การป้องกันการเกิดลูปของระบบเครือข่ายด้วยโปรโตคอลสแปนนิงทรี
LOOP PROTECTION FOR NETWORK SYSTEM USING SPANNING TREE PROTOCOL

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

การป้องกันการเกิดลูปของระบบเครือข่ายด้วยโปรโตคอลสแปนนิงทรี

LOOP PROTECTION FOR NETWORK SYSTEM USING SPANNING TREE PROTOCOL

วรินทร สุธีนิยมเดช

b.00265714
TR00172

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOOP PROTECTION FOR NETWORK SYSTEM USING SPANNING TREE PROTOCOL

WARINTHON SUTEENIYOMDAT

A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การป้องกันการเกิดลูปของระบบเครือข่ายด้วยโปรโตคอลสแปนนิงทรี Loop Protection for Network System using Spanning Tree Protocol
ชื่อนักศึกษา	นางสาววรินทร สุธีนิยมเดช รหัสนักศึกษา 56050367
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2559
อาจารย์ที่ปรึกษา	ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง(สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2559

คณะกรรมการสอบ	ลายมือชื่อ
ดร.ไพรัตน์ ธรเจริญศรี ประธานกรรมการ	
ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา การป้องกันการเกิดลูปของระบบเครือข่ายด้วยโปรโตคอลสแปนนิงทรี
ชื่อนักศึกษา นางสาววรินทร สุธีนิยมเดช รหัสนักศึกษา 56050367
ปริญญา วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา วิทยาการคอมพิวเตอร์
คณะ วิทยาศาสตร์
มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา 2559
อาจารย์ที่ปรึกษา ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์

บทคัดย่อ

สหกิจศึกษานี้มีวัตถุประสงค์เพื่อศึกษาและพัฒนาโปรแกรมที่ใช้ในการแก้ปัญหาเรื่องลูป (Loop) ในเลเยอร์สอง หรือสถานะที่สวิตช์ (Switch) มีการรับส่งข้อมูลวนแบบเดิมซ้ำในระบบเครือข่าย โดยการใช้โปรโตคอลสแปนนิงทรี (Spanning Tree Protocol) แบ่งการดำเนินงานเป็น 3 ส่วน 1. การหาค่าคอนฟิกที่เหมาะสมในการทำสแปนนิงทรีให้กับสวิตช์ในระบบเครือข่าย 2. การสร้างสคริปต์ (Script) คำสั่งภาษาทีซีแอล (Tool Command Language) เพื่อช่วยในการคอนฟิกค่าต่างๆแต่ละค่าของสวิตช์ที่เกี่ยวกับสแปนนิงทรี เช่น ค่าพอร์ต (Port) ค่า Path Cost และค่า Priority ได้โดยง่าย และ 3. การพัฒนาโปรแกรมภาษาจาวาในการนำเข้าค่าจากไฟล์ .csv และสร้างเป็นไฟล์สคริปต์ภาษาทีซีแอลจำนวนเท่ากับจำนวนสวิตช์เพื่อนำไปใช้ในการคอนฟิกค่าทั้งหมดในการตั้งค่าสแปนนิงทรีให้กับสวิตช์ทุกตัวในระบบ ผู้ดูแลระบบสามารถรันไฟล์เหล่านี้ที่สวิตช์แต่ละอุปกรณ์แทนการตั้งค่าทุกค่าด้วยมือ ซึ่งเป็นการประหยัดเวลาและอำนวยความสะดวกแก่ผู้ดูแลระบบ

คำสำคัญ : ทีซีแอล พอร์ต ลูป สแปนนิงทรี สวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Loop Protection for Network System using Spanning Tree Protocol
Student	Miss. Warinthon Suteeniyomdat Student ID 56050367
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2016
Advisor	Dr.Rungrat Wiangsripanawan

Abstract

The purposes of this cooperative education is to study, and develop a program to solve the switching loop problem by using Spanning Tree Protocol (STP). The operation is divided into three parts. The first part is to find proper STP's configuration for switches in the network. The second part is to create a Tool Command Language (Tcl) scripts to assist a network administrators when setting each STP switch's values such as port, path cost and priority. The final part is to develop a java program to import a csv file that storing all switches' STP configurations and convert it into Tcl files. The number of Tcl files depends on the number of the switches. Rather than configuring all STP values manually one by one in each switch, network operators can run these STP's Tcl files – one for each switch instead. These can save time and facilitate network administrators.

Keywords : Loop, Port, STP, Switch, Tcl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปัญหาพิเศษเล่มนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความช่วยเหลือและความกรุณาอย่างดียิ่งจาก ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ อาจารย์ที่ปรึกษาปัญหาพิเศษ ที่ได้ให้คำปรึกษาอย่างใกล้ชิดและเสนอแนะแนวทางแก้ปัญหา รวมทั้งตรวจแก้ปัญหาพิเศษฉบับนี้ให้มีความสมบูรณ์เพิ่มขึ้น ผู้จัดทำปัญหาพิเศษจึงใคร่ขอขอบพระคุณท่านเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณ บริษัท ซีเอส ล็อกซอินโฟ จำกัด (มหาชน) และคณะกรรมการผู้รับผิดชอบในการดำเนินงาน ที่ได้ให้คำปรึกษาและอำนวยความสะดวกในการดำเนินงานมาโดยตลอด

ขอขอบพระคุณคณาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ ให้แก่ข้าพเจ้าได้นำความรู้ที่ได้ร่ำเรียนมาจัดทำโครงงานสหกิจศึกษานี้ให้สำเร็จไปด้วยดี

ท้ายที่สุดนี้ ผู้จัดทำปัญหาพิเศษขอกราบขอบพระคุณบิดา มารดา ผู้ที่ให้กำเนิดและอุปการะอบรมเลี้ยงดูมาโดยตลอด ผู้จัดทำปัญหาพิเศษจึงใคร่ขอขอบพระคุณ ทุกท่านเป็นอย่างสูงไว้ ณ ที่นี้

วรินทร สุธีนิยมเดช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ
คำย่อ/สัญลักษณ์ ..... ฌ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญ 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขต 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ 3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4
2.1 ระบบเครือข่าย 4
2.1.1 ชนิดของเครือข่าย 4
2.1.2 เน็ตเวิร์คโปรโตคอล 5
2.2 อุปกรณ์ 8
2.2.1 สวิตช์ 8
2.2.3 สาย UTP 11
2.2.4 การเชื่อมต่ออุปกรณ์ 12
2.3 ลูป 13
2.4 Redundancy 15
2.4.1 โปรโตคอลสแปนนิงทรี 15
2.4.2 EtherChannel 26
2.5 Tool Command Language 27
2.6 โปรแกรมที่เกี่ยวข้อง 29
2.6.2 โปรแกรม Putty 29
2.6.3 โปรแกรม Wireshark 30
2.6.4 โปรแกรม Networx 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.6.5 โปรแกรม Tftpd32 31
2.6.6 โปรแกรม Netbeans 32
บทที่ 3 วิธีการดำเนินงาน 33
3.1 ขั้นตอนการศึกษารวบรวมข้อมูลของระบบ 34
3.1.1 ศึกษาระบบเครือข่ายเดิม 34
3.1.2 การตรวจสอบการเกิดลูป 39
3.2 ขั้นตอนการออกแบบระบบ 48
3.2.1 การออกแบบระบบเครือข่าย 48
3.2.2 อุปกรณ์ที่นำมาใช้ 52
3.3 ขั้นตอนการพัฒนาระบบ 53
3.3.1 แผนการดำเนินงาน 53
3.3.2 การคอนฟิก 54
3.4 ขั้นตอนการออกแบบสคริปต์ช่วยในการพัฒนาระบบ 58
3.5 ขั้นตอนการออกแบบโปรแกรมช่วยในการพัฒนาระบบ 62
บทที่ 4 ผลการดำเนินงานและการอภิปรายผล 64
4.1 ผลการคอนฟิก 64
4.1.1 การแสดงผล Etherchannel 64
4.1.2 การแสดงผล Trunk 65
4.1.3 การแสดงผล Vlan 66
4.2 ผลการติดตั้งอุปกรณ์ 66
4.3 ผลการทดสอบการเกิดลูป 67
4.3.1 สวิตช์ SW1 เชื่อมต่อกับสวิตช์ SWB 68
4.3.2 สวิตช์ SW1 เชื่อมต่อกับสวิตช์ SWC 72
4.3.3 สวิตช์ SW1 เชื่อมต่อกับสวิตช์ SW2 75
4.3.4 สวิตช์ SWB เชื่อมต่อกับสวิตช์ SWC 78
4.3.5 สวิตช์ SWB เชื่อมต่อกับสวิตช์ SW2 81
4.3.6 สวิตช์ SWC เชื่อมต่อกับสวิตช์ SW2 84
4.3.7 สวิตช์ SWB เชื่อมต่อกับสวิตช์ CORE 87
4.4 การทำงานของสคริปต์ช่วยในการพัฒนาระบบ 90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.5 การทำงานของโปรแกรมช่วยในการพัฒนาระบบ 97
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ 101
5.1 สรุปผลการดำเนินงาน 101
5.2 ข้อจำกัดในการพัฒนาระบบ 101
5.3 ข้อเสนอแนะ 102
เอกสารอ้างอิง 103
ภาคผนวก 104
ภาคผนวก ก วิธีการใช้งานพื้นฐานโปรแกรม Putty 109
ภาคผนวก ข วิธีการใช้งานพื้นฐานโปรแกรม NetBeans 108
ภาคผนวก ค Core Tcl Commands 112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงค่า Path Cost เทียบกับค่าแบนด์วิธ 20
2.2 สรุปสถานะและความสามารถของพอร์ต 24
3.1 แสดงการเชื่อมต่อของสวิตช์ CORE 36
3.2 แสดงการเชื่อมต่อของสวิตช์ Cisco 2960S-24TS-L (SW1) 37
3.3 แสดงการเชื่อมต่อของสวิตช์ Cisco 2960S-24TS-L (SW2) 37
3.4 แสดงการเชื่อมต่อของสวิตช์ 3COM (SWB) 38
3.5 แสดงการเชื่อมต่อของสวิตช์ 3COM (SWC) 38
3.6 แสดงรายละเอียดการเชื่อมต่อของอุปกรณ์ 40
3.7 แสดงรายละเอียดการเชื่อมต่อของสวิตช์ CORE 50
3.8 แสดงรายละเอียดการเชื่อมต่อของสวิตช์ SW1 51
3.9 แสดงรายละเอียดการเชื่อมต่อของสวิตช์ SWB 51
3.10 แสดงรายละเอียดการเชื่อมต่อของสวิตช์ SWC 51
3.11 แสดงรายละเอียดการเชื่อมต่อของสวิตช์ SW2 51
3.12 แสดงแผนการดำเนินงาน 53
3.13 แสดงรูปแบบตารางเก็บค่าที่ใช้คำนวณ STP 62
4.1 แสดงตัวอย่างการเก็บข้อมูลเพื่อใช้ในโปรแกรม 97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 แสดงการเชื่อมต่อของเครือข่ายภายใน	4
รูปที่ 2.2 แสดงการเชื่อมต่อของเครือข่ายวงกว้าง	5
รูปที่ 2.3 แสดงเฟรมของ ICMP Message	6
รูปที่ 2.4 แสดงคำสั่งในการ ping	6
รูปที่ 2.5 แสดงลักษณะของ Data Link เฟรม	7
รูปที่ 2.6 แสดงการทำงานเริ่มต้นของสวิตช์	9
รูปที่ 2.7 แสดงการทำงานของสวิตช์	10
รูปที่ 2.8 แสดงการเข้าหัวสาย UTP แบบ Straight Through	11
รูปที่ 2.9 แสดงการเข้าหัวสาย UTP แบบ Crossover	12
รูปที่ 2.10 แสดงประเภทการเชื่อมต่อของอุปกรณ์เน็ตเวิร์ค	13
รูปที่ 2.11 แสดงตัวอย่างการเกิดลูป	14
รูปที่ 2.12 Data Flow Diagram แสดงการทำงานของ STP	16
รูปที่ 2.13 Data Flow Diagram แสดงการทำงานในการหา Root Bridge	18
รูปที่ 2.14 แสดงตัวอย่างการหา Root Bridge	19
รูปที่ 2.15 Data Flow Diagram แสดงการทำงานในการหา Root Port	21
รูปที่ 2.16 แสดงตัวอย่างการหา Root Port	22
รูปที่ 2.17 แสดงตัวอย่างการหา Designated Port	23
รูปที่ 2.18 แสดงสถานะของพอร์ต	23
รูปที่ 2.19 แสดงลักษณะการทำ EtherChannel	26
รูปที่ 2.20 แสดงโปรแกรม Putty	29
รูปที่ 2.21 แสดงโปรแกรม Wireshark	30
รูปที่ 2.22 แสดงโปรแกรม Networx	31
รูปที่ 2.23 แสดงโปรแกรม Tftpd32	32
รูปที่ 2.24 แสดงโปรแกรม NetBeans	32
รูปที่ 3.1 Data Flow Diagram แสดงวิธีการดำเนินงาน	33
รูปที่ 3.2 Network Diagram ของระบบเครือข่ายเดิม	35
รูปที่ 3.3 แสดงการเชื่อมต่อหลังเกิดปัญหา	35
รูปที่ 3.4 แสดงลักษณะสวิตช์ Cisco รุ่น 2960G	36
รูปที่ 3.5 แสดงลักษณะสวิตช์ Cisco รุ่น 2960S-24TS-L	38
รูปที่ 3.6 แสดงลักษณะสวิตช์ 3COM รุ่น 3C16476CS	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

รูปที่ 3.7 แสดงการเชื่อมต่อก่อนเกิดลูป 39
รูปที่ 3.8 แสดงผลการ ping จาก PC1 ไป PC2 40
รูปที่ 3.9 แสดงผลการ ping จาก PC1 ไป PC2 41
รูปที่ 3.10 แสดง MAC address Table ของสวิตช์ CORE ก่อนเกิดลูป 41
รูปที่ 3.11 แสดง MAC address Table ของสวิตช์ SWB ก่อนเกิดลูป 42
รูปที่ 3.12 แสดง MAC address Table ของสวิตช์ SWC ก่อนเกิดลูป 42
รูปที่ 3.13 กราฟแสดงอัตราการรับส่งข้อมูลก่อนเกิดลูปจากโปรแกรม Networx 43
รูปที่ 3.14 แสดงการเชื่อมต่อที่ทำให้เกิดลูป 43
รูปที่ 3.15 แสดงผลการ ping จาก PC1 ไป PC2 44
รูปที่ 3.16 แสดงผลการ ping จาก PC2 ไป PC1 44
รูปที่ 3.17 แสดง Mac Address Table ของสวิตช์ CORE หลังเกิดลูป 45
รูปที่ 3.18 แสดง Mac Address Table ของสวิตช์ SWB หลังเกิดลูป 45
รูปที่ 3.19 แสดง MAC Address Table ของสวิตช์ SWC หลังเกิดลูป 46
รูปที่ 3.20 กราฟแสดงอัตราการรับส่งข้อมูลหลังเกิดลูปจากโปรแกรม Networx 47
รูปที่ 3.21 แสดงการตรวจสอบข้อมูลจากโปรแกรม Wireshark 47
รูปที่ 3.22 แสดงลักษณะการเชื่อมต่อที่สามารถทำให้เกิดลูปขึ้นของสวิตช์ทั้ง 5 ตัว 48
รูปที่ 3.23 แสดงการเชื่อมต่อของระบบเครือข่ายใหม่ 49
รูปที่ 3.24 แสดงการเชื่อมต่อของระบบใหม่กับอุปกรณ์ลูกค้า 49
รูปที่ 3.25 แสดงลักษณะสวิตช์ Cisco รุ่น 2960S 53
รูปที่ 3.26 Data Flow Diagram แสดงการทำงานของสคริปต์ 59
รูปที่ 3.27 Data Flow Diagram แสดงการทำงานของโปรแกรม 63
รูปที่ 4.1 แสดงผล Etherchannel ของสวิตช์ CORE 64
รูปที่ 4.2 แสดงผล Etherchannel ของสวิตช์ SW1 64
รูปที่ 4.3 แสดงผล Trunk ของสวิตช์ CORE 65
รูปที่ 4.4 แสดงผล Trunk ของสวิตช์ SW1 65
รูปที่ 4.5 แสดงผล Vlan ของสวิตช์ CORE 66
รูปที่ 4.6 แสดงการติดตั้งอุปกรณ์ 66
รูปที่ 4.7 แสดงคำสั่งที่ใช้ใน STP 67
รูปที่ 4.8 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SW1 ต่อกับ SWB 68
รูปที่ 4.9 แสดงการหา Designated Port ในการทดสอบสวิตช์ SW1 ต่อกับ SWB 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

รูปที่ 4.10 แสดงผล STP บนสวิตช์ CORE ในการทดสอบสวิตช์ SW1 ต่อกับ SWB 70
รูปที่ 4.11 แสดงผล STP บนสวิตช์ SW1 ในการทดสอบสวิตช์ SW1 ต่อกับ SWB 71
รูปที่ 4.12 แสดงผล STP บนสวิตช์ SWB ในการทดสอบสวิตช์ SW1 ต่อกับ SWB 71
รูปที่ 4.13 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SW1 ต่อกับ SWC. 72
รูปที่ 4.14 แสดงการหา Desiganted Port ในการทดสอบสวิตช์ SW1 ต่อกับ SWC 73
รูปที่ 4.15 แสดงผล STP บนสวิตช์ SW1 ในการทดสอบสวิตช์ SW1 ต่อกับสวิตช์ SWC..... 74
รูปที่ 4.16 แสดงผล STP บนสวิตช์ SWC ในการทดสอบสวิตช์ SW1 ต่อกับสวิตช์ SWC..... 74
รูปที่ 4.17 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SW1 ต่อกับ SW2.. 75
รูปที่ 4.18 แสดงการหา Designated Port ในการทดสอบสวิตช์ SW1 ต่อกับ SW2..... 76
รูปที่ 4.19 แสดงผล STP บนสวิตช์ SW1 ในการทดสอบสวิตช์ SW1 ต่อกับ SW2 77
รูปที่ 4.20 แสดงผล STP บนสวิตช์ SW2 ในการทดสอบสวิตช์ SW1 ต่อกับ SW2 77
รูปที่ 4.21 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SWB ต่อกับ SWC. 78
รูปที่ 4.22 แสดงการหา Designated Port ในการทดสอบสวิตช์ SWB ต่อกับ SWC 79
รูปที่ 4.23 แสดงผล STP บนสวิตช์ SWB ในการทดสอบสวิตช์ SWB ต่อกับ SWC..... 80
รูปที่ 4.24 แสดงผล STP บนสวิตช์ SWC ในการทดสอบสวิตช์ SWB ต่อกับ SWC 80
รูปที่ 4.25 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SWB ต่อกับ SW2.. 81
รูปที่ 4.26 แสดงการหา Designated Port ในการทดสอบสวิตช์ SWB ต่อกับ SW2..... 82
รูปที่ 4.27 แสดงผล STP บนสวิตช์ SWB ในการทดสอบสวิตช์ SWB ต่อกับ SW2..... 83
รูปที่ 4.28 แสดงผล STP บนสวิตช์ SW2 ในการทดสอบสวิตช์ SWB ต่อกับ SW2 83
รูปที่ 4.29 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SWC ต่อกับ SW2. 84
รูปที่ 4.30 แสดงการหา Designated Port ในการทดสอบสวิตช์ SWC ต่อกับ SW2 85
รูปที่ 4.31 แสดงผล STP บนสวิตช์ SWC ในการทดสอบสวิตช์ SWC ต่อกับ SW2..... 86
รูปที่ 4.32 แสดงผล STP บนสวิตช์ SW2 ในการทดสอบสวิตช์ SWC ต่อกับ SW2..... 86
รูปที่ 4.33 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ CORE ต่อกับ SWB 87
รูปที่ 4.34 แสดงการหา Designated Port ในการทดสอบสวิตช์ CORE ต่อกับ SWB..... 88
รูปที่ 4.35 แสดงผล STP บนสวิตช์ CORE ในการทดสอบสวิตช์ CORE ต่อกับ SWB 89
รูปที่ 4.36 แสดงผล STP บนสวิตช์ SWB ในการทดสอบสวิตช์ CORE ต่อกับ SWB..... 89
รูปที่ 4.37 แสดงการคัดลอกไฟล์สคริปต์บนสวิตช์จากคอมพิวเตอร์ 90
รูปที่ 4.38 แสดงการทำงานของโปรแกรม Tftpd 32 90
รูปที่ 4.39 แสดงรายละเอียดไฟล์ในหน่วยความจำแฟลช 91

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

รูปที่ 4.40 แสดงการเรียกไฟล์สคริปต์ขึ้นมาทำงาน 91
รูปที่ 4.41 แสดงข้อความเตือนเมื่อผู้ใช้งานใส่ค่าผิดในโหมดการเลือก Configuration Types.... 91
รูปที่ 4.42 แสดงการรับข้อมูลในโหมด Auto Configuration 92
รูปที่ 4.43 แสดงการรับข้อมูลในโหมด Root Bridge Configuration 92
รูปที่ 4.44 ผลการคอนฟิกจากโหมด Root Bridge Configuration 92
รูปที่ 4.45 แสดงการรับข้อมูลในโหมด Non Root Bridge Configuration 93
รูปที่ 4.46 ผลการคอนฟิกจากโหมด Non Root Bridge Configuration 93
รูปที่ 4.47 แสดงการรับข้อมูลในโหมด Manual Configuration 93
รูปที่ 4.48 แสดงข้อความเตือนเมื่อผู้ใช้งานใส่ค่าผิดในการเลือก Bridge Priority Configuration.. 94
รูปที่ 4.49 แสดงการป้อนค่า no ในการเลือก Bridge Priority Configuration 94
รูปที่ 4.50 แสดงการรับค่า y ในการเลือก Bridge Priority Configuration 94
รูปที่ 4.51 แสดงการรับข้อมูลในโหมด Bridge Priority Configuration 95
รูปที่ 4.52 แสดงรายละเอียดและรูปแบบในการป้อนค่า 95
รูปที่ 4.53 แสดงการทดสอบการป้อนค่าในโหมดการคอนฟิกพื้นฐาน 96
รูปที่ 4.54 แสดงการป้อนค่า exit เพื่อออกจากโหมดคอนฟิกพื้นฐาน 96
รูปที่ 4.55 แสดงการป้อนค่า ex เพื่อออกจากโหมดคอนฟิกพื้นฐาน 96
รูปที่ 4.56 ผลการคอนฟิกจากโหมดการคอนฟิกพื้นฐาน 97
รูปที่ 4.57 แสดงหน้าจอเมื่อเปิดโปรแกรม 98
รูปที่ 4.58 แสดงหน้าจอสำหรับให้ผู้ใช้งานเลือกไฟล์ .csv 98
รูปที่ 4.59 แสดงหน้าจอเมื่อผู้ใช้งานเลือกไฟล์ .csv แล้ว 99
รูปที่ 4.60 แสดงหน้าจอเมื่อระบบทำงานเสร็จ 99
รูปที่ 4.61 แสดงที่อยู่ของไฟล์ .tcl 100
รูปที่ 4.62 แสดงคำสั่งในไฟล์ .tcl 100
รูปที่ ก.1 โปรแกรม putty 104
รูปที่ ก.2 แสดงส่วนประกอบของโปรแกรม putty 104
รูปที่ ก.3 การหา Serial line 105
รูปที่ ก.4 การเชื่อมต่อแบบ Serial 106
รูปที่ ก.5 การเชื่อมต่อแบบ Telnet 106
รูปที่ ก.6 การเชื่อมต่อแบบ SSH 107
รูปที่ ข.1 JDK1 108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

รูปที่ ข.2 โปรแกรม NetBeans 108
รูปที่ ข.3 แสดงหน้าจอหลักของโปรแกรม NetBeans 108
รูปที่ ข. 4 หน้าจอสร้าง Project 109
รูปที่ ข.5 หน้าจอตั้งชื่อ Project และ Class 109
รูปที่ ข.6 แสดงการสร้าง class ขึ้นมาใหม่ 110
รูปที่ ข.7 หน้าจอการตั้งชื่อ class 110
รูปที่ ข.8 หน้าจอสำหรับเขียน Code และ Run โปรแกรม 111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำย่อ/สัญลักษณ์

คำย่อ/สัญลักษณ์	คำอธิบาย
STP	Spanning-Tree Protocal
Tcl	Tool Command Language
LAN	Local Area Network
WAN	Wide Area Network
ICMP	Internet Control Message Protocol
ARP	Address Resolution Protocol
TFTP	Trivial File Transfer Protocol
UTP	Unshield Twisted Pair
MDI	Medium Dependent Interface
MDI-X	Medium Dependent Interface Crossover
BPDU	Bridge Protocol Data Unit
TCN	Topology Change Notification
TTL	Tool Command Language
CSV	Comma Separated Value
config	Configuration
sh	Show

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบัน การติดต่อสื่อสารบนระบบเครือข่ายมีความจำเป็นอย่างยิ่งในทุกระดับ การออกแบบระบบเครือข่ายให้สามารถติดต่อสื่อสารกันได้อย่างมีประสิทธิภาพจึงเป็นสิ่งสำคัญมาก เพื่อให้สามารถรองรับการใช้งานจากผู้ใช้จำนวนมากได้อย่างเหมาะสม รวมถึงการเตรียมพร้อมป้องกันกับปัญหาที่มีโอกาสจะเกิดขึ้น เพื่อให้ระบบและอุปกรณ์สามารถพร้อมทำงานได้อยู่เสมอ ภายใต้ทรัพยากรที่พอเหมาะและเทคโนโลยีที่เปลี่ยนแปลงไปอย่างรวดเร็ว

ในการติดต่อสื่อสารเพื่อรับส่งข้อมูลระหว่างแต่ละเครือข่ายทั้ง LAN และ WAN จำเป็นต้องใช้อุปกรณ์ที่มีความสามารถในการค้นหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ โดยอุปกรณ์แต่ละชนิดมีขีดความสามารถแตกต่างกันออกไป อุปกรณ์ที่นิยมใช้เป็นศูนย์กลางในการเชื่อมแต่ละเครือข่ายคือสวิตช์ ซึ่งเป็นอุปกรณ์ที่ติดต่อกันโดยใช้โปรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) เมื่อมีการต่อใช้งานระบบเครือข่ายผ่านอุปกรณ์สวิตช์โดยมีเส้นทางเดียวจากต้นทางไปยังปลายทาง ถ้าลิงค์ (Link) ระหว่างอุปกรณ์สวิตช์ทั้งสองตัวเกิดมีปัญหาขึ้นมา ก็ทำให้ระบบไม่สามารถใช้งานได้ จึงต้องมีการเพิ่มอุปกรณ์สวิตช์ขึ้นมาอีกหนึ่งตัว เพื่อให้มีเส้นทางสำรอง เมื่อลิงค์ระหว่างอุปกรณ์สวิตช์เดิมมีปัญหา ระบบจะยังสามารถใช้งานต่อได้ ผ่านไปทางอุปกรณ์สวิตช์อีกตัว ซึ่งหลักการดังกล่าวนั้นมักจะเกิดปัญหาในเรื่องของลูป (Loop) หรือสถานะที่อุปกรณ์ส่งข้อมูลออกไปแล้วเกิดการวนแบบไม่มีที่สิ้นสุด อาจเป็นผลมาจากการออกแบบระบบไม่ถูกหลักและอุปกรณ์ที่ไม่มีประสิทธิภาพเพียงพอ เมื่อมีลูปเกิดขึ้นจะทำให้เกิดปัญหาตามมา 3 อย่างหลักๆ คือ

