การพัฒนาระบบการจัดการแบบมัลติเทอร์มินอล

THE DEVELOPMENT MULTI TERMINAL MANAGEMENT SYSTEM

โดย

วุฒิภูมิ อกนิษฐ์

WUTTIPUME AKANIT

อาจารย์ที่ปรึกษา

ผศ. อัครินทร์ คุณกิตติ

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2550

THE DEVELOPMENT MULTI TERMINAL MANAGEMENT SYSTEM

WUTTIPUME AKANIT

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

1/ 2007

หัวข้อ	การพัฒนาระบบการจัดการแบบมัลติเทอร์มินอล
นักศึกษา	นาย วุฒิภูมิ อกนิษฐ์
รหัสนักศึกษา	48066825
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	ผศ. อัครินทร์ คุณกิตติ

บทคัดย่อ

ในปัจจุบันการตรวจสอบคุณภาพของเครือข่ายไม่สามารถทำได้โดยอัตโนมัติ เนื่องจากไม่มีบุคลากรประจำอยู่หน้างาน เพื่อทำการทดสอบตลอดเวลาซึ่งทำให้สิ้นเปลืองทรัพยากรในการทดสอบปัญหาและผลที่ได้ในการทดสอบไม่น่าเชื่อถือ อาจมีสาเหตุจากทดสอบได้ทีละช่วงเวลาไม่ต่อเนื่อง ความสามารถของบุคลากรไม่เท่ากัน การทดสอบควบคุมได้ยากไม่สามารถจัดการทดสอบการทำงานได้จากส่วนกลางและไม่สามารถรู้ปัญหาที่เกิดขึ้นได้ล่วงหน้า

ดังนั้น ผู้จัดทำจึงมีแนวคิดที่จะพัฒนาระบบการจัดการอุปกรณ์ปลายทาง (คอมพิวเตอร์) ที่ตั้งอยู่หลายๆแห่งโดยควบคุมจากส่วนกลางผ่านเครือข่ายอินเตอร์เน็ต โดยผู้จัดทำได้นำระบบการจัดการการตรวจสอบคุณภาพของเครือข่ายอินเตอร์เน็ต และการจัดการการทำงานของไฟร์วอลล์มานำเสนอ โดยมีวัตถุประสงค์เพื่อแสดงการควบคุมอุปกรณ์ปลายทางได้จากส่วนกลาง โดยผ่านระบบโดยเว็ปแอปพลิเคชันเมเนทเมนต์ (Web Application Management) เพื่อง่ายในการใช้งาน โดยพัฒนาฟังก์ชันทำงานการตรวจสอบเกี่ยวกับ เรื่องการทดสอบแบนด์วิธเพอฟอร์แมนซ์นั้นจะมีการทดสอบในส่วนของการใช้งาน HTTP ,PING ส่วนการควบคุมการทำงานของไฟร์วอล์ ของอุปกรณ์ปลายทางจะใช้ฟังก์ชันการทำงานของ “iptables” บนระบบปฏิบัติการ Linux รวมถึงการทำงานเกี่ยวกับการเก็บข้อมูล(Backup) ของอุปกรณ์ปลายทางการนำข้อมูลกลับไปใช้งาน(Restore) ได้ และการ Remote ไปยังอุปกรณ์ปลายทางผ่านช่องทางที่ปลอดภัย (Secure channel)

โดยโครงการนี้จะช่วยเกี่ยวกับการจัดการการทดสอบคุณภาพตามที่กำหนด ซึ่งสามารถทำการวัดผลการทดสอบคุณภาพของเครือข่ายได้ จากการจัดการจากส่วนกลางและควบคุมการจัดการการทำงานของไฟร์วอลล์ โดยผ่านเว็ปแอปพลิเคชันเมเนทเมนต์ และการติดต่อกับเครื่องปลายทางทั้งหมดสามารถทำงานผ่านช่องทางที่ปลอดภัย (Secure channel) โดยเครื่องปลายทางสามารถทำงานได้โดยอัตโนมัติ

Title	The Development of Multi Terminal Management System
Student	Mr.Wuttipume Akanit
Student ID.	4206825
Degree	Master of Science
Programme	Information Science
Academic Year	200
Advisor	Asst.Prof. Akharin Khunkitti

ABSTRACT

In the present we can't automatically test the performance because of insufficiency about the man who testing all time, wasted the man for testing and the results are unbelievable. Causes are testing a little at a time, discontinue, capability of each people, can't test from the center and unpredictable for the problems.

So, the purpose of this project is developing for management many terminals from center by passing network. For easy to use I'll present about the system of management network and Firewall management via Web Application Management. I've tried to develop the functions of HTTP and PING for testing Bandwidth performance. By the way, managing the terminals I'll use "iptables" on Linux Operating system from Firewall management. For this project it must has the important functions such as backup, restore and remote to the secure channel.

The useful purpose helps about the management of Bandwidth performance. We can measure the quality of network from center controlled and Firewall management via Web Application Management. All of the communications can remote to the secure channel and automatically works at terminals.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานนี้เกิดขึ้น และสำเร็จลุล่วงไปด้วยดี ผู้จัดทำโครงงานขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ อัครินทร์ คุณกิตติ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ที่ได้กรุณาเสียสละเวลาในการให้คำแนะนำและให้คำปรึกษาด้านวิชาการที่เป็นประโยชน์ต่อโครงการ ผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ขอกราบขอบพระคุณคณาจารย์ทุกท่านในคณะที่ได้ให้ความรู้ซึ่งเป็นประโยชน์อย่างยิ่งกับข้าพเจ้าจึง

ขอขอบคุณอรรถวิท รัชตภูมิ เพื่อนร่วมงานและเพื่อนที่คณะที่คอยให้ความช่วยเหลือคำปรึกษา ให้กำลังใจและให้ข้อมูลความรู้ที่เป็นประโยชน์ต่อโครงการตลอดมา

และสุดท้ายขอกราบขอบพระคุณ คุณพ่อคุณแม่ที่ให้กำเนิดและ ให้กำลังใจตลอดมา

วุฒิภูมิ อกนิษฐ์

กันยายน 2550

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในอดีตการควบคุมอุปกรณ์ปลายทาง หรือ เครื่องคอมพิวเตอร์ปลายทางผ่านทางอินเตอร์เน็ต ทำได้ยากมากอาจจะเนื่องจาก ความเร็วของอินเตอร์เน็ตในสมัยนั้นช้าและมีราคาแพงกว่าในสมัยนี้มาก ทำให้ไม่ค่อยมีผู้พัฒนาการควบคุมอุปกรณ์ปลายทาง ผ่านทางเครือข่ายอินเตอร์เน็ต แต่ในปัจจุบันได้มีเทคโนโลยีอินเตอร์เน็ตบอร์ดแบรนด์ มีความเร็วที่สูงกว่าระบบอินเตอร์เน็ตสมัยเก่า และมีราคาไม่แพงมาก ทำให้ในปัจจุบันมีการประยุกต์การใช้งานผ่านอินเตอร์เน็ต เพื่อควบคุมการทำงานของอุปกรณ์ปลายทางมากขึ้น เช่นระบบกล้องวงจรปิด ที่คอยตรวจสอบผู้บุกรุกภายในบ้านและสามารถทำการ Monitor ดูผ่านทางอินเตอร์เน็ตได้ตลอดเวลา

ดังนั้นผู้จัดทำจึงมีแนวคิดที่จะมีการจัดการควบคุมอุปกรณ์ปลายทางผ่านทางเครือข่ายอินเตอร์เน็ตที่สามารถจะทำการ ควบคุมการทำงานได้จากส่วนกลางทั้งหมดเพื่อทำให้เห็นภาพการจัดการมากขึ้นผู้จัดทำจึงนำระบบการจัดการการตรวจสอบ คุณภาพของอินเตอร์เน็ตบอร์ดแบรนด์ และการจัดการการทำงานของไฟร์วอล์ ของอุปกรณ์ปลายทางมาแสดง โดยสามารถจัดการได้โดยง่ายผ่านทางเว็ปเมเนทเมนต์ (Web Management) เพื่อจะได้ง่ายในการใช้งานแต่เนื่องจากในปัจจุบันการใช้งานอินเตอร์เน็ตมีผู้ใช้งานกันอย่างกว่างขวางระบบความปลอดภัยในเครื่อข่ายจึงมีความสำคัญ ดังนั้นระบบจะต้องมีความปลอดภัยในการสื่อสารกันระหว่างอุปกรณ์ปลายทาง และตัวจัดการส่วนกลาง(server)

1.2 เป้าหมายในการพัฒนา

ระบบสามารถควบคุมการทำงานของอุปกรณ์ปลายทางจากส่วนกลางได้ โดยระบบจะทำการติดต่อสื่อสารกันผ่านทางเว็ปแอปพลิเคชันเมเนทเมนต์ (Web Application Management) ที่พัฒนาฟังก์ชันทำงานการตรวจสอบเกี่ยวกับเรื่องการทดสอบแบนด์วิธเพอฟอร์แมนซ์นั้นจะมีการจำลองการทดสอบบนเครือข่ายอินเตอร์เน็ตบอร์ดแบรนด์ ที่มีการทดสอบในส่วนของการใช้งาน HTTP , PING และการควบคุมการทำงานของไฟร์วอล์จะเป็นการกำหนดให้ Firewall ที่ Client ทำงานได้ตามที่ Server กำหนดได้ โดยใช้คำสั่งเพื่อควบคุมการทำงานผ่านทางส่วนกลาง รวมถึงการทำงานเกี่ยวกับการเก็บข้อมูล(Backup) ของอุปกรณ์ปลายทาง และการนำข้อมูลกลับไปใช้งาน (Restore) ได้ โดยจะต้องสามารถกำหนดการทำงานต่างๆ ตามวันและ เวลาที่กำหนดผ่านจากส่วนกลางได้ทั้งหมด

1.3 ขอบเขตในการพัฒนาระบบ

การพัฒนาโปรแกรมจะเป็นการพัฒนาโปรแกรมให้กับระบบทั้งส่วนของตัว Client และ Server โดยระบบจะต้องสามารถทำงานได้ดังนี้

- Client จะต้องสามารถ Auto power on after power lost (สามารถบูทเครื่องได้เอง ในกรณีเกิดไฟฟ้าดับ)
- Server สามารถป้อนข้อมูลที่จำเป็นสำหรับการกำหนดการทำงานของเครื่องปลายทางได้ผ่าน Web Application ผ่าน User Interface เช่นตั้งค่าคำสั่ง เพื่อให้การทดสอบการเรียกใช้งานของฟังก์ชั่น HTTP และ PING
- สามารถเก็บข้อมูล (Backup) ของกฎของไฟร์วอลล์เพื่อใช้งานใหม่ ผ่านทาง secure channel ได้
- สามารถแสดง Graph Traffic ของข้อมูลที่เครื่องปลายทางทดสอบได้ผ่านทาง Web Application
- Client จะต้องมีความสามารถในการเก็บข้อมูล การทำงานเพื่อทำการทดสอบคุณภาพของเครือข่ายได้โดยจะต้องทำการเก็บข้อมูล ที่ถูกส่งการทำงานมาจาก Server ได้
- Client จะต้องสามารถทำการทดสอบการทำงานตามฟังก์ชัน ที่กำหนดมาจาก server ได้ โดยจะมีฟังก์ชันการทำงาน คือ PING และ HTTP
- ทั้ง Client และ Server จะมีการทำงานในการส่งผ่านข้อมูลจะต้องเป็นไปตามเงื่อนไขการทำงานที่ได้กำหนดไว้คือ Client จะส่งข้อมูลให้ Server นั้น Client จะต้องส่งชื่อ Node ของตัวเองออกไปด้วย
- Client จะต้องทำงานได้อย่างอัตโนมัติ โดยจะต้องทำการทดสอบตามที่ Server กำหนด จนกว่าจะได้ทำการ Update การทำงานอีกครั้งจาก Server ถึงจะมีการเปลี่ยนแปลงการทำงาน

1.4 องค์ประกอบของระบบงาน

ระบบงานประกอบด้วยองค์ประกอบต่างๆ ดังต่อไปนี้

เครื่องคอมพิวเตอร์ส่วนที่เป็นเซิร์ฟเวอร์

- ติดตั้งระบบปฏิบัติการ Linux
- ติดตั้ง PHP
- ติดตั้งส่วนที่ให้บริการเว็ปเซอร์เวอร์ Apache web server
- ติดตั้งซอร์ฟแวร์ระบบฐานข้อมูลโดยใช้ MySQL
- ติดตั้งซอร์ฟแวร์ไฟร์วอลล์

เครื่องคอมพิวเตอร์ส่วนที่เป็นอุปกรณ์ปลายทาง(Client)

1.5 ขั้นตอนในการพัฒนาระบบ

ศึกษาความเป็นไปได้ในการพัฒนาระบบ เพื่อกำหนดขอบเขตของปัญหาและวางแผนวิธีการพัฒนาโปรแกรม และกำหนดเป้าหมายในการพัฒนาโครงการ

- ศึกษาติดตั้งและการใช้งานซอร์ฟแวร์ต่างๆ ที่ใช้สำหรับการทำโครงงาน
- ได้แก่ Linux, Apache Web Server , PHP , MySQL
- ศึกษาการใช้งานโปรแกรมฐานข้อมูล MySQL เพื่อเก็บข้อมูลของระบบ
- ศึกษาการใช้งานโปรแกรมภาษา PHP
- ศึกษาเทคโนโลยีการทำงานของ HTTP, PING เพื่อใช้ในการคำนวณค่าแบนด์วิต
- ศึกษาการทำงานของไฟร์วอลล์
- ศึกษาการออกแบบ การวิเคราะห์แบบจำลองเชิงแนวคิดของระบบโดยอาศัย Unified Modeling Language (UML)
- ศึกษาการทำงานของการติดต่อสื่อสารกันผ่านทาง secure channel จากเครื่องส่วนกลางไปยังเครื่องปลายทาง

1.5.1 การวิเคราะห์และการออกแบบ

ทำการวิเคราะห์และออกแบบรวมถึงกำหนดความต้องการของโครงการพัฒนาระบบ โดยได้ออกแบบให้ ระบบสามารถเพิ่มหรือลบ คำสั่ง และป้อนข้อมูลที่จำเป็นสำหรับการกำหนดคำสั่ง เช่นการ ตั้งค่าทดสอบฟังก์ชันการเรียกใช้ HTTP ของ www.google.com รวมถึงการเก็บข้อมูล (Backup) การนำข้อมูลกลับ(Restore) และการแสดงผลการทำงานของเครื่องอุปกรณ์ปลายทาง

1.5.2 การพัฒนาและทดสอบ

- ทำการพัฒนาโปรแกรมทั้งในส่วนของเครื่องส่วนกลาง(Server) และเครื่องปลายทาง
- ทำการติดตั้งโปรแกรม บนส่วนของอุปกรณ์ปลายทาง เพื่อทำการเก็บข้อมูลของ Bandwidth ของข้อมูลที่ต้องการทดสอบ
- ทำการทดสองการทำงานตามฟังก์ชันการทำงานต่างๆ

1.5.3 การทดสอบการใช้งานและปรับปรุงแก้ไข

นำโปรแกรมมาทดลองการใช้งานและปรับปรุงแก้ไขเพื่อให้ทำงานได้อย่างถูกต้องและง่ายยิ่งขึ้น

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- ได้พัฒนาความรู้ความเข้าใจเรื่องการทำงานพื้นฐานของฟังก์ชัน PING, HTTP
- ได้พัฒนาความรู้ความเข้าใจเรื่องการทำงานของไฟร์วอลล์
- ได้พัฒนาความรู้ความสามารถในการวิเคราะห์ ออกแบบและพัฒนาระบบงานและสามารถนำไปใช้ประโยชน์ต่อการทำงานในอนาคต
- ได้โปรแกรมเพื่อใช้ทดสอบการทำงานของเครื่องปลายทาง โดยใช้ทดสอบฟังก์ชันการทำงานของ PING, HTTP และการตั้งค่าไฟร์วอลล์ของเครื่องปลายทางผ่านทางส่วนกลาง
- สามารถนำไปใช้ประยุกต์การทำงานอื่นๆ ได้เช่น การตั้งค่าการทำงานอื่นๆ ของเครื่องปลายทางเช่น การตั้งค่า Proxy หรือฟังก์ชันอื่นๆ

บทที่ 2
การทำงานระหว่างเครื่องไคล์เอนต์และเซิร์ฟเวอร์

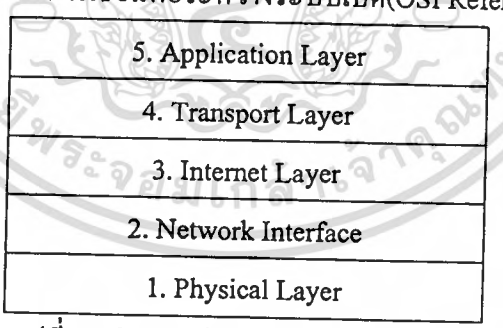
ในหัวข้อนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องในการวิจัยการทำงานของระบบการจัดการการทำงานระหว่างเครื่องไคล์เอนต์และเซิร์ฟเวอร์ การจัดการการทำงานไฟร์วอลล์ การทำงานของระบบเว็ปบราวเซอร์ และการจัดการต่างๆ

2.1 โปรโตคอลทีซีพี/ไอพี(TCP/IP protocol)

แรกเริ่มโปรโตคอลทีซีพี/ไอพี เป็นโปรโตคอลที่ใช้ในการสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ที่เชื่อมต่อกันในระบบยูนิกซ์ (Unix) สำหรับปัจจุบันนี้โปรโตคอลทีซีพี/ไอพีมีการใช้งานในเครื่องคอมพิวเตอร์ทุกรุ่นทำแบบ ทำให้เครื่องคอมพิวเตอร์แบบใดๆก็ตามที่ทำงานกับซอฟต์แวร์โปรโตคอลทีซีพี/ไอพีก็สามารถเชื่อมเข้าในเครือข่ายโปรโตคอลทีซีพี/ไอพีได้ และแต่ละเครือข่ายก็สามารถเชิ่มโยงกันทำให้กลายเป็นเครือข่ายอินเตอร์น็ตในที่สุด

2.1.1 โครงสร้างของโปรโตคอลทีซีพี/ไอพี

โปรโตคอลทีซีพี/ไอพีเป็นชุดโปรโตคอลที่ประกอบด้วยไอพี (IP: Internet Protocol), ทีซีพี (TCP : Transmission Comtrol Protocol), ยูดีพี (UDP: User Datagram Protocol) ฯลฯ โครงสร้างของโปรโตคอลทีซีพี/ไอพี แสดงอยู่ในภาพที่2.1 มีจำนวน 5 ชั้น สอดคล้องกับชั้นต่างๆ ของแบบ จำลองอ้างอิงสำหรับการเชื่อมต่อระหว่างระบบเปิด(OSI Reference Model) ดังนี้



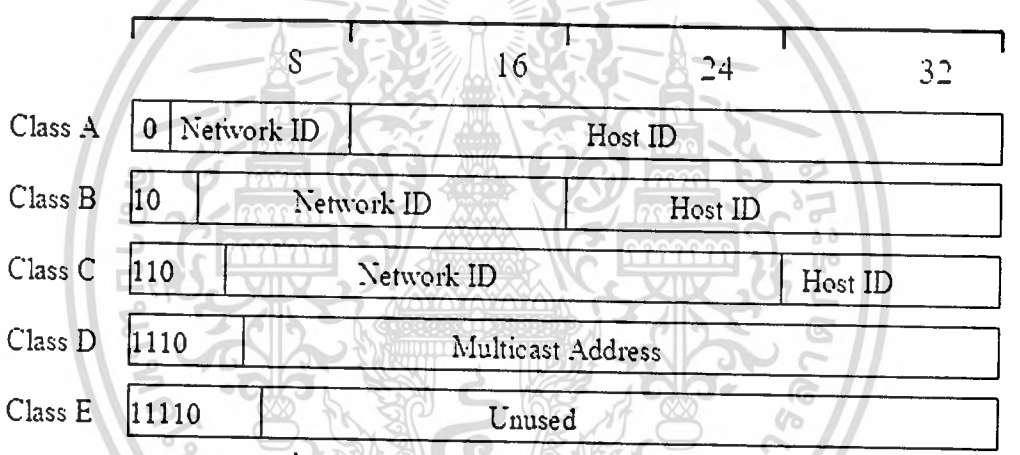
รูปที่ 2.1 โครงสร้างโปรโตคอลทีซีพีไอพี

- **Layer 1: Physical** ในชั้นที่ จะเกี่ยวข้องกับอุปกรณ์เครือข่ายพื้นฐานอย่างเดียว ซึ่งสอดคล้องกับชั้นที่ 1 ของโมเดลการลำดับชั้นมาตรฐาน
- **Layer 2: Network Interface** โปรโตคอลของชั้นที่ 2 จะเกี่ยวข้องกับการจัดข้อมูลลงในเฟรม(Frame) และการส่งเฟรมต่างๆ ข้ามเครือข่ายซึ่งคล้ายกับการทำงานในชั้นที่ 2 ของโมเดลการลำดับชั้นที่เป็นมาตรฐาน
- **Layer 3: Internet** โปรโตคอลต่างๆในชั้นที่ 3 จะกำหนดรูปแบบของแพ็คเก็ตที่จะส่งข้ามเครือข่ายอินเตอร์เน็ต การทำงานจะคล้ายกับกลไกที่ใช้ในการส่งแพ็คเก็ตจาก

คอมพิวเตอร์เครื่องกหนึ่งไปยังเครื่องปลายทางผ่านเราท์เตอร์(router)

- **Layer 4: Transport** การทำงานของโปรโตคอลต่างๆในชั้นที่ 4 จะคล้ายกับการทำงานของชั้นที่ 4 ของโมเดลการลำดับชั้นที่เป็นมาตรฐาน คือ ได้มีการกำหนดถึงการรับรองความไว้ใจในการส่งผ่านข้อมูล
- **Layer 5: Application** ในลำดับที่ 5 จะสอดคล้องกับชั้นที่ 6 และชั้นที่ 7 ของโมเดลการลำดับชั้นมาตรฐานโดยโปรโตคอลแต่ละตัวจะมีการระบุถึงแอปพลิเคชั่นต่างๆ (Applicaions) ที่จะใช้ในอินเตอร์เน็ต

โปรโตคอลที่อยู่ในชั้นอินเตอร์เน็ตจะประกอบด้วยโปรโตคอลหลายตัวเช่น ไอพี ไอซีเอ็มพี(ICMP: Internet Control Message Protocol) โปรโตคอลเกตเวย์ (Gateway Protocol) ฯลฯ แต่ในเนื้อหาจะกล่าวถึงไอพีเพียงอย่างเดียวเท่านั้น



รูปที่ 2.2 หมายเลขเครื่องอินเตอร์เน็ตทั้ง 5 ประเภท

2.1.2 ไอพี

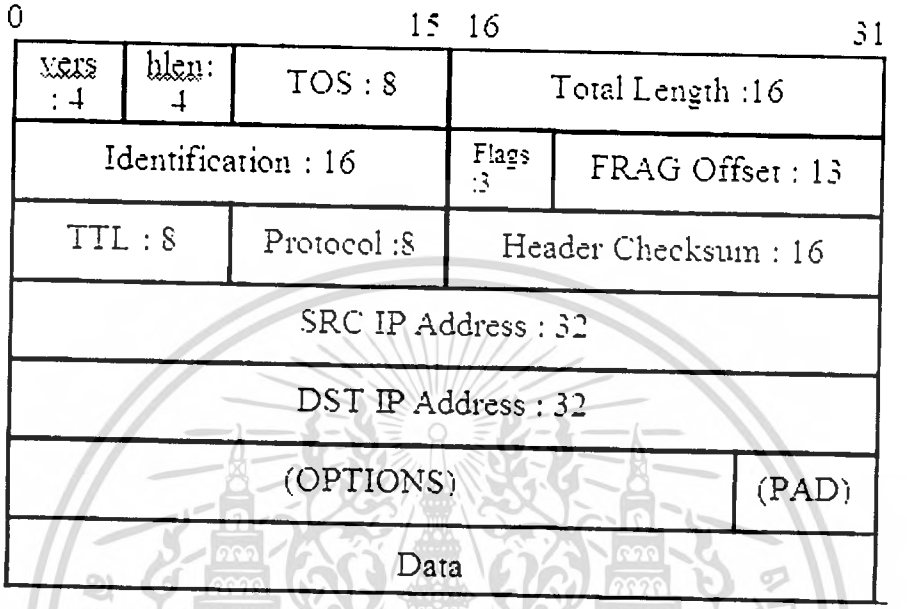
ไอพีเป็นโปรโตคอลระหว่างเครือข่าย (Internetworking Protocol) ที่ถูกพัฒนาโดยกระทรวงกลาโหมสหรัฐอเมริกา ระบบต่างๆได้ถูกนำพัฒนาให้เป็นส่วนหนึ่งของโครงการดาร์ปาอินเตอร์เน็ตเวิร์คโปรโตคอล (CAPRA internetwork protocol) ซึ่งเป็นระบบที่มีการใช้ทั่วโลก

- **การทำงานของส่วนไอพี**

เนื่องจากไอพีคือขั้นตอนของการส่งข้อมูลระหว่างเครือข่าย จะมีหมายเลขเครื่องในระบบอินเตอร์เน็ต (Internet Address) ซึ่งเป็นตัวกำหนดว่าจะส่งข้อมูลไปที่ส่วนใดในเครือข่ายลักษณะของหมายเลขเครื่องในระบบอินเตอร์เน็ตแต่ละหมายเลขมี 32 บิต และได้มีการแบ่งหมายเลขออกเป็น 2 ส่วนคือส่วนที่เป็นหมายเลขเครือข่าย (Prefix) และส่วนที่เป็นหมายเลขประจำเครื่อง (Suffix)

ส่วนหมายเลขเครือข่ายจะกำหนดโดยหน่วยงาน interNIC เพื่อให้ไม่มีการซ้ำซ้อนกัน ส่วนหมายเลขประจำเครื่องเป็นเลขที่กำหนดโดยผู้บริหารเครือข่ายให้กับคอมพิวเตอร์ที่อยู่ภายในเครือข่าย

แต่หมายเลขที่ปรากฏก็จะเป็นอย่างใดอย่างหนึ่งในหน้าหนึ่งใน 5 ประเภท ซึ่งแต่ละประเภทก็ต่างกันที่ขนาดของหมายเลขประจำเครื่องนั้นเอง ในรูปที่2,2 จะแสดงหมายเลขเครื่องในระบบอินเตอร์เน็ตทั้ง 5 ประเภท และไอพีได้กำหนดไอพีดาต้าแกรม ซึ่งหน่วยพื้นฐานสำหรับการส่งผ่านข้ามไปในดปรโตคอลทีซีพี/ไอพี ดังแสดงในรูปที่2.3



รูปที่ 2.3 รูปแบบของไอพีดาต้าแกรม[1]

โปรโตคอลในชั้นทรานสปอร์ตมีหลายดโปรโตคอล โปรโตคอลที่เราสนใจคือ ทีซีพี และยูดีพี โดยที่ทีซีพีเป็นการติดต่อสื่อสารแบบมีการเชื่อมต่อ (Connection-oriented) และยูดีพีเป็นการติดต่อแบบไม่มีการเชื่อมต่อ (Connectionless)

2.1.3 ทีซีพี

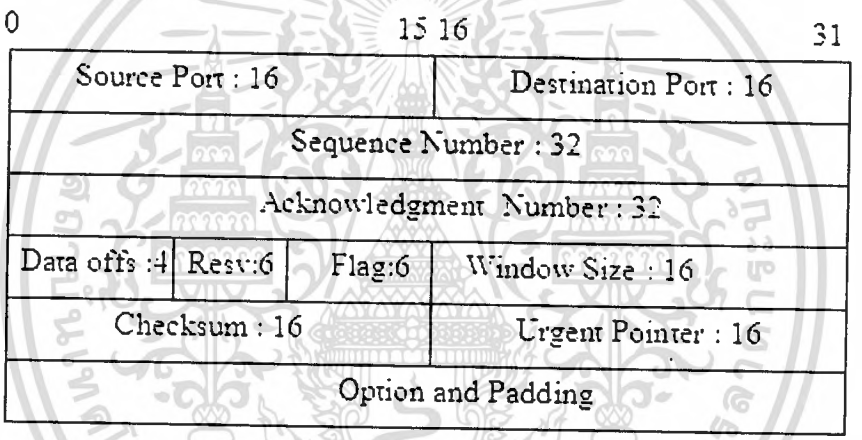
โปรโตคอลทีซีพีจะกำหนดช่วงเวลสำหรับการติดต่อเพื่อยืนยันการรับ-ส่งข้อมูลระหว่างคอมพิวเตอร์สองเครื่อง ทำให้โปรโตคอลทีซีพีเป็นโปรโตคอลที่มีความน่าเชื่อถือ (Reliable) เพราะให้ความแน่นอนว่าแพ็คเก็ตข้อมูลที่ถูกส่งออกไปจากต้นทางจะไปถึงยังปลายทางอย่างเป็นลำดับ และไม่เกิดการผิดพลาด หรือการสูญหายของข้อมูล ดังนั้นโปรโตคอลทีซีพีเป็นโปรโตคอลสำหรับควบคุมการสื่อสาร กำหนดตำแหน่งต้นทาง และปลายทาง และอื่นๆ กับข้อมูล

การให้บริการของ ทีซีพีประกอบด้วยบริการต่างๆ ดังต่อไปนี้ คือ Connection-oriented service, Reliable transport service ซึ่งถ้าแอพพลิเคชั่นใดๆ ก็ตามที่ใช้ TCP เป็น Transport protocol ก็จะได้รับบริการครบทั้ง 2 อย่าง ซึ่งมีรายละเอียด ดังนี้

- Connection-oriented service: แอพพลิเคชั่นที่ใช้ TCP นั้นจะต้องมีการแลกเปลี่ยน Transport layer Control Information กันก่อนที่จะทำการแลกเปลี่ยน message กันซึ่งการทำการแลก information ระหว่างกันนี้ เราเรียกว่า กระบวนการการทำ

handshaking -ซึ่ง หลังจาก เสร็จขั้นตอนการ handshaking แล้ว TCP Connection จะเกิดขึ้นระ หว่าง socket ของทั้ง 2 process และ process จะสามารถส่ง message ระหว่างกันได้ โดยเมื่อแอพพลิเคชั่นทำการส่ง messageทั้งหมดเสร็จสิ้น แล้ว connection นั้นก็จะถูก ปิดลง

