

# การพัฒนาซอฟต์แวร์ที่ใช้ในการกระจายและการปรับแต่งค่าของเราเตอร์

## Distribution of Router Configuration Software Development

โดย

นายมหัทธวัฒน์ รักษาเกียรติศักดิ์

รหัส 41067041



\*H001627\*

อาจารย์ที่ปรึกษา

อาจารย์อัครินทร์ คุณกิตติ

วัน เดือน ปี..... 22 S.A. 2549

เลขทะเบียน..... 01627

เลขเรียกหนังสือ..... 016.1/2549

"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจธ."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาคเรียนที่ 2 ปีการศึกษา 2542  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาซอฟต์แวร์ที่ใช้ในการกระจายและการปรับแต่งค่าของเราเตอร์
นักศึกษา	นายมหัทธวัฒน์ รักษาเกียรติศักดิ์
อาจารย์ที่ปรึกษา	อาจารย์อักรินทร์ คุณกิตติ
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2542

## บทคัดย่อ

เราเตอร์เป็นอุปกรณ์ตัวหนึ่งที่อยู่บนเครือข่าย ซึ่งใช้ทำหน้าที่ในการหาเส้นทางการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์และระบบเครือข่าย เมื่อเราเตอร์มีมากขึ้นทำให้ระบบจะต้องมีการแก้ไขข้อมูลเป็นจำนวนมาก จึงเกิดความยุ่งยากในการปรับแต่งค่าข้อมูลของเราเตอร์ ซึ่งโครงการนี้เป็นการพัฒนาระบบการกระจายข้อมูลการปรับแต่งค่าต่างๆบนซิสโทเราเตอร์ เพื่อช่วยในการจัดการกับข้อมูลการปรับแต่งค่าของซิสโทเราเตอร์ ในการพัฒนาจะทำบนเครื่องยูนิกซ์ โดยในส่วนของเครื่องยูนิกซ์จะประกอบด้วย 3 ส่วน คือ การกระจายข้อมูลการปรับแต่งไปยังเราเตอร์ตัวเดียว, การกระจายข้อมูลการปรับแต่งไปยังกลุ่มของเราเตอร์ และการจัดการกับข้อมูลกลุ่มของเราเตอร์ ซึ่งใน 2 ส่วนแรกนั้นใช้ในส่วนของเชลล์สคริปต์ ไปเรียกโมดูลที่เป็นโปรแกรมเอ็กเพ็ค (expect) ในส่วนของเชลล์สคริปต์นั้นจะทำการรับค่าแล้วส่งไปยังโปรแกรมเอ็กเพ็ค ส่วนโปรแกรมเอ็กเพ็คนั้นจะเรียกคำสั่ง telnet แล้วส่งพารามิเตอร์ในการกำหนดคำสั่งไปยังเราเตอร์ ซึ่งกลไกในการติดต่อกันระหว่างเครื่องยูนิกซ์กับเราเตอร์นั้นจะผ่านกลไก RPC และในส่วนการจัดการกับข้อมูลกลุ่มของเราเตอร์นั้น ข้อมูลที่เกี่ยวกับกลุ่มของเราเตอร์ และเราเตอร์แต่ละตัวนั้นจะถูกเข้ารหัสโดยวิธี DES (Data Encryption Standard) เพื่อให้ข้อมูลของเราเตอร์มีความปลอดภัยมากที่สุด

<b>Title</b>	Distribution of Router Configuration Software Development
<b>Student</b>	Mr.Mahutthawat Raksakiettisak
<b>Advisor</b>	Mr.Akharin Khunkitti
<b>Level of Study</b>	Master of Science in Information Technology
<b>Major</b>	Information Science
<b>Academic Year</b>	1999

## ABSTRACT

In rapid increasing number of routers in enterprise network bring configuration trouble to network administrator. This project aims for managing the distribution of Cisco router configuration. It is implemented using bash shell script and Expect programming language on Linux Redhat 6.0. Unix machine is set as RCP server in order to retrieve and copy configuration file from router. The Shell script gets data from user input and sends it to Expect program. The Expect program invokes telnet command and sends parameters from shell script to Cisco routers. This software has 3 sets of operations, single router, a group of router and managing group of router. For single router operation, program manages running configuration in a router. In a group of router operation, program add local configuration file of a router and global configuration file, which common to its group, then send it to all member in that group using single router operation. Information of group is encrypted using DES (Data Encryption Standard) for security purpose.

## กิตติกรรมประกาศ

ขอขอบคุณอาจารย์อักรินทร์ คุณกิตติ ที่กรุณาให้คำปรึกษาแนะนำในด้านการศึกษา การ  
ออกแบบระบบงาน แนวทางแก้ปัญหาต่างๆ และดูแลช่วยเหลือมาโดยตลอด

ขอขอบคุณอาจารย์ประจำคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้า  
เจ้าคุณทหารลาดกระบังที่ได้ประสาทวิชาความรู้ทั้งภาคทฤษฎีและภาคปฏิบัติ

ขอขอบคุณเจ้าหน้าที่ฝ่ายระบบเครือข่ายคอมพิวเตอร์และการสื่อสารข้อมูล ที่ให้คำแนะนำ  
เกี่ยวกับระบบเครือข่ายของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ขอขอบคุณเจ้าหน้าที่ฝ่ายระบบคอมพิวเตอร์และเครือข่ายมหาวิทยาลัยศรีนครินทรวิโรฒ  
ที่ให้คำแนะนำและข้อมูลต่างๆเกี่ยวกับระบบเครือข่าย

และขอขอบคุณเพื่อนนักศึกษาทุกท่านที่ได้ให้ความช่วยเหลือด้วยดีตลอดมา

นายมหัทธวัฒน์ รักษาเกียรติศักดิ์

ผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	VI
บทที่	
1. บทนำ	
1.1 ความสำคัญและที่มาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 ขั้นตอนในการดำเนินงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 แผนการดำเนินงาน.....	4
2. ระบบการกระจาย Configuration ของ Cisco Router	
2.1 ระบบการทำงานของ Cisco Router.....	5
2.2 วิธีการต่างๆในการปรับแต่งค่าของ Cisco Router.....	7
2.3 หลักการในการติดต่อระหว่าง Router กับระบบ.....	10
2.4 วิธีการในการเข้ารหัสข้อมูลที่เกี่ยวข้องกับระบบ.....	11
3. การออกแบบระบบงานการกระจายค่าการปรับแต่ง ของ Cisco Router	
3.1 การออกแบบระบบงาน.....	13
3.2 ลักษณะการเก็บข้อมูลต่างๆในระบบงาน.....	16

## สารบัญ (ต่อ)

4. การพัฒนาระบบงาน	
4.1 หลักการทำงานของโปรแกรม.....	21
4.2 โครงสร้างของโปรแกรม.....	22
4.3 การพัฒนาโปรแกรม.....	27
4.4 วิธีการทดสอบโปรแกรม.....	32
4.5 ผลการทดสอบโปรแกรม.....	34
5. สรุปและข้อเสนอแนะ	
5.1 การเปรียบเทียบวิธีการในการ Configure แบบเดิมกับระบบใหม่.....	43
5.2 ปัญหาที่เกิดขึ้น.....	43
5.3 ข้อเสนอแนะ.....	44
บรรณานุกรม.....	45
ภาคผนวก	
ภาคผนวก ก คู่มือการใช้งานระบบ.....	46
ภาคผนวก ข วิธีการติดตั้งโปรแกรมและตัวอย่าง source code.....	52

# สารบัญภาพ

หน้า

ภาพที่

2.1 ลำดับการอ่านค่าต่างๆ บนเราเตอร์.....	6
2.2 การเชื่อมต่อระหว่างระบบกับเราเตอร์.....	10
3.1 Context Diagram ของระบบ.....	13
3.2 DFD level 1 ของระบบ.....	14
3.3 DFD level 2 of process 2.....	14
3.4 DFD level 2 of process 3.....	15
3.5 การตรวจสอบรหัสลับของกลุ่ม .....	18
3.6 ความสัมพันธ์ระหว่างข้อมูลการปรับแต่งค่าต่างๆของ cisco router.....	19
3.7 ความสัมพันธ์ระหว่างเพิ่มข้อมูลกลุ่มกับรายละเอียดของกลุ่ม.....	20
4.1 การแก้ไขข้อมูลของเราเตอร์ .....	22
4.2 การเปลี่ยนแปลงข้อมูลของเราเตอร์ .....	23
4.3 การเพิ่มค่าของข้อมูลการปรับแต่งให้กับกลุ่มของเราเตอร์.....	24
4.4 การลบค่าของข้อมูลต่างๆให้กับกลุ่มของเราเตอร์.....	24
4.5 การเพิ่มกลุ่มของเราเตอร์.....	25
4.6 การลบกลุ่มของเราเตอร์.....	26
4.7 การแก้ไขข้อมูลภายในกลุ่ม.....	26
4.8 หน้าจอโปรแกรมหลัก .....	28
4.9 หน้าจอการจัดการกับเราเตอร์ตัวเดียว.....	28
4.10 หน้าจอการจัดการกับกลุ่มของเราเตอร์.....	29
4.11 หน้าจอการกระจายการเพิ่มข้อมูลไปยังกลุ่มของเราเตอร์.....	29
4.12 หน้าจอการจัดการกับกลุ่มของเราเตอร์.....	30
4.13 หน้าจอการเพิ่มกลุ่มของเราเตอร์.....	30
4.14 หน้าจอการกรอกข้อมูลของเราเตอร์ .....	31

## สารบัญภาพ (ต่อ)

หน้า

ภาพที่

4.15 การแสดงการกำหนดค่าของเราเตอร์บนเครื่องเราเตอร์.....	35
4.16 การแสดงการกำหนดค่าของเราเตอร์บนเครื่องยูนิคซ์.....	36
4.17 ค่าการปรับแต่งข้อมูลของเราเตอร์หลังจากที่เพิ่มข้อมูลส่วนกลาง.....	38
4.18 ค่าการปรับแต่งข้อมูลของเราเตอร์หลังจากที่ลบข้อมูลส่วนกลาง.....	39
4.19 เพิ่มข้อมูลที่ถูกเข้ารหัส .....	40
4.20 ข้อมูลของเราเตอร์ที่ถอดรหัสเรียบร้อยแล้ว.....	40
4.21 เพิ่มข้อมูลที่ถูกเข้ารหัสหลังจากที่มีการเพิ่มเราเตอร์ car14.....	41
4.22 ข้อมูลของเราเตอร์ที่ถอดรหัสเรียบร้อยแล้วหลังจากที่มีการเพิ่ม car14.....	41

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของโครงการ

ปัจจุบันเริ่มมีการขยายเครือข่ายเพิ่มมากขึ้น อุปกรณ์ต่างๆที่ใช้ในการเชื่อมต่อกันระหว่างเครือข่ายก็มีการเพิ่มมากขึ้น ซึ่งหนึ่งในอุปกรณ์เครือข่ายนั้นก็คือ router ซึ่งโดยหน้าที่หลักของ router จะทำหน้าที่ในการหาเส้นทางต่างๆ เพื่อให้ข้อมูลต่างๆมีการเชื่อมต่อถึงกันได้ ซึ่งในปัจจุบัน อุปกรณ์ router ก็มีอยู่หลายบริษัทที่ทำการผลิตออกมา ไม่ว่าจะเป็น Cisco, 3com, Cabletron เป็นต้น แต่ที่จะกล่าวถึงในโครงการนี้จะกล่าวถึงเฉพาะในส่วนของ Cisco router ซึ่งเป็น router ที่นิยมใช้กันมากที่สุดในขณะนี้ ซึ่งในระบบเครือข่ายขนาดใหญ่นั้นก็จะมี router เป็นจำนวนมาก ซึ่งการที่มี Router ในเครือข่ายขนาดใหญ่เป็นจำนวนมาก โดยที่ Router บางตัวอาจจะมี Configuration บางส่วนที่คล้ายคลึงกัน ดังนั้นถ้าผู้ดูแลระบบต้องการที่จะ Configure ค่าต่างๆที่เหมือนกันไปยัง Router แต่ละตัว ก็จะต้องทำวิธีที่เป็น manual ซึ่งทำการเพิ่มหรือแก้ไขคำสั่งแต่ละคำสั่งใน Router ทุกๆตัวที่ต้องการจะเปลี่ยนแปลงข้อมูลที่เหมือนกัน ซึ่งจะเกิดความซ้ำซ้อนในระบบงานขึ้น จึงเกิดแนวความคิดที่ว่า ในการ configure router ถ้า Router ตัวใดที่มี Configuration ที่คล้ายกันจะรวมอยู่ในกลุ่มเดียวกัน แล้วทำการ configure กับระบบงานระบบหนึ่งแล้วระบบนั้นจะเป็นตัวที่จะกระจาย Configuration ของ Router ที่เหมือนกัน ออกไปยัง Router ที่อยู่ในกลุ่มนั้นๆ ทำให้การทำงานลดลงและการจัดการกับข้อมูลต่างๆก็จะมีประสิทธิภาพมากขึ้น

### 1.2 วัตถุประสงค์ของโครงการ

จากที่ได้กล่าวมาแล้วข้างต้น จะเห็นได้ว่าการที่มีระบบการกระจาย Configuration นั้นจะช่วยให้งานต่าง ๆ ลดลงไม่ว่าจะเป็นในส่วนที่เป็น การ Configure ในแต่ละตัว หรือจะเป็นการ Configure router หลายๆตัวที่อยู่ในกลุ่มเดียวกัน ทำให้ลดความซ้ำซ้อนของงานลงได้ นอกจากนี้ยังช่วยในการตรวจสอบความถูกต้องของ Configuration ที่จะนำไปกำหนดค่าต่างๆของ router ด้วย ทำให้การทำงานในด้านต่างๆดีขึ้น

### 1.3 ขอบเขตของโครงการ

ในส่วนของระบบโครงการการกระจาย Configuration นั้นจะเป็นลักษณะการทำงานคล้ายกับการทำงานที่เป็น Host terminal ซึ่งจะการทำงานจะแบ่งเป็น 2 ส่วน คือ ในส่วนที่เป็นการ Configure router และส่วนที่สองจะเป็นส่วนที่จะต้องทำการเขียนโปรแกรมที่เครื่อง Host ที่เป็นระบบปฏิบัติการ Unix ซึ่งการทำงานนั้น จะต้องเริ่มต้นโดยการทำการกำหนดค่าต่างๆของ router ให้สามารถสนับสนุนการทำงานในลักษณะของ RCP server ซึ่งการที่ router สนับสนุนการทำงานนี้ก็เพื่อให้ข้อมูลที่เขียน Configuration สามารถสำเนาจาก router ไปยังเครื่อง Host ที่ทำการกำหนดไว้ได้ หรือ ทำการสำเนาข้อมูลที่เป็น Configuration จากเครื่องที่เป็น Host ไปยัง router ได้ ต่อไปก็จะเป็นส่วนของระบบการกระจายซึ่งจะทำการกระจายข้อมูล Configuration ไปยัง router ต่างๆ ซึ่งในส่วนนี้ระบบจะทำงานโดยให้ผู้ดูแลระบบทำการเลือกว่าต้องการจะจัดการกับข้อมูลของ router เพียงหนึ่งตัวหรือ หลายตัว

ในกรณีที่ทำการ Configure router เพียงตัวเดียวก็จะให้มีการใส่ รหัสผู้ใช้ และ รหัสลับทั้งหมดพร้อมกับ เครื่อง router ที่ต้องการจะ Configure จากนั้นระบบก็จะทำการดึงค่าข้อมูล Configuration มาที่ระบบ แล้วทำการเปรียบเทียบกับข้อมูล Configuration ที่เก็บไว้อยู่ในเครื่อง Host ซึ่งการทำการเปรียบเทียบนั้นจะแสดงขึ้นมาว่าข้อมูลของ Configuration ที่ได้จาก router นั้นแตกต่างกับที่เก็บไว้บนเครื่อง Host หรือไม่ และผู้ดูแลระบบต้องการจะเปลี่ยนแปลงค่าข้อมูลที่อยู่บน Host ก็สามารถเปลี่ยนได้

ในกรณีที่ผู้ดูแลระบบต้องการจะ Configure router หลายๆตัวจะต้องมีการระบุว่าจะกระจาย Configuration ไปในกลุ่มใด ซึ่งในการกำหนดกลุ่มผู้ดูแลระบบสามารถกำหนดกลุ่มขึ้นมาใหม่ หรือ จะทำการลบกลุ่มที่มีอยู่ก็ได้ หรือต้องการแก้ไขข้อมูลของกลุ่มที่มีอยู่ได้ ซึ่งข้อมูลของกลุ่ม router ในแต่ละกลุ่ม จะมีการเข้ารหัสในแบบ DES (Data Encryption Standard) ถ้าผู้ที่ใช้ระบบจะต้องผ่านการตรวจสอบรหัสลับของแต่ละกลุ่ม ผู้ที่มีรหัสนั้นจึงจะทำการลบ หรือ แก้ไขข้อมูลภายในกลุ่มได้

### 1.4 ขั้นตอนในการดำเนินงาน

#### 1.4.1 ศึกษาการทำงานของเราเตอร์

เป็นการศึกษาถึงโครงสร้างต่างๆของระบบการปรับแต่งค่าต่างๆของเราเตอร์ และวิธีการกำหนดค่าต่างๆของเราเตอร์ และวิธีการในการปรับแต่งค่าต่างๆของเราเตอร์

เพื่อใช้เป็นแนวทางในการออกแบบระบบการกระจายการปรับแต่งค่าของเราเตอร์ให้มีประสิทธิภาพสูงสุด

1.4.2 ออกแบบการจัดเก็บข้อมูลที่ใช้ในระบบการกระจายค่าการปรับแต่งของเราเตอร์เป็นการวิเคราะห์และออกแบบการจัดเก็บข้อมูลต่างๆที่ใช้ในการปรับแต่งค่าของเราเตอร์ รวมถึงความปลอดภัยของข้อมูลที่จัดเก็บในระบบการกระจายและการปรับแต่งค่าของเราเตอร์

1.4.3 กำหนดค่าต่างๆของเราเตอร์เพื่อใช้ในโปรแกรมเป็นการกำหนดค่าของเราเตอร์เพื่อให้สนับสนุนการทำงานของระบบการกระจายและการปรับแต่งค่าของเราเตอร์

1.4.4 พัฒนาโปรแกรมบนเครื่องโฮสต์ยูนิคซ์  
การพัฒนาโปรแกรมจะใช้ลักษณะของสคริปต์ที่มีอยู่บนยูนิคซ์ และภาษาเอ็กซ์เพกต์ (expect) ซึ่งเป็นส่วนหนึ่งของภาษา TCL เนื่องจากภาษานี้เหมาะกับงานที่ทำในลักษณะ offline ซึ่งหน้าที่การทำงานของโปรแกรมนี้นี้

1.4.4.1 ทำการกระจายค่าของการปรับแต่งจากโฮสต์ยูนิคซ์ไปยังเราเตอร์

1.4.4.2 ทำการกระจายค่าของการปรับแต่งจากโฮสต์ยูนิคซ์ไปยังกลุ่มของเราเตอร์

1.4.4.3 สามารถกำหนดการเพิ่ม ลบ หรือ แก้ไข ข้อมูลต่างของกลุ่มได้

1.4.5 ทดสอบและปรับปรุงโปรแกรม

ทดสอบการทำงานของโปรแกรม เพื่อดูความสามารถทั้งหมดที่เป็นไปได้ของโปรแกรม รวมถึงข้อจำกัดและขจัดปัญหาที่เกิดขึ้นกับระบบ

1.4.6 สรุปผลการพัฒนา

สรุปผลการพัฒนาระบบและจัดทำเอกสารประกอบ เพื่อสรุปผลการดำเนินงานที่ได้จัดทำไว้แล้ว ปัญหาและอุปสรรคที่เกิดขึ้น ข้อเสนอแนะ รวมทั้งแนวทางในการพัฒนาให้มีประสิทธิภาพมากขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ระบบนี้สามารถทำการกระจายค่าการปรับแต่งไปยังเราเตอร์ได้ถูกต้อง

1.5.2 ประหยัดเวลาในการกำหนดค่าต่างๆของเราเตอร์ที่เหมือนๆกัน

1.5.3 มีการเก็บข้อมูลต่างๆของเราเตอร์ไว้เป็นข้อมูลสำรองอยู่ที่เครื่องโฮสต์ยูนิคซ์

## 1.6 แผนการดำเนินงาน

ส่วนของงานที่จะต้องพัฒนา	พ.ย	ธ.ค	ม.ค	ก.พ
1.6.1 ศึกษาการทำงานของเราเตอร์	■			
1.6.2 ออกแบบการจัดเก็บข้อมูลที่ใช้ในระบบการกระจายค่าการปรับแต่งของเราเตอร์	■			
1.6.3 กำหนดค่าต่างๆของเราเตอร์เพื่อใช้ในโปรแกรม		■		
1.6.4 พัฒนาโปรแกรมบนเครื่องไฮสปีดยูนิกซ์			■	
1.6.5 ทดสอบและปรับปรุงโปรแกรม			■	
1.6.6 สรุปผลการพัฒนา				■