1) Broadcast Storm คือการที่สวิตช์ทุกตัวส่ง Broadcast ออกมาทุกพอร์ตและทำแบบเดิมซ้ำไม่หยุด ทำให้อุปกรณ์สวิตช์ทุกตัวซึ่งต่อกันเป็นวงกลมหรือลูปต้องรับ Broadcast เหล่านั้นมาประมวลผลอย่างหลีกเลี่ยงไม่ได้ ทำให้เสียเนื้อที่และเวลาโดยไร้ประโยชน์ เมื่อทิ้งไว้อาจทำให้ระบบไม่สามารถใช้งานได้
2) Multiple Frame Transmissions คือการที่เครื่องปลายทางได้รับข้อมูลเฟรม (Frame) เข้ามาซ้ำทำให้เสียเวลาในการประมวลผล
3) MAC Database Instability คือการที่อุปกรณ์สวิตช์ได้รับ MAC address ของอุปกรณ์เดียว เข้ามาหลายทาง ทำให้อุปกรณ์สวิตช์เกิดการเรียนรู้ MAC address ที่ผิดเพี้ยนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นผู้จัดทำจึงใช้หลักการของ Redundancy โดยนำโปรโตคอลสแปนนิงทรี หรือ Spanning Tree Protocol (STP) เข้ามาช่วยในการป้องกันลูปที่เกิดขึ้นบน Layer 2 โดย STP จะทำการเปลี่ยนสถานะของพอร์ตหนึ่ง ให้เป็น Blocking เพื่อไม่ให้รับส่งข้อมูลได้ชั่วคราว จนกว่าเส้นทางหลักที่ใช้งานจะมีปัญหา และ STP มีระบบช่วยป้องกันไม่ให้ระบบหยุดการทำงาน ถ้าเส้นทางหนึ่งมีปัญหาก็สามารถเปลี่ยนไปใช้เส้นทางอื่นได้ จากการแลกเปลี่ยนเฟรมที่มีชื่อเรียกว่า BPDU (Bride Protocol Data Unit) ซึ่งเก็บค่าหลักของกลไกที่ใช้ในการเลือกพอร์ตที่ต้องการ Block เช่น Bridge ID (Priority, Mac address), Port ID และ Path Cost

นอกจากนี้ในขั้นตอนการพัฒนาระบบจำเป็นต้องมีการคอนฟิก (Configuration) สวิตช์ให้สามารถทำงานได้อย่างถูกต้อง ซึ่งการคอนฟิกแบบปกติต้องใช้เวลาในการศึกษาเรียนรู้และคอนฟิกค่อนข้างมาก ส่งผลให้เกิดความล่าช้าในการพัฒนาระบบ ผู้จัดทำจึงได้พัฒนาสคริปต์เพื่อช่วยในการคอนฟิกค่าในการคำนวณ STP ด้วยภาษาทีซีแอล หรือ Tool Command Language (Tcl) ขึ้น เนื่องจากภาษา Tcl เป็นภาษาสคริปต์ที่สามารถทำงานบนอุปกรณ์สวิตช์ได้ดี และมีความเสถียรภาพสูง อีกทั้งยังพัฒนาโปรแกรมด้วยภาษา Java เพื่อช่วยในการสร้างไฟล์สคริปต์เก็บคำสั่งเป็นภาษา Tcl จากการที่ผู้ใช้งานเพียงเก็บค่าที่ใช้ในการคำนวณ STP ลงในตารางเป็นไฟล์ .csv ซึ่งรองรับกับระบบเครือข่ายที่มีการใช้สวิตช์จำนวนมาก ทั้งนี้เพื่อช่วยลดระยะเวลาและเป็นการอำนวยความสะดวกแก่ผู้ใช้งานในกรณีที่ระบบเครือข่ายเกิดปัญหาลูป และต้องการใช้ STP เข้ามาช่วย

1.2 วัตถุประสงค์

1) เพื่อศึกษาและหาแนวทางแก้ไขปัญหาลูปที่เกิดขึ้นกับสวิตช์ในระบบเครือข่ายอินเทอร์เน็ต
2) เพื่อออกแบบระบบเครือข่ายสำรองในกรณีที่เส้นทางหลักเสียหายเนื่องจากเกิดลูปให้สามารถใช้งานได้ปกติ
3) เพื่อเพิ่มความสามารถด้านความพร้อมใช้งาน (Availability) ของระบบเครือข่าย
4) เพื่อพัฒนาสคริปต์ด้วยภาษา Tcl ช่วยในการคอนฟิกค่าที่ใช้ในการคำนวณ STP
5) เพื่อให้สคริปต์สามารถช่วยลดระยะเวลาในการคอนฟิกสวิตช์ได้
6) เพื่อพัฒนาโปรแกรมช่วยลดระยะเวลาในการคอนฟิกสวิตช์ได้

1.3 ขอบเขต

1) สามารถปรับปรุงรูปแบบระบบเครือข่ายของลูกค้า จากเดิมที่เกิดปัญหาเรื่องลูปได้ โดยการใช้หลักการ Redundancy แบบ STP (Spanning Tree Protocol)
2) อุปกรณ์สวิตช์สามารถติดต่อสื่อสารถึงกันได้ในขณะที่เกิดลูปขึ้น
3) อุปกรณ์สวิตช์สามารถเลือก Block พอร์ตที่มีปัญหาให้หยุดการรับส่งข้อมูลชั่วคราวได้
4) อุปกรณ์สวิตช์สามารถหาเส้นทางใหม่เมื่อเส้นทางหลักเสียหายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) สามารถคอนฟิกค่าที่ใช้คำนวณ STP ให้กับสวิตช์โดยใช้สคริปต์ได้อย่างถูกต้อง
6) โปรแกรมสามารถนำข้อมูลจากไฟล์ .csv มาแปลงเป็นคำสั่งโดยใช้ภาษา tcl ในการคอนฟิกสวิตช์แต่ละตัวได้ถูกต้อง
7) สามารถติดตั้งอุปกรณ์ระบบเครือข่าย ได้อย่างมีประสิทธิภาพ และถูกต้อง
8) สามารถสอนการใช้งานเบื้องต้น ของตัวอุปกรณ์สวติช์ รุ่น Cisco C2960X-48TS-LL ด้วยวิธีการ OJT (On the job training) ให้แก่ลูกค้า เพื่อให้ลูกค้าสามารถใช้งานตัวอุปกรณ์ในเบื้องต้นได้
9) สามารถทำเอกสารรายงานในส่วนต่างๆ ของการออกแบบระบบความปลอดภัยเครือข่าย รูปแบบการติดตั้ง และอธิบายการคอนฟิกในระบบเครือข่ายใหม่ ว่าสามารถแก้ไขให้มีประสิทธิภาพที่ดีกว่าเดิมอย่างไร เพื่อนำเสนอแก่ลูกค้า

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ระบบเครือข่ายสามารถทำงานได้อย่างมีประสิทธิภาพในขณะที่มีการต่อระบบเครือข่ายในรูปแบบวนลูป
2) ระบบเครือข่ายสามารถทำงานได้อย่างปกติในขณะที่เส้นทางหลักเสียหายเนื่องจากเกิดลูป
3) สามารถอำนวยความสะดวกให้ผู้ใช้งานสามารถนำสคริปต์ไปใช้ช่วยในการคอนฟิกได้โดยไม่ต้องเสียเวลาในการเรียนรู้มาก
4) สามารถลดระยะเวลาการดำเนินงานในการคอนฟิกอุปกรณ์สวิตช์โดยการใช้สคริปต์ได้
5) สามารถลดระยะเวลาการดำเนินงานโดยการใช้โปรแกรมในระบบเครือข่ายที่มีการใช้สวิตช์จำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

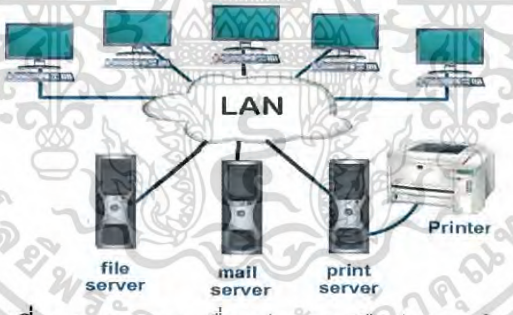
2.1 ระบบเครือข่าย

ระบบเครือข่าย คือ ระบบที่มีการนำคอมพิวเตอร์อย่างน้อยสองเครื่องมาเชื่อมต่อกัน โดยใช้สื่อกลาง เพื่อใช้ในการสื่อสารและเปลี่ยนข้อมูลถึงกันได้อย่างมีประสิทธิภาพ นอกจากนี้ยังสามารถใช้ทรัพยากรร่วมกันได้ ทำให้ประหยัดทรัพยากรในการใช้งาน

2.1.1 ชนิดของเครือข่าย

1) เครือข่ายภายใน หรือ Local Area Network

Local Area Network (LAN) [2] คือ ระบบเครือข่ายท้องถิ่นเป็นการเชื่อมต่อคอมพิวเตอร์และอุปกรณ์ต่าง ๆ ในระยะใกล้ ๆ ภายในสำนักงานหรืออาคารเดียวกันหรืออาคารที่อยู่ใกล้กัน เพื่อติดต่อสื่อสารใช้ทรัพยากรของระบบร่วมกัน การเชื่อมต่อมีทั้งระบบที่มีคอมพิวเตอร์ตั้งแต่ 2-3 เครื่องเชื่อมต่อกันผ่านอุปกรณ์กระจายสัญญาณตัวเดียวกัน จนถึงระบบขนาดใหญ่ที่มีการเชื่อมต่อที่ซับซ้อนมีอุปกรณ์เครือข่ายนับร้อยตัว

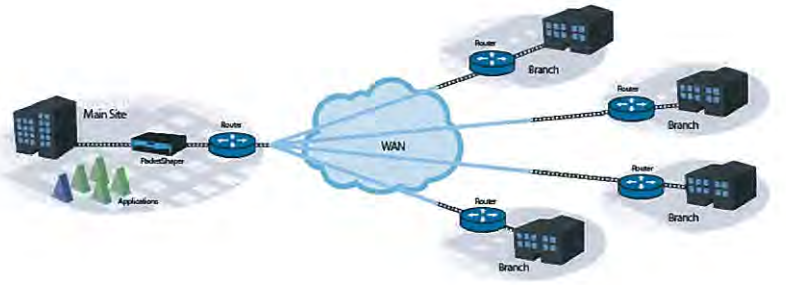


รูปที่ 2.1 แสดงการเชื่อมต่อของเครือข่ายภายใน

2) เครือข่ายวงกว้าง หรือ Wide Area Network

Wide Area Network (WAN) [2] คือ เครือข่ายงานที่อยู่ห่างไกลกันมาก อาจจะอยู่ระหว่างเมือง หรือระหว่างประเทศ เช่นการเชื่อมต่อเครือข่ายของสานักงานสาขาย่อยเข้ากับเครือข่ายของสานักงานใหญ่ ที่อยู่ห่างกันไกล อาจจะอยู่กันคนละที่หรือคนละเมืองกัน แต่ติดต่อกันด้วยระบบการสื่อสารทางไกลความเร็วสูง โดยการใช้การส่งสัญญาณ ผ่านดาวเทียมเพื่อเชื่อมโยงคอมพิวเตอร์ให้ติดต่อถึงกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการเชื่อมต่อของเครือข่ายวงกว้าง

2.1.2 เน็ตเวิร์คโปรโตคอล

การที่เครื่องคอมพิวเตอร์ที่ถูกเชื่อมโยงกันไว้ในระบบ จะสามารถติดต่อสื่อสารกันได้นั้น จำเป็นจะต้องมีภาษาสื่อสารที่เรียกว่า โปรโตคอล (Protocol)

1) TCP/IP

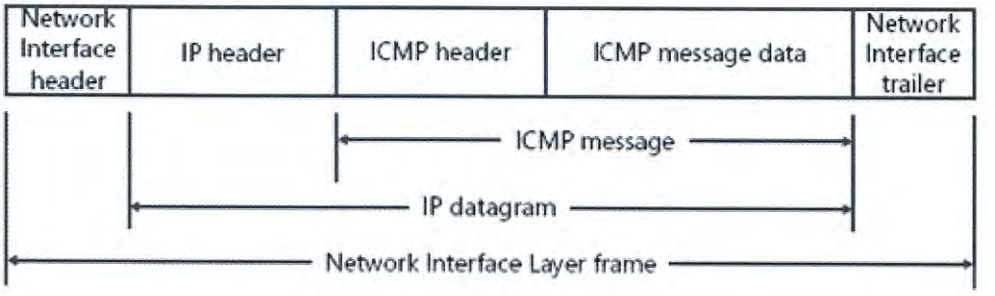
TCP/IP (Transmission Control Protocol/Internet Protocol) [1] โดย TCP จะทำหน้าที่ในการแยกข้อมูลเป็นส่วนๆ หรือ Package ส่งออกไป TCP ปลายทางก็จะทำการรวบรวมข้อมูลแต่ละส่วนเข้าด้วยกัน โดยระหว่างการรับส่งข้อมูลนั้นก็จะมีการตรวจสอบความถูกต้อง ถ้าเกิดผิดพลาด TCP ปลายทางก็จะขอไปยัง TCP ต้นทางให้ส่งข้อมูลมาใหม่ ส่วน IP จะทำหน้าที่ในการจัดส่งข้อมูลจากเครื่องต้นทางไปยังเครื่องปลายทางโดยอาศัย IP Address

2) Internet Control Message Protocol

Internet Control Message Protocol (ICMP) [6] เป็นโปรโตคอลที่สามารถทำหน้าที่ตรวจสอบการเชื่อมต่อ โดยการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งปัญหาส่วนมากที่พบคือ ส่งไปไม่ได้ หรือปลายทางรับข้อมูลไม่ได้เป็นต้น นอกจากนี้ โปรโตคอล ICMP ยังถูกเรียกใช้งานจากเครื่องเซิฟเวอร์ (Server) และเราเตอร์อีกด้วย เพื่อแลกเปลี่ยนข้อมูลที่ใช้ควบคุม ส่วนรูปแบบการทำงานของโปรโตคอล ICMP นั้นจะทำควบคู่กับโปรโตคอล IP ในระบบเดียวกัน และข้อความ ต่างๆที่แจ้งให้ทราบจะถูกผนึกอยู่ภายในข้อมูล IP (IP datagram) อีกทีหนึ่ง ข้อความที่โปรโตคอล ICMP ส่งนั้น แบ่งออกได้เป็น 2 แบบ คือ ICMP Error Message หรือข้อความแจ้งข้อผิดพลาด และ ICMP Query หรือข้อความเรียกขอข้อมูลเพิ่ม ตัวอย่างกลไกการทำงานของโปรโตคอล ICMP เช่น เมื่อมีการส่งผ่านข้อมูลจากผู้ใช้ไปยังปลายทางที่ไม่ถูกต้อง หรือขณะนั้นเครื่องปลายทางเกิดปัญหาจนไม่สามารถรับข้อมูลได้ ที่ Router จะส่งข้อความแจ้งเป็น ICMP Message ที่ชื่อ Destination Unreachable ให้กับผู้ส่งข้อมูลนั้น นอกจากนี้ตัวข้อมูลที่แจ้งข้อความ ก็จะมีส่วนของข้อมูล IP Datagram ที่เกิดปัญหาด้วย ดังนั้นเมื่อผู้ส่งข้อมูลได้รับข้อความแจ้งแล้วก็จะทราบได้ว่าจุดที่เกิดปัญหานั้นอยู่ที่ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

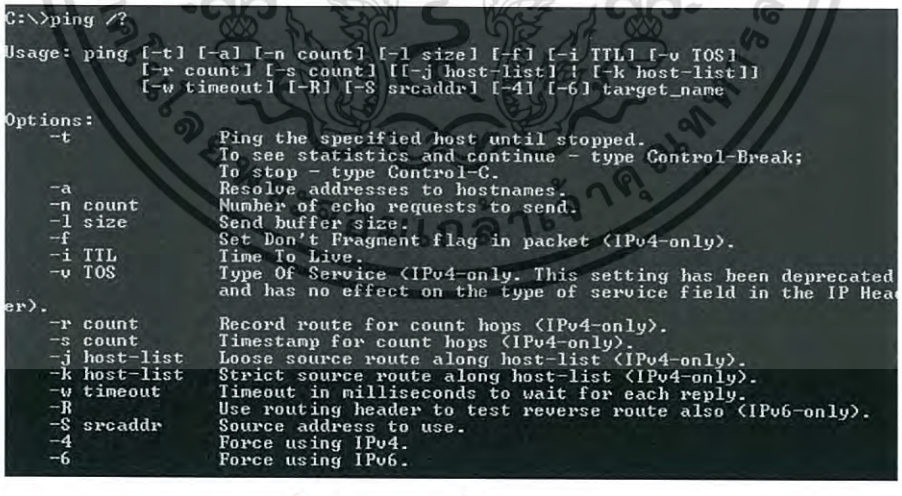
ICMP messages จะประกอบด้วยส่วนของ ICMP header และ ICMP messages ที่ถูกห่อหุ้มไว้ในส่วนของ ICMP header แสดงดังรูปที่ 2.3



รูปที่ 2.3 แสดงเฟรมของ ICMP Message

ดังนั้นโปรโตคอล ICMP จึงกลายมาเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง ping ที่มักใช้ทดสอบว่าเครื่องเซิฟเวอร์ที่ให้บริการหรืออุปกรณ์ที่ต่ออยู่ในระบบเครือข่ายนั้นยังทำงานเป็นปกติหรือไม่ โดย ping จะบอกว่าใช้เวลาไปเท่าไหร่ในการส่ง ในกรณีที่ไม่มีการตอบกลับจากอุปกรณ์เน็ตเวิร์ค ก็จะแสดง Error ให้เห็น

การ ping สามารถทำในเครื่องคอมพิวเตอร์ได้ โดยผ่าน Command Prompt คำสั่งในการ ping สามารถตรวจสอบได้จากคำสั่ง **ping /?** เพื่อเป็นการแสดงรายละเอียดในการเรียกดูว่าต้องการใช้คำสั่ง ping ในรูปแบบไหน ดังรูปที่ 2.4



```
C:\>ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -t                  Ping the specified host until stopped.
                        To see statistics and continue - type Control-Break;
                        To stop - type Control-C.
    -a                  Resolve addresses to hostnames.
    -n count            Number of echo requests to send.
    -l size             Send buffer size.
    -f                  Set Don't Fragment flag in packet (IPv4-only).
    -i TTL              Time To Live.
    -v TOS              Type Of Service (IPv4-only. This setting has been deprecated
                        and has no effect on the type of service field in the IP Hea
er).
    -r count            Record route for count hops (IPv4-only).
    -s count            Timestamp for count hops (IPv4-only).
    -j host-list        Loose source route along host-list (IPv4-only).
    -k host-list        Strict source route along host-list (IPv4-only).
    -w timeout          Timeout in milliseconds to wait for each reply.
    -R                  Use routing header to test reverse route also (IPv6-only).
    -S srcaddr          Source address to use.
    -4                  Force using IPv4.
    -6                  Force using IPv6.
```

รูปที่ 2.4 แสดงคำสั่งในการ ping

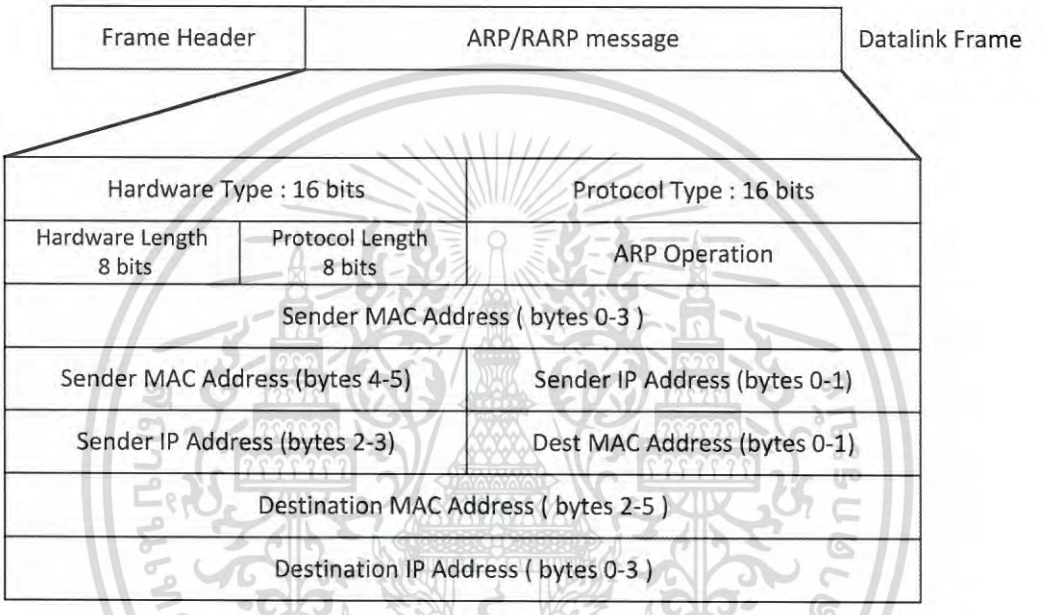
คำสั่งในการ ping เพื่อทดสอบการเชื่อมต่อจะใช้คำสั่ง

ping หมายเลข IP Address,Network Address,Dns เช่น ping www.google.com, ping 8.8.8.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Address Resolution Protocol

Address Resolution Protocol (ARP) [5] เป็นโปรโตคอลที่ใช้ในการสื่อสาร ทำหน้าที่ในการจับคู่ระหว่าง IP Address ทาง Logocal กับ IP Address ทาง Physical ทั้งนี้เนื่องจากระบบของการส่งข้อมูลในระบบ IP นั้น เป็นระบบที่ไม่ขึ้นกับ Hardware ใดๆ ทำให้เมื่อระบบ IP ต้องการส่งข้อมูล จะต้องร้องขอบริการจากระดับชั้น Data link แต่เนื่องจากระดับชั้น Data link ไม่รู้จัก Address ในระบบ IP ดังนั้นระบบ IP จึงต้องทำการหา Address ที่ระดับชั้น Data link รู้จัก ซึ่งก็คือ Hardware Address



รูปที่ 2.5 แสดงลักษณะของ Data Link เฟรม

กลการทำงานของ ARP มี 2 ขั้นตอนคือ

1. เครื่องที่ต้องการสอบถาม MAC address ส่ง ARP Packet เรียกว่า ARP Request ซึ่งบรรจุ IP, MAC address ของตนเอง และ IP Address ของเครื่องที่ต้องการทราบ MAC address ส่วน MAC address ปลายทางนั้น จะถูกกำหนดเป็น FF:FF:FF:FF:FF:FF ซึ่งเป็น Broardcast Address เพื่อให้ ARP Packet ถูกส่งไปยังเครื่องทุกเครื่องที่อยู่ในเครือข่ายเดียวกัน

2. เฉพาะเครื่องที่มี IP Address ตรงกับที่ระบุใน ARP Packet จะตอบกลับมาด้วย ARP Packet เช่นกัน โดยใส่ MAC address และ IP Address ของตนเองเป็นผู้ส่ง และใส่ MAC address และ IP address ของเครื่องที่ส่งมาเป็นผู้รับ Packet ที่ตอบกลับนี้เรียกว่า ARP Reply

นอกเหนือจากโปรโตคอล ARP แล้วยังมีอีกโปรโตคอลหนึ่งที่ถือว่าเป็นโปรโตคอลคู่แฝดของ ARP โดยจะมีการทำงานที่ย้อนกลับกันกับโปรโตคอล ARP ดังนั้นจึงมีชื่อว่า RARP (Reverse ARP) โดยรูปแบบเฟรมของ ARP และ RARP จะมีลักษณะเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) Trivial File Transfer Protocol

Trivial File Transfer Protocol (TFTP) [1] เป็นกระบวนการรับส่งไฟล์ที่ใช้กลไกการสื่อสารแบบ User Datagram Protocol (UDP) ซึ่งเป็นโปรโตคอลที่ทำงานแบบ Connectionless ซึ่งผู้ใช้ไม่จำเป็นต้องใส่รหัสหรือ Password แต่จะทำได้เพียงโอนข้อมูลที่จัดเตรียมไว้แล้วเท่านั้น แต่จะไม่มีฟังก์ชั่นอื่นๆ เช่น การแสดงรายชื่อไฟล์, การเปลี่ยนไดเร็คทอรี เป็นต้น

การทำงานของ TFTP ไม่ซับซ้อนยุ่งยาก ดังนั้นโปรแกรมที่ใช้งานจะมีขนาดเล็ก ใช้เนื้อที่ในหน่วยความจำน้อย สามารถบรรจุโปรแกรมลงในชิพประเภทที่เป็น Programmable Read-Only Memory (PROM) เพื่อนำไปใช้งานในเครื่องที่ใช้พกพาหรืออุปกรณ์เราเตอร์และสวิตช์ได้ง่าย TFTP สามารถนำมาใช้ในการถ่ายโอนไฟล์ระหว่างคอมพิวเตอร์กับเราเตอร์และสวิตช์ได้ โดยไฟล์ที่ทำการถ่ายโอนนั้นอาจจะเป็น Configuration file ที่มีผลต่อการทำงานของอุปกรณ์จริง นั่นหมายถึงว่าสามารถถ่ายโอน Configuration file จากอุปกรณ์เราเตอร์และสวิตช์มาไว้ในเครื่องคอมพิวเตอร์ได้ และสามารถนำไฟล์ดังกล่าวนี้ส่งไปยังเราเตอร์และสวิตช์ เพื่อให้ทำงานตามคอนฟิกไฟล์ได้ ซึ่งจะใช้วิธีการ copy จาก Command Prompt (จากหน้าต่างการ Telnet หรือจากหน้าต่างการคอนฟิกด้วยสาย Console) แล้วนำมาเก็บไว้ในแฟลช (Flash) ของอุปกรณ์