- Reliable transport service: Process ที่มีการติดต่อสื่อสารกันโดยใช้ TCP นั้นจะวางใจได้ว่าข้อมูลที่ ทำการส่งไปยังฝั่งรับนั้นจะไม่เกิด error และข้อมูลจะถูกส่งเรียงตามลำดับด้วยนอกจากนี้ TCP ยังให้บริการอื่นๆ อีกเช่น Congestion control ซึ่งเป็นบริการ ที่ TCP เตรียมไว้เพื่อควบคุมไม่ให้ฝั่งส่ง ทำการส่งข้อมูลมากเกินไป ในขณะที่ network เกิดความคับคั่ง และยังมีบริการ Flow control อีกด้วย ซึ่งเป็นบริการที่ใช้เพื่อควบคุมไม่ให้ฝั่งส่ง ส่งข้อมูลมากเกินความสามารถในการรับไปประมวลผลของฝั่งรับ

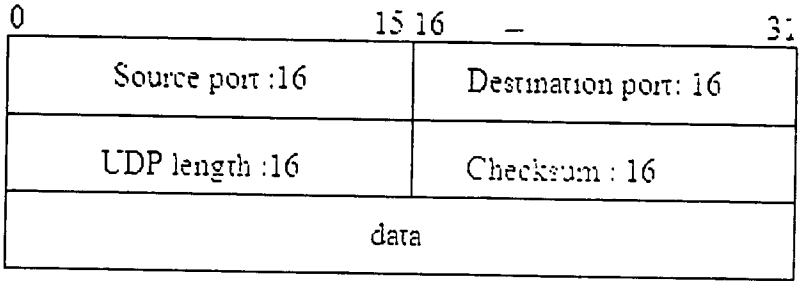


รูปที่ 2.4 รูปแบบของเซ็กเมนต์

2.1.4 ยูดีพี

โปรโตคอลยูดีพีคือโปรโตคอลที่ทำหน้าที่ควบคุมการรับ-ส่งข้อมูลโดยไม่มีการรอคอยการยืนยันการตอบรับจากปลายทาง ทำให้การบริการแบบนี้ให้ความน่าเชื่อถือน้อยกว่า แต่ก็ทำให้การสื่อสารข้อมูลเป็นไปอย่างรวดเร็วมากขึ้นถ้าไม่มีความผิดพลาดเกิดขึ้นระหว่างการรับ-ส่งข้อมูล (Comer:1995)

ในลำดับชั้นตามรูปที่2.1 ยูดีพีจะอยู่เหนือไอพีเนื่องจากยูดีพีเป็นการสื่อสารแบบไม่มีการเชื่อมต่อ และยูดีพีจะเพิ่มความสามารถในการหาที่อยู่ของพอร์ต (port) ให้กับไอพีโดยจะพิจรณาที่ส่วนหัวของยูดีพีดังแสดงในรูปที่ 2.5



รูปที่ 2.5 รูปแบบของยูดีพีดาต้าแกรม

2.1.5 โปรโตคอลที่เกี่ยวข้องกับฟังก์ชันการทำงานของโครงงาน

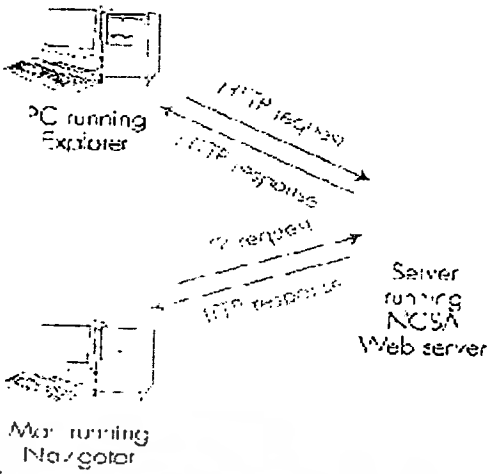
- **HTTP (Hypertext Transfer Protocol)**

HTTP เป็น protocol ใน Application Layer ซึ่งเป็นหัวใจสำคัญของเว็บ ซึ่งถูกแบ่งการทำงาน ออกเป็น 2 ส่วน คือ Client program และ Server program โดยอาจจะมีการทำงานบนระบบที่ต่างกัน มีการ แลกเปลี่ยน HTTP message กันด้วย ซึ่ง HTTP นั้นจะทำการกำหนดรูปแบบของข้อมูล (message) และวิธีการแลกเปลี่ยนข้อมูลระหว่าง client และ server

Web page ประกอบด้วยออบเจ็กต์ (Object) คือ ไฟล์ประเภทต่างๆ เช่น HTML, JPEG, GIF, Java applet, audio clip เป็นต้น ซึ่งไฟล์เหล่านี้จะมี URL เป็นตัวบอกที่อยู่ของไฟล์และจะเก็บอยู่ในรปแบบของไฟล์ HTML สมมติว่าเว็บเพจประกอบด้วยไฟล์รูป JPEG 5 ไฟล์และเท็กซ์ไฟล์ HTML 1 ไฟล์ ดังนั้นเว็บเพจนี้จะประกอบด้วย 6 ออบเจ็กต์ ทำให้เวลาที่เว็บเพจจะอ้างถึงออบเจกต์ต่าง ๆ จึงต้องมีการใช้ด้วย URL ของแต่ละออบเจ็กต์นั่นเอง

ในแต่ละ URL จะประกอบด้วย 2 ส่วน คือ Host name ของ server และ path ของออบเจ็กต์นั้นๆ ตัวอย่างเช่น www.someSchool.edu/someDepartment/picture.gif โดยมี www.someSchool.edu เป็น Host name และ /someDepatment/picture.gif เป็น path name เป็นต้น

ส่วน Browser ก็คือโปรแกรมที่ทำหน้าทติดต่อระหว่างผู้ใช้ (user) และ web server ที่เป็นที่นิยมก็ ได้แก่ Netscape Communication และ Microsoft Internet Explorer นอกจากนี้ Browser ยังทำหน้าที่ implement HTTP ของฝั่ง Client อีกด้วย ในขณะที่ Web server นั้นที่เก็บของอ็อบเจกต์ ต่างๆ ซึ่งในแต่ละ ออบเจ็กต์จะระบุที่อยู่โดย URL ซึ่ง Web Server ที่เป็นที่นิยมได้แก่ Apache, Microsoft Internet Information Server และ Netscape Enterprise Server โดยที่ Web Server จะทำหน้าที่ implement HTTP ของฝั่ง Server ด้วย เมื่อผู้ใช้ตองการข้อมูลของเว็บเพจ (เช่น คลิก Hyperlink) browser จะส่ง HTTP request เพื่อขอ รายละเอียดของ web page คือออบเจ็กต์ต่าง ๆ ไปยัง server เมื่อ server ได้รับ request จะส่ง HTTP response message ซึ่งมีออบเจ็กต์ต่าง ๆ ไป ด้วย



รูปที่ 2.6 HTTP request-response behavior

จากรูป 2.6 แสดงการติดต่อสื่อสารกันระหว่าง client และ server เมื่อ user ร้องขอเว็บเพจ browser จะทำการส่ง HTTP request message ไปยัง server และเมื่อ server ได้รับ request ก็จะส่ง HTTP response message และออบเจ็กต์ที่ร้องขอกลับไปยัง client

HTTP เป็น protocol ใน Application layer โดยใช้บริการของ TCP ดังนั้น browser ต้องทำการสร้าง connection โดยการสร้าง socket ขึ้นมาและติดต่อไปยัง server ที่ port เบอร์ 80 ทางด้าน server เมื่อได้ รับคำร้องขอ connection ก็จะตอบกลับไปยัง client เพื่อเริ่มการส่ง message ต่อไป โดย message จะมีการ แลกเปลี่ยนกันระหว่าง browser กับ server และเมื่อทำการแลกเปลี่ยน message ทั้งหมดกันเสร็จเรียบร้อยแล้วก็จะทำการปิด connection นั้น

HTTP เป็น protocol แบบ stateless คือมันจะไม่เก็บรักษาสถานะต่างๆ ของ client เอาไว้ เช่น Client ใดทำการร้องขอเว็บเพจมาเมื่อไร ร้องขออะไรบ้าง เป็นต้น

Non-Persistent และ Persistent Connections

HTTP สามารถใช้ได้ทั้ง non-persistent connections และ persistent connections โดย Nonpersistent connections เป็น default mode ของ HTTP/1.0 (Version 1.0) ส่วน persistent connections เป็น default mode ของ HTTP/1.1 (Version 1.1)

Non-Persistent Connection

พิจารณาขั้นตอนของการส่งเว็บเพจจาก server มายัง client สำหรับกรณีของ non-persistent connections นั้นสมมติว่าเว็บเพจประกอบด้วย HTML file และรูปภาพชนิด JPEG 10 รูป และ URL เป็นดังนี้ www.someSchool.edu/someDepartment/home.html

เหตุการณ์ที่เกิดขึ้น คือ

1. HTTP client สร้าง TCP connection ไปยัง HTTP server www.someSchool.edu โดยใช้ port 80

2. HTTP client ส่ง HTTP request message ไปยัง TCP socket

3. HTTP server รับ request message ผ่านทาง socket ที่ถูกสร้างขึ้นมาตอนสร้าง connection แล้วก็ทำการดึงออบเจ็กต์ /someDepartment/home.html จากหน่วยความจำ (RAM หรือ DISK) แล้วทำการ encapsulate ออบเจ็กต์ใน HTTP response message และส่ง response message ไปใน TCP connection

4. HTTP server ติดต่อกับ TCP เพื่อทำการปิด connection (แต่ TCP จะไม่ทำการปิด connection จนกว่า client จะได้รับ response message)

5. HTTP client ได้รับ response message กลับมาและ TCP connection ถูกปิดแล้ว มันจะทำการ แสดงผล HTMLและทำการ parsing เพื่อหาออบเจ็กต์อื่นๆ ซึ่งอยู่ในเว็บเพจนั้นอีก

6. ทำ 4 ขั้นตอนแรกซ้ำ สำหรับแต่ละ JPEG object

จะเห็นได้ว่า ขั้นตอนดังกล่าวจะใช้หลักการของ Non-persistent Connections เนื่องจากแต่ละ TCP connection ถูกปิดภายหลังจาก server ส่งออบเจ็กต์ไปยัง client แล้วโดยแต่ละ TCP connection จะทำการ รับ-ส่ง เพียง 1 request message และ 1 response message เท่านั้น จากตัวอย่าง เมื่อ user ร้องขอเว็บเพจนี้ จะมีการสร้าง TCP connection ทั้งหมด 11 ครั้ง

เวลาที่ Packet ใช้ในการเดินทางจาก client ไปยัง server และจาก server กลับมายง client เรียกว่า Round-trip Time (RTT) ซึ่ง RTT จะรวมถึง Propagation delay, queuing delay ที่เกิดขึ้นใน router และ switch และ Processing delay ด้วย ลองพิจารณาเหตุการณ์ที่เกิดขึ้นเมื่อ User ทำการ click hyperlink แล้ว browser จะทำการสร้าง TCP connection ระหว่าง browser และ web server ซึ่งจะต้องทำ "handshake" โดย client จะทำการส่ง TCP message ไปยัง server หลังจากนั้น server ทำการแจ้งว่าได้รับ message นั้นแล้ว และทำการตอบกลับแล้วดวย สุดท้าย client จะทำการแจ้งกลับไปยัง server ว่าได้รับ message เรียบร้อยแล้วและร้องขอ object ไปด้วย ซึ่ง Server ก็จะทำการส่ง object นั้นมา ซึ่งจะเห็นว่ามีการใช้เวลาในการที่ client ส่ง request message ไปยัง server และ server สง HTML file กลับมา นั้นเท่ากับ 2 RTT โดย RTT แรกจะใช้ในตอนเริ่มตดตั้ง TCP connection และ RTT ที่ 2 จะใช้ตอนทำการร้องขอออบเจ็กต์และได้ ออบเจ็กต์กลับมา

Persistent Connections

ใน Persistent connections การร้องขอและตอบกลับระหว่าง client และ server เดียวกันจะทำผ่าน connection เดียวกันเท่านั้น จากตัวอย่างที่กล่าวมาแล้ว การรับ-ส่ง HTML file และรูปภาพ 10 รูป สามารถ ทำโดยใช้ persistent TCP connection 1 connection เท่านั้น

Persistent connection มีอยู่ 2 version คือ แบบ without pipelining และแบบ with pipelining สำหรับแบบ without pipelining นั้น client จะทำการร้องขอครั้งออบเจ็กต์ตัวใหม่ได้ ก็ต่อเมื่อ server ตอบกลับการร้องขอครั้งที่แล้วมาแล้วเท่านั้น ซึ่งจะใช้เวลา 1 RTT ในการ request และreceive ออบเจ็กต์ แต่มี ข้อเสียคือ ระหว่างการส่งออบเจ็กต์ ถ้าเกิด connection ติดค้างไม่สามารถทำงานต่อได้จะทำให้เกิดการรอออบเจ็กต์นั้นๆ อยู่ตลอดทำให้ไม่สามารถส่ง request ใหม่ออกไปได้ ซึ่งเหตุการณ์นีจะทำให้เปลืองทรัพยากรของ server ส่วนอีก version หนึ่งก็คือ แบบ

with pipelining ซึ่ง Client จะไม่รอจนกว่าจะได้รับ response ที่ ส่งไปแต่มันจะส่ง request ไปทันทีที่สามารถจะส่งออกไปได้ (เป็นลักษณะของการส่ง Request ต่อเนื่องกันไปโดยไม่ต้องรอ Response) ซึ่งใน HTTP/1.1 จะใชหลักการทำงานของ persistent connections with pipelining

HTTP message format

ทั้ง HTTP/1.0 และ HTTP/1.1 กำหนดรูปแบบของ HTTP message ไว้ 2 แบบด้วยกันคือ HTTP request message และ HTTP response message HTTP Request Message

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: Close
User-agent: Mozilla/4.0
Accept-language: fr
(Extra carriage return, line feed)
```

รูปที่ 2.7 HTTP Request Message[2]

จะเห็นว่า Message ประกอบด้วย 5 บรรทัด โดยแต่ละบรรทัดจะจบด้วย carriage return, line feed ซึ่งจริงแล้วใน message อาจมีมากกว่า 5 บรรทัด หรือมีเพียง 1 บรรทัดก็ได้ โดยบรรทัดแรกใน HTTP request message เรียกว่า request line ส่วนบรรทัดอื่น ๆ เรียกว่า header line ใน request line จะประกอบด้วย 3 field ได้แก่ method field, URL field และ HTTP version field ซึ่ง HTTP request message ส่วนใหญ่จะใช้ GET method โดยถ้าใช้ GET method ข้อความที่ user กรอกใน Form ในเว็บเพจจะปรากฏ อยู่ข้างหลัง URL และถ้ายาวเกินไปจะถูกตัดออกไป แต่ถ้าใช้ Post method ข้อความที่ user กรอกจะปรากฏ ในส่วน body ของ request message

ในส่วนของ Header line ในตัวอย่างนั้นจะมีความหมาย ดังนี้

- Host: ใช้ระบุ host (หรือ URL) ที่ทำการเก็บไฟล์ (Object ที่ต้องการ)
- Connection: close นั้น browser จะบอก server ว่ามันไม่ต้องการใช้ persistent connection แต่ต้องการให้ server ทำการปิด connection หลังจาก server ส่งออบเจ็กต์ที่ร้องขอมายัง client แล้ว
- ดังนั้นจะพบว่า browser ที่สร้าง request message นี้ถูก implement โดย HTTP/1.1 แต่ไม่ต้องการ
- ใช้ persistent connection
- User-agent: ใช้ทำการระบุชนิดของ browser ที่สร้างการร้องขอไปยัง server ในกรณีนี้ user agent คือ Mozilla/4.0 ซึ่งก็คือ Netscape browser
- Accept-language: ใช้บอก server ว่าชนิดของออบเจ็กต์ที่ browser สามารถรับได้นั้นเป็นภาษาอะไรบ้าง

HTTP Response Message

```
HTTP/1.1 200 OK
Connection: Close
Date: Thu, 06 Aug 1998 10:00:15 GMT
Server: Apache/1.3.0( Unix )
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html