## บทที่ 2

### ระบบการกระจาย การปรับแต่งค่า ของ Cisco Router

เนื่องจากในสมัยแรกการนำคอมพิวเตอร์มาเชื่อมต่อกันยังมีไม่มากนัก จึงทำให้การเชื่อมต่อระหว่างคอมพิวเตอร์หรือเครือข่ายยังไม่มีปัญหา ในกรณีที่ไม่มีเครื่องอยู่ไม่มากนักก็อาจจะอาศัยการเชื่อมต่อเครื่องต่างๆผ่านฮับ (Hub) ได้ แต่ในกรณีที่ระบบขนาดใหญ่ขึ้นนั้นมีการเชื่อมต่อระหว่างเครือข่ายเป็นจำนวนมากย่อมมีการนำเราเตอร์เข้ามาใช้งาน เมื่อมีการใช้งานในระบบเครือข่ายมากขึ้นทำให้ภายในเครือข่ายคอมพิวเตอร์มีจำนวนเราเตอร์มากขึ้น เมื่อมีเราเตอร์มากขึ้น ทำให้เกิดปัญหาในการปรับแต่งค่าของเราเตอร์เกิดขึ้น ซึ่งในการปรับแต่งค่าต่างๆของซิสโกเราเตอร์ (cisco router) นั้นโดยทั่วไปแล้วจะทำการกำหนดค่าโดยการใช้คำสั่ง telnet เข้าไปที่เราเตอร์ตัวที่ต้องการจะกำหนดค่าแล้วจึงกำหนดค่าต่างๆของเราเตอร์ในแต่ละตัว ซึ่งการที่จะใช้คำสั่งต่างๆของเราเตอร์ก็ต้องเข้าใจถึงค่าต่างๆก่อนจึงจะสามารถกำหนดค่าให้ได้ตามความต้องการได้อย่างถูกต้อง

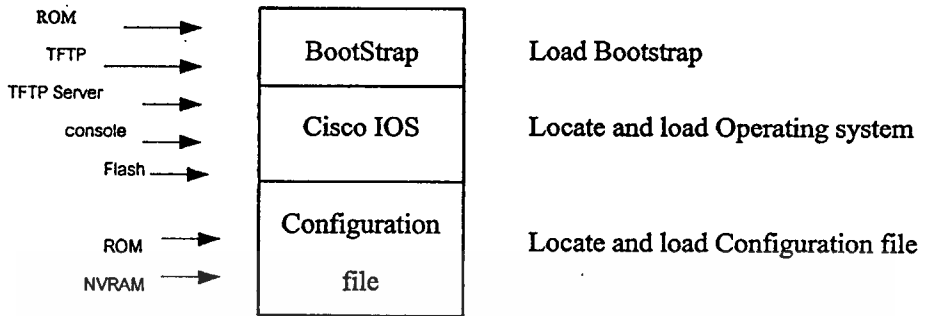
#### 2.1 ระบบการทำงานของ Cisco Router

ในกรณีที่เครื่องใหม่การทำงานของ Router จะไม่แตกต่างกับเครื่องที่มีการใช้งานอยู่แล้วแต่จะต้องมีการ ตรวจสอบซอฟต์แวร์เวอร์ชัน และฮาร์ดแวร์ ซอฟต์แวร์ ที่ได้ install ลงไปแล้ว ว่าตรงกับเวอร์ชันที่เราต้องการหรือไม่ การทำงานจะเป็นการให้ใส่ค่า parameter ต่างๆในการกำหนดค่าของซิสโกเราเตอร์ โดยที่ค่าต่างๆนั้นจะเก็บรวมกันเป็นแฟ้มข้อมูลเดียว

การทำงานของ Router จะทำงานโดยจะเริ่มต้นจากการอ่านค่าที่ ROM เพื่อทำการดึงค่า bootstrap (ค่าต่างๆที่ใช้ในการเปิดเครื่อง) ขึ้นมาเพื่อใช้ในการดึงระบบปฏิบัติการของ router ขึ้นมาซึ่งการดึงระบบปฏิบัติการขึ้นมาจะดูจาก boot field ของ configuration registerว่าจะไปดึงระบบปฏิบัติการมาจากที่ใด โดยระบบปฏิบัติการจะสามารถถูกดึงจาก flash หรือ เครื่องคอมพิวเตอร์เครือข่าย (network server) หรือจาก ROM ก็ได้ซึ่งโดยปกติจะดึงระบบปฏิบัติการมาจาก flash เมื่อระบบดึงระบบปฏิบัติการขึ้นมาแล้วนั้น ก็จะทำการอ่านค่า configuration file ซึ่ง configuration file ก็สามารเลือกได้ว่าจะสามารถนำเอา configuration file มาจาก NVRAM หรือเครื่องคอมพิวเตอร์เครือข่าย หรือ มาจาก Console ก็ได้ซึ่งโดยทั่วไปแล้วจะเก็บค่า configuration file ไว้ใน NVRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(nonvolatile memory) และทำการ configure ค่าต่างๆตาม configuration ที่อ่านมาได้ เป็นไปตามรูปที่ 2.1



### รูปที่ 2.1 ลำดับการอ่านค่าต่างๆบนเราเตอร์

ในการเปิดเครื่องครั้งต่อไป Router จะนำค่าต่างๆของการตั้งระบบปฏิบัติการ หรือ configuration file ตามที่ได้กำหนดเอาไว้

การกำหนดระบบปฏิบัติการ (system image) ของ router เมื่อเกิดการเปิดเครื่องครั้งใหม่เราสามารถกำหนดการอ่านค่าระบบปฏิบัติการได้จากทั้ง 3 ที่ เพื่อเป็นการสำรองข้อมูล ซึ่งมีอยู่ 3 วิธีด้วยกันคือ

- จาก flash memory ซึ่ง flash memory จะสามารถเก็บระบบปฏิบัติการลงใน flash ได้ซึ่งวิธีการนี้จะไม่ไปเปลี่ยนค่าใน EPROM
- จากเครื่องคอมพิวเตอร์เครือข่าย ในกรณีที่ flash memory เกิดการเสียหาย ก็สามารถระบุให้ตั้งระบบปฏิบัติการ จากเครื่องคอมพิวเตอร์เครือข่ายได้โดยใช้ TFTP หรือ scp ก็ได้
- จาก ROM ในกรณีที่ทั้ง flash memory และเครื่องคอมพิวเตอร์เครือข่ายเสียหายก็สามารถระบุให้ตั้งระบบปฏิบัติการ จาก ROM แต่ระบบปฏิบัติการที่เก็บใน ROM อาจจะไม่ล่าสุดเหมือนกับที่เก็บใน flash memory หรือ เครื่องคอมพิวเตอร์เครือข่าย

ในส่วนของ configuration file ทั่วไปจะดึงค่า configuration file ใน nonvolatile memory มาใช้งาน ถ้า Router ตรวจพบว่า configuration file ใน nonvolatile memory นั้นมีปัญหา ก็จะดูว่ามีกำหนดให้นำค่า configuration มาจากเครื่องคอมพิวเตอร์เครือข่ายหรือไม่ ถ้ามีก็จะดึงค่า configuration มาจากเครื่องคอมพิวเตอร์เครือข่าย ถ้าไม่ได้มีการกำหนดเอาไว้ก็จะเข้าสู่ mode setup ให้โดยอัตโนมัติเพื่อรอ configuration file อันใหม่ ซึ่งปัญหาที่ไม่สามารถดึงค่า configuration file

ใน nonvolatile memory อาจเกิดจากการ checksum แล้วพบว่าเกิด error ก็จะไม่สามารถอ่านค่า configuration ได้

## 2.2 วิธีการต่างๆในการปรับแต่งค่าของ Cisco Router

ในการปรับแต่งค่าของเราเตอร์นั้นทำได้หลายวิธีด้วยกัน ดังนี้คือ

2.2.1 Online Configure เป็นวิธีที่มีการทำการแก้ไขและมีผลทันทีซึ่งในแบบ online configure นี้จะแบ่งเป็น 2 แบบ คือ วิธีที่ใช้ telnet เข้าไปที่ router ซึ่งวิธีที่ใช้ telnet นี้จะต้องมีการกำหนด IP address ก่อนและ router ต้องค่ออยู่บนระบบเครือข่าย ซึ่งวิธีการ telnet เข้าไปที่ router จะสามารถกำหนดค่าต่างของ router ได้ 2 วิธีซึ่งวิธีแรกจะเป็นวิธีที่ใช้ได้โดยการพิมพ์คำสั่ง setup เพื่อเข้าสู่ setup mode ซึ่งโดยทั่วไปแล้ววิธีนี้จะใช้กับครั้งแรก เนื่องจากคำสั่ง setup นั้นจะสามารถกำหนดค่าต่างๆที่เป็นค่ามาตรฐานเท่านั้น ถ้าต้องการจะมีการกำหนดค่าต่างๆที่เพิ่มมากขึ้น ก็จะต้องใช้วิธีสั่งคำสั่งแต่ละคำสั่ง ซึ่งวิธีที่สั่งแต่ละคำสั่งนี้เป็นวิธีที่คนโดยทั่วไปนิยมใช้ในการ configure router ซึ่งวิธีนี้ต้องอาศัยความรู้และความชำนาญพอสมควรในการสั่งแต่ละค่าซึ่งวิธีนี้สามารถกำหนดค่า configuration ทุกอย่างที่มีของระบบได้ และอีกวิธีหนึ่งที่สามารถเข้าไป configure ได้ก็คือวิธีในการ configure ผ่าน console port ซึ่งเมื่อต่อเทอร์มินัลเข้ากับ router โดยผ่าน console port แล้วก็ยังสามารถทำการกำหนดค่าต่างเหมือนกับการ telnet เข้าไปที่ router ซึ่งก็สามารถเข้าไปสู่ setup mode หรือสั่งแต่ละคำสั่งได้เช่นกัน

2.2.2 Offline Configure วิธีนี้เป็นการ configure ที่บนเครื่องคอมพิวเตอร์เครื่องหนึ่งจนเสร็จแล้วจึงทำการส่งข้อมูลที่ทำการ configure ไปยัง router และบันทึกเก็บไว้ใน nonvolatile memory ซึ่งวิธีการนี้สามารถทำได้หลายวิธีซึ่งจะอธิบายในแต่ละวิธีดังนี้

### 2.2.2.1 หลักการทำงานของ TFTP server

หลักการทำงานของ TFTP server สามารถทำได้ 2 แบบซึ่งแบบแรกจะเป็นวิธีที่กำหนดให้เครื่องที่เป็น Unix host เป็น TFTP server และ router เป็นเครื่องร้องขอ (client) กับอีกวิธีหนึ่งคือวิธีที่ให้เครื่อง router เป็นเครื่อง TFTP server และเครื่อง Unix host เป็นเครื่องที่ทำการร้องขอ

กรณีที่เครื่อง Unix host เป็น TFTP server และเครื่อง router เป็นเครื่องร้องขอนั้น ที่เครื่อง router สามารถทำการสำเนา configuration file จาก Unix host มาที่ router ได้และก็สามารถทำสำเนาจากเครื่อง router ไปยัง Unix host ได้เช่นกัน แต่ในกรณีที่ทำการสำเนาไปยังเครื่อง Unix host จะต้องทำการสร้างแฟ้มนั้นขึ้นมาก่อน และกรณีที่สำเนาข้อมูลจาก router มายัง Unix host นั้นได้ เครื่องที่เป็น Unix host จะต้องเปิดบริการที่เป็น TFTP server เอาไว้ ซึ่งเหตุนี้เอง ถ้ามีการเปิดบริการ

TFTP ที่ Unix host จะทำให้เกิดความเสียหายทางด้านความปลอดภัยได้ ซึ่งโดยทั่วไปแล้ว เครื่องให้บริการ Unix host จะไม่เปิดบริการ TFTP เอาไว้

กรณีที่ให้เครื่อง router เป็นเครื่อง TFTP server และเครื่อง Unix host เป็นเครื่องที่ร้องขอ ส่วนใหญ่จะใช้ในกรณีที่ต้องการสำเนาระบบปฏิบัติการ เมื่อทำการตั้งค่าให้ router เป็น TFTP server แล้ว router จะรับการร้องขอที่เป็น TFTP เมื่อได้รับการร้องขอที่เป็น TFTP ก็จะทำการทำสำเนาระบบปฏิบัติการ (system image) ที่อยู่ใน ROM หรือ ระบบปฏิบัติการ(system image) ที่อยู่ใน flash memory ส่งไปที่เครื่องที่ร้องขอ ซึ่งการร้องขอที่เป็น TFTP จะต้องใช้ชื่อ file ที่มีการระบุใน router configuration โดยหลักการของ router ในการเลือกว่าจะสำเนาจากใน ROM หรือ ใน flash memory นั้นมีอยู่ว่า ถ้า file ที่ระบุไม่ได้เก็บอยู่ใน flash memory ส่วนที่เก็บอยู่ใน ROM ก็จะถูกส่งออกไป ถ้า file ที่ระบุไว้มีอยู่ใน flash memory ก็จะส่ง file นั้นไปที่เครื่องที่ร้องขอมา ในบางครั้งการส่งโดยใช้ TFTP ก็จะมีผลผิดพลาดเกิดขึ้น และเนื่องจากการใช้ TFTP นั้นมีจุดอ่อนอยู่หลายจุดในด้านความปลอดภัยทำให้วิธีนี้จึงไม่เหมาะกับการใช้งาน

#### 2.1.2.2 หลักการทำงาน RCP (remote copy) server

หลักการการทำงานของ RCP server สามารถทำได้ 2 แบบซึ่งแบบแรกจะเป็นวิธีที่กำหนดให้เครื่องที่เป็น Unix host เป็น RCP server และ router เป็นเครื่องร้องขอ (client) กับอีกวิธีหนึ่งคือวิธีที่ให้เครื่อง router เป็นเครื่อง RCP server และเครื่อง Unix host เป็นเครื่องที่ทำการร้องขอ

กรณีที่ให้เครื่อง Unix host เป็นเครื่อง RCP server และให้เครื่อง router เป็นเครื่องที่ทำการร้องขอ ถ้าเครื่อง router ต้องการที่จะทำการสำเนา configuration file มายังเครื่อง Unix host ที่เครื่อง Unix host จะต้องมีการกำหนดสิทธิ์ให้เครื่องที่ทำการร้องขอมีสิทธิ์ในการสำเนาไปที่เครื่อง Unix host โดยเครื่อง Unix host จะมีการกำหนดสิทธิ์ในแฟ้ม .rhosts ซึ่งแฟ้มนี้จะกำหนดชื่อเครื่องคอมพิวเตอร์และชื่อผู้ที่ต้องการทำการสำเนาไปยังเครื่อง Unix host จึงจะทำการสำเนา configuration file มาได้ แต่ถ้าในกรณีที่เครื่องร้องขอที่เป็น router ต้องการที่จะสำเนา configuration file จากเครื่องที่เป็น Unix host มายัง router ก็สามารถทำการสำเนาได้เลย ซึ่งวิธีนี้จะมีการจัดการทางด้านความปลอดภัยมากกว่าวิธีที่ใช้เป็น TFTP server เนื่องจากวิธีนี้มีการตรวจสอบสิทธิ์แต่การใช้งาน TFTP server ไม่ต้องการตรวจสอบสิทธิ์อย่างใดเลย

กรณีที่เครื่อง router เป็นเครื่อง RCP server และเครื่อง Unix host เป็นเครื่องที่ใช้ในการร้องขอ วิธีนี้สามารถทำได้ทั้งในกรณีที่ทั้ง rcp และ rsh ซึ่ง rcp คือการที่เครื่อง Unix host สามารถทำการสำเนาข้อมูลต่างๆมายังเครื่อง router และยังสามารถสำเนาข้อมูลจาก router ไปยังเครื่อง Unix host ได้ด้วย ส่วนในกรณีของ rsh คือการที่เครื่อง Unix host สามารถทำคำสั่งต่างบน router ได้และที่เครื่อง router ก็สามารถทำคำสั่งต่างๆบน Unix host ได้ โดยไม่ต้องทำการเข้าไปที่

เครื่องนั้นๆ เมื่อทำคำสั่งบนเครื่องต่างๆเสร็จเรียบร้อยแล้วก็จะจบการเชื่อมต่อเองโดยอัตโนมัติ ซึ่งการใช้ rsh นั้นจะมีประโยชน์ในแง่ของการดูสถานะของอุปกรณ์ต่างๆหลายๆเครื่อง โดยไม่ต้องทำการเชื่อมต่อกับเครื่องนั้นก่อน ในกรณีที่ต้องการทำคำสั่งต่างๆบนเครื่อง Unix host ก็จะใช้เพิ่ม .rhosts เช่นเดียวกับ rcp ในการกำหนดสิทธิในการทำคำสั่งต่างๆ

หลักการในการ configure router โดยผ่านทาง rcp นั้น จะต้องกำหนดการให้สิทธิในการเข้าถึงข้อมูลบน router ซึ่งการกำหนดสิทธิในการเข้าถึงข้อมูลในตัว router จะมีการเก็บค่าคล้ายๆ กับ .rhost file ในระบบ Unix แต่การเก็บฐานข้อมูลการเข้าถึงข้อมูลบน router จะแบ่งออกเป็น 3 ส่วนด้วยกันคือ

1. Local username (ชื่อผู้ใช้งานบน router)
2. Remote host (ชื่อเครื่องเครือข่าย)
3. Remote user (ชื่อผู้ใช้ในเครื่องเครือข่าย)

ในการตรวจสอบการใช้สิทธิในใช้งานบน router จะแตกต่างตรงกับในกรณีของเพิ่ม .rhosts ซึ่งจะถูกเก็บใน directory ของบัญชีผู้ใช้ (user account) ในเครื่องเครือข่ายนั้น ดังนั้นในเพิ่ม .rhosts จึงมีการเก็บค่าเฉพาะ remote user และ remote host เท่านั้น แต่ใน router ไม่ได้ใช้หลักการของบัญชีผู้ใช้จึงต้องมีการเก็บ local username , remote host และ remote user ซึ่งใน router สามารถกำหนดชื่อ router เป็น local username ก็ได้ ซึ่งในกรณีของแคว้นบน router จะใช้ DNS ในการตรวจสอบชื่อเครื่องคอมพิวเตอร์ กับ IP address ถ้าทำการตรวจสอบแล้วว่า IP address ไม่ตรงกัน router ก็จะไม่ปฏิบัติการใช้งานผ่าน rcp เนื่องจากวิธีนี้ในด้านความปลอดภัยก็จะมีมากกว่าในกรณีที่เป็นการใช้ในโดยการกำหนดให้เป็น TFTP server จึงเหมาะสมกับการกระจาย configuration ไปยัง router หลากๆตัว นอกจากนี้การเริ่มต้นระบบปฏิบัติการ (boot system image) จากเครื่องคอมพิวเตอร์เครือข่ายโดยใช้ rcp นั้น เนื่องจาก rcp จะใช้งานผ่าน TCP protocol ทำให้การส่งข้อมูลเชื่อถือได้ ถ้ามีการ boot router จากเครื่องเครือข่ายโดยใช้ rcp นั้น router จะหาระบบปฏิบัติการ (system image) ในห้อง (directory) ของผู้ใช้งานในเครื่องเครือข่าย ซึ่งถ้าเป็นในระบบ Unix จะสามารถระบุชื่อผู้ใช้งานในเครื่องเครือข่ายนั้นได้ นอกจากนี้ยังสามารถบีบอัดระบบปฏิบัติการที่ส่งมาจากเครื่องเครือข่ายได้ด้วย การที่จะต้องมีมีการบีบอัดข้อมูลนั้นเนื่องจากว่า ถ้าในกรณีที่ ระบบปฏิบัติการมีขนาดใหญ่จะไม่สามารถใส่ใน หน่วยความจำได้ เพราะทุกๆครั้งที่มีการดึงระบบปฏิบัติการมาจากเครื่องเครือข่ายจะต้องมาทำการนำระบบปฏิบัติการมาไว้ที่หน่วยความจำ ทุกครั้ง

