การพัฒนาระบบสร้างสมดุลภาระบนระบบปฏิบัติการลินุกซ์ เพื่อรวมช่องสัญญาณการเข้าถึงอินเทอร์เน็ต

IMPLEMENTATION OF LINUX-BASED LOAD BALANCING SYSTEM FOR AGGREGATING INTERNET ACCESS BANDWIDTH

โดย

เจนณรงค์ มุสิกพันธ์

JANENARONG MUSIGAPHAN

อาจารย์ที่ปรึกษา

รศ.ดร. โชติพัชร์ ภรณวลัย

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2550

IMPLEMENTATION OF LINUX-BASED LOAD BALANCING SYSTEM FOR AGGREGATING INTERNET ACCESS BANDWIDTH

JANENARONG MUSIGAPHAN

A SYSTEM DEVELOPMENT PROJECT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/ 2007

หัวข้อ	การพัฒนาระบบสร้างสมดุลภาระบนระบบปฏิบัติการลินุกซ์เพื่อรวมช่องสัญญาณการเข้าถึงอินเทอร์เน็ต
นักศึกษา	เรืออากาศเอก เจนณรงค์ มุสิกพันธ์
รหัสนักศึกษา	48066540
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	รศ.ดร. โชติพัชร์ ภรณวลัย

บทคัดย่อ

การพัฒนาระบบ Load Balancing บนระบบปฏิบัติการ Linux เพื่อรวมช่องสัญญาณการเข้าถึงอินเตอร์เน็ตนี้ เป็นผลมาจากการศึกษาความเป็นไปได้ในการพัฒนาระบบดังกล่าวในวิชาสัมมนา 2 เพื่อสร้างระบบที่สามารถรวมการใช้ช่องสัญญาณการเชื่อมต่ออินเทอร์เน็ตหลายช่องทาง โดยการกระจายกระแสข้อมูลที่เกิดจากภายในองค์กรออกสู่ช่องทางการเชื่อมต่อที่มีอยู่ ซึ่งจะส่งผลให้องค์กรสามารถใช้ขนาดช่องสัญญาณจากแต่ละช่องทางได้พร้อมกัน ดังนั้นขนาดช่องสัญญาณรวมที่องค์กรมีใช้งานจึงมีเพียงพอและพร้อมใช้งานในระดับที่สามารถใช้งานกับองค์กรขนาดเล็กหรือขนาดกลางได้ ทั้งนี้การพัฒนาระบบดังกล่าวมีเป้าหมายเพื่อลดภาระการใช้งบประมาณด้านเทคโนโลยีสารสนเทศขององค์กรในการลงทุนเช่าสายเคเบิลเชื่อมต่ออินเตอร์เน็ตความเร็วสูงหรือการจัดซื้อระบบที่ถูกพัฒนาเป็นผลิตภัณฑ์จำหน่ายโดยบริษัทต่างๆ สำหรับองค์กรที่มีข้อจำกัดด้านงบประมาณ

Title	Implementation of Linux-based Load Balancing System for Aggregating Internet Access Bandwidth
Student	Flt.Lt. Janenarong Musigaphan
Student ID.	48066540
Degree	Master of Science
Programme	Information Science
Academic Year	2007
Advisor	Assoc.Prof. Dr.Chotipat Pornavalai

ABSTRACT

The implementation of Linux-based load balancing system for aggregating Internet access bandwidth is followed upon the same study area done in seminar 2. The resulting information was shown that it was feasible to implement the system based on Linux technology. Consequently, the project is due to initiate to achieve the system prototype. Hopefully, the system could be deployed in small and mid-sized enterprises with small IT budget in order to provide them an access to higher speed and more reliable Internet access in which is sufficiently robust for enterprise applications.

กิตติกรรมประกาศ

ในการพัฒนาโครงงานระบบสร้างสมดุลภาระบนระบบปฏิบัติการลินุกซ์ เพื่อรวมช่องสัญญาณการเข้าถึงอินเทอร์เน็ตนี้ จะไม่สามารถเกิดขึ้นได้หลายถ้าไม่ได้รับความอนุเคราะห์จาก รศ.ดร. โชติพัชร์ ภรณวลัย ที่กรุณารับเป็นอาจารย์ที่ปรึกษาสำหรับโครงงานนี้ ด้วยความเข้าใจและความไว้วางใจที่อาจารย์มีต่อแนวทางที่ข้าพเจ้าดำเนินงานในโครงงานนี้ ตั้งแต่เริ่มต้นการนำเสนอหัวข้อโครงงาน โดยเฉพาะอย่างยิ่งคำชี้แนะในช่วงท้ายของการดำเนินการที่เป็นปัจจัยสำคัญต่อความสำเร็จของการดำเนินงานรวมทั้งการจัดทำรายงานฉบับสมบูรณ์นี้ ทำให้ข้าพเจ้ารู้สึกเคารพและซาบซึ้งในความกรุณาที่อาจารย์มีต่อข้าพเจ้าเป็นอย่างมาก

ขอกราบขอบพระคุณคณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆท่าน ที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า และรู้สึกเป็นเกียรติมากที่ได้มีโอกาสเข้าศึกษาในสถาบันแห่งนี้

ขอขอบคุณ นาย ทวีวัฒน์ สุวรรณกนิษฐ์ เพื่อนร่วมหลักสูตรวิทยาศาสตรมาบัณฑิต สาขาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง รุ่นที่ 19 (IS19.2) เป็นพิเศษที่สรรหาและแนะนำแหล่งข้อมูลเกี่ยวกับเทคโนโลยี Linux ที่ข้าพเจ้าสามารถนำมาใช้ประโยชน์ในการพัฒนาระบบนี้ เป็นอย่างมาก รวมทั้งเพื่อนๆ รวมหลักสูตรทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมอ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้า รวมทั้งคนพิเศษที่คอยให้กำลังใจและสนับสนุนในทุกๆเรื่อง ที่เป็นแรงผลักดันที่สำคัญส่วนหนึ่งที่ทำให้ข้าพเจ้าบากบั่นมาถึงขั้นตอนสุดท้ายในการศึกษาหลักสูตรนี้

ด้วยความหวังว่า ผลที่เกิดขึ้นจากโครงงานนี้อาจจะถูกนำไปใช้ให้เกิดประโยชน์หรือเป็นแหล่งความรู้สำหรับผู้ที่สนใจการพัฒนาระบบลักษณะนี้ต่อไป

สุดท้ายนี้ คุณค่าและประโยชน์อันพึงมาจากโครงงานนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

เจนณรงค์ มุสิกพันธ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

เนื่องด้วยกองทัพอากาศมีขนาดขององค์กรที่ใหญ่และประกอบโครงสร้างภายในหน่วยที่แบ่งเป็นหลายระดับตั้งแต่ระดับบังคับบัญชาจนถึงระดับปฏิบัติการที่ตั้งอยู่กระจายครอบคลุมพื้นที่ทั่วประเทศไทย ทำให้แต่ละหน่วยมีศักยภาพในการเข้าถึงช่องทางการเชื่อมต่ออินเทอร์เน็ตได้แตกต่างกันทั้งนี้ขึ้นอยู่กับที่ตั้งของหน่วยและงบประมาณด้านเทคโนโลยีสารสนเทศที่ได้รับในแต่ละปี

สำหรับหน่วยงานระดับปฏิบัติการที่มีขนาดเล็กที่มีกำลังพลไม่เกิน 60 คนและมีงบประมาณด้านเทคโนโลยีสารสนเทศไม่เกินปีละ 18,000 บาทนั้น แต่มีความจำเป็นต้องใช้ช่องทางการเชื่อมต่ออินเทอร์เน็ตที่มีขนาดช่องสัญญาณ (Bandwidth) ที่สูง เช่น 1 Mbps ขึ้นไปและมีความพร้อมใช้งานสูง อย่างเช่นการเช่าสายเคเบิลและบริการอินเทอร์เน็ตที่เชื่อมต่อโดยตรงกับผู้ให้บริการทางอินเทอร์เน็ตที่เรียกกันว่า Leased line นั้น ต้องใช้งบประมาณมากกว่า 300,000 บาทต่อปี สำหรับช่องสัญญาณขนาด 2Mbps (ที่มาจาก บริษัท สามารถอินโฟเนต จำกัด) ซึ่งเป็นทางออกที่ทำไม่ได้สำหรับหน่วยงานดังกล่าว

แต่ด้วยความแพร่หลายของการใช้การเชื่อมต่ออินเทอร์เน็ตแบบบรอดแบรนด์ในระดับครัวเรือนนั้นได้ขยายไปสู่การใช้ในระดับองค์กรมากขึ้น ซึ่งเป็นผลมาจากราคาของบริการที่มีแนวโน้มลดลงในขณะที่สามารถได้รับช่องสัญญาณการเชื่อมต่ออินเทอร์เน็ตที่เพียงพอสำหรับการใช้งานในองค์กรขนาดเล็กถึงขนาดกลางที่มีความต้องการใช้บริการเชื่อมต่อออกสู่อินเทอร์เน็ตเพียงอย่างเดียว เช่น บริการ one2connext โดย Ji-Net ที่ความเร็ว 1 Mbps (ดาวน์โหลด) ในราคา 690.-บาท/เดือน เป็นต้น

ด้วยโอกาสข้างต้นนี้ อำนวยให้องค์กรที่มีขีดจำกัดในงบประมาณด้านเทคโนโลยีสารสนเทศสามารถเข้าถึงบริการอินเทอร์เน็ตที่มีขนาดช่องสัญญาณมากขึ้น ประกอบกับการนำเทคโนโลยีที่เรียกว่า load balancing มาประยุกต์ใช้เพื่อทำให้เกิดการรวมช่องสัญญาณเชื่อมต่ออินเทอร์เน็ตให้มีขนาดเพิ่มขึ้นจากการใช้การเชื่อมต่ออินเทอร์เน็ตแบบบรอดแบรนด์จากผู้ให้บริการอินเทอร์เน็ต (Internet Service Provider : ISP) มากกว่าหนึ่งรายร่วมกัน และเพื่อปรับปรุงความน่าเชื่อถือและความพร้อมใช้งาน (Reliability and Availability) ซึ่งเป็นข้อด้อยของการเชื่อมต่อแบบบรอดแบนด์ให้มีประสิทธิภาพที่สูงขึ้น โดยมีเป้าหมายเพื่อทำให้ช่องสัญญาณการเชื่อมต่อรวมมีขนาดใหญ่และมีความพร้อมเพียงพอในระดับที่สามารถรองรับการใช้งานใน

ระดับองค์กรได้ และเพื่อทดแทนการลงทุนเช่าสายเคเบิลเชื่อมต่ออินเทอร์เน็ตความเร็วสูงที่ต้องเสียค่าใช้จ่ายที่สูงกว่าการเชื่อมต่อแบบบรอดแบนด์มาก

อย่างไรก็ตามเทคโนโลยี load balancing ลักษณะดังกล่าวสามารถถูกจัดหาได้จากการซื้อผลิตภัณฑ์ที่ถูกพัฒนาโดยผู้ผลิตต่างๆ ที่มีอยู่แล้ว เช่น NetComm , XRoads Network , F5 , Edimax เป็นต้น อย่างไรก็ตามการจัดซื้อผลิตภัณฑ์ดังกล่าวเป็นไปได้ยากสำหรับองค์กรที่มีขีดจำกัดในงบประมาณด้านเทคโนโลยีสารสนเทศโดยเฉพาะอย่างยิ่งในส่วนของงบประมาณที่ถูกจัดสรรเพื่อจัดหาบริการเชื่อมต่ออินเทอร์เน็ต ทางออกหนึ่งที่เป็นไปได้คือการพัฒนาระบบ load balancing สำหรับเชื่อมต่อออกสู่อินเทอร์เน็ตขึ้นมาใช้เองภายในองค์กร

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

การวิเคราะห์ออกแบบและพัฒนาระบบ load balancing นี้มุ่งเน้นการพัฒนาให้เกิดระบบต้นแบบที่สามารถนำไปประยุกต์ใช้งานได้จริง และยังสามารถใช้เป็นระบบรากฐานสำหรับการปรับปรุงเพิ่มศักยภาพของระบบในด้านต่างๆต่อไปในอนาคต โดยมีวัตถุประสงค์ดังนี้

1. เพื่อเพิ่มศักยภาพการเข้าถึงบริการในอินเทอร์เน็ตขององค์กรขนาดกลางและเล็ก ผ่านช่องทางการเชื่อมต่อที่มีขนาดช่องสัญญาณรวมมากขึ้น
2. เพื่อรักษาความพร้อมใช้งานช่องทางการเชื่อมต่ออินเทอร์เน็ตขององค์กรขนาดกลางและเล็กให้สามารถตอบสนองความต้องการขององค์กรได้ตลอดเวลาที่มีการใช้งาน
3. เพื่อลดภาระค่าใช้จ่ายด้านเทคโนโลยีสารสนเทศขององค์กรขนาดกลางและเล็กที่มีขีดจำกัดในงบประมาณด้านนี้ ให้สามารถมีโอกาสเข้าถึงอินเทอร์เน็ตความเร็วสูงที่มีระดับความพร้อมใช้งานที่เทียบเท่ากับการเชื่อมต่ออินเทอร์เน็ตผ่านช่องทางการเชื่อมต่อแบบเหมาเช่าที่ราคาสูง

1.3 ขอบเขตของการพัฒนาโครงงาน

ระบบที่องค์กรใช้เพื่อเข้าถึงบริการที่อยู่บนอินเทอร์เน็ตเพียงอย่างเดียว และไม่มีบริการใดๆ ภายในเครือข่ายขององค์กรที่สนับสนุนให้เข้าถึงได้จากภายนอกองค์กรโดยใช้ช่องทางการเชื่อมต่ออินเทอร์เน็ตที่องค์กรมีใช้งานอยู่ ฉะนั้นกระแสข้อมูลอินเทอร์เน็ตทั้งหมดจึงเริ่มต้นจากเครือข่ายภายในองค์กรและถูกส่งมารวมอยู่ที่ระบบก่อนส่งต่อไปยังเส้นทางการเชื่อมต่ออินเทอร์เน็ต ต่อไปนี้จะเรียกระบบ Load Balancing นี้ว่า LBS-IA (Load Balancing System for Internet Access) ซึ่งหมายถึงระบบ Load Balancing เพื่อสนับสนุนการเข้าถึงบริการที่อยู่ในอินเทอร์เน็ต โดยการพัฒนาจะประกอบไปด้วยรายละเอียดเบื้องต้นดังต่อไปนี้

1. พัฒนาระบบ LBS-IA เพื่อจัดการการจราจรข้อมูลอินเทอร์เน็ตขององค์กร

2. จัดหาระบบ Management Station ที่เหมาะสมเพื่อควบคุมและจัดการการทำงานของ LBS-IA สำหรับผู้ใช้งาน

1.4 ขั้นตอนและแนวทางการพัฒนา

แนวทางในการพัฒนาระบบในโครงงานนี้มุ่งเน้นการนำเทคโนโลยี Linux ที่ได้ถูกพัฒนาไว้แล้วมาประยุกต์มาใช้ให้เกิดเป็นระบบที่อยู่ภายใต้ขอบเขตและสอดคล้องกับวัตถุประสงค์ของโครงงานที่ได้ระบุไว้ ทั้งนี้เนื่องจากผู้เขียนได้ประเมินแล้วว่า ระบบ LBS-IA นี้มีความซับซ้อนในองค์ประกอบของระบบมากเกินกว่าที่จะพัฒนาซอฟต์แวร์ขึ้นมาใช้งานเองภายใต้ระยะเวลาที่จำกัด ยิ่งไปกว่านั้นผู้เขียนยังไม่มีพื้นฐานในด้านการพัฒนาซอฟต์แวร์รวมอยู่ด้วย ดังนั้นจึงสรุปเป็นขั้นตอนในการพัฒนาระบบงานได้ดังนี้

1. ศึกษาหลักการและองค์ประกอบหลักของระบบ Load Balancing
2. ศึกษาเทคโนโลยี Linux ที่เกี่ยวข้องสำหรับนำมาประยุกต์ใช้งานกับระบบ
3. วิเคราะห์และออกแบบระบบบนพื้นฐานของเทคโนโลยี Linux ที่จะเลือกใช้
4. การประยุกต์ใช้เทคโนโลยี Linux ให้ตรงกับความต้องการของระบบที่ได้วิเคราะห์และออกแบบไว้
5. ทดสอบระบบบนสภาพแวดล้อมจำลอง
6. สรุปผลการพัฒนาระบบ
7. จัดทำเอกสารประกอบการใช้งาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ศึกษาเทคโนโลยี Load Balancing เพื่อการนำมาประยุกต์ใช้งานและศึกษาแนวโน้มและทางเลือกรวมทั้งข้อจำกัดต่างๆของเทคโนโลยี เพื่อใช้เป็นแนวทางในการพัฒนาระบบขึ้นมาใช้งานเองภายในองค์กร
2. ประยุกต์ใช้งานเทคโนโลยี Load Balancing ให้เหมาะสมกับโครงงาน
3. เป็นระบบต้นแบบที่ได้ศึกษา วิเคราะห์และออกแบบ รวมทั้งการพัฒนาเพื่อการนำไปใช้งานกับสภาพแวดล้อมจริงต่อไป

บทที่ 2
ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

แนวคิดของการใช้หลายช่องทางการเชื่อมต่อเพื่อปรับปรุงขนาดช่องสัญญาณและความพร้อมใช้งานของช่องสัญญาณในการเข้าถึงอินเทอร์เน็ตสำหรับองค์กรนั้นไม่ใช่เป็นเรื่องใหม่ เทคโนโลยีหนึ่งที่ทำให้แนวคิดนี้เป็นไปได้คือการใช้ระบบที่กันเรียกว่า load balancing ระบบดังกล่าวนี้ช่วยให้สามารถกระจายการจราจรของข้อมูลออกสู่และมาจากอินเทอร์เน็ตผ่านช่องทางการเชื่อมต่อที่มีอยู่เพื่อเพิ่มปริมาณการใช้ช่องสัญญาณรวม และยังสามารถปรับเปลี่ยนเส้นทางการจราจรของข้อมูลไปใช้ช่องทางการเชื่อมต่อที่พร้อมใช้งานได้ในกรณีที่เกิดการขัดข้องในช่องทางการเชื่อมต่ออื่น

ในระดับของหลักการนั้น ระบบ load balancing ต้องมีความสามารถในเลือกช่องทางการเชื่อมต่อที่จะใช้กระจายการจราจรของข้อมูล พร้อมทั้งการกำหนดการจราจรของข้อมูลกับช่องทางการเชื่อมต่อที่เลือกไว้ได้ และยังต้องสามารถเฝ้าตรวจข้อขัดข้องที่เกิดกับช่องทางการเชื่อมต่อที่มีอยู่เพื่อให้สามารถเบี่ยงการจราจรของข้อมูลไปบนช่องทางการเชื่อมต่อที่พร้อมใช้งานได้ อย่างไรก็ตามมีตัวแปรหลักที่ส่งผลต่อแนทางการออกแบบระบบ load balancing คือการที่องค์กรได้รับมอบชุดของ public IP address จาก Internet Assigned Number Authority (IANA) มาให้บริหารจัดการเองหรือไม่

ในกรณีที่องค์กรได้รับมอบชุดของ public IP address ที่ไม่ขึ้นอยู่กับชุดของ IP address ของ ISP ที่องค์กรนั้นใช้บริการอยู่ สามารถประยุกต์ใช้ระบบ load balancing ได้โดยการใช้เทคนิคที่เรียกว่า BGP peering ที่ตั้งค่าให้ BGP router ขององค์กรเชื่อมต่อโดยตรงคู่กับ BGP router ของ ISP และใช้ความสามารถของ routing protocol ที่มีอยู่เพื่อการรวมช่องสัญญาณการเข้าถึงอินเทอร์เน็ตและการเบี่ยงเบนการจราจรของไปยังช่องทางการเชื่อมต่อที่พร้อมใช้งานกว่า

ในกรณีที่องค์กรได้รับการกำหนดชุดของ public IP address โดย ISP ที่องค์กรนั้นใช้บริการอยู่ สามารถประยุกต์ใช้ระบบ load balancing ได้โดยการใช้ Network Address Translation (NAT) เข้ามาช่วยในการกำหนดการจราจรของข้อมูลไปบนช่องทางการเชื่อมต่อที่มีอยู่ได้

อย่างไรก็ตาม ไม่ว่าการออกแบบจะเป็นไปในทิศทางใด แนวทางการประยุกต์ใช้ระบบ load balancing ยังคงต้องถูกพัฒนาบนพื้นฐานขององค์ประกอบหลักเดียวกัน

2.1 องค์ประกอบหลักของระบบ Load Balancing

องค์ประกอบหลักที่สนับสนุนให้ระบบ load balancing ทำงานได้นั้น ประกอบด้วย 3 ส่วนที่ทำงานสัมพันธ์กันคือ

2.1.1 ส่วนการนำส่งกระแสข้อมูลอินเทอร์เน็ต

สำหรับองค์กรที่ถูกกำหนดให้ใช้กลุ่มของ public IP address ที่ต่างกันจากแต่ละ ISP (องค์กรเป้าหมายสำหรับการพัฒนาระบบ load balancing ในโครงงานนี้) แนวทางในการจัดการให้สามารถนำส่งกระแสข้อมูล หรือ "load" ไปยังแต่ละช่องทางการเชื่อมต่อได้คือการนำ NAT (Network Address Translation) เข้ามาประยุกต์ใช้ เช่นนี้จะทำให้ private IP address โฮสภายในองค์กรสามารถผูกติดกับ public IP address ที่แตกต่างกันได้พร้อมกัน เป็นผลให้ load ที่เกิดจากโฮสแต่ละเครื่องสามารถถูกนำส่งออกสู่อินเทอร์เน็ตผ่านแต่ละช่องทางการเชื่อมต่อที่มีอยู่ได้ตามนโยบายการกระจาย load ของ LBS-IA

2.1.2 ส่วนการตรวจสอบความพร้อมใช้งานของช่องทางการเชื่อมต่ออินเทอร์เน็ต

การตรวจสอบความพร้อมใช้งานของช่องทางการเชื่อมต่อมุ่งเน้นที่การตรวจสอบแต่ละช่องทางการเชื่อมต่อที่ LBS-IA ใช้เชื่อมต่อไปยังแต่ละ ISP ซึ่งทำได้โดยการเฝ้าตรวจโดยหลักการนั้นสามารถใช้ 2 แนวทางร่วมกัน คือ แบบ Passive และ Active

โดยหลักการนั้น การเฝ้าตรวจแบบ Passive หรือเรียกอีกอย่างว่า "Passive Monitoring" เป็นความสามารถในการตรวจจับแพ็กเก็ตที่เกิดขึ้นและนำมาวิเคราะห์เพื่อหาข้อมูลที่กระบวนการเฝ้าตรวจกำลังค้นหาอยู่ ในกรณีของ LBS-IA ข้อมูลดังกล่าวคือสถานะที่แสดงว่าช่องทางการเชื่อมต่ออินเทอร์เน็ตมีความพร้อมให้ใช้งานหรือไม่

โดยทั่วไปลักษณะการเชื่อมต่อเครือข่ายจะเกิดขึ้นแบบสองกระแสเส้นทางคือกระแสข้อมูลขาออกและกระแสข้อมูลขาเข้า ฉะนั้นตราบใดที่มีการสร้างการเชื่อมต่อเริ่มจากเครือข่ายภายในองค์กรย่อมมีกระแสข้อมูลขาเข้าบนช่องทางการเชื่อมต่อเสมอถ้าช่องทางนั้นยังสามารถใช้งานได้ การที่ LBS-IA สามารถตรวจจับแพ็กเก็ตของกระแสข้อมูลขาเข้าของช่องทางการเชื่อมต่ออินเทอร์เน็ตที่มีอยู่ก็ทำให้สามารถระบุได้ว่าช่องทางการเชื่อมต่อนั้นๆมีความพร้อมในการใช้งานเพื่อรับส่งกระแสข้อมูลอินเทอร์เน็ต

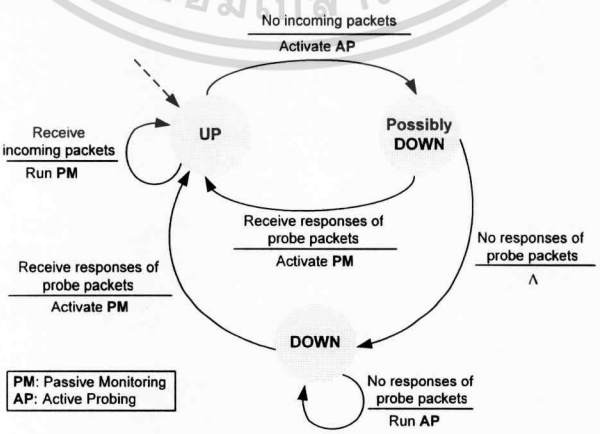
อย่างไรก็ตามยังคงมีข้อจำกัดของการนำ Passive monitoring มาประยุกต์ใช้กับ LBS-IA ลักษณะนี้ คือการเฝ้าตรวจกระแสข้อมูลขาเข้าเพียงอย่างเดียวทำให้ LBS-IA มีข้อมูลไม่เพียงพอในการแยกแยะว่าช่องทางการเชื่อมต่อที่มีอยู่นั้นมีความพร้อมใช้งานได้หรือไม่ถ้าไม่สามารถตรวจจับแพ็กเก็ตของกระแสข้อมูลขาเข้าได้เลย เพื่อแก้ข้อจำกัดดังกล่าว LBS-IA จำเป็นต้องมีการเฝ้าตรวจแบบ Active ทำงานร่วมอยู่ด้วย

การเฝ้าตรวจแบบ Active หรือเรียกอีกอย่างว่า "Active Probing" เป็นความสามารถในการสร้างและส่งกระแสข้อมูลไปบนช่องทางการเชื่อมต่อเพื่อจำลองพฤติกรรมของการเชื่อมต่อ