Data Data Data Data Data Data . ..... . ....
```

รูปที่ 2.8 HTTP Response Message

HTTP response message ประกอบด้วย 3 ส่วน ได้แก่ status line, header line และ Entity Body โดย Entity Body คือเนื้อของ message ซึ่งบรรจุ object ที่ client ร้องขอ

ใน Status line มีทั้งหมด 3 field ได้แก่ protocol response field, status code และ status message ของ status code นั้น จากตัวอย่างนี้ status line HTTP/1.1 200 OK จะบอกว่า server ใช้ HTTP/1.1 และทุกอย่าง OK หมายถึง server พบออบเจ็กต์ที่ร้องขอแล้วและกำลังจัดส่งออบเจ็กต์ไปให้ client

ในส่วนของ Header line server มีรายละเอียด ดังนี้

- Connection: close header line เพื่อบอก Client ว่ามันจะทำการปิด TCP connection ทันที
- หลังจากทำการส่ง message เสร็จสิ้นแล้ว
- Date: header line ระบุเวลาและวันที่ที่ server สร้างและส่ง HTTP response โดยเวลาดัง กล่าวไม่ใช่เวลาที่สร้างออบเจ็กต์หรือเวลาล่าสุดที่แก้ไขออบเจกต์ แต่เป็นเวลาที่ server ดึง
- ออบเจ็กต์ออกมาจากระบบไฟล์ แล้วใส่ออบเจกต์ไปใน response message และส่ง response message
- Server: header line ระบุว่าใช้ Web server อะไร ตามตัวอย่างเป็นการบอก client วาใช้ Apache web server ซึ่งคล้าย User-agent: header line ใน HTTP request message
- Content-Length: header line ระบุจำนวน byte ของออบเจ็กต์ที่จะส่งไป
- Content-Type: header line ระบุว่าออบเจ็กต์ในสวน entity body เป็น text ชนิด HTML

- **ICMP: Internet Control Message Protocol**

ICMP เป็น protocol ที่ถูกใช้โดย hosts, routers และ gateways ในการติดต่อสื่อสารใน network layer ซึ่งกันและกัน รายละเอียดเฉพาะของ ICMP จะมีการกำหนดไว้ใน RFC 792 โดย

ปกติ ICMP จะถูกใช้มากในกรณี ที่มี error reporting เช่น ในขณะใช้ระบบ Telnet, FTP หรือ HTTP เรามักจะพบกับปัญหา error message เช่น ไม่สามารถเข้าสู่ network ปลายทางได้ ข้อความเหล่านี้มาจาก ICMP ในบางกรณที่ IP router ไม่ สามารถหาทางไปสู่ host ตามคำสั่งของ Telnet, FTP หรือ HTTP application ได้ router ก็จะสร้างและส่ง type-3 ICMP message ไปให้ host เพื่อแจงความผิดปกตินั้น เมื่อ host ได้รับ ICMP message แล้ว ก็จะส่ง error code ไปให้ TCP code ที่พยายามติดต่อไปที่ remote host จากนั้น TCP ก็จะส่ง error code กลับไป ให้แอพพลิเคชั่น

ICMP มักถูกมองว่าเป็นส่วนหนึ่งของ IP เพราะ ICMP message จะใช้อยู่ภายใน IP packets นั่นก็ คือ ICMP message บรรจุอยู่ใน IP payload เหมือนๆ กับที่ TCP หรือ UDP packets ที่ถูกบรรจุอยู่ใน IP payload เช่นกัน ในทำนองเดียวกันเมื่อ host ได้รับ IP packet ที่ระบุมี ICMP จะทำการแปลง packet นั้นเป็น ICMP เช่นเดียวกับที่แปลงเป็น packet ให้ TCP หรือ UDP

ICMP messages จะมี field ระบุ type และ code ตัวอย่างของ type, code และ description ของ ICMP message แสดงได้ในรูปที่ 2.9 ICMP messages ไม่เพียงมีไว้เพื่อแจงสัญญาณให้ทราบถึงความ ผิดปกติเท่านั้น ยังมีโปรแกรมที่รู้จักกันดี คือ ping program ที่ใช้ ICMP อีกด้วย โดยคำสั่ง ping จะส่ง ICMP message ที่ระบุ type = 8, code = 0 (echo request) ไปให้ host ผู้รับ เมื่อ host ผู้รับไดรับ message ก็จะ ส่ง ICMP reply ที่ระบุ type = 0, code = 0 เพื่อตอบรับ message ที่ได้รับ

ICMP message ที่น่าสนใจอีกชนิดหนึ่ง คือ source quench message. ซึ่งไม่ค่อยจะได้ใช้บ่อยนักในการทำงานจริง จุดมุ่งหมายดั้งเดิมก็เพื่อใช้ในการควบคุมความคับคั่งของเครือขาย (congestion control) โดยให้ congested router สามารถส่ง ICMP source quench message ไปที่ host เพื่อใหลดอัตราการส่ง แต่ เมื่อเปรียบเทียบกับ TCP จะเห็นว่าใน TCP มีการควบคุมความคับคั่งของมันเอง ซึ่งพร้อมที่จะใช้ได้ที่ transport layer โดยไม่ต้องใช้ network layer มาช่วย เหมือนกับใน ICMP source quench message

Traceroute program ก็เป็นอีก program หนึ่งที่ใช้ในการหาเส้นทางจาก host ต้นทางไป host ปลายทาง โดยใช้ ICMP messages ในการค้นหาชื่อและที่อยู่ของ routers ระหว่างต้นทางถึงปลายทาง ซึ่ง traceroute ต้นทางจะส่ง IP datagram ไปยังปลายทาง โดย datagram อันแรกจะมีค่า TTL = 1 ส่วน datagram อันต่อๆไป ก็จะมีค่าเพิ่มขึ้นตามลำดับของ datagram นั้นๆ ต้นทาง ก็จะเริ่มจบเวลาของแต่ละ datagrams เมื่อ datagram อันที่ N ไปถึง router ที่ N แล้ว router นั้น หากเห็นว่า TTL ของ datagram เพิ่ง หมดอายุไป ก็จะทำการทิ้ง datagram นันไป จากนั้นก็จะส่ง ICMP warning message ไปที่ต้นทาง (type 11 code 0) ซึ่ง message นี้ จะมีชื่อของ router พร้อมกับ IP address เมื่อต้นทางได้รบ ICMP message ตอบ

ICMP Type	Code	Description
0	0	Echo reply (to ping)
3	0	Destination network unreachable
3	1	Destination host unreachable
3	2	Destination protocol unreachable
3	3	Destination port unreachable
3	6	Destination network unknown
3	7	Destination host unknown
4	0	source quench (congestion control)
8	0	Echo request
9	0	Router advertisement
10	0	Router discovery
11	0	TTL expired
12	0	IP header bad

รูปที่ 2.9 ICMP Message

2.2 ไฟร์วอลล์

ไฟร์วอลล์ได้ถูกออกแบบมาเพื่อจัดการกับการเข้ามา(incoming)และการออกไป (outgoing) ของการจราจร(traffic) โดยตัวของระบบ โดย Firewall สามารถกำหนดกลุ่มของกฎเพื่อจัดการกับแพ็คเก็ต ที่เข้ามา หรือ ออกจากระบบเน็ตเวอร์ค ที่มีการเชื่อมต่ออยู่ โดยไฟร์วอลล์ส่วนใหญ่จะทำหน้าที่เป็น เราท์เตอร์ และจะทำการกรองข้อมูลที่ผ่านเข้ามา หรือส่งออกไปโดยอาศัยนโยบายในการรักษาความปลอดภัย หรือตัดสิน ใจของผู้จัดการให้การดูแลระบบเครือข่าย

ในปัจจุบันมีการแบ่งชนิดของไฟร์วอลล์ตามเทคโนโลยีที่ใช้ในการตรวจสอบและควบคุมได้เป็น 3 ชนิดคือ

- ไฟร์วอลล์ชนิดกรองแพ็คเก็ต (Packet Filtering)
- พร็อกซี่บริการ (Proxy Service)
- Stateful Multilayer Inspection Firewall

2.2.1 ไฟร์วอลล์ชนิดกรองแพ็คเก็ต (Packet Filtering)

เป็นรูปแบบการทำงานที่ใช้ในการควบคุมการไหลของข้อมูลซึ่งการกรองแพ็คเก็ตจะอนุญาตหรือปฏิเสธแพ็คเก็ตเหล่านั้น โดยจะพิจารณาจากข้อมูลส่วนที่อยู่ในเฮดเดอร์ (Header) ของแพ็คเก็ตที่จะผ่านเข้ามาเทียบกับกฎ (Rule) ที่กำหนดไว้ ในการพิจารณาเฮดเดอร์ Packet Filter จะ

ทำการตรวจสอบในระดับของอินเตอร์เน็ตเลเยอร์ (Internet Layer) และทรานสปอร์ตเลเยอร์ (Transport Layer) ซึ่งในอินเตอร์ เน็ตเลเยอร์จะมีแอตทริบิวต์ ที่สำคัญต่อ Packet Filtering ดังนี้

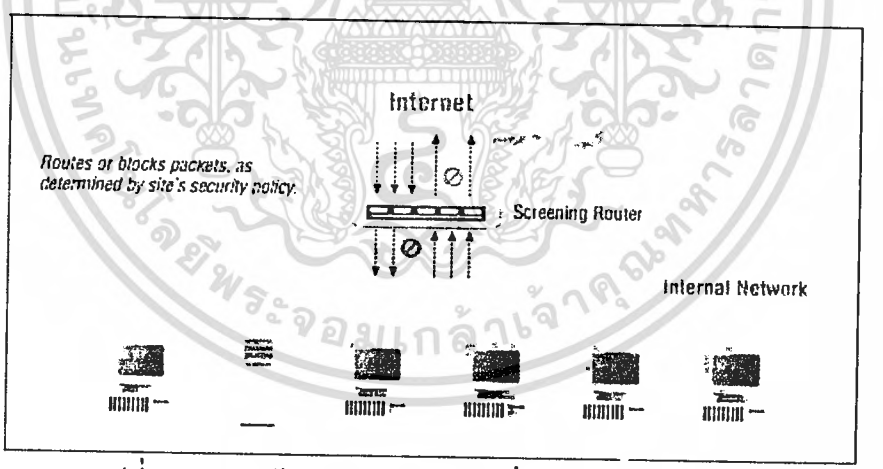
- ไอพีแอดเดรสต้นทาง
- ไอพีแอดเดรสปลายทาง
- ชนิดของโปรโตคอล

ในทรานสปอร์ตเลเยอร์มีแอตทริบิวต์ ที่สำคัญคือ

- พอร์ตต้นทาง (TCP/UDP)
- พอร์ตปลายทาง (TCP/UDP)
- แฟล็ก (Flag ซึ่งจะมีเฉพาะในเฮดเดอร์ของแพ็คเก็ต TCP)
- ชนิดของ ICMP message (ในแพ็คเก็ต ICMP)

การกรองแพ็คเก็ตอาจจะพบในอุปกรณ์ต่างๆ เช่น ในเราท์เตอร์ บริดจ์ หรือบนเครื่องโฮสท์เองซึ่งเราอาจเรียกวิธี การนี้ว่า "Screening"

ระบบการกรองแพ็คเก็ตจะทำการส่งแพ็คเก็ตระหว่างเครือข่ายทั้งภายในเครือข่ายและภายนอกเครือข่าย โดยในรูปที่ 2.10 จะแสดงการไหลของแพ็คเก็ตโดยจะเป็นการแสดงชนิด ของเราเตอร์ที่ทำหน้าที่เป็นไฟร์วอลล์แบบกรองแพ็คเก็ตหรือเรียกว่า "Screening Router"



รูปที่ 2.10 การใช้ Screening Router เพื่อทำการกรองแพ็คเก็ต

จากรูปที่ 2.10 แสดงการใช้ Screening Router เพื่อทำการกรองแพ็คเก็ตทุกแพ็คเก็ตจะประกอบด้วยรายละเอียดในส่วนของเฮดเดอร์

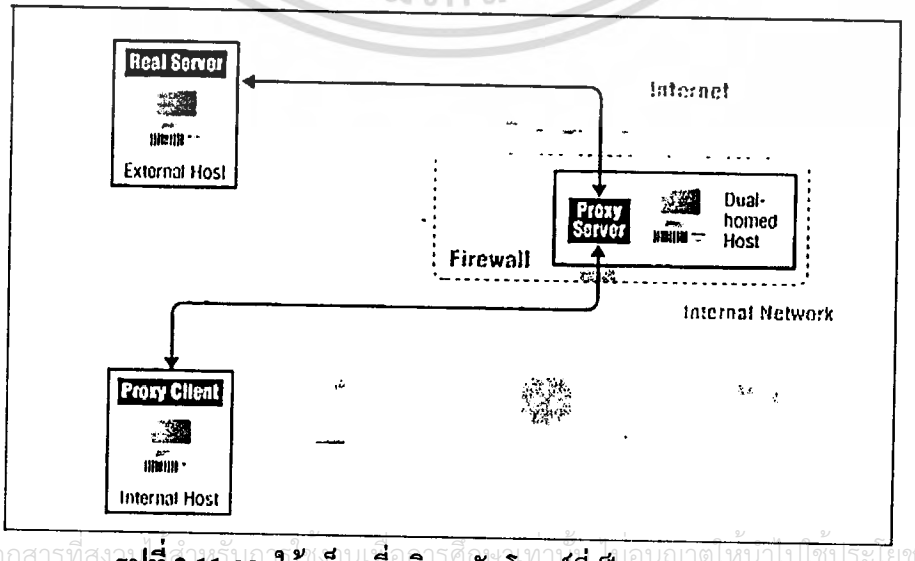
ขั้นตอนการทำงานของการกรองแพ็คเก็ตในอุปกรณ์มีดังนี้

- กฎการกรองแพ็คเก็ต(Packet filter rules) หรือ เงื่อนไขในการกรองแพ็คเก็ต
- เมื่อมีแพ็คเก็ตเข้ามาข้อมูลส่วนหัวของแพ็คเก็ตจะถูกกระจายออก เช่นอุปกรณ์เราท์เตอร์ส่วนใหญ่ที่กรองแพคเก็ตจะพิจารณาเฉพาะหัวของไอพีทั้งทีซีพีและ ยูดีพีเท่านั้น

- การกรองแพ็คเก็ตจะดูตามกฎที่มีการระบุถึงลำดับเพื่อให้มีการตรวจสอบเงื่อนไขแบบเป็นลำดับการทำงาน
- ถ้ากฎเป็นการบล็อกการส่งผ่านหรือการบล็อกการรับ แพ็คเก็ตเข้ามา แสดงว่าแพ็คเก็ตนั้นไม่ได้รับอนุญาตที่จะส่งหรือการรับแพ็คเก็ตนั้นเข้ามา
- ถ้ากฎเป็นการอนุญาตการส่งผ่าน หรือการอนุญาตรับแพ็ค เก็ตเข้ามา แสดงว่าแพ็คเก็ตนั้นได้รับอนุญาต ที่จะส่งหรือการรับแพ็คเก็ตนั้นเข้ามา
- ถ้าแพ็คเก็ตไม่เข้ากฎเงื่อนไขทั้งหมด แพ็คเก็ตนั้นจะถูกทำการบล็อก

2.2.2 พร็อกซี่บริการ

พร็อกซี่บริการเป็นแอปพลิเคชั่นที่ทำขึ้นมาเฉพาะ หรือเป็นโปรแกรมของเซิร์ฟเวอร์ (Server program) ที่ทำงานอยู่บนโฮสต์ที่เป็นไฟร์วอลล์ (Firewall Host) พร็อกซี่บริการจะตั้งอยู่ระหว่างผู้ใช้ที่อยู่ในเครือข่ายภายในและการบริการ อินเตอร์เน็ตที่อยู่ภายนอก ซึ่งเดิมผู้ใช้จะติดต่อโดยตรงก็จะติดต่อกับพร็อกซี่แทน ดังนั้นเสมือนกับพร็อกซี่จะทำหน้าที่ดูแล การติดต่อการสื่อสารทั้งหมดระหว่างผู้ใช้กับการบริการอิน เตอร์เน็ตเช่น FTP หรือ Telnet จากรูปที่2.11 แสดงการทำงานของพร็อกซี่บริการบนเครื่องโฮสต์ที่เป็นไฟร์วอลล์ ซึ่งจะเรียกเครื่องดังกล่าวว่าเป็น "Dual-homed Host" โดยที่พร็อกซี่เซิร์ฟเวอร์จะเป็นตัวแทนของเซิร์ฟเวอร์จริงสามารถที่จะติดต่อกับผู้ใช้ผ่านตัวแทนโฮสต์ ที่เป็นเครื่องไคล์เอนต์ซึ่งเรียกว่า "พร็อกซี่ไคล์เอนต์" โดยพร็อกซี่ทั้งสองนี้ต่างก็เป็นตัวแทนของทั้งเครื่องไคลแอนต์ และเครื่องเซิร์ฟเวอร์ซึ่งสามารถติดต่อกันได้แต่ว่า การติดต่อออกไปยังเครือข่ายภายนอก พร็อกซี่เซิร์ฟเวอร์จะทำหน้าที่แทนพร็อกซี่ไคล์เอนต์อีกทีหนึ่ง ดังนั้นพร็อกซี่เซิร์ฟเวอร์สามารถที่จะตัดสินใจในการอนุญาตหรือปฏิเสธ การร้องขอที่มาจากเครือข่ายภายนอกได้



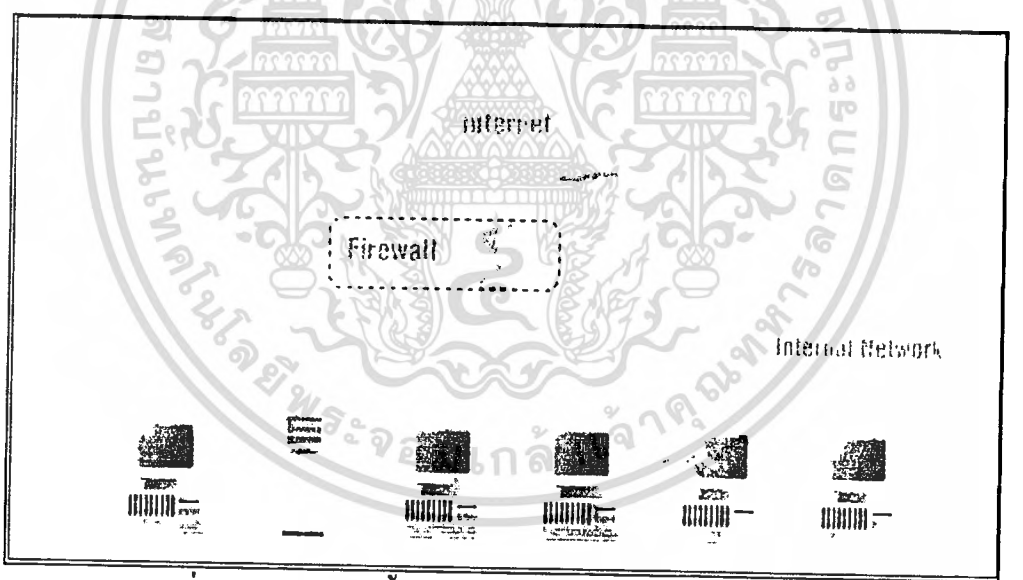
รูปที่ 2.11 การใช้พร็อกซี่บริการกับโฮสต์ที่เป็น Dual-home

2.2.3 ไฟร์วอลล์ชนิด Stateful Multilayer Inspaction Technology

Stateful Multilayer Inspection เป็นเทคโนโลยีที่เพิ่มเข้าไปใน Packet Filtering โดยในการพิจารณาว่าจะยอมให้แพ็คเก็ตใดผ่านไปนั้น แทนที่จะดูจะ Header เพียงอย่างเดียว Stateful Multilayer Inspection จะนำเอาส่วนข้อมูลของแพ็คเก็ต (Message Content) และข้อมูลที่ได้จากแพ็คเก็ตก่อนหน้านี้ที่ได้ทำการบันทึกเอาไว้ นำมาพิจารณาด้วยจึงทำให้สามารถระบุได้ว่าแพ็คเก็ตใด เป็นแพ็คเก็ตที่ติดต่อเข้ามาใหม่ หรือว่าเป็นส่วนหนึ่งของการเชื่อมต่อที่มีอยู่แล้วตัวอย่างผลิตภัณฑ์การค้าที่ใช้ Stateful Multilayer Inspaction Technology ได้แก่

- Check Point Firwall-1
- Cisco Secure Pix Firewall
- SunScreen Secure Net
- ส่วนที่เป็น Open Source ได้แก่ NetFilter ใน Linux (มี iptables ในลีนุกซ์เคอร์เนล 2.3)

ไฟร์วอลล์เป็นระบบที่ป้องกันอันตรายจากอินเตอร์เน็ต หรือเครือข่ายภายนอกโดย การควบคุมการเข้า-ออกของข้อมูล ดังนั้นไฟร์วอลล์จึงเป็นคอมโพเน็นต์หรือกลุ่มของคอมโพเน็นต์



รูปที่ 2.12 ไฟร์วอลล์กั้นระหว่างอินเตอร์เน็ตกับเครือข่ายภายใน

จากรูปที่ 2.12 ไฟร์วอลล์จะทำหน้าที่ในการควบคุมการเข้าถึงข้อมูลระหว่างเครือข่ายภายนอก(เครือข่ายที่คิดว่าไม่ปลอดภัย)

2.2.4 สถาปัตยกรรมไฟร์วอลล์ (Firewall Architecture)

ในส่วนของสถาปัตยกรรมไฟรวออล์ จะกล่าวถึงการจัดวางไฟร์วอลลล์ในแบบต่างๆเพื่อทำให้เกิดเป็นระบบ ไฟร์วอลล์ขึ้น โดยแบ่งรูปแบบสถาปัตยกรรมได้ดังนี้

2.2.4.1 Single Box Architecture

Single Box Architecture เป็นสถาปัตยกรรมแบบง่ายๆ ที่มีคอมโพเน็นต์ทำหน้าที่เป็นไฟร์วอลล์เพียงอันเดียวตั้งอยู่ระหว่างเครือข่ายภายในกับเครือข่ายภายนอก ข้อดีของวิธีนี้ก็คือ การควบคุมการเข้าออกของข้อมูลทำได้ง่ายแต่ก็มีข้อเสีย คือการที่มีเพียงจุดเดียวนี้ทำให้มีความเสี่ยงสูง หากมีการคอนฟิกูเลชันผิดพลาดหรือมีช่องโหว่เพียงเล็กน้อยก็อาจจะทำให้ระบบผิดพลาดได้ โดยคอมโพเน็นต์ที่ใช้ในสถา ปัตยกรรมนี้อาจเป็น Screening Router, Dual-Homed Host หรือ Multi-Purposed Firewall Box ก็ได้

- Screening Router Architecture สามารถใช้เราเตอร์ทำ Packet Filtering ซึ่งจะทำให้ประหยัดค่าใช้จ่ายเนื่อง จากส่วนใหญ่จะใช้เราเตอร์ต่อกับเครือข่ายภายนอกอยู่แล้ว
- Dual-Home Host Architecture สามารถใช้ Dual-Home Host (คอมพิวเตอร์ที่มีเครือข่ายอินเตอร์เฟสอย่างน้อย 2 สองอินเตอร์เฟส) เพื่อให้การบริการเป็น Proxy ให้กับเครื่องภายในเครือข่าย
- Multi-Purposed Firewall Box เป็นผลิตภัณฑ์หลายชนิดที่ผลิตออกมาเป็นกล่องๆเดียว และสามารถทำหน้าที่ได้หลายอย่างเช่นเป็น Packet Filtering, Proxy

2.2.4.2 Screened Host Architecture

Screened Host Architecture จะมีโฮสต์ซึ่งให้บริการ Proxy เหมือนกับใน Single Box Architecture ที่เป็น Dual-Home Host แต่จะแตกต่างกันตรงที่ว่า โฮสต์นั้นจะอยู่ภายในเครือข่าย ไม่ต่ออยู่กับเครือข่ายภายนอกอื่นๆ และจะมีเราเตอร์ ที่ทำหน้าที่ Packet Filtering ช่วยบังคับให้เครื่องภายในเครือข่ายต้องติดต่อบริการผ่าน Proxy โดยไม่ยอมให้ติดต่อใช้บริการจากภายนอกโดยตรง และก็ให้ภายนอกเข้าถึงได้เฉพาะ Bastion Host (Host ที่มีความเสี่ยงสูงต่อการถูกโจมตี มักจะเป็น host ที่ให้บริการทางอินเตอร์เน็ต)

วิธีนี้ถึงแม้จะมีทั้ง Proxy และ เราเตอร์ทำหน้าที่ Packet Filtering แต่ก็ยังคงอันตรายอยู่ เพราะว่าเราเตอร์ต้องยอมให้ภายนอกสามารถติดต่อกับ Bastion Host ได้ดังนั้นหากเฮกเกอร์สามารถเจาะเข้ามายัง Bastion Host ได้ก็ทำให้เกิดปัญหาขึ้นได้

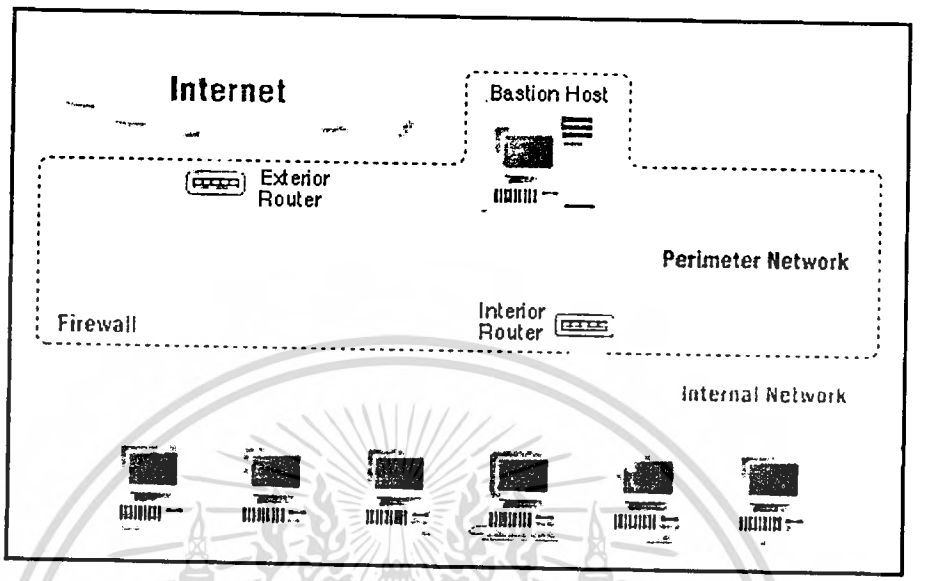
2.2.4.3 Multi Layer Architecture

Multi Layer Architecture เป็นสถาปัตยกรรมแบบหลายชั้น ไฟร์วอลล์จะเกิดขึ้นจากคอมโพเน็นต์หลายๆส่วน ทำหน้าที่ประกอบกันขึ้นเป็นระบบ วิธีการนี้สามารถเพิ่มความปลอดภัยได้มาก เนื่องจากเป็นการลดความเสี่ยงต่อความผิดพลาดที่อาจเกิดขึ้น โดยสถาปัตยกรรมแบบหลายชั้นนี้ จะเป็นการต่อกันเป็นลำดับโดยมี Perimeter Network (หรือบางทีเรียกว่า DMZ Network) อยู่ตรงกลาง เรียกว่า Screened Subnet Architecture