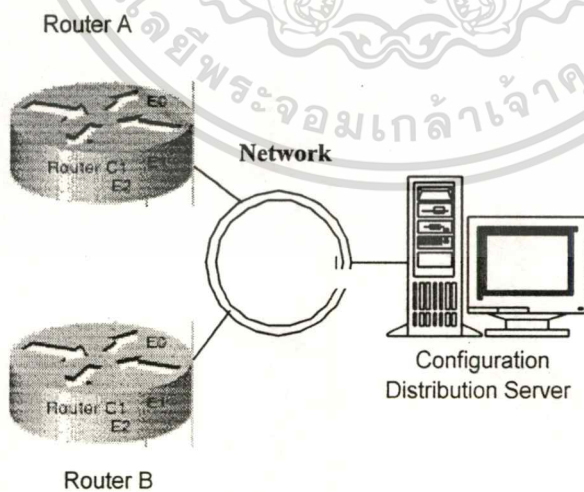
### 2.1.2.3 หลักการทำงานของ SNMP protocol

การทำงานในลักษณะนี้จะมีการร้องขอโดยผ่าน Object id เพื่อไปกำหนดค่าต่างๆของ MIB และถ้าต้องการสำเนาข้อมูลที่เป็น configuration file นั้น ไปยังเครื่อง router ใต้นั้นจะต้องเปิด

สิทธิในการอ่านและเขียน SNMP community ได้ ซึ่งถ้าเปิดให้อ่านและเขียนได้นั้นอาจจะเกิดความเสียหายได้ และกรณีนี้จะไม่เหมาะกับการ Configure เนื่องจากโครงสร้างของ Configuration file นั้นเป็น text ที่เก็บอยู่ใน NVRAM และวิธีการนี้จะไม่สามารถกำหนดค่า configuration ต่างๆ ได้ทั้งหมดเนื่องจากตัวแปร MIB ที่ใช้ในการกำหนดค่าต่างๆของ router มีไม่ครบทั้งหมดที่ต้องการ ในกรณีของ router จะมีค่า MIB เฉพาะในส่วนที่เป็นการตรวจสอบเครือข่ายว่ามีการใช้งานมากน้อยเพียงใด มีจำนวน packet เข้ามาผ่านที่ port router มากน้อยเพียงใด เป็นต้น เพราะฉะนั้น วิธีการนี้จึงไม่เหมาะในการ configure router

### 2.3 หลักการในการติดต่อระหว่างเราเตอร์ กับระบบ

การติดต่อระหว่างเราเตอร์กับระบบการกระจายและการปรับแต่งข้อมูลของเราเตอร์นั้น จะต้องมีการกำหนดค่าต่างๆในการอนุญาตในการติดต่อระหว่างเราเตอร์กับระบบ ซึ่งระบบที่อยู่บนเครื่องโฮสต์ยูนิกซ์ จะต้องอนุญาตให้เราเตอร์สามารถสำเนาข้อมูลจากเราเตอร์มายังเครื่องโฮสต์ยูนิกซ์ได้ ซึ่งในการติดต่อระหว่างเราเตอร์กับระบบนั้น ที่เครื่องโฮสต์ยูนิกซ์นั้นจะต้อง telnet เข้าไปที่เราเตอร์และทำการสำเนาข้อมูลจากเราเตอร์มายังเครื่องโฮสต์ยูนิกซ์ หรือ สำเนาจากเครื่องโฮสต์ยูนิกซ์มายังเราเตอร์ ซึ่งการส่งงานจะตั้งเป็นชุดคำสั่งต่างๆ ซึ่งเปรียบเสมือนการจำลองการทำงานในลักษณะคล้ายกับ telnet แต่เป็นลักษณะ offline คือสั่งเป็นชุดๆตามที่ต้องการ ซึ่งการที่จะสำเนาข้อมูลจากเครื่องโฮสต์ยูนิกซ์ไปยังเราเตอร์นั้น จะสั่งงานโดยเราเตอร์ โดยที่โปรแกรมจะ telnet เข้าไปที่เราเตอร์ และเราเตอร์จะเป็นตัวส่งงานในการสำเนาข้อมูลต่างๆดังรูปที่ 2.2



รูปที่ 2.2 การเชื่อมต่อระหว่างระบบกับเราเตอร์

จากรูปถ้าต้องการติดต่อระหว่างเราเตอร์ A และเราเตอร์ B กับเครื่องโฮสต์ยูนิคซ์ โปรแกรมที่เครื่องโฮสต์ยูนิคซ์จะทำการ telnet ไปยังเราเตอร์ A และ เราเตอร์ B ตามลำดับจากนั้น จะทำการสำเนาข้อมูลการปรับแต่งค่าของเราเตอร์มายังระบบ จากนั้นระบบจะทำงานตามที่ผู้ดูแลระบบเป็นคนสั่งงาน เช่น ถ้าต้องการเพิ่มการปรับแต่งข้อมูลของเราเตอร์ก็ตรวจสอบดูข้อมูลที่สำเนาว่ามีส่วนไหนต้องการเพิ่ม ถ้าต้องการเพิ่มก็ทำการปรับปรุงค่าการปรับแต่งของเราเตอร์จากนั้นก็ทำการสำเนาจากเครื่องโฮสต์ยูนิคซ์ไปยังเราเตอร์ ตามลำดับ

## 2.4 วิธีการในการเข้ารหัสข้อมูลที่เกิดขึ้นบนระบบ

การเก็บข้อมูลที่เป็นตัวอักษรอยู่บนระบบ มีความไม่ปลอดภัยสูง เนื่องจากมีบางแฟ้มข้อมูล เป็นข้อมูลที่เก็บเกี่ยวกับรหัสลับทั้งหมดของเราเตอร์ ซึ่งถ้ามีผู้บุกรุกสามารถนำแฟ้มข้อมูลนั้นไปได้ ก็จะทำให้เกิดความเสียหายได้ ดังนั้นแฟ้มที่ใช้ในการเก็บข้อมูลเกี่ยวกับรหัสนั้นจึงต้องมีการเข้ารหัสข้อมูลเกิดขึ้น ซึ่งในปัจจุบันการเข้ารหัสก็มีด้วยกันหลายวิธี เช่น DES (Data Encryption Standard) , public key private key เป็นต้น

การเก็บแฟ้มข้อมูลของรหัสนั้นในระบบนั้น เพื่อความสะดวกจึงได้เลือกใช้การเข้ารหัสข้อมูล ในแบบ DES ซึ่งเป็นการเข้ารหัสข้อมูลที่ใช้รหัส (key) เดียว คือ ใช้ตัวที่เข้ารหัสและถอดรหัสนั้นเป็น ตัวเดียวกัน ซึ่งการเข้ารหัสแบบ DES นี้เป็นวิธีที่ใช้กันอย่างแพร่หลาย โดยการเข้ารหัสวิธีนี้จะเข้ารหัสเป็นบล็อก ข้อมูลที่จะเข้ารหัสจะถูกแบ่งออกเป็นบล็อกเท่าๆกัน ซึ่งการใช้ DES อัลกอริทึมนั้น จะแบ่งข้อมูลเป็นบล็อกโดยจะแบ่งเป็นบล็อกละ 64 บิตนำมาเข้ารหัสเป็นข้อมูลที่เข้ารหัส (ciphertext) โดยใช้ 56 บิตในการควบคุม ซึ่งกระบวนการเข้ารหัสนั้นจะเป็นกระบวนการที่ทำการเข้ารหัสซ้ำกัน 16 รอบ และในการถอดรหัสนั้นก็จะเป็นกระบวนการที่ทำซ้ำเหมือนกับการเข้ารหัส ซึ่งในการพัฒนาและหัวใจหลักของการออกแบบการเข้ารหัสข้อมูลแบบ DES นั้นอยู่ที่ 3 ส่วนด้วยกันคือ DES permutation (P-box) ,substitutions (S-box) และ key schedules<sup>1</sup>

<sup>1</sup> Permutation เป็นคำเริ่มต้นของข้อความที่ถูกแบ่งเป็นบล็อกพร้อมที่จะทำการโยกย้ายและสลับที่

Substitution เป็นการสลับที่ของข้อมูล permutation ซึ่งถ้ามีการสลับกันเป็นจำนวน 16 รอบจะได้เป็นค่า Inverse permutation ซึ่งทำให้ที่แบ่งออกเป็น 2 ส่วนไม่เปลี่ยนแปลง ในด้านความปลอดภัยใช้เพียง 5 รอบก็เพียงพอแล้ว

Key schedule เป็นการเลือกเอา key 56 บิตจาก key 64 บิต ซึ่งทั้ง 56 บิตสามารถจะเลื่อนเพื่อใช้ในการเข้ารหัส ซึ่งการเข้ารหัสที่เป็นไปได้จะเป็นไปได้ถึง  $2^{56}$  รูปแบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวมาแล้วข้างต้นเกี่ยวกับการทำงานของซิสโรรเตอร์ และวิธีการปรับแต่งค่าต่างๆของเรเตอร์นั้นจะเห็นได้ว่าวิธีการปรับแต่งค่าของเรเตอร์นั้นมีอยู่ด้วยกันหลายวิธี ซึ่งแต่ละวิธีก็จะมีข้อดีข้อเสียที่แตกต่างกันไป ซึ่งในที่นี้โปรแกรมจะใช้วิธีการปรับแต่งค่าของเรเตอร์โดยการหลักการของการกำหนดค่าของเครื่องยูนิคซ์ให้เป็น rcp server เนื่องจากวิธีนี้เป็นวิธีที่ดีที่สุดและง่ายที่สุดเมื่อเทียบกับการกำหนดเป็น ftp หรือ การใช้ snmp นอกจากนี้ในการเข้ารหัสข้อมูลที่เป็นรหัสลับนั้นก็เลือกใช้วิธี DES เนื่องจากการเข้ารหัสแบบ DES เป็นการเข้ารหัสที่เป็นมาตรฐานที่มีการใช้งานอยู่บนเครื่องยูนิคซ์ซึ่งในกรณีที่ต้องการย้ายโปรแกรมไปรันบนเครื่องยูนิคซ์อื่นๆ เช่น SUN OS หรือ Digital unix ก็สามารถทำได้



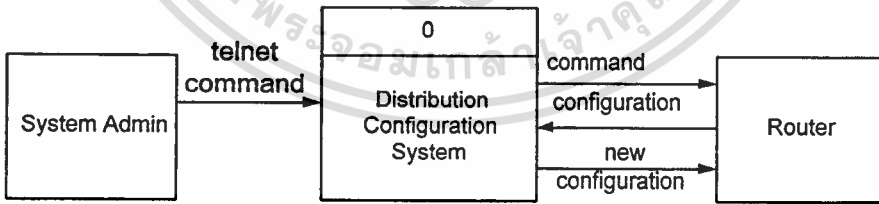
# บทที่ 3

## การออกแบบระบบงานการกระจายค่าการปรับแต่ง ของ Cisco Router

การออกแบบระบบงานการกระจายค่าการปรับแต่งของ Cisco เราเตอร์ หลังจากที่ได้มีการวิเคราะห์หาข้อมูลเกี่ยวกับระบบของเราเตอร์ และระบบงานที่จะเชื่อมต่อกับโฮสต์ยูนิคซ์แล้ว สามารถนำเสนอความสัมพันธ์ระหว่างระบบกับเราเตอร์ได้ในรูปแบบของ DFD (Data Flow Diagram) และนอกจากนี้ยังแสดงถึงรายละเอียดของแฟ้มข้อมูลต่างๆ ในระบบการกระจายและการปรับแต่งค่าของเราเตอร์ซึ่งเก็บอยู่ในรูปของแฟ้มข้อมูลต่าง ซึ่งการที่เก็บข้อมูลต่างๆ ในรูปของแฟ้มข้อมูลเนื่องจากข้อมูลที่เก็บอยู่ในเราเตอร์นั้นเป็นข้อมูลที่เป็นแฟ้มหนึ่งแฟ้ม เพื่อให้สะดวกในการค้นหาข้อมูลจากระบบไปยังเราเตอร์เป็นไปได้สะดวก และเนื่องจากข้อมูลที่เก็บอยู่ในระบบไม่มาก จึงสามารถเก็บอยู่ในรูปของแฟ้มข้อมูลได้

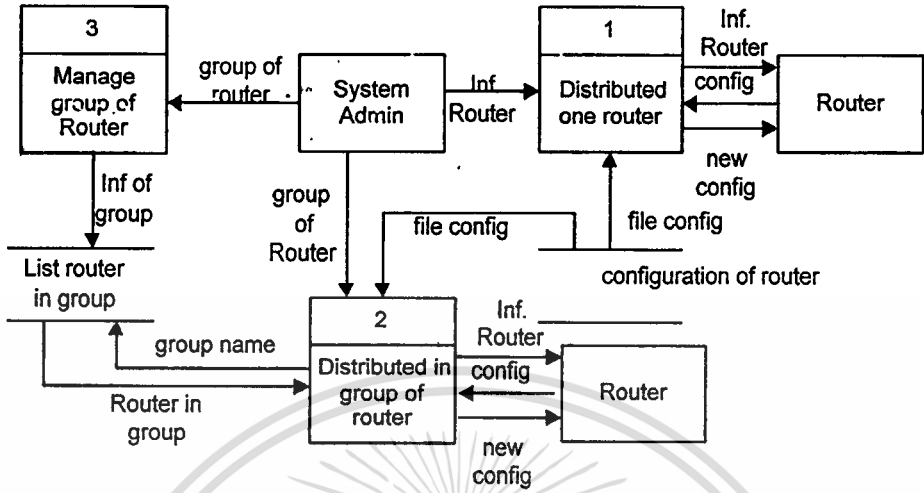
### 3.1 การออกแบบระบบงาน

การพัฒนาโปรแกรมการกระจายและการปรับแต่งค่าของเราเตอร์นี้ ใช้การวิเคราะห์ความสัมพันธ์โดยใช้วิธีดาต้าโฟลว์ไดอะแกรม DFD (Data Flow Diagram) ซึ่งสามารถแสดงความสัมพันธ์ในแต่ละระดับชั้นได้ดังต่อไปนี้



รูปที่ 3.1 Context Diagram ของระบบ

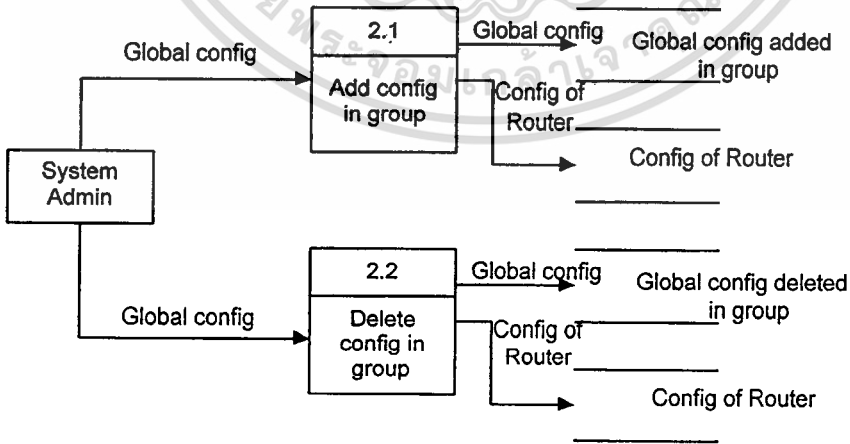
จากรูปที่ 3.1 Context Diagram เป็นการแสดงถึงภาพรวมของระบบการทำงานต่างของระบบซึ่งจะมีผู้ดูแลระบบทำการติดต่อกับระบบการกระจายและการปรับแต่งข้อมูลและระบบจะทำการติดต่อกับเราเตอร์



รูปที่ 3.2 DFD level 1 ของระบบ

จากรูปที่ 3.2 จะแบ่งเป็น 3 กระบวนการด้วยกัน คือ

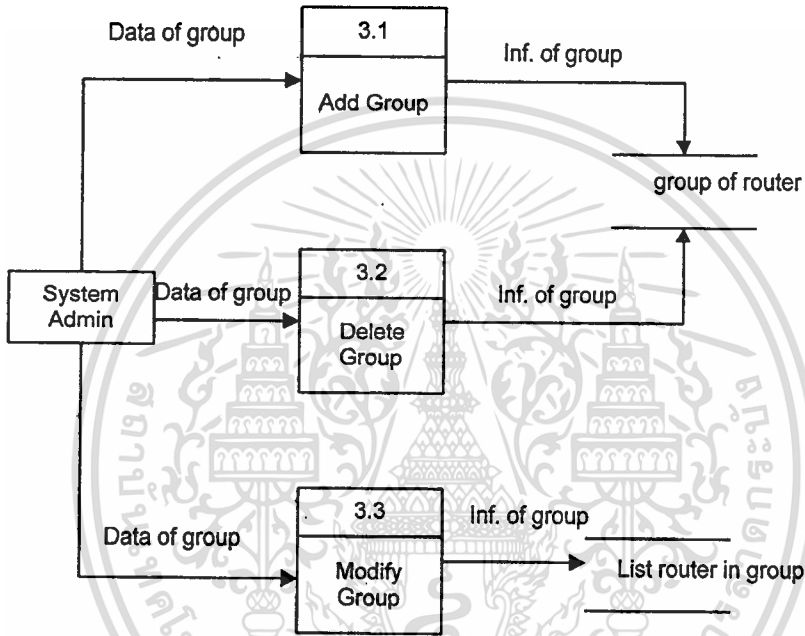
1. การปรับแต่งค่าของเราเตอร์เพียงตัวเดียว
2. การปรับแต่งค่าของเราเตอร์เป็นกลุ่มของเราเตอร์
3. การจัดการกับกลุ่มของเราเตอร์



รูปที่ 3.3 DFD level 2 of process 2

จากรูปที่ 3.3 จะเป็นการแสดงรายละเอียดในการกระจายค่าการปรับแต่งข้อมูลของเราเตอร์ ซึ่งจะมีอยู่ 2 กระบวนการด้วยกันคือ

1. การเพิ่มค่าการปรับแต่งข้อมูลการกระจายภายในกลุ่ม
2. การลบค่าการปรับแต่งข้อมูลการกระจายภายในกลุ่ม



รูปที่ 3.4 DFD level 2 of process 3

จากรูปที่ 6. เป็นการแสดงรายละเอียดการจัดการกับกลุ่มของเราเตอร์ ซึ่งการจัดการจะแบ่งเป็น 3 ส่วนด้วยกันคือ

1. การเพิ่มกลุ่มของเราเตอร์ว่ากลุ่มนี้มีเราเตอร์ตัวใดอยู่ในกลุ่มบ้าง
2. การลบกลุ่มของเราเตอร์
3. การแก้ไขข้อมูลของเราเตอร์ภายในกลุ่ม

### 3.2 ลักษณะการเก็บข้อมูลต่างๆในระบบงาน

การเก็บข้อมูลต่างๆจะเก็บในลักษณะเป็นแฟ้มข้อมูลที่เป็นตัวอักษร (text) ซึ่งรายละเอียดและรูปแบบการเก็บข้อมูลในแต่ละแฟ้มข้อมูลจะแสดงในรายละเอียดดังนี้

1. แฟ้มข้อมูลการปรับแต่งค่าของเราเตอร์ โดยแฟ้มนี้จะเก็บข้อมูลของการกำหนดค่าของเราเตอร์แต่ละตัว

```
Using 27817 out of 129016 bytes
!
version 11.3
no service pad
service timestamps debug uptime
service timestamps log uptime
service udp-small-servers
service tcp-small-servers
!
hostname CAR10
!
boot system flash slot0:slot0
enable secret xxxxxxxxxxxxxxxx
enable password xxxxxxxxxxxxxxxx
!
ip domain-name net.kmitl.ac.th
ip name-server 161.246.52.21
!
interface Ethernet0
 ip address 161.246.52.101 255.255.255.0
 ip broadcast-address 161.246.52.255
 ipx network A1F60B00
 no mop enabled
!
router rip
 redistribute connected
 network 161.246.0.0
!
ip classless

snmp-server community xxxxxx RO 60
!
line con 0
 transport input none
line aux 0
line vty 0 4
 password xxxx
 login

end
```

รูปแบบการจัดเก็บข้อมูลนั้นจะเป็นในลักษณะเดียวกับที่เก็บในซิสโทเรเตอร์ ซึ่งจะแบ่งเป็นส่วนๆ โดยมีเครื่องหมาย “!” เป็นตัวคั่นระหว่างชุดคำสั่ง โดยที่คำสั่งที่เป็นคำสั่งชนิดเดียวกันจะจัดอยู่ในกลุ่มเดียวกัน จากตัวอย่างข้างบนจะเห็นว่าแต่ละชุดคำสั่ง เช่น ชุดคำสั่ง service เมื่อหมดชุดคำสั่งของ service แล้วจะมีเครื่องหมาย “!” จากนั้นก็จะมีชุดคำสั่งเกี่ยวกับ hostname, ip หรือ interface