เครือข่ายและตรวจจับผลของการจำลองพฤติกรรมนั้น ผลที่ได้จะถูกนำมาวิเคราะห์ตามวัตถุประสงค์ของการเฝ้าตรวจ สำหรับเป้าหมายของการเฝ้าตรวจโดย LBS-IA คือการตรวจจับพฤติกรรมการส่งและรับกระแสข้อมูลระหว่าง LBS-IA และเป้าหมายที่ถูกใช้เพื่อการจำลองพฤติกรรม (probe target) เพื่อวิเคราะห์ว่าช่องทางการเชื่อมต่อที่ใช้นั้นมีความพร้อมในการใช้งานหรือไม่

ประเด็นหลักที่ต้องนำมาพิจารณาเพื่อพัฒนา Active Probing นี้จึงเกี่ยวพันกับการเลือกชนิดของแพ็กเก็ตที่จะใช้สร้างกระแสข้อมูลจำลองพฤติกรรม (probe packet) และการเลือก probe target เนื่องจากการใช้ Active Probing เป็นส่วนหนึ่งที่ทำให้เกิด load บนช่องทางการเชื่อมต่ออินเทอร์เน็ตนอกเหนือจาก load ที่เกิดจากโฮสภายในเครือข่ายขององค์กร ดังนั้น probe packet ไม่ควรใช้ทรัพยากรที่มีอยู่บนช่องทางการเชื่อมต่ออินเตอร์ เช่น bandwidth มากเกินความจำเป็น แพ็กเก็ตประเภท ICMP และ UDP น่าจะเป็นตัวเลือกที่เหมาะสม สำหรับ probe target นั้นจำเป็นต้องคำนึงถึงการเลือกใช้โฮสที่สามารถรองรับปริมาณของ probe packet ได้และพร้อมให้ใช้งานได้ตลอดเวลา นอกจากนั้นจะต้องมีที่ตั้งอยู่อีกด้านหนึ่งของช่องทางการเชื่อมต่อที่ LBS-IA เชื่อมต่ออยู่เพื่อให้สามารถทดสอบความพร้อมใช้งานของช่องทางการเชื่อมต่อได้จริง เช่นการเลือกใช้ DNS server ของ ISP เป็น probe target ก็นับว่าเป็นทางเลือกที่เหมาะสมทางหนึ่ง

การประยุกต์ใช้ Active Probing ร่วมกับ Passive Monitoring ทำได้โดยการกำหนดให้ Passive Monitoring ทำหน้าที่เป็นระบบเฝ้าตรวจหลักเนื่องจากการเฝ้าตรวจลักษณะดังกล่าวสามารถตรวจจับแพ็กเก็ตของกระแสข้อมูลขาเข้าได้เกือบทุกกรณีถ้ามีการใช้งานอินเทอร์เน็ตจากโฮสในองค์กรและเป็นการเฝ้าตรวจที่ไม่ทำให้เกิด load บนช่องทางการเชื่อมต่ออินเทอร์เน็ต สำหรับ Active Probing นั้นจะถูกใช้เป็นระบบเฝ้าตรวจเสริมในกรณีที่ Passive Monitoring ไม่สามารถยืนยันได้ว่าช่องทางการเชื่อมต่ออินเทอร์เน็ตมีความพร้อมใช้งานหรือไม่ซึ่งเป็นผลจากการที่ไม่สามารถตรวจจับแพ็กเก็ตของกระแสข้อมูลขาเข้าได้



รูปที่ 2.1 การแสดงสถานะของช่องทางการเชื่อมต่อด้วยการใช้ระบบเฝ้าตรวจแบบ Passive Monitoring และ Active Probing ร่วมกัน

ในรูปที่ 2 แสดงให้เห็นหลักการทำงานร่วมกันของ Passive Monitoring และ Active Probing ในรูปแบบของ state machine จะเห็นได้ว่าตราบใดที่ PM สามารถตรวจจับแพ็กเก็ตของกระแสข้อมูลขาเข้าบนช่องทางการเชื่อมต่ออินเทอร์เน็ตได้ ช่องทางนั้นจะถูกระบุว่าอยู่ในสถานะที่พร้อมใช้งาน หรือ UP แต่เมื่อแพ็กเก็ตดังกล่าวไม่สามารถถูกตรวจจับได้ตามเกณฑ์ที่กำหนดไว้โดย PM แล้ว AP จะถูกเรียกให้ทำงานเพื่อตรวจสอบให้แน่ใจว่าไม่สามารถใช้งานช่องทางการเชื่อมต่อที่กำลังเฝ้าตรวจอยู่ได้ ในระหว่างนี้สถานะของช่องทางการเชื่อมต่อยังคงถูกพิจารณาว่าพร้อมใช้งานอยู่โดยถูกกำหนดเป็น Possibly DOWN ถ้า AP ตรวจสอบได้ผลว่าช่องทางการเชื่อมต่อยังคงพร้อมใช้งานอยู่นั่นคือมีการตอบกลับมาจาก probe target แล้ว AP จะหยุดทำงานและรอจนกว่า PM จะรายงานพฤติกรรมความไม่พร้อมใช้งานของช่องทางการเชื่อมต่ออีกครั้ง ถ้า AP ตรวจสอบอีกครั้งและได้ผลว่าไม่มีการตอบกลับมาจาก probe target เช่นนี้สถานะของช่องทางการเชื่อมต่อจะถูกเปลี่ยนเป็นไม่พร้อมใช้งาน หรือ DOWN และการทำงานของ AP จะดำเนินไปอย่างไม่มีกำหนดจนกว่าสถานะของช่องทางการเชื่อมต่อจะถูกเปลี่ยนเป็น UP อีกครั้ง

2.1.3 ส่วนการกระจายกระแสข้อมูลบนช่องทางการเชื่อมต่ออินเทอร์เน็ต

ความสมดุลของกระแสข้อมูลอินเทอร์เน็ตที่ถูกกระจายไปบนช่องทางการเชื่อมต่อที่มีอยู่นั้นขึ้นอยู่กับแนวทางที่ LBS-IA นำมาใช้ในการเลือกช่องทางการเชื่อมต่อที่เหมาะสมกับการนำส่งกระแสข้อมูลที่เกิดจากแอ็พพลิเคชั่นแต่ละชนิด เช่น HTTP และ FTP เป็นต้น โดยมีเป้าหมายเพื่อทำให้ load ที่เกิดขึ้นมีความสมดุลกันในทุกช่องทางมากที่สุดเท่าที่เป็นไปได้

เนื่องจากบางแอ็พพลิชันจำเป็นต้องสร้างมากกว่าหนึ่งการเชื่อมต่อที่มีต้นกำเนิดจาก IP address เดียวกันเพื่อให้สามารถรับส่งข้อมูลกับบริการที่อยู่บนอินเทอร์เน็ตได้สำเร็จ ตัวอย่างเช่น FTP จำเป็นต้องมีการสร้างการเชื่อมต่อ 2 แบบคือ control connection และ data connection ไปยัง FTP server ถ้าทั้งสองการเชื่อมต่อถูกนำส่งผ่านช่องทางการเชื่อมต่อที่ต่างกัน นั่นคือแต่ละการเชื่อมต่อถูกผูกติดกับ public IP address ของแต่ละช่องทางการเชื่อมต่อ เป็นผลให้ FTP sever ปฏิเสธให้บริการเนื่องจาก IP address ต้นกำเนิดของ data connection ต่างไปจากของ control connection ดังนั้นจึงมีความจำเป็นที่ต้องกำหนดให้ทั้งสองการเชื่อมต่อถูกนำส่งไปบนช่องทางการเชื่อมต่อเดียวกัน ในทางตรงกันข้าม บางบริการในอินเทอร์เน็ตไม่มีข้อจำกัดในเรื่อง IP address ต้นกำเนิดของแพ็กเก็ตฉะนั้นการกำหนดการเชื่อมต่อของแอ็พพลิเคชันที่ใช้บริการลักษณะนี้ไปบนช่องทางการเชื่อมต่อที่ต่างกันจะไม่ได้รับผลกระทบเช่นเดียวกับ FTP ตัวอย่างเช่น HTTP เป็นต้น จะเห็นได้ว่าการเลือกช่องทางการเชื่อมต่อโดย LBS-IA จำเป็นต้องพิจาณาให้เหมาะสมกับชนิดของแอ็พพลิเคชันที่สร้างการเชื่อมต่อด้วย ในกรณีของแอ็พพลิเคชันที่มีพฤติกรรมการเชื่อมต่อเป็นไปในแนวทางเดียวกับ FTP นั้นการเลือกช่องทางการเชื่อมต่อต้องพิจารณาในมุมมองของเซสชั่นที่ประกอบด้วย IP address ของโฮสต้นทางและของปลายทาง (Two-tuple) ของการเชื่อมต่อ เพื่อกำหนดให้ใช้ช่องทางการเชื่อมต่อเดียวกันจนสิ้นสุดเซสชั่น

ในส่วนของการกระจาย load ไปบนช่องทางการเชื่อมต่อนั้นขึ้นอยู่กับนโยบายที่ LBS-IA นำมาใช้เพื่อกำหนดให้แต่ละกระแสข้อมูลถูกส่งออกไปยังช่องทางการเชื่อมต่อที่มีอยู่ นโยบายดังกล่าวมีผลมาจากการเน้นพิจารณาคุณลักษณะของ bandwidth หรือจำนวนการเชื่อมต่อที่เกิดขึ้นในแต่ละช่องทางเป็นหลัก

สำหรับการพิจารณาโดยใช้คุณลักษณะของ bandwidth นั้นโดยทั่วไปแล้ว LBS-IA ควรเลือกใช้ช่องทางการเชื่อมต่อที่มีการใช้งานต่ำสุดเพราะว่างช่องทางการเชื่อมต่อลักษณะดังกล่าวดูเหมือนว่าจะมี bandwidth เหลือมากที่สุดสำหรับแต่ละกระแสข้อมูลที่กำลังจะถูกสร้างขึ้น

เกณฑ์ทั่วไปที่นำมาใช้พิจารณาในมุมมองของ bandwidth ประกอบด้วย 2 ส่วนคือ ปริมาณ bandwidth ที่ถูกจัดสรร (capacity) และ ปริมาณ bandwidth ที่เหลือ (available bandwidth) ซึ่งครอบคลุมถึงช่องทางการเชื่อมต่อระหว่าง LBS-IA กับ ISP และตลอดเส้นทางการเชื่อมต่อจาก LBS-IA ไปถึงปลายทาง เกณฑ์ดังกล่าวประกอบด้วย

- *Link capacity* หมายถึง ปริมาณ bandwidth ที่ถูกจัดสรรบนช่องทางการเชื่อมต่อระหว่าง LBS-IA และ ISP
- *Path capacity* หมายถึง ปริมาณ bandwidth ที่เหลือตลอดเส้นทางการเชื่อมต่อจาก LBS-IA ไปจนถึงปลายทางของการเชื่อมต่อ
- *Available bandwidth of link* หมายถึง ปริมาณ bandwidth ที่เหลือบนช่องทางการเชื่อมต่อระหว่าง LBS-IA และ ISP
- *Available bandwidth of path* หมายถึง ปริมาณ bandwidth ที่เหลือตลอดเส้นทางการเชื่อมต่อจาก LBS-IA ไปจนถึงปลายทางการเชื่อมต่อ

ในส่วนของการพิจารณาโดยใช้จำนวนการเชื่อมต่อในแต่ละช่องทางเป็นเกณฑ์เพื่อกระจาย load ให้สมดุลในแต่ละช่องทางนั้น LBS-IA จำเป็นต้องมีกลไกที่ใช้บันทึกจำนวนการเชื่อมต่อที่เกิดขึ้นเพื่อช่วยให้การกระจาย load เกิดความสมดุลกันตามจำนวนการเชื่อมต่อโดยรวมที่ LBS-IA กำลังรับภาระอยู่

สำหรับการเลือกช่องทางการเชื่อมต่อนั้นจำเป็นต้องเกิดขึ้น ณ จุดเริ่มต้นของการสร้างการเชื่อมต่อเสมอเพื่อให้ NAT สามารถผูกติด public IP address ของช่องทางการเชื่อมต่อที่ถูกเลือกกับ private IP address ของโฮสก่อนที่จะนำส่งกระแสข้อมูลไปบนช่องทางการเชื่อมต่อนั้นๆ ได้ การเลือกช่องทางการเชื่อมต่อแบบบันทึกสถานะ (Stateful Link Assignment) เป็นแนวทางหนึ่งที่ทำได้โดยการบันทึกการตัดสินใจเลือกใช้ช่องทางการเชื่อมต่อสำหรับสำหรับแพ็กเก็ตแรกเพื่อให้แน่ใจว่าแพ็กเก็ตต่อๆไปของการเชื่อมต่อเดียวกันจะถูกส่งไปบนช่องทางการเชื่อมต่อที่เลือกไว้ตั้งแต่เริ่มต้นสร้างการเชื่อมต่อ เช่นนี้จึงจำเป็นต้องมีการประยุกต์ใช้รูปแบบหนึ่งของตารางการบันทึกข้อมูล (lookup table) เพื่อบันทึกข้อมูลการตัดสินใจเลือกใช้ช่องทางการเชื่อมต่อข้างต้น

2.2 Linux Policy Routing

การค้นหาเส้นทางเพื่อนำส่งข้อมูลโดย router ทั่วไปนั้นจะตัดสินใจเลือกเส้นทางบนพื้นฐานของ destination IP address ของแพ็กเก็ต แต่สำหรับการนำส่งกระแสข้อมูลไปบนเส้นทางการเชื่อมต่อที่ LBS-IA ดูแลจัดการอยู่นั้น จำเป็นต้องตัดสินใจเลือกช่องทางการเชื่อมต่อเพื่อรักษาสมดุลของ load ที่เกิดขึ้นบนช่องทางที่มีอยู่ทั้งหมด ฉะนั้นการพิจารณาเลือกช่องทางการเชื่อมต่อด้วยการใช้ destination IP address ของแพ็กเก็ตเพียงอย่างเดียวจึงไม่มีความยืดหยุ่นเพียงพอที่จะสนับสนุนการทำงานของระบบ ฉะนั้น LBS-IA จึงต้องสามารถเลือกช่องทางการเชื่อมต่อได้โดยการพิจารณา field อื่นๆ ของแพ็กเก็ต เช่น source IP address, IP protocol, Transport protocol เป็นต้น แนวทางการค้นหาเส้นทางเช่นนี้เรียกว่า Policy Routing

สำหรับการใช้ Policy Routing บนพื้นฐานของระบบปฏิบัติการ Linux สามารถสรุปเป็นหลักการที่ประกอบด้วย 2 ส่วนดังนี้

2.2.1 องค์ประกอบหลักภายใน Policy Routing

หลักการทำ Policy Routing อยู่บนพื้นฐานของการประยุกต์ใช้องค์ประกอบ 3 อย่างคือ

- *Address* เป็นองค์ประกอบที่ใช้ระบุตำแหน่งของบริการที่อยู่บนเครือข่ายคอมพิวเตอร์ นั่นคือการระบุวัตถุที่กำลังทำงานหรือกำลังถูกกระทำที่เป็นผลมาจากการตัดสินใจเลือกตามนโยบายที่กำหนดไว้ใน Policy Routing
- *Route* เป็นองค์ประกอบที่ใช้ระบุตำแหน่งของ Address นั่นคือเป็นองค์ประกอบที่ช่วยในการตัดสินใจเลือกเส้นทางการนำส่งแพ็กเก็ตด้วยการใช้เกณฑ์การตัดสินใจอื่นนอกเหนือจากการตัดสินใจเลือกโดยใช้เกณฑ์ของ destination IP address แบบเดิม ดังนั้นวิธีการที่ใช้เลือกเส้นทางการนำส่งแพ็กเก็ตจึงเปลี่ยนไปจากเดิม แต่เมื่อเส้นทางถูกเลือกแล้ววิธีการใช้เส้นทางนั้นยังคงเป็นไปตามวิธีการเดิมของการนำส่งแพ็กเก็ตผ่านเครือข่ายคอมพิวเตอร์
- *Rule* เป็นองค์ประกอบที่ใช้ระบุตำแหน่งของ Route นั่นคือสามารถนำกระบวนการที่เรียกว่าการคัดกรอง (filter) มาใช้กับแพ็กเก็ตเพื่อจับคู่กับแพ็กเก็ตที่มีคุณลักษณะที่สอดคล้องกับข้อกำหนดต่างๆ ภายใน Rule และเลือกโครงสร้าง Route ให้กับแพ็กเก็ตที่ผ่านการคัดกรองนั้นแล้ว

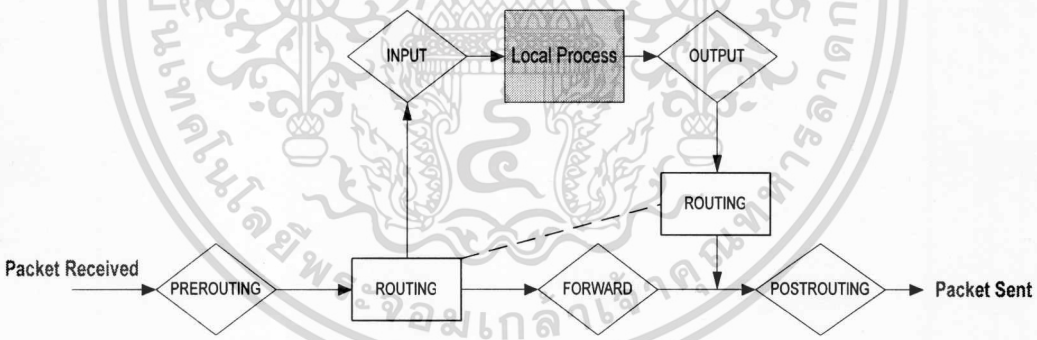
องค์ประกอบทั้งสามนี้ทำให้เกิดโครงสร้างพื้นฐานที่เป็นแกนหลักของการใช้ Policy Routing โดยสามารถนำประยุกต์ใช้แยกจากกันโดยอิสระได้ แต่การประยุกต์ใช้ Policy Routing ที่มีประสิทธิภาพเกิดจากการรวมการใช้งานองค์ประกอบทั้งสามร่วมกันได้อย่างสอดคล้องเป็นหนึ่งเดียว

2.2.2 Routing Policy Database

บนพื้นฐานของระบบปฏิบัติการ Linux นั้น การประยุกต์ใช้ Policy Routing ถูกกระทำผ่านกลไกที่เรียกว่า Routing Policy Database (RPDB) ที่ซึ่งประกอบไปด้วยชุดของ Route, Routing table และ Rule สิ่งที่ RPDB ทำคือการเตรียมให้มีโครงสร้างภายในและกลไกต่างๆ สำหรับการประยุกต์ใช้ Rule ใน Policy Routing และยังสามารถรองรับการทำหลาย Routing table ที่มีให้ใช้งานภานในระบบปฏิบัติการ Linux ด้วย

2.3 เทคโนโลยีของ Linux ที่นำมาประยุกต์ใช้เพื่อพัฒนาระบบ

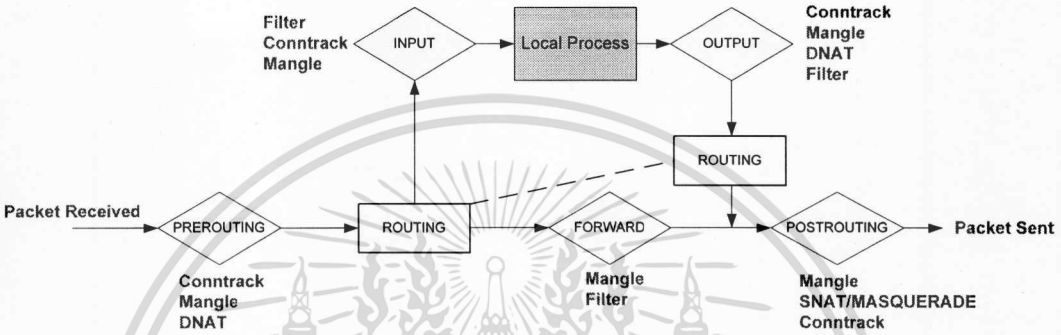
LBS-IA จะถูกพัฒนาบนโครงสร้างพื้นฐานใน kernel ของ Linux ที่เรียกว่า Netfilter ซึ่งสนับสนุนให้สามารถนำ module ต่างๆมาติดตั้งทำงานร่วมกันได้ และ iptables modules จะเป็นชุดของเครื่องมือที่จะนำมาประยุกต์ใช้เพื่อทำหน้าที่เกี่ยวกับการคัดเลือกแพ็กเก็ต เพื่อสนับสนุนกลไกของ LBS-IA สำหรับ Netfilter และ iptables สามาถถูกนำมาใช้ได้จาก Linux kernel 2.4 ขึ้นไป ภายในโครงสร้างของ Netfilter ประกอบด้วยจุดหลักหรือเรียกอีกอย่างว่า chain ที่กำหนดให้แต่ละแพ็กเก็ตต้องผ่านเพื่อให้ iptable modules ที่ลงทะเบียนใช้งานไว้นั้นสามารถคัดเลือกแพ็กเก็ตได้ ดังแสดงให้เห็นในรูปที่ 2.2



รูปที่ 2.2 เส้นทางของแพ็กเก็ตในโครงสร้างของ Netfilter

- PREROUTING เป็นจุดแรกที่รับแพ็กเก็ตจากภายนอกเข้าสู่ระบบ เพื่อการประมวลผลตาม module ที่ได้ลงทะเบียนไว้
- ROUTING เป็นที่ซึ่งพิจารณาว่าแพ็กเก็ตมีปลายทางอยู่ภายในตัว LBS-IA เองหรือควรถูกส่งไปยังช่องทางการเชื่อมต่อภายนอก
- FORWARD เป็นที่ซึ่งรับแพ็กเก็ตที่มีปลายทางออกสู่ภายนอกเครือข่าย ณ จุดนี้แพ็กเก็ตจะถูกกระทำตามหลักการของ Firewall เช่น drop, queue, accept
- POSTROUTING เป็นจุดสุดท้ายที่ประมวลผลแพ็กเก็ตก่อนส่งออกสู่ช่องทางการเชื่อมต่อภายนอก การประมวลผลจะเกิดขึ้นตามการเรียกใช้ module ที่ได้ลงทะเบียนไว้

iptables จะประกอบด้วย modules ต่างๆ ที่ถูกลงทะเบียนเข้าไว้กับ chain ใน Netfilter เพื่อทำหน้าที่คัดเลือกแพ็กเก็ตที่ผ่านจุดต่างๆ ภายในโครงสร้างของ Netfilter กฎของการคัดเลือกแพ็กเก็ตจะอยู่ในรูปแบบของตาราง (table) ซึ่ง kernel สามารถนำไปใช้เป็นเกณฑ์ในการคัดเลือกแพ็กเก็ตที่ผ่านเข้ามา วิธีการคัดเลือกแพ็กเก็ตของ iptables ที่สามารถเลือกนำมาใช้ร่วมกันใน LBS-IA นี้แบ่งเป็น 3 กลุ่มหลักคือ NAT table, Filter table และ Mangle table ในรูปที่ 2.3 แสดงให้ภาพรวมของ iptables modules และลำดับการเรียกใช้แต่ละโมดูลในโครงสร้างของ Netfilter



รูปที่ 2.3 การลงทะเบียนใช้ iptables modules ในโครงสร้างของ Netfilter

2.3.1 Connection Tracking

กลไกการติดตามการเชื่อมต่อของแพ็กเก็ตจะถูกลงทะเบียนใช้ที่ PREROUTING ซึ่งเป็นจุดแรกที่แพ็กเก็ตถูกรับเข้าสู่ระบบ อันที่จริงแล้ว module นี้เป็นส่วนหนึ่งของ NAT table ที่สามารถแยกการทำงานออกจาก NAT code ได้เพื่อให้ระบบสามารถติดตามสถานะการเชื่อมต่อของแต่ละแพ็กเก็ตก่อนที่จะถูกกระทำโดย NAT code สถานะการเชื่อมต่อของแพ็กเก็ตสามารถถูกระบุได้ดังนี้

- ESTABLISHED แพ็กเก็ตเป็นส่วนหนึ่งของการเชื่อมต่อที่เกิดขึ้นแล้ว
- RELATED แพ็กเก็ตของการเชื่อมต่อนี้เกี่ยวข้องกับการเชื่อมต่ออื่นที่เกิดขึ้นแล้ว
- NEW แพ็กเก็ตนี้แสดงถึงการสร้างการเชื่อมต่อใหม่
- ESTABLISHED AND REPLY แพ็กเก็ตเป็นส่วนหนึ่งของการเชื่อมต่อที่เกิดขึ้นแล้วและอยู่ในทิศทางตอบกลับ
- RELATED AND REPLY แพ็กเก็ตของการเชื่อมต่อนี้เกี่ยวข้องกับการเชื่อมต่ออื่นที่เกิดขึ้นแล้ว และอยู่ในทิศทางตอบกลับ

สำหรับชื่อของ module ที่ต้องลงทะเบียนเพื่อใช้งานคือ ip_conntrack module (connection tracking)

2.3.2 Routing System

ระบบการเลือกใช้เส้นทางจะเกิดขึ้นที่ ROUTING ของ Netfilter การเลือกใช้ Routing table สำหรับแต่ละแพ็กเก็ตจะถูกควบคุมโดย Routing rules

สำหรับการบังคับให้แพ็กเก็ตถูกส่งต่อไปตามการตัดสินใจใช้ Routing table ใดๆ นั้นจะใช้การ tag ที่เรียกว่า fwmark ของ iptables เพื่อระบุไว้ที่แพ็กเก็ตที่ถูกจัดการโดย Routing table เรียบร้อยแล้ว

2.3.3 NAT Function

NAT code จะถูกเรียกใช้ที่ POSTROUTING chain สำหรับชื่อของ module ที่ต้องลงทะเบียนเพื่อใช้ NAT code คือ iptable_nat module