2.2 อุปกรณ์

2.2.1 สวิตช์

สวิตช์ (Switch) [1] เป็นอุปกรณ์ที่ทำหน้าที่เชื่อมต่อกับคอมพิวเตอร์ หรืออุปกรณ์ทำงาน Network อื่นๆ เพื่อให้อุปกรณ์นั้นๆ สามารถเชื่อมต่อกันได้ โดยการทำงานของสวิตช์จะส่งข้อมูลออกไปเฉพาะพอร์ตที่ใช้ในการติดต่อกับเครื่องคอมพิวเตอร์ปลายทางเท่านั้น ไม่ส่งกระจายข้อมูลไปยังทุกพอร์ต ทำให้ในสวิตช์ไม่มีปัญหาการชนของข้อมูล สวิตช์จะทำงานอยู่ในชั้น Data link layer คือจะรับผิดชอบในการเชื่อมโยงของข้อมูล ตรวจสอบความถูกต้องและความสมบูรณ์ของการรับส่งข้อมูล สำหรับในชั้นเชื่อมโยงข้อมูลนั่นจะทำการแบ่งข้อมูลระดับบิตที่ได้รับจากชั้น Physical layer เป็นข้อมูลชนิดที่เรียกว่าเฟรมก่อนจะส่งไปยังชั้นถัดไป คือ Network layer

Layer 3 Switch จะสามารถทำงานได้ในทั้งระดับของ Layer 2 และ Layer 3 โดยถ้าเป็นการส่งข้อมูลกันในระดับ Layer 2 จะพิจารณาจาก Mac address เหมือนเดิม แต่ถ้าเป็นการติดต่อกันในระดับ Layer 3 จะพิจารณา IP Address เป็นหลัก สำหรับข้อมูล ที่ Layer 3 ส่งต่อออกมานั้น ถ้าทำงานในระดับของ Layer 2 จะส่งข้อมูลเป็นเฟรม ถ้าทำงานในระดับ Layer 3 จะส่งข้อมูลเป็น Packet นอกจากนี้ Layer 3 มีความสามารถด้านการ Routing เหมือนกับเราเตอร์ โดยจะต่างกับเราเตอร์ที่ไม่กันการส่ง Broadcast ข้ามเครือข่าย

สวิตช์จะมีอยู่หลายแบบขึ้นอยู่กับลักษณะการใช้งานโดยจะมีพอร์ตให้เลือกใช้จะแยกเป็นพอร์ตที่เป็น Fast Ethernet จะมีความเร็วในการรับส่งข้อมูลอยู่ที่ 100 Mbps และพอร์ตที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

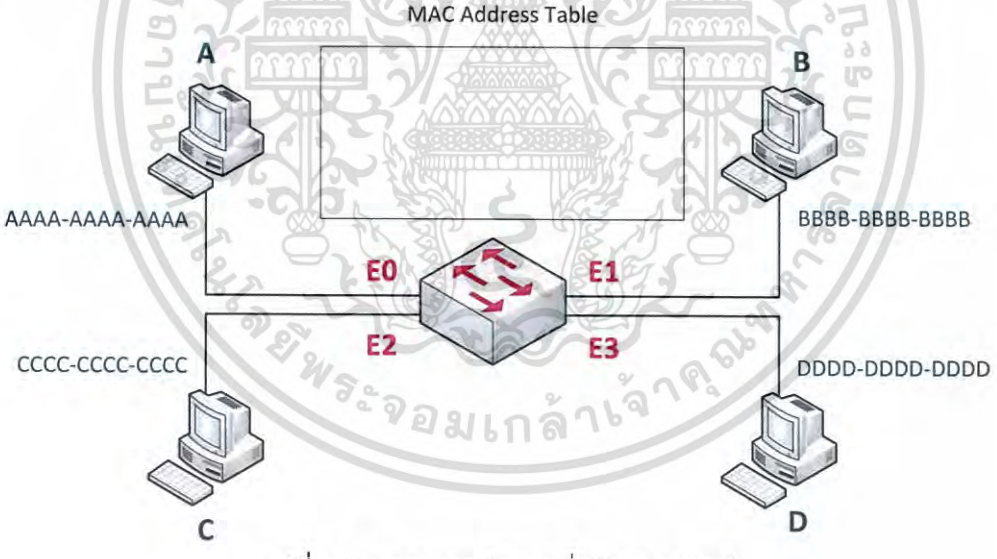
Gigabit Ethernet มีความเร็วในการรับส่งข้อมูลอยู่ที่ 1 Gbps และอุปกรณ์สวิตช์แต่ละตัวมีจำนวนพอร์ตไม่เท่ากัน ซึ่งมีตั้งแต่ 8 16 24 48 พอร์ต

การเชื่อมต่ออุปกรณ์สวิตช์เข้ากับอุปกรณ์สวิตช์ด้วยกัน จำเป็นต้องเชื่อมต่อผ่านพอร์ต Uplink ซึ่งถ้าหากพอร์ต Uplink ของอุปกรณ์มีไม่เพียงพอ สามารถเชื่อมต่อผ่านพอร์ตปกติโดยการใช้สาย Cross ในการเชื่อมต่อ

หลักการทำงานของสวิตช์

ในกรณีของสวิตช์ Layer 2 พอร์ตทุกพอร์ตจะเป็นคนละ “Collision Domain” คือเครื่องคอมพิวเตอร์แต่ละเครื่องที่เชื่อมต่อเข้าพอร์ตจะได้รับแบนด์วิธ (Bandwidth) ในการใช้งานไปเต็มๆ เช่น 10 Mbps หรือ 100 Mbps โดยที่ไม่ต้องแชร์ร่วมกับใคร เมื่อสวิตช์ได้รับเฟรมมาหนึ่งเฟรม สวิตช์จะพิจารณาว่าจะส่งออกไปทางพอร์ตไหน ซึ่งจะตรวจสอบจากตาราง MAC address Table โดยภายในตารางนี้จะประกอบไปด้วยการจับคู่ความสัมพันธ์ระหว่างหมายเลข MAC address ของเครื่องคอมพิวเตอร์ที่เชื่อมต่ออยู่กับพอร์ตของสวิตช์กับหมายเลขพอร์ตนั้นๆ

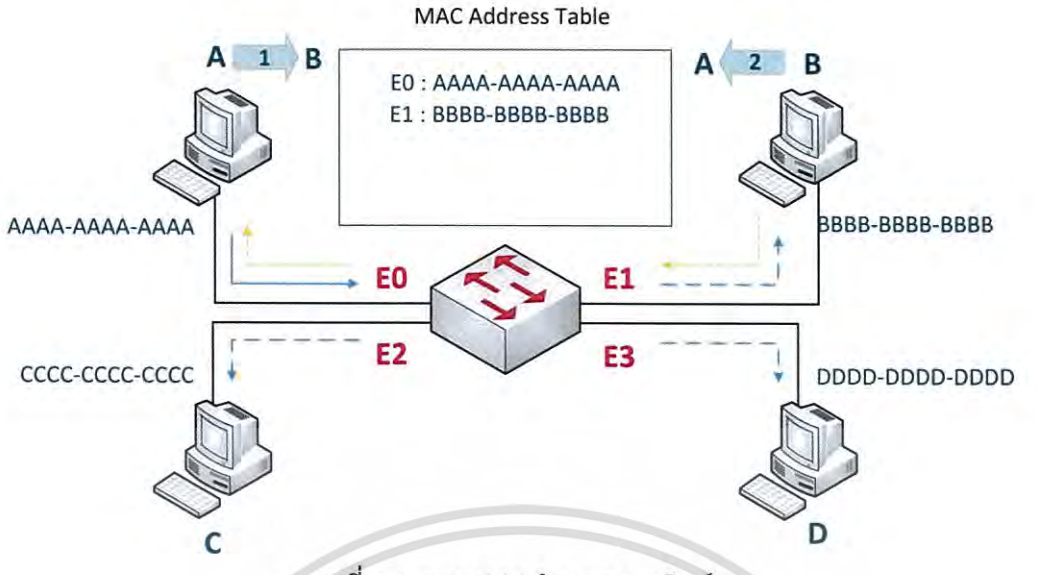
ตารางดังกล่าวไม่ปรากฏขึ้นมาเลยหลังจากติดตั้งสวิตช์ แต่สวิตช์จะคอยเรียนรู้และเก็บข้อมูลภายในเวลาอันรวดเร็ว (กระบวนการนี้เรียกว่าการเรียนรู้แอดเดรสของสวิตช์ หรือ Address Learning)



รูปที่ 2.6 แสดงการทำงานเริ่มต้นของสวิตช์

จากรูปที่ 2.6 เมื่อสวิตช์ได้รับการเปิด power on ขึ้นมาใหม่ จะไม่มีข้อมูลใดเลยในตาราง MAC address Table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงการทำงานของสวิตช์

จากรูปที่ 2.7 เครื่อง A ส่งเฟรมไปหาเครื่อง B เมื่อสวิตช์ได้รับเฟรมเข้ามาทางพอร์ต E0 สวิตช์จะอ่านเฮดเดอร์ (Header) ของเฟรมเพื่อเรียนรู้หมายเลข MAC address ต้นทาง (Source MAC address) ของเฟรมที่ได้รับเข้ามา และทำการจับคู่ความสัมพันธ์ระหว่างหมายเลข MAC Address ดังกล่าวซึ่งคือ MAC address ของเครื่อง A (AAAA-AAAA-AAAA) ไว้คู่กับพอร์ต E0 และเนื่องจากยังไม่ทราบว่าต้องส่งเฟรมออกไปทางพอร์ตใดเพื่อให้ไปถึงเครื่อง B สวิตช์จึงใช้วิธีการส่งแบบ “FLOOD” คือการกระจายออกไปที่ทุกพอร์ต ยกเว้นพอร์ตที่ได้รับเฟรมเข้ามาคือพอร์ต E0

สิ่งที่ถูกบรรจุเข้าไปในตาราง MAC address Table ในขณะนี้คือ ความสัมพันธ์ระหว่างหมายเลขพอร์ต E0 คู่กับหมายเลข MAC address ของเครื่อง A และในขณะเดียวกัน เมื่อเครื่อง B มีการส่งเฟรมตอบกลับมา สวิตช์จะเรียนรู้ด้วยว่าเฟรมที่มีหมายเลข MAC address ต้นทางเป็นของเครื่อง B ได้รับเข้ามาทางพอร์ต E1 และจะนำมาจับคู่ความสัมพันธ์ระหว่างหมายเลขพอร์ต E1 คู่กับหมายเลข MAC address ของ B ไว้ด้วย

ไม่ว่าสวิตช์จะมีกี่พอร์ตและมีกี่เครื่องคอมพิวเตอร์ที่เชื่อมต่ออยู่กับพอร์ตของมัน เมื่อสวิตช์ทำเช่นนี้ไปเรื่อยๆ จะได้ตารางที่เก็บความสัมพันธ์ระหว่างหมายเลขพอร์ตกับหมายเลข MAC address ไว้อย่างสมบูรณ์ ตารางนี้ช่วยให้สวิตช์สามารถตัดสินใจได้ว่า เมื่อได้รับเฟรมเฟรมหนึ่งเข้ามาแล้วจะต้องส่งเฟรมออกไปทางพอร์ตไหน โดยการนำเอา MAC address ปลายทางของเฟรมไปค้นดูในตาราง เมื่อพบแล้ว สวิตช์จะส่งผ่าน (Forward) เฟรมออกไปทางพอร์ตหมายเลขนั้นเพียงพอร์ตเดียว โดยไม่ส่งออกไปทางพอร์ตอื่น (กระบวนการนี้เรียกว่า การส่งผ่าน / การฟิลเตอร์เฟรม หรือ Forwarding / Filtering)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

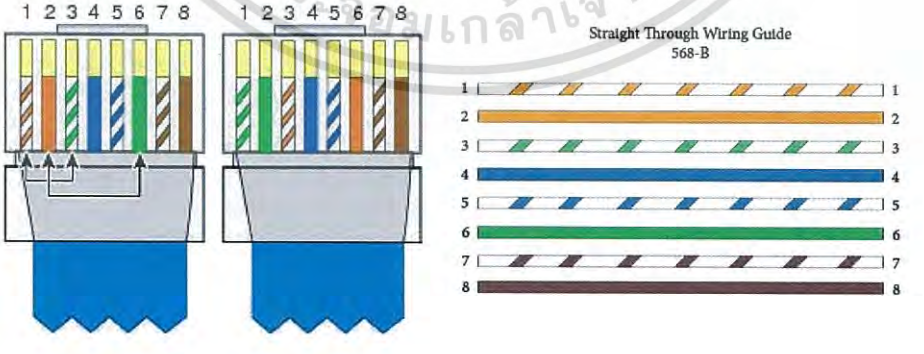
2.2.3 สาย UTP

สาย UTP (Unshield Twisted Pair) [1] เป็นสายที่นิยมใช้อย่างกว้างขวางในระบบเครือข่ายที่เป็น LAN โดยสามารถใช้เพื่อรับส่งข้อมูลข่าวสารทั้งที่เป็น Voice และ Data ทั้งที่มีความเร็วต่ำและความเร็วสูง สาย UTP ใช้ต่อกับคอมพิวเตอร์และอุปกรณ์เชื่อมต่อเครือข่ายอย่างอุปกรณ์สวิตช์ หรือ ฮับ และสายแลนก็ใช้ต่อกับโมเด็มเราเตอร์เพื่อเชื่อมต่อสัญญาณอินเตอร์เน็ตได้อีกด้วย

สาย UTP ประกอบด้วยคู่สายตีเกลียวจำนวนหลายคู่เพื่อลดสัญญาณรบกวน แต่สำหรับการสื่อสารในระบบเครือข่าย LAN มีมาตรฐานกำหนดไว้ว่าให้มีเพียง 4 คู่เท่านั้น
แบ่งประเภทของสาย UTP ตามคุณภาพความถี่ที่รองรับได้ดังนี้

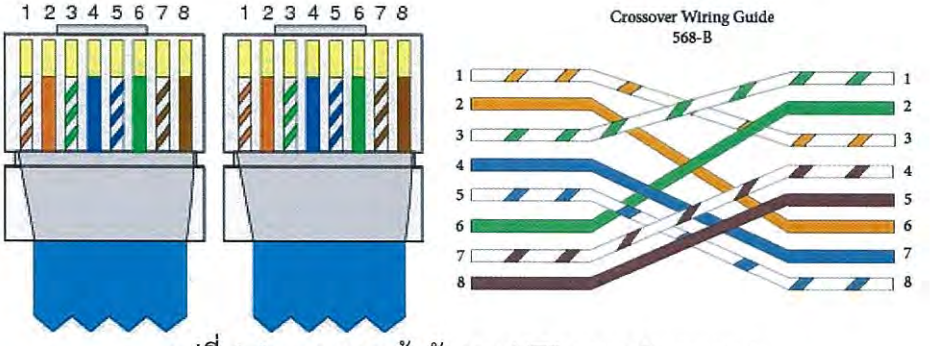
1) ประเภทที่หนึ่ง คือ UTP UTP CAT5 คือ สายแลน ที่เป็นสายทองแดงมีความเร็วต่ำ ความเร็วสูงสุดอยู่ที่ 100 Mbps สายชนิดนี้ไม่เป็นที่นิยมใช้งาน เนื่องด้วยความเร็วในการถ่ายโอนข้อมูลที่ต่ำ
2) ประเภทที่สอง UTP CAT5e คือ สายแลนที่เป็นสายทองแดงที่มีความเร็วที่ต่ำ ความเร็วสูงสุดอยู่ที่ 1 Gpbs
3) ประเภทที่สาม UTP CAT6 คือ สายแลนที่เป็นสายทองแดงที่มีความเร็วที่ต่ำ ความเร็วสูงสุดอยู่ที่ 10 Gpbs แบนด์วิธอยู่ที่ 250 MHz
4) ประเภทที่สี่ UTP CAT7 คือ สายแลนที่เป็นสายทองแดงที่มีความเร็วที่ต่ำ ความเร็วสูงสุดอยู่ที่ 10 Gpbs แบนด์วิธอยู่ที่ 600 MHz

การเข้าหัวของสาย UTP สามารถแบ่งได้ตามการเชื่อมต่ออุปกรณ์แต่ละประเภท สามารถแบ่งได้เป็น Straight Through หรือ สายตรง และ Crossover หรือสายครอส



รูปที่ 2.8 แสดงการเข้าหัวสาย UTP แบบ Straight Through

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงการเข้าหัวสาย UTP แบบ Crossover

2.2.1 การเชื่อมต่ออุปกรณ์

1) Medium Dependent Interface (MDI)

คือชนิดของ Ethernet port ซึ่งจะถูกใช้อยู่บน Network Interface Card (NIC) หรือ Card LAN ซึ่ง Card LAN ถูกเสียบอยู่บนคอมพิวเตอร์ รวมถึง Ethernet port บนอุปกรณ์เราเตอร์ก็เป็นชนิด MDI เช่นกัน

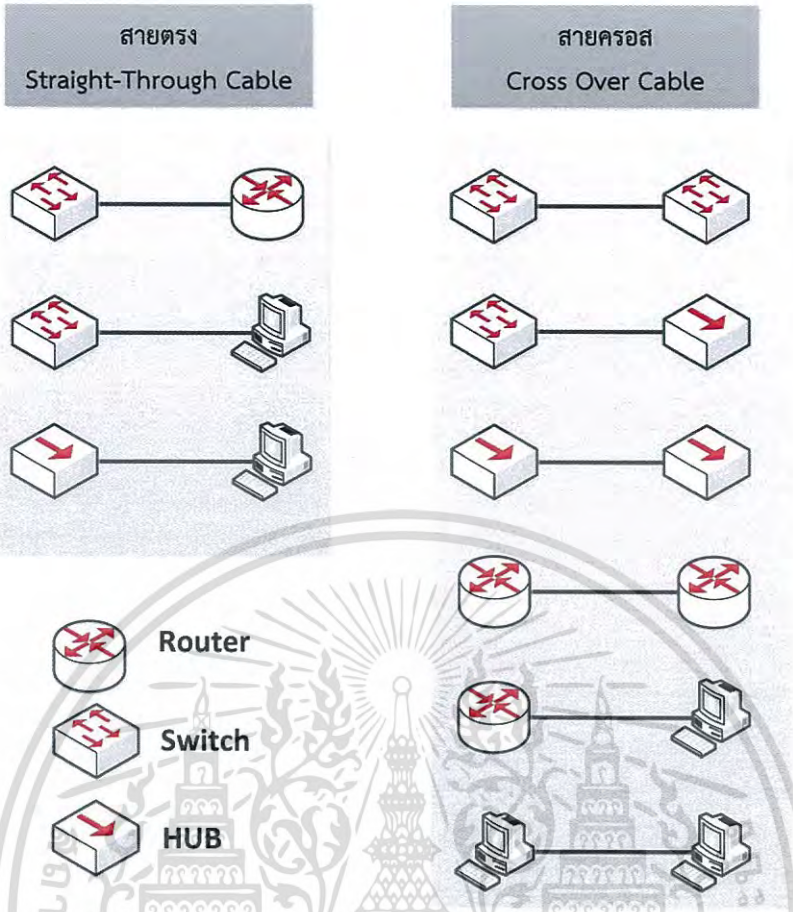
2) Medium Dependent Interface Crossover (MDI-X)

คือชนิดของ Ethernet port ที่อยู่บนอุปกรณ์ฮับ และสวิตช์โดยตัวอักษร X เป็นการแทนคำว่า Crossover

ดังนั้นการเชื่อมต่อของแต่ละอุปกรณ์เข้าด้วยกันจึงต้องคำนึงถึงชนิดของ Ethernet port ในแต่ละอุปกรณ์ และประเภทของสาย UTP ที่นำมาเชื่อมต่อ โดยสามารถจำแนกได้ดังนี้

- MDI ต่อกับ MDI เป็นชนิดเดียวกันใช้สายครอส (Crossover cable)
- MDI-X ต่อกับ MDI-X เป็นชนิดเดียวกันใช้สายครอส (Crossover cable)
- MDI ต่อกับ MDI-X เป็นคนละชนิดกันใช้สายตรง (Straight-Through Cable)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



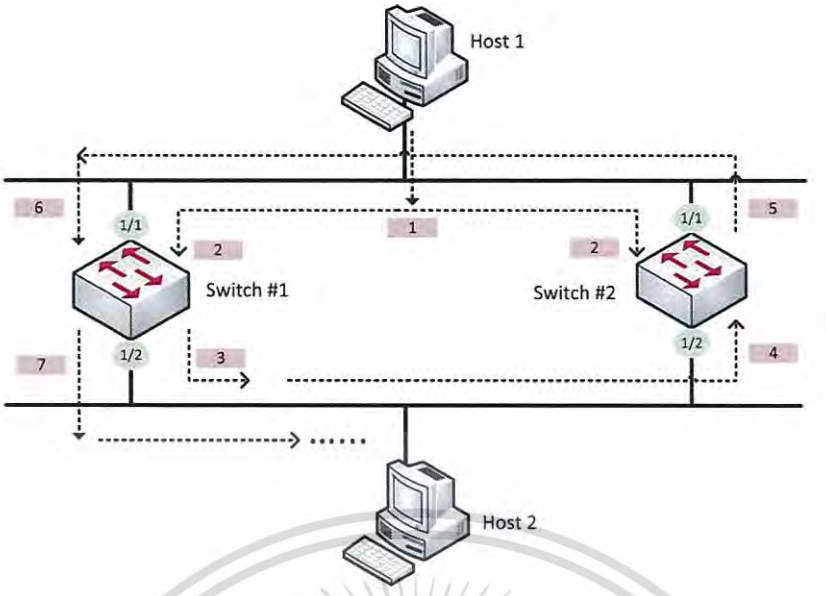
รูปที่ 2.10 แสดงประเภทการเชื่อมต่อของอุปกรณ์เน็ตเวิร์ค

2.3 ลูป (Loop)

การเพิ่มอุปกรณ์สวิตช์เข้าไปนั้นเป็นเสมือนการเพิ่มเส้นทางสำรอง แม้จะมีข้อดีแต่มันก็ก่อให้เกิดผลกระทบข้างเคียงตามมา คือการเกิดลูป โดยการเกิดลูปคือสถานะที่อุปกรณ์ส่งข้อมูลออกไปแล้วเกิดการวนแบบไม่มีที่สิ้นสุด เมื่อมีลูปเกิดขึ้นจะทำให้เกิดปัญหาตามมา 3 อย่างหลักๆ คือ

1) Broadcast Storm คือการที่สวิตช์ทุกตัวส่ง Broadcast ออกมาทุกพอร์ตและทำแบบเดิมซ้ำไม่หยุด ทำให้สวิตช์ทุกตัวซึ่งต่อกันเป็นวงกลมหรือลูปต้องรับ Broadcast เหล่านั้นมาประมวลผลอย่างหลีกเลี่ยงไม่ได้ ทำให้เสียเนื้อที่และเวลาโดยไร้ประโยชน์ เมื่อทิ้งไว้อาจทำให้ระบบใช้งานไม่ได้ ขั้นตอนการเกิดลูปสามารถอธิบายได้ดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงตัวอย่างการเกิดลูป

1) Host A เริ่มต้นส่งเฟรมแบบ Boardcast (เฟรมที่มี MAC address ปลายทางเป็น FF-FF-FF-FF-FF-FF) Switch#1 และSwitch#2 จึงได้รับเฟรมเข้าไปทางพอร์ต 1/1
2) เมื่อสวิตช์ได้รับเฟรม Boardcast มา นั่นคือการ Flood หรือการส่งผ่านเฟรมออกไปทุกพอร์ต (ยกเว้น พอร์ตที่รับเข้ามา) เฟรมนี้จึงถูกส่งออกทางพอร์ต 1/2
3) เฟรมจากสวิตช์ 1 ถูกส่งต่อไปยังพอร์ต 1/2 ของสวิตช์ 2
4) เมื่อสวิตช์ 2 ได้รับเฟรม Boardcast เข้ามาก็จะทำการส่งต่อออกทุกพอร์ตเช่นกัน Switch#2 จึงส่งต่อไปที่พอร์ต 1/1
5) เฟรม Boardcast จะถูกส่งต่อเข้าทางพอร์ต 1/1 ของ Switch#1 อีกรอบ และวนซ้ำไปเรื่อยๆ

2) Multiple Frame Transmissions คือการที่เครื่องปลายทางได้รับเฟรมข้อมูลเข้ามาซ้ำ ในตัวอย่างจะเกิดที่จากเฟรมที่ส่งจาก Host A ไปยัง Host B จะถูกส่งไป 2 เฟรม แทนที่จะส่งไปเฟรมเดียว ซึ่งส่งผลให้เสียเวลาในการประมวลผลและสิ้นเปลืองแบนด์วิดธ์

3) MAC Database Instability คือการที่อุปกรณ์สวิตช์ได้รับ MAC address เดียวเข้ามาหลายทาง ทำให้อุปกรณ์สวิตช์เกิดการเรียนรู้ MAC address ที่ผิดเพี้ยนไปในการส่งต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Redundancy

2.3.1 โปรโตคอลสแปนนิงทรี

โปรโตคอลสแปนนิงทรี [1] หรือ Spanning Tree Protocol (STP) เป็นเทคโนโลยีที่ช่วยป้องกันการเกิดลูปในสวิตช์ และช่วยเสริมให้มีเส้นทางสำรองใช้ในกรณีที่พอร์ตเสีย โดยจะทำการ Block พอร์ตที่มีความเสี่ยงสูงไม่ให้รับส่งเฟรมไว้ชั่วคราว ซึ่งพอร์ตนั้นจะอยู่ในสถานะ Blocking state จนกว่าจะมีการเปลี่ยนแปลง โทโพโลยีในระบบ จึงจะมีการคำนวณใหม่ว่าจะให้ยกเลิกการ Block ออกไป สถานะจะเปลี่ยนเป็น Forwarding state

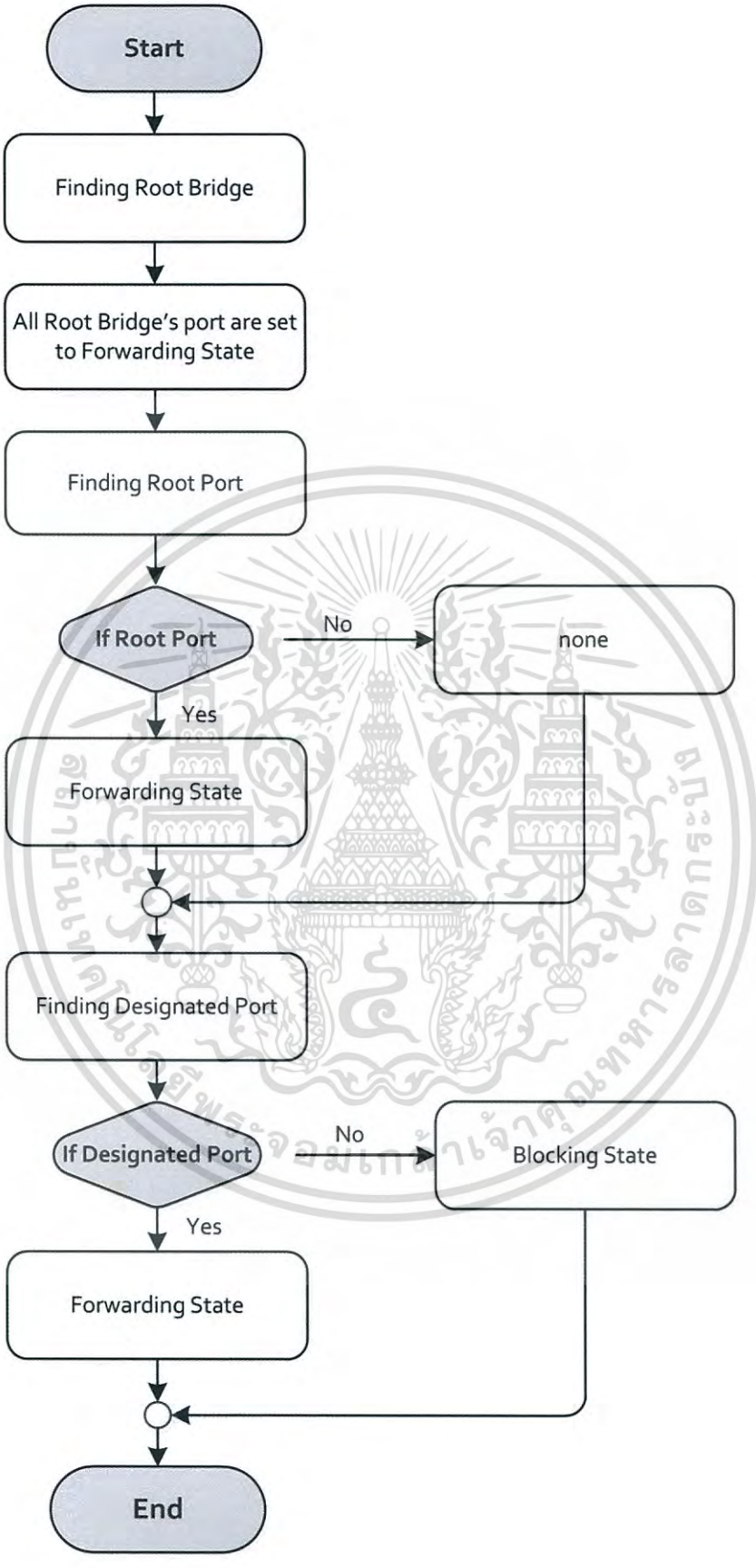
จุดประสงค์หลักของ STP คือ การสร้างเส้นทางเพียงเส้นทางเดียวที่ปราศจากลูป โดยเริ่มต้นจากสวิตช์ที่ทำหน้าที่เป็นศูนย์กลางหลัก หรือ Root Bridge ผ่านไปยังสวิตช์แต่ละตัวที่เชื่อมต่ออยู่ในลักษณะคล้ายกับโครงสร้างต้นไม้ที่มีจุดศูนย์กลางคือ ราก และผ่านลำต้นกระจายออกไปเป็นกิ่งและใบตามลำดับ

หลักการทำงานของโปรโตคอลสแปนนิงทรี

STP ถือเป็นมาตรฐานกลางที่ทางองค์กร IEEE ได้กำหนดนิยามไว้ในมาตรฐาน 802.1D มีเงื่อนไขในการพิจารณาว่าพอร์ตใดจะอยู่ในสถานะ Forwarding State ดังนี้

1) STP จะเลือก Root Bridge โดยพอร์ตทุกพอร์ตบนสวิตช์ตัวที่ถูกเลือกเป็น Root Bridge ถูกกำหนดให้อยู่ในสถานะ Forwarding State
2) สวิตช์ที่เหลือจะพิจารณาเลือกหนึ่งพอร์ตจากพอร์ตที่มีอยู่ทั้งหมด ให้ทำหน้าที่เป็น Root Port โดยพอร์ตที่เป็น Root Port จะต้องมีค่า Path Cost จากพอร์ตนั้นๆไปยัง Root Bridge น้อยที่สุดเมื่อเทียบกับพอร์ตอื่นๆบนสวิตช์ตัวเดียวกัน เพราะฉะนั้นจะมี Root Port 1 พอร์ต บนสวิตช์ 1 ตัว และมีสถานะเป็น Forwarding State
3) บนแต่ละเส้นทางของการเชื่อมต่อระหว่างสวิตช์ โดยปกติจะมีสวิตช์อย่างน้อยสองตัวต่อกัน จึงจะต้องมีการเลือกพอร์ตของสวิตช์ตัวใดตัวหนึ่งเป็น Designated Port โดยจะต้องเป็นพอร์ตที่มีค่า Path Cost จากสวิตช์ตัวนั้นๆไป Root Bridge ต่ำที่สุด และจะมีสถานะเป็น Forwarding
4) พอร์ตที่เหลือจะถูกกำหนดให้อยู่ในสถานะ Blocking State

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 Data Flow Diagram แสดงการทำงานของ STP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเลือกว่าพอร์ตหรือสวิตช์ตัวไหนจะถูกเลือกเป็นอะไรนั้น ขึ้นอยู่กับเงื่อนไขดังต่อไปนี้

1) การเลือก Root Bridge

สวิตช์จะส่งเฟรมที่เรียกว่า BPDU (Bridge Protocol Data Unit) แลกเปลี่ยนระหว่างสวิตช์เพื่อเลือกหา Root Bridge โดยเฟรม BPDU มีฟิลด์ (Field) สำคัญหนึ่งที่เรียกว่า Bridge ID เป็นค่าตัวเลข 8 ไบต์ (Byte) แบ่งออกเป็น 2 ส่วนได้แก่

1. Bridge Priority (2 ไบต์) ระบุความสำคัญของสวิตช์นั้นๆ เมื่อเปรียบเทียบกับสวิตช์อื่นในระบบเครือข่าย โดยมีค่าได้ตั้งแต่ 0 ถึง 65,535
2. MAC address (6 ไบต์) เนื่องจากค่า Bridge Priority สามารถที่กำหนดเองได้ ทำให้สวิตช์แต่ละตัวมีโอกาสที่จะมีบริดจ์ไพเออริตี้เท่ากัน จึงมีการนำค่า MAC address ของสวิตช์ ซึ่งเป็นค่าเฉพาะของอุปกรณ์ มาเป็นส่วนหนึ่งของ Bridge ID

โดยปกติสวิตช์จะส่งเฟรม BPDU ออกไปทุกพอร์ตที่รัน STP โดยเฟรมที่ว่าจะมีหมายเลข MAC address ต้นทางเป็นหมายเลข MAC address ของพอร์ตนั้นๆ และหมายเลข MAC address ปลายทางเป็น Multicast address พิเศษที่รู้จักกันในหมู่สวิตช์ที่รัน STP

BPDU มีอยู่ด้วยกัน 2 ประเภท

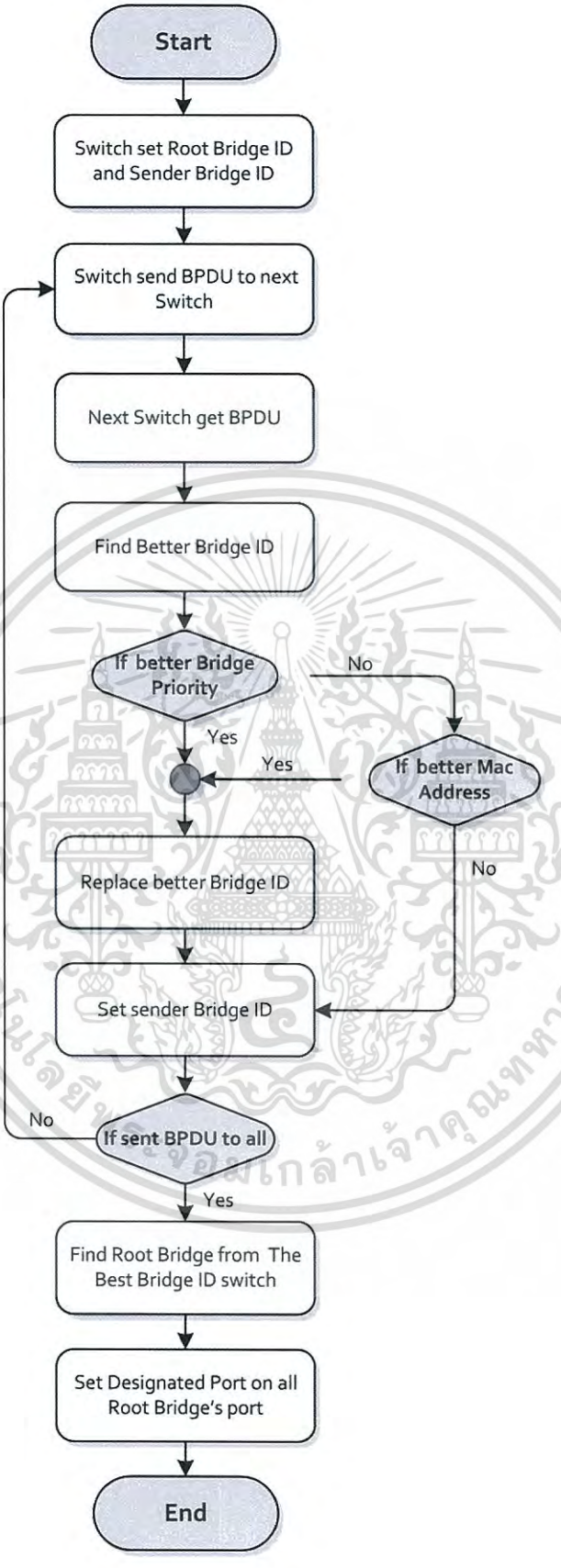
1. Configuration BPDU ใช้เพื่อคำนวณ STP
2. Topology Change Notification (TCN) ใช้เพื่อประกาศให้สวิตช์ตัวอื่นๆทราบถึงการเปลี่ยนแปลงโทโพโลยีที่เกิดขึ้น โดยจะถูกส่งออกทางทุกพอร์ตทุกๆ 2 วินาที

ภายในเฟรม BPDU จะมีฟิลด์ที่สำคัญ ได้แก่

1. Message Type ระบุว่าเป็น Configuration BPDU หรือ TNC
2. Root Bridge ID
3. Sender Bridge ID
4. Root Path Cost
5. Sending Port ID
6. Message Age
7. Maximum Age
8. Hello Time
9. Forward Delay

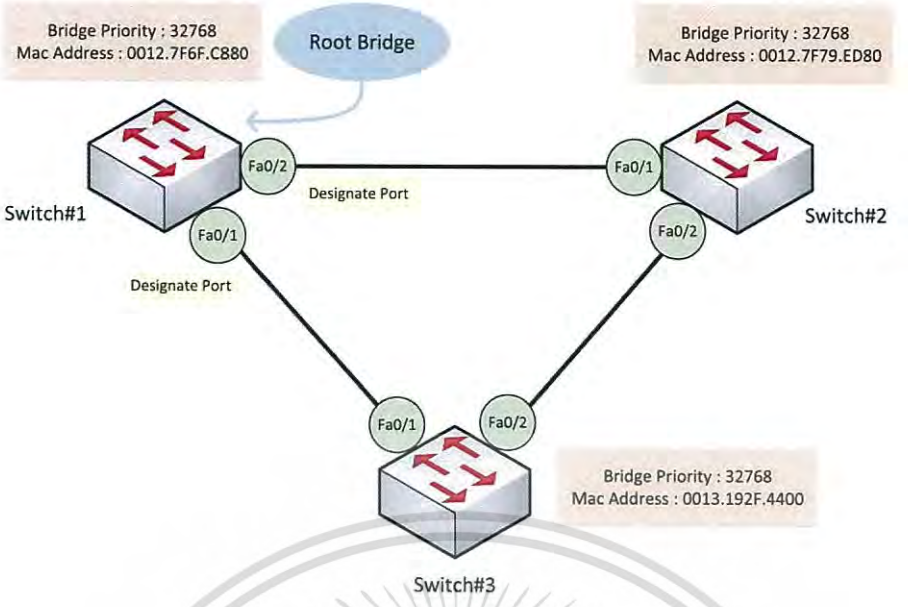
สวิตช์แต่ละตัวเมื่อได้รับเฟรม BPDU เข้ามา มันจะทำการวิเคราะห์ว่ามี BPDU ไหนที่มีค่าต่ำ Bridge ID ของตนเอง หรือต่ำกว่า Bridge ID อื่นๆที่มัน หากสวิตช์พบ BPDU ที่มีค่า Bridge ID ที่ต่ำกว่า มันจะแทนที่ฟิลด์ Root Bridge ID ในเฟรม BPDU ที่ตัวมันประกาศออกไปด้วยค่า Bridge ID ที่ต่ำกว่า โดยยังกำหนดค่าฟิลด์ Sender Bridge ID (ฟิลด์ที่บอกแหล่งที่มา) ให้เท่ากับค่า Bridge ID ปัจจุบันของมันเหมือนเดิม หลังจากนั้นพอร์ตทุกพอร์ตบน Root Bridge จะถูกกำหนดให้เป็น Designated Port อัตโนมัติ ลำดับการหา Root Port จะแสดงตามรูปที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 Data Flow Diagram แสดงการทำงานในการหา Root Bridge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดงตัวอย่างการหา Root Bridge

จากตัวอย่างดังรูปที่ 2.14 สวิตช์ทุกตัวมี Bridge Priority เท่ากันหมด คือ 32768 จึงใช้ค่า MAC address ของสวิตช์มาคำนวณเพื่อเลือกหา Root Bridge แทน โดย Switch#1 มีค่า MAC address น้อยที่สุด จึงได้เลือกเป็น Root Bridge และทุกพอร์ตบนสวิตช์จะถูกกำหนดให้เป็น Desingnated Port โดยอัตโนมัติ

2) การเลือก Root Port

ในการหา Root Port จะพิจารณาค่า Root Path Cost ที่น้อยที่สุดเมื่อเทียบกับพอร์ตอื่นมาเป็นตัวตัดสิน แต่ละลิงค์หรือการเชื่อมต่อระหว่างสวิตช์จะมีค่าที่เรียกว่า Path Cost โดยค่า Path Cost นี้เป็นอัตราส่วนผกผันกันกับค่าแบนด์วิธของลิงค์นั้นๆ ในการพิจารณาหาค่า Root Path Cost นั้นจะได้จากผลบวกของ Path Cost ทั้งหมดนับจาก Root Bridge จนถึงพอร์ตนั้นๆ ในความเป็นจริงแล้วการเลือก Root Port จะพิจารณาที่ค่า Path Cost ที่กำหนดเอาไว้ก่อน ซึ่งถ้ามีค่าเท่ากันจึงจะพิจารณาที่ค่าของ Port ID ซึ่งใน Port ID จะมี Port Priority และ Port Number อยู่ จึงพิจารณาที่ Port Priority ที่ถูกกำหนดขึ้นมาก่อน และถ้าค่าเท่ากันอีกจึงพิจารณาที่ Port Number ที่น้อยที่สุด เช่น FastEthernet0/1 เป็นลำดับสุดท้าย

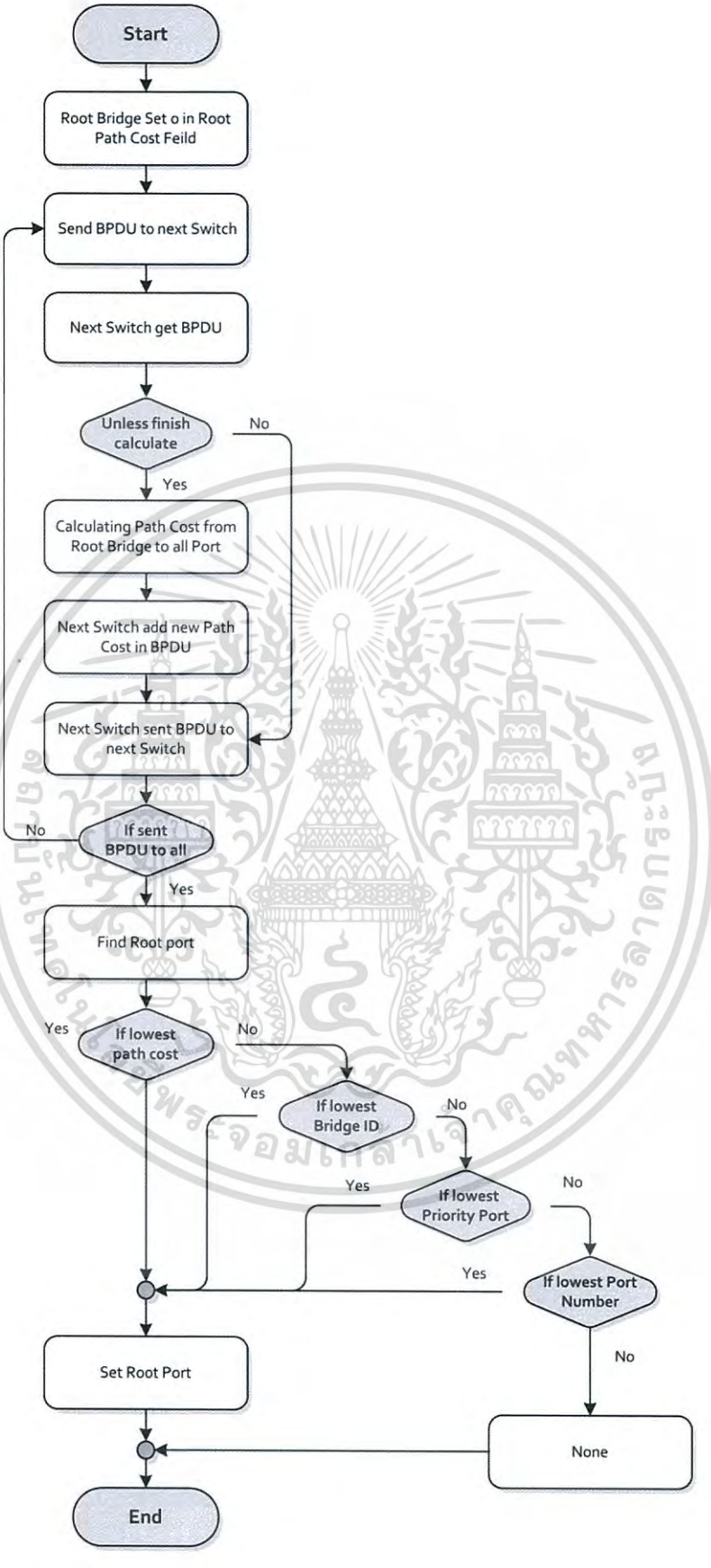
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงค่า Path Cost เทียบกับค่าแบนด์วิธ

แบนด์วิธ	Path Cost
4 Mbps	250
10 Mbps	100
16 Mbps	62
45 Mbps	39
100 Mbps	19
155 Mbps	14
622 Mbps	6
1 Gbps	4
10 Gbps	2

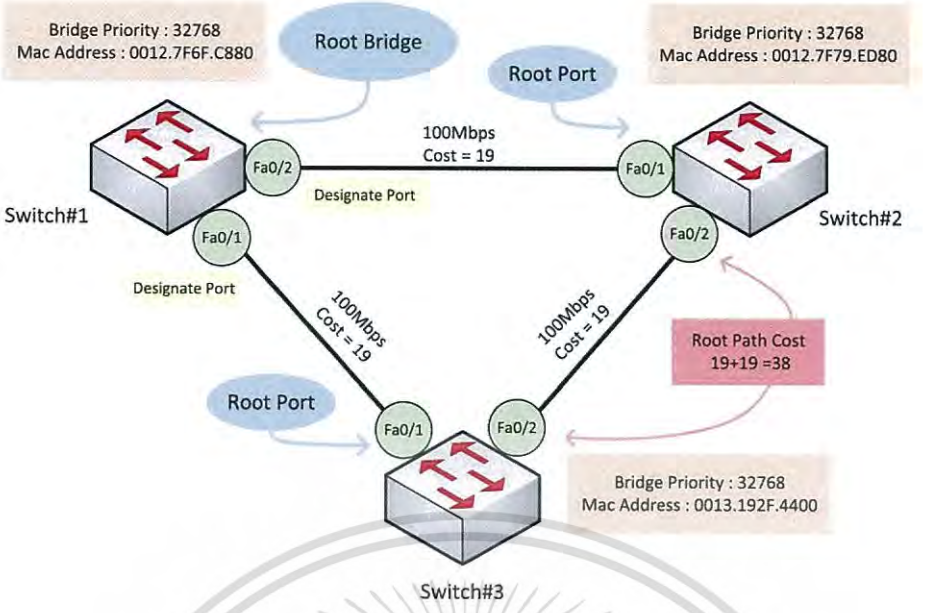
Root Bridge ส่งเฟรม BPDU ออกไปโดยกำหนดค่าฟิลด์ Root Path Cost ให้เท่ากับศูนย์ เพราะพอร์ตของมันต่อตรงกับ Root Bridge เมื่อสวิตช์ตัวอื่นๆที่อยู่ถัดจาก Root Bridge ลงมาได้รับ BPDU นี้มันจะเพิ่มค่า Path Cost ของพอร์ตมันที่ชี้ไปยัง Root Bridge บวกเข้าไปในค่าฟิลด์ Root Path Cost ของเฟรม BPDU ที่ได้รับมา และส่งต่อไปยังอุปกรณ์สวิตช์ตัวอื่นตามเส้นทาง ลำดับการหา Root Port จะแสดงตามรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 Data Flow Diagram แสดงการทำงานในการหา Root Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงตัวอย่างการหา Root Port

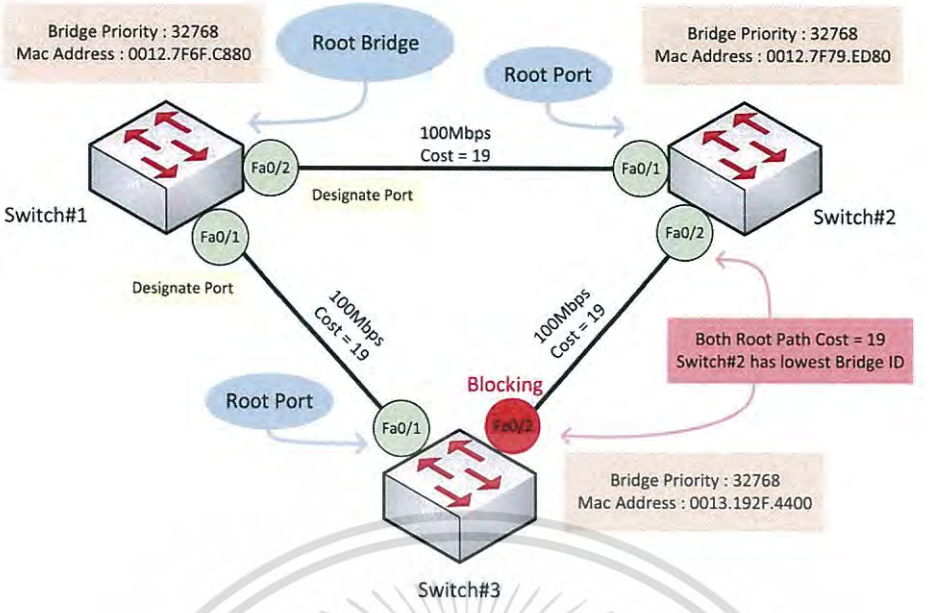
จากตัวอย่างดังรูปที่ 2.16 เส้นทางในแต่ละเส้นทางมีแบนด์วิธเท่ากับ 100 Mbps ค่า Path Cost จึงเท่ากับ 19 พอร์ตบน Switch#2 มี 2 พอร์ต พอร์ต FastEthernet0/1 มีค่า Root Path Cost = 19 นับจาก Root Bridge ถึงพอร์ตนั้นๆ และพอร์ต FastEthernet0/2 มีค่า Root Path Cost = 19+19 = 38 (ได้จากฟิลด์ Root Path Cost ที่เข้ามาทางพอร์ตนี้ ซึ่งเริ่มจาก BPDU ที่ FastEthernet0/1 ของ Switch#3 = 19) เพราะฉะนั้น พอร์ต FastEthernet0/1 ของ Switch#2 จึงได้เป็น Root Port

ในทำนองเดียวกัน Switch#3 พอร์ต FastEthernet0/1 จะมี Root Path Cost = 19 ในขณะที่พอร์ต FastEthernet0/2 มี Root Path Cost = 19+19 = 38 ดังนั้นพอร์ต FastEthernet0/1 จึงได้เป็น Root Port

3) การเลือก Designated Port

ในการเลือก Designate Port จะพิจารณาว่าพอร์ตไหนมี Root Path Cost ที่ดีกว่า ซึ่งถ้ามีค่าเท่ากันจึงจะพิจารณาที่ค่าของ Bridge ID หลังจากนั้นถ้าหากยังมีค่าที่เท่ากันอีกจะพิจารณาที่ Port ID ซึ่งใน Port ID จะมี Port Priority และ Port Number อยู่ จึงพิจารณาที่ Port Priority ที่ถูกกำหนดขึ้นมาก่อน และ Port Number ที่น้อยที่สุด เช่น FastEthernet0/1 เป็นลำดับสุดท้าย ตัวอย่างการหา Designated Port จะแสดงตามรูปที่ 2.17

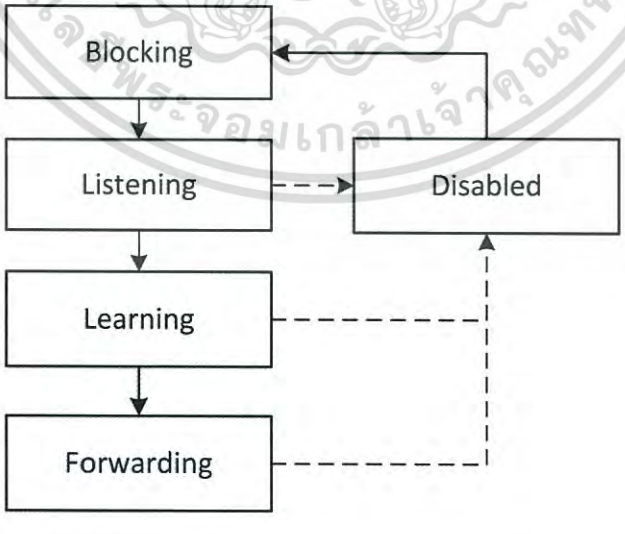
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 แสดงตัวอย่างการหา Designated Port

จากตัวอย่างรูปที่ 2.17 พอร์ต FastEthernet0/2 ของ Switch#2 และ Switch#3 มี Root Path Cost เท่ากันคือ 19 เพราะ ค่าของ Root Path Cost ที่ใช้เพื่อการเลือก Designated Port จะพิจารณาค่าของ Root Path Cost ที่อยู่ใน BPDU ที่กำลังถูกส่งออกจากพอร์ตนั้นๆ

เพราะฉะนั้น ในกรณีที่มีค่า Root Path Cost เท่ากัน จึงพิจารณาจากค่า Bridge ID แทน ซึ่ง Switch#2 มีค่า Bridge ID ที่ต่ำกว่า พอร์ต FastEthernet0/2 ของ Switch#2 จึงได้เป็น Designated Port ลำดับสถานะในการจัดการของ STP จะแสดงดังรูปที่ 2.18



รูปที่ 2.18 แสดงสถานะของพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Blocking State เมื่อมีการเลือก Block พอร์ตเรียบร้อยแล้ว จะไม่มีการรับส่งข้อมูลชั่วคราว และไม่สามารถเพิ่ม MAC address เข้าไปในตาราง MAC address Table ได้ สวิตช์แต่ละตัวยังคงส่งเฟรม BPDU เพื่อแลกเปลี่ยนกันได้อยู่ทุก 20 วินาที ถ้าเริ่มมีการผิดปกติเกิดขึ้นก็จะเปลี่ยนสถานะไปสู่ Listening

Listening State สวิตช์จะเริ่มเชื่อมลิงค์ (Enable Link) และทำการรอฟังว่าพอร์ตนั้นสามารถเป็น Designated Port หรือ Root Port ได้หรือไม่ ใช้เวลาประมาณ 15 วินาที

Learning State สวิตช์จะเริ่มมีการเรียนรู้ MAC address จากที่อื่นๆ และรอดูว่าจะมีอะไรเกิดขึ้น ในขั้นตอนนี้ใช้เวลาประมาณ 15 วินาที

Forwading State หลังจากที่พอร์ตอยู่ในสถานะ Learning จนครบช่วงเวลา Forward Delay แล้ว พอร์ตที่ถูก Block ไว้จะสามารถกลับมารับส่งข้อมูลได้

STP จะคำนวณตลอดว่า BPDU ที่ได้รับมานั้นถูกต้องหรือไม่ ได้รับจากเส้นทางเดิม อุปกรณ์สวิตช์ตัวเดิมหรือไม่ ถ้าหากว่าเส้นทางใดเสียหาย เท่ากับว่า BPDU เปลี่ยน มันก็จะเข้า step เปลี่ยนสถานะเป็น Listening การเข้าสู่โหมดต่างๆ ของ STP ต้องใช้เวลาประมาณ 15 วินาที ถ้าหากว่าระบบเสียหายการส่งข้อมูลของ User อาจต้องใช้เวลาถึง 50 วินาที

ตารางที่ 2.2 สรุปสถานะและความสามารถของพอร์ต

สถานะ	สามารถรับส่งเฟรมข้อมูลได้หรือไม่	สามารถเรียนรู้เฟรมที่ได้รับและเพิ่ม MAC address เข้าไปได้หรือไม่	เป็นสถานะชั่วคราวหรือสถานะที่อยู่ตัว
Blocking	ไม่ได้	ไม่ได้	สถานะอยู่ตัว
Listening	ไม่ได้	ไม่ได้	สถานะชั่วคราว
Learning	ไม่ได้	ได้	สถานะชั่วคราว
Forwarding	ได้	ได้	สถานะอยู่ตัว

ช่วงเวลาที่เกี่ยวข้องกับ STP

- **Hello Time**

  โดยดีฟอลต์ (Default) ค่าเวลานี้เท่ากับ 2 วินาที เพราะฉะนั้น เมื่อครบทุก 2 วินาที Root Bridge จะทำการส่ง BPDU ไปทุกพอร์ต ค่า Hello Time นี้สามารถถูกกำหนดค่าใหม่โดยการแก้ไขค่า Hello Time บน Root Bridge แล้ว Root Bridge จะเป็นผู้อัปเดตค่า Hello Time ไปยัง Non Root Bridge อื่นๆเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Forward Delay**

เป็นช่วงเวลาที่พอร์ตของสวิตช์ใช้ไประหว่างการเปลี่ยนสถานะจาก Listening เข้าสู่ Learning และเป็นช่วงเวลาที่พอร์ตของสวิตช์ใช้ไประหว่างการเปลี่ยนสถานะจาก Learning เข้าสู่ Forwarding โดยดีฟอลต์ช่วงเวลานี้เท่ากับ 15 วินาที

- **Maximum (Max) Age**

เป็นช่วงเวลาสูงสุดที่สวิตช์จะรอคอย หลังจากที่เริ่มสังเกตว่าไม่ได้รับ BPDU ที่ "ดีที่สุด" ที่ตนเองเคยได้รับ โดยเมื่อคอยจนกระทั่งครบช่วงเวลาแล้วยังคงไม่ได้รับ BPDU เข้ามาอีก มันจะสรุปว่าขณะนี้มีการเปลี่ยนแปลงโทโพโลยีเกิดขึ้นแล้ว โดยดีฟอลต์ค่าเวลานี้เท่ากับ 20 วินาที

การเปลี่ยนแปลงโทโพโลยี

จากรูปที่ 2.20 พอร์ต Fa0/2 ของ Switch#3 เท่านั้นที่อยู่ในสถานะ Blocking ถ้าเกิดลิงค์ระหว่าง Switch#1 และ Switch#3 ขาดการเชื่อมต่อกัน ในกรณีนี้ Switch#3 จะเป็นสวิตช์ที่รับรู้การเปลี่ยนแปลงได้ก่อนสวิตช์ตัวอื่น เพราะ Switch#3 ไม่ได้รับเฟรม Hello BPDU มาจาก Root Port (พอร์ต Fa0/1) ภายในช่วงเวลา Max Age ในขณะที่ Switch#2 ยังคงได้รับ Hello BPDU เข้ามาทาง Root Port

หลังจาก Max Age หมดอายุ Switch#3 จะส่ง TCN ไปทางพอร์ต Fa0/2 (ซึ่งเป็นพอร์ตที่เหลืออยู่) เพื่ออ้างว่าตนเองเป็น Root Bridge เมื่อ Switch#2 ได้รับก็จะส่ง BPDU ไปยัง Switch#1 ซึ่งปัจจุบันเป็น Root Bridge เมื่อ Switch#1 ได้รับ TCN BPDU มันก็จะส่ง BPDU ตอบกลับมาทางเดิมจนถึง Switch#3

เมื่อ Switch#3 ได้รับ BPDU กลับมา มันก็จะรู้ว่าตัวมันเองไม่สามารถเป็น Root Bridge ได้ เนื่องจากมีค่า Bridge ID ที่มากกว่าของ Switch#1 ดังนั้น Switch#3 จึงตัดสินใจทำต่อไปคือ

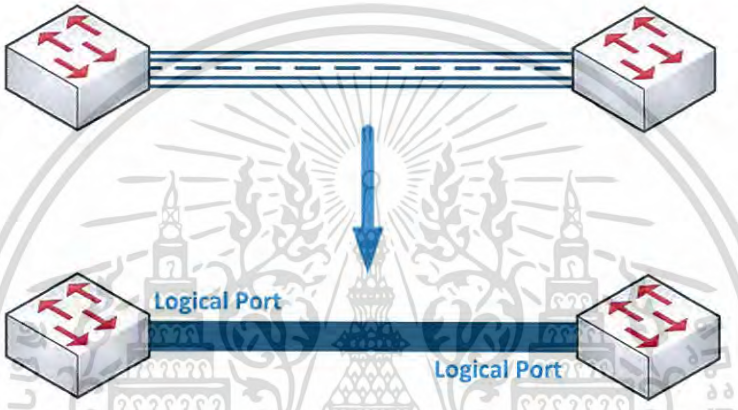
1) ตัดสินใจใหม่ว่าให้พอร์ต Fa0/2 เปลี่ยนสถานะจาก Blocking กลายเป็น Forwarding
2) พอร์ต Fa0/1 อาจมีปัญหา ดังนั้นมันจะเปลี่ยนให้อยู่ในสถานะ Blocking
3) สวิตช์จะเคลียร์ MAC address Table ทิ้ง เพราะเป็นไปได้ว่าหมายเลข MAC address ที่เคยเรียนรู้ว่าได้รับเข้ามาทางพอร์ต Fa0/1 อาจต้องเปลี่ยนแปลงไป กลายเป็นให้มาเรียนรู้ทางพอร์ต Fa0/2 แทน

Switch#3 ไม่สามารถเปลี่ยนแปลงพอร์ต Fa0/2 เข้าสู่สถานะ Forwarding State ได้ในทันที เพราะสวิตช์ตัวอื่นอาจมีการเปลี่ยนสถานะอยู่พร้อมๆกัน ทำให้เกิดลูปได้ ดังนั้นเพื่อป้องกันจึงได้กำหนดให้มีการเปลี่ยนแปลงเข้าสู่สถานะ Listenning และ Learning ก่อน โดยเวลาที่ใช้ไปในแต่ละสถานะเท่ากับ 15 วินาที ดังนั้นใช้เวลาทั้งหมด รวมกันแล้วประมาณ 50 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 EtherChannel

EtherChannel [4] หรือ Link Aggregation เป็นคุณสมบัติที่ทำให้สามารถรวมหลายๆ Interface แบบ Physical เข้าด้วยกันเป็น Interface แบบ Logical เพียงอันเดียวได้ ใช้เพื่อแก้ปัญหาความคับคั่ง (Congestion) ในการใช้งานระบบเครือข่าย หรือใช้แก้ปัญหาเมื่อมีปริมาณการใช้งานระบบเครือข่ายมากเกินไปจนทำให้การรับ-ส่งข้อมูลทำได้ช้า เสมือนเป็นการเพิ่มเส้นทางข้อมูลขึ้นจากเส้นทางเดียวเป็นหลายๆเส้นทาง เพื่อที่จะทำการ Shared Traffic ไปในหลายๆเส้นทางนั่นเอง พร้อมทั้งยังเป็นการทำ Redundant ในกรณีที่มีบางลิงค์ที่เป็นสมาชิกของ EtherChannel เกิดมีปัญหาขึ้นมาได้อีกด้วย



รูปที่ 2.19 แสดงลักษณะการทำ EtherChannel

EtherChannel ไม่ได้เป็นการเพิ่มแบนด์วิธเนื่องมาจากในการส่งข้อมูลไปยังอุปกรณ์ฝั่งตรงข้ามนั้น จะมีการเลือกเส้นทางที่จะใช้ส่งข้อมูลตามเงื่อนไขเพียงเส้นทางเดียวเท่านั้น เพราะฉะนั้น ถ้าแต่ละเส้นทางมีแบนด์วิธเท่ากับ 100 Mb เมื่อทำ EtherChannel โดยใช้ 2 ลิงค์แล้วแบนด์วิธก็ไม่ได้เพิ่มเป็น 200 Mb แต่อย่างใด (เนื่องจากในแต่ละลิงค์ก็ยังมีแบนด์วิธเท่ากับ 100 Mb เช่นเดิม) ในการที่จะใช้งาน EtherChannel ให้ได้เห็นผลลัพธ์อย่างชัดเจนนั้น จึงจะต้องใช้กับระบบเครือข่ายที่มีความคับคั่งที่ค่อนข้างสูง เพื่อที่จะได้ใช้ EtherChannel นี้ไปช่วยในการ Shared Load เพื่อลดค่าความคับคั่งลง ซึ่งถ้านำ EtherChannel ไปใช้กับเครือข่ายที่ไม่ได้มีการใช้งานในปริมาณที่มากแล้ว ก็อาจจะไม่เห็นความแตกต่างในการใช้งานเท่าไร แต่ประโยชน์ของ EtherChannel นั้นยังมีมากกว่าการลดค่า Congestion ลง นั่นก็คือเป็นการสำรองเส้นทางระหว่างอุปกรณ์ที่ใช้ในการทำ EtherChannel ทั้งสองฝั่งนั่นเอง ถ้าเกิดกรณีที่ลิงค์ใดมีปัญหาขึ้นมา ก็ยังมีลิงค์อื่นๆที่สามารถใช้ในการส่งข้อมูลต่อไปได้นั่นเอง

สำหรับเงื่อนไขที่นำมาใช้ในการตัดสินใจว่าจะส่ง Packet ไปในเส้นทางใดนั้น สามารถเลือกได้จาก MAC address, IP Address, Port Number, Source หรือ Destination โดยเมื่อทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกเงื่อนไขใดแล้วเงื่อนไขนั้นก็จะนำไปใช้กับทุก EtherChannel ที่มีการใช้งานอยู่บนสวิตช์ตัวนั้น และการที่จะเลือกว่าจะใช้เงื่อนไขใดก็ควรพิจารณาตามแต่ละสถานการณ์ไป

การทำงานของ EtherChannel สำหรับ STP จะมองว่า Physical พอร์ตหลายๆพอร์ตที่เข้ามาอยู่ใน EtherChannel นี้ เป็นเสมือนพอร์ตเดียวกัน และทำการคำนวณโดยมองว่า 1 EtherChannel พอร์ตเป็นเพียง 1 เส้นทางเท่านั้น จะไม่มองว่าภายใน 1 EtherChannel นี้จริงๆแล้วมีกี่ Physical พอร์ต เช่น ถ้าหากพอร์ต Fasethernet 0/1 และ Fastethernet 0/2 อยู่ใน EtherChannel เดียวกันแล้ว พอร์ตทั้งคู่จะอยู่ในสถานะเดียวกัน เช่น ถ้าสถานะเป็น Forwarding ก็จะเป็นสถานะ Forwarding ทั้งคู่

2.4 Tool Command Language

Tcl (Tool Command Language) [7] เป็นภาษาสคริปต์ที่สามารถใช้บน Cisco IOS ตั้งแต่ Version 12.2(25) ขึ้นไป เป็นการเขียนโปรแกรมเพื่อสั่งให้อุปกรณ์ทำตามคำสั่งที่เราเขียนไว้

Tclsh เป็น shell ของ Tcl สำหรับการอ่านคำสั่งต่างๆ ของ Tcl จาก Standard input หรือไฟล์ แล้วทำการตีความคำสั่งนั้นๆ Tclsh จะตีความให้เป็น Tcl สคริปต์ หากการเรียกใช้ shell ปราศจาก Argument จะเป็นการใช้งาน Tcl แบบ Interface Active เมื่อมีการเรียกใช้ Tclsh จะมีพรอมต์ (Prompt) สำหรับสั่งงานด้วยคำสั่ง Tcl และจะออกจาก Tclsh เมื่อได้รับคำสั่ง exit หรือจบแฟ้มข้อมูล

การใช้ Tcl/Tk แบบ Non-Interface Active คือ การเขียนสคริปต์คำสั่งเก็บไว้ในไฟล์ก่อน จากนั้นจึงเรียก tclsh มาแปลคำสั่งในสคริปต์ที่ได้เขียนอีกครั้งหนึ่ง วิธีนี้มีประโยชน์มากกว่าการใช้งานแบบ Interface Active เนื่องจากทำให้สามารถสั่งงานได้ต่อเนื่องด้วยชุดคำสั่ง ทำให้ได้แอปพลิเคชั่นที่เต็มรูปแบบ สมมติว่าได้ทำการเขียนสคริปต์ไว้ในไฟล์ myprog.tcl แล้วจะเรียกใช้โปรแกรมที่เขียนมาทำงานที่ยูนิกส์พรอมต์ (Unix Prompt) ได้ดังนี้ $tclsh filename.tcl
นอกจากนี้ยังสามารถเรียกชื่อสคริปต์ได้โดยตรง แต่ในไฟล์สคริปต์จะต้องมีคำสั่ง Shell ดังนี้

#!/bin/sh

exec wish "$0" "#@"

วิธีการนี้ช่วยป้องกันความผิดพลาดที่เกิดจากการติดตั้ง Tcl/Tk ไว้ในไดเรคทอรีที่ต่างกัน หลังจากคำสั่ง shell ทั้งสองบรรทัดนี้แล้วจึงจะเป็นคำสั่งอื่นๆ ของโปรแกรมที่เขียนขึ้น นอกจากนี้ถ้าต้องการเรียกชื่อไฟล์สคริปต์จากยูนิกส์พรอมต์ให้ขึ้นมาทำงานได้เองโดยไม่ต้องเรียก tclsh ก่อน ใช้คำสั่ง

chmod u+x .tcl // เพื่อเป็นการเปลี่ยนโหมดไฟล์ให้เป็น execute จึงจะทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) Basic syntax

Tcl เป็นภาษาแบบสคริปต์ โดยที่คำสั่งต่างๆ จะแยกกันด้วยการขึ้นบรรทัดใหม่ หรือเครื่องหมายอัฒภาค (Semicolon) และทุกๆ คำสั่งจะมีลักษณะของคำสั่งเหมือนกัน เช่น

expr 2 + 1

คำสั่งนี้จะทำการคำนวณผลบวกของ 2 และ 1 และทำการส่งผลลัพธ์ให้เป็น 3 ซึ่งลักษณะของผลลัพธ์จะเป็นลักษณะของสายอักขระ (String) หากคำสั่งนั้นไม่มีความหมาย (No Meaningful) ผลลัพธ์ที่ได้จะเป็น String ว่าง คำสั่งแต่ละคำสั่งของ Tcl จะประกอบด้วยคำ (Word) 1 คำ หรือมากกว่า โดยที่แต่ละคำจะแยกโดยใช้ช่องว่าง (Space)

2) Variables

Tcl อนุญาติให้ทำการเก็บค่าด้วยการใช้ตัวแปรได้ และใช้ค่านั้นในคำสั่งได้ คำสั่ง set ใช้ในการกำหนดค่าและอ่านค่าของตัวแปร

set x 32 //เป็นการกำหนดค่าให้กับตัวแปร x

set x //เป็นการอ่านค่าของตัวแปร x

สำหรับ Tcl ไม่มีความจำเป็นที่จะต้องทำการประกาศตัวแปรขึ้นใช้งาน เพราะตัวแปรจะถูกสร้างขึ้นมาอย่างอัตโนมัติในครั้งแรกที่ถูกใช้ และตัวแปรของ Tcl ไม่มีประเภทของตัวแปร ทำให้ตัวแปรสามารถเก็บค่าต่างๆ ได้ การใช้เครื่องหมาย $ ในคำสั่งจะเป็นการระบุว่าคำหรือตัวเลขที่ตามหลังนั้นเป็นชื่อของตัวแปรและจะทำการแทนค่าของตัวแปร (Variable substitution) ตัวนั้นด้วยค่าของมัน

3) Command substitution

วิธีการที่มีการใช้ผลลัพธ์ของคำสั่งหนึ่ง มาเป็น Argument ของคำสั่งอื่นๆ เรียกวิธีการนี้ว่า Command substitution เช่น

set A 44

*set B [expr $A*4]*

การใช้เครื่องหมาย [ ] จะเป็นการกำหนดให้ Tcl ทำงานทุกๆ อย่างที่อยู่ใน [ ] เป็นแบบ Nested Tcl command นั่นคือ Tcl จะทำการแทนค่าผลลัพธ์ที่ได้จากคำสั่งใน [ ] เพราะฉะนั้น Argument ของคำสั่งที่แสดงในการ set ครั้งที่สองจะเป็น 176

4) Quotes and Braces

การใช้ Double-quotates (“ ”) เป็นการสร้างกลุ่มของคำให้เป็น Argument เดียว โดยที่เครื่องหมาย $ และคำที่อยู่ในเครื่องหมาย [ ] ยังคงถูกแปลเพื่อให้ทำงาน เช่น

set x 1

set y 2

set z “$x + $y is [expr $x + $y]”

puts “$z”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากทำทั้ง 3 คำสั่งแล้ว ตัวแปร z จะมีค่าเป็น 1 + 2 เท่ากับ 3 ในการทำงานทุกๆ อย่างที่อยู่ใน Double-quotes จะถูกส่งไปยังคำสั่ง set เหมือนเป็นคำเดียว จากตัวอย่างหากไม่มีเครื่องหมาย Double-quotes คำสั่ง set จะมี Argument อยู่ 6 ตัว คือ z, $x, +, $y, is, [expr $x + $y] ซึ่งจะส่งผลให้เกิดความผิดพลาด และคำสั่ง Puts คือการแสดงผล Argument ที่อยู่ใน Double-quotes จะถูกแสดงผลออกมาทางหน้าจอ

5) Control structures

Tcl มีคำสั่งที่ใช้ในการควบคุมการทำงานต่างๆ คือ

- โปรแกรมย่อย (Procedures)
- คำสั่งเงื่อนไขการทำงาน (Condition execution)
- คำสั่งเกี่ยวกับการวนทำซ้ำ (Looping)

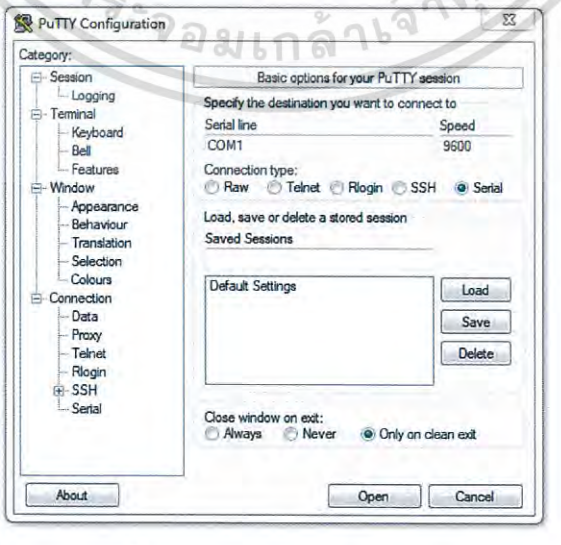
6) คำสั่งคอนฟิกค่าในโหมด Configuration บนสวิตช์ เช่น

los_config"interface gigabitethernet 0/1"

2.6 โปรแกรมที่เกี่ยวข้อง

2.6.1) โปรแกรม Putty

PuTTY [10] หรือโปรแกรม Telnet จากเครื่องลูก ใช้ในการเชื่อมต่อรีโมท (Remote) จากเครื่องคอมพิวเตอร์ตัวเอง เข้าไปยังเครื่องคอมพิวเตอร์อื่นหรือเซิร์ฟเวอร์ โดยวิธีการ Telnet หรือ Secure Shell (SSH) จากเครื่องลูก (Client) เข้าไปจัดการ พิมพ์คำสั่ง หรือส่งคำสั่ง ในเครื่องเซิร์ฟเวอร์ด้วยระบบ Command-line Interface (ใช้พิมพ์คำสั่ง) ในการเชื่อมต่อเข้าไปยังเครื่องเซิร์ฟเวอร์ ได้ง่ายๆ เพียงแค่พิมพ์ชื่อ Hostname พอร์ตที่ต้องการเชื่อมต่อ ซึ่งโดยส่วนใหญ่ค่าปกติคือ 22 ขึ้นอยู่กับอุปกรณ์ปลายทาง

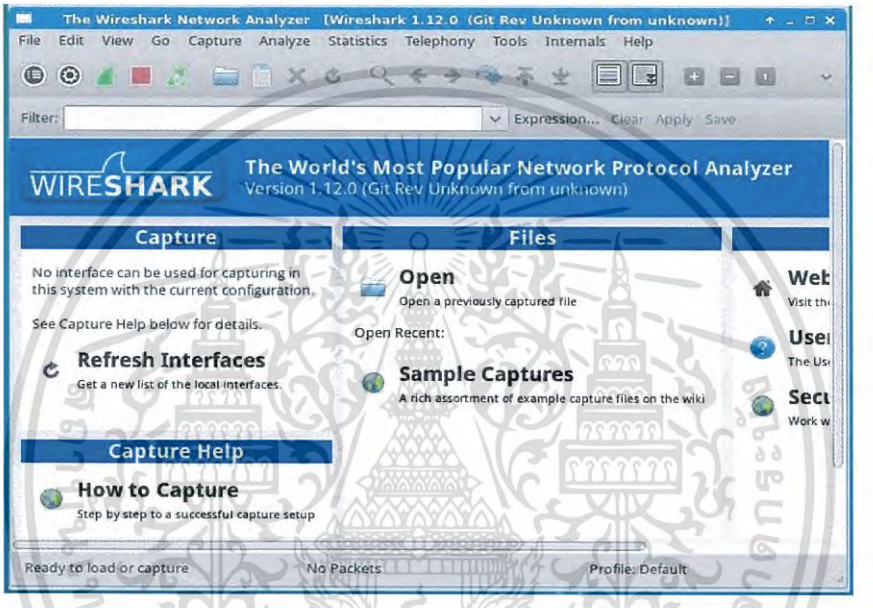


รูปที่ 2.20 แสดงโปรแกรม Putty

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2) โปรแกรม Wireshark

Wireshark เป็นโปรแกรมประเภท Packet Analyzer หรือ Sniffer (แอบดักฟัง) ไว้ใช้วิเคราะห์ข้อมูลที่รับส่งกันในระบบเครือข่าย ความสามารถของ Wireshark มี 2 อย่างคือ ดักจับข้อและบันทึกข้อมูลทุกอย่างที่ผ่านการ์ด LAN หรือ Network Interface Card ใดๆ และอีกข้อคือแสดงผลให้ดู เพื่อที่จะสามารถวิเคราะห์ได้ว่าตอนนี้มีข้อมูลอะไรที่วิ่งในระบบเครือข่าย ส่วนใหญ่แล้วโปรแกรม Wireshark มักนำไปใช้แก้ไขปัญหาบน Network เช่น Loopback, Worm, Netcut, Bittorrent

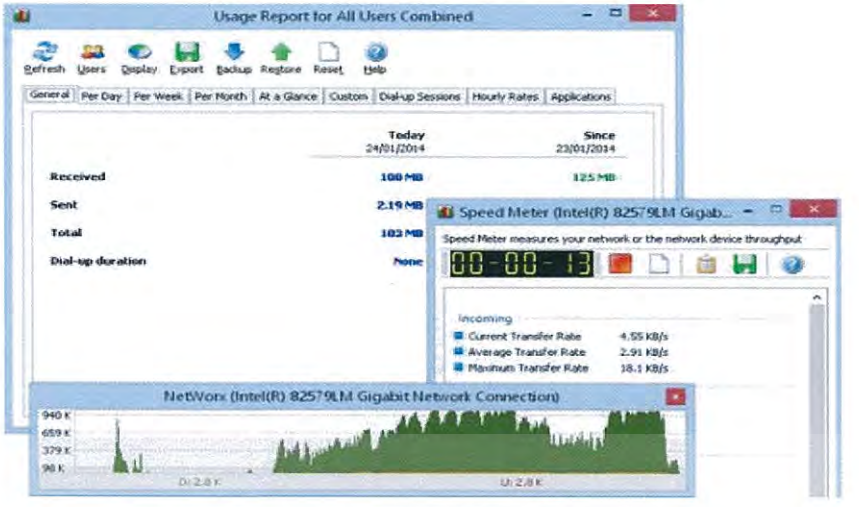


รูปที่ 2.21 แสดงโปรแกรม Wireshark

2.6.3) โปรแกรม Networx

Networx เป็นโปรแกรมที่มีประโยชน์สำหรับผู้ดูแลระบบทั้งหลาย ที่จะใช้มอนิเตอร์ (Monitor) เพื่อดูข้อมูลต่างๆ ที่เกิดขึ้นในระบบเครือข่าย หรือ วงเครือข่ายที่รับผิดชอบดูแลอยู่ โปรแกรมนี้สามารถที่จะตรวจวัดปริมาณข้อมูลการรับส่ง ได้ทั้งจากการเชื่อมต่อแบบใช้สายโทรศัพท์ ไม่ว่าจะเป็นผ่านการหมุนโมเด็ม (Dial-up Modem) รวมไปถึงเอดีเอสแอล (ADSL) เคเบิ้ลโมเด็ม (Cable Modem) ผ่านระบบเครือข่ายภายใน ผ่านดาวเทียม (Satellite) ซึ่งการแสดงผลสามารถแสดงผลข้อมูลขาเข้า (Data in) และข้อมูลขาออก (Data out) ได้ทั้งในรูปแบบของกราฟ หรือเป็นตัวเลข

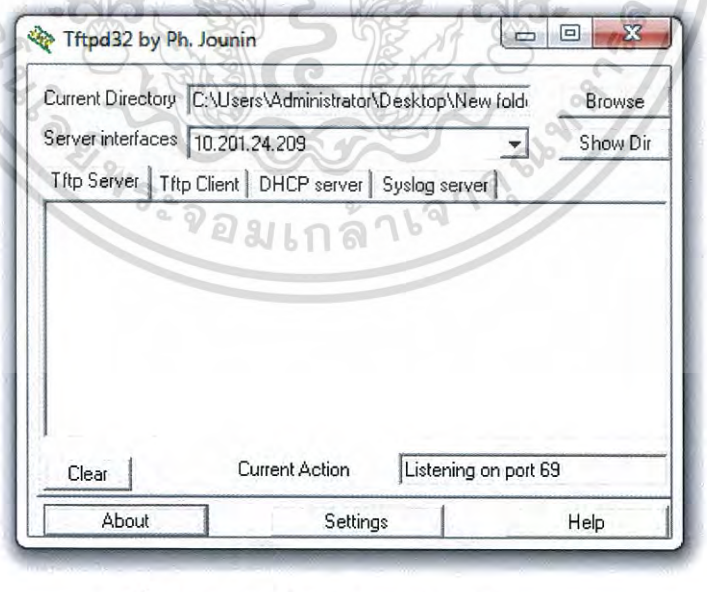
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 แสดงโปรแกรม Networx

2.6.4) โปรแกรม TFTPD32

TFTPD32 เป็นโปรแกรมที่รองรับการใช้งานของกระบวนการ TFTP สำหรับการถ่ายโอนไฟล์ระหว่างเครื่องคอมพิวเตอร์และอุปกรณ์ ทั้งเราเตอร์และสวิตช์ โดยไฟล์ที่ทำการถ่ายโอนนั้นอาจจะเป็นประเภท Configuration File ที่มีผลต่อการทำงานของอุปกรณ์จริง โปรแกรมจะมีขนาดเล็กซึ่งใช้เนื้อที่หน่วยความจำน้อย เครื่องที่ติดตั้ง tftpd32 ไว้แล้วจะมีสถานะเป็นเครื่อง Tftp server ซึ่งในขั้นแรกต้องทำการตั้งค่าให้เครื่องดังกล่าวสามารถติดต่อกับสวิตช์ได้ นั่นคือ ต้องกำหนด IP Address ของเครื่องคอมพิวเตอร์ดังกล่าวให้สามารถติดต่อไปยังสวิตช์ได้ เช่นสามารถ ping ไปยังสวิตช์ได้



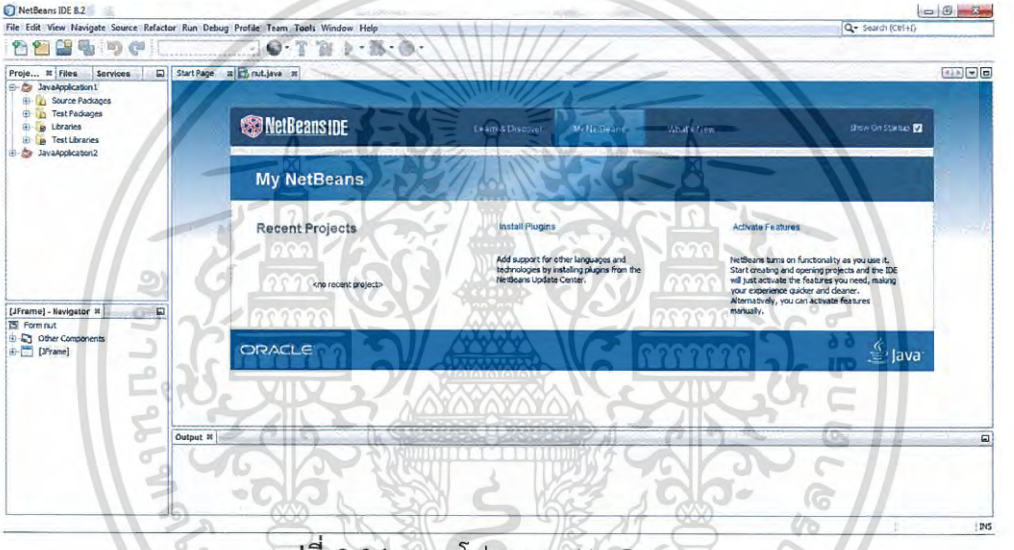
รูปที่ 2.23 แสดงโปรแกรม Tftpd32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5) โปรแกรม NetBeans

NetBeans [9] คือโปรแกรมที่ใช้พัฒนา Application เป็นโปรแกรมประเภท OpenSource software นอกจากนี้ยังสามารถพัฒนาอื่นๆได้อีกหลากหลายโดยติดตั้งโปรแกรมเสริม (Add-on) ได้จากเว็บไซต์ หรือผ่านตัวอัปเดตเซนเตอร์ (Update Center) ของ NetBeans เช่น ภาษาซี/ซีพลัสพลัส (C/C++), Ruby, UML, SOA, Web Application, Java EE, Mobility (Java ME), Java FX, Java Script, PHP เป็นต้น

ข้อดีของโปรแกรมนี้ก็คือ โปรแกรม NetBeans ทำงานแยกส่วนต่างๆออกจากกันเป็น Module จึงทำให้สามารถนำ Module ต่างๆที่มีผู้ที่ได้พัฒนาต่อเติมมาติดตั้งเพิ่มเติมในภายหลังได้ ใช้งานได้กับระบบปฏิบัติการ Windows, Linux, Mac OS X และ Solaris



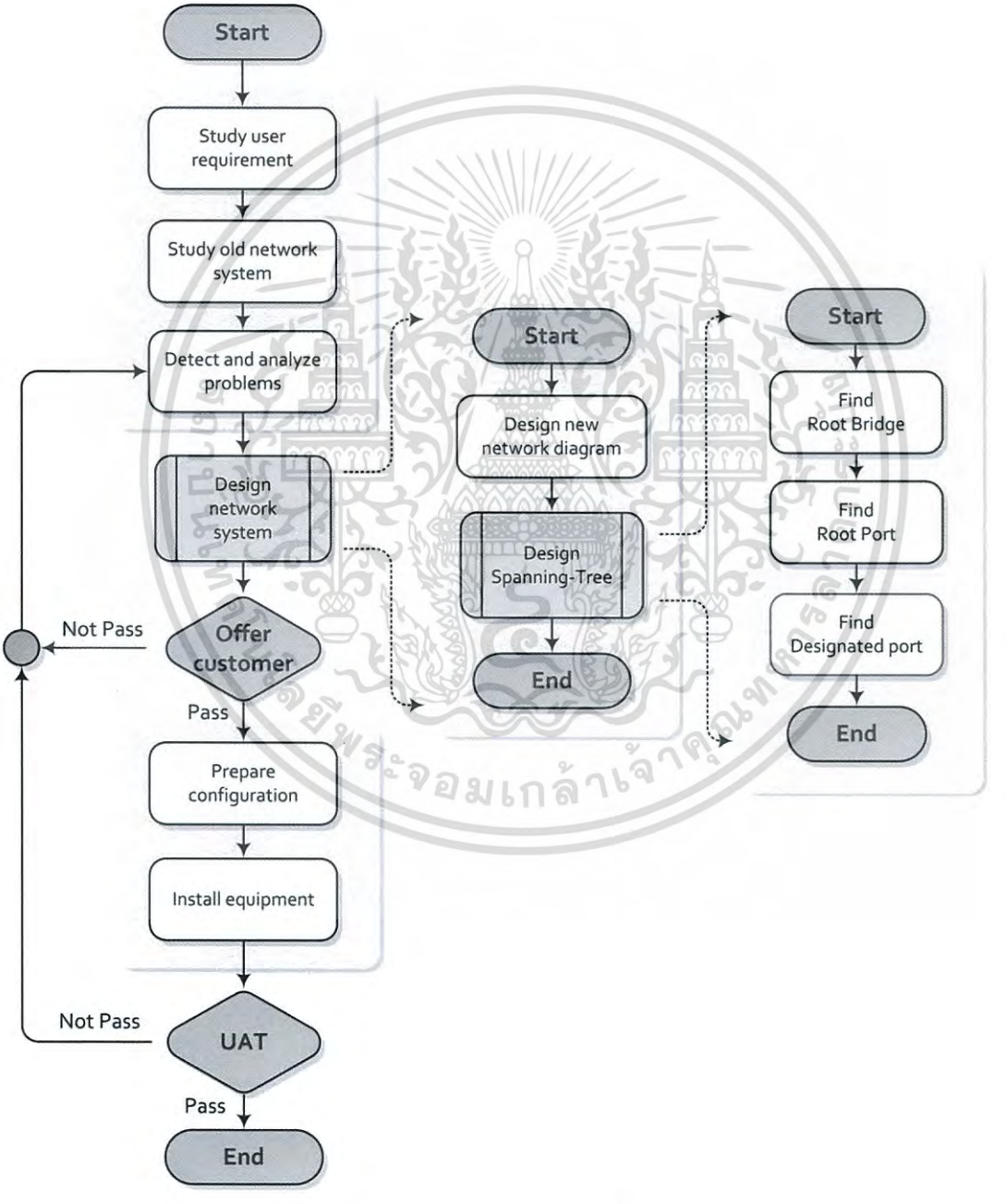
รูปที่ 2.24 แสดงโปรแกรม NetBeans

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงาน

วิธีดำเนินงานของการออกแบบและพัฒนาระบบเครือข่ายเพื่อแก้ปัญหาลูปที่เกิดขึ้นจากระบบเก่า ผู้จัดทำได้แบ่งขั้นตอนในการดำเนินงานออกเป็น 3 ขั้นตอน ดังนี้



รูปที่ 3.1 Data Flow Diagram แสดงวิธีการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ขั้นตอนการศึกษารวบรวมข้อมูลของระบบ

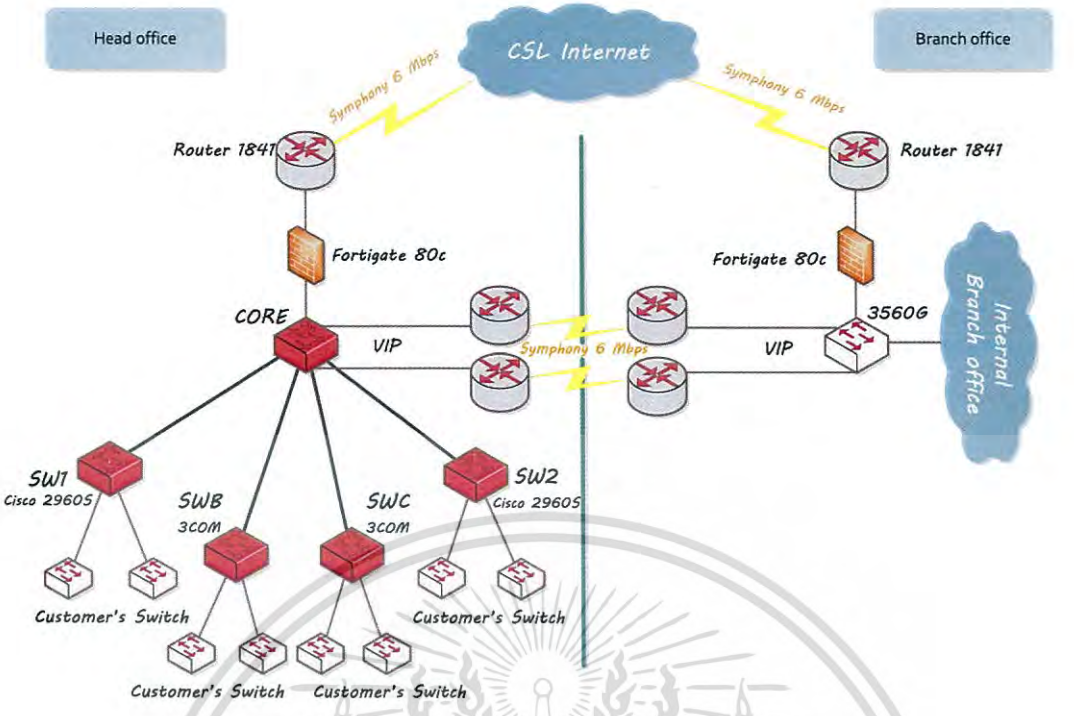
3.1.1 ศึกษาระบบเครือข่ายเดิม

ความสำคัญของการออกแบบระบบเครือข่ายที่มีระบบเดิมอยู่แล้ว สิ่งที่สาคัญเป็นอันดับแรก คือการสำรวจโครงสร้างระบบเครือข่ายเดิมของผู้ใช้งาน และการใช้งานขององค์กรแต่ละองค์กร เพราะจะทำให้สามารถทราบถึงปัญหาและสาเหตุของปัญหาที่เกิดขึ้น เพื่อให้สามารถทราบถึงวิธีการแก้ไขปัญหาของการเกิดลูป รวมถึงศึกษาและรวบรวมความต้องการของผู้ใช้งานเพื่อเป็นแนวทางในการพัฒนาระบบให้มีประสิทธิภาพมากยิ่งขึ้นโดยการศึกษาระบบเครือข่ายเดิมแบ่งได้เป็น 2 ขั้นตอน ดังนี้

1) ศึกษาจาก Network Diagram
   Network Diagram คือแผนผังโครงสร้างของระบบเครือข่ายที่อธิบายถึงการเชื่อมต่อทั้งระบบภายในและภายนอก ซึ่งจะมีรายละเอียดของอุปกรณ์แต่ละตัวว่ามีการเชื่อมต่อกันอย่างไร ซึ่งใน Network Diagram นี้จะอธิบายถึงโครงสร้างการเชื่อมต่อเดิมก่อนที่จะเกิดปัญหาลูปขึ้น
2) ศึกษาจากคอนฟิกและการเชื่อมต่อปัจจุบันของอุปกรณ์แต่ละตัว
   นอกจากการศึกษาจาก Network Diagram แล้ว เพื่อให้ได้ข้อมูลที่ถูกต้องและแม่นยำ ควรศึกษาจากคอนฟิกของอุปกรณ์แต่ละตัวในระบบเครือข่ายด้วย ว่ามีการเชื่อมต่อหรือใช้คำสั่งใดบ้าง เพื่อให้ง่ายต่อการหาวิธีการแก้ไขปัญหาและออกแบบระบบเครือข่ายใหม่ อุปกรณ์แต่ละตัวสามารถดูรายละเอียดคอนฟิกได้จากการใช้คำสั่ง **show run** ซึ่งในการศึกษาคอนฟิกของอุปกรณ์แต่ละตัวนั้นจะต้องศึกษาควบคู่ไปกับการดูสายที่เชื่อมต่อของแต่ละตัวอุปกรณ์จริงๆด้วยว่าเชื่อมต่อไปที่ใด ตรงกับคอนฟิกหรือไม่

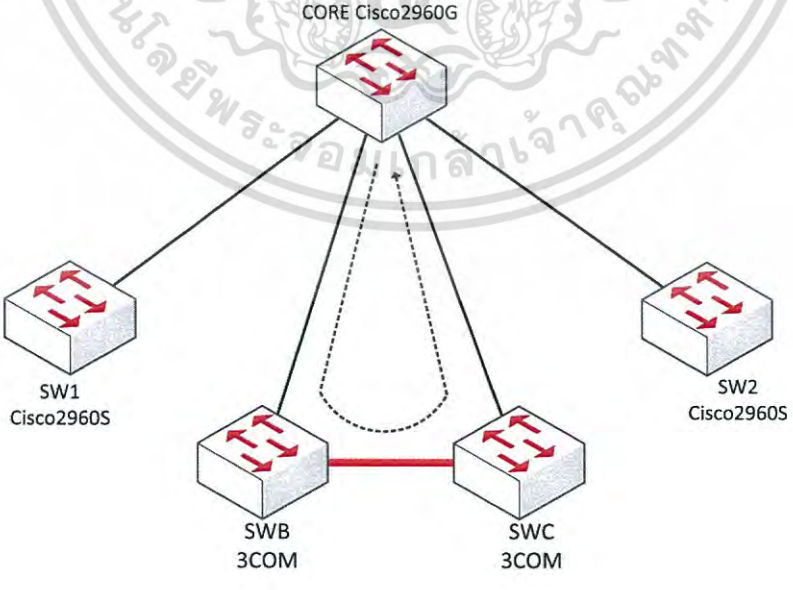
สำหรับโครงงานสหกิจศึกษาครั้งนี้ การเชื่อมต่อระบบเครือข่ายเดิมของลูกค้าซึ่งได้จากการศึกษา Network Diagram ก่อนเกิดปัญหาแสดงให้เห็นว่าบริษัทของลูกค้ามีระบบเครือข่ายเชื่อมต่อกันอยู่ระหว่างที่ Head office และที่ Branch office แสดงดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 Network Diagram ของระบบเครือข่ายเดิม

เมื่อศึกษาการเชื่อมต่อหลังจากเกิดปัญหา ทั้งจากการสำรวจสายที่เชื่อมต่อระหว่างอุปกรณ์ รวมถึงศึกษาจากคอนฟิกของแต่ละตัวอุปกรณ์ พบว่ามีการเปลี่ยนแปลงเกิดขึ้นที่ Abdulrahim Place โดยมีการเชื่อมต่อเพิ่มขึ้นมาในลักษณะลูประหว่างสวิตช์ CORE, SWB และ SWC ซึ่งอาจเกิดจากการที่มีผู้ใช้งานเชื่อมต่ออุปกรณ์เข้าหากันโดยขาดความรู้และรอบคอบ แสดงการเชื่อมต่อดังรูปที่ 3.3



รูปที่ 3.3 แสดงการเชื่อมต่อหลังเกิดปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดคุณสมบัติและการเชื่อมต่อของสวิตช์ในระบบเครือข่ายเก่ามีดังนี้

1) สวิตช์ Cisco รุ่น 2960G

รูปที่ 3.4 แสดงลักษณะสวิตช์ Cisco รุ่น 2960G

คุณสมบัติ

Ports : 48 ports

Cabling Type : Ethernet 1000Base-T, Ethernet 100Base-TX, Ethernet 10Base-T

Frequency Required : 50/60 Hz

Encryption Algorithm : SSL

Switching capacity : 32 Gbps

Sample Features : Access Control List (ACL), Multiple Spanning Tree Protocol (MSTP), Quality of Service (QoS), IPv6 support, load balancing

ตารางที่ 3.1 แสดงการเชื่อมต่อของสวิตช์ CORE

Node	Port	Remote Node	Port
CORE	Gi0/1	SW1	Gi0/1
CORE	Gi0/2	SWB	Gi0/1
CORE	Fa0/1	SWC	Gi0/1
CORE	Fa0/2	SW2	Gi0/1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) สวิตช์ Cisco 2960S-24TS-L

รูปที่ 3.5 แสดงลักษณะสวิตช์ Cisco รุ่น 2960S-24TS-L

คุณสมบัติ

Ports : 24 ports

Cabling Type : Ethernet 10Base-T/100Base-TX, console

RAM : 128 MB

Encryption Algorithm : SSL

Switching capacity : 176 Gbps

Sample Features : Access Control List (ACL), Multiple Spanning Tree Protocol (MSTP), Quality of Service (QoS), IPv6 support, load balancing

2.1) สวิตช์ Cisco 2960S-24TS-L (SW1)

ตารางที่ 3.2 แสดงการเชื่อมต่อของสวิตช์ Cisco 2960S-24TS-L (SW1)

Node	Port	Remote Node	Port
SW1	Gi1/0/1	CORE	Gi0/1

2.2) สวิตช์ Cisco 2960S-24TS-L (SW2)

ตารางที่ 3.3 แสดงการเชื่อมต่อของสวิตช์ Cisco 2960S-24TS-L (SW2)

Node	Port	Remote Node	Port
SW2	Gi1/0/1	CORE	Fa0/2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) สวิตช์ 3COM รุ่น 3C16476CS

รูปที่ 3.6 แสดงลักษณะสวิตช์ 3COM รุ่น 3C16476CS

คุณสมบัติ

Ports : 48 ports

Primary Port Speed : 10/100Mbps + 1000Mbps

Standards : IEEE 802.1d Spanning Tree, IEEE 802.1p Priority Tags, IEEE 802.1X Port Security, IEEE 802.3 Ethernet, IEEE 802.3u Fast Ethernet

3.1) สวิตช์ 3COM (SWB)

ตารางที่ 3.4 แสดงการเชื่อมต่อของสวิตช์ 3COM (SWB)

Node	Port	Remote Node	Port
SWB	Gi0/1	CORE	Gi0/2
SWB	Gi0/2	SWC	Gi0/2

3.2) สวิตช์ 3COM (SWC)

ตารางที่ 3.5 แสดงการเชื่อมต่อของสวิตช์ 3COM (SWC)

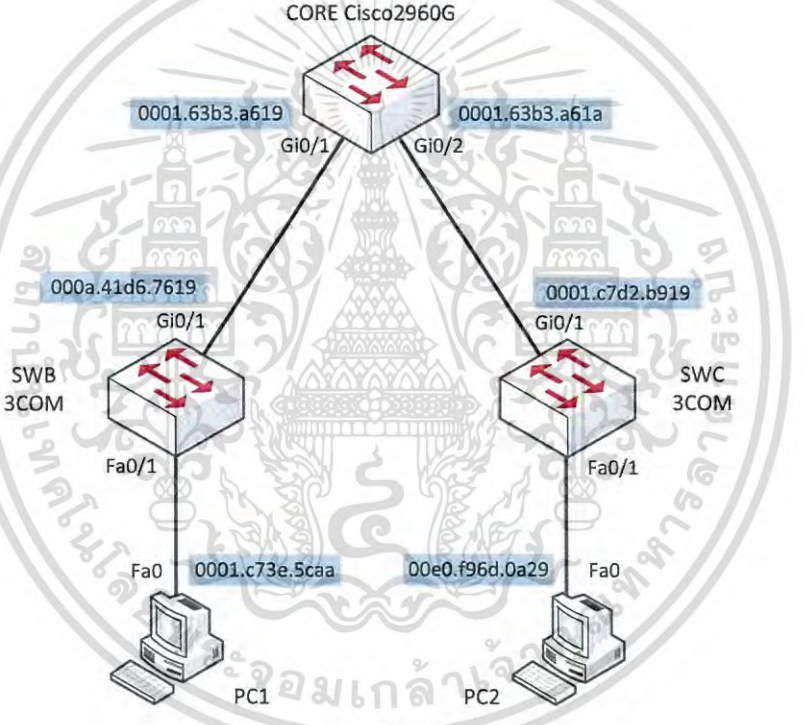
Node	Port	Remote Node	Port
SWC	Gi0/1	CORE	Fa0/1
SWC	Gi0/2	SWB	Gi0/2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การตรวจสอบการเกิดลูป

จากการศึกษาโครงสร้างการเชื่อมต่อของระบบเก่าทั้งก่อนเกิดลูปและหลังเกิดลูปทำให้ทราบได้ว่าระบบมีการเชื่อมต่ออย่างไร และมีการเปลี่ยนแปลงอย่างไรเทียบระบบเดิมกับปัจจุบัน แต่วิธีการดังกล่าวอาจไม่สามารถทำให้ทราบได้ว่าปัญหาลูปนั้นเกิดที่อุปกรณ์ใดและส่งผลถึงอุปกรณ์ใดบ้าง ซึ่งปัญหาดังกล่าวสามารถตรวจสอบได้จาก 3 วิธีการหลักๆ คือ การใช้คำสั่ง Ping บน ICMP, การตรวจสอบ Mac address ของอุปกรณ์ และการตรวจสอบแบนด์วิธในการรับส่งข้อมูล

ในการแสดงผลการตรวจสอบการเกิดลูปด้วย 3 วิธีการดังกล่าวจะจำลองการเชื่อมต่อโดยการเชื่อมต่อคอมพิวเตอร์เข้ากับสวิตช์ทั้ง 2 จำนวนอย่างละ 1 เครื่อง โดยก่อนเกิดลูปจะได้การเชื่อมต่อระบบเครือบข่ายดังนี้



รูปที่ 3.7 แสดงการเชื่อมต่อก่อนเกิดลูป

แต่ละพอร์ตของอุปกรณ์จะมีเลข MAC address เป็นของตัวเอง ซึ่งแต่ละพอร์ตจะมีเลข MAC address ที่ไม่เท่ากัน รายละเอียดการเชื่อมต่อของอุปกรณ์สามารถแสดงได้ดังตารางที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

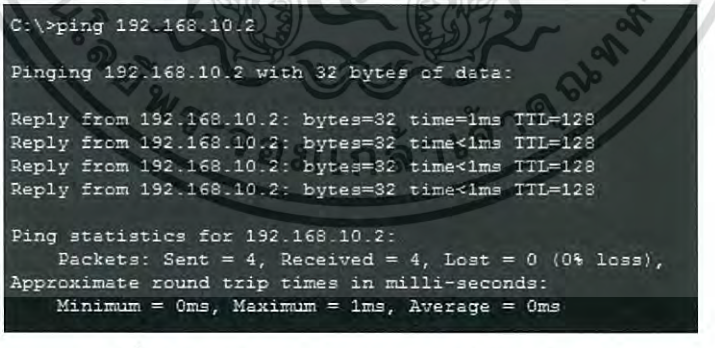
ตารางที่ 3.6 แสดงรายละเอียดการเชื่อมต่อของอุปกรณ์

Node	Port	MAC address	IP Address	Remote Node	Port
CORE	Gi0/1	0001.63B3.A619	-	Connect to SWB	Gi0/1
	Gi0/2	0001.63B3.A61A	-	Connect to SWC	Gi0/2
SWB	Gi0/1	000A.41D6.7619	-	Connect to CORE	Gi0/1
	Gi0/2	000A.41D6.761A	-	-	-
SWC	Gi0/1	0001.C7D2.B919	-	Connect to CORE	Gi0/2
	Gi0/2	0001.C7D2.B91A	-	-	-
PC1	Fa0	0001.C73E.5CAA	192.168.10.1	Connect to SW1	Fa0/1
PC2	Fa0	00E0.F96D.0A29	192.168.10.2	Connect to SwitchB	Fa0/1

1) ขั้นตอนการ Ping ก่อนที่จะเกิดลูป

เมื่ออุปกรณ์แต่ละตัวมีการเชื่อมต่อเข้าหากันโดยไม่มีการเกิดลูป ทำให้ PC1 สามารถติดต่อรับส่งข้อมูลถึง PC2 ได้ ในขั้นตอนการตรวจสอบจะใช้ ICMP โดยการใช้คำสั่ง ping จาก PC1 ไป PC2 และจาก PC2 ไป PC1 ผลที่ได้เป็นดังนี้

1.1 PC1



รูปที่ 3.8 แสดงผลการ ping จาก PC1 ไป PC2

จากผลการ ping ของ PC1 ที่ส่งไปที่ IP Address 192.168.10.2 ของ PC2 แสดงให้เห็นว่าสามารถส่งข้อมูลถึงกันได้ จากการที่มี Reply From ตอบกลับมาจาก PC2 ซึ่ง bytes = 32 หมายถึงจำนวนไบต์ที่ส่งกลับมา, time = 1ms หมายถึงเวลาที่ใช้ในการตอบกลับ มีหน่วยเป็นมิลลิวินาที, TTL = 120 TTL (Time To Live) หมายถึงจุดผ่านเราเตอร์ต่างๆ ก่อนจะเดินไปถึงเป้าหมาย TTL, Packet ที่ส่งไปและที่สามารถรับได้เท่ากับ 4 ซึ่งหมายถึงไม่มีการสูญเสียเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 PC2

```
C:\>ping 192.168.10.1

Pinging 192.168.10.1 with 32 bytes of data:

Reply from 192.168.10.1: bytes=32 time<1ms TTL=128
Reply from 192.168.10.1: bytes=32 time<1ms TTL=128
Reply from 192.168.10.1: bytes=32 time<1ms TTL=128
Reply from 192.168.10.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

รูปที่ 3.9 แสดงผลการ ping จาก PC1 ไป PC2

จากผลการ ping ของ PC2 ที่ส่งไปที่ IP Address 192.168.10.1 ของ PC1 แสดงให้เห็นว่าสามารถส่งข้อมูลถึงกันได้เช่นเดียวกับ PC1 มา PC2

2) ขั้นตอนการตรวจสอบ MAC address ก่อนที่จะเกิดลูป

ปัญหาหนึ่งที่เกิดขึ้นหลังจากการเกิดลูปคือ MAC Database Instability หรือการที่อุปกรณ์ได้รับ MAC address ที่ผิดเพี้ยนไป สามารถแสดง MAC address ก่อนและหลังเกิดลูปได้โดยการใช้คำสั่ง **show mac-address-table** สามารถแสดง MAC address Table ของอุปกรณ์ที่เชื่อมต่ออยู่ได้ดังนี้

2.1 CORE

```
Core#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.c73e.5caa    DYNAMIC     Gig0/1
   1    0001.c7d2.b919    DYNAMIC     Gig0/2
   1    000a.41d6.7619    DYNAMIC     Gig0/1
   1    00e0.f96d.0a29    DYNAMIC     Gig0/2
```

รูปที่ 3.10 แสดง MAC address Table ของสวิตช์ CORE ก่อนเกิดลูป

จากรูปที่ 3.10 MAC address Table ของสวิตช์ CORE ทางพอร์ต Gig0/1 มีการรับ MAC address เข้ามาสองค่าคือ 000a.41d6.7619 (พอร์ต Gig0/1 ของสวิตช์ SWB) ซึ่งต่อโดยตรงกับสวิตช์ CORE และ 0001.c73e.5caa (พอร์ต Fa0 ของ PC1) ซึ่งต่อกับสวิตช์ SWB ทางพอร์ต Gi0/2 มีการรับ MAC address เข้ามาสองค่าคือ 0001.c7d2.b919 (พอร์ต Gig0/1 ของสวิตช์ SWC) ซึ่งต่อโดยตรง และ 00e0.f96d.0a29 (พอร์ต Fa0 ของ PC2) ซึ่งต่อกับสวิตช์ SWC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 SWB

```
SwB#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.63b3.a619    DYNAMIC     Gig0/1
   1    0001.c73e.5caa    DYNAMIC     Fa0/1
   1    00e0.f96d.0a29    DYNAMIC     Gig0/1
```

รูปที่ 3.11 แสดง MAC address Table ของสวิตช์ SWB ก่อนเกิดลูป

จากรูปที่ 3.11 MAC address Table ของสวิตช์ SWB ทางพอร์ต Gig0/1 มีการรับ MAC address เข้ามาสองค่าคือ 0001.63b3.a619 (พอร์ต Gig0/1 ของสวิตช์ CORE) ซึ่งต่อโดยตรงกับสวิตช์ SWB และ 00e0.f96d.0a29 (พอร์ต Fa0 ของ PC2) ซึ่งต่อกับสวิตช์ SWC ที่ต่อเข้ากับสวิตช์ CORE ทางพอร์ต Fa0/1 มีการรับ MAC address เข้ามาค่าเดียวคือ 0001.c73e.5caa (พอร์ต Fa0 ของ PC1)

2.3 SWC

```
SwC#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.63b3.a61a    DYNAMIC     Gig0/1
   1    0001.c73e.5caa    DYNAMIC     Gig0/1
   1    00e0.f96d.0a29    DYNAMIC     Fa0/1
```

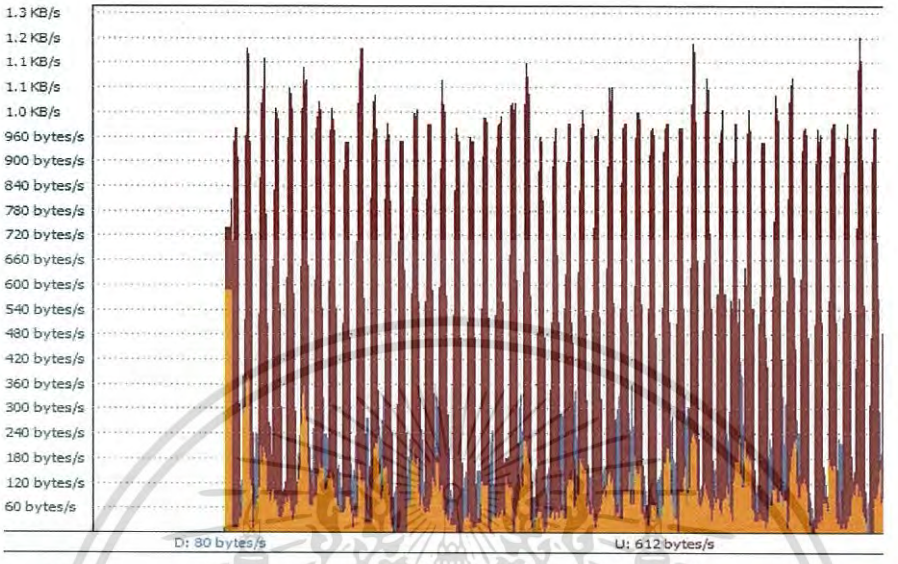
รูปที่ 3.12 แสดง MAC address Table ของสวิตช์ SWC ก่อนเกิดลูป

จากรูปที่ 3.12 MAC Address Table ของสวิตช์ SWC ทางพอร์ต Gig0/1 มีการรับ MAC address เข้ามาสองค่าคือ 0001.63b3.a61a (พอร์ต Gig0/2 ของสวิตช์ CORE) ซึ่งต่อโดยตรงกับสวิตช์ SWC และ 0001.c73e.5caa (พอร์ต Fa0 ของ PC1) ซึ่งต่อกับสวิตช์ SWB ที่ต่อเข้ากับสวิตช์ CORE ทางพอร์ต Fa0/1 มีการรับ MAC address เข้ามาค่าเดียวคือ 00e0.f96d.0a29 (พอร์ต Fa0 ของอุปกรณ์ PC2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ขั้นตอนการตรวจสอบแบนด์วิธก่อนที่จะเกิดลูป

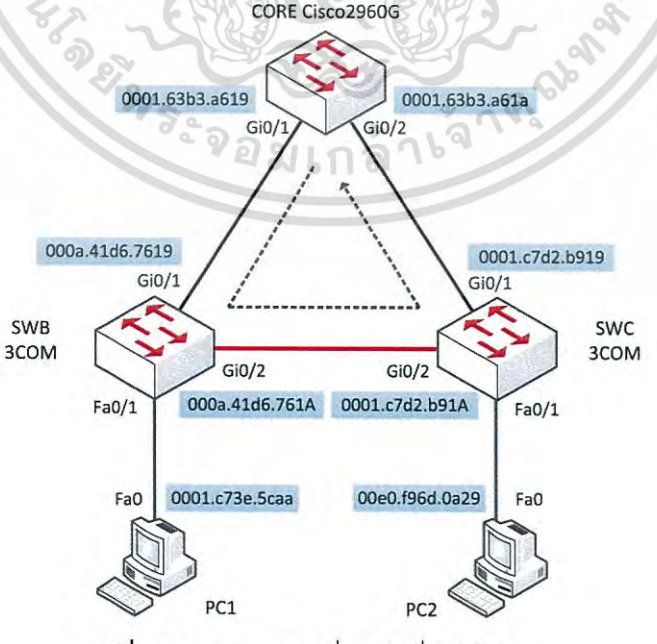
อัตราการการรับส่งข้อมูลบนแบนด์วิธจาก PC1 ก่อนที่จะเกิดลูป สามารถตรวจสอบได้จากโปรแกรม NetWorx ผลที่ได้แสดงดังรูปที่ 3.13



รูปที่ 3.13 กราฟแสดงอัตราการรับส่งข้อมูลก่อนเกิดลูปจากโปรแกรม Networx

ผลจากกราฟพบว่ามีอัตราการรับข้อมูลสูงสุดอยู่ที่ประมาณ 360 ไบต์ต่อวินาที (เส้นสีฟ้า) และอัตราการส่งข้อมูลสูงสุดอยู่ที่ประมาณ 1200 ไบต์ต่อวินาที (เส้นสีแดง)

เมื่อมีการเชื่อมต่อสวิตช์ SWB ทางพอร์ต Gig0/2 เข้ากับพอร์ต Gig0/2 ของสวิตช์ SWC จึงทำให้เกิดลูปขึ้น ลักษณะการเชื่อมต่อแบบที่ทำให้เกิดลูปแสดงดังรูปที่ 3.14



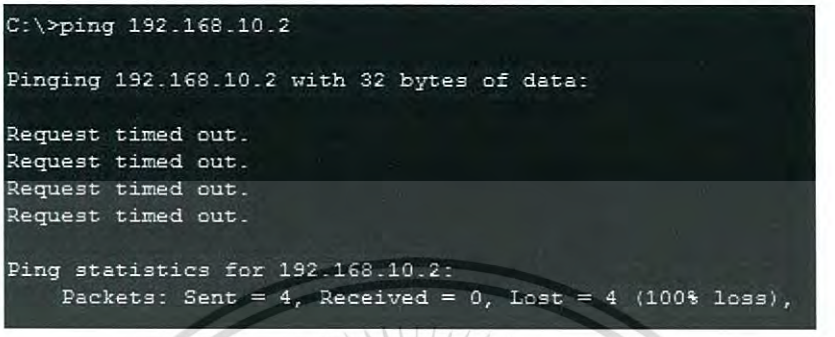
รูปที่ 3.14 แสดงการเชื่อมต่อที่ทำให้เกิดลูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ขั้นตอนการ Ping หลังจากเกิดลูป

ในขั้นตอนการทดสอบการเชื่อมต่อระหว่าง PC1 และ PC2 โดยการ ping ผลที่ได้หลังจากการเกิดลูปเป็นดังนี้

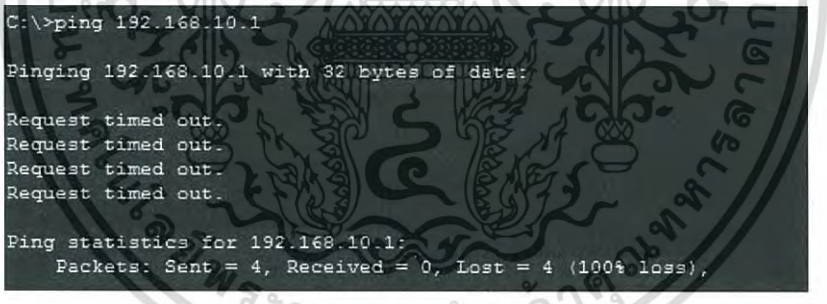
1.1 PC1



รูปที่ 3.15 แสดงผลการ ping จาก PC1 ไป PC2

จากรูปที่ 3.15 ผลการ ping ของ PC1 ที่ส่งไป IP Address 192.168.10.2 ของ PC2 ข้อความที่แสดงคือ Request timed out แสดงให้เห็นว่าไม่มีมีการตอบรับจาก PC2 โดย Packet ที่ส่งไปเท่ากับ 4 แต่ที่ได้รับนั้นเท่ากับ 0 ซึ่งเกิดการสูญเสียเท่ากับ 4 หรือ 100%

1.2 PC2



รูปที่ 3.16 แสดงผลการ ping จาก PC2 ไป PC1

จากรูปที่ 3.16 ผลการ ping ของ PC2 ที่ส่งไป IP Address 192.168.10.1 ของ PC1 ข้อความที่แสดงคือ Request timed out แสดงให้เห็นว่าไม่มีมีการตอบรับจาก PC1 โดย Packet ที่ส่งไปเท่ากับ 4 แต่ที่ได้รับนั้นเท่ากับ 0 ซึ่งเกิดการสูญเสียเท่ากับ 4 หรือ 100%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ขั้นตอนการตรวจสอบ MAC address หลังจากเกิดลูป

เมื่อมีการเกิดลูปขึ้น สวิตช์จะมีการรับ MAC address ผิดเพี้ยนไป รายละเอียด MAC address ในตาราง MAC address Table ของอุปกรณ์ที่เชื่อมต่ออยู่หลังจากเกิดลูปแสดงดังนี้

2.1 CORE

```
Core#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.c73e.5caa    DYNAMIC     Gig0/2
   1    0001.c7d2.b919    DYNAMIC     Gig0/2
   1    000a.41d6.7619    DYNAMIC     Gig0/1
   1    00e0.f96d.0a29    DYNAMIC     Gig0/2
```

รูปที่ 3.17 แสดง Mac address Table ของสวิตช์ CORE หลังเกิดลูป

จากรูปที่ 3.17 MAC address Table ของสวิตช์ CORE หลังเกิดลูป ทางพอร์ต Gig0/1 มีการรับ MAC address เข้ามาเหลือเพียงค่าเดียวคือ 000a.41d6.7619 (พอร์ต Gig0/1 ของสวิตช์ SWB) ซึ่งต่อโดยตรงกับสวิตช์ CORE

ทางพอร์ต Gig0/2 มีการรับ MAC address เข้ามาสามค่าคือ 0001.c7d2.b919 (พอร์ต Gi0/1 ของสวิตช์ SWC), 00e0.f96d.0a29 (พอร์ต Fa0 ของ PC2) ซึ่งต่อกับสวิตช์ SWC และสุดท้ายคือ 0001.c73e.5caa (พอร์ต Fa0 ของ PC1) โดยก่อนที่จะเกิดลูป MAC address นี้เข้ามาทางพอร์ต Gig0/1 ที่เชื่อมต่อกับสวิตช์ SWB

2.2 SWB

```
SwB#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.63b3.a619    DYNAMIC     Gig0/1
   1    0001.c73e.5caa    DYNAMIC     Fa0/1
   1    0001.c7d2.b919    DYNAMIC     Gig0/1
   1    0001.c7d2.b91a    DYNAMIC     Gig0/2
   1    00e0.f96d.0a29    DYNAMIC     Gig0/2
```

รูปที่ 3.18 แสดง Mac address Table ของสวิตช์ SWB หลังเกิดลูป

จากรูปที่ 3.18 MAC address Table ของสวิตช์ SWB หลังเกิดลูป ทางพอร์ต Gig0/1 มีการรับ MAC address เข้ามาสองค่าคือ 0001.63b3.a619 (พอร์ต Gig0/1 ของสวิตช์ CORE) ซึ่งต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตรงกับสวิตช์ SWB และ 0001.c7d2.b919 (พอร์ต Gig0/1 ของสวิตช์ SWC) ซึ่งต่อกับสวิตช์ CORE ทางพอร์ต Gig0/2 ไม่ใช่ Gig0/1

พอร์ต Gig0/2 มีการรับ MAC address เข้ามาสองค่าคือ 0001.c7d2.b91a (พอร์ต Gig0/2 ของสวิตช์ SWC) และ 00e0.f96d.0a29 (พอร์ต Fa0 ของ PC2) ซึ่งต่อกับสวิตช์ SWC

พอร์ต Fa0/1 มีการรับ MAC address เข้ามาค่าเดียวคือ 0001.c73e.5caa (พอร์ต Fa0 ของ PC1) ซึ่งต่อโดยตรงกับสวิตช์ SWB

2.3 SWC

```
SwC#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.63b3.a61a    DYNAMIC     Gig0/1
   1    0001.c73e.5caa    DYNAMIC     Gig0/2
   1    0001.c7d2.b919    DYNAMIC     Gig0/2
   1    000a.41d6.761a    DYNAMIC     Gig0/2
   1    00e0.f96d.0a29    DYNAMIC     Fa0/1
```

รูปที่ 3.19 แสดง MAC address Table ของสวิตช์ SWC หลังเกิดลูป

จากรูปที่ 3.19 MAC address Table ของสวิตช์ CORE ทางพอร์ต Gi0/1 มีการรับ MAC address เข้ามาเหลือเพียงค่าเดียวคือ 0001.63b3.a61a (พอร์ต Gi0/2 ของสวิตช์ CORE) ซึ่งต่อโดยตรงกับสวิตช์ SWC และ 0001.c73e.5ccaa (พอร์ต Fa0 ของ PC1) ซึ่งต่อกับสวิตช์ SWB

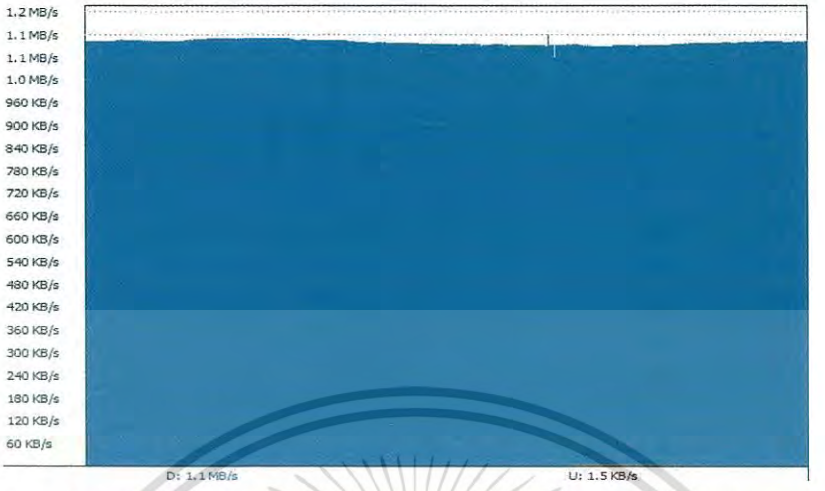
พอร์ต Gi0/2 มีการรับ MAC address เข้ามาสามค่าคือ 000a.41d6.761a (พอร์ต Gi0/2 ของสวิตช์ SWB) ซึ่งต่อโดยตรงกับสวิตช์ SWB, 0001.c73e.5caa (พอร์ต Fa0 ของ PC1) ซึ่งต่อกับสวิตช์ SWB โดยตอนแรกเข้ามาทางพอร์ต Gig0/1 ของสวิตช์ SWC และ 0001.c7d2.b919 (พอร์ต Gig0/1 ของตัวสวิตช์ SWC เอง) ที่วนมาจากสวิตช์ CORE และเข้ามาทางสวิตช์ SWB

จาก MAC address Table ก่อนและหลังเกิดลูปของสวิตช์ทั้งสามตัว แสดงให้เห็นว่ามีการเรียนรู้และรับ MAC address ที่ผิดเพี้ยนไปจากเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

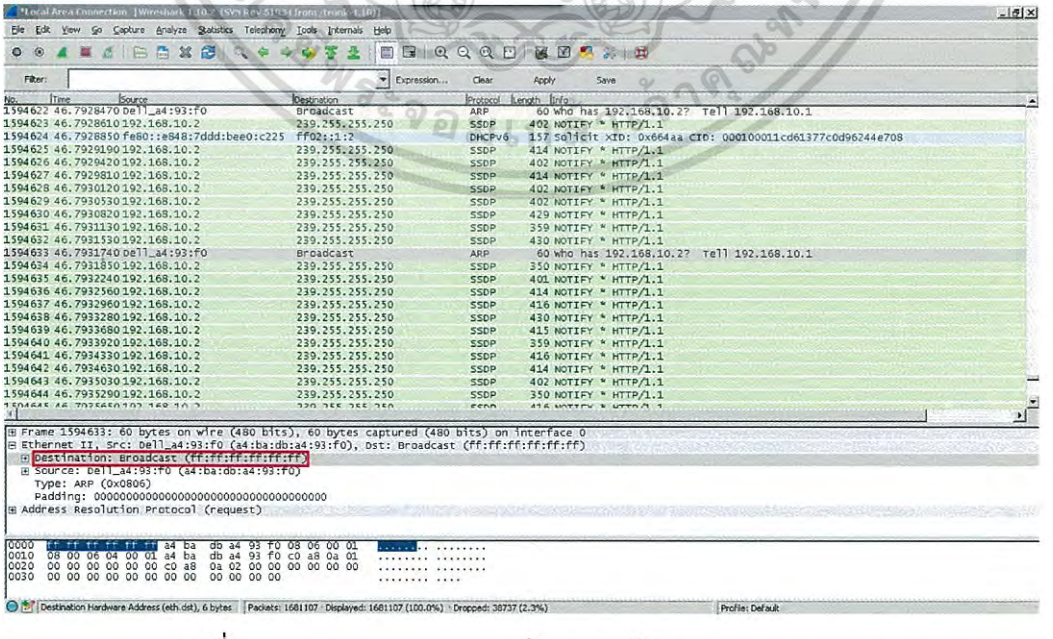
3) ขั้นตอนการตรวจสอบแบนด์วิธหลังจากเกิดลูป

อัตราการการรับส่งข้อมูลจาก PC1 หลังจากที่เกิดลูป ผลที่ได้เป็นดังรูปที่ 3.20



รูปที่ 3.20 กราฟแสดงอัตราการรับส่งข้อมูลหลังเกิดลูปจากโปรแกรม Networx

ผลจากกราฟพบว่ามีอัตราการรับข้อมูลหรือ Download สูงสุดอยู่ที่ประมาณ 1100000 ไบต์ต่อวินาที (เส้นสีฟ้า) และอัตราการส่งข้อมูลหรือ Upload สูงสุดอยู่ที่ประมาณ 1500 ไบต์ต่อวินาที (เส้นสีแดง) ซึ่งอัตราการรับข้อมูลมีการเพิ่มขึ้นมามากจากก่อนที่จะเกิดลูป แสดงให้เห็นว่า มีการรับข้อมูลเข้ามาในรูปแบบที่ผิดปกติเนื่องจากเมื่อเกิดลูปขึ้น สวิตช์มีการ “Flood” หรือส่งข้อมูลออกมาทุกพอร์ต อุปกรณ์ต่างๆจึงรับข้อมูลนั้นเข้ามาอย่างหลีกเลี่ยงไม่ได้ และในการตรวจสอบข้อมูลในการรับส่ง ไม่ว่าจะเป็นที่มาหรือปลายทางของข้อมูล รวมถึงโปรโตคอลที่ใช้ในการรับส่ง สามารถตรวจสอบได้จากโปรแกรม Wireshark แสดงได้ดังรูปที่ 3.21



รูปที่ 3.21 แสดงการตรวจสอบข้อมูลจากโปรแกรม Wireshark

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

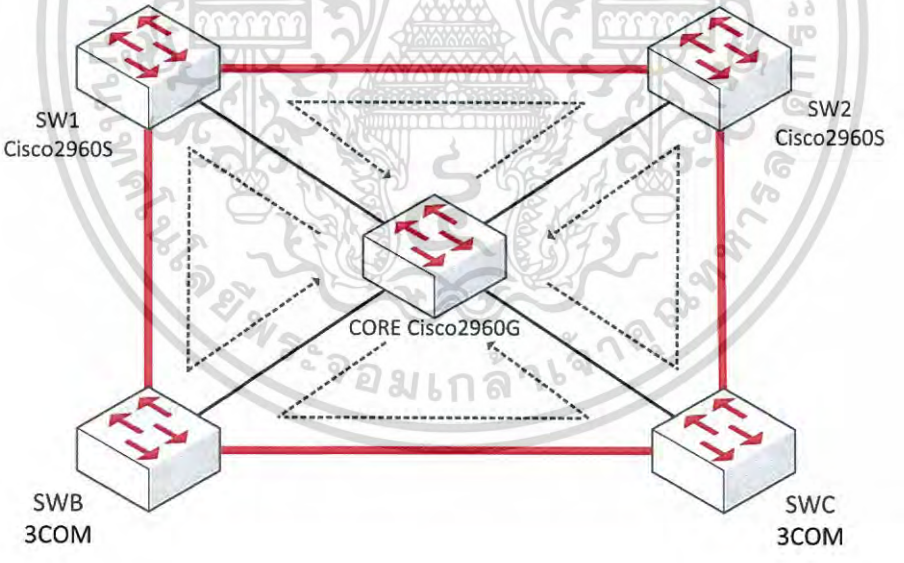
จากข้อมูล packet ในการรับส่งข้อมูล แสดงให้เห็นว่ามีการ Broadcast เกิดขึ้นในระบบ ซึ่งโปรโตคอลที่ใช้นั้นคือ ARP โดยมี Destination คือ FF:FF:FF:FF:FF:FF ซึ่งเป็น Destination ที่เกิดขึ้นเนื่องจาก Broadcast

3.2 ขั้นตอนการออกแบบระบบเครือข่าย

3.2.1 การออกแบบระบบเครือข่าย

ทำการออกแบบปรับปรุงพัฒนาระบบเครือข่ายเพื่อแก้ไขปัญหาที่เกิดขึ้นจากระบบเครือข่ายเดิม เพื่อเพิ่มประสิทธิภาพและแก้ไขปัญหาของระบบเครือข่ายในปัจจุบันให้สามารถรองรับการทำงานในอนาคตได้โดยอ้างอิงจากอุปกรณ์เดิมที่มีอยู่และเพิ่มอุปกรณ์ใหม่แทนของเดิมที่พบว่าเป็นสาเหตุที่ทำให้เกิดปัญหาการลูปขึ้นในระบบ การออกแบบระบบเครือข่ายจำเป็นต้องทำงานอย่างเป็นระบบเพื่อให้ได้ระบบเครือข่ายที่ตรงตามความต้องการสามารถใช้งานได้จริงและคุ้มค่ากับการลงทุน

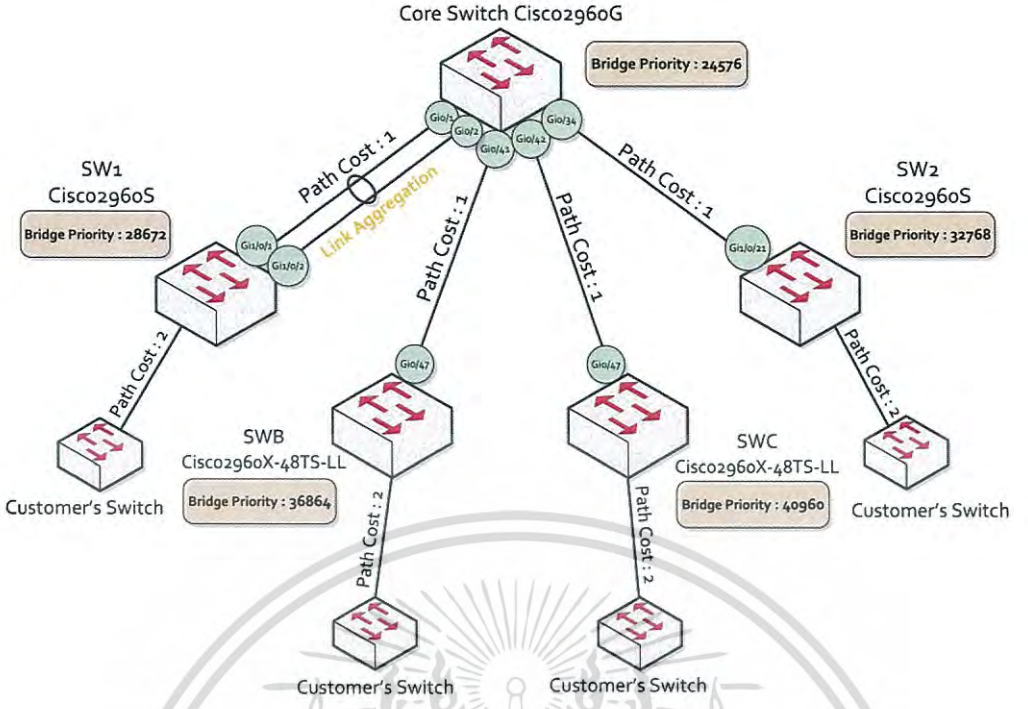
จากการศึกษาโครงสร้างการเชื่อมต่อของระบบ รวมถึงปัญหาและสาเหตุของการเกิดลูป พบว่าการเชื่อมต่อของสวิตช์ทั้ง 5 ตัว มีโอกาสที่จะถูกเชื่อมต่อเข้าหากันจนเป็นลักษณะลูปได้ทั้งหมดทุกตัว แสดงดังรูปที่ 3.22



รูปที่ 3.22 แสดงลักษณะการเชื่อมต่อที่สามารถทำให้เกิดลูปขึ้นของสวิตช์ทั้ง 5 ตัว

โดยระบบเครือข่ายใหม่มีการนำ STP เข้ามาแก้ไขปัญหาลูปที่เกิดขึ้น ดังนั้นในการทำ STP จึงจำเป็นต้องทำให้กับสวิตช์ทั้ง 5 ตัว ในการออกแบบระบบเครือข่ายใหม่แสดงดังรูปที่ 3.23

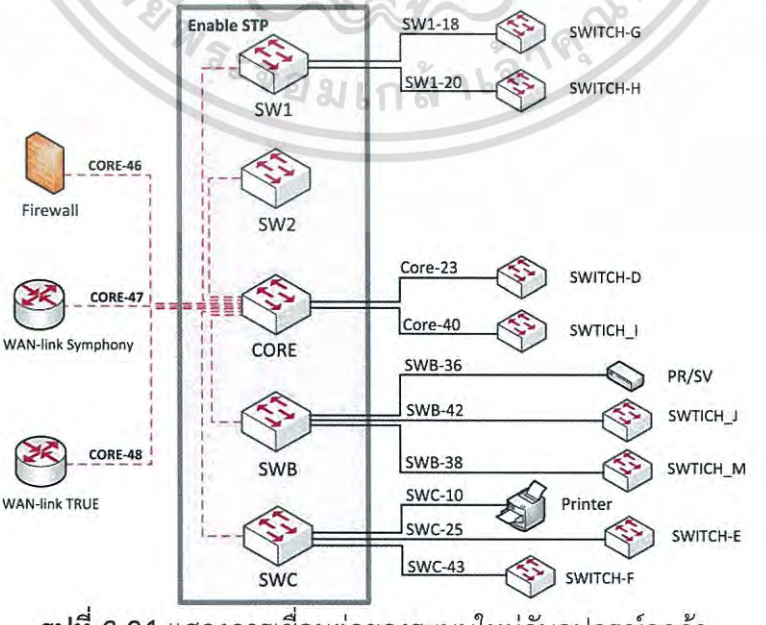
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 แสดงการเชื่อมต่อของระบบเครือข่ายใหม่

เนื่องจากสวิตช์ SWB และสวิตช์ SWC เป็นสวิตช์ยี่ห้อ 3COM มีพอร์ตในการใช้งานจำนวน 48 พอร์ต ซึ่งการใช้สวิตช์ต่างยี่ห้ออาจก่อให้เกิดผิดพลากในการติดต่อสื่อสาร เพราะค่าพื้นฐานของอุปกรณ์อาจมีค่าที่แตกต่างกัน ในระบบใหม่จึงมีการนำสวิตช์ยี่ห้อ Cisco รุ่น 2960X-48TS-LL ซึ่งมีจำนวน 48 พอร์ต เข้ามาใช้แทนสวิตช์ 3COM ซึ่งเป็นการรองรับการใช้งานของระบบเดิม

โดยสวิตช์แต่ละตัวมีการเชื่อมต่อไปยังอุปกรณ์ตัวอื่นๆของลูกค้าแสดงดังรูปที่ 3.24



รูปที่ 3.24 แสดงการเชื่อมต่อของระบบใหม่กับอุปกรณ์ลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การตั้งค่า Bridge Priority

กำหนดให้สวิตช์ CORE เป็น Root Bridge จึงมีค่า Priority ที่น้อยที่สุด เนื่องจากเป็นตัวหลักในการเชื่อมต่อสวิตช์ตัวอื่นๆเข้าด้วยกัน ซึ่งค่า Priority ดีฟอลต์มีค่าเท่ากับ 32768 โดยการกำหนดค่าจะเริ่มที่ 0 และเพิ่มขึ้นทีละ 4096 ซึ่งสวิตช์ CORE มีการกำหนดค่าไว้ที่ 24576 สวิตช์ตัวที่ไม่ใช่ Root Bridge จึงจะมีค่า Priority ที่มากกว่า โดยเพิ่มขึ้นตัวละ 4096 ทั้งนี้เพื่อเป็นการป้องกันในกรณีที่มีการนำสวิตช์ตัวอื่นที่ไม่ได้ตั้งค่า Priority ไว้มาเสียบเข้าหากัน แน่นอนว่าสวิตช์ CORE จะมีค่า Priority ที่ต่ำที่สุด

- การตั้งค่า Path Cost ประจำพอร์ต

กำหนดให้พอร์ตที่เชื่อมต่อกับสวิตช์ CORE มีค่า Path Cost เท่ากับ 1 ซึ่งเป็นเส้นทางหลัก รวมถึงพอร์ตที่เชื่อมต่อกับ Firewall และ เราเตอร์ ที่เชื่อมต่อกับ WAN ส่วนฝั่งที่เชื่อมต่อกับสวิตช์ของลูกค้าหรือพอร์ตที่มีการใช้งานอยู่ จะกำหนดให้เท่ากับ 2 เนื่องจากเป็นพอร์ตที่มีโอกาสน้อยในการจะถูกนำมาเสียบให้เข้ากับสวิตช์อีกตัวและทำให้เกิดลูปขึ้น และพอร์ตปกติของสวิตช์แต่ละตัวเป็นพอร์ต Gigabitethernet ซึ่งมีค่าแบนด์วิธเท่ากับ 1 Gbps หรือค่า Path Cost ดีฟอลต์เท่ากับ 4 จึงตั้งค่าพอร์ตที่เป็นเส้นทางหลักหรือมีการใช้งานอยู่ให้มีค่าน้อยกว่า 4 รายละเอียดการเชื่อมต่อของสวิตช์และการกำหนดค่า Cost แสดงดังตารางที่ 3.7

ตารางที่ 3.7 แสดงรายละเอียดการเชื่อมต่อของสวิตช์ CORE

Node	Port	Remote Node	Cost
CORE	Gi0/1	Connect to SW1	1
	Gi0/2	Connect to SW1	1
	Gi0/23	Connect to SWITCH-D	1
	Gi0/34	Connect to SW2	1
	Gi0/40	Connect to SWITCH-I	2
	Gi0/41	Connect to SWB	1
	Gi0/42	Connect to SWC	1
	Gi0/46	Connect to Firewall	1
	Gi0/47	Connect to WAN-link Symphony	1
	Gi0/48	Connect to WAN-link TRUE	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 แสดงรายละเอียดการเชื่อมต่อของสวิตช์ SW1

Node	Port	Remote Node	Cost
SW1	Gi1/0/1	Connect to CORE	1
	Gi1/0/2	Connect to CORE	1
	Gi1/0/18	Connect to SWITCH-G	2
	Gi1/0/20	Connect to SWITCH-H	2

ตารางที่ 3.9 แสดงรายละเอียดการเชื่อมต่อขอสวิตช์ SWB

Node	Port	Remote Node	Cost
SWB	Gi0/36	Connect to PR/SV	2
	Gi0/38	Connect to SWITCH-M	2
	Gi0/42	Connect to SWITCH-J	2
	Gi0/47	Connect to CORE	1

ตารางที่ 3.10 แสดงรายละเอียดการเชื่อมต่อของสวิตช์ SWC

Node	Port	Remote Node	Cost
SWC	Gi0/10	Connect to Printer	2
	Gi0/25	Connect to SWITCH-E	2
	Gi0/43	Connect to SWITCH-F	2
	Gi0/47	Connect to CORE	1

ตารางที่ 3.11 แสดงรายละเอียดการเชื่อมต่อของสวิตช์ SW2

Node	Port	Remote Node	Cost
SW2	Gi1/0/21	Connect to CORE	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การตั้งค่า Port Priority

จะเริ่มจาก 0 และเพิ่มขึ้นทีละ 16 ซึ่งในระบบนี้ไม่มีการตั้งค่า Port Priority ดังนั้นจึงมีค่าเป็นค่าดีฟอลต์ หรือเท่ากับ 128 เนื่องจากว่ามีการตั้งค่า Bridge Priority และค่า Path Cost ไว้ให้ไม่สามารถเท่ากันได้อยู่แล้ว จึงไม่จำเป็นต้องพิจารณาที่ค่า Port Priority อีก

- ส่วนของ VLAN

กำหนดให้ทุกพอร์ตบนสวิตช์ทุกตัวมี VLAN เป็นค่าดีฟอลต์ หรือก็คือ VLAN 1 ซึ่ง STP จะทำงานและมีผลเพียงบน VLAN เดียวกันเท่านั้น เนื่องจากไม่มีการกำหนดค่าให้กับ VLAN อื่นๆ หรือก็คือ 1 STP ต่อ 1 VLAN แต่ละ VLAN จะมี Root Bridge, Root Port, Designated Port เป็นของตนเอง ข้อดีในการตั้งค่ารูปแบบนี้คือ

1) เพื่อให้สามารถกระจายทราฟฟิคไปบนแต่ละเส้นทาง
2) เพื่อให้สามารถแบ่งโหลดของสวิตช์ที่ทำหน้าที่เป็น Root Bridge ด้วยการแบ่งให้สวิตช์ที่เหมาะสมทำหน้าที่เป็น Root Bridge ของแต่ละ VLAN แยกกันไป
3) เพื่อให้เกิดความเสถียรภาพ โดยจะช่วยป้องกันไม่ให้การแลกเปลี่ยนโทโพโลยีต่างๆ การมีปัญหาวงลูป และอื่นๆของ VLAN หนึ่งเข้ามามีผลกระทบกับอีก VLAN หนึ่ง

- EtherChannel

ทำในส่วนของสวิตช์ CORE กับ SW1 ที่พอร์ต Gig0/1 และ Gig0/2 และทำ Trunk ให้กับเส้นทางดังกล่าวด้วย เพื่อให้หลายๆ VLAN สามารถวิ่งผ่านได้พร้อมกันในหนึ่งลิงค์

3.2.2 อุปกรณ์ที่นำมาใช้

การเลือกใช้อุปกรณ์เครือข่ายให้เหมาะสมกับระบบงานนั้นเป็นสิ่งจำเป็น เพื่อให้การเลือกใช้อุปกรณ์นั้นได้เป็นไปอย่างประหยัดคุ้มค่าและถูกต้องตามหลักการทำงานจริง สามารถใช้งานร่วมกับระบบเครือข่ายเดิม รวมถึงทำให้ระบบเครือข่ายของระบบงานนั้นๆมีประสิทธิภาพ และสามารถแก้ไขปัญญาของระบบที่เกิดขึ้นได้ โดยอุปกรณ์ที่นำมาใช้ในระบบใหม่มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สวิตช์ Cisco รุ่น 2960X-48TS-LL

รูปที่ 3.25 แสดงลักษณะสวิตช์ Cisco รุ่น 2960X-48TS-LL

คุณสมบัติ

Ports : GigE 48 ports

Cabling Type : Ethernet 10Base-T/100Base-TX, console

Power Consumption Operational : 53 Watt

Encryption Algorithm : SSL

Sample Features : Multiple Spanning Tree Protocol (MSTP), Quality of Service (QoS), Port Security

3.3 ขั้นตอนการพัฒนาระบบ

3.3.1 แผนการดำเนินงาน

ตารางที่ 3.12 แสดงแผนการดำเนินงาน

No.	Tentative schedule / Main Task	June				July				August				September				October			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
	Meeting and Prepare config																				
1	Meeting and discuss solution																				
2	Design																				
3	Prepare config all equipment (Switch Cisco)																				
4	Prepare Proposal																				
	Migration Switch																				
5	Replace switch B and switch C with Cisco 2960																				
6	Move UTP on 3Com especially uplink that are connected with customer's switch																				
7	Adjust configuration on core switch																				
8	Adjust configuration on switch 1																				
9	Adjust configuration on switch 2 and change duplicated IP Address																				
10	Connect switch 2 with core switch																				
11	All switches in server room can ping to each others																				
12	Check state on switch that are not loop																				
13	Test loop prevention																				
14	Connect UTP																				
	1) core switch and switch 1																				
	2) core switch and switch 2																				
	3) core switch and switch B																				
	4) core switch and switch C																				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การคอนฟิก

1) การคอนฟิกพื้นฐานของสวิตช์

1.1) คำสั่งที่ใช้ในโหมด User Exec Mode (Switch >)

Switch> **enable** // คำสั่งเข้าสู่ privileged Exec mode (Switch#)

1.2) คำสั่งที่ใช้ได้ในโหมด Privileged Exec mode (Switch#)

Switch# **show ip route** // ใช้เพื่อแสดงข้อมูลที่อยู่ใน Routing Table

Switch# **show ip protocol** // ใช้แสดงโปรโตคอลในสวิตช์

Switch# **show mac-address-table** // ใช้แสดง MAC address อุปกรณ์ที่ต่อ

Switch# **show Ip interface brief** // ตรวจสอบสถานะของพอร์ต

*คำสั่งในการ show สามารถพิมพ์แบบย่อได้เป็น sh

Switch# **write** // เป็นคำสั่ง save

Switch# **erase startup-config** // เป็นคำสั่งลบข้อมูลทั้งหมดที่คอนฟิกไว้ในสวิตช์

Switch# **reload** // เป็นคำสั่ง reboot เครื่องใหม่

Switch# **ping** <IP ที่จะping> // คำสั่งในการทดสอบการส่งข้อมูลไปยัง ip นั้นๆ

Switch# **configure terminal** // เข้าสู่โหมด Global Configuration Mode

* สามารถพิมพ์แบบย่อได้เป็น conf t

1.3) คำสั่งที่ใช้ได้ในโหมด Global Configuration Mode (Switch(config)#)

Switch(config)# **hostname** <ชื่อสวิตช์> //เปลี่ยนชื่อสวิตช์

Switch(config)# **interface fastEthernet** <เลขพอร์ต>

Switch(config-f)# **ip address** <ip Address> <subnet mask> // คำสั่งกำหนด Ip Address ให้กับพอร์ต

1.4) คำสั่งตั้งหรัสผ่าน line vty และ line console สำหรับ Remote เข้ามาคอนฟิก

Switch(config)# **line console** 0 // กำหนดจำนวนผู้ที่สามารถ console เข้าไปได้

Switch(config)# **line vty** 0 4 // กำหนดจำนวนผู้ที่สามารถ Remote เข้าไปได้

Switch(config-line)# **password** <รหัส>

Switch(config-line)# **login**

Switch(config-line)# **exit**

Switch(config)# **enable secret** <รหัส> // การตั้งรหัสผ่าน

และคำสั่งในการทำ Routing เช่น RIP, RIPv2, EIGRP, และ Static ทำในโหมดนี้

2) วิธีการคอนฟิก EtherChannel สำหรับ STP

แต่ละพอร์ตที่ถูกทำเป็น EtherChannel จะต้องมีหมายเลข channel เดียวกัน ซึ่งจะต้องเข้าสู่โหมด Interface Config ก่อน และจึงใช้คำสั่ง group หรือ channel number ให้พอร์ตนั้นๆ คำสั่งมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Switch(config)# **interface fastethernet/gigabitethernet** <number>

Switch(config-if)# **channel-group** <channel-number>

ในระบบใหม่มีการทำ EtherChannel ที่พอร์ต Gi0/1 และ Gi0/2 ของสวิตช์ CORE และ SW1 มีการคอนฟิกดังนี้

- CORE : คำสั่งที่ใช้ได้แก่

CORE(config)# **interface range gigabitethernet** 0/1 - 2

CORE(config-if)# **channel-group** 2 **mode on**

- SW1 : คำสั่งที่ใช้ได้แก่

SW1(config)# **interface gigabitethernet** 0/1 -2

SW1(config-if)# **channel-group** 2 **mode on**

3) วิธีการคอนฟิก Trunk

เมื่อมีการกำหนดให้มีหลาย VLAN ในระบบ ดังนั้นการทำให้หลายๆ VLAN วิ่งผ่านได้พร้อมกันในหนึ่งคอนเน็กชัน จำเป็นต้องมีการทำ "Trunk Port" ซึ่งในการคอนฟิกจะต้องเข้าสู่โหมด Interface Config หรือ Port Channel ก่อน และจึงใช้คำสั่ง Trunk ในพอร์ตนั้นๆ คำสั่งในการทำ Trunk มีดังนี้

Switch(config)# **interface / port channel** <channel number>

Switch(config-if)# **switchport trunk encapsulation dot1q**

Switch(config-if)# **switchport mode trunk**

ในระบบใหม่มีการทำ Trunk ให้กับ EtherChannel group 2 หรือ พอร์ต Gi0/1 และ Gi0/2 ของสวิตช์ CORE และ SW1 มีการคอนฟิกดังนี้

- CORE : คำสั่งที่ใช้ได้แก่

CORE(config)# **port channel** 2

CORE(config-if)# **switchport trunk encapsulation dot1q**

CORE(config-if)# **switchport mode trunk**

- SW1 : คำสั่งที่ใช้ได้แก่

SW1(config)# **port channel** 2

SW1(config-if)# **switchport trunk encapsulation dot1q**

SW1(config-if)# **switchport mode trunk**

4) วิธีการคอนฟิกเพื่อเปิดการใช้งาน Spanning Tree

Switch(config)# **spanning-tree vlan** <หมายเลข VLAN>

ในระบบใหม่นี้มีการกำหนดให้เป็น VLAN 1 สามารถคอนฟิกได้ดังนี้

Switch(config)# **spanning-tree vlan** 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1) วิธีการคอนฟิกค่า Bridge Priority

โดยหลักการคอนฟิกค่า Bridge Priority ของสวิตช์ที่ต้องการเป็น Root Bridge จะต้องให้มีค่าต่ำกว่าค่า Bridge Priority ของสวิตช์ตัวอื่นๆ รวมถึงต้องต่ำกว่าค่าดีฟอลต์ของสวิตช์ตัวที่ไม่ได้คอนฟิกไว้ด้วย โดยการใช้คำสั่งดังนี้

Switch(config)#**spanning-tree vlan** <เลข VLAN>**priority**<ค่า Bridge Priority>

ดังนั้นสวิตช์ที่ถูกกำหนดให้เป็น Root Bridge ในระบบใหม่ก็คือสวิตช์ CORE จึงมีการกำหนดให้มีค่า Bridge Priority น้อยที่สุด คำสั่งที่ใช้กับสวิตช์แต่ละตัวมีดังนี้

CORE(config)# **spanning-tree vlan** 1 **priority** 24576

SW1(config)# **spanning-tree vlan** 1 **priority** 28672

SWB(config)# **spanning-tree vlan** 1 **priority** 36864

SWC(config)# **spanning-tree vlan** 1 **priority** 40960

SW2(config)# **spanning-tree vlan** 1 **priority** 32768

4.2) วิธีการคอนฟิก Path Cost สำหรับการเลือก Root Port และ Designated Port

การคอนฟิกค่า Path Cost บนแต่ละพอร์ตสามารถกำหนดได้เอง ซึ่งมีผลต่อการคำนวณ Root Path Cost ของพอร์ตนั้นๆ ในการคอนฟิกจำเป็นต้องเข้าสู่โหมด Interface Config ก่อนและใช้คำสั่งดังนี้

Switch(config-if)# **spanning-tree vlan** <หมายเลข VLAN>**cost**<ค่า Path Cost>

ดังนั้นการคอนฟิก Path Cost ในแต่ละสวิตช์ของระบบเครือข่ายใหม่มีดังนี้

- CORE : คำสั่งที่ใช้ได้แก่

CORE(config)# **interface gigabitethernet** 0/1

CORE(config-if)# **spanning-tree vlan** 1 **cost** 1

CORE(config)# **interface gigabitethernet** 0/23

CORE(config-if)# **spanning-tree vlan** 1 cost 1

CORE(config)# **interface gigabitethernet** 0/34

CORE(config-if)# **spanning-tree vlan** 1 **cost** 1

CORE(config)# **interface gigabitethernet** 0/40

CORE(config-if)# **spanning-tree vlan** 1 **cost** 1

CORE(config)# **interface gigabitethernet** 0/41

CORE(config-if)# **spanning-tree vlan** 1 **cost** 1

CORE(config)# **interface gigabitethernet** 0/42

CORE(config-if)# **spanning-tree vlan** 1 **cost** 1

CORE(config)# **interface gigabitethernet** 0/46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CORE(config-if)# spanning-tree vlan 1 cost 1
CORE(config)# interface gigabitethernet 0/47
CORE(config-if)# spanning-tree vlan 1 cost 1
CORE(config)# interface gigabitethernet 0/48
CORE(config-if)# spanning-tree vlan 1 cost 1
```

- SW1 : คำสั่งที่ใช้ได้แก่

```
SW1(config)# interface gigabitethernet 1/0/1
SW1(config-if)# spanning-tree vlan 1 cost 1
SW1(config)# interface gigabitethernet 1/0/2
SW1(config-if)# spanning-tree vlan 1 cost 1
SW1(config)# interface gigabitethernet 1/0/18
SW1(config-if)# spanning-tree vlan 1 cost 2
SW1(config)# interface gigabitethernet 1/0/20
SW1(config-if)# spanning-tree vlan 1 cost 2
```

- SWB : คำสั่งที่ใช้ได้แก่

```
SWB(config)# interface gigabitethernet 0/36
SWB(config-if)# spanning-tree vlan 1 cost 2
SWB(config)# interface gigabitethernet 0/38
SWB(config-if)# spanning-tree vlan 1 cost 2
SWB(config)# interface gigabitethernet 0/42
SWB(config-if)# spanning-tree vlan 1 cost 2
SWB(config)# interface gigabitethernet 0/47
SWB(config-if)# spanning-tree vlan 1 cost 1
```

- SWC : คำสั่งที่ใช้ได้แก่

```
SWC(config)# interface gigabitethernet 0/10
SWC(config-if)# spanning-tree vlan 1 cost 2
SWC(config)# interface gigabitethernet 0/25
SWC(config-if)# spanning-tree vlan 1 cost 2
SWC(config)# interface gigabitethernet 0/43
SWC(config-if)# spanning-tree vlan 1 cost 2
SWC(config)# interface gigabitethernet 0/47
SWC(config-if)# spanning-tree vlan 1 cost 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SW2 : คำสั่งที่ใช้ได้แก่

SW2(config)# **interface gigabitethernet** 1/0/21

SW2(config-if)# **spanning-tree vlan** 1 **cost** 1

5) วิธีการคอนฟิก Port ID

Port ID เป็นพารามิเตอร์สุดท้ายที่มีผลต่อการเลือก Root Port และ Designate Port หลังจากที่ผ่านการเปรียบเทียบ Root Path Cost และ Bridge ID ค่า Port ID ประกอบไปด้วย 2 ฟิลด์มารวมกันได้แก่ Port Piority และ Port number ซึ่งสวิตช์ตัวที่มีค่า Port ID ต่ำกว่าจะถูกเลือกเป็น Root Port คำสั่งที่ใช้ในการคอนฟิก Port ID จึงจำเป็นต้องเข้าสู่โหมด Interface ก่อน และใช้คำสั่งดังนี้

Switch(config-if)#**spanning-tree vlan** <หมายเลข VLAN> **port-priority** <ค่า Port Priority>