2. เพิ่มข้อมูลที่เกี่ยวข้องของเราเตอร์ ซึ่งเพิ่มข้อมูลนี้จะเป็นตัวเก็บว่า ณ ขณะนี้มีกลุ่มของเราเตอร์อยู่ทั้งหมดกี่กลุ่ม และมีชื่อกลุ่มว่าอะไรบ้าง ซึ่งลักษณะข้อมูลจะเป็นดังนี้

```
List of Router Group
Groupname
Groupname
Groupname
Groupname
```

รูปแบบของการเก็บเพิ่มข้อมูลนี้จะเก็บเรียงลำดับของกลุ่มที่ทำการเพิ่มเข้าไปซึ่งแต่ละกลุ่มจะเก็บอยู่ในแต่ละบรรทัดตามลำดับ

3. เพิ่มข้อมูลรายละเอียดของเราเตอร์ภายในกลุ่ม ว่ามีเราเตอร์ตัวใดอยู่ในกลุ่มนี้บ้าง และเพิ่มข้อมูลนี้จะเก็บรายละเอียดข้อมูลที่เป็นในการติดต่อกับเราเตอร์ตัวนี้ด้วย เช่น รหัสผู้ใช้ (username), รหัสผ่าน (password) และรหัสลับ (enable password) เป็นต้น ซึ่งลักษณะข้อมูลที่เก็บอยู่ในเพิ่มข้อมูลนี้จะทำการเข้ารหัสข้อมูลโดยใช้เทคนิคการเข้ารหัส DES (Data Encryption Standard) เพื่อความปลอดภัยในการเก็บรักษาข้อมูลรหัสลับของเราเตอร์ทุกๆ ตัว และการที่จะเข้าถึงเพิ่มข้อมูลนี้จะต้องทำการถามรหัสลับก่อนซึ่งถ้ารหัสลับที่ใส่ถูกต้องก็จะยอมให้เข้าถึงและอ่านเพิ่มข้อมูลที่เป็นรหัสลับของเราเตอร์แต่ละตัวได้ ซึ่งลักษณะการเก็บข้อมูลจะเป็นดังนี้

```
Groupname: [groupname]
[Routrname] [username] [password] [enablepassword]
[Routrname] [username] [password] [enablepassword]
[Routrname] [username] [password] [enablepassword]
[Routrname] [username] [password] [enablepassword]
```

ข้อมูลที่ปรากฏนี้จะแบ่งออกเป็น 2 ส่วน ส่วนแรกจะเป็นส่วนของชื่อกลุ่มของเราเตอร์ ซึ่งในส่วนนี้จะเป็นตัวที่ใช้ในการตรวจสอบรหัสลับในการถอดรหัสข้อมูล เมื่อผู้ใช้ใส่รหัสลับของกลุ่มก็จะทำการถอดรหัสซึ่งในกรณีที่ถอดรหัสได้ตรงกับชื่อกลุ่มแล้วแสดงว่ารหัสลับของกลุ่มถูกต้อง



จากตัวอย่างข้างบนจะเป็นกรณีที่ต้องการเพิ่มตารางเส้นทางก็จะเป็นข้อมูลดังนี้

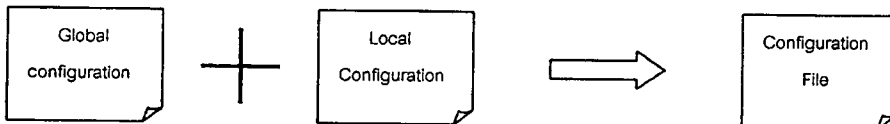
```
ip route 133.29.0.0 255.255.0.0 161.246.51.18
ip route 133.30.0.0 255.255.0.0 161.246.51.18
ip route 133.31.0.0 255.255.0.0 161.246.51.18
ip route 133.32.0.0 255.255.0.0 161.246.51.18
ip route 133.33.0.0 255.255.0.0 161.246.51.18
ip route 133.34.0.0 255.255.0.0 161.246.51.18
ip route 133.35.0.0 255.255.0.0 161.246.51.18
ip route 133.36.0.0 255.255.0.0 161.246.51.18
ip route 133.37.0.0 255.255.0.0 161.246.51.18
ip route 133.38.0.0 255.255.0.0 161.246.51.18
```

5. เพิ่มข้อมูลที่เป็นกรปรับแต่งค่าในส่วนที่มีเฉพาะเราเตอร์ตัวใดตัวหนึ่งเท่านั้น (local configuration) ซึ่งในส่วนนี้จะเป็นส่วนที่ไม่เหมือนกันในแต่ละตัวจึงต้องมีการแยกเก็บข้อมูลเหล่านี้ในเราเตอร์ทุกๆตัวด้วย ซึ่งข้อมูลของเราเตอร์ที่เป็นเฉพาะเราเตอร์นั้นจะเกี่ยวข้องกับ interface เป็นต้น ซึ่งรูปแบบของการเก็บก็จะเป็นส่วนๆแล้วมีเครื่องหมาย “!” คั่นระหว่าง interface ดังนี้

```
!
interface Ethernet0
ip address 161.246.52.101 255.255.255.0
ip broadcast-address 161.246.52.255
ipx network A1F60B00
no mop enabled
!
```

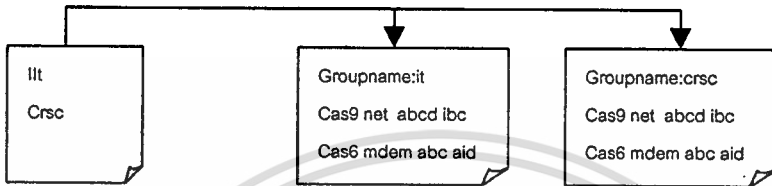
เพิ่มข้อมูลต่างๆสามารถสรุปได้ดังนี้

เพิ่มข้อมูลของเราเตอร์แต่ละตัวจะเป็นเพิ่มข้อมูลที่เป็นข้อมูลส่วนกลางกับข้อมูลส่วนที่เป็นเฉพาะเราเตอร์แต่ละตัว ซึ่งเพิ่มข้อมูลของเราเตอร์ก่อนที่จะส่งไปยังเราเตอร์นั้นจะต้องทำการรวมข้อมูลในส่วนที่เป็น global กับ local configuration ก่อนดังรูปที่ 3.6



รูปที่ 3.6 ความสัมพันธ์ระหว่างข้อมูลการปรับแต่งค่าต่างๆของ cisco router

เพิ่มข้อมูลของกลุ่มของเราเตอร์นั้นเมื่อสร้างขึ้นมาแล้วจะเก็บชื่อกลุ่มในข้อที่ 2 ส่วนข้อมูลของเราเตอร์และรายละเอียดของเราเตอร์ เช่นข้อมูลของรหัสผู้ใช้, รหัสผ่าน, รหัสลับ นั้นจะเก็บในข้อที่ 3. ซึ่งทั้ง 2 ส่วนนี้จะต้องสัมพันธ์กัน ถ้ามีข้อมูลของกลุ่มอยู่ในเพิ่มข้อมูลกลุ่ม แล้วจะต้องมีข้อมูลรายละเอียดของกลุ่มด้วยดังรูปที่ 3.7



เพิ่มข้อมูลกลุ่ม

รายละเอียดของแต่ละกลุ่ม

รูปที่ 3.7 ความสัมพันธ์ระหว่างเพิ่มข้อมูลกลุ่มกับรายละเอียดของกลุ่ม

การออกแบบระบบที่จะใช้ในการกระจายนี้ได้ใช้ DFD (Data Flow Diagram) ในการออกแบบซึ่งระบบจะแบ่งงานออกเป็นส่วนๆตาม DFD และนอกจากนี้ในส่วนของรูปแบบเพิ่มข้อมูลทีในโปรแกรมนี้จะเก็บอยู่ในรูปของตัวอักษร (text file) ซึ่งลักษณะการเก็บ ในกรณีที่เป็นข้อมูลการปรับแต่งค่าของซิสโคนั้นจะใช้รูปแบบการเก็บที่เหมือนกับค่าการปรับแต่งบนซิสโก ซึ่งง่ายต่อการเข้าใจและการส่งข้อมูล อีกส่วนหนึ่งจะเป็นการเก็บเพิ่มข้อมูลในรูปของตัวอักษรเช่นกัน ซึ่งจะมีรูปแบบในการเก็บตามที่ได้กล่าวมาแล้ว เพื่อความสะดวกในการดึงข้อมูลต่างๆไปใช้งาน

## บทที่ 4

### การพัฒนาระบบงาน

การพัฒนาระบบงานขึ้นมาจะต้องมีการวิเคราะห์และออกแบบระบบมาก่อนจึงจะทำระบบได้ดี เมื่อได้มีการวิเคราะห์เรียบร้อยแล้วจึงได้มีการพัฒนา ซึ่งแนวทางในการพัฒนาระบบนั้นจะเป็นในรูปแบบใดนั้นก็ขึ้นอยู่กับปัจจัยต่างๆ ไม่ว่าจะเป็นทางด้านที่ข้องเกี่ยวกับอุปกรณ์ หรือ ทางด้านแอปพลิเคชัน ดังนั้นการจะเลือกพัฒนาไปในแนวทางใดนั้นก็จะต้องดูให้เหมาะสมกับงานที่จะทำ และต้องมองในรายละเอียดว่างานที่จะพัฒนาขึ้นมา นั้น ผู้ใช้เป็นคนใด และระบบที่พัฒนาขึ้นมา นั้นจะช่วยแก้ปัญหาให้กับผู้ใช้ได้มากน้อยเพียงใด

#### 4.1 หลักการทำงานของโปรแกรม

การออกแบบและพัฒนาโปรแกรมให้มีลักษณะการทำงานเป็นแบบโฮสต์เทอร์มินัล (Host-Terminal) ซึ่งการทำงานจะแบ่งออกเป็น 2 ส่วนคือ ส่วนแรกจะเป็นส่วนของเราเตอร์ และส่วนที่สองจะเป็นส่วนของเครื่องยูนิคซ์โฮสต์ โดยในส่วนของเราเตอร์จะต้องมีการกำหนดค่าให้สามารถทำการติดต่อกับเครื่องยูนิคซ์โฮสต์ได้ และในส่วนของโปรแกรมที่ทำการพัฒนานั้นจะทำอยู่บนเครื่องยูนิคซ์โฮสต์ ซึ่งการทำงานผู้ใช้สามารถอยู่ที่เครื่องเทอร์มินัลที่ใดก็ได้หรือจะทำงานที่เครื่องยูนิคซ์โฮสต์ก็ได้ ถ้าผู้ใช้อยู่ที่เครื่องเทอร์มินัลก็ใช้คำสั่ง telnet ไปยังเครื่องยูนิคซ์โฮสต์ก่อนแล้วจึงเรียกใช้โปรแกรม โดยโปรแกรมจะทำหน้าที่ในการติดต่อกับเราเตอร์ และนำค่าต่างๆที่ผู้ใช้กำหนดขึ้นไปใส่ไว้ยังกลุ่มของเราเตอร์ที่ต้องการ ซึ่งกลไกที่เครื่องยูนิคซ์โฮสต์ใช้ในการติดต่อกับเครื่องเราเตอร์ก็ใช้กลไกของการ telnet เข้าไปยังเครื่องเราเตอร์แล้วสั่งให้เราเตอร์ทำงานเพียงแต่การทำงานของโปรแกรมนี้อาจจะทำในลักษณะคล้ายๆกับ batch processing สั่งให้โปรแกรมทำงานจากนั้นโปรแกรมก็จะส่งค่าต่างๆที่ต้องการไปยังเราเตอร์ทุกๆตัวภายในกลุ่ม

## 4.2 โครงสร้างของโปรแกรม

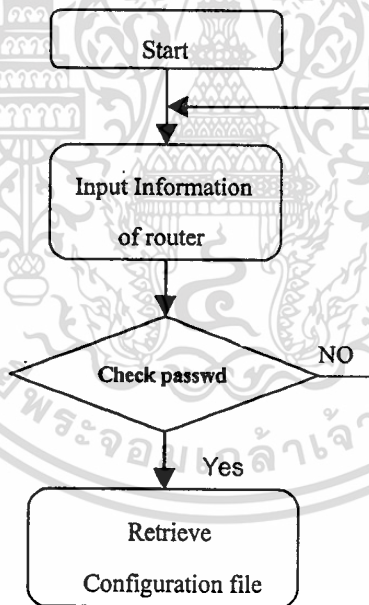
ลักษณะของโครงสร้าง โปรแกรมนั้นจะเป็นสคริปต์การทำงาน ซึ่งแบ่งออกเป็นส่วนต่างๆ ดังนี้

### 4.2.1 ส่วนที่ใช้ในการติดต่อกับเราเตอร์ตัวเดียว

ส่วนนี้จะเป็นส่วนที่ใช้ในการจัดการข้อมูลการกำหนดค่าต่างๆของเราเตอร์เพียงตัวเดียวซึ่งในส่วนนี้จะประกอบด้วยฟังก์ชันการทำงาน 2 ส่วนด้วยกันคือ

#### 4.2.1.1 การแก้ไขข้อมูลของเราเตอร์ (edit configuration of router)

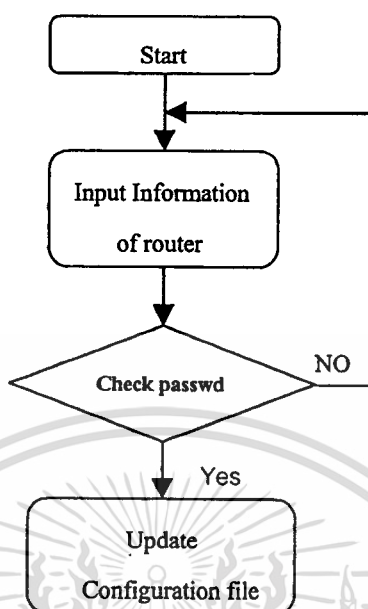
ในกระบวนการนี้ระบบจะมีการตรวจสอบ โดยการให้ใส่ชื่อผู้ใช้ (username), รหัสลับ (password) และ รหัสผ่าน(enable) จากนั้นจึงทำการดึงข้อมูลจากเราเตอร์ เมื่อได้ข้อมูลของเราเตอร์แล้วก็จะทำการตรวจสอบกับข้อมูลที่มีอยู่ที่เครื่องยูนิกซ์โฮสต์ว่าตรงกันหรือเปล่านั้นส่วนใดที่ผิดพลาดบ้าง และทำการแก้ไขค่าต่างๆของเราเตอร์



### รูปที่ 4.1 การแก้ไขข้อมูลของเราเตอร์

#### 4.2.1.2 การเปลี่ยนแปลงข้อมูลของเราเตอร์ (update configuration of router)

ในกระบวนการนี้เมื่อผู้ใช้ทำการเปลี่ยนแปลงแก้ไขข้อมูลต่างๆของเราเตอร์เรียบร้อยแล้วทำการส่งข้อมูลการกำหนดค่าของเราเตอร์ไปยังเราเตอร์ที่กำหนด



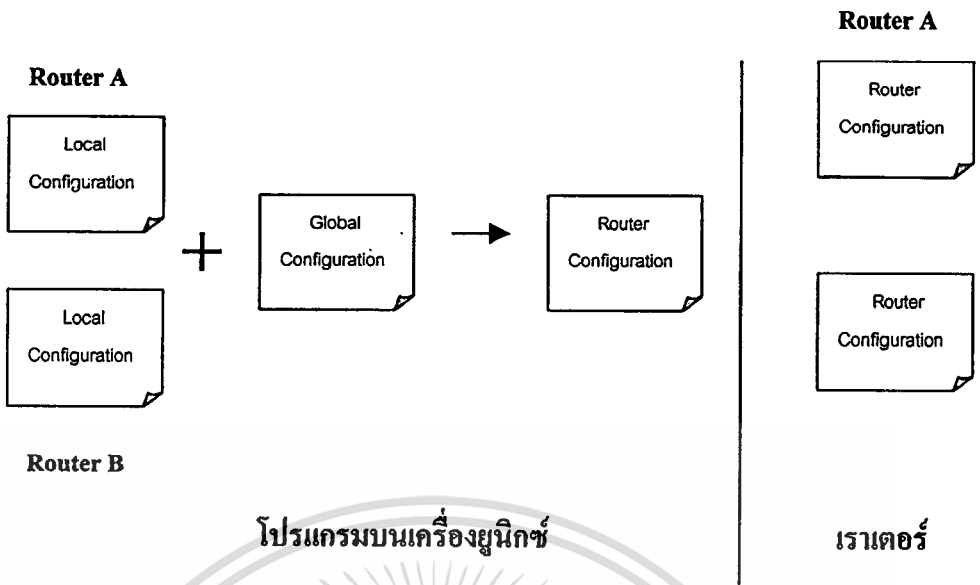
#### รูปที่ 4.2 การเปลี่ยนแปลงข้อมูลของเราเตอร์

##### 4.2.2 ส่วนที่ใช้ในการติดต่อกับกลุ่มของเราเตอร์

ส่วนนี้จะเป็นส่วนที่ใช้ในการจัดการข้อมูลการกำหนดค่าของเราเตอร์ ซึ่งในส่วนนี้จะเป็นการกำหนดค่าของกลุ่มของเราเตอร์ซึ่งในส่วนนี้จะประกอบด้วยฟังก์ชันการทำงาน 2 ส่วนด้วยกันคือ

##### 4.2.2.1 การเพิ่มค่าของข้อมูลการปรับแต่งให้กับกลุ่มของเราเตอร์

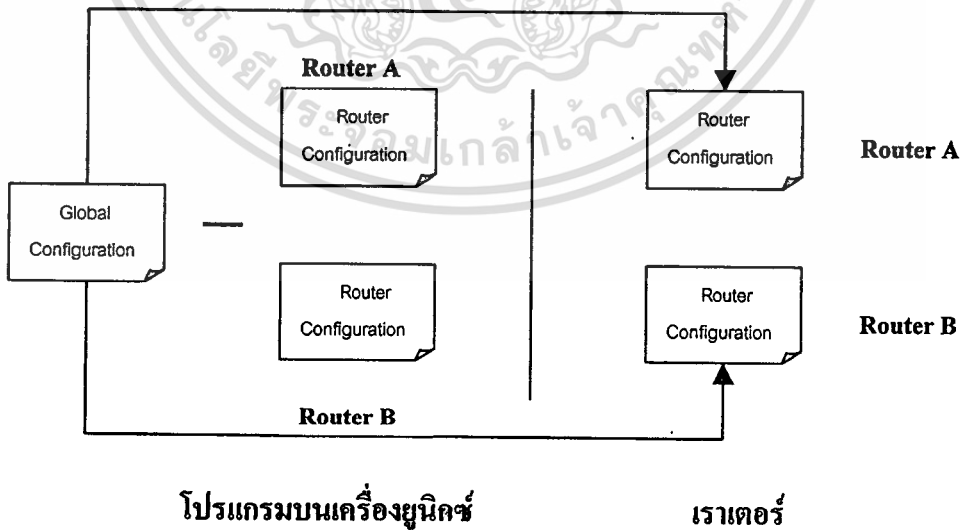
ในส่วนนี้จะเป็นการเพิ่มการกำหนดค่าที่เหมือนกันในกลุ่มของเราเตอร์ เมื่อผู้ใช้ต้องการนำค่าที่เหมือนกันของเราเตอร์ส่งเข้าไปในกลุ่มของเราเตอร์ โปรแกรมจะนำเอาค่าที่ต้องการจะเพิ่มไปใส่ในแฟ้มข้อมูลที่เป็นส่วนกลาง (global Configuration) จากนั้นนำมาพร้อมกับแฟ้มข้อมูลที่เป็นส่วนเฉพาะของเราเตอร์แต่ละตัว (local Configuration) เมื่อรวมกันระหว่างแฟ้มข้อมูลส่วนกลางกับแฟ้มข้อมูลเฉพาะของแต่ละเราเตอร์แล้วจะได้ข้อมูลการปรับแต่งค่าของเราเตอร์จากนั้นจึงทำการส่งข้อมูลไปยังเราเตอร์แต่ละตัวที่เป็นสมาชิกภายในกลุ่มที่ต้องการส่ง



รูปที่ 4.3 การเพิ่มค่าของข้อมูลการปรับแต่งให้กับกลุ่มของเราเตอร์

4.2.2.2 การลบค่าของข้อมูลต่างๆ ให้กับกลุ่มของเราเตอร์

ในส่วนนี้จะคล้ายกับส่วนของการเพิ่มการกำหนดค่าของเราเตอร์ เมื่อผู้ใช้ใส่ค่าของข้อมูลที่ต้องการจะลบเรียบร้อยแล้ว โปรแกรมก็จะทำการนำค่าที่ผู้ใช้ต้องการจะลบ ไปลบในเพิ่มข้อมูลการกำหนดค่าของเราเตอร์จากนั้นก็ทำการส่งเพิ่มข้อมูลของเราเตอร์ไปยังเราเตอร์แต่ละตัวที่อยู่ภายในกลุ่มเดียวกัน