2.2.4 Screened Subnet Architechure

Screened Subnet Architechure เป็นสถาปัตยกรรมที่มีการเพิ่ม Perimeter Network เข้าไป

กั้นระหว่างอินเตอร์เน็ตกับเครือข่ายภายในไม่ให้มีการเชื่อมต่อกันโดยตรง ทำให้เครือ ข่ายภายในมีความปลอดภัยมากขึ้น



รูปที่ 2.13 แสดง Screened Subnet Architecture

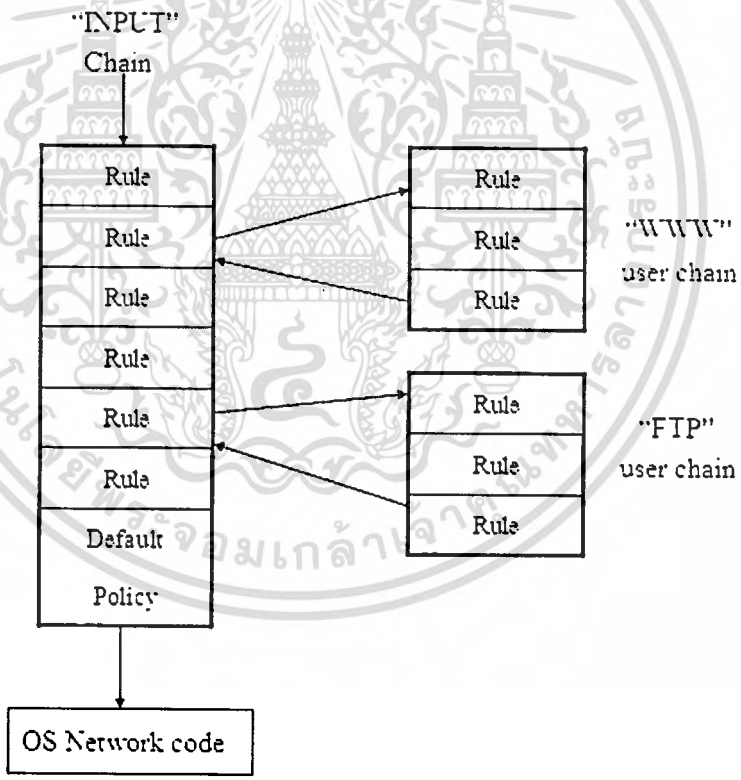
จากรูปที่ 2.13 จะแสดงรูปแบบอย่างง่าย ประกอบด้วย เราเตอร์ 2 ตัวตัวนึงอยู่ระหว่างอินเตอร์เน็ต กับPerimeter Network ส่วนอีกตัวหนึ่งอยู่ระหว่าง Perimeter Network กับเครือข่ายภายใน คอมโพเน็นต์ของ Screen Subnet Architecture

- Perimeter Network เป็นเครือข่ายที่เพิ่มเข้ามาเพื่อความปลอดภัยอยู่ระหว่างเครือข่ายภายนอกและเครือข่ายภายใน โดยจะแบ่งเครือข่ายออกเป็นส่วนๆ อย่างชัดเจน
- Bastion Host ตั้งอยู่บน Perimeter Network ทำหน้าที่ให้บริการเครือข่ายภายในและให้บริการต่างๆกับผู้ใช้ บนอิน เตอร์เน็ต Bastion Host นั้นจะมีความเสี่ยงต่อการถูกโจมตีสูง จึงต้องมีการดูแลความปลอดภัยเป็นพิเศษ
- Interior Router ตั้งระหว่าง Perimeter Network กับเครือข่ายภายใน ทำหน้าที่ Packet Filtering ป้องกันเครือข่ายภายในจาก Perimeter Network
- Exterior Router ตั้งอยู่ระหว่างเครือข่ายภายนอกกับ
- Perimeter Network เนื่องจาก Exterior Router นี้เป็นจุดที่ต่ออยู่กับเครือข่ายภายนอก จึงมีหน้าที่สำคัญคือ ป้องกัน Packet ที่มีการ Forged IP Address เข้ามาโดยอ้างว่ามาจาก เครือข่ายภายในทั้งที่จริงแล้วมาจาก เครือข่ายภายนอก

2.3 Linux Kernel Firewall

ลินุกซ์เป็นระบบปฏิบัติการที่ออกแบบมาเพื่อความเสถียรภาพและมีความปลอดภัยมากตัวหนึ่ง ได้มีการพัฒนาให้สนับสนุน Firwall ตั้งแต่ Linux Kernel 2.0 คือ ipfwadm ซึ่งทำงานเป็น Packet Filtering แบบ Rule Stack พอมาถึง Linux Kernel 2.1 จึงได้มีการพัฒนาไฟร์วอลล์ตัวใหม่ คือ ipchains มีประสิทธิภาพและจัดการกับกฎได้ดีกว่า จากนั้น Linux Kernel 2.4 ได้ถูกพัฒนาขึ้นและสร้างไฟร์วอลล์ ให้มีความสามารถ ตรวจสอบสถานะการทำงานของทราฟิกส์ บนแอพพลิเคชั่นต่าง ๆ ได้ คือ iptables ซึ่งนิยมใช้งานกันมากในปัจจุบัน

ipchains เป็นโปรแกรมที่พัฒนาบน Linux Kernel 2.1.102 ซึ่งมีประสิทธิภาพการทำงานที่ดีกว่า ipfwadm โปรแกรม ipchains ใช้เทคนิคการทำงานของ chain ที่สามารถจะสร้างกฎกติกาต่างๆ แล้วนำมาร้อยเรียงกันเป็นลูกโซ่ เพื่อสอดแทรกเข้าไปในณ. จุดใดจุดหนึ่งของ Stack (ต่างจาก ipfwadm ที่ทำได้เพียงการเพิ่มหรือลดกฎได้เฉพาะส่วนบุคคล หรือเพียงส่วนล่างของ Stack เท่านั้น)

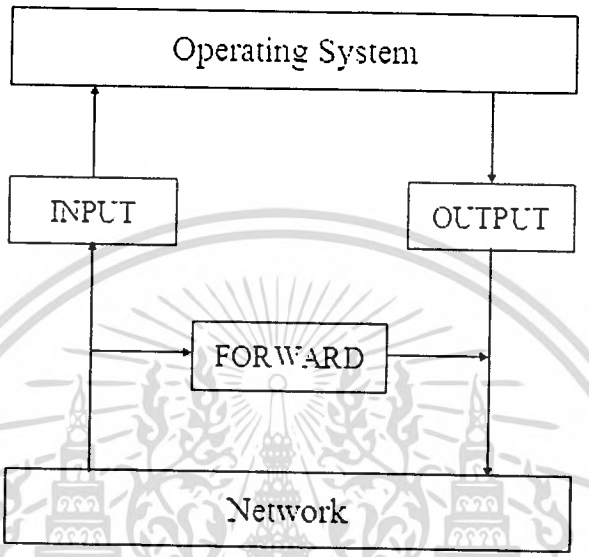


รูปที่ 2.14 การเชื่อมโยงของกฎบน ipchains

รูปที่ 2.14 จะเห็นว่ามีการแทรกกฎทราฟิกส์ของ WWW และ FTP เข้าไปได้ทุกจุดบน Stack ที่ต้องการ(ไม่ต้องเรียงลำดับจากบนลงล่าง) ทำให้สามารถตรวจสอบกฎกติกานั้นได้อย่างรวดเร็วสามารถลบหรือแทนที่กฎในตำแหน่งใดบน Stack ก็ได้ นอกจากนี้ ipchains ยังสามารถป้องกันการปลอม IP Address ต้นฉบับหรือ IP Spoofing ได้อีกด้วย

Iptables เป็นโปรแกรมไฟร์วอลล์ที่พัฒนาบน Linux Kernel 2.4 มีความสามารถในการ

ตรวจสอบสถานะการทำงานของทราฟิกส์บนแอพพลิเคชั่นต่างๆ ได้อย่างดี สนับสนุนการทำงานแบบ SPI (Stateful Inspection) โดยทำการวิเคราะห์และตรวจสอบดูพฤติกรรมการทำงานของตัวแอพพลิเคชั่นว่ามีอะไรผิดปกติหรือไม่ สามารถวิเคราะห์การทำงานของทราฟิกส์ระดับแอพพลิเคชั่นที่ทำงานบนโปรโตคอล FTP HTTP SMTP POP ได้อย่างดี iptables จึงจำเป็น Stateful Firewall ที่ทำงานได้อย่างมีประสิทธิภาพ



รูปที่ 2.15 แสดงโครงสร้างการทำงานของ iptables

จากรูปที่ 2.15 เป็นโครงสร้างของการทำงานของ iptables จะเห็นว่าแพ็กเก็ตที่วิ่งผ่านเข้ามาทางเน็ตเวิร์ก ต้องผ่าน INPUT Chain และถูกตรวจสอบจาก INPUT จนถึง Operating System ส่วนแพ็กเก็ตที่ส่งออกมาจะผ่าน OUTPUT Chain และถูกตรวจสอบเช่นกัน ถ้าไม่มีเงื่อนไขตรงกับกฎในการห้ามเข้า แพ็คเก็ตนั้นจะถูกโยนทิ้ง (drop)

บทที่ 3

วิเคราะห์และออกแบบระบบงาน

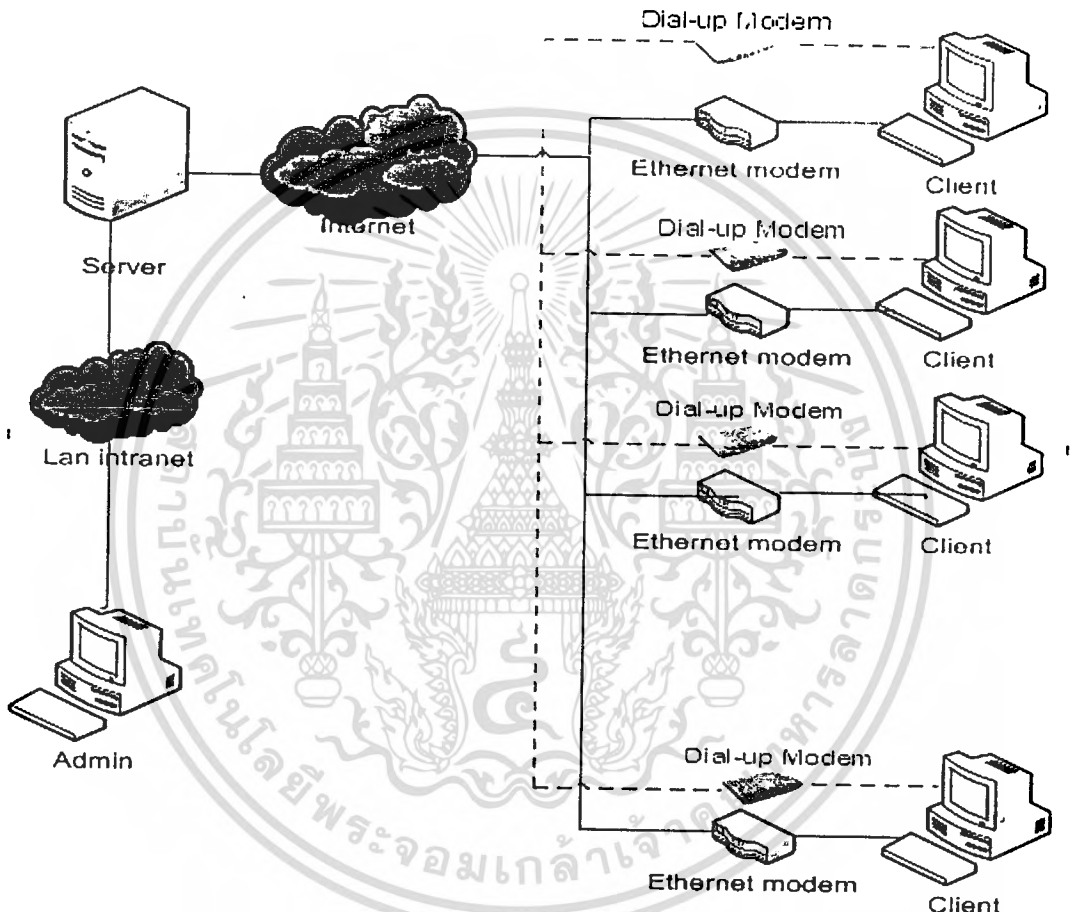
3.1 ความต้องการของระบบ

ระบบที่สร้างขึ้น ต้องมีความสามารถดังนี้

- ระบบมีความสามารถควบคุมการทำงานของอุปกรณ์ปลายทางจากส่วนกลางได้
  - ระบบสามารถทำการกำหนดกฎ การทำงานของไฟร์วอล์ปลายทาง ได้จากส่วนกลาง
  - ระบบสามารถจัดการ การทำงานการทดสอบ Traffic Performance ได้จากส่วนกลาง
  - ระบบสามารถกำหนดวันและ เวลาการทำงาน(Backup ,Restore)ของอุปกรณ์ปลายทางได้จากส่วนกลาง
- ระบบสามารถเก็บข้อมูล(Backup) ของอุปกรณ์ปลายทางได้
- ระบบสามารถ Restore ข้อมูลกลับไปยังอุปกรณ์ปลายทางได้
- ระบบจะต้องสามารถติดต่อกับอุปกรณ์ปลายทางได้แม้ว่าไม่สามารถใช้ผ่านบริการ Broadband ได้จากทาง Dial up
- ระบบสามารถทำงานได้ตามฟังก์ชันดังนี้
  - สามารถบูทเครื่องได้เอง ในกรณีเกิดไฟฟ้าดับ
  - มีระบบจัดการตัวเองทำให้ คน ไม่จำเป็นต้องเข้าไปหน้าเครื่องเพื่อทำงาน routine โดยมีฟังก์ชันก์การทำงานดังนี้
  - Admin สามารถ remote เข้ามาทำงานที่ Client ได้ผ่าน service secure shell
  - Client แต่ละตัวจะมี "คำสั่ง" โดย "คำสั่ง" (command) นี้จะถูกกำหนดโดย Admin ที่ server ผ่านทางเว็บ เช่น "ให้ Terminal1 ไป เล่นเว็บ www.hotmail.com" จากตัวอย่างหมายถึงให้ ให้ Terminal1 ไปทำ service HTTP กับ Host www.hotmail.com เป็นต้น
  - การทำคำสั่งแต่ละคำสั่งนั้น จะทำเป็นลักษณะ automatic script โดยทำตามที่ schedule กำหนด และทำไปเรื่อยๆ ไม่มีวันสิ้นสุด จนกว่า server จะลบ schedule นั้นทิ้งไป
  - เมื่อ Client ทำคำสั่งเสร็จแล้ว จะทำการเก็บข้อมูลภายในตัวเองก่อน (MySQL database) โดยจะมีอีก process หนึ่งทำหน้าที่ส่งข้อมูลที่ได้ มาที่ server ผ่านทาง program application
  - ทาง server จะมีไฟล์ ที่ทำหน้าที่เป็นตัวเก็บข้อมูลที่จะถูกส่งมาจาก Client แต่ละตัว โดย หลังจากได้ข้อมูลแล้วจะเก็บเข้า Database (MySQL)

- เครื่องที่เป็น server จะมี script ดึงข้อมูลจาก database ไปแสดงเป็น graph เพื่อให้ web application เรียกกราฟไปแสดงผล ในลักษณะของ monitor tools
- เครื่องที่เป็น server เป็นตัว manage ระบบทั้งหมดทั้ง client , command หรือการแสดงผล สำหรับ user (Admin จะมาเพิ่ม command ที่นี่)

3.2 หลักการการทำงานของระบบ

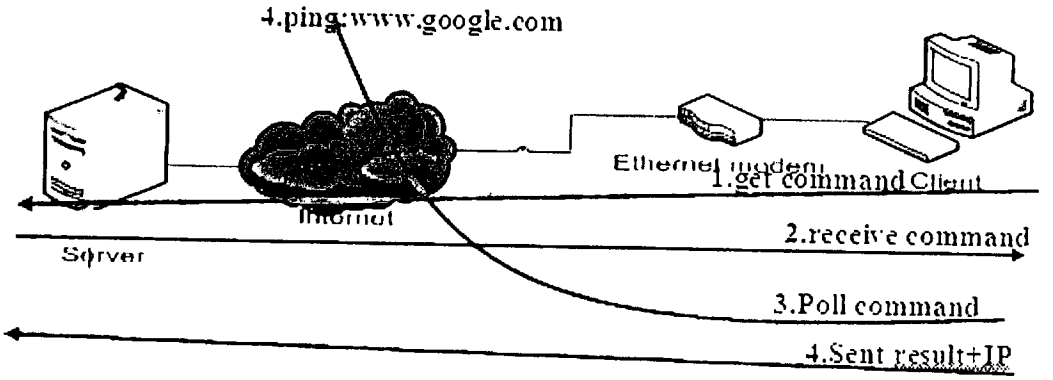


รูปที่ 3.1 แสดงการเชื่อมต่อทางกายภาพของระบบ

แบ่งการทำงานตามฟังก์ชั่นเป็น 3 รูปแบบ

1. หลักการการออกแบบของระบบการแสดงผลทราฟฟิกส์

จะเป็นการแสดงรูปแบบการทำงานของระบบเพื่อใช้ในการเก็บข้อมูลของ Client แต่ละตัวเพื่อนำข้อมูลนั้นไปประมวลผลเพื่อใช้ในการแสดงผลของทราฟฟิกส์ตามที่ Server กำหนดการทำงานให้กับ Client แต่ละตัว เช่น เมื่อ Server กำหนดให้ Client ทำการทดสอบ PING ไปยัง www.google.com และเมื่อ Client ได้ทำงานฟังก์ชั่น PING แล้วจะทำการส่งผลการทดสอบของ www.google.com ที่ Client นั้นได้ทำการทดสอบกลับมายัง Server ตามระยะเวลาที่กำหนด ทุก ๆ 2 นาที



รูปที่ 3.2 แสดงการออกแบบของระบบการแสดงผลทราฟฟิกส์

1.1 Client จะทำการร้องขอการทำงานจาก Server โดยการส่งชื่อของตัว Client เองไปยัง Server เพื่อร้องของานที่จะให้ Client ทำงานนั้น ๆ โดยชื่อของ Client จะต้องถูกกำหนดไว้ที่ Server ด้วยจึงสามารถทำงานได้

ฟังก์ชั่นของการเรียกใช้งานจาก Client มายัง Server จะเรียกใช้งานผ่าน Class httpclient ซึ่งจะทำการดึงหน้า page content จาก Server ตามงานที่ Client ต้องทำ

1.2 Server ทำการตรวจสอบชื่อของ Client ว่ามีอยู่ในระบบหรือไม่ หากไม่มีอยู่ในระบบ Server จะไม่ตอบข้อมูลกลับไปยัง Client ที่ร้องขอมา หากมีชื่อของ Client อยู่ในระบบ Server จะทำการตอบงานของ Client ตามที่กำหนดไว้บน Server

1.3 Client ทำการรับข้อมูลที่ส่งกลับมาจาก Server โดย Client จะทำการตรวจสอบงานที่ Server ส่งมาให้และจะทำงานตามฟังก์ชั่นงานนั้น ๆ โดยฟังก์ชั่นงานจะมีดังนี้

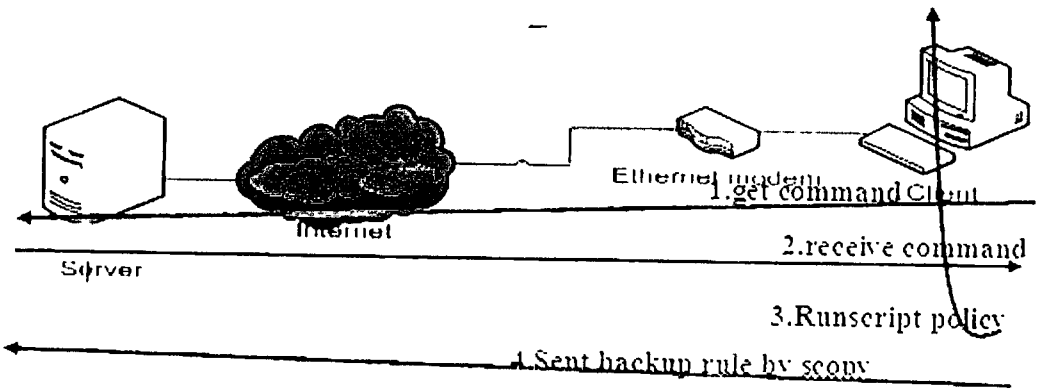
ฟังก์ชั่น PING : Client จะทำการเรียกใช้งาน Module PING ที่อยู่ใน Class_ICMP เพื่อเป็นการหาค่า round trip time (rtt) และจะเป็นการกำหนดการทดสอบซึ่งให้ทำงาน 5 ครั้งต่อผลการทดลอง 1 ครั้ง

ฟังก์ชั่น HTTP : Client จะทำการเรียกใช้งาน Module ของการดึงหน้า page content ที่อยู่ใน Class httpclient เพื่อคำนวณหาขนาดของหน้า page เพื่อนำมาคำนวณหาความเร็วในการเข้าถึงหน้า page นั้น ๆ โดยใช้สูตร bpf = (ขนาดของหน้า page content x 8)/เวลาการเข้าถึง

1.4 เมื่อ Client ได้ผลการทดสอบตามฟังก์ชั่น PING หรือ HTTP แล้วจะทำการส่งผลลัพธ์กลับมายัง Server เพื่อ Server จะได้นำผลนั้นไปประมวลผลต่อไปและ Client จะทำการ update IP address นั้น ๆ กลับไปยัง Server เพื่อบอกสถานะของ IP address และเวลาที่ update ล่าสุดของ Client

2. หลักการการออกแบบของระบบการทำงาน Firewall

จะเป็นการออกแบบให้ Client สามารถทำงานในฟังก์ชั่น Firewall ได้โดยการกำหนดการทำงานจากส่วนกลางซึ่ง Server จะเป็นผู้กำหนดกฎการทำงานต่างๆให้กับ Client แต่ละตัว



รูปที่ 3.3 แสดงการออกแบบของระบบการทำงาน Firewall

2.1 Client จะทำการร้องขอการทำงานจาก Server โดยการส่งชื่อของตัว Client เองไปยัง Server เพื่อร้องของานที่จะให้ Client ทำงานนั้น ๆ โดยชื่อของ Client จะต้องถูกกำหนดไว้ที่ Server ด้วยจึงสามารถทำงานได้

ฟังก์ชั่นของการเรียกใช้งานจาก Client มายัง Server จะเรียกใช้งานผ่าน Class httpclient ซึ่งจะทำการดึงหน้า page content จาก Server ตามงานที่ Client ต้องทำ

2.2 Server ทำการตรวจสอบชื่อของ Client ว่ามีอยู่ในระบบหรือไม่ หากไม่มีอยู่ในระบบ Server จะไม่ตอบข้อมูลกลับไปยัง Client ที่ร้องขอมา หากมีชื่อของ Client อยู่ในระบบ Server จะทำการตอบงานของ Client ตามที่กำหนดไว้บน Server

2.3 Client ทำการรับข้อมูลที่ส่งกลับมาจาก Server โดย Client จะทำการตรวจสอบงานที่ Server ส่งมาให้และจะทำงานตามฟังก์ชั่น Firewall (iptables)

2.4 ก่อนการทำงานฟังก์ชั่นของ Firewall Client จะทำการ Backup iptables เก่ามาเก็บที่ Server ก่อนแล้วจึงทำงานตามฟังก์ชั่น Firewall เมื่อหลังจากการทำงานเสร็จแล้วจะทำการ Backup iptables ใหม่มาเก็บที่ Server อีกครั้งหนึ่ง

3. การ Backup ข้อมูล

การ Backup ข้อมูลจะใช้ช่องทาง secure channel ที่มีความปลอดภัยโดยเลือกใช้ฟังก์ชั่น SCP โดยก่อนทำการ Backup เครื่อง Client จะต้องนำ public key ของตัวเองมาเก็บที่ Server และจะต้องนำ public key ของ Server มาเก็บที่ Client ด้วยเพื่อใช้ในการ authenticate ระหว่าง client และ server

ตัวอย่าง ค่าของ public key ที่ client ซึ่งประกอบไปด้วยค่า public key ของ Server (**root@decho-desktop**) และ public key ของ client เอง (root@ASD-03_1)

Client : ASD-03_1:~/.ssh# more authorized_keys

ssh-dss AAAAB3NzaC1kc3MAAACBAJQMCr8CjhOJ/mi6g25gKRK9R4ol+GjoHUzYPfA8jOZtVXtdGuhBupeX9a1qSXEkor48jHy5dA5vyUhNqYm2nhlUALQOFwRj+O3AMOMU
jOwpcXLhJisLhmV2VWQWTLQ3aPCcxes0VnVShETOZveUxfk+lu/QSKjXxWkGChJOfBuHAAAAFQD/2FDm+R0yGbtYijQBPYNQJIINawAAAIEAiKmEno+wc7njEY9m+6P4dOX5
3BS5IXl80B7PKkEth08LsMI8QsuuIEaUUxsoZvIIISKp1r1JjN+ECvvAC0uVexOdz8Q3FaZpcbSrx644By9Ve0LOQbKWVgX5dxfxM+Dk/3b/DkIwVpGxOrs6r1RgBbu80OQ
PBMhkmoYhJw1a9EAAACATy0JLb4XNAn9rIWI4eDl++4EwD2KoDLblbBZWaKNcf7iT6Ct8ygsYpIWyYFsKeDJkOawLjfLKNe4LY60qfUsX0DPJWfcTeXRZPasrPtu+8YxaFZ
g+851Ck9Un/0+tbtNmih5fBb9OYvsIKLSrEo6UYMTIchRIUQDCintjirY8k= **root@ASD-03_1**

ssh-dss AAAAB3NzaC1kc3MAAACBAOgovfD3jIwOOhnEagL5sW2pUYL7hYJy/Q1Ve+d8TnhDlrgtAIsmMorFgmA/WYBSHSqJR0fKLg/lvNEsvDo5xUf4VV/1pno84K1jtOO8
muLZ4VWgb4ez90DIBjuZwZIVqdJp+Q38ZcIVPAPNAjHQG4MOTiprCEb11aX9UOgDq6ZrAAAAFQDD7/x1lRtCjGgIMe5YNcfl4O2bmQAAAIEAyvPrySARU3bwCIkFAY/K57Fe
4M2JAtEnMjJ4gR0IViIUYOOSCentIAXrk79Cpo1k+UiGLnV1Xesweg46XTyQtVMKWjRnFRVHDUW9C5wYtKaCDtTJS7vpcsvwJl8nLnyQn3EonXozp/cChp5SZxV3hJoTlNHT
Z0UokjccSa2RNTEAAACBAJ73mXJFeSpgmbRnRJ6VJxe+23UjCwftk8MqR9xqnK3khAF58A7CD6ysa9RpBzi1leEBOs2SVHomvUtq2Ahg5wDvxjxJQ7Ew1Ob43MvgNzqAub/r
AHmJPTiUmGYpU4S3Er86LIqpOH3qwbN01tl+dL2YQ4hhayw1ywU+r2O1jWFO root@decho-desktop

ตัวอย่าง ค่าของ public key ที่ Server ซึ่งประกอบไปด้วยค่า public key ของ Client (root@ITM-03_1, root@ASD-03_1) และ public key ของ server เองด้วย (root@decho-desktop)

Server : root@decho-desktop:~/.ssh# more authorized_keys

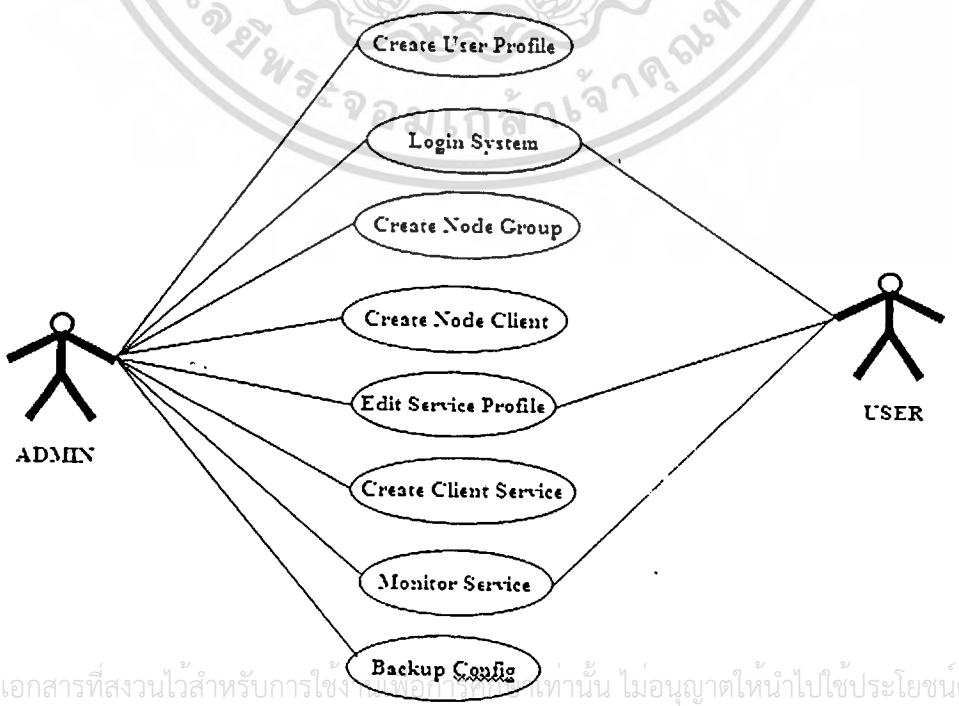
ssh-dss AAAAB3NzaC1kc3MAAACBAOgovfD3jIwOOhnEagL5sW2pUYL7hYJy/Q1Ve+d8TnhDlrgtAIsmMorFgmA/WYBSHSqJR0fKLg/lvNEsvDo5xUf4VV/1pno84K1jtOO8
muLZ4VWgb4ez90DIBjuZwZIVqdJp+Q38ZcIVPAPNAjHQG4MOTiprCEb11aX9UOgDq6ZrAAAAFQDD7/x1lRtCjGgIMe5YNcfl4O2bmQAAAIEAyvPrySARU3bwCIkFAY/K57Fe
4M2JAtEnMjJ4gR0IViIUYOOSCentIAXrk79Cpo1k+UiGLnV1Xesweg46XTyQtVMKWjRnFRVHDUW9C5wYtKaCDtTJS7vpcsvwJl8nLnyQn3EonXozp/cChp5SZxV3hJoTlNHT
Z0UokjccSa2RNTEAAACBAJ73mXJFeSpgmbRnRJ6VJxe+23UjCwftk8MqR9xqnK3khAF58A7CD6ysa9RpBzi1leEBOs2SVHomvUtq2Ahg5wDvxjxJQ7Ew1Ob43MvgNzqAub/r
AHmJPTiUmGYpU4S3Er86LIqpOH3qwbN01tl+dL2YQ4hhayw1ywU+r2O1jWFO root@decho-desktop
ssh-dss AAAAB3NzaC1kc3MAAACBAJm7oPU6qR1xUNcBkyNK+sYi50gKRIvKOrV6tZ+wdGMKBTi0M9Hk8rEU3CaCPaL6h8d2DHKjUkBIvgdprhXjFyxtwtboHHSpnHcQ6IEe
R4C4Zv4Wdq9+A8h+JFd3U5eFs7b4Nsh18icdTuHASzNz+SZtraYAD0BCVwlM4pZ6bPurAAAAFQD06wfRGQFznbIT+WCjmz0H8rz2KQAAAIBzLp7Uy14BEKxHbpi6QZg/K5E/
A5YIqKwJv0fGanvcaTKmuijlsitnNJSggqUluY167nAiA9hGGul4pfaBz+pTe3FtvW1GBcMXG71t/i0gNo2N75HDF7ricoxc4ILBK6KG549dzm74aAT4TlrEw3ZwnsA41m1N
qjkyXls3IrD1KgAAAIEAmOtCia/XKfzSos7PytTPtXKyo/8yol0WZWkiu0Zsuhi+phSkVRLa9Ax5WpQYlIIHbaS5ti1+mKUTK6wB0mBekp0LJM6m8btR3lBForGvXltVAGkf
yIK1DzlrWfUfxLZr8wzRxt3wbpUbIVvzkLuge3xYFlKqJBAlVTlCvnEowyQ= root@ITM-03_1
ssh-dss AAAAB3NzaC1kc3MAAACBAJQMCr8CjhOJ/mi6g25gKRK9R4ol+GjoHUzYPfA8jOZtVXtdGuhBupeX9a1qSXEkor48jHy5dA5vyUhNqYm2nhIUALQOFwRj+O3AMOMU
jOwpcXLhJisLhmV2VWQWTLQ3aPCexes0VnVShETOZveUxIk+lu/QSKjXxWkGChJOfBuHAAAAFQDV2FDm+R0yGbtYijQBPYNQJIlNawAAAIEAiKmEno+we7njEY9m+6P4dOX5
3BS5IXl80B7PKkEth08LsMI8QsuuIEaUUxsoZvIIISKp1r1JjN+ECvvAC0uVexOdz8Q3FsZpcbSrx644By9Ve0LOQbKWVgX5dxfxM+Dk/3b/DkIwVpGxOrs6r1RgBbtr80OQ
PBMhkmoYhJw1a9EAAAACATy0JLb4XNAn9rlWI4eDl++4EwD2KoDLblbBZWaKNcf7iT6Ctl8ygsYpIWyYFsKeDJkOawLjfLKNe4LY60qfUsX0DPJWfcTeXRZPasrPtu+8YxaFZ
g+851Ck9Un/0+tbtNmih5fBb9OYvsIKLSrEo6UYMTlchRJUQDCintJirY8k= root@ASD-03_1

root@decho-desktop:~/.ssh#

3.3 แบบจำลองแนวคิดของระบบ

Model ที่ใช้ในการวิเคราะห์การทำงานของระบบจะใช้วิธีการวิเคราะห์การทำงานแบบ Unified Modeling Language(UML) โดยจะแบ่งออกเป็น 3 มุมมอง

3.3.1 Use case Model ใช้ในการอธิบายระบบงานทั้งหมด



รูปที่ 3.4 แสดง Use Case Diagram ของระบบ

USE CASE DESCRIPTION

Use Case: Create User Profile

Brief Description: จะทำสร้าง Username และ Password และกำหนดสิทธ์ของผู้ใช้แต่ละราย

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login โดยจะต้องได้รับสิทธิ์การใช้เป็น Super Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการสร้างข้อมูลและรายละเอียดของข้อมูล
2. ทำการป้อน Username และ Password ของผู้จะเข้าใช้ระบบ และสิทธ์ ของผู้ใช้แต่ละราย
3. ทำการบันทึกข้อมูลลงสู่ระบบ

Use Case: Login System

Brief Description: ทำการตรวจสอบว่า Username และ Password อยู่ในระบบหรือไม่

Actor: Administrator, User

Precondition: ผู้ใช้งานจะต้องมี Username และ Password เก็บอยู่ในระบบ

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการ Login เข้าสู่ระบบ
2. ทำการป้อน Username และ Password ของผู้จะเข้าใช้ระบบ

Use Case : Create Node Group

Brief Description: ทำการสร้าง Group ของ Client Node

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login โดยจะต้องได้รับสิทธิ์การใช้เป็น Admin

Basic Flows:

1. ผู้ใช้เรียก Web page ขึ้นเพื่อทำการสร้าง Group Name และะรายละเอียดของข้อมูล
2. ทำการป้อน Groupname ลงสู่ระบบ
3. ทำการบันทึกข้อมูลลงสู่ระบบ

Use Case : Create Node Client

Brief Description: ทำการสร้าง Node Client

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login โดยจะต้องได้รับสิทธิ์การใช้เป็น Admin

Basic Flows:

1. ผู้ใช้เรียก Web page เพื่อทำการสร้าง Node Client
2. ทำการป้อนชื่อ Node Client และทำการกำหนด Group ของ Node ลงสู่ระบบ
3. ทำการบันทึกข้อมูลลงสู่ระบบ

Use Case : Edit Service Profile

Brief Description: ทำการสร้าง Profile Configuration

Actor: Administrator, user

Precondition: ผู้ใช้งานจะต้องทำการ Login โดยจะต้องได้รับสิทธิ์การใช้เป็น Admin หรือสูงกว่า

Basic Flows:

1. ผู้ใช้เรียก Web page เพื่อทำการสร้าง Profile Configuration
2. ทำการเลือก ชนิดของ Profile ที่ต้องการสร้าง เช่น HTTP และทำกำหนดชื่อของ Profile HTTP นี้ ทำการ กำหนดค่าต่างๆ ของ Profile นี้ เช่น Domain ปลายทางที่ต้องการจะทดสอบ
3. ทำการบันทึกข้อมูลลงสู่ระบบ

Use Case : Create Client Service

Brief Description: ทำการกำหนด Service ที่ให้กับ Client

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login โดยจะต้องได้รับสิทธิ์การใช้เป็น Admin

Basic Flows:

1. ผู้ใช้เรียก Web page เพื่อทำการกำหนด Service ให้กับ Client ในแบบต่าง ๆ
2. ทำการเลือก Client Node หรือ Group Node หรือ Node ทั้งหมด และทำการเลือก Service ที่ต้องการใช้งาน
3. ทำการบันทึกข้อมูลลงสู่ระบบ

Use Case : Monitor Service

Brief Description: เข้าไปดูการทำงานของ Service ทั้งหมด

Actor: Administrator, User

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบ

Basic Flows:

1. ผู้ใช้เรียก Web page เพื่อทำการเลือก Service ที่เราต้องการ Monitor

Use Case : Backup Config

Brief Description: ทำการเก็บข้อมูลของ Client Node

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login โดยจะต้องได้รับสิทธิ์การใช้เป็น Admin

Basic Flows:

1. ผู้ใช้เรียก Web page เพื่อทำการเก็บ ข้อมูล Node Client
2. ทำการกำหนด Node Client และทำการกำหนด Group ของ Node หรือเลือกข้อมูลทั้งหมดที่ต้องการเก็บข้อมูล
3. กำหนดเวลาที่ต้องการในการเก็บข้อมูล
4. ทำการบันทึกข้อมูลลงสู่ระบบ

Use Case : Restore Config

Brief Description: ทำการนำข้อมูลที่ได้ทำการ Backup นำกลับไปยัง Client Node

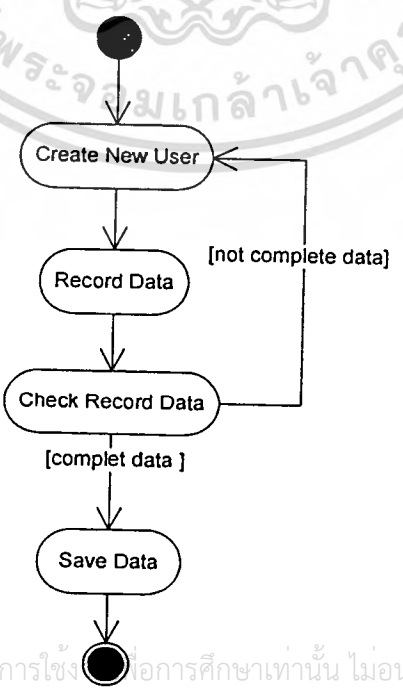
Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login โดยจะต้องได้รับสิทธิ์การใช้เป็น Admin

Basic Flows:

1. ผู้ใช้เรียก Web page เพื่อทำการ Restore ข้อมูลลงใน Node Client
2. ทำการกำหนด Node Client และทำการกำหนด Group ของ Node
3. กำหนดเวลาที่ต้องการในการทำ Restore ข้อมูล
4. ทำการบันทึกข้อมูลลงสู่ระบบ

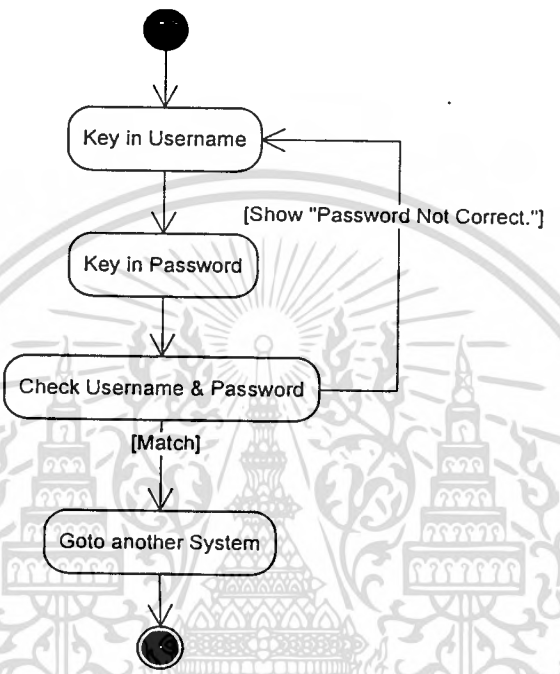
3.3.2 Activity Diagram



รูปที่ 3.5 Activity Diagram ของ Create User Profile

อธิบายการทำงานของ Activity Diagram : Create User Profile

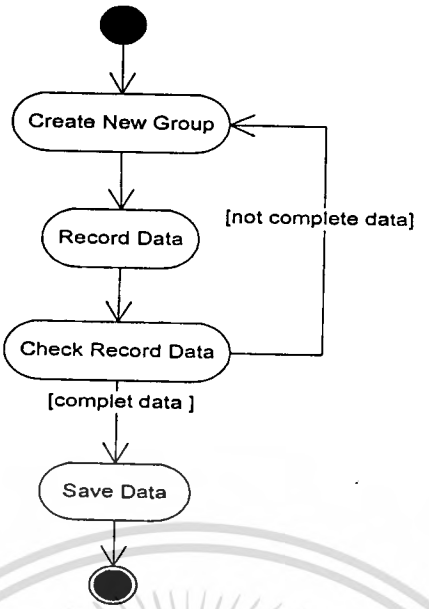
1. จากหน้าจอหลัก คลิกเลือกเมนู User
2. ระบบแสดงหน้าจอเข้าสู่ระบบ
3. ป้อนข้อมูล Username , password และ level ของ User นั้น
4. คลิกปุ่ม Add
5. ระบบทำการตรวจสอบ Username ว่ามีอยู่ในระบบหรือไม่ หากไม่มีจะทำการบันทึกข้อมูล



รูปที่ 3.6 Activity Diagram ของ Login System

อธิบายการทำงานของ Activity Diagram : Login System

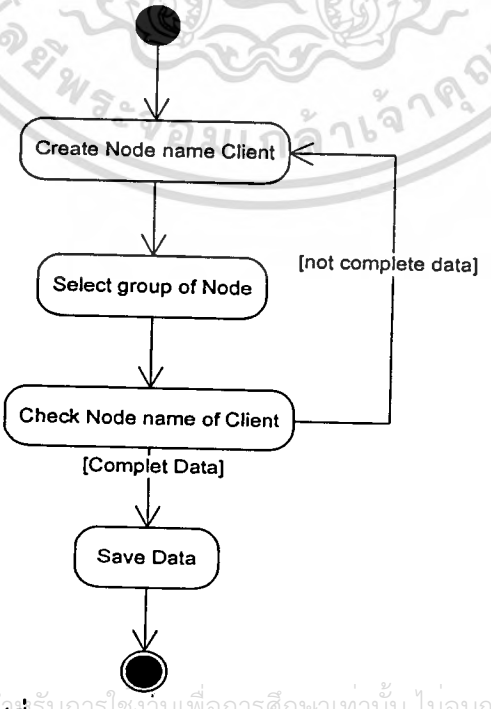
1. เข้าสู่ระบบ ระบบจะแสดงหน้าจอให้ใส่ Username และ Password
2. ป้อนข้อมูล Username และ Password ของ User นั้น
3. คลิกปุ่ม Login
4. ระบบทำการตรวจสอบ Username และ Password ว่ามีอยู่ในระบบหรือไม่ หากมีอยู่ในระบบจะสามารถเข้าใช้ระบบได้ หากไม่มีอยู่ในระบบจะขึ้นหน้าจอให้ใส่ Username และ Password อีกครั้ง



รูปที่ 3.7 Activity Diagram ของ Create Node Group

อธิบายการทำงานของ Activity Diagram : Create Node Group

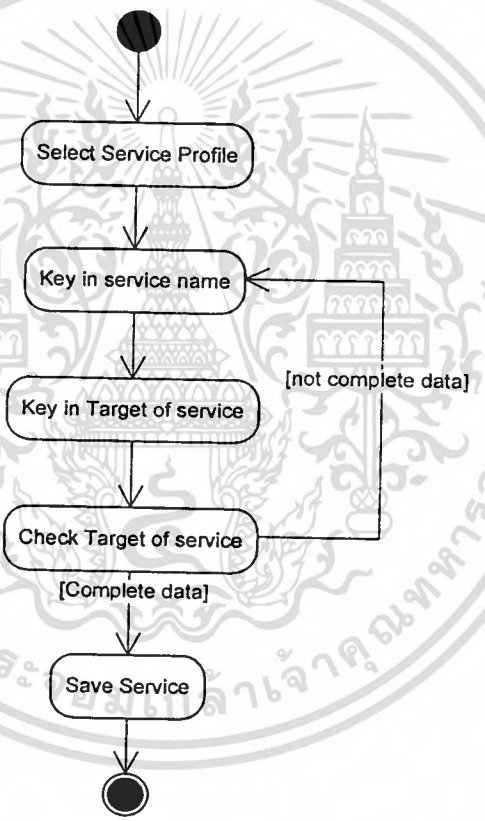
1. เข้าสู่ระบบเลือกเมนู Group
2. ระบบแสดงหน้าจอเข้าสู่ระบบ
3. ป้อนข้อมูล groupID และ groupName
4. คลิกปุ่ม save
5. ระบบทำการตรวจสอบ groupID และ groupName ว่ามีอยู่ในระบบหรือไม่ หากมีข้อมูลดังกล่าวอยู่ในระบบ ระบบจะทำการส่งค่า Error กลับไป โดยหากระบบไม่มีข้อมูลดังกล่าวในระบบระบบจะทำการบันทึกข้อมูล



รูปที่ 3.8 Activity Diagram ของ Create Node Client

อธิบายการทำงานของ Activity Diagram : Create Node Client

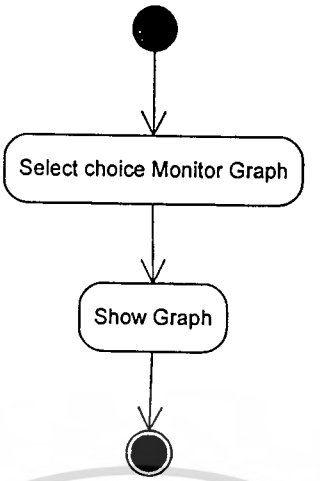
1. เข้าสู่ระบบเลือกเมนู : Client
2. ระบบแสดงหน้าจอเข้าสู่ระบบ
3. ป้อนข้อมูล groupName และ nodeName
4. คลิกปุ่ม save
5. ระบบทำการตรวจสอบ nodeName และ groupName ว่ามีอยู่ในระบบหรือไม่ หากมีข้อมูลดังกล่าวอยู่ในระบบ ระบบจะทำการส่งค่า Error กลับไป โดยหากระบบไม่มีข้อมูลดังกล่าวในระบบ ระบบจะทำการบันทึกข้อมูล



รูปที่ 3.9 Activity Diagram ของ Edit Service Profile

อธิบายการทำงานของ Activity Diagram : Edit Service Profile

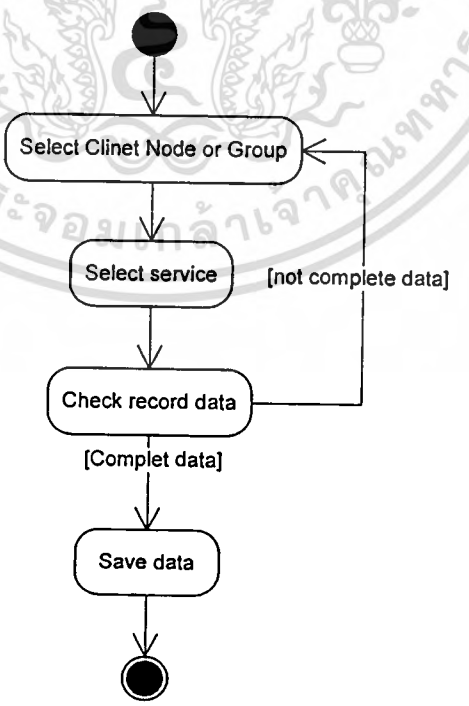
1. เข้าสู่ระบบเลือกเมนู Command Target
2. ระบบแสดงหน้าจอเข้าสู่ระบบ
3. ป้อนข้อมูล serviceName และ รายละเอียดของ Service ต่างๆ
4. คลิกปุ่ม save
5. ระบบทำการตรวจสอบ serviceName ว่ามีอยู่ในระบบหรือไม่ หากมีข้อมูลดังกล่าวอยู่ในระบบ ระบบจะทำการส่งค่า Error กลับไป โดยหากระบบไม่มีข้อมูลดังกล่าวในระบบ ระบบจะทำการบันทึกข้อมูล



รูปที่ 3.10 Activity Diagram ของ Monitor Service

อธิบายการทำงานของ Activity Diagram : Monitor Service

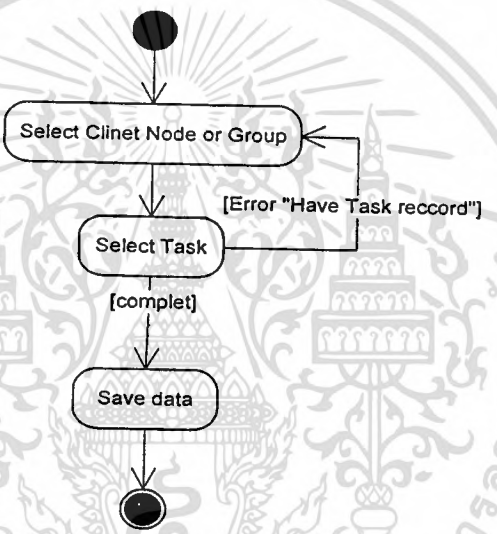
1. เข้าสู่ระบบเลือกเมนู Monitor Service
2. ระบบแสดงหน้าจอเข้าสู่ระบบ
3. เลือก Client ที่ต้องการดูค่า
4. โดยระบบจะทำการตรวจสอบข้อมูลที่มีอยู่ในระบบ หากไม่มีข้อมูลระบบจะส่งค่ากลับไป เพื่อบอกว่าไม่มีข้อมูลในระบบ หากมีข้อมูลในระบบ ระบบจะทำการส่งค่ากลับไป เช่น Traffic ที่ได้ทดสอบ



รูปที่ 3.11 Activity Diagram ของ Create Client Service

อธิบายการทำงานของ Activity Diagram : Create Client Service

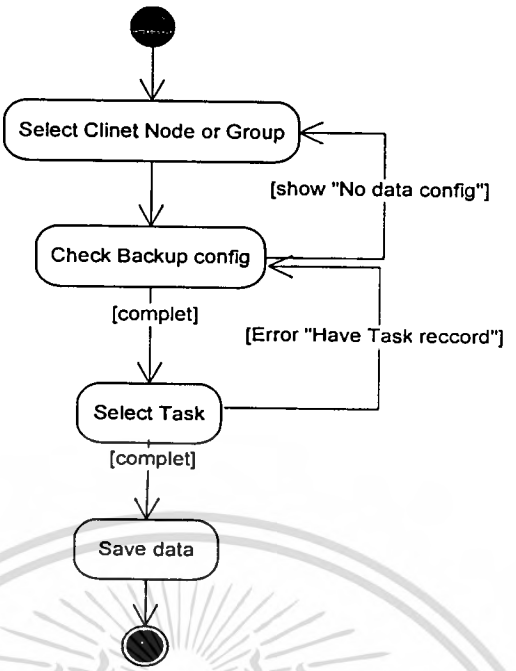
1. เข้าสู่ระบบเลือกเมนู Target
2. ระบบแสดงหน้าจอเข้าสู่ระบบ
3. เลือกข้อมูล Node Client หรือ Node Group
4. กำหนด Service ให้กับ Node Client หรือ Node Group
5. คลิกปุ่ม save
6. ระบบทำการตรวจสอบ หากมี service ที่ได้เลือกมาในระบบแล้ว ระบบจะทำการส่งค่า Error กลับไป โดยหาก Node Client หรือ Node Group ไม่มีข้อมูลดังกล่าวในระบบ ระบบจะทำการบันทึกข้อมูล



รูปที่ 3.12 Activity Diagram ของ Backup Config

อธิบายการทำงานของ Activity Diagram : Backup Config

1. เข้าสู่ระบบเลือกเมนู Backup Config
2. ระบบแสดงหน้าจอเข้าสู่ระบบ
3. เลือกข้อมูล nodeName หรือ nodeGroup
4. ทำการกำหนดเวลาที่ต้องการให้ระบบทำการBackup ข้อมูล
5. คลิกปุ่ม save
6. ระบบจะทำการตรวจสอบว่ามีการตั้งเวลาของ Client ไว้ก่อนหน้าหรือยังหากมีแล้ว ระบบจะทำการส่งค่า Error ไปบอกเพื่อให้ Admin ทำการกำหนดใหม่ หากไม่มีข้อมูลในระบบ ระบบจะทำการบันทึกข้อมูล

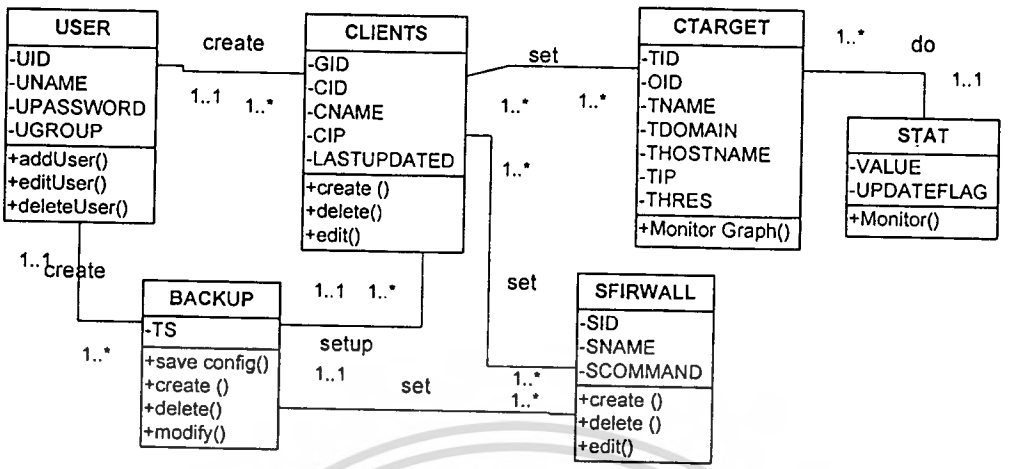


รูปที่ 3.13 Activity Diagram ของ Restore Config

อธิบายการทำงานของ Activity Diagram : Restore Config

1. เข้าสู่ระบบเลือกเมนู Restore Config
2. ระบบแสดงหน้าจอเข้าสู่ระบบ
3. เลือกข้อมูล nodeName หรือ nodeGroup
4. ระบบจะทำการตรวจสอบข้อมูลในระบบก่อนว่ามีข้อมูลที่ ได้เลือกนั้นมีการ Backup ไว้หรือไม่ หากไม่มีข้อมูล Backup อยู่ระบบจะส่งค่า error กลับไปว่าไม่มีข้อมูลในระบบ
5. ทำการกำหนดเวลาที่ต้องการให้ระบบทำการ Restore ข้อมูล
6. คลิกปุ่ม save
7. ระบบจะทำการตรวจสอบว่ามีการตั้งเวลาของ Client ไว้ก่อนหน้าหรือยังหากมีแล้ว ระบบจะทำการส่งค่า Error ไปบอกเพื่อให้ Admin ทำการกำหนดใหม่ หากไม่มีข้อมูลในระบบ ระบบจะทำการบันทึกข้อมูล

3.3.3 Class Diagram

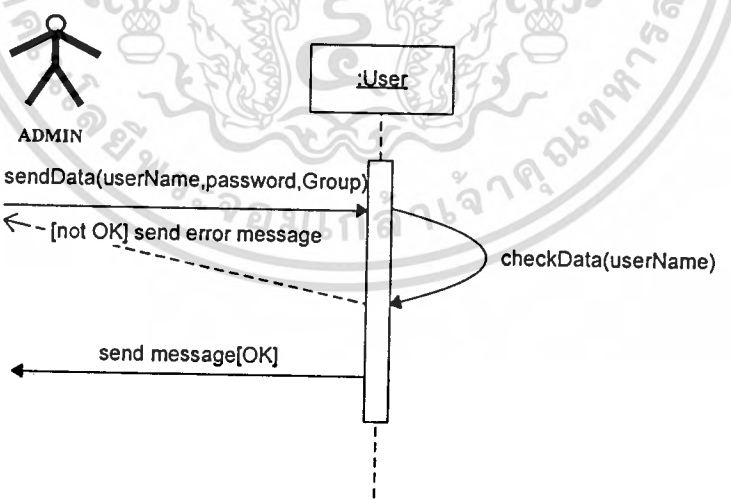


รูปที่ 3.14 Class Diagram

3.3.4 Behavioral Models

เป็นการมองกระบวนการของระบบกรือกลไกของระบบ โดยมองในลักษณะพฤติกรรมของระบบ ว่าระบบทำงานอย่างไร

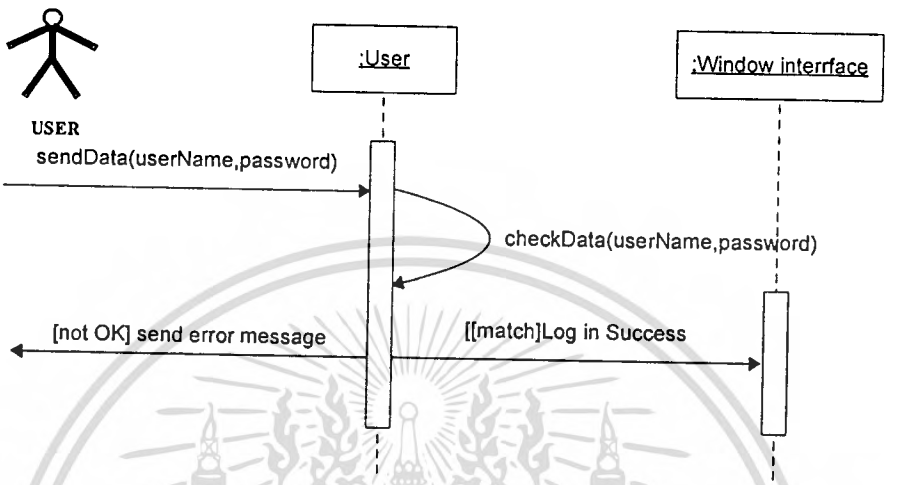
ซึ่งในที่นี้ใช้ Sequence Diagram เพื่ออธิบายกลไกของระบบในลักษณะพฤติกรรมของระบบ



รูปที่ 3.15 Sequence Diagram ของ Create User Profile

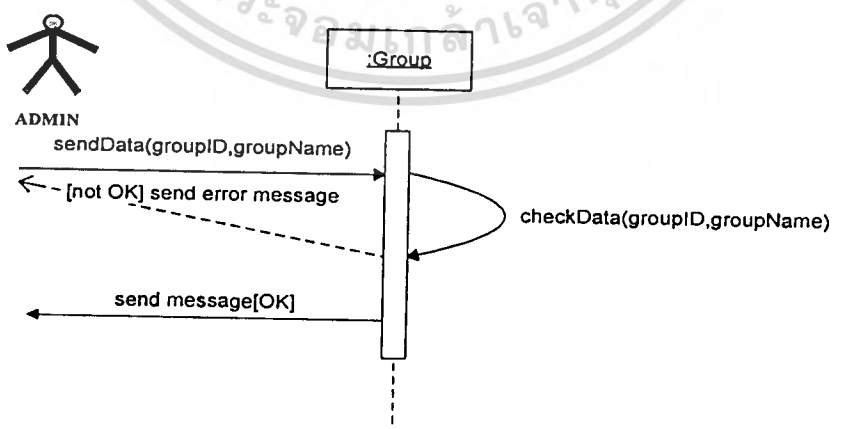
จากรูปเป็นการแสดงถึงการทำงานของ Create User Profile โดย ผู้ที่ได้รับสิทธ์ Admin เท่านั้นที่สามารถทำการสร้าง Login ให้กับ user รายใหม่โดย Admin จะต้องป้อนข้อมูล username ,password และ Group เพื่อจะระบุ Level ของ user แต่ละรายโดยข้อมูลที่ได้ป้อนเข้ามานั้น ส่วน

ของ userName จะต้องไม่ซ้ำกับข้อมูลเก่าที่มีอยู่ โดยหาก Admin ใส่ข้อมูล userName ที่เคยมีในระบบ ระบบจะแจ้งข้อผิดพลาดขึ้น



รูปที่ 3.16 Sequence Diagram ของ Login System

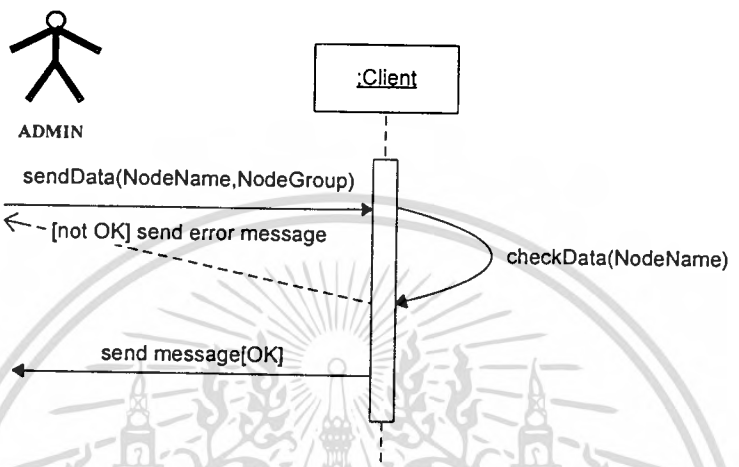
จากรูปเป็นการแสดงถึงการทำงานของ Login System โดยผู้ที่จะใช้ระบบจะต้องทำการ Login เข้าสู่ระบบ ซึ่งผู้ใช้จะต้องป้อนข้อมูล userName , password เข้าสู่ระบบ เพื่อให้ได้สิทธ์ใน level ต่างๆ โดยผู้ที่จะ Login เข้ามาได้จะต้องมีชื่อในระบบก่อนถ้า userName และ password นั้นมีในระบบ ผู้ใช้จะสามารถเข้าไปทำงานได้ตามสิทธ์ที่ตนมี หากไม่มีชื่อในระบบ ระบบส่งค่า error กลับไปเพื่อให้ผู้ใช้ทำการ Login เข้าสู่ระบบใหม่



รูปที่ 3.17 Sequence Diagram ของ Create Node Group

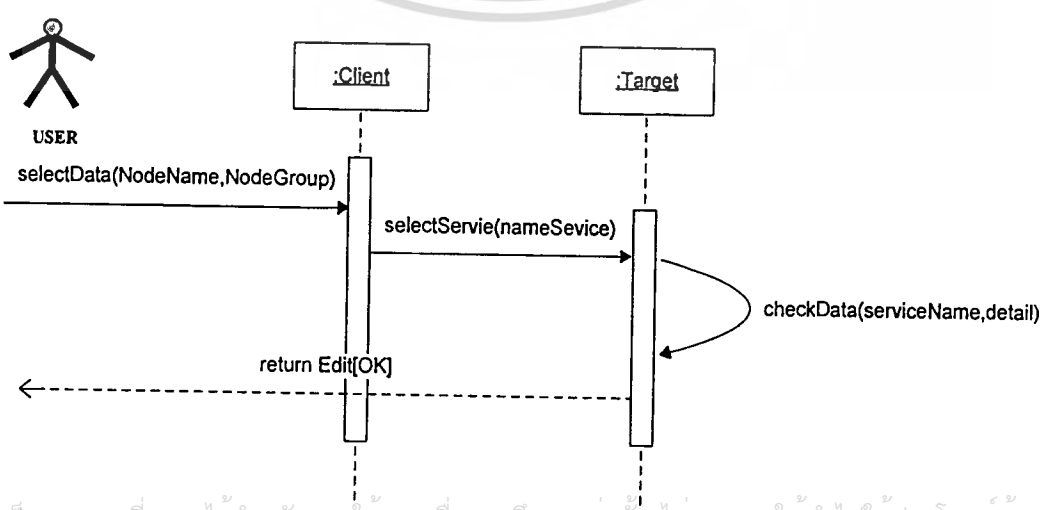
จากรูปเป็นการแสดงถึงการทำงานของ Create Node Group โดยผู้ที่ได้รับสิทธ์ Admin เท่านั้นที่สามารถทำการสร้าง Node Group ได้ โดย Admin จะต้องป้อนข้อมูล groupID และ

groupName โดยระบบจะทำการตรวจสอบว่ามีข้อมูล groupID และ groupName อยู่ในระบบหรือไม่ หากมีข้อมูลดังกล่าวอยู่ในระบบ ระบบจะทาการส่งค่า Error กลับไป โดยหากระบบไม่มีข้อมูลดังกล่าวในระบบ ระบบจะทำการบันทึกข้อมูลและ ส่งค่ากลับไปบอกว่าระบบทำการข้อมูลเรียบร้อยแล้ว



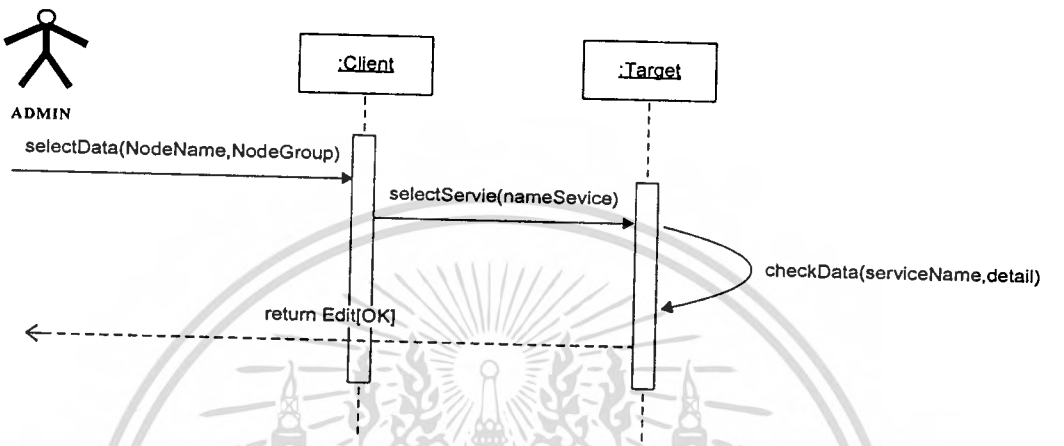
รูปที่ 3.18 Sequence Diagram ของ Create Node Client

จากรูปเป็นการแสดงถึงการทำงานของ Create Node Client โดยผู้ที่ได้รับสิทธ์ Admin เท่านั้นที่สามารถทำการสร้าง Node Client ได้โดย Admin จะต้องป้อนข้อมูล nodeName และทำการเลือก groupName โดยระบบจะทำการตรวจสอบว่ามีข้อมูล nodeName อยู่ในระบบหรือไม่ หากมีข้อมูลดังกล่าวอยู่ในระบบ ระบบจะทาการส่งค่า Error กลับไป โดยหากระบบไม่มีข้อมูลดังกล่าวในระบบ ระบบจะทำการบันทึกข้อมูลและ ส่งค่ากลับไปบอกว่าระบบทำการข้อมูลเรียบร้อยแล้ว



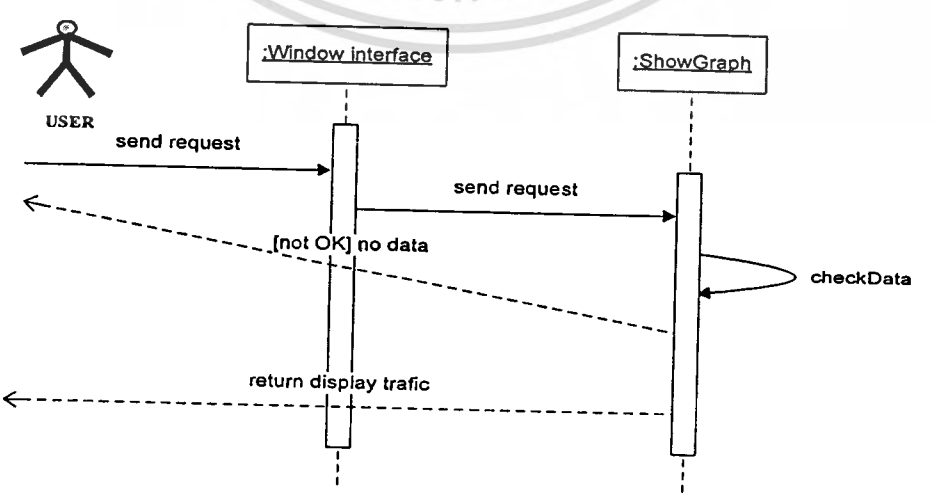
รูปที่ 3.19 Sequence Diagram ของ Edit Service Profile

จากรูปเป็นการแสดงถึงการทำงานของ Edit Service Profile โดยผู้ที่จะใช้จะต้องป้อนข้อมูล serviceName และ รายละเอียดของ Service ต่างๆโดยระบบจะทำการตรวจสอบว่ามีข้อมูล serviceName อยู่ในระบบหรือไม่ หากมีข้อมูลดังกล่าวอยู่ในระบบ ระบบจะทาการส่งค่า Error กลับไป โดยหากระบบไม่มีข้อมูลดังกล่าวในระบบ ระบบจะทำการบันทึกข้อมูลและ ส่งค่ากลับไปบอกว่าระบบทำการข้อมูลเรียบร้อยแล้ว



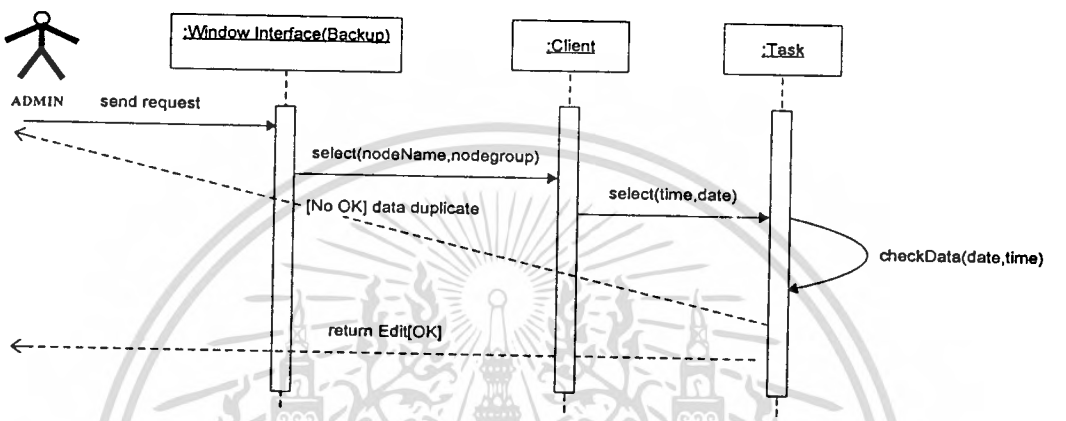
รูปที่ 3.20 Sequence Diagram ของ Create Client Service

