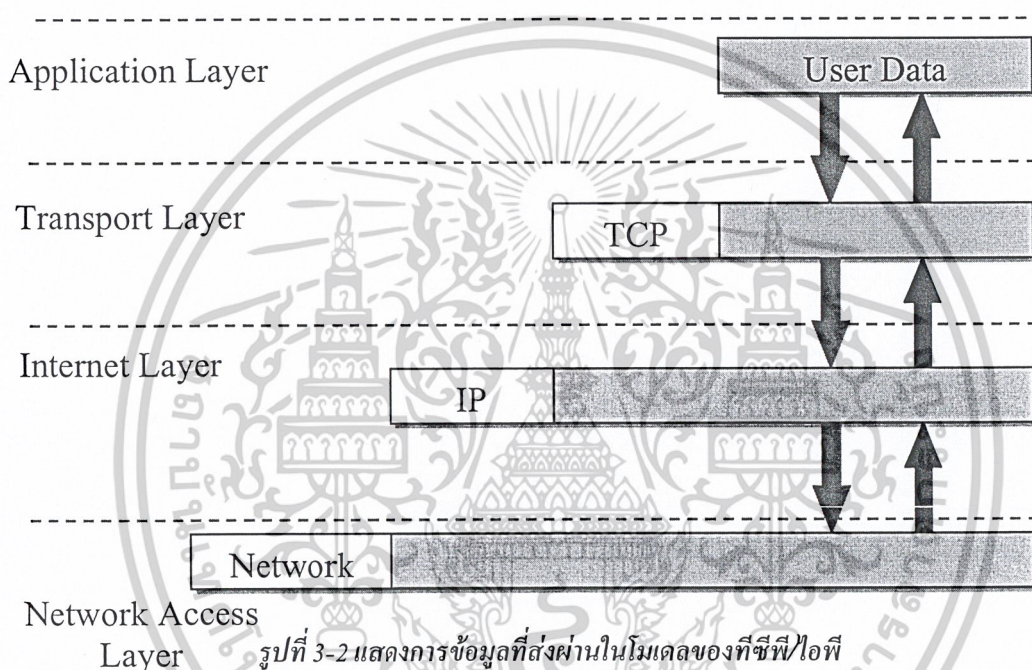
ในแต่ละระดับชั้นของทีซีพี/ไอพีมีการทำงานที่แตกต่างกัน ตั้งแต่การติดต่อกับแอปพลิเคชัน จนกระทั่งแปลงเป็นสัญญาณส่งไปตามสายสัญญาณ ซึ่งการทำงานในแต่ละระดับชั้นของทีซีพี/ไอพี มีดังนี้

ชื่อระดับชั้น	หน้าที่
1. ชั้นแอปพลิเคชัน (Application Layer)	รองรับการทำงานของแอปพลิเคชันต่างๆ ที่ทำงานเป็นโพรเซสอยู่ในเครื่องต้นทางและปลายทาง โดยจัดการเชื่อมต่อระหว่างโพรเซส หรือแอปพลิเคชันที่อยู่ต่างเครื่องกัน โดยการทำงานของแอปพลิเคชันต่างๆ มีการติดต่อกันตามแต่ละโพรโตคอลเฉพาะแล้วแต่แอปพลิเคชันที่ใช้งาน ซึ่งจะขอบริการจากชั้นทรานสปอร์ตอีกทีหนึ่ง
2. ชั้นทรานสปอร์ต (Transport Layer)	สร้างการเชื่อมต่อกันระหว่างแอปพลิเคชันแบบ end-to-end โดยจุดที่เชื่อมต่อกันเพื่อรับส่งข้อมูลนี้เรียกว่า พอร์ต (port) หรือซ็อกเก็ต (Socket) ในชั้นนี้มีบริการหลักอยู่ 2 แบบ คือ Connection Oriented โดยเรียกผ่านโพรโตคอลทีซีพี (TCP: Transmission Control Protocol) และ Connectionless ซึ่งเรียกผ่านโพรโตคอลยูดีพี (UDP: User Datagram Protocol) ซึ่งกล่าวถึงในหัวข้อถัดไป
3. ชั้นอินเทอร์เน็ต (Internet Layer)	ส่งผ่านข้อมูลระหว่างเครือข่าย โดยมีโพรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใดๆ ในอินเทอร์เน็ต คือ ไอพี (Internet Protocol: IP) ซึ่งกล่าวถึงในหัวข้อถัดไป นอกจากนี้ในชั้นนี้ยังมีโพรโตคอลทำงานอยู่ด้วยอีก 2 ชนิด คือ ไอซีเอ็มพี (Internet Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Message Protocol: ICMP) และเออาร์พี (Address Resolution Protocol: ARP)
4. ชั้นเน็ตเวิร์กอินเทอร์เฟซ (Network Interface Layer)	แปลงข้อมูลให้อยู่ในรูปที่เหมาะสมกับเครือข่ายแต่ละแบบ ซึ่งแตกต่างกันออกไป และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่าย

ตารางที่ 3-1 การทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี



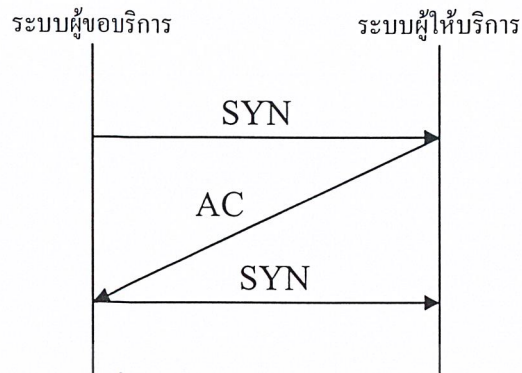
รูปที่ 3-2 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี

ในชุดโพรโทคอลทีซีพี/ไอพีนี้ มีโพรโทคอลหลัก ที่ขอกว่าถึง 3 โพรโทคอล ได้แก่ โพรโทคอลทีซีพี โพรโทคอลยูดีพี ซึ่งทำงานในชั้นทรานสปอร์ต และโพรโทคอลไอพี ซึ่งทำงานในชั้นอินเทอร์เน็ต โดยมีรายละเอียดดังต่อไปนี้

3.3 โพรโทคอลทีซีพี (TCP: Transmission Control Protocol)

การทำงานที่สำคัญอย่างหนึ่งของโพรโทคอลทีซีพี คือ การทำ “3-Way Handshake” ซึ่งเป็นกระบวนการเริ่มต้นในการสร้างการเชื่อมต่อในชั้นทรานสปอร์ต กล่าวคือ ในการติดต่อกันระหว่างระบบ ในเครือข่ายต้องมีการสร้างการเชื่อมต่อไปยังระบบที่ให้บริการก่อน โดยผู้ขอบริการส่งสัญญาณ SYN เพื่อขอบริการ จากนั้นผู้ให้บริการจะส่งสัญญาณ ACK เพื่อตอบรับการเชื่อมต่อที่ร้องขอมา จึงสามารถรับส่งข้อมูลกันได้ ดังรูปที่ 3-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 แสดงการทำ 3-Way Handshake

การเชื่อมต่อแบบ 3-Way Handshake นี้ เป็นการตรวจสอบความพร้อมของทั้งฝ่ายส่ง และฝ่ายรับ และการกำหนดค่าเริ่มต้นของพารามิเตอร์ต่างๆ ของทั้งสองฝ่ายให้ตรงกัน หลังจากกระบวนการทำ

3-Way Handshake สิ้นสุด ทั้งสองฝ่ายจึงสามารถรับ และส่งข้อมูลซึ่งกัน และกันได้

ดังนั้นโปรโตคอลทีซีพีจึงเป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ “Connection Oriented” ทำให้การทำงานของทีซีพีมีความน่าเชื่อถือมากขึ้น หน้าที่การทำงานของทีซีพีในการรับส่งข้อมูลมีหน้าที่หลัก 6 ข้อคือ

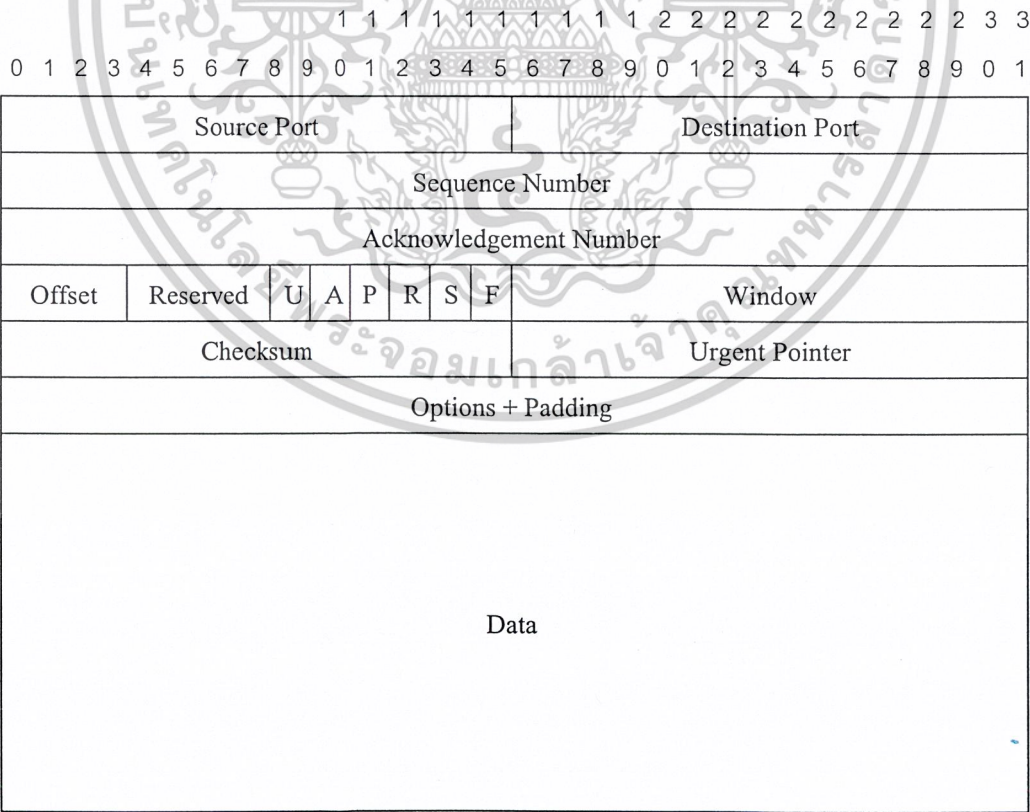
1. ควบคุมการรับส่งข้อมูล (Basic Data Transfer)
2. ความน่าเชื่อถือในการรับส่งข้อมูล (Reliability)
3. ควบคุมการไหลของข้อมูล (Flow Control)
4. การทำมัลติเพล็กซ์ (Multiplexing)
5. ควบคุมการเชื่อมต่อ (Connection)
6. ความปลอดภัยในการรับส่งข้อมูล (Security)

ส่วนประกอบของทีซีพีเฮดเดอร์

1. *Source Port* : เป็นหมายเลขพอร์ตของบริการที่เครื่องต้นทาง
2. *Destination Port* : เป็นหมายเลขพอร์ตของบริการเครื่องปลายทาง
3. *Sequence Number* : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลของเครื่องที่ต้องการขอส่งข้อมูล
4. *Acknowledgement Number* : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลที่ฝั่งรับข้อมูลปกติ ค่าของ Acknowledgement Number มีค่าเท่ากับ Sequence Number (ของอีกฝั่งหนึ่ง) + 1 เสมอ
5. *Data Offset* : เป็นตัวบอกค่าออฟเซตของข้อมูล เพราะทีซีพีนั้นไม่มีการกำหนดความยาวที่แน่นอนของข้อมูล จึงต้องมีออฟเซตเป็นตัวบอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. *Flag* : เป็นบิตที่บอกชนิดของข้อมูล ได้แก่
- URG : Urgent Pointer Field Significant - แสดง Urgent Pointer
 - ACK : Acknowledgement Field Significant – แสดงการ Acknowledgement
 - PSH : Push Function
 - RST : Reset The Connection - แสดงเมื่อรีเซ็ตการเชื่อมต่อ
 - SYN : Synchronize Sequence Number - หมายเลขแพ็กเก็ตที่ส่งแบบซิงโครนัส
 - FIN : No more data from sender - แสดงว่าไม่มีข้อมูลที่ส่งจากผู้ส่งแล้ว
7. *Window* : เป็นเลขบอกจำนวนของอ็อกเต็ต (octet) ของข้อมูล จัดการในส่วนของ end-to-end flow control
8. *Checksum* : เป็นส่วนที่ตรวจสอบความถูกต้องของข้อมูล
9. *Urgent Pointer* : เป็นตัวชี้ตำแหน่งของ Urgent Data
10. *Option and Padding* : เป็นตัวบอกופןชันของโปรเซสที่ใช้ที่ซีพี
11. *Data* : เนื้อข้อมูลที่ต้องการสื่อสาร มีขนาดได้ไม่ต่ำกว่า 5 32-บิตเวิร์ด (6 บิตแรกสงวนไว้และกำหนดให้เป็นศูนย์)



รูปที่ 3-4 แสดงแพ็กเก็ตที่ซีพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โพรโทคอลยูดีพี (UDP: User Datagram Protocol)

โพรโทคอลยูดีพีเป็นโพรโทคอลในการติดต่อสื่อสารในชั้นทรานสปอร์ต (Transport Layer) การทำงานคล้ายกับที่ซีพีมาก คือจัดการเกี่ยวกับการสื่อสารระหว่างเครื่อง แต่เป็นแบบ Connectionless คือทั้งฝ่ายส่ง และฝ่ายรับไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน โดยไม่ต้องมีการแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโพรโทคอลทีซีพี และไม่มีการส่งสัญญาณตรวจสอบว่าข้อมูลถึงเครื่องปลายทางอย่างถูกต้องครบถ้วนในการส่งข้อมูลแต่ละครั้ง จึงไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล

ส่วนประกอบของ UDP Frame

1. *Source Port* : เป็นค่าตัวเลข 16 บิต บอกรหัสของบริการที่เครื่องต้นทาง
2. *Destination Port* : เป็นค่าตัวเลข 16 บิต บอกรหัสของบริการที่เครื่องปลายทาง
3. *Length* : เป็นค่าตัวเลข 16 บิต บอกความยาวของข้อมูล
4. *Checksum* : เป็นค่าตัวเลข 16 บิต ตรวจสอบความถูกต้องของข้อมูลที่ส่ง

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Source Port	Destination Port
Length	Checksum
Data	

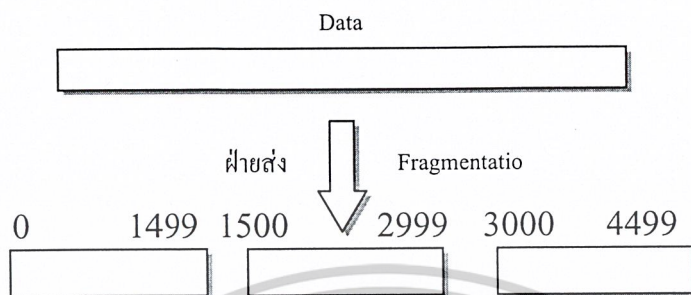
รูปที่ 3-5 แสดงแพ็กเก็ตยูดีพี

3.5 โพรโทคอลไอพี (IP: Internet Protocol)

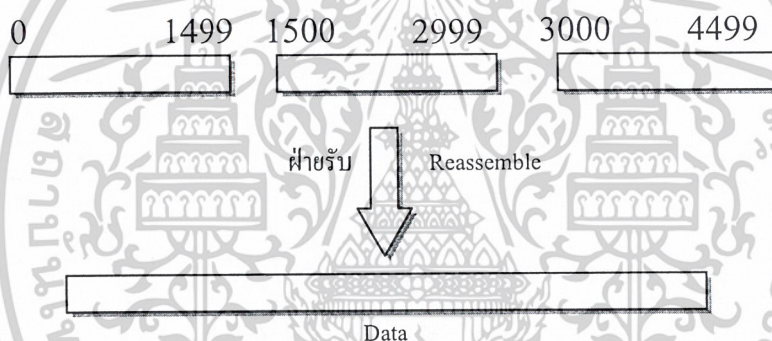
โพรโทคอลไอพีเป็นโพรโทคอลที่จัดการเกี่ยวกับแอดเดรสของแต่ละแพ็กเก็ต เพื่อให้ส่งแพ็กเก็ตต่างๆ ไปยังเป้าหมายได้ถูกต้อง การทำงานของไอพีเป็นเพียงการส่งข้อมูลไปยังเครื่องเป้าหมายเท่านั้น ไม่มีการส่งสัญญาณขอรับบริการ หรือสัญญาณให้บริการระหว่างกันเหมือนที่ซีพี เรียกว่าการเชื่อมต่อแบบ Connectionless ซึ่งระบบทั้งสองตั้งสมมติฐานว่าการเชื่อมต่อระหว่างกันไม่มีความผิดพลาดเกิดขึ้นแน่

เนื่องจากมาตรฐานในเครือข่ายมีหลากหลาย ขนาดของแพ็กเก็ตในแต่ละมาตรฐานจึงมีความแตกต่างกันออกไป ทำให้การส่งข้อมูลระหว่างอุปกรณ์ในเครือข่ายนั้นอาจมีการแบ่งข้อมูลออกเป็นแพ็กเก็ตย่อยๆ ในระหว่างการส่ง เรียกว่า การทำแฟร็กเมนต์ชัน (Fragmentation) เช่น แพ็กเก็ตของ FDDI มีขนาด 4,500 ไบต์ หากเครื่องปลายทางอยู่ในเครือข่าย Ethernet ซึ่งมีขนาดของแพ็กเก็ตสูงสุดเพียง 1,500 ไบต์

ดังนั้นการส่งแพ็กเก็ตไปยังเครื่องปลายทางจึงต้องมีการแบ่งเป็นแพ็กเก็ตย่อย และเมื่อแพ็กเก็ตย่อยมาถึงเครื่องเป้าหมายก็จะมารวมกันเป็นแพ็กเก็ตเดิมที่มีขนาด 4,500 ไบต์อีกครั้ง เรียกการรวมกันนี้ว่าการรีแอสเซมเบิล (Reassemble) ซึ่งทำให้ได้ข้อมูลเหมือนที่ส่งมาจากเครื่องต้นทาง



รูปที่ 3-6 แสดงการทำแฟร็กเมนต์ชัน



รูปที่ 3-7 แสดงการรีแอสเซมเบิล

ส่วนประกอบของแพ็กเก็ตไอพี

1. *version* : เป็นค่าตัวเลข 4 บิต บอกเวอร์ชันของมาตรฐานไอพีที่ใช้ โดยปกติมีค่าเป็น 4 ซึ่งหมายถึง IPv4
2. *Internet Header Length (IHL)* : เป็นตัวบอกความยาวเฮดเดอร์ของไอพี
3. *Type of Service* : เป็นส่วนที่บอกการทำงานของแพ็กเก็ตที่ส่งว่าทำหน้าที่อะไร มีทั้งหมด 8 บิต โดย

Bit 0-2 : บอกรายละเอียดการทำงานของแพ็กเก็ตนั้นๆ

111 - Network Control

110 - Internetwork Control

101 - CRITIC / ECP

100 - Flash Override

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

011 - Flash

010 - Immediate

001 - Priority

000 - Routine

Bit 3 : บอกถึงลักษณะของดีเลย์

0 = Normal Delay - มีดีเลย์ปกติ

1 = Low Delay - มีดีเลย์ต่ำ

Bit 4 : บอกถึงประเภทของทรูพุต

0 = Normal Throughput - มีทรูพุตปกติ

1 = High Throughput - มีทรูพุตสูง

Bit 5 : บอกถึงประเภทของความน่าเชื่อถือ

0 = Normal Reliability - มีความน่าเชื่อถือพอประมาณ

1 = High Reliability - มีความน่าเชื่อถือสูง

Bit 6-7 : กันไว้ใช้ในอนาคค

4. *Total Length* : มีขนาด 16 บิต บอกถึงความยาวในดาต้าแกรมของไอพี
5. *Identification field* : เป็นตัวเลข 16 บิต เป็นค่าประจำตัวของไอพีนั้น โดยโฮสต์ที่ส่งเป็นผู้กำหนด และเพิ่มค่าขึ้นหนึ่งเมื่อมีการส่งดาต้าแกรมของไอพีใหม่ ซึ่งใช้ในการประกอบกลับ
6. *Flag* : เป็นตัวเลข 3 bit บอกลักษณะของแฟ็กเก็ตว่ามีการเฟร็กเมนต์หรือไม่

Bit 0 : สงวนไว้ ปกติเป็น 0

Bit 1 : 0 = บอกว่าแฟ็กเก็ตมีการแตกแฟ็กเก็ตย่อย

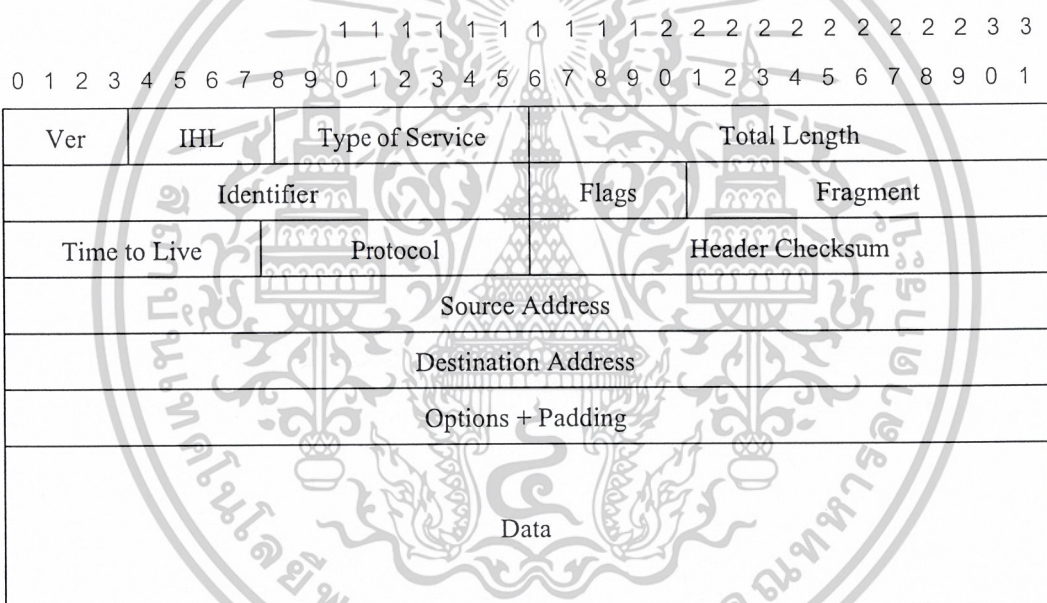
1 = บอกว่าแฟ็กเก็ตไม่มีการแตกแฟ็กเก็ตย่อย

Bit 2 : 0 = บอกว่าแฟ็กเก็ตนั้นเป็นแฟ็กเก็ตสุดท้ายที่ได้จากการแตกแฟ็กเก็ตย่อย

1 = บอกว่าแฟ็กเก็ตนั้นยังไม่ใช่แฟ็กเก็ตสุดท้ายที่ได้จากการแตกแฟ็กเก็ตย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. *Fragment Offset* : เป็นค่าตัวเลข 13 บิต บอกออฟเซตของแฟร็กเมนต์เมื่อเทียบในดาต้าแกรม
8. *Time To Live (TTL)* : เป็นตัวเลข 8 บิต บอกช่วงเวลาของแพ็กเก็ตที่ยังอยู่ในเครือข่ายได้ โดยกำหนดค่าเป็นจำนวนเร้าเตอร์สูงสุดที่ดาต้าแกรมผ่านได้ ซึ่งโดยทั่วไปที่ค่าระหว่าง 32 ถึง 64 และลดค่าลงเรื่อยๆ เมื่อผ่านเร้าเตอร์ เพื่อเป็นการป้องกันแพ็กเก็ตเคลื่อนเครือข่าย
9. *Protocol* : เป็นตัวเลข 8 bit บอกถึงโปรโตคอลที่อยู่เหนือขึ้นไป ว่าเป็นโปรโตคอลระดับสูงกว่าประเภทใด
10. *Header Checksum* : เป็นค่าตัวเลข 32 บิต ใช้ตรวจสอบความถูกต้องของเฮดเดอร์
11. *Source Address* : เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องต้นทาง
12. *Destination Address* : เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องปลายทาง



รูปที่ 3-8 แสดงแพ็กเก็ตไอพี

บทที่ 4

ไอพีแอดเดรส (IP Address)

ไอพีแอดเดรสเป็นส่วนประกอบอย่างหนึ่งในเครือข่ายที่ซีพี/ไอพี ไอพีแอดเดรสเป็นแอดเดรสทางซอฟต์แวร์ประจำเครือข่าย ผู้ออกแบบเครือข่ายจำเป็นต้องศึกษาและเข้าใจรูปแบบของไอพีแอดเดรสโดยละเอียด การจัดแบ่งแอดเดรสออกเป็นคลาส และการจัดแบ่งเครือข่ายย่อยหรือซับเน็ต รวมทั้งสามารถออกแบบซับเน็ตโดยเลือกใช้ไอพีแอดเดรสได้อย่างถูกต้อง เนื้อหาซึ่งจะกล่าวถึงในบทนี้มีดังนี้

- รูปแบบไอพีแอดเดรส
- การจัดคลาส
- การจัดแบ่งเครือข่ายย่อย
- ไพรเวตไอพีแอดเดรส

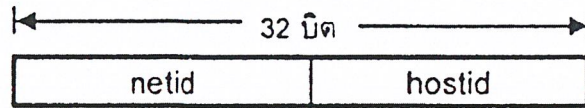
4.1 ไอพีแอดเดรส

อุปกรณ์ที่เชื่อมต่อเข้าเครือข่ายและสามารถทำงานตามข้อกำหนดของซีพี/ไอพีจะต้องมีแอดเดรสประจำอุปกรณ์นั้น อุปกรณ์ที่กล่าวถึงนี้อาจเป็น โฮสต์ เราเตอร์ เครื่องพิมพ์ หรือแม้กระทั่งอุปกรณ์สำนักงานยุคใหม่เช่นโทรศัพท์หรือเครื่องถ่ายเอกสาร ไอพีรุ่นที่กำหนดให้ใช้ไอพีแอดเดรสขนาด 32 บิต อุปกรณ์ที่เชื่อมกับอินเทอร์เน็ตจะมีไอพีแอดเดรส 32 บิตประจำอินเทอร์เน็ตเฟสที่ไม่ซ้ำกัน อุปกรณ์เช่นเราเตอร์ซึ่งมีหลายอินเทอร์เน็ตเฟสจะมีไอพีแอดเดรสหลายค่าตามจำนวนอินเทอร์เน็ตเฟสที่มีอยู่โดยไม่ซ้ำค่ากัน แต่คอมพิวเตอร์หรือโฮสต์ที่พบโดยทั่วไปมักมีเพียงอินเทอร์เน็ตเฟสเดียวจึงมักเรียกว่าไอพีแอดเดรสเป็นแอดเดรสประจำโฮสต์

แอดเดรสขนาด 32 บิตมีจำนวนแอดเดรสรวมเท่ากับ 2^{32} (4,294,967,296) แต่เมื่อนำมาจัดสรรแล้วจะไม่สามารถใช้งานได้ครบทั้งหมด ไอพีแอดเดรสนิยมเขียนในรูปแบบเลขฐานสิบตามแบบ “dotted decimal” โดยแบ่งตัวเลข 32 บิตออกเป็น 4 ไบต์ แต่ละไบต์แทนด้วยเลขฐานสิบหนึ่งตัวและคั่นแต่ละไบต์ด้วยเครื่องหมายจุด เช่นแอดเดรส 10100001 11110110 00000010 00000001 จะเขียนได้เป็น 161.246.2.1

แอดเดรสขนาด 32 บิต ประกอบขึ้นจากหมายเลขสองส่วนคือ เลขเครือข่าย (Network number หรือ network identifier หรือ netid) และ เลขโฮสต์ (host number หรือ host identifier หรือ hostid) เลขเครือข่ายใช้สำหรับจัดคลาสเครือข่าย ส่วนเลขโฮสต์ใช้ระบุหมายเลขโฮสต์ (หรืออีกนัยหนึ่งคืออินเทอร์เน็ตเฟสของโฮสต์) ในเครือข่าย ไอพีแอดเดรสจึงแบ่งได้เป็นสองส่วนตามรูปที่ 4-1 จำนวนบิตที่ใช้สำหรับเลขเครือข่ายและเลขโฮสต์จะขึ้นอยู่กับคลาสที่สังกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



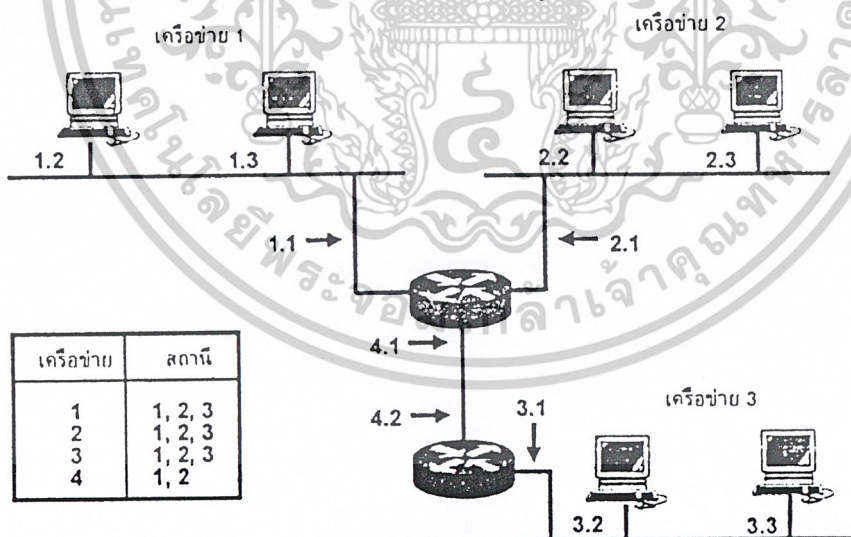
รูปที่ 4-1 รูปแบบของไอพีแอดเดรส

ในปัจจุบันฟิลด์กำหนดเลขเครือข่ายนิยมเรียกว่า พรีฟิกซ์เครือข่าย (network-prefix) เพราะทุกโฮสต์ในเครือข่ายจะต้องมีพรีฟิกซ์หรือบิตนำหน้าเหมือนกัน ตัวอย่างเช่นหากมีเลขเครือข่ายจำนวน 16 บิตก็จะเรียกว่า พรีฟิกซ์ 16 เป็นต้น

4.1.1 ความสำคัญของเลขเครือข่ายและเลขโฮสต์

การจัดแบ่งไอพีแอดเดรสออกเป็นสองส่วนที่ประกอบด้วยเลขเครือข่ายและเลขโฮสต์ ก็เพื่อประโยชน์ในการดูแลระบบ เราเตอร์จะอาศัยเลขเครือข่ายเพื่อแลกเส้นทางส่งแพ็กเก็ตเกิดด้วยหลักการโดยสังเขปต่อไปนี้

- โฮสต์ที่มีเลขเครือข่ายชุดเดียวกันย่อมอยู่ภายในเครือข่ายเดียวกัน และสามารถสื่อสารถึงกันโดยใช้เฟรมค่าลิงก์โดยไม่ต้องพึ่งพาเราเตอร์
- โฮสต์ที่มีเลขเครือข่ายต่างกันจะอยู่ต่างเครือข่ายกัน การสื่อสารระหว่างโฮสต์จะอาศัยเราเตอร์ที่เชื่อมเครือข่ายเป็นผู้นำส่งแพ็กเก็ต เราเตอร์อาจเชื่อมเครือข่ายที่อยู่ติดกันหรือส่งแพ็กเก็ตผ่านเราเตอร์อื่นไปยังปลายทางดังรูปที่ 4-2

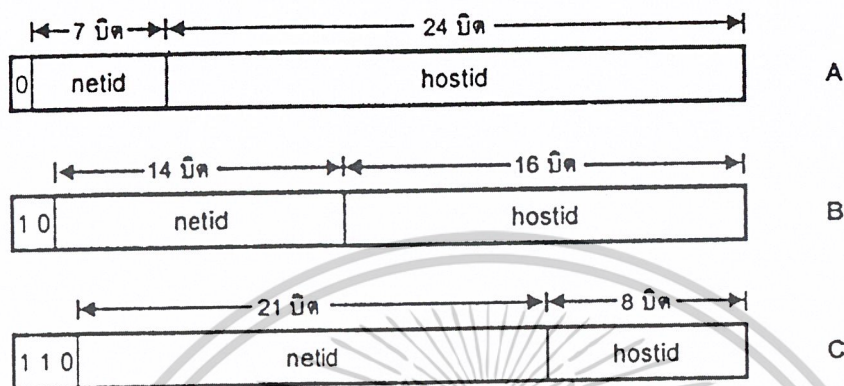


รูปที่ 4-2 เราเตอร์เชื่อมโยงเครือข่ายที่มีเลขเครือข่ายต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

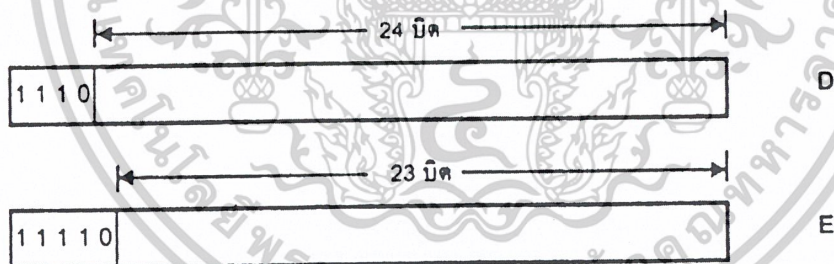
4.1.2 การจัดคลาสเครือข่าย

ไอพีแอดเดรสมีการจัดแบ่งออกเป็นกลุ่มหรือคลาส (class) เครือข่ายที่ใช้งานในปัจจุบันมักสังกัดอยู่ในคลาสใดคลาสหนึ่งคือคลาส A, B หรือ C การแบ่งคลาสอาศัยจำนวนพรีฟิกซ์เครือข่ายที่แตกต่างกันตามรูปที่ 4-3 แต่ละคลาสจึงมีจำนวนเครือข่ายในสังกัดและจำนวนโฮสต์ต่อเครือข่ายไม่เท่ากัน



รูปที่ 4-3 การแบ่งคลาสเครือข่าย

การจัดคลาสตามรูปที่ 4-3 เป็นการจัดแบ่งตามการใช้งานเครือข่ายทั่วไป ในขณะที่ยังมีอีกสองคลาสซึ่งใช้เพื่อจุดประสงค์เฉพาะได้แก่ คลาส D และ E ดังรูปที่ 4-4 เครือข่ายคลาส D เป็นเครือข่ายแบบมัลติคาสต์ ส่วนคลาส E สงวนไว้ใช้งานหากมีความจำเป็นอันใดในอนาคต ทั้งสองคลาสนี้ไม่ได้แบ่งเลขโฮสต์จึงไม่กำหนดโฮสต์ไว้



รูปที่ 4-4 การแบ่งคลาส D และ E

การจัดแบ่งคลาสโดยใช้พรีฟิกซ์เป็นการผนวกข้อมูลเพื่อใช้ในการเลือกเส้นทาง เช่น หากตรวจพบว่าพรีฟิกซ์ 2 บิตแรกมีค่าเป็น 10 แสดงว่าเป็นแอดเดรสในคลาส B ซึ่งมีค่า 16 บิตแรกกำหนดกลุ่มเครือข่ายและ 16 บิตถัดมาเป็นเลขโฮสต์

4.1.3 ลักษณะสำคัญของแต่ละคลาส

จำนวนเครือข่ายในแต่ละคลาสและจำนวนโฮสต์สูงสุดที่มีได้ สามารถคำนวณได้จากจำนวนบิตที่ใช้ทำงานตามสูตร 2^n เมื่อ n คือจำนวนบิต ตัวอย่างเช่นในคลาส B มีเลขโฮสต์จำนวน 16 บิต จึงมีโฮสต์ได้ไม่เกิน 2^{16} ซึ่งเท่ากับ 65,536 แต่เลขโฮสต์ที่ทุกบิตเป็น “0” และเป็น “1” จะสงวนไว้ใช้งานเฉพาะกรณีจำนวนโฮสต์จึงลดลงไป 2 โฮสต์ทุกเครือข่าย หรือมีโฮสต์ได้ไม่เกิน

$2^{16} - 2 = 65,534$ สูตร $2^n - 2$ นี้จะใช้ในการคำนวณเครือข่ายในคลาสและจำนวนโฮสต์ทั้งคลาส A, B และ C ดังนี้

คลาส A

เครือข่ายในคลาส A มีบิตซ้ายสุดเป็น 0 และใช้ 7 บิตถัดมากำหนดเลขเครือข่ายส่วนอีก 24 เป็นเลขโฮสต์ คลาส A จึงมีเลขเครือข่ายได้ 2^7 หรือ 128 ค่า แต่เครือข่าย 0.0.0.0 และ 127.0.0.0 สงวนไว้เป็นแอแดเรสเฉพาะงานคือ แอแดเรส 0.0.0.0 ใช้เป็นแอแดเรสกำหนดเส้นทางโดยปริยาย (default router) ส่วน 127.0.0.0 เป็นแอแดเรสลูบแเบ็ก (ดูหัวข้อ 4.3) จำนวนเครือข่ายที่สังกัดในคลาส A จึงมีได้ 126 เครือข่ายคือเลขที่ขึ้นต้นด้วย 1.0.0.0 ถึง 126.0.0.0

แต่ละเครือข่ายในคลาส A มีแอแดเรสได้ $2^{24} - 2$ หรือเท่ากับ 16,777,214 คือตั้งแต่ 0.0.1 ถึง 255.255.255.254 เครือข่ายในคลาส A ใช้กับหน่วยงานขนาดใหญ่ที่ต้องการแอแดเรสเป็นจำนวนมาก เครือข่ายคลาสนี้จัดสรรให้กับหน่วยงานในยุคแรกเริ่มของอินเทอร์เน็ต แอแดเรสเครือข่ายที่เหลืออยู่ส่วนใหญ่จะสงวนไว้ ตัวอย่างของเครือข่ายในคลาสนี้มีเช่น

9.0.0.0 ibm.com

15.0.0.0 hp.com

20.0.0.0 csc.net

ขอให้สังเกตว่าในคลาส A นี้เมื่อกล่าวถึงเฉพาะเลขเครือข่ายก็จะเขียนเฉพาะค่าที่แสดงเลขเครือข่ายที่ขนาด 8 บิต เท่านั้นเช่น 2 หรือ 26 ในทำนองเดียวกันเมื่อกล่าวถึงเฉพาะเลขโฮสต์ก็จะเขียนเฉพาะ 24 บิต เช่น 0.0.1 แต่ถ้าต้องการอ้างอิงโดยรวมทั้งเครือข่ายจะเขียนเฉพาะหมายเลขเครือข่ายโดยให้เลขโฮสต์เป็น “0” เช่น 2.0.0.0 รูปแบบการเขียนเช่นนี้ใช้กับคลาส B และ C เช่นกัน

คลาส B

เครือข่ายในคลาส B มีบิตแรกเริ่มเป็น 10 และใช้ 14 บิตถัดมากำหนดเลขเครือข่ายจำนวนบิตที่กำหนดเลขโฮสต์มีขนาด 16 บิต คลาส B จึงมีสมาชิกเครือข่ายได้ $2^{14} - 2$ หรือ 16,382 คือตั้งแต่ 128.1.0.0 ถึง 191.254.0.0 แต่ละเครือข่ายมีเลขโฮสต์ได้ $2^{16} - 2$ หรือเท่ากับ 65,534 แอแดเรส หรือตั้งแต่ 0.1 ถึง 255.254

เครือข่ายในคลาส B มักจัดสรรให้กับหน่วยงานขนาดกลาง ในปัจจุบันมีเครือข่ายในคลาส B เหลือไม่มากนัก และมักไม่จัดสรรเครือข่ายในคลาสนี้ให้กับผู้จดทะเบียนรายใหม่หากไม่มีความจำเป็นอย่างแท้จริง ตัวอย่างของเครือข่ายนี้เช่น

129.123.0.0	usu.edu
130.149.0.0	tu-berlin.de
161.246.0.0	kmitl.ac.th

คลาส C

เครือข่ายในคลาส C มีพรีฟิกซ์ 110 และใช้ 21 บิตถัดมาเป็นเลขเครือข่าย จำนวนบิตที่เป็นเลขโฮสต์มีเพียง 8 บิต คลาส C จึงมีเลขเครือข่ายได้ตั้งแต่ 192.0.1.0 ถึง 223.255.254.0 รวม 2,097,150 เครือข่าย แต่ละเครือข่ายมีเลขโฮสต์ได้ตั้งแต่ 1 ถึง 254