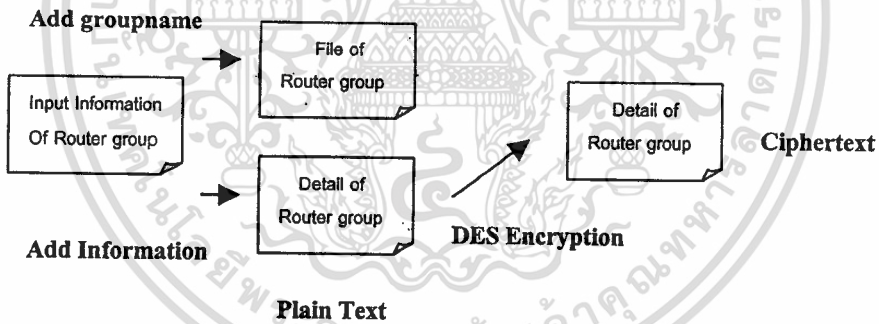
รูปที่ 4.4 การลบค่าของข้อมูลต่างๆ ให้กับกลุ่มของเราเตอร์

4.2.3 ส่วนที่ใช้ในการจัดการกับกลุ่มของเราเตอร์

ส่วนนี้จะเป็นส่วนที่ใช้ในการจัดการกับกลุ่มของเราเตอร์ ซึ่งในส่วนนี้จะเป็นการกำหนดค่าต่างๆของกลุ่มของเราเตอร์ เช่น สมาชิกภายในกลุ่มมีเราเตอร์ตัวใดอยู่บ้าง และข้อมูลของเราเตอร์แต่ละกลุ่มเป็นอะไร โดยในส่วนนี้ข้อมูลที่เก็บไว้ในแฟ้มข้อมูลจะถูกเข้ารหัสข้อมูลเอาไว้ ในกรณีของการจัดการกับกลุ่มของเราเตอร์ที่กล่าวไว้แล้วก็จะต้องถอดรหัสก่อนจึงจะได้ข้อมูลของเราเตอร์แต่ละตัวไปใช้ได้ ซึ่งในส่วนนี้จะประกอบด้วยฟังก์ชันการทำงาน 3 ส่วนด้วยกันคือ

4.2.3.1 การเพิ่มกลุ่มของเราเตอร์ (Add group)

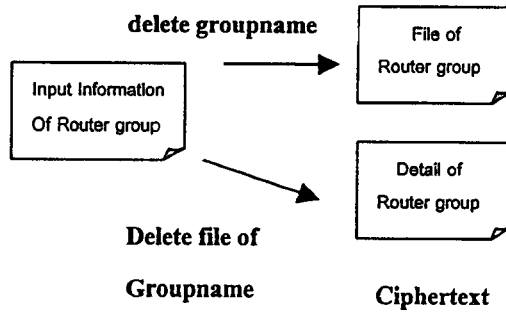
การเพิ่มกลุ่มของเราเตอร์นั้น โปรแกรมก็จะให้ใส่ข้อมูลต่างๆเกี่ยวกับกลุ่ม เช่น ชื่อกลุ่ม, รหัสลับของกลุ่ม, สมาชิกภายในกลุ่ม ซึ่งสมาชิกภายในกลุ่มก็ต้องรู้รายละเอียดเกี่ยวกับสมาชิกแต่ละตัว เช่น รหัสผู้ใช้ (username), รหัสผ่าน (password) และ รหัสลับ (enable) เมื่อผู้ใช้ป้อนข้อมูลเรียบร้อยแล้วข้อมูลจะถูกเข้ารหัสด้วยรหัสลับของกลุ่ม ซึ่งผู้ที่ทำการแก้ไข หรือลบข้อมูลของกลุ่มนี้ก็ต้องรู้รหัสลับของกลุ่ม หรือในการเรียกใช้งานการกระจายการกำหนดค่าของเราเตอร์ภายในกลุ่ม ก็ต้องรู้ว่ารหัสลับของกลุ่มนั้นเป็นอะไรจึงจะกระจายข้อมูลไปยังกลุ่มนั้นได้ ซึ่งลักษณะการทำงานจะเป็นดังรูปที่ 4.5



รูปที่ 4.5 การเพิ่มกลุ่มของเราเตอร์

4.2.3.2 การลบกลุ่มของเราเตอร์ (Delete group)

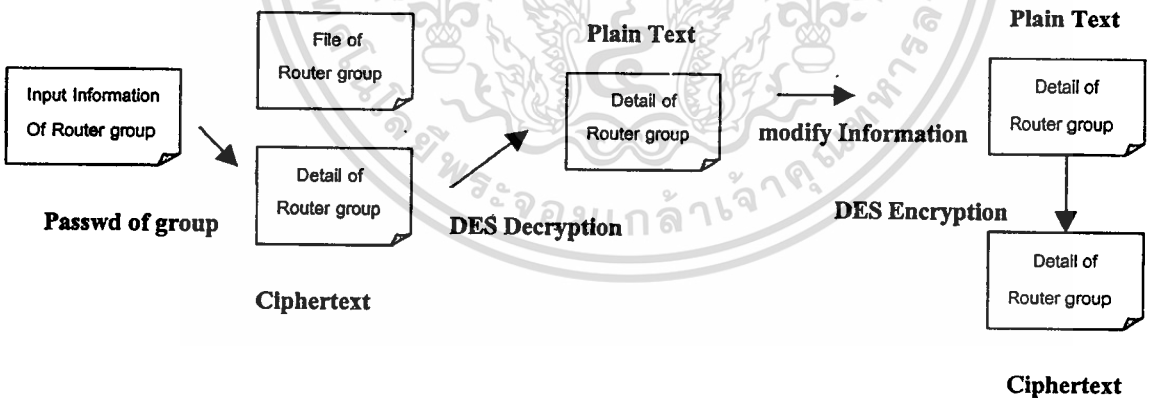
การลบกลุ่มของเราเตอร์ ซึ่งการลบกลุ่มของเราเตอร์นั้นจะต้องรู้รหัสลับของกลุ่มนั้นเพื่อเป็นการตรวจสอบว่า บุคคลนั้นเป็นคนที่สร้างกลุ่มนั้นขึ้นมาจึงจะทำการลบข้อมูลของกลุ่มนั้นได้ เมื่อผู้ใช้ต้องการจะลบข้อมูลของกลุ่ม โปรแกรมก็จะถามว่ารหัสลับของกลุ่มที่ต้องการจะลบจากนั้นระบบก็จะตรวจสอบว่ารหัสลับถูกต้องหรือไม่ ถ้ารหัสลับของกลุ่มถูกต้องก็จะยอมให้ลบกลุ่มนั้น



รูปที่ 4.6 การลบกลุ่มของเราเตอร์

### 4.2.3.3 การแก้ไขข้อมูลภายในกลุ่ม

การแก้ไขข้อมูลภายในกลุ่มนั้น ผู้ใช้จะต้องรู้รหัสลับของกลุ่มนั้นเช่นเดียวกับการลบกลุ่มของข้อมูล เมื่อผู้ใช้ใส่รหัสลับของกลุ่มถูกต้อง โปรแกรมก็จะถามว่าต้องการจะลบเราเตอร์ หรือแก้ไขข้อมูลของเราเตอร์ ถ้าต้องการลบเราเตอร์ โปรแกรมก็จะให้ใส่ชื่อเราเตอร์และทำการลบทันที แต่ถ้าผู้ใช้ต้องการแก้ไขข้อมูลของเราเตอร์เช่น แก้ชื่อผู้ใช้, รหัสผ่าน หรือ รหัสลับก็สามารถทำได้



รูปที่ 4.7 การแก้ไขข้อมูลภายในกลุ่ม

### 4.3 การพัฒนาโปรแกรม

การพัฒนาโปรแกรมในระบบนี้จะเป็นการพัฒนาบนเครื่องยูนิกซ์ โดยใช้สคริปต์ และ คำสั่งบางส่วนของ TCL (TCL command) ซึ่งโปรแกรมนี้จะมีลักษณะเป็นเมนู ให้ผู้ใช้เลือกจะทำในระบบใด ซึ่งการพัฒนานี้จะแบ่งออกเป็น ส่วนของการกำหนดค่าซึ่งในส่วนของการกำหนดค่านั้น จะเป็นการกำหนดในส่วนของเครื่องยูนิกซ์และซิสโกราเตอร์ โดยที่การกำหนดค่านั้นจะเป็นการกำหนดให้เครื่องยูนิกซ์เปิดการทำงานในลักษณะของ RCP เพื่อให้เราเตอร์สามารถส่งข้อมูลการปรับแต่งมายังเครื่องยูนิกซ์ได้ อีกส่วนหนึ่งจะเป็นการพัฒนาโปรแกรมบนเครื่องยูนิกซ์ ซึ่งการกำหนดค่าจะแบ่งออกเป็น 2 ส่วนดังนี้

#### 4.3.1 ส่วนที่เป็นการกำหนดค่าของเราเตอร์

การกำหนดค่าของเราเตอร์ ก็สามารถทำได้โดยการ telnet เข้าไปที่เราเตอร์จากนั้นใส่รหัสผ่าน และ รหัสลับของเราเตอร์จากนั้นก็ทำการกำหนดค่าต่างๆของเราเตอร์โดยการสั่งดังนี้

```
routername> config term
routername\config# ip rcmd remote-username <username>
```

การสั่งในลักษณะนี้เป็นการกำหนดว่า เมื่อมีการส่งข้อมูลการปรับแต่งค่าไปยังเครื่องยูนิกซ์ นั้นจะส่งไปยังรหัสผู้ใช้ที่ได้มีการกำหนดไว้

#### 4.3.2 ส่วนที่เป็นการกำหนดค่าของเครื่องยูนิกซ์ ที่ใช้ในการพัฒนาโปรแกรม

การกำหนดค่าของเครื่องที่ใช้ในการพัฒนาโปรแกรมนั้นจะต้องมีใช้งานผ่านกลไก RPC (Remote Procedure Call) ดังนั้นจึงต้องมีการกำหนดเพิ่มข้อมูลที่ชื่อ .rhosts ในห้อง (directory) ของผู้ใช้ที่ต้องการให้สามารถส่งข้อมูลจากเราเตอร์มายังส่วนนี้ได้ ซึ่งการกำหนดในส่วนนี้จะเป็นอย่างนี้

```
IP address1   Username
IP address2   Username
IP address3   Username
```

ในแฟ้มนี้จะต้องกำหนดไอพีแอดเดรสของเราเตอร์และชื่อผู้ใช้ของเราเตอร์ที่ต้องการจะส่งข้อมูลมายังเครื่องยูนิกซ์ ในกรณีที่เรามีเราเตอร์หลายๆตัวต้องการจะส่งข้อมูลมายังเครื่องยูนิกซ์ ก็สามารถใส่หมายเลขไอพีแอดเดรสของเครื่องที่ 2 และ เครื่องที่ 3 ได้ เป็นต้น

4.3.3 การพัฒนาโปรแกรมบนเครื่องยูนิกซ์ ซึ่งรูปแบบของการใช้งานระบบนี้จะเป็นแบบอักษรทั้งหมด (text mode) และลักษณะการใช้งานจะอยู่ในรูปของเมนูการทำงานในส่วนต่างๆซึ่งในส่วนของหน้าจอหลักจะเป็นรูปที่ 4.8

```

#####
Main Menu of Distributed config Router
1).   Select one Router
2).   Select group of Router
3).   Manage group of Router
x).   exit of program
Please enter your choice => _____
#####

```

รูปที่ 4.8 หน้าจอโปรแกรมหลัก

หน้าจอนี้เป็นหน้าจอเริ่มต้นของโปรแกรม จากนั้นผู้ใช้สามารถเลือกตามที่ต้องการได้ในกรณีที่ต้องการจัดการข้อมูลกับเราเตอร์เพียงตัวเดียวจะปรากฏดังรูปที่ 4.9

```

#####
Do you want to edit or update config
1).   Edit one router
2).   Update one router
x).   Back to main page
Please enter your choice => _____
#####

```

รูปที่ 4.9 หน้าจอการจัดการกับเราเตอร์ตัวเดียว

หน้าจอนี้เป็นหน้าจอการจัดการกับเราเตอร์เพียง 1 ตัวซึ่งจะมีการแก้ไขข้อมูลการกำหนดค่าของเราเตอร์และ การเปลี่ยนแปลงค่าของเราเตอร์

ในกรณีที่ผู้ใช้ต้องการจะกระจายค่าข้อมูลของเราเตอร์เป็นกลุ่มสามารถเลือกจากหน้าจอโปรแกรมหลักโดยเลือกข้อที่ 2. ซึ่งจะปรากฏดังรูปที่ 4.10

```

#####
Do you want to add or delete config in group of router

1). Add config in group of router
2). Delete config in group of router
x). Back to main page
Please enter your choice => ____
#####

```

#### รูปที่ 4.10 หน้าจอการจัดการกับกลุ่มของเราเตอร์

หน้าจอนี้เป็นการจัดการกับกลุ่มของเราเตอร์ซึ่งผู้ใช้สามารถเลือกได้ว่าต้องการจะเพิ่มการกำหนดค่าข้อมูลของเราเตอร์หรือลบข้อมูลเป็นกลุ่มของเราเตอร์ ถ้าผู้ใช้ต้องการเพิ่มข้อมูลการกำหนดค่าผู้ใช้ต้องเตรียมข้อมูลที่จะเพิ่มให้เรียบร้อย จากนั้นเลือกในข้อที่ 1. จะปรากฏดังรูปที่ 4.11

```

#####
What group do you want to add _____
Please enter your group name _____
Please enter your password of group _____
Do you create config file that you want to add in group ALREADY (y/n) ____
#####

```

#### รูปที่ 4.11 หน้าจอการกระจายการเพิ่มข้อมูลไปยังกลุ่มของเราเตอร์

หน้าจอนี้เป็นการกระจายข้อมูลที่เพิ่มเข้าไปยังกลุ่มของเราเตอร์ ซึ่งเมื่อผู้ใช้ตอบตกลงเพื่อต้องการกระจายข้อมูลระบบก็จะทำงานให้โดยทำการสำเนาข้อมูลที่ได้มีการเตรียมไว้ และส่งไปยังเราเตอร์ให้โดยอัตโนมัติ

ในกรณีที่ผู้ใช้ต้องการจะจัดการกับกลุ่มของเราเตอร์ ในหน้าจอโปรแกรมหลักก็จะมีการจัดการกับกลุ่มของเราเตอร์โดยผู้ใช้สามารถเลือกข้อ 3. ซึ่งเมื่อเลือกแล้วจะปรากฏดังรูปที่ 4.12

```

#####
Do you want to add or delete or modify group of router

1). Add group of router
2). Delete group of router
3). Modify group of router
x). Back to main page

Please enter your choice => _____
#####

```

รูปที่ 4.12 หน้าจอการจัดการกับกลุ่มของเราเตอร์

หน้าจอนี้เป็นการจัดการกับกลุ่มของเราเตอร์ ซึ่งจะมีอยู่ 3 หัวข้อด้วยกันคือ การเพิ่มกลุ่มของเราเตอร์ , การลบกลุ่มของเราเตอร์ และ การแก้ไขกลุ่มของเราเตอร์ ถ้าผู้ใช้ต้องการที่จะเพิ่มกลุ่มของเราเตอร์ ก็สามารเลือกข้อที่ 1. ซึ่งจะปรากฏดังรูปที่ 4.13

```

#####
Please enter your group name _____

How many router do you want to add _____

Please enter your password of group name _____
#####

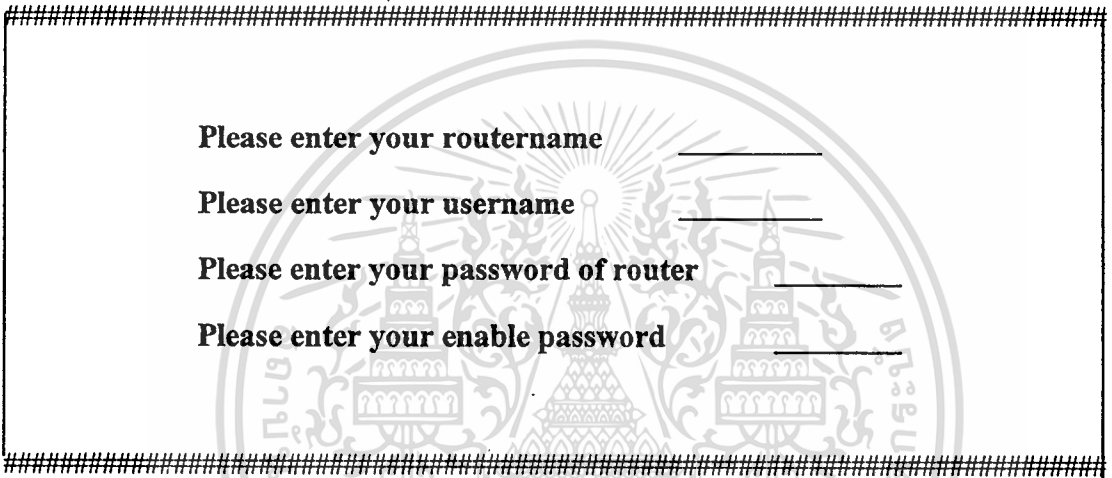
```

รูปที่ 4.13 หน้าจอการเพิ่มกลุ่มของเราเตอร์

จากนั้น โปรแกรมจะให้ผู้ใช้ใส่ข้อมูลของเราเตอร์แต่ละตัว ซึ่งข้อมูลที่ต้องรู้เกี่ยวกับเราเตอร์ที่ต้องจะเพิ่มเข้าไปในกลุ่มนั้นจะมีดังนี้

- รหัสผู้ใช้ (username)
- รหัสผ่าน (password)
- รหัสลับ (enable password)

ซึ่งจะปรากฏดังรูปที่ 4.14



The screenshot shows a configuration window with a decorative border and a watermark of a university seal. It contains four prompts for user input:

- Please enter your routename \_\_\_\_\_
- Please enter your username \_\_\_\_\_
- Please enter your password of router \_\_\_\_\_
- Please enter your enable password \_\_\_\_\_

รูปที่ 4.14 หน้าจอการกรอกข้อมูลของเราเตอร์

หน้าจอนี้เมื่อผู้ใช้กรอกข้อมูลของเราเตอร์ตัวแรกเสร็จเรียบร้อยแล้ว โปรแกรมก็จะขึ้นหน้าจอให้กรอกข้อมูลจนกระทั่งผู้ใช้กรอกข้อมูลของเราเตอร์ครบตามจำนวนที่ระบุไว้ตอนแรกว่าต้องการเราเตอร์จำนวนเท่าใด เมื่อผู้ใช้กรอกข้อมูลเราเตอร์เรียบร้อยแล้ว ข้อมูลทั้งหมดจะถูกเข้ารหัสด้วยรหัสลับของกลุ่มนั้น

ในกรณีที่ผู้ใช้ลืมรหัสลับจะไม่สามารถถอดรหัสข้อมูลที่เข้ารหัสเอาไว้ได้ ซึ่งในการถอดรหัสไม่ได้หมายถึงผู้ใช้จะไม่สามารถเข้าไปแก้ไขข้อมูลภายในกลุ่ม หรือลบข้อมูลของกลุ่มนั้นได้ โดยผ่านทางโปรแกรม เนื่องจากโปรแกรมจะไม่ยอมให้ผ่าน ถ้าผู้ใช้ใส่รหัสลับของกลุ่มไม่ถูกต้อง ดังนั้น ผู้ใช้จะต้องทำการลบข้อมูลกลุ่มโดยการลบเพิ่มข้อมูลของกลุ่มที่อยู่ในห้องของกลุ่มและลบชื่อกลุ่มนั้นทิ้ง จากนั้นทำการสร้างกลุ่มใหม่ขึ้นมา

## 4.4 วิธีการทดสอบโปรแกรม

การทดสอบโปรแกรมนั้นเป็นส่วนสำคัญอีกส่วนหนึ่ง ไม่น้อยไปกว่าการวิเคราะห์และออกแบบระบบซึ่งการทดสอบนั้นในโปรแกรมนี้อาจแบ่งออกเป็น 2 ส่วนด้วยกัน คือ

### 4.4.1 การทดสอบในส่วนของเราเตอร์

การทดสอบในส่วนของเราเตอร์นั้นจะทำการทดสอบ โดยทำการกำหนดค่าต่างๆของเราเตอร์ที่จะทำให้เครื่องยูนิคซ์สามารถทำการสำเนาข้อมูลของการกำหนดค่าต่างๆจากเราเตอร์ไปยังเครื่องยูนิคซ์และสามารถทำการสำเนาจากเครื่องยูนิคซ์ไปยังเราเตอร์ได้ ซึ่งการทดสอบนั้นจะทำการ telnet เข้าไปยังเครื่องเราเตอร์จากนั้นทำการกำหนดค่าดังกล่าวตามที่ได้กล่าวไว้ส่วนที่แล้ว จากนั้นทำการสำเนาข้อมูลจากเราเตอร์ไปยังเครื่องยูนิคซ์ และทำการสำเนาจากเครื่องยูนิคซ์ ไปยังเครื่องเราเตอร์ได้