สำหรับกรณีของ LBS-IA เพื่อให้สามารถแปลง IP address ของแพ็กเก็ตเป็น IP address ของช่องทางการเชื่อมต่อที่จะใช้เพื่อนำส่งแพ็กเก็ตออกสู่อินเทอร์เน็ต สำหรับการแปลง IP address โดยใช้ module ของ iptables นั้นสามารถทำได้สองแนวทางขึ้นอยู่กับว่า public IP address ถูกกำหนดแบบ static หรือ dynamic ในกรณีของ static IP address จะใช้ฟังก์ชันที่เรียกว่า SNAT (source NAT) เพื่อแปลงกลุ่มของ private IP address ให้เป็น public IP address ที่กำหนดไว้บนช่องทางการเชื่อมต่อ ในกรณีของ public IP address จะใช้ฟังก์ชันที่เรียกว่า MASQUERADE เพื่อแปลงกลุ่มของ private IP address ให้เป็น public IP address ของช่องทางการเชื่อมต่อที่จะเลือกใช้

2.3.4 Stateful Protocol

บางแอ็พพลิเคชันที่ต้องสร้างหลายการเชื่อมต่อเพื่อให้สามารถใช้บริการที่อยู่บนอินเทอร์เน็ตได้ อย่างเช่น FTP นั้น จำเป็นต้องมีการเขียนโปรแกรมเพื่อจัดการแอ็พพลิเคชันลักษณะดังกล่าวเพิ่มเติมโดยใช้หลักการที่เรียกกว่า “Protocol Helper” ซึ่งมีวัตถุประสงค์ให้ connection tracking module สามารถเข้าใจคุณลักษณะของแอ็พพลิเคชันที่ใช้หลายการเชื่อมต่อและทำให้สามารถส่งการเชื่อมต่อออกสู่ช่องทางการเชื่อมต่อเดียวกันได้จนสิ้นสุดการใช้งาน

Protocol Helper ประกอบด้วยสองส่วนคือ Connection Tracking Helper Module และ NAT Helper Module

Connection Tracking Helper Module ช่วยระบุชนิดของแอ็พลิเคชันที่ Netfilter ต้องติดตามเป็นพิเศษเพื่อให้สามารถตรวจพบและเรียกใช้การกระทำ เช่น Ip_conntrack_expect_related() ที่เกี่ยวข้องกับแอ็พลิเคชันนั้นๆ ได้อย่างเหมาะสม

NAT Helper Module ช่วยเสริมให้ NAT code สามารถเข้าไปจัดการภายในแพ็กเก็ตเพื่อเปลี่ยน IP address ให้สอดคล้องกับการเชื่อมต่อที่เกี่ยวข้องกันได้ เพื่อให้นำส่งแพ็กเก็ตของแอ็พพลิเคชันเดียวกันออกสู่ช่องทางการเชื่อมต่อเดียวกัน

การเขียนโปรแกรมดังกล่าวอาศัยโครงสร้างที่มีอยู่แล้วภายใน Netfilter เพื่อเรียกใช้ฟังก์ชันต่างๆ ทั้งนี้แต่ละแอ็พพลิเคชันต้อง Protocol Helper ที่แตกต่างกันน

สำหรับ FTP นั้น สามารถเรียกใช้ Module ที่มีอยู่แล้วเพื่อจัดการกับประเด็นที่กล่าวมานี้ได้จากภายในโครงสร้างของ Netfilter เอง module ดังกล่าวคือ ip_conntrack_ftp และ ip_nat_ftp

2.3.5 Gateway Detection

การเฝ้าตรวจความพร้อมใช้งานของช่องทางการเชื่อมต่อจะอาศัยความสามารถของ kernel ที่สามารถกำหนดได้เองว่าช่องทางการเชื่อมต่อใดสามารถนำมาใช้งานได้ ทั้งนี้ต้องกระทำบนพื้นฐานของหลักการใช้งานช่องทางการเชื่อมต่อของ kernel นั่นคือ kernel จะสามารถรับทราบสถานะของช่องทางการเชื่อมต่อได้ก็ต่อเมื่อมีการใช้งานช่องทางนั้นๆ

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 โครงสร้างพื้นฐานของระบบ

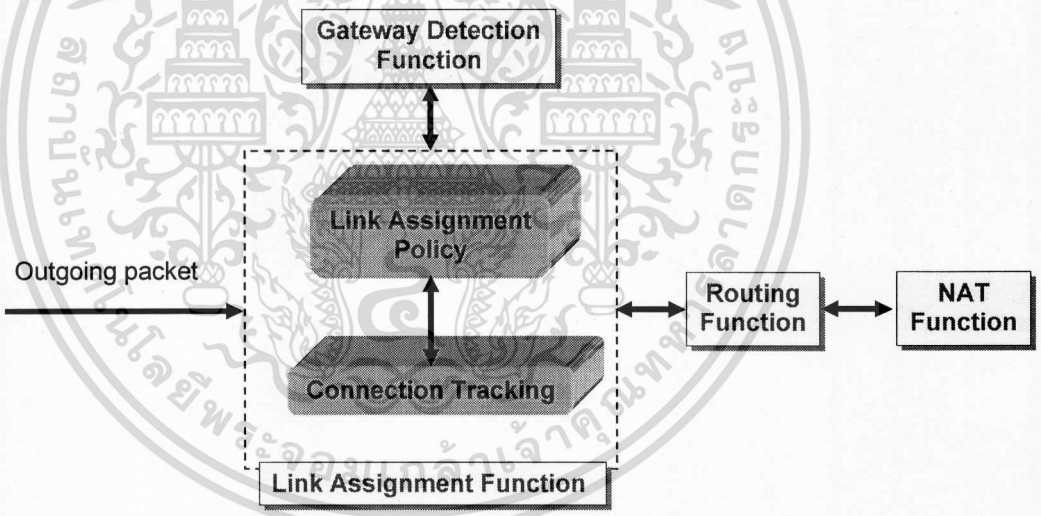
โครงสร้างพื้นฐานของระบบประกอบด้วยชุดของกฎคำสั่งของ iptables และ Routing Policy สำหรับการคัดกรองและจัดการแพ็กเก็ตที่ระบบรับเข้ามาเพื่อประสานให้องค์ประกอบหลักทั้งสามส่วนที่กล่าวไว้ในบทที่ 2 สามารถทำงานร่วมกันได้ สำหรับชุดของกฎคำสั่งดังกล่าวสามารถแบ่งได้เป็น 4 ชุด ดังนี้

1) Load Balancing Rule

2) Link Availability Rule

3) NAT Rule

4) Routing Rule



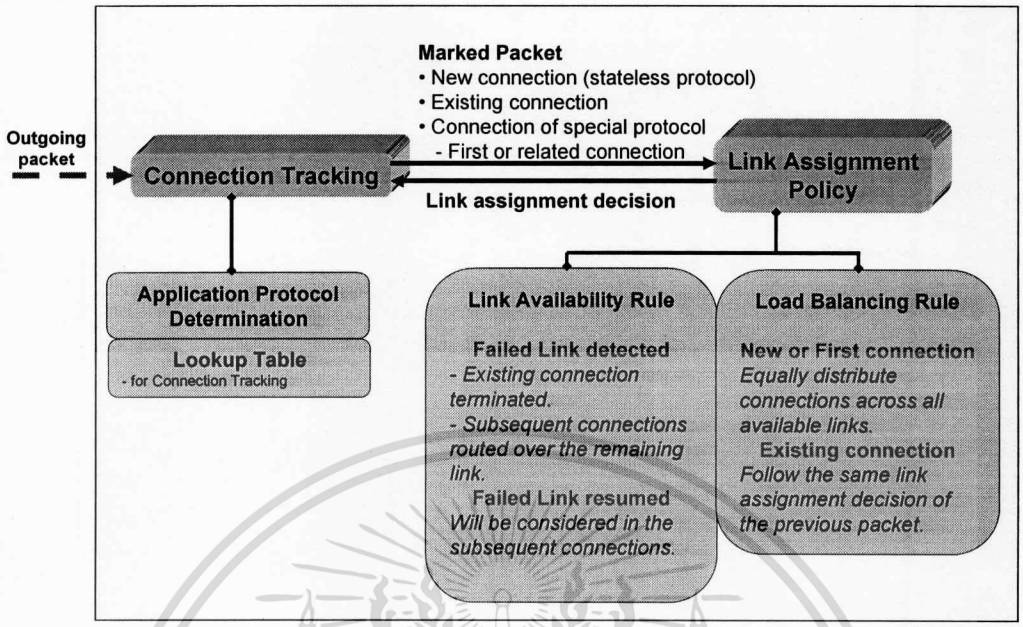
รูปที่ 3.1 โครงสร้างพื้นฐานของระบบ

การสร้างความสมดุลของ load บนช่องทางการเชื่อมต่อนั้นใช้เทคนิคการกระจายจำนวนการเชื่อมต่อที่เกิดขึ้นให้สมดุลกันบนช่องทางการเชื่อมต่ออินเทอร์เน็ตที่มีอยู่ ในกรณีของ LBS-IA นี้ประกอบด้วยช่องทางการเชื่อมต่ออินเทอร์เน็ตจำนวน 2 ช่องทาง ดังนั้น 50% ของการเชื่อมต่อทั้งหมดจะถูกนำส่งไปยังช่องทางการเชื่อมต่อหนึ่ง และที่เหลือจะถูกนำส่งไปยังช่องทางการเชื่อมต่อที่เหลือ

ในภาพรวมนั้นองค์ประกอบหลักทั้ง 3 ของ LBS-IA ที่แสดงให้เห็นในรูปที่ 3.1 นั้น มีหน้าที่ดังต่อไปนี้

- Link Assignment Function เป็นส่วนแรกที่รับแพ็กเก็ตของเครือข่ายเข้าสู่ระบบ เพื่อกำหนดช่องทางการเชื่อมต่อให้กับแพ็กเก็ตด้วยการใช้ Load Balancing Rule ที่กำหนดไว้ภายในระบบ โดยมี Connection Tracking ซึ่งเป็นโมดูลแรกที่ทำหน้าที่ตรวจสอบแพ็กเก็ตที่ส่งมาจากภายในเครือข่ายขององค์กร โดยมีหลักการทำงานดังนี้
    - ตรวจสอบสถานะของแพ็กเก็ตว่าเป็นแพ็กเก็ตแรกสำหรับการเชื่อมต่อหรือไม่
    - ตรวจสอบชนิดของ Application Protocol ของแพ็กเก็ตว่าเป็นแบบ Stateful หรือ Stateless สำหรับระบบต้นแบบนี้จะมุ่งเน้นการตรวจสอบ Stateful Protocol ชนิด FTP เพียงอย่างเดียว
    - สำหรับแพ็กเก็ตแรกของการเชื่อมต่อจะถูกส่งไปยัง Link Assignment Policy เพื่อกำหนดช่องทางการเชื่อมต่อที่จะใช้นำส่งแพ็กเก็ตต่อไป
    - สำหรับแพ็กเก็ตที่เป็นส่วนหนึ่งของการเชื่อมต่อเดิมจะถูกนำส่งออกไปยังช่องทางการเชื่อมต่อเดียวกันกับแพ็กเก็ตก่อนหน้านี้โดยอัตโนมัติโดยใช้กลไกที่มีอยู่แล้วของ Connection Tracking
    - ผลของการนำส่งแพ็กเก็ตใหม่ไปยังช่องทางการเชื่อมต่อจะถูกบันทึกเป็นสถานะของแพ็กเก็ตไว้ใน Connection Tracking Cache เพื่อใช้ในการตรวจสอบสถานะของแพ็กเก็ตต่อไปโดยที่แพ็กเก็ตแรกของการเชื่อมต่อใหม่จะถูกคัดกรองเพื่อกำหนดเครื่องหมายด้วย fwmark ให้กับแพ็กเก็ตที่บ่งชี้ถึง Routing Table ที่ต้องใช้กับแพ็กเก็ต และยังทำหน้าที่ติดตามการเลือกใช้ช่องทางการเชื่อมต่อกับแพ็กเก็ตผ่าน Connection Tracking เพื่อให้แน่ใจว่าแพ็กเก็ตที่เป็นของการเชื่อมต่อเดียวกันสามารถถูกส่งออกผ่านช่องทางการเชื่อมต่อเดียวกันเสมอ

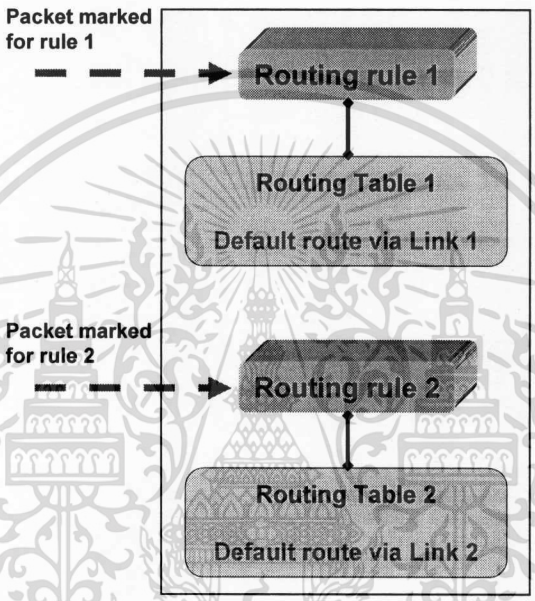
  ในกรณีที่เกิดความล้มเหลวบนช่องทางการเชื่อมต่อหนึ่งที่ทำให้ไม่สามารถนำมาใช้กับ Load Balancing Rule ได้นั้น ระบบจะเรียกใช้ Link Availability Rule เพื่อแก้ปัญหาให้สามารถนำส่งการเชื่อมต่อที่เกิดขึ้นผ่านช่องทางการเชื่อมต่อที่พร้อมใช้งาน ในรูปที่ 3.2 แสดงให้เห็นกระบวนการทำงานภายในของ Link Availability Rule ประกอบด้วยชุดของกฎสำหรับการเลือกใช้ช่องทางการเชื่อมต่อที่พร้อมใช้งานในกรณีที่ kernel ตรวจพบความล้มเหลวของช่องทางการเชื่อมต่อหนึ่ง



รูปที่ 3.2 กระบวนการทำงานของ Link Assignment Function

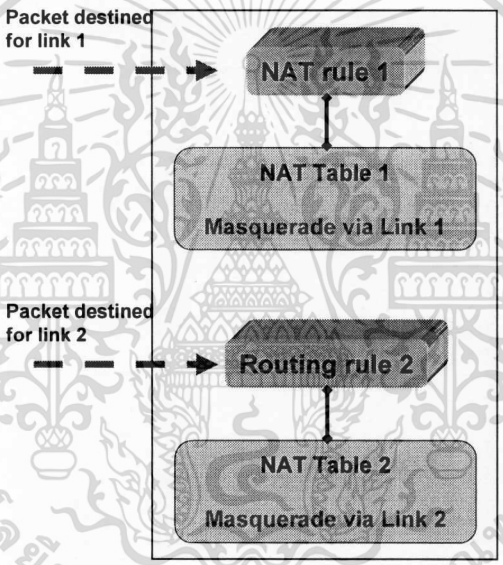
- Gateway Detection Function ทำหน้าที่ในการตรวจสอบความพร้อมใช้งานของช่องทางการเชื่อมต่อที่มีอยู่โดยใช้ความสามารถของ kernel ร่วมกับการประยุกต์ใช้ Link Availability Rule ใน Link Assignment Function เพื่อส่งสถานะที่ตรวจสอบได้ให้กับ Link Assignment Function เพื่อใช้เป็นข้อมูลในการพิจารณาเลือกช่องทางการเชื่อมต่อที่พร้อมใช้งานสำหรับแพ็กเก็ตต่อไป

- Routing Function เป็นส่วนที่รับแพ็กเก็ตที่ถูกเลือกช่องทางการเชื่อมต่อแล้วโดย Link Assignment Function เพื่อเลือก Routing Table ที่เหมาะสมสำหรับแพ็กเก็ตดังกล่าวในการนำส่งแพ็กเก็ตออกสู่ช่องทางการเชื่อมต่อที่เลือกไว้ต่อไป ภายในประกอบด้วย Routing Rule จำนวน 2 ชุด ที่ใช้คัดกรองแพ็กเก็ตที่มี fwmark ที่แตกต่างกัน และ Routing Table จำนวน 2 ชุดสำหรับบังคับใช้กับแพ็กเก็ตที่ผ่านการจับคู่กับ Routing Table ซึ่งแสดงให้เห็นในรูปที่ 3.3



รูปที่ 3.3 กระบวนการทำงานของ Routing Function

- NAT Function ทำหน้าที่นำส่งแพ็กเก็ตออกสู่ช่องทางการเชื่อมต่อที่ได้เลือกไว้แล้วตามการตัดสินใจของ Routing Function โดยแปลง Source IP address ของแพ็กเก็ตให้เป็น IP address ของช่องทางการเชื่อมต่อที่เลือกไว้ ภายในประกอบด้วย NAT Rule จำนวน 2 ชุดที่ใช้แยกกันสำหรับแต่ละช่องทางการเชื่อมต่อ สำหรับการแปลง IP address ของแพ็กเก็ตใช้วิธี MASQUERADE ซึ่งทำให้แพ็กเก็ตถูกแปลง IP address ให้เป็น IP address ของช่องทางการเชื่อมต่อที่เลือกใช้ได้โดยอัตโนมัติโดยไม่ขึ้นอยู่กับว่ามีการเปลี่ยน IP address ของช่องทางการเชื่อมต่อนั้นถูกกำหนดแบบ static หรือ dynamic ยิ่งไปกว่านั้นการทำ NAT เช่นนี้ทำให้ทุกๆ แพ็กเก็ตที่อยู่ภายใต้การเชื่อมต่อเดียวกันได้สืบทอดการแปลง IP address เดียวกันกับแพ็กเก็ตแรกของการเชื่อมต่อเสมอ



รูปที่ 3.4 กระบวนการทำงานของ NAT Function

3.2 องค์ประกอบหลักภายในชุดของกฎคำสั่ง iptables

3.2.1 Table

สำหรับ Table ที่นำมาใช้ในการคัดเลือกแพ็กเก็ตนั้นประกอบด้วย

- NAT table สำหรับเป็นโครงสร้างข้อมูลการแปลง source IP address ของแพ็กเก็ตแบบ MASQUERADE ก่อนที่จะนำส่งแพ็กเก็ตออกสู่ช่องทางการเชื่อมต่อที่เลือกไว้
- Mangle table สำหรับเป็นโครงสร้างข้อมูลสำหรับการระบุเครื่องหมาย fwmark ให้กับแพ็กเก็ตที่ผ่านการตรวจสอบแล้ว เพื่อวัตถุประสงค์ของการกระจายการเชื่อมต่อให้สมดุลกันบนช่องทางการเชื่อมต่อที่มีอยู่

3.2.2 Chain

สำหรับ chain หลักที่เกี่ยวข้องกับกระบวนการภายในของ LBS-IA เพื่อสร้างความสมดุลของ load ที่เกิดขึ้นจากเครือข่ายภายในองค์กรนั้น ประกอบด้วย 4 ส่วนคือ

- PREROUTING chain ใช้เป็นจุดตรวจสอบสถานะการเชื่อมต่อของแพ็กเก็ตที่รับมาจากเครือข่ายภายในองค์กร และการตัดสินใจเลือกช่องทางการเชื่อมต่อสำหรับแพ็กเก็ตภายใต้โครงสร้างข้อมูลของ Mangle table
- NEW_OUT_CONN chain เป็นจุดที่ถูกสร้างขึ้นเพิ่มเติมเพื่อใช้งานเฉพาะกับ LBS-IA นี้ สำหรับ chain นี้จะเป็นจุดที่รับแพ็กเก็ตแรกของการเชื่อมต่อเข้ามาเพื่อระบุเครื่องหมายให้กับแพ็กเก็ตที่ผ่านการตรวจสอบแล้วภายใต้ Mangle table และนำส่งต่อไปยัง FORWARD chain ใช้ในการเลือกช่องทางการเชื่อมต่อสำหรับแพ็กเก็ตตามเครื่องหมายไว้แพ็กเก็ต
- POSTROUTING chain เป็นจุดสุดท้ายที่แพ็กเก็ตต้องผ่านในโครงสร้างของ Netfilter เพื่อการแปลง Source IP address ของแพ็กเก็ตแบบ MASQUERADE ให้เปลี่ยนไปใช้ IP address ของช่องทางการเชื่อมต่อที่เลือกไว้สำหรับแพ็กเก็ตนั้นๆ ภายใต้โครงสร้างข้อมูลของ NAT table

3.2.3 Random Match

การจับคู่แพ็กเก็ตที่ผ่านเข้ามาใน NEW_OUT_CCONN chain กับกฎการทำเครื่องหมายแพ็กเก็ตให้มีอัตราส่วนเป็น 50:50 สำหรับกฎการทำเครื่องหมายที่มีอยู่ 2 กฎ เพื่อใช้ในการกระจายแพ็กเก็ตไปยังช่องทางการเชื่อมต่อที่มีอยู่ให้สมดุลกัน

3.2.4 MARK Target

สำหรับ Target เป็นส่วนหนึ่งของกฎที่ระบุการกระทำที่จะต้องเกิดกับแพ็กเก็ตหลังจากที่แพ็กเก็ตนั้นผ่านการตรวจสอบด้วยกฎนั้นๆ ชนิดของ Target ที่เป็นปัจจัยหลักสนับสนุนการทำงานของ LBS-IA คือ MARK target เป็นการระบุการทำเครื่องหมาย fwmark ให้แพ็กเก็ตแรกของการเชื่อมต่อที่ผ่านเข้ามาตรวจสอบภายใน NEW_OUT_CONN chain

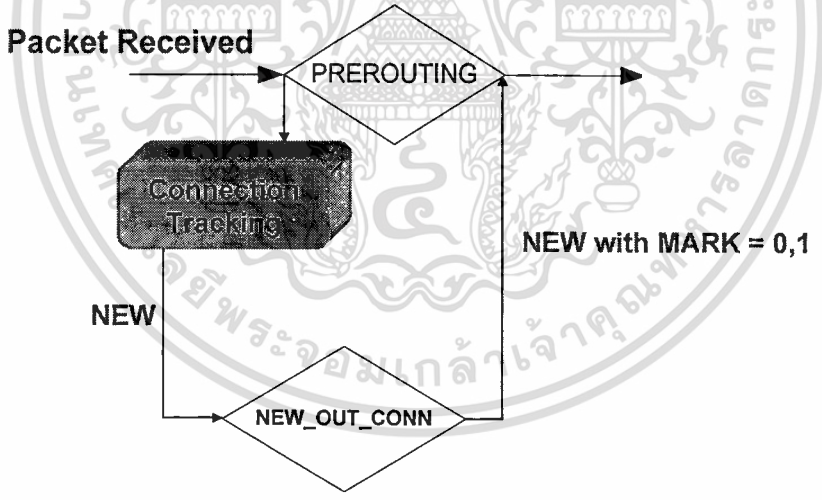
3.3 แนวทางการออกแบบชุดของกฎคำสั่ง

3.3.1 Load Balancing Rule

มีขั้นตอนการทำงานตามลำดับดังนี้

- ที่ PREROUTING chain สำหรับแพ็กเก็ตแรกของการเชื่อมต่อซึ่งสามารถระบุได้โดย Connection Tracking ให้ถูกส่งไปผ่านกระบวนการเลือกช่องทางการเชื่อมต่อใน NEW_OUT_CONN chain ให้เสร็จสิ้นก่อนส่งกลับมา
- ที่ NEW_OUT_CONN chain แพ็กเก็ตทั้งหมดต้องผ่านกระบวนการต่อไปนี้
    - 50% ของแพ็กเก็ตที่ถูกกำหนดเครื่องหมาย fwmark เป็นหมายเลข 0 แล้วจะถูกส่งกลับไปยัง PREROUTING chain
    - 50% ของแพ็กเก็ตที่ถูกกำหนดเครื่องหมาย fwmark เป็นหมายเลข 1 แล้วจะถูกส่งกลับไปยัง PREROUTING chain

ที่ PREROUTING chain แพ็กเก็ตที่มีเครื่องหมาย fwmark เป็นหมายเลข 0 ถูกส่งผ่านช่องทางการเชื่อมต่อที่ 1 และแพ็กเก็ตที่มีเครื่องหมาย fwmark เป็นหมายเลข 1 ถูกส่งผ่านช่องทางการเชื่อมต่อที่ 2 ภาพรวมของกฎแสดงให้เห็นในรูปที่ 3.5



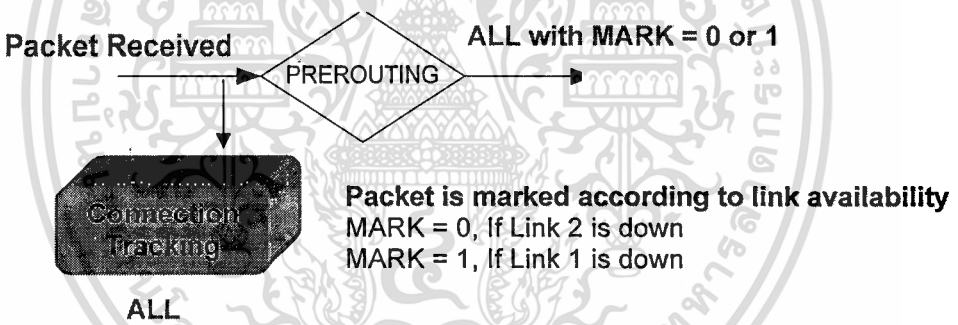
รูปที่ 3.5 การออกแบบ Load Balancing Rule

3.3.2 Link Availability Rule

กฎคำสั่งนี้จะถูกสั่งให้ทำงานไปพร้อมๆ กับกลไกการตรวจสอบความพร้อมใช้งานของ kernel เพื่อปรับเปลี่ยนพฤติกรรมการเลือกใช้ช่องทางการเชื่อมต่อของ Load Balancing Rule โดยมีขั้นตอนการทำงานตามลำดับดังนี้