จำนวนแอดเดรสได้จำกัดเพียง 254 แอดเดรสทำให้เครือข่ายจึงเหมาะสมสำหรับหน่วยงานขนาดเล็ก หากจำเป็นต้องใช้โฮสต์มากกว่านี้ต้องขอใช้เครือข่ายคลาส C หลายเครือข่ายตัวอย่างของเครือข่ายคลาส C ได้แก่

198.6.250.0	isoc.org
198.137.240.0	whitehouse.gov
203.144.152.0	mcot.or.th

คลาส D และ E

เครือข่ายในคลาส D และ E ไม่มีการจัดแบ่งเลขเครือข่ายและเลขโฮสต์ คลาส D โดยมี 3 บิตแรกเป็น 111 จึงมีแอดเดรสตั้งแต่ 224.0.0.0 ถึง 239.255.255.255 แอดเดรสในคลาสนี้เรียกว่า มัลติคาสต์แอดเดรส (multicast address) เนื่องจากใช้ในเครือข่ายมัลติคาสต์

สำหรับคลาส E มีแอดเดรสจาก 240.0.0.0 ถึง 254.255.255.255 ซึ่งสำรองไว้เพื่อความจำเป็นเฉพาะงานในอนาคต

4.2 การแบ่งเครือข่ายย่อย

เครือข่ายที่สังกัดในคลาส A และ B เป็นเครือข่ายที่มีจำนวนโฮสต์ได้เป็นจำนวนมากกล่าวคือ 16,777,214 และ 65,534 ตามลำดับ ในทางปฏิบัติแล้วเราไม่สามารถต่อเชื่อมโฮสต์ทั้งหมดในเครือข่ายเดียวได้ เพราะข้อจำกัดทางฮาร์ดแวร์ ผู้วางระบบจึงต้องจัดแบ่งเครือข่ายขนาดใหญ่ให้เล็กลงเป็นเครือข่ายย่อย หรือ ซับเน็ต (subnet) การแบ่งซับเน็ต นอกจากจะจัดจำนวนโฮสต์ให้เหมาะสมกับฮาร์ดแวร์ของเครือข่ายแล้วยังช่วยอำนวยความสะดวกในการบริหารเครือข่าย

การจัดซับเน็ต [RFC 950] ใช้วิธีแบ่งบางส่วนของเลขโฮสต์มาใช้เป็น เลขซับเน็ต (subnetid) เพื่อกำหนดว่าเป็นเครือข่ายย่อยที่เท่าใด ตัวอย่างเช่นเครือข่าย 161.246.0.0 ซึ่งอยู่ในคลาส B อาจใช้ 8 บิตแรกของเลขโฮสต์เป็นเลขซับเน็ต และ 8 บิตที่เหลือใช้สำหรับเลขโฮสต์ดังรูปที่ 4-5

16 บิต	8 บิต	8 บิต
161.246	subnetid	hostid

รูปที่ 4-5 ตัวอย่างการแบ่งเครือข่ายย่อยของ 161.246

จำนวนบิตของเลขซับเน็ตเป็นตัวกำหนดจำนวนเครือข่ายย่อย ซับเน็ตขนาด 8 บิต สำหรับเครือข่าย 161.246.0.0 จะมี 254 ซับเน็ต ($2^{\text{subnet}} - 2$) แต่ละซับเน็ตมี 254 โฮสต์ ($2^{\text{hostid}} - 2$) ดังตารางที่ 4-1 เลขซับเน็ตที่ทุกบิตเป็น “1” และ “0” จะสงวนไว้ใช้งานเฉพาะ (ดูหัวข้อ4.3) ดังนั้นซับเน็ต 161.246.0.0 และ 161.246.255.0 จึงนำมาใช้ไม่ได้

ตารางที่ 4-1 การจัดแบ่งเครือข่าย 161.246 ด้วยซับเน็ต 8 บิต

ซับเน็ตที่	เครือข่ายย่อย	แอดเดรสเริ่มต้น	แอดเดรสสุดท้าย
1	161.246.1.0	161.246.1.1	161.246.1.254
2	161.246.2.0	161.246.2.1	161.246.2.254
3	161.246.3.0	161.246.3.1	161.246.3.254
..	..	..	..
..	..	..	..
252	161.246.252.0	161.246.252.1	161.246.252.254
253	161.246.253.0	161.246.253.1	161.246.253.254
254	161.246.254.0	161.246.254.1	161.246.254.254

4.2.1 ซับเน็ตมาสก์

เมื่อผู้วางระบบเลือกขนาดซับเน็ตแล้วจะกำหนดพารามิเตอร์เพื่อใช้บอกให้โฮสต์และเราเตอร์ทราบว่าซับเน็ตที่ใช้งานมีขนาดกี่บิต คำนี้เรียกว่า ซับเน็ตมาสก์ (subnet mask)

ซับเน็ตมาสก์เป็นตัวเลข 32 บิต ซึ่งเขียนอยู่ในรูป dotted-decimal เช่นเดียวกับการเขียนไอพีแอดเดรส ซับเน็ตมาสก์จะมีบิตที่ตรงกับเลขเครือข่ายและเลขซับเน็ตเท่ากับ “1” ส่วนบิตที่ตรงกับเลขโฮสต์มีค่าเท่ากับ “0” การเลือกซับเน็ตมาสก์ควรใช้ค่าที่บิต “1” อยู่ติดกับจากซ้ายมือไปทางขวามือเสมอ

ตัวอย่างเช่นเครือข่าย 161.246.0.0 ซึ่งแบ่งให้มีเลขซับเน็ตและเลขโฮสต์อย่างละ 8 บิตจะมีค่าซับเน็ตมาสก์เท่ากับ 255.255.255.0 คำนี้คำนวณได้จากการเขียนไอพีแอดเดรสทั้งสี่หลัก และใส่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขฐานสองค่า “1” ให้ครบทุกบิตที่เป็นเลขเครือข่ายและเลขซับเน็ต จากนั้นให้ใส่ค่า “0” สำหรับเลขโฮสต์ แล้วจึงแปลงเลขฐานสองที่ได้เป็นเลขฐานสิบตามขั้นตอนต่อไปนี้

	8 บิต	8 บิต	8 บิต	8 บิต
(1) นำค่าไอพีแอดเดรส	161	246	subnetid	hostid
(2) กำหนดบิต “1” และ “0”	11111111	11111111	11111111	00000000
(3) แปลงเป็นฐานสิบ	255	255	255	0

เครือข่าย 161.246.0.0 ซึ่งใช้ซับเน็ตมาสก์เท่ากับ 255.255.255.0 เรียกว่ามีซับเน็ตมาสก์ 24 บิต เนื่องจากมีบิตที่มีค่า “1” จำนวน 24 บิต หรือเขียนตามรูปแบบที่นิยมใช้ในปัจจุบันคือ 161.246.0.0/24 โดยเรียกว่าเครือข่าย 161.246.0.0 มี 프리ฟิกซ์ (prefix) 24 บิต

โปรดสังเกตว่า 161.246.0.0/24 ใช้เลขซับเน็ตจำนวน 8 บิต ดังนั้นนอกจากจะเรียกว่ามี 프리ฟิกซ์ 24 บิตแล้ว ยังเรียกได้อีกว่าใช้ซับเน็ตจำนวน 8 บิต

4.2.2 ดีฟอลต์ซับเน็ตมาสก์

การติดตั้งโฮสต์เข้าเครือข่ายนอกจากจะต้องกำหนดไอพีแอดเดรสแล้วต้องกำหนดค่าซับเน็ตมาสก์ตามที่ผู้ดูแลระบบกำหนดไว้ด้วย ถึงแม้ว่าในบางเครือข่ายเช่นเครือข่ายในคลาส C ซึ่งมีโฮสต์ได้ 254 โฮสต์และไม่ได้แบ่งให้มีซับเน็ต ขั้นตอนการติดตั้งโฮสต์ยังจำเป็นต้องใส่ค่าซับเน็ตมาสก์เช่นกัน แต่ค่าซับเน็ตมาสก์นี้เรียกว่า ดีฟอลต์ซับเน็ตมาสก์ (default subnet mask) ดีฟอลต์ซับเน็ตมาสก์ของเครือข่ายคลาส A, B และ คลาส C แสดงได้ดังตารางที่ 4-2

ผู้ดูแลระบบสามารถวางแผนจัดการเครือข่ายโดยเลือกทำซับเน็ตหรือไม่ทำซับเน็ตตามความต้องการ โดยปกติแล้วผู้ดูแลระบบเครือข่ายในคลาส A และ B ไม่สามารถหลีกเลี่ยงการใช้ซับเน็ตได้

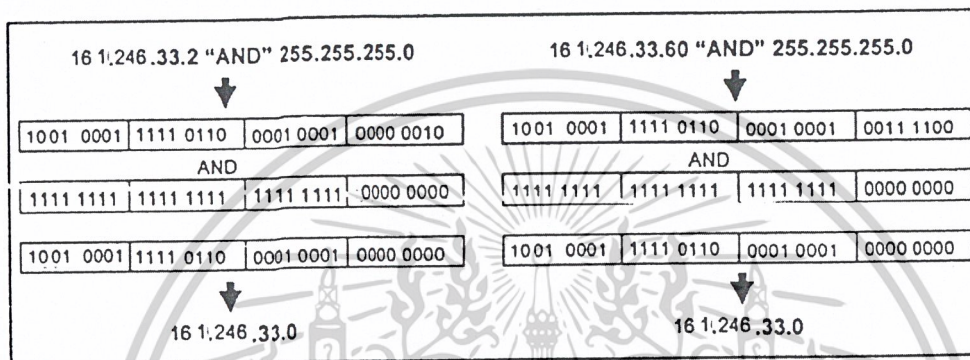
ตาราง 4-2 ค่าดีฟอลต์ซับเน็ตมาสก์

คลาส	ดีฟอลต์ซับเน็ตมาสก์	ดีฟอลต์ซับเน็ตมาสก์ (ฐานสอง)
A	255.0.0.0	11111111.00000000.00000000.00000000
B	255.255.0.0	11111111.11111111.00000000.00000000
C	255.255.255.0	11111111.11111111.11111111.00000000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การเลือกเส้นทางในชั้นเน็ต

ชั้นเน็ตนอกจากจะช่วยจัดการแบ่งเครือข่ายย่อยแล้ว ยังใช้ประโยชน์ในการเลือกเส้นทางส่งไอพีเดตาแกรมระหว่างเครือข่ายย่อยด้วย เช่น โฮสต์ 161.246.33.2 ในเครือข่าย 161.246.0.0/24 (ชั้นเน็ตมาสก์ 255.255.255.0) ต้องการส่งข้อมูลไปยังโฮสต์ 161.246.33.60 โปรโตคอลไอพีจะทำหน้าที่เลือกเส้นทางโดยนำแอดเดรส 161.246.33.2 และ 161.246.33.60 มาผ่านลอจิก “AND” บิตต่อบิตกับค่าชั้นเน็ตมาสก์ดังรูปที่ 4-6



รูปที่ 4-6 การตรวจสอบหาแอดเดรสชั้นเน็ตเพื่อเลือกเส้นทาง

ผลลัพธ์จากลอจิก “AND” ของแอดเดรสและมาสก์ข้างต้นได้ค่าแอดเดรสชั้นเน็ต 161.246.33.0 เท่ากัน ซึ่งหมายความว่าโฮสต์ทั้งสองอยู่ในชั้นเน็ตเดียวกัน หากเครือข่ายที่ใช้คืออีเทอร์เน็ตแล้ว โฮสต์ 161.246.33.2 จะสร้างแพ็กเก็ตโดยระบุอีเทอร์เน็ตแอดเดรสของ 161.246.33.60 โดยไม่ต้องส่งแพ็กเก็ตให้เราเตอร์ดำเนินการ โปรดสังเกตว่าการใช้ลอจิก “AND” เป็นการให้ชั้นเน็ตมาสก์เพื่อ “มาสก์” ให้ได้เฉพาะเลขเครือข่าย ค่าชั้นเน็ตมาสก์จึงเป็นเสมือน หน้ากากครอบเอาเลขเครือข่ายออกมา

ในกรณีที่โฮสต์ปลายทางอยู่ต่างเครือข่ายกับโฮสต์ต้นทาง เช่นแอดเดรสของโฮสต์ต้นทางคือ 161.246.33.2 และโฮสต์ปลายทางคือ 161.246.40.5 ผลจากลอจิก “AND” 161.246.33.0 ดังนั้นโฮสต์ 161.246.33.2 จะสรุปว่า 161.246.40.0 อยู่ต่างชั้นเน็ต และจะส่งแพ็กเก็ตไปยังเราเตอร์เพื่อให้เราเตอร์นำส่งแพ็กเก็ตต่อไป

4.2.4 การเลือกขนาดชั้นเน็ต

จำนวนบิตที่ใช้ในการแบ่งชั้นเน็ตขึ้นอยู่กับการใช้ของผู้วางระบบเอง ค่าชั้นเน็ตมาสก์เป็นตัวกำหนดจำนวนชั้นเน็ตและจำนวนโฮสต์ที่มีได้ในเครือข่าย ผู้วางระบบเครือข่ายต้องเลือกจำนวนบิตที่เหมาะสมกับการใช้งานในปัจจุบันและการเพิ่มขยายในอนาคต

จำนวนชั้นเน็ตที่เลือกใช้จะลงตัวที่ค่า $2^n - 2$ เช่นเมื่อต้องการใช้ 5 ชั้นเน็ต อาจเลือกใช้ค่า $2^5 - 2$ หรือ 6 ชั้นเน็ตซึ่งจะเหลืออีกหนึ่งชั้นเน็ตสำรอง หรือเมื่อต้องการใช้ 10 ชั้นเน็ตจะต้องเลือกใช้ชั้นเน็ต $2^4 - 2$ หรือ 14 ชั้นเน็ตโดยมีสำรองไว้ 4 ชั้นเน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4-3 แสดงจำนวนชั้นเน็ตคลาส B ที่ค่าที่ใช้ได้ตั้งแต่ 2 ชั้นเน็ต (16,382 แอดเดรสต่อชั้นเน็ต) ถึง 16,382 ชั้นเน็ต (2 แอดเดรสต่อชั้นเน็ต) ดังตารางที่ 4-3 ชั้นเน็ตที่นิยมใช้ได้แก่ชั้นเน็ต /22, /23, /24, /25 และ /26

ในคลาส C ที่มีโฮสต์เพียง 254 โฮสต์ก็สามารถเลือกทำชั้นเน็ตได้ เพียงแต่มีจำนวนบิตให้ดำเนินการได้จำกัด ตัวอย่างเช่นแบ่งเครือข่ายเป็น 2 ชั้นเน็ต ต้องใช้เลขชั้นเน็ต 2 บิตส่วน 6 บิตที่เหลือใช้เป็นเลขโฮสต์ทำให้มีโฮสต์ได้ 62 โฮสต์ต่อชั้นเน็ตดังตารางที่ 4-4

ข้อพึงระลึกสำคัญ คือ การแบ่งให้มีชั้นเน็ตมากต้องใช้อินเทอร์เน็ตเฟสมากขึ้น ตัวอย่างเช่นแอดเดรส 194.1.1.0 ในเครือข่ายคลาส C ที่ไม่มีชั้นเน็ตหรือมีเครือข่ายเดียวสามารถเลือกเครือข่ายที่เชื่อมโยงกันโดยฮับหรือสวิตช์ได้ดังรูปที่ 4-7 (ก) หากมี 2 ชั้นเน็ตจำเป็นต้องใช้อินเทอร์เน็ตของเราเตอร์อย่างน้อย 2 อินเทอร์เน็ตดังรูปที่ 4-7 (ข) แต่ถ้านแบ่งออกเป็น 6 ชั้นเน็ตก็ต้องมีอย่างน้อย 6 อินเทอร์เน็ตดังรูปที่ 4-7 (ค)

ตารางที่ 4-3 ชั้นเน็ตมาตรฐานในเครือข่ายคลาส B

ชั้นเน็ตบิต	พรีฟิกซ์	จำนวนชั้นเน็ต	โฮสต์ต่อชั้นเน็ต	ชั้นเน็ตมาสก์
0	/16	0	65534	255.255.0.0
1	/17	ไม่อนุญาตให้ใช้		
2	/18	2	16382	255.255.192.0
3	/19	6	8190	255.255.224.0
4	/20	14	4094	255.255.240.0
5	/21	30	2046	255.255.248.0
6	/22	62	1022	255.255.252.0
7	/23	126	510	255.255.254.0
8	/24	254	254	255.255.255.0
9	/25	510	126	255.255.255.128
10	/26	1022	62	255.255.255.192
11	/27	2046	30	255.255.255.224
12	/28	4096	14	255.255.255.240
13	/29	8190	6	255.255.255.248
14	/30	16382	2	255.255.255.252
15	/31	ไม่อนุญาตให้ใช้		

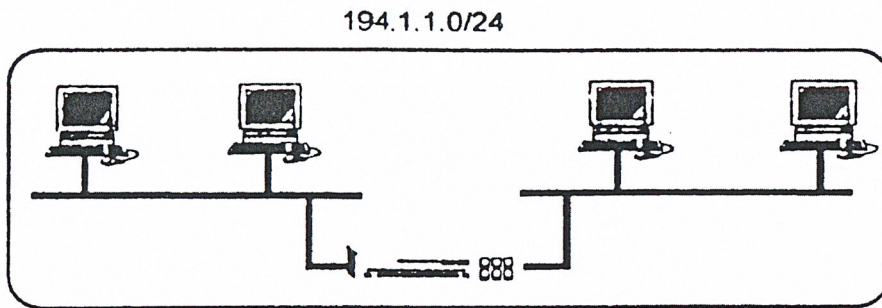
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4-4 ชั้นเน็ตมาสดในเครือข่ายคลาส C

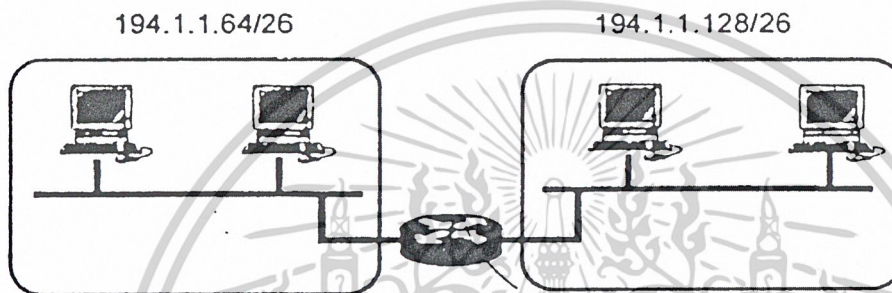
ชั้นเน็ตบิต	พรีฟิกซ์	จำนวนชั้นเน็ต	โฮสต์ต่อชั้นเน็ต	ชั้นเน็ตมาสด
0	/24	0	254	255.255.255.0
1	/25	ไม่อนุญาตให้ใช้		
2	/26	2	62	255.255.255.192
3	/27	6	30	255.255.255.224
4	/28	14	14	255.255.255.240
5	/29	30	6	255.255.255.248
6	/30	62	2	255.255.255.252
7	/31	ไม่อนุญาตให้ใช้		



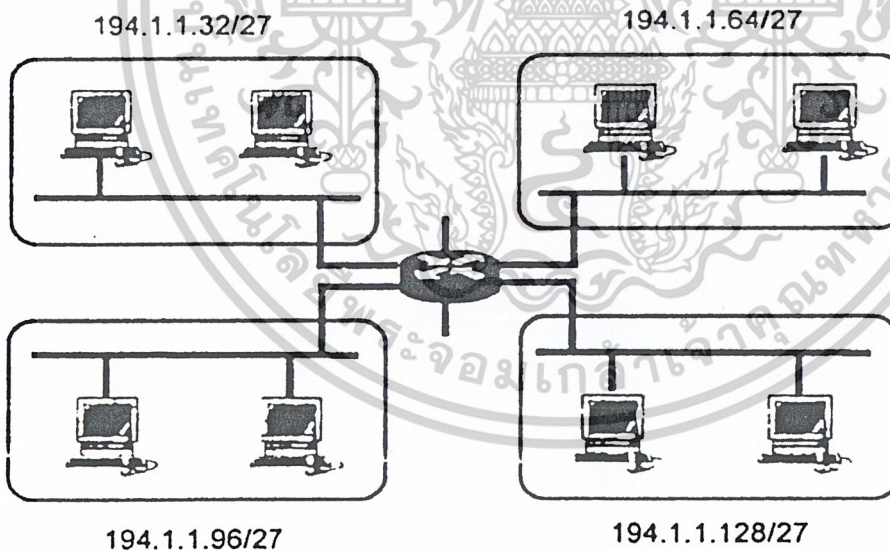
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) เชื่อมเครือข่ายด้วยฮับโดยไม่แบ่งซับเน็ต



(ข) ใช้เราเตอร์แบ่งเป็น 2 ซับเน็ต



(ค) ใช้เราเตอร์แบ่งเป็น 6 ซับเน็ต (แสดงแอดเดรสเพียง 4 ซับเน็ต)

รูปที่ 4-7 ตัวอย่างการจัดแบ่งซับเน็ต 194.1.1.0 แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ไอพีแอดเดรสเฉพาะกรณี

การใช้ไอพีแอดเดรสกำหนดเลขเครือข่ายและโฮสต์มีข้อกำหนดปลีกย่อยที่ผู้วางระบบเครือข่ายต้องเข้าใจเพื่อเลือกใช้ได้อย่างถูกต้อง ข้อกำหนดที่กล่าวถึงนี้คือแอดเดรสที่ไม่อนุญาตให้ใช้งานและแอดเดรสที่สงวนไว้ใช้งานในความหมายเฉพาะอย่างตามรูปที่ 4-8 โดยมีรายละเอียดดังนี้

netid	0000 0000	แอดเดรสเครือข่าย
1111 1111 1111 1111 1111 1111 1111 1111		บรอดคาสต์ในเครือข่าย
netid	1111 1111	บรอดคาสต์ในเครือข่าย netid
0000 0000	0000 0000 0000 0000	โฮสต์
127	ค่าใดก็ได้	ลูปแบ็ค

รูปที่ 4-8 ไอพีแอดเดรสเฉพาะกรณี

แอดเดรสกำหนดเครือข่าย

แอดเดรสที่มีเลขโฮสต์เป็น “0” ทั้งหมด ใช้กำหนดตัวเครือข่ายจึงไม่สามารถนำมาใช้เป็นไอพีแอดเดรสสำหรับอินเทอร์เน็ตได้ เช่น 5.0.0.0 เป็นเครือข่ายหนึ่งในคลาส A 161.246.0.0 เป็นเครือข่ายในคลาส B และ 202.105.100.0 เป็นเครือข่ายในคลาส C

ในเครือข่ายที่แบ่งซับเน็ตก็จะมีแอดเดรสเครือข่ายที่ไม่อนุญาตให้นำมาใช้งาน เช่น เครือข่าย 161.246.0.0 ซึ่งใช้ซับเน็ตมาสก์ 255.255.255.0 จะไม่สามารถใช้แอดเดรส 161.246.1.0, 161.246.2.0, ..., 161.246.254.0

แอดเดรสบรอดคาสต์

แอดเดรสที่มีเลขโฮสต์ทุกบิตเป็น “1” ใช้เป็นแอดเดรสบรอดคาสต์โดยตรงภายในเครือข่าย (net directed broadcast) เช่น เครือข่าย 161.246.0.0/16 มีแอดเดรสบรอดคาสต์เท่ากับ 161.246.255.255/24 และเครือข่าย 202.105.100.0/24 มีแอดเดรสบรอดคาสต์เท่ากับ 202.105.100.255 เป็นต้น

เมื่อมีการแบ่งซับเน็ตจะเรียกแอดเดรสที่มีเลขโฮสต์เป็น “1” ทั้งหมดว่า แอดเดรสบรอดคาสต์โดยตรงในซับเน็ต (subnet directed broadcast) เช่น 161.246.0.0/24 มีแอดเดรส 161.246.1.255 เป็นแอดเดรสบรอดคาสต์โดยตรงของซับเน็ต 161.246.1.0

ขอให้พิจารณาค่า 161.246.255.255 ซึ่งมีความหมายได้สองกรณีคือ กรณีแรกหากไม่ใช่ซับเน็ต (161.246.0.0/16) ถือเป็นการบรอดคาสต์ทั้งเครือข่าย 161.246.0.0 หรือเรียกว่า “net directed broadcast” กรณีที่สองหากใช้พรีฟิกซ์ 24 บิต (161.246.0.0/24) ดังนั้นแอดเดรส 161.246.255.255 จะหมายถึงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรอดคาสต์ตรงทุกซบเน็ต (all-subnet-directed broadcast) ถึงแม้ว่า RFC 922 จะกำหนดให้แอดเดรสนี้เป็นการบรอดคาสต์ไปยังทุกซบเน็ต แต่ปัจจุบันเราเตอร์จะไม่กระจายบรอดคาสต์ไปยังซบเน็ตอื่น

แอดเดรสบรอดคาสต์ 255.255.255.255 เป็นแอดเดรสบรอดคาสต์สำหรับทุกโฮสต์ในซบเน็ตเช่นกัน และเรียกว่า “limited broadcast” เพราะเราเตอร์จะกั้นการบรอดคาสต์ให้อยู่เฉพาะในเครือข่ายที่ปล่อยแอดเดรสนี้เท่านั้น โฮสต์จะใช้แอดเดรสนี้กรณีที่ต้องการถามหาเซิร์ฟเวอร์เพื่อขอบริการ

แอดเดรสซบเน็ตที่สงวนไว้

เมื่อมีการแบ่งซบเน็ตแล้ว เลขซบเน็ตที่มีค่า 0 ทุกบิต (เรียกเลขซบเน็ตนี้ว่า All-0s subnet) และ 1 ทุกบิต (เรียกเลขซบเน็ตนี้ว่า All-1s subnet) จะสงวนไว้ไม่ให้นำมาใช้กำหนดแอดเดรส [RFC 950, RFC 1122]

เราเตอร์หลายยี่ห้อในปัจจุบันอนุญาตให้ใช้แอดเดรสซบเน็ตที่มีเป็น 0 ทั้งหมด และ 1 ทั้งหมดได้ เนื่องจากการจัดแอดเดรสในปัจจุบันมีรูปแบบที่ไม่จัดคลาส (classless) ตามหลักการของ “ไซเคอร์” (ดูหัวข้อ 4.5.2) วิธีนี้ช่วยให้มีซบเน็ตใช้งานเพิ่มขึ้นจากเดิมที่เคยห้ามใช้

แอดเดรสกำหนดโฮสต์

แอดเดรส 0.0.0.0 หมายถึงตัวโฮสต์เอง (this host on this net) แอดเดรสนี้สงวนไว้ใช้สำหรับโฮสต์ในการเริ่มต้นระบบด้วยโปรโตคอลบูต และไมอนุญาตให้ใช้เป็นไอพีแอดเดรสปลายทาง นอกจากนี้แอดเดรส 0.0.0.0 ยังมีความหมายเฉพาะในการเลือกเส้นทาง โฮสต์หรือเราเตอร์ใช้แอดเดรสนี้เป็น ดีฟอลต์เร้าท์ (default route) หรือเส้นทางโดยปริยาย

แอดเดรสลูปแบ็ก

แอดเดรส 127.x.x.x ใช้เป็นแอดเดรสเพื่อเชื่อมเข้าสู่อินเทอร์เน็ตลูปแบ็ก ค่า x มีค่าเท่าใดก็ได้ แต่โดยปกติแล้วจะนิยมใช้แอดเดรส 127.0.0.1

4.3.1 ตัวอย่างการทำซบเน็ต

เพื่อให้เห็นภาพการออกแบบซบเน็ตและการหาค่าแอดเดรสที่ใช้งานได้จึงยกตัวอย่างเครือข่าย 194.1.1.0 ซึ่งต้องการแบ่งให้แต่ละซบเน็ตมีโฮสต์อย่างน้อย 25 โฮสต์

ขั้นตอนที่ 1 กำหนดค่าซบเน็ตมาสก์

ขั้นตอนแรกคือการหาค่าซบเน็ตมาสก์ โดยก่อนอื่นต้องพิจารณาว่า 194.1.1.0 อยู่ในคลาส C ซึ่งมีเลขโฮสต์ 8 บิตสำหรับทำซบเน็ต ข้อกำหนดระบุว่าต้องมีจำนวนโฮสต์อย่างน้อย 25 โฮสต์ต่อซบเน็ต ดังนั้นจำนวนบิตที่ใช้เป็นเลขโฮสต์คือ 5 บิต เพราะเลขโฮสต์ 5 บิตทำให้มีโฮสต์ในซบเน็ตได้ $2^5 - 2 = 30$ ผลลัพธ์ที่ตามมาคือจำนวนบิตที่ใช้เป็นเลขซบเน็ตจะเหลือเท่ากับ $8 - 5 = 3$ บิต ซึ่งสร้างซบเน็ตได้ $2^3 - 2 = 6$ ซบเน็ต ซบเน็ตมาสก์สำหรับเครือข่ายนี้จึงเท่ากับ 255.255.255.224 ตามการคำนวณต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{array}{rcl}
 194.1.1.0 & = & \left| 1100\ 0010 \cdot 0000\ 0001 \cdot 0000\ 0001 \cdot \left| 000 \right| 00000 \right| \\
 255.255.255.224 & = & \left| 1111\ 1111 \cdot 1111\ 1111 \cdot 1111\ 1111 \cdot \left| 111 \right| 00000 \right| \\
 & & \xleftarrow{\text{netid}} \quad \quad \quad \xrightarrow{\text{subnetid}} \quad \quad \quad \uparrow \quad \quad \uparrow \\
 & & \quad \quad \quad \text{subnetid} \quad \quad \text{hostid}
 \end{array}$$

ขั้นตอนที่ 2 หาเลขเครือข่ายประจำแต่ละซับเน็ต

ซับเน็ตทั้งหมดรวม 8 ซับเน็ต (ซึ่งใช้ได้เพียง 6 ซับเน็ต) จะมีเลขเครือข่ายประจำโดยปรับเปลี่ยนเลขซับเน็ต 3 บิต จาก 000 ถึง 111 ดังต่อไปนี้

ซับเน็ต #0 : 1100 0010 . 0000 0001 . 0000 0001 . 000 00000 = 194.1.1.0 (ไม่อนุญาตให้ใช้)
 ซับเน็ต #1 : 1100 0010 . 0000 0001 . 0000 0001 . 001 00000 = 194.1.1.32
 ซับเน็ต #2 : 1100 0010 . 0000 0001 . 0000 0001 . 010 00000 = 194.1.1.64
 ซับเน็ต #3 : 1100 0010 . 0000 0001 . 0000 0001 . 011 00000 = 194.1.1.96
 ซับเน็ต #4 : 1100 0010 . 0000 0001 . 0000 0001 . 100 00000 = 194.1.1.128
 ซับเน็ต #5 : 1100 0010 . 0000 0001 . 0000 0001 . 101 00000 = 194.1.1.160
 ซับเน็ต #6 : 1100 0010 . 0000 0001 . 0000 0001 . 110 00000 = 194.1.1.192
 ซับเน็ต #7 : 1100 0010 . 0000 0001 . 0000 0001 . 111 00000 = 194.1.1.224 (ไม่อนุญาตให้ใช้)

จากข้อกำหนดของ RFC 950 ซับเน็ต #0 และ #7 นำมาใช้ไม่ได้เนื่องจากมีเลขซับเน็ตเป็น 0 และ 1 ทุกบิต เราสามารถตรวจสอบความถูกต้องของค่าที่ได้ี้โดยพิจารณาจากช่วงห่างแต่ละซับเน็ตซึ่งเท่ากับ 32 ตามจำนวนแอดเดรสที่มีได้ในแต่ละซับเน็ต

ขั้นตอนที่ 3 หาเลขเครือข่ายประจำแต่ละซับเน็ต

เมื่อได้เลขเครือข่ายประจำแต่ละซับเน็ตแล้วก็สามารถหาแอดเดรสในแต่ละซับเน็ตได้โดยปรับเปลี่ยนเลขโฮสต์ 5 บิต จาก 00001 ถึง 11110 ในแต่ละซับเน็ต ในที่นี้จะยกตัวอย่างเฉพาะซับเน็ต #1 ดังนี้

ซับเน็ต #1 : 1100 0010 . 0000 0001 . 0000 0001 . 001 00000 = 194.1.1.32

โฮสต์ #1 : 1100 0010 . 0000 0001 . 0000 0001 . 001 00001 = 194.1.1.33

โฮสต์ #2 : 1100 0010 . 0000 0001 . 0000 0001 . 001 00010 = 194.1.1.34

โฮสต์ #3 : 1100 0010 . 0000 0001 . 0000 0001 . 001 00011 = 194.1.1.35

: : :

โฮสต์ #27 : 1100 0010 . 0000 0001 . 0000 0001 . 001 11011 = 194.1.1.59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอสต์ #28 : 1100 0010 . 0000 0001 . 0000 0001 . 001 11100 = 194.1.1.60

ไอสต์ #29 : 1100 0010 . 0000 0001 . 0000 0001 . 001 11101 = 194.1.1.61

ไอสต์ #30 : 1100 0010 . 0000 0001 . 0000 0001 . 001 11110 = 194.1.1.62

ขั้นตอนที่ 4 แอดเรสโปรดคาสต์

แอดเรสโปรดคาสต์คำนวณได้โดยให้เลขไอสต์ทั้งหมดเป็น 1 จากตัวอย่างชั้นเน็ต#1 จะได้
แอดเรสโปรดคาสต์ดังนี้

ชั้นเน็ต #1 : 1100 0010 . 0000 0001 . 0000 0001 . 001 00000 = 194.1.1.32

โปรดคาสต์ #1 : 1100 0010 . 0000 0001 . 0000 0001 . 001 11111 = 194.1.1.63

โปรดสังเกตว่าแอดเรสโปรดคาสต์ในชั้นเน็ตใดๆจะมีค่าลดลงหนึ่งค่าจากแอดเรสชั้นเน็ตของ
ชั้นเน็ตที่อยู่ถัดไป

ขั้นตอนที่ 5 สร้างตารางสรุป

จัดทำตารางสรุปเพื่อใช้ประโยชน์ได้ภายหลัง ในกรณีจะได้ค่าดังนี้

ชั้นเน็ต	แอดเรสชั้นเน็ต	แอดเรสแรก	แอดเรสสุดท้าย	แอดเรสโปรด คาสต์
ชั้นเน็ต #1	194.1.1.32	194.1.1.33	194.1.1.62	194.1.1.63
ชั้นเน็ต #2	194.1.1.64	194.1.1.65	194.1.1.94	194.1.1.95
ชั้นเน็ต #3	194.1.1.96	194.1.1.97	194.1.1.126	194.1.1.127
ชั้นเน็ต #4	194.1.1.128	194.1.1.129	194.1.1.158	194.1.1.159
ชั้นเน็ต #5	194.1.1.160	194.1.1.161	194.1.1.190	194.1.1.191
ชั้นเน็ต #6	194.1.1.192	194.1.1.193	194.1.1.222	194.1.1.223

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

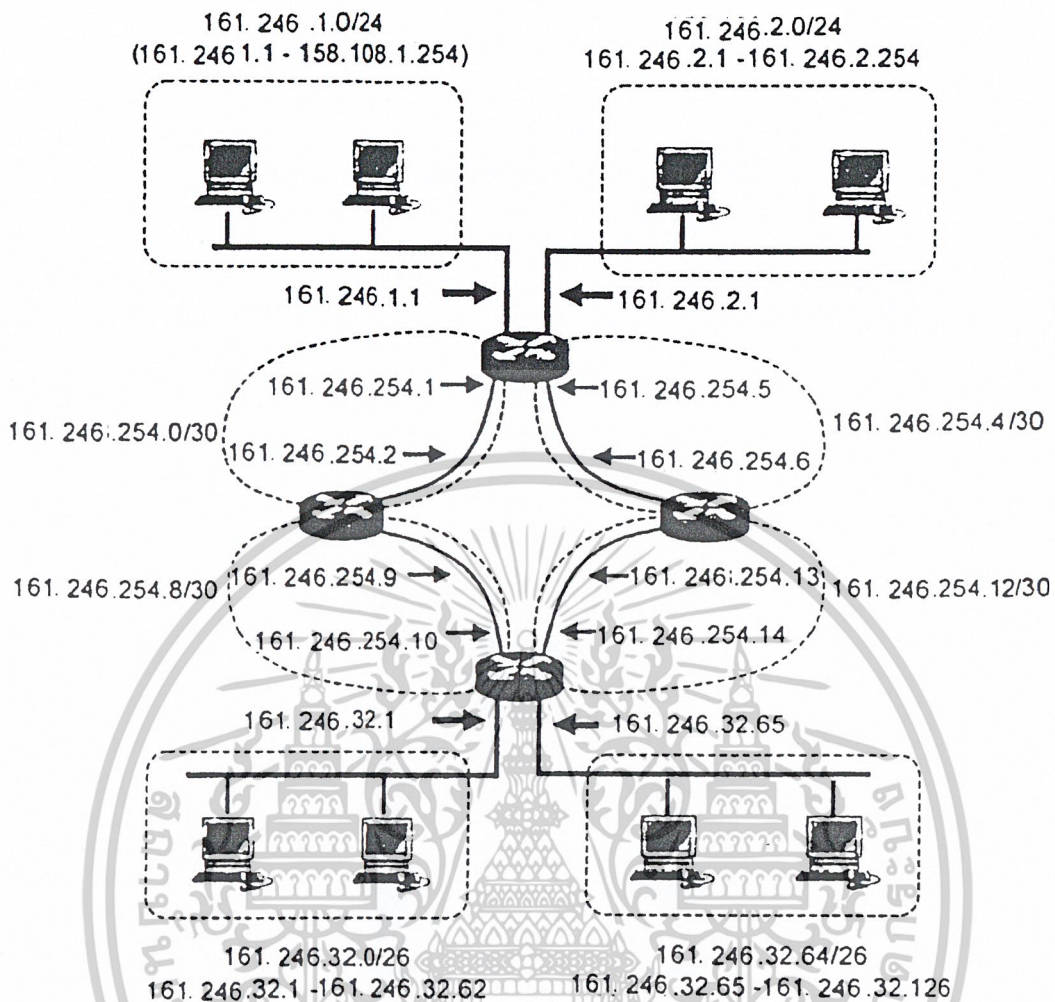
4.4 ชั้นเน็ตชนิดแปรเปลี่ยนได้

การเลือกชั้นเน็ตมาสก์ค่าหนึ่งๆเป็นการกำหนดจำนวนชั้นเน็ตและจำนวนโฮสต์ไว้ตายตัวตามค่าชั้นเน็ตนั้น ตัวอย่างเช่นเครือข่าย 161.246.0.0 ที่ใช้ชั้นเน็ตมาสก์ 255.255.255.0 จะมี 254 ชั้นเน็ตๆละ 254 โฮสต์ หากโฮสต์และเราเตอร์ใช้ค่าชั้นเน็ตมาสก์นี้ไปตลอดทั้งเครือข่ายจะเรียกชั้นเน็ตมาสก์แบบนี้ว่าชั้นเน็ตความยาวคงที่ (fix-length subnet mask) หากพิจารณาถึงหน่วยงานย่อยบางหน่วยงานหนึ่งในเครือข่ายนี้ที่ต้องการใช้โฮสต์จำนวนอย่างน้อยเพียง 50 โฮสต์ ค่าชั้นเน็ตมาสก์นี้จะทำให้สูญเสียไอพีแอดเดรสไปกว่า 200 แอดเดรส เพราะทั้ง 254 แอดเดรสจะถูกกันไว้เฉพาะชั้นเน็ตนี้เท่านั้น วิธีหนึ่งที่จะช่วยลดการสูญเสียไอพีแอดเดรสได้คือใช้ชั้นเน็ตมาสก์ต่างกันตามความจำเป็นในแต่ละชั้นเน็ต

RFC 1009 กำหนดวิธีใช้ชั้นเน็ตมาสก์ต่างค่ากันในเครือข่าย ชั้นเน็ตใดที่ต้องการแอดเดรสมากให้ใช้เลขชั้นเน็ตน้อยบิต ชั้นเน็ตที่ใช้แอดเดรสน้อยให้ใช้เลขชั้นเน็ตหลายบิต ชั้นเน็ตมาสก์ทั้งเครือข่ายจึงมีได้หลายค่า และเรียกว่า ชั้นเน็ตมาสก์ความยาวแปรค่า (VLSM : Variable Length Subnet Mask)