### 4.4.2 การทดสอบในส่วนของโปรแกรมบนเครื่องยูนิคซ์

การทดสอบในส่วนของโปรแกรมจะแบ่งตามหัวข้อของเมนูในโปรแกรมซึ่งแบ่งได้เป็น 3 หัวข้อดังนี้

#### 4.4.2.1 การทดสอบในส่วนของการส่งข้อมูลไปยังเราเตอร์ตัวเดียว

การส่งข้อมูลไปยังเราเตอร์ตัวเดียวนั้นจะมีการติดต่อกันระหว่างเราเตอร์กับ โปรแกรมซึ่งมีอยู่ 2 ส่วนคือ ส่วนที่เป็นการแก้ไขข้อมูลของเราเตอร์ที่มีอยู่เดิมบนเครื่องเราเตอร์ และ ส่วนที่สองจะเป็นส่วนที่ทำการเปลี่ยนแปลงข้อมูลของเราเตอร์โดยการทดสอบนั้นก็จะเป็นโปรแกรมในส่วนของฟังก์ชันที่ใช้ในการแก้ไขข้อมูลและเรียกใช้ฟังก์ชันของการเปลี่ยนแปลงข้อมูลว่ามีการทำงานถูกต้องหรือไม่

#### 4.4.2.2 การทดสอบในส่วนของการส่งข้อมูลไปยังกลุ่มของเราเตอร์

การส่งข้อมูลของเราเตอร์ไปยังกลุ่มของเราเตอร์นั้นจะมีอยู่ด้วยกัน 2 ฟังก์ชันด้วยกันคือ ฟังก์ชันในการเพิ่มข้อมูลส่วนกลางที่เหมือนกัน ไปยังเราเตอร์ภายในกลุ่ม และ ฟังก์ชันที่สองจะเป็นการลบข้อมูลส่วนกลางที่เหมือนกัน ไปยังเราเตอร์ที่อยู่ภายในกลุ่ม

การทดสอบนั้นจะทำการทดสอบการเพิ่มข้อมูลโดยการสร้างแฟ้มข้อมูลที่ต้องการจะเพิ่มแล้วทำการเพิ่มข้อมูลที่เตรียมไว้ลงในเราเตอร์ทุกๆตัวที่อยู่ภายในกลุ่มนั้น จากนั้นดูว่าข้อมูลที่เพิ่มเข้าไบนั้นถูกต้องหรือไม่ และเพิ่มเข้าไปในเราเตอร์ทุกๆตัวภายในกลุ่มหรือไม่

การทดสอบการลบข้อมูลส่วนกลางของเราเตอร์นั้นจะทำการทดสอบโดยการสร้างแฟ้มข้อมูลที่ต้องการจะลบภายในกลุ่มของเราเตอร์นั้น จากนั้นทำการเรียกฟังก์ชันในการลบนั้น จากนั้น

ตรวจสอบข้อมูลการกำหนดกลุ่มของเราเตอร์ทุกๆตัวที่อยู่ภายในกลุ่มว่าข้อมูลที่ต้องการฉบับนั้นยังมีอยู่ในเราเตอร์นั้นทุกตัวหรือไม่

#### 4.4.2.3 การทดสอบการจัดการเกี่ยวกับกลุ่มของเราเตอร์

การจัดการกับกลุ่มของเราเตอร์นั้นจะแบ่งออกด้วยกันเป็น 3 ฟังก์ชันคือ

- การเพิ่มกลุ่มของเราเตอร์
- การลบกลุ่มของเราเตอร์
- การแก้ไขข้อมูลภายในกลุ่มของเราเตอร์

การทดสอบการจัดการเกี่ยวกับกลุ่มของเราเตอร์นั้นก็จะสามารถทำได้โดยเรียกโปรแกรมและเรียกฟังก์ชันในการเพิ่มกลุ่มของเราเตอร์ เมื่อผู้ใช้กรอกข้อมูลชื่อกลุ่ม, จำนวนเราเตอร์ที่อยู่ในกลุ่ม, รหัสลับของกลุ่ม และรายละเอียดเกี่ยวกับข้อมูลของเราเตอร์เรียบร้อยแล้ว ก็สามารถทำการตรวจสอบ โดยการดูว่าข้อมูลที่ได้เพิ่มลงไปแล้วมีอยู่จริงหรือไม่ และข้อมูลที่เก็บอยู่นั้นถูกเข้ารหัสไว้หรือไม่

การทดสอบเกี่ยวกับการลบข้อมูลของกลุ่มเราเตอร์ สามารถทำได้โดยการเรียกโปรแกรมขึ้นมาจากนั้นก็เรียกฟังก์ชันการลบกลุ่มของเราเตอร์ เมื่อผู้ใช้ใส่ชื่อกลุ่มและใส่รหัสลับของกลุ่มถูกต้องก็จะสามารถลบกลุ่มนั้นได้ ซึ่งการที่ต้องใส่รหัสลับก็เพื่อเป็นการตรวจสอบว่าผู้ใช้ที่จะลบกลุ่มนั้นเป็นผู้ที่เพิ่มกลุ่มนั้นขึ้นมา เมื่อผู้ใช้โปรแกรมลบกลุ่มไปแล้วจากนั้นทำการตรวจสอบเพิ่มข้อมูลของกลุ่มว่าข้อมูลของกลุ่มนั้นยังปรากฏอยู่หรือไม่ และข้อมูลทั้งหมดถูกเข้ารหัสไว้หรือไม่

การทดสอบเกี่ยวกับการแก้ไขข้อมูลภายในกลุ่ม ซึ่งการแก้ไขข้อมูลภายในกลุ่มนั้นมีอยู่ด้วยกัน 3 ส่วนคือ

- การเพิ่มข้อมูลของเราเตอร์ภายในกลุ่ม
- การลบข้อมูลของเราเตอร์ภายในกลุ่ม
- การแก้ไขข้อมูลของเราเตอร์แต่ละตัวภายในกลุ่ม

การทดสอบนั้นก็จะทำการทดสอบทีละส่วน โดยส่วนแรกก็จะทำการทดสอบโดยการเพิ่มข้อมูลของเราเตอร์ก็ทำการเรียกฟังก์ชันการเพิ่มข้อมูลของเราเตอร์ภายในกลุ่ม และก่อนที่โปรแกรมจะให้ใส่รายละเอียดของเราเตอร์ที่จะเพิ่มจะมีการตรวจสอบรหัสลับของกลุ่มก่อน ซึ่งในจุดนี้เป็นการทดสอบว่ารหัสลับของกลุ่มที่สร้างตอนการเพิ่มกลุ่มนั้นใช้ได้ถูกต้องหรือไม่ เมื่อใส่รหัสลับถูกต้องโปรแกรมจะให้ใส่ข้อมูลของเราเตอร์ที่ต้องการเพิ่ม โดยจะให้ใส่รหัสผู้ใช้, รหัสผ่าน และรหัสลับ จากนั้นทำการตรวจสอบดูในเพิ่มข้อมูลว่าเราเตอร์ที่เพิ่มเข้าไปในนั้นถูกต้องหรือไม่และข้อมูลของกลุ่มถูกเข้ารหัสไว้หรือไม่

การลบข้อมูลของเราเตอร์ก็ทำการเรียกฟังก์ชันการลบข้อมูลของเราเตอร์ภายในกลุ่ม และก่อนที่โปรแกรมจะให้ใส่ชื่อของเราเตอร์ที่จะลบจะมีการตรวจสอบรหัสลับของกลุ่มก่อน ซึ่งในจุดนี้เป็นการทดสอบว่ารหัสลับของกลุ่มที่สร้างตอนการเพิ่มกลุ่มนั้นใช้ได้ถูกต้องหรือไม่ เมื่อใส่รหัสลับถูกต้องโปรแกรมจะให้ใส่ชื่อของเราเตอร์ที่ต้องการจะลบ เมื่อโปรแกรมทำงานเสร็จ ทำการตรวจสอบดูในแฟ้มข้อมูลว่าเราเตอร์ที่ลบไปนั้นถูกต้องหรือไม่และข้อมูลของกลุ่มถูกเข้ารหัสไว้หรือไม่

การแก้ไขข้อมูลของเราเตอร์ก็ทำการเรียกฟังก์ชันการแก้ไขข้อมูลของเราเตอร์ภายในกลุ่ม และก่อนที่โปรแกรมจะให้ใส่ชื่อของเราเตอร์ที่จะแก้ไขจะมีการตรวจสอบรหัสลับของกลุ่มก่อน ซึ่งในจุดนี้เป็นการทดสอบว่ารหัสลับของกลุ่มที่สร้างตอนการเพิ่มกลุ่มนั้นใช้ได้ถูกต้องหรือไม่ เมื่อใส่รหัสลับถูกต้องโปรแกรมจะให้ใส่ชื่อของเราเตอร์ที่ต้องการจะแก้ไข และข้อมูลรายละเอียดของเราเตอร์นั้นใหม่ ซึ่งผู้ใช้สามารถอาจจะใส่ข้อมูลของรหัสผู้ใช้, รหัสผ่าน หรือรหัสลับใหม่หรือมีบางส่วนเป็นเหมือนเดิมก็ได้ เมื่อโปรแกรมทำงานเสร็จ ทำการตรวจสอบดูในแฟ้มข้อมูลว่าเราเตอร์ที่แก้ไขไปนั้นถูกต้องหรือไม่และข้อมูลของกลุ่มถูกเข้ารหัสไว้หรือไม่

#### 4.5 ผลการทดสอบ โปรแกรม

หลังจากดำเนินการพัฒนาและแก้ไขปรับปรุงโปรแกรมเสร็จแล้ว ผลการทดสอบโปรแกรมทั้งหมดมีดังต่อไปนี้

##### 4.5.1 ผลการทดสอบบนเครื่องเราเตอร์

ผลการทดสอบปรากฏว่าเมื่อมีการกำหนดค่าต่างๆบนเราเตอร์ เมื่อ telnet เข้าไปที่เราเตอร์ จากนั้นทำการสั่งให้เราเตอร์สำเนาข้อมูลการกำหนดค่าต่างๆมายังเครื่องยูนิคซ์ ปรากฏว่าสามารถถ่ายโอนข้อมูลมาได้ และในการทดสอบการถ่ายโอนข้อมูลการกำหนดค่าจากเครื่องยูนิคซ์ไปยังเราเตอร์ก็สามารถทำได้

จากผลการทดสอบข้อมูลของการปรับแต่งที่อยู่บนเราเตอร์ จะดูได้จากการตั้งคำสั่ง show config ซึ่งค่าต่างๆจะถูกแสดงออกมามีดังรูปที่ 4.15

```

router# show config
!
service password-encryption
service udp-small-servers
service tcp-small-servers
!
aaa new-model
aaa authentication login default radius
aaa authentication login admin.local
aaa authentication login test radius local
aaa authentication ppp default if-needed radius
aaa authentication ppp user radius local
aaa authorization exec if-authenticated radius local
aaa authorization network radius local
aaa accounting exec start-stop radius
aaa accounting network start-stop radius
!
ip rcmd remote-username mahuttha
ip domain-name net.kmitl.ac.th
ip name-server 161.246.10.21
ip name-server 161.246.10.22
ipx routing 0000.0c91.80b8
clock timezone mst 7
!
router rip
 network 161.246.0.0
!
router igrp 8
 network 161.246.0.0
!
no ip classless
ip route 0.0.0.0 0.0.0.0 161.246.51.1
ip route 161.246.0.0 255.255.0.0 161.246.51.1
!
hostname CAS5
!

```

#### รูปที่ 4.15 การแสดงการกำหนดค่าของเราเตอร์บนเครื่องเราเตอร์

จากที่ได้ดูค่าต่างๆบนเราเตอร์แล้วเมื่อส่งข้อมูลมายังเครื่องยูนิกซ์ก็จะได้เป็นข้อมูลตัวอักษร (text file) ซึ่งข้อมูลที่สำคัญได้นั้นเป็นค่าที่ตรงกันระหว่างเราเตอร์กับเครื่องยูนิกซ์ ค่าที่ได้บนเครื่องยูนิกซ์สามารถดูได้โดยใช้คำสั่ง `cat <filename>` ซึ่งเพิ่มข้อมูลที่ต้องการในที่นี้เป็นชื่อเพิ่มของเราเตอร์ที่สำเนา มา จากตัวอย่างนี้เพิ่มข้อมูลที่สำคัญมาชื่อ `cas6-config` จะได้ดังรูปที่ 4.16

```

hostname# cat cas6-config
!
service password-encryption
service udp-small-servers
service tcp-small-servers
!
aaa new-model
aaa authentication login default radius
aaa authentication login admin local
aaa authentication login test radius local
aaa authentication ppp default if-needed radius
aaa authentication ppp user radius local
aaa authorization exec if-authenticated radius local
aaa authorization network radius local
aaa accounting exec start-stop radius
aaa accounting network start-stop radius
!
ip rcmd remote-username mahuttha
ip domain-name net.kmitl.ac.th
ip name-server 161.246.10.21
ip name-server 161.246.10.22
ipx routing 0000.0c91.80b8
clock timezone mst 7
!
router rip
 network 161.246.0.0
!
router igrp 8
 network 161.246.0.0
!
no ip classless
ip route 0.0.0.0 0.0.0.0 161.246.51.1
ip route 161.246.0.0 255.255.0.0 161.246.51.1
!
hostname CAS5
!

```

รูปที่ 4.16 การแสดงการกำหนดค่าของเราเตอร์บนเครื่องยูนิกซ์

#### 4.5.2 ผลการทดสอบบนเครื่องยูนิคซ์

ผลการทดสอบจะแบ่งออกเป็นส่วนๆตามที่ได้มีการทดสอบในหัวข้อที่ 4. จะแบ่งได้ดังนี้

##### 4.5.2.1 ผลการทดสอบการส่งข้อมูลกับเราเตอร์เพียงตัวเดียว

จากการทดสอบปรากฏว่าสามารถดึงข้อมูลการกำหนดค่าของเราเตอร์มายังเครื่องยูนิคซ์ได้ และสามารถสำเนาข้อมูลการกำหนดค่าของเราเตอร์จากเครื่องยูนิคซ์ไปยังเราเตอร์ได้ และ เราเตอร์ทำงานได้ ซึ่งค่าต่างๆสามารถดูได้จากรูปที่ 24. กับ รูปที่ 25.

##### 4.5.2.2 ผลการทดสอบการส่งข้อมูลกับเราเตอร์ภายในกลุ่ม

จากการทดสอบการส่งข้อมูลกับเราเตอร์ภายในกลุ่มนั้น ซึ่งการทดสอบได้แบ่งออกเป็น 3 กลุ่มด้วยกันคือ

- การเพิ่มข้อมูลการกำหนดค่าของเราเตอร์ภายในกลุ่ม

การเพิ่มข้อมูลภายในกลุ่มจากการทดสอบเมื่อผู้ใช้ทำการเพิ่มข้อมูลเข้าไปในกลุ่มแล้ว โปรแกรมทำการเพิ่มข้อมูลการกำหนดค่าในแต่ละเราเตอร์และสามารถส่งข้อมูลการกำหนดค่านั้นไปยังกลุ่มได้อย่างถูกต้อง

จากตัวอย่างผลการทดสอบข้อมูลที่ต้องการเพิ่มเป็นดังนี้

```
ip name-server 161.246.11.3
ip name-server 161.246.52.21
```

ข้อมูลก่อนที่จะทำการเพิ่มการกำหนดค่าไปยังเราเตอร์ที่อยู่ภายในกลุ่มที่ต้องการนั้นเป็นดังรูปที่ 24. สังเกตว่าในส่วนที่เป็น ip name-server จะมีอยู่ 2 ส่วนตัวด้วยกันคือ name server ที่มี IP address เป็น 161.246.10.21 กับ 161.246.10.22 เมื่อทำการทดสอบโปรแกรมการเพิ่มข้อมูลส่วนกลางไปยังเราเตอร์ภายในกลุ่มเมื่อ โปรแกรมทำงานเสร็จแล้วสามารถตรวจสอบดูได้จากเราเตอร์โดยใช้คำสั่ง show config ปรากฏว่าข้อมูลที่เพิ่มเข้าไปนั้นเข้าไปอยู่ที่เราเตอร์ทุกๆตัวซึ่งเป็นดังรูปที่

26.

```

router# show config
!
service password-encryption
service udp-small-servers
service tcp-small-servers
!
aaa new-model
aaa authentication login default radius
aaa authentication login admin local
aaa authentication login test radius local
aaa authentication ppp default if-needed radius
aaa authentication ppp user radius local
aaa authorization exec if-authenticated radius
aaa authorization network radius local
aaa accounting exec start-stop radius
aaa accounting network start-stop radius
!
ip rcmd remote-username mahuttha
ip domain-name net.kmitl.ac.th
ip name-server 161.246.10.21
ip name-server 161.246.10.22
ip name-server 161.246.11.3
ip name-server 161.246.52.21
ipx routing 0000.0c91.80b8
clock timezone mst 7
!
router rip
 network 161.246.0.0
!
router igrp 8
 network 161.246.0.0
!
no ip classless
ip route 0.0.0.0 0.0.0.0 161.246.51.1
ip route 161.246.0.0 255.255.0.0 161.246.51.1
!
hostname CAS5
!

```

#### รูปที่ 4.17 คำการปรับแต่งข้อมูลของเราเตอร์หลังจากที่เพิ่มข้อมูลส่วนกลาง

จะเห็นได้ว่าข้อมูลส่วนกลางที่เพิ่มเข้าไปนั้นจะเข้าไปอยู่ในเราเตอร์ทุกๆตัวที่เป็นสมาชิกภายในกลุ่มที่ต้องการเพิ่ม

- การลบข้อมูลการกำหนดค่าของเราเตอร์ภายในกลุ่ม

ผลการทดสอบการลบข้อมูลการกำหนดค่าของเราเตอร์ภายในกลุ่มปรากฏว่าข้อมูลที่ต้องการลบจะถูกโปรแกรมลบในเราเตอร์ทุกๆตัวที่อยู่ในกลุ่มได้อย่างถูกต้อง ถ้าในกรณีที่ข้อมูลที่ต้องการจะลบเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

aaa authentication login test radius local
aaa authentication ppp default if-needed radius
aaa authentication ppp user radius local
aaa authorization exec if-authenticated radius local
aaa authorization network radius local
aaa accounting exec start-stop radius
aaa accounting network start-stop radius

```

ในส่วนของคุณสมบัติก่อนการลบจะเป็นดังรูปที่ 4.16 เมื่อโปรแกรมทำการลบข้อมูลการปรับแต่งค่าของเราเตอร์เรียบร้อยแล้วข้อมูลการปรับแต่งค่าของเราเตอร์ที่ถูกลบแล้วจะเป็นดังรูปที่ 4.18

```

router# show config
!
service password-encryption
service udp-small-servers
service tcp-small-servers
!
aaa new-model
aaa authentication login default radius
aaa authentication login admin local
!
ip rcmd remote-username mahuttha
ip domain-name net.kmitl.ac.th
ip name-server 161.246.10.21
ip name-server 161.246.10.22
ip name-server 161.246.11.3
ip name-server 161.246.52.21
ipx routing 0000.0c91.80b8
clock timezone mst 7
!
router rip
 network 161.246.0.0
!
router igrp 8
 network 161.246.0.0
!
no ip classless
ip route 0.0.0.0 0.0.0.0 161.246.51.1
ip route 161.246.0.0 255.255.0.0 161.246.51.1
!
hostname CAS5

```

รูปที่ 4.18 ค่าการปรับแต่งข้อมูลของเราเตอร์หลังจากที่ลบข้อมูลส่วนกลาง

จะสังเกตเห็นว่าข้อมูลส่วนกลางที่ต้องการลบสามารถทำการลบเรียบร้อยแล้ว ซึ่ง จะไม่มีในส่วนที่เป็น aaa authentication ที่เป็น radius เนื่องจากทำการลบไปเรียบร้อยแล้ว

#### 4.5.2.3 ผลการทดสอบการจัดการกับกลุ่มข้อมูลกลุ่มของเราเตอร์

จากการทดสอบการจัดการกับกลุ่มของเราเตอร์จะแบ่งผลของการทดสอบออกเป็น 3 ส่วนด้วยกันคือ

##### - การเพิ่มข้อมูลกลุ่มของเราเตอร์

การเพิ่มกลุ่มของเราเตอร์เมื่อผู้ใช้กรอกข้อมูลของเราเตอร์ที่อยู่ภายในกลุ่มทั้งหมดและครบ ตามจำนวนของเราเตอร์ที่อยู่ภายในกลุ่ม ปรากฏว่าข้อมูลได้ถูกเก็บอยู่ในรูปแบบที่สมบูรณ์ คือได้มีการเข้ารหัสข้อมูลได้อย่างถูกต้องสามารถทำการถอดรหัสข้อมูลได้ออกมาเป็นค่าต่างๆของเราเตอร์ได้

ตัวอย่างผลการทดสอบ ข้อมูลของเราเตอร์ที่อยู่ภายในกลุ่มจะถูกเข้ารหัสไว้เมื่อทำการเพิ่มข้อมูลกลุ่มของเราเตอร์ ซึ่งข้อมูลของเราเตอร์ที่เก็บอยู่บนเครื่องยูนิกซ์นั้น สามารถแสดงโดยใช้คำสั่ง `cat <filename>` แต่จะไม่สามารถอ่านได้เนื่องจากข้อมูลได้ถูกเข้ารหัสเอาไว้ ข้อมูลที่ถูกเข้ารหัสจะเป็นดังรูปที่ 4.19

