

บทที่ 3

การออกแบบและโครงสร้างของโปรแกรม

3.1 ภาษาและเครื่องมือที่ใช้ในการพัฒนา

3.1.1 ภาษาที่ใช้ในการพัฒนา

- 1) ภาษา C#
- 2) ภาษา ActionScript 3.0

3.1.2 เครื่องมือที่ใช้ในการพัฒนา

- 1) โปรแกรม Microsoft Visual Studio 2008
- 2) โปรแกรม Adobe Flash CS4
- 3) ระบบปฏิบัติการ Windows
- 4) .NET Framework version 3.5

3.2 อินพุตและเอาต์พุตที่เข้าและออกจากระบบ

3.2.1 ข้อมูลเข้าของระบบ (System Input)

ส่วนของการออกแบบโครงสร้างของระบบเครือข่าย

- 1) การเลือกชนิดของอุปกรณ์เครือข่ายในการวางระบบ
- 2) การวางรูปแบบโครงข่ายของระบบเครือข่าย

ส่วนของคำสั่งที่ใช้ในการกำหนดค่าการทำงาน

คำสั่งในการกำหนดค่าการทำงานของเราท์เตอร์

- 1) คำสั่งที่ใช้ในการเปลี่ยนชื่อเราท์เตอร์
- 2) คำสั่งที่ใช้ในการกำหนดค่ารหัสผ่านของเราท์เตอร์
- 3) คำสั่งที่ใช้ในการกำหนดค่ารหัสผ่านในส่วนของ

Virtual Terminal Lines ของเราท์เตอร์

- 4) คำสั่งที่ใช้กำหนดค่ารหัสผ่านในการเข้าใช้ Privileged

Exec Mode ของเราท์เตอร์

- 5) คำสั่งที่ใช้ในการกำหนดค่าให้กับอินเตอร์เฟซ

- 6) คำสั่งที่ใช้ในการกำหนดค่าการส่งต่อแพคเกจของ

เราท์เตอร์

3.2.2 ข้อมูลออกที่ได้จากระบบ (System Output)

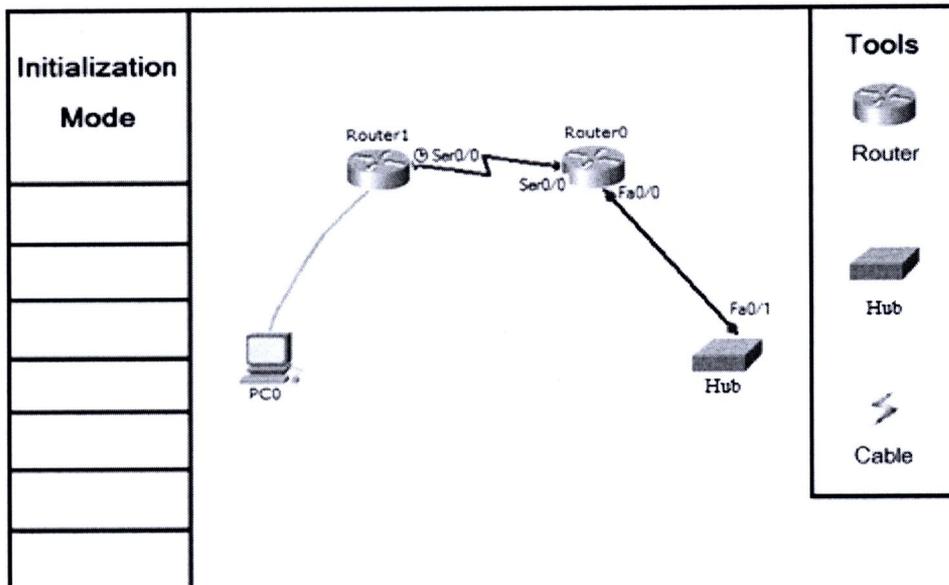
- 1) Dialog Box แสดงการทำการตั้งค่าเริ่มต้น
- 2) เราท์เตอร์แต่ละตัวมีค่าที่เกิดจากการตั้งค่าเริ่มต้น
- 3) Dialog box แสดงการตรวจสอบการทำงานของผู้ใช้งานระบบเมื่อมีสิ่งผิดปกติเกิดขึ้น
- 4) อุปกรณ์เครือข่ายทำงานตามสิ่งที่ผู้ใช้งานระบบกำหนดไว้ใน โปรแกรม

3.3 หลักการทำงานของโปรแกรม

เมื่อผู้ใช้งานระบบมีอุปกรณ์เครือข่ายที่ต้องการจะทำการตั้งค่าการทำงานซึ่งเป็นรุ่นที่ตรงตามความสามารถของโปรแกรม สามารถทำการตั้งค่าหรือแสดงค่าการทำงานโดยผ่านอุปกรณ์เครือข่ายเพียงตัวเดียวได้โดยอาศัยโปรแกรมนี้ ซึ่งมีหลักการและขั้นตอนการทำงานดังนี้

3.3.1 การออกแบบโครงสร้างของระบบเครือข่าย

เมื่อทำการเปิดใช้งาน โปรแกรมจะเข้าสู่โหมดการตั้งค่าเริ่มต้นของอุปกรณ์เครือข่าย ซึ่งโปรแกรมจะมีพื้นที่สำหรับให้ผู้ใช้งานทำการออกแบบการเชื่อมต่ออุปกรณ์เครือข่ายแต่ละตัวตามที่ต้องการ ผู้ใช้งาน โปรแกรมสามารถออกแบบระบบเครือข่ายได้โดยการเลือกวางอุปกรณ์ที่มีให้จากแถบเครื่องมือ

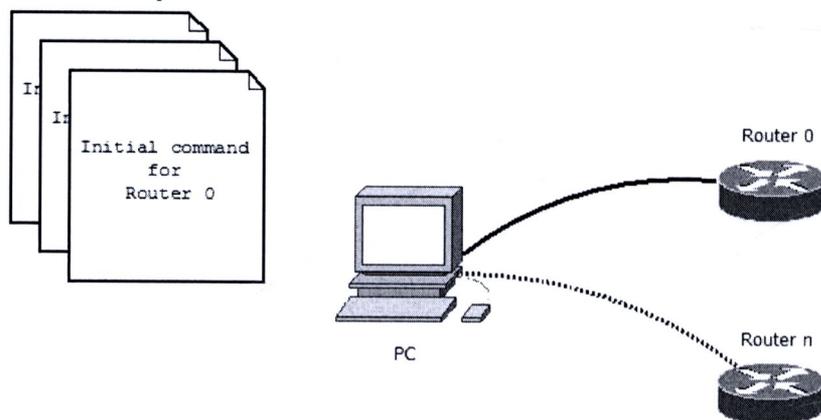


รูป 3.1 การออกแบบระบบเครือข่าย

หลังจากผู้ใช้งาน โปรแกรมทำการออกแบบเครือข่ายเรียบร้อยแล้ว ผู้ใช้งาน โปรแกรมจะต้องทำการเชื่อมต่ออุปกรณ์จริงให้มีลักษณะเหมือนกับที่ผู้ใช้งานออกแบบไว้ในระบบ ซึ่งตัวโปรแกรมจะมีส่วนที่ใช้ในการกำหนดค่าเริ่มต้นให้กับอุปกรณ์ที่ผู้ใช้งานได้ออกแบบไว้ โดยที่ผู้ใช้งานจะทำการเชื่อมต่อพอร์ตอนุกรมเข้ากับสายคอนโซล หลังจากนั้นจึงนำไปเชื่อมต่อกับอุปกรณ์ และผู้ใช้งานทำการเลือกตัวอุปกรณ์ที่จะทำการตั้งค่าเริ่มต้น ซึ่งขั้นตอนนี้ผู้ใช้งานจะต้องเชื่อมต่อตัวโปรแกรมเข้ากับอุปกรณ์จริงให้ตรงกับตัวอุปกรณ์ที่ได้ทำการเลือกไว้ในตัวโปรแกรม หลังจากทำการตั้งค่าเริ่มต้นให้กับอุปกรณ์เรียบร้อยแล้ว โปรแกรมจะทำการตรวจสอบการเชื่อมต่อของอุปกรณ์ต่างๆ ซึ่งถ้าเกิดความผิดพลาดในการตั้งค่าเริ่มต้นกับอุปกรณ์จริง

3.3.2 การตั้งค่าเริ่มต้นให้กับอุปกรณ์

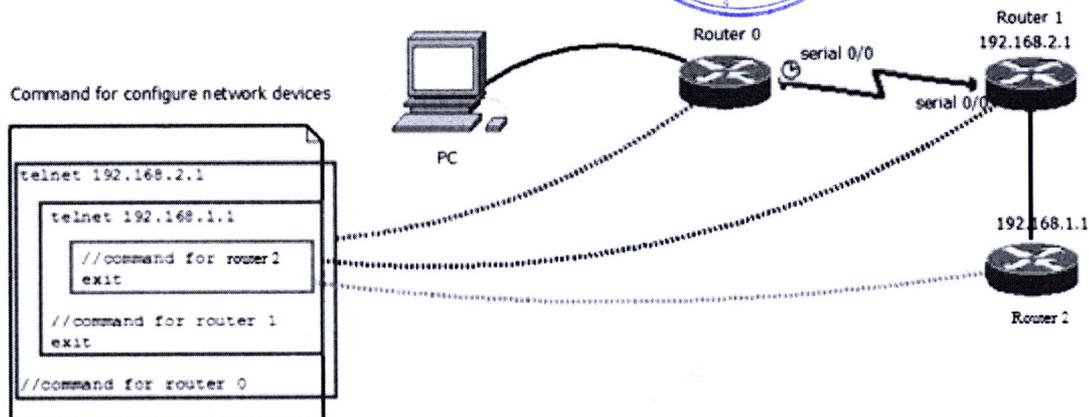
เมื่อผู้ใช้งาน โปรแกรมได้ทำการเชื่อมต่อตัวโปรแกรมและอุปกรณ์จริงเข้าด้วยกัน หลังจากทำการออกแบบเครือข่ายเรียบร้อยแล้ว โปรแกรมจะทำการตั้งค่าเริ่มต้นให้กับอุปกรณ์ โดยจะมีการสร้างชุดคำสั่งขึ้นมาเพื่อทำการตั้งค่าเริ่มต้นให้กับตัวของอุปกรณ์ต่างๆ เช่น ค่า IP Address ของอินเทอร์เฟซที่มีการใช้งานตามแบบเครือข่ายที่ได้ออกแบบไว้แล้ว ชื่อของอุปกรณ์ และรหัสผ่าน เป็นต้น เมื่อโปรแกรมทำการตั้งค่าเริ่มต้นได้สำเร็จ โปรแกรมจะมีการแจ้งแก่ผู้ใช้งาน หรือเมื่อทำการติดตั้งค่าเริ่มต้นให้กับอุปกรณ์ไม่สำเร็จ โปรแกรมจะมีการแจ้งเตือนสิ่งที่ควรตรวจสอบ เช่น การตรวจสอบสายและพอร์ตที่ใช้ในการออกแบบ และ/หรือ แจ้งให้ผู้ใช้งานเริ่มต้นการตั้งค่าเริ่มต้นใหม่



รูป 3.2 การตั้งค่าเริ่มต้นให้กับเราเตอร์



3.3.3 การตั้งค่าอุปกรณ์



รูป 3.3 การตั้งค่าให้ระบบเครือข่าย

เมื่อผู้ใช้โปรแกรมทำการตั้งค่าเริ่มต้นให้กับอุปกรณ์ ผู้ใช้งานยังสามารถที่จะทำการตั้งค่าอื่นๆ หรือเปลี่ยนแปลงค่าที่ได้ตั้งไว้โดยโปรแกรมได้ ซึ่งค่าต่างๆ ที่ผู้ใช้งานทำการตั้งค่าเพิ่มเติม หรือ เปลี่ยนแปลงค่าต่างๆที่ได้ตั้งไว้แล้วนั้น ทางตัวโปรแกรมจะนำไปตั้งค่าที่อุปกรณ์ต่างๆ ให้ โดยใช้อัลกอริทึมของการค้นหาแบบกว้างก่อน (Breadth first search) ในการสร้างการจำลองทางเดินของเครือข่ายภายในโปรแกรม เป็นอัลกอริทึมที่มีความเหมาะสมในการใช้งาน และประหยัดเวลาในหาเส้นทางในการติดต่อระหว่างอุปกรณ์แล้วจึงทำการกำหนดค่าอุปกรณ์ตัวนั้นๆ เหตุผลที่ต้องทำการกำหนดค่าในลักษณะนี้เป็นเพราะว่า หากทำการกำหนดค่าของอุปกรณ์ที่บริเวณใดก็ได้ของระบบ จะพบปัญหา คือระบบไม่สามารถที่จะติดต่อกันได้เป็นปกติ ซึ่งถ้าเกิดปัญหานี้ขึ้นระบบจะไม่สามารถตั้งค่าที่ได้เพิ่มเติมลงในตัวอุปกรณ์หรือทำการเปลี่ยนแปลงค่าต่างๆให้กับตัวอุปกรณ์ที่โปรแกรมไม่สามารถติดต่อไปถึงได้ จึงทำให้การตั้งค่านั้นล้มเหลวและระบบใช้งานไม่ได้

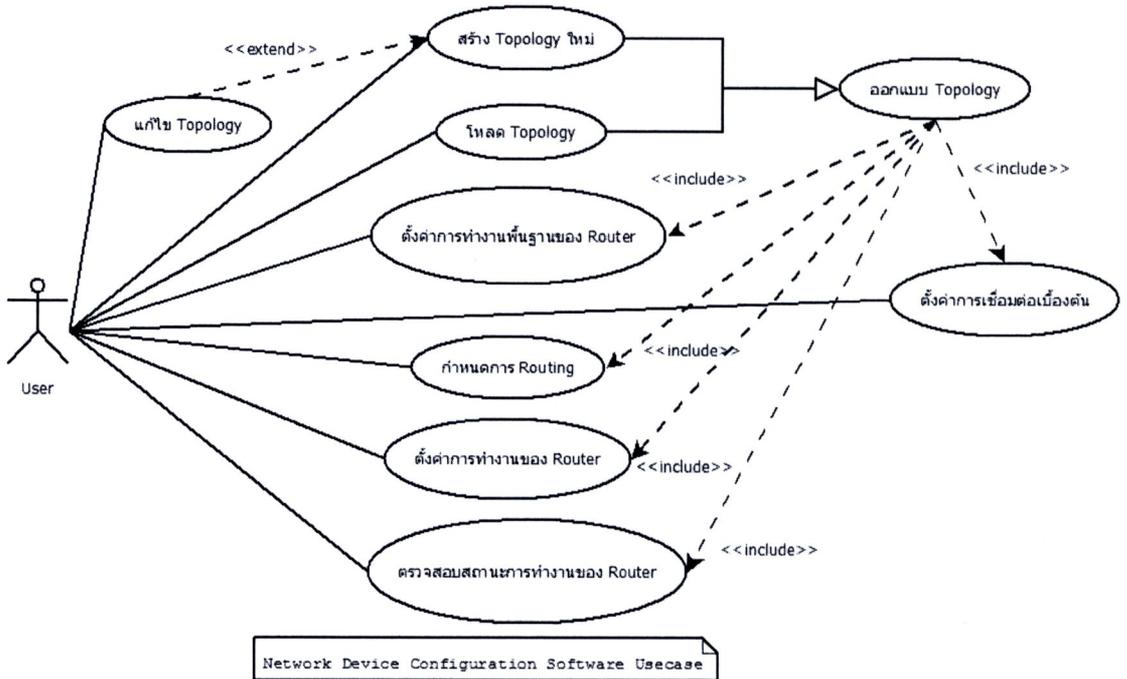
อย่างที่ผู้ใช้งานโปรแกรมต้องการดังรูป 3.4

3.4 การวิเคราะห์ความต้องการของระบบ

จากการวิเคราะห์ความต้องการของระบบโดยยึดถือตามขอบเขตของระบบที่จะใช้งานโดยผู้ใช้งาน โปรแกรม สามารถสรุปความต้องการของระบบในรูปแบบภาพ UML โดยแบ่งเป็น แผนภาพ Usecase และแผนภาพ Class ดังต่อไปนี้

3.4.1 แผนภาพ Usecase

แผนภาพ Usecase ใช้แสดงถึงความสัมพันธ์ระหว่างผู้ใช้และบริการที่ระบบมีให้ใช้งาน



รูป 3.4 แผนภาพ Usecase diagram ของ Network Device Configuration Software

3.4.2 ผู้ใช้งานระบบ

User คือผู้ที่ต้องการใช้งานโปรแกรม (หรือ Actor ของระบบ ซึ่งมีสัญลักษณ์เป็นรูปคน) โดยมีความต้องการที่จะทำการตั้งค่าเริ่มต้นของอุปกรณ์เครือข่ายให้สามารถเชื่อมต่อกันเบื้องต้นก่อน และต้องการที่จะทำการตั้งค่าการทำงานหรือแสดงค่าการทำงาน ของอุปกรณ์ที่อยู่ในเครือข่ายผ่านอุปกรณ์เครือข่ายเพียงตัวเดียว

3.4.3 ความสัมพันธ์ระหว่างผู้ใช้งานระบบกับ Usecase

ความสัมพันธ์ระหว่างผู้ใช้งานระบบกับ Usecase แทนด้วยเส้นลูกศร ซึ่งแสดงให้เห็นว่าผู้ใช้งานระบบสามารถใช้งานบริการใดของระบบได้บ้าง จากรูปสามารถแจกแจงได้ดังนี้

ผู้ใช้งานระบบสามารถเรียกใช้บริการ

- 1) สร้าง Topology ใหม่
- 2) แก้ไข Topology
- 3) โหลด Topology
- 4) ตั้งค่าการเชื่อมต่อเบื้องต้น

- 5) ตั้งค่าการทำงานพื้นฐานของเราเตอร์
- 6) กำหนดการ เราท์ติ้ง
- 7) ตั้งค่าการทำงานของเราเตอร์
- 8) ตรวจสอบสถานการณ์ทำงานของเราเตอร์

3.4.4 ความหมายของแต่ละ Usecase

รูปวงรีในแผนภาพ Usecase ใช้แทนแต่ละ Usecase ซึ่งเป็นบริการที่ระบบมีไว้ให้บริการกับผู้ใช้งานระบบ บริการของระบบที่มีให้ใช้แสดงในรูปมีดังนี้

3.4.4.1 บริการสร้าง Topology ใหม่

เป็นบริการหลักที่โปรแกรมจำเป็นต้องมี โดยจะมีพื้นที่ให้ผู้ใช้งานระบบใช้ในการออกแบบโครงสร้างของระบบเครือข่ายตามที่ผู้ใช้งานต้องการ ซึ่งจะมีอุปกรณ์เครือข่ายให้เลือกใช้ในการออกแบบอยู่สองประเภทคือเราเตอร์และสับ และมีสายที่เอาไว้ใช้ในการเชื่อมต่ออุปกรณ์เครือข่ายแต่ละตัวเข้าด้วยกัน โดยในบริการนี้ผู้ใช้งานระบบสามารถสร้าง Topology ได้ตามจำนวนที่ต้องการ

3.4.4.2 บริการแก้ไข Topology

เป็นบริการที่มีความสามารถให้ทำการแก้ไขเพิ่มเติมโครงสร้างของระบบเครือข่ายได้ โดยยังมีอุปกรณ์และเครื่องมือให้ใช้ในการแก้ไขเหมือนกับบริการสร้าง Topology ใหม่ โดยในบริการนี้สามารถทำได้กับทั้ง Topology ที่สร้างขึ้นใหม่หรือ Topology ที่ทำการไหลดมา

3.4.4.3 บริการไหลด Topology

เป็นบริการที่มีความสามารถให้ผู้ใช้งานโปรแกรมสามารถเรียกใช้งาน Topology ที่เคยสร้างไว้กลับมาใช้ใหม่ได้ ซึ่ง Topology เหล่านั้นจะต้องเคยถูกบันทึกจากโปรแกรมก่อน

3.4.4.4 บริการตั้งค่าการเชื่อมต่อเบื้องต้น

เป็นบริการที่จะทำการกำหนดค่าการทำงานของอุปกรณ์เครือข่ายทุกตัวให้โดยอัตโนมัติ เพื่อให้อุปกรณ์เครือข่ายแต่ละตัวสามารถเชื่อมต่อกันได้ โดยที่ผู้ใช้งานไม่ต้องทำการกำหนดค่าใดๆ แต่จะต้องทำการเชื่อมต่อกับอุปกรณ์เครือข่ายผ่านพอร์ตอนุกรมเพื่อส่งค่าการทำงานเบื้องต้นนี้ไปยังอุปกรณ์

3.4.4.5 บริการตั้งค่าการทำงานพื้นฐานของเราเตอร์

เป็นบริการที่มีให้สำหรับผู้ใช้งานระบบใช้ในการกำหนดค่าการทำงานพื้นฐานของเราเตอร์โดยง่ายได้ ซึ่งจะประกอบไปด้วย

- 1) ชื่อของเราเตอร์
- 2) รหัสผ่านที่ใช้ในการเข้าใช้งานผ่าน Console
- 3) รหัสผ่านที่ใช้ในการเข้าใช้งานผ่าน Virtual Terminal Lines
- 4) รหัสผ่านที่ใช้ในการเข้าใช้งาน Privileged โหมด
- 5) เนตเวิร์กแอคเดรส
- 6) สับเน็ตมาร์ก

โดยในบริการส่วนนี้ผู้ใช้งานจะทำการกำหนดค่าแค่บางส่วนที่ต้องการ หรือใช้ค่าการทำงานที่ถูกกำหนดมาแต่ต้นก็ได้

3.4.4.6 บริการกำหนดการเราต์ติ้ง

เป็นบริการที่ทำให้ผู้ใช้งานระบบสามารถเลือกรูปแบบในการส่งผ่านแพคเกจข้ามเน็ตเวิร์กที่ต่างกัน ได้ ซึ่งมีทั้งหมด 5 รูปแบบดังนี้

- 1) Static routing
- 2) RIP
- 3) RIPv2
- 4) EIGRP
- 5) OSPF

ซึ่งบริการเราต์ติ้งนี้จะทำการสร้างและส่งคำสั่งที่ใช้ในการเราต์ติ้งไปยังอุปกรณ์เครือข่ายทุกตัวที่ทำการเลือกโดยอัตโนมัติ

3.4.4.7 บริการตั้งค่าการทำงานของเราเตอร์

เป็นบริการที่มีให้สำหรับผู้ใช้งานระบบทำการกำหนดค่าการทำงานพื้นฐาน รวมไปถึงการทำงานอื่นๆของเราเตอร์ได้ ซึ่งประกอบไปด้วย

- 1) ชื่อของเราเตอร์
- 2) รหัสผ่านที่ใช้ในการเข้าใช้งานผ่าน Console
- 3) รหัสผ่านที่ใช้ในการเข้าใช้งานผ่าน Virtual Terminal Lines
- 4) รหัสผ่านที่ใช้ในการเข้าใช้งาน Privileged โหมด
- 5) อินเตอร์เฟส
- 6) การเราต์ติ้ง

ซึ่งบริการตั้งค่าการทำงานของเราเตอร์นี้จะทำการสร้างและส่งคำสั่งที่ใช้ในการกำหนดค่าการทำงานไปยังเราเตอร์แต่ละตัวเพื่อทำการตั้งค่าโดยอัตโนมัติ

3.4.4.8 บริการตรวจสอบสถานะของการทำงานของเราเตอร์

เป็นบริการที่จะทำการรวบรวมข้อมูลการทำงานทั้งหมดของเราเตอร์แต่ละตัว ตัวอย่างเช่น สถานะของอินเตอร์เฟซ เป็นต้น มาแสดงให้ผู้ใช้งานเห็นในรูปแบบที่เข้าใจง่าย

3.4.5 แผนภาพการวิเคราะห์ในเชิงวัตถุ

แผนภาพการวิเคราะห์ในเชิงวัตถุ หรือแผนภาพ Class ใช้แสดงความสัมพันธ์ของ Class ต่างๆ ในระบบ ตามมุมมองของวิธีการวิเคราะห์ในเชิงวัตถุ

เริ่มต้นด้วยแผนภาพ Component ซึ่งแสดงให้เห็นถึงการจัดเก็บกลุ่มของ Class ตามหมวดหมู่ที่สัมพันธ์กัน กลุ่มของ Class จะถูกจัดเก็บไว้ด้วยกันใน “แพคเกจ” แทนด้วยสัญลักษณ์โฟลเดอร์ ดังรูป 3.6

จากการวิเคราะห์ระบบ เราได้แบ่งการจัดเก็บ Class ออกเป็นหลายแพคเกจ ดังนี้

1) แพคเกจ Topology ใช้เก็บเหล่าคลาสที่เกี่ยวข้องกับการออกแบบโครงสร้างระบบ เครือข่ายซึ่งมีการอ้างอิงหรือมีความสัมพันธ์ถึง คลาสในแพคเกจ Initial_Configuration แพคเกจ Basic_Configuration แพคเกจ เราต์ติ้ง แพคเกจ Device_Configuration และ แพคเกจ Monitoring (แสดงด้วยเส้นประ ในรูป 3.6)

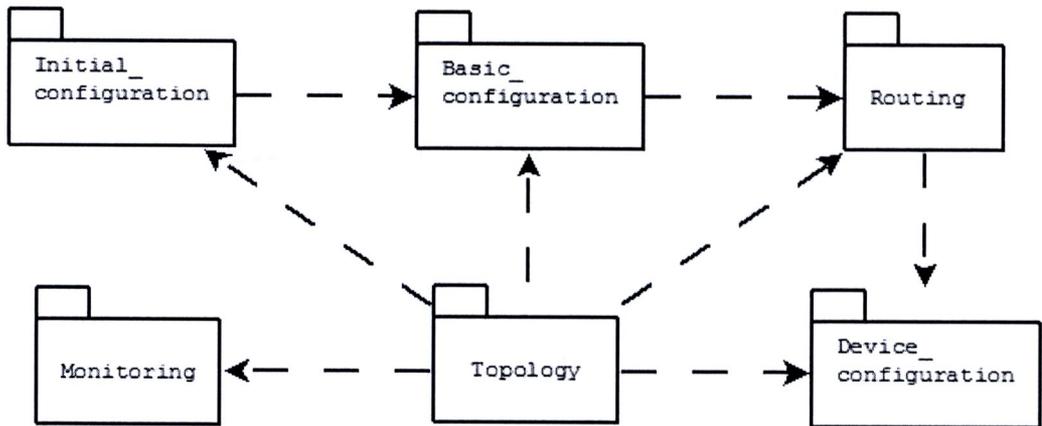
2) แพคเกจ Initial_Configuration ใช้เก็บเหล่าคลาสที่ใช้ในการตั้งค่าการทำงานเริ่มต้น ของอุปกรณ์เครือข่าย ซึ่งมีการอ้างอิงหรือมีความสัมพันธ์ถึง คลาสในแพคเกจ Basic_Configuration (แสดงด้วยเส้นประ ในรูป 3.6)

3) แพคเกจ Basic_Configuration ใช้เก็บเหล่าคลาสที่ใช้ในการตั้งค่าการทำงานพื้นฐานของอุปกรณ์เครือข่าย ซึ่งมีการอ้างอิงหรือมีความสัมพันธ์ถึง คลาสในแพคเกจ เราต์ติ้ง (แสดงด้วยเส้นประ ในรูป 3.6)

4) แพคเกจ Routing ใช้เก็บเหล่าคลาสที่ใช้ในการกำหนดค่าการส่งผ่านแพคเกจข้ามเน็ตเวิร์ก ซึ่งมีการอ้างอิงหรือมีความสัมพันธ์ถึง คลาสในแพคเกจ Device_Configuration (แสดงด้วยเส้นประ ในรูป 3.6)

5) แพคเกจ Device_Configuration ใช้เก็บเหล่าคลาสที่ใช้ในการตั้งค่าการทำงานต่างๆ ของอุปกรณ์เครือข่าย

6) แพคเกจ Monitoring ใช้เก็บเหล่าคลาสที่ใช้ในการตรวจสอบสถานะ และค่าที่ทำการตั้งค่าไว้ของอุปกรณ์เครือข่ายทุกตัวในระบบ

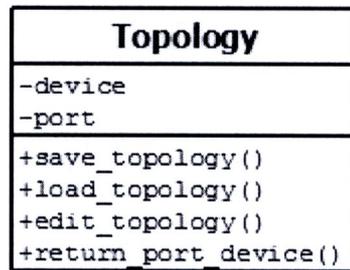


รูป 3.5 การแบ่งแพ็คเกจ

ต่อไปจะกล่าวถึงรายละเอียดภายในแต่ละแพ็คเกจ ซึ่งมีรายละเอียดดังต่อไปนี้

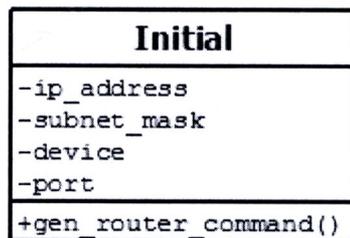
3.4.5.1 แพ็คเกจ *Topology*

ภายในจะประกอบด้วยคลาส *Topology*

รูป 3.6 คลาสในแพ็คเกจ *Topology*

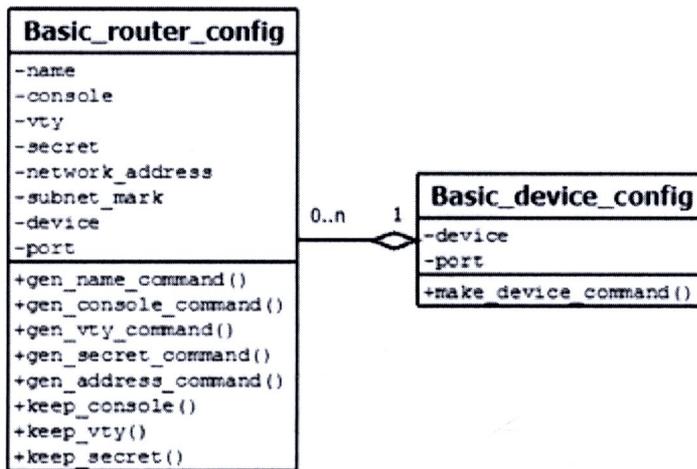
3.4.5.2 แพ็คเกจ *Initial_Configuration*

ภายในประกอบไปด้วยคลาส *Initial*

รูป 3.7 คลาสในแพ็คเกจ *Initial_Configuration*

3.4.5.3 แพ็คเกจ *Basic_Configuration*

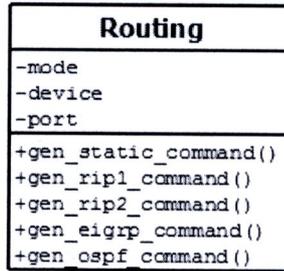
ภายในประกอบไปด้วยคลาส *Basic_device_config* ซึ่งเป็นคอนเทนเนอร์ สำหรับคลาส *Basic_router_config*



รูป 3.8 คลาสในแพ็คเกจ Basic_configuration

3.4.5.4 แพคเกจ Routing

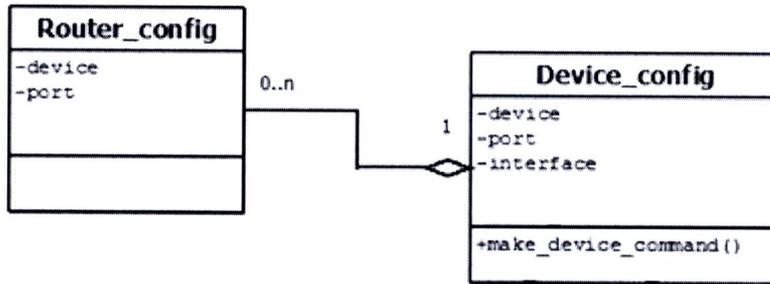
ภายในประกอบไปด้วยคลาส Routing



รูป 3.9 คลาสในแพคเกจ Routing

3.4.5.5 แพคเกจ Device_Configuration

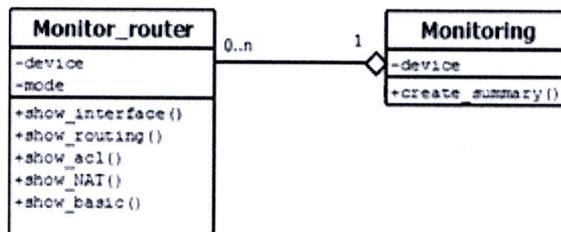
ภายในประกอบไปด้วยคลาส Device_config ซึ่งเป็นคอนเทนเนอร์ สำหรับคลาส Router_config



รูป 3.10 คลาสในแพคเกจ Device_configuration

3.4.5.6 แพคเกจ Monitoring

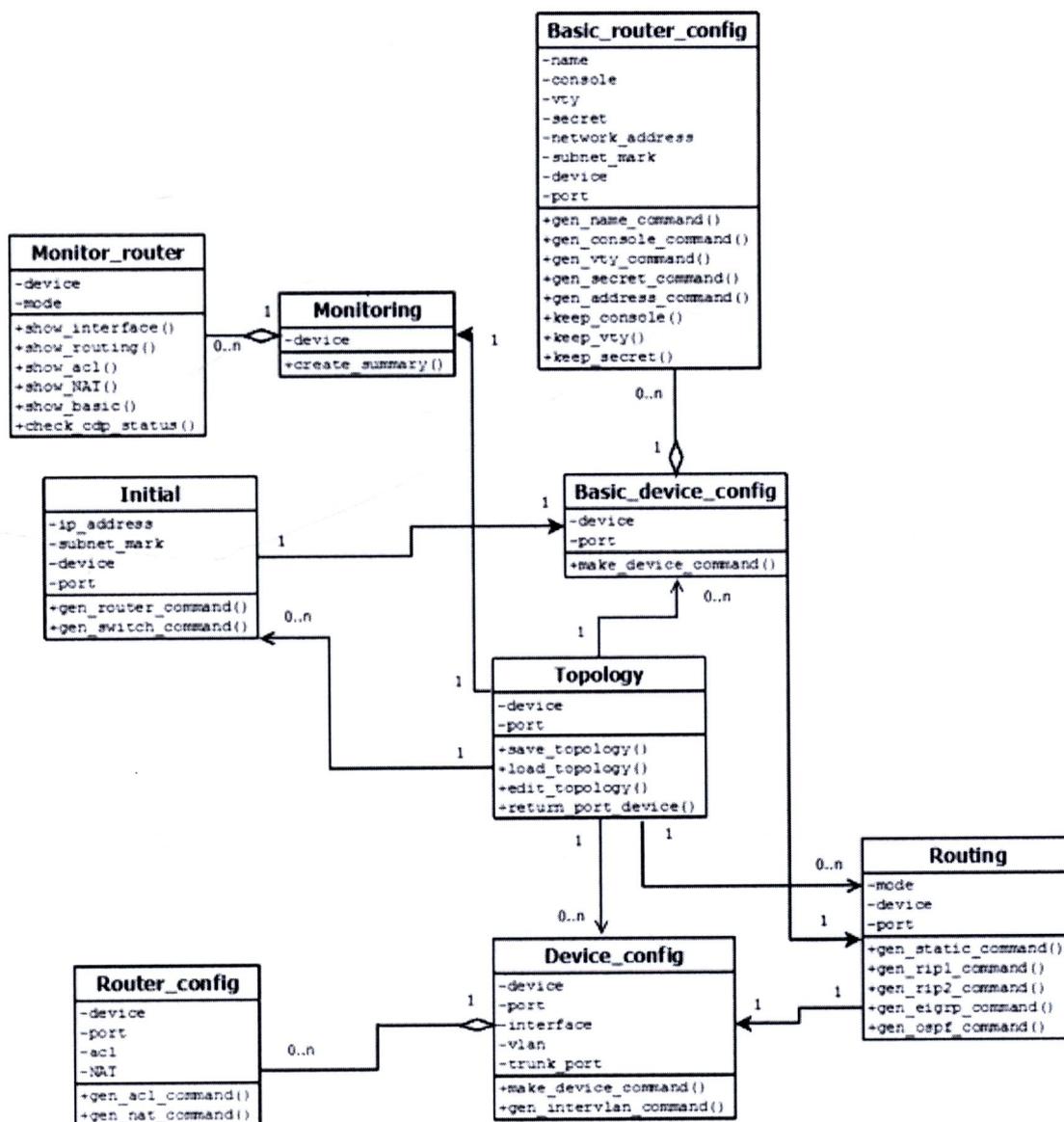
ภายในประกอบไปด้วยคลาส Monitoring ซึ่งเป็นคอนเทนเนอร์ สำหรับคลาส Monitor_router



รูป 3.11 คลาสในแพคเกจ Monitoring

ซึ่งเมื่อนำมาเชื่อมโยงความสัมพันธ์ระหว่างแต่ละคลาสแล้วจะได้เป็นแผนภาพ

Class ดังนี้



รูป 3.12 แผนภาพ Class ของโปรแกรม Network Device Configuration