รูปที่ 4-10 แสดงตัวอย่างการใช้ชั้นเน็ตมาสก์ความยาวแปรค่า แต่ละเครือข่ายจะมีจำนวนโฮสต์ได้ไม่เท่ากัน ไอพีแอดเดรส 161.246.0.0 ที่ใช้ในรูปแสดงเฉพาะบางส่วนของส่วนที่นำมากำหนดเครือข่ายเท่านั้น ชั้นเน็ตมาสก์ที่ใช้มี 3 ค่าต่อไปนี้

- 255.255.255.0 สำหรับชั้นเน็ตที่มีโฮสต์จำนวนมาก (254 โฮสต์)
- 255.255.255.192 สำหรับชั้นเน็ตที่มีโอกาสจำนวนน้อย (62 โฮสต์) หรือ
- 255.255.255.252 สำหรับอินเทอร์เฟซแบบจุดต่อจุด (ใช้ได้ 16,382 จุดๆละ 2 แอดเดรส)



รูปที่ 4-9 เครือข่ายที่ใช้ชั้นเน็ตความยาวแปรค่า

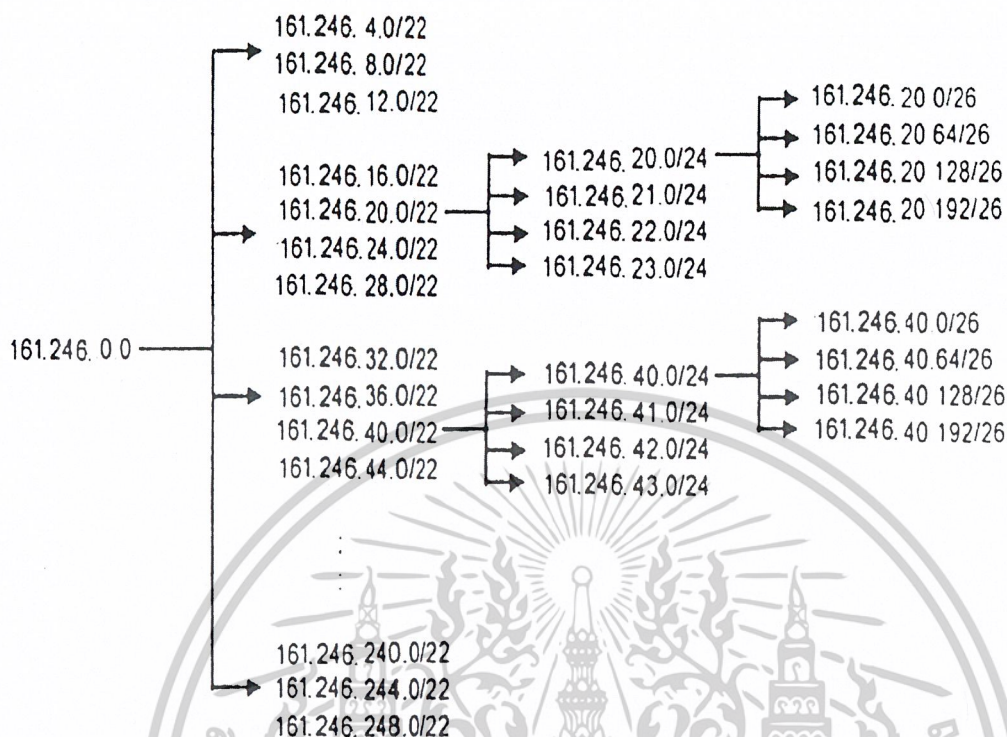
4.4.1 การแบ่งชั้นเน็ตตามสเกลความยาวแปรค่า

ชั้นเน็ตตามสเกลความยาวแปรค่าสามารถจัดแบ่งชั้นเน็ตย่อยแบบรีเคอร์ซีฟได้ เช่นการแบ่งเครือข่าย 161.246.0.0 ตามรูปที่ 4-11 เมื่อใช้ชั้นเน็ต /22 หรือค่ามาสก์ 255.255.192.0 จะมีชั้นเน็ต แต่ละชั้นเน็ตมี 1022 แอดเดรส

แต่ถ้าต้องการให้มีชั้นเน็ตที่ใช้แอดเดรสจำนวนน้อยเช่นใช้ได้ไม่เกิน 200 แอดเดรส ก็สามารถนำชั้นเน็ตชุดใหญ่มาแบ่งย่อยโดยใช้ชั้นเน็ต /24 หรือแบ่งชั้นเน็ตลงไปอีก 4 ชั้นเน็ตย่อย เช่นแบ่งชั้นเน็ต 161.246.20.0 ออกเป็น 4 ชั้นเน็ต ชั้นเน็ตย่อยที่แบ่งแล้วนี้ยังสามารถแบ่งต่อได้อีกเช่นใช้ชั้นเน็ต /26 กับ 161.246.20.0 จะได้อีก 4 ชั้นเน็ตย่อย

โปรดสังเกตว่าคุณสมบัติการแบ่งรีเคอร์ซีฟไม่ได้บังคับให้แบ่งย่อยทุกระดับ และไม่บังคับว่าจะต้องแบ่งให้ได้ชั้นเน็ตย่อยในแต่ละระดับอย่างเท่ากัน ผู้ดูแลระบบสามารถเลือกระดับการแบ่งและ

จำนวนชั้นนี้ขึ้นอยู่กับความต้องการ



รูปที่ 4-10 การแบ่งชั้นเน็ตความยาวแบบรีเคอร์ซีฟ

4.4.2 ปัญหาชั้นเน็ตมาสค์ความยาวแปรค่า

ถึงแม้ว่าชั้นเน็ตมาสค์ความยาวแปรค่าจะช่วยให้ใช้ไอพีแอดเดรสได้อย่างคุ้มค่า แต่ก็ส่งผลกระทบต่อด้านอื่นด้วย เครื่องมือที่ใช้ชั้นเน็ตมาสค์ชนิดความยาวแปรค่าไม่สามารถใช้โปรโตคอลเลือกเส้นทางบางชนิด เช่นอาร์ไอพีรุ่น 1 (RIPv1) ได้ แต่ถ้าต้องใช้โปรโตคอลอื่นที่ซับซ้อนขึ้นเช่น โอเอสพีเอฟ (OSPF)

4.5 ปัญหาการจัดสรรไอพีแอดเดรส

เครือข่ายอินเทอร์เน็ตทวีจำนวนเป็นเกือบสองเท่าในทุกๆปี จากการสำรวจข้อมูลระหว่าง ค.ศ. 1994 ถึง 1996 พบว่าแอดเดรสในกลุ่ม A และ B ซึ่งมีอยู่จำนวนน้อยกำลังจะหมดอย่างรวดเร็ว นอกจากความต้องการใช้เครือข่ายที่สูงจนกระทั่งจำนวนเครือข่ายที่จัดสรรให้ได้จะมีจำนวนลดลงอย่างมากจนเริ่มมีปัญหาขาดแคลนแอดเดรส อีกทั้งการแบ่งเครือข่ายออกเป็นคลาสก็นับเป็นสาเหตุร่วมส่วนหนึ่งด้วย เพราะเมื่อพิจารณาถึงจำนวนโฮสต์ที่รวมทั้งหมดจะพบว่าเครือข่ายคลาส A จะรวมจำนวนโฮสต์ได้ถึง 50% ของโฮสต์ทั้งหมด ส่วนคลาส B และคลาส C มีจำนวนโฮสต์ 25% และ 12.5% ของโฮสต์ทั้งหมดตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครือข่ายคลาส A ที่มีโฮสต์ประมาณสองพันล้านเครื่องแต่กลับจัดสรรได้เพียง 126 กลุ่มๆละประมาณ 16.7 ล้านเครื่อง สถิติการขอแอดเดรสคลาส A ในช่วง ค.ศ. 1994 ถึง 1996 พบว่ามีการจัดสรรให้ใช้เพิ่มขึ้น 11 เครือข่าย รวมเป็นเครือข่ายทั้งหมดที่จัดสรรให้ใช้ไปแล้วทั้งสิ้น 64 เครือข่ายและสงวนไว้อีก 63 เครือข่าย

เมื่อพิจารณาในแง่จำนวนโฮสต์ที่มีได้ในแต่ละเครือข่าย B และ C กล่าวคือ คลาส B มีโฮสต์ได้ 63,564 โฮสต์ ซึ่งค่อนข้างมากสำหรับองค์กรระดับทั่วไป ในขณะที่คลาส C มีโฮสต์ได้เพียง 254 โฮสต์ ซึ่งนับว่าน้อยเกินไปแม้สำหรับเครือข่ายขนาดเล็กก็ตาม ด้วยเหตุนี้ในช่วงก่อน ค.ศ. 1990 จึงมีผู้ขอเครือข่ายคลาส B เพิ่มขึ้นในอัตราที่สูง โดยไม่ได้ใช้โฮสต์อย่างเต็มอัตราซึ่งส่งผลให้ต้องสูญเสียแอดเดรสไม่ได้ใช้งานไปเป็นจำนวนมาก

4.5.1 นโยบายจัดสรรเครือข่าย

จากการวิเคราะห์อัตราเพิ่มของเครือข่ายคลาส B ในค.ศ. 1990 พบว่าเครือข่ายคลาส B จะหมดลง ในค.ศ. 1994 ด้วยเหตุนี้จึงต้องมีนโยบายและวิธีบรรเทาปัญหาขาดแคลนเครือข่ายคลาส B และใช้เครือข่ายคลาสอื่นอย่างเหมาะสมดังสรุปมาได้ดังนี้

- * เครือข่ายคลาส A ที่เหลืออยู่ให้สงวนไว้โดยไม่มีกำหนด
- * เครือข่ายคลาส B จะอนุมัติให้ใช้สำหรับองค์กรที่แสดงให้เห็นได้อย่างชัดเจนว่ามี ความจำเป็นอย่างแท้จริง องค์กรมีเครือข่ายย่อยภายในมากกว่า 32 เครือข่ายและใช้โฮสต์ไม่ต่ำกว่า 4096 โฮสต์
- * องค์กรใดที่ไม่เข้าข้อกำหนดเบื้องต้น ให้อนุมัติใช้เครือข่ายคลาส C หลายเครือข่าย ติดต่อกันตามจำนวนโฮสต์ที่ต้องการใช้งาน
- * เครือข่ายคลาส C ครึ่งบน (192.0.0 ถึง 207.255.255) ให้แบ่งออกเป็น 8 ส่วนแต่ละส่วน จัดสรรให้กับภูมิภาคต่างๆดังนี้
 - 192.0.0.0 – 193.255.255.255 ภูมิภาคทั่วไป
 - 194.0.0.0 – 195.255.255.255 ยุโรป
 - 196.0.0.0 – 197.255.255.255 อื่นๆ
 - 200.0.0.0 – 201.255.255.255 อเมริกากลางและใต้
 - 202.0.0.0 – 203.255.255.0 แปซิฟิก
 - 204.0.0.0 – 205.255.255.255 อื่นๆ
 - 206.0.0.0 – 207.255.255.255 อื่นๆ
- * การจัดสรรเครือข่ายคลาส C ให้จัดสรรตามความต้องการใช้งานเป็นชุดดังนี้
 - ใช้น้อยกว่า 256 โฮสต์ให้ 1 หมายเลขเครือข่าย
 - ใช้น้อยกว่า 512 โฮสต์ให้ 2 หมายเลขเครือข่ายติดกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้น้อยกว่า 1024 โฮสต์ให้ 4 หลายหมายเครือข่ายติดกัน
- ใช้น้อยกว่า 2048 โฮสต์ให้ 8 หลายหมายเครือข่ายติดกัน
- ใช้น้อยกว่า 4096 โฮสต์ให้ 16 หมายเลขเครือข่ายติดกัน

* เครือข่ายคลาส C ครึ่งล่าง (208.0.0 ถึง 223.255.255) สงวนไว้ยังไม่จัดสรร

4.5.2 (CIDR)

ถึงแม้ว่าการจัดสรรเครือข่ายคลาส C จำนวนหลายเครือข่ายแทนการขอคลาส B จะช่วยบรรเทาปัญหาขาดแคลนเครือข่าย B ลงได้ แต่ก็สร้างปัญหาใหม่ขึ้นมาแทนที่คือ เพิ่มค่าในตารางเลือกเส้นทางอย่างรวดเร็ว ทั้งนี้เนื่องจากเครือข่ายคลาส C หนึ่งคลาสจะต้องมีข้อมูลกำหนดเส้นทางไปยังแต่ละเครือข่ายแยกออกจากกัน หากพิจารณาเครือข่ายคลาส B เครือข่ายหนึ่งที่มีโฮสต์ 3000 โฮสต์ จะมีข้อมูลกำหนดเส้นทางเพียงค่าเดียวในตาราง แต่ถ้าจัดสรรให้ใช้เครือข่ายคลาส C แทนแล้วจะต้องมีข้อมูลกำหนดเส้นทาง 16 ค่าในตาราง

เครือข่ายคลาส C ที่เกิดขึ้นเป็นจำนวนมากทำให้ตารางเลือกเส้นทางในเราเตอร์หลักของอินเทอร์เน็ตมีขนาดใหญ่ ในปัจจุบันเราเตอร์หลักของอินเทอร์เน็ตต้องรับภาระเก็บตารางเส้นทางเกือบ 70,000 ค่า และหากยังคงเพิ่มขึ้นอีกในอัตราเดิมจะส่งผลให้เราเตอร์ไม่สามารถจัดการกับตารางขนาดใหญ่ได้อีกต่อไป

เพื่อแก้ปัญหานี้จึงมีการจับกลุ่มของเครือข่ายคลาส C ให้เข้ากันเพื่อให้สามารถใช้ค่ากำหนดเส้นทางเพียงค่าเดียวเป็นทางออกไปยังกลุ่มเครือข่านั้น การเลือกเส้นทางด้วยวิธีนี้ไม่คำนึงถึงการจัด “คลาส” อีกต่อไป จึงเรียกการเลือกเส้นทางแบบนี้ว่า Classless Inter Domain Routing หรือ CIDR (อ่านว่า ไซเดอร์) [RFC 1518, RFC1519, RFC1520]

ก่อนที่จะอธิบายถึงรูปแบบของไซเดอร์ต่อไป ขอให้พิจารณาถึงลักษณะสำคัญของการจัดสรรเครือข่ายที่ผ่านมาคือ ไม่ว่าจะเป็นเครือข่ายในคลาสใดก็ตาม การจัดสรรจะไม่ขึ้นอยู่กับสภาพทางภูมิศาสตร์ว่าอยู่ในทวีปใด หมายเลขเครือข่ายที่อยู่ติดกันอาจจะกระจายไปอยู่ต่างทวีปกันและจำเป็นต้องมีค่ากำหนดเส้นทางเครือข่ายนั้นเก็บประจำไว้ ในทางกลับกันหากจัดสรรเครือข่ายตามปฏิบัติตามภูมิประเทศจะสามารถลดขนาดของตารางเส้นทางเราเตอร์ลงได้มาก เพราะเราเตอร์เพียงแต่เก็บเส้นทางต่อไป ด้วยเหตุนี้การจัดสรรเครือข่ายคลาส C จึงจัดแบ่งตามภูมิประเทศเพื่อช่วยลดขนาดตารางเลือกเส้นทางลงไป

เนื่องจากก่อนหน้านี้ได้มีการจัดสรรเครือข่ายคลาส C ในช่วงตั้งแต่ 192.0.0.0 ถึง 193.255.255.255 ไปก่อนที่จะกำหนดนโยบายจัดแบ่งตามภูมิภาค แอดเดรสในช่วงดังกล่าวจึงกำหนดให้เป็นอยู่ในกลุ่มภูมิภาคทั่วไป ส่วนช่วงแอดเดรสในกลุ่ม “อื่นๆ” ยังไม่กำหนดให้กับภูมิภาคใด

แนวคิดของไซเดอร์คือการจับกลุ่มเครือข่ายด้วยกันเพื่อให้สามารถรวมค่ากำหนดเส้นทางลงเหลือเพียงค่าเดียวสำหรับเครือข่ายกลุ่มเดียวกัน เช่นองค์กรได้รับการจัดสรรเครือข่ายคลาส C 16 คลาส ตามแบบของไซเดอร์จะสามารถลดค่ากำหนดเส้นทางจาก 16 ค่าเหลือเพียงค่าเดียวได้

ข้อกำหนดสำคัญที่ทำให้เครือข่ายจับกลุ่มได้คือต้องจัดสรรเครือข่ายให้บิตลำดับสูงมีหมายเลขร่วมกัน ตัวอย่างเช่นองค์กรในยุโรปที่ต้องการใช้โฮสต์ 900 โฮสต์ อาจได้รับการจัดสรรให้ใช้ 4 เครือข่ายคลาส C คือ 195.0.32.0, 195.0.33.0, 195.0.34.0, 195.0.35.0 ทั้ง 4 เครือข่ายมีแอดเดรส 22 บิตแรกเหมือนกันคือ

1100 0011 . 0000 0000 . 0010 0000 . 0000 0000 = 195.0.32.0

1100 0011 . 0000 0000 . 0010 0001 . 0000 0000 = 195.0.33.0

1100 0011 . 0000 0000 . 0010 0010 . 0000 0000 = 195.0.34.0

1100 0011 . 0000 0000 . 0010 0011 . 0000 0000 = 195.0.35.0

เมื่อต้องการจับกลุ่มเข้าด้วยกันให้ใช้ค่ามาสก์ที่มี 22 บิตแรก เป็น “1” ซึ่งเท่ากับ 1111 1111 . 1111 1111 . 1111 1100 . 0000 0000 หรือ 255.255.252.0 ดังนั้นไม่ว่าแอดเดรสใดก็ตามใน 4 เครือข่ายนี้เมื่อนำผ่านลอจิก “AND” กับค่ามาสก์จะได้ค่า 195.0.32.0 เสมอ ค่านี้จะเป็นค่ากำหนดเส้นทางประจำเครือข่ายทั้งกลุ่ม

โปรดสังเกตว่าจำนวนบิตที่ใช้ทำมาสก์ในไซเคอร์จะมีค่าน้อยกว่าดีฟอลต์มาสก์ จากตัวอย่างนี้คือใช้น้อยลงไป 2 บิต จำนวนบิตที่น้อยลงไปจะเป็นตัวกำหนดจำนวนเครือข่ายในกลุ่มคือ $2^2 = 4$ หรือหากใช้แอดเดรส 21 บิตรวมกันจะเหลือ 3 บิตสำหรับกำหนดเครือข่ายจำนวน 8 เครือข่ายในกลุ่ม ด้วยวิธีการนี้ต้องจัดสรรให้แอดเดรสเครือข่ายมีค่า *ต่อเนื่องกัน* และมีจำนวนเครือข่ายเป็นค่า 2^n คือ 1, 2, 4, 8, 16, 32 หรือ 64 สำหรับผู้ให้บริการอินเทอร์เน็ตรายใหญ่อาจได้รับการจัดสรรแอดเดรสคลาส C หลายร้อยแอดเดรสต่อเนื่องกันเพื่อนำไปจัดสรรต่อให้กับผู้ให้บริการในรูปแบบเดียวกัน

รูปแบบการขุบรวมค่ากำหนดเส้นทางของไซเคอร์มันเรียกว่าเป็น **การรวมแอดเดรส** (address aggregation) หรือ **การสรุปรวมแอดเดรส** (address summarization) หรือ อีกชื่อหนึ่งที่นิยมเรียกคือ **ซูเปอร์เน็ต** (supernetting) เพราะเป็นการรวมเครือข่ายย่อยเข้าด้วยกันเป็นเครือข่ายใหญ่ซึ่งตรงข้ามกับซับเน็ตที่แบ่งเครือข่ายใหญ่ออกเป็นเครือข่ายย่อย

เราเตอร์ในปัจจุบันจะใช้โปรโตคอลเลือกเส้นทางระหว่างเครือข่ายด้วย บีจีพี ซึ่งรองรับการทำงานร่วมกับไซเคอร์ ผู้ดูแลระบบสามารถกำหนดให้เราเตอร์ประกาศเส้นทางสรุปไปยังเครือข่ายอื่นได้

4.5.3 ไพรเวตแอดเดรส

ทางหนึ่งในการสงวนไอพีแอดเดรสคือการกำหนดไอพีแอดเดรสขึ้นชุดหนึ่งเพื่อให้องค์กรใด ๆ สามารถนำมาใช้ได้ ไอพีดังกล่าวนี้เรียกว่า **ไพรเวตแอดเดรส** (private address) ซึ่งกำหนดขึ้นเพื่อใช้กับองค์กรที่ไม่ได้ต่อเชื่อมเข้าอินเทอร์เน็ต หรือหากองค์กรนั้นเชื่อมต่อกับอินเทอร์เน็ตก็ต้องจำกันไม่ให้ปล่อยแพ็กเก็ตเกิดจากแอดเดรสนี้สู่อินเทอร์เน็ต ไพรเวตแอดเดรสที่จัดสรรไว้มีอยู่ 3 กลุ่มแยกตามแต่ละคลาสคือ

10.0.0.0	ถึง	10.255.255.255	(1 ชุด คลาส A)
172.16.0.0	ถึง	172.31.255.255	(16 ชุด คลาส B)
192.168.0.0	ถึง	192.168.255.255	(256 ชุด คลาส C)

องค์กรที่ต่อเชื่อมเข้ากับอินเทอร์เน็ตและประสบปัญหาแอตเดรสไม่พอใช้ ตัวอย่างเช่น องค์กรที่ได้แอตเดรสคลาส C สามารถใช้ไพรเวตแอตเดรสโดยต้องแยกส่วนของเครือข่ายไพรเวตไม่ให้เชื่อมกับอินเทอร์เน็ตโดยตรง การป้องกันทำได้โดยใช้เราเตอร์หรือไฟร์วอลล์กรองแพ็กเก็ตระหว่างเครือข่ายกับอินเทอร์เน็ต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

โพรโทคอลเลือกเส้นทาง (Routing Protocol)

การเลือกเส้นทางเป็นหน้าที่สำคัญอย่างหนึ่งของไอพี เราเตอร์อาศัยตารางเส้นทาง (Routing Table) เพื่อนำส่งคำคำแกรมไปยังเครือข่ายปลายทาง ตารางเส้นทางอาจปรับเปลี่ยนได้ตามสภาพเครือข่ายอัตโนมัติโดยใช้โพรโทคอลเลือกเส้นทาง หรือผู้ดูแลระบบเป็นคนปรับเปลี่ยนเส้นทางเอง

5.1 การเลือกเส้นทาง (Routing)

การเลือกเส้นทางเป็นกระบวนการที่เกิดขึ้นในระดับชั้นที่ 3 หรือระดับเน็ตเวิร์ก ของแบบจำลองทียซีพี/ไอพี ซอฟต์แวร์ไอพีที่อยู่ในโฮสต์หรือเราเตอร์จะนำส่งคำคำแกรมไปตามเส้นทาง โดยอาศัยเลขเครือข่ายของไอพีแอดเดรสตามแต่ละคลาส เลขเครือข่ายเป็นเสมือนคำกำหนดตำแหน่งปลายทางของเครือข่ายซึ่งคล้ายกับรหัส 3 ตัวแรกที่กำหนดหมายเลขโทรศัพท์ที่ ระบบรหัสหมายเลขจะกำหนดที่ตั้งทางภูมิศาสตร์ด้วยว่าอยู่ที่ทิศทางใด แต่เลขเครือข่ายในไอพีแอดเดรสไม่ได้มีส่วนสัมพันธ์กับที่ตั้งของเครือข่าย

เครือข่ายโดยทั่วไปประกอบด้วยสถานีปลายทาง (End Node) และเราเตอร์หรืออุปกรณ์อื่นทำงานร่วมกัน โดยกระบวนการเลือกหาเส้นทางจะเกิดขึ้นทั้งที่สถานีปลายทางและที่เรเตอร์ คือ

1. สถานีปลายทางที่เป็นสถานีส่งต้องทราบว่าจะนำส่งคำคำแกรมให้เรเตอร์ได้อย่างไรและเมื่อใด
2. เราเตอร์ต้องทราบเส้นทางเชื่อมโยงไปยังเรเตอร์ตัวอื่น เพื่อส่งคำคำแกรมไปตามเส้นทางที่เหมาะสมที่สุด
3. เราเตอร์ที่เชื่อมกับเครือข่ายของสถานีปลายทางต้องทราบถึงวิธีส่งคำคำแกรมไปยังสถานีปลายทางนั้น

ในตอนแรกสถานีต้นทางจะเป็นผู้ตัดสินใจขั้นแรกว่าต้องส่งคำคำแกรมไปยังสถานีในเครือข่ายด้วยกันเองหรือต้องส่งผ่านเรเตอร์ โดยสถานีต้นทางจะเปรียบเทียบเลขเครือข่ายของแอดเดรสต้นทางและปลายทางกับค่าซับเน็ตมาสก์ หากได้เลขเครือข่ายเหมือนกันแสดงว่าสถานีปลายทางอยู่ในเครือข่ายเดียวกัน สถานีต้นทางจะใช้เออาร์ทีสอปรแกรมฮาร์ดแวร์แอดเดรสหรืออ่านจากแคช และบรรจุฮาร์ดแวร์แอดเดรสเข้าสู่เฟรมคำคำแกรมเพื่อส่งตรงไปยังสถานีปลายทาง แต่ถ้าเลขเครือข่ายมีค่าต่างกันแสดงว่าสถานีปลายทางอยู่ต่างเครือข่ายกัน สถานีส่งก็จะส่งเฟรมไปให้เรเตอร์เพื่อให้เรเตอร์นำส่งต่อไป เมื่อเรเตอร์ได้รับเฟรมนี้ก็จะส่งต่อไป ไอพีแอดเดรสต้นทางและปลายทางในไอพีคำคำแกรมจะไม่เปลี่ยนแปลงค่าระหว่างการลำเลียง แต่ฮาร์ดแวร์แอดเดรสจะเปลี่ยนแปลงไปตามเครือข่าย

5.2 ตารางเส้นทาง(Routing Table)

โฮสต์และเราเตอร์จะเก็บแอตทริบิวต์ปลายทางสำหรับใช้เป็นเส้นทางส่งค่าคำกรณไว้ในรูปตารางที่เรียกว่า ตารางเส้นทาง (Routing Table) ค่าในตารางเส้นทางมักประกอบด้วยไอพีแอตทริบิวต์ของเครือข่ายปลายทาง และเกตเวย์ซึ่งเป็นทางออกของค่าคำกรณ

ตารางเส้นทางจะบรรจุแอตทริบิวต์หนึ่งซึ่งทำหน้าที่เป็นช่องทางออกไปสู่เครือข่ายใดๆ ที่ไม่ได้ระบุอยู่ในตารางเส้นทาง แอตทริบิวต์นี้เรียกว่า ดีฟอลต์เกตเวย์ (Default Gateway) หรือ ดีฟอลต์เราเตอร์ (Default Router) ซึ่งแทนด้วยแอตทริบิวต์ 0.0.0.0 ไอพีจะเลือกเส้นทางโดยตรวจสอบว่าแอตทริบิวต์ปลายทางตรงกับรายการใดในตาราง แล้วส่งค่าคำกรณไปยังเกตเวย์ของรายการนั้น หากไม่พบรายการใดในตารางก็จะส่งไปยังดีฟอลต์เกตเวย์

5.3 ประเภทของการเลือกเส้นทาง

เมื่อค่าคำกรณเดินทางออกนอกเครือข่ายที่อาจต้องผ่านเราเตอร์จำนวนมากโดยมีเส้นทางลำเลียงได้หลายเส้นทาง ปัญหาที่สำคัญคือเราเตอร์จะทราบได้อย่างไรว่ามีเส้นทางใดบ้าง และเส้นทางใดที่เป็นเส้นทางที่ดีที่สุด

เราเตอร์จะส่งค่าคำกรณได้จำเป็นต้องมีแผนที่เครือข่าย ซึ่งก็คือตารางเส้นทาง สำหรับวิธีที่ใช้ในการสร้างตารางเส้นทางนี้มี 2 แบบคือ

5.3.1 การเลือกเส้นทางแบบสถิต (Static Routing) ตารางเลือกเส้นทางสร้างขึ้นและแก้ไขโดยผู้ดูแลระบบเครือข่าย

5.3.2 การเลือกเส้นทางแบบไดนามิก (Dynamic Routing) ใช้ซอฟต์แวร์คำนวณหาค่าตารางเลือกเส้นทาง ตารางสามารถปรับเปลี่ยนได้หากสภาพเครือข่ายเปลี่ยนไป

5.3.1 การเลือกเส้นทางแบบสถิต

การเลือกเส้นทางแบบนี้ผู้ดูแลระบบเครือข่ายเป็นผู้พิจารณาและคำนวณหาเส้นทางทั้งหมด โดยใส่ค่าตารางเส้นทางให้กับเราเตอร์ทุกตัว ตารางเลือกเส้นทางนี้จะมีค่าตายตัวตลอดถึงแม้ว่าสภาพของเครือข่ายจะเปลี่ยนไปดังนั้นผู้ดูแลระบบจะต้องคอยตรวจสอบเครือข่ายและปรับเปลี่ยนตารางเส้นทางให้ถูก

รูปที่ 5-1 แสดงเครือข่ายที่ประกอบด้วยเราเตอร์ 2 ตัว เราเตอร์แต่ละตัวจะมีตารางเส้นทางที่ผู้ดูแลระบบป้อนเพื่อใช้กำหนดทิศทางของการส่งค่าคำกรณ ทุกเครือข่ายที่เชื่อมโดยตรงกับเราเตอร์อื่นจะมีค่าเกตเวย์เท่ากับไอพีแอตทริบิวต์ของเราเตอร์ที่เชื่อม ส่วนเครือข่ายที่ต้องผ่านเราเตอร์อื่นจะมีค่าเกตเวย์เท่ากับไอพีแอตทริบิวต์ของเราเตอร์ขั้นถัดไป (Next Hop Router) เช่นตารางเส้นทางในเราเตอร์ R1 มีค่าเกตเวย์ประจำเครือข่าย 1 และเครือข่าย 2 เท่ากับ 1.1 และ 2.1 ตามลำดับ ส่วนเกตเวย์สำหรับเครือข่าย 4 จะมีค่าเท่ากับ 3.2 ซึ่งเป็นแอตทริบิวต์ของเราเตอร์ R2 (R2 เป็นเราเตอร์ขั้นถัดไปของ R1)

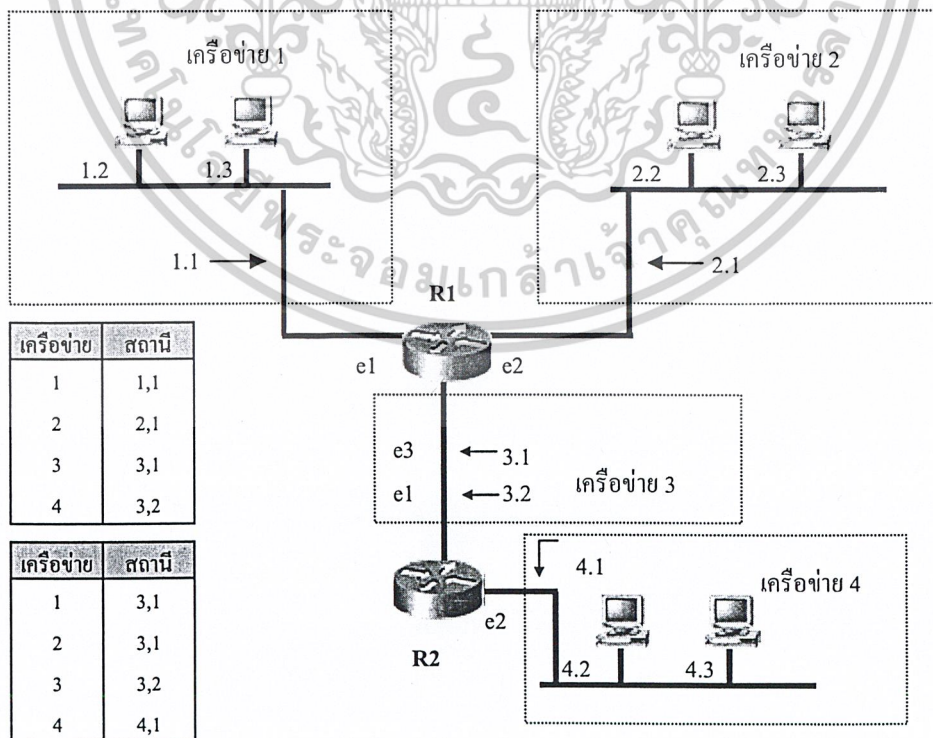
การเลือกเส้นทางแบบสแตติกนิยมใช้กับการเชื่อมโยงแบบจุดต่อจุดระหว่างเราเตอร์สองตัว เช่น เครือข่ายมีทางออกไปสู่ภายนอกหรืออินเทอร์เน็ตเพียงช่องทางเดียว มักจะกำหนดเส้นทางแบบสแตติก การเลือกเส้นทางแบบสแตติกมีข้อดีข้อเสียดังนี้

ข้อดี

1. สะดวกต่อการใช้งานกับเครือข่ายขนาดเล็ก
2. ไม่ต้องใช้ซอฟต์แวร์เลือกเส้นทาง เราเตอร์ไม่จำเป็นต้องมีซอฟต์แวร์ที่ซับซ้อน
3. ประหยัดแบนด์วิดท์เครือข่ายเนื่องจากไม่ต้องแลกเปลี่ยนข้อมูลระหว่างเราเตอร์
4. สามารถจำกัดให้เข้าถึงได้เฉพาะในเครือข่ายที่ต้องการ

ข้อเสีย

1. ไม่เหมาะกับเครือข่ายขนาดใหญ่ เพราะผู้ดูแลระบบสามารถคำนวณเส้นทางในเครือข่ายที่มีขนาดเล็กได้ แต่ในเครือข่ายขนาดใหญ่ที่มีเราเตอร์เป็นจำนวนมากการคำนวณและป้อนค่าเข้าสู่เราเตอร์โดยตรงเป็นสิ่งที่เกินวิสัย
2. ไม่สะดวกต่อการเปลี่ยนโครงสร้างของเครือข่าย เพราะผู้ดูแลต้องคำนวณหาเส้นทางใหม่ทุกครั้งที่มีการเปลี่ยนโครงสร้างของเครือข่าย
3. ตารางเส้นทางเป็นตารางคงตัวไม่สามารถเปลี่ยนแปลงได้เอง ถ้าเส้นทางใดถูกตัดขาดไป ผู้ดูแลระบบจะต้องคอยตรวจสอบและแก้ไขเอง



รูปที่ 5-1 เครือข่ายการแสดงการเลือกเส้นทางแบบสแตติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.2 การเลือกเส้นทางแบบไดนามิก

การเลือกเส้นทางแบบนี้จะใช้ซอฟต์แวร์ทำหน้าที่แลกเปลี่ยนข้อมูลการเลือกเส้นทางระหว่างเราเตอร์ด้วยกัน โดยใช้โพรโทคอลเลือกเส้นทาง เราเตอร์จะสร้างตารางเลือกเส้นทางจากสภาพเครือข่ายขณะนั้น หากเครือข่ายมีการเปลี่ยนแปลงตารางเส้นทางก็จะเปลี่ยนแปลงตามไปด้วย

การเลือกเส้นทางแบบไดนามิกต้องอาศัยการแลกเปลี่ยนค่าเส้นทางระหว่างเราเตอร์และใช้ชีพียุในเราเตอร์เพื่อสร้างตารางเส้นทาง เราเตอร์ประเภทนี้จึงมักมีราคาสูงกว่าเราเตอร์ที่มีโพรโทคอลแบบสแตติกอย่างเดียว เพราะต้องออกแบบซอฟต์แวร์ให้ทำงานตามโพรโทคอลเลือกเส้นทาง และชีพียุต้องมีสมรรถนะสูงพอในการคำนวณหาตารางเส้นทาง

ข้อดีของการการเลือกเส้นทางแบบไดนามิก

1. รองรับขนาดเครือข่ายที่ขยายขึ้นเป็นลำดับได้
2. ตารางเส้นทางเปลี่ยนแปลงค่าเองตามการทำงานของซอฟต์แวร์ เส้นทางใดที่ถูกตัดขาดจะมีการหาเส้นทางใหม่ทดแทน

ประเภทโพรโทคอลเลือกเส้นทางแบบไดนามิก

โพรโทคอลเลือกเส้นทางแบบไดนามิกสามารถจัดแยกประเภทออกได้หลายรูปแบบ ในที่นี้จะกล่าวเพียง 2 รูปแบบคือ

- 5.3.2.1 โพรโทคอลเกตเวย์ภายนอกและภายใน (Interior Gateway Protocol และ Exterior Gateway Protocol)
- 5.3.2.2 โพรโทคอลดิสเทนซ์เวกเตอร์และลิงก์สเทต (Distance Vector Protocol และ Link State Protocol)

5.3.2.1 โพรโทคอลเกตเวย์ภายนอกและภายใน

การจัดแบบนี้แบ่งชนิดโพรโทคอลออกเป็นโพรโทคอลเกตเวย์ภายนอกและโพรโทคอลเกตเวย์ภายใน โพรโทคอลเกตเวย์ภายนอกมีหน้าที่แลกเปลี่ยนข้อมูลเส้นทางระหว่างเครือข่ายที่มีการบริหารงานเป็นอิสระออกจากกัน โดยแต่ละเครือข่ายที่มีการบริหารเป็นอิสระออกจากกันจะเรียกว่าระบบออโตโนมัส (Autonomous System) แต่ละระบบออโตโนมัสมีหมายเลขประจำของตนเองเรียกว่า เลขระบบออโตโนมัส (Autonomous System Number) หมายเลขนี้สามารถขอได้จากหน่วยงานนิค (NIC) ประจำภูมิภาค เลขระบบออโตโนมัสเป็นค่าที่ระบุข้อมูลเส้นทางที่แลกเปลี่ยนระหว่างเครือข่ายนั้นมาจากที่ใด

อินเทอร์เน็ตในยุคแรกใช้โพรโทคอล อีจีพี (EGP: Exterior Gateway Protocol) เป็นโพรโทคอลเกตเวย์ภายนอก แต่ในปัจจุบันโพรโทคอลที่นิยมใช้ระหว่างเครือข่ายคือบีจีพี (BGP : Border Gateway Protocol) และนำมาใช้งานแทนอีจีพี บีจีพีผ่านการพัฒนามาเป็นลำดับกระทั่งปัจจุบันเป็นรุ่นที่ 4 จึงเรียกว่า บีจีพี-4

โปรโตคอลเกตเวย์ภายในเป็นโปรโตคอลที่ออกแบบเพื่อใช้งานในระบบอัตโนมัติหนึ่งๆ เช่น อาร์ไอพี (RIP : Routing Information Protocol) และ โอเอสพีเอฟ (OSPF: Open Shortest Path First) โปรโตคอลทั้งสองนี้เป็นที่ยอมรับเป็นมาตรฐานสากลและใช้งานอย่างแพร่หลายในเครือข่ายทั่วไป ในขณะที่โปรโตคอลเฉพาะที่ออกแบบโดยบริษัทซิสโก้ (Cisco) คืออีไอจีอาร์พี (EIGRP : Enhanced Interior Gateway Routing Protocol) เป็นอีกโปรโตคอลหนึ่งที่นิยมใช้ตามความแพร่หลายของเราเตอร์ของซิสโก้

เมตริก (Metric)

เมตริกเป็นค่าที่นำมาใช้คำนวณหาว่าเส้นทางใดเหมาะต่อการใช้มากกว่าเส้นทางอื่น ค่าเมตริกที่ใช้อาจเป็นได้ทั้งระยะทาง เวลาหน่วง ความน่าเชื่อถือ หรือความเร็ว โปรโตคอลที่ไม่สลับซับซ้อนอาจจะเลือกใช้เมตริกเพียงประเภทใดประเภทหนึ่ง เช่นอาจใช้เฉพาะระยะทางซึ่งนับจากจำนวนเราเตอร์ที่ต้องส่งผ่านหรือเรียกว่าจำนวนขั้น (Hop Count)

จำนวนขั้นนับจากจำนวนเราเตอร์ที่ปลายทางต้องเดินผ่านดังตัวอย่างในรูปที่ 5-2 หากค่าค่าแกรมไม่ต้องเดินทางผ่านเราเตอร์จำนวนขั้นจะเท่ากับ 0 หากต้องข้ามเราเตอร์หนึ่งตัว จำนวนขั้นจะเพิ่มขึ้นทีละ 1 แต่โปรโตคอลบางชนิดอาจนับจำนวนขั้นตามจำนวนลิงก์ที่ปลายทางเดินผ่านแทนการนับด้วยจำนวนเราเตอร์ หากเครือข่ายอยู่ติดกับเราเตอร์จะมีจำนวนขั้นเท่ากับ 1 แทนที่จะเท่ากับ 0

