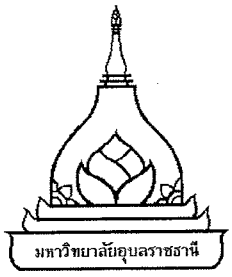
การจัดช่องทางการสื่อสารแบบสมดุล

ปกรณ์ กัลปดี

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์
มหาวิทยาลัยอุบลราชธานี
พ.ศ. 2551

ลิขสิทธิ์เป็นของมหาวิทยาลัยอุบลราชธานี



NETWORK LOAD BALANCING

PAKORN KALLAPADEE

AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

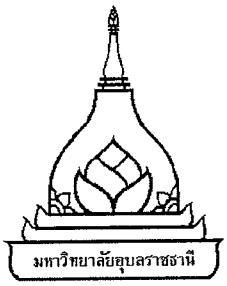
MAJOR IN INFORMATION TECHNOLOGY

FACULTY OF SCIENCE

UBON RAJATHANEE UNIVERSITY

YEAR 2008

COPYRIGHT OF UBON RAJATHANEE UNIVERSITY



ใบรับรองการค้นคว้าอิสระ

มหาวิทยาลัยอุบลราชธานี

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์

เรื่อง การจัดช่องทางการสื่อสารแบบสมดุล

ผู้วิจัย นายปกรณ์ กัลปดี

ได้พิจารณาเห็นชอบโดย

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.จิรดา แฮร์บส์ท)

..... กรรมการ

(ดร.ลักษณ์ เจริญวัฒนา)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ สุวัฒน์ บรรลือ)

..... คณบดี

(ผู้ช่วยศาสตราจารย์ ดร.จันทร์เพ็ญ อินทรประเสริฐ)

มหาวิทยาลัยอุบลราชธานี รับรองแล้ว

.....

(ผู้ช่วยศาสตราจารย์ ดร.อุทิศ อินทร์ประสิทธิ์)

รองอธิการบดีฝ่ายวิชาการ

ปฏิบัติราชการแทนอธิการบดี มหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2551

กิตติกรรมประกาศ

การค้นคว้าอิสระเรื่องการจัดช่องทางการสื่อสารแบบสมดุลครั้งนี้ สำเร็จลุล่วงไปด้วยดี ด้วยได้รับความกรุณาเป็นอย่างดียิ่งจากท่านอาจารย์ ผู้ช่วยศาสตราจารย์ ดร.จิรดา แฮร์บส์ท และอาจารย์ ดร.ลักษณ์ เจริญวัฒนา ที่ได้ให้คำแนะนำปรึกษาแนวทางในการแก้ไขปัญหาต่างๆ ผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอบคุณคณาจารย์ ทุกท่านถ่ายทอดวิชาความรู้ในด้านต่างๆ ให้สามารถนำมาประยุกต์และทำการค้นคว้าอิสระได้อย่างมีประสิทธิภาพ

ขอกราบขอบพระคุณ มารดา ที่เป็นกำลังใจช่วยส่งเสริมให้สามารถทำงานได้บรรลุจุดมุ่งหมาย

ขอขอบคุณเพื่อนร่วมงานที่อำนวยความสะดวก และช่วยเหลือให้กำลังใจตลอดมา และบุคคลอื่นที่เกี่ยวข้องที่ยังไม่ได้กล่าวนามในที่นี้

คุณค่าและประโยชน์ที่พึงได้จากการทำการค้นคว้าอิสระในครั้งนี้ ผู้จัดทำขออุทิศเพื่อบูชาพระคุณบิดา มารดา และครูอาจารย์ทุกท่าน

(นายปกรณ์ กัลปดี)
ผู้วิจัย

บทคัดย่อ

ชื่อเรื่อง : การจัดช่องทางการสื่อสารแบบสมดุล

โดย : ปกรณ์ กัลปดี

ชื่อปริญญา : วิทยาศาสตรมหาบัณฑิต

สาขาวิชา : เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.จิรดา แฮร์บส์ท

คำสำคัญ : การจัดช่องทางการสื่อสารแบบสมดุล ช่องทางการสื่อสารไดนามิคเวดจ์ราวด์-โรบิน

เนื่องจากในปัจจุบันองค์กรหรือหน่วยงานต่าง ๆ มีความต้องการใช้งานเครือข่ายอินเตอร์เน็ตเพิ่มมากขึ้น จึงต้องมีการเชื่อมต่อเครือข่ายภายในเข้ากับเครือข่ายอินเตอร์เน็ตโดยการใช้ช่องทางการสื่อสารหลายช่องทาง ซึ่งอาจเกิดปัญหาในกรณีที่ช่องทางการสื่อสารหนึ่งมีปริมาณการจราจรหนาแน่น ในขณะที่ช่องทางการสื่อสารอื่นๆ มีปริมาณการจราจรเพียงเล็กน้อย งานวิจัยนี้จึงนำเสนอการจัดช่องทางการสื่อสารแบบสมดุล ซึ่งมีความสามารถในการจัดปริมาณการจราจรและหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ขัดข้อง โดยพัฒนาซอฟต์แวร์ในการควบคุมอุปกรณ์เครือข่ายเพื่อกำหนดปริมาณการจราจรขาออกที่เหมาะสมให้กับช่องทางการสื่อสาร และการนำอัลกอริทึม Round-Robin, Weighted Round-Robin และ Dynamic Weighted Round-Robin มาประยุกต์ใช้ พร้อมกับทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมทั้งสามนี้ จากผลการทดลองพบว่าอัลกอริทึมสำหรับการจัดช่องทางการสื่อสารแบบสมดุลที่มีประสิทธิภาพมากที่สุดคือ Dynamic Weighted Round-Robin

ABSTRACT

TITLE : NETWORK LOAD BALANCING

BY : PAKORN KALLAPADEE

DEGREE : MASTER OF SCIENCE

MAJOR : INFORMATION TECHNOLOGY

CHAIR : ASST.PROF.JIRADA HERBST,Ph.D.

KEYWORDS : NETWORK LOAD BALANCING / GATEWAY / DYNAMIC WEIGHTED ROUND-ROBIN

Nowadays, the demand for internet access has increased in almost every organizations. The connections to internet via multi-channel communication can be used to facilitate these high demands. This may cause a problem when one channel is overloaded meanwhile the others have only small amount of traffic. This research purposed the network load balancing strategy which could balance the traffic on the network and also avoid the inefficient communication channel. We developed a software to control the network equipment which in turn direct the out-going traffic to the suitable communication channel. We implemented and compared three network load balancing algorithms: Round-Robin, Weighted Round-Robin and Dynamic Weighted Round-Robin. The empirical results showed that the Dynamic Weighted Round-Robin was the most effective algorithm.

สารบัญ

	หน้า
กิตติกรรมประกาศ	**ก**
บทคัดย่อภาษาไทย	**ข**
บทคัดย่อภาษาอังกฤษ	**ค**
สารบัญตาราง	**ฉ**
สารบัญภาพ	**ช**
บทที่	
1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	3
1.4 ประโยชน์ที่ได้รับจากงานวิจัย	3
1.5 สถานที่ทำวิจัย	4
2 เอกสารและงานวิจัยที่เกี่ยวข้อง	
2.1 ความรู้พื้นฐานเกี่ยวกับช่องทางการสื่อสารแบบสมดุล	5
2.2 อัลกอริทึมในการรักษาสมดุลของปริมาณการจราจร	15
2.3 การตรวจสอบสถานะและปริมาณการจราจรในช่องทางการสื่อสาร	16
2.4 โปรแกรมที่ใช้ในการทดลอง	24
2.5 งานวิจัยที่เกี่ยวข้อง	27
3 วิธีดำเนินการวิจัย	
3.1 การจัดสภาพแวดล้อมการทำงานของช่องทางการสื่อสารแบบสมดุล	29
3.2 การพัฒนาและการใช้งานโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล	35

สารบัญ(ต่อ)

หน้า

4 การทดลองและผลการทดลอง

4.1 การทดลอง 40

4.2 ผลการทดลองเพื่อเปรียบเทียบประสิทธิภาพของอัลกอริทึม 45

4.3 ผลการทดลองการจัดช่องทางการสื่อสารเมื่อช่องทางการสื่อสารเกิดขัดข้อง 51

5 สรุปผลการศึกษา

5.1 สรุปผลการศึกษา 52

5.2 ข้อเสนอแนะ 52

5.3 ปัญหาและอุปสรรค 53

เอกสารอ้างอิง **56**

ประวัติผู้วิจัย **58**

สารบัญตาราง

ตารางที่		หน้า
2.1	แสดงคำสั่งพื้นฐานที่ใช้ใน SNMP	12
4.1	เปรียบเทียบประสิทธิภาพของอัลกอริทึม	50

สารบัญภาพ

ภาพที่		หน้า
2.1	ลักษณะของอุปกรณ์ Load Balancing Router	5
2.2	ลักษณะการเชื่อมต่อของ Load Balancing Router	6
2.3	BIG-IP Link Controller ของบริษัท F5 Networks,Inc.	6
2.4	การใช้งาน Advanced Router ของบริษัท Sys Master Corp.	7
2.5	ระบบเครือข่ายและคอมพิวเตอร์ที่ใช้สำหรับโปรแกรม PF Sense	8
2.6	การกำหนดค่าการใช้งานของโปรแกรม PF Sense	8
2.7	ลำดับชั้นสื่อสารของ TCP/IP เมื่อเปรียบเทียบกับ OSI Model	9
2.8	องค์ประกอบของ Simple Network Management Protocol (SNMP)	12
2.9	โครงสร้างของ Object Identifier	13
2.10	การจัดกลุ่มคอมมูนิตี้ (Communities) ของอุปกรณ์ที่สนับสนุน SNMP	14
2.11	กราฟแสดงปริมาณการจราจรที่ถูกสร้างขึ้นโดยโปรแกรม MRTG	25
2.12	แฟ้มบันทึกปริมาณการจราจรของโปรแกรม MRTG	26
3.1	แบบจำลองแสดงสภาพแวดล้อมของการจัดช่องทางการสื่อสารแบบสมดุล	29
3.2	การกำหนด IP Address และ Subnet Mask	26
3.3	การกำหนดชื่อผู้ใช้และรหัสผ่าน	31
3.4	สถานะของเราท์เตอร์ (router) และ IP Address ที่กำหนดให้โดยผู้ให้บริการอินเตอร์เน็ท	32
3.5	การกำหนด community ของ SNMP ให้กับเราท์เตอร์ (router)	33
3.6	การกำหนด IP Address และ Subnet Mask ให้กับสวิทช์ (switch)	34
3.7	การกำหนดตารางเส้นทาง (routing table) ให้กับสวิทช์ (switch)	34
3.8	การกำหนด community ให้กับสวิทช์ (switch)	34
3.9	คลาสไดอะแกรม (class diagram) ของโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล	36
3.10	การพัฒนาโปรแกรมการจัดช่องทางการสื่อสารแบบสมดุลด้วยโปรแกรม Microsoft Visual Studio.Net 2003	37
3.11	การกำหนดช่องทางการสื่อสารให้กับโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล	38
3.12	การกำหนดค่าของ Balancer	39

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3.13	การกำหนดอัลกอริทึมในการทำงานของโปรแกรม จัดช่องทางการสื่อสารแบบสมดุล	39
4.1	สภาพแวดล้อมการทดลองการจัดช่องทางการสื่อสารแบบสมดุลบนเครือข่ายระบบปิด	41
4.2	อุปกรณ์ที่ใช้ในการทดลองการจัดช่องทางการสื่อสารแบบสมดุลบนเครือข่ายระบบปิด	42
4.3	สภาพแวดล้อมการทดลองการจัดช่องทางการสื่อสารแบบสมดุลบนเครือข่ายระบบอินเตอร์เน็ต	43
4.4	อุปกรณ์ที่ใช้ในการทดลองการจัดช่องทางการสื่อสารแบบสมดุลบนเครือข่ายระบบอินเตอร์เน็ต	44
4.5	กราฟแสดงปริมาณการจราจรตามช่วงเวลาทุกๆ 3 นาที โดยอัลกอริทึม Round-Robin	45
4.6	กราฟแสดงปริมาณการจราจรตามรอบเวลาการทำงานของโปรแกรมในทุกๆ 7 วินาที โดยอัลกอริทึม Round-Robin	46
4.7	กราฟแสดงปริมาณการจราจรตามช่วงเวลาทุกๆ 3 นาที โดยอัลกอริทึม Weighted Round-Robin	47
4.8	กราฟแสดงปริมาณการจราจรตามรอบเวลาการทำงานของโปรแกรมในทุกๆ 7 วินาที โดยอัลกอริทึม Weighted Round-Robin	48
4.9	กราฟแสดงปริมาณการจราจรตามช่วงเวลาทุกๆ 3 นาที โดยอัลกอริทึม Dynamic Weighted Round-Robin	49
4.10	กราฟแสดงปริมาณการจราจรตามรอบเวลาการทำงานของโปรแกรมในทุกๆ 7 วินาที โดยอัลกอริทึม Dynamic Weighted Round-Robin	49
4.11	กราฟแสดงปริมาณการจราจรเฉลี่ยในทุกๆ 5 นาที ของช่องทางการสื่อสารที่ 1 และ 2	51

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเครือข่ายอินเตอร์เน็ตมีส่วนสำคัญในชีวิตประจำวันเป็นอย่างมาก เนื่องจากมีบริการในด้านต่างๆ บนระบบอินเตอร์เน็ตอยู่อย่างมากมายหลายด้าน ไม่ว่าจะเป็นบริการด้านการค้นหาข้อมูล (Search engine) ธุรกรรมทางด้านการค้าอิเล็กทรอนิกส์ (e-commerce) การธนาคารอิเล็คทรอนิกส์ (e-banking) หรือการส่งข้อมูลในรูปแบบอิเล็กทรอนิกส์ต่างๆ เช่น จดหมายอิเล็กทรอนิกส์ (e-mail) ได้มีปริมาณการใช้งานเพิ่มมากขึ้นอย่างต่อเนื่อง ดังนั้นหน่วยงานหรือองค์กรต่างๆ จึงมีการจัดตั้งช่องทางการสื่อสารเพื่อเชื่อมต่อระหว่างเครือข่ายภายในหน่วยงานหรือองค์กรของตนเข้ากับระบบอินเตอร์เน็ต เพื่อให้ใช้งานได้อย่างรวดเร็วและเพียงพอต่อความต้องการของผู้ใช้ ซึ่งวิธีการหนึ่งก็คือการใช้ช่องทางการสื่อสารขนาดใหญ่ (high-bandwidth) หนึ่งช่องทางไว้เพื่อบริการ ส่วนอีกวิธีการหนึ่งก็คือการรวมช่องทางการสื่อสารขนาดเล็กหลายช่องทางเข้าด้วยกัน (multiple-Internet-link)[1] ซึ่งอาจเกิดปัญหาในกรณีที่ช่องทางการสื่อสารหนึ่งๆ มีปริมาณการจราจรที่หนาแน่น ในขณะที่ช่องทางการสื่อสารอื่นๆ มีปริมาณการจราจรเพียงเล็กน้อย ทำให้ไม่สามารถใช้ช่องทางการสื่อสารที่มีอยู่ทั้งหมดได้อย่างเต็มประสิทธิภาพ (Bandwidth-optimization) ปัญหาดังกล่าวสามารถแก้ไขได้โดยการจัดการช่องทางการสื่อสารแบบสมดุล (Network Load Balancing : NLB)[2]

การจัดช่องทางการสื่อสารแบบสมดุลมีทั้งลักษณะที่เป็นฮาร์ดแวร์และซอฟต์แวร์ ซึ่งต่างก็อาศัยหลักการในการกระจายการจราจร (traffic) ให้กับช่องทางการสื่อสารทั้งหมดที่มีอยู่ในปริมาณที่เท่ากัน แต่ทั้งสองลักษณะก็มีข้อจำกัดที่แตกต่างกัน กล่าวคือ ลักษณะที่เป็นฮาร์ดแวร์ก็ไม่อาจที่จะเพิ่มช่องทางการสื่อสารหรือเพิ่มความเร็วเกินกว่าที่อุปกรณ์นั้นๆ มีอยู่ได้ และไม่สามารถพัฒนาหรือนำอัลกอริทึมอื่นๆ มาใช้นอกจากที่มีในฮาร์ดแวร์อยู่แล้ว ส่วนในลักษณะที่เป็นซอฟต์แวร์ระบบก็อาจต้องทำงานหนัก เนื่องจากต้องตรวจสอบสถานะและปริมาณการจราจรในช่องทางการสื่อสาร เพื่อนำข้อมูลมาประมวลผลในการจัดช่องทางการจราจรแบบสมดุล

ดังนั้น เพื่อลดข้อจำกัดในการเพิ่มจำนวนช่องทางการสื่อสารของฮาร์ดแวร์ที่ใช้ในการรักษาสมดุลของระบบเครือข่าย และลดภาระการทำงานในซอฟต์แวร์รักษาสมดุลของเครือข่าย ตลอดจนเพื่อเพิ่มความสามารถในการพัฒนาอัลกอริทึมเพื่อใช้ในการรักษาสมดุลของเครือข่าย การ

จัดช่องทางการสื่อสารแบบสมดุลในงานวิจัยนี้ จึงได้เสนอการนำอุปกรณ์เครือข่ายที่ใช้กันอยู่โดยทั่วไป มาทำงานร่วมกับซอฟต์แวร์ที่พัฒนาขึ้นเพื่อบริหารจัดการปริมาณการจราจรในช่องทางการสื่อสารแบบสมดุล ซึ่งทั้งสองส่วนจะทำงานอยู่บนพื้นฐานของ SNMP: Simple Network Management Protocol[3][4][5][6] โดยได้นำอัลกอริทึม Round-Robin[7][8] อัลกอริทึม Weighted Round-Robin[9] และอัลกอริทึม Dynamic Weighted Round-Robin[10] มาใช้ในการจัดการปริมาณการจราจรในช่องทางการสื่อสาร และทำการเปรียบเทียบความสามารถในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสารในแต่ละอัลกอริทึม อีกทั้งได้พัฒนาความสามารถในการทนทานต่อความผิดพลาด (Fault-tolerance) และสามารถทำงานได้อย่างต่อเนื่องโดยไม่หยุดชะงัก (Availability) ในกรณีที่ช่องทางการสื่อสารบางช่องทางไม่สามารถใช้งานได้ ทำให้ระบบมีความสามารถรวมช่องทางการสื่อสารหลายช่องทางให้เสมือนเป็นช่องทางเดียว (Virtual gateway) โดยใช้โปรแกรมการจัดช่องทางการสื่อสารแบบสมดุลที่ได้พัฒนาขึ้นในงานวิจัยนี้

สำหรับหลักการในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสารที่ใช้ในงานวิจัยนี้ โดยให้มีความสมดุลของปริมาณการจราจรทั้งในขาเข้า (incoming traffic) และขาออก (outgoing traffic) นั้น ได้กระทำโดยอาศัยหลักการของการจัดเส้นทางบนเครือข่ายอินเตอร์เน็ต กล่าวคือ การส่งข้อมูลขาออกโดยเส้นทางใด ก็จะใช้เส้นทางนั้นในการรับข้อมูลในขาเข้ากลับมาเช่นกัน ดังนั้นจึงได้อาศัยหลักการนี้ในรักษาสมดุลของปริมาณการจราจรในขาเข้า (incoming traffic) โดยการควบคุมปริมาณการจราจรขาออก (incoming traffic) ให้อยู่ในปริมาณที่เหมาะสม

ทั้งนี้ สภาพแวดล้อมในการทำงานของระบบจะต้องอาศัยอุปกรณ์เครือข่ายที่ทำหน้าที่ในการเชื่อมต่อระหว่างเครือข่ายภายในกับเครือข่ายอินเตอร์เน็ต ซึ่งก็คือเราท์เตอร์ (router) ที่มีฟังก์ชัน NAT: Network Address Translation เพื่อให้สามารถใช้หลักการของการจัดเส้นทางบนเครือข่ายอินเตอร์เน็ตได้ กล่าวคือ ฟังก์ชัน NAT: Network Address Translation จะทำให้สามารถส่งข้อมูลกลับมาทางเส้นทางเดิมได้อย่างถูกต้อง ซึ่งเราท์เตอร์ (router) ที่ใช้งานกันอยู่ในปัจจุบันต่างก็มีฟังก์ชันการทำงานดังกล่าวอยู่แล้ว

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อทำการวิเคราะห์และเปรียบเทียบประสิทธิภาพของอัลกอริทึม Round-Robin, Weighted Round-Robin และ Dynamic Weighted Round-Robin ในการจัดช่องทางการจัดช่องทางการสื่อสารแบบสมดุลตามสภาพแวดล้อมที่ได้กำหนดขึ้น

1.2.2 เพื่อสร้างระบบเครือข่ายแบบสมดุล ที่มีความสามารถในการรวมช่องทางการสื่อสารหลายช่องทางให้เสมือนเป็นช่องทางเดียวกัน และสามารถหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ไม่สามารถใช้งานได้

1.3 ขอบเขตการวิจัย

ขอบเขตการศึกษาในงานวิจัยนี้ได้เสนอการจัดการช่องทางการสื่อสารแบบสมดุล โดยอาศัยการทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์ ซึ่งมีความสามารถในการรวมช่องทางการสื่อสารหลายช่องทางให้เสมือนเป็นช่องทางเดียว โดยการใช้ SNMP (Simple Network Management Protocol) ในการตรวจสอบสถานะและปริมาณการจราจร และทดสอบประสิทธิภาพการทำงานของอัลกอริทึมต่างๆ ซึ่งประกอบไปด้วยอัลกอริทึม Round-Robin, Weighted Round-Robin และ Dynamic Weighted Round-Robin

โดยในการวัดประสิทธิภาพการทำงานของอัลกอริทึมต่างๆ จะพิจารณาเปรียบเทียบจากความเหมาะสมในการจัดการปริมาณข้อมูลขาออก (outgoing traffic) ให้กับช่องทางการสื่อสารตามปริมาณการจราจรของแต่ละช่องทางฯ ในขณะนั้น เพื่อให้ปริมาณการจราจรในแต่ละช่องทางมีความสมดุลกันมากที่สุด ตามข้อกำหนด ความสามารถ และสภาพแวดล้อมดังต่อไปนี้

1.3.1 บริหารจัดการการจราจรในส่วนของข้อมูลที่ออกจากเครือข่าย (outgoing traffic)

1.3.2 สามารถตรวจสอบปริมาณการจราจรในช่องทางการสื่อสารจากค่าร้อยละของปริมาณข้อมูลขาเข้า (incoming traffic) และข้อมูลขาออก (outgoing traffic) ในแต่ละช่องทางการสื่อสาร

1.3.3 สามารถตรวจสอบการคงอยู่ของช่องทางการสื่อสาร โดยสามารถยกเลิกการใช้งานช่องทางการสื่อสารที่ไม่สามารถใช้งานออกจากเครือข่ายและสามารถนำกลับมาใช้งานใหม่ได้

1.3.4 สามารถบริหารจัดการเครือข่ายโดยการกำหนดภาระงานไปให้กับช่องทางการสื่อสารที่มีปริมาณการจราจรน้อย ให้อยู่ในลักษณะสมดุลกันในทุกช่องทางที่มีอยู่

1.3.5 ผู้บริหารเครือข่ายสามารถกำหนดการทำงานของโปรแกรมโดยผ่าน Graphic User Interface (GUI) ได้

1.3.6 ระบบทำงานภายในสภาวะแวดล้อมของอุปกรณ์ที่สนับสนุน Simple Network Management Protocol (SNMP)

1.4 ประโยชน์ที่จะได้รับจากงานวิจัย

1.4.1 สามารถนำโปรแกรมการจัดช่องทางการการสื่อสารแบบสมดุลที่พัฒนาขึ้นไปใช้ในจัดการปริมาณการจราจรในระบบเครือข่ายที่มีช่องทางการสื่อสารหลายช่องทาง โดยอาศัยหลักการของการจัดการเครือข่ายแบบสมดุล

1.4.2 สามารถแก้ไขปัญหาที่เกิดจากการที่ช่องทางการสื่อสารช่องทางใดช่องทางหนึ่งหรือหลายช่องทางไม่สามารถใช้งานได้ และสามารถนำช่องทางการสื่อสารสำรองมาใช้ในกรณีที่ต้องการ

1.5 สถานที่ทำวิจัย

1.5.1 สถาบันวิจัยและพัฒนาอุตสาหกรรมซอฟต์แวร์ อาคารศูนย์วัฒนธรรม มหาวิทยาลัยราชภัฏอุบลราชธานี

1.5.2 บ้านเลขที่ 47 ถนน ชลประทาน-ท่าบ่อ ตำบล ในเมือง อำเภอ เมือง จังหวัด อุบลราชธานี

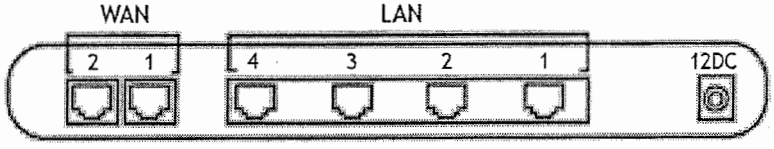
บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 ความรู้พื้นฐานเกี่ยวกับช่องทางการสื่อสารแบบสมดุล

2.1.1 การจัดช่องทางการสื่อสารแบบสมดุล (Network Load Balancing)