จากรูปเป็นการแสดงถึงการทำงานของ Create Client Service โดยผู้ที่ได้รับสิทธ์ Admin เท่านั้นที่สามารถทำการสร้าง กำหนด Service ให้กับ Client Node หรือ group ของ Node ได้โดย Admin จะต้องเลือก Node Client หรือ Node Group จากนั้น Admin จะต้องทำการกำหนด Service ให้กับ Node Client หรือ Node Group หากมี service ที่ได้เลือกมาในระบบแล้ว ระบบจะทำการส่งค่า Error กลับไป โดยหาก Node Client หรือ Node Group ไม่มีข้อมูลดังกล่าวในระบบ ระบบจะทำการบันทึกข้อมูลและ ส่งค่ากลับไปบอกว่าระบบทำการข้อมูลเรียบร้อยแล้ว



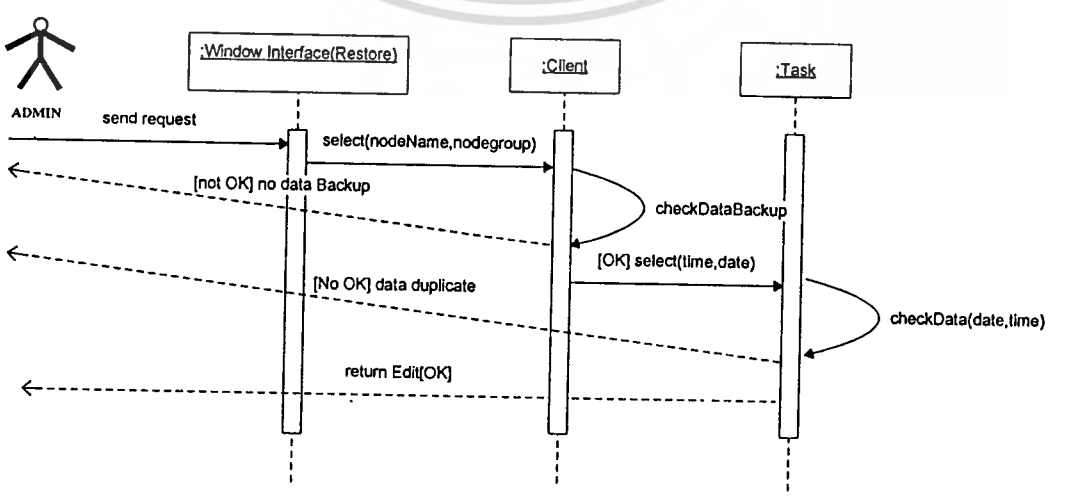
รูปที่ 3.21 Sequence Diagram ของ Monitor Service

จากรูปเป็นการแสดงถึงการทำงานของ Monitor Service โดยผู้ที่จะใช้ระบบจะต้องทำการ Login เข้าสู่ระบบ ซึ่งผู้ใช้จะต้องทำการเลือก Window Monitor และจากนั้นทำการ เลือก Service ที่ต้องการดูข้อมูลเช่น graph ผลการทดสอบ Traffic โดยระบบจะทำการตรวจสอบข้อมูลที่มีอยู่ในระบบ หากไม่มีข้อมูลระบบจะส่งค่ากลับไป เพื่อบอกว่าไม่มีข้อมูลในระบบ หากมีข้อมูลในระบบ ระบบจะทำการส่งค่ากลับไป เช่น Traffic ที่ได้ทดสอบ



รูปที่ 3.22 Sequence Diagram ของ Backup Config

จากรูปเป็นการแสดงถึงการทำงานของ Backup Config โดยผู้ที่ได้รับสิทธ์ Admin เท่านั้นที่สามารถทำการตั้งค่าการ Backup ข้อมูลได้ โดย Admin ต้องเรียก Window Backup ข้อมูลขึ้นมา จากนั้น Admin จะต้องทำการเลือก nodeName หรือ nodeGroup หลังจากนั้น Admin จะต้องทำการกำหนดเวลาที่ต้องการให้ระบบทำการBackup ข้อมูล โดยระบบจะทำการตรวจสอบว่ามีการตั้งเวลาของ Client ไว้ก่อนหน้าหรือยังหากมีแล้ว ระบบจะทำการส่งค่า Error ไปบอกเพื่อให้ Admin ทำการกำหนดใหม่ หากไม่มีข้อมูลในระบบ ระบบจะส่งค่ากลับไปบอกว่าทำการบันทึกข้อมูลเรียบร้อยแล้ว



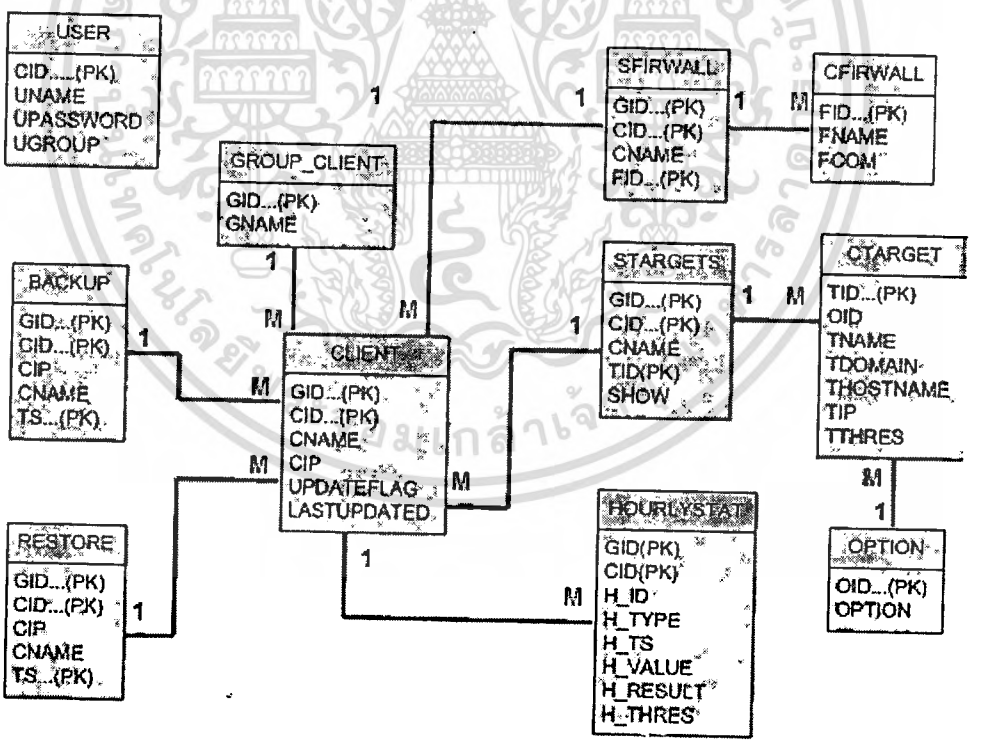
รูปที่ 3.23 Sequence Diagram ของ Restore Config

จากรูปเป็นการแสดงถึงการทำงานของ Restore Config โดยผู้ที่ได้รับสิทธ์ Admin เท่านั้นที่สามารถทำการตั้งค่าการ Restore ข้อมูลได้ โดย Admin ต้องเรียก Window Restore ข้อมูลขึ้นมา จากนั้น Admin จะต้องทำการเลือก nodeName หรือ nodeGroup โดยระบบจะทำการตรวจสอบข้อมูลในระบบก่อนว่ามีข้อมูลที่ ได้เลือกนั้นมีการ Backup ไว้หรือไม่ หากไม่มีข้อมูล Backup อยู่ระบบจะส่งค่า error กลับไปว่าไม่มีข้อมูลในระบบ หากมีข้อมุลอยู่ในระบบ แล้ว Admin จะต้องทำการกำหนดเวลาที่ต้องการให้ระบบทำการ Restore ข้อมูล โดยระบบจะทำการตรวจสอบว่ามีการตั้งเวลาของ Client ไว้ก่อนหน้าหรือยังหากมีแล้ว ระบบจะทำการส่งค่า Error ไปบอกเพื่อให้ Admin ทำการกำหนดใหม่ หากไม่มีข้อมูลในระบบ ระบบจะส่งค่ากลับไปบอกว่าทำการบันทึกข้อมูลเรียบร้อยแล้ว

3.4 โครงสร้างฐานข้อมูล

เนื่องจากในระบบการจัดการแบบมัลติเทอร์มินอลนั้นได้มีการเก็บข้อมูลออกเป็นสองส่วนด้วยกันคือ

3.4.1 โครงสร้างฐานข้อมูลในส่วนของตัวเซิร์ฟเวอร์



รูปที่ 3.24 แสดงโครงสร้างของ Database บน Server

การออกแบบฐานขอ้มูลได้แสดงความสัมพันธ์ของตารางต่างๆ ไว้ใน E/R Diagram ตามรูปด้านบนซึ่งประกอบด้วยตารางต่างๆ ดังนี้

ตารางที่ 3.1 ตาราง USER

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่	CID	INTEGER(10)	เพิ่มเป็น increment
ชื่อผู้ใช้	UNAME	VARCHAR(20)	User name ที่ใช้ในการเข้าระบบ
พาสเวอร์ด ผู้ใช้	UPASSWORD	VARCHAR(20)	Password ที่ใช้ในการเข้าระบบ
กลุ่มของผู้ใช้	UGROUP	VARCHAR(20)	มี 3 ระดับ SUPERADMIN , ADMIN , USER

ตารางที่ 3.2 ตาราง CLIENTS

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่Group	GID	INTEGER	อ้างอิงจาก Table GROUP
เลขที่Client	CID	INTEGER(10)	เพิ่มเป็น increment
ชื่อ ของ Client	CNAME	VARCHAR(255)	ชื่อของ Client
IP ของ Client	CIP	VARCHAR(26)	จะมีค่าเมื่อ Script มีการทำงาน
การรับข้อมูล	UPDATEFLAG	VARCHAR(1)	default จะเป็น N (offline) เป็น Y (online) ก็ต่อเมื่อมี server ได้รับข้อมูล
เวลารับข้อมูล	LASTUPDATED	VARCHAR(30)	อัพเดทก็ต่อเมื่อ server ได้รับข้อมูลของ client

ตารางที่ 3.3 ตาราง CTARGETS

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่	TID	VARCHAR(255)	เพิ่มเป็น increment อ้างอิงจาก
ประเภท service	OID	INTEGER	ประเภท service(PING,HTTP)
ชื่อของ TARGETS	TNAME	VARCHAR(255)	ชื่อของ TARGETS
ชื่อ DOMAIN	TDOMAIN	VARCHAR(255)	กำหนด Parameter สำหรับ Service HTTP และ DNS
ชื่อ HOSTNAME	THOSTNAME	VARCHAR(255)	กำหนด Parameter สำหรับ Service PING
IP ที่ใช้	TIP	VARCHAR(26)	กำหนด Parameter สำหรับ Service DNS
ค่า Threshold	TTHRES	INTEGER(38)	กำหนดค่า Threshold

ตารางที่ 3.4 ตาราง CFIREWALL

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่	FID	VARCHAR(255)	เพิ่มเป็น increment
ชื่อของกฎ	FNAME	VARCHAR(255)	ชื่อของกฎ
Command ที่ใช้	FCOM	VARCHAR(255)	Command ที่ใช้

ตารางที่ 3.5 ตาราง GCLIENT

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่	GID	INTEGER(10)	ตั้งค่า GroupID
ชื่อของ GROUP	GNAME	VARCHAR(255)	ชื่อของ GROUP

ตารางที่ 3.6 ตาราง SFIREWALL

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่ GROUP	GID	INTEGER(11)	เพิ่มเป็น increment
เลขที่ CLIENT	CID	INTEGER(11)	ลำดับของคำสั่ง
ชื่อ CLIENT	CNAME	VARCHAR(30)	ชื่อ CLIENT
เลขที่ของ FID	FID	INTEGER(11)	เลขที่ของ FID

ตารางที่ 3.7 ตาราง STARGETS

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่ GROUP	GID	INTEGER(11)	เพิ่มเป็น increment
เลขที่ CLIENT	CID	INTEGER(11)	ลำดับของคำสั่ง
ชื่อ CLIENT	CNAME	VARCHAR(30)	ชื่อ CLIENT
เลขที่ของ FID	TID	INTEGER(11)	เลขที่ของ TID
บอกการแสดงกราฟ	SHOW	INTEGER(2)	บอกว่าต้องการแสดงกราฟ

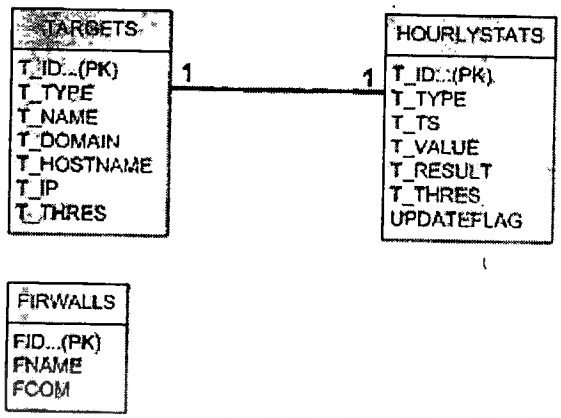
ตารางที่ 3.8 ตาราง OPTION

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่	OID	INTEGER(10)	ตั้งค่า OID
ชื่อของ OPTION	OPTION	VARCHAR(255)	ชื่อของ OPTION

ตารางที่ 3.9 ตาราง HOURLYSTAT

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่	TID	VARCHAR(255)	เพิ่มเป็น increment อ้างอิงจาก
Time stamp	TS	VARCHAR(30)	Format YYYYMMDDHHmmss
เลขที่ GROUP	GID	INTEGER(11)	เพิ่มเป็น increment
เลขที่ CLIENT	CID	INTEGER(11)	ลำดับของคำสั่ง
เลขที่	OID	INTEGER(10)	ตั้งค่า OID
ค่าที่ได้	VALUE	INTEGER(38)	ค่าที่ได้จากการทดสอบ
ประเภทข้อมูล	RESULT	INTEGER(3)	ประเภทข้อมูล
ค่า Threshold	THRES	INTEGER(38)	กำหนดค่า Threshold

3.4.2 โครงสร้างฐานข้อมูลในส่วนของตัวปลายทาง (Client)



รูปที่ 3.25 แสดง Database ในส่วนของ Client

การออกแบบฐานขอ้มูลได้แสดงความสัมพันธ์ของตารางต่างๆ ไว้ใน E/R Diagram ตามรูปด้านบนซึ่งประกอบด้วยตารางต่างๆ ดังนี้

ตารางที่ 3.10 ตาราง HOURLYSTATS

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่	T_ID	bigint(20) unsigned	เพิ่มเป็น increment
ประเภทข้อมูล	T_TYPE	smallint(5) unsigned	กำหนด Parameter
Timestamp	T_TS	int(10) unsigned	YYYYMMDDHHmmss
ผลที่วัดได้	T_VALUE	bigint(20) unsigned	ผลที่วัดได้
ประเภทข้อมูล	T_RESULT	smallint(5) unsigned	ประเภทข้อมูล
ค่า Threshold	T_THRES	bigint(20) unsigned	กำหนดค่า Threshold

ตาราง 3.11 ตาราง FIREWALLS

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่	FID	INT(10)UNSIGNED	เพิ่มเป็น increment
ชื่อดับของคำสั่ง	FNAME	VARCHAR(30)	ชื่อดับของคำสั่ง
command ที่ใช้	FCOM	INTEGER(38)	command ที่ใช้

ตาราง 3.12 ตาราง TARGETS

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่	T_ID	INT(10)UNSIGNED	เพิ่มเป็น increment
ชื่อ TARGETS	T_NAME	VARCHAR(255)	ชื่อของ TARGETS
ชื่อ DOMAIN	T_DOMAIN	VARCHAR(255)	Service HTTP
ชื่อ HOSTNAME	T_HOSTNAME	VARCHAR(255)	Service PING
IP ที่ใช้	T_IP	VARCHAR(16)	IP ที่ใช้
ค่า Threshold	T_THRES	BIGINT(20)UNSIGNED	กำหนดค่า Threshold
ชนิด OPTION	T_TYPE	SMALLINT(5)UNSIGNED	service แต่ละชนิด

บทที่ 4
การพัฒนาโปรแกรม

4.1 การวางแผนปฏิบัติงาน

ได้เลือกใช้ระบบปฏิบัติการและซอฟแวร์ดังต่อไปนี้

1. **ระบบปฏิบัติการเลือกใช้ Linux**

เป็นระบบปฏิบัติการยูนิกซ์ที่มีการใช้งานกันอย่างแพร่หลาย และไม่ต้องเสียค่าใช้จ่าย เพราะเป็นระบบยูนิกซ์แบบ Open Source ทั่วๆไป

2. **เว็บเซิร์ฟเวอร์ที่รองรับการทำงานของ PHP และ MySQL เลือกใช้ Apache**

Apache เป็นซอฟต์แวร์ที่สำหรับให้บริการเว็บเซิร์ฟเวอร์ (HTTP/Web Server) ผ่านโปรโตคอล HTTP โดยเป็นซอร์ฟแวร์แบบ Open Source สามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่าย

3. **ซอฟต์แวร์ภาษา เลือกใช้ PHP**

PHP Extension เป็น scripting language ที่ทำงานร่วมกับ HTML โดย PHP เป็น Open Source ที่สามารถใช้ ได้กับหลายๆ ระบบปฏิบัติการ ไม่ว่าจะเป็น windows, Unix, Linux, และยังรองรับการติดต่อกับฐานข้อมูลได้หลายชนิด

4. **ซอฟต์แวร์ระบบฐานข้อมูลเลือกใช้ MySQL**

MySQL เป็นโปรแกรมฐานข้อมูลที่นิยมใช้งานกันอย่างแพร่หลาย และมีประสิทธิภาพ สามารถนำมาใช้ทดแทนโปรแกรมฐานข้อมูลที่มีจำหน่ายในเชิงพานิชย์ได้

5. **ซอฟต์แวร์ที่ใช้ในการจัดทำกราฟเลือกใช้ JP Graph**

JP Graph เป็นซอฟต์แวร์ ทีได้พัฒนาจาก ซอฟต์แวร์ภาษา PHP โดยสามารถแสดงแผนภาพได้หลายรูปแบบ และเป็น Open Source สามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่าย

6. **ซอฟต์แวร์ไฟร์วอลล์เลือกใช้ iptables**

เป็นโปรแกรมไฟร์วอลล์ที่พัฒนาบน Linux Kernel 2.4 มีความสามารถในการตรวจสอบสถานะการทำงานของทราฟิกส์บนแอพพลิเคชั่นต่างๆ ได้อย่างดี สนับสนุนการทำงานแบบ SPI (Stateful Inspection)เป็นระบบปฏิบัติการยูนิกซ์และเป็นระบบยูนิกซ์แบบ Open Source

7. **ฟังก์ชัน HTTP จะใช้ Class httpclient.php**

เข้ามาช่วยซึ่งสามารถ ทำการ download จาก อินเตอร์เน็ตมาใช้งานได้ โดยจะนำมาช่วยในการดึงหน้า Page content จาก server ของ client และการหาขนาดของหน้า page เพื่อนำมาคำนวณหาแบนด์วิธในในการเรียกใช้หน้า page ว่ามีความเร็วเป็นเท่าใด(bps)

8. **ฟังก์ชัน PING จะใช้ class_ICMP**

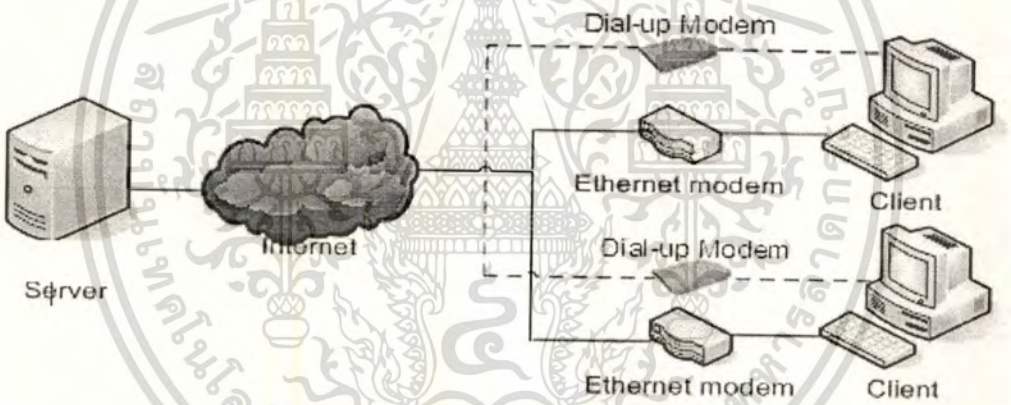
เข้ามาช่วยเพื่อเป็น การกำหนดการทดสอบว่าจะทำการทดสอบเป็นจำนวนกี่ครั้ง จาก โครงการจะใช้การทดสอบจำนวน 5 ครั้งในการเก็บข้อมูลในครั้งหนึ่งๆ

9. **การเข้าใช้งาน รวมถึงการ Backup ข้อมูล**

จะใช้ช่องทางของ secure channel โดยฟังก์ชันดังกล่าวจะมีอยู่แล้วในระบบปฏิบัติการ Linux โดยจะนำ ระบบ SSH เข้ามาช่วยในการ remote เข้าไปยัง client แต่ละตัวแทนการใช้งาน Telnet และจะใช้ฟังก์ชัน SCP ในการ backup ข้อมูลแทน การ ftp ปกติโดยในการใช้งานฟังก์ชัน SCP จะต้องมีการนำ public key บน client ทุกตัวมาเก็บที่ server และจะต้องนำ public key ของ server มาเก็บไว้ที่ client ทุกตัวด้วย

4.2 การจัดการรูปแบบในการทดสอบ

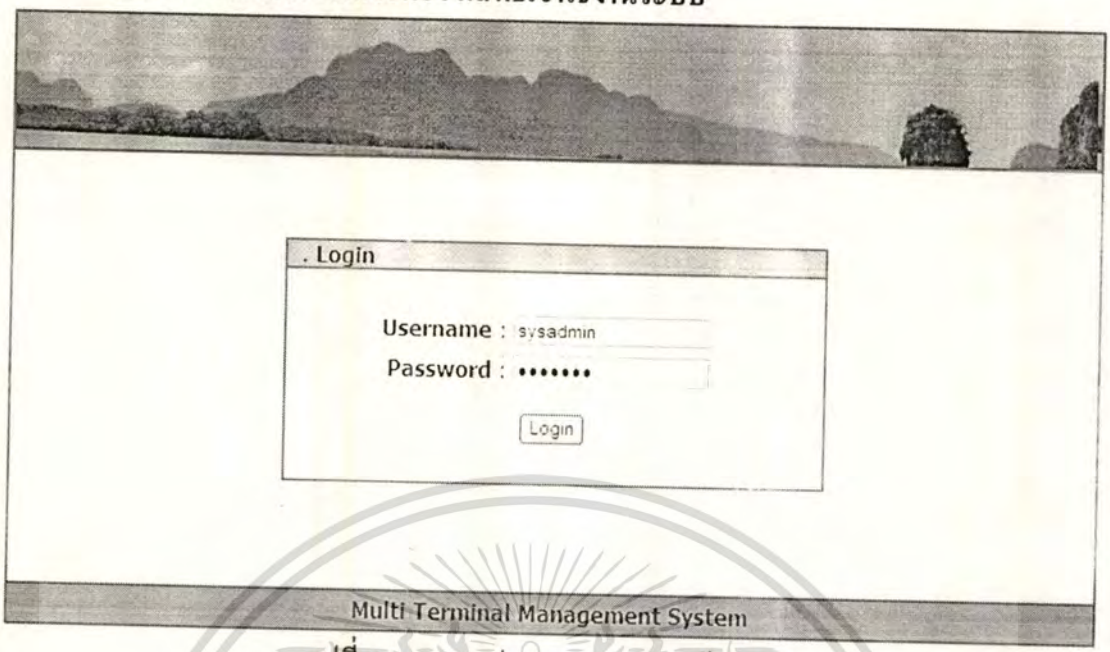
4.2.1 การทดสอบการทำงานของฟังก์ชันไฟร์วอลล์ และการจัดการทราฟิกส์



รูปที่ 4.1 การทดสอบการทำงานของการ Monitor Traffic

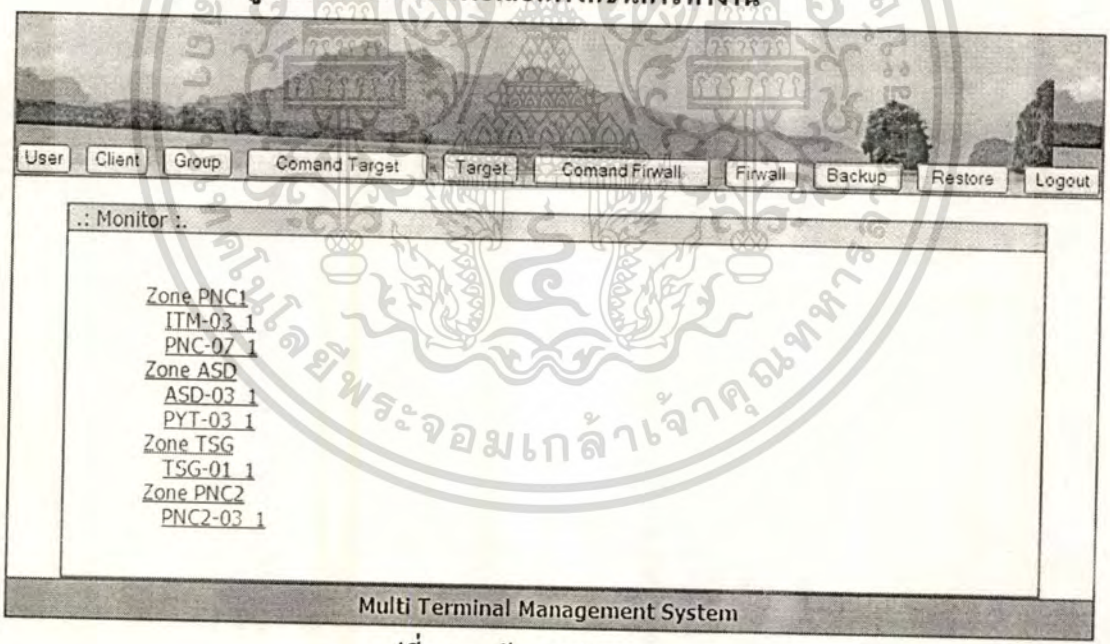
จากรูปเป็นการจัดการการทำงานของการตรวจสอบคุณภาพของเครือข่ายที่มีการนำเครื่อง Client ไปติดตั้งไว้ในแต่ละที่ โดย Client ทั้งหมดจะต้องมีการติดต่อกับเครื่อง Server เพื่อทำการรับ – ส่งข้อมูลการทำงานระหว่างเครื่อง Client และ Server

4.2.1.1 ก่อนเข้าใช้งานจะต้อง มีสิทธิเข้าใช้งานระบบ



รูปที่ 4.2 แสดงรูปแบบการ Login เข้าระบบ

4.2.1.2 เข้าสู่หน้าหน้าจอหลักเพื่อเลือกฟังก์ชันการทำงาน

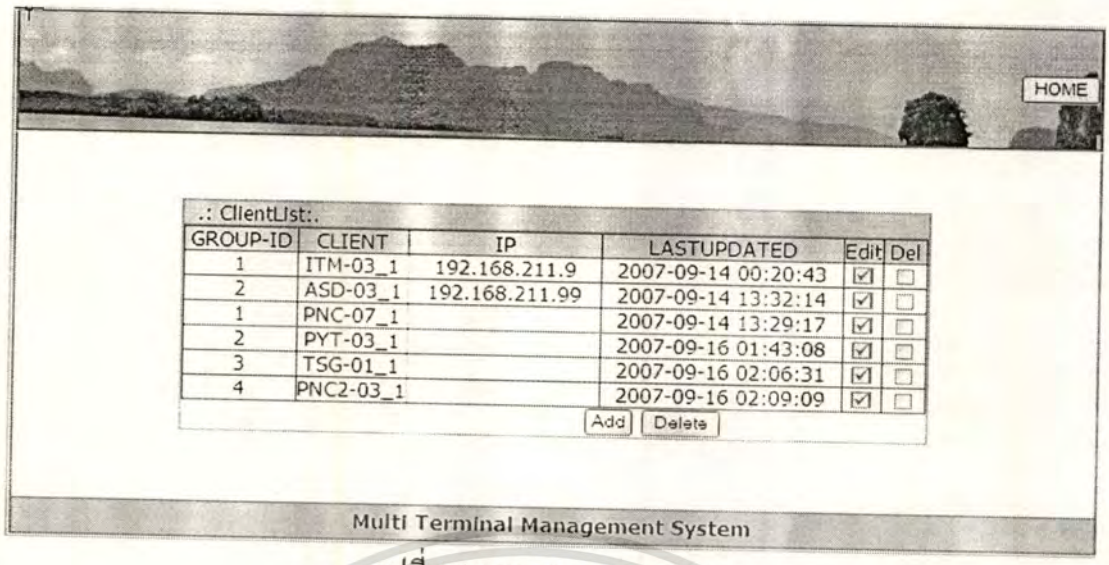


รูปที่ 4.3 หน้าจอหลักของระบบ

4.3 ฟังก์ชันการทำงาน จะมีฟังก์ชันดังนี้

ฟังก์ชัน TARGETS เป็นการกำหนดการทำงานให้กับเครื่อง Client แต่ละตัวทำงานตามที่เรากำหนดได้ โดยตัวระบบสามารถที่จะกำหนดการทำงานได้ดังนี้

- สามารถกำหนดการสร้าง Group ของ Client โดยสามารถสร้าง Group ใหม่ได้
- สามารถสร้าง Client ได้แต่ Client นั้นจะต้องมีการกำหนดชื่อใน Database ให้ตรงกับหน้างานที่มีอยู่จริง(ตาม Location)

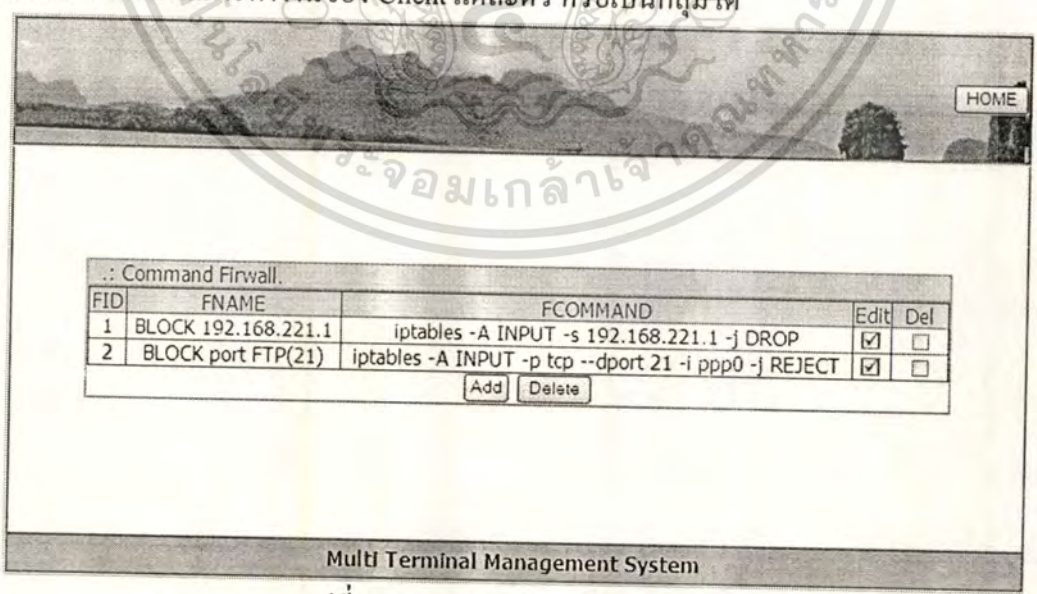


รูปที่ 4.4 แสดง List ของ Client

จากรูปที่ 4.4 แสดง Client List ซึ่งจะเป็น List ของ Client ที่ถูกสร้างขึ้นมา โดย จากรูปจะแสดงชื่อของ Client , IP address ของ Client และ เวลาที่ Update ล่าสุดที่ Clint นั้นยังสามารถจัดการได้ค่าแต่ละค่าได้มาโดย หลังจาก Client หน้างานที่มีอยู่จริง(ตาม Location site) สามารถที่จะใช้งานอินเตอร์เน็ตได้ Client จะทำการ Post ข้อมูล คือ IP Address ,Note Name ,และ Time stamp ที่ update ล่าสุดส่งมาที่ Server หลังจากนั้น server จะนำข้อมูลนั้น เก็บใส่ฐานข้อมูลที่สร้างขึ้นก่อนหน้านี้ตามชื่อ และ Group name ที่ตั้งไว้หากไม่ตรงกับ Client หน้างานก็จะไม่สามารถ ใส่ข้อมูลในฐานข้อมูลได้

4.3.1 สามารถสร้างกฎต่างๆ ทั้งส่วนของ Firewall

โดยสามารถกำหนดการทำงานของ Client แต่ละตัว หรือเป็นกลุ่มได้

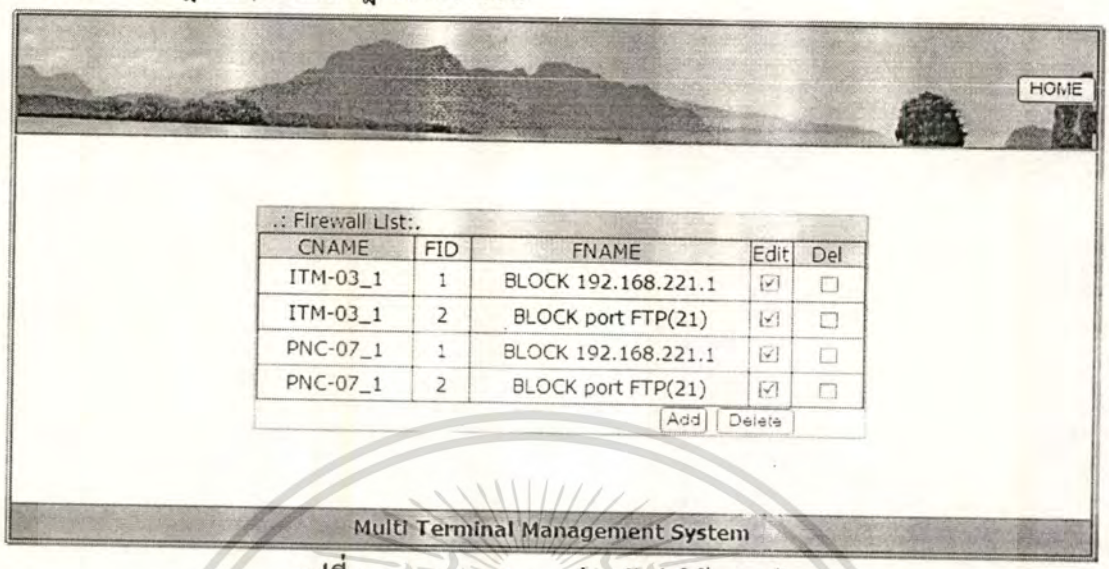


รูปที่ 4.5 List ของ Firewall command

จากรูปจะที่ 4.5 แสดง Firewall command ที่เราสร้างขึ้นบน Server โดยเราสามารถสร้างได้ตามหลักการการสร้างตาม iptables

- สามารถกำหนดการทำงานของ Client แต่ละตัวให้ทำงานตามกฎที่กำหนด โดยจากรูป

ที่ 4.7 จะแสดงการกำหนดคำสั่งต่าง ๆ ให้กับ client แต่ละตัวตามสิ่งที่ผู้ดูแลระบบต้องการ โดยสามารถเลือกกฎต่างๆ ได้จากกฎที่เราสร้างขึ้น



รูปที่ 4.6 แสดง Client แต่ละตัวว่าใช้กฎอะไร

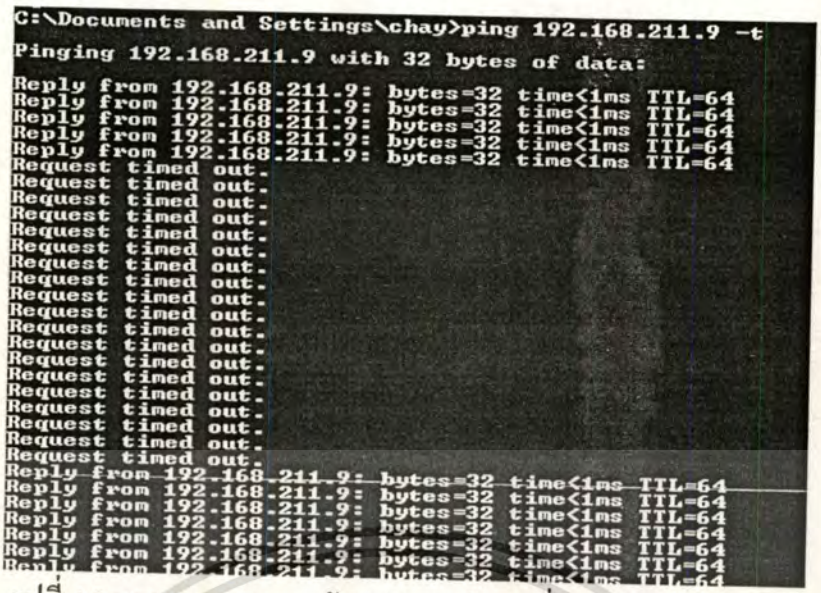
ทำการตรงจสอบที่ Client เมื่อ Client สามารถติดต่อผ่านอินเตอร์เน็ตได้ Client จะดึงข้อมูลของกฎที่สร้างขึ้นไปเก็บไว้ที่ ฐานข้อมูลของ Client แต่ละตัว ซึ่ง command ของ Client แต่ละตัวอาจจะแตกต่างกันก็ได้ตามสิ่งที่ผู้ดูแลระบบสร้างมา

```
mysql>
mysql> select * from FTARGETS;
+------+--------------------+-----------------------------------------------------+
| F_ID | F_NAME             | F_COM                                               |
+------+--------------------+-----------------------------------------------------+
|    2 | BLOCK port FTP(21) | iptables -A INPUT -p tcp --dport 21 -i ppp0 -j REJECT |
|    1 | BLOCK 192.168.221.1 | iptables -A INPUT -s 192.168.221.1 -j DROP          |
+------+--------------------+-----------------------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

รูปที่ 4.7 แสดงฐานข้อมูลกฎของ Firewall ที่ Client

จากรูปที่ 4.7 แสดงกฎของไฟร์วอลล์ที่ Client หลังจากที่ Client สามารถติดต่อกับ Server เพื่อดึงข้อมูล

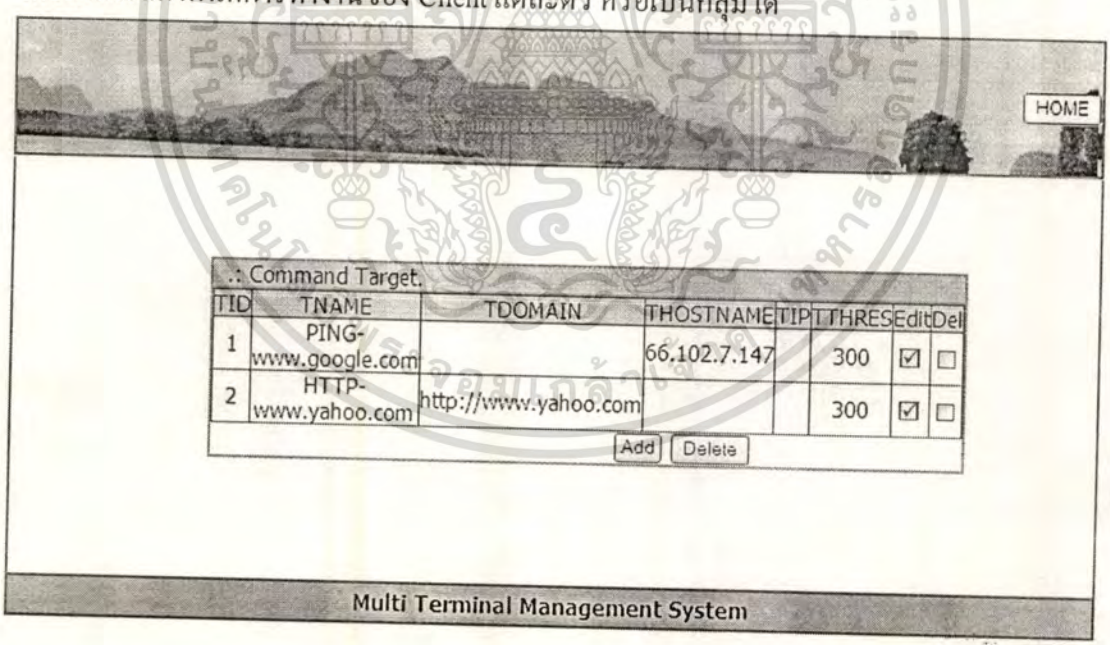


รูปที่ 4.8 ผลการทดสอบบนตัว Server หลังจากที่ Client ทำงานตามกฎ

จากรูปที่ 4.8 แสดงผลการทดสอบบนตัว Server หลังจากที่ Client ทำงานตามกฎ โดยตัว server จะเป็น ip address 192.168.211.1 ซึ่งจะถูก Drop packet ลงตามกฎข้อที่ 1

4.3.2 สามารถสร้างการทดสอบการจัดการทราฟิกส์ได้

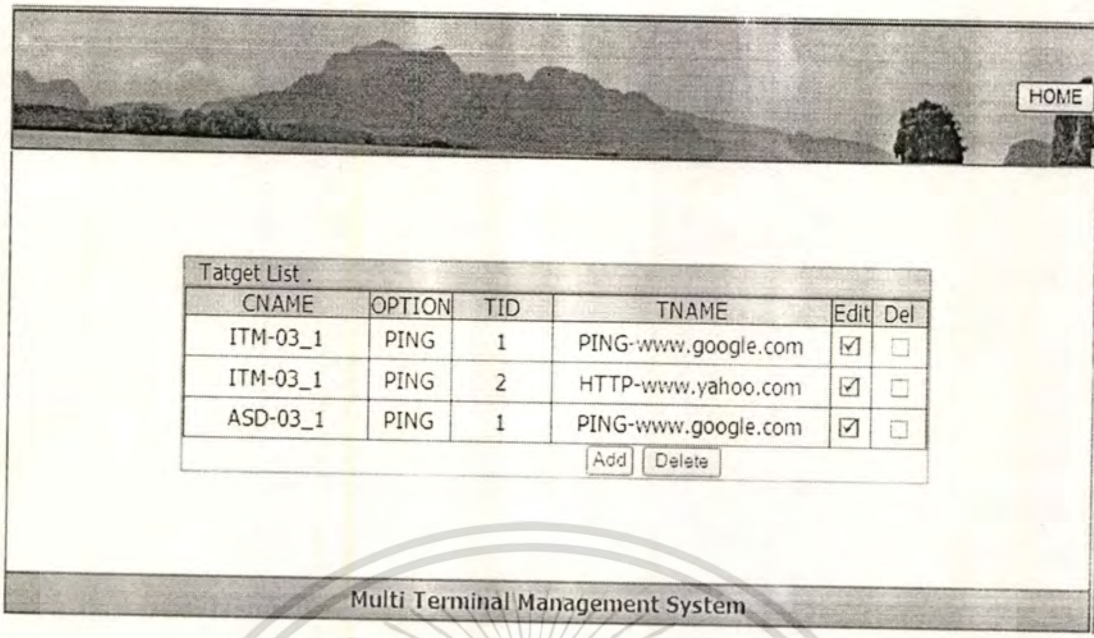
โดยสามารถกำหนดการทำงานของ Client แต่ละตัว หรือเป็นกลุ่มได้



รูปที่ 4.9 List ของ Target ที่ต้องการทดสอบ

จากรูปจะที่ 4.9 แสดง Target List ที่เราสร้างขึ้นบน Server โดยเราสามารถสร้างได้ว่าเราต้องการจะทดสอบอะไร

- สามารถกำหนดการทำงานของ Client แต่ละตัวให้ทำงานตามสิ่งที่กำหนด โดยจากรูปที่ 4.11 จะแสดงการกำหนดให้ Client ไปทดสอบการทำงานยังที่ต่างๆ



รูปที่ 4.10 แสดงการทดสอบทราฟิกส์ ของ Client

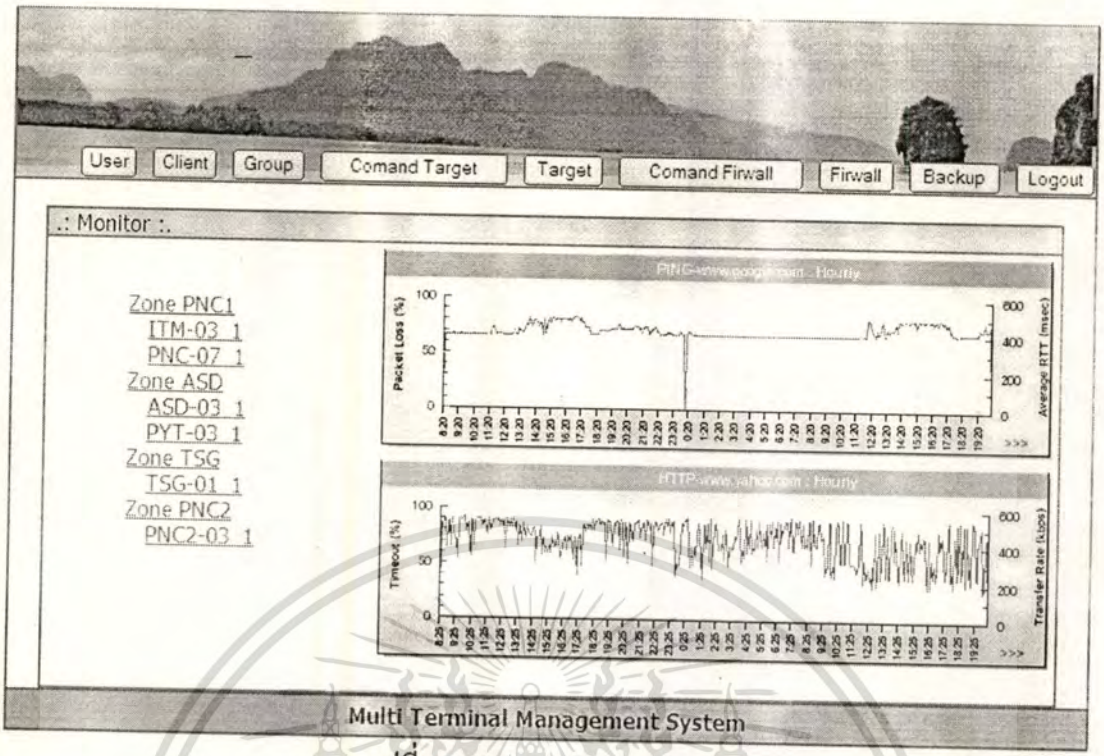
ทำการตรวจสอบที่ Client เมื่อ Client สามารถติดต่อผ่านอินเตอร์เน็ตได้ Client จะดึงข้อมูลของกฎที่สร้างขึ้นไปเก็บไว้ที่ ฐานข้อมูลของ Client แต่ละตัว ซึ่ง command ของ Client แต่ละตัวอาจจะแตกต่างกันกันก็ได้ตามสิ่งที่ผู้ดูแลระบบสร้างมา

```
mysql>
mysql> select * from TARGETS;
+------+---------------------+---------------------+---------------+------+---------+-------+----------+--------+--------+--------+-
---------+
| T_ID | T_NAME              | T_DOMAIN            | T_HOSTNAME    | T_IP | T_THRES | T_DNS | T_RADIUS | T_HTTP | T_MAIL | T_PING |
T_OPTION |
+------+---------------------+---------------------+---------------+------+---------+-------+----------+--------+--------+--------+-
---------+
|    2 | HTTP-www.yahoo.com  | http://www.yahoo.com |              |      |     300 | NULL  |     NULL |      1 |   NULL |   NULL |
       3 |
|    1 | PING-www.google.com |                     | 66.102.7.147  |      |     300 | NULL  |     NULL |   NULL |   NULL |      1 |
       5 |
+------+---------------------+---------------------+---------------+------+---------+-------+----------+--------+--------+--------+-
---------+
2 rows in set (0.00 sec)

mysql>
```

รูปที่ 4.11 แสดงฐานข้อมูลของสิ่งที่ต้องการทดสอบ ที่ Client

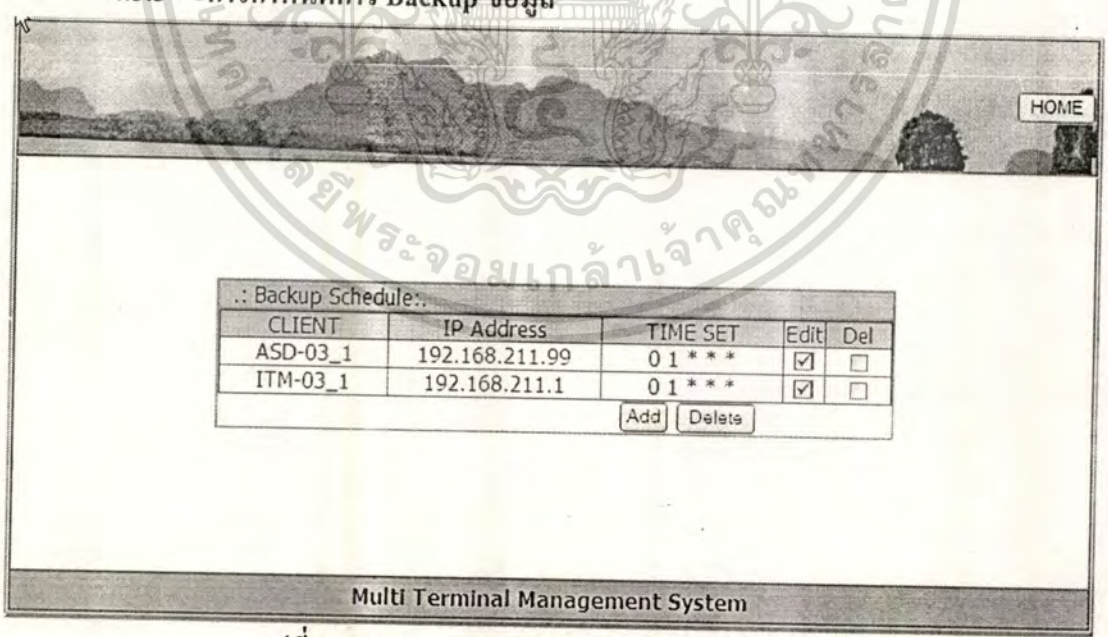
จากรูปที่ 4.11 แสดงการกำหนดการทำงานของ Client หลังจากที่ Client สามารถติดต่อกับ Server เพื่อดึงข้อมูล



รูปที่ 4.12 แสดงผลการทดสอบ

จากรูปที่ 4.12 แสดงผลการทดลอง โดยข้อมูลจะถูกส่งจาก Client มาที่ Server เพื่อทำการเขียนกราฟ

4.3.3 การกำหนดการ Backup ข้อมูล



รูปที่ 4.13 แสดง Schedule Backup ของ Node Client

จากรูปที่ 4.13 เป็นการตั้งการทำงานของการเก็บข้อมูล ของ Client แต่ละตัว โดยการตั้งการทำงานจะอ้างอิงเวลาจาก การ set crontab บนระบบ Linux โดยการตั้งค่าเป็นดังนี้

ตัวอย่าง 0 1 * * *

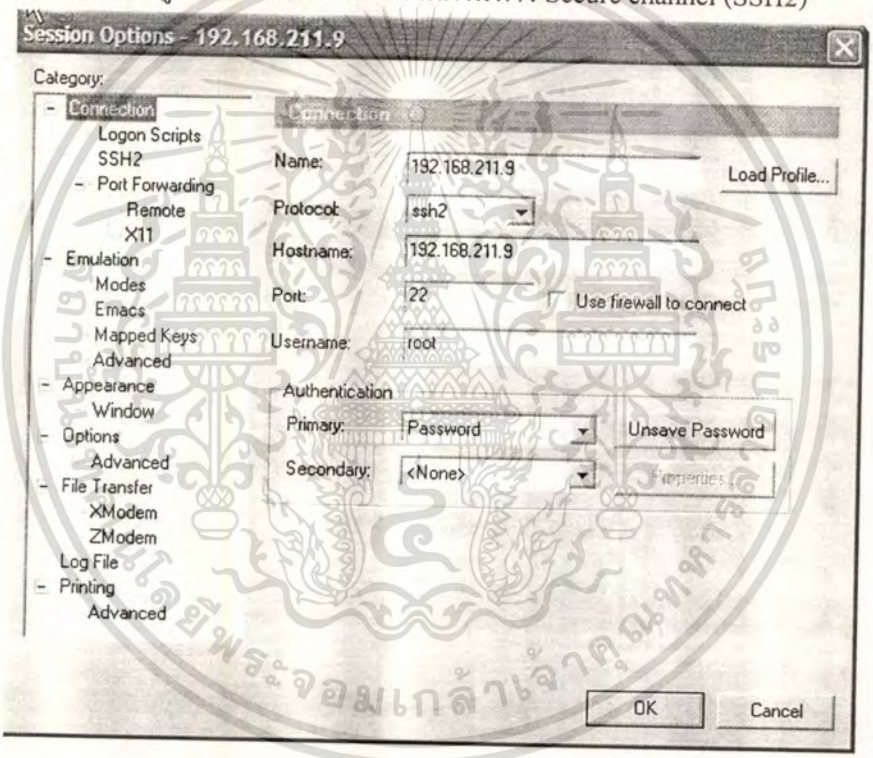
"Minutes" "Hours" "Day of month" "Month" "Day of week"

Minutes	กำหนดนาทีมีค่าตั้งแต่ 0 -59
Hours	กำหนดชั่วโมงมีค่าตั้งแต่ 0 -23 โดย 0 = เที่ยงคืน
Day of month	กำหนดวันที่ของเดือนมีค่าตั้งแต่ 0 -31
Month	กำหนดเดือนมีค่าตั้งแต่ 1-12
Day of week	กำหนดวันในสัปดาห์มีค่าตั้งแต่ 0 - 6 โดย 0 = วันอาทิตย์

จากตัวอย่างจะเป็นการกำหนดการเก็บข้อมูลช่วงเวลา 1 นาฬิการของทุกๆ วัน โดยฟังก์ชันการทำงานนี้จะมีการตรวจสอบสถานะของ client ก่อนทุกครั้งที่ทำงาน

4.3.4 การ Login เข้าสู่ Client

การ Login เข้าสู่ Client จะต้องใช้งานได้ผ่านทาง Secure channel (SSH2)



รูปที่ 4.14 แสดง Tool Secure channel

จากรูปที่ 14,5 จะเป็นการแสดง Tool ที่สามารถใช้ในการ Remote ไปยัง Client โดย Client จะเปิดฟังก์ชันการ Remote แบบ Secure channel เท่านั้น

สรุปการทำงานของ ระบบการจัดการแบบมัลติเทอร์มินอล สามารถควบคุมการทำงานของ client หลายๆ ตัวได้จากส่วนกลางผ่านระบบเครือข่ายอินเตอร์เน็ต โดยสามารถจัดการกับระบบไฟร์วอลล์ และระบบสามารถทำการทดสอบค่าทราฟิกส์ของไคลเอนต์ในแต่ละที่ได้ ซึ่งผู้ใช้สามารถที่จะนำไปประยุกต์สู่การทำงานในรูปแบบ อื่นๆ เช่นการจัดการการทำงานของปลายทางในรูปแบบอื่นๆ ในอนาคต

บทที่ 5
บทสรุปและแนวทางการพัฒนาในอนาคต

5.1 บทสรุปของโครงการ

- ระบบสามารถแสดง แสดงทราฟฟิกส์ ในการทดสอบของ client แต่ละตัวได้
- Server สามารถกำหนดการทำงานของ client ในแต่ละตัวได้ หรือกำหนดการ ทำงานให้กับ client เป็นกรุปได้
- สามารถกำหนดการทำงานของฟังก์ชันไฟร์วอลล์ได้
- ระบบสามารถทำการ Backup ข้อมูล ได้ผ่านทาง secure channel (SCP)
- การ remote เข้าไปยัง client จะทำงานผ่าน secure channel(SSH)

5.2 ข้อจำกัด และปัญหา

- ต้องติดตั้ง Software ที่ client ตามฟังก์ชันการทำงานทุกฟังก์ชันการทำงาน
- ผู้ใช้งานจะต้องมีความรู้ด้านการกำหนดการทำงานของไฟร์วอลล์
- หากระบบอินเตอร์เน็ต มีปัญหาหรือหลุดบ่อย จะมีปัญหาเนื่องจากระบบการจัดการแบบมัลติเทอร์มินอลเป็นการจัดการผ่านทาง เครือข่ายอินเตอร์ เน็ต โดยการทำงานนั้นจะมีการติดต่อกันระหว่างไคล์เอนต์ และเครื่องเซิร์ฟเวอร์ตลอดเวลา ดังนั้น หากระบบเครือข่ายอินเตอร์เน็ตมีปัญหา หรือ เครื่องไคล์เอนต์ เชื่อมต่อกับเครือข่ายอินเตอร์เน็ต ที่ช้า หรือ Speed ต่ำ หรือหลุดบ่อย อาจจะทำให้การส่งข้อมูลกันระหว่างไคล์เอนต์และเครื่องเซิร์ฟเวอร์มีปัญหาทำให้ไม่สามารถควบคุมการทำงานของไคล์เอนต์ได้ การส่งข้อมูลอาจจะไม่ครบตาม Process ที่ต้องการ
- หากมีไคล์เอนต์ เป็นจำนวนมากขึ้น ควรคำนึงถึง CPU ของเครื่องเซิร์ฟเวอร์ ที่จะต้องมีการประมวลผล รวมถึงการสร้างกราฟ จึงควรที่จะขยายให้ CPU มีความเร็วในการทำงานมากขึ้น หรือแยกฟังก์ชันการทำงานออกจากกัน

5.2 แนวทางในการพัฒนาในอนาคต

เนื่องจากโครงงานนี้มีจุดประสงค์เพื่อแสดงถึงการจัดการกับอุปกรณ์ปลายทาง ผ่านทางเครือข่ายอินเตอร์เน็ต อาจจะไม่ครอบคลุมฟังก์ชันการทำงานทั้งหมดดังนั้นจึงสามารถนำโครงงานนี้ไปพัฒนาต่อ ในเรื่องของฟังก์ชันการทำงานแบบอื่นๆ เพื่อให้มีการทำงานที่ครอบคลุมและมีการใช้งานให้ง่ายยิ่งขึ้นใช้งานได้รวดเร็ว เพิ่มฟังก์ชันการทำงานในส่วนอื่นๆเข้าไป เช่นการตั้งค่า

Proxy การตั้งค่าการ Update ไวรัส การตั้งค่า Policy ต่างๆ จากตัวเซิร์ฟเวอร์ เพื่อควบคุมการทำงานของ ไคล์เอนต์ เพื่อทำให้ ไคล์เอนต์ ทำงานได้ครอบคลุมทั้งระบบ

บรรณานุกรม

ก่อกิจ วีระอาชากุลม . 2529 . **ติดตั้งและปรับแต่งเซิร์ฟเวอร์ Linux สำหรับ Admin Linux โดยเฉพาะ** . กรุงเทพ ฯ: อินโฟเพรส

กิติศักดิ์ เจริญโภคานนท์, 2548 . **คู่มือเรียนเขียนเว็บอีคอมเมิร์ซด้วย PHP5 ครอบคลุมเวอร์ชั่นล่าสุด 5.1**. กรุงเทพ ฯ: ซัคเซส มีเดีย

ธัญลักษณ์ ผังชัยมงคล, 2548 . **"การพัฒนายูสเซอร์อินเตอร์เฟสแบบเว็บสำหรับไอพีไฟร์วอลล์ของ Free BSD"**, โครงการพัฒนาระบบงาน วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

บัณฑิต จามรภูติ, 2547 . **คัมภีร์ RedHat Enterprise Linux . เล่มที่ 2** . กรุงเทพ ฯ: ซัคเซส มีเดีย

James F, Kurose and Keith W. Ross . 2001 . **Computer Networking** . Boston : Addison Wesley.

John W, Satzinger, Robert B, Jackson, Stephen D. Burd . 2004. **System Analysis & Design IN A CHIANGING WORD**. Boston : Thomson/Course Technology

Peter Rob, Carlos Coronel. 2004. **Database System: DESIGN, IMPLEMENTZATION , MANAGEMENT** . Boston : Thomson/Course Technology

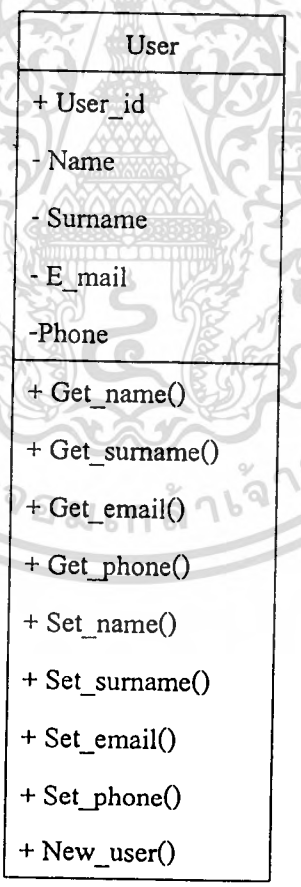
ภาคผนวก ก

หลักการแปลงคลาสไปเป็นตาราง

การแปลงคลาสไปเป็นตาราง คือ การวิเคราะห์จากความสัมพันธ์ เช่น แอสโซซิเอชั่น, เจเนอรัลไรเซชั่นและคอมโพสซิชัน เพื่อแปลงเป็นตาราง

- **หลักในการแปลงคลาสให้เป็นตาราง**

1. กำหนดให้แอตทิบิวต์ตัวใดตัวหนึ่งหรือกลุ่มใดกลุ่มหนึ่งเป็นคีย์หลัก
2. สร้างตารางที่มีทุกแอตทิบิวต์ของคลาสนั้นและมีคีย์หลักตามที่กำหนดแล้ว
3. แอตทิบิวต์หรือกลุ่มของแอตทิบิวต์ที่เป็นคีย์หลักต้องถูกกำหนดเป็น Not Null เสมอ
4. สำหรับแอตทิบิวต์อื่น ๆ ที่ไม่ได้ถูกเลือกให้เป็นคีย์หลัก ให้พิจารณาว่าแอตทิบิวต์ใดเป็นนัล (Null) ได้และแอตทิบิวต์ใดเป็นนัลไม่ได้
5. ในการออกแบบตารางไม่ต้องสนใจในส่วนของฟังก์ชั่น ให้สนใจที่แอตทิบิวต์เท่านั้น

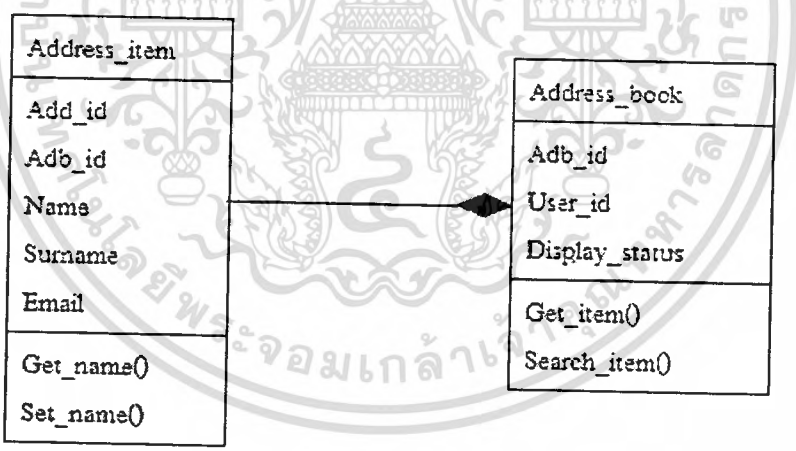


รูปที่ 1 แสดงคลาสตัวอย่าง 1

จากรูป 1 สามารถสร้างตารางของคลาสได้ดังนี้

```
Create Table User
( User_id        Interger(4)  Not Null.
 Name            Varchar(50)  Not Null,
 Surname         Varchar(50) Not Null,
E_mail           Varchar(50) Not Null,
Phone            Varchar(20),
 Primary Key User_id
)
```

- หลักการแปลงคลาสที่มีความสัมพันธ์แบบแอกรีเกรชันให้เป็นตารางที่สัมพันธ์กัน
  1. ออกแบบตารางจากคลาสทั้งสองของเครื่องหมายแอกรีเกรชัน
  2. การแสดงความสัมพันธ์ของตารางนั้น ให้นำเอาคีย์หลัก (อาจเป็นฟิลด์เดียวหรือกลุ่มของฟิลด์เดียวก็ได้) ของคลาสหลักมาเป็นคีย์รองของคอมโพซิสคลาส (Composite class) ซึ่งก็เหมือนกับการเอาชื่อพ่อแม่ไปเก็บไว้ที่ลูก
  3. ในการใส่คีย์รองเข้าไปยังคอมโพซิสคลาสนั้น ต้องพิจารณาด้วยว่าคีย์รองนั้น เป็นค่า Null หรือไม่



รูปที่ 2 แสดงคลาสตัวอย่าง 2

จากรูป 2 สามารถสร้างตารางได้ ดังนี้

```
Create Table Address_item
(Adb_id          Int(4) Not Null,
 Add_id          Int(4) Not Null,
 Name            Varchar(50) Not Null,
 Surname         Varchar(50) Not Null,
 Email           Varchar(50) Not Null,
```

```
Primary Key    Add_id,
Foreign Key    Adb_id References Address_book(Adb_id)
)
```

- **หลักการแปลงคลาสที่มีความสัมพันธ์แบบแอสโซซิเอชันให้เป็นตารางที่สัมพันธ์กัน**

1. แอสโซซิเอชันแบบ 1:1

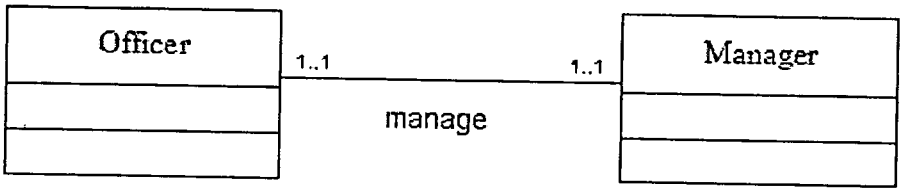
- ออกแบบตารางของคลาสทั้งสองข้างของเครื่องหมายแอสโซซิเอชัน
- ให้เลือกเอาคีย์หลักของตารางตัวใดก็ได้เป็นคีย์รองของอีกตารางหนึ่ง

รูปที่ 3 แสดงความสัมพันธ์แบบแอสโซซิเอชัน

จากรูป 3 สามารถสร้างตาราง Man และ Woman ที่สัมพันธ์กันดังนี้

```
Create Table Man
( ManId          Char(10)        Not null,
 WomanId         Char(10),
 Primary Key     ManId,
 Foreign Key     Woman_Id References Woman(Woman_Id))

Create Table Woman
( WomanId        Char(10)        Not null
 Primary Key     WomanId)
```



รูปที่ 4 แสดงความสัมพันธ์แบบแอสโซซิเอชัน แบบ 1:1

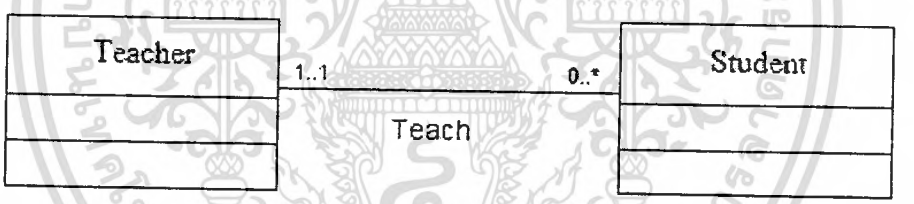
จากรูป 4 สามารถสร้างตาราง Manager และ Office ที่สัมพันธ์กันดังนี้

```
Create Table Manager
( ManGid      Char(10)      Not null,
  OfficeId    Char(10)      Not null
  Primary Key ManGid
  Foreign Key OfficeId References Office(OfficeId))

Create Table Office
( OfficeId    Char(10)      Not null
  Primary key OfficeId)
```

2. แอสโซซิเอชันแบบ 1:N

หลักในการสร้างตารางจากแอสโซซิเอชัน 1:N นั้นมีหลักการเดียวกับการสร้างตารางจากแอกกรีเกรชันโดยตารางในด้าน 1 จะเหมือนกับตารางของคลาสหลักของแอกกรีเกรชัน นั่นคือให้เอาคีย์หลักของตารางในด้าน 1 ไปเป็นคีย์รองของตารางในด้าน N ดังรูปที่ 5



รูปที่ 5 แสดงความสัมพันธ์แบบแอสโซซิเอชันแบบ 1:N

จากรูป 5 สามารถสร้างตาราง Teacher และ Student ได้ดังนี้

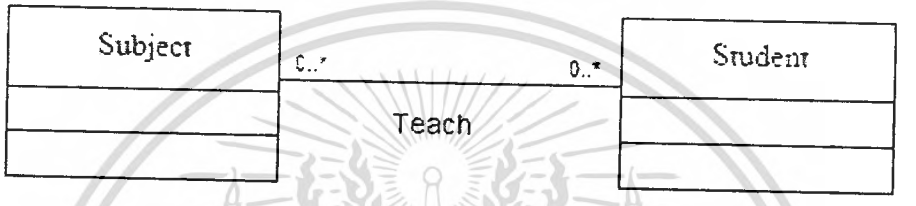
```
Create Table Student
( StuId       Char(10)      Not null,
  TchId       Char(10)      Not null
  Primary Key StdId
  Foreign Key TchId References Teacher(TchId))

Create Table Teacher
( TchId       Char(10)      Not null
  Primary Key TchId)
```

3. แอสโซซิเอชันแบบ N:N

หลักในการสร้างตารางจากคลาสที่มีความสัมพันธ์แบบ N:N มีหลักการดังนี้

1. สร้างตารางของคลาสทั้งสองข้างของแอสโซซิเอชัน
2. สร้างตารางอีกหนึ่งตาราง ที่มีอย่างน้อย 2 คอลัมน์ ซึ่งก็คือคีย์หลักของตารา ทั้งสองและให้คอลัมน์ทั้งหมดเป็นคีย์ของตาราง ดังกล่าว ซึ่งจะเรียกว่าเป็นตารางแอสโซซิเอชัน
3. ให้ส่วนหนึ่งของคีย์หลักที่เป็นคีย์หลักของตารางข้างใดข้างหนึ่งเป็นคีย์รองอ้างอิงไปยังตารางนั้น ๆ ดังรูปที่ 6



รูปที่ 6 แสดงความสัมพันธ์แบบแอสโซซิเอชันแบบ N:N

จากรูป 6 สามารถสร้างตาราง Student และ Subject ได้ดังนี้

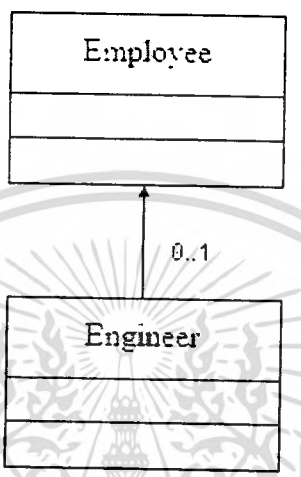
```
Create Table Student
( StuId        Char(10)        Not null,
 Primary key   StuId)

Create Table subject
( SubId        Char(10)        Not null,
 Primary Key   StuId)

Create Table Std_Sub
( StdId        Char(10),
 SubId         Char(10),
 Primary Key   (StdId, SubId),
 Foreign Key   StdId References Student
 Foreign Key   SubId References Subject)
```

- **หลักการแปลงคลาสที่มีความสัมพันธ์กันแบบเจเนอรัลไรเซชันให้เป็นตารางที่สัมพันธ์กัน**

1. ในกรณีที่เจเนอรัลไรเซชันที่เกิดขึ้นเป็นแบบโททัล-โอเวอร์แลปปิ้ง, พาร์เชียลโอเวอร์แลป ปิ้ง (Total-Overlapping, Partial-overlapping) ให้สร้างตารางของซุปเปอร์คลาสและของทุก ๆ คลาสย่อย (ใช้หลักการแอสโซซิเอชันแบบ 1:1) โดยให้สร้างคีย์รองไว้ที่ตารางของคลาสย่อยและคีย์รองนั้น ต้องกำหนดให้เป็น Not null และเป็นคีย์หลักในตารางย่อยด้วย ดังรูปที่ 7



รูปที่ 7 แสดงความสัมพันธ์แบบเจเนอรัลไรเซชันชนิดพาร์เชียลโอเวอร์แลปปิ้ง

จากรูป 7 สามารถสร้างตาราง employee และ engineer ที่สัมพันธ์กันดังนี้

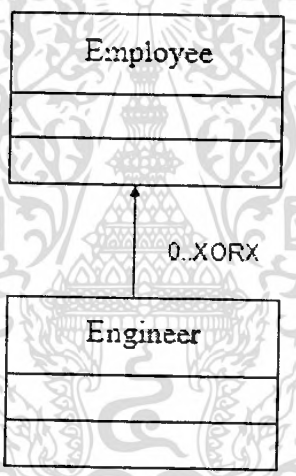
```
Create Table employee
( EmpId          Char(10)      Not null,
  Name           Char(10)      Not null,
  Surname        Char(10)      Not null,
  Primary  Key   ManId)
```

```
Create Table engineer
( EngId          Char(10)      Not null,
  MemId          Char(10)      Not null,
  EmpId          Char(10)      Not null,
  Engtype        Char(10)      Not null,
  Foreign  Key   EmpId References employee(EmpId))
  Primary  Key   (EngId,MemId))
```

2. ในกรณีที่เจเนอรัลไรเซชันที่เกิดขึ้น เป็นแบบโททัลเอ็กซ์คลูซีฟ (Total-Exclusive), พาร์เชียลเอ็กซ์คลูซีฟ (Partial Exclusive) ให้สร้างตารางของซุปเปอร์คลาสและของทุก ๆ คลาสย่อย (ใช้หลักการแอสโซซิเอชันแบบ 1:1) หลังจากนั้นให้สร้างตารางแอสโซซิเอชัน ซึ่งมีฟิลด์เดียว คือ คีย์หลักและดำเนินการ ดังนี้

- ในกรณีที่เป็นโททัลเอ็กซ์คลูซีฟ ให้เอาคีย์รองของตารางของคลาสย่อยซึ่งเป็นตัวเดียวกันกับคีย์หลักอ้างอิงมาที่คีย์หลักของตารางแอสโซซิเอชันที่สร้างขึ้น แต่ถ้าเป็นกรณีพาเชียลเอ็กซ์คลูซีฟให้เพิ่มคีย์รองซึ่งจะมีค่าเดียวกันกับคีย์หลักเสมอ หรือมีค่าเป็นนัลก็ได้ ให้อ้างอิงไปยังตารางแอสโซซิเอชัน

- ให้สร้างคีย์รองเพิ่มเข้าไปยังตารางของซุปเปอร์คลาสอ้างอิงมายังตารางแอสโซซิเอชันโดยมีเงื่อนไขว่า คีย์รองที่มีในตารางแอสโซซิเอชันนั้น จะสามารถมีค่าเป็นนัลได้ ในกรณีที่เป็นพาเชียลเอ็กซ์คลูซีฟและถูกตั้งเป็น Not Null ในกรณีเป็นโททัลเอ็กซ์คลูซีฟ ดังรูปที่ 8



รูปที่ 8 แสดงความสัมพันธ์แบบเจเนอรัลไรเซชันชนิดพาเชียลเอ็กซ์คลูซีฟ

จากรูปที่ 8 สามารถสร้างตาราง employee และ engineer ที่สัมพันธ์กันดังนี้

```
Create Table employee
( EmpId        Char(10)     Not null,
  AltId        Char(10)     Not null,
  Name         Char(10)     Not null,
  Surname      Char(10)     Not null,
  Primary Key  ManId)

Create Table Sub_employee
( SubId        Char(10)     Not null,
```

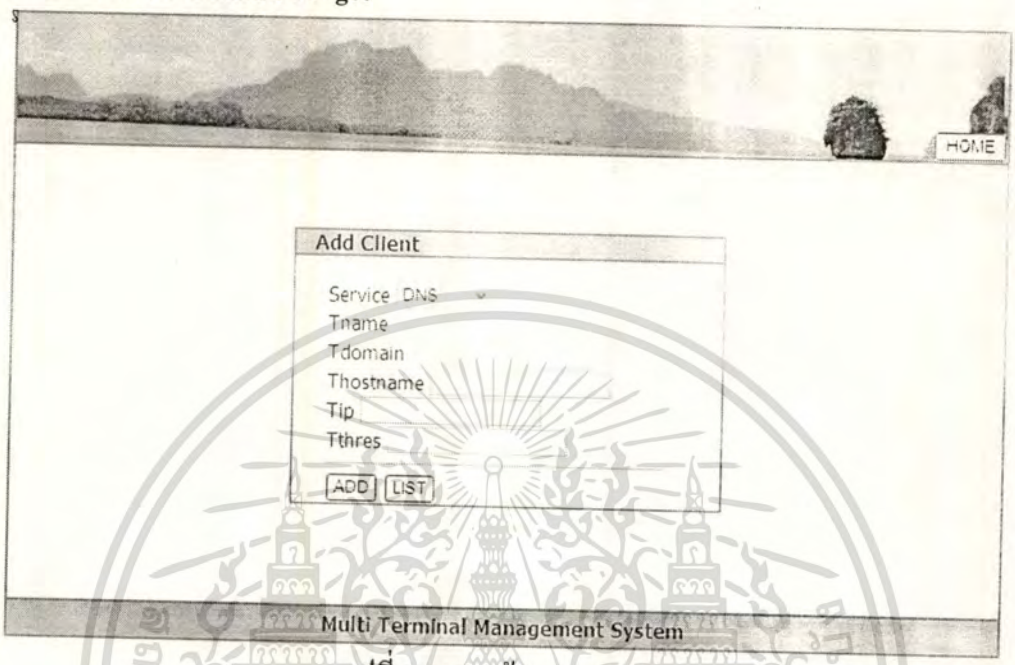
EngId	Char(10)	Not null,
AltId	Char(10)	Null,
Foreign Key	EngId References engineer(EngId))	
Foreign Key	AltId References employee(AltId))	
Primary Key	SubId)	

Create Table engineer

(EngId	Char(10)	Not null,
EmpId	Char(10)	Not null,
Engtype	Char(10)	Not null,
Foreign Key	EmpId References employee(EmpId))	
Primary Key	EngId))	

การสร้าง Node Client จะต้องมีการกำหนด group ของ client ด้วยโดยจะต้องมีการตั้งชื่อให้ตรงกับ Client ที่ตรงกับหน้างาน เพราะจะต้องนำมาใช้ในการ update ip ว่า Client Node นั้นมีการ update ข้อมูลมาครั้งสุดท้ายเมื่อใด

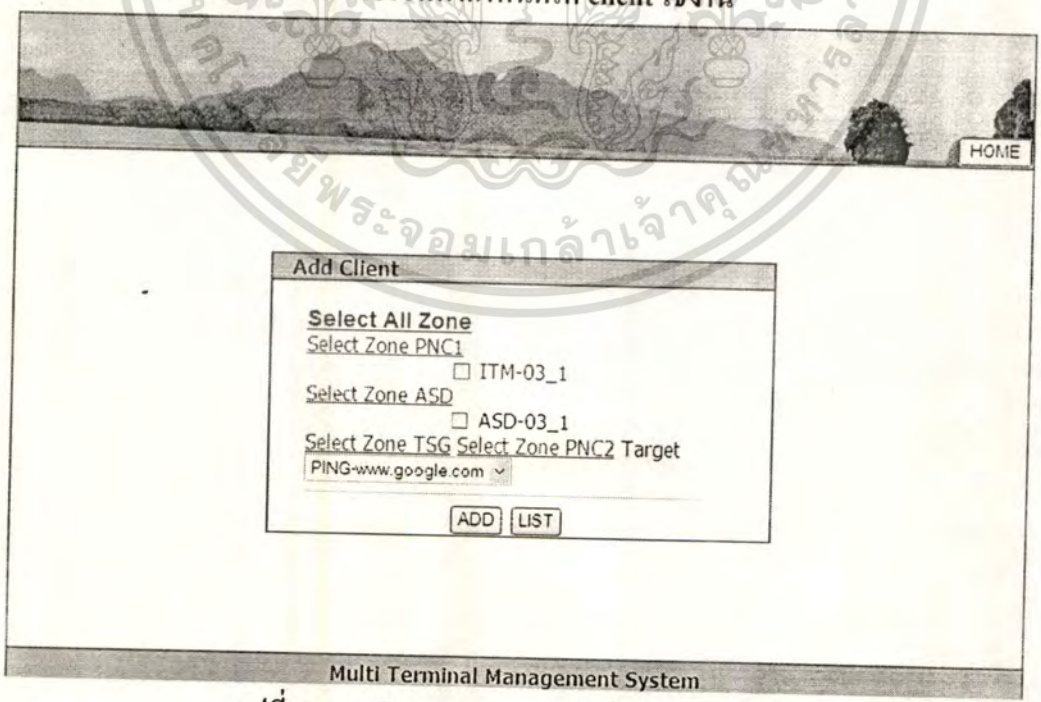
4. สร้าง command Target



รูปที่ 14 การสร้าง command

เป็นการกำหนดเงื่อนไขในการทดสอบเพื่อที่จะนำไปกำหนดให้ clients แต่ละตัวใช้ต่อไป

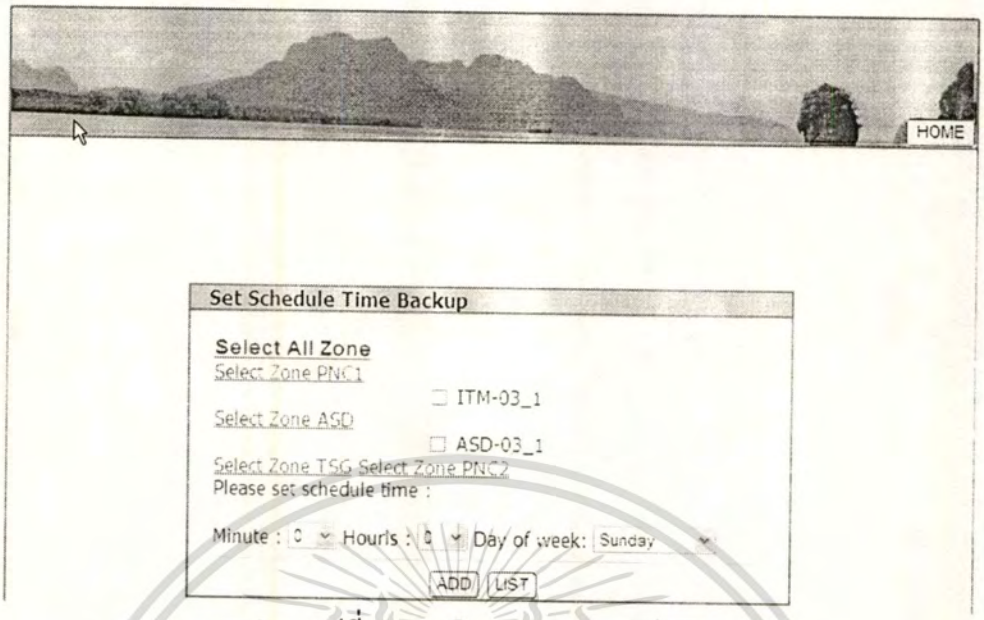
5. การนำ command ที่สร้างขึ้นมากำหนดให้ client ใช้งาน



รูปที่ 15 การกำหนด command ให้ client แต่ละตัว

เป็นการกำหนดให้ clients แต่ละตัวทำงานตามที่กำหนด โดยสามารถกำหนดเป็น group หรือ client ตัวเดียวก็ได้

6. การกำหนดการ Backup



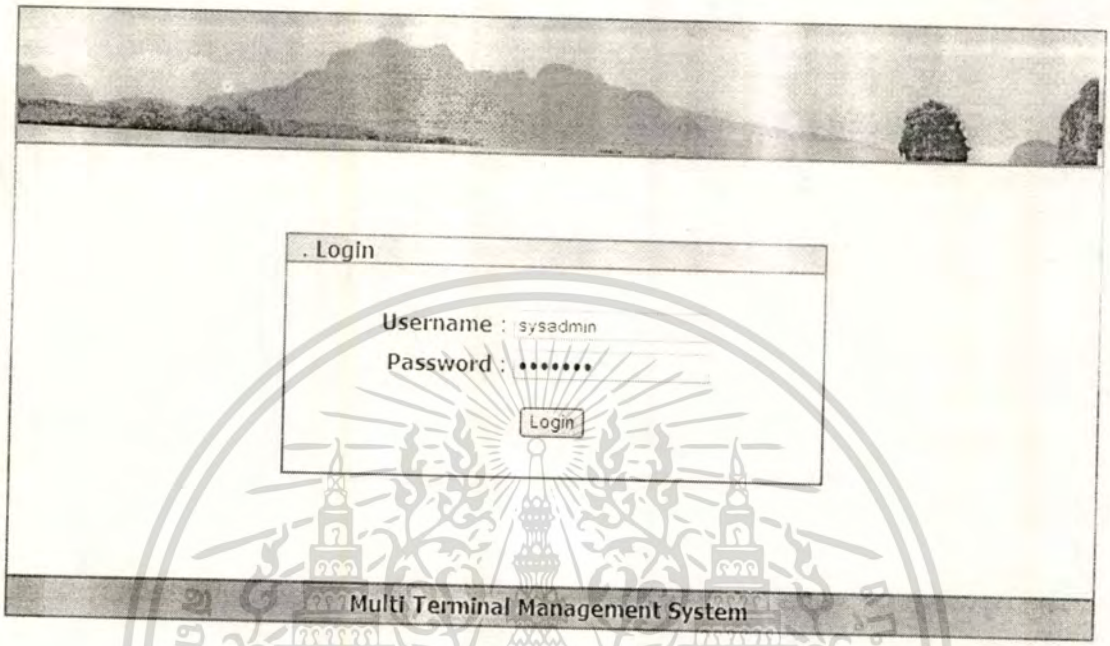
รูปที่ 16 การกำหนดการ backup

จะเป็นการตั้งเวลาการทำงาน backup โดยสามารถกำหนดเป็น Group หรือ กำหนด client เป็นแต่ละตัวได้

– ภาคผนวก ค

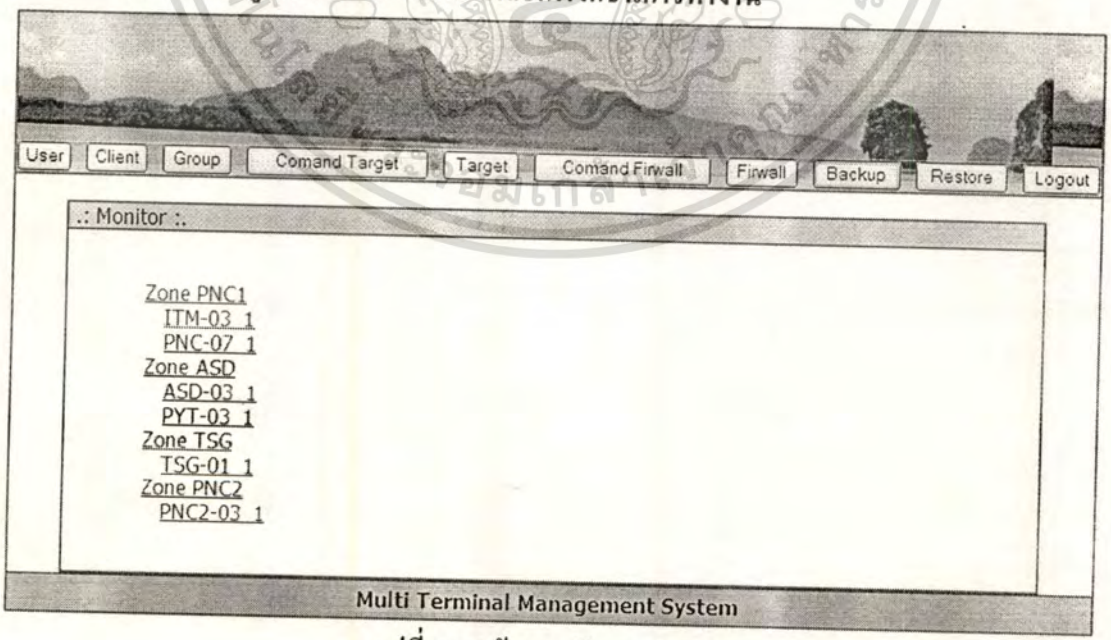
คู่มือการใช้งาน

ก่อนเข้าใช้งานจะต้อง มีสิทธิเข้าใช้งานระบบ



รูปที่ 9 แสดงรูปแบบการ Login เข้าระบบ

4.2.1.1 เข้าสู่หน้าหน้าจอหลักเพื่อเลือกฟังก์ชันการทำงาน



รูปที่ 10 หน้าจอหลักของระบบ

ภาคผนวก ข

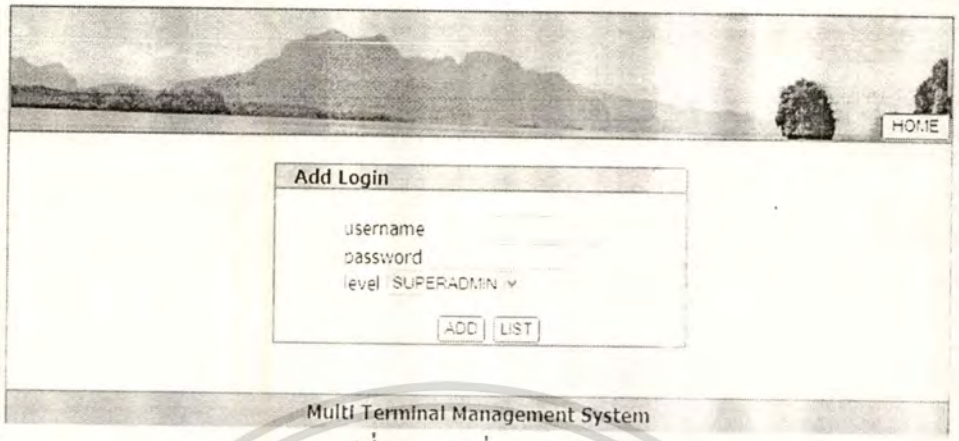
การติดตั้งแบ่งการติดตั้งออกเป็น 2 ส่วน

1. **การติดตั้งการทำงานที่ server**
   - ติดตั้ง OS Linux platform โดยระบบผู้จัดทำติดตั้ง Linux Ubuntu
   - ติดตั้งระบบจัดการฐานข้อมูล MySQL version ที่สูงกว่า 4.0
   - ติดตั้งซอร์ฟแวร์ภาษา PHP ตั้งแต่ version 4.3.2 ขึ้นไป
   - ติดตั้ง Apache webserver
   - ทำการเปิด service ssh(secure channel)
   - ทำการสร้าง public key ที่ส่วน server และนำ public key ของ client และ server เก็บใน file authorize_key เพื่อเก็บ public key
   - นำ program JP graph มาใช้ในการแสดงผลของกราฟ
   - นำ Software PHP(ใน CD file server)ไว้ใน directory ที่ต้องการ run
   - ทำการสร้าง Database บน Server โดยสามารถเรียกใช้ได้จาก phpmyadmin ได้ทันที

2. **การติดตั้งการทำงานที่ Client**
   - ติดตั้ง OS Linux platform โดยระบบผู้จัดทำติดตั้ง Linux debian
   - ติดตั้งระบบจัดการฐานข้อมูล MySQL version ที่สูงกว่า 4.0
   - ติดตั้ง Apache Webserver
   - ติดตั้งซอร์ฟแวร์ภาษา PHP ตั้งแต่ version 4.3.2 ขึ้นไป
   - ทำการเปิด service ssh(secure channel)
   - ทำการสร้าง public key ที่ส่วน client และนำ public key ของ client และ server เก็บใน file authorize_key เพื่อเก็บ public key
   - นำ Software PHP(ใน CD file client) มาเก็บไว้ใน directory ที่ต้องการ run
   - ทำการสร้าง database โดย นำ คำสั่งใน file pms.sql มาสร้าง file บน mysql

ฟังก์ชันการทำงาน จะมีฟังก์ชันดังนี้

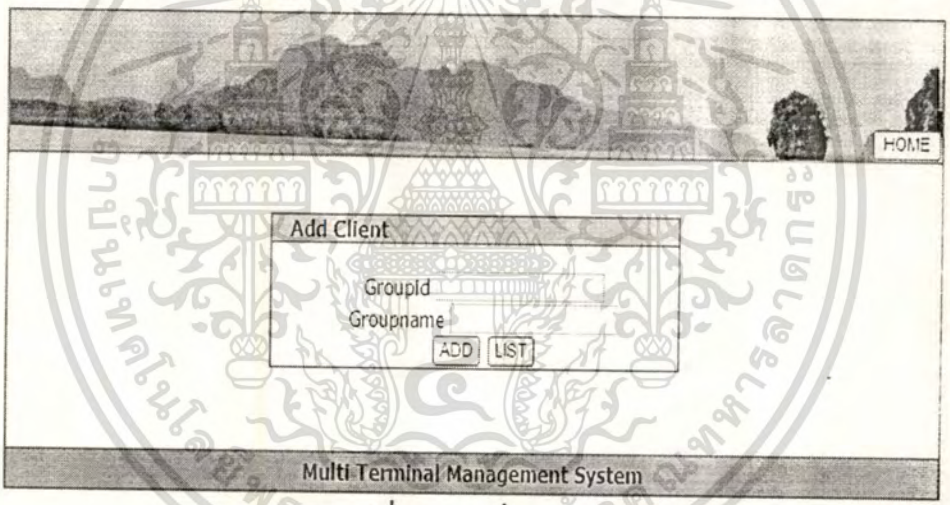
1. การจัดการกับ user



รูปที่ 11 การเพิ่ม user login

เป็นการกำหนดสิทธ์การเข้าใช้งานระบบ

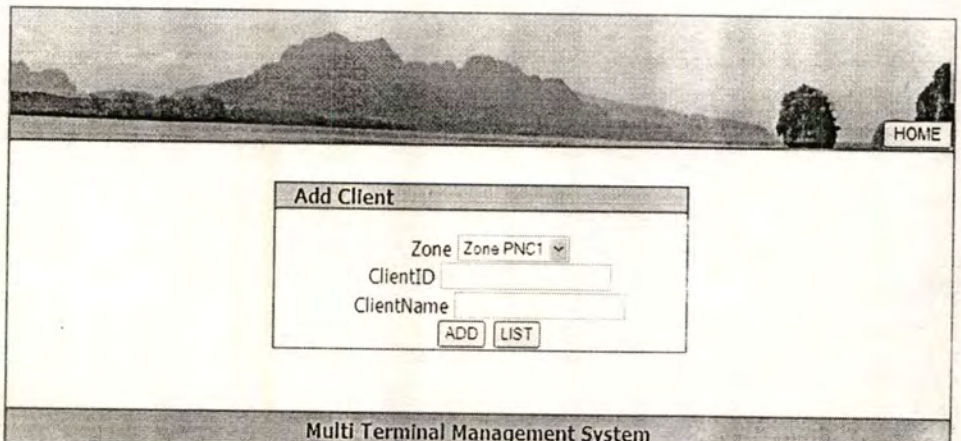
2. การสร้าง Group Node



รูปที่ 12 การเพิ่ม group

เป็นการสร้าง Group เพื่อกำหนดกลุ่มของ client หรือการแบ่ง Zone

3. การสร้าง Client Node



รูปที่ 13 การเพิ่ม group

ประวัติผู้เขียน

ชื่อผู้เขียน	นายวุฒิภูมิ อกนิษฐ์
วันเดือนปีเกิด	20 มกราคม 2524
สถานที่เกิด	จ.พิจิตร
วุฒิการศึกษาระดับปริญญษตรี	วศ.บ (วิศวกรรมคอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2544