5.3.2.2 โปรโตคอลดิสเทนซ์เวกเตอร์และลิงก์สเตต

การจัดแบ่งโปรโตคอลในแบบนี้อาศัยรูปแบบของข้อมูลที่ส่งผ่านระหว่างเราเตอร์และวิธีที่เรารู้อะไรว่าเราเตอร์สร้างตารางเลือกเส้นทางจากข้อมูลนั้น

เราเตอร์ที่ใช้โปรโตคอลดิสเทนซ์เวกเตอร์อาศัยระยะทางเพื่อกำหนดว่าเส้นทางใดเหมาะสมกว่าเส้นทางอื่น ความหมายของดิสเทนซ์เวกเตอร์คือใช้ระยะทางเป็นค่าเมตริก และเอเจนต์ของเครือข่ายปลายทางเป็นเวกเตอร์กำหนดจุดหมายปลายทาง

การทำงานพื้นฐานของโปรโตคอลดิสเทนซ์เวกเตอร์จะส่งข้อมูลเลือกเส้นทางไปยังเราเตอร์ที่อยู่ข้างเคียงทุกตัวอย่างสม่ำเสมอเป็นช่วงเวลา ข้อมูลเลือกเส้นทางประกอบด้วยตารางเลือกเส้นทางของตนเองทั้งหมดที่กำกับด้วยเมตริก เราเตอร์แต่ละตัวจะใช้ตารางเส้นทางของตนเองร่วมกับตารางเส้นทางที่ได้รับมาใหม่เพื่อคำนวณหาระยะทางที่สั้นที่สุด โปรโตคอลดิสเทนซ์เวกเตอร์ที่นิยมใช้ในปัจจุบันได้แก่ อาร์ไอพี ค่าเมตริกที่ใช้คือจำนวนขั้น อาร์ไอพีถือว่าเส้นทางที่ดีที่สุดคือเส้นทางที่มีจำนวนขั้นน้อยที่สุด

โปรโตคอลลิงก์สเตตไม่ได้แลกเปลี่ยนตารางเส้นทางโดยตรงเหมือนกับที่ใช้ในดิสเทนซ์เวกเตอร์ หากแต่เราเตอร์แต่ละตัวจะตรวจสอบ สถานะลิงก์ (Link State) ที่เชื่อมไปยังเราเตอร์ข้างเคียงว่าใช้งานได้หรือไม่พร้อมกับค่าเมตริกซึ่งโดยทั่วไปแล้วเป็นความเร็วของสายสื่อสาร เช่นให้ค่า 1 สำหรับสายสื่อสารที่มีความเร็ว 2 เมกะบิตต่อวินาที และ 10 สำหรับสายสื่อสารที่มีความเร็ว 9600 บิตต่อวินาที โปรโตคอลลิงก์สเตตที่นิยมใช้คือโอเอสพีเอฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 อาร์ไอพี(RIP)

อาร์ไอพีมีกำเนิดมาจากโพรโตคอลเลือกเส้นทางในระบบเครือข่ายซีร็อกซ์ซึ่งต่อมาได้พัฒนาเป็น อาร์ไอพีและโพรโตคอลอื่นได้แก่ไอพีเอกซ์ (โนเวลล์) และไอจีอาร์พี (ซิสโก้) อาร์ไอพีเป็นโพรโตคอลแบบดิสแทนซ์เวกเตอร์และใช้ระยะทางเป็นค่าเมตริกเพื่อหาเส้นทางที่ดีที่สุดสำหรับการเลือกเส้นทาง

อาร์ไอพีเป็นโพรโตคอลแบบดิสแทนซ์เวกเตอร์ ขั้นตอนพื้นฐานที่อาร์ไอพีใช้คือขั้นตอนวิธีของฟอร์ด/ฟูกเกอร์สัน หรือเรียกอีกชื่อว่าเบลแมน-ฟอร์ด เราเตอร์อาร์ไอพีจะเก็บตารางเส้นทางซึ่งประกอบด้วยแอดเดรสเครือข่าย แอดเดรสของเราเตอร์ถัดไปซึ่งเป็นเกตเวย์ไปยังเครือข่ายนั้น และเมตริกประจำเส้นทางซึ่งโดยปกติจะนับตามจำนวนเราเตอร์ระหว่างทาง

5.4.1 การทำงานของอาร์ไอพี

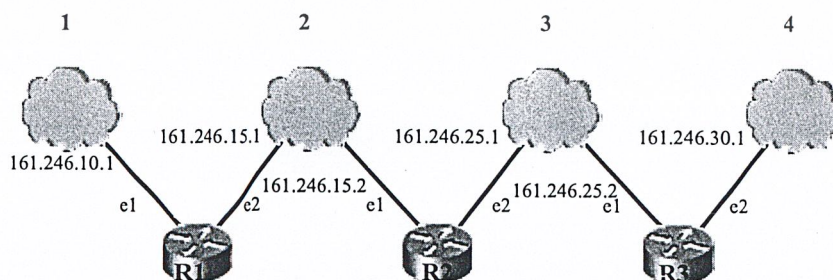
เริ่มแรกเราเตอร์แต่ละตัวจะได้รับการกำหนดแอดเดรสเครือข่ายประจำแต่ละอินเทอร์เฟซของเราเตอร์นั้น ซึ่งเราเตอร์ก็จะสร้างตารางเส้นทางจากแอดเดรสเครือข่ายที่มี แล้วส่งตารางเส้นทางของตัวเองไปให้เราเตอร์ข้างเคียงด้วยการบรอดคาสต์

เมื่อเราเตอร์ได้รับตารางเส้นทางจากเราเตอร์ตัวอื่น ก็จะปรับตารางเส้นทางของตัวเองพร้อมทั้งบรอดคาสต์ตารางของตัวเองไปให้เราเตอร์ข้างเคียงอีกเช่นกัน วิธีนี้ทำให้เราเตอร์แต่ละตัวสามารถคำนวณระยะทางและทิศทางของเครือข่ายอื่นที่เราเตอร์นั้นไม่ได้เชื่อมต่อได้ ในที่สุดเราเตอร์ทุกตัวก็จะทราบแอดเดรสทั้งหมดของเครือข่าย สำหรับการปรับค่าตารางเส้นทางเราเตอร์จะพิจารณาโดย

1. ถ้าเป็นเส้นทางใหม่ที่ไม่อยู่ในตารางเส้นทาง จะใส่เส้นทางนั้นเข้าตารางเลย
2. ถ้าเป็นเส้นทางที่มีข้อมูลอยู่ในตารางเส้นทางแล้ว เราเตอร์จะพิจารณาว่าเป็นเส้นทางที่สั้นกว่าข้อมูลที่มีอยู่ในตาราง จะแทนที่ข้อมูลในตารางด้วยข้อมูลใหม่
3. ถ้าได้รับข้อมูลเส้นทางจากเราเตอร์ R ใดๆ และตรวจพบว่าในตารางมีเส้นทางซึ่งเราเตอร์ R เป็นเกตเวย์อยู่แล้ว ให้ปรับค่าเส้นทางใหม่ตามค่าที่ได้รับจากเราเตอร์ R นั้น

ตัวอย่างการสร้างตารางเส้นทาง

สมมติให้มีเครือข่ายดังรูป 5.2



รูปที่ 5-2 เครือข่ายสาธิตการทำงานของอาร์ไอพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตอนแรกที่เราเตอร์ทั้ง 3 ตัวเริ่มทำงาน ค่าในตารางเส้นทางของเราเตอร์แต่ละตัวที่เกิดจากการติดตั้งค่าของผู้ดูแลเครือข่ายจะเป็นเครือข่ายที่เชื่อมต่อกับเราเตอร์นั้นโดยตรงดังนี้

ตาราง R1

เครือข่าย	เกตเวย์	อินเทอร์เฟซ	เมตริก
161.246.10.0	0.0.0.0	e1	1
161.246.15.0	0.0.0.0	e2	1

ตาราง R2

เครือข่าย	เกตเวย์	อินเทอร์เฟซ	เมตริก
161.246.15.0	0.0.0.0	e1	1
161.246.25.0	0.0.0.0	e2	1

ตาราง R3

เครือข่าย	เกตเวย์	อินเทอร์เฟซ	เมตริก
161.246.25.0	0.0.0.0	e1	1
161.246.30.0	0.0.0.0	e2	1

ตารางเลือกเส้นทางตามตัวอย่างข้างต้นแสดงเพียงฟิลด์สำคัญซึ่งประกอบด้วย

เครือข่าย : ไอพีแอดเดรสของเครือข่ายปลายทาง

เกตเวย์ : ไอพีแอดเดรสของเราเตอร์ซึ่งเป็นทางออกไปสู่เครือข่ายปลายทาง

อินเทอร์เฟซ : อินเทอร์เฟซของเราเตอร์

เมตริก : จำนวนขั้น

ในตอนแรกเราเตอร์จะมีตารางเส้นทางเฉพาะเครือข่ายที่อยู่ติดกับเราเตอร์ แต่เส้นทางไปเครือข่ายอื่นนั้นจะได้จากการแลกเปลี่ยนตารางเส้นทางกับเราเตอร์ตัวอื่น ซึ่งอาร์ไอพีนี้กำหนดให้เราเตอร์ประกาศเส้นทางให้กับเราเตอร์ข้างเคียงทุกๆ 30 วินาที หากสมมติให้ R1 บรอดแคสต์ R2 จะได้รับแอดเดรส 161.246.10.0 และ 161.246.15.0 จาก R1

เครือข่าย 161.246.15.0 เป็นค่าที่มีอยู่ในตารางแล้วและเป็นเครือข่ายที่อยู่ติดกับเราเตอร์ R2 จึงไม่เปลี่ยนค่า ส่วนเครือข่าย 161.246.10.0 เป็นค่าใหม่ R2 จะเพิ่มค่าเข้าไปในตารางโดยมองว่า “เราเตอร์ R2 สามารถไปถึงเครือข่าย 161.246.10.0 ได้โดยผ่านเราเตอร์ R1 ซึ่งถ้าเราเตอร์ R1 สามารถไปยังเครือข่าย 161.246.10.0 ได้ด้วยระยะทาง 1 จำนวนขั้น ดังนั้นถ้าเราเตอร์ R2 สามารถไปเครือข่ายนั้นโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านเราเตอร์ R1 ก็สามารถึงได้ด้วยระยะทางที่เราเตอร์ R1 ไป + 1 จึงเท่ากับ 1+1 = 2 จำนวนชั้น และใช้เกตเวย์ 161.246.15.1 เป็นทางออก ตาราง ของ R2 หลังจากได้รับตารางเส้นทางที่ R1 บรอดแคสต์ให้จะเป็น

ตาราง R2

เครือข่าย	เกตเวย์	อินเทอร์เฟซ	เมตริก
161.246.10.0	161.246.15.1	e1	2
161.246.15.0	0.0.0.0	e1	1
161.246.25.0	0.0.0.0	e2	1

← ค่าที่ได้ใหม่จาก R1

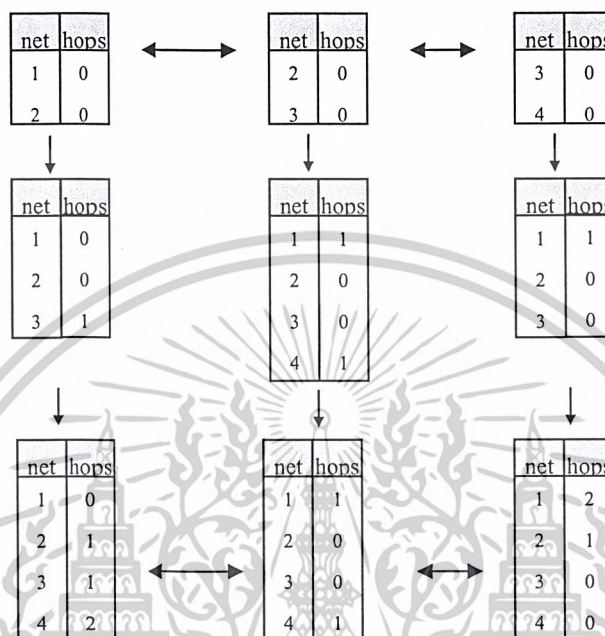
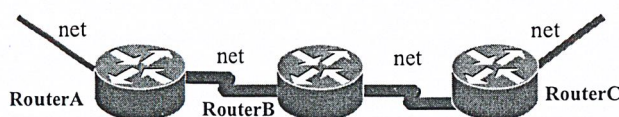
ถ้าต่อมา R3 บรอดคาสต์เฉพาะเราเตอร์ R2 ก็จะได้รับค่าและเปลี่ยนค่าในตารางเช่นเดียวกัน ดังนั้นตารางเส้นทางของเราเตอร์ R2 จะเป็น

ตาราง R2

เครือข่าย	เกตเวย์	อินเทอร์เฟซ	เมตริก
161.246.10.0	161.246.15.1	e1	2
61.246.15.0	0.0.0.0	e1	1
161.246.25.0	0.0.0.0	e2	1
161.246.30.0	161.246.25.2	e2	2

← ค่าที่ได้ใหม่จาก R3

ตารางของ R2 ในตอนแรกมีเพียงแค่เครือข่ายที่เชื่อมต่อโดยตรงกับเราเตอร์นั้น หลังจากที่ได้รับบรอดคาสต์จาก R1 และ R3 ตารางของ R2 ก็จะมีข้อมูลของเครือข่ายครบทุกเครือข่าย รูปที่ 5-3 แสดงการประกาศและปรับค่าตารางเส้นทางของเราเตอร์ทั้งหมด โดยสมมติให้แต่ละเราเตอร์ประกาศค่าตารางพร้อมกัน



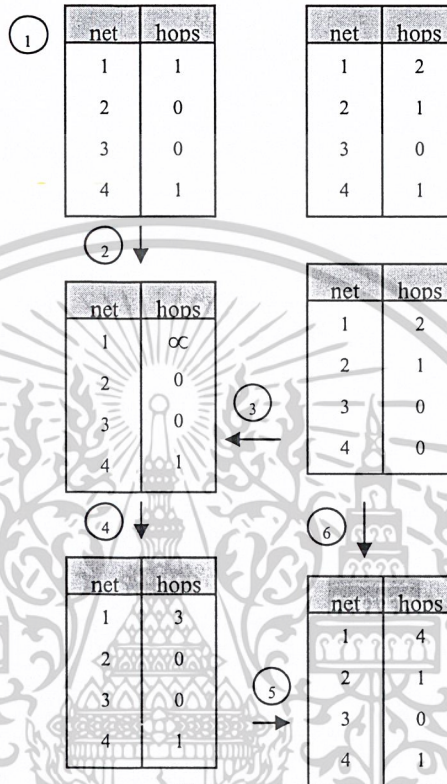
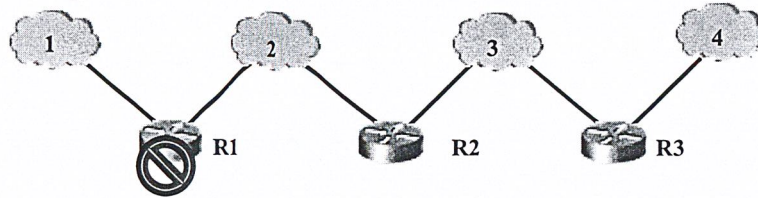
รูปที่ 5-3 การประกาศค่าและปรับค่าและตารางของอาร์ไอพี

5.4.2 การปรับค่าเมื่อเครือข่ายเปลี่ยน

กรณีที่โครงสร้างของเครือข่ายไม่เปลี่ยนแปลง ตารางเส้นทางของอาร์ไอพีก็จะมีเสถียรภาพ ถึงแม้ว่าเราเตอร์จะประกาศตารางออกไปทุกๆ 30 วินาที จึงเป็นการตรวจสอบสภาพเส้นทางและพร้อมที่จะปรับตารางเส้นทางให้เหมาะสมกับสภาพของเครือข่ายถ้าเส้นทางเดิมนั้นมีปัญหา

เมื่อพิจารณาในทางปฏิบัติแล้วเราเตอร์อาร์ไอพีใดๆ จะไม่สามารถปรับตารางเส้นทางใหม่ได้หากเราเตอร์ข้างเคียงไม่ประกาศค่าเส้นทางให้ ซึ่งอาจเป็นเพราะเราเตอร์ข้างเคียงตัวนั้นไม่สามารถทำงานได้หรือหยุดทำงานชั่วคราว อาร์ไอพีจึงหาวิธีแก้ปัญหาโดยการกำหนดให้เส้นทางทุกเส้นทางมีอายุการใช้งานปกติจะกำหนดไว้ที่ 180 วินาที ถ้าเส้นทางใดไม่ได้รับการประกาศค่าเส้นทางใดๆ เป็นเวลาตามที่กำหนดไว้ จะถือว่าเส้นทางนั้นเป็นเส้นทางที่ใช้งานไม่ได้อีกต่อไป และเราเตอร์จะเปลี่ยนเมตริกของเส้นทางนั้นให้เป็นอนันต์ แต่จะยังไม่ลบเส้นทางนั้นออกจากตาราง หากมีการประกาศเส้นทางนั้นมาจากเราเตอร์อื่น เราเตอร์จะปรับเส้นทางให้ใช้เส้นทางตามที่ได้รับจากเราเตอร์อื่นนั้น

ปัญหาการนับเข้าสู่อนันต์



รูป 5-4 การนับเข้าสู่อนันต์

การทำงานของอาร์โอพีอาจมีปัญหาเวลาที่เราเตอร์ R พบเส้นทางไปยังเครือข่ายหนึ่งผิดปกติ ในขณะเดียวกันเราเตอร์ข้างเคียงตัวหนึ่งแจ้งว่ามีเส้นทางไปยังเครือข่ายที่เกิดผิดปกติ เราเตอร์ R ที่พบปัญหาจะปรับค่าเส้นทางตามค่าที่ได้รับมาใหม่ แต่เนื่องจากเป็นเส้นทางที่เราเตอร์ข้างเคียงสามารถไปถึงเครือข่ายนั้นได้โดยผ่านทางเราเตอร์ R ทำให้เกิดการปรับเมตริกต่อไปแบบไม่มีที่สิ้นสุด ปรากฏการณ์นี้เราเรียกว่า การนับเข้าสู่อนันต์ (Count to Infinity)

ให้ตารางเส้นทางที่เราเตอร์ R2 และ R3 มีค่าดังตำแหน่ง ในจังหวะที่เราเตอร์ R1 ทำงานผิดปกติ ทำให้เครือข่าย 1 จะถูกตัดขาดออกไป ค่าเส้นทางของเครือข่าย 1 ใน R2 จะลดลงตามการทำงานของตัวจับเวลา เมื่อเมื่อ 180 วินาที เส้นทางไปเครือข่าย 1 จะเปลี่ยนเป็นอนันต์ดังตำแหน่ง

หากเวลานั้นถึงเวลาที่ R3 บรอดแคสต์ R3 จะส่งค่าเส้นทางมาให้ R2 ทำให้ R2 ได้รับเส้นทางไปเครือข่าย 1 มาด้วยเมตริก 2 ดังนั้น R2 จึงเปลี่ยนค่าในตารางเส้นทางให้เส้นทางไปเครือข่าย 1 สามารถไปโดยผ่านทาง R3 ด้วยเมตริก 3 แต่ที่จริงแล้วเครือข่าย 1 นั้นไม่สามารถใช้งานได้อยู่ในขณะนั้น R2 และ R3 ก็ไม่ทราบได้โดยตรง ทำให้เมื่อถึงเวลา R2 และ R3 ประกาศตารางเส้นทางตารางของทั้งสองจะปรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าเส้นทางเพื่อไปยังเครือข่าย 1 อย่างไม่มีที่สิ้นสุด ซึ่งอาร์ไอพีได้กำหนดค่าเมตริกไว้ค่าหนึ่งเพื่อแก้ปัญหา
นี้ ค่าเมตริกนี้ควรเป็นค่าที่ไม่สูงมากเพราะจะได้ไม่ต้องใช้เวลานาน แต่ถ้าน้อยเกินไปก็อาจทำให้เครือข่าย
ถูกจำกัดให้เล็กลง ซึ่งปกติจะกำหนดไว้ที่ 16 เมื่อเราเตอร์เพิ่มค่าไปถึง 16 ก็จะทราบว่าเส้นทางนั้นเป็น
เส้นทางที่ไม่สามารถไปถึงได้ และจะเข้าสู่กระบวนการจับเวลาเพื่อกำจัดเส้นทางออกจากตาราง

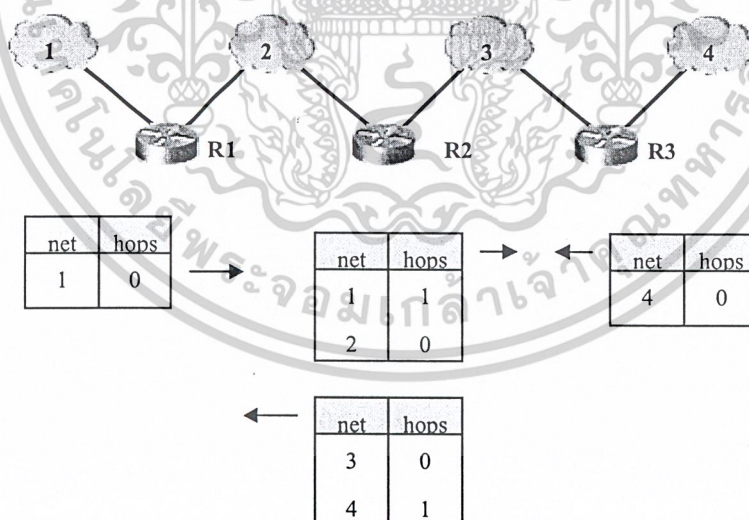
5.4.3 เวลาเข้าสู่

เวลาเข้าสู่ของอาร์ไอพีหมายถึงเวลาที่เราเตอร์ใช้ปรับตารางจนกระทั่งตารางมีค่าถูกต้องตาม
โครงสร้างของเครือข่ายจริง จากข้อกำหนดของการประกาศค่าของเราเตอร์อาร์ไอพีทุกๆ 30 วินาที ปัญหา
การนับเข้าสู่ส่วนนั้นในเครือข่ายที่ใช้อาร์ไอพีจะใช้เวลาสูงสุดประมาณ 7 นาที

หากเวลาเข้าสู่ของอาร์ไอพีสั้นลง เครือข่ายก็ย่อมที่จะเข้าสู่เสถียรภาพได้เร็วขึ้น วิธีปรับเวลาเข้าสู่
ของอาร์ไอพีให้สั้นลงมีหลายวิธี ดังนี้

5.4.4 สปลิตออไรซัน(Split Horizon)

วิธีนี้จะช่วยแก้ปัญหาการนับเข้าสู่ส่วนนั้น (เฉพาะเครือข่ายบางรูปแบบเท่านั้น) โดยเราเตอร์จะไม่
ประกาศตารางเส้นทางทั้งตารางให้กับเราเตอร์ข้างเคียงแต่จะประกาศเฉพาะเส้นทางที่ไม่ได้เรียนรู้มาจาก
เราเตอร์ตัวนั้นเท่านั้น เช่นเราเตอร์ R2 ได้รับตารางเส้นทางไปเครือข่าย 1 มาจากเราเตอร์ R1 เราเตอร์ R2
ก็จะประกาศเฉพาะเส้นทางของเครือข่าย 3 และ 4 ไปให้กับ R1 ดังรูป



รูปที่ 5-5 การประกาศเส้นทางด้วยวิธีสปลิตออไรซัน

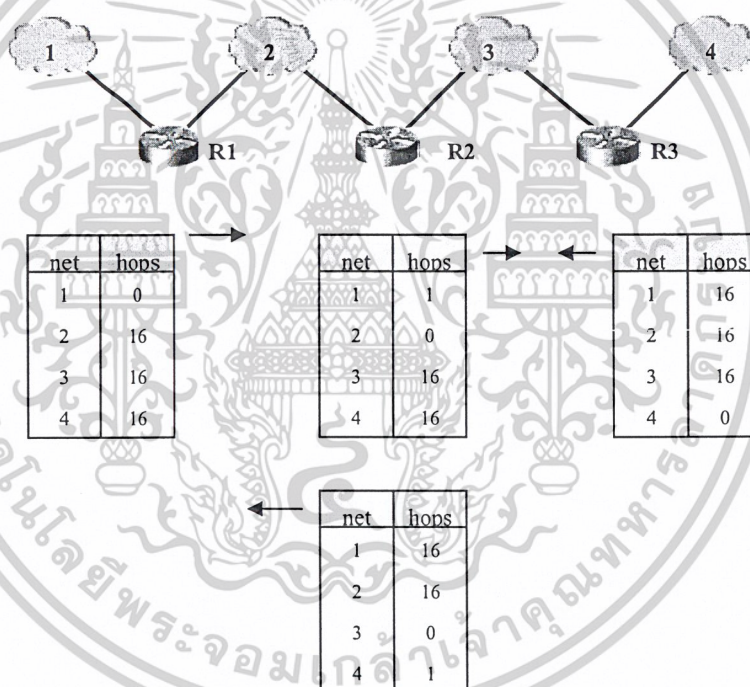
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากเราเตอร์ R1 หยุดทำงานไป ค่าเส้นทางของเครือข่าย 1 ใน R2 จะค่อยๆ ลดลงจนครบ 180 วินาที เส้นทางไปเครือข่าย 1 จะเปลี่ยนเป็นอนันต์ โดยที่ R2 จะไม่ได้รับการประกาศเส้นทางไปเครือข่าย 1 จาก R3 เนื่องจากเป็นเส้นทางที่ R2 ประกาศให้ R3 รู้วิธีไปยังเครือข่าย 1 เอง ดังนั้นจึงไม่เกิดปัญหานับเข้าสู่อนันต์

5.4.5 สปลิตฮอไรซันแบบพอยซันรีเวอร์ส (Split Horizon with Poisoned Reverse)

วิธีนี้เป็นเทคนิคอีกแบบหนึ่งของสปลิตฮอไรซันที่ใช้วิธีแจ้งค่าเส้นทางที่ชัดเจนมากยิ่งขึ้น เพื่อไม่ให้เกิดการนับเข้าสู่อนันต์

วิธีนี้จะยอมให้เราเตอร์ประกาศเส้นทางที่เรียนรู้จากลิงก์หนึ่งๆ กลับไปได้ แต่ให้เมตริกประจำเส้นทางนั้นเป็นอนันต์หรือเท่ากับ 16 เพื่อกำกับว่าเส้นทางนั้นไม่สามารถใช้งานได้ เช่น มีเราเตอร์ต่อกัน ดังรูป



รูปที่ 5-6 การประกาศเส้นทางด้วยวิธีสปลิตฮอไรซันแบบพอยซันรีเวอร์ส

หากเราเตอร์ R1 หยุดทำงาน ค่าเส้นทางของเครือข่าย 1 ใน R2 จะลดลงตามเวลาการทำงานของตัวจับเวลา และเมื่อครบ 180 วินาที เส้นทางไปเครือข่าย 1 จะเปลี่ยนเป็นอนันต์ เมื่อทั้ง R2 และ R3 ต่างประกาศค่าซึ่งแจ้งว่าระยะทางไปเครือข่าย 1 มีค่าเท่ากับ 16 ส่วนทางกันก็จะทราบได้ว่าเครือข่าย 1 ใช้งานไม่ได้ ทั้ง R2 และ R3 ก็จะกำจัดเส้นทางไปเครือข่าย 1 โดยไม่เกิดปัญหานับเข้าสู่อนันต์

วิธีนี้จะสร้างแพ็กเก็ตมากกว่าวิธีสปลิตซอไรชันแบบธรรมดา เนื่องจากมีค่าที่ต้องประกาศมากกว่า แต่ก็เป็นวิธีที่เราเตอร์ในปัจจุบันนิยมใช้เนื่องจากให้ประสิทธิภาพดีกว่า

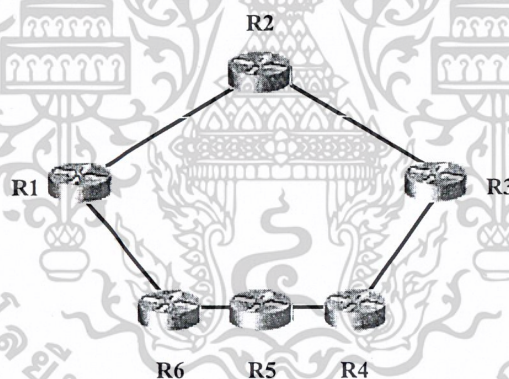
เทคนิคของสปลิตซอไรชันทั้งสองแบบนี้ไม่สามารถป้องกันปัญหานับเข้าสู่อนันต์ได้ทุกรูปแบบ เครือข่ายช่วยแก้ปัญหาสำหรับเราเตอร์สองตัวที่อยู่ข้างเคียงกันเท่านั้น

5.4.6 ทริกเกอร์อัปเดต

สปลิตซอไรชันทั้ง 2 วิธีนั้นช่วยแก้ปัญหานับเวลาเข้าสู่อนันต์เพื่อลดเวลาเข้าสู่ หากแต่ยังต้องใช้เวลาคำนวณก่อนที่จะตรวจสอบว่าเส้นทางหนึ่งๆ มีปัญหา ทริกเกอร์อัปเดตเป็นเทคนิคเสริมที่ช่วยลดเวลาเข้าสู่ของอาร์โอที คือกำหนดให้เราเตอร์ประกาศค่าเส้นทางออกไปทันทีที่พบว่าโครงสร้างเครือข่ายมีการเปลี่ยนแปลงโดยไม่ต้องรอให้ครบเวลา 30 วินาที

หากเราเตอร์ทุกตัวใช้ทริกเกอร์อัปเดตก็จะสามารถปรับปรุงตารางได้อย่างรวดเร็ว แต่ก็อาจจะสร้างภาระในเครือข่ายขนาดใหญ่ที่มีเราเตอร์หลายตัว เพราะการเปลี่ยนแปลงใดๆ ก็ย่อมทำให้เราเตอร์ตัวหนึ่งส่งข้อมูลให้เราเตอร์ตัวอื่นประกาศค่าตามกันไปอย่างต่อเนื่องหรือเกิดแพ็กเก็ตบรอดแคสต์เป็นจำนวนมาก

แต่เทคนิคนี้อาจทำให้เกิดเหตุการณ์ที่ไม่คาดฝันขึ้นได้ เช่น ในกรณีที่มีเครือข่ายดังรูป



รูปที่ 5-7 เครือข่ายที่จำเป็นต้องใช้โฮลดาวน์ร่วมกับทริกเกอร์อัปเดต

ถ้าเส้นทางที่เชื่อมระหว่าง R3 กับ R4 ไม่สามารถใช้งานได้และใช้เทคนิคทริกเกอร์อัปเดตแจ้งไปยังเราเตอร์ที่อยู่ข้างเคียง ปัญหาจะเกิดขึ้นเมื่อ R1 ได้รับการแจ้งค่าจาก R2 ก่อน R6 ทำให้ R1 ปรับตารางว่าเส้นทางที่เชื่อมระหว่าง R3 กับ R4 ใช้งานไม่ได้

ต่อมาก่อนที่ R6 จะได้รับแจ้งจาก R5 ถ้าถึงเวลาประกาศเส้นทางออกไป ทำให้ R1 ได้รับแจ้งว่ามีเส้นทางระหว่าง R3 กับ R4 ใช้งานได้ผ่านไประหว่าง R6 ทำให้ R1 ปรับตารางเส้นทางกลับไปในห้วงว่าเส้นทางระหว่าง R3 กับ R4 ใช้งานได้ ทำให้กว่าที่เราเตอร์ทุกตัวจะรู้ว่าเส้นทางระหว่าง R3 กับ R4 ไม่สามารถใช้งานได้ ก็ใช้เวลานานพอสมควร เพื่อแก้ปัญหาเรื่องนี้จึงใช้กฎเพิ่มเติมสำหรับการปรับตารางที่เรียกว่าโฮลดาวน์(Hold Down)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โสดคาว์กำหนดช่วงเวลาให้เราเตอร์ปรับเปลี่ยนค่าที่เพิ่งจะตรวจพบว่าไม่สามารถใช้งานได้ จากตัวอย่างข้างต้นนี้ เมื่อ R1 ปรับเส้นทางระหว่าง R3 กับ R4 ว่าไม่สามารถใช้งานได้ โสดคาว์จะบังคับให้ R1 รอเวลาโดยไม่รับการแจ้งค่าว่าเส้นทางระหว่าง R3 กับ R4 สามารถใช้งานได้จากเราเตอร์อื่น เป็นเวลาหนึ่ง โดยช่วงเวลานี้ควรเป็นช่วงเวลาที่มาพอจะให้ทริกเกอร์อัปเดตกระจายไปยังเราเตอร์ทุกตัว ซึ่งการใช้โสดคาว์ถึงแม้ว่าจะทำให้ช่วงเวลาลู่ว่านานกว่าแบบไม่ใช่ แต่ทำให้การทำงานของระบบดีกว่า แบบไม่ใช่ด้วย

ตัวจับเวลาในอาร์ไอพี

ตัวจับเวลาในอาร์ไอพีจะมีอยู่ 3 ประเภทคือ

1. ตัวจับเวลาปรับค่า (Update Timer) เป็นช่วงเวลาที่กำหนดการบรอดคาสต์ของเราเตอร์ ซึ่งปกติจะให้บรอดคาสต์ตารางทุกๆ 30 วินาที แต่ในทางปฏิบัติเราไม่สามารถหลีกเลี่ยงไม่ให้เกิดการบรอดคาสต์พร้อมกัน เช่น ในเราเตอร์ซิสโก้จะสุ่มค่า RIP_JITTER ซึ่งมีค่าได้ไม่เกิน 4.5 วินาทีแล้วนำมาลบกับค่า 30 ตัวจับเวลาปรับค่าของเราเตอร์ซิสโก้จึงอยู่ระหว่าง 25.5 ถึง 30 วินาที วิธีนี้จึงช่วยลดโอกาสเกิดการบรอดคาสต์แฟล็กเก็ตพร้อมกันเป็นจำนวนมากลงไปในนอกจากนี้เพื่อป้องกันการกลับมาเข้าจังหวะของเราเตอร์เมื่อเกิดทริกเกอร์อัปเดต เราเตอร์จะต้องไม่รีเซ็ตตัวจับเวลาให้กลับมาเป็นศูนย์เมื่อเกิดทริกเกอร์อัปเดต แต่ให้จับเวลาต่อไปตามปกติและบรอดคาสต์ออกไปเมื่อถึงเวลา
2. ตัวจับเวลาหมดอายุ (Expiration Timer) เป็นตัวกำหนดอายุของแต่ละเส้นทาง โดยถ้าเส้นทางใดไม่ได้รับประกาศเส้นทางนั้นมาจากเราเตอร์อื่นเป็นเวลาเท่าที่กำหนดนี้ ค่าเมตริกของเส้นทางนั้นก็จะถูกเปลี่ยนเป็น 16 เพื่อแสดงว่าไม่สามารถไปถึงได้ โดยปกติจะกำหนดเวลานี้เป็น 180 วินาที
3. ตัวจับเวลากำจัดเส้นทาง (Garbage Collection Timer) สำหรับเส้นทางที่หมดอายุจากการจับเวลา 180 วินาทีข้างต้นแล้วจะยังไม่ถูกลบออกจากตารางทันที แต่จะนับเวลาถอยหลังเป็นจำนวนเวลาตามที่กำหนดนี้เพื่อลบเส้นทางนี้ออกไปจากตาราง แต่ในระหว่างที่นับเวลาถอยหลังนี้ก็จะบรอดคาสต์เส้นทางนี้ออกไปด้วย เพื่อให้เราเตอร์ตัวอื่นนับเวลาถอยหลังเพื่อลบเส้นทางนี้ด้วย

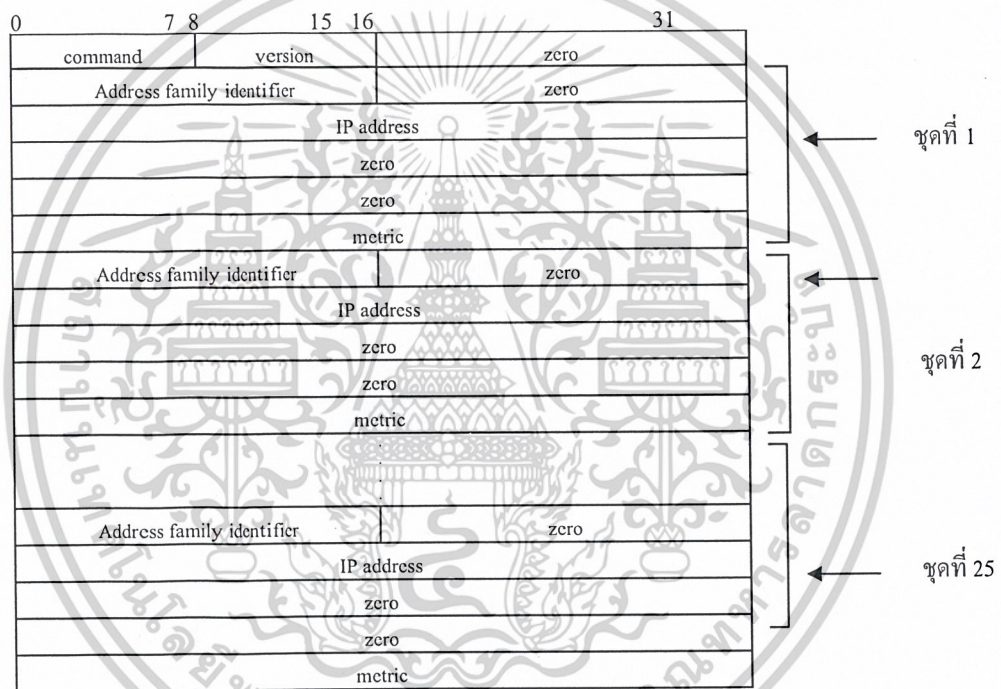
5.4.7 เฟรมอาร์ไอพี

เฟรมอาร์ไอพีจะนำส่งโดยบรรจุอยู่ในยูนิคิพี ลักษณะของเฟรมอาร์ไอพีจะเป็นดังรูปแบบเฟรมในทีนี้แสดงเฉพาะการใช้อาร์ไอพีในเครือข่ายไอพีเท่านั้น หากเป็นเครือข่ายอื่นจะมีรูปแบบแตกต่างออกไป โดยแต่ละฟิลด์มีความหมายดังนี้

- command ขนาด 8 บิต : กำหนดแบบการทำงานว่าเป็นการร้องขอ (ค่าเท่ากับ 1) หรือการตอบรับ (ค่าเท่ากับ 2) การบรอดคาสต์ของอาร์ไอพีจัดเป็นการตอบรับ (เพราะไม่มีการร้องขอมาก่อน) อาร์ไอพีจะกำหนดคำสั่งให้เป็น 1 เพื่อร้องขอต่อเมื่อต้องการทราบตารางเส้นทางของเราเตอร์ใดเป็นพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- version ขนาด 8 บิต : กำหนดรุ่นของโปรโตคอล หากเท่ากับ 1 หมายถึงรุ่น 1
- zero ขนาด 16 บิต : สงวนไว้และต้องมีค่าเป็น 0
- address family identifier ขนาด 16 บิต : การออกแบบอาร์ไอพีในขั้นต้นไม่ได้กำหนดให้ใช้เฉพาะกับทีซีพี/ไอพีเท่านั้น จึงมีฟิลด์กำกับชุดแอดเดรสเพื่อแยกแยะชนิดโปรโตคอลที่อาร์ไอพีทำงานด้วย ค่านี้เท่ากับ 2 สำหรับไอพี
- IP address ขนาด 32 บิต : ไอพีแอดเดรสของเครือข่ายที่ประกาศค่าออกไป
- Metric ขนาด 32 บิต : เมตริกกำหนดระยะทางให้ใช้ค่าได้ตั้งแต่ 1 ถึง 15 หากมีค่าเป็น 16 หมายถึงเครือข่ายปลายทางไม่สามารถไปถึงได้



รูปที่ 5-8 ฟอรัมเมตของเฟรมอาร์ไอพี

ตั้งแต่ฟิลด์ address family identifier กระทั่งถึงเมตริกสามารถมีค่าซ้ำได้ 25 ชุด เพื่อให้เราเตอร์สามารถบรอดคาสต์เส้นทางได้ 25 ค่าในดาต้าแกรมเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.8 เฟรมอาร์ไอพี 2