การจัดช่องทางการสื่อสารแบบสมดุลจะทำการกระจายการจราจร (traffic) ไปให้กับช่องทางการสื่อสาร (gateways) ที่มีอยู่ในปริมาณที่เหมาะสม โดยมีทั้งลักษณะที่เป็นฮาร์ดแวร์และซอฟต์แวร์ซึ่งมีข้อดีและข้อเสียต่างกัน ข้อดีของลักษณะที่เป็นฮาร์ดแวร์ก็คือสามารถทำงานได้รวดเร็ว แต่ก็มีข้อเสียคือ ไม่สามารถขยายขีดความสามารถในการเชื่อมต่อกับช่องทางการสื่อสารเกินกว่าจำนวนที่กำหนดไว้ได้ และอาจไม่รองรับเทคโนโลยีทางด้านเครือข่ายที่อาจเกิดขึ้นในอนาคต ส่วนข้อดีของลักษณะที่เป็นซอฟต์แวร์ก็คือ สามารถรองรับเทคโนโลยีทางด้านเครือข่ายที่เกิดขึ้นในอนาคต เช่น ในด้านความเร็วที่เพิ่มขึ้น เป็นต้น นอกจากนี้ยังสามารถพัฒนาอัลกอริทึม (algorithms) ซึ่งก็คือขั้นตอนวิธีการที่ใช้รักษาสมดุลของเครือข่ายให้มีประสิทธิภาพมากยิ่งๆ ขึ้นได้ ส่วนข้อเสียก็คือ ประสิทธิภาพของระบบจะขึ้นอยู่กับสภาพแวดล้อมของระบบโดยรวม เนื่องจากต้องอาศัยการทำงานร่วมกัน ระหว่างคอมพิวเตอร์ที่ทำหน้าที่ในการกระจายการจราจรให้กับช่องทางการสื่อสารที่มีอยู่ ตลอดจนขึ้นอยู่กับปริมาณการจราจรระบบเครือข่ายในขณะนั้นด้วย

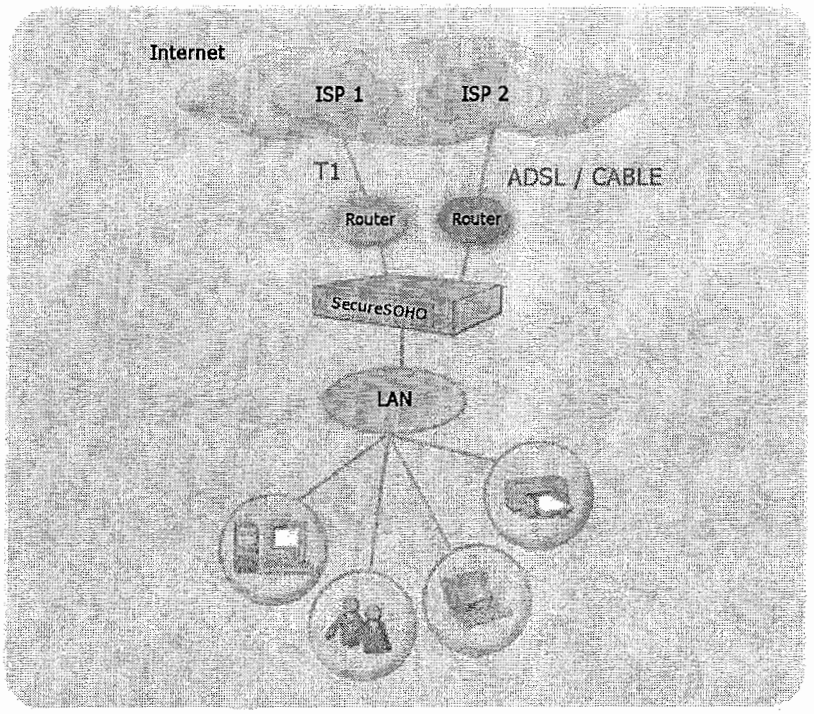
ฮาร์ดแวร์ที่ใช้ในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสารที่เรียกว่า Load Balancing Router [11] มีลักษณะดังในภาพที่ 2.1 โดยอุปกรณ์ดังกล่าวจะมีพอร์ต (port) อยู่ 2 กลุ่ม คือ WAN และ LAN ซึ่งกลุ่มหนึ่งจะใช้เชื่อมต่อกับเครือข่ายภายนอก ส่วนอีกกลุ่มจะใช้เชื่อมต่อเข้ากับเครือข่ายภายในตามลำดับ



ภาพที่ 2.1 ลักษณะของอุปกรณ์ Load Balancing Router

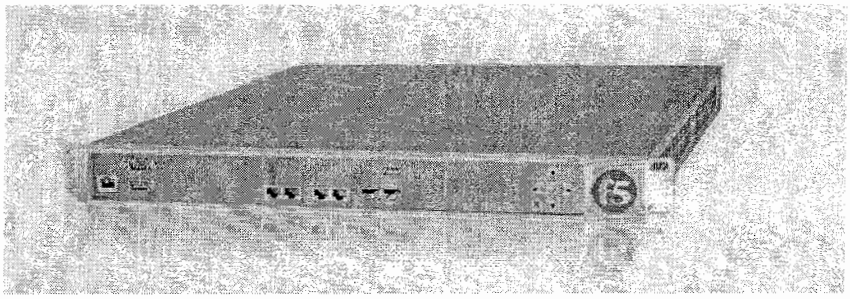
จากภาพที่ 2.1 แสดงให้เห็นถึงข้อจำกัดของอุปกรณ์ทางด้านกายภาพ เนื่องจากพอร์ต (port) ต่างๆ นั้นมีอยู่อย่างจำกัด จึงไม่สามารถที่จะขยายขีดความสามารถในการเชื่อมต่อกับทั้งเครือข่ายภายในและเครือข่ายภายนอกได้โดยปราศจากการใช้อุปกรณ์อื่นๆ มาเพิ่มเติม

ส่วนในภาพที่ 2.2 แสดงลักษณะการเชื่อมต่อของอุปกรณ์ดังกล่าว ซึ่งจะเห็นว่าสามารถเชื่อมต่อกับผู้ให้บริการอินเตอร์เน็ตได้จำกัดเพียง 2 ช่องทาง

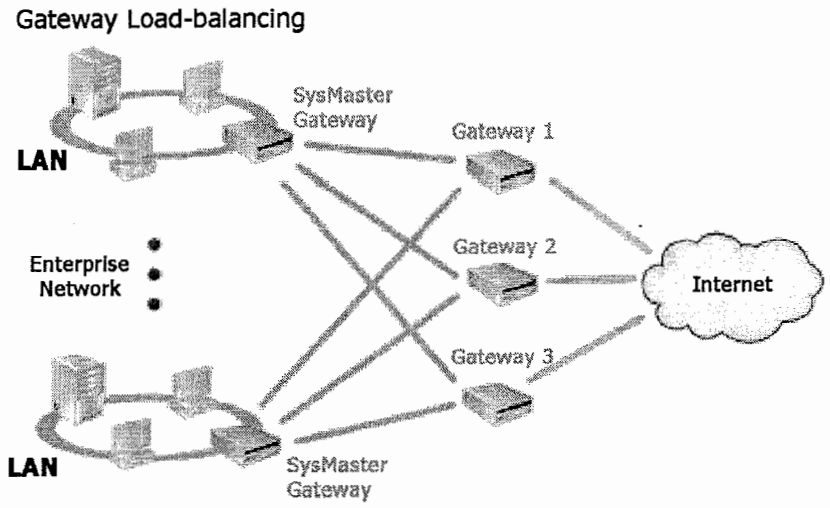


ภาพที่ 2.2 ลักษณะการเชื่อมต่อของ Load Balancing Router

ตัวอย่างอุปกรณ์ที่สามารถเชื่อมต่อกับช่องทางการสื่อสารได้หลายช่องทางและมีความสามารถในการจัดช่องทางการสื่อสารแบบสมดุล ได้แก่ BIG-IP Link Controller ของบริษัท F5 Networks, Inc ซึ่งใช้อัลกอริทึม Round-Robin ในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสารดังภาพที่ 2.3 และ Advanced Router [12] ของบริษัท SysMaster Corp. ซึ่งใช้อัลกอริทึม Weighted Round-Robin โดยมีรูปแบบการเชื่อมต่อเพื่อใช้งานดังแสดงในภาพที่ 2.4



ภาพที่ 2.3 BIG-IP Link Controller ของบริษัท F5 Networks, Inc.

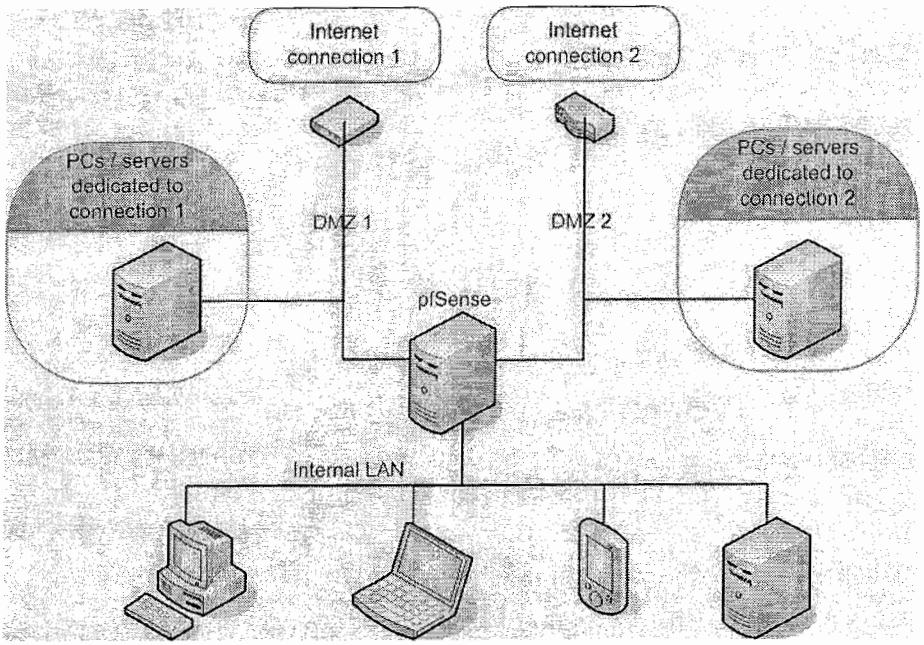


ภาพที่ 2.4 การใช้งาน Advanced Router ของบริษัท SysMaster Corp.

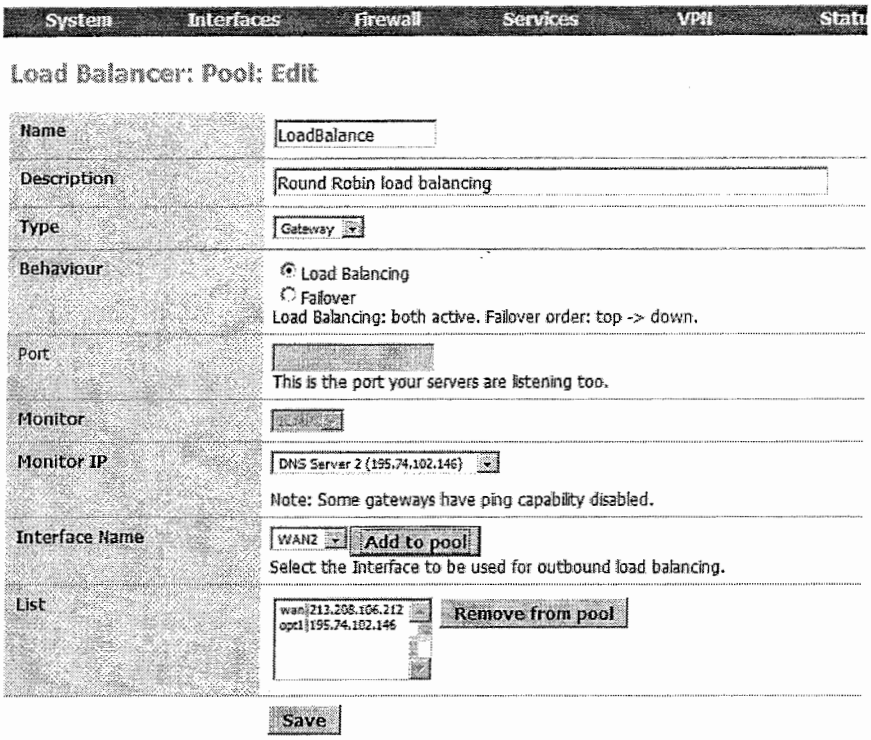
สำหรับซอฟต์แวร์ที่ใช้ในการจัดช่องทางการสื่อสารแบบสมดุลนั้น ได้แก่ โปรแกรม PFSense[13] ซึ่งเป็นโปรแกรมที่มีความสามารถในการรักษาสมดุลของปริมาณการจราจร และแก้ปัญหาในกรณีที่ช่องทางการสื่อสารไม่สามารถใช้งานได้ (Load Balance with Fail Over) โดยที่โปรแกรม PFSense จะทำงานโดยการจัดการกับข้อมูลที่ออกจากเครือข่ายภายในออกไปยังเครือข่ายอินเตอร์เน็ต (outbound internet traffic) โดยใช้อัลกอริทึม Round-Robin

ในส่วนของการใช้งานโปรแกรม PFSense ผู้ใช้หรือผู้ดูแลระบบเครือข่าย จะต้องติดตั้งโปรแกรมลงบนระบบปฏิบัติการ FreeBSD ซึ่งเป็นระบบปฏิบัติการลีนุกส์ (Linux) โดยจะต้องติดตั้งแผงวงจรเครือข่าย (Network Interface Adaptor) อย่างน้อย 3 ใบ สำหรับระบบเครือข่ายที่มีช่องทางการสื่อสาร 2 ช่องทาง

โดยอุปกรณ์จะถูกติดตั้งและกำหนดค่าต่างๆ ลงในโปรแกรมดังที่แสดงในภาพที่ 2.5 และภาพที่ 2.6 ดังนี้



ภาพที่ 2.5 ระบบเครือข่ายและคอมพิวเตอร์ที่ใช้สำหรับโปรแกรม PFSense



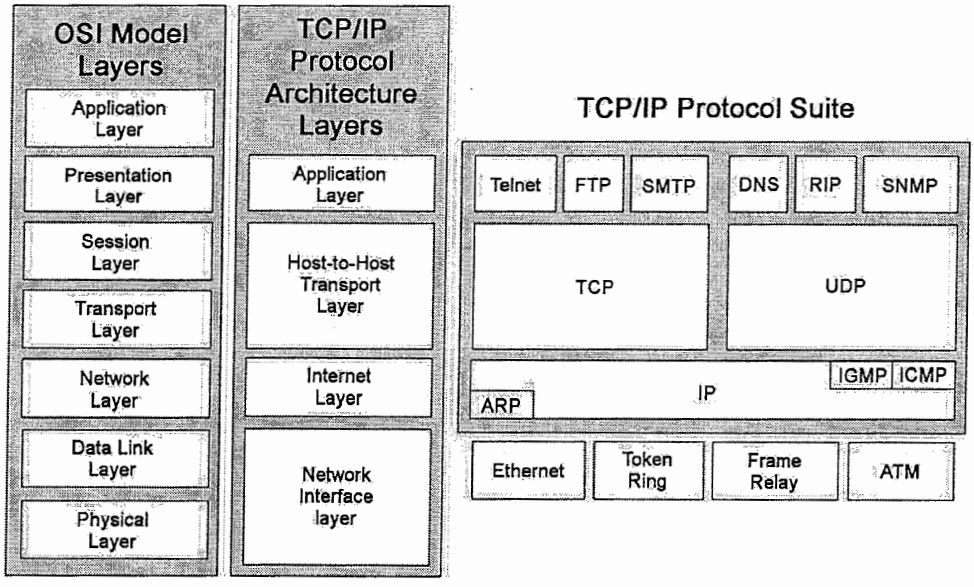
ภาพที่ 2.6 การกำหนดค่าการใช้งานของโปรแกรม PFSense

สำหรับการจัดการช่องทางการสื่อสารแบบสมดุลในงานวิจัยนี้ จะได้นำเสนอการพัฒนาซอฟต์แวร์เพื่อทำงานร่วมกับฮาร์ดแวร์ เช่น สวิทช์ (switch) หรือเราท์เตอร์ (router) ซึ่งเป็นอุปกรณ์ที่

ใช้ในระบบเครือข่ายอยู่โดยทั่วไป โดยที่ซอฟต์แวร์และฮาร์ดแวร์เหล่านี้ต่างก็มีการทำงานอยู่ภายใต้ข้อกำหนดและมาตรฐานเดียวกันทำให้สามารถนำมาใช้งานร่วมกันได้ ดังนั้นจึงจะได้กล่าวถึงข้อกำหนดและมาตรฐานของโปรโตคอล TCP/IP และ SNMP ที่ได้นำมาใช้ในงานวิจัยนี้ โดยที่โปรโตคอลเหล่านี้ต่างก็มีการอ้างอิงถึงรูปแบบของเครือข่ายมาตรฐานสากล ซึ่งมีรายละเอียดดังต่อไปนี้

2.1.2 เครือข่ายมาตรฐานสากลและโปรโตคอลที่ใช้ในการจัดช่องทางการสื่อสาร

เนื่องจากการจัดการช่องทางการสื่อสารแบบสมดุลในงานวิจัยนี้ ได้ออกแบบขึ้นมาเพื่อนำไปใช้กับระบบเครือข่าย TCP/IP ซึ่งเป็นโปรโตคอลของระบบเครือข่ายการสื่อสารที่มีการอ้างอิงถึงแบบจำลองขององค์กรเครือข่ายมาตรฐานสากล (International Standard Organization: ISO) ที่ได้กำหนดรูปแบบโครงสร้างมาตรฐานสำหรับการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ที่เรียกว่า "Open System Interconnection : OSI"[14] เพื่อเป็นข้อกำหนดหรือข้อตกลงในการสื่อสารข้อมูลระหว่างอุปกรณ์ต่างๆ ที่ทำงานบนระบบเครือข่ายให้สามารถทำงานร่วมกันได้ โดยประกอบขึ้นจากลำดับชั้นทั้งหมด 7 ลำดับชั้น ดังแสดงในภาพที่ 2.7 โครงสร้างดังกล่าวได้เน้นความสำคัญของรูปแบบการติดต่อสื่อสารด้วยระบบเปิด (open systems) จึงทำให้ผู้ผลิตแต่ละรายสามารถพัฒนาฮาร์ดแวร์และซอฟต์แวร์สำหรับระบบเครือข่ายได้โดยอิสระ



ภาพที่ 2.7 ลำดับชั้นสื่อสารของ TCP/IP เมื่อเปรียบเทียบกับ OSI Model

โดยที่โครงสร้างแบบจำลองขององค์กรเครือข่ายมาตรฐานสากลได้มีการกำหนดหน้าที่การทำงานของแต่ละลำดับชั้นสื่อสารต่างๆ ไว้ดังต่อไปนี้

2.1.2.1 ชั้นสื่อสารกายภาพ (Physical layer) ลำดับชั้นนี้ทำหน้าที่ในการกำหนดมาตรฐานของสัญญาณทางไฟฟ้า มาตรฐานการเชื่อมต่อของอุปกรณ์ที่ใช้ในการเชื่อมต่อ มาตรฐานของสื่อนำสัญญาณ ตลอดจนมาตรฐานของแรงดันไฟฟ้าและรูปแบบการรับ-ส่งบิตข้อมูลที่เกิดขึ้นในสื่อนำสัญญาณ

2.1.2.2 ชั้นสื่อสารเชื่อมต่อข้อมูล (Data link layer) เป็นลำดับชั้นที่รับผิดชอบในการส่งข้อมูลบนเครือข่ายแต่ละประเภท เช่น Ethernet, Token Ring, FDDI หรือ WAN ในลักษณะต่างๆ และดูแลการห่อหุ้มข้อมูลจากลำดับชั้นที่อยู่ในลำดับสูงกว่า เช่น IP มาบรรจุไว้ภายในเฟรม (frame) แล้วส่งข้อมูลจากต้นทางไปยังอุปกรณ์ต่างๆ ลำดับชั้นนี้จะทำงานตามกลไกและอัลกอริทึมตลอดจนรูปแบบของเฟรมที่ใช้ในเครือข่ายประเภทต่างๆ ได้

2.1.2.3 ชั้นสื่อสารควบคุมเครือข่าย (Network Layer) เป็นลำดับชั้นที่มีหน้าที่หลักในการรับ-ส่งข้อมูลจากต้นทางไปยังปลายทาง ซึ่งมีหน่วยเป็นแพ็กเก็ต (packet) โดยมีการกำหนดลอจิคัลแอดเดรส (logical address) ให้กับเครื่องคอมพิวเตอร์ และอุปกรณ์การสื่อสารเพื่อใช้ระบุตัวตน ตัวอย่างของโปรโตคอลในลำดับชั้นนี้ก็คือ IP (Internet Protocol) และ ลอจิคัลแอดเดรส (logical address) ที่ใช้ก็คือหมายเลข IP Address

สำหรับโปรแกรมเพื่อควบคุมและติดต่อกับอุปกรณ์เครือข่ายในงานวิจัยนี้ ได้ใช้ API ของโปรโตคอล Simple Network Management Protocol (SNMP) ซึ่งเป็นโปรโตคอลที่ใช้ในการบริหารจัดการอุปกรณ์เครือข่าย ซึ่งจะได้กล่าวถึงรายละเอียดในหัวข้อถัดไป

2.1.2.4 ชั้นจัดการนำส่งข้อมูล (Transport Layer) เป็นลำดับชั้นที่มีหน้าที่ในการแบ่งข้อมูล (segmentation) ในลำดับชั้นที่อยู่ด้านบนให้พอเหมาะกับการจัดส่งไปในลำดับชั้นที่อยู่ต่ำกว่า และทำการประกอบรวบรวมข้อมูล (assembly) ที่ได้รับมาจากลำดับชั้นที่ต่ำกว่าเพื่อส่งไปให้กับลำดับชั้นที่อยู่ด้านบน ตลอดจนการให้บริการในการแก้ปัญหาเมื่อเกิดข้อผิดพลาดขึ้นในระหว่างการส่งข้อมูล (error recovery) ตัวอย่าง ของโปรโตคอลที่อยู่ในลำดับชั้นนี้คือโปรโตคอล TCP และ UDP

2.1.2.5 ชั้นสื่อสารควบคุมการเชื่อมต่อ (Session Layer) เป็นลำดับชั้นที่ควบคุมการสื่อสารจากต้นทางไปยังปลายทาง และคอยควบคุมช่องทางการสื่อสารในกรณีที่มีหลายๆ โปรเซสต้องการรับส่งข้อมูลพร้อมๆ กันบนเครื่องเดียวกัน และยังให้ลำดับชั้นที่อยู่ด้านบนสามารถควบคุมขั้นตอนการทำงานของโปรโตคอลที่อยู่ในระดับต่ำกว่า เช่น Windows Socket ที่ใช้ในระบบปฏิบัติการวินโดวส์ ซึ่งเป็น Application Program Interface (API) ที่ทำให้ผู้พัฒนาซอฟต์แวร์ในระดับบนสามารถเขียนโปรแกรมเพื่อควบคุมโปรโตคอล TCP/IP ในระดับล่างได้

2.1.2.6 ชั้นสื่อสารนำเสนอข้อมูล (Presentation Layer) เป็นลำดับชั้นที่มีวัตถุประสงค์ในการกำหนดรูปแบบของการสื่อสาร เช่น ข้อมูลที่เป็นข้อความ (text) ที่ใช้รหัส ASCII หรือ EBCDIC และการเข้ารหัสลักษณะต่างๆ เป็นต้น

2.1.2.7 ชั้นสื่อสารการประยุกต์ (Application Layer) เป็นลำดับชั้นที่กำหนดการเชื่อมต่อระหว่างแอพพลิเคชันที่ทำงานบนเครื่องคอมพิวเตอร์กับซอฟต์แวร์สื่อสารต่างๆ ที่ทำงานอยู่บนเครื่องคอมพิวเตอร์

เนื่องจากการทำงานของระบบเครือข่ายต้องเกี่ยวข้องกับอุปกรณ์และข้อกำหนดที่ใช้ในการสื่อสารเป็นจำนวนมาก แต่จากการที่องค์กรเครือข่ายมาตรฐานสากลได้แบ่งการทำงานออกเป็นลำดับชั้นต่างๆ โดยให้แต่ละลำดับชั้นก็มีขอบเขตการทำงานของตนเอง จึงทำให้ผู้ผลิตและผู้พัฒนาสามารถสร้างซอฟต์แวร์และฮาร์ดแวร์ของระบบเครือข่ายขึ้นมาใช้งานร่วมกันได้ ซึ่งในงานวิจัยชิ้นนี้ก็ได้อาศัยคุณสมบัติดังกล่าวมาใช้ในการพัฒนาการจัดการเครือข่ายแบบสมดุลด้วยเช่นกัน ดังนั้นผู้วิจัยจึงจะขออธิบายและกล่าวถึงโปรโตคอลสำคัญที่ใช้ในงานวิจัยคือ SNMP โดยมีรายละเอียดดังต่อไปนี้

Simple Network Management Protocol (SNMP) เป็นโปรโตคอลในแอพพลิเคชันเลเยอร์ (Application Layer) ใช้สำหรับแลกเปลี่ยนข้อมูลเกี่ยวกับการจัดการเครือข่ายระหว่างอุปกรณ์เครือข่ายต่างโปรโตคอลนี้เป็นส่วนหนึ่งในชุดโปรโตคอล TCP/IP ซึ่งช่วยให้ผู้ดูแลระบบสามารถจัดการประสิทธิภาพ วิเคราะห์ปัญหา และให้ข้อมูลเพื่อใช้สำหรับการวางแผนเพื่อการขยายเครือข่ายในอนาคต

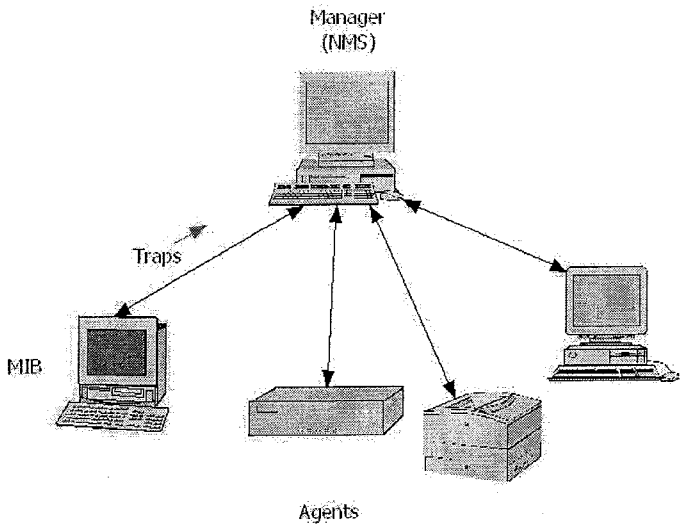
SNMP ได้ถูกกำหนดขึ้นในปี ค.ศ. 1988 เพื่อใช้แก้ปัญหาสำหรับการบริหารจัดการอุปกรณ์ต่างๆ จากการใช้งานอินเตอร์เน็ตและการเชื่อมต่อระหว่างระบบเครือข่ายที่เพิ่มขึ้น โปรโตคอลนี้เป็นที่ยอมรับอย่างกว้างขวาง โดยมีต้นกำเนิดจาก Simple Gateway Management Protocol (SGMP) โดยมุ่งหวังให้มาแทนที่สถาปัตยกรรม Common Management Information Service/Protocol (CMIS/CMIP) ซึ่งพัฒนาขึ้นมาก่อนแต่ก็ไม่รับความนิยมเท่ากับ SNMP

สำหรับองค์ประกอบของระบบจัดการระบบเครือข่ายที่ใช้โปรโตคอล SNMP จะประกอบด้วย 3 ส่วนคือ

(1) Managed Devices

(2) Agents

(3) Network Management Systems (NMS)



ภาพที่ 2.8 องค์ประกอบของ Simple Network Management Protocol (SNMP)

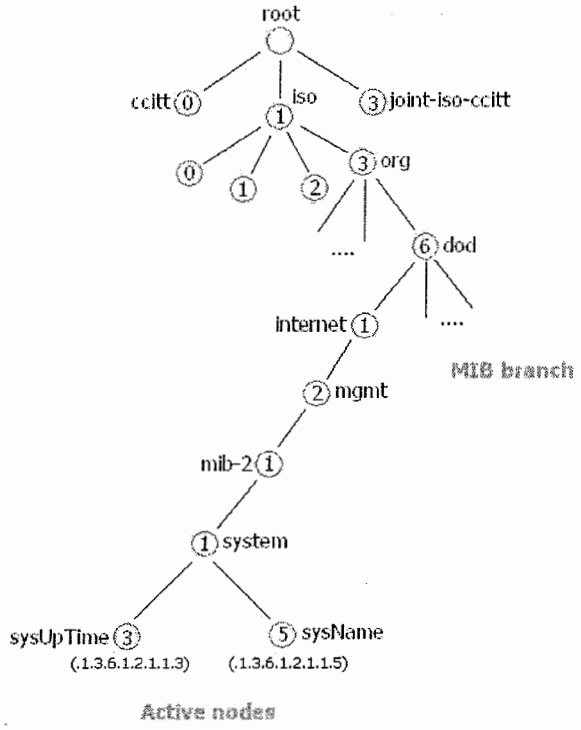
Management Devices หมายถึง อุปกรณ์ในเครือข่ายที่สามารถบริหารจัดการได้ ส่วน Agents เป็นซอฟต์แวร์ที่ติดตั้งลงในอุปกรณ์เครือข่ายที่ต้องการบริหารจัดการ ซึ่งซอฟต์แวร์นี้จะจัดการข้อมูลของอุปกรณ์ และจัดรูปแบบให้สามารถใช้งานได้กับโปรโตคอล SNMP เครือข่ายที่ใช้ SNMP จะสามารถรวบรวมข้อมูลของอุปกรณ์และส่งข้อมูลให้กับ Manager (NMS) ซึ่งเป็นอุปกรณ์ที่คอยติดตามการทำงานและควบคุมอุปกรณ์เครือข่าย อุปกรณ์เครือข่ายที่สามารถใช้ SNMP ในการบริหารจัดการได้แก่ เราท์เตอร์ (router) สวิตช์ (switch) เป็นต้น

ข้อมูลที่อยู่ใน SNMP เรียกว่า "เมสเซจ" (message) ซึ่งจะถูกบรรจุอยู่ภายในดาต้าแกรม (Datagram) โดยการที่จะนำข้อมูลต่างๆ มาใช้จะต้องอาศัยคำสั่งพื้นฐานของ SNMP คือ Protocol Data Units (PDU) ซึ่งใช้ในการร้องขอข้อมูล คำสั่งใน PDU ที่ใช้ใน SNMP นั้นแสดงอยู่ในตารางที่ 2.1

ตารางที่ 2.1 คำสั่งพื้นฐานที่ใช้ใน SNMP

PDU	หน้าที่
getRequest	ใช้เพื่อร้องขอข้อมูล
getNextRequest	ใช้เพื่อร้องขอข้อมูลถัดไปในตาราง
getResponse	ใช้ส่งข้อมูลไปยังผู้ร้องขอ
setRequest	ใช้ในการแก้ไขข้อมูล
Trap	ใช้แจ้งเตือนเหตุการณ์ผิดปกติไปยังผู้ดูแลระบบเครือข่าย

ส่วนของค่าหรือสถานะต่างๆ ของอุปกรณ์ที่ต้องการควบคุมต้องสามารถกำหนดเป็นชื่อที่ไม่ซ้ำกัน (unique name) เรียกว่า "object identifier" ซึ่ง object identifier ในแต่ละระดับชั้นจะถูกแทนด้วยโหนด (node) ซึ่งถูกกำกับด้วยหมายเลขที่ต่างกันโดยมีโครงสร้างแบบต้นไม้ (tree) ดังในภาพที่ 2.9



ภาพที่ 2.9 โครงสร้างของ object identifier

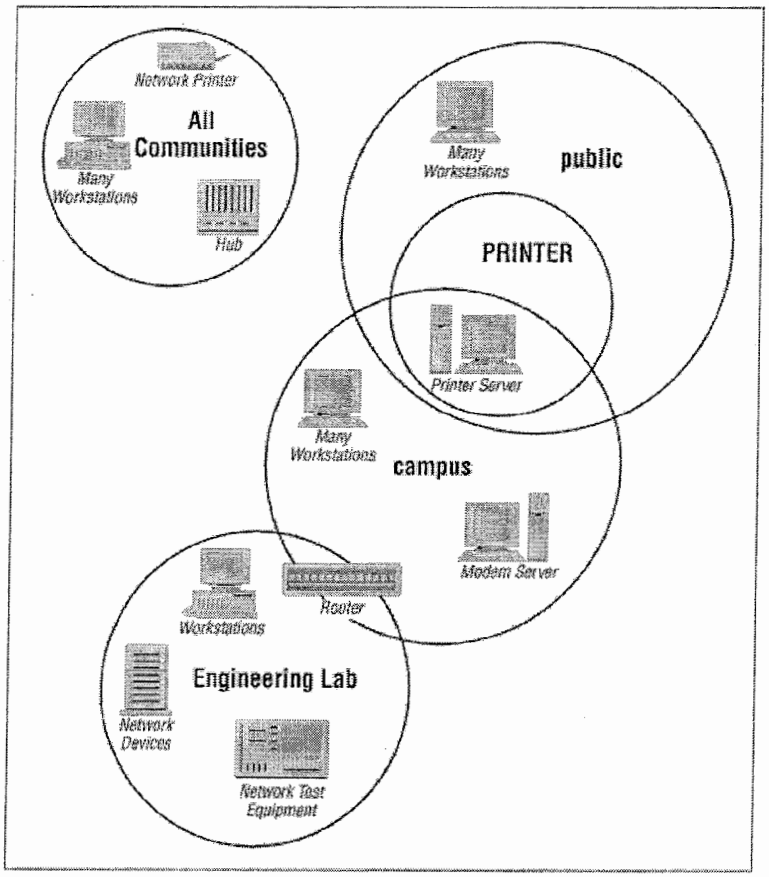
จากโครงสร้างแบบต้นไม้ของ object identifier ดังในภาพที่ 2.9 จะพบว่าแต่ละโหนดของต้นไม้ (tree) ในแต่ละลำดับชั้นจะแทนด้วยโหนด (node) ของข้อมูลในแต่ละกลุ่ม และข้อมูลในแต่ละกลุ่มก็สามารถมีโหนดลูก (child node) ในลำดับชั้นต่อไปได้อีก ดังนั้นเมื่อต้องการข้อมูลในกลุ่มใดก็สามารถระบุหมายเลขของกลุ่มตามด้วยเครื่องหมายจุด (.) และหมายเลขของกลุ่มย่อยเช่นนี้ไปเรื่อยๆ จนถึงข้อมูลที่ต้องการ เช่น การการกำหนดหมายเลข object identifier เป็น .1.3.6.1.2.1.1.3 ก็คือ object identifier ที่อยู่ในกลุ่ม iso (1), org (3), dod (6), internet (1), mgmt (2), mib-2 (1), system (1) และ sysUpTime (3) ซึ่งหมายถึง จำนวนเวลาที่ระบบถูกเปิดขึ้นใช้งาน

object identifier ทั้งหมดของอุปกรณ์เครือข่ายจะถูกเก็บไว้ในฐานข้อมูลที่เรียกว่า Management Information Base (MIB) ซึ่งเป็นข้อมูลทั้งหมดที่ SNMP ใช้ในการบริหารจัดการหรือ

ควบคุมอุปกรณ์ต่างๆ จึงอาจกล่าวได้ว่า Management Information Base (MIB) ก็คือฐานข้อมูลที่ใช้ในการบริหารจัดการเครือข่าย

SNMP ได้กำหนดคอมมูนิตี้ (community) ซึ่งก็คือกลุ่มของการใช้งานสำหรับการบริหารอุปกรณ์ต่างๆ ที่สนับสนุน SNMP โดยแต่ละคอมมูนิตี้ (community) ที่ถูกกำหนดขึ้นก็คือกลุ่มของอุปกรณ์ที่ประกอบไปด้วย agent และ NMS จำนวนอย่างละหนึ่งเป็นอย่างน้อย ชื่อที่ใช้ในการกำหนดกลุ่มของอุปกรณ์นี้เรียกว่า "community name" ซึ่งจะถูกเข้ารหัส (encode) อยู่ในเมสเซจ (message) ที่ส่งโดยสมาชิกที่อยู่ในกลุ่ม ซึ่งเรียกว่า "community string" เพื่อให้เมสเซจ (message) ที่ส่งนั้นสามารถระบุได้ว่าใครจะเป็นผู้รับเมสเซจ (message) ดังกล่าว

อุปกรณ์เครือข่ายจะยอมรับเมสเซจ (message) ก็ต่อเมื่อได้ทำการตรวจสอบ community string ที่กำหนดเอาไว้ว่าตรงกันหรือไม่ ตัวอย่างเช่น ถ้าอุปกรณ์เครือข่ายยอมรับเมสเซจ (message) มี community string เป็น "public" ดังนั้นอุปกรณ์ฯ ดังกล่าวก็จะไม่ยอมรับเมสเซจ (message) ที่มาจากอุปกรณ์ที่มี community string เป็น "private" เป็นต้น



ภาพที่ 2.10 การจัดกลุ่มคอมมูนิตี้ (communities) ของอุปกรณ์ที่สนับสนุน SNMP

ภาพที่ 2.10 แสดงกลุ่มของอุปกรณ์ที่อยู่ในเครือข่ายซึ่งสนับสนุนการทำงานของ SNMP โดยที่อุปกรณ์ส่วนใหญ่ถูกกำหนดให้อยู่ในกลุ่มเพียงกลุ่มเดียว แต่บางอุปกรณ์ถูกกำหนดให้อยู่ในสองหรือสามกลุ่ม เช่น เราท์เตอร์ (router) ถูกกำหนดให้อยู่ในกลุ่ม "Engineering Lab" และ "campus" สำหรับ printer server ถูกกำหนดให้อยู่ในกลุ่ม "campus" "PRINTER" และ "public" ในส่วนของ "All Communities" นั้นประกอบไปด้วยอุปกรณ์ต่างๆ ที่ไม่ได้ถูกกำหนดให้มี community name ดังนั้นจึงสามารถจัดอยู่ได้ในทุกกลุ่ม

2.2 อัลกอริทึมในการรักษาสมดุลของปริมาณการจราจร

อัลกอริทึมที่ใช้ในการกำหนดปริมาณการจราจรให้กับช่องทางการสื่อสารแบบสมดุลในงานวิจัยนี้ ได้นำอัลกอริทึมที่มีพื้นฐานมาจากวิธีการในการจัดเวลาซีพียู (CPU Scheduling) ซึ่งเป็นหลักการทำงานของระบบปฏิบัติการ ที่ทำให้คอมพิวเตอร์มีความสามารถในการประมวลผลโปรแกรมหลายๆ โปรแกรมในเวลาเดียวกัน มาประยุกต์ใช้กับการจัดช่องทางสื่อสารแบบสมดุล โดยอัลกอริทึมที่นำมาประยุกต์ใช้งานมีดังต่อไปนี้

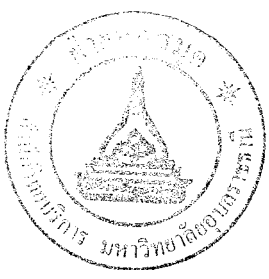
2.2.1 อัลกอริทึม Round-Robin

อัลกอริทึม Round-Robin เป็นวิธีการจัดตารางการทำงานแบบวนรอบที่ใช้ในระบบปฏิบัติการเพื่อใช้กับระบบคอมพิวเตอร์แบบแบ่งเวลา โดยแต่ละงานที่เข้ามาในระบบจะถูกแบ่งเวลาการเข้าไปใช้หน่วยประมวลผลกลางเท่าๆ กัน ซึ่งในงานวิจัยนี้ได้นำอัลกอริทึม Round-Robin มาประยุกต์ใช้ในการรักษาสมดุลของระบบเครือข่าย โดยการแบ่งเวลาในการใช้งานช่องทางการสื่อสารในแต่ละช่องทางด้วยเวลาที่เท่ากัน ข้อดีของอัลกอริทึม Round-Robin ก็คือเป็นวิธีการที่ไม่ต้องคอยตรวจสอบปริมาณการจราจรในช่องทางการสื่อสาร จึงไม่เพิ่มภาระงานให้แก่ระบบ แต่ก็อาจทำให้เกิดปัญหาคอขวด (bottle-neck) ขึ้นได้

อัลกอริทึมแบบ Round-Robin นั้นจัดว่าเป็นวิธีการที่ง่ายที่สุดวิธีการหนึ่ง โดยอาศัยการกระจายภาระงานโดยการวนรอบ ตามหลักการแล้ว ถ้าจำนวนช่องทางการสื่อสารทั้งหมดมีอยู่เป็นจำนวน n ถ้าในขณะนั้นใช้ $link_n$ เป็นช่องทางในการส่งข้อมูล โดยที่ $1 \le i \le n$ ดังนั้น ช่องทางการสื่อสารที่จะถูกใช้ถัดไปก็คือ $link_{(i+1) \bmod n}$

2.2.2 อัลกอริทึม Weighted Round-Robin

อัลกอริทึม Weighted Round-Robin เป็นอัลกอริทึมที่มีพื้นฐานมาจาก Round-Robin แต่ได้เพิ่มการกำหนดค่าน้ำหนัก (weight) ให้กับแต่ละช่องทางการสื่อสาร โดยช่องทางการสื่อสารที่มีปริมาณการจราจรน้อยก็จะถูกกำหนดค่าน้ำหนักไว้มาก เพื่อให้ช่องทางการสื่อสารนั้นถูกนำมาใช้ในการรับ-ส่งข้อมูลมากขึ้น



2.2.3 อัลกอริทึม Dynamic Weighted Round-Robin

อัลกอริทึม Dynamic Weighted Round-Robin เป็นอัลกอริทึมที่ถูกพัฒนาขึ้นจาก Weighted Round-Robin โดยมีการเพิ่มความสามารถในการคำนวณเวลาที่เหมาะสมสำหรับการตรวจสอบปริมาณการจราจรในแต่ละช่องทางสื่อสารในแต่ละรอบการทำงาน กล่าวคือ ถ้าปริมาณการจราจรอยู่ในภาวะสมดุล เวลาของรอบการทำงานก็จะเพิ่มขึ้นทำให้ระบบทำงานน้อยลง แต่ถ้าปริมาณการจราจรไม่อยู่ในภาวะสมดุล เวลาของรอบการทำงานก็จะลดลง ทำให้มีการตรวจสอบปริมาณการจราจรถี่ขึ้น เพื่อที่จะปรับให้ปริมาณการจราจรอยู่ในภาวะสมดุล

2.3 การตรวจสอบสถานะและปริมาณการจราจรในช่องทางการสื่อสาร

เนื่องจากการพัฒนาซอฟต์แวร์เพื่อใช้ในการจัดช่องทางการสื่อสารแบบสมดุล ภายใต้ข้อกำหนดและมาตรฐานของระบบเครือข่าย TCP/IP โดยการนำโปรโตคอล SNMP มาประยุกต์ใช้ในการตรวจสอบสถานะและปริมาณการใช้งานช่องทางการสื่อสาร จำเป็นต้องอาศัยองค์ประกอบและการทำงานของระบบเครือข่ายดังที่กล่าวมา นอกจากนี้แล้ว การพัฒนาโปรแกรมจัดการช่องทางการสื่อสารแบบสมดุลที่ทำงานในลำดับชั้นสื่อสารการประยุกต์ (Application Layer) ในแบบจำลอง OSI จะต้องมีการพัฒนาโปรแกรมร่วมกับชุดคำสั่งที่มีอยู่ใน Application Program Interface (API) เพื่อให้โปรแกรมเมอร์สามารถพัฒนาโปรแกรมให้ทำงานได้ตามความต้องการ โดยอาศัยคำสั่งที่มีอยู่ใน API เพื่อการอ่านข้อมูลหรือสั่งงานไปยังอุปกรณ์เครือข่าย ซึ่ง API ที่นำมาใช้ร่วมกับโปรแกรมจัดช่องทางการสื่อสารที่ได้พัฒนาขึ้นก็คือไลบรารี SNMP++ .NET API โดยที่ API ดังกล่าวจะมีหน้าที่ในการใช้โปรโตคอล SNMP ติดต่อกับอุปกรณ์เครือข่ายเพื่อให้ข้อมูลซึ่งจะถูกนำมาใช้ในการตรวจสอบสถานะของช่องทางการสื่อสาร การคำนวณปริมาณการจราจรในช่องทางการสื่อสาร การคำนวณค่าสมดุลของภาระงาน การคำนวณค่าน้ำหนักและการกำหนดรอบการทำงานของช่องทางการสื่อสาร และการคำนวณค่าความถี่ของเวลาในการตรวจสอบช่องทางการสื่อสาร ซึ่งมีรายละเอียดดังต่อไปนี้

2.3.1 การตรวจสอบสถานะของช่องทางการสื่อสาร

เนื่องจากงานวิจัยนี้ ส่วนหนึ่งมีวัตถุประสงค์ให้โปรแกรมมีความสามารถในการหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ไม่สามารถใช้งานได้ (Fault-tolerance) ดังนั้นโปรแกรมต้องมีความสามารถในการตรวจสอบการคงอยู่ของช่องทางการสื่อสารที่ใช้ในการรับ-ส่งข้อมูล ถ้าโปรแกรมตรวจพบช่องทางการสื่อสารที่ไม่สามารถใช้งานได้ ก็จะยกเลิกการรับ-ส่งข้อมูลในช่องทางการสื่อสารนั้น โดยที่โปรแกรมการจัดช่องทางการสื่อสารแบบสมดุลจะทำการติดต่อไปยังอุปกรณ์เครือข่ายที่ใช้เป็นช่องทางการสื่อสาร โดยการอ้างถึงหมายเลข object identifier ที่เป็น

สถานะของอุปกรณ์เครือข่ายที่กำหนดโดยผู้ดูแลระบบ (*ifAdminStatus*) และสถานะการใช้งานของอุปกรณ์เครือข่าย (*ifOperStatus*) ซึ่งมีหมายเลข object identifier คือ .1.3.6.1.2.1.2.2.1.7 และ .1.3.6.1.2.1.2.2.1.8 ตามลำดับ โดยค่าที่อ่านได้อาจมีสถานะเป็น "up" ซึ่งหมายถึงการที่ช่องทางการสื่อสารนั้นสามารถใช้งานได้ หรือ "down" ซึ่งหมายถึงการที่ช่องทางการสื่อสารนั้นไม่สามารถใช้งานได้

2.3.2 การคำนวณปริมาณการจราจรในช่องทางการสื่อสาร

ปริมาณการจราจรในช่องทางการสื่อสาร ประกอบด้วยทั้งปริมาณการจราจรขาเข้าและปริมาณการจราจรขาออก ซึ่งสามารถคำนวณได้จากการนำปริมาณข้อมูลขาเข้าหรือและขาออกในช่วงเวลาหนึ่งๆ หารด้วยช่วงเวลา และนำมาเปรียบเทียบอัตราส่วนกับขนาดของช่องทางการสื่อสาร เพื่อหาร้อยละของปริมาณการจราจรในช่องทางการสื่อสารนั้น โดยการนำบันทึกข้อมูลจากอุปกรณ์เครือข่ายโดยใช้โปรโตคอล SNMP ซึ่งมีข้อมูลและหมายเลขของ object identifier ที่เกี่ยวข้องดังนี้

2.3.2.1 จำนวนเวลาที่อุปกรณ์ถูกเปิดใช้งาน *(sysUpTime)* คือ .1.3.6.1.2.1.1.3

2.3.2.2 ความเร็วของช่องทางการสื่อสาร (*ifSpeed*) คือ 1.3.6.1.2.1.2.2.1.5

2.3.2.3 ปริมาณข้อมูลขาเข้า (*ifInOctets*) คือ .1.3.6.1.2.1.2.2.1.10 มีหน่วยเป็นไบต์ (byte)

2.3.2.4 ปริมาณข้อมูลขาออก (*ifOutOctets*) หมายเลขคือ .1.3.6.1.2.1.2.2.1.16 มีหน่วยเป็นไบต์ (byte)

สำหรับการคำนวณเพื่อหาร้อยละของปริมาณการจราจรจะกำหนดให้ *t0* และ *t1* คือเวลาจาก *sysUpTime* ของอุปกรณ์เครือข่าย มีหน่วยเป็นวินาที โดยที่จำนวนของบิตของข้อมูลขาเข้าต่อวินาที ($Traffic_{in}$)ในช่วงเวลา *t0* ถึง *t1* คือ

$$Traffic_{in} = \frac{(ifInOctets_{t1} + ifInOctets_{t0}) * 8}{(t1 - t0)} \quad \text{(B/s.)} \qquad .....(2.1)$$

เมื่อ $Traffic_{in}$ แทน ปริมาณการจราจรของข้อมูลขาเข้า

$ifInOctets_{t0}$ แทน ปริมาณข้อมูลขาเข้า ณ เวลาเริ่มต้น

$ifInOctets_{t1}$ แทน ปริมาณข้อมูลขาเข้า ณ เวลาสิ้นสุด

$t0$ แทน เวลาเริ่มต้น

$t1$ แทน เวลาสิ้นสุด

ดังนั้น ร้อยละของปริมาณการจราจรขาเข้า ($Load_n$) ในช่องทางการสื่อสาร คือ

$$Load_{in} = \frac{Traffic_{in} * 100}{ifSpeed} (\%) \qquad .....(2.2)$$

เมื่อ $Load_{in}$ แทน ปริมาณร้อยละของการจราจรของข้อมูลขาเข้า

$Traffic_{in}$ แทน ปริมาณการจราจรของข้อมูลขาเข้า

$ifSpeed$ แทน ความเร็วของช่องทางการสื่อสาร

สำหรับปริมาณข้อมูลขาออก ($Traffic_{out}$) สามารถคำนวณได้จาก

$$Traffic_{out} = \frac{(ifOutOctets_{t1} + ifOutOctets_{t0}) * 8}{(t1 - t0)} \text{ (B/s.)} \qquad .....(2.3)$$

เมื่อ $Traffic_{out}$ แทน ปริมาณการจราจรของข้อมูลขาออก

$ifOutOctets_{t0}$ แทน ปริมาณข้อมูลขาออก ณ เวลาเริ่มต้น

$ifOutOctets_{t1}$ แทน ปริมาณข้อมูลขาออก ณ เวลาสิ้นสุด

$t0$ แทน เวลาเริ่มต้น

$t1$ แทน เวลาสิ้นสุด

ดังนั้น ร้อยละของปริมาณข้อมูลขาออก ($Load_{out}$) ในช่องทางการสื่อสาร คือ

$$Load_{out} = \frac{Traffic_{out} * 100}{ifSpeed} (\%) \qquad .....(2.4)$$

เมื่อ $Load$ แทน ปริมาณร้อยละของการจราจรของข้อมูลขาออก

$Traffic_{out}$ แทน ปริมาณการจราจรของข้อมูลขาออก

$ifSpeed$ แทน ความเร็วของช่องทางการสื่อสาร

ปริมาณภาระงานทั้งหมด ($Load$) สามารถคำนวณได้จากปริมาณข้อมูลเฉลี่ยทั้งขาเข้าและขาออกในแต่ละช่องทางการสื่อสารได้จากสมการต่อไปนี้

$$Load = \frac{Load_{in} + Load_{out}}{bandwidth_{in} + bandwidth_{out}} \qquad .....(2.5)$$

เมื่อ $Load$ แทน ปริมาณภาระงานทั้งหมด

$Load_{in}$ แทน ปริมาณร้อยละของการจราจรของข้อมูลขาเข้า

$Load_{out}$ แทน ปริมาณร้อยละของการจราจรของข้อมูลขาออก

$bandwidth_{in}$ แทน ความกว้างของช่วงคลื่นขาเข้า

$bandwidth_{out}$ แทน ความกว้างของช่วงคลื่นขาออก

ตัวอย่างเช่น

แบนด์วิดธ์ของช่องทางการสื่อสาร (*ifSpeed*) คือ 2,048,000 B/s.

เวลา (*sysUpTime*) *t0* คือ 7 days, 11:31:41.38

เวลา (*sysUpTime*) *t1* คือ 7 days, 11:36:42.35

ปริมาณข้อมูลขาเข้า (*ifInOctets*) ณ เวลา *t0* คือ 3,390,369,342 ไบต์

ปริมาณข้อมูลขาเข้า (*ifInOctets*) ณ เวลา *t1* คือ 3,418,690,493 ไบต์

ปริมาณข้อมูลขาออก (*ifOutOctets*) ณ เวลา *t0* คือ 2,696,978,663 ไบต์

ปริมาณข้อมูลขาออก (*ifOutOctets*) ณ เวลา *t1* คือ 2,730,913,602 ไบต์

จึงได้ว่า $Traffic_{in} = \frac{(3,418,690,493 - 3,390,369,342) * 8}{301}$

$$= \frac{226,569,208}{301}$$

$$= 752,772 \text{ bps. (94 KB/s)}$$

ดังนั้น

$$Load_{in} = \frac{752,772 * 100}{2,048,000}$$

$$= 36\%$$

ปริมาณข้อมูลขาเข้า ($Load_{in}$) ในช่วงเวลา *t0* ถึง *t1* คือ 36%

สำหรับร้อยละของปริมาณข้อมูลขาออก ($Load_{out}$) ในช่องทางการสื่อสาร คือ

$$Traffic_{out} = \frac{(2{,}730{,}913{,}602 - 2{,}696{,}978{,}663) * 8}{301}$$

$$= \frac{271{,}479{,}512}{301}$$

$$= 901{,}925 \text{ B/s. (112 KBytes/s)}$$

ดังนั้น

$$Load_{in} = \frac{901{,}925 * 100}{2{,}048{,}000}$$

$$= 44\%$$

ปริมาณข้อมูลขาออก ($Load_{out}$) ในช่วงเวลา *t0* ถึง *t1* คือ 44%

ดังนั้นปริมาณภาระงานทั้งหมด (*Load*) คือ

$$Load = \frac{752{,}772 + 901{,}925}{2{,}048{,}000 + 2{,}048{,}000} \qquad = 40\%$$

2.3.3 การคำนวณค่าสมดุลของภาระงาน

เมื่อสามารถหาปริมาณภาระงานของช่องทางการสื่อสารหลักที่มีอยู่ทั้งหมดแล้ว ก็จะต้องนำปริมาณภาระงานของแต่ละช่องทางการสื่อสารมาคำนวณหาค่าสมดุลของภาระงาน จากสมการต่อไปนี้

$$\bar{v} = \frac{\sum(Load_{t,i} - \overline{Load_t})}{n} \quad \text{เมื่อ} \quad \overline{Load_t} = \frac{\sum_{i=1}^{n} L_{t,i}}{n} \quad .....(2.6)$$

เมื่อ $\bar{v}$ แทน ค่าแปรผันของภาระงานทั้งหมด

$\overline{Load_t}$ แทน ค่าสมดุลของภาระงาน

n แทน จำนวนช่องทางการสื่อสารทั้งหมด

ตัวอย่างเช่น ถ้ามีจำนวนช่องทางการสื่อสารทั้งหมด 3 ช่องทาง คือ *Load1, Load2* และ *Load3* โดยที่แต่ละช่องทางการสื่อสารมีปริมาณภาระงานเป็น 50%, 60% และ 30% ตามลำดับ ในขณะเมื่อเวลา t และช่วงเวลาที่ผ่านไปทั้งหมด (T) จะได้ว่า

$$\overline{Load_t} = \frac{50+60+30}{3}$$

$$= 47$$

ดังนั้น ค่าแปรผันของภาระงานทั้งหมดในระบบในช่วงเวลาหนึ่งๆ ($\bar{v}$) จึงหาได้จาก

$$\bar{v} = (|50\text{-}47| + |60\text{-}47| + |30\text{-}47|)/3 = 11$$

2.3.4 การคำนวณค่าค่าน้ำหนักและการกำหนดรอบการทำงานของช่องทางการสื่อสาร

การคำนวณค่าน้ำหนัก (weight) และการกำหนดรอบการทำงาน (round) ในการใช้งานช่องทางการสื่อสารนั้นเป็นส่วนสำคัญของอัลกอริทึม Dynamic Weighted Round-Robin เนื่องจากประสิทธิภาพของการจัดช่องทางการสื่อสารให้เกิดความสมดุลนั้นขึ้นอยู่กับค่าทั้งสองนี้ จากช่องทางการสื่อสาร *L1, L2* และ *L3* ที่มีปริมาณภาระงานเป็น 50%, 60% และ 30% ตามลำดับ สามารถนำมากำหนดค่าน้ำหนักและรอบการทำงานได้ดังต่อไปนี้

เนื่องจาก $load_{t,1} : load_{t,2} : load_{t,3} = 5:6:3$

ดังนั้น $w_{t,1} : w_{t,2} : w_{t,3} = \frac{1}{5} : \frac{1}{6} : \frac{1}{3}$

$= 6:5:10$

ลำดับของรอบการทำงานในการใช้งานช่องทางการสื่อสารจึงเป็นดังนี้

$link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow$
$link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_3 \rightarrow link_3 \rightarrow$
$link_3 \rightarrow link_3 \rightarrow link_3$

หลังจากที่โปรแกรมจัดช่องทางการสื่อสารตามรอบการทำงานเบื้องต้นไปแล้วนั้น ก็จะทำการวัดค่าปริมาณภาระงานอีกครั้ง เพื่อตรวจสอบความสมดุลของช่องทางการสื่อสารเพื่อนำมากำหนดค่าน้ำหนักที่จะใช้ในการรักษาสมดุลในรอบการทำงานต่อไป โดยใช้สมการต่อไปนี้

$$w_{t,i} = \frac{1}{\left[\left(load_{t,i} / k\right)+1\right]} \qquad \text{.....(2.7)}$$

เมื่อ w แทน ค่าน้ำหนัก

$\overline{Load_t}$ แทน ปริมาณภาระงานทั้งหมด

k มีค่าเท่ากับ $\sqrt{v} = \sqrt{11} = 3.3$

กำหนดให้ช่องทางการสื่อสาร *Load1, Load2* และ *Load3* วัดค่าภาระงานได้เป็น 50%, 60% และ 30% ตามลำดับ

จึงได้ว่า $w_{t,1} = \frac{1}{\left[\left(50/3.3\right)+1\right]} = \frac{1}{16}$

$$w_{t,2} = \frac{1}{\left[\left(60/3.3\right)+1\right]} = \frac{1}{20}$$

$$w_{t,3} = \frac{1}{\left[\left(30/3.3\right)+1\right]} = \frac{1}{10}$$

ดังนั้น $w_{t,1} : w_{t,2} : w_{t,3} = \frac{1}{16} : \frac{1}{20} : \frac{1}{10}$

$$= 10 : 8 : 16$$

ลำดับการทำงานของช่องทางการชื่อสารในรอบถัดไปจึงเป็นดังนี้

$$link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow$$
$$link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow$$
$$link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_2 \rightarrow link_3 \rightarrow link_1 \rightarrow link_3 \rightarrow link_1 \rightarrow$$
$$link_3 \rightarrow link_3 \rightarrow link_3 \rightarrow link_3 \rightarrow link_3 \rightarrow link_3 \rightarrow link_3$$

2.3.5 การคำนวณค่าความถี่ของเวลาในการตรวจสอบช่องทางการสื่อสาร

เนื่องจากงานวิจัยนี้มีวัตถุประสงค์ในการรักษาสมดุลของเครือข่าย โดยให้มีการตรวจสอบช่องทางการสื่อสารน้อยที่สุด เพื่อลดการทำงานของระบบโดยรวม จึงได้ใช้การคำนวณหาช่วงเวลาที่น้อยที่สุด (Smallest detecting interval) และช่วงเวลาที่มากที่สุด (Longest detecting interval) มาใช้ในการหาช่วงเวลาที่เหมาะสมในการตรวจสอบช่องทางการสื่อสาร เพื่อให้การรักษาสมดุลของช่องทางการสื่อสารให้มีประสิทธิภาพมากที่สุดดังต่อไปนี้

กำหนดค่าภาระงานของช่องทางการสื่อสาร *Load1,Load2* และ*Load3* เป็น 50%, 60% และ 30% ตามลำดับ และช่วงเวลาที่ใช้ในการตรวจสอบช่องทางการสื่อสาร (interval) คือ 5000ms.

ค่าน้ำหนักภาระงานทั้งหมดของ *Load1, Load2* และ *Load3* คือ 10 + 8 + 16 = 34

เนื่องจาก	k	=	$\sqrt{\bar{v}}$
ดังนั้น	k	=	$\sqrt{11} = 3.3$
	$\bar{a}$	=	$5000/34 = 147$
	$\bar{L}$	=	$(34/3)\cdot 11 = 1617$
	n	=	จำนวนช่องทางการสื่อสารทั้งหมด

จากสมการในการคำนวณช่วงเวลาที่น้อยที่สุด (Smallest detecting interval)

คือ $$t_l = \frac{\bar{a}k}{\bar{L} - \bar{a}} \qquad .....(2.8)$$

ดังนั้น

$$t_l = \frac{(147)\cdot(3.3)}{1617 - 147} \cdot 5000 = (0.3401)\cdot 5000 = 1701ms.$$

จากสมการในการคำนวณช่วงเวลาที่มากที่สุด (Longest detecting interval)

คือ $$t_u = \frac{\bar{a}n^2k}{(n-1)\bar{L}} \quad \text{.....(2.9)}$$

ดังนั้น

$$t_u = \frac{(147)\cdot 3^2 \cdot (3.3)}{(3-1)\cdot 1617}\cdot 5000 = (1.3908)\cdot 5000 = 6954ms.$$

จากสมการคำนวณช่วงเวลาที่เหมาะสมในการตรวจสอบช่องทางการสื่อสาร (t_b)

คือ $$\frac{\bar{a}k}{\bar{L}-\bar{a}} \le t_b < \frac{\bar{a}n^2k}{(n-1)\bar{L}}$$

ดังนั้น

$$t_b = \frac{1701+6954}{2} = 4328ms.$$

ดังนั้นช่วงเวลาที่เหมาะสมในการตรวจสอบช่องทางการสื่อสาร คือ 4328ms.

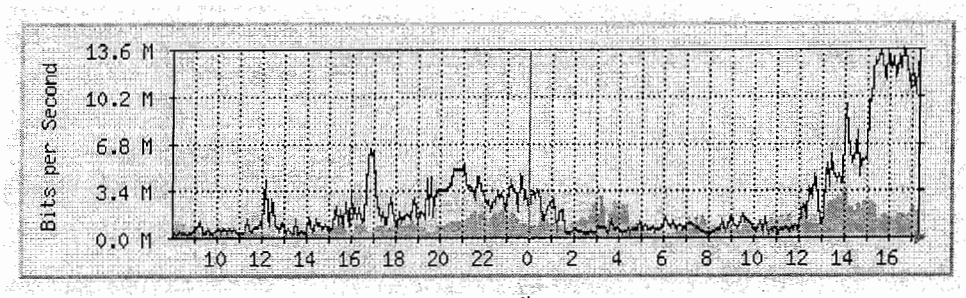
2.4 โปรแกรมที่ใช้ในการทดลอง

ในการทดสอบการทำงานของโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล ได้มีการนำซอฟต์แวร์อื่นๆ มาใช้ร่วมในการทดลองเพื่อวัดประสิทธิภาพและทดสอบการทำงาน ซึ่งซอฟต์แวร์ที่นำมาใช้ก็คือ โปรแกรม MRTG[15] ซึ่งนำมาใช้ในการบันทึกปริมาณการจราจร เพื่อนำข้อมูลที่ได้ไปเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมต่างๆ และโปรแกรมมิวทอเรนท์ (μTorrent)[16] ซึ่งใช้ในการทดสอบการรับ-ส่งข้อมูล โดยรายละเอียดของโปรแกรมดังกล่าว มีดังต่อไปนี้

2.4.1 โปรแกรม MRTG

โปรแกรม MRTG (Multi Router Traffic Grapher) เป็นโปรแกรมที่ใช้เป็นเครื่องมือในการวัดและตรวจสอบปริมาณการจราจรในระบบเครือข่าย โดยจะแสดงปริมาณการจราจรในรูปแบบของกราฟ และมีการบันทึกสถิติในแฟ้มบันทึกปริมาณการจราจร (log-file)

ผู้ริเริ่มในการพัฒนาโปรแกรม MRTG ขึ้นก็คือ Tobias Oetiker และ Dave Rand เพื่อตรวจสอบปริมาณการจราจรของเราท์เตอร์ (router) แต่ต่อมาได้ถูกพัฒนาขึ้นให้เป็นเครื่องมือที่สามารถสร้างกราฟและบันทึกสถิติซึ่งสามารถใช้งานได้กับอุปกรณ์เกือบทุกประเภท



ภาพที่ 2.11 กราฟแสดงปริมาณการจราจรที่ถูกสร้างขึ้นโดยโปรแกรม MRTG

โปรแกรม MRTG ใช้โปรโตคอล SNMP ในการร้องขอข้อมูลโดยการส่ง object indentifiers ที่ต้องการไปยังอุปกรณ์ ดังนั้นอุปกรณ์ดังกล่าวจะต้องสามารถสนับสนุนโปรโตคอล SNMP และมี management information base: MIB สำหรับใช้ในการค้นหา object identifier ที่กำหนดมาให้ หลังจากที่พบข้อมูลที่ต้องการแล้วอุปกรณ์ก็จะส่งข้อมูลกลับไปยังโปรแกรม MRTG โดยใช้โปรโตคอล SNMP ด้วยเช่นกัน เมื่อโปรแกรม MRTG ได้รับข้อมูลที่ต้องการแล้วก็จะบันทึกข้อมูลลงในล็อกไฟล์ (log-file) และทำการสร้างไฟล์เอชทีเอ็มแอล (HTML) จากข้อมูลที่ได้รับ ซึ่งประกอบไปด้วยกราฟแสดงปริมาณการจราจรและรายละเอียดของอุปกรณ์นั้นๆ

แฟ้มบันทึกปริมาณการจราจร (log-file) ที่สร้างขึ้นโดยโปรแกรม MRTG ดังที่แสดงในภาพที่ 2.11 แบ่งออกเป็น 2 ส่วนคือ บรรทัดแรก และบรรทัดส่วนที่เหลือของแฟ้มทั้งหมด ซึ่งมีรายละเอียดดังนี้

บรรทัดแรกของแฟ้มบันทึกปริมาณการจราจร ประกอบไปด้วย 3 คอลัมน์ ซึ่งมีรายละเอียดคือ

คอลัมน์ที่ 1 บันทึกเวลาตามมาตรฐานของระบบยูนิกซ์ (UNIX)

คอลัมน์ที่ 2 บันทึกปริมาณข้อมูลขาออกทั้งหมด (หน่วยเป็นไบต์)

คอลัมน์ที่ 3 บันทึกปริมาณข้อมูลขาเข้าทั้งหมด (หน่วยเป็นไบต์)

บรรทัดส่วนที่เหลือของแฟ้มบันทึกปริมาณการจราจร ประกอบไปด้วย 5 คอลัมน์ มีรายละเอียดดังนี้

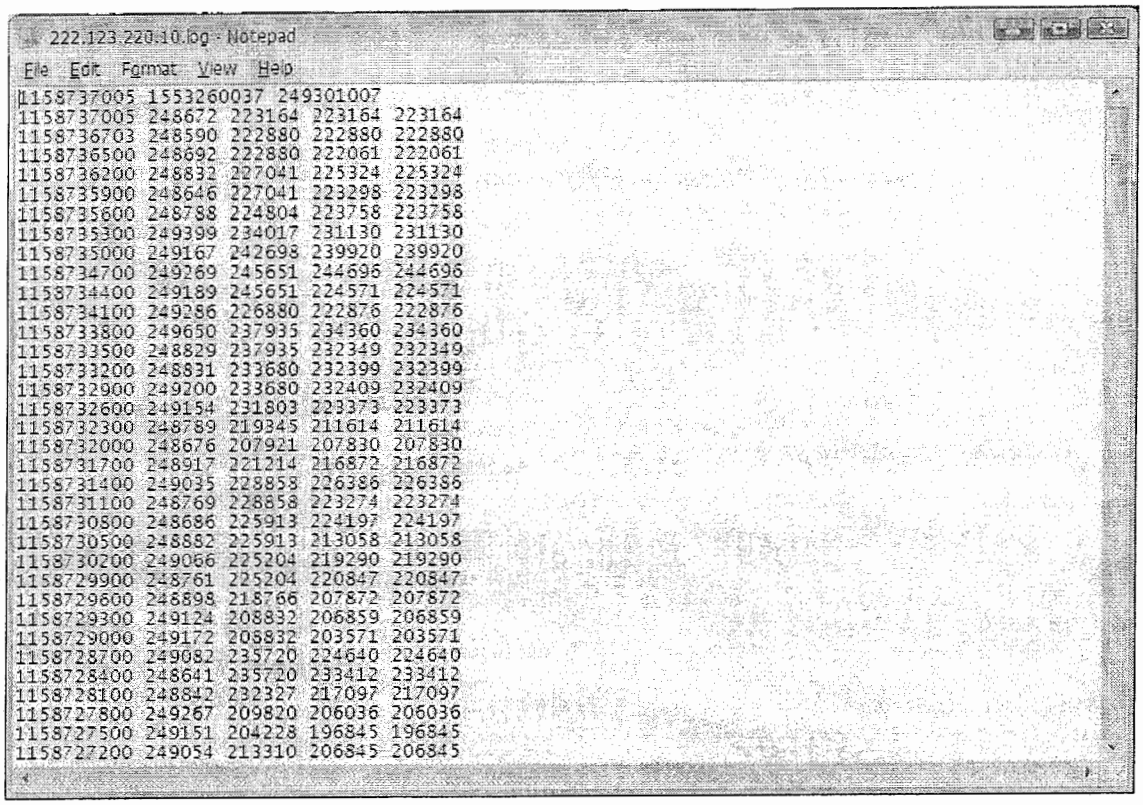
คอลัมน์ที่ 1 บันทึกเวลา ซึ่งในแต่ละบรรทัดห่างกันจะเป็นเวลา 5 นาที

คอลัมน์ที่ 2 บันทึกปริมาณข้อมูลเฉลี่ยขาเข้า (มีหน่วยเป็นไบต์ต่อวินาที)

คอลัมน์ที่ 3 บันทึกปริมาณข้อมูลเฉลี่ยขาออก (มีหน่วยเป็นไบต์ต่อวินาที)

คอลัมน์ที่ 4 บันทึกปริมาณสูงสุดของข้อมูลขาเข้า (มีหน่วยเป็นไบต์ต่อวินาที)

คอลัมน์ที่ 5 บันทึกปริมาณสูงสุดของข้อมูลขาออก (มีหน่วยเป็นไบต์ต่อวินาที)



ภาพที่ 2.12 แฟ้มบันทึกปริมาณการจราจรของโปรแกรม MRTG

2.4.2 โปรแกรมมิวทอเร้นท์ (μTorrent)

มิวทอเร้นท์ (μTorrent) เป็นโปรแกรมสำหรับเครื่องลูกข่ายที่ใช้ในการการรับและส่งแฟ้มข้อมูลบนเครือข่ายอินเตอร์เน็ตโดยใช้โปรโตคอล BitTorrent ซึ่งพัฒนาขึ้นโดย Bram Cohen ในปีพ.ศ. 2544 โดยมีหลักการทำงานคือทุกเครื่องในเครือข่ายจะมีสิทธิเท่าเทียมกันในการรับและส่งข้อมูล ข้อมูลที่ต้องการส่งจะถูกแบ่งออกเป็นชิ้นและจะถูกบรรจุไว้ใน TCP sockets ซึ่งเป็นลำดับชั้นย่อยของโปรโตคอล TCP/IP เพื่อส่งข้อมูลไปให้กับเครื่องลูกข่าย เมื่อเครื่องลูกข่ายได้รับชิ้นส่วนของข้อมูลแล้วก็จะส่งไปให้กับเครื่องลูกข่ายที่ต้องการข้อมูลชุดเดียวกันเช่นนี้ไปเรื่อยๆ จึงทำให้สามารถรับและส่งข้อมูลในขณะเดียวกันได้อย่างรวดเร็ว

การที่จะส่งข้อมูลด้วยโปรโตคอล BitTorrent ผู้ส่งจะต้องสร้างแฟ้มข้อมูล torrent ขนาดเล็กขึ้นมาหนึ่งแฟ้ม เช่น MyFile.torrent เป็นต้น ซึ่งแฟ้มข้อมูลนี้จะเป็นที่รวบรวมรายละเอียดต่างๆ ที่เกี่ยวข้องกับข้อมูลที่ต้องการส่ง ส่วนผู้ที่ต้องการรับข้อมูลจะต้องเปิดแฟ้มข้อมูลนี้ด้วยโปรแกรมลูกข่ายของโปรโตคอล BitTorrent เพื่อทำการรับชิ้นส่วนของแฟ้มข้อมูลจากเครื่องอื่นที่มีข้อมูลอยู่ และส่งชิ้นส่วนที่ตนมีอยู่ไปให้กับผู้อื่นที่ยังไม่ได้รับชิ้นส่วยนี้ไปเรื่อยๆ จนได้รับข้อมูลครบทั้งหมด

สำหรับความเร็วในการรับและส่งข้อมูลนั้นโดยทั่วไปจะขึ้นอยู่กับความเร็วในการเชื่อมต่อของวงจรสื่อสาร เช่น วงจรสื่อสารที่มีความเร็วในการรับข้อมูล (download) 1,024 Kbps และส่งข้อมูล (upload) 512 Kbps ก็จะมีความเร็วในการรับข้อมูลด้วยโปรโตคอล BitTorrent เป็น 128 Kbps และส่งข้อมูลเป็น 64 Kbps โดยประมาณ

2.5 งานวิจัยที่เกี่ยวข้อง

จากการค้นคว้าเอกสารงานวิจัยที่เกี่ยวข้องกับการจัดช่องทางการสื่อสารแบบสมดุล ผู้จัดทำได้ทำการค้นคว้างานวิจัยดังต่อไปนี้

Der-Chiang Li, Chihsen Wu และ Fengming M. Chang ได้กล่าวถึงวิธีการในการแก้ปัญหาการจราจรในช่องทางการสื่อสารด้วยซอฟต์แวร์ที่ใช้อัลกอริทึม Dynamic Weighted Round-Robin ซึ่งมีพื้นฐานมาจาก Round-Robin โดยที่อัลกอริทึม Dynamic Weighted Round-Robin ได้ถูกพัฒนาขึ้นเพื่อใช้ในการควบคุมปริมาณการจราจรของข้อมูลในเครือข่าย ที่มีหลายช่องทางการสื่อสาร (multiple-link network) โดยการใช้ฟังก์ชั่นทางคณิตศาสตร์ที่ถูกพัฒนาขึ้นเพื่อคาดคะเนเวลาที่ใช้ในการตรวจสอบปริมาณการจราจรที่เหมาะสม จึงเป็นการลดภาระการทำงานของระบบลงได้ และผลที่ได้จากงานวิจัยดังกล่าวได้แสดงให้เห็นว่าอัลกอริทึม Dynamic Weighted Round-Robin นั้นมีประสิทธิภาพมากกว่าอัลกอริทึม Weighted Round-Robin และ Round-Robin ตามลำดับ

P. Levillain[17] ได้กล่าวถึงการรักษาสมดุลของปริมาณการจราจรให้กับเครื่องแม่ข่าย (server load balancing) ด้วยซอฟต์แวร์ โดยบริหารจัดการข้อมูลในระดับแพ็กเก็ต (packet) ด้วยวิธี Single packet rewriter และ Double packet rewriter ซึ่งจะทำการแก้ไข (rewrite) ข้อมูลในส่วนของ Virtual IP address ให้กับแพ็กเก็ต (packet) เพื่อกำหนดเส้นทางให้กับแพ็กเก็ต (packet) นั้นๆ แต่วิธีการดังกล่าวเป็นการใช้ซอฟต์แวร์ซึ่งต้องอาศัยการทำงานของหน่วยประมวลกลาง (processor) หน่วยความจำ (memory) และสื่อบันทึกข้อมูลสำรอง (disk) ในปริมาณมาก จึงไม่เหมาะกับระบบเครือข่ายความเร็วสูงในปัจจุบัน ดังนั้นจึงเสนอให้ใช้ Switch-based server load balancing ซึ่งเป็นฮาร์ดแวร์ในการเพิ่มประสิทธิภาพให้กับการรักษาสมดุลของปริมาณการจราจรให้กับเครื่องแม่ข่าย (server load balancing)

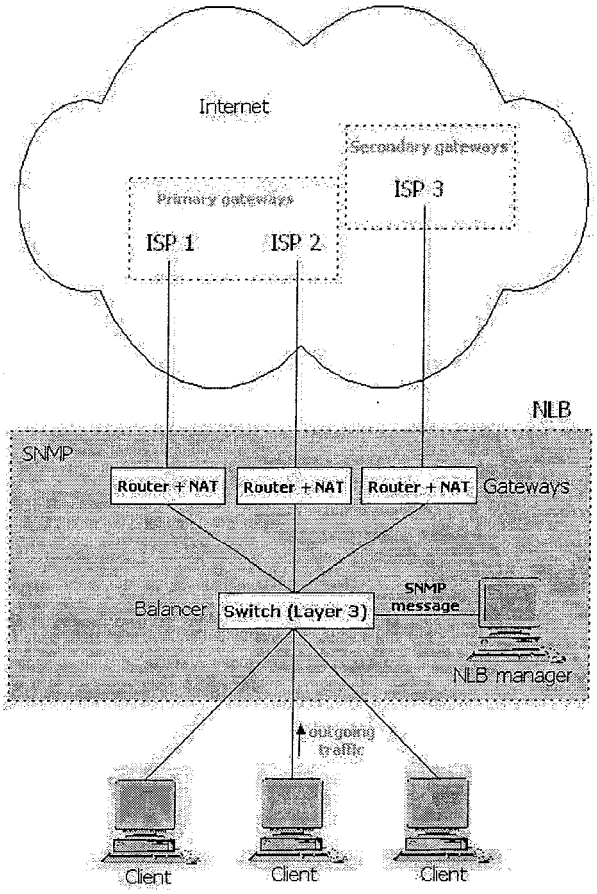
Mohammed H. Sqalli และ Shaik Sirajuddin[18] ได้เสนอการนำอัลกอริทึม Adaptive Load-balancing ร่วมกับ XML-based ในการประมวลผลแบบขนาน ซึ่งเป็นการประยุกต์ใช้ JPVM (Java Parallel Virtual Machine) ร่วมกับโปรโตคอล SNMP เพื่อกำหนดภาระงานให้กับหน่วยประมวลผล (processor) โดยมีวัตถุประสงค์ในการกระจายปริมาณภาระงาน (Load-balancing) เพื่อเพิ่มประสิทธิภาพในการประมวลผลและลดเวลาที่ใช้ติดต่อกับโหนด (node) ต่างๆ

สรุปจากการค้นคว้าเอกสารงานวิจัยดังกล่าว ทำให้เกิดแนวคิดในการจัดช่องทางการสื่อสารแบบสมดุล โดยการพัฒนาซอฟต์แวร์ควบคุมอุปกรณ์เครือข่ายเพื่อจัดปริมาณการจราจรให้มีความสมดุลกัน โดยใช้โปรโตคอล SNMP ในการตรวจสอบสถานะและปริมาณการจราจรในช่องทางการสื่อสาร ตลอดจนการนำอัลกอริทึม Round-Robin, Weighted Round-Robin และ Dynamic Weighted Round-Robin มาประยุกต์ใช้และเปรียบเทียบถึงประสิทธิภาพในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสาร

บทที่ 3
วิธีดำเนินการวิจัย

3.1 การจัดสภาพแวดล้อมการทำงานของช่องทางการสื่อสารแบบสมดุล

เนื่องจากงานวิจัยนี้ต้องอาศัยการทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์ โดยอาศัยการนำอุปกรณ์เครือข่ายมาใช้ร่วมกับการพัฒนาซอฟต์แวร์ในการบริหารจัดการเครือข่ายแบบสมดุล ดังนั้นจึงจะได้กล่าวถึงการจัดสภาพแวดล้อมการทำงานของระบบโดยอาศัยแบบจำลองดังที่แสดงในภาพที่ 3.1



ภาพที่ 3.1 แบบจำลองแสดงสภาพแวดล้อมของการจัดช่องทางการสื่อสารแบบสมดุล

จากภาพดังกล่าว แสดงถึงสภาพแวดล้อมของการจัดช่องทางการสื่อสารแบบสมดุล (Network Load Balance: NLB) ภายใต้การทำงานของ SNMP ซึ่งเป็นโปรโตคอลที่ใช้บริหารจัดการ

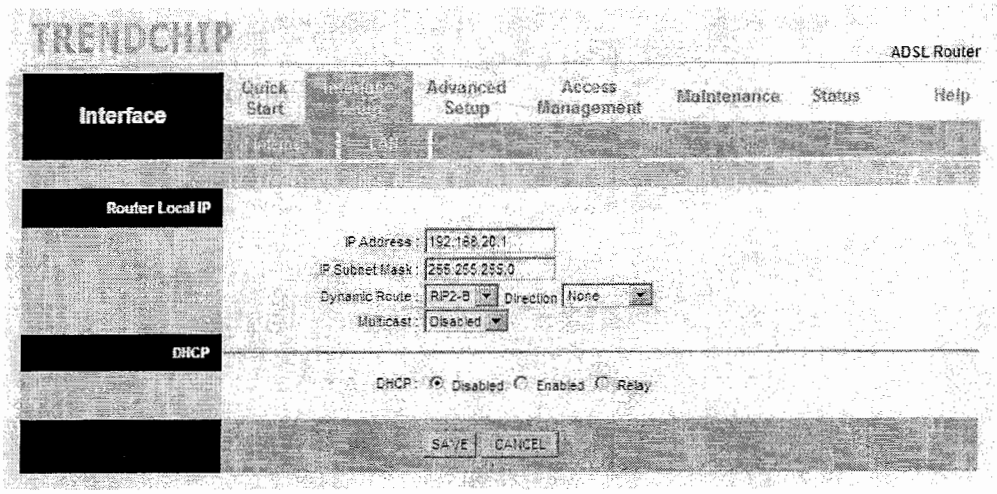
อุปกรณ์ต่างๆ ในระบบเครือข่าย TCP/IP ดังนั้นอุปกรณ์ที่ใช้ในระบบ ไม่ว่าจะเป็น เราท์เตอร์ (router) สวิทช์ (switch) หรือเครื่องคอมพิวเตอร์ จะต้องสนับสนุนคำสั่งต่างๆ ของ SNMP และสำหรับเครื่องคอมพิวเตอร์ที่นำมาใช้ควบคุมช่องทางการสื่อสาร (NLB manager) จะต้องมีการติดตั้งโปรโตคอลดังกล่าวลงในระบบปฏิบัติการ เพื่อให้สามารถติดต่อกับอุปกรณ์เครือข่ายที่อยู่ภายในระบบได้

สำหรับช่องทางการสื่อสารที่เชื่อมต่อกับเครือข่ายอินเตอร์เน็ตในงานวิจัยนี้ แบ่งออกเป็น 2 ลักษณะคือ ช่องทางการสื่อสารหลัก (primary gateway) และช่องทางการสื่อสารสำรอง (secondary gateway) โดยที่ช่องทางการสื่อสารหลัก (primary gateway) จะเป็นช่องทางการสื่อสารที่ถูกใช้งานในสภาวะปกติ ส่วนช่องทางการสื่อสารสำรอง (secondary gateway) จะเป็นช่องทางการสื่อสารที่ถูกนำมาใช้ในกรณีที่ช่องทางการสื่อสารหลัก (primary gateway) ไม่สามารถใช้งานได้ ซึ่งช่องทางการสื่อสารแต่ละช่องทางจะต้องมีอุปกรณ์ในการเชื่อมต่อก็คือ เราท์เตอร์ (router) ซึ่งมีการเปิดใช้งานฟังก์ชัน NAT: Network Address Translation เพื่อให้สามารถแปลงหมายเลข IP Address เครือข่ายภายในให้เป็น IP Address ของผู้ให้บริการอินเตอร์เน็ต โดยที่เราท์เตอร์ (router) แต่ละตัวจะถูกนำมาเชื่อมต่อกับสวิทช์ (switch) ซึ่งในงานวิจัยนี้ได้ใช้สวิทช์ (switch) เลเยอร์ 3 ซึ่งเป็นอุปกรณ์ที่สามารถกำหนด IP Address และ routing table โดยที่สวิทช์ (switch) ที่นำมาใช้นี้จะถูกควบคุมโดยโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล (NLB manager) เพื่อทำหน้าที่ในการรักษาสมดุลของปริมาณการจราจรในขาออก (outgoing-traffic) ให้กับช่องทางการสื่อสาร โดยการกำหนด default gateway ที่เหมาะสมในการส่งข้อมูล ซึ่งขึ้นอยู่กับปริมาณการจราจรในช่องทางการสื่อสารที่มีอยู่ในขณะนั้น

ในส่วนของอุปกรณ์ที่ใช้ในการทดลองดังที่กล่าวมานั้น ต้องมีการติดตั้งและกำหนดค่าต่างๆ ในแต่ละส่วนแตกต่างกันไป จึงจะได้อธิบายถึงขั้นตอนในการติดตั้งดังต่อไปนี้

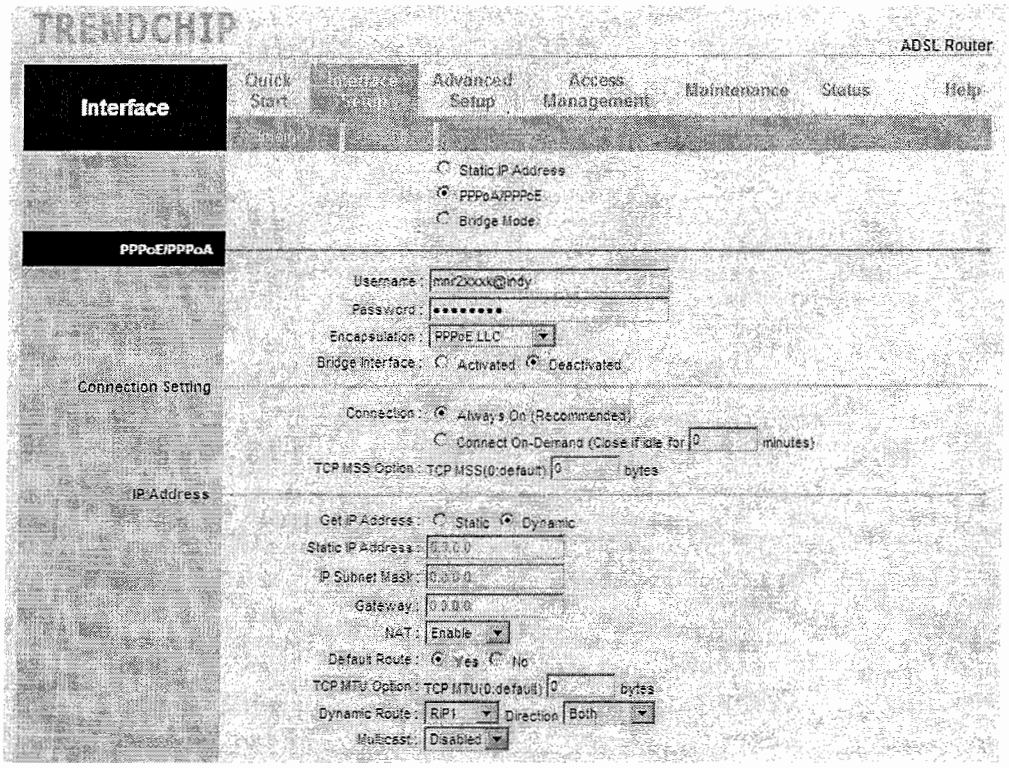
3.1.1 ADSL Router

เราท์เตอร์ (router) เป็นอุปกรณ์เครือข่ายที่ใช้ในการเชื่อมต่อระหว่างเครือข่ายภายในกับเครือข่ายภายนอก ซึ่งในที่นี้ก็คือเครือข่ายอินเตอร์เน็ต เนื่องจากวงจรเครือข่ายที่ใช้ในการทดลองเป็นวงจรคู่สายเช่าแบบ ADSL: Asynchronous Digital Subscriber Line ดังนั้นจึงใช้ ADSL Router เป็นอุปกรณ์ในการเชื่อมต่อกับเครือข่ายอินเตอร์เน็ต แต่เนื่องจากวงจรแบบ ADSL ที่ใช้ในการการทดลองนั้นเป็นลักษณะของ Dynamic IP Address จึงทำให้ IP Address ที่เราท์เตอร์ (router) ได้รับจากการเชื่อมต่อในแต่ละครั้งแตกต่างกันได้ สำหรับการกำหนดค่าต่างๆ ให้กับเราท์เตอร์ (router) เช่น IP Address ตลอดจนชื่อผู้ใช้และรหัสผ่านในการใช้งานตามปกติแล้ว จะต้องมีการกำหนดลักษณะการทำงานต่างๆ ของเราท์เตอร์ (router) ดังแสดงในภาพที่ 3.2 และภาพที่ 3.3



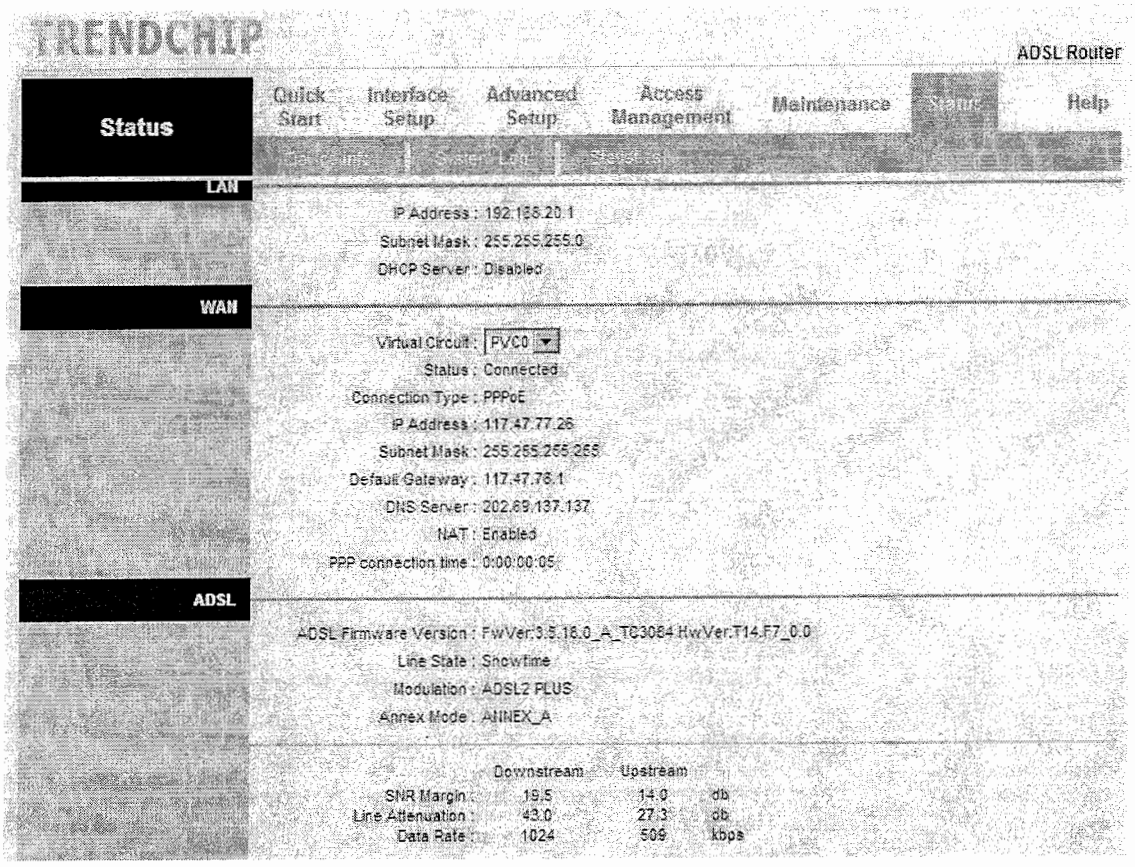
ภาพที่ 3.2 การกำหนด IP Address และ Subnet Mask

การกำหนด IP Address ให้กับเราท์เตอร์ (router) ในการทดลองนี้ได้ใช้ IP Address ซึ่งเป็นเครือข่ายภายใน โดยที่เราท์เตอร์ (router) แต่ละตัวจะต้องมีหมายเลขไอพีต่างเครือข่ายกัน เช่น เราท์เตอร์ (router) ตัวที่หนึ่งกำหนดหมายเลขไอพีเป็น 192.168.10.1 เราท์เตอร์ (router) ตัวที่สองกำหนดเป็น 192.168.20.1 เป็นต้น โดยจากภาพที่ 3.2 เป็นการกำหนด IP Address ของเราท์เตอร์ (router) เป็น 192.168.20.1 และกำหนด Subnet mask เป็น 255.255.255.0



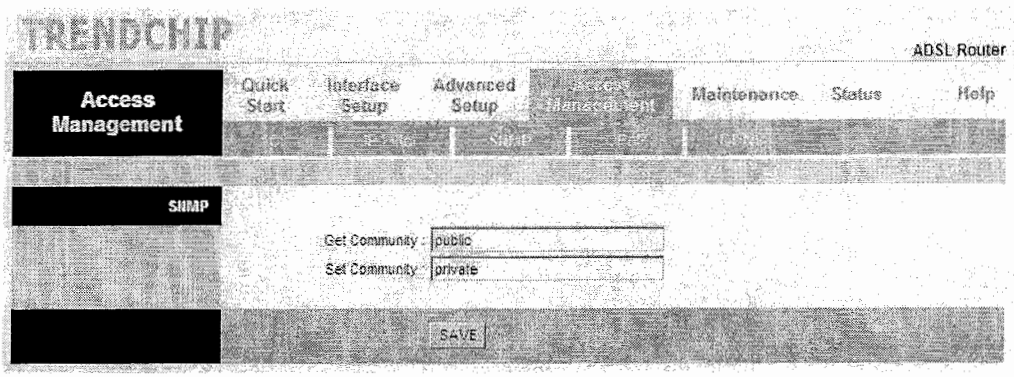
ภาพที่ 3.3 การกำหนดชื่อผู้ใช้และรหัสผ่าน

เมื่อการกำหนด IP Address ให้กับเราท์เตอร์ (router) แล้ว ขั้นตอนต่อไปก็คือการกำหนดชื่อผู้ใช้และรหัสผ่านดังในภาพที่ 3.3 และทำการเชื่อมต่อกับผู้ให้บริการอินเตอร์เน็ต จากนั้นขั้นตอนต่อไปก็คือการตรวจสอบสถานะของการเชื่อมต่อและนำ IP Address ที่ได้รับ เพื่อนำไปกำหนดค่าในตารางการจัดเส้นทาง (routing table) ให้กับสวิทช์ดังที่จะกล่าวถึงในหัวข้อต่อไป



ภาพที่ 3.4 สถานะของเราท์เตอร์ (router) และ IP Address ที่กำหนดให้โดยผู้ให้บริการอินเตอร์เน็ต

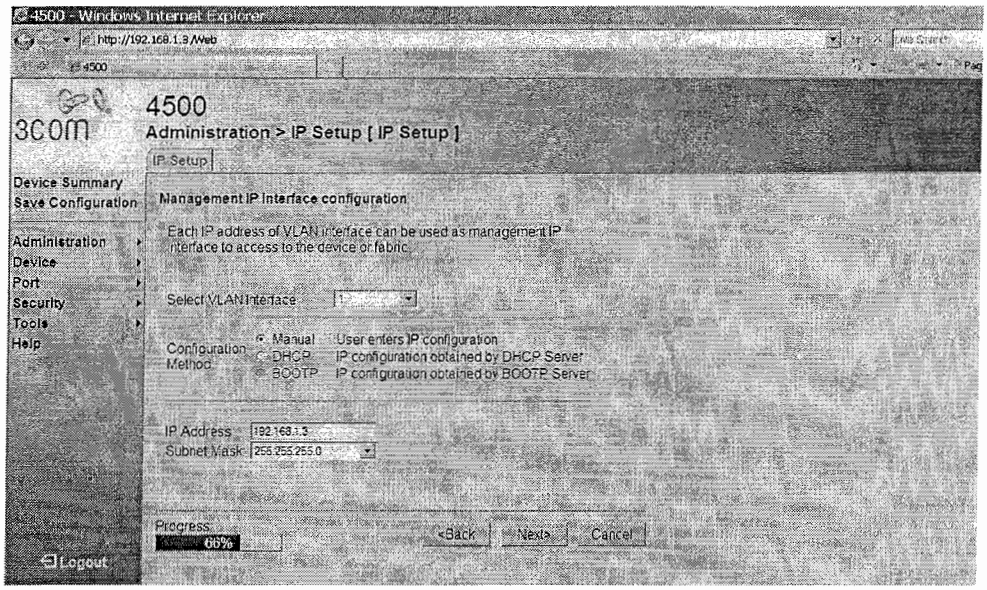
ขั้นตอนสุดท้ายของการกำหนดค่าให้กับเราท์เตอร์ (router) ก็คือ การกำหนด community ของ SNMP ให้กับเราท์เตอร์ (router) ซึ่ง community ที่กำหนดนี้ จะถูกนำไปกำหนดในโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล ร่วมกับการกำหนด IP Address ที่เราท์เตอร์ (router)ได้รับจากผู้ให้บริการอินเตอร์เน็ต เพื่อที่โปรแกรมจัดช่องทางการสื่อสารแบบสมดุลจะได้นำไปใช้ในการติดต่อกับเราท์เตอร์ (router) เพื่อตรวจสอบปริมาณการจราจรในช่องทางการสื่อสารได้ ซึ่งในที่นี้ได้กำหนด Get Community เป็น “public” และ Set Community เป็น “private” ดังแสดงในภาพที่ 3.5



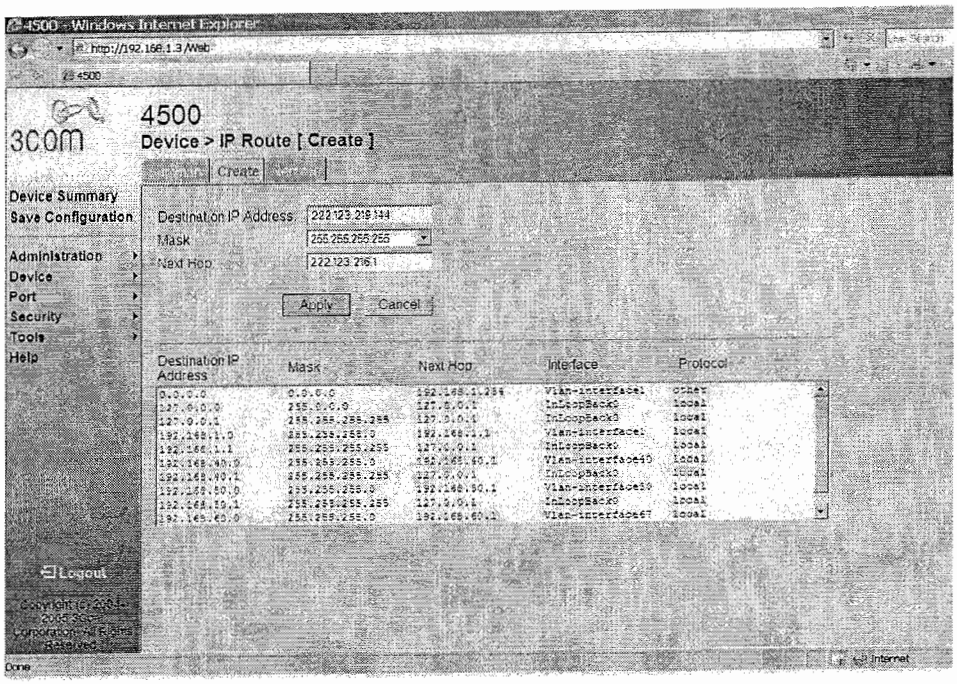
ภาพที่ 3.5 การกำหนด community ของ SNMP ให้กับเราท์เตอร์ (router)

3.1.2 สวิทช์ (Switch)

เมื่อทำการเชื่อมต่อเราท์เตอร์ (router) กับผู้ให้บริการอินเตอร์เน็ตแล้ว ขั้นตอนต่อไปก็จะเป็นการกำหนด IP Address ให้กับสวิทช์ (switch) ดังภาพที่ 3.5 โดยสวิทช์ (switch) ที่นำมาใช้นั้นจะต้องเป็นสวิทช์ (switch) ที่ทำงานอยู่ในลำดับชั้นที่ 3 ของแบบจำลอง OSI 7 layers ซึ่งมีความสามารถในการกำหนดตารางเส้นทาง (routing table) ด้วย IP Address ได้ และนอกจากนี้สวิทช์ (switch) ดังกล่าวยังจะต้องสนับสนุนโปรโตคอล SNMP เพื่อที่โปรแกรมจัดช่องทางการสื่อสารแบบสมดุลที่ได้พัฒนาขึ้น จะได้สามารถกำหนดช่องทางการสื่อสารให้กับเครือข่ายภายในสำหรับเป็นเส้นทางในการติดต่อกับเครือข่ายอินเตอร์เน็ต โดยทั้งนี้จะต้องมีการนำ IP Address ของเราท์เตอร์ (router) มากำหนดไว้ในตารางเส้นทาง (routing table) ของสวิทช์ (switch) ดังภาพที่ 3.6

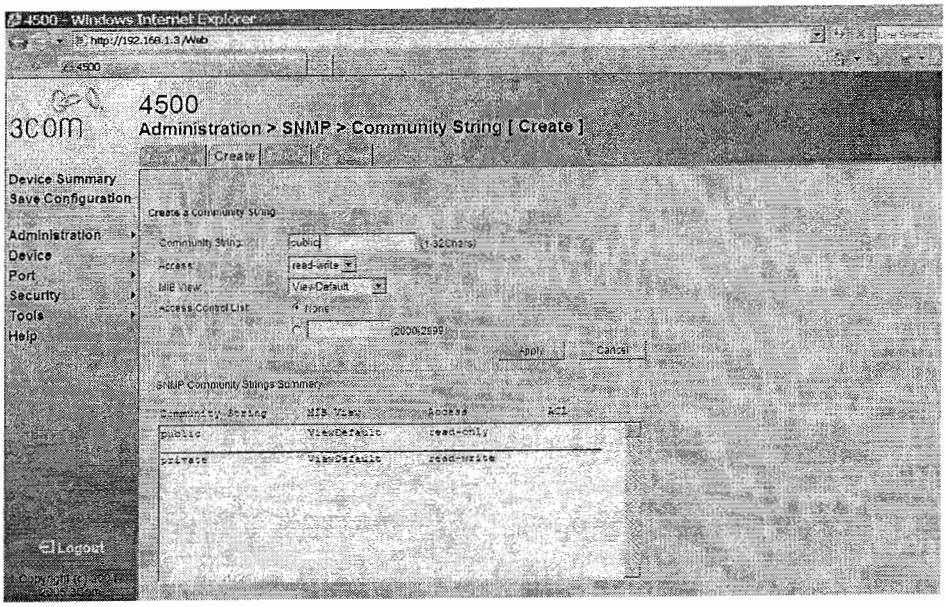


ภาพที่ 3.6 การกำหนด IP Address และ Subnet Mask ให้กับสวิทช์ (switch)



ภาพที่ 3.7 การกำหนดตารางเส้นทาง (routing table) ให้กับสวิทช์ (switch)

สำหรับขั้นตอนสุดท้ายในการกำหนดค่าให้กับเราท์เตอร์ (router) ก็คือ การกำหนด community ให้กับสวิทช์ (switch) เช่นเดียวกันกับเราท์เตอร์ (router) แต่สิ่งที่ต่างกันก็คือ โปรแกรมจัดช่องทางการสื่อสารแบบสมดุลจะต้องใช้ทั้งคอมมูนิตี้ในการอ่านค่า (Get Community) และคอมมูนิตี้ในการกำหนดค่า (Set Community) ในการติดต่อและควบคุมการทำงานของสวิทช์ (switch)



ภาพที่ 3.8 การกำหนด community ให้กับสวิทช์ (switch)

นอกจากนี้ หน้าที่สำคัญอีกประการหนึ่งของสวิทช์ (switch) ในงานวิจัยนี้ นอกจากการเป็นอุปกรณ์ในการเชื่อมต่อเราท์เตอร์ (router) และคอมพิวเตอร์แล้วก็คือ การทำหน้าที่เป็นอุปกรณ์ในการรักษาสมดุลของปริมาณการจราจรในเครือข่าย หรือที่เรียกว่า "Balancer" ซึ่งถูกควบคุมโดยโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล โดยที่โปรแกรมจัดช่องทางการสื่อสารแบบสมดุลจะทำการตรวจสอบปริมาณการจราจรในช่องทางการสื่อสาร แล้วนำมากำหนดช่องทางการสื่อสารที่เหมาะสมในการส่งข้อมูลออกจากเครือข่าย (outgoing traffic) โดยการกำหนดค่า default gateway ของสวิทช์ (switch) ให้ชี้ไปยังช่องทางการสื่อสารที่ต้องการใช้งาน ซึ่งจะต้องมีการตรวจสอบ object ientifier ที่เป็น default gateway ของสวิทช์ (switch) แล้วจึงนำค่าของ object identifer ดังกล่าวไปกำหนดในโปรแกรมจัดช่องทางการสื่อสารแบบสมดุลต่อไป

3.2 การพัฒนาและการใช้งานโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล

การพัฒนาโปรแกรมจัดช่องทางการสื่อสารแบบสมดุลในงานวิจัยนี้ ได้ใช้โปรแกรมภาษา C# ซึ่งเป็นภาษาหนึ่งในโปรแกรม Microsoft Visual Studio .NET 2003 และไลบรารี SNMP++ .NET API[19] ที่พัฒนาโดย Johen Katz และ Frank Fock ซึ่งได้นำไลบรารี SNMP++ ของ Peter Mellquist มาปรับปรุงให้สามารถใช้คำสั่งใน SNMP ทุกคำสั่ง

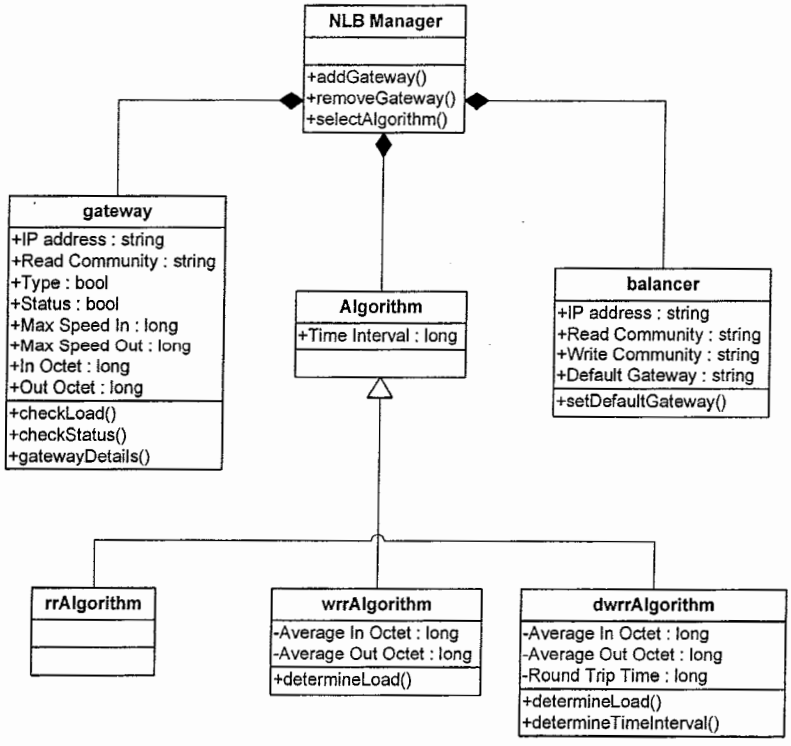
สำหรับการออกแบบโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล สามารถแสดงเป็นคลาสไดอะแกรม (class diagram) ได้ดังภาพที่ 3.9 โดยโปรแกรมประกอบด้วยคลาสหลักคือ NLB manager ซึ่งทำหน้าที่ในการติดต่อกับผู้ใช้โปรแกรมเพื่อความสะดวกในการกำหนดค่าในการใช้งานต่างๆ ของโปรแกรม เช่น การเพิ่มช่องทางการสื่อสาร (addGateway) การลบช่องทางการสื่อสาร (removeGateway) การกำหนดอัลกอริทึมในการรักษาสมดุลของปริมาณการจราจร (selectAlgorithm) และทำหน้าที่ติดต่อกับคลาสอื่นๆ ที่อยู่ในโปรแกรม เป็นต้น

ส่วนคลาสเกตเวย์ (gateway) จะเป็นคลาสของช่องทางการการสื่อสารซึ่งประกอบไปด้วยแอดทริบิวต์ (attribute) ต่างๆ เช่น IP address, Read Community และประเภท (Type) ของช่องทางการสื่อสาร เป็นต้น ซึ่งประเภทของช่องทางการสื่อสารมีอยู่ 2 ลักษณะคือ ช่องทางการสื่อสารหลัก (Primary Gateway) และช่องทางการสื่อสารสำรอง (Secondary Gateway) และภายในคลาสยังประกอบไปด้วยโอเปอเรชัน (operation) ต่างๆ คือ การตรวจสอบปริมาณการจราจร (checkLoad) การตรวจสอบสถานะ (checkStatus) และการแสดงรายละเอียดของช่องทางการสื่อสาร (gatewayDetails)

สำหรับคลาสอัลกอริทึม (Algorithm) ซึ่งเป็นคลาสที่มีความสัมพันธ์ในลักษณะของการสืบทอดคุณสมบัติ (Inheritance) ระหว่างคลาสแม่ (superclass) คือคลาสอัลกอริทึม (Algorithm) กับ

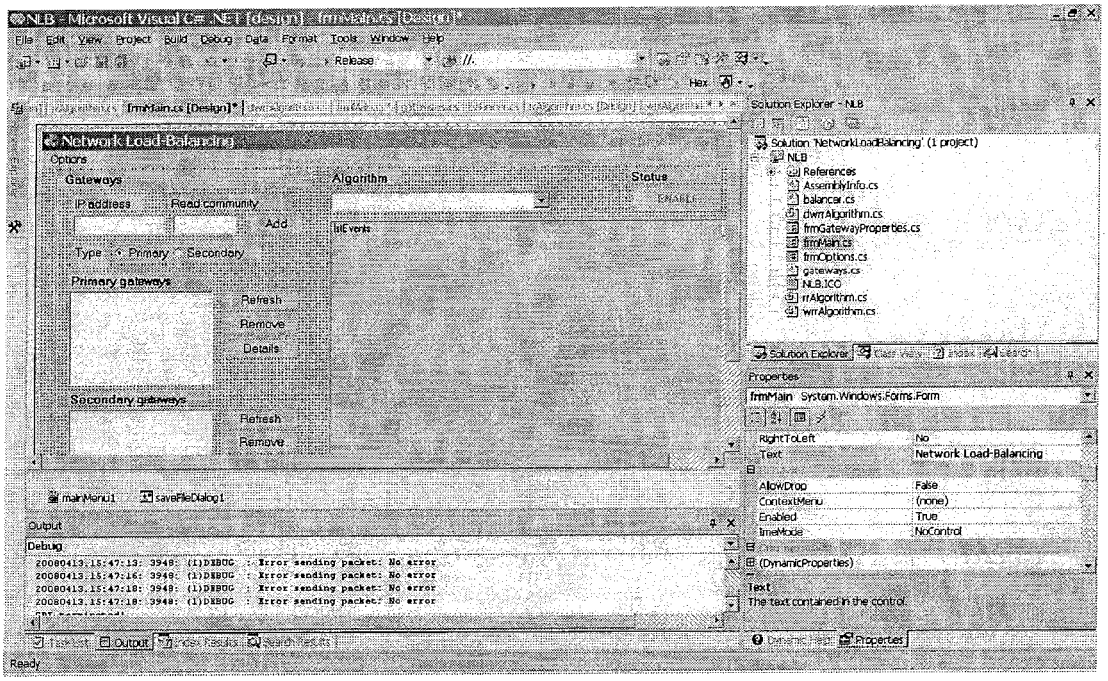
คลาสลูก (subclass) คือ คลาส rrAlgorithm ซึ่งจะทำการรักษาสมดุลของปริมาณการจราจรในเครือข่ายด้วยอัลอริทึม Round-Robin โดยจะทำงานตามรอบเวลา (Time Interval) ที่ผู้ใช้กำหนด ส่วนคลาส wrrAlgorithm จะทำการรักษาสมดุลของปริมาณการจราจรในเครือข่ายด้วยอัลอริทึม Weighted Round-Robin โดยจะมีการตรวจสอบปริมาณการจราจรในช่องทางการสื่อสารเพื่อนำมากำหนดค่าน้ำหนัก (weight) ของภาระงานให้กับในแต่ละช่องทางการสื่อสารตามความเหมาะสมด้วยคลาส (determineLoad) กล่าวคือ ถ้าช่องทางการสื่อสารใดมีปริมาณการจราจรน้อยก็จะถูกกำหนดให้มีการใช้งานมาก ส่วนช่องทางที่มีการจราจรมากก็จะถูกกำหนดให้ใช้งานน้อยลง และสำหรับคลาส dwrrAlgorithm นั้นก็มีพื้นฐานการทำงานเช่นเดียวกันกับคลาส Weighted Round-Robin แต่ได้เพิ่มความสามารถในการคำนวณรอบเวลาการทำงานด้วยคลาส determineTimeInterval ให้มีความเหมาะสมกับปริมาณการจราจรมากขึ้น

ในส่วนของคลาส balancer ก็คือคลาสของสวิทช์ (switch) ที่ทำหน้าที่เป็น balancer ซึ่งประกอบไปด้วยแอดทริบิวต์ต่างๆ คือ IP address, Read Community, Write Community และ Default Gateway และโอเปอเรชั่น setDefaultGateway ที่ใช้ในการกำหนดช่องทางการสื่อสาร



ภาพที่ 3.9 คลาสไดอะแกรม (class diagram) ของโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล

จากคลาสไดอะแกรมในภาพที่ 3.9 ได้นำมาพัฒนาเป็นโปรแกรมการจัดช่องทางการสื่อสารแบบสมดุล ด้วยโปรแกรม Microsoft Visual Studio .NET 2003 ได้ดังภาพที่ 3.10

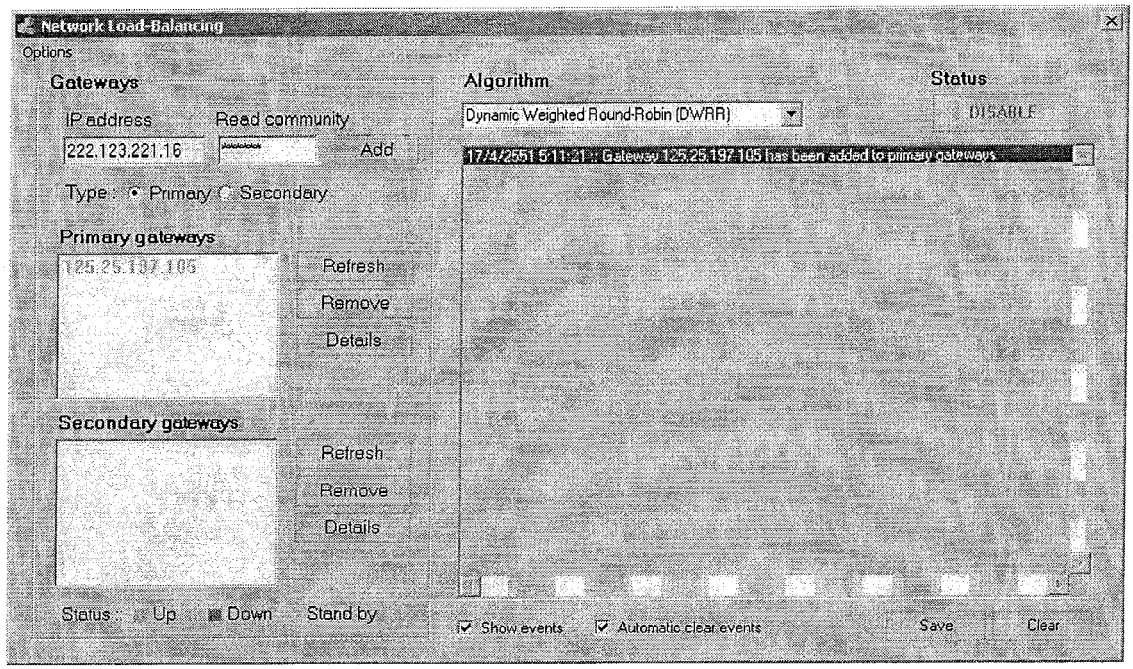


ภาพที่ 3.10 การพัฒนาโปรแกรมการจัดช่องทางการสื่อสารแบบสมดุลด้วยโปรแกรม Microsoft Visual Studio .NET 2003

เมื่อได้ทำการพัฒนาและทดสอบโปรแกรมเป็นที่เรียบร้อยแล้ว จึงได้ทำการติดตั้งโปรแกรมการจัดช่องทางการสื่อสารแบบสมดุลในเครื่องคอมพิวเตอร์ที่จะใช้ในการตรวจสอบและควบคุมปริมาณการจราจรในช่องทางการสื่อสาร ซึ่งใช้ระบบปฏิบัติการ Microsoft Windows Server 2003 ซึ่งได้ทำการติดตั้งโปรแกรม Microsoft .NET Framework 2.0 และ SNMP จากนั้นจึงทำการกำหนดค่าต่างๆ ภายในโปรแกรมจัดช่องทางการสื่อสารแบบสมดุลเพื่อเตรียมความพร้อมในการใช้งานดังต่อไปนี้

3.2.1 กำหนดช่องทางการสื่อสาร

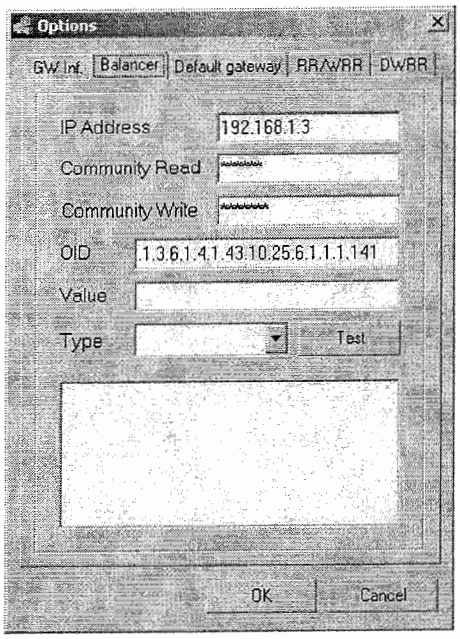
การกำหนดช่องทางการสื่อสารให้กับโปรแกรมจัดช่องทางการสื่อสารแบบสมดุลจะต้องทำการนำหมายเลขไอพี (IP Address) ของเราท์เตอร์ (router) ที่ใช้ติดต่อผู้ให้บริการอินเตอร์เน็ต ดังภาพที่ 3.4 มากำหนดลงในช่อง IP Address พร้อมกับระบุ Read Community ที่ได้กำหนดไว้ให้กับเราท์เตอร์ (router) ดังในภาพที่ 3.5 และกำหนดประเภท (Type) ของช่องทางการสื่อสารให้เป็นช่องทางหลัก (Primary) ดังภาพที่ 3.11



ภาพที่ 3.11 การกำหนดช่องทางการสื่อสารให้กับโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล

3.2.2 การกำหนดค่า Balancer

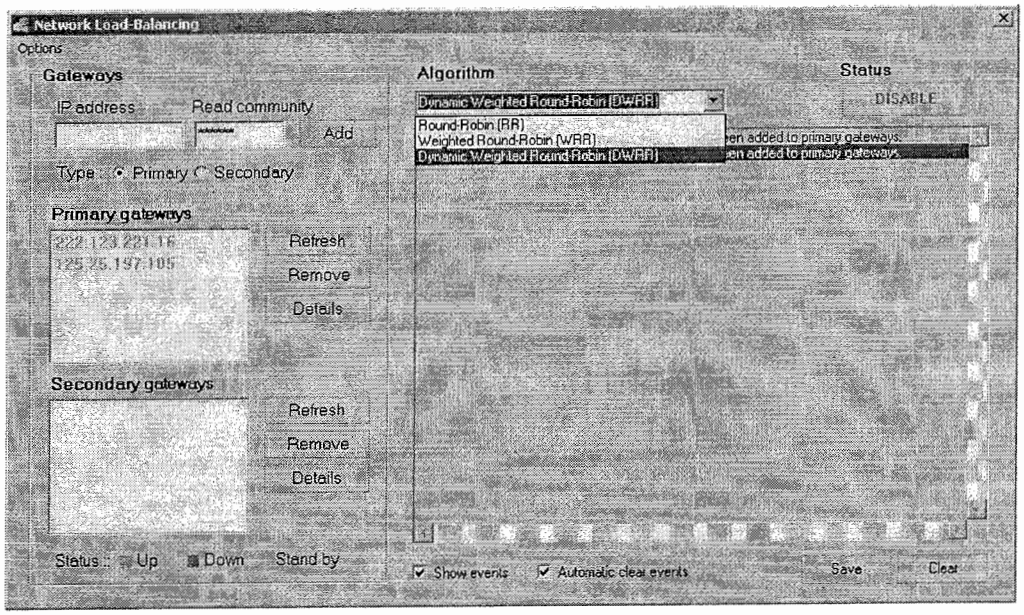
เมื่อทำการกำหนดช่องทางการสื่อสารสารทั้งหมดที่จะนำมาใช้เป็นที่เรียบร้อย แล้วขั้นตอนต่อไปก็คือการกำหนดค่าให้กับ Balancer โดยกำหนดได้จากเมนู Options ของโปรแกรม โดยค่าที่จะต้องทำการกำหนดคือ IP Address ของสวิทช์ที่ทำหน้าที่เป็น Balancer, Read Community และ Write Community ที่ได้กำหนดไว้ให้กับสวิทช์ (switch) ดังภาพที่ 3.8 ซึ่งก็คือ "public" และ "private" ตามลำดับ และค่าสุดท้ายที่ต้องกำหนดก็คือหมายเลข object identifier ซึ่งเป็นค่าที่อยู่ใน Management Information Base (MIB) ของสวิทช์ (switch) ซึ่งใช้กำหนดช่องทางการสื่อสารในการส่งข้อมูลออกจากเครือข่าย หรือ Default Gateway ดังภาพที่ 3.12



ภาพที่ 3.12 การกำหนดค่าของ Balancer

3.2.3 กำหนดอัลกอริทึมในการทำงานของโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล

ขั้นตอนสุดท้ายในการกำหนดค่าการทำงานให้กับโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล ก็คือการกำหนดอัลกอริทึมที่ใช้ในการรักษาสมดุลของปริมาณการจราจรให้กับช่องทางการสื่อสาร ดังในภาพที่ 3.13 โดยกำหนดเป็น Round-Robin, Weighted Round-Robin หรือ Dynamic Weighted Round-Robin ได้ตามความต้องการ



ภาพที่ 3.13 การกำหนดอัลกอริทึมในการทำงานของโปรแกรมจัดช่องทางการสื่อสารแบบสมดุล

บทที่ 4
การทดลองและผลการทดลอง

การทดลองในงานวิจัยนี้ มีวัตถุประสงค์ในการทดสอบโปรแกรมจัดช่องทางการสื่อสารแบบสมดุลเพื่อเปรียบเทียบประสิทธิภาพของอัลกอริทึมแบบ Round-Robin (RR), Weighted Round-Robin (WRR) และ Dynamic Weighted Round-Robin (DWRR) ในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสาร และในด้านความสามารถในการหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ขัดข้อง

4.1 การทดลอง

การทดลองในงานวิจัยนี้จึงได้แบ่งออกเป็น 2 ส่วน โดยส่วนที่ 1 เป็นการทดลองบนเครือข่ายระบบปิด ซึ่งทำการทดลองบนเครือข่ายแลน (LAN: Local Area Network) และส่วนที่ 2 เป็นการทดลองบนเครือข่ายอินเตอร์เน็ตจริง ดังมีรายละเอียดดังต่อไปนี้

4.1.1 การทดลองบนเครือข่ายระบบปิด

การทดลองในส่วนที่ 1 เป็นการทดลองเพื่อเปรียบเทียบประสิทธิภาพของอัลกอริทึม Round-Robin, Weighted Round-Robin และ Dynamic Weighted Round-Robin ในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสารในสภาพแวดล้อมแบบปิด ซึ่งเป็นการทดลองที่กระทำบนระบบเครือข่ายแลน (LAN: Local Area Network) เนื่องจากเป็นระบบเครือข่ายที่สามารถควบคุมปัจจัยที่มีผลกระทบต่อการวัดประสิทธิภาพของอัลกอริทึมต่างๆ ได้ดีกว่าเครือข่ายอินเตอร์เน็ต ซึ่งมีปัจจัยที่ไม่สามารถควบคุมได้อยู่เป็นจำนวนมาก โดยประกอบด้วยการทดลองดังต่อไปนี้

การทดลองที่ 1 การจัดช่องทางการสื่อสารโดยใช้อัลกอริทึม Round-Robin

การทดลองที่ 2 การจัดช่องทางการสื่อสารโดยใช้อัลกอริทึม Weighted Round-Robin

การทดลองที่ 3 การจัดช่องทางการสื่อสารโดยใช้อัลกอริทึม Dynamic Weighted Round-Robin

อุปกรณ์ที่ใช้ในการทดลองในส่วนที่ 1 มีดังต่อไปนี้

4.1.1.1 ส่วนประกอบของฮาร์ดแวร์

1) เครื่องคอมพิวเตอร์ส่วนบุคคล จำนวน 6 เครื่อง

2) สวิทช์ เลเยอร์ 3 (switch layer 3) จำนวน 1 ตัว

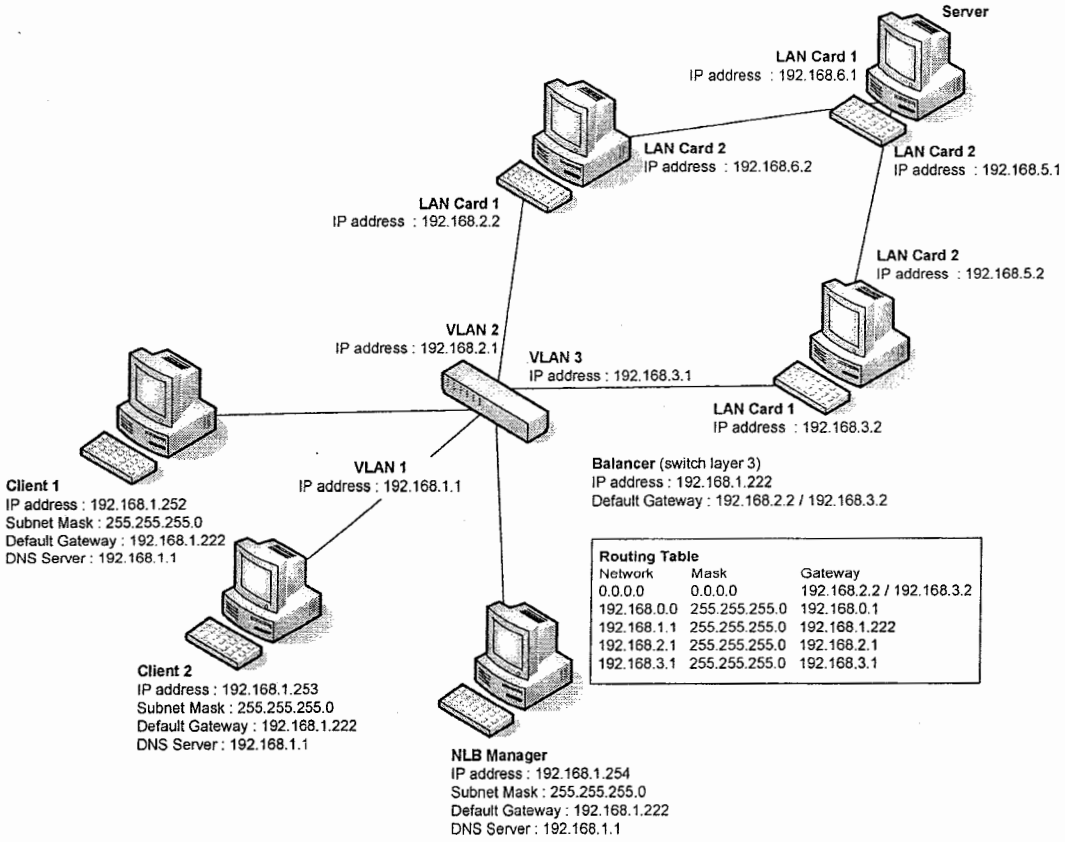
3) แผงวงจรเครือข่าย (network adapter) จำนวน 3 ใบ

4.1.1.2 ส่วนประกอบของซอฟต์แวร์

1) ระบบปฏิบัติการ Microsoft Windows Server 2003 ที่ติดตั้งโปรแกรม Routing and Remote Access และโปรแกรม Internet Information Services (IIS)

2) โปรแกรม Microsoft .NET Framework version 1.1

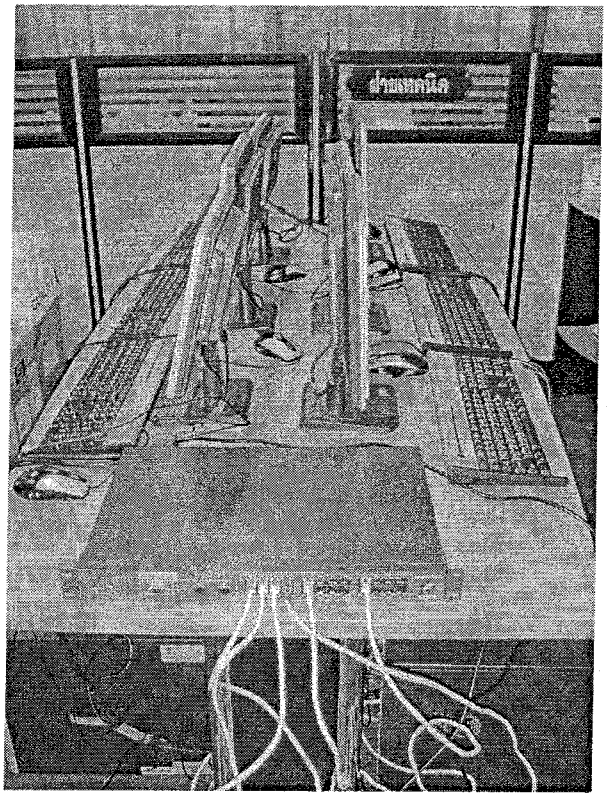
โดยที่การทดลองทั้งหมดได้นำอุปกรณ์ที่ดังกล่าว มากำหนดเป็นสภาพแวดล้อมในการทดลอง ตามแบบจำลองสภาพแวดล้อมของการจัดช่องทางการสื่อสารแบบสมดุลดังภาพที่ 3.1 โดยมีการติดตั้งและกำหนดค่าของอุปกรณ์ต่างๆ ดังภาพที่ 4.3



ภาพที่ 4.1 สภาพแวดล้อมการทดลองการจัดช่องทางการสื่อสารแบบสมดุลบนเครือข่ายระบบปิด

จากภาพที่ 4.1 เป็นการจำลองสภาพแวดล้อมของเครือข่ายแลน (LAN: Local Area Network) ให้คล้ายกับเครือข่ายอินเตอร์เน็ต โดยการใช้เครื่องคอมพิวเตอร์ส่วนบุคคลจำนวน 2 เครื่องแทนอุปกรณ์เราท์เตอร์ (router) และเชื่อมต่อเครื่องคอมพิวเตอร์ทั้งสองเครื่องไปยังเครื่อง

คอมพิวเตอร์ส่วนบุคคลอีกเครื่องหนึ่ง ซึ่งทำหน้าที่เป็นเครื่องแม่ข่าย (server) ในการบริการข้อมูลให้แก่เครื่องลูกข่าย (client) โดยผ่านแผงวงจรเครือข่าย (network adapter) ซึ่งเครื่องคอมพิวเตอร์ทั้ง 3 เครื่องที่กล่าวมา จะต้องติดตั้งโปรแกรม Routing and Remote Access ในระบบปฏิบัติการ Microsoft Windows Server 2003 โดยปิดฟังก์ชัน NAT: Network Address Translation และกำหนดเส้นทางในการรับ-ส่งข้อมูลระหว่างสวิทช์กับเครื่องแม่ข่ายเป็นแบบ static route เพื่อบังคับให้การจราจรในขาเข้า (incoming traffic) ใช้เพียงช่องทางเดียว ทั้งนี้เพื่อความชัดเจนในการเปรียบเทียบประสิทธิภาพของอัลกอริทึมต่างๆ ในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสาร โดยกำหนดให้ใช้เส้นทางที่ 1 คือ 192.168.2.2 ในการรับข้อมูลขาเข้า (incoming traffic) ส่วนข้อมูลขาออก (outgoing traffic) นั้นสามารถใช้ได้ในทั้งสองเส้นทาง คือ 192.168.2.2 และ 192.168.3.2



ภาพที่ 4.2 อุปกรณ์ที่ใช้ในการทดลองการจัดช่องทางการสื่อสารแบบสมดุลบนเครือข่ายระบบปิด

4.1.2 การทดลองบนเครือข่ายอินเตอร์เน็ตจริง

การทดลองในส่วนที่ 2 เป็นการทดลองเพื่อทดสอบโปรแกรมจัดช่องทางการสื่อสารแบบสมดุลในการใช้งานกับเครือข่ายอินเตอร์เน็ต เพื่อทดสอบความสามารถในด้านการหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ขัดข้อง ประกอบด้วยการทดลองคือ

การทดลองที่ 4 ทดสอบการจัดช่องทางการสื่อสารเมื่อช่องทางการสื่อสารเกิดขัดข้อง

อุปกรณ์ที่ใช้ในการทดลองมีดังต่อไปนี้

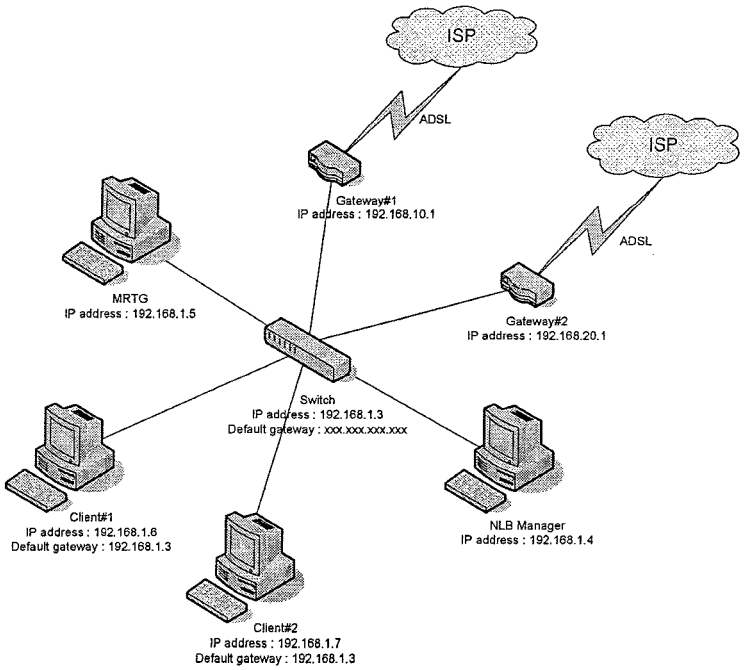
4.1.2.1 ส่วนประกอบของฮาร์ดแวร์

1) เครื่องคอมพิวเตอร์ส่วนบุคคล จำนวน 4 เครื่อง
2) สวิทช์ เลเยอร์ 3 (switch layer 3) จำนวน 1 ตัว
3) ADSL router จำนวน 2 ตัว
4) วงจรแบบ ADSL จำนวน 2 วงจร

4.1.2.2 ส่วนประกอบของซอฟต์แวร์

1) ระบบปฏิบัติการ Microsoft Windows Server 2003
2) โปรแกรม Microsoft .NET Framework version 1.1
3) โปรแกรม MRTG 2.16.1
4) โปรแกรม μTorrent 1.7.7

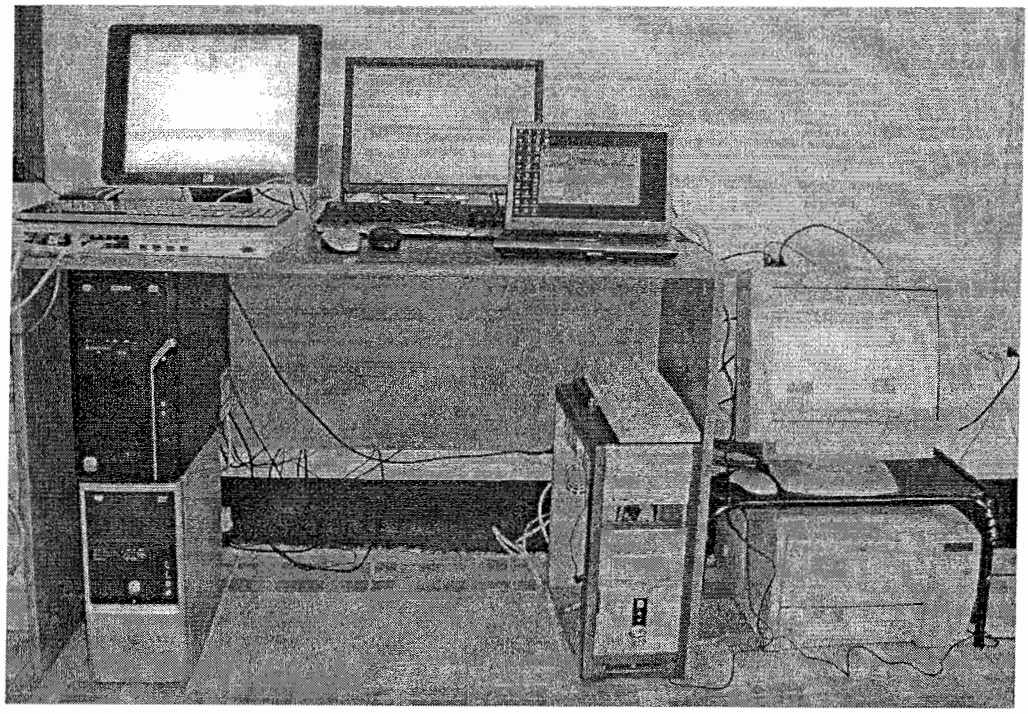
โดยในการทดลองได้นำอุปกรณ์ดังกล่าว มากำหนดเป็นสภาพแวดล้อมในการทดลองตามแบบจำลองสภาพแวดล้อมของการจัดช่องทางการสื่อสารแบบสมดุล ดังภาพที่ 3.1 โดยมีการติดตั้งและกำหนดค่าของอุปกรณ์ต่างๆ ดังภาพที่ 4.1



ภาพที่ 4.3 สภาพแวดล้อมในการทดลองการจัดช่องทางการสื่อสารแบบสมดุลบนเครือข่ายอินเตอร์เน็ต

จากภาพที่ 4.3 ได้กำหนดให้มีการเชื่อมต่อระหว่างเครือข่ายภายในกับเครือข่ายภายนอก โดยใช้วงจรแบบ ADSL จำนวน 2 วงจร คือ ช่องทางการสื่อสารที่ 1 (Gateway#1) และช่องทางการสื่อสารที่ 2 (Gateway#2) โดยให้อุปกรณ์เราเตอร์ (router) ของแต่ละช่องทางการสื่อสารต่อเข้ากับสวิทช์ (switch) ของเครือข่ายแลน (LAN: Local Area Network) ในส่วนของเครื่องคอมพิวเตอร์ประกอบด้วยเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรมรักษาสมดุลของปริมาณจราจร (NLB manager) จำนวน 1 เครื่อง เพื่อทำหน้าที่ควบคุมการกระจายปริมาณจราจรให้กับช่องทางการสื่อสาร และเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรม MRTG (Multi Router Traffic Grapher) เพื่อทำหน้าที่บันทึกปริมาณการจราจรในแต่ละช่องทางการสื่อสาร โดยที่อุปกรณ์ที่กล่าวมาทั้งหมดนั้นเป็นอุปกรณ์ที่สนับสนุนโปรโตคอล SNMP

ในส่วนของเครื่องลูกข่ายได้ใช้โปรแกรมมิวทอเร้นท์ (μTorrent) ในการทดสอบการรับส่งข้อมูลจากเครือข่ายอินเตอร์เน็ต เพื่อจะได้นำบันทึกปริมาณการจราจรในช่องทางการสื่อสารจากแฟ้มบันทึกปริมาณการจราจรของโปรแกรม MRTG มาแสดงเป็นกราฟ เพื่อแสดงให้เห็นถึงประสิทธิภาพของโปรแกรมการจัดช่องทางการสื่อสารแบบสมดุลในการหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ไม่สามารถใช้งานได้



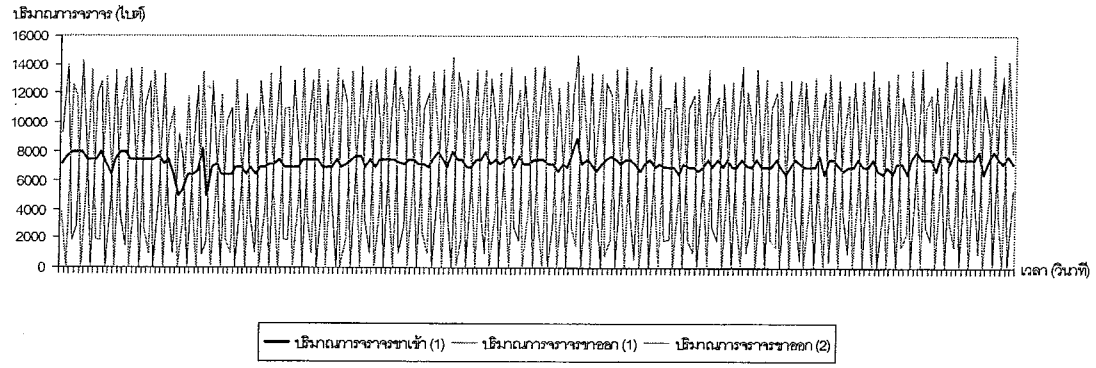
ภาพที่ 4.4 อุปกรณ์ที่ใช้ในการทดลองการจัดช่องทางการสื่อสารแบบสมดุลบนเครือข่ายอินเตอร์เน็ต

4.2 ผลการทดลองเพื่อเปรียบเทียบประสิทธิภาพของอัลกอริทึม

การทดลองเพื่อเปรียบเทียบประสิทธิภาพของอัลกอริทึมในการรักษาสมดุลของปริมาณการจราจรในงานวิจัยนี้ ได้นำอัลกอริทึม Round-Robin, Weighted Round-Robin และ Dynamic Weighted Round-Robin มาทำการเปรียบประสิทธิภาพบนเครือข่ายระบบปิด เพื่อเป็นการกำจัดปัจจัยที่อาจมีผลกระทบต่อการทำงานของระบบ และทำให้สามารถเปรียบเทียบประสิทธิภาพการทำงานได้อย่างชัดเจน จึงได้แบ่งการทดลองออกตามการทำงานของอัลกอริทึมดังที่กล่าวมาแล้วในหัวข้อ 4.1 ได้ผลการทดลองดังต่อไปนี้

4.2.1 ผลการทดลองที่ 1 การจัดช่องทางการสื่อสารโดยใช้อัลกอริทึม Round-Robin

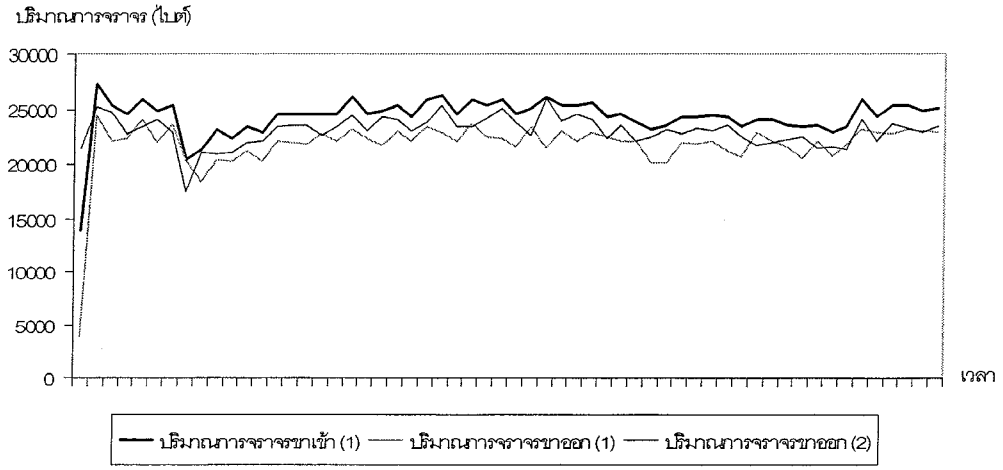
การทดลองนี้ได้ใช้อัลกอริทึม Round-Robin ในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสาร โดยการบันทึกปริมาณการจราจรในช่องทางการสื่อสารจากอุปกรณ์สวิทช์ (switch) ซึ่งทำหน้าที่เป็นบาลานเซอร์ (balancer) ใน 2 ลักษณะด้วยกันคือ บันทึกปริมาณการจราจรตามช่วงเวลาทุกๆ 3 วินาที และบันทึกปริมาณการจราจรตามรอบเวลาการทำงานของโปรแกรมจัดการเครือข่ายแบบสมดุลทุกๆ 7 วินาที และได้นำมาแสดงในลักษณะของกราฟได้ดังในภาพที่ 4.5 และภาพที่ 4.6 ดังนี้



ภาพที่ 4.5 กราฟแสดงปริมาณการจราจรตามช่วงเวลาทุกๆ 3 วินาทีโดยอัลกอริทึม Round-Robin

จากกราฟในภาพที่ 4.5 แสดงให้เห็นว่าปริมาณการจราจรขาออกในช่องทางการสื่อสารที่ 1 และช่องทางการสื่อสารที่ 2 จะขึ้นอยู่กับปริมาณการจราจรขาออกเท่านั้น ไม่ได้มีความสัมพันธ์กับปริมาณการจราจรขาเข้าในช่องทางการสื่อสารที่ 1 จึงแสดงให้เห็นว่าอัลกอริทึม Round-Robin มีการทำงานโดยการสลับการใช้งานช่องทางการสื่อสารตามช่วงเวลาที่กำหนด โดยไม่คำนึงถึงปริมาณการจราจรขาเข้า (incoming traffic) ในช่องทางการสื่อสาร จึงทำให้กราฟแสดงปริมาณ

ข้อมูลขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 1 และช่องทางการสื่อสารที่ 2 มีปริมาณใกล้เคียงกันอยู่ในช่วง 0 ถึง 14,000 ไบต์

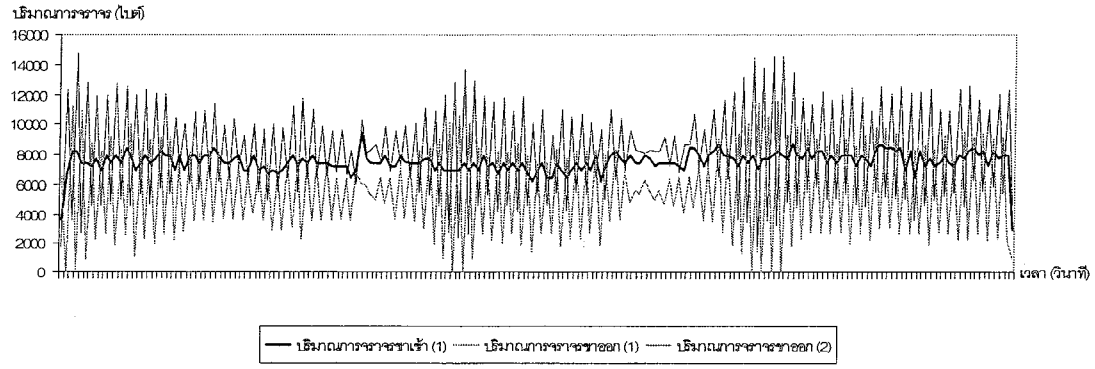


ภาพที่ 4.6 กราฟแสดงปริมาณการจราจรตามรอบเวลาการทำงานของโปรแกรมในทุกๆ 7 วินาที โดยอัลกอริทึม Round-Robin

จากกราฟในภาพที่ 4.6 ซึ่งแสดงปริมาณการจราจรตามรอบการทำงานของโปรแกรมโดยอัลกอริทึม Round-Robin พบว่ากราฟแต่ละเส้นมีจุดที่ตัดกัน จึงแสดงให้เห็นว่าปริมาณการจราจรในขาเข้า (incoming traffic) และปริมาณการจราจรขาออก (outgoing traffic) ไม่ได้มีความสัมพันธ์ต่อกันในเชิงปริมาณ

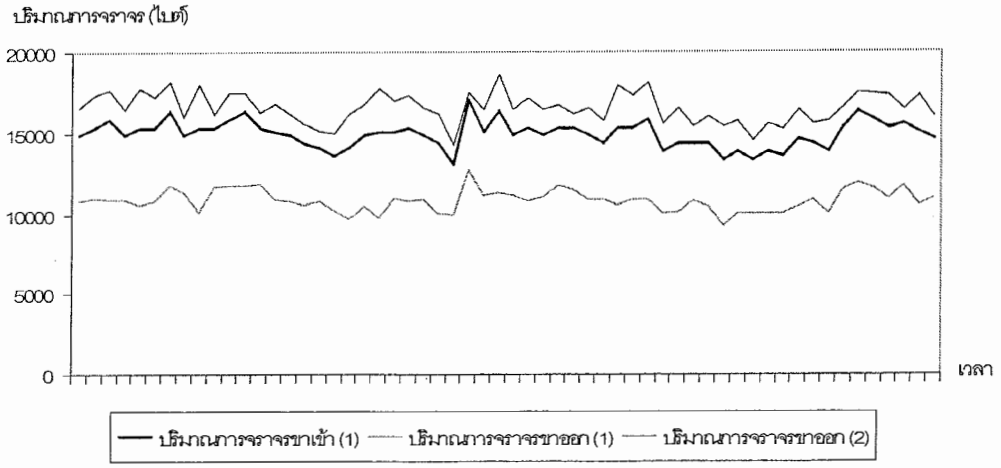
4.2.2 ผลการทดลองที่ 2 การจัดช่องทางการสื่อสารโดยใช้อัลกอริทึม Weighted Round-Robin

การทดลองนี้ ได้ใช้อัลกอริทึม Weighted Round-Robin เพื่อรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสาร โดยการบันทึกปริมาณการจราจรในช่องทางการสื่อสารจากอุปกรณ์สวิทช์ซึ่งทำหน้าที่เป็นบาลานเซอร์ (balancer) ใน 2 ลักษณะเช่นเดียวกันกับในการทดลองที่ 1 และนำมาแสดงเป็นกราฟดังในภาพที่ 4.7 และภาพที่ 4.8 ดังนี้



ภาพที่ 4.7 กราฟแสดงปริมาณการจราจรตามช่วงเวลาทุกๆ 3 วินาทีโดยอัลกอริทึม Weighted Round-Robin

จากกราฟในภาพที่ 4.7 แสดงให้เห็นว่าปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 1 และ 2 จะขึ้นอยู่กับปริมาณการจราจรขาออก (outgoing traffic) และมีความสัมพันธ์กับปริมาณการจราจรขาเข้า (incoming traffic) ของช่องทางการสื่อสารที่ 1 ซึ่งมีปริมาณค่อนข้างคงที่ระหว่าง 6,000 ถึง 8,000 ไบต์ ส่วนกราฟแสดงปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 1 มีปริมาณอยู่ระหว่าง 0 ถึง 12,000 ไบต์ ซึ่งมีค่าน้อยกว่าปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 2 ซึ่งมีปริมาณอยู่ระหว่าง 2,000 ถึง 14,000 ไบต์ จึงแสดงให้เห็นว่าอัลกอริทึม Weighted Round-Robin มีการจัดช่องทางการสื่อสารโดยคำนึงถึงปริมาณการจราจรในแต่ละช่องทางการสื่อสาร กล่าวคือ เมื่อการจราจรขาเข้า (incoming traffic) ของช่องทางการสื่อสารที่ 1 มีปริมาณมาก ปริมาณการจราจรขาออก (outgoing traffic) ของช่องทางการสื่อสารที่ 1 จะถูกลดลง

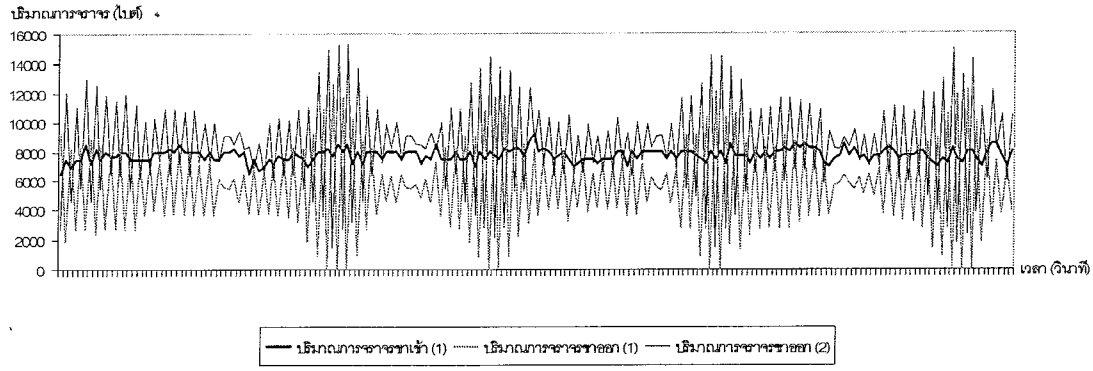


ภาพที่ 4.8 กราฟแสดงปริมาณการจราจรตามรอบเวลาการทำงานของโปรแกรมในทุกๆ 7 วินาที โดยอัลกอริทึม Weighted Round-Robin

จากกราฟในภาพที่ 4.8 แสดงให้เห็นว่าปริมาณการจราจรขาออกในช่องทางการสื่อสารที่ 1 และช่องทางการสื่อสารที่ 2 มีปริมาณที่ผกผันต่อซึ่งกันและกัน อันเนื่องมาจากการทำงานของอัลกอริทึม Weighted Round-Robin ซึ่งมีการกำหนดค่าน้ำหนักในการใช้งานช่องทางการสื่อสารตามปริมาณการจราจรในรอบการทำงานของโปรแกรม ทำให้กราฟแสดงปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 1 มีปริมาณอยู่ระหว่าง 10,000 ถึง 15,000 ไบต์ น้อยกว่าปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 2 ซึ่งมีปริมาณอยู่ระหว่าง 15,000 ถึง 18,000 ไบต์ เนื่องจากในช่องทางการสื่อสารที่ 1 มีปริมาณการจราจรขาเข้า (incoming traffic) ซึ่งมีปริมาณเฉลี่ย 15,000 ไบต์ รวมอยู่ด้วย

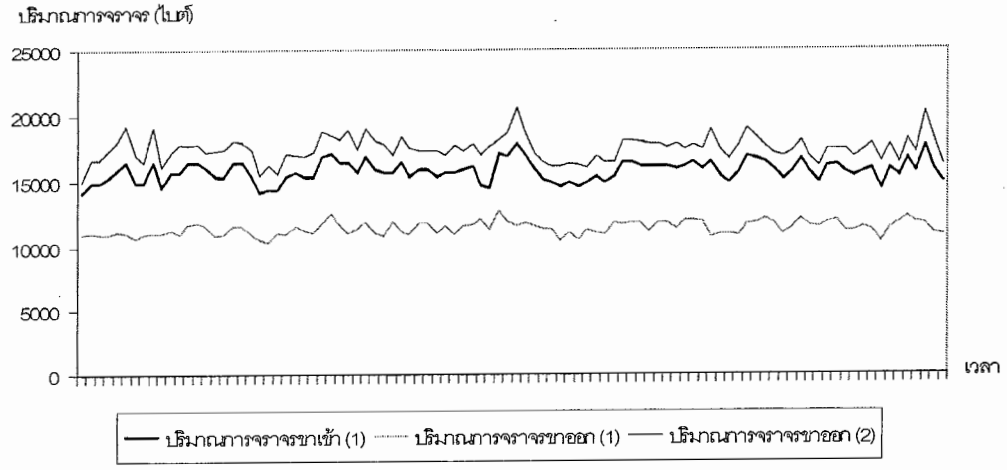
4.2.3 ผลการทดลองที่ 3 การจัดช่องทางการสื่อสารโดยใช้อัลกอริทึม Dynamic Weighted Round-Robin

การทดลองนี้ใช้อัลกอริทึม Dynamic Weighted Round-Robin เพื่อรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสาร โดยการบันทึกปริมาณการจราจรในช่องทางการสื่อสารจากอุปกรณ์สวิทช์ (switch) ซึ่งทำหน้าที่เป็นบาลานเซอร์ (balancer) ในลักษณะเช่นเดียวกันกับในการทดลองที่ 1 และการทดลองที่ 2 แล้วจึงนำมาแสดงเป็นกราฟได้ดังในภาพที่ 4.9 และภาพที่ 4.10 ดังนี้



ภาพที่ 4.9 กราฟแสดงปริมาณการจราจรตามช่วงเวลาทุกๆ 3 วินาทีโดยอัลกอริทึม Dynamic Weighted Round-Robin

จากกราฟในภาพที่ 4.9 แสดงให้เห็นว่าปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 1 และ 2 จะขึ้นอยู่กับปริมาณการจราจรขาออก (outgoing traffic) และมีความสัมพันธ์กับปริมาณการจราจรขาเข้า (incoming traffic) ของช่องทางการสื่อสารที่ 1 ซึ่งมีปริมาณค่อนข้างคงที่ระหว่าง 0 ถึง 8,000 ไบต์ ส่วนกราฟแสดงปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 1 มีปริมาณอยู่ระหว่าง 0 ถึง 12,000 ไบต์ ซึ่งมีค่าน้อยกว่าปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 2 ซึ่งมีปริมาณอยู่ระหว่าง 2,000 ถึง 14,000 ไบต์ จึงแสดงให้เห็นว่าอัลกอริทึม Dynamic Weighted Round-Robin มีการจัดช่องทางการสื่อสารโดยคำนึงถึงปริมาณการจราจรในแต่ละช่องทางการสื่อสาร เช่นเดียวกันกับอัลกอริทึม Weighted Round-Robin กล่าวคือ เมื่อการจราจรขาเข้า (incoming traffic) ของช่องทางการสื่อสารที่ 1 มีปริมาณมาก ปริมาณการจราจรขาออก (outgoing traffic) ของช่องทางการสื่อสารที่ 1 จะถูกลดลง



ภาพที่ 4.10 กราฟแสดงปริมาณการจราจรตามรอบเวลาการทำงานของโปรแกรมในทุกๆ 7 วินาที โดยอัลกอริทึม Dynamic Weighted Round-Robin

จากกราฟในภาพที่ 4.10 แสดงให้เห็นว่าปริมาณการจราจรขาออกในช่องทางการสื่อสารที่ 1 และช่องทางการสื่อสารที่ 2 มีปริมาณที่ผกผันต่อซึ่งกันและกัน อันเนื่องมาจากการทำงานของอัลกอริทึม Dynamic Weighted Round-Robin ซึ่งมีการกำหนดค่าน้ำหนักในการใช้งานช่องทางการสื่อสารตามปริมาณการจราจรในรอบการทำงานของโปรแกรม เช่นเดียวกันกับอัลกอริทึม Weighted Round-Robin ทำให้กราฟแสดงปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 1 มีปริมาณอยู่ระหว่าง 10,000 ถึง 12,000 ไบต์ น้อยกว่าปริมาณการจราจรขาออก (outgoing traffic) ในช่องทางการสื่อสารที่ 2 ซึ่งมีปริมาณอยู่ระหว่าง 15,000 ถึง 20,000 ไบต์ เนื่องจากในช่องทางการสื่อสารที่ 1 มีปริมาณการจราจรขาเข้า (incoming traffic) ซึ่งมีปริมาณเฉลี่ย 15,000 ไบต์ รวมอยู่ด้วย เนื่องจากอัลกอริทึม Dynamic Weighted Round-Robin ได้มีการคำนวณเวลาของรอบการทำงานที่เหมาะสมใหม่ทุกครั้งในแต่ละรอบการทำงาน ทำให้ในเวลาทำงานของโปรแกรมที่เท่ากัน อัลกอริทึม Dynamic Weighted Round-Robin อาจมีจำนวนรอบการทำงานที่แตกต่างจากอัลกอริทึม Weighted Round-Robin กล่าวคือ ถ้าปริมาณการจราจรในแต่ละช่องทางการไม่สมดุลกัน อัลกอริทึม Dynamic Weighted Round-Robin จะทำการลดเวลาของรอบการทำงานลงให้มีการทำงานถี่มากขึ้น แต่ถ้าปริมาณการจราจรสมดุลกันก็จะเพิ่มเวลาของรอบการทำงานขึ้นเพื่อให้ระบบทำงานน้อยลง ซึ่งจากการทดลองอัลกอริทึม Dynamic Weighted Round-Robin จะมีรอบการทำงานที่มากกว่าอัลกอริทึม Weighted Round-Robin (94 และ 57 รอบตามลำดับ) สาเหตุเนื่องจากสภาพแวดล้อมในการทดลองได้กำหนดให้ช่องทางที่ 1 เท่านั้นที่ใช้ในการรับข้อมูลขาเข้า ทำให้ปริมาณการจราจรอยู่ในภาวะที่ไม่สมดุลกันระบบจึงทำการลดเวลาใน

แต่ละรอบการทำงานลง เป็นเหตุให้อัลกอริทึม Dynamic Weighted Round-Robin มีรอบการทำงานที่มากกว่า Weighted Round-Robin

4.2.4 ผลการเปรียบเทียบประสิทธิภาพของอัลกอริทึม

ในการศึกษาเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมแบบ Round-Robin, Weighted Round-Robin และ Dynamic Weighted Round-Robin เพื่อการรักษาความสมดุลของปริมาณการจราจรในช่องทางการสื่อสารตามสภาพแวดล้อมที่ได้กำหนดขึ้น ดังนั้นผู้วิจัยจึงได้ทำการวิเคราะห์ข้อมูลโดยใช้หลักการทางสถิติ เพื่อเปรียบเทียบประสิทธิภาพการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสาร ดังตารางที่ 4.1

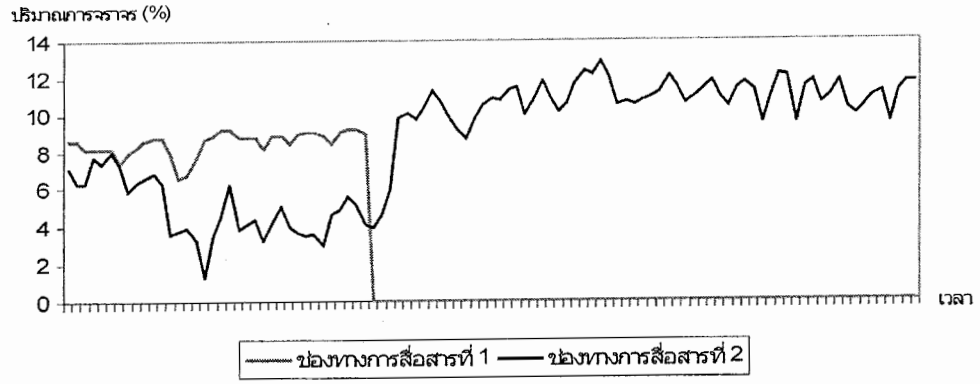
ตารางที่ 4.1 เปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึม

อัลกอริทึม	Mean	SD
Round-Robin	6873.62	10464.80
Weighted Round-Robin	4628.37	6177.97
Dynamic Weighted Round-Robin	4742.02	6064.00

จากตารางที่ 4.1 พบว่าจาก อัลกอริทึม Dynamic Weighted Round-Robin มีประสิทธิภาพในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสารมากที่สุด พิจารณาจากค่าส่วนเบี่ยงเบนมาตรฐาน (SD: Std. Deviation) ต่ำที่สุดเท่ากับ 6064.00 ค่าเฉลี่ย (Mean) เท่ากับ 4742.02 ซึ่งหมายถึงปริมาณการจราจรในช่องทางการสื่อสารมีประสิทธิภาพในด้านการทำงานของอัลกอริทึมที่มีแนวโน้มการทำงานดีกว่า ส่วนอัลกอริทึมที่มีประสิทธิภาพรองลงมาก็คือ Weighted Round-Robin ค่าส่วนเบี่ยงเบนมาตรฐาน (SD: Std. Deviation) เท่ากับ 6177.97 ค่าเฉลี่ย (Mean) เท่ากับ 4628.37 และอัลกอริทึมที่มีประสิทธิภาพในการรักษาสมดุลของปริมาณการจราจรในช่องทางการสื่อสารต่ำที่สุดในการทดลองนี้คืออัลกอริทึม Round-Robin ซึ่งมีค่าส่วนเบี่ยงเบนมาตรฐาน (SD: Std. Deviation) เท่ากับ 10464.80 ค่าเฉลี่ย (Mean) เท่ากับ 6873.62

4.3 ผลการทดลองการจัดช่องทางการสื่อสารเมื่อช่องทางการสื่อสารเกิดขัดข้อง

ผลการทดลองที่ 4 ทดสอบการจัดช่องทางการสื่อสารเมื่อช่องทางการสื่อสารเกิดขัดข้อง จากการทดลองนี้ได้ใช้อัลกอริทึม Round-Robin เพื่อทดสอบความสามารถในการหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ขัดข้องโดยในช่วงแรกของการทดลองได้กำหนดให้ใช้ช่องทางการสื่อสารหลักจำนวน 2 ช่องทาง เมื่อเวลาผ่านไประยะหนึ่งจึงได้ปิดการใช้งานช่องทางการสื่อสารที่ 1 ซึ่งได้ผลการทดลองดังแสดงในภาพที่ 4.11



ภาพที่ 4.11 กราฟแสดงปริมาณการจราจรเฉลี่ยในทุกๆ 5 นาทีของช่องทางการสื่อสารที่ 1 และ 2

จากกราฟแสดงปริมาณการจราจรเฉลี่ยในภาพที่ 4.11 พบว่า เมื่อปิดการทำงานของช่องทางการสื่อสารที่ 1 จึงทำให้มีปริมาณการจราจรเป็น 0 ส่วนในช่องทางการสื่อสารที่ 2 พบว่ามีปริมาณการจราจรเพิ่มขึ้น เนื่องจากปริมาณการจราจรทั้งหมดที่เกิดขึ้นในเครือข่ายถูกกำหนดให้กับช่องทางการสื่อสารที่ 2 จึงกล่าวได้ว่า โปรแกรมการจัดช่องทางการสื่อสารแบบสมดุลมีความสามารถในการหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ขัดข้องไปใช้งานช่องทางการสื่อสารอื่นที่มีอยู่แทน

บทที่ 5
สรุปผลการศึกษา

5.1 สรุปผลการศึกษา

งานวิจัยนี้ต้องการศึกษาวิธีการจัดช่องทางการสื่อสารแบบสมดุล ซึ่งอาศัยการทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์ เพื่อลดข้อจำกัดทางด้านต่างๆ ที่เกิดขึ้นจากการใช้ฮาร์ดแวร์หรือซอฟต์แวร์เพียงอย่างเดียว โดยการทำงานของสภาพแวดล้อมและโปรแกรมที่ได้พัฒนาขึ้นนี้ ได้มีความสามารถในการตรวจสอบปริมาณการจราจร เพื่อกำหนดปริมาณข้อมูลขาออกที่เหมาะสมให้กับช่องทางการสื่อสาร โดยใช้อัลกอริทึม Round-Robin, Weighted Round-Robin และ Dynamic Weighted Round-Robin มาประยุกต์ใช้ในการจัดช่องทางการสื่อสารแบบสมดุล ตลอดจนมีความสามารถในการหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ขัดข้อง

สำหรับด้านการเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึม จากการศึกษาพบว่าอัลกอริทึมที่ให้ประสิทธิภาพในการรักษาสมดุลของปริมาณการจราจรดีที่สุด คือ อัลกอริทึม Dynamic Weighted Round-Robin ส่วนอัลกอริทึม Weighted Round-Robin มีประสิทธิภาพรองลงมา และอัลกอริทึมที่มีประสิทธิภาพต่ำที่สุดคืออัลกอริทึม Round-Robin

นอกจากนี้ยังพบว่า โปรแกรมจัดช่องทางการสื่อสารแบบสมดุลมีความสามารถในการตรวจสอบสถานะของช่องทางการสื่อสาร ทำให้สามารถหลีกเลี่ยงการใช้งานช่องทางการสื่อสารที่ขัดข้องไปใช้งานช่องทางการสื่อสารอื่นแทน ซึ่งจะเห็นได้จากกราฟในภาพที่ 4.11 ที่แสดงปริมาณการจราจรในช่องทางที่ 1 มีค่าเป็นศูนย์ และปริมาณการจราจรในช่องทางการสื่อสารที่ 2 มีค่าเพิ่มขึ้นเนื่องจากการจราจรทั้งหมดที่เกิดขึ้นถูกกำหนดมาให้กับช่องทางการสื่อสารที่ 2 แทน

5.2 ข้อเสนอแนะ

5.2.1 เนื่องจากโปรแกรมที่ได้พัฒนาขึ้นได้นำอัลกอริทึมบางส่วนมาใช้ในการทดลอง แต่จากการศึกษาพบว่ายังมีอัลกอริทึมอื่นๆ ที่สามารถนำมาใช้ในการรักษาสมดุลของปริมาณการจราจรในเครือข่ายได้ ในอนาคตจึงอาจมีการนำอัลกอริทึมอื่นๆ หรือการพัฒนาอัลกอริทึมขึ้นใหม่เพื่อนำมาใช้กับโปรแกรมการจัดช่องทางการสื่อสารแบบสมดุล เช่น การนำอัลกอริทึมทางด้านปัญญาประดิษฐ์ที่มีความสามารถในการคาดคะเนปริมาณการจราจรในช่องทางการสื่อสารล่วงหน้ามาประยุกต์ใช้กับการจัดช่องทางการสื่อสารแบบสมดุล เป็นต้น

5.2.2 เนื่องจากโปรแกรมได้ถูกพัฒนาขึ้นเพื่อใช้กับวงจรการสื่อสารแบบ ADSL ซึ่งเป็นวงจรสื่อสารที่ใช้สำหรับบ้านพักอาศัย แต่ถ้ามีการนำไปใช้ในภาคธุรกิจซึ่งอาจมีการใช้วงจรสื่อสารในลักษณะอื่นๆ เช่น ISDN เป็นต้น จึงอาจพัฒนาความสามารถของโปรแกรมโดยการเพิ่มฟังก์ชันในการตรวจสอบลักษณะของช่องทางการสื่อสารให้มีความหลากหลายมากขึ้น

5.2.3 เนื่องจากโปรแกรมการจัดช่องทางการสื่อสารแบบสมดุลที่พัฒนาขึ้น ได้ใช้ SNMP ในลักษณะของการโพลลิ่ง (polling) ซึ่งโปรแกรมต้องคอยตรวจสอบสถานะของช่องทางการสื่อสาร แต่จากการค้นคว้ายังพบว่า SNMP ยังมีการทำงานโดยใช้คำสั่ง Trap message ทำให้อุปกรณ์เครือข่าย สามารถรายงานสถานะของช่องทางการสื่อสารโดยที่โปรแกรมไม่ต้องทำการตรวจสอบได้ ในอนาคตจึงอาจพัฒนาโปรแกรมให้สามารถใช้คำสั่ง Trap message เพื่อช่วยลดขั้นตอนในการติดต่อสื่อสารกับอุปกรณ์เครือข่ายได้

5.2.4 เนื่องจากโปรแกรมที่ได้พัฒนาขึ้นยังไม่สามารถตรวจสอบ object Identifier ของ default gateway ที่เป็นอุปกรณ์เครือข่ายที่ทำหน้าที่เป็น balancer ได้โดยอัตโนมัติ ซึ่งค่าดังกล่าวมีความสำคัญในการกำหนดการจราจรให้กับช่องทางการสื่อสาร ดังนั้นจึงอาจพัฒนาโปรแกรมให้มีความสามารถในการตรวจสอบค่าดังกล่าวโดยที่ผู้ใช้งานโปรแกรมไม่ต้องกำหนดเอง

5.3 ปัญหาและอุปสรรค

เนื่องจากบางการทดลองได้ใช้เครือข่ายอินเตอร์เน็ตซึ่งเป็นเครือข่ายสาธารณะ ทำให้มีปัจจัยที่อาจส่งผลกระทบต่างๆ มากมาย อาทิเช่น ปริมาณของผู้ใช้งานในช่วงเวลาที่แตกต่างกัน ปริมาณข้อมูลในแต่ละช่วงเวลา ความเร็วในการรับ-ส่งข้อมูลในช่วงเวลาต่างกัน เป็นต้น ซึ่งล้วนอาจเป็นสาเหตุที่ทำให้ผลการทดลองเกิดความคลาดเคลื่อนได้ทั้งสิ้น การทดลองบางส่วนจึงได้กระทำในระบบปิดโดยใช้เครือข่ายแลน (LAN: Local Area Network) เพื่อลดปัจจัยที่อาจส่งผลกระทบต่อการทำการทดลอง

เอกสารอ้างอิง

เอกสารอ้างอิง

[1] Microsoft Windows 2000 Server. (2000). Network Load Balancing Technical Overview. http://www.microsoft.com/technet/prodtechnol/windows2000serv. May 28, 2006.

[2] Cardellini V, Colajanni M, Yu PS. (1999). "Dynamic load balancing on Web-Server systems", IEEE Internet Computing. http://ieeexplore.ieee.org/servlet/opac?punumber=4236, April 30, 2006.

[3] James D. Murray. (1998). Windows NT SNMP. Sebastopol, CA, USA: O'Reilly & Associates Inc.

[4] Cisco Systems Inc. (2005). Simple Network Management Protocol (SNMP). http://www.cisco.com/warp/public/535/3.html. March 15, 2006.

[5] Perl for System Administration (2001). The Twenty-Minute SNMP Tutorial, O'Reilly & Associates. http://www.unix.org.ua/ orelly /perl/sysadmin/appe_01.htm. May 31, 2006.

[6] Craig Hunt. (1998). TCP/IP Network Administration, Sebastopol, CA, USA: O'Reilly & Associates, Inc.

[7] Bryhni H, Klovning E, Kure O. (2000). "A comparison of load balancing techniques for scalable web servers", IEEE Network. http://ieeexplore.ieee.org/iel1/49/3226/00105173.pdf. March 15, 2006.

[8] Nagle J. (1987). "Packet switches with infinite storage", IEEE Transactions on Communications. http://www.elsevier.com/locate/dsw. January 9, 2006.

[9] Katevenis M, Sidiropoulos C, Courcoubetis C. (1991). "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip", IEEE Journal on Selected Areas in Communications. http://ieeexplore.ieee.org/iel1/49/3226/00105173.pdf. March 15, 2006

[10] Der-Chiang Li, Chihsen Wu and Fengming M. Chang. (2004). Determination of the parameters in the dynamic weighted Round-Robin method for network load balancing. http://www/elsevier.com/ locate/dsw. December 4, 2005.

[11] BIG-IP Link Controller Datasheet. (2008). F5 Networks, Inc. http://www.f5.com/pdf/products/big-ip-link-controller-ds.pdf. May 10, 2008.

เอกสารอ้างอิง (ต่อ)

[12] Advanced Router. (2005). SysMaster. http://www.sysmaster.com. November 10, 2005.

[13] Load Balance with Fail Over. (2007). PFSENSE. http://pfsense.icehosting.com/tutorials/outgoing_loadbalancing/outgoing_loadbalancing.pdf. Febuary 22, 2008

[14] เอกสิทธิ์ วิริยจารี. เรียนรู้ระบบเน็ตเวิร์กจากอุปกรณ์ของ Cisco ภาคปฏิบัติ. กรุงเทพมหานคร : สำนักพิมพ์ซีเอ็ดยูเคชั่น.

[15] Tobias Oetiker. (2006). Multi Router Traffic Grapher. http://www.mrtg.net. September.17, 2005

[16] BitTorrent, Inc. (2008). A beginner's guide to BitTorrent. http://www.utorrent.com/beginners-guide.php. June. 27, 2007.

[17] P. Levillain. (2002). "Switch-Based Server Load Balancing for Enterprises". Alcatel Telecommunications Review. 4th Quarter 2002. http://www1.alcatel-lucent.com/doctypes/articlepaperlibrary/pdf/ATR2002Q4/T0212-Load_Balancing-EN.pdf. March 10, 2008.

[18] Mohammed H. Sqalli and Shaik Sirajuddin (2005). An Adaptive Load-balancing Approach to XML-based Network Management using JPVM. https://eprints.kfupm.edu.sa/249/1/Sqalli-Sirajuddin-Adaptive-LB-XNM-ICON-05-CR.pdf. June 10, 2007.

[19] Johen Katz and Frank Fock. (2006). SNMP++ .NET API. http://maom_onet.republika.pl/snmp/snmp_ppnet. June 23, 2005

ประวัติผู้วิจัย

ชื่อ นายปกรณ์ กัลปดี

ประวัติการศึกษา พ.ศ. 2537 วิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยหอการค้าไทย

ประวัติการทำงาน พ.ศ. 2537 – 2538
ตำแหน่งโปรแกรมเมอร์
ศูนย์บริการโลหิตแห่งชาติ สภากาชาดไทย
พ.ศ. 2539 – 2540
ตำแหน่ง MIS OFFICER
บริษัท เวฟ ดีเวลลอปเม้นท์ จำกัด
พ.ศ. 2541 – 2542
ตำแหน่ง MIS SUPERVISOR
บริษัท แมนฮัทตัน โฟโต้ ออง โวค จำกัด
พ.ศ. 2542 – ปัจจุบัน
ตำแหน่งอาจารย์ประจำ
มหาวิทยาลัยราชภัฏอุบลราชธานี