```
Hostname# cat it
Al;dks3ldkfja;alseuriuujl;kajoirupew]fsdlfjq';afjkogjoeij;
oqwjfl;ajfopieqjf;lsdjfqpowlfjaiejqf
```

รูปที่ 4.19 เพิ่มข้อมูลที่ถูกเข้ารหัส

ข้อมูลที่ถูกเข้ารหัสเมื่อทำการถอดรหัสข้อมูลที่เก็บอยู่บนเครื่องจะเก็บอยู่ในลักษณะดังรูปที่ 4.20

```
Hostname# cat it.decrypt
groupname: crsc
car10:net:km23jq1:wd12lkqk
car11:net:alld1lju;;liad19ja
car12:net:aljau11:kam221j
car13:net:qei231o:Land009
```

รูปที่ 4.20 ข้อมูลของเราเตอร์ที่ถอดรหัสเรียบร้อยแล้วจากรูปที่ 28.

เมื่อโปรแกรมทำการเพิ่มข้อมูลของเราเตอร์เช่น โปรแกรมของเราเตอร์ทำการเพิ่มข้อมูลของเราเตอร์ car14 เข้าไป เมื่อโปรแกรมทำงานเสร็จข้อมูลที่อยู่บนเครื่องยูนิกซ์จะเป็นดังรูป 4.21

```
Hostname# cat it
Al;dks3ldkfja;alseuriuujl;kajoirupew]fsdlfjq';afjkoei
qj490oijlksajdfflqjoeij;oqwjfl;ajfopieqjf;lsdjfqpowl
fjaiejqffqjpweirjewuf;
```

รูปที่ 4.21 เพิ่มข้อมูลที่ถูกรหัสหลังจากที่มีการเพิ่มเราเตอร์ car14

เมื่อถอดรหัสข้อมูลของเพิ่มข้อมูลที่ทำการเพิ่มเราเตอร์ car14 จะเป็นดังรูปที่ 4.22

```
Hostname# cat it.decrypt
groupname: crsc
car10:net:km23jq1:wd121kqk
car11:net:alld11ju;;liad19ja
car12:net:aljau11:kam221j
car13:net:qei23lo:Land009
car14:net:ladjfoew:ldfjal
```

รูปที่ 4.22 ข้อมูลของเราเตอร์ที่ถอดรหัสเรียบร้อยแล้วหลังจากที่มีการเพิ่ม car14

ผลการทดสอบปรากฏว่าข้อมูลเราเตอร์ที่เพิ่มเข้าไปถูกเข้ารหัสถูกต้องและเมื่อทำการถอดรหัสข้อมูลออกมาปรากฏว่าข้อมูลที่เพิ่มเข้าไปถูกต้อง

- การลบข้อมูลกลุ่มของเราเตอร์

การลบข้อมูลกลุ่มของเราเตอร์ ผลปรากฏว่าสามารถลบข้อมูลของกลุ่มได้อย่างถูกต้องเมื่อโปรแกรมตรวจสอบได้ว่ารหัสลับของกลุ่มนั้นถูกต้องก็สามารถลบข้อมูลของกลุ่มได้อย่างถูกต้อง

- การจัดการกับข้อมูลภายในกลุ่มของเราเตอร์

การจัดการกับข้อมูลภายในกลุ่มของเราเตอร์ ผลการทดสอบจะแบ่งออกเป็น 3 ส่วนด้วยกันคือ การเพิ่มเราเตอร์ในกลุ่มที่มีอยู่แล้ว, การลบเราเตอร์ภายในกลุ่ม และการแก้ไขข้อมูลของเรา

เตอร์ที่อยู่ภายในกลุ่ม โดยที่การเพิ่มเราเตอร์ที่อยู่ภายในกลุ่มนั้นจะต้องตรวจสอบรหัสลับของกลุ่ม ซึ่งสามารถทำการตรวจสอบรหัสลับได้อย่างถูกต้อง และสามารถเพิ่มข้อมูลของเราเตอร์ได้อย่างถูกต้องและเข้ารหัสข้อมูลได้

การลบข้อมูลของเราเตอร์ที่อยู่ในกลุ่มนั้นสามารถลบข้อมูลของเราเตอร์ได้อย่างถูกต้องและเมื่อลบข้อมูลของเราเตอร์แล้วก็ทำการเข้ารหัสได้อย่างถูกต้อง

การจัดการกับข้อมูลของเราเตอร์ ซึ่งในกรณีนี้ผลการทดสอบก็จะทดสอบโดยการเปลี่ยนแปลงแก้ไขข้อมูลบางส่วนของเราเตอร์ที่อยู่ภายในกลุ่ม ซึ่งก็สามารถแก้ไขข้อมูลของเราเตอร์ภายในกลุ่มได้และเมื่อแก้ไขเรียบร้อยแล้วข้อมูลของเราเตอร์จะถูกเข้ารหัสได้อย่างถูกต้อง



## บทที่ 5

### สรุปและข้อเสนอแนะ

ในการพัฒนาโปรแกรมการกระจายข้อมูลการปรับแต่งค่าของเราเตอร์นั้นเป็นการช่วยในเรื่องของการจัดการเกี่ยวกับข้อมูลการปรับแต่งที่เป็นส่วนกลางที่เราเตอร์สามารถใช้ร่วมกันได้ และยังช่วยในเรื่องของการจัดการกับข้อมูลของเราเตอร์ในด้านต่างๆ ไม่ว่าจะเป็นในส่วนของการปรับแต่งค่าของเราเตอร์ หรือจะเป็นข้อมูลในส่วนของการความปลอดภัยเกี่ยวกับการเก็บรหัสผ่านและรหัสลับ เป็นต้น

#### 5.1 การเปรียบเทียบวิธีการในการ Configure แบบเดิมกับระบบใหม่

การทำงานในระบบเดิมนั้นเป็นการทำงานแบบโต้ตอบ (interactive) กับผู้ใช้โดยตรง ซึ่งในกรณีของระบบงานหรือโปรแกรมที่สร้างขึ้นใหม่เป็นระบบงานที่ทำงานเป็นแบบ off-line คือจะทำงานต่างๆ ให้เสร็จบนเครื่องยูนิกซ์ จากนั้นจึงทำการส่งข้อมูลที่ต้องการไปยังเราเตอร์ ซึ่งวิธีการนี้เราสามารถตรวจสอบความถูกต้องก่อนที่จะส่งข้อมูลไปยังเราเตอร์ได้ และยังมีข้อมูลการปรับแต่งค่าของเราเตอร์เก็บไว้สำรองในกรณีที่ต้องการใช้ค่าการปรับแต่งเราเตอร์ค่าเดิมก็สามารถทำได้ และสะดวกในการที่จะส่งข้อมูลการปรับแต่งค่าต่างๆ ที่เหมือนกัน ไปยังเราเตอร์ทุกๆ ตัว ซึ่งในกรณีการทำงานในลักษณะเดิมนั้นจะต้องทำการกำหนดค่าต่างๆ ที่เหมือนกันทุกๆ ตัว ทำให้เสียเวลา และอาจเกิดข้อผิดพลาดขึ้นได้ง่ายด้วย

#### 5.2 ปัญหาและอุปสรรคที่เกิดขึ้น

ในการพัฒนาโปรแกรมนี้นั้น เนื่องจากการพัฒนาโปรแกรมเขียนโดยใช้เซตสคริปต์ ซึ่งคอนไคราะห์และออกแบบระบบคิดว่าเป็นภาษาที่เข้าใจง่าย และร่วมกับคำสั่ง TCL ก็สามารถทำงานได้ครอบคลุมทั้งหมด แต่ปัญหาที่เกิดขึ้น เนื่องจากในบางฟังก์ชันมีการทำงานซับซ้อน และไม่มีคำสั่งที่ใช้ในการทำงานโดยตรงจึงต้องมีการเขียน โปรแกรมที่ยาวขึ้น เพื่อให้การทำงานมีความถูกต้องและสมบูรณ์มากยิ่งขึ้น

นอกจากนี้ในส่วนของการกำหนดค่าของเราเตอร์นั้นเนื่องจากการใช้งานกลไกการทำงานของ RCP ในกรณีที่ต้องการเพิ่มข้อมูลส่วนกลางสามารถทำการเพิ่มข้อมูลส่วนกลางแล้วทำการส่งข้อมูลการปรับแต่งค่าทั้งหมดไปยังเราเตอร์ได้โดยไม่ต้องทำการรีสตาร์ทเครื่องใหม่ แต่ถ้าในกรณีที่เป็นการลบข้อมูลส่วนกลางจะต้องทำการรีสตาร์ทเราเตอร์ใหม่ ซึ่งในการใช้งานจริงจะไม่เหมาะ ดังนั้นโปรแกรมในส่วนของการลบข้อมูลส่วนกลางจึงไม่ได้ใช้การทำงานในลักษณะของ RCP แต่จะใช้การทำงานเหมือนกับการส่งงานเราเตอร์แบบ online โดยการส่ง no แล้วตามด้วยคำสั่งต่างๆที่ต้องการจะลบไปยังเราเตอร์ที่ต้องการ

### 5.3 ข้อเสนอแนะ

ในการพัฒนาโปรแกรมครั้งนี้เป็นการพัฒนาในส่วนของการกระจายข้อมูลส่วนกลางไปยังสมาชิกของเราเตอร์ภายในกลุ่ม และการจัดการกับกลุ่มของเราเตอร์ ซึ่งในกรณีที่ต้องการจะศึกษาต่อสามารถทำได้ในหลายลักษณะดังนี้

1. การพัฒนาครั้งนี้เป็นการพัฒนาให้เหมาะกับผู้ใช้ โดยผู้ที่จะใช้โปรแกรมนี้ น่าจะเป็นผู้ที่ดูแลระบบ ซึ่งมีความรู้เกี่ยวกับเราเตอร์พอสมควรจึงได้มีการพัฒนาในลักษณะตัวอักษรทั้งหมด ซึ่งต่อไปอาจจะมีการพัฒนาในลักษณะของกราฟฟิก ซึ่งอาจจะทำให้ผู้ใช้ไม่จำเป็นต้องมีความรู้เกี่ยวกับเราเตอร์ได้
2. ในเรื่องของความปลอดภัยของข้อมูลรหัสผ่านและรหัสลับนั้น การส่งข้อมูลยังส่งไปในลักษณะของ plain text ซึ่งอาจจะมีการดักจับข้อมูลรหัสผ่านและรหัสลับได้ ดังนั้นโปรแกรมควรที่จะตั้งอยู่ในระบบเครือข่ายที่มีความปลอดภัยสูง ซึ่งไม่มีใครสามารถที่จะดักจับข้อมูลที่ส่งไปใช้ได้
3. ในการส่งข้อมูลควรจะมีการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะส่งข้อมูลไปยังเราเตอร์ เนื่องจากถ้ามีการส่งข้อมูลที่ผิดพลาดเข้าไปอาจทำให้การจัดตั้งค่าผิดพลาดได้ ซึ่งในการพัฒนาครั้งต่อไปอาจมีการตรวจสอบความถูกต้องก่อนการส่งข้อมูลไปยังเราเตอร์

## บรรณานุกรม

Cisco system, 1996.**Cisco Connection Documentation** [Online] .Available

URL:<http://www.cisco.com/univercd/home/home.html>

Harrison,Mark and other contributor.1997.**TCL/TK Tools**, o'Reilly & Associates,Inc.

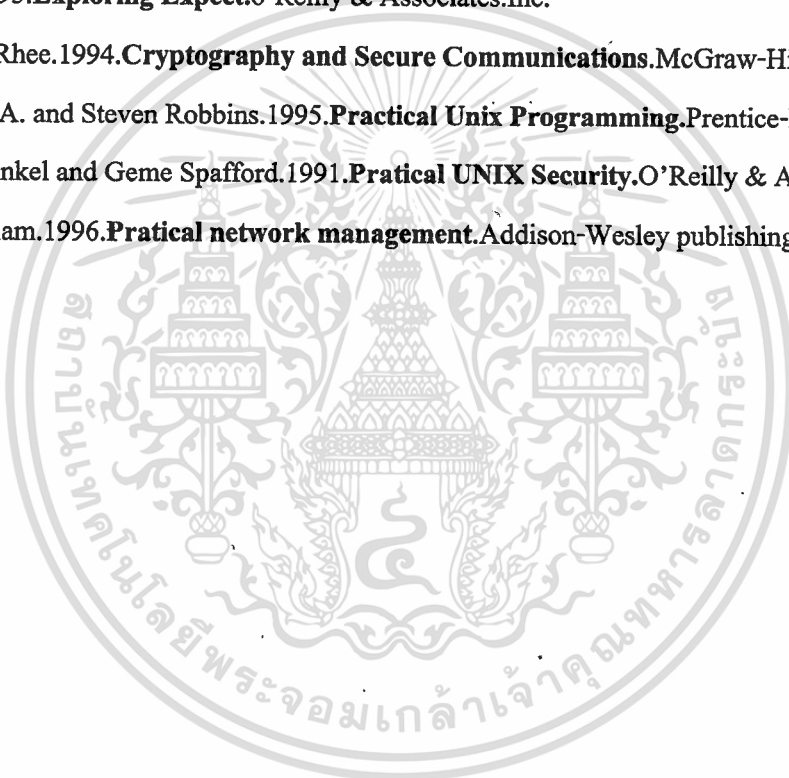
Libés,Don.1995.**Exploring Expect**.o'Reilly & Associates.Inc.

Man ,Young Rhee.1994.**Cryptography and Secure Communications**.McGraw-Hill Book Co.

Robbins,Kay A. and Steven Robbins.1995.**Practical Unix Programming**.Prentice-Hall,Inc

Simson, Garfinkel and Geme Spafford.1991.**Practical UNIX Security**.O'Reilly & Associates,Inc.

Stalling, William.1996.**Practical network management**.Addison-Wesley publishing company.



## ภาคผนวก ก

### คู่มือการใช้งานระบบ

เนื่องจากในระบบการกระจายการปรับแต่งค่าของข้อมูลนั้นส่วนที่สำคัญจะประกอบไปด้วย 2 ส่วนด้วยกันคือในส่วนของเครื่องที่จะนำมาลงระบบปฏิบัติการยูนิกซ์ และส่วนของซิสโคโนเตอร์ ซึ่งทั้ง 2 ส่วนนี้จะต้องสามารถทำงานด้วยกันได้

ส่วนของโปรแกรมจะสามารถรันได้บนเครื่องที่ใช้งานยูนิกซ์ทั่วไป ซึ่งเครื่องที่ใช้งานยูนิกซ์ อาจใช้เครื่อง PC ที่มีคุณลักษณะดังนี้

1. CPU Pentium
2. Harddisk 1 GB ขึ้นไป
3. RAM 64 MB
4. OS ที่เป็นยูนิกซ์ที่สามารถรันบนเครื่อง PC เช่น linux , FreeBSD ,OpenBSD
5. โปรแกรมแพ็คเกจ TCL/TK

ส่วนของซิสโคโนเตอร์ที่สามารถใช้งานได้นั้นจะต้องเป็นซิสโคโน IOS version 11.3 เป็นอย่างต่ำ เนื่องจากถ้าเป็น version ที่ต่ำกว่า 11.3 นั้นจะไม่สามารถใช้ feature ในส่วนของ RCP ได้

เนื่องจากในระบบยูนิกซ์นั้นการทำงานจะต้องมีการ login เพื่อตรวจสอบว่าผู้ใช้นั้นมีสิทธิในการใช้ข้อมูลหรือไม่ นอกจากนี้จะทำการตรวจสอบว่าสามารถใช้ข้อมูลได้หรือไม่แล้วยังตรวจสอบในเรื่องของการเข้าถึงข้อมูลว่าผู้ใช้คนใดสามารถเข้าถึงข้อมูลในส่วนใดได้บ้าง

#### ก.1 การเข้าใช้งานระบบ

ในการใช้งาน โปรแกรมนี้ผู้ใช้จะต้องทำการ telnet ไปยังเครื่องยูนิกซ์ที่โปรแกรมนี้รันอยู่ ซึ่งในที่นี้รันอยู่บนเครื่อง akajunior.it.kmitl.ac.th เมื่อผู้ใช้ telnet เข้าไปยังเครื่องนี้แล้วให้ใส่รหัสผู้ใช้และรหัสลับของผู้ใช้ที่มีสิทธิในการเข้าใช้งาน โปรแกรมได้ เมื่อเข้าไปได้แล้วให้เข้าไปที่ไคเรททอรี /home/mahuttha/cisco จากนั้นสั่ง ./main จะปรากฏดังรูป ก.1

**Main Menu of Distributed config Router**

- 1). **Select one Router**
- 2). **Select group of Router**
- 3). **Manage group of Router**
- x). **exit of program**

Please enter your choice => \_\_\_\_\_

**รูปที่ ก.1 หน้าจอโปรแกรมหลัก**

จากนั้นให้เลือกหัวข้อตามต้องการในกรณีที่ต้องการจัดการเกี่ยวกับเราเตอร์เพียงตัวเดียวก็สามารถเลือกในหัวข้อที่ 1. ซึ่งจะปรากฏดังรูป ก.2

**Do you want to edit or update config**

- 1). **Edit one router**
- 2). **Update one router**
- x). **Back to main page**

Please enter your choice => \_\_\_\_\_

**รูปที่ ก.2 หน้าจอโปรแกรมการทำงานกับเราเตอร์ 1 ตัว**

จากนั้นจะปรากฏเมนูย่อยขึ้นมา ให้ผู้ใช้เลือกว่าต้องการที่จะทำการแก้ไขค่าการปรับแต่งของเราเตอร์หรือต้องการเปลี่ยนแปลงข้อมูลของเราเตอร์ ในกรณีที่ต้องการแก้ไขข้อมูลของเราเตอร์นั้นให้เลือกในหัวข้อที่ 1. เมื่อเลือกแล้วจะปรากฏดังรูป ก.3

What router do you want to edit configuration	_____
Please fill your username	_____
Please fill your password of router	_____
Please fill your enable password of router	_____
Please fill your host that you want to edit configuration	_____

### รูปที่ ก.3 หน้าจอโปรแกรมการใส่ข้อมูลการจัดการแก้ไขข้อมูลเราเตอร์ 1 ตัว

เมื่อใส่ข้อมูลเรียบร้อยแล้วโปรแกรมจะทำการติดต่อไปยังเราเตอร์จากนั้นเราเตอร์จะส่งข้อมูลการปรับแต่งค่ามายังเครื่องฮุคซึ่งเพิ่มข้อมูลที่เราเตอร์ส่งมานั้นจะอยู่ในชื่อของ [routername]-config ซึ่งอยู่ในห้องของผู้ใช้ที่ได้มีการกำหนดไว้ในเราเตอร์ ซึ่งในที่นี้จะอยู่ในไดเรกทอรี /home/mahuttha จากนั้นเมื่อผู้ใช้ทำการแก้ไขข้อมูลการปรับแต่งค่าของเราเตอร์เรียบร้อยแล้วถ้าต้องการเปลี่ยนแปลงข้อมูลการปรับแต่งค่าของเราเตอร์ ก็สามารถส่งข้อมูลไปยังเราเตอร์ได้โดยเลือกในส่วนของการ update configuration ซึ่งเมื่อเลือกหัวข้อนี้แล้วจะปรากฏดังรูปที่ ก.4

What router do you want to update configuration	_____
Please fill your username	_____
Please fill your password of router	_____
Please fill your enable password of router	_____
Please fill your host that you want to send configuration	_____

### รูปที่ ก.4 หน้าจอโปรแกรมการใส่ข้อมูลการจัดการเปลี่ยนแปลงข้อมูลเราเตอร์ 1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น, อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อใส่ข้อมูลของเราเตอร์เรียบร้อยแล้วโปรแกรมก็จะทำการส่งข้อมูลของเราเตอร์จากเครื่องยูนิคซ์ไปยังเราเตอร์

ในกรณีที่ต้องการส่งข้อมูลส่วนกลางไปยังกลุ่มของเราเตอร์นั้นจะต้องมีการสร้างกลุ่มของเราเตอร์ขึ้นมาก่อน ซึ่งกลุ่มของเราเตอร์นั้นสามารถสร้างได้โดยเลือกเมนูในส่วนของ การเพิ่มกลุ่มของเราเตอร์ดังรูปที่ ก.5

**Do you want to add or delete or modify group of router**

- 1). Add group of router
- 2). Delete group of router
- 3). Modify group of router
- x). Back to main page

**Please enter your choice => \_\_\_\_\_**

รูปที่ ก.5 หน้าจอโปรแกรมหลักในการจัดการกับกลุ่มของเราเตอร์

ในส่วนของ การเพิ่มกลุ่มของข้อมูลนั้นเมื่อเลือกในส่วนที่ 1. การเพิ่มกลุ่มของเราเตอร์นั้นจะปรากฏหน้าจอ ดังรูปที่ ก.6

**Please fill groupname that you want to add** \_\_\_\_\_

**How many router do you want to add** \_\_\_\_\_

**Please fill password of router group** \_\_\_\_\_

รูปที่ ก.6 หน้าจอโปรแกรมการใส่ข้อมูลกับกลุ่มของเราเตอร์ที่ต้องการเพิ่ม

จากนั้น โปรแกรมจะให้ใส่ข้อมูลของเราเตอร์ในแต่ละตัว ดังรูปที่ ก.7

**Information of Router NO. 1**

**Please fill your routername** \_\_\_\_\_

**Please fill your username** \_\_\_\_\_

**Please fill your password** \_\_\_\_\_

**Please fill your enable password** \_\_\_\_\_

**รูปที่ ก.7 หน้าจอการใส่ข้อมูลของเราเตอร์**

เมื่อใส่ข้อมูลของเราเตอร์เรียบร้อยแล้วข้อมูลของเราเตอร์ก็จะถูกเข้ารหัสแล้วเก็บในแฟ้มข้อมูลของกลุ่มและกลับไปยังหน้าจอหลัก เมื่อทำการเพิ่มกลุ่มของข้อมูลเรียบร้อยแล้วก็สามารถทำการกระจายข้อมูลส่วนกลางไปยังกลุ่มของเราเตอร์ได้โดยเลือกในหัวข้อที่ 2.เลือกกลุ่มของเราเตอร์จะปรากฏดังรูปที่ ก.8

**Do you want to add or delete config in group of router**

- 1). **Add config in group of router**
- 2). **Delete config in group of router**
- x). **Back to main page**

**Please enter your choice => \_\_\_\_\_**

**รูปที่ ก.8 หน้าจอโปรแกรมหลักในการกระจายข้อมูลไปยังกลุ่มของเราเตอร์**

ในกรณีที่เรากำลังต้องการที่จะเพิ่มข้อมูลส่วนกลางกระจายไปยังกลุ่มของเราเตอร์ให้เลือกหัวข้อที่ 1. ซึ่งในการเพิ่มหรือลบข้อมูลส่วนกลางจะต้องทำการใส่ข้อมูลที่ต้องการจะเพิ่มหรือลบเก็บอยู่ในเพิ่มข้อมูล [groupname].added หรือ [groupname].deleted ก่อนซึ่งการสร้างทั้ง 2 แฟ้มนี้ให้เก็บที่ไคลเรททอรีเดียวกับโปรแกรม ซึ่งในที่นี้เก็บที่ /home/mahuttha/cisco เมื่อสร้างแฟ้มข้อมูลเรียบร้อยแล้วจากนั้นเลือกหัวข้อตามต้องการ เมื่อเลือกเพิ่มข้อมูลแล้วจะปรากฏดังรูปที่ ก.9