- ถ้า kernel ตรวจพบความไม่พร้อมใช้งานในช่องทางการเชื่อมต่อใดๆ จะมีการระงับการใช้กฎเพื่อการส่งแพ็กเก็ตไปยังช่องทางการเชื่อมต่อนั้นๆ ใน PREROUTING chain โดยการทำเครื่องหมาย fwmark เป็นหมายเลขที่สอดคล้องกับช่องทางการเชื่อมต่อที่พร้อมใช้งานสำหรับทุกๆแพ็กเก็ต
- ถ้า kernel ตรวจพบความพร้อมใช้งานในช่องทางการเชื่อมต่อใดๆ จะมีการเรียกใช้ Load Balancing Rule ให้กลับคืนมาเหมือนเดิม

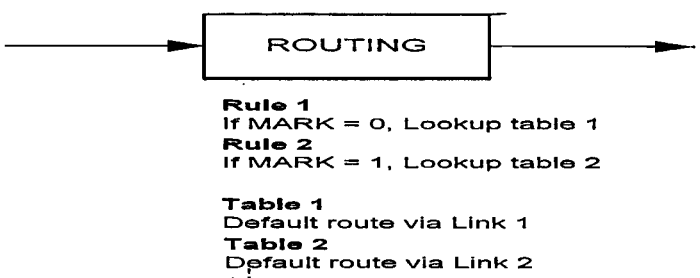
ภาพรวมของกฎแสดงให้เห็นในรูปที่ 3.6



รูปที่ 3.6 การออกแบบ Link Availability Rule

3.3.3 Routing Rule

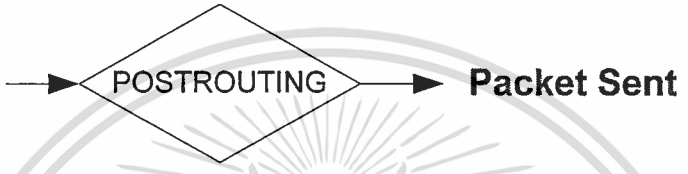
สำหรับ Routing Function ของระบบจะประกอบด้วยกฎที่ใช้คัดกรองแพ็กเก็ตที่ถูกทำเครื่องหมาย fwmark โดย Load Balancing Rule หรือ Link Availability Rule เพื่อเลือกใช้ Routing Table ที่สอดคล้องกับ fwmark ของแพ็กเก็ต แต่ละ Routing table จะประกอบด้วย Default Route สำหรับใช้กับแต่ละช่องทางการเชื่อมต่อที่ระบบมีใช้งาน ซึ่งทั้งหมดแสดงให้เห็นในรูปที่ 3.7



รูปที่ 3.7 การออกแบบ Routing Rule

3.3.4 NAT Rule

ที่ POSTROUTING chain แพ็กเก็ตที่ถูกกำหนดให้ออกสู่ช่องทางการการเชื่อมต่อที่ 1 ต้องถูกทำ MASQUERADE ให้ใช้ IP address ของช่องทางการเชื่อมต่อที่ 1 ก่อนส่งแพ็กเก็ตออกจาก LBS-IA และเช่นเดียวกันกับแพ็กเก็ตที่ถูกกำหนดให้ออกสู่ช่องทางการเชื่อมต่อที่ 2 ต้องถูกทำ MASQUERADE ให้ใช้ IP address ของช่องทางการเชื่อมต่อที่ 2 ก่อนส่งแพ็กเก็ตออกจาก LBS-IA เช่นกัน ดังแสดงให้เห็นในรูปที่ 3.8



Rule 1
If destined for Link 1, MASQUERADE using IP Address of Link 1's interface
Rule 2
If destined for Link 2, MASQUERADE using IP Address of Link 2's interface

รูปที่ 3.8 การออกแบบ NAT Rule

บทที่ 4
การประยุกต์ใช้งานเทคโนโลยี Linux ในระบบ

4.1 รายละเอียดของระบบที่นำมาประยุกต์ใช้งาน

1) เครื่องคอมพิวเตอร์ส่วนบุคคลแบบ laptop สำหรับจำลองการทำงานของระบบทั้งหมด มีคุณลักษณะหลักดังนี้
   - CPU: Intel Centrino Mobile Technology 1.6 GHz
   - RAM: 760 MB
   - Hard Disk: 60 GB
2) VMware Workstation 5.0 build-13124 สำหรับการจำลองระบบภายในเครื่องคอมพิวเตอร์ส่วนบุคคลแบบ laptop
3) ระบบปฏิบัติการ Red Hat Linux version 9 พร้อมกับ kernel version 2.4.20-8 ใช้เป็น Management Workstation และ LBS-IA
4) ความต้องการระบบพื้นฐานของ Management Workstation
   - RedHat package ที่ประกอบด้วย
     - bind-utils, gdk-pixbuf, glib, glibc, gtk+, gtkmm, libsigc++, libstdc++, libxml2, libxslt, openssl-0.96b, ucd-snmp, X-Windows system with XFree86-libs, qt
   - Firewall Builder vesion 2.1.16

ซอฟต์แวร์สำหรับทำหน้าที่เป็น Management Workstation ถูกเลือกมาจากซอฟร์แวร์ที่ถูกพัฒนาขึ้นภายใต้ GNU General Public License โดยนักพัฒนาซอฟต์แวร์ชื่อ Mr. Vadim Kurland ซอฟต์แวร์ดังกล่าวมีชื่อว่า Firewall Builder 2.1.16 และสามารถดาวน์โหลดได้จาก

http://sourceforge.net/project/showfiles.php?group_id=5314&package_id=125361

Firewall Builder เป็นซอฟต์แวร์สำหรับจัดการและตั้งค่า policy ซึ่งมีคุณลักษณะพื้นฐานต่อไปนี้

- สนับสนุนการจัดการ policy ในรูปแบบ object-oriented approach เป็นผลให้การเปลี่ยนแปลงที่เกิดขึ้นกับวัตถุในฐานข้อมูลมีผลกับกฎต่างๆ ที่ใช้วัตถุนั้นทันที
- สนับสนุนการใช้ Policy Compiler ร่วมกับ firewall ได้หลายแบบ (multi-platform firewall) เช่น iptables, ipfilter และ OpenBSD PF เป็นต้น

- สนับสนุนการใช้ GUI สำหรับการใช้งานโดยผู้ใช้เพื่อการจัดการ policy ของ firewall เป้าหมาย

โครงสร้างพื้นฐานภายใน Firewall Builder ประกอบด้วย 3 modules หลักที่ทำงานร่วมกันคือ

- Application Programming Interface (API) สำหรับการเข้าถึงและใช้งาน objects database และ subclass อื่นๆ ภายในซอฟต์แวร์ โมดูลที่นำมาติดตั้งใช้งานชื่อ libfwbuilder-2.1.16-1.rh90.i386.rpm

- Policy Compiler สำหรับการแปลง policy ให้อยู่ในรูปแบบของ configuration file หรือ script สำหรับ firewall เป้าหมาย โมดูลที่นำมาติดตั้งใช้งานชื่อ fwbuilder-ipt-2.1.16-1.rh90.1386.rpm

- GUI สำหรับใช้เป็นเครื่องมือในการสร้าง policy ในรูปแบบ graphic เพื่อให้ผู้ใช้สามารถเปิดดู สร้าง ปรับแต่ง ลบ ชุดของ object ได้ โมดูลที่นำมาติดตั้งใช้งานชื่อ fwbuilder-2.1.16-1.rh90.i386.rpm

5) ความต้องการระบบพื้นฐานของ LBS-IA

- Kernel Compilation Software
  - GNU C 2.95.3
  - GNU make 3.77
  - binutils 2.9.1.0.25
  - Kernel source 2.4.20-8
- File Compression Software
  - GNU tar 1.13.25
  - bzip2 1.0.
- OpenSSH_3.5 p1
- iptables modules
  - iptables verion 1.2.9
  - Patch-O-Matic 20031219
    - Random match module
    - MARK target module
- Vi text editor

4.2 การติดตั้งระบบ

กระบวนการติดตั้งระบบเพื่อประยุกต์ใช้งานเทคโนโลยี Linux นั้นประกอบด้วย 3 ส่วนหลัก คือ การติดตั้ง Management Workstation การติดตั้ง LBS-IA และการจัดการ iptables script ด้วย Management Workstation ดังต่อไปนี้

4.2.1 การติดตั้ง Management Workstation

การติดตั้ง Firewall Builder สำหรับทำเป็น Management Workstation นั้น กระทำผ่านเครื่องมือการติดตั้งซอฟต์แวร์ของ Red Hat ที่เรียกว่า Red Hat Package Management

สำหรับ Read Hat Linux 9 นี้ประกอบด้วย RedHat package พื้นฐานส่วนหนึ่งสำหรับการติดตั้ง Firewall Builder อยู่แล้ว ยกเว้น ucd-snmp ฉะนั้นก่อนการติดตั้งชุดโมดูลของ Firewall Builder จึงจำเป็นต้องติดตั้ง package ดังกล่าวก่อน ซึ่งกระทำด้วยคำสั่งดังต่อไปนี้

- rpm –i net-snmp-5.0.6-17.i386.rpm

- rpm –i net-snmp-utils-5.0.6-17.i386.rpm

จากนั้นติดตั้งชุดโมดูลของ Firewall Builder ซึ่งประกอบด้วย API, GUI และ Policy compiler for iptables ตามลำดับดังนี้

- rpm –i libfwbuilder-2.1.16-1.rh90.i386.rpm

- rpm –i fwbuilder-2.1.16-1.rh90.i386.rpm

- rpm –i fwbuilder-ipt-2.1.16-1.rh90.1386.rpm

4.2.2 การติดตั้ง LBS-IA

ในภาพรวมของการติดตั้ง LBS-IA เพื่อให้พร้อมทำงานร่วมกับ Management Workstation และประยุกต์ใช้ชุดของกฎคำสั่งที่ออกแบบไว้ในบทที่ 3 นั้นแบ่งออกเป็น 3 ส่วนหลักดังนี้

4.2.2.1 การติดตั้ง iptables modules

กระบวนการติดตั้ง iptables modules นี้กระทำเพื่อลงทะเบียนโครงสร้างข้อมูลของ iptables version 1.2.9 ซึ่งประกอบด้วย NAT table และ Mangle Table พร้อมกับโมดูลเสริมที่จะประยุกต์ใช้ตามวัตถุประสงค์ของระบบ โดยมีลำดับการติดตั้งดังนี้

1) คัดลอก source file ของ iptables version 1.2.9 ไปยังโฟลเดอร์ /usr/src/linux-2.4

- คัดลอก source file ของ iptables version 1.2.9 จาก CD-ROM ไปยังโฟลเดอร์ /usr/src/linux-2.4 และทำการ extract ไฟล์ออกมา ด้วยคำสั่งตามลำดับต่อไปนี้

- cp iptables-1.2.9.tar.bz2 /usr/src/linux-2.4

- tar –xjvf iptables-1.2.9.tar.bz2

(คำสั่งนี้ต้องกระทำภายใต้โฟลเดอร์ /usr/src/liniux-2.4)

2) ติดตั้ง Random match module และ MARK target module เข้ากับโครงสร้างข้อมูลของ iptables 1.2.9 (สำหรับ MARK target module ได้ถูกติดตั้งมาพร้อมกับ Kernel เรียบร้อยแล้ว)

- คัดลอก source file ของ Patch-O-Matic 20030107 จาก CD-ROM ไปยังโฟลเดอร์ /usr/src/linux-2.4 และทำการ extract ไฟล์ออกมา ด้วยคำสั่งตามลำดับต่อไปนี้

  - cp patch-o-matic-20031219.tar.bz2 /usr/src/linux-2.4

  - tar –xjvf patch-o-matic-20031219.tar.bz2

  (คำสั่งนี้ต้องกระทำภายใต้โฟลเดอร์ /usr/src/liniux-2.4)

- ภายในโฟลเดอร์ /usr/src/linux-2.4/patch-o-matic ติดตั้ง Random match module ด้วยคำสั่งตามลำดับต่อไปนี้

  - KERNEL_DIR=/usr/src/linux-2.4

  IPTABLES_DIR=/usr/src/linux-2.4/iptables-1.2.9 sh runme

  base/random.patch

3) ติดตั้ง iptables 1.2.9 เข้าสู่ kernel

- ภายในโฟลเดอร์ /usr/src/linux-2.4/iptables-1.2.8 ดำเนิการ compile ไฟล์ข้อมูลของ iptables เข้าสู่โฟลเดอร์ที่เก็บ source file ของ kernel ด้วยคำสั่ง

  - make KERNEL_DIR=/usr/src/linux-2.4

  ติดตั้ง shared library และ binaries ของ iptables เข้าสู่ source file ของ kernel ด้วยคำสั่ง

  - make install KERNEL_DIR=/usr/src/linux-2.4

4.2.2.2 การ compile kernel

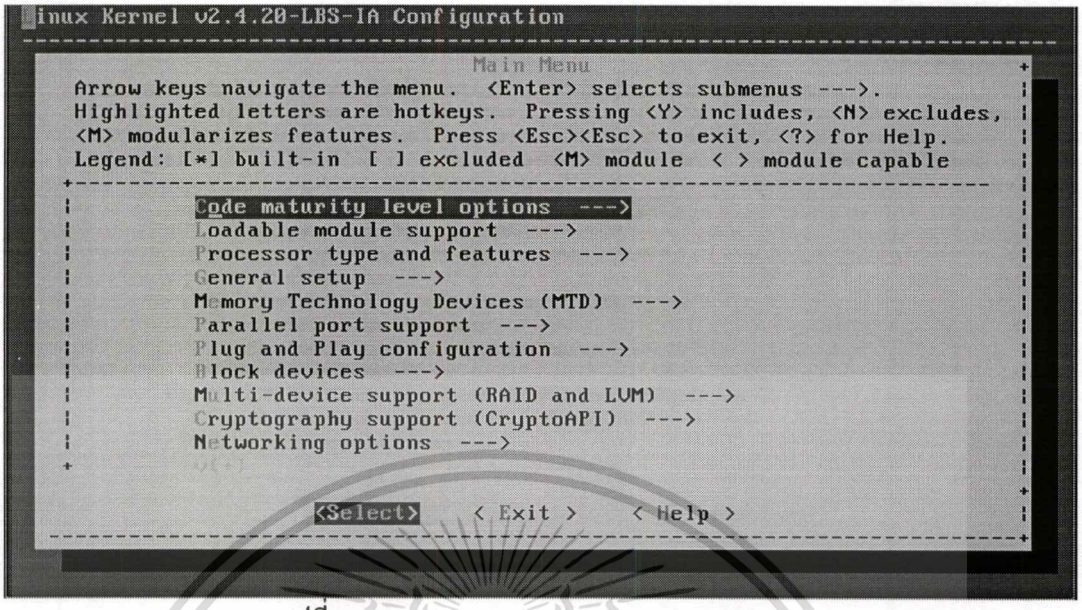
เพื่อสร้าง LBS-IA ที่ประกอบด้วยโมดูลต่างๆที่เกี่ยวข้องกับการจัดการกระแสข้อมูลให้เป็นไปตามวัตถุประสงค์ของระบบ จำเป็นต้องปรับแต่ง option ภายในของ Kernel ให้ตรงกับการใช้งานจริงรวมทั้งการลงทะเบียนโมดูลของ iptables ที่จำเป็นเข้าสู่โครงสร้างของ Netfilter ด้วยการสร้าง kernel เฉพาะสำหรับ LBS-IA ตามขั้นตอนต่อไปนี้ ซึ่งถูกกระทำภายในโฟลเดอร์ /usr/src/linux-2.4 ทั้งหมด

1) ปรับแต่ง Kernel options เพื่อเลือกใช้โมดูลที่เกี่ยวข้องกับ LBS-IA

- แสดงรายการของ Kernel options ด้วยคำสั่ง

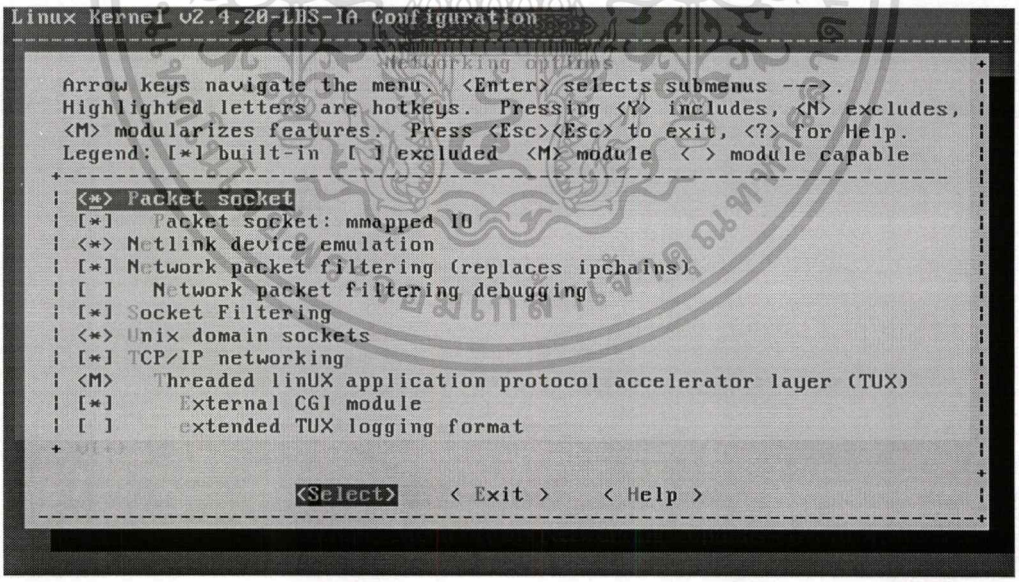
  - make menuconfig

หน้าหลักของ Kernel options ปรากฏให้เห็นตามรูปที่ 4.1



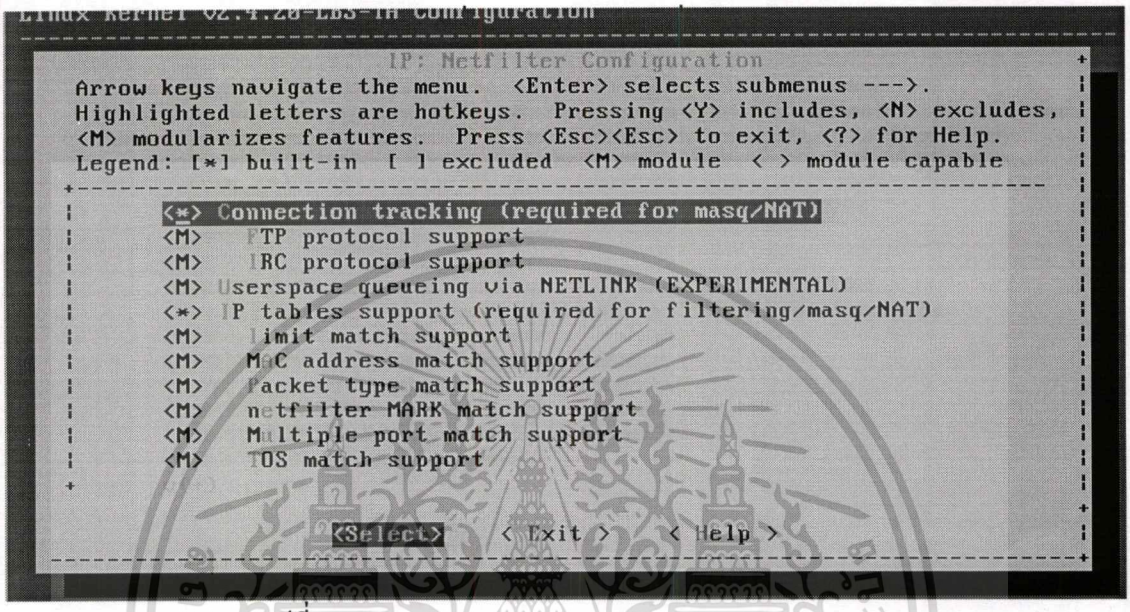
รูปที่ 4.1 Kernel Configuration – Main Menu

- ภายใน Main Menu เลือก Networking options เพื่อเลือกติดตั้งโมดูลที่เกี่ยวข้องกับการใช้งาน iptables หน้าจอภายใต้ Networking options แสดงให้เห็นในรูปที่ 4.2



รูปที่ 4.2 Kernel Configuration – Networking options

รายการโมดูลที่ต้องเลือกภายใน Networking options นี้ประกอบด้วย Packet socket, Network packet filtering (replace ipchains), Socket filtering, Unix domain sockets, TCP/IP networking -> [IP: advanced router, IP: policy routing,], เลือก IP: Netfilter Configuration เพื่อแสดงรายการโมดูลที่เกี่ยวข้อง



รูปที่ 4.3 Kernel Configuration – Netfilter Configuration

รายการโมดูลที่ต้องเลือกภายใน Netfilter Configuration นี้ประกอบด้วย Connection Tracking ->[FTP protocol support, Connection mark tracking support], IP tables support ->[Packet type match support, netfilter mark match support, random match support, Helper match support, Connection state match support, Connection tracking match support, Full NAT support -> MASQUERADE target support

- กลับสู่ Main Menu แล้วเลือก Exit เพื่อบันทึกการเลือก options ซึ่งระบบจะบันทึกข้อมูลไว้ในไฟล์ .config โดยอัตโนมัติ

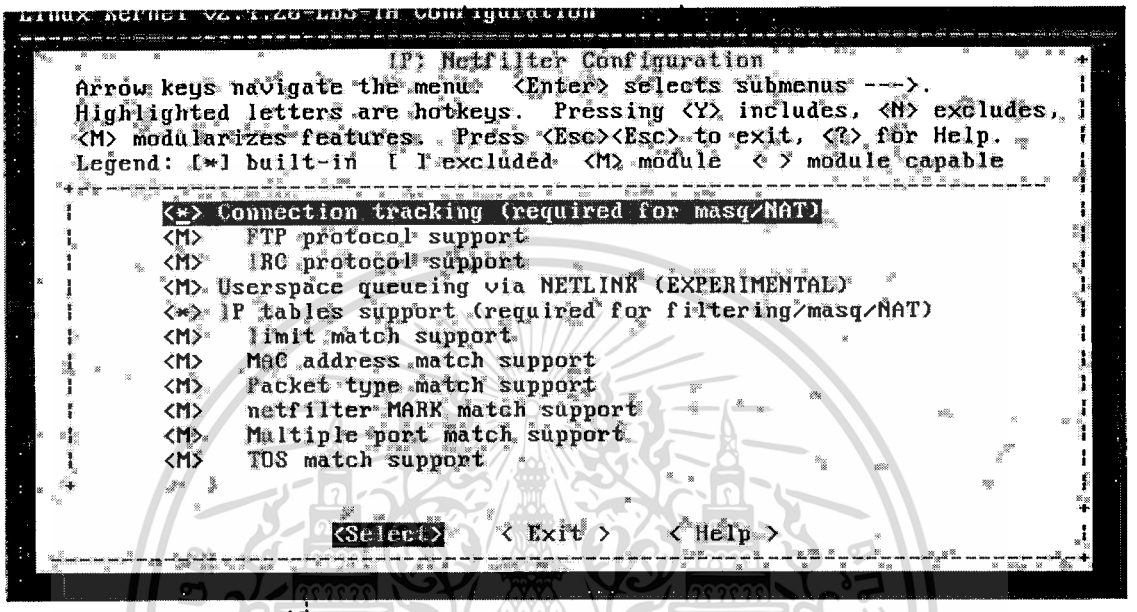
2) สร้าง Kernel ที่มีคุณสมบัติเฉพาะการใช้งานกับ LBS-IA ด้วยการ compile kernel โดยมีลำดับการกระทำดังนี้

- แยก kernel ของ LBS-IA ออกจาก kernel เดิมที่กำลังใช้งานอยู่ โดยการปรับแต่งเนื้อหาภายในไฟล์ Makefile ดังนี้

  - vi Makefile

  ที่บรรทัด EXTRAVERSION=-20custom ให้แก้เป็น EXTRAVERSION=-LBS-IA และบันทึกการเปลี่ยนแปลงที่เกิดขึ้นและออกจากไฟล์

รายการโมดูลที่ต้องเลือกภายใน Networking options นี้ประกอบด้วย Packet socket, Network packet filtering (replace ipchains), Socket filtering, Unix domain sockets, TCP/IP networking -> [IP: advanced router, IP: policy routing,], เลือก IP: Netfilter Configuration เพื่อแสดงรายการโมดูลที่เกี่ยวข้อง



รูปที่ 4.3 Kernel Configuration – Netfilter Configuration

รายการโมดูลที่ต้องเลือกภายใน Netfilter Configuration นี้ประกอบด้วย Connection Tracking ->[FTP protocol support, Connection mark tracking support], IP tables support ->[Packet type match support, netfilter mark match support, random match support, Helper match support, Connection state match support, Connection tracking match support, Full NAT support -> MASQUERADE target support

- กลับสู่ Main Menu แล้วเลือก Exit เพื่อบันทึกการเลือก options ซึ่งระบบจะบันทึกข้อมูลไว้ในไฟล์ .config โดยอัตโนมัติ

2) สร้าง Kernel ที่มีคุณสมบัติเฉพาะการใช้งานกับ LBS-IA ด้วยการ compile kernel โดยมีลำดับการกระทำดังนี้

- แยก kernel ของ LBS-IA ออกจาก kernel เดิมที่กำลังใช้งานอยู่ โดยการปรับแต่งเนื้อหาภายในไฟล์ Makefile ดังนี้

  - vi Makefile

  ที่บรรทัด EXTRAVERSION=-20custom ให้แก้เป็น EXTRAVERSION=-LBS-IA และบันทึกการเปลี่ยนแปลงที่เกิดขึ้นและออกจากไฟล์

เริ่มการ compile kernel ด้วยการเรียกใช้คำสั่งตามลำดับต่อไปนี้

- **make clean** (เตรียมโครงสร้างของ kernel source ให้พร้อมสำหรับการ compile)
- **make dep** (เตรียม module dependencies)
- **make bzImage** (สร้าง kernel image สำหรับ LBS-IA)
- **make modules** (สร้างโมดูลที่ถูกเลือกไว้)
- **make modules_install** (ติดตั้งโมดูลทั้งหมดเข้ากับ kernel)
- **make install** (ติดตั้ง kernel ใหม่พร้อมกับไฟล์ที่เกี่ยวข้องลงในโครงสร้างของ LBS-IA)

4.2.2.3 การตั้งค่าการใช้งาน iptables 1.2.9

1) กำหนดให้ใช้ executable file ของ iptables 1.2.9 เมื่อเริ่มต้นเปิดใช้งานระบบ ตามที่กำหนดไว้ใน startup script ของ iptables (/etc/rc.d/init.d/iptables) โดยการคัดลอกไฟล์ iptables, iptables-save และ iptables-restore ไปยังโฟลเดอร์ /sbin ด้วยคำสั่งต่อไปนี้

- **cp /usr/local/sbin/iptables /sbin/iptables** (executable file สำหรับ iptables service)
- **cp /usr/local/sbin/iptables-save /sbin/iptables-save** (executable file สำหรับการบันทึกกฎการทำ load balancing ที่ถูกสร้างขึ้น)
- **cp /usr/local/sbin/iptables-restore /sbin/iptables-restore** (executable file สำหรับการเรียกใช้กฎการทำ load balancing ที่สร้างไว้)

2) กำหนดให้ระบบเปิดใช้ iptables service โดยอัตโนมัติเมื่อเริ่มเปิดใช้งานระบบ ด้วยคำสั่งต่อไปนี้

- **chkconfig --level 2345 iptables on**

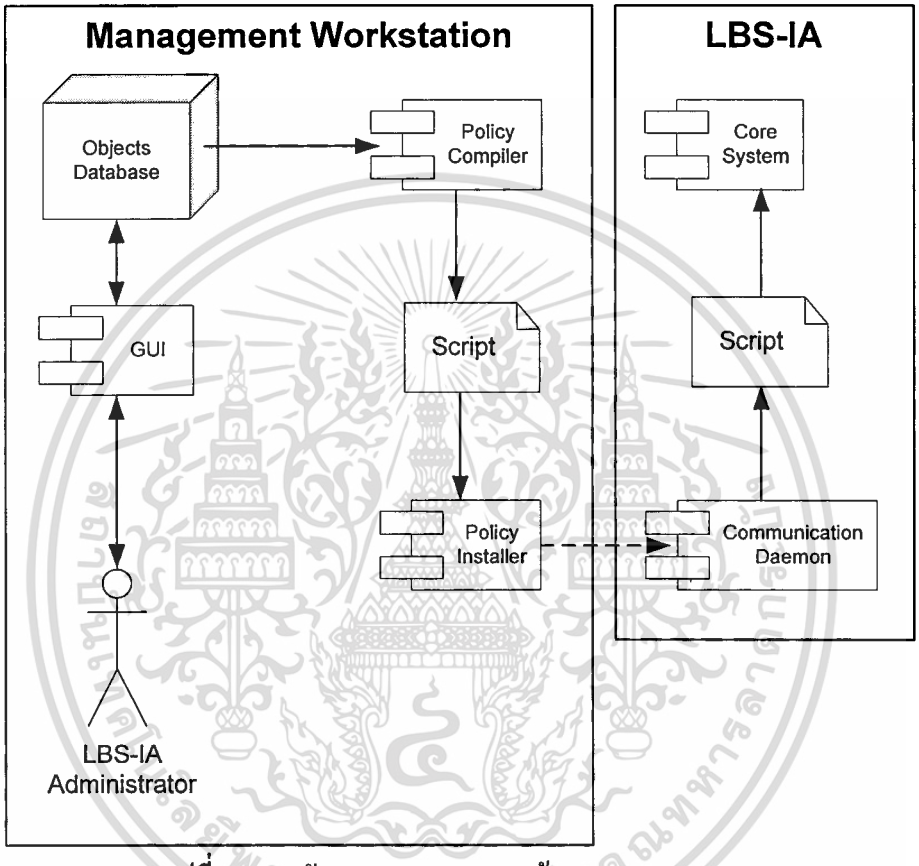
3) สร้าง iptables script เพื่อรองรับการบันทึกกฎที่จะถูกสร้างขึ้นเพื่อการทำ load balancing ด้วยคำสั่งต่อไปนี้

- **service iptables save**

  คำสั่งนี้จะสร้างไฟล์ /etc/sysconfig/iptables ขึ้นไว้เพื่อรองรับการบันทึกกฎที่จะถูกสร้างขึ้น

4.2.3 การจัดการ iptables script ด้วย Management Workstation

การจัดการ iptables script ของ LBS-IA ด้วยซอฟต์แวร์ Firewall Builder version 2.1.16 ที่ติดตั้งใช้งานใน Management Workstation สามารถกระทำผ่านช่องทางความปลอดภัยแบบ Secured shell ที่ถูกสร้างขึ้นเพื่อการติดต่อสื่อสารเฉพาะสองระบบนี้ ในรูปที่ 4.4 แสดงให้เห็นกระบวนการทำงานร่วมกันระหว่างระบบทั้งสองผ่านช่องทางการติดต่อสื่อสารดังกล่าว



รูปที่ 4.4 การจัดการ iptables script ด้วย Firewall Builder

4.2.3.1 Management Workstation

ระบบนี้ทำหน้าที่ควบคุมและจัดการการทำงานของ LBS-IA ซึ่งมีคุณลักษณะหลักดังต่อไปนี้

1) Graphical User Interface (GUI) สำหรับผู้ใช้งานระบบ

2) Object Database สำหรับจัดเก็บข้อมูลวัตถุที่ใช้สร้าง policy สำหรับ LBS-IA

3) Policy Compiler สำหรับการแปลงข้อมูลวัตถุที่ใช้สร้าง policy ให้อยู่ในรูปแบบของ script ที่ LBS-IA สามารถนำไปใช้งาน

4) Policy Installer สำหรับการติดตั้ง script บน LBS-IA

4.2.3.2 LBS-IA

ระบบนี้บังคับใช้ policy ที่ถูกสร้างโดย Management Workstation เพื่อควบคุมการจราจรของข้อมูลอินเทอร์เน็ตให้กระจายออกสู่ช่องทางการเชื่อมต่อที่มีอยู่อย่างสมดุล มีคุณลักษณะหลักดังต่อไปนี้

1) Communication daemon สำหรับติดต่อสื่อสารกับ Management Workstation เพื่อการติดตั้งใช้งาน script ที่ถูกสร้างขึ้น

2) Policy Database สำหรับจัดเก็บ script ที่ถูกติดตั้งใช้งาน

3) Core system สำหรับตรวจสอบและจัดการแพ็กเก็ตให้เป็นไปตาม policy ที่กำหนดไว้ใน script นั่นคือ โครงสร้างของ Netfilter และ iptables module ตามที่ได้กล่าวไว้ในบทที่ 2

4.2.3.3 การสร้างช่องทางการเชื่อมต่อแบบ Secured Shell

มีขั้นตอนการกระทำดังต่อไปนี้

1) ทดสอบให้แน่ใจว่าระบบทั้งสองกำลังเปิดใช้งาน Secured shell อยู่โดยการใช้คำสั่ง

ps -ax | grep sshd

ssh –l root 192.168.1.1

2) ติดตั้ง Secured shell key บน Management Workstation

ssh-keygen –t dsa

คำสั่งนี้จะสร้าง DSA key ซึ่งเป็น private key (id_dsa) รวมทั้ง public key (id_dsa.pub) สำหรับใช้ติดต่อกับ LBS-IA ซึ่งจะถูกเก็บไว้ใน /root

3) คัดลอก id_dsa.pub ไปยัง LBS-IA โดยเปลี่ยนชื่อ public key เป็น authorized_keys

scp .ssh/id_dsa.pub root@192.168.1.1:.ssh/authorized_keys

4) ทดสอบการใช้ Secured shell กับ LBS-IA

ssh root@192.168.1.1

4.3 iptables script

ในส่วนนี้จะแสดงให้เห็นว่า วิธีการสร้าง iptables script ที่สอดคล้องกับการออกแบบชุดของกฎคำสั่งที่กล่าวถึงบทที่ 3 เพื่อการคัดกรองแพ็กเก็ตให้เป็นไปตามวัตถุประสงค์ของระบบ จะมีรูปแบบภายใน script เป็นอย่างไร

การสร้างชุดของกฎคำสั่งที่แสดงให้เห็นนี้เกิดจากการใช้ shell command ของ iptables และ IPROUTE ซึ่งเป็นโปรแกรมที่ใช้งานด้าน Policy Routing บนระบบปฏิบัติการ Linux ที่มีอยู่แล้ว สำหรับผู้ใช้งานระบบที่ไม่มีความชำนาญเรื่องการใช้ shell command ของ iptables ก็สามารถจัดการชุดของกฎคำสั่งในส่วนของ iptables ได้ด้วยการใช้ Firewall Builder ที่ติดตั้งใช้งานใน Management Workstation ซึ่งมีรายละเอียดการใช้งานในบทที่ 6

4.3.1 การสร้าง Load Balancing Rule

- iptables –t mangle –N NEW_OUT_CONN

  คำอธิบาย: สร้าง NEW_OUT_CONN ภายใต้โครงสร้างข้อมูลของ Mangle table เพื่อใช้เป็นจุดตรวจสอบแพ็กเก็ตแรกของการเชื่อมต่อ

- iptables –t mangle –A PREROUTING –m state --state NEW –j NEW_OUT_CONN

  คำอธิบาย: คัดกรองแพ็กเก็ตที่มีสถานะเป็นแพ็กเก็ตแรกของการเชื่อมต่อที่ผ่านเข้ามาใน PREROUTING chain ภายใต้โครงสร้างข้อมูลของ Mangle table เพื่อส่งต่อไปยัง NEW_OUT_CHAIN

- iptables –t mangle –A NEW_OUT_CONN –j MARK --set–mark 0

  iptables –t mangle –A NEW_OUT_CONN –m random --average 50 –j RETURN

  คำอธิบาย: ทุก 50% ของจำนวนแพ็กเก็ตที่ถูกส่งมาใน NEW_OUT_CONN chain ให้ทำเครื่องหมายเป็นหมายเลข 1 กำกับไว้ภายใต้โครงสร้างข้อมูลของ Mangle table แล้วจากนั้นส่งกลับไปยัง PREROUTING chain

- iptables –t mangle –A NEW_OUT_CONN –j MARK --set–mark 1

  iptables –t mangle –A NEW_OUT_CONN –m random --average 50 –j RETURN

  คำอธิบาย: ทุก 50% ของจำนวนแพ็กเก็ตที่ถูกส่งมาใน NEW_OUT_CONN chain ให้ทำเครื่องหมายเป็นหมายเลข 2 กำกับไว้ภายใต้โครงสร้างข้อมูลของ Mangle table แล้วจากนั้นส่งกลับไปยัง PREROUTING chain

4.3.2 การสร้าง NAT Rule

- iptables –t nat –A POSTROUTING –o eth1 –j MASQUERADE

  คำอธิบาย: ทำ MASQUERADE แพ็กเก็ตภายใต้โครงสร้างข้อมูลของ NAT table สำหรับแพ็กเก็ตที่เป็นแพ็กเก็ตแรกของการเชื่อมต่อหรือเป็นแพ็กเก็ตแรกของการเชื่อมต่อที่เกี่ยวข้องกับการเชื่อมต่ออื่นที่เกิดขึ้นก่อนหน้านี้แล้ว และถูกกำหนดให้ส่งออกช่องทางการเชื่อมต่อชื่อ eth1

- iptables –t nat –A POSTROUTING –o eth2 –j MASQUERADE

  คำอธิบาย: ทำ MASQUERADE แพ็กเก็ตภายใต้โครงสร้างข้อมูลของ NAT table สำหรับแพ็กเก็ตที่เป็นแพ็กเก็ตแรกของการเชื่อมต่อหรือเป็นแพ็กเก็ตแรกของการเชื่อมต่อที่เกี่ยวข้องกับการเชื่อมต่ออื่นที่เกิดขึ้นก่อนหน้านี้แล้ว และถูกกำหนดให้ส่งออกช่องทางการเชื่อมต่อชื่อ eth2

4.3.3 การสร้าง Routing Policy

ชุดของกฎคำสั่งของ Routing Policy ถูกสร้างขึ้นโดยการใช้ shell command ของ IPROUTE โดยมีขั้นตอนดังนี้

1) คัดลอก Main Routing Table ที่ถูกสร้างโดย kernel โดยอัตโนมัติทุกครั้งที่ระบบเริ่มทำงานมาทำเป็น Routing Table 1 สำหรับการเลือกใช้โดย Routing Rule การคัดลอกดังกล่าวดังกระทำโดยอัตโนมัติเมื่อระบบเริ่มทำงาน ฉะนั้นจึงสั่งการคัดลอกดังกล่าวไว้ในไฟล์ /etc/rc.local ซึ่งเป็นไฟล์ที่ถูกเรียกใช้งานทุกครั้งที่เปิดให้ระบบทำงาน ภายในไฟล์ประกอบด้วยคำสั่งต่อไปนี้

- ip route show table main | grep –Ev ^default | while read ROUTE ; do ip route add table 1 $ROUTE; done

- ip route add table 1 default via <*IP address ของ Internet Gateway*> eth2

(ในกรณีนี้ได้ทำการตั้งค่า default route ไว้ที่ eth1 เรียบร้อยแล้วโดยการกำหนด Gateway ไว้บน eth1 โดยตรง)

2) สร้าง Routing Rule เพิ่มเติมจากที่มีอยู่แล้วเพื่อใช้คู่กับ Routing Table 1 ซึ่งโดยปกติ kernel จะมี Routing Rule เริ่มต้นสำหรับการใช้งานกับ Main Routing Table สำหรับการจับคู่กับแพ็กเก็ตที่มีค่า fwmark เป็นหมายเลข 0 อยู่แล้ว (ค่า fwmark เริ่มต้นของทุกแพ็กเก็ต) ฉะนั้นจึงสร้างเฉพาะ Routing Rule ที่จำเป็นต้องใช้คู่กับ Routing Table 1 เท่านั้น โดยมีขั้นตอนต่อไปนี้

- สร้างไฟล์ชื่อ /etc/sysconfig/static-rules โดยมีคำสั่งระบุไว้ภายในไฟล์ดังนี้

  ip rule add fwmark 1 lookup 1

- เพิ่มคำสั่งต่อไปนี้ลงไปในไฟล์ /etc/rc.d/init.d/network

```
if [ -f /etc/sysconfig/static-rules ]; then
    sh /etc/sysconfig/static-rules
fi
```

4.3.4 การสร้าง Link Availability Rule

เพื่อให้ Link Availability Rule สามารถทำงานร่วมกับกลไกการตรวจสอบสถานะของช่องทางการเชื่อมต่อที่ทำโดย kernel นั้น จำเป็นต้องกำหนดรูปแบบของกฎการเลือกใช้ช่องทางการเชื่อมต่อที่พร้อมใช้งาน ไว้ภายใน script การตรวจสอบที่มีอยู่แล้วของ kernel คือภายใต้ /etc/sysconfig/network-scripts/ ของ LBS-IA กฎดังกล่าวนี้เพื่อถูกเรียกใช้ผ่าน script ที่กำลังจะกล่าวถึงข้างล่างนี้ จะเป็นผลให้มีการเปลี่ยนแปลงชุดของกฎคำสั่งที่กำลังใช้งานโดย kernel ให้สามารถเลือกใช้เฉพาะช่องทางการเชื่อมต่ออินเทอร์เน็ตที่พร้อมใช้งานได้ script ดังกล่าวแบ่งเป็น 2 ส่วนดังนี้

4.3.4.1 Script เมื่อตรวจพบความไม่พร้อมใช้งานของช่องทางการเชื่อมต่อ

ในส่วนนี้ต้องกำหนดกฎในรูปแบบของ iptables ไว้ในไฟล์ชื่อ /etc/sysconfig/network-scripts/ifdown ดังนี้

```
if [ "${DEVICE}" = "eth1" ]; then
    iptables –t mantle -F
    iptables –t mangle –A PREROUTING –j MARK –set-mark 1
fi
if [ "${DEVICE}" = "eth2" ]; then
    iptables –t mantle -F
    iptables –t mangle –A PREROUTING –j MARK –set-mark 0
fi
```

เมื่อ script นี้ถูกสั่งให้ทำงาน จะเป็นผลให้กฎการนำส่งแพ็กเก็ตไปยังช่องทางการเชื่อมต่อที่ตรวจพบว่าไม่พร้อมใช้งานนั้น ถูกลบออกจาก Load Balancing Rule พร้อมกับสร้าง Link Availability Rule ขึ้นมาใหม่เพื่อกำหนดให้ทุกแพ็กเก็ตของการเชื่อมต่อที่เกิดขึ้นใหม่ถูกกำหนด fwmark ให้เป็นหมายเลขที่ตรงกับช่องทางการเชื่อมต่อที่พร้อมใช้งาน

4.3.4.2 Script เมื่อตรวจพบความพร้อมใช้งานของช่องทางการเชื่อมต่อ

ในส่วนนี้ต้องกำหนดกฎในรูปแบบของ iptables ไว้ในไฟล์ชื่อ /etc/sysconfig/network-scripts/ifup ดังนี้

```
if [ "${DEVICE}" = "eth1" ]; then
    service iptables restart
fi
if [ "${DEVICE}" = "eth2" ]; then
    service iptables restart
    ip route show table main | grep –Ev ^default | while read ROUTE ; do ip route add table 1
    $ROUTE; done
    ip route add table 1 default via <IP address ของ Internet Gateway> eth2
fi
```

เมื่อ script นี้ถูกสั่งให้ทำงาน สำหรับในกรณีของ eth1 ที่กลับสู่ความพร้อมใช้งาน จะเป็นผลให้มีการเรียกใช้ Load Balancing Rule กลับคืนมาเหมือนเดิม สำหรับในกรณีของ eth2 ที่กลับสู่ความพร้อมใช้งาน จะเป็นผลให้มีการเรียกใช้ Load Balancing Rule กลับคืนมาเหมือนเดิม พร้อมกับการสร้าง Routing Table 1 ขึ้นมาอีกครั้งเนื่องจาก kernel ลบออกจากระบบเมื่อตรวจพบความไม่พร้อมใช้งานของ eth2

บทที่ 5

การทดสอบระบบ

5.1 วัตถุประสงค์ของการทดสอบระบบ

การกำหนดแนวทางในการทดสอบระบบ LBS-IA นี้จะอยู่บนพื้นฐานของวัตถุประสงค์ของการทดสอบดังต่อไปนี้

1) เพื่อพิสูจน์การทำงานร่วมกันระหว่างองค์ประกอบหลักทั้งสามของระบบ ได้แก่ ส่วนการนำส่งกระแสข้อมูลอินเทอร์เน็ต ส่วนการตรวจสอบความพร้อมใช้งานของช่องทางการเชื่อมต่ออินเทอร์เน็ต และ ส่วนการกระจายกระแสข้อมูลบนช่องทางการเชื่อมต่ออินเทอร์เน็ต

2) เพื่อประเมินประสิทธิภาพการทำงานของระบบตามแนวทางที่ได้ออกแบบไว้ สำหรับใช้เป็นแนวทางในการพิจารณาเพื่อติดตั้งใช้งานในสภาพแวดล้อมจริงที่ระบบสามารถรองรับการใช้งานได้

5.2 แนวทางและผลของการทดสอบระบบ

5.2.1 สภาพแวดล้อมของการทดสอบระบบ

เนื่องด้วยข้อจำกัดเรื่องจำนวนช่องทางการเชื่อมต่ออินเทอร์เน็ตที่มีอยู่จริงในการทดสอบระบบ ซึ่งมีเพียงหนึ่งช่องทางการเชื่อมต่อภายในสำนักงานของข้าพเจ้า จึงจำเป็นต้องทำการทดสอบระบบนี้ภายใต้เครือข่ายคอมพิวเตอร์จำลองที่สร้างขึ้นภายในคอมพิวเตอร์ส่วนบุคคลแบบ laptop เครือข่ายฯจำลองดังกล่าวถูกสร้างโดยซอฟต์แวร์ Virtual Machine Workstation 5.0 ที่ติดตั้งใช้งานบนคอมพิวเตอร์แบบ laptop ที่ระบุไว้แล้วในข้อ 4.1

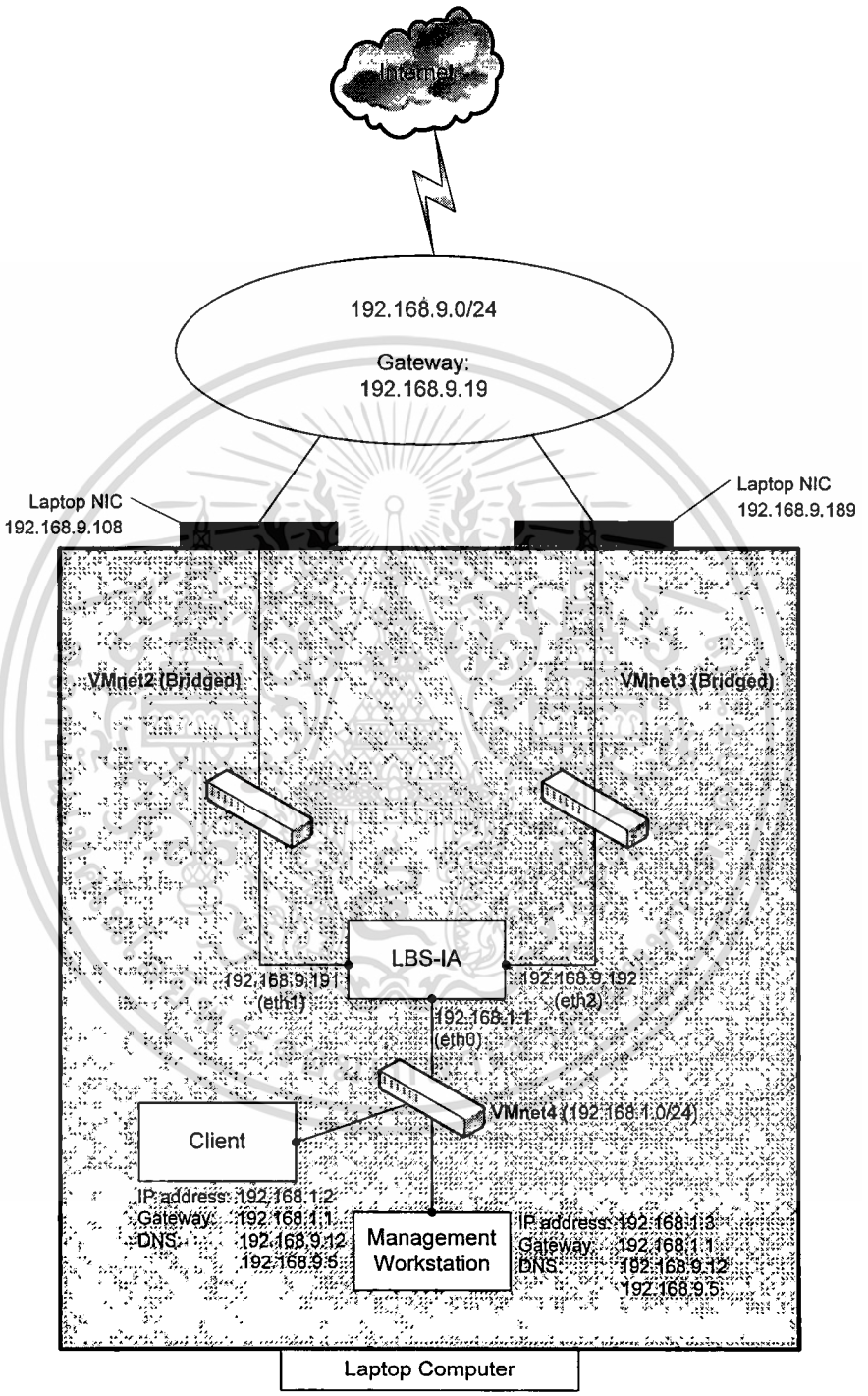
เครือข่ายจำลองที่ถูกสร้างขึ้นประกอบด้วยคอมพิวเตอร์ลูกข่ายจำนวน 2 เครื่อง และ LBS-IA จำนวน 1 ชุด ซึ่งมีรายละเอียดในการเชื่อมต่อดังนี้

1) คอมพิวเตอร์ลูกข่ายจำนวน 2 เครื่อง เชื่อมต่ออยู่ภายในเครือข่ายเดียวกันคือ 192.168.1.0/24 (VMnet4) โดยใช้ DNS server ของเครือข่ายฯจริงที่คอมพิวเตอร์ส่วนบุคคลแบบ laptop เชื่อมต่ออยู่ ดังแสดงให้เห็นในรูปที่ 5.1

- Client ติดตั้งระบบปฏิบัติการ Microsoft Windows XP Professional เพื่อใช้งานเป็นคอมพิวเตอร์ลูกข่ายที่เชื่อมต่อกับอินเทอร์เน็ตผ่าน LBS-IA
- Management Workstation ติดตั้งระบบปฏิบัติการ Red Hat Linux 9 เพื่อใช้ในการจัดการ script ของ iptables ที่จะถูกติดตั้งใช้งานบน LBS-IA

2) LBS-IA เชื่อมต่ออยู่กับสองเครือข่ายฯ คือ VMnet4 ด้วย Network Interface Card (NIC) ชื่อ eth0 และเครือข่ายฯจริงที่คอมพิวเตอร์ส่วนบุคคลแบบ laptop เชื่อมต่ออยู่ผ่าน NIC ชื่อ

eth1 และ eth2 ซึ่งเป็นการจำลองให้เห็นว่าระบบมีช่องทางการเชื่อมต่อออกสู่อินเทอร์เน็ตจำนวน 2 ช่องทาง รายละเอียดการเชื่อมต่อแสดงให้เห็นในรูปที่ 5.1



รูปที่ 5.1 เครือข่ายจำลองสำหรับการทดสอบระบบ

5.2.2 การทดสอบการทำงานร่วมกันระหว่างองค์ประกอบหลักของระบบ

ขอบเขตของการทดสอบประกอบด้วย

- การทดสอบการกระจายการเชื่อมต่อจาก client ออกสู่ช่องทางการเชื่อมต่ออินเทอร์เน็ตของระบบ
- การทดสอบการเลือกใช้ช่องทางการเชื่อมต่อที่พร้อมใช้งาน

5.2.2.1 กระจายการเชื่อมต่อจาก client ออกสู่ช่องทางการเชื่อมต่ออินเทอร์เน็ตของระบบ

<u>เป้าหมายของการทดสอบ</u>

เพื่อทดสอบว่าระบบสามารถใช้ช่องทางการเชื่อมต่ออินเทอร์เน็ตที่มีอยู่ในกระจายการเชื่อมต่อที่เกิดจาก client ได้จริง ซึ่งจะเป็นการพิสูจน์ว่า ส่วนการนำส่งกระแสข้อมูลอินเทอร์เน็ตและส่วนการกระจายกระแสข้อมูลบนช่องทางการเชื่อมต่ออินเทอร์เน็ตสามารถทำงานร่วมกันได้

<u>วิธีการทดสอบระบบ</u>

กำหนดให้ client เชื่อมต่อไปยังเว็บไซต์ www.google.com เพื่อค้นหารูปภาพที่มีชื่อว่า ARSENAL เมื่อผลการค้นหาปรากฏขึ้นให้ทำการเปิดรูปภาพจำนวน 4 รูป

<u>ผลการทดสอบ</u>

การตรวจสอบผลการทดสอบสามารถกระทำได้โดยการแสดงข้อมูล Connection Tracking ที่ระบบจัดเก็บไว้ (**cat \proc\net\ip_contrack | less**) เพื่อการตรวจสอบสถานะการเชื่อมต่อที่ระบบกำลังจัดการอยู่ ซึ่งแสดงให้เห็นว่ามีการสถาปนาการเชื่อมต่อระหว่าง client (192.168.1.2) กับคอมพิวเตอร์แม่ข่ายที่ให้บริการในอินเทอร์เน็ตโดยผ่านการใช้ช่องทางการเชื่อมต่อทั้งสองของ LBS-IA คือ eth1 (192.168.9.191) และ eth2 (192.168.9.192) ดังแสดงให้เห็นในรูปที่ 5.2

รูปที่ 5.2 การใช้ช่องทางการเชื่อมต่อของระบบเพื่อกระจายการเชื่อมต่อออกสู่อินเทอร์เน็ต

5.2.2.2 การเลือกใช้ช่องทางการเชื่อมต่อที่พร้อมใช้งาน

<u>เป้าหมายของการทดสอบ</u>

เพื่อทดสอบว่าในกรณีที่เกิดความล้มเหลวในการใช้ช่องทางการเชื่อมต่อหนึ่ง ระบบสามารถเบี่ยงการเชื่อมต่ออินเทอร์เน็ตที่เกิดขึ้นใหม่ไปใช้ช่องทางการเชื่อมต่อที่เหลืออยู่ได้หรือไม่ ซึ่งเป็นการพิสูจน์ว่าส่วนการตรวจสอบความพร้อมใช้งานของช่องทางการเชื่อมต่ออินเทอร์เน็ต และส่วนการนำส่งกระแสข้อมูลอินเทอร์เน็ตสามารถทำงานร่วมกันได้

<u>วิธีการทดสอบระบบ</u>

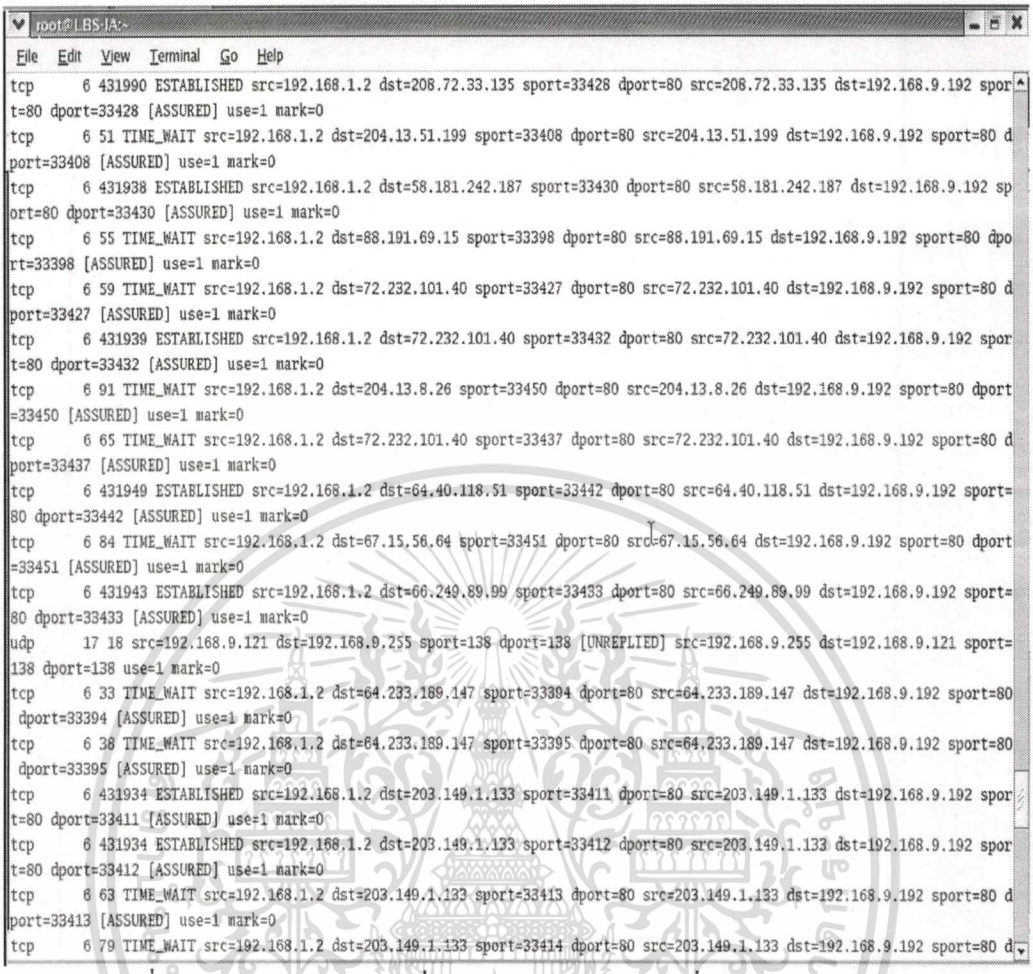
กำหนดให้ client เชื่อมต่อออกสู่อินเทอร์เน็ตในขณะที่สามารถใช้ช่องทางการเชื่อมต่อผ่านระบบได้เพียงช่องทางเดียว สำหรับการกำหนดให้ระบบมีช่องทางการเชื่อมต่อได้ครั้งละหนึ่งช่องทางนั้นสามารถทำได้โดยการระงับการใช้งานช่องทางการเชื่อมต่อด้วยคำสั่ง **ifdown**

<u>ผลการทดสอบ</u>

การตรวจสอบผลการทดสอบสามารถกระทำได้โดยการแสดงข้อมูล Connection Tracking ที่ระบบจัดเก็บไว้ (**cat \proc\net\ip_contrack | less**) เพื่อการตรวจสอบสถานะการเชื่อมต่อที่ระบบกำลังจัดการอยู่ ซึ่งแสดงให้เห็นว่า ณ เวลาที่ระบบสามารถใช้ช่องทางการเชื่อมต่อ

ได้เพียงช่องทางเดียวนั้น จะมีการสถาปนาการเชื่อมต่อระหว่าง client (192.168.1.2) กับคอมพิวเตอร์แม่ข่ายในอินเทอร์เน็ตผ่านช่องทางการเชื่อมต่อเดียวของ LBS-IA คือ eth1 (192.168.9.191) หรือ eth2 (192.168.9.192) ดังแสดงให้เห็นในรูปที่ 5.3 และ 5.4 ตามลำดับ

รูปที่ 5.3 การใช้ช่องทางการเชื่อมต่อของระบบในกรณีที่ eth2 ไม่พร้อมใช้งาน



รูปที่ 5.4 การใช้ช่องทางการเชื่อมต่อของระบบในกรณีที่ eth1 ไม่พร้อมใช้งาน

5.2.3 การทดสอบประสิทธิภาพการทำงานของระบบ

ขอบเขตของการทดสอบประกอบด้วย

- การทดสอบประสิทธิภาพการสร้างสมดุลของ load บนช่องทางการเชื่อมต่อ
- การทดสอบประสิทธิภาพการรวมช่องสัญญาณเชื่อมต่ออินเทอร์เน็ต
- การทดสอบขีดจำกัดในการรองรับจำนวนการเชื่อมต่อของระบบ

5.2.3.1 ประสิทธิภาพการสร้างสมดุลของ load บนช่องทางการเชื่อมต่อ

<u>เป้าหมายของการทดสอบ</u>

เพื่อตรวจสอบว่าจำนวนการเชื่อมต่อที่ถูกกระจายผ่านช่องทางการเชื่อมต่อทั้งสองมีความสมดุลกันโดยตรวจสอบปริมาณข้อมูลที่เกิดขึ้นบน eth1 และ eth2

<u>วิธีการทดสอบ</u>

การทดสอบนี้ถูกกระทำพร้อมๆกับการทดสอบในข้อ 5.2.2.1 โดยใช้ซอฟต์แวร์ vnStat ซึ่งสามารถวิเคราะห์ระบบไฟล์ /proc ภายในของระบบที่เก็บข้อมูลเกี่ยวกับ NIC แล้วนำมาแสดงผลสรุปให้เห็น

<u>ผลการทดสอบ</u>

จากรูปที่ 5.5 แสดงให้เห็นว่าปริมาณข้อมูลที่เกิดบน eth1 และ eth2 มีขนาดที่ใกล้เคียงกัน ซึ่งบ่งชี้ว่าระบบสามารถกระจายจำนวนการเชื่อมต่อที่เกิดขึ้นออกสู่ช่องทางการเชื่อมต่อทั้งสองได้อย่างสมดุลกัน (แพ็คเก็ตทั้งหมดมีขนาดเท่าหรือใกล้เคียงกัน)

```
[root@LBS-IA root]# vnstat -u
[root@LBS-IA root]# vnstat
                          rx      /      tx      /     total    /  estimated
 eth1:
        today       25.17 MB   /    1.76 MB   /    26.93 MB   /       38 MB

 eth2:
        today       26.26 MB   /    0.29 MB   /    26.56 MB   /       39 MB

 eth0:
        today        2.07 MB   /   47.75 MB   /    49.83 MB   /       74 MB

[root@LBS-IA root]# _
```

รูปที่ 5.5 ความสมดุลของ load บนช่องทางการเชื่อมต่อทั้งสอง

5.2.3.2 ประสิทธิภาพการรวมช่องสัญญาณเชื่อมต่ออินเทอร์เน็ต

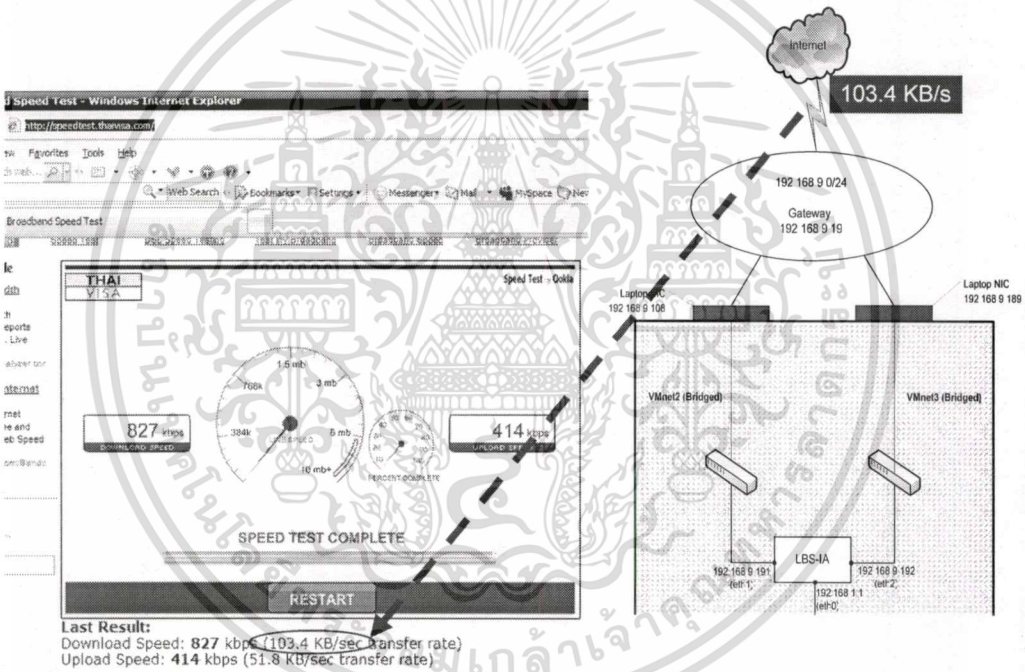
<u>เป้าหมายของการทดสอบ</u>

เพื่อตรวจสอบว่าระบบสามารถรวมช่องสัญญาณเชื่อมต่ออินเทอร์เน็ตที่มีอยู่ได้หรือไม่ รวมทั้งการตรวจสอบประสิทธิภาพการรวมช่องสัญญาณของระบบเมื่อเปรียบเทียบกับการเชื่อมต่ออินเตอร์แบบปกติโดยไม่ใช้ระบบ LBS-IA

<u>วิธีการทดสอบ</u>

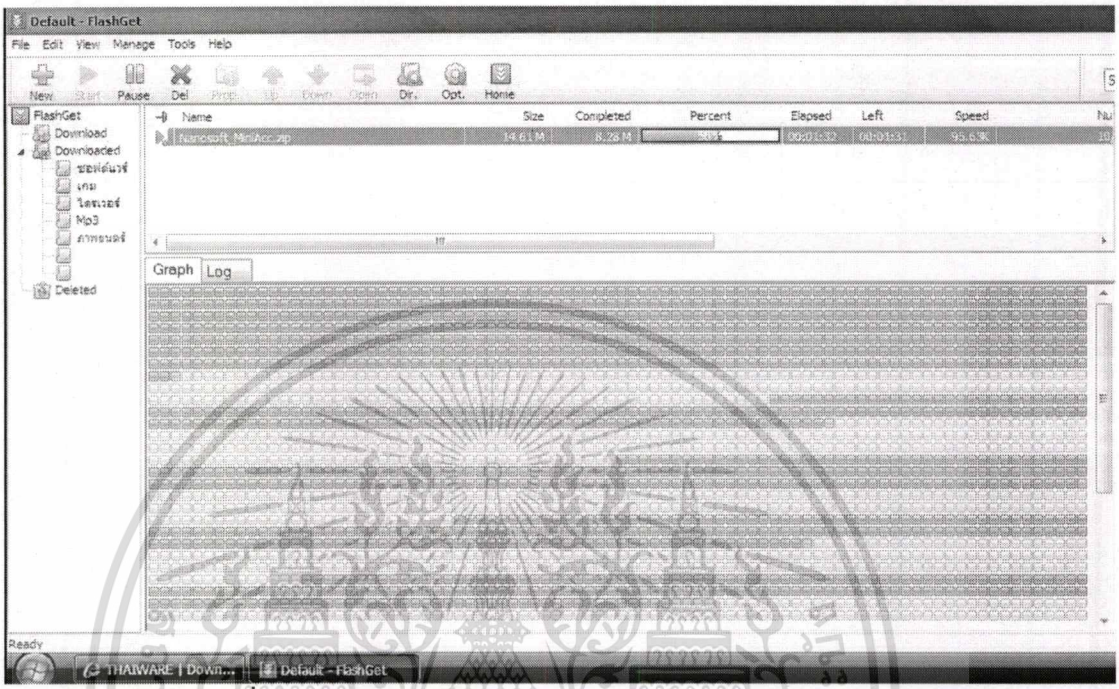
มีขั้นตอนดังต่อไปนี้

1. บันทึกขนาดช่องสัญญาณที่มีอยู่ล่าสุดในการเชื่อมต่ออินเทอร์เน็ตของสำนักงานด้วยการใช้บริการจากเว็บไซต์ http://speedtest.thaivisa.com เพื่อใช้เป็นเกณฑ์ของขนาดช่องสัญญาณรวมที่ระบบสามารถใช้งานได้ ณ เวลาของการทดสอบ ดังแสดงให้เห็นในรูปที่ 5.6



รูปที่ 5.6 การตวจสอบขนาดช่องสัญญาณเชื่อมต่ออินเทอร์เน็ตด้วยเว็บไซต์ speedtest.thaivisa.com

2. การทดสอบการใช้ช่องทางการเชื่อมต่ออินเทอร์เน็ตทั้งสองของระบบทำได้โดยการใช้ซอฟต์แวร์ Flash Get 1.9 เพื่อสร้างการเชื่อมต่อพร้อมกัน 10 การเชื่อมต่อในการดาวน์โหลดไฟล์หนึ่งจากเว็บไซต์ www.thaiware.com ในรูปที่ 5.7 แสดงให้เห็นการใช้ซอฟต์แวร์ Flash Get 1.9



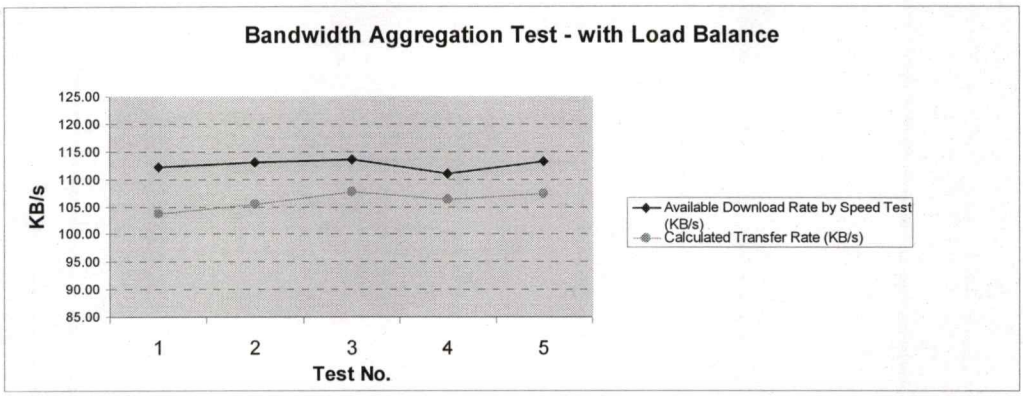
รูปที่ 5.7 การดาวน์โหลดไฟล์ด้วยซอฟต์แวร์ Flash Get 1.9

ผลการทดสอบ

1. จากผลการทดสอบที่แสดงให้เห็นใน ตารางที่ 5.1 และ รูปที่ 5.8 เป็นการทดสอบดาวน์โหลดไฟล์ขนาด 14610 KB โดยการทำการทดสอบจำนวน 5 ครั้ง ซึ่งแสดงให้เห็นว่าระบบสามารถใช้ขนาดช่องสัญญาณอินเทอร์เน็ตรวมที่มีอยู่ (Available Download Rate) ได้เกือบเต็มขนาดช่องสัญญาณ โดยคิดเป็นร้อยละ 94.22 ของขนาดช่องสัญญาณรวมเฉลี่ย ทั้งนี้ขึ้นอยู่กับตัวแปรที่ไม่สามารถควบคุมได้ในระหว่างการทดสอบ เช่น จำนวนการเชื่อมต่ออินเทอร์เน็ตภายในสำนักงานที่กำลังใช้ช่องสัญญาณอินเตอร์ร่วมกันกับการทดสอบ เป็นต้น

ตารางที่ 5.1 ผลการทดสอบการใช้ช่องสัญญาณเชื่อมต่ออินเทอร์เน็ตรวม

	Test No.					
Speed Test at http://speedtest.thaivisa.com	1	2	3	4	5	Average
Available Download Rate by Speed Test (KB/s)	112.10	113.00	113.50	111.00	113.30	**112.58**
Time to complete (seconds)	141.00	138.60	135.60	137.40	136.20	
Calculated Transfer Rate (KB/s)	103.62	105.41	107.74	106.33	107.27	**106.07**
Percentage of Aggregation	92.43	93.28	94.93	95.79	94.68	**94.22**



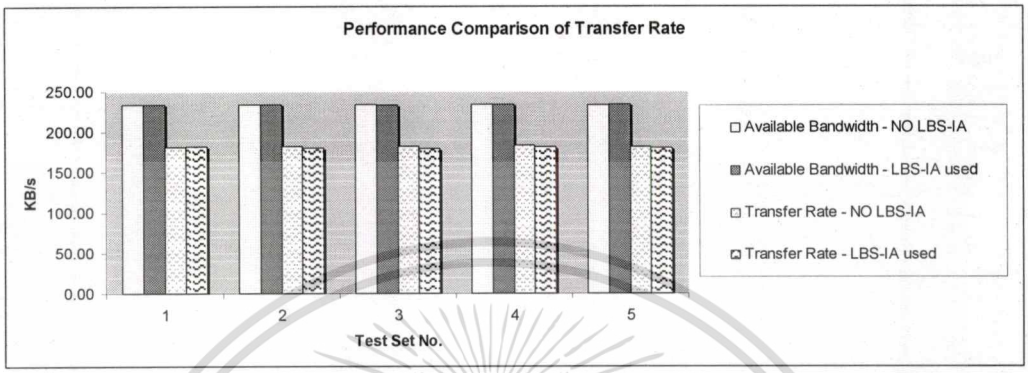
รูปที่ 5.8 กราฟเปรียบเทียบช่องสัญญาณอินเทอร์เน็ตรวมที่มีอยู่กับการใช้งานจริงในการดาวน์โหลดไฟล์ขนาด 14610 KB

นอกจากนั้นในรูปที่ 5.9 ซึ่งเกิดจากการตรวจสอบอัตราการถ่านโอนข้อมูลของช่องทางการเชื่อมต่ออินเตอร์ทั้งสองของระบบด้วยซอฟต์แวร์ vnStat ทำให้เห็นว่าช่องสัญญาณรวมเฉลี่ยที่ระบบสามารถใช้งานได้ประมาณ 106 KB/s (ในตารางที่ 5.1) เกิดจากการรวมอัตราการถ่ายโอนข้อมูลของแต่ละช่องทางการเชื่อมต่อของระบบ

```
[root@LBS-IA root]# vnstat -i eth1 -tr
491 packets sampled in 5 seconds
Traffic average for eth1

     rx         50.09 kB/s              36 packets/s
     tx          4.31 kB/s              61 packets/s

[root@LBS-IA root]# vnstat -i eth2 -tr
186 packets sampled in 5 seconds
Traffic average for eth2

     rx         52.94 kB/s              37 packets/s
     tx          0.00 kB/s               0 packets/s
```

รูปที่ 5.9 อัตราการถ่ายโอนข้อมูลในช่องทางการเชื่อมต่ออินเทอร์เน็ตของระบบ

2. จากผลการทดสอบที่แสดงให้เห็นใน รูปที่ 5.10 และ ตารางที่ 5.2 – 5.6 แสดงให้เห็นผลการทดสอบเปรียบเทียบประสิทธิภาพการรวมช่องสัญญาณเชื่อมต่ออินเทอร์เน็ตของระบบกับการเชื่อมต่ออินเตอร์แบบปกติที่ไม่ใช้ระบบ load balancing



รูปที่ 5.10 แผนภูมิแท่งเปรียบเทียบประสิทธิภาพการรวมช่องสัญญาณเชื่อมต่ออินเทอร์เน็ตของระบบกับการเชื่อมต่ออินเทอร์เน็ตแบบปกติ

การทดสอบดังกล่าวถูกแบ่งออกเป็น 5 ชุด แต่ละชุดประกอบด้วยการทดสอบจำนวน 20 ครั้ง ครึ่งหนึ่งใช้สำหรับทดสอบความสามารถในการใช้ช่องสัญญาณเชื่อมต่ออินเทอร์เน็ตแบบปกติ และ อีก 10 ครั้งสำหรับทดสอบความสามารถในการใช้ช่องสัญญาณเชื่อมต่อที่ผ่านการจัดการของระบบ โดยการดาวน์โหลดไฟล์ขนาด 3250 KB

จากรูปที่ 5.10 แสดงให้เห็นว่าการนำระบบมาใช้งานเพื่อรวมช่องสัญญาณการเชื่อมต่ออินเทอร์เน็ตไม่ได้ทำให้ความสามารถในการใช้ช่องสัญญาณรวมด้อยลงมากจนไม่สามารถยอมรับได้ ถึงแม้ว่าค่าเฉลี่ยของความสามารถจะลดลงไปประมาณร้อยละ 0.73 ก็ตาม แต่ทั้งนี้อาจเป็นเหตุได้จากตัวแปรที่ไม่สามารถควบคุมได้ในระหว่างการทดลองดังที่ได้เคยกล่าวไว้แล้ว

ตารางที่ 5.2 – 5.6 ผลการทดสอบประสิทธิภาพการใช้ช่องสัญญาณเชื่อมต่ออินเตอร์เน็ตของระบบ

ตารางที่ 5.2 ผลการทดสอบ ชุดที่ 1

Test Set No. 1	Test No.										
No LBS-IA	1	2	3	4	5	6	7	8	9	10	Average
Available Download Rate by Speed Test (KB/s)	234.10	235.00	233.50	234.80	234.40	235.10	233.60	232.90	234.60	234.60	**234.26**
Calculated Transfer Rate (KB/s)	182.58	181.56	181.56	181.56	180.56	186.78	179.56	178.57	181.56	182.58	**181.69**
Time to complete (seconds)	17.80	17.90	17.90	17.90	18.00	17.40	18.10	18.20	17.90	17.80	**17.89**
LBS-IA used											
Available Download Rate by Speed Test (KB/s)	234.50	235.10	234.50	234.50	234.40	234.50	234.10	234.90	234.40	233.30	**234.42**
Calculated Transfer Rate (KB/s)	186.78	182.58	183.62	181.56	180.56	182.58	181.56	181.56	181.56	181.56	**182.39**
Time to complete (seconds)	17.40	17.80	17.70	17.90	18.00	17.80	17.90	17.90	17.90	17.90	**17.82**

ตารางที่ 5.3 ผลการทดสอบ ชุดที่ 2

Test Set No. 2	Test No.										
No LBS-IA	1	2	3	4	5	6	7	8	9	10	Average
Available Download Rate by Speed Test (KB/s)	233.30	234.30	232.30	234.60	235.30	234.30	234.40	234.60	234.50	234.50	**234.21**
Calculated Transfer Rate (KB/s)	180.56	181.56	182.58	181.56	186.78	182.58	182.58	182.58	182.58	181.56	**182.50**
Time to complete (seconds)	18.00	17.90	17.80	17.90	17.40	17.80	17.80	17.80	17.80	17.90	**17.81**
LBS-IA used											
Available Download Rate by Speed Test (KB/s)	233.40	234.00	234.10	233.50	232.60	234.50	234.10	234.40	234.40	234.00	**233.90**
Calculated Transfer Rate (KB/s)	180.56	181.56	181.56	180.56	177.60	182.58	180.56	180.56	182.58	182.58	**181.07**
Time to complete (seconds)	18.00	17.90	17.90	18.00	18.30	17.80	18.00	18.00	17.80	17.80	**17.95**

ตารางที่ 5.4 ผลการทดสอบ ชุดที่ 3

Test Set No. 3	Test No.										
No LBS-IA	1	2	3	4	5	6	7	8	9	10	Average
Available Download Rate by Speed Test (KB/s)	234.10	234.60	234.10	234.60	234.60	234.30	234.10	234.80	234.40	234.10	**234.37**
Calculated Transfer Rate (KB/s)	181.56	182.58	181.56	187.86	182.58	181.56	184.66	182.58	182.58	182.58	**183.01**
Time to complete (seconds)	17.90	17.80	17.90	17.30	17.80	17.90	17.60	17.80	17.80	17.80	**17.76**
LBS-IA used											
Available Download Rate by Speed Test (KB/s)	233.60	229.10	234.50	234.00	232.40	234.50	234.30	235.00	232.40	234.10	**233.39**
Calculated Transfer Rate (KB/s)	180.56	168.39	182.58	181.56	180.56	181.56	180.56	182.58	180.56	180.56	**179.95**
Time to complete (seconds)	18.00	19.30	17.80	17.90	18.00	17.90	18.00	17.80	18.00	18.00	**18.07**

ตารางที่ 5.5 ผลการทดสอบ ชุดที่ 4

Test Set No. 4	Test No.										
No LBS-IA	1	2	3	4	5	6	7	8	9	10	Average
Available Download Rate by Speed Test (KB/s)	234.90	234.60	234.60	234.10	234.80	233.90	234.60	234.50	234.80	234.40	**234.52**
Calculated Transfer Rate (KB/s)	182.58	186.78	182.58	181.56	183.62	181.56	183.62	186.78	185.71	184.66	**183.95**
Time to complete (seconds)	17.80	17.40	17.80	17.90	17.70	17.90	17.70	17.40	17.50	17.60	**17.67**
LBS-IA used											
Available Download Rate by Speed Test (KB/s)	234.10	234.00	234.00	234.50	234.60	234.30	234.50	235.30	234.50	234.00	**234.38**
Calculated Transfer Rate (KB/s)	182.58	180.56	181.56	180.56	181.56	179.56	180.56	183.62	181.56	187.86	**182.00**
Time to complete (seconds)	17.80	18.00	17.90	18.00	17.90	18.10	18.00	17.70	17.90	17.30	**17.86**

ตารางที่ 5.6 ผลการทดสอบ ชุดที่ 5

Test Set No. 5	Test No.										
No LBS-IA	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Average**
Available Download Rate by Speed Test (KB/s)	233.50	233.60	234.10	233.40	232.60	234.40	234.10	234.10	234.10	234.40	**233.83**
Calculated Transfer Rate (KB/s)	182.58	182.58	182.58	172.87	184.66	184.66	182.58	183.62	181.56	180.56	**181.83**
Time to complete (seconds)	17.80	17.80	17.80	18.80	17.60	17.60	17.80	17.70	17.90	18.00	**17.88**
LBS-IA used											
Available Download Rate by Speed Test (KB/s)	234.50	234.10	234.10	234.90	234.50	234.50	235.00	234.50	234.40	234.10	**234.46**
Calculated Transfer Rate (KB/s)	181.56	179.56	179.56	181.56	179.56	180.56	181.56	182.58	180.56	181.56	**180.86**
Time to complete (seconds)	17.90	18.10	18.10	17.90	18.10	18.00	17.90	17.80	18.00	17.90	**17.97**

5.2.3.3 ขีดจำกัดในการรองรับจำนวนการเชื่อมต่ออินเทอร์เน็ตของระบบ

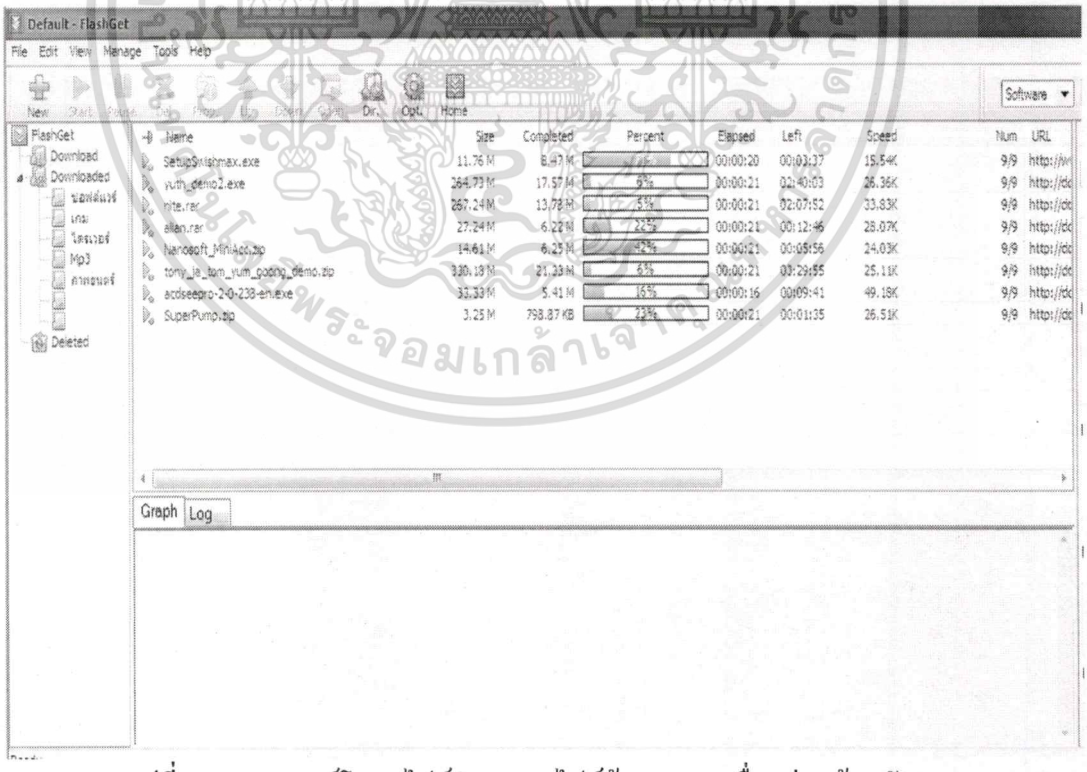
<u>เป้าหมายของการทดสอบ</u>

เพื่อตรวจสอบว่าระบบสามารถรองรับจำนวนการเชื่อมต่อที่เกิดจากภายในเครือข่ายจำลองได้สูงสุดเท่าใด

<u>วิธีการทดสอบ</u>

จากการศึกษาข้อมูลเกี่ยวกับ Connection Tracking พบว่าจำนวนการเชื่อมต่อที่ระบบสามารถรองรับได้ขึ้นอยู่กับขนาดของหน่วยความจำที่ระบบมีใช้งานและการตั้งค่าจำนวนรายการของการเชื่อมต่อสูงสุดที่ระบบไฟล์ proc ของ Connection Tracking (/proc/sys/net/ipv4/ip_conntrack_max) นั่นคือที่ขนาดหน่วยความจำ 64 MB ระบบสามารถรองรับรายการของการเชื่อมต่อที่บันทึกไว้ Connection Tracking ได้สูงสุด 4,096 รายการ ที่หน่วยความจำ 128 MB สามารถรองรับได้สูงสุด 8,182 รายการ ซึ่งโดยเฉลี่ยคิดเป็น 1 MB ต่อ 64 รายการ

อย่างไรก็ตามเพื่อพิสูจน์ให้เห็นว่าระบบต้นแบบนี้สามารถรองรับการใช้งานจริงขององค์กรที่มีกำลังพลไม่เกิน 60 คน จึงได้ทำการทดลองสร้างการเชื่อมต่ออินเทอร์เน็ตพร้อมกันมากว่า 60 การเชื่อมต่อเพื่อดาวน์โหลดไฟล์ขนาดต่างกันจำนวน 8 ไฟล์ ดังแสดงให้เห็นในรูปที่ 5.11



รูปที่ 5.11 การดาวน์โหลดไฟล์จำนวน 8 ไฟล์ด้วย 80 การเชื่อมต่อพร้อมกัน

<u>ผลการทดสอบ</u>

สามารถดาวน์โหลดไฟล์ทั้ง 8 ได้เสร็จสมบูรณ์โดยไม่เกิดปัญหากับการทำงานของระบบ

บทที่ 6
การใช้งานระบบ

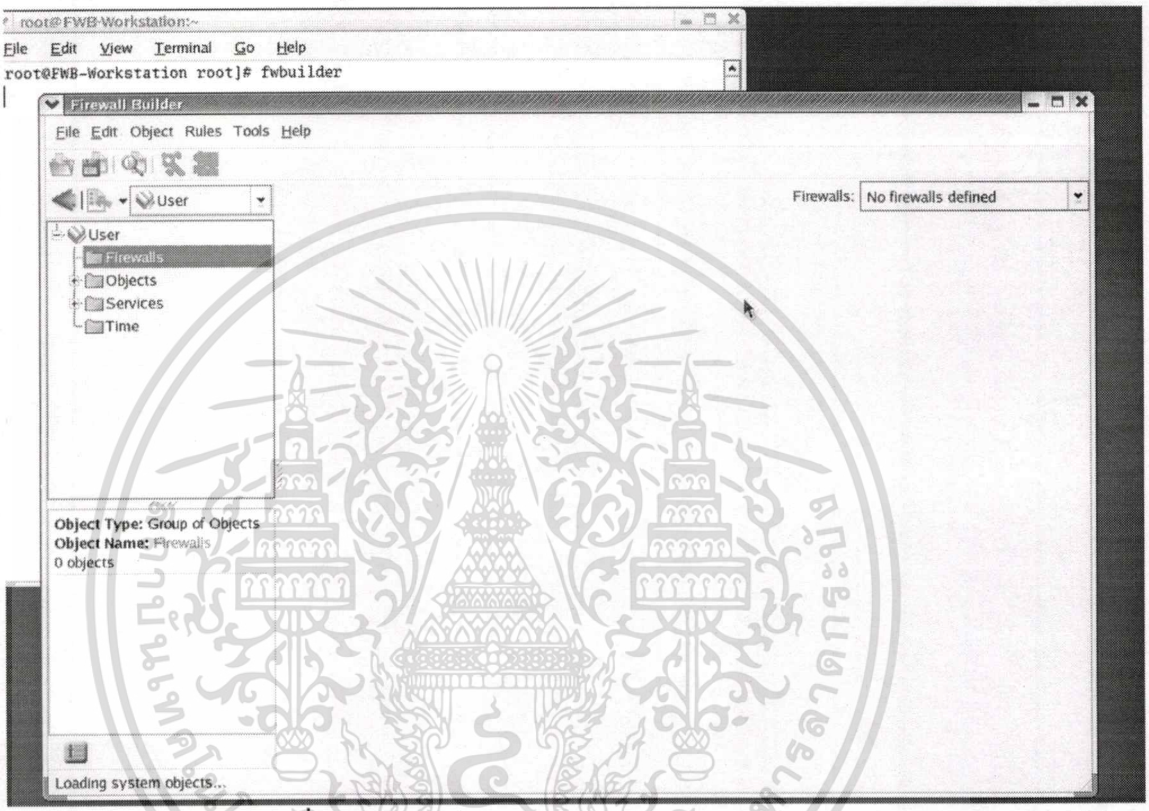
ทันทีที่ระบบทั้งหมดถูกติดตั้งพร้อมใช้งานเรียบร้อยแล้ว ซึ่งหมายถึงการติดตั้งระบบทั้ง 2 เสร็จสิ้นสมบูรณ์ การสร้างการติดต่อสื่อสารระหว่าง Management Workstation และ LBS-IA ประสบความสำเร็จแล้ว ทั้งนี้ยังไม่จำเป็นต้องเชื่อมต่อ LBS-IA เข้ากับอุปกรณ์เครือข่ายของ ISP ก็สามารถใช้ Management Workstation สร้าง iptables script เพื่อติดตั้งรอการใช้งานได้ทันที

ฉะนั้นการใช้งานระบบทั้งหมดโดยผู้ใช้ที่ไม่มีความชำนาญการใช้ shell command จะกระทำบน Management Workstation ที่เดียวโดยมีลำดับการใช้งานเพื่อสร้าง script สำหรับ LBS-IA ประกอบด้วย

1) การเข้าสู่ระบบของ Firewall Builder
2) การสร้างวัตถุสำหรับ LBS-IA
3) การสร้าง Policy สำหรับ LBS-IA
4) การสร้าง iptables script
5) การติดตั้ง iptables script ใน LBS-IA

6.1 การเข้าสู่ระบบของ Firewall Builder

การสั่งการทำงาน Firewall Builder สามารถทำได้โดยการใช้คำสั่ง **fwbuilder** เพื่อเรียกใช้ GUI ของ Firewall Builder ตามรูปที่ 6.1

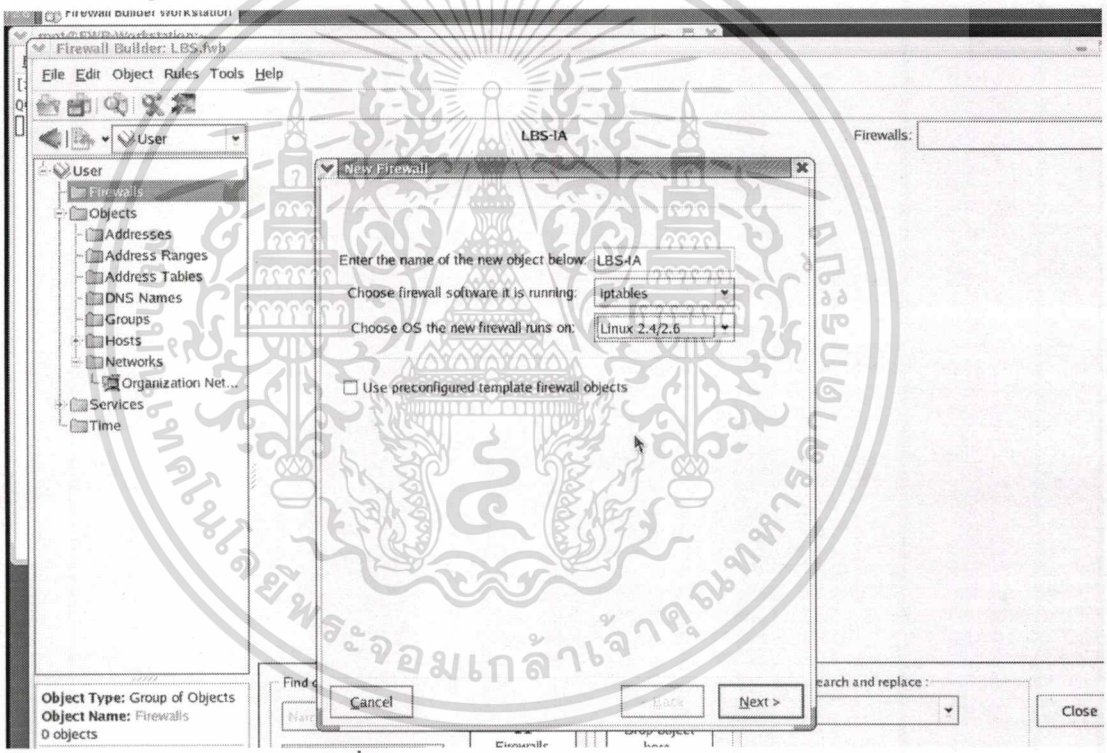


รูปที่ 6.1 หน้าจอเข้าสู่ระบบของ Firewall Builder

6.2 การสร้างวัตถุสำหรับ LBS-IA

กระบวนการสร้าง Policy ที่จะถูกติดตั้งใช้งานใน LBS-IA เริ่มจากการสร้างวัตถุที่เกี่ยวข้องกับ policy ตามรูปที่ 6.2 ซึ่งประกอบไปด้วย

- เครือข่ายคอมพิวเตอร์ของ ISP และเครือข่ายภายในขององค์กร
- การกำหนด Network Interface Card ของ LBS-IA
- การกำหนดระบบปฏิบัติการของ LBS-IA
- การกำหนดระบบการคัดกรองแพ็กเก็ตที่ติดตั้งใช้งานภายใน LBS-IA ในที่นี้คือ iptables



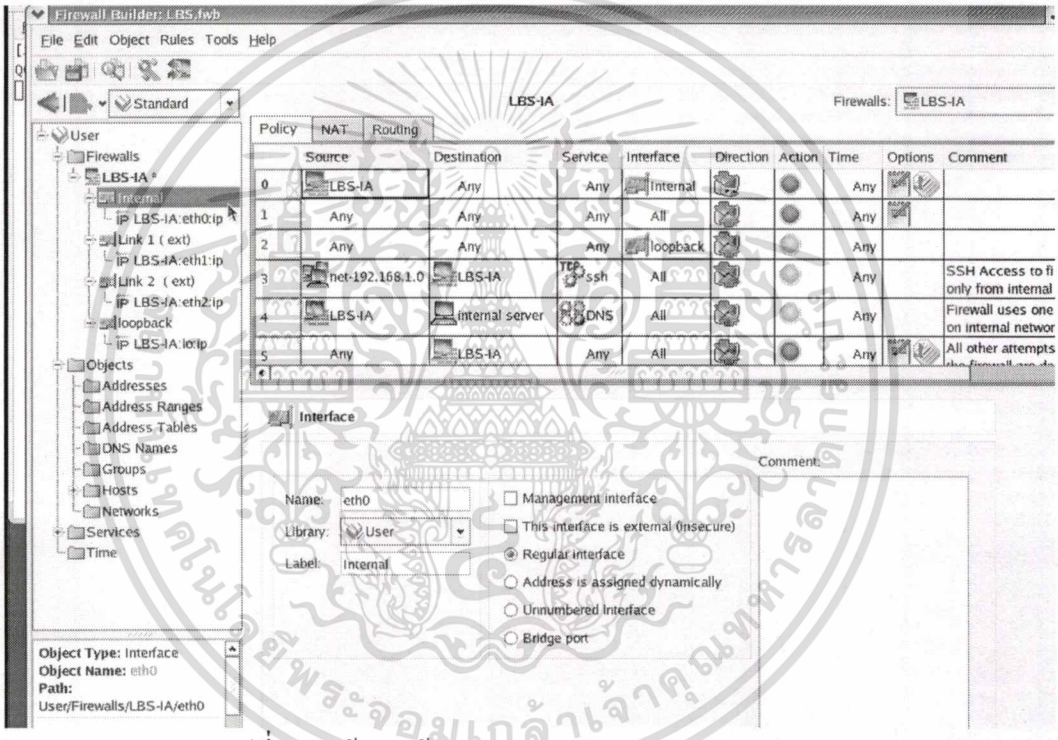
รูปที่ 6.2 หน้าจอสร้างวัตถุของ LBS-IA

6.3 การสร้าง Policy สำหรับ LBS-IA

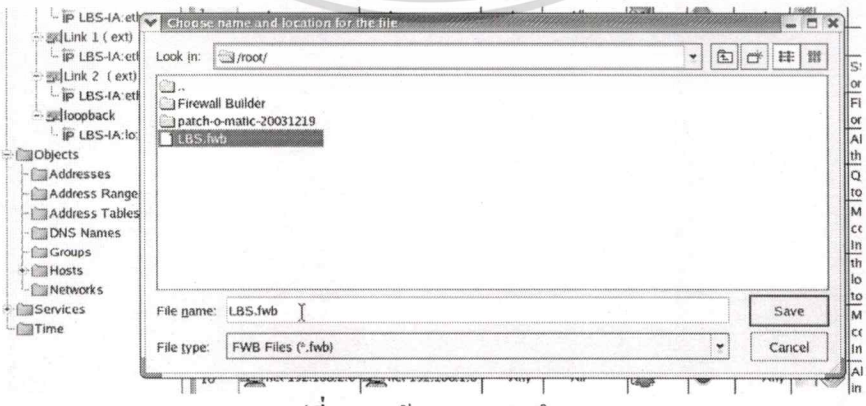
การสร้าง Policy ที่จะใช้สำหรับ LBS-IA จะอยู่ในรูปแบบของ Interface policy นั่นคือ policy จะถูกเชื่อมโยงกับการใช้ Network interface ที่ได้สร้างไว้แล้วในกระบวนตามข้อ 6.3

Link Assignment Policy จะถูกกำหนดไว้ในแท็บ Policy และ Routing ของแต่ละ Network interface (eth1 และ eth2) ส่วน NAT Policy จะถูกกำหนดไว้ที่แท็บ NAT ของแต่ละ Network interface เช่นกัน

รูปที่ 6.3 แสดงให้เห็นภาพของการสร้าง Policy ที่อยู่ในรูปแบบของกฎที่เชื่อมโยงกับแต่ละ Network interface



รูปที่ 6.3 หน้าจอสร้าง Policy สำหรับ Network Interface



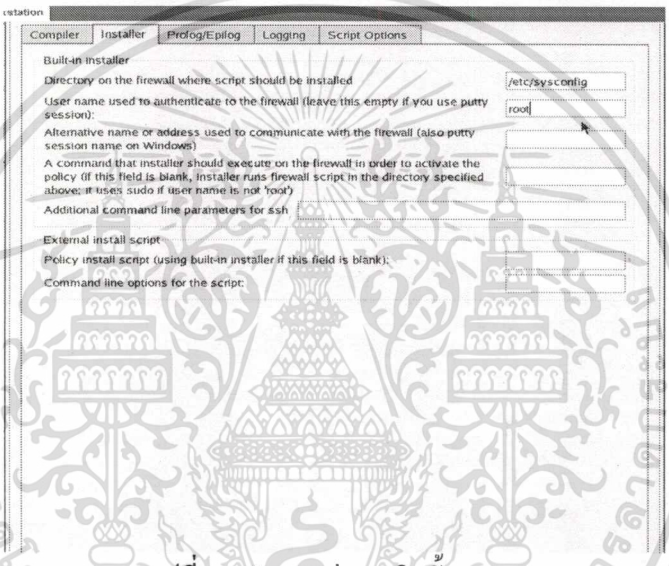
รูปที่ 6.4 หน้าจอการบันทึก Policy

หลังจากเสร็จการกำหนด Policy ทั้งหมดแล้ว ทำการบันทึกข้อมูลเพื่อ compile policy ต่อไปตามรูปที่ 6.4

6.4 การสร้าง iptable script

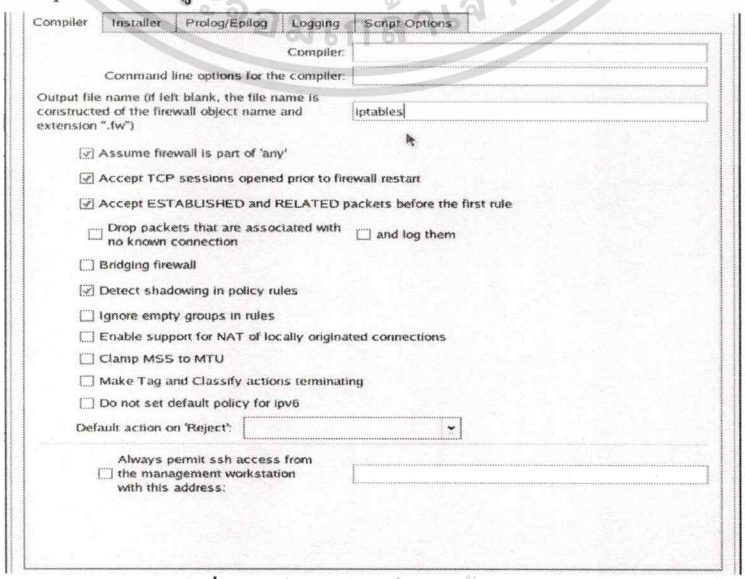
หลังจาก Policy ทั้งหมดถูกบันทึกเป็นไฟล์ในรูปแบบของ Firewall Builder แล้ว ต่อไปเป็นขั้นตอนการแปลงไฟล์ดังกล่าวให้อยู่ในรูปแบบ iptables script ที่ LBS-IA สามารถนำไปใช้งานได้ ขั้นตอนนี้เรียกว่าการ Compile policy โดยมีลำดับการกระทำดังนี้

- กำหนดค่าการติดตั้ง script บน LBS-IA โดยคลิกปุ่ม Firewall Settings และกำหนดค่าดังต่อไปนี้
  - โฟลเดอร์ที่ script จะถูกติดตั้งใน LBS-IA คือ /etc/sysconfig และ User name สำหรับ root ที่ใช้ในการติดตั้ง script ตามรูปที่ 6.5



รูปที่ 6.5 กำหนดค่าการติดตั้ง script

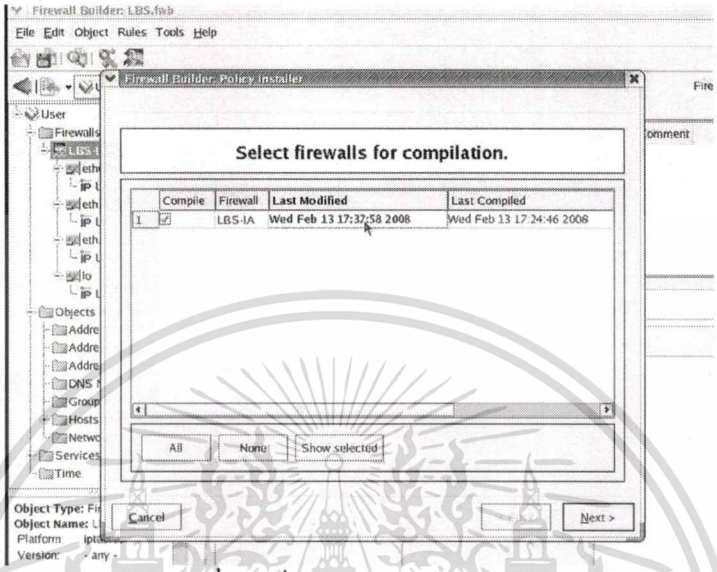
- กำหนดชื่อของไฟล์ script ให้ใช้ชื่อเดียวกับที่ LBS-IA กำลังใช้งานอยู่ นั่นคือ iptables ตามรูปที่ 6.6



รูปที่ 6.6 การกำหนดชื่อไฟล์ของ script

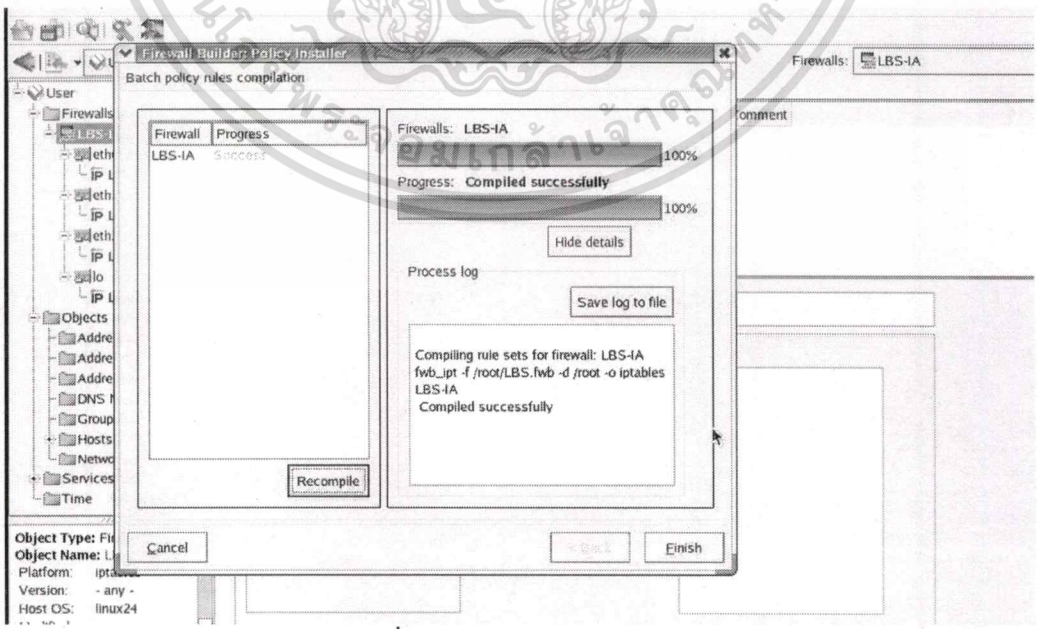
สั่งการทำงานให้ Firewall Builder ทำการ compile policy เป็น script ที่ตั้งค่าไว้แล้ว

○ ที่เมนู Rule เลือก Compile หน้าจอการเลือก policy เพื่อนำมา compile จะปรากฏให้เห็นตามรูปที่ 6.7



รูปที่ 6.7 เริ่มต้นการ compile policy

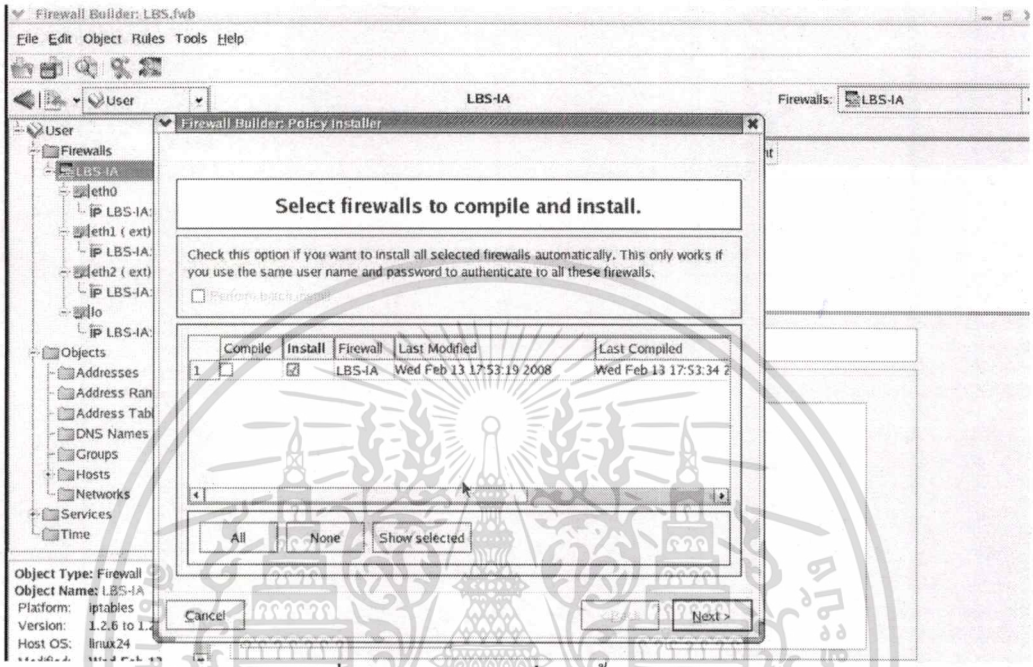
○ คลิกปุ่ม Next เพื่อ compile policy ผลการกระทำจะแสดงให้เห็นตามรูปที่ 6.8 ซึ่งแสดงให้เห็นว่าไฟล์ LBS.fwb ถูก compile ให้อยู่ในรูปแบบของ iptables script ที่มีชื่อไฟล์ iptables เสร็จสิ้นสมบูรณ์



รูปที่ 6.8 ผลการ compile policy

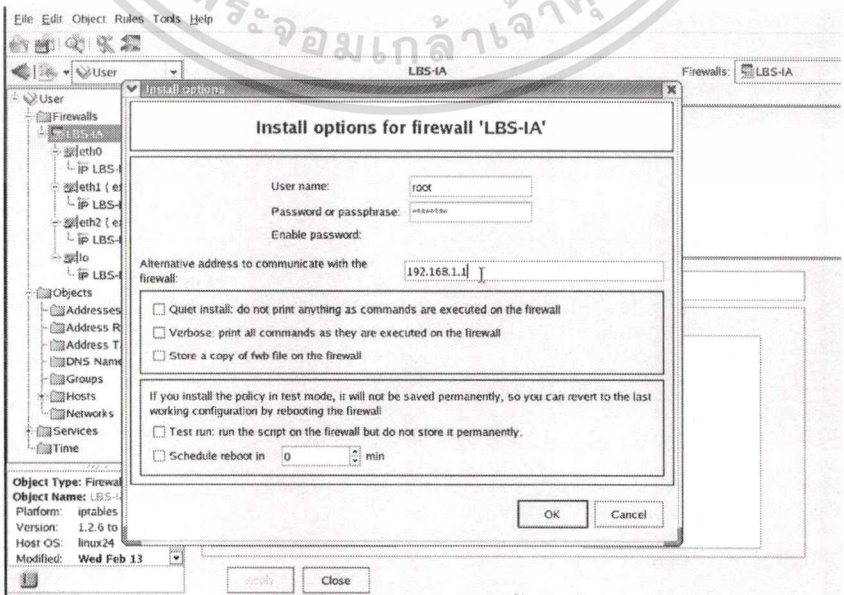
6.5 การติดตั้ง iptable script

เมื่อ iptables script ถูกสร้างพร้อมติดตั้งใน LBS-IA แล้ว สามารถดำเนินการติดตั้ง script ได้จากเมนู Rules แล้วคลิกที่ Install หน้าจอการเลือก script จะปรากฏขึ้นตามรูปที่ 6.9



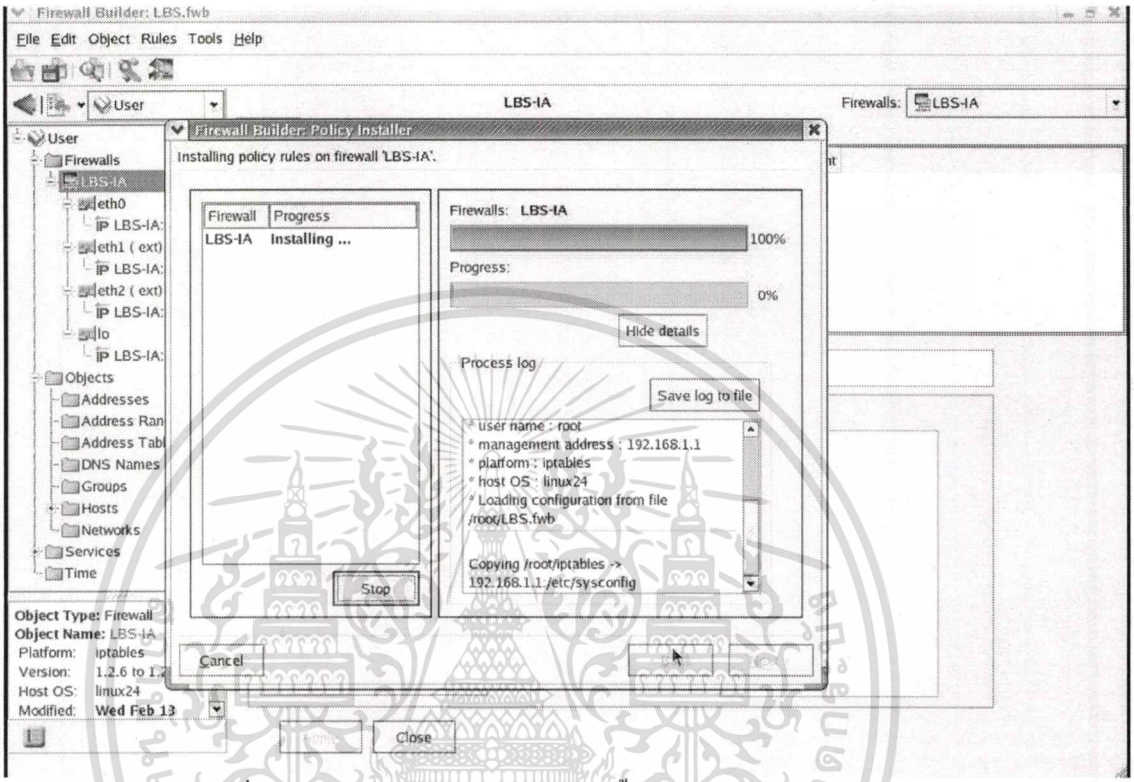
รูปที่ 6.9 เลือก script เพื่อติดตั้งใน LBS-IA

จากนั้นให้คลิกที่ปุ่ม Next เพื่อกำหนด password สำหรับ Root ที่ใช้ในการติดตั้ง script พร้อมทั้งระบุ IP address ของ LBS-IA คือ 192.168.1.1 เพื่อการติดต่อสื่อสารผ่าน Secured shell (ในกรณีนี้ไม่สามารถระบุเป็นชื่อ LBS-IA ได้เพราะไม่ได้ติดตั้งใช้งาน Name Resolution service ภายในเครือข่าย) ตามรูปที่ 6.10



รูปที่ 6.10 ตัวเลือกก่อนการติดตั้ง script

เมื่อคลิก OK จากในรูปที่ 6.10 Firewall Builder จะเริ่มกระบวนการติดตั้ง iptables script ใน LBS-IA ลงในโฟลเดอร์ /etc/sysconfig พร้อมกับสั่งการทำงาน script ทันทีที่ติดตั้งเสร็จสิ้นสมบูรณ์ ซึ่งแสดงให้เห็นเป็นความก้าวหน้าและผลการติดตั้ง script ในรูปที่ 6.11



รูปที่ 6.11 ความก้าวหน้าและผลการติดตั้ง script ใน LBS-IA

บทที่ 7
บทสรุป

7.1 สรุปผลการดำเนินงาน

การพัฒนาระบบ Load Balancing บนระบบปฏิบัติการ Linux เพื่อรวมช่องสัญญาณการเข้าถึงอินเทอร์เน็ตนั้น เป็นการพัฒนาในเชิงของการประยุกต์ใช้งานเทคโนโลยี Linux ที่มีอยู่แล้ว ซึ่งพัฒนาไว้โดยกลุ่มผู้เชี่ยวชาญในวงการ Linux และนักพัฒนาซอฟต์แวร์สำหรับระบบปฏิบัติการ Linux เช่น องค์กรของ Netfilter และนักพัฒนาซอฟต์แวร์การใช้งาน iptables เพื่อนำมาเชื่อมโยงเข้ากันเป็นระบบให้สามารถทำงานได้ตามวัตถุประสงค์ที่ระบุไว้ในโครงงานนี้ เหตุผลที่เลือกใช้แนวทางการพัฒนาระบบเช่นนี้คือ

1) จากผลการศึกษาระบบในสัมมนา 2 พบว่าระบบเป้าหมายประกอบด้วยองค์ประกอบหลักที่จำเป็นต้องทำงานร่วมกันเป็นหนึ่งเดียวเพื่อให้บรรลุเป้าหมายของระบบ ซึ่งแสดงให้เห็นถึงความซับซ้อนที่มีอยู่ในการพัฒนาระบบในเวลาที่มีอยู่อย่างจำกัด
2) ผู้เขียนมีขีดความสามารถที่จำกัดในเรื่องการพัฒนาซอฟต์แวร์ขึ้นใช้งานเอง เนื่องจากไม่มีประสบการณ์ด้านการเขียนโปรแกรม โดยเฉพาะการเขียนโปรแกรมเพื่อเชื่อมโยงเข้ากับการทำงานของ kernel ในระบบปฏิบัติการ Linux สำหรับระบบเป้าหมาย

อย่างไรก็ตามการดำเนินงานในโครงงานนี้ถูกกระทำบนภายใต้กรอบการทำงานเพื่อการพัฒนาระบบงานที่ได้รับการศึกษามาจากหลักสูตรวิทยาศาสตร์มหาบัณฑิตนี้ ซึ่งสรุปได้ดังนี้คือ

1) การศึกษาหลักการทำงานของระบบ Load Balancing
2) การศึกษาเทคโนโลยี Linux ที่จะนำมาประยุกต์ใช้งานและความเป็นไปได้ในการนำมาใช้งาน
3) การวิเคราะห์เทคโนโลยี Linux ที่ประกอบด้วย โครงสร้างการทำงานของ Netfilter และหลักการทำงานของ iptables ภายใต้โครงสร้างของ Netfilter รวมทั้งการทำงานของซอฟต์แวร์ Firewall Builder สำหรับการจัดการ iptables
4) การออกแบบการประยุกต์ใช้งาน iptables ที่เหมาะสมกับการทำงานเพื่อกระจายกระแสข้อมูลอินเตอร์ออกสู่ช่องทางการเชื่อมต่ออินเตอร์ 2 ช่องทางเพื่อให้สมดุลกัน
5) การประยุกต์ใช้งานองค์ประกอบต่างๆ ของระบบที่ได้วิเคราะห์และออกแบบไว้ภายใต้เครือข่ายจำลองที่สร้างขึ้นโดยซอฟต์แวร์ Virtual Machine ในคอมพิวเตอร์ส่วนบุคคลแบบ laptop

ผลของการพัฒนาระบบนี้สามารถสรุปได้ดังนี้

1) ระบบ Load Balancing ในโครงงานนี้ใช้เทคนิคการสร้างความสมดุลของ load โดยพิจารณาจำนวนการเชื่อมต่อที่เกิดจากเครือข่ายภายในองค์กรเป็นเกณฑ์ เพื่อให้การเชื่อมต่อที่เกิดขึ้นสามารถถูกกระจายออกสู่ช่องทางการเชื่อมต่ออินเทอร์เน็ตที่มีอยู่ได้อย่างเท่าเทียมกัน อย่างไรก็ตามแนวทางนี้มีข้อจำกัดที่เกิดจากเทคโนโลยีที่นำมาประยุกต์ใช้งาน (Random Match Module) ดังนี้
   - ความสมดุลของจำนวนการเชื่อมต่อที่ถูกกระจายออกสู่ช่องทางการเชื่อมต่ออินเทอร์เน็ตที่มีอยู่นั้นขึ้นอยู่กับจำนวนการเชื่อมต่อที่ระบบยอมรับเข้ามาจัดการในแต่ละช่วงเวลา ประกอบกับระบบไม่มีกลไกที่ใช้บันทึกจำนวนการเชื่อมต่อที่ถูกกระจายออกสู่แต่ละช่องทาง ด้วยเหตุนี้จึงมีโอกาสทำให้สูญเสียความสมดุลของ load บนช่องทางการเชื่อมต่อได้ถ้าแต่ละช่วงเวลามีจำนวนการเชื่อมต่อรวมถูกนำส่งเข้าสู่ระบบไม่เท่ากัน
   - ในกรณีที่แต่ละช่องทางการเชื่อมต่ออินเทอร์เน็ตของระบบมีขนาดช่องสัญญาณ (Bandwidth) ที่ไม่เท่ากัน ระบบจะไม่สามารถเลือกใช้ประโยชน์จากช่องทางการเชื่อมต่อที่มีขนาดช่องสัญญาณสูงกว่าได้
2) อีกแนวทางหนึ่งที่สามารถนำมาใช้สร้างความสมดุลของ load คือใช้เทคนิคของการตรวจสอบขนาดช่องสัญญาณของช่องทางการเชื่อมต่ออินเทอร์เน็ต ระบบจำเป็นต้องมีกลไกภายในสำหรับตรวจสอบขนาดช่องสัญญาณที่มีอยู่ตลอดเวลาเพื่อใช้เป็นเกณฑ์ในการเลือกกระจายการเชื่อมต่อที่เกิดขึ้นไปยังช่องทางการเชื่อมต่ออินเทอร์เน็ตที่มีขนาดช่องสัญญาณที่เหลือมากกว่าช่องทางอื่นได้ ด้วยเทคนิคดังกล่าวทำให้สามารถใช้ประโยชน์ขนาดช่องสัญญาณที่มีอยู่จริงได้อย่างมีประสิทธิภาพมากกว่าเทคนิคในข้อ 1 อย่างไรก็ตามแนวทางนี้ก็ยังมีข้อจำกัดดังต่อไปนี้
   - อุปกรณ์ที่นำมาใช้พัฒนาระบบจำเป็นต้องมีประสิทธิภาพการทำงานที่สูงเพียงพอที่จะรับภาระการทำงานที่เกิดขึ้นจากเทคนิคนี้
   - แพ็กเก็ตของการเชื่อมต่อหนึ่งสามารถถูกกระจายออกสู่ช่องทางการเชื่อมต่ออินเทอร์เน็ตที่แตกต่างกันได้ เช่นนี้ทำให้มีโอกาสเกิดปัญหากับลำดับของแพ็กเก็ตในการเชื่อมต่อเดียวกันที่ถูกนำส่งออกสู่ผู้ให้บริการอินเทอร์เน็ตที่แตกต่างกัน เป็นผลให้การเชื่อมต่อลักษณะดังกล่าวเกิดการล้มเหลว

3) การจัดการระบบ Load Balancing สามารถจัดการผ่านการใช้ซอฟแวร์ Firewall Builder ที่ถูกติดตั้งใช้งานบนคอมพิวเตอร์ที่สามารถเชื่อมต่อกับระบบ Load Balancing ผ่านเครือข่ายภายในขององค์กร การจัดการดังกล่าวรวมถึงการสร้างชุดของกฎคำสั่งที่เหมาะสมสำหรับระบบ การติดตั้งและสั่งการทำงานของชุดของกฎคำสั่งบนระบบ และการปรับเปลี่ยนชุดของกฎคำสั่งให้สอดคล้องกับการเปลี่ยนแปลงที่เกิดขึ้นกับระบบ เช่น การเพิ่มหรือลดช่องทางการเชื่อมต่ออินเทอร์เน็ตที่ระบบมีอยู่ เป็นต้น
4) การออกแบบโครงสร้างของชุดของกฎคำสั่งที่เหมาะสมตามหลักการทำงานของโครงสร้าง Netfilter, iptables และ Policy Routing

7.2 ปัญหาที่พบในการพัฒนาระบบ

ปัญหาที่พบในการพัฒนาระบบคือ

1) ข้อจำกัดด้านทรัพยากรของคอมพิวเตอร์ส่วนบุคคลแบบ laptop ที่นำมาใช้ เรื่องของความจุของหน่วยความจำที่น้อยเกินไป ทำให้การใช้งานเครือข่ายจำลองและการใช้งานระบบที่อยู่ภายในเครือข่ายจำลองเป็นไปด้วยความล่าช้า
2) ความไม่พร้อมในเรื่องอุปกรณ์และทรัพยากรที่สามารถนำมาทดสอบการใช้งานบนเครือข่ายคอมพิวเตอร์จริง ทำให้ไม่สามารถหาข้อสรุปที่เกิดจากการใช้งานจริงได้

7.3 ข้อเสนอแนะ

ระบบ Load Balancing ต้นแบบนี้เป็นระบบที่เน้นสร้างความสมดุลจำนวนการเชื่อมต่อที่ถูกกระจายออกสู่ช่องทางการเชื่อมต่อที่มีอยู่เท่านั้น ทั้งนี้ด้วยขอบเขตของการพัฒนาระบบนี้ที่มุ่งเน้นการพัฒนาให้เกิดระบบต้นแบบที่สามารถพัฒนาขึ้นมาใช้งานเองได้ จึงทำให้ยังมีส่วนประกอบของระบบและการเพิ่มความสามารถของระบบอีกหลายส่วนที่ยังไม่ได้มีการพัฒนา ซึ่งสามารถสรุปได้ดังนี้

1) ระบบการติดตามการกระจายจำนวนการเชื่อมต่อที่กำลังใช้งานอยู่
2) ระบบการเก็บบันทึกการใช้งานอินเทอร์เน็ตขององค์กร
3) ระบบการจัดการทรัพยากรภายในของระบบให้เหมาะสมกับ load ที่ระบบต้องรับภาระ
4) ระบบการคำนวณปริมาณสัญญาณการเชื่อมต่อที่เกิดขึ้นจริงเพื่อให้สามารถเพิ่มศักยภาพให้ระบบสร้างความสมดุลของ load ที่เกิดขึ้นจริงบนพื้นฐานของขนาดช่องสัญญาณรวมที่มีใช้งานทั้งหมด

บรรณานุกรม

A. Akella, A. Shaikh, S. Seshan. 2004. **Multihoming Performance Benefits: An Experimental Evaluation of Practical Enterprise Strategies**. Proceedings of the USENIX Annual Technical Conference (Boston, MA).

Edimax. 2007. [Online]. Available : http://www.edimax.com

F. Marie. **Netfilter Extension HOWTO.**

F5. [Online]. 2007. Available : http://www.f5.com

Jasmine Internet Co., Ltd.,. 2007. **DSL one2connext**. [Online]. Available : http://www.ji-net.com

Joe Brockmeir, Dee-Ann LeBlanc, Ron McCarth. **Linux Routing**. New Riders. 2001

Matthew G. Marsh. **Policy Routing with Linux – Online Edition.** [Online]. Available : http://www. policyrouting.org/PolicyRoutingBook/ONLINE/TOC.html

NetCitadel LLC. **Firewall Builder's User Guide – Online Edition.** [Online]. Available: http://www.fwbuilder.org/archives/cat_about.html

NetComm Broadband Solutions. 2007. [Online]. Available : http://www.netcomm.com.au

Oskar Andreasson. **Iptables Tutorial 1.2.2 – Online Edition**. [Online]. Available: http://iptables-tutorial.frozentux.net/

R. Russel. **Linux netfilter Hacking HOWTO.**

R. S. Prasad, M. Murray, C. Dovrolis, K. Claffy. **Bandwidth estimation: metrics, measurement techniques, and tools.**

T. Lindh. **A New Approach to Performance Monitoring in IP Networks – combining active and passive methods.**

Tony Mancill. **Linux Routers, 2nd Edition**. Prentice Hall PTR. 2002

XRoads Network. 2007. [Online]. Available : http://www.xroadsnetworks.com

ประวัติผู้เขียน

ชื่อ	เรืออากาศเอก เจนณรงค์ มุสิกพันธ์
วัน/เดือน/ปี เกิด	5 พฤษภาคม 2518
ประวัติการศึกษา	
ปริญญาตรี	Degree of Bachelor of Engineering with Third Class Honours in a programme of Command & Control, Communications & Information Systems The Royal Military College of Science, Cranfield University, England
ประวัติการทำงาน	นายทหาร บริหารและควบคุมเครือข่ายสารสนเทศสำหรับผู้บังคับบัญชาระดับสูงของกองทัพอากาศ ส่วนปฏิบัติการ สำนักงานเทคโนโลยีสารสนเทศทหารอากาศ กองบัญชาการกองทัพอากาศ ดอนเมือง กรุงเทพมหานคร