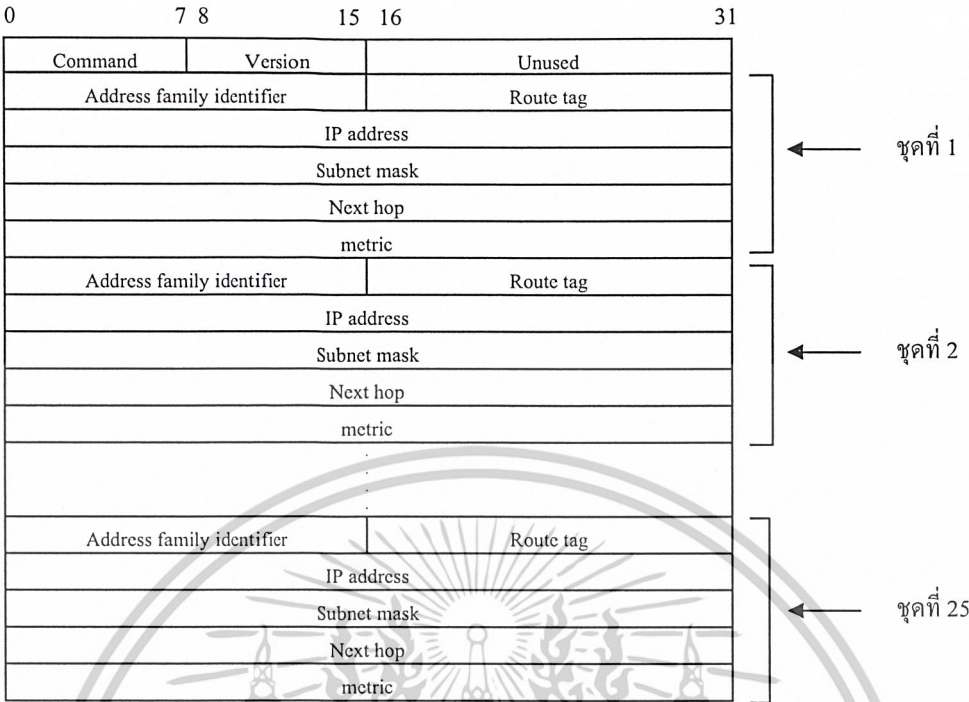
อาร์ไอพีรุ่น 2 มีส่วนขยายการทำงานเพิ่มเติมจากอาร์ไอพีรุ่นแรกและเพิ่มระบบความปลอดภัยในการส่งข้อมูลเส้นทางระหว่างเราเตอร์ อาร์ไอพีรุ่น 1 ไม่สนับสนุนการใช้งานในเครือข่ายที่มีซับเน็ตแบบแปรค่า ไม่สามารถแจ้งหมายเลขระบบอโตโนมัสเพื่อเชื่อมความปลอดภัยในการส่งตาราง อาร์ไอพีรุ่น 2 จึงได้รับการพัฒนาขึ้นเพื่อรักษาความปลอดภัยในการส่งตาราง อาร์ไอพีรุ่น 2 จึงได้รับการพัฒนาขึ้นเพื่อแก้ปัญหาดังกล่าวและกลายเป็นมาตรฐานใหม่ทดแทนอาร์ไอพีรุ่น 1

ฟิลด์อาร์ไอพีรุ่น 2 ยังคงลักษณะของอาร์ไอพีรุ่น 1 แต่ได้เพิ่มเติมฟิลด์ใหม่ และมีรูปแบบดังรูปที่

5-10

ความหมายของแต่ละฟิลด์

- Command ขนาด 8 บิต : กำหนดแบบการทำงานว่าเป็นการร้องขอ (ค่าเท่ากับ 1) หรือเป็นการตอบรับ (ค่าเท่ากับ 2)
- Version ขนาด 16 บิต : กำหนดรุ่นของโปรโตคอล ค่าที่ใช้คือ 2
- Unused ขนาด 16 บิต : ไม่ได้ใช้งาน
- Address family identifier ขนาด 16 บิต : ฟิลด์กำกับชุดแอดเดรสเพื่อแยกแยะชนิดของโปรโตคอลที่อาร์ไอพีทำงานร่วมกับ สำหรับไอพีใช้ค่าเท่ากับ 2
- Route tag ขนาด 8 บิต : ใช้แยกความแตกต่างระหว่างการได้มาของค่าเส้นทางว่าเรียนรู้จากภายในหรือภายนอก ค่าที่ใช้จะเป็นเลขระบบอโตโนมัสสำหรับเส้นทางที่ได้มาจากโปรโตคอลภายนอก
- IP Address ขนาด 32 บิต : ไอพีแอดเดรสที่กำหนดเส้นทาง
- Subnet Mask ขนาด 32 บิต : ซับเน็ตมาสก์กำกับไอพีแอดเดรส หากมีค่าเป็น 0 หมายถึงไม่ได้กำหนดซับเน็ตมาสก์ให้
- Next Hop ขนาด 32 บิต : ไอพีแอดเดรสของเราเตอร์ถัดไปที่ควรจะนำส่งแพ็กเก็ตซึ่งมีแอดเดรสปลายทางตามที่กำหนดด้วยค่าเลือกเส้นทางในเฟรมอาร์ไอพีนี้ ประโยชน์ของฟิลด์นี้ใช้เพื่อลดจำนวนเส้นทางที่แพ็กเก็ตต้องเดินทางผ่านในกรณีที่มิโปรโตคอลเลือกเส้นทางอื่นนอกเหนือไปจากอาร์ไอพีทำงานร่วมกัน หากมีค่า 0.0.0.0 หมายถึงให้ส่งแพ็กเก็ตไปยังเราเตอร์ที่ประกาศค่าเส้นทางนี้มาให้
- Metric ขนาด 32 บิต : เมตริกกำหนดระยะทางยังคงมีค่าได้ตั้งแต่ 1 ถึง 15 เช่นเดียวกับอาร์ไอพีรุ่น 1

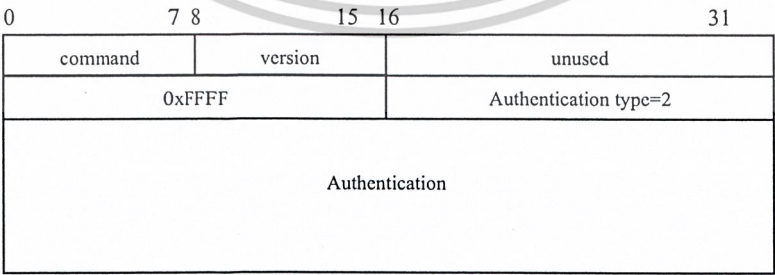


รูปที่ 5-9 ฟอรัมของเฟรมอาร์พีไอพีเวอร์ชัน 2

5.4.9 การพิสูจน์ตัวตนจริง

ตั้งแต่ฟิลด์ address family identifier ถึงฟิลด์ metric มีค่าซ้ำกันได้ 25 ชุดเช่นเดียวกับเวอร์ชันแรกเพื่อการประกาศเส้นทางไปพร้อมกันได้ 25 เส้นทางในเฟรมเดียว แต่ถ้าฟิลด์ address family identifier ชุดแรกมีค่าเป็น 0xFFFF (ให้ค่านี้มีได้เฉพาะในชุดแรกเท่านั้น) หมายความว่าเฟรมกำหนดการพิสูจน์ตัวตนจริง และฟิลด์ถัดมาจะถูกแปลความหมายใหม่ตามรูป 5-10

มีฟิลด์ authentication type ขนาด 16 บิตอยู่ต่อจาก address family identifier และถัดไปอีก 128 บิตเป็นฟิลด์ authentication สำหรับใช้ตรวจสอบว่าเราเตอร์ที่ส่งเฟรมนี้เป็นเราเตอร์ตัวจริงที่ได้รับอนุญาต



รูปที่ 5-10 เฟรมอาร์พีไอพีเวอร์ชัน 2 เมื่อใช้การพิสูจน์ตัวตนจริง

ในปัจจุบันกำหนดให้ฟิลด์ authentication type มีค่าเท่ากับ 2 ซึ่งหมายถึงการพิสูจน์ด้วยรหัสผ่าน(ชนิดที่ไม่เข้ารหัสลับ) และค่ารหัสนั้นจะต้องส่งมาในฟิลด์ authentication ด้วยเหตุนี้เฟรมอาร์ไอพีที่ใช้การพิสูจน์ตัวจริงจะมีจำนวนเส้นทางที่บรอดคาสต์พร้อมกันในหนึ่งเฟรมได้เพียง 24 เส้นทาง

5.5 โอเอสพีเอฟ (OSPF : Open Shortest Path First)

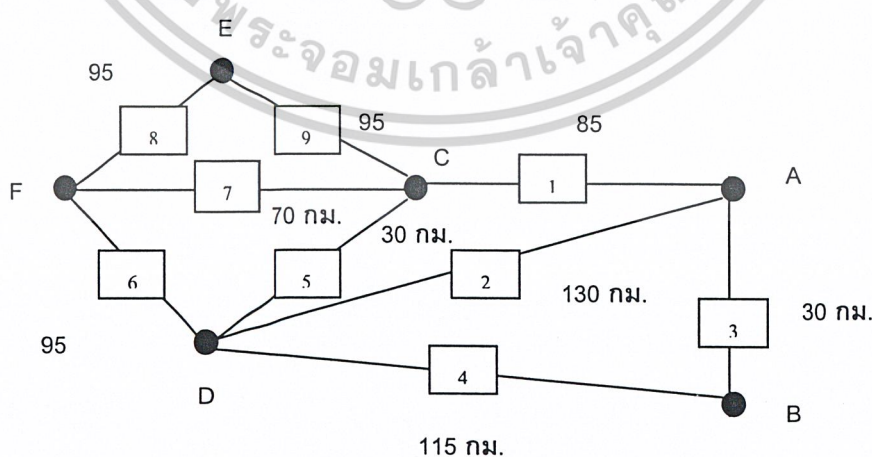
โอเอสพีเอฟเป็นโพรโทคอลเกตเวย์ภายในที่นิยมใช้อย่างแพร่หลายเนื่องจากสามารถปรับเปลี่ยนการเลือกเส้นทางตามสภาพเครือข่ายได้รวดเร็วและรองรับการทำงานกับเครือข่ายขนาดใหญ่ได้

5.5.1 หลักการพื้นฐานของโพรโทคอลแบบโอเอสพีเอฟ

เป็นโพรโทคอลแบบลิงค์สเตต โดยโอเอสพีเอฟจะทำหน้าที่สร้างแผนที่ให้กับเราเตอร์ 3 ขั้นตอนคือ ขั้นแรกเราเตอร์จะตรวจหาเราเตอร์อื่นที่อยู่ข้างเคียง ขั้นที่สองเราเตอร์จะประกาศค่านี้ออกไปให้เราเตอร์อื่นทุกตัว เมื่อเสร็จสิ้นขั้นตอนนี้ เราเตอร์จะทราบว่ามีเราเตอร์อื่นใดอยู่บ้าง ขั้นสุดท้ายเราเตอร์จะอาศัยข้อมูลจากขั้นที่สองสร้างแผนที่ไปยังเราเตอร์อื่นทุกตัวในเครือข่าย เราเตอร์ทุกตัวต่างก็มีแผนที่ประจำตัวที่ทราบถึงเส้นทางทั้งหมดในเครือข่าย

5.5.1.1 การทำงานของโอเอสพีเอฟโดยสรุป

เพื่อให้สามารถศึกษาการทำงานของโอเอสพีเอฟได้อย่างชัดเจนเป็นขั้นตอนจะใช้เส้นทางจากเมืองหนึ่งไปยังอีกเมืองหนึ่งดังรูป 6.1 แต่ละเมืองเปรียบเสมือนเราเตอร์และทางหลวงที่เชื่อมต่อกันถือเป็นสายสื่อสารระหว่างเราเตอร์ เส้นทางในรูปเป็นเพียงเส้นทางที่จำลองจากทางหลวงระหว่างเมืองและระยะทางวัดเป็นกิโลเมตร โดยประมาณ หมายเลขประจำเส้นทางตั้งแต่ 1-9 ใช้เรียกชื่อแทนหมายเลขทางหลวงเพื่อความสะดวกต่อการอ้างอิง

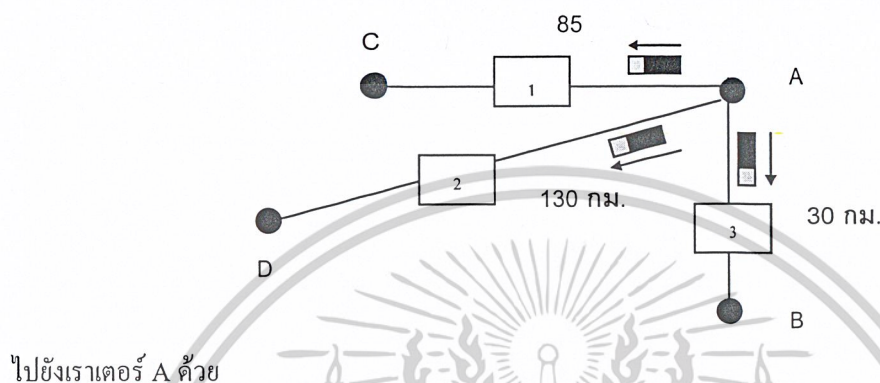


รูปที่ 5-11 แผนที่ทางหลวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรียนรู้เครือข่าย

เราเตอร์ในเครือข่ายที่ใช้โอเอสทีเอฟจะเริ่มต้นเรียนรู้ว่ามีเราเตอร์ใดที่อยู่ข้างเคียงโดยการส่งแพ็กเก็ต “ทักทาย” หรือ ฮัลโลแพ็กเก็ต (Hello Packet) ออกไปทุกๆ ลิงค์ของเราเตอร์ ตัวอย่างรูปที่ 6-2 เราเตอร์ A จะส่งแพ็กเก็ตทักทายไปยังเราเตอร์ที่ B, C และ D ในขณะที่เราเตอร์ทั้งสามก็ส่งแพ็กเก็ตทักทาย



รูปที่ 5-12 เราเตอร์ A ส่งแพ็กเก็ตทักทาย

เมื่อเราเตอร์ A ได้รับแพ็กเก็ตตอบกลับจาก B, C และ D เราเตอร์ A ก็จะทราบถึงเส้นทางข้างเคียงโดยตรง เราเตอร์ A จะสร้างตารางระยะทางไปยังเราเตอร์ข้างเคียงตามตารางที่ 6.1 ในขณะเดียวกันและในที่สุดเราเตอร์แต่ละจังหวัดต่างก็จะมีข้อมูลระยะทางไปยังเราเตอร์ข้างเคียงทุกตัว

เราเตอร์ข้างเคียง	เส้นทาง	ระยะทาง
B	3	30 กม.
C	1	85 กม.
D	2	130 กม.

ตารางที่ 5-1 ตารางเส้นทางที่เราเตอร์ A

สร้างฐานข้อมูลเส้นทาง

ขั้นตอนที่สองนี้แต่ละเราเตอร์จะสร้างฐานข้อมูลที่เก็บเส้นทางไปยังเราเตอร์ทุกตัวในเครือข่าย วิธีการที่ใช้คือเราเตอร์สร้างแพ็กเก็ตบรรจุเส้นทางที่มีอยู่ทั้งหมดในขณะนั้นและ “ปล่อย” (Flooding) ให้แพ็กเก็ตไหลจากเราเตอร์หนึ่งไปยังอีกเราเตอร์หนึ่ง

เมื่อเราเตอร์ได้รับแพ็กเก็ตก็จะสำเนาแพ็กเก็ตนั้นส่งออกไปยังลิงค์อื่นด้วย เราเตอร์ต้องตรวจสอบว่าเป็นแพ็กเก็ตเก่าซ้ำของเดิมหรือเป็นแพ็กเก็ตใหม่ หากได้รับแพ็กเก็ตเก่าจากเราเตอร์อื่นให้กำจัดทิ้งไป หากเป็นแพ็กเก็ตใหม่จะต้องปล่อยแพ็กเก็ตไปทุกอินเทอร์เฟซยกเว้นอินเทอร์เฟซที่ได้รับแพ็กเก็ต

รูปแบบของแพ็กเก็ตที่เราเตอร์ส่งออกไปในขั้นนี้เรียกว่า แพ็กเก็ตประกาศลิงก์สเตต (Link State Advertisement) หรือ แพ็กเก็ตแอลเอสเอ (LSA) เมื่อเราเตอร์ทุกตัวปล่อยแพ็กเก็ตแอลเอสเอเช่นเดียวกันเพียงชั่วระยะหนึ่งแพ็กเก็ตจะกระจายไปทั่วทั้งเครือข่าย ท้ายที่สุดแล้วเราเตอร์ทุกตัวจะทราบเส้นทางไปยังเราเตอร์อื่นและได้ตารางเส้นทางเรียกว่า ฐานข้อมูลลิงก์สเตต (Link State Database) จากตัวอย่างนี้ ฐานข้อมูลลิงก์สเตตมีค่าตามตารางที่ 6.2

Router A	Router B	Router C	Router D	Router E	Router F
B : 30	A : 30	A : 85	A : 130	C : 110	C : 70
C : 85	D : 115	D : 30	B : 115	F : 95	D : 95
D : 130		E : 110	C : 30		E : 95
		F : 70	F : 95		

ตารางที่ 5-2 ฐานข้อมูลลิงก์สเตตประจำแต่ละเราเตอร์

การคำนวณเส้นทาง

ฐานข้อมูลที่ได้จากขั้นที่แล้วยังไม่ใช้ตารางเส้นทางที่นำไปใช้ได้ ตารางเส้นทางจะได้จากการคำนวณหาเส้นทางที่สั้นที่สุดตามขั้นตอนวิธีของไดจ์สตรา (Dijkstra Algorithm)

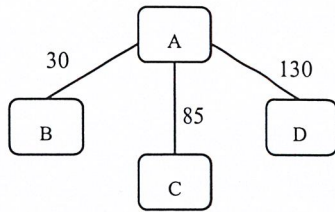
5.5.1.2 ขั้นตอนวิธีของไดจ์สตรา (Dijkstra Algorithm)

ตามขั้นตอนวิธีไดจ์สตราเราเตอร์จะสร้างเส้นทางที่เขียนแทนด้วยต้นไม้กำกับทิศทางโดยมีกิ่งเชื่อมไปยังเราเตอร์อื่น รากของต้นไม้ก็คือเราเตอร์ที่เป็นจุดตั้งต้นคำนวณหาเส้นทาง แต่ละเส้นจะกำกับด้วยเลเบลซึ่งแบ่งเส้นทางออกเป็นสองชนิดคือ เส้นทางชั่วคราว และ เส้นทางถาวร เส้นทางชั่วคราวใช้กำกับเส้นทางที่ค้นพบแต่ยังไม่ผ่านการคำนวณว่าเป็นเส้นทางที่สั้นที่สุด ส่วนเส้นทางถาวรใช้กำกับเส้นทางที่คำนวณแล้วว่าสั้นที่สุด

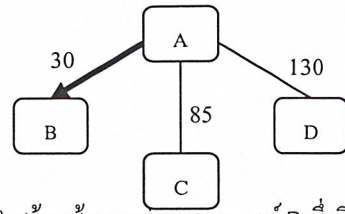
เมื่อเริ่มต้นทำงานที่เราเตอร์ใด เราเตอร์นั้นจะมีระยะทางถึงตัวเองเป็นศูนย์และสร้างเส้นทางชั่วคราวไปยังเราเตอร์ข้างเคียงก่อน ส่วนเส้นทางไปยังเราเตอร์อื่นให้มีค่าเป็นอนันต์

เมื่อใช้ขั้นตอนวิธีของไดจ์สตรากับรูปที่ 6-1 โดยพิจารณาเราเตอร์ A จะได้เส้นทางแรกตามรูปที่ 6-3 (1) โดยมีเส้นทางไปยัง B, C และ D ทั้งสามเส้นทางไม่มีหัวลูกศรซึ่งหมายถึงเส้นทางชั่วคราว ลำดับขั้นที่ (2) ถึง (6) แสดงการหาเส้นทางที่สั้นที่สุดไปยังที่อื่นๆ โดยเส้นทางที่มีหัวลูกศรหมายถึงเส้นทางถาวร

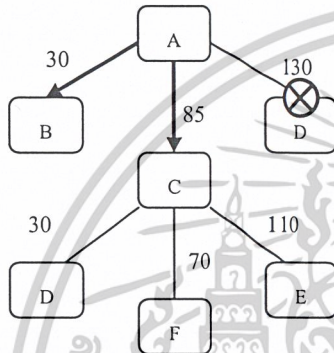
เมื่อเสร็จสิ้นขั้นตอนสุดท้ายแล้วเราเตอร์ A จะมีแผนที่ทั้งเครือข่าย เมื่อต้องการไปยังเส้นทางใดก็เพียงแต่หาเส้นทางจากต้นไม้ที่สร้างขึ้นเท่านั้น ในขณะที่เราเตอร์อื่นก็จะดำเนินการตามขั้นตอนเพื่อหาระยะทางที่สั้นที่สุดไปยังที่อื่นๆ ในลักษณะเดียวกัน



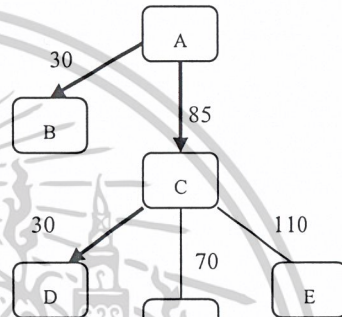
(1) เราเตอร์ A สร้างต้นไม้แสดงเส้นทางไปยังเราเตอร์ข้างเคียง



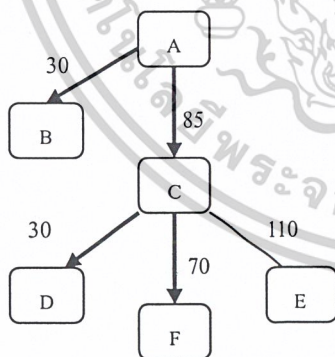
(2) สร้างเส้นทางต่อจากเราเตอร์ B ซึ่งมีเพียงเราเตอร์ D เท่านั้นแต่ระยะทางจาก A ไป D โดยตรงมีค่าเท่ากับ 130 ซึ่งน้อยกว่าการไป D โดยผ่าน B ก่อนซึ่งจะมีค่าเท่ากับ $115+30 = 145$ ดังนั้นให้เส้นทางไป B เป็นเส้นทางถาวร



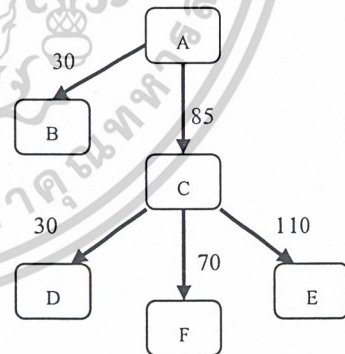
(3) สร้างเส้นทางต่อจากเราเตอร์ C แล้วพบว่า เส้นทางไป D ผ่านทาง C ($85+30=115$) สั้นกว่าไป D (130) โดยตรง ดังนั้นให้เส้นทางไป C เป็นเส้นทางถาวร และยกเลิกเส้นทางไป D



(4) จากเราเตอร์ D ไม่พบเส้นทางไปเราเตอร์ใดที่สั้นกว่า จึงให้เส้นทางที่ D เป็นเส้นทางถาวร



(5) จากเราเตอร์ F ไม่พบเส้นทางไปเราเตอร์ใดที่สั้นกว่า จึงให้เส้นทางที่ F เป็นเส้นทางถาวร



(6) จากเราเตอร์ E ไม่พบเส้นทางไปเราเตอร์ใดที่สั้นกว่า จึงให้เส้นทางที่ E เป็นเส้นทางถาวร

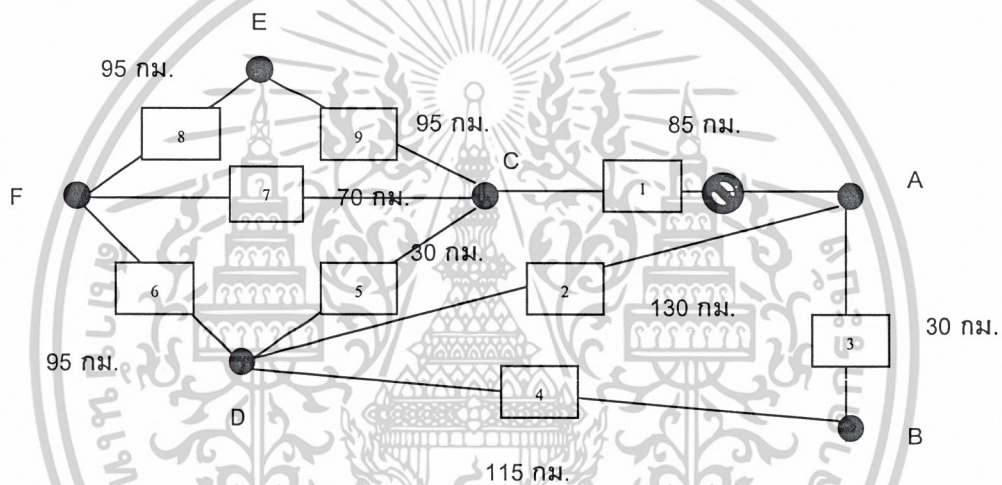
รูปที่ 5-13 การคำนวณหาเส้นทางสั้นที่สุดตามขั้นตอนวิธีของไดจ์สตรา

5.5.1.3 การปรับเปลี่ยนเมื่อเครือข่ายเปลี่ยน

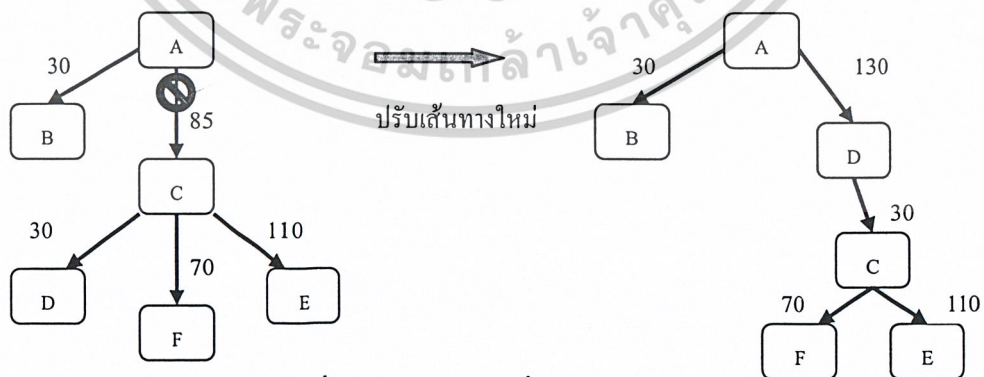
การทำงานของโอเอฟพีเอฟซึ่งประกอบด้วย 3 ขั้นตอนที่กล่าวไปแล้วนั้น ในทางปฏิบัติจะเกิดขึ้นแบบขนานพร้อมกันไปในเรเตอร์แต่ละตัว เรเตอร์จะส่งและได้รับฮัลโหลแพ็กเก็ตกับเรเตอร์ข้างเคียงอยู่ตลอดเวลา หากสายสื่อสารเกิดผิดปกติทำให้เรเตอร์ไม่ได้รับฮัลโหลแพ็กเก็ตเกิดภายในระยะเวลาที่กำหนดก็จะทราบว่าเป็นเส้นทางไปไม่ถึงและจะเริ่มคำนวณหาเส้นทางใหม่

ตัวอย่างเช่นเส้นทางจากเรเตอร์ A ไปยังเรเตอร์ C มีปัญหาขัดข้องดังรูปที่ 6-4 เรเตอร์ A และ C ต่างไม่ได้รับฮัลโหลแพ็กเก็ตซึ่งกันและกัน เมื่อถึงกำหนดเวลาต่างก็ทราบว่าเส้นทางไม่สามารถใช้งานได้และต้องเปลี่ยนฐานข้อมูลลิงก์สเตทใหม่

หากพิจารณาเฉพาะที่เรเตอร์ A ซึ่งจะคำนวณหาเส้นทางใหม่ กรณีเรเตอร์จะใช้เส้นทางไปยัง D เพื่อไป C โดยเส้นทางอื่นยังคงเดิม ดังนั้นไม่แสดงเส้นทางสั้นที่สุดจะมีลักษณะดังรูปที่ 6-5



รูปที่ 5-14 ปัญหาเส้นทางขัดข้อง



รูปที่ 5-15 เส้นทางใหม่ที่คำนวณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

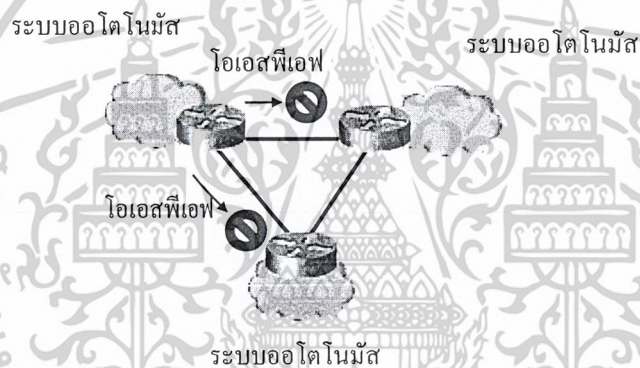
ขั้นตอนวิธีของไดจ์สตราเป็นขั้นตอนวิธีที่ต้องใช้กำลังการคำนวณของซีพียูอย่างมากโดยเฉพาะอย่างยิ่งในเครือข่ายขนาดใหญ่ เราเตอร์จะต้องคำนวณหาต้นไม้ที่แทนเส้นทางสั้นที่สุดใหม่ทุกครั้งที่มีการเปลี่ยนแปลงใดๆในเครือข่าย

แต่โอเอสพีเอฟมีวิธีการจัดรูปแบบเครือข่ายเพื่อลดการคำนวณเมื่อเครือข่ายเปลี่ยนแปลงซึ่งจะกล่าวในหัวข้อถัดไป

5.5.2 การจัดองค์ประกอบเครือข่าย

ระบบบอโตโนมัสเป็นตัวกำหนดขอบเขตการทำงานของโอเอสพีเอฟ ข้อมูลเส้นทางโอเอสพีเอฟจะแลกเปลี่ยนระหว่างเราเตอร์ที่อยู่ภายในบอโตโนมัสหนึ่งๆ เท่านั้น

ในระบบบอโตโนมัสหนึ่งๆ ย่อมมีเราเตอร์เชื่อมต่อไปยังระบบอื่นดังตัวอย่างในรูปที่ 6-6 เราเตอร์ในระบบบอโตโนมัสที่เชื่อมกับระบบบอโตโนมัสอื่นเรียกว่า เราเตอร์ขอบปลายระบบบอโตโนมัส (AS boundary routers) และเป็นตัวกันไม่ให้ข้อมูลโอเอสพีเอฟออกไปสู่ระบบบอโตโนมัสอื่น



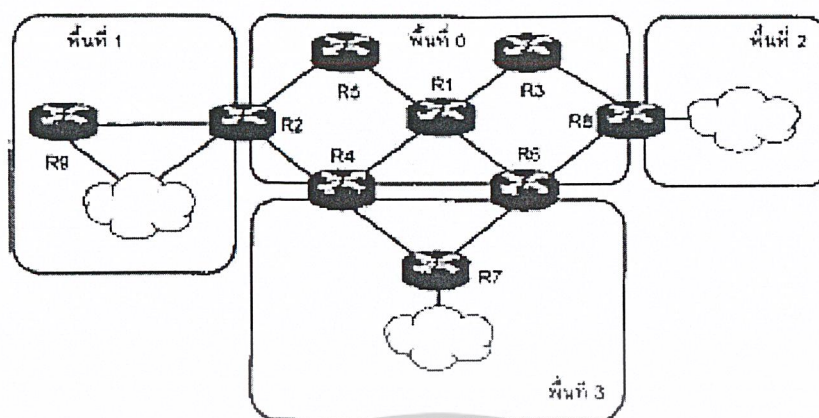
รูปที่ 5-16 เราเตอร์ขอบปลายของระบบบอโตโนมัส

5.5.2.1 การแบ่งระบบบอโตโนมัสออกเป็นพื้นที่

เครือข่ายที่ใช้งานโอเอสพีเอฟย่อมอยู่ในระบบบอโตโนมัสเดียวกัน และอาจเป็นได้ทั้งเครือข่ายขนาดเล็กไปถึงเครือข่ายขนาดใหญ่ โอเอสพีเอฟสามารถทำงานได้กับเครือข่ายขนาดใหญ่และยังรองรับการขยายขนาดขึ้นไปอีก เพราะมีคุณสมบัติการทำงานที่รองรับการขยายอยู่ในตัว

คุณสมบัติหนึ่งได้แก่การแบ่งเครือข่ายออกเป็นกลุ่มย่อยเพื่อจำกัดขอบเขตการทำงานบางอย่างให้อยู่ในบริเวณใดบริเวณหนึ่งหรือกลุ่มของเราเตอร์กลุ่มหนึ่งๆ ขอบเขตของเครือข่ายที่จัดแบ่งแล้วเรียกว่า พื้นที่ (Area)

พื้นที่ในความหมายของโอเอสพีเอฟคือกลุ่มของโฮสต์หรือเราเตอร์ภายใต้ขอบเขตที่กำหนดเครือข่ายหนึ่งๆ อาจแบ่งออกเป็นหลายพื้นที่ แต่ละพื้นที่จะเชื่อมโยงกับพื้นที่กลางที่เรียกว่า พื้นที่แบ็คโบน (Backbone Area) รูปที่ 6-7 แสดงเครือข่ายที่แบ่งออกเป็น 4 ส่วนแต่ละพื้นที่จะมีหมายเลขประจำพื้นที่ (Area ID) โดยแบ็คโบนจะมีหมายเลขเท่ากับ 0



รูปที่ 5-17 ตัวอย่างการจัดพื้นที่ในระบบออตโนมัสของโอเอสพีเอฟ

เราเตอร์บางตัวอาจสังกัดอยู่ที่พื้นที่เพียงพื้นที่เดียว ในขณะที่เราเตอร์บางตัวอาจเป็นตัวเชื่อมระหว่างพื้นที่เข้าด้วยกันและเรียกว่า เราเตอร์ขอบปลาย (area border router) ในรูปนี้ เราเตอร์ R2, R4, R6 และ R8 เป็นเราเตอร์ของปลายพื้นที่ เราเตอร์อื่นที่ไม่ได้เป็นเราเตอร์ขอบปลายพื้นที่ เรียกว่าเป็น เราเตอร์ภายใน (internal router) เช่น R1, R3, R5, R7 และ R9 จัดเป็น เราเตอร์ภายใน หน้าที่ของเราเตอร์ภายในคือเก็บตารางเส้นทางทั้งหมดภายในพื้นที่ซึ่งสังกัดอยู่และตารางเส้นทางสรุปของพื้นที่อื่นๆ

เราเตอร์ขอบปลายพื้นที่ไม่ได้แลกเปลี่ยนแพ็กเก็ตถึงคลัสเตอร์ตามปกติ หากแต่ใช้แพ็กเก็ตรูปแบบพิเศษที่สรุปรวมเกี่ยวกับเส้นทางทั้งหมดภายในพื้นที่นั้นเพื่อแลกเปลี่ยนกับเราเตอร์ขอบปลายอื่น การทำงานของโอเอสพีเอฟภายในพื้นที่ใดๆยังคงหลักการตามที่ได้กล่าวไว้แล้วคือ เราเตอร์ สร้างและปล่อยแพ็กเก็ตถึงคลัสเตอร์ไปยังเราเตอร์อื่นในพื้นที่เดียวกัน เราเตอร์ที่อยู่ในพื้นที่หนึ่งจะทราบถึงโทโปโลยีของเครือข่ายเฉพาะในพื้นที่ที่สังกัดอยู่เท่านั้น

5.5.2.2 แบ็คโบน

แบ็คโบนเป็นพื้นที่พิเศษในระบบออตโนมัสซึ่งทำหน้าที่เป็นเสมือนศูนย์กลางของระบบ พื้นที่อื่นจะต้องมีจุดเชื่อมโยงเข้ากับแบ็คโบนเสมอ และอาจมีจุดเชื่อมโยงเข้ากับพื้นที่อื่นด้วยก็ได้

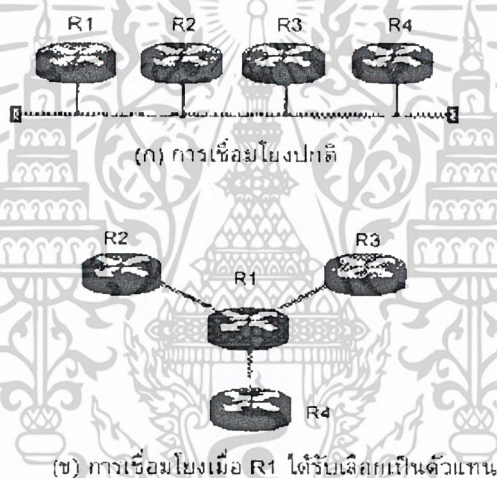
เราเตอร์ที่อยู่ในแบ็คโบนจัดเป็น แบ็คโบนเราเตอร์ (backbone router) รูปที่ 6-7 ประกอบด้วย แบ็คโบนเราเตอร์ 7 ตัว ในจำนวนนี้ 4 ตัวทำหน้าที่เป็นเราเตอร์ขอบปลายด้วย คือ R2, R4, R5 และ R8 ที่เหลืออีก 3 ตัวเป็นแบ็คโบนเราเตอร์เพียงอย่างเดียว

5.5.3 เราเตอร์ตัวแทน

เราเตอร์โอเอสพีเอฟใช้ โปรโตคอลฮัลโล (hello protocol) เพื่อตรวจหาเราเตอร์ข้างเคียง โปรโตคอลฮัลโลจะเลือกเราเตอร์หนึ่งตัวขึ้นมาทำหน้าที่เป็นตัวแทนประกาศว่าลิงสเทตให้กับเราเตอร์อื่นในเครือข่าย เราเตอร์ที่ได้รับเลือกนี้เรียกว่า เราเตอร์ตัวแทน (designate router) ประโยชน์การใช้เราเตอร์ตัวแทนคือช่วยลดปริมาณข้อมูลที่ต้องส่งระหว่างเราเตอร์ด้วยกันเอง

ขอให้อธิบายการเชื่อมต่อเครือข่ายแบบบรอดคาสต์อย่างเช่นอีเทอร์เน็ต ในรูปที่ 6-8 (ก) ซึ่งมีเราเตอร์ต่อเชื่อมกัน 4 ตัว เราเตอร์ตัวหนึ่งๆต้องมีเราเตอร์ข้างเคียงอยู่ 3 ตัว และส่งลิงค์สเทตระหว่างกันทุกตัว แต่เมื่อใช้เราเตอร์ตัวแทนจะเกิดโทโปโลยีเทียมให้เราเตอร์ทุกตัวเชื่อมไปยังเราเตอร์ตัวแทนตามรูปที่ 6-8 (ข) จำนวนเราเตอร์ข้างเคียงซึ่งเดิมอยู่ 3 ตัวจะลดเหลือเพียงตัวเดียว คือเราเตอร์ตัวแทนเท่านั้น ปริมาณการส่งลิงค์สเทตลดลงจากเอกซ์โพเนนเชียลมาเป็นเชิงเส้นตามจำนวนเราเตอร์ที่มีอยู่

การทำงานของโอเอสพีเอฟยังคงเป็นไปตามขั้นตอนที่กล่าวในหัวข้อ 6.1.1 ยกเว้น เพียงแต่เราเตอร์แต่ละตัวจะไม่ส่งค่าลิงค์สเทตที่สร้างขึ้นระหว่างเราเตอร์ด้วยกันเอง แต่จะส่งกับเราเตอร์แทนเท่านั้น



รูปที่ 5-18 การใช้เราเตอร์ตัวแทน

5.5.3.1 มัลติคาสต์แอดเดรสของเราเตอร์โอเอสพีเอฟ

ตามที่ได้อธิบายไว้แล้วว่าเราเตอร์ใช้วิธีปล่อยลิงค์สเทตออกไป แต่เพื่อลดปริมาณการส่งค่าลิงค์สเทตลงไปอีก เราเตอร์ตัวแทนจะไม่ใช้วิธีการบรอดคาสต์ค่าลิงค์สเทต หากแต่จะใช้วิธีการมัลติคาสต์แทน โอเอสพีเอฟเราเตอร์ทุกตัวกำหนดให้มีแอดเดรสมัลติคาสต์ 224.0.0.5 เป็นแอดเดรสประจำตัวเพื่อรับลิงค์สเทตจากเราเตอร์ตัวแทน แอดเดรสนี้เรียกว่า “AllSPFRouter” โปรโตคอลฮัลโลจะใช้แอดเดรสนี้ในการทำงานเช่นกัน

เราเตอร์ตัวแทนจะมีแอดเดรสมัลติคาสต์ประจำตัวต่างหากอีกหนึ่งแอดเดรสคือ 224.0.0.6 โดยเรียกแอดเดรสนี้ว่า “AllDRouters” เราเตอร์อื่นจะใช้แอดเดรสนี้เพื่อส่งลิงค์สเทตไปยังเราเตอร์ตัวแทน

เราเตอร์ตัวแทนซึ่งเป็นเสมือนศูนย์กลางการรับส่งลิงค์สเทตจะเป็นจุดประจําของเครือข่ายอยู่ในตัว เพื่อรับประกันการทำงานให้มีความน่าเชื่อถือมากขึ้น โปรโตคอลฮัลโลจะเลือก เราเตอร์ตัวแทน

สำรอง (backup designated router) ขึ้นอีกหนึ่งตัว หน้าที่ของเราเตอร์ตัวแทนสำรองจะสร้างโทโปโลยีเทียมและรวบรวมข้อมูลในรูปแบบเช่นเดียวกับเราเตอร์ตัวแทนปกติ เมื่อใดที่เราเตอร์ตัวแทนไม่สามารถทำงานได้ เราเตอร์ตัวแทนสำรองจะรับรองหน้าที่ต่อไปทันที

5.5.3.2 การเลือกเราเตอร์ตัวแทน

การเลือกเราเตอร์ตัวแทนจะใช้ลำดับความสำคัญของเราเตอร์ที่กำหนดในแต่ละอินเทอร์เฟซของเราเตอร์โดยผู้ดูแลระบบ เราเตอร์ที่มีลำดับความสำคัญสูงสุดจะได้รับเลือกเป็นเราเตอร์ตัวแทน ในกรณีเราเตอร์สองตัวมีลำดับความสำคัญเท่ากัน เราเตอร์ที่มีเลขประจำตัวมากกว่าจะรับหน้าที่เป็นเราเตอร์ตัวแทน

กรณีที่เรามีเราเตอร์ใหม่เพิ่มเข้าสู่เครือข่าย เราเตอร์ใหม่จะตรวจสอบว่ามีเราเตอร์ตัวแทนอยู่แล้วหรือไม่ หากพบมีเราเตอร์ตัวแทนอยู่แล้วเราเตอร์ใหม่จะให้เราเตอร์ตัวแทนนั้นทำหน้าที่ต่อไปถึงแม้ว่าตัวเองจะมีลำดับความสำคัญสูงกว่าก็ตาม ทั้งนี้เพื่อหลีกเลี่ยงการเปลี่ยนแปลงเราเตอร์ตัวแทนซึ่งจำเป็นส่งลิงค์สแตตกันใหม่

5.5.4 การเลือกเส้นทาง

ภายในพื้นที่หนึ่งๆ เราเตอร์ไอเอสพีเอฟเริ่มทำงานโดยใช้โปรโตคอลฮิลโหลเพื่อตรวจหาเราเตอร์ข้างเคียง โปรโตคอลฮิลโหลจะทำหน้าที่เลือกเราเตอร์ตัวแทนเมื่อทำงานในเครือข่ายประเภทมัลติแอ็กเซส เราเตอร์แต่ละตัวจะสร้างฐานข้อมูลลิงค์สแตตโดยใช้เราเตอร์ตัวแทนเป็นศูนย์กลางการรับส่ง จากฐานข้อมูลนี้เราเตอร์แต่ละตัวจะสร้างระยะทางที่สั้นที่สุด จากตัวเองไปยังเราเตอร์ตัวอื่นในรูปของคันทัน ไม่ระยะทางที่มีตัวเองเป็นราก คันทันนี้ทำหน้าที่เป็นตารางเส้นทางให้กับโปรโตคอลสำหรับส่งข้อมูลภายในพื้นที่เดียวกัน

5.5.4.1 การเลือกเส้นทางระหว่างพื้นที่

เราเตอร์ขอบปลายพื้นที่รับหน้าที่จัดหาเส้นทางเพื่อส่งแพ็กเก็ตไปยังปลายทางที่อยู่นอกพื้นที่ เราเตอร์ขอบปลายพื้นที่จะป้อนข้อมูลเกี่ยวกับเส้นทางนอกพื้นที่ให้เราเตอร์ ภายในข้อมูลทีป้อนให้เป็นเส้นทางสรุปหรือโทโปโลยีของพื้นที่อื่นๆที่อยู่ติดกันรวมทั้งเบ็คโบน เพื่อให้เราเตอร์ใดๆในพื้นที่หนึ่งสามารถใช้เส้นทางสรุปนี้คำนวณหาระยะทางสั้นที่สุดไปยังเครือข่ายย่อยอื่นๆที่อยู่ต่างพื้นที่ได้

5.5.4.2 เส้นทางไปนอกระบบออโตโนมัส

เราเตอร์ขอบปลายของระบบออโตโนมัสซึ่งมีข้อมูลเส้นทางของระบบออโตโนมัสอื่น จะปล่อยข้อมูลนี้ไปทั่วเราเตอร์ในระบบออโตโนมัสของตนเอง ยกเว้นไม่ส่งไปยังเราเตอร์ในพื้นที่สแตบ(stub area)

เราเตอร์ขอบปลายของระบบออโตโนมัสจะสรุปรวมเส้นทางและระยะห่างไปยังระบบออโตโนมัสอื่น ไอเอสพีเอฟใช้ระยะห่างเป็นค่าเมตริกกำกับเส้นทางนอกระบบออโตโนมัส เนื่องจากโครงสร้างภายในของไอเอสพีเอฟในระบบหนึ่งจะไม่ส่งผ่านออกไปภายนอก เมตริกที่ใช้กันระหว่างต่างระบบ

ออโตโนมัสจึงอาจไม่เหมือนกัน

5.5.4.3 พื้นที่สตัป

พื้นที่สตัปในโอเอสพีเอฟหมายถึงพื้นที่ซึ่งมีเราเตอร์ขอบปลายเชื่อมต่อกับพื้นที่อื่นเพียงตัวเดียว หากย้อนกลับไปพิจารณารูปที่ 6-7 พื้นที่สตัปคือพื้นที่ 1 และพื้นที่ 2 ซึ่งต่างมีเราเตอร์ R2 และ R8 ตามลำดับ เป็นช่องทางเดียวออกไปสู่พื้นที่อื่น

เราเตอร์ภายในซึ่งสังกัดในพื้นที่สตัปใดๆจะไม่ได้รับค่าตารางเส้นทางจากภายนอกขอให้ย้อนไปพิจารณารูปที่ 6-7 ในพื้นที่ 1 เราเตอร์ R2 เป็นเราเตอร์ขอบปลายที่ทราบถึงเส้นทางออกไประบบออโตโนมัสอื่น ดังนั้น R1 จึงไม่ส่งค่าเส้นทางนี้ไปยัง R9

เราเตอร์ R2 จะส่งเฉพาะค่าเส้นทางดีฟอลต์ ซึ่งก็คือเส้นทางผ่านตัวเองไปยังเราเตอร์ภายในทุกตัว เราเตอร์ R9 ไม่มีความจำเป็นใดที่จะต้องทราบถึงเส้นทางภายนอกเพราะมีช่องทางออกเดียวผ่าน R2 ดังนั้น R9 จะใช้ดีฟอลต์เราต์ไปยัง R2 เสมอ เพื่อให้ R2 ดำเนินการเลือกหาเส้นทางต่อไป ค่าดีฟอลต์เราต์ในพื้นที่สตัปจะเกิดขึ้นอย่างอัตโนมัติจากกลไกการทำงานของโอเอสพีเอฟโดยไม่ต้องให้ผู้ดูแลเครือข่ายกำหนด

ประโยชน์ของการใช้พื้นที่สตัปคือ เมื่อเราเตอร์ภายในไม่ต้องเก็บเส้นทางออกไปภายนอก ตารางเลือกเส้นทางของเราเตอร์ในพื้นที่สตัปก็จะมีขนาดเล็กลง หน่วยความจำในเราเตอร์สามารถสงวนไว้ใช้ประโยชน์อื่นและช่วยลดภาระการทำงานของซีพียูลงได้

5.5.5 เมตริกในโอเอสพีเอฟ

อินเทอร์เฟซของเราเตอร์โอเอสพีเอฟจะมีเมตริกเรียกว่า ต้นทุนเส้นทาง (path cost) ประจำทุกอินเทอร์เฟซเพื่อใช้ตัดสินใจว่าเส้นทางใดเป็นเส้นทางที่เหมาะสมที่สุด ต้นทุนเส้นทางเป็นตัวเลขใดๆที่อยู่ระหว่าง 1 ถึง 65535 ตัวเลขที่น้อยหมายถึงเส้นทางที่ดีกว่า เมื่อนำต้นทุนเส้นทางมาคำนวณหาระยะที่สั้นที่สุดหากได้ค่าน้อยกว่าแสดงว่าเป็นเส้นทางที่สั้นกว่า

ต้นทุนเส้นทางอาจเป็นค่าเวลาหน่วง แบนวิดท์ หรือความน่าเชื่อถือ ตามข้อกำหนดของโอเอสพีเอฟให้ใช้ต้นทุนเส้นทางโดยคำนวณจากแบนวิดท์ของสายสื่อสารตามสูตร

$$\text{ต้นทุนเส้นทาง} = 10^8 / \text{แบนวิดท์สายสื่อสาร}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.6 รูปแบบเฟรมโปรโตคอลไอเอสพีเอฟ

ไอเอสพีเอฟเป็นโปรโตคอลหนึ่งที่มีหลักการซับซ้อนและรูปแบบเฟรมจำลองมากในที่นี้จะอธิบายเพียงรูปแบบเฟรมเบื้องต้นเท่านั้น

5.5.6.1 ไอเอสพีเอฟเฮดเดอร์

ไอเอสพีเอฟไม่ใช้ทรานสปอร์ตโปรโตคอลในการเ็นแคปซูลแพ็กเก็ต แต่จะใช้ไอฟีดาแกรมโดยตรง แพ็กเก็ตไอเอสพีเอฟทุกรูปแบบจะมีเฮดเดอร์ขนาด 24 ไบต์กำกับอยู่ตามรูปที่ 6-10 ถัดไปจากนั้นจะมีเฮดเดอร์เฉพาะการทำงานแต่ละรูปแบบต่อไป

0	7	8	15	16	31
Version		Type		Message Length	
Router ID					
Area ID					
Checksum			Authentication Type		
Authentication					
Authentication					
Data					

รูปที่ 5-19 เฮดเดอร์ไอเอสพีเอฟแพ็กเก็ต

- version ขนาด 8 บิต : ปัจจุบันคือเวอร์ชัน 2
- type ขนาด 8 บิต : กำหนดชนิดของแพ็กเก็ต ไอเอสพีเอฟ จะกล่าวในหัวข้อถัดไป
- message length ขนาด 16 บิต : กำหนดความยาวทั้งหมดของแพ็กเก็ตในหน่วยไบต์ (รวมเฮดเดอร์)
- router ID ขนาด 32 บิต : ไอฟีแอดเรสของเราเตอร์ที่ส่งแพ็กเก็ต
- area ID ขนาด 32 บิต : เลขพื้นที่ซึ่งเป็นเจ้าของแพ็กเก็ต
- checksum ขนาด 16 บิต : ค่าผลรวมตรวจสอบขนาด 16 บิต แบบเดิมเต็มหนึ่งโดยไม่รวมฟิลด์ authentication ขนาด 64 บิต
- authentication type ขนาด 16 บิต : ชนิดของการพิสูจน์ตัวตน ปัจจุบันมีค่าได้ 2 แบบ คือ 0 (ไม่มี) และ 1 (ใช้รหัสผ่าน)
- authentication ขนาด 64 บิต : ค่าพิสูจน์ตัวตนจริง

5.5.6.2 ชนิดแพ็กเก็ตโอเอสพีเอฟ

ฟิลด์กำหนดชนิดของแพ็กเก็ตโอเอสพีเอฟ มีอยู่ 5 แบบดังนี้คือ

1. แพ็กเก็ตฮัลโล (hello packet)
2. แพ็กเก็ตบรรยายฐานข้อมูล (database description packet)
3. แพ็กเก็ตร้องขอลิงก์สเทต (link state request)
4. แพ็กเก็ตปรับปรุงลิงก์สเทต (link state update)
5. แพ็กเก็ตร้องขอรับยืนยันลิงก์สเทต (link state acknowledgement)



บทที่ 6

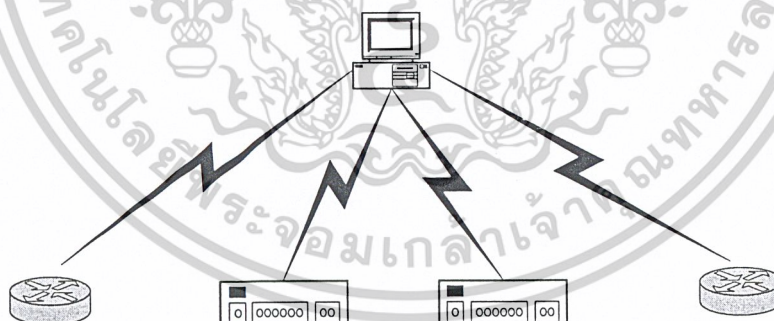
SNMP Service

การจัดการเครือข่ายใน TCP/IP อาศัยรูปแบบในการจัดการมาตรฐานตามข้อกำหนดของ โพรโทคอล SNMP ซึ่งเป็นโพรโทคอลประยุกต์ที่กำหนดรูปแบบและกรรมวิธีการจัดการเครือข่าย โดยมี สถานีจัดการเครือข่ายส่วนกลางทำหน้าที่ดูแล ตรวจสอบ และควบคุมการทำงานของอุปกรณ์เครือข่าย

6.1 พื้นฐานการบริหารเครือข่าย

ประโยชน์จากการใช้คอมพิวเตอร์และเครือข่ายคือประหยัดเวลาและค่าใช้จ่าย แต่ในขณะเดียวกัน การใช้คอมพิวเตอร์ก็ต้องลงทุนทั้งเวลาและค่าใช้จ่ายเพื่อดูแลให้คอมพิวเตอร์และเครือข่ายทำงานได้ด้วย เครือข่ายขนาดใหญ่ที่ประกอบด้วยคอมพิวเตอร์และอุปกรณ์จำนวนมากจะทำงานได้อย่างมีประสิทธิภาพ จำเป็นต้องใช้คอมพิวเตอร์ช่วยบริหารและจัดการด้วยระบบเองด้วย

การบริหารเครือข่ายคือการตรวจ ควบคุม และวางแผนการใช้ทรัพยากรระบบเพื่อให้เครือข่าย ทำงานได้อย่างมีประสิทธิภาพและสามารถตรวจหาจุดบกพร่องที่เกิดขึ้นเพื่อแก้ไขปัญหาได้อย่างรวดเร็ว คอมพิวเตอร์อย่างน้อยหนึ่งเครื่องในเครือข่ายทำหน้าที่เป็น แมเนเจอร์ (manager) เพื่อใช้เป็นสถานีจัดการ แมเนเจอร์อาจเรียกอีกชื่อว่า สถานีจัดการเครือข่าย (network management station) หรือ เอ็นเอ็มเอส (NMS)



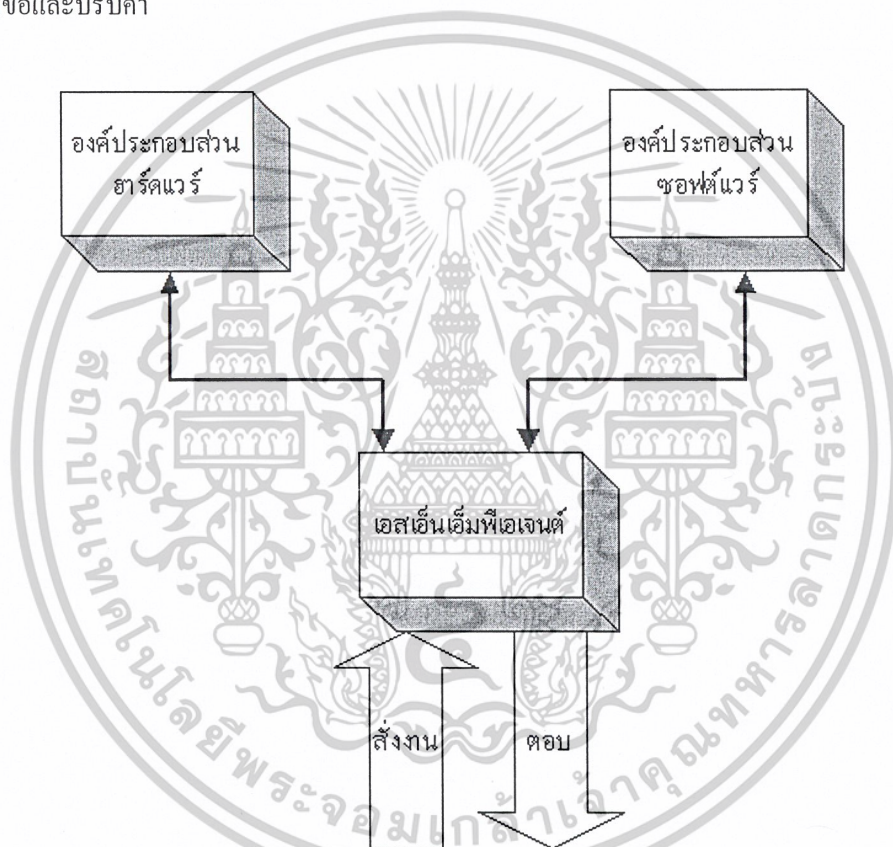
รูปที่ 6-1 องค์ประกอบในระบบจัดการเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 เอสเอ็นเอ็มพีเอเจนต์

การจัดการเครือข่ายในพีซีพี/ไอพีอาศัยรูปแบบการจัดการมาตรฐานตามข้อกำหนดของ โพรโทคอล เอสเอ็นเอ็มพี (SNMP: Simple Network Management Protocol) ซึ่งเป็นโพรโทคอล ประยุกต์ที่กำหนดรูปแบบและกรรมวิธีการจัดการเครือข่าย

อุปกรณ์เครือข่ายที่เป็นเอเจนต์อาจเป็นพีซี โมเด็ม ฮับ สวิตช์ หรือเราเตอร์ อุปกรณ์เหล่านี้จะมี ส่วนทำงานที่เป็นซอฟต์แวร์และฮาร์ดแวร์และมีเอสเอ็นเอ็มพีเอเจนต์เชื่อมต่อ เอเจนต์จะนำข้อมูลจาก ส่วนซอฟต์แวร์หรือฮาร์ดแวร์เมื่อเอ็นเอ็มเอสร้องขอข้อมูล และปรับเปลี่ยนการทำงานของซอฟต์แวร์หรือ ฮาร์ดแวร์เมื่อเอ็นเอ็มเอสสั่งงาน โดยมีการแจ้งยืนยันสิทธิในรูปรหัสผ่านว่าเอ็นเอ็มเอสมีอำนาจหน้าที่ใน การร้องขอและปรับค่า



รูปที่ 6-2 เอสเอ็นเอ็มพีเอเจนต์

6.2.1 เอ็มไอบี

เอเจนต์ประกอบด้วยส่วนสำคัญสองส่วนคือ โพรโทคอลเอ็นจิน และฐานข้อมูลสารสนเทศการ จัดการ โพรโทคอลเอ็นจินทำหน้าที่ประมวลคำสั่งที่มาจากเอ็นเอ็มเอสซึ่งได้แก่ รับคำสั่ง ถอดรหัสคำสั่ง ทำงานตามคำสั่งและส่งผลตอบกลับ ฐานสารสนเทศการจัดการหรือเรียกสั้นๆว่า เอ็มไอบี เป็นส่วนที่เก็บ ตัวแปรและค่ากำหนดการทำงานประจำอุปกรณ์

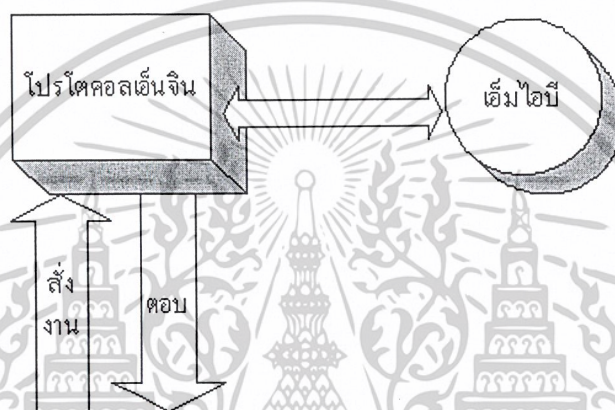
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2 อ็อบเจ็กต์ไอดีเอ็นทีไฟเออร์

ภายในเอ็มไอบีประกอบด้วยตัวแปรจำนวนมากที่เรียกโดยทั่วไปว่า อ็อบเจ็กต์จัดการ อ็อบเจ็กต์ในความหมายนี้เป็นชื่อที่เรียกตัวแปรและลักษณะเฉพาะของตัวแปรในเอ็มไอบีโดยไม่เกี่ยวข้องกับเรื่องเชิงวัตถุพิสัยแต่อย่างใด ขอให้พิจารณาว่าอ็อบเจ็กต์ในเอสเอ็นเอ็มพีมีลักษณะเช่นเดียวกับคอร์ดในฐานข้อมูล

แต่ละอ็อบเจ็กต์จะมีชื่อเฉพาะเรียกว่า อ็อบเจ็กต์ไอดีเอ็นทีไฟเออร์ หรือเรียกโดยย่อว่า ไอดีเอ็นทีไฟเออร์ เพื่อใช้อ้างอิงถึงอ็อบเจ็กต์นั้น

อ็อบเจ็กต์ทุกตัวมีนิยามที่กำหนด ชื่อ แบบข้อมูล สิทธิการเข้าถึง คำอธิบายลักษณะและค่าข้อมูล การนิยามอ็อบเจ็กต์มีกฎเกณฑ์ตามข้อกำหนด โครงสร้างฐานข้อมูลสารสนเทศการจัดการ



รูปที่ 6-3 โครงสร้างของเอเจนต์

6.3 โพรโทคอล

การติดต่อระหว่างสถานีจัดการกับเอเจนต์มีรูปแบบในการติดต่อหลายรูปแบบด้วยกันตามวัตถุประสงค์ในการติดต่อ แบบของการติดต่อในเอสเอ็นเอ็มพีรุ่น 1 มี 5 แบบคือ

1. get-request ใช้สอบถามข้อมูลจากตัวเอเจนต์ที่อยู่บนอุปกรณ์ที่ต้องการตรวจสอบในระบบเครือข่าย
2. get-next-request ใช้สอบถามข้อมูลที่เรียงเป็นลำดับ เช่นข้อมูลที่เก็บอยู่ในรูปตาราง หรือในกรณีที่ไม่ทราบชื่อตัวแปรที่แน่ชัด
3. get-response เอเจนต์ส่งคำตอบกลับมายังผู้สอบถาม
4. set-request ใช้เปลี่ยนแปลงค่าตัวแปรที่เอเจนต์รับผิดชอบอยู่
5. trap ใช้แจ้งเหตุการณ์ที่เกิดขึ้นในระบบเครือข่าย เช่นการเริ่มต้นทำงานใหม่ของอุปกรณ์ หรือเส้นทางขัดข้อง

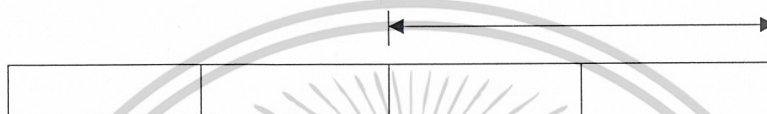
เอสเอ็นเอ็มพีอาศัยโพรโทคอลยูติลิตี้ โดยใช้พอร์ตหมายเลข 161 สำหรับการติดต่อในแบบที่ 1 ถึง 4 และใช้พอร์ตหมายเลข 162 สำหรับแบบที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.1 การเ็นแคปซูล

การเ็นแคปซูลคำสั่งและข้อมูลในเอสเอ็นเอ็มพีมีวิธีตามรูปที่ ---- ฟอ์แมตของเอสเอ็นเอ็มพีประกอบด้วย 2 ส่วนคือ เฮดเดอร์และพืดย เฮดเดอร์ประกอบด้วยฟิลด์ย่อยสองฟิลด์คือ

- Version รุ่นของโพร โทคอลที่ใช้ ถ้าเป็นโพร โทคอลรุ่น 1 จะมีค่า 0 หากเป็นรุ่น 2 จะมีค่า 1
- Community รหัสผ่านในรูปสายอักขระเพื่อให้เอเจนต์ใช้ตรวจสอบว่าข้อความที่ส่งมามีสิทธิ์ในการสอบถามหรือเปลี่ยนแปลงข้อมูลหรือไม่

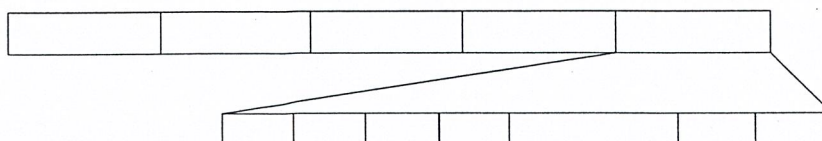


รูปที่ 6-4 การเ็นแคปซูลเอสเอ็นเอ็มพี

ในส่วนของพืดยประกอบด้วยฟิลด์ย่อยตามชนิดของข้อความ หากเป็นข้อความ get, get-next และ get-response จะมีโครงสร้างเดียวกัน รูปที่ 6-5 แสดงโครงสร้างของพืดยโดยแต่ละฟิลด์มีความหมายดังนี้

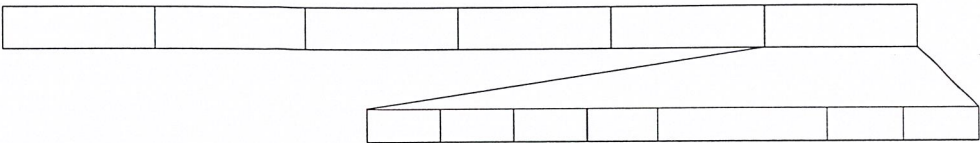
- PDU type แบบการติดต่อ (1 ถึง 5)
- Request ID กำหนดบอกหมายเลขข้อความเพื่อให้จับคู่เมื่อรับคำตอบกลับมา
- Error Status สถานะความผิดพลาดที่เกิดขึ้น
- Error Index ดรรชนีชี้ค่าผิดพลาดที่เกิดขึ้นเกิดจากตัวแปรตัวลำดับที่เท่าไรของตัวแปรทั้งหมดที่สอบถามไป
- VarBindList ค่าผูกพันตัวแปร(variable binding) แสดงอยู่ในรูปของตัวแปรและค่าของตัวแปรต่อเนื่องกันไปเป็นรายการ

สำหรับข้อความไทร์บมีลักษณะแตกต่างกันออกไปดังรูปที่ ----- โปรดสังเกตว่าในที่นี้ไม่ได้กล่าวขนาดความยาวของแต่ละฟิลด์ เพราะทุกฟิลด์ในเอสเอ็นเอ็มพีต้องเข้ารหัสและจะได้ขนาดของแต่ละฟิลด์ที่มีความยาวแตกต่างกันไปตามชนิดข้อมูล



รูปที่ 6-5 โครงสร้างพืดยของ get, get-next และ get-response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-6 โครงสร้างพีคียูของคำสั่ง trap

6.3.2 รหัสผิดพลาด

เอเจนต์จะตอบคำถามพร้อมทั้งแจ้งรหัสการทำงานกลับไปยังเอ็นเอ็มเอส ตารางที่ 6-1 แสดงรหัสผิดพลาดที่ใช้ในเอสเอ็นเอ็มพี

Enterprise

รหัสผิดพลาด	ชื่อ	คำอธิบาย
0	noError	ไม่มีข้อผิดพลาด
1	tooBig	เอเจนต์ไม่สามารถส่งคำตอบได้ในเฟรมเดียว
2	noSuchName	ไม่มีตัวแปรที่ต้องการสอบถามอยู่ในฐานข้อมูล
3	badValue	ค่าที่กำหนดให้ตัวแปรไม่ถูกต้อง
4	readOnly	เปลี่ยนค่าตัวแปรไม่ได้เพราะอ่านค่าได้เพียงอย่างเดียว
20	genErr	มีข้อผิดพลาดอื่นๆเกิดขึ้น

ตารางที่ 6-1 รหัสผิดพลาดในเอสเอ็นเอ็มพี

6.4 โครงสร้างเอ็มไอบี

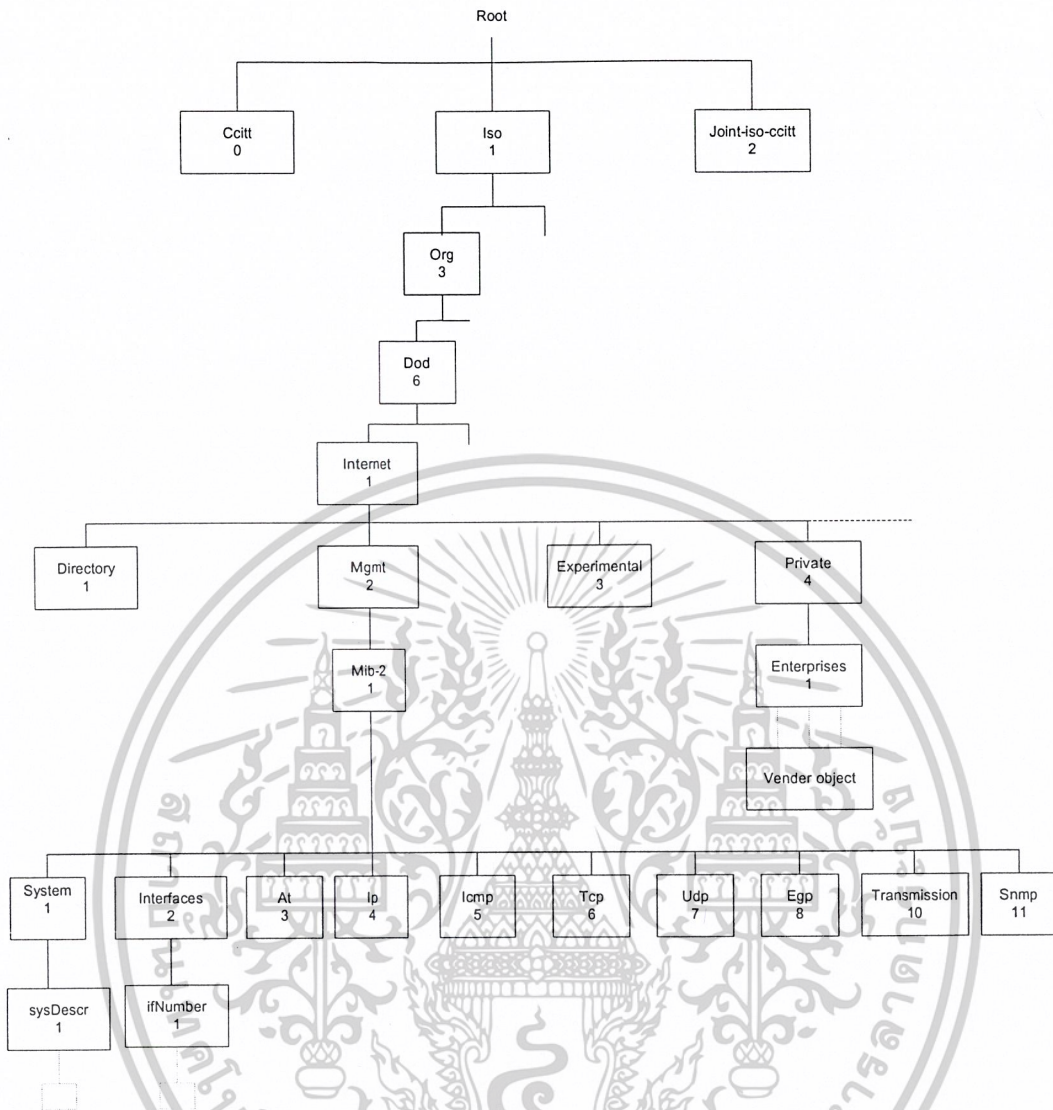
ข้อมูลประจำอุปกรณ์เครือข่ายชั้นหนึ่งๆมีได้อย่างหลากหลาย อีกทั้งอุปกรณ์ต่างประเภทกันย่อมมีข้อมูลประจำอุปกรณ์แตกต่างกัน ดังนั้นการสอบถาม หรือเปลี่ยนค่าฐานข้อมูลจึงต้องมีรูปแบบมาตรฐานให้กับอุปกรณ์ทุกประเภท โครงสร้างต้นไม้แบบลำดับชั้นเป็น โครงสร้างที่เหมาะสมสำหรับใช้เป็นฐานข้อมูลเพื่อจัดเก็บตัวแปรเหล่านี้

รูปนี้ 6-7 แสดงข้อมูลหรืออ็อบเจกต์ของเอสเอ็นเอ็มพีในโครงสร้างแบบต้นไม้ซึ่งนิยมเรียกว่าเอ็มไอบีทีรี แต่ละโหนดซึ่งแทนอ็อบเจกต์หนึ่งๆมีชื่อพร้อมทั้งตัวเลขฐานสิบกำกับประจำโหนดเพื่อใช้อ้างอิง ยกเว้นรากซึ่งไม่มีชื่อกำกับ

ลำดับชั้นแรกจะมีโหนดหลักสามโหนดหลักสามโหนดซึ่งกำหนดกลุ่มองค์กรสามกลุ่มคือ ITU-T(0), ISO(1), และ Joint-ISO-ITU-T(2) ภายใต้โหนด ISO มีโหนดลำดับที่สามคือ org(3) กำหนดองค์กรนานาชาติ และส่วนหนึ่งขององค์กรนี้คือ dod(6) หรือ Department of Defense และมีโหนด internet(1) เพื่อกำหนดกลุ่มการจัดการเครือข่ายในอินเทอร์เน็ต

เมื่อต้องการอ้างอิงถึงโหนดใดในโครงสร้างให้เขียนหมายเลขจากรากไปตามเส้นทางถึงโหนดนั้นและค้นด้วยจุด ลำดับตัวเลขนี้เรียกว่า อ็อบเจกต์ไเดนติไฟเออร์ (object identifier) หรือ โอไอดี (OID)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-7 อ็อบเจ็กต์ไอดีเอ็นทีไฟเบอร์ในโครงสร้างฐานข้อมูลสารสนเทศ

ตัวอย่างเช่น 1.3.6.1.2.1.1 เป็นอ็อบเจ็กต์ไอดีเอ็นทีไฟเบอร์ โดยมีชื่อที่สมนัยกันคือ iso.org.dod.internet.mgmt.mib-2.system โหนดที่อยู่ภายใต้ 1.3.6.1.2.1 หรือในกลุ่ม mib-2 เป็นโหนดสำหรับใช้งานเอสเอ็นเอ็มพี แต่ละโหนดจะมีโหนดย่อยเพื่ออ้างอิงถึงตัวแปรเช่น 1.3.6.1.2.1.1.1 คือตัวแปร sysDescr (System Description) ซึ่งเก็บคำอธิบายเกี่ยวกับอุปกรณ์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.1 กลุ่มในเอ็มไอบี

เอ็มไอบีภายใต้ internet มีกลุ่มย่อยทั้งหมด 6 กลุ่มคือ

- directory(1) สงวนไว้สำหรับใช้งานในอนาคต
- mgmt(2) กลุ่มเอ็มไอบีที่ใช้ในการจัดการภายใต้เอสเอ็นเอ็มพีรุ่น 1
- experimental(3) ใช้สำหรับการทดลอง
- private(4) สำหรับให้ผู้ผลิตกำหนดตัวแปรเฉพาะอุปกรณ์
- security(5) ใช้ในระบบรักษาความปลอดภัย
- SNMPv2(6) ใช้ในเอสเอ็นเอ็มพีรุ่น 2

ภายใต้กลุ่ม mib-2 บรรจุกลุ่มย่อยที่ใช้ในเอสเอ็นเอ็มพีซึ่งประกอบด้วย interfaces, at และ ip และอื่นๆ ความหมายของแต่ละกลุ่มอธิบายไว้ในตารางที่ 6-2 แต่ละกลุ่มจะประกอบด้วยตัวแปรซึ่งมีแบบต่างๆกันไป

ลำดับ	ชื่อ	ความหมาย
1	system	ข้อมูลระบบ
2	interfaces	ข้อมูลอินเทอร์เฟซที่ใช้เชื่อมต่อ
3	at	ข้อมูลการแปลงแอดเดรส
4	ip	ข้อมูลไอพี
5	icmp	ข้อมูลไอซีเอ็มพี
6	tcp	ข้อมูลที่ซีพี
7	udp	ข้อมูลยูดีพี
8	egp	ข้อมูลโพรโทคอลเกตเวย์ภายนอก
10	transmission	ข้อมูลสายสื่อสาร
11	snmp	ข้อมูลเอสเอ็นเอ็มพี

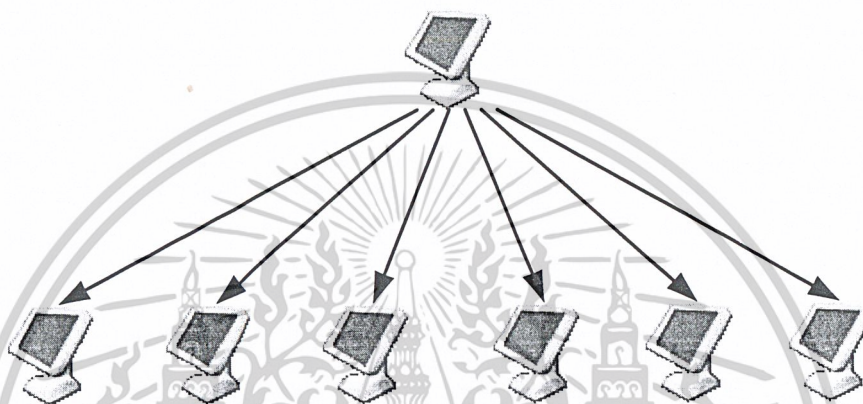
ตาราง 6-2 กลุ่มย่อยภายใต้ mgmt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การสแกนเพื่อตรวจสอบ

7.1 Network Ping Sweeps



รูปที่ 7-1 แสดงการทำงานของ ping sweep

หนึ่งในเทคนิคพื้นฐานที่นิยมกระทำกันก็คือ การ ping ไปยังเครื่องเป้าหมายจำนวนมากพร้อมๆ กัน ในลักษณะคล้ายกับการกวาดหรือการยิง ซึ่งเราเรียกว่าการทำ ping sweep เพื่อตรวจสอบว่าเครื่องปลายทางใดบ้างที่ยังเปิดทำงานอยู่ โดยปกติ ถ้าคุณใช้คำสั่ง ping ธรรมดา, ping จะมีการส่ง แพ็กเก็ต ICMP ECHO (Type 8) ออกไปยังเครื่องปลายทางและรอคอย ICMP ECHO_REPLY (Type 0) ที่ถูกส่งกลับมา ถึงแม้ ping จะมีประโยชน์สำหรับการทดสอบว่าเครื่องปลายทางเปิดอยู่หรือไม่ก็ตาม แต่มันจะเหมาะสมสำหรับเครื่องที่อยู่บนเน็ตเวิร์กขนาดเล็กถึงขนาดกลางเท่านั้น มันจะไม่มีประสิทธิภาพเพียงพอที่จะนำมาใช้ตรวจสอบเครื่องที่อยู่บนเน็ตเวิร์กขนาดใหญ่ได้ การตรวจสอบเครื่องที่อยู่ในเน็ตเวิร์กที่ใช้แอตเดรสในคลาส A อาจกินเวลานานหลายชั่วโมงกว่าจะทราบผล เทคนิคในการ ping sweep มีหลายเทคนิคแตกต่างกันไป เช่น ปกติแล้วการ ping จะรอคอยการตอบสนองจากเครื่องทีละเครื่อง ก่อนจะเปลี่ยนไปทดสอบเครื่องอื่นๆ ถัดไป มันจะใช้การส่ง แพ็กเก็ต ICMP ออกไปพร้อมๆ กันแบบขนานไปยังเครื่องปลายทางหลายๆ เครื่อง ในลักษณะคล้าย “Round Robin” (คือส่งแพ็กเก็ต ICMP ไปที่เครื่อง 1,2,3,... ถึงเครื่องสุดท้าย แล้ววนกลับมาส่งแพ็กเก็ตไปที่เครื่อง 1,2,3 ใหม่ไปเรื่อยๆ แล้ววนกลับมาอีก โดยไม่จำเป็นต้องหยุดรอจากตอบสนองจากเครื่องแรก) ดังนั้น จะทำงานได้รวดเร็วกว่าคำสั่ง ping ธรรมดามาก แต่อาจทำให้เกิดทราฟฟิกจำนวนมากที่เกิดขึ้นอาจเข้าไปรบกวนแบนด์วิธของ WAN Link ความเร็วอย่างต่ำเช่น 128K ISDN หรือ เฟรมรีเลย์ (Frame relay)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในบางครั้งจะอย่างไรถ้าที่เน็ตเวิร์กเป้าหมายได้มีการบล็อกห้ามแพ็กเก็ต ICMP ไว้ไม่ให้เข้าถึงเน็ตเวิร์กภายในได้

ถ้าแพ็กเก็ต ICMP ถูกบล็อกเอาไว้ เรามีเทคนิคและเครื่องมืออื่นที่ใช้ตรวจสอบได้ว่าเครื่องปลายทางใดบ้างที่เข้าถึงได้และเปิดทำงานอยู่ แต่อย่างไรก็ตาม มันอาจไม่ถูกต้องและรวดเร็วเท่ากับตรวจสอบด้วย ping sweep

เทคนิคที่ว่ามันก็คือ การสแกนพอร์ต นั่นเอง ซึ่งเป็นเทคนิคแรกที่ใช้ถ้าใช้ปลายทางบล็อกแพ็กเก็ต ICMP ไว้ โดยการสแกนพอร์ตต่างๆไปบนแต่ละเครื่องเราจะสามารถคาดการณ์ได้ว่าเครื่องไหนเปิดอยู่บ้าง เทคนิคนี้ค่อนข้างกินเวลาและอาจไม่สามารถสรุปแน่ชัดได้ TCP Ping Scan ด้วย TCP Ping scan ไปที่พอร์ตที่ต้องการส่วนมากจะเป็นพอร์ต 80 เนื่องจากเป็นพอร์ตมาตรฐานที่เราเตอร์หรือไฟร์วอลล์ของเน็ตเวิร์กส่วนใหญ่จะเปิดไว้ให้ผ่านเข้าไปได้ยังเว็บเซิร์ฟเวอร์ที่มักตั้งอยู่ในส่วนที่เรียกว่า Demilitarized zone (DMZ) หรือยิ่งไปกว่านั้น อาจทะลุผ่านเข้าไปในอินทราเน็ตภายในด้วย โดยจะทำงานดังนี้คือสร้างแพ็กเก็ตของโปรโตคอล TCP ซึ่งได้เซตแฟล็ก ACK (Acknowledge) ไว้เป็น 1 ด้วย (ต่อไปเราจะเรียกแพ็กเก็ตลักษณะนี้สั้นๆว่า แพ็กเก็ต TCP ACK) แล้วส่งแพ็กเก็ตเหล่านี้ไปยังเน็ตเวิร์กเป้าหมายปลายทางหรือไม่ ถ้ามีก็แสดงว่าเครื่องปลายทางสามารถติดต่อได้และเปิดทำงานอยู่ แพ็กเก็ต TCP ACK นี้มักวิ่งทะลุผ่านไฟร์วอลล์ที่ไม่ค่อยฉลาดนักหรือคอนฟิกไว้ไม่ดีพอให้เข้าไปได้

อย่างที่ได้เห็นไปแล้วว่าเทคนิคนี้ค่อนข้างใช้ได้ผลถึงแม้เน็ตเวิร์กปลายทางจะบล็อกแพ็กเก็ต ICMP ไว้ ซึ่งถ้า พอร์ต 80 ใช้ไม่ได้อาจใช้เป็นพอร์ตมาตรฐานที่มักพบบ่อย อย่างเช่น พอร์ตของเซิร์ฟเวอร์ SMTP (หมายเลข 25), POP (110), AUTH (113), IMAP (143) หรือพอร์ตอื่นๆ ที่เป็นไปได้

ถ้าพอร์ตที่เครื่องปลายทางได้เปิดไว้แพ็กเก็ตของโปรโตคอล TCP ที่มีการเซตแฟล็ก SYN(S bit) ไว้และแพ็กเก็ต TCP ที่มีการเซตแฟล็ก ACK ไว้จะถูกส่งกลับมา

นอกจากนี้ยังมีเทคนิคอื่นในการที่จะตรวจสอบว่าเครื่องปลายทางยังเปิดทำงานอยู่หรือไม่ด้วยการส่งแพ็กเก็ต ICMP ECHO ออกไป พร้อมทั้งแพ็กเก็ตของโปรโตคอล ICMP ใน Type อื่นๆด้วย ได้แก่ ICMP Type TIME STAMP REQUEST และ ICMP INFO request เพื่อที่ว่า ถึงแม้แพ็กเก็ต ICMP ใน Type ECHO ถูกสกัดกั้นไว้ที่เราเตอร์ตัวที่เชื่อมต่อออกอินเทอร์เน็ตหรือที่ไฟร์วอลล์ก็ตาม แต่โอกาสที่แพ็กเก็ต ICMP ใน Type อื่นๆจะวิ่งทะลุผ่านเข้าไปเพื่อตรวจสอบได้ก็ยังมีอยู่ หรืออาจจะส่งไปด้วยแพ็กเก็ตปลอม (spoofed packet) เพื่อป้องกันการตรวจสอบย้อนกลับไปได้ว่าแพ็กเก็ตถูกส่งมาจากที่ไหน

กล่าวโดยสรุป ขั้นตอนนี้จะทำให้เราสามารถตรวจสอบได้อย่างแท้จริงว่าเครื่องปลายทางใดบ้างที่เราสามารถติดต่อได้โดยตรงผ่านอินเทอร์เน็ต ด้วยการส่งแพ็กเก็ต ICMP ใน TYPE ต่างๆ หรือใช้เทคนิคของการสแกนพอร์ตเข้าไปตรวจสอบ จากลิสต์รายการหมายเลข IP Address ที่อยู่ในคลาส C จำนวนกว่า 200 หมายเลข เราสามารถตรวจสอบได้ว่าแท้จริงแล้วมีเครื่องไหนบ้างที่อยู่ในขอบข่ายของการเป็นเครื่องเป้าหมายได้ เพื่อช่วยลดประสิทธิภาพของการตรวจสอบให้แคบลงและประหยัดเวลามากขึ้น

- TCP Null scan เทคนิคนี้จะส่งแพ็กเก็ต TCP ออกไปโดยเซตค่าของทุกๆฟิลด์ให้เป็น 0 หมด และ อาศัยมาตรฐาน RFC 793 อีกเช่นกัน เครื่องปลายทางจะส่งแพ็กเก็ต TCP RST ของทุกๆพอร์ตที่เปิดอยู่กลับมาให้
- TCP ACK scan เทคนิคนี้จะถูกใช้เพื่อค้นหา “rule” และ “policy” ต่างๆที่เซตไว้ที่ไฟร์วอลล์ เพื่อตรวจสอบดูว่าไฟร์วอลล์นั้นๆทำหน้าที่แค่เพียงกรองแพ็กเก็ตได้อย่างง่าย ๆ หรือเป็นไฟร์วอลล์ที่มีความฉลาดพอสมควรและใช้เทคนิคการกรองแพ็กเก็ตขั้นสูง
- TCP Window scan เทคนิคนี้จะตรวจสอบพอร์ตที่เปิดอยู่ รวมทั้งตรวจสอบว่า พอร์ตใดบ้างที่ถูกฟิลเตอร์เอาไว้ไม่ให้ผ่านเข้าไปถึง และพอร์ตหมายเลขใดได้รับการอนุญาตไว้บ้าง โดยอาศัยช่องโหว่จากความคิดปกติบางอย่างในการแจ้งค่าของ TCP Window Size ของโปรโตคอล TCP/IP
- TCP RPC scan เทคนิคนี้ใช้งานได้เฉพาะกับเครื่องปลายทางที่รันยูนิกซ์เท่านั้น มันถูกใช้เพื่อตรวจสอบดูว่ามีเซอร์วิสใดทำงานอยู่บนเซอร์วิส RPC บ้าง รวมทั้งตรวจสอบดูเวอร์ชันของเซอร์วิสนั้นและโปรแกรมอื่นที่เกี่ยวข้อง
- UDP scan เทคนิคนี้จะส่งแพ็กเก็ตของโปรโตคอล UDP ไปยังพอร์ตเป้าหมาย ถ้าเครื่องปลายทางตอบกลับมาด้วยแพ็กเก็ต ICMP type PORT UNREACHABLE นั้นหมายความว่าพอร์ตนั้นปิดอยู่ในทางตรงกันข้าม ถ้าเราไม่ได้รับแพ็กเก็ต ICMP type ดังกล่าว เราสามารถสรุปได้ว่าพอร์ตนั้นเปิดอยู่ เนื่องจากโปรโตคอล UDP เป็นโปรโตคอลลักษณะ connectionless คือไม่รับรองว่าแพ็กเก็ตที่ส่งไปจะถึงเครื่องปลายทางครบถ้วนหรือไม่ ดังนั้นความถูกต้องของผลลัพธ์ที่ได้จากเทคนิคนี้ก็อาจขึ้นกับปัจจัยอื่นๆ ด้วยเช่น ปริมาณทราฟฟิกในเน็ตเวิร์กและทรัพยากรบนเครื่องปลายทาง นอกจากนั้นมันยังเป็นเทคนิคที่ค่อนข้างช้าอีกด้วยถ้าคุณกำลังสแกนเน็ตเวิร์กที่ใช้งานไฟร์วอลล์หรือเราเตอร์ที่มีการฟิลเตอร์กรองแพ็กเก็ต จึงขอให้เตรียมใจไว้ด้วยกับผลลัพธ์ที่ไม่คาดคิดของ UDP scan

ไคร์เวอร์ของโปรโตคอล TCP/IP ของระบบปฏิบัติการบางตัวอาจส่งแพ็กเก็ต TCP RST กลับไปยังเครื่องต้นทางตลอดไม่ว่าพอร์ตๆ นั้นจะเปิดหรือปิดอยู่ ดังนั้นจึงเป็นไปได้ว่า ผลลัพธ์ที่ได้ อาจแตกต่างกันไปไม่แน่นอน แต่อย่างไรก็ดี การตรวจสอบด้วยแพ็กเก็ต TCP SYN ธรรมดานั้นจะทำงานได้ดีสำหรับทุกๆ โฮสต์

7.4 การตรวจหาประเภทของระบบปฏิบัติการ

จุดประสงค์ประการแรกของการสแกนคือ การค้นหา TCP port และ UDP port ที่เปิดอยู่บนเครื่องปลายทาง ส่วนจุดประสงค์ประการที่สองคือ การค้นหาประเภทของระบบปฏิบัติการที่รันอยู่บนเครื่องปลายทาง ประเภทของระบบปฏิบัติการที่เราทราบจะถูกนำไปใช้ประโยชน์ในขั้นตอนการค้นหารายละเอียดต่างๆ (enumeration) โดยอาศัยข้อบกพร่องต่างๆ ที่แต่ละระบบปฏิบัติการนั้นมี ดังนั้นเราจึงต้องมั่นใจว่า ระบบปฏิบัติการที่เราทราบนั้นเป็นข้อมูลที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Active Stack Fingerprinting

Stack fingerprinting เป็นเทคโนโลยีที่ช่วยทำให้มั่นใจได้ว่าประเภทของระบบปฏิบัติการที่ค้นหาได้นั้นเป็นระบบปฏิบัติการที่ถูกต้อง มีเปอร์เซ็นต์ความน่าเชื่อถือสูง โดยอาศัยหลักการที่ว่า ผู้ผลิตระบบปฏิบัติการแต่ละรายมักพัฒนาไครเวอร์ของโพรโตคอล TCP/IP และเซอร์วิสที่ทำงานบนโพรโตคอล TCP/IP ให้มีเอกลักษณ์เฉพาะตัวเป็นของตัวเอง ซึ่งมักแตกต่างกับของระบบปฏิบัติการอื่น

ดังนั้น โดยการตรวจวัดความแตกต่างเหล่านี้ เราจะสามารถเริ่มคาดเดาได้อย่างมีเหตุผล แต่เพื่อให้มีความน่าเชื่อถือสูงสุด Stack fingerprinting จำเป็นต้องอาศัยการคอนเน็กไปพอร์ตที่เปิดอยู่น้อยหนึ่งพอร์ต แต่ การทำงานของโครงงานนี้ สามารถคาดเดาได้อย่างมีเหตุผล ถึงแม้ว่าจะไม่ได้คอนเน็กไปที่พอร์ตใดเลย แต่ผลที่ได้ก็อาจยังไม่น่าเชื่อถือนัก โดยให้สำรวจวิธีการตรวจวัดความแตกต่างกันทีละวิธี

- Fin probe ใช้การส่งแพ็กเก็ตของโพรโตคอล TCP โดยเซตแฟล็ก Fin ให้เป็น 1 ไว้ ตามมาตรฐาน RFC 793 ระบุว่า พฤติกรรมที่ถูกต้องจะต้องไม่มีการส่งแพ็กเก็ตอะไรตอบสนองกลับไป แต่ทว่า ไครเวอร์ของโพรโตคอล TCP/IP ในระบบปฏิบัติการบางตัว อย่างเช่น วินโดวส์เอ็นที จะตอบสนองกลับมามีแพ็กเก็ตของโพรโตคอล TCP ที่เซตแฟล็ก Fin และ ACK ให้เป็น 1 ซึ่งเครื่องมือส่วนมากได้นำเทคนิคนี้ไปใช้กัน
- Bogus Flag probe ใช้การส่งแพ็กเก็ตของโพรโตคอล TCP โดยเซตแฟล็ก SYN ให้เป็น 1 พร้อมทั้งเซตบิตตำแหน่งที่ยังไม่ได้ใช้งานให้เป็น 1 ด้วย บางระบบปฏิบัติการเช่น ลินุกซ์ จะตอบสนองกลับมามีการเซตแฟล็กบางตัวในแพ็กเก็ต ว่ามักเป็นค่าอะไร อย่างไรก็ตามบางระบบปฏิบัติการจะทำการ reset connection เมื่อได้รับแพ็กเก็ตนี้
- TCP Initial Sequence Number (ISN) sampling ใช้วิธีการตรวจหาแพทเทิร์นของ Sequence number ที่อยู่ในส่วนหัวของแพ็กเก็ตที่ได้รับการตอบสนองต่อการร้องขอการเชื่อมต่อ ว่าเป็นค่าอะไร ซึ่งสามารถแบ่งได้หลายกลุ่มเช่น ถ้าเป็นแบบ traditional 64K จะพบใน UNIX รุ่นเก่า, แบบ Random increments จะพบใน Solaris, IRIX, FreeBSD, Digital UNIX, Cray ในเวอร์ชันใหม่, แบบ True "random" จะพบใน Linux 2.0.*, OpenVMS, AIX เวอร์ชันใหม่, ส่วน Windows จะเป็นแบบ "time dependent" model คือค่าของ ISN จะเพิ่มขึ้นจำนวนหนึ่งที่กำหนดไว้ในแต่ละช่วงเวลา, และในบางครั้งจะใช้ค่า ISN ค่าเดียวไม่เปลี่ยนแปลง เช่นใน 3Com hubs จะใช้เป็น 0x803 ตลอดและใน Apple LaserWriter printers จะใช้เป็น 0xC7001
- "Don't fragment bit" monitoring บางระบบปฏิบัติการได้เซตบิต "Don't fragment bit" ไว้เพื่อเพิ่มความเร็วในการส่งข้อมูล ให้มอนิเตอร์บิตนี้เพื่อตรวจดูว่าระบบปฏิบัติการไหนเซตบิตนี้บ้าง
- TCP initial window size ใช้วิธีดูขนาดของ TCP window เพราะบางระบบปฏิบัติการจะมีการลือค่าไว้เลยว่าขนาดของ TCP window เป็นเท่าไร (เช่น AIX เป็นเพียงระบบปฏิบัติการเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มีขนาดของ window เท่ากับ 0x3F25 ส่วน window NT มีขนาด 0x402E) ค่านี้มักเป็นค่าเฉพาะตัวของแต่ละระบบปฏิบัติการด้วย จึงยังทำให้ผลที่ได้มีความถูกต้องมากขึ้น

- ACK value ระบบปฏิบัติการแต่ละระบบมักเซตค่าในฟิลด์ ACK ไม่เหมือนกัน บางระบบเซตค่าฟิลด์ ACK ให้สอดคล้องตามค่าของฟิลด์ SYN ที่เซตไว้ในฝั่งผู้ส่ง บางระบบจะเซตค่าฟิลด์ ACK ให้บวกจากค่าของฟิลด์ SYN ไปอีกหนึ่ง
- ICMP error message quenching บางระบบปฏิบัติการอาจปฏิบัติตามมาตรฐาน RFC 1812 และมีการจำกัดอัตราการส่งข้อความแจ้งข้อผิดพลาดดังนั้น (ใน Linux kernel จะจำกัดการส่งข้อความแจ้งข้อผิดพลาดไปยังปลายทางที่อัตรา 80 แพ็กเก็ตต่อ 4 ± 0.25 วินาที) โดยการส่ง
- แพ็กเก็ตของโปรโตคอล UDP ไปยังหมายเลขพอร์ตสูงๆ แล้วค่อยนับจำนวนของ ICMP type PORT UNREACHABLE Message ที่ตอบกลับภายในช่วงเวลาที่กำหนด
- ICMP message quoting เมื่อพบข้อผิดพลาดเกี่ยวกับโปรโตคอล TCP/IP ระบบปฏิบัติการแต่ละประเภทจะให้ข้อมูลและสาเหตุต่างๆมาในแพ็กเก็ตของ ICMP ไม่เท่ากัน บางระบบจะให้รายละเอียดมากบางระบบจะให้รายละเอียดน้อย (ใน Solaris จะส่งกลับมา a bit more และใน Linux จะส่งกลับมา even more than that) ดังนั้นโดยการสำรวจข้อมูลที่ได้มานี้ เราพอจะใช้ในการคาดเดาประเภทของระบบปฏิบัติการได้
- ICMP error message-echoing integrity บางระบบปฏิบัติการอาจดัดแปลงส่วนหัวของแพ็กเก็ต IP เมื่อมีการส่ง ICMP error message ด้วยการตรวจสอบลักษณะของการดัดแปลงดังกล่าวนี้คุณสามารถนำมาใช้ในการคาดเดาได้
- Type of service (TOS) ให้คุณสำรวจฟิลด์ที่ชื่อ Type of service ที่อยู่ในแพ็กเก็ต ICMP ประเภท port unreachable ซึ่งโดยปกติหลายระบบปฏิบัติการจะใช้ค่าศูนย์ แต่บางระบบอาจใช้ค่าอื่น เช่น Linux ใช้ 0xC0
- Fragmentation handling แต่ละระบบปฏิบัติการจะจัดการกับแพ็กเก็ตที่ถูกเฟรกเมนต์หรือหั่นซอยไม่เหมือนกัน เมื่อมีการประกอบแพ็กเก็ตย่อยขึ้นมาเป็นแพ็กเก็ตที่สมบูรณ์ บางระบบจะเขียนทับ
แพ็กเก็ตย่อยอันเก่าด้วยแพ็กเก็ตย่อยอันใหม่หรือบางระบบก็ทำในทางตรงกันข้าม โดยการสังเกตพฤติกรรมตรงนี้ เราพอจะใช้ในการคาดเดาประเภทของระบบปฏิบัติการได้เช่นเดียวกัน สามารถหาข้อมูลเพิ่มเติมได้ที่ www.secnet.com
- TCP options ได้ถูกกำหนดไว้ในมาตรฐาน RFC 793 และได้รับการปรับปรุงในมาตรฐาน RFC 1323 ในปัจจุบัน ผู้ผลิตหลายรายได้มีการอิมพลีเมนต์ออปชันพิเศษที่อยู่ใน RFC 1323 ไว้ในระบบปฏิบัติการของตน ดังนั้น ด้วยการส่งแพ็กเก็ตที่เซตออปชันหลายๆออปชันไปทดสอบ อย่างเช่น no operation, maximum segment size, window scale factor และ timestampsมัน เป็นไปได้ที่เราจะตั้งสมมติฐานเกี่ยวกับระบบปฏิบัติการที่รันที่เครื่องเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยกตัวอย่าง

Window Scale=10; NOP; Max Segment Size = 265; Timestamp; End of Ops;

บางระบบปฏิบัติการ เช่น FreeBSD จะ support ทุกออพชั่นข้างต้นขณะที่ Linux 2.0.X จะ support บางออพชั่น Linux 2.1.x support ทุกออพชั่น

แม้ว่าหลายระบบปฏิบัติการจะ support ออพชั่นเดียวกันแต่บางที่เราก็สามารถแยกแยะได้โดยจาก ออพชั่น the_values_of เช่น ถ้าส่งค่า MSS เล็กๆไปที่ Linux โดยทั่วไปแล้วมันจะส่ง MSS echo กลับไป ส่วนระบบปฏิบัติการอื่นจะส่งค่าที่แตกต่างกันกลับไป แต่ถ้ายังได้ผลลัพธ์เหมือนกันอีกก็สังเกตลำดับของออพชั่นที่ได้รับมา เช่น Solaris จะเป็น 'NNTNWME' ซึ่งหมายความว่า <no op><no op><timestamp><no op><window scale><echoed MSS> ขณะที่ Linux 2.1.122 จะเป็น MENNTNW ซึ่งจะเห็นได้ว่าเป็นออพชั่นเดียวกัน ส่ง MSS echo เหมือนกัน แต่ลำดับที่ส่งมาไม่เหมือนกัน

- SYN Flood Resistance บางระบบปฏิบัติการจะไม่ยอมรับการ connection ครั้งใหม่ ถ้ามีการส่งแพ็กเก็ต SYN ไปมากเกินไป ซึ่งหลายระบบปฏิบัติการนั้นสามารถที่จะจัดการกับแพ็กเก็ตที่ส่งเข้ามาได้ไม่เกินครั้งละ 8 แพ็กเก็ต แต่ใน kernel ของ Linux รุ่นใหม่มีวิธีในการป้องกันปัญหานี้เช่น การใช้ SYN cookies ดังนั้นเราก็สามารถรู้ได้ว่าเป็นระบบปฏิบัติการอะไร โดยการส่งแพ็กเก็ตไป 8 แพ็กเก็ต ไปยังพอร์ตที่เปิดอยู่แล้วดูว่าสามารถ connect ไปยังพอร์ตนั้นได้หรือไม่

7.5 ตัวอย่างเครื่องมือที่ใช้ในการสำรวจระบบเครือข่าย

Pinger จากบริษัท Rhino9 โปรแกรมตัวนี้ทำงานโดยจะส่งแพ็กเก็ต ICMP ECHO ออกไปพร้อมๆ กันแบบขนานในลักษณะของ Round robin และรอคอยการตอบสนอง นอกจากนั้น Pinger ยังอนุญาตให้คุณค้นหาชื่อโฮสต์และบันทึกผลลัพธ์ที่ได้ลงในไฟล์ด้วย ส่วนเครื่องมือที่เป็นผลิตภัณฑ์ทางการค้าอีกตัวหนึ่งซึ่งทำงานได้รวดเร็วพอๆ กับ Fping หรืออาจจะเร็วกว่าด้วย ก็คือเครื่องมือจากบริษัท SolarWinds สาเหตุที่มันทำงานได้เร็วกว่าก็เพราะว่า มันอนุญาตให้กำหนดช่วงเวลาการรอคอย ระหว่างการส่งแพ็กเก็ต ICMP แต่ละครั้งด้วย ถ้าระบุ delay time ให้เป็น 0 หรือ 1 คุณจะสามารถสแกน IP Address ที่อยู่ ในคลาส C พร้อมทั้งแปลง IP Address เป็นชื่อโฮสต์ได้ในเวลาน้อยกว่า 7 วินาที

Network Mapper (nmap) โดย Fyodor นอกจาก NMAP จะให้ทั้งความสามารถพื้นฐานในการสแกนต่างๆที่ได้กล่าวไปข้างต้นทั้งหมด ซึ่งค่อนข้างหายากเหมือนกันสำหรับเครื่องมือตัวเดียวแต่ให้ได้ครบทุกเทคนิค

บทที่ 8

การออกแบบและพัฒนาโปรแกรม

8.1 รายละเอียดของการพัฒนา

ในการจัดทำโปรแกรมสำรวจและสังเคราะห์เครือข่ายและระบบคอมพิวเตอร์จำเป็นต้องศึกษา ทฤษฎีและข้อมูลที่ใช้ในการค้นหาและรวบรวมข้อมูลของคอมพิวเตอร์เครื่องอื่นที่อยู่ในเครือข่าย โดยใน โปรแกรมของเราใช้เทคนิคหลายๆอย่าง ดังนี้

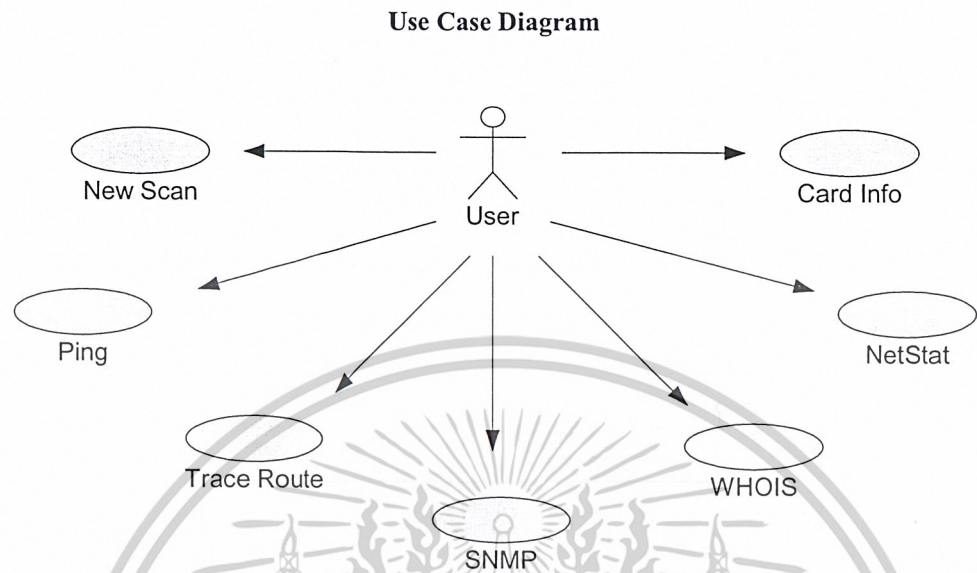
- ใช้เทคนิค Ping sweep เพื่อค้นหาว่ามีเครื่องไหนอยู่ในเครือข่ายบ้าง
- ใช้เทคนิค Trace route เพื่อสำรวจ Router
- ใช้เทคนิค Netstat โดยอาศัยการเปิด Service RIP Listener เพื่อหา Routing Table
- ใช้เทคนิค SNMP เพื่อ Query ข้อมูลของ Router
- ใช้ Whois เพื่อหา Network Address

ส่วนเครื่องมือต่าง ๆ ที่ใช้ในการพัฒนาได้แก่

- Microsoft Visual C++ 6.0 MSDN Library Visual Studio 6.0
- Winpcap 3.0
- Nmap 3.00
- Microsoft Windows XP

โดยรูปแบบของโปรแกรมจะมี Input เป็น IP address จะใส่แค่ค่าเดียวหรือ ใส่เป็นช่วงก็ได้ หรือ จะใส่เป็นจำนวน Hop ผลลัพธ์ที่ได้ออกมาจะแสดงในรูปแบบ Graphic โดยจะวาดเป็นการเชื่อมต่อ ว่ามีการเชื่อมต่อจากเครือข่ายใดไปยังเครือข่ายใด และเครือข่ายใดอยู่กลุ่มเดียวกัน แต่จะวาดโดยแบ่งตามเครือข่ายที่เราทำการสำรวจ โดยสามารถแสดงเครื่องคอมพิวเตอร์ทั้งหมดที่สำรวจได้ที่อยู่บนเครือข่ายที่ทำการสำรวจ และหากมีการเปิด SNMP Service ก็จะอ่านรายละเอียดอื่น ๆ เพิ่มเติมได้เช่น CPU ที่ใช้เป็น รูนไหน มีจำนวน Interface ที่ติดต่อยู่บนเครือข่ายเป็นเท่าไร เป็นต้น

8.2 การออกแบบโครงสร้างของโปรแกรม

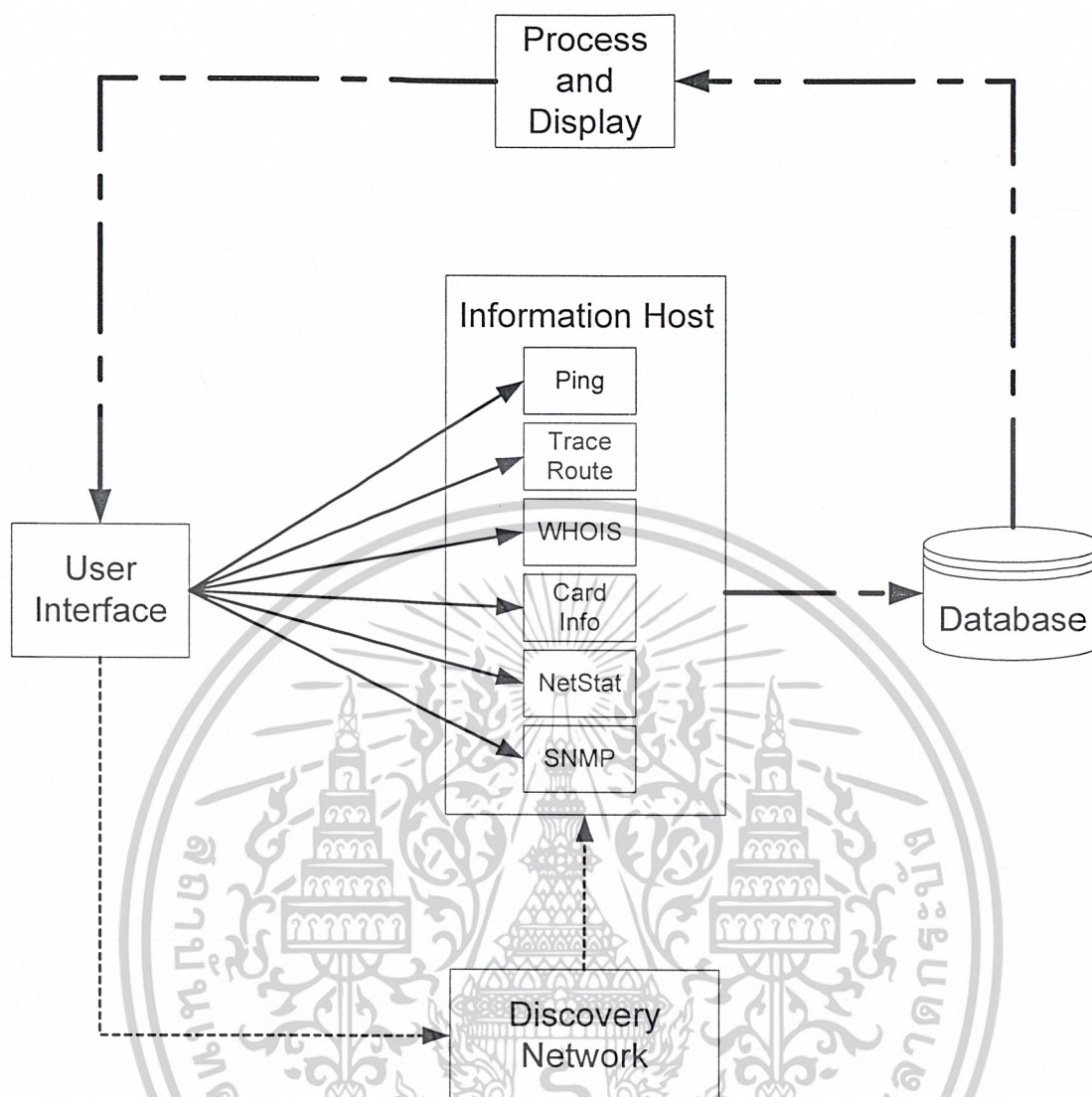


รูปที่ 8-1 แสดง Use Case ของโปรแกรม

คำศัพท์	คำอธิบาย
User	ผู้ใช้โปรแกรมที่ต้องการทำการสำรวจระบบเครือข่าย เช่น ผู้ดูแลระบบ
New Scan	ใช้โปรแกรมทำการสร้างแผนภาพของระบบเครือข่ายรวมโดยผู้ใช้ใส่ข้อมูลที่จำเป็นในการใช้สำรวจ
Ping	สั่งให้โปรแกรม Ping ไปยัง IP ที่สนใจ
Trace Route	สั่งให้โปรแกรมทำการค้นหาเส้นทางไปยัง IP ที่สนใจ
SNMP	สั่งให้โปรแกรมทำการอ่านข้อมูลจาก SNMP Service ของคอมพิวเตอร์ในระบบ จาก IP ที่สนใจ
WHOIS	สั่งให้โปรแกรมให้ทำการค้นหา Network Address ของ IP ที่สนใจ
NetStat	สั่งให้โปรแกรมทำการอ่านค่า Connection ต่างๆ หรือ อ่านค่า Routing Table ของตนเอง
Card Info	สั่งให้โปรแกรมทำการอ่านรายละเอียดจาก Network Card ตนเอง

ตารางที่ 8-1 แสดงคำอธิบาย Use Case Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-2 แสดงโครงสร้างโปรแกรม

ส่วนหลัก ๆ ของโปรแกรมแบ่งเป็น 5 ส่วนหลักๆ

- User Interface ทำหน้าที่รับและแสดงผลติดต่อกับผู้ใช้
- Information Host ทำหน้าที่ค้นหาข้อมูลของเครื่องที่ผู้ใช้สนใจ
- Discovery Network ทำหน้าที่ค้นหาเครือข่าย
- Database ทำหน้าที่เก็บข้อมูลที่ค้นหาได้
- Process and Display ทำหน้าที่วิเคราะห์ข้อมูลที่เก็บไว้ในส่วนเก็บข้อมูลและ นำมาสร้างแผนภาพของระบบเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถการทำงานของโปรแกรมแบ่งเป็น 2 ส่วนหลักๆ

1. Information Host (โดยมีฟังก์ชันภายในดังนี้)

- Ping
- Trace Route
- WHOIS
- Card Info
- NetStat
- SNMP

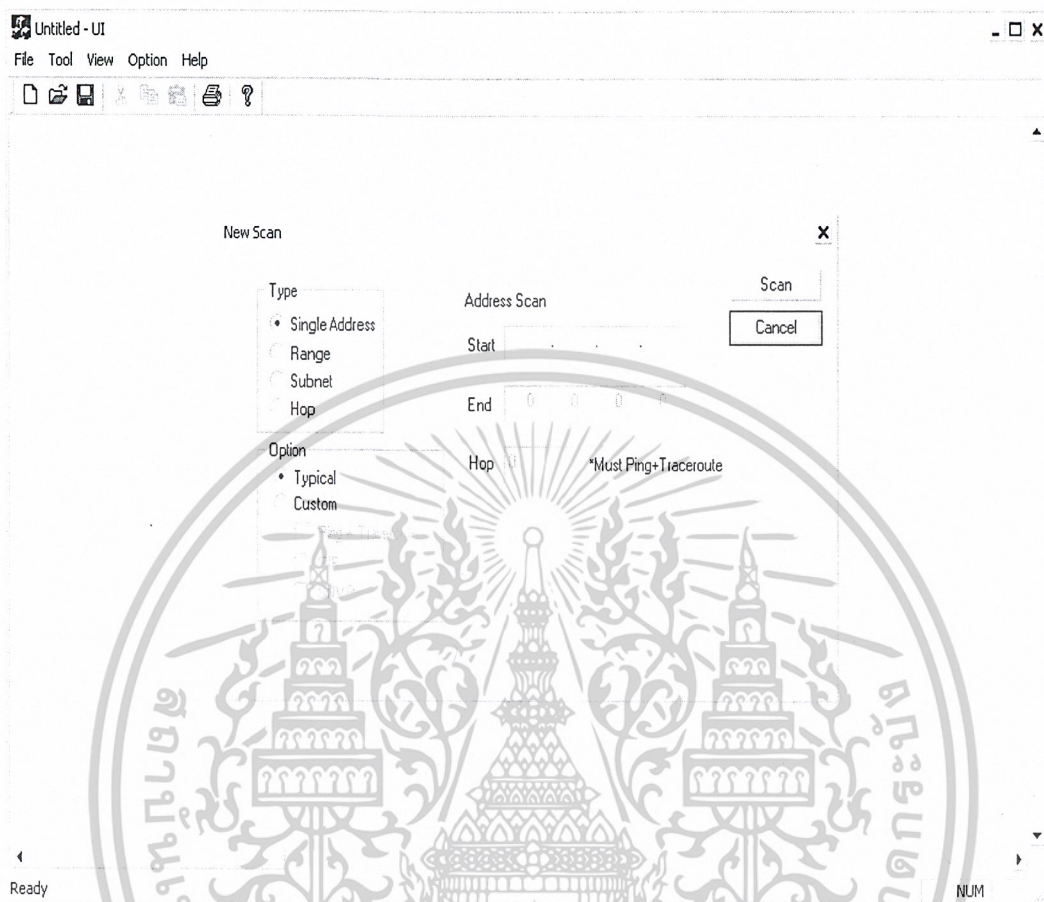
ทั้งหมดนี้จะทำการค้นหาข้อมูลของเครื่องที่ผู้ใช้สนใจและนำข้อมูลนั้นมาแสดงผล โดยไม่ได้มีการเก็บข้อมูลลงฐานข้อมูลแต่อย่างใด

2. Discovery Network จะทำงานโดยเรียกใช้ฟังก์ชันต่างๆบนส่วน Information Host และเมื่อเรียกใช้ฟังก์ชันใดฟังก์ชันหนึ่งเสร็จก็จะเก็บลงฐานข้อมูล และเรียกใช้ฟังก์ชันอื่นต่อไป (ตามแต่ผู้ใช้งานจะเลือกใช้งานฟังก์ชันใดบ้าง โดยค่าเริ่มต้นจะเรียกทุกฟังก์ชัน) หลังจากเก็บข้อมูลทั้งหมดที่หามาได้ ก็จะนำข้อมูลเหล่านั้นไปผ่านส่วนวิเคราะห์ข้อมูลและนำมาสร้างแผนภาพเพื่อแสดงผลให้กับผู้ใช้งาน

โดยมีขั้นตอนการทำงานดังนี้

- Ping Sweep ไปยัง IP ที่สนใจและเก็บสถานะของ IP เหล่านั้นลงฐานข้อมูล
- Trace Route ไปยัง IP ที่มีสถานะ ON อยู่ในขณะนั้น โดยตรวจสอบจากฐานข้อมูล
ถ้าไม่มีข้อมูลก็จะ Trace Route ไปทุก IP ที่ป้อนเข้ามา และเก็บลงฐานข้อมูล
- Query SNMP ไปยัง IP ที่มีสถานะ ON อยู่ในขณะนั้น โดยตรวจสอบจากฐานข้อมูล
ถ้าไม่มีข้อมูลก็จะ Query SNMP ไปทุก IP ที่ป้อนเข้ามา และเก็บลงฐานข้อมูล
- Query NetStat จากเครื่องที่ผู้ใช้งาน
- นำข้อมูลที่เก็บลงฐานข้อมูลมาประมวลผลเป็นภาพ

8.3 ตัวอย่างส่วนติดต่อกับผู้ใช้

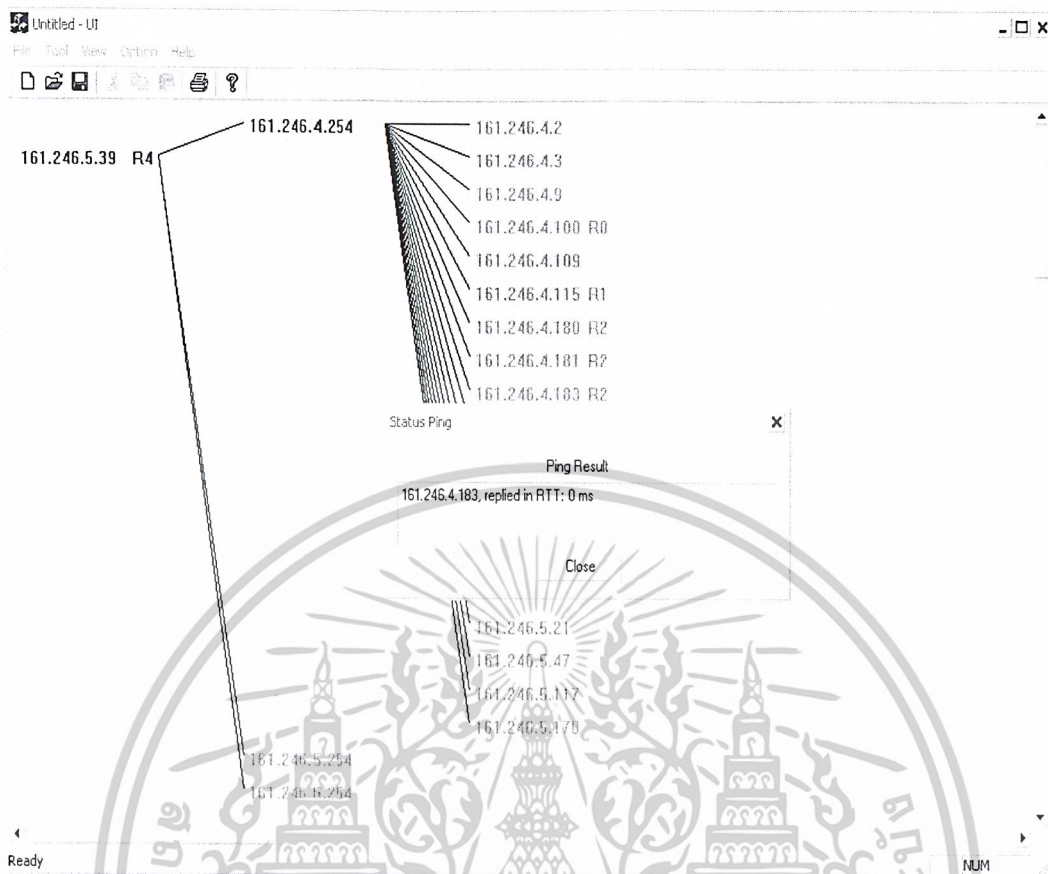


รูปที่ 8-3 ตัวอย่างต้นแบบหน้าจอที่ใช้ในการใส่ Input ของโปรแกรม

Input ที่ต้องการของโปรแกรม

- Single Address ของคอมพิวเตอร์ที่ต้องการสำรวจกรณี โดยรู้เพียง IP ของเครื่องนั้น
- Range ของคอมพิวเตอร์ที่ต้องการสำรวจระบบเครือข่ายคอมพิวเตอร์ โดยทราบเป็นช่วงของคอมพิวเตอร์ในระบบเครือข่าย
- Subnet โดยต้องใส่ IP Address และ Subnet Mark ของกลุ่มเครือข่ายคอมพิวเตอร์ที่ต้องการทำการสำรวจ
- Hop โดยใส่เป็นจำนวนของ Hop Count ที่ต้องการสำรวจระบบเครือข่าย ว่ารูปแบบของเครือข่ายภายในจำนวนที่ป้อนไว้เป็นอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-4 ตัวอย่างต้นแบบหน้าจอ แผนภาพที่ได้จากการสำรวจ

Output ของโปรแกรมที่ได้จากการสำรวจ

- แผนภาพทางกายภาพของระบบเครือข่ายที่สำรวจมาได้ จากเครือข่ายตัวอย่างที่สนใจ (161.246.4.0 - 161.246.6.0) โดยจะแสดงเป็น โหนดหรือรายชื่อเครื่องที่ยังมีชีวิตภายใน โหนดนั้นก็ได้
- ถ้าเครื่องหรือโหนดใด มีการเปิด SNMP Service ก็จะมีการแสดงตัวอักษรท้าย IP Address นั้น
- โดยสามารถตรวจสอบสถานะของเครื่องที่แสดงได้ว่ายังมีชีวิตอยู่หรือหายไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

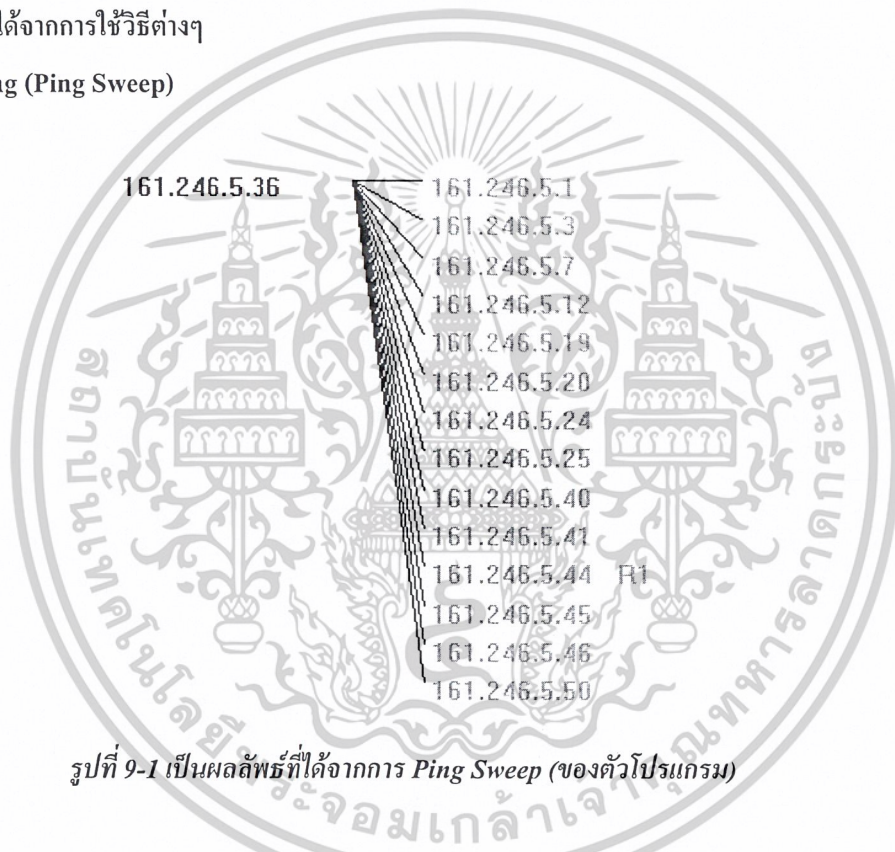
บทที่ 9

การทำงานของโปรแกรมสำรวจเครือข่าย

มีการทำงานโดยจะสำรวจเครือข่ายว่าเครือข่ายมีรูปแบบการเชื่อมต่อแบบใด โดยใช้วิธีการ Ping (Ping Sweep), Trace Route, Whois, RIP (Listener), SNMP ในการเก็บข้อมูล เพื่อมาประมวลผลและตรวจสอบการเชื่อมต่อ ซึ่งนำมาแสดงเป็นรูปภาพได้

9.1 ข้อมูลที่ได้จากการใช้วิธีต่างๆ

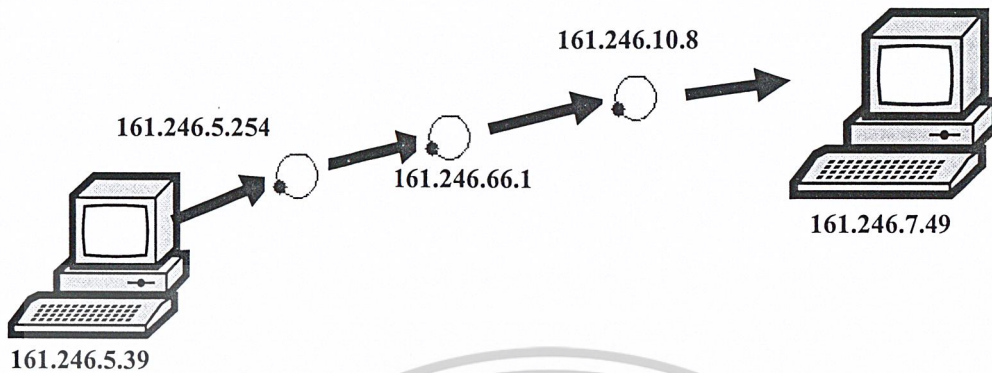
Ping (Ping Sweep)



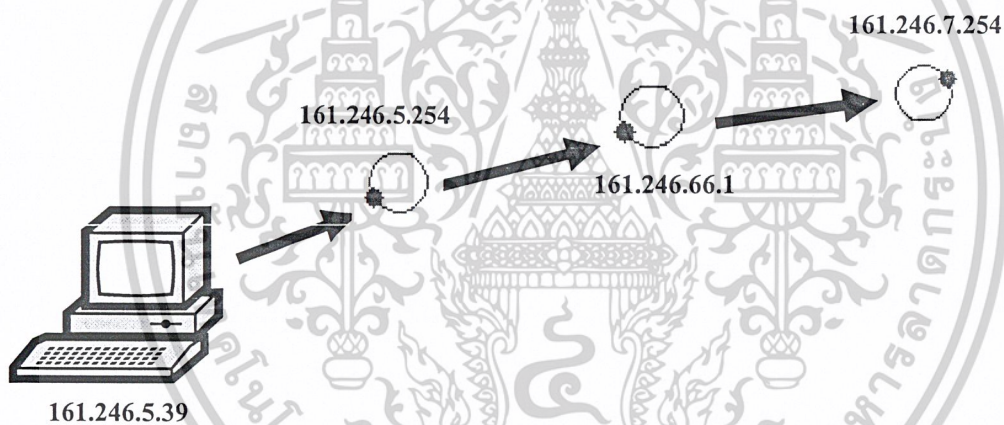
รูปที่ 9-1 เป็นผลลัพธ์ที่ได้จากการ Ping Sweep (ของตัวโปรแกรม)

จากรูปจะแสดงให้เห็นว่าไอพีใดบ้างในเครือข่ายที่ออนไลน์อยู่ โดยจะเก็บเฉพาะตัวไอพีที่ออนไลน์ไว้และสถานะนั้น ถ้าเกิดมีการตรวจสอบอีกครั้งแล้วไอพีนั้นออฟไลน์ไปก็จะเปลี่ยนสถานะของไอพีนั้นด้วย

Trace Route



รูปที่ 9-2 เป็นผลลัพธ์ที่ได้จาก Trace Route ไปยัง Host



รูปที่ 9-3 เป็นผลลัพธ์ที่ได้จาก Trace Route ไปยัง Router หรือ Gateway

จากรูปจะแสดงให้เห็นว่าผลของการ Trace Route ไปยังไอพี หรือแต่ละโหนดจะให้ผลลัพธ์ที่ต่างกันอยู่บ้าง คือถ้า Trace Route ไปยัง Host ภายในเครือข่ายนั้นๆ ผลที่ได้จะไม่แสดงไอพีของ Gateway ของเครือข่ายนั้น แต่ในทางกลับกันถ้า Trace Route ไปยัง

Whois

OrgName: CommuniTech.Net, Inc.
OrgID: CMTK
Address: 1627 Main Street Suite 1100
City: Kansas City
StateProv: MO
PostalCode: 64108
Country: US
NetRange: 209.15.0.0 - 209.15.255.255
CIDR: 209.15.0.0/16
NetName: COMMUNITECHNET
NetHandle: NET-209-15-0-0-1
Parent: NET-209-0-0-0-0
NetType: Direct Allocation
NameServer: NS.HOSTING4U.NET
NameServer: NS2.HOSTING4U.NET
NameServer: NS4.HOSTING4U.NET
NameServer: NS3.HOSTING4U.NET
Comment: ADDRESSES WITHIN THIS
 BLOCK ARE NON-PORTABLE
RegDate: 1999-03-17
Updated: 2001-09-26
TechHandle: IC117-ARIN
TechName: CommuniTech.Net, Inc.
TechPhone: +1-877-722-8662
TechEmail: noc@hosting4u.net

รูปที่ 9-4 เป็นการสอบถามข้อมูลไอพี 209.15.15.15 จาก Whois.internic.net

โดยผลลัพธ์ที่ได้จะนำเพียงบางส่วนมาใช้งาน คือ เนตเวิร์กแอดเดรส หรือช่วงของไอพีใน
 เนตเวิร์กที่เราสนใจ เพื่อช่วยในการหาค่าเน็ตเวิร์กแอดเดรสจากไอพีที่รับเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RIP (Listener)

Active Routes:					
Network	Destination	Netmask	Gateway	Interface	Metric
0.0.0.0		0.0.0.0	161.246.5.254	161.246.5.39	20
127.0.0.0		255.0.0.0	127.0.0.1	127.0.0.1	1
161.246.4.0		255.255.255.0	161.246.5.254	161.246.5.39	2
161.246.5.0		255.255.255.0	161.246.5.39	161.246.5.39	20
161.246.5.39		255.255.255.255	127.0.0.1	127.0.0.1	20
161.246.6.0		255.255.255.0	161.246.5.254	161.246.5.39	2
161.246.66.0		255.255.255.0	161.246.5.254	161.246.5.39	2
161.246.70.0		255.255.255.0	161.246.5.254	161.246.5.39	2
161.246.255.255		255.255.255.255	161.246.5.39	161.246.5.39	20
224.0.0.0		240.0.0.0	161.246.5.39	161.246.5.39	20
255.255.255.255		255.255.255.255	161.246.5.39	161.246.5.39	1
Default Gateway:		161.246.5.254			

รูปที่ 9-5 เราดิ่งเทเบิลภายในเครื่อง จากการ *Netstat -nr*

เป็นการดิ่งข้อมูลเราดิ่งเทเบิลจากเครื่องของผู้ใช้งาน โดยที่ผู้ใช้งานควรเปิด RIP Listener Service เพื่อจะได้ให้เครื่องของตนมีการรับเราดิ่งอินฟอรม์เมชันจากเราเตอร์ใกล้เคียง และจะได้นำข้อมูลเราดิ่งเทเบิลนั้นมาใช้งานให้เกิดประโยชน์เพิ่มขึ้น

SNMP

เป็นสอบถามข้อมูลจากอุปกรณ์หรือเครื่องที่เปิด SNMP Service และใช้ Community Name เป็น public (เพราะ Community Name เปรียบเสมือน Password ที่ใช้ตรวจสอบการเข้าไปใช้บริการนั้น โดย public จะเป็นค่าเริ่มต้น แต่ค่านี้ถูกเปลี่ยนแล้วเราไม่ทราบก็ไม่สามารถเข้าไปใช้บริการได้) ส่วนออบเจก ที่สนใจคือ ipNetToMediaTable, ipRoutingTable, ipAddrTable เนื่องจากที่เลือกมาเพียง 3 กลุ่ม เพราะตัว SNMP เองมีออบเจกเป็นจำนวนมาก และที่ทั้งส่วนที่เกี่ยวข้องและไม่เกี่ยวข้องกับข้อมูลที่ต้องการใช้

Index	3.161.246.5.254
ipNetToMediaIndex	3
ipNetToMediaPhysAddress	0008.e277.8409
ipNetToMediaNetAddress	161.246.5.254
ipNetToMediaType	dynamic(3)

รูปที่ 9-6 ipNetToMediaTable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Index:	161.246.5.0	161.246.6.0
ipRouteDest	161.246.5.0	161.246.6.0
ipRouteIfIndex	0	1
ipRouteMetric1	0	0
ipRouteMetric2	-1	-1
ipRouteMetric3	-1	-1
ipRouteMetric4	-1	-1
ipRouteNextHop	161.246.6.254	161.246.6.136
ipRouteType	indirect(4)	direct(3)
ipRouteProto	local(2)	local(2)
ipRouteAge	3	0
ipRouteMask	255.255.255.0	255.255.255.0
ipRouteMetric5	-1	-1
ipRouteInfo		

รูปที่ 9-7 ipRoutingTable

Index:	161.246.6.136	192.168.0.2
ipAdEntAddr	161.246.6.136	192.168.0.2
ipAdEntIfIndex	1	2
ipAdEntNetMask	255.255.255.0	255.255.0.0
ipAdEntBroadcastAddr	1	1
ipAdEntReasmMaxSize	18024	18024

รูปที่ 9-8 ipAddrTable

9.2 การสำรวจเครือข่าย

รูปที่ 9-9 แสดงการเลือก New Scan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Type

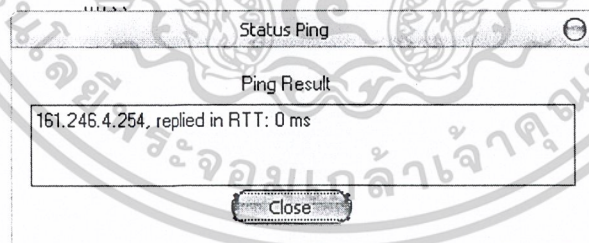
- Single Address จะเป็นการใส่ไอพีเดียวลงไป และจะทำการ WHOIS ดูว่าไอพีดังกล่าวมี Network Address เป็นอย่างไร แล้วจะเอาช่วงไอพีนั้นมาทำการสำรวจ
- Range Address จะเป็นการใส่ช่วงไอพีที่ต้องการสำรวจ
- Subnet Address จะเป็นการหาไอพีในช่วงซับเน็ตนั้น
- Hop จะทำการสำรวจรอบๆ ตัวเราตามจำนวน Hop ที่ใส่ไปว่ามีไอพีอะไรอยู่บ้าง (จะมีช่วงไอพีไม่เกินหนึ่งคลาส B)

Option

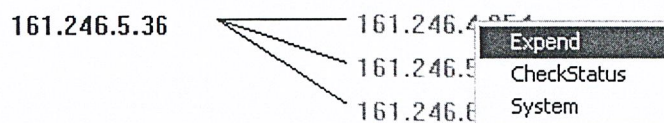
- Typical จะใช้วิธีการทั้งหมดในการสำรวจ คือ Ping, Trace Route, RIP และ SNMP
- Custom จะเลือกจะใช้วิธีการใดบ้างในการสำรวจเครือข่าย

9.3 การตรวจสอบเครื่อง

- จะสามารถตรวจสอบได้ว่าเครื่องนั้นมีการทำงานอยู่หรือไม่ และนำค่าที่ได้ไปอัปเดตข้อมูลแสดงเป็นภาพออกมา โดยที่สีแดงคือไม่ได้มีการทำงานอยู่ (รูปที่ 9-10)
- สามารถขยายเพื่อดูว่ามีเครื่องไหนทำงานอยู่บ้าง (รูปที่ 9-11) หรือย่อเน็ตเวิร์คได้ในกรณีที่ไม่ต้องการที่จะดูเครื่องภายใน (รูปที่ 9-12)
- สามารถบอก System อย่างคร่าวๆ ได้จากการ SNMP Query (รูปที่ 9-13)

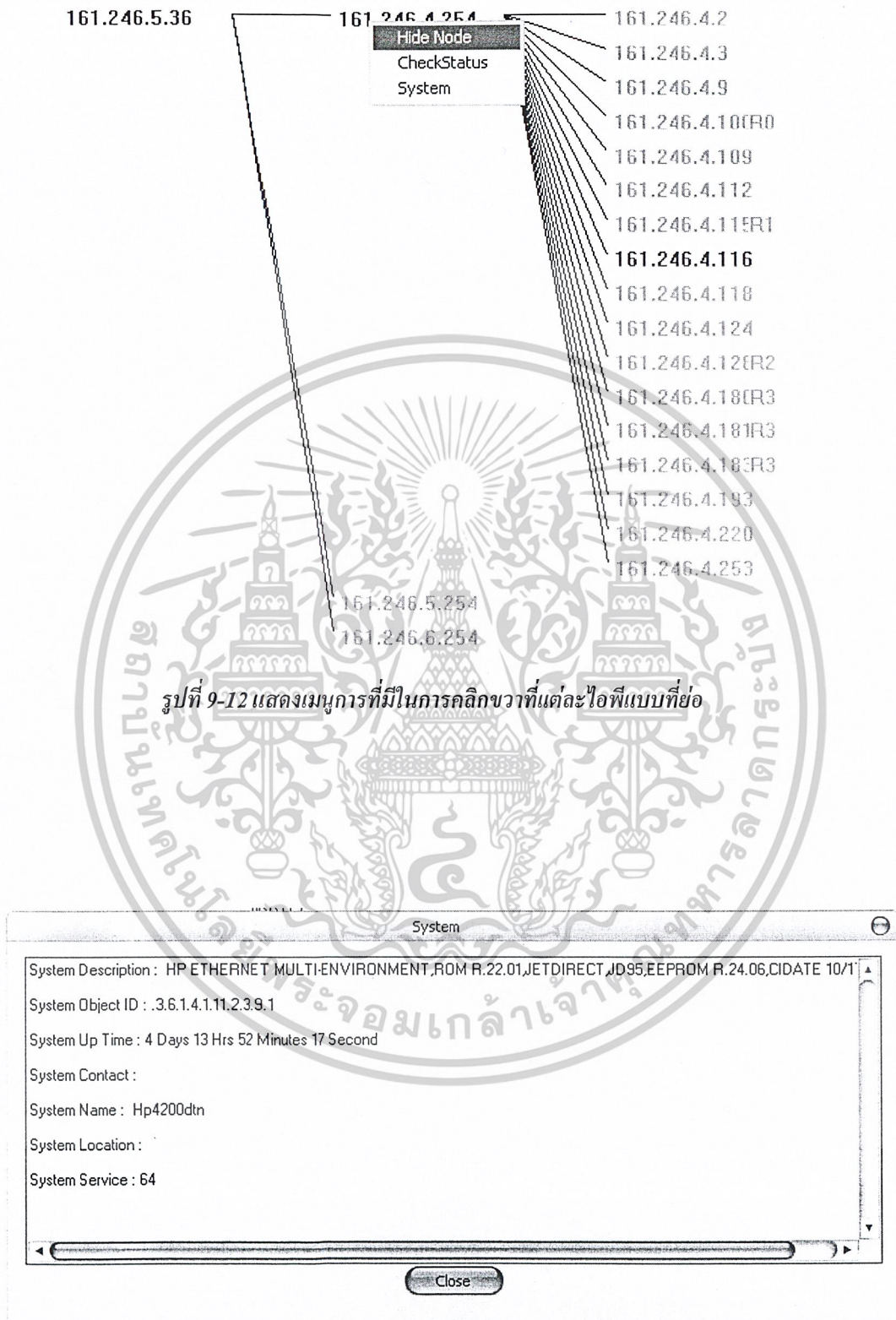


รูปที่ 9-10 แสดงว่าเครื่องยังมีการทำงานอยู่



รูปที่ 9-11 แสดงเมนูการที่มีในการคลิกขวาที่แต่ละไอพีแบบที่เป็นขยาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



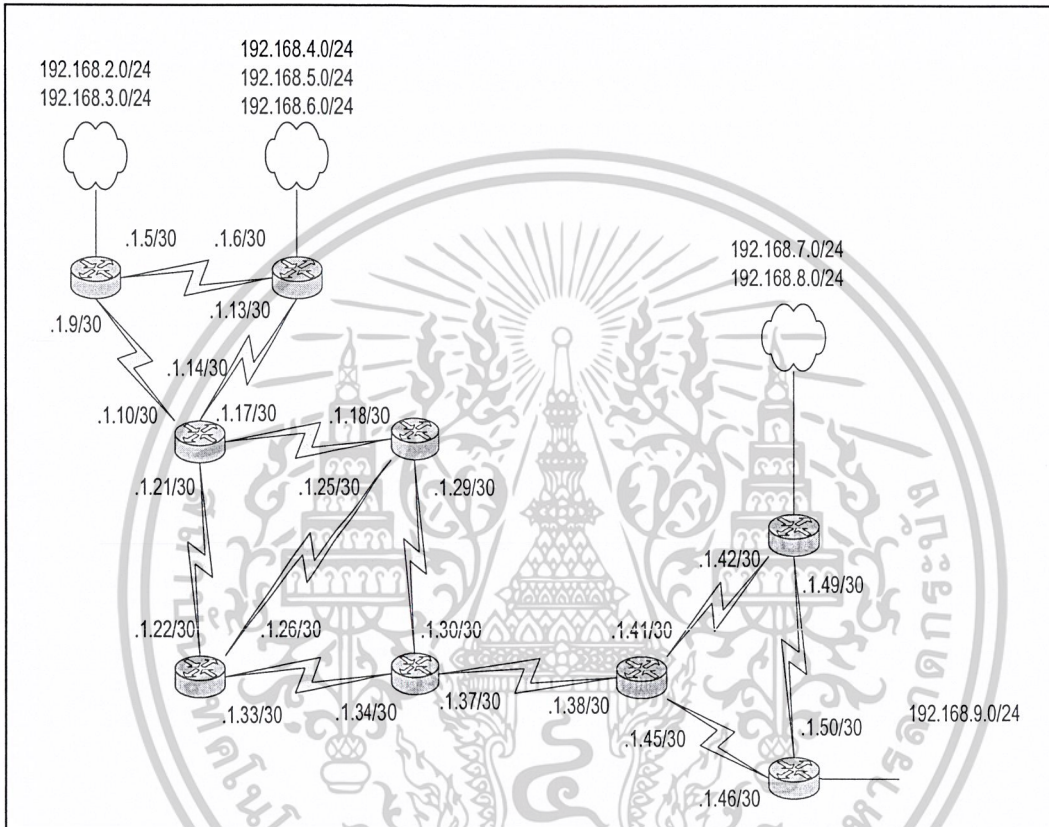
รูปที่ 9-13 แสดง System อย่างคร่าวๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10

วิเคราะห์ผลการทดลองและสรุป

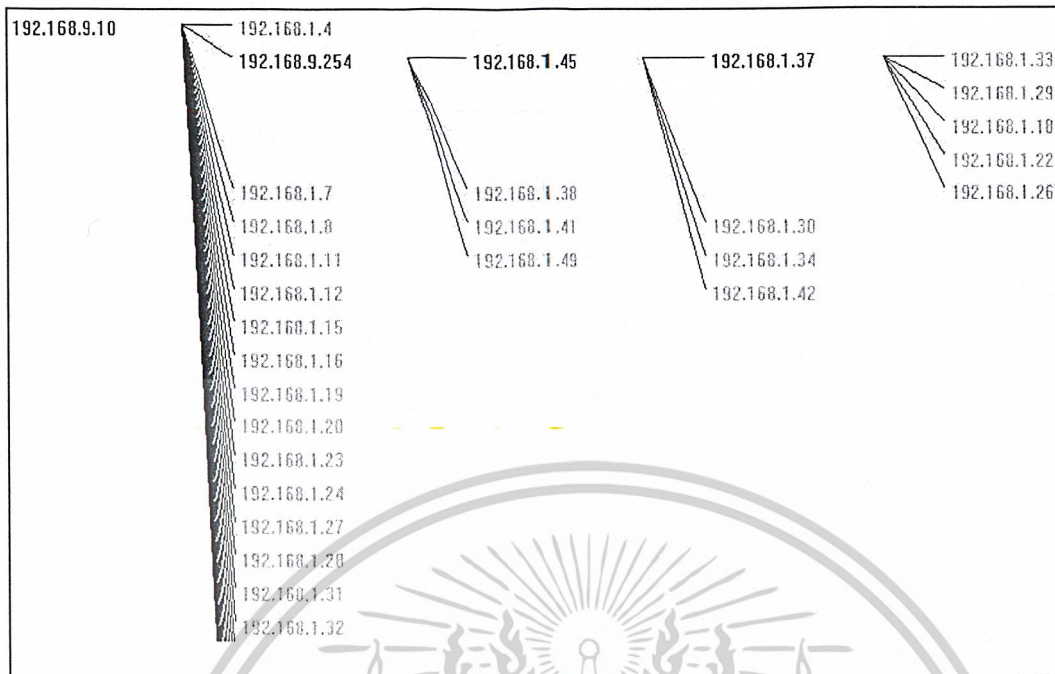
10.1 วิเคราะห์ผลการทดลอง



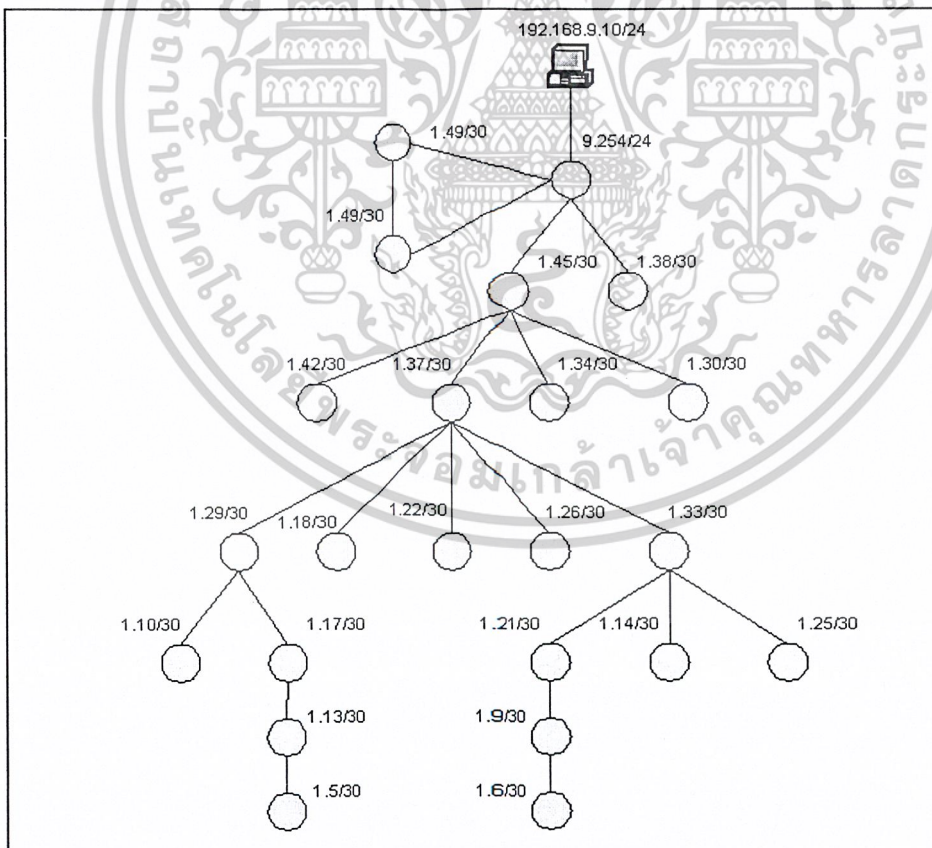
รูปที่ 10-1 ภาพตัวอย่างเครือข่ายที่ต้องการที่จะสำรวจ

10.2 การทดลองด้วยวิธีการ Ping และ Trace Route

จากรูปผลลัพธ์ของโปรแกรม และแผนภาพที่ถูกสร้างขึ้น (รูปที่ 10-2 และ 10-3 ตามลำดับ) จะแสดงให้เห็นว่าได้แผนภาพออกมาในลักษณะของต้นไม้ (Tree) โดยที่แตกต่างจากรูปของเครือข่ายจริงค่อนข้างมาก เนื่องจากการใช้เพียงวิธี Ping และ Trace Route จึงทำให้ได้ข้อมูลที่จำเป็นต่อการสร้างภาพไม่เพียงพอ



รูปที่ 10-2 ภาพผลลัพธ์บางส่วนของโปรแกรมจากการสำรวจโดยเลือกใช้วิธีการ Ping + Trace Route

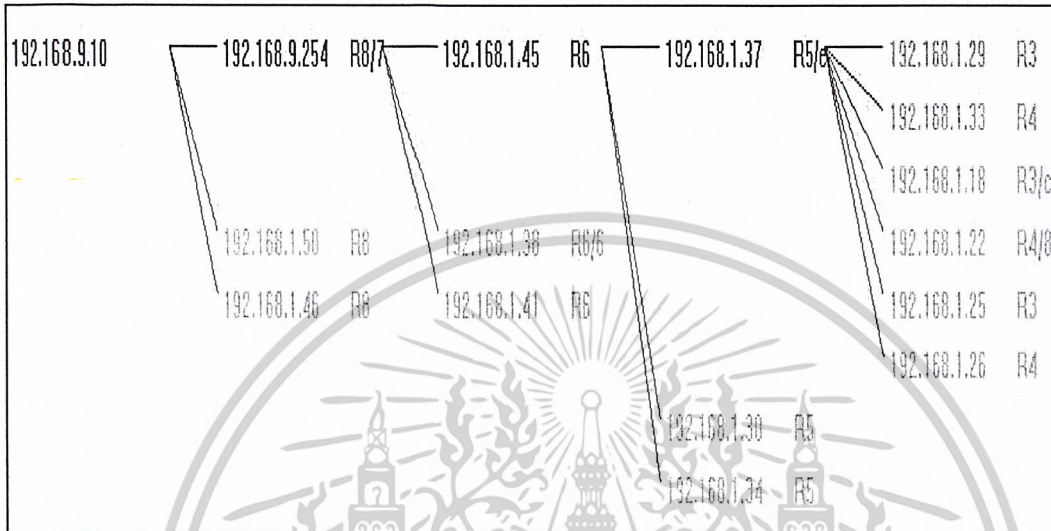


รูปที่ 10-3 ภาพจากการนำผลลัพธ์ที่ได้มาวาดเป็นแผนภาพ (Ping + Trace Route)

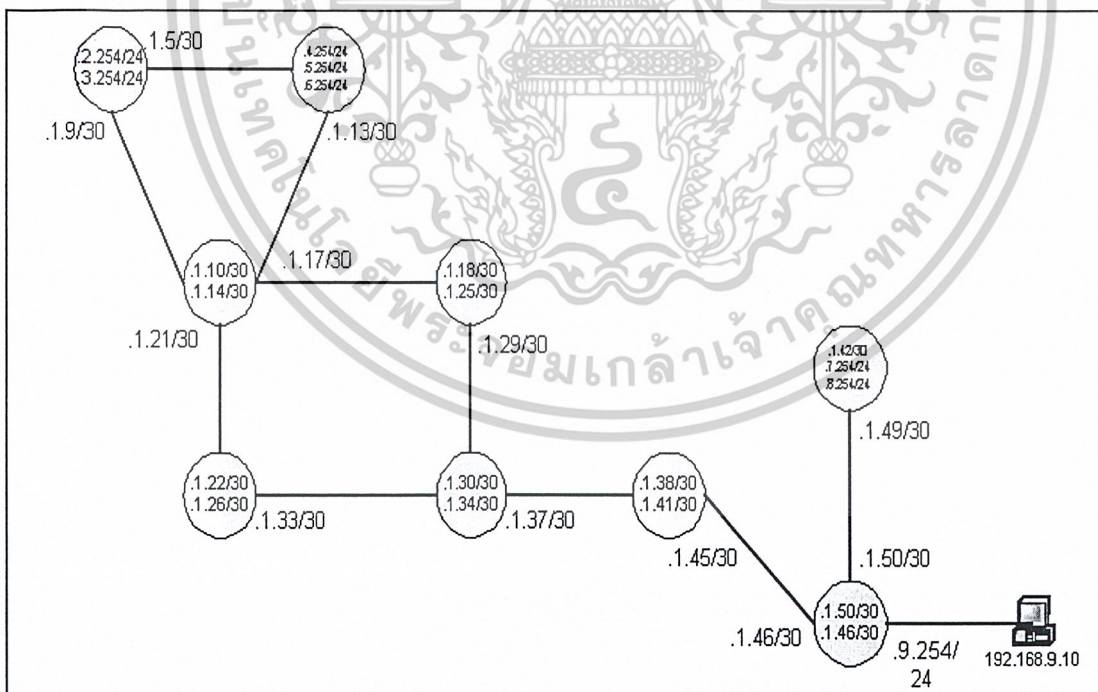
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.3 การทดลองด้วยวิธีการทั้งหมด (Typical)

จากรูปผลลัพธ์ของโปรแกรมและแผนภาพที่ถูกสร้างขึ้น (รูปที่ 10-4 และ 10-5 ตามลำดับ) จะแสดงให้เห็นว่าได้แผนภาพที่ออกมาค่อนข้างใกล้เคียงกับเครือข่ายตัวอย่าง เนื่องจากการใช้วิธีการทั้งหมดที่มี ทำให้ได้ข้อมูลมากกว่าวิธีการแรกจึงช่วยในการสร้างภาพที่ได้ใกล้เคียงขึ้นดังต่อไปนี้



รูปที่ 10-4 ภาพผลลัพธ์บางส่วนของโปรแกรมจากการสำรวจโดยเลือกใช้วิธีการทั้งหมด (Typical)



รูปที่ 10-5 ภาพจากการนำผลลัพธ์ที่ได้มาวาดเป็นแผนภาพ (Typical)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสามารถทำงานได้ตามวัตถุประสงค์ดังนี้

- สามารถหาว่าเครือข่ายมีการเชื่อมต่อแบบใด โดยสามารถที่จะทำการขยายเพื่อดูได้ง่ายยิ่งขึ้น
- สามารถตรวจสอบได้ว่ามีเครื่องไหนมีการทำงานอยู่บ้าง
- สามารถตรวจสอบระบบเครื่องจากการ SNMP ได้

10.4 สรุปผล

- โปรแกรมสามารถทำงานได้ตามวัตถุประสงค์ที่วางไว้
- สามารถช่วยทำให้เราทราบว่าเครือข่ายมีรูปแบบใด เช่นว่ามีการเชื่อมต่อจากไอพีใดไปยังไอพีใด
- ข้อจำกัดของโปรแกรม
 1. พบปัญหาในการสำรวจเนื่องจากเครื่องผู้ใช้ในการติดตั้ง Firewall ไว้ทำให้ไม่สามารถรู้ได้ว่าเครื่องนั้นเปิดอยู่หรือไม่
 2. มีการ Block ICMP ทำให้เราไม่สามารถที่จะสำรวจเครือข่ายนั้นได้
 3. การตรวจสอบเครือข่ายด้วย SNMP ไม่สามารถที่จะตรวจสอบได้ทุกเครื่องเนื่องจากไม่ได้มีการเปิด Service ไว้ทุกเครื่อง ทำให้เก็บข้อมูลได้ไม่เพียงพอ
 4. เครื่องผู้ใช้งานต้องเปิด RIP Listener

10.5 แนวทางในการพัฒนาสำหรับผู้สนใจในอนาคต

- พัฒนาส่วนตรวจสอบโฮสต์ เช่น เทคนิคนอกเหนือจากการใช้ Ping
- พัฒนาส่วนที่จะช่วยตรวจสอบเครื่องในเครือข่ายว่ามีเปิดพอร์ต อะไรบ้าง มีการ Share Folder อะไรบ้าง และเพิ่มเติมในส่วนว่าเครื่องใดมีความเสี่ยงที่จะติดไวรัส หรือเปิดพอร์ตที่น่าสงสัยไว้
- การพัฒนาส่วนประมวลผลให้มีความเร็วในการทำงานมากขึ้นกว่าเดิม
- พัฒนาส่วนแสดงผลให้เป็น Graphic ที่ใช้และสวยงามกว่าเดิม

บรรณานุกรม

- [1] ยุทธนา ติลาศวัฒนกุล: “คู่มือการเขียนโปรแกรม และใช้งาน Visual C++ 6.0 ฉบับโปรแกรมเมอร์”
อินโฟเพรส, 2001
- [2] สุรศักดิ์ สงวนพงษ์: “สถาปัตยกรรมและโปรโตคอลที่ซีพี/ไอพี”, ซีเอ็ด, 2002
- [3] สุวัฒน์ ปุณณชัยยะ, ตัน ตันท์สุทริวงศ์, สุพจน์ ปุณณชัยชนะ, “เปิดโลกของ TCP/IP และโปรโตคอล
ของอินเทอร์เน็ต”, โปรวิชั่น, 2543
- [4] William Stallings: “SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, Addison-Wesley, 1999
- [5] Larry J. Hughes, Jr., “Internet Security Techniques”, New Riders Publishing, 1991, pp. 130-
144, 170-174, 192-207
- [6] ANTHONY JONES: “NETWORK PROGRAMMING For MICROSOFT WINDOWS”
SECOND EDITION

เว็บไซต์อ้างอิง

1. <http://winpcap.polito.it/>
2. <http://www.codeguru.com>
3. <http://www.codeproject.com>
4. http://www.cs.cornell.edu/cnrg/topology_aware/topology/topology.html
5. <http://vc4you.hypermart.net/>
6. <http://www.naughter.com/>
7. <http://www.cs.cornell.edu/boom/1999sp/projects/network%20topology/topology.html#Obtaining-source>
8. <http://www.janenet.com/sourcecode/code.asp?group=5>
9. <http://sern.ucalgary.ca/courses/SENG/611/F99/ConceptMaps/ConceptMaps.html>
10. <http://tangentsoft.net/wskfaq/>
11. <http://www.caida.org/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้