```
#####
What group do you want to add global config _____
Please fill password of group _____
Do you have file that you want to add ALREADY(y/n) _____
#####
```

### รูปที่ ก.9 หน้าจอโปรแกรมการใส่ข้อมูลของกลุ่มที่ต้องการจะเพิ่มข้อมูลการปรับแต่งของเราเตอร์

เมื่อใส่ข้อมูลของกลุ่มเรียบร้อยแล้ว โปรแกรมจะทำการเพิ่มข้อมูลส่วนกลางไปยังเราเตอร์ที่เป็นสมาชิกภายในกลุ่มที่เราต้องการ เมื่อโปรแกรมทำงานเสร็จจะกลับไปหน้าจอหลัก

ในกรณีของการลบข้อมูลส่วนกลางของกลุ่มก็จะคล้ายกับการเพิ่มข้อมูลส่วนกลางของกลุ่ม ถ้าผู้ใช้เลือกหัวข้อการลบข้อมูลส่วนกลางแล้วใส่ข้อมูลของกลุ่มของเราเตอร์เรียบร้อยแล้ว โปรแกรมก็จะทำการลบข้อมูลส่วนกลาง เมื่อโปรแกรมทำงานเสร็จก็จะกลับไปหน้าจอหลัก

## ภาคผนวก ข

### วิธีการ ติดตั้งโปรแกรม และ ตัวอย่าง source code

ขั้นตอนการลงโปรแกรมนั้น เมื่อได้เพิ่มข้อมูลโปรแกรมที่ชื่อ router.tar แล้วนั้นให้ทำการแตกเพิ่มข้อมูลโปรแกรมลงในไคลเรกทอรีตามที่ต้องการโดยใช้คำสั่ง

```
hostname> tar -xvf router.tar
```

เมื่อแตกเพิ่มข้อมูลแล้วจะได้เพิ่มข้อมูลทั้งหมดอยู่ในไคลเรกทอรีที่สั่งคำสั่ง tar เมื่อได้เพิ่มข้อมูลที่ต้องการแล้วให้เข้าไปที่โปรแกรมดังต่อไปนี้

1. main
2. add\_global
3. del\_global
4. copyh-r.expect
5. copyr-h.expect
6. distribute.expect
7. delete.expect

จากนั้นเข้าไปแก้ไขในส่วนของค่าตัวแปรที่เกี่ยวข้องกับ path ซึ่งตัวแปรที่จะต้องแก้ไขเกี่ยวข้องกับ path จะมีดังนี้

1. ตัวแปร HOME
2. ตัวแปร PARENT
3. ตัวแปร GROUP

โดยที่ตอนเริ่มต้นที่โปรแกรมถูกแตกออกมานั้น ค่าตัวแปรต่างจะอยู่ที่ /home/mahuttha ซึ่งในกรณีที่ต้องการเปลี่ยนแปลงไคลเรกทอรีของตัวแปร HOME นั้นให้แก้ path ไปอยู่ที่ home directory ของผู้ใช้คนนั้น ส่วนตัวแปร PARENT นั้นจะต้องแก้ไขเป็น /home/[username]/cisco จากนั้นในส่วนของตัวแปร GROUP นั้นจะต้องแก้ไขเป็น /home/[username]/cisco/group ซึ่งในส่วนของผู้ใช้นั้นจะต้องเป็นผู้ใช้ที่ถูกต้องตรงกับที่มีการกำหนดค่าของผู้ใช้ในซิสโตนีเตอร์ด้วยการกำหนดชื่อผู้ใช้ในเรเตอร์ อธิบายในบทที่ 4. ในส่วนของการกำหนดค่าของเรเตอร์ เมื่อทำการแก้ไขเรียบร้อยแล้วสามารถรันโปรแกรมได้ตามภาคผนวก ก.

## ข.1 เพิ่มข้อมูล add\_global

Source code โปรแกรมที่ใช้ในการรับข้อมูลในการตรวจสอบการเพิ่มข้อมูลส่วนกลางในการกระจายข้อมูลไปยังกลุ่มของเราเตอร์ที่ใช้ bash shell script

```
#!/bin/bash
HOME=/home/mahuttha
PARENT=/home/mahuttha/cisco
GROUP=/home/mahuttha/cisco/group
tput clear
tput cup 2 18
echo ' What group do you want to add global config'
tput cup 5 15
echo 'Please enter your host that you will send config'
tput cup 5 66
stty erase ^H
read HOST
tput cup 7 25
echo 'Please enter your group name'
tput cup 7 56
stty erase ^H
read GROUP_NAME
tput cup 9 25
echo 'Please enter your passwd'
tput cup 9 52
stty erase ^H
stty -echo
read PASSWD
stty echo
crypt $PASSWD < $GROUP/$GROUP_NAME > $GROUP/$GROUP_NAME.decrypt
grep -q $GROUP_NAME $GROUP/$GROUP_NAME.decrypt
if [ $? -eq 0 ]
then
    echo 'your passwd is correct'
    tput cup 11 1
    echo ' Do you create config file that you want to add in group
ALREADY (y/n) '
    tput cup 11 73
    read KEY
    if [ $KEY = 'y' ] || [ $KEY = 'Y' ]
    then
        if test -s $PARENT/$GROUP_NAME.added
        then
            TOTCNT=`cat $GROUP/$GROUP_NAME.decrypt|wc -l`
            ROUND=2
            while [ $ROUND -le $TOTCNT ]
            do
                ROUTER=`head -$ROUND
$GROUP/$GROUP_NAME.decrypt |tail -1|awk '{printf $1}`
                USERNAME=`head -$ROUND
$GROUP/$GROUP_NAME.decrypt|tail -1|awk '{printf $2}`
                PASSWD=`head -$ROUND
$GROUP/$GROUP_NAME.decrypt |tail -1|awk '{printf $3}`
                ENABLE=`head -$ROUND
$GROUP/$GROUP_NAME.decrypt |tail -1|awk '{printf $4}`
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cat $PARENT/g_$ROUTER-config >
$PARENT/$ROUTER-config
cat $PARENT/$GROUP_NAME.added >>
$PARENT/$ROUTER-config
cat $PARENT/l_$ROUTER-config >>
$PARENT/$ROUTER-config
cp $PARENT/$ROUTER-config $HOME/$ROUTER-config
echo $ROUTER > $PARENT/temp
grep ^[a-z] $PARENT/temp > $PARENT/templ
if [ $? = 0 ]
then
ROUTERNAME=`grep $ROUTER
$PARENT/routername | cut -d : -f 3`
else
ROUTERNAME=$ROUTER
ROUTER=`grep $ROUTER $PARENT/routername
| cut -d : -f 1`
fi
if test -s $HOME/$ROUTER-config
then
$PARENT/distribute.expect ${ROUTERNAME} $
{ROUTER} ${USERNAME} ${PASSWD} ${ENABLE} ${HOST}
else
echo 'Not found configuration of $ROUTER
router'
fi
ROUND=`expr $ROUND + 1`
done
else
echo $GROUP_NAME.added 'not found in this
directory'
fi
else
tput clear
echo 'See instruction'
cat $PARENT/instruction_added |more
read anykey
fi
else
echo 'your passwd is incorrect'
exit 0
read anykey
fi
rm -f $GROUP/$GROUP_NAME.decrypt

```

## ข.2 เพิ่มข้อมูล distribute.expect

Source code โปรแกรมที่ใช้ในการเพิ่มค่าข้อมูลการปรับแต่งค่าต่างๆภายในกลุ่มที่ใช้ expect script

```
#!/usr/local/bin/expect -f

set timeout 1
set blnk " "
set routename [lindex $argv 0]
set router [lindex $argv 1]
set username [lindex $argv 2]
set passwd [lindex $argv 3]
set enable [lindex $argv 4]
set host [lindex $argv 5]

spawn telnet $routename
expect "Username:"
send "$username\r"
expect "Password:"
send "$passwd\r"
expect {
    "$router>" {
        send "en\r"
        expect "Password:"
        send "$enable\r"
        expect {
            "$router#" {
                send "copy rcp startup-config\r"
                expect "Address of remote host "
                send "$host\r"
                expect "Name of configuration file "
                send "\r"
                expect "Configure using "
                send "\r"
                expect -re "\n Loading"
                expect -re "\n$router"
                send "copy start run\r"
                sleep 20
                expect "$router"
                send "exit\r"
                return 0
            }
            "Password:" {
                send_user "Enable Password incorrect\n"
                return 1
            }
        }
    }
}

send_user "Password incorrect\n"
return 1
}
interact
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ข.3 เพิ่มข้อมูล del\_global

Source code โปรแกรมที่ใช้ในการรับข้อมูลในการตรวจสอบการลบข้อมูลส่วนกลางในการกระจายข้อมูลไปยังกลุ่มของเราเตอร์ ที่ใช้ bash shell script

```
#!/bin/bash

HOME=/home/mahuttha/
PARENT=/home/mahuttha/cisco
GROUP=/home/mahuttha/cisco/group
tput clear
tput cup 2 18
echo ' What group do you want to delete global config'
tput cup 5 25
echo 'Please enter your group name'
tput cup 5 60
stty erase ^H
read GROUP_NAME
tput cup 7 25
echo 'Please enter your passwd'
tput cup 7 53
stty -echo
read PASSWD
stty echo
crypt $PASSWD < $GROUP/$GROUP_NAME > $GROUP/$GROUP_NAME.decrypt
grep -q $GROUP_NAME $GROUP/$GROUP_NAME.decrypt
if [ $? -eq 0 ]
then
    echo 'your passwd is correct'
    echo ' Do you create config file that you want to delete in
group ALREADY (y/n) '
    read KEY
    if [ $KEY = 'y' ] || [ $KEY = 'Y' ]
    then
        if test -s $PARENT/$GROUP_NAME.deleted
        then
            TOTCNT=`cat $GROUP/$GROUP_NAME.decrypt|wc -l`
            ROUND=2
            while [ $ROUND -le $TOTCNT ]
            do
                ROUTER=`head -$ROUND
$GROUP/$GROUP_NAME.decrypt |tail -1|awk '{printf $1}`
                USERNAME=`head -$ROUND
$GROUP/$GROUP_NAME.decrypt|tail -1|awk '{printf $2}`
                PASSWD=`head -$ROUND
$GROUP/$GROUP_NAME.decrypt |tail -1|awk '{printf $3}`
                ENABLE=`head -$ROUND
$GROUP/$GROUP_NAME.decrypt |tail -1|awk '{printf $4}`

                DELLINE=`cat $PARENT/$GROUP_NAME.deleted|wc -
l`

                COUNT=1
                while [ $COUNT -le $DELLINE ]
                do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1`
TOTALINE=`cat $PARENT/$ROUTER-config|wc -
$PARENT/$GROUP_NAME.deleted|tail -1`
CMD=`head -$COUNT
LINE=`grep -n "$CMD" $PARENT/$ROUTER-
config|awk -F : '{printf $1}`
if [ $LINE > 0 ]
then
UP=`expr $LINE - 1`
DOWN=`expr $TOTALINE - $LINE`
head -$UP $PARENT/$ROUTER-config >
$PARENT/$ROUTER.config1
tail -$DOWN $PARENT/$ROUTER-config >>
$PARENT/$ROUTER.config1
mv $PARENT/$ROUTER.config1
$PARENT/$ROUTER-config
COUNT=`expr $COUNT + 1`
else
COUNT=`expr $COUNT + 1`
fi
done
echo $ROUTER > $PARENT/temp
grep ^[a-z] $PARENT/temp > $PARENT/temp1
if [ $? = 0 ]
then
ROUTERNAME=`grep $ROUTER
$PARENT/routername | cut -d : -f 3`
else
ROUTERNAME=$ROUTER
ROUTER=`grep $ROUTER $PARENT/routername
| cut -d : -f 1`
fi
CNT=1
while [ $CNT -le $DELLINE ]
do
if [ $CNT = 1 ]
then
CMD1=`head -1
$PARENT/$GROUP_NAME.deleted|tail -1`
fi
if [ $CNT = 2 ]
then
CMD2=`head -2
$PARENT/$GROUP_NAME.deleted|tail -1`
fi
if [ $CNT = 3 ]
then
CMD3=`head -3
$PARENT/$GROUP_NAME.deleted|tail -1`
fi
if [ $CNT = 4 ]
then
CMD4=`head -4
$PARENT/$GROUP_NAME.deleted|tail -1`

```

```

        if [ $CNT = 5 ]
        then
            CMD5=`head -5
$PARENT/$GROUP_NAME.deleted|tail -1`
            fi
            if [ $CNT = 6 ]
            then
                CMD6=`head -6
$PARENT/$GROUP_NAME.deleted|tail -1`
                fi
                if [ $CNT = 7 ]
                then
                    CMD7=`head -7
$PARENT/$GROUP_NAME.deleted|tail -1`
                    fi
                    if [ $CNT = 8 ]
                    then
                        CMD8=`head -8
$PARENT/$GROUP_NAME.deleted|tail -1`
                        fi
                        if [ $CNT = 9 ]
                        then
                            CMD9=`head -9
$PARENT/$GROUP_NAME.deleted|tail -1`
                            fi
                            if [ $CNT = 10 ]
                            then
                                CMD10=`head -10
$PARENT/$GROUP_NAME.deleted|tail -1`
                                fi
                                CNT=`expr $CNT + 1`
                            done
                            $PARENT/delete.expect ${ROUTERNAME} ${ROUTER}
                            ${USERNAME} ${PASSWD} ${ENABLE} ${DELLINE} "$CMD1" "$CMD2" "$CMD3"
                            "$CMD4" "$CMD5" "$CMD6" "$CMD7" "$CMD8" "$CMD9" "$CMD10"
                            ROUND=`expr $ROUND + 1`
                        done
                    else
                        echo $GROUP_NAME.deleted 'not found in this
directory'
                    fi
                else
                    tput clear
                    echo 'See instruction'
                    cat $PARENT/instruction_deleted |more
                    read anykey
                fi
            else
                echo 'your passwd is incorrect'
                read anykey
            fi
        rm -f $GROUP/$GROUP_NAME.decrypt

```

#### ข.4 เพิ่มข้อมูล delete.expect

Source code โปรแกรมที่ใช้ในการลบค่าข้อมูลการปรับแต่งค่าต่างๆภายในกลุ่มที่ใช้ expect script

```
#!/usr/local/bin/expect -f

set timeout 1
set blk " "
set routename [lindex $argv 0]
set router [lindex $argv 1]
set username [lindex $argv 2]
set passwd [lindex $argv 3]
set enable [lindex $argv 4]
set delline [lindex $argv 5]
set cmd1 [lindex $argv 6]
set cmd2 [lindex $argv 7]
set cmd3 [lindex $argv 8]
set cmd4 [lindex $argv 9]
set cmd5 [lindex $argv 10]
set cmd6 [lindex $argv 11]
set cmd7 [lindex $argv 12]
set cmd8 [lindex $argv 13]
set cmd9 [lindex $argv 14]
set cmd10 [lindex $argv 15]

spawn telnet $routename
expect "Username:"
send "$username\r"
expect "Password:"
send "$passwd\r"
expect {
    "$router>" {
        send "en\r"
        expect "Password:"
        send "$enable\r"
        expect {
            "$router#" {
                send "conf term\r"
                expect "$router"
                set i 1
                while {$i <= $delline} {
                    if {$i == 1} {
                        send "no $cmd1\r"
                        expect "$router" }
                    if {$i == 2} {
                        send "no $cmd2\r"
                        expect "$router" }
                    if {$i == 3} {
                        send "no $cmd3\r"
                        expect "$router" }
                    if {$i == 4} {
                        send "no $cmd4\r"
                        expect "$router" }
                    if {$i == 5} {
                        send "no $cmd5\r"
```

```

        expect "$router" }
    if {$i == 6} {
        send "no $cmd6\r"
        expect "$router" }
        if {$i == 7} {
            send "no $cmd7\r"
            expect "$router" }
    if {$i == 8} {
        send "no $cmd8\r"
        expect "$router" }
    if {$i == 9} {
        send "no $cmd9\r"
        expect "$router" }
    if {$i == 10} {
        send "no $cmd10\r"
        expect "$router" }
        incr i +1
    }
    sleep 5
    send "exit\r"
    expect "$router"
    send "exit\r"
    return 0
}
>Password:" {
    send_user "Enable Password incorrect\n"
    return 1
}
}
"% Login invalid" {
    send_user "Password incorrect\n"
    return 1
}
}

```

## ประวัติผู้เขียน

ชื่อผู้เขียน (ไทย)	นายมหัทธวัฒน์ รักษาเกียรติศักดิ์
ชื่อผู้เขียน (อังกฤษ)	Mr. Mahutthawat Raksakiettisak
วันเดือนปีเกิด	18 กันยายน 2518
สถานที่เกิด	จ.กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. (วิทยาการคอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2539
ประวัติการทำงาน	นักวิชาการคอมพิวเตอร์ 3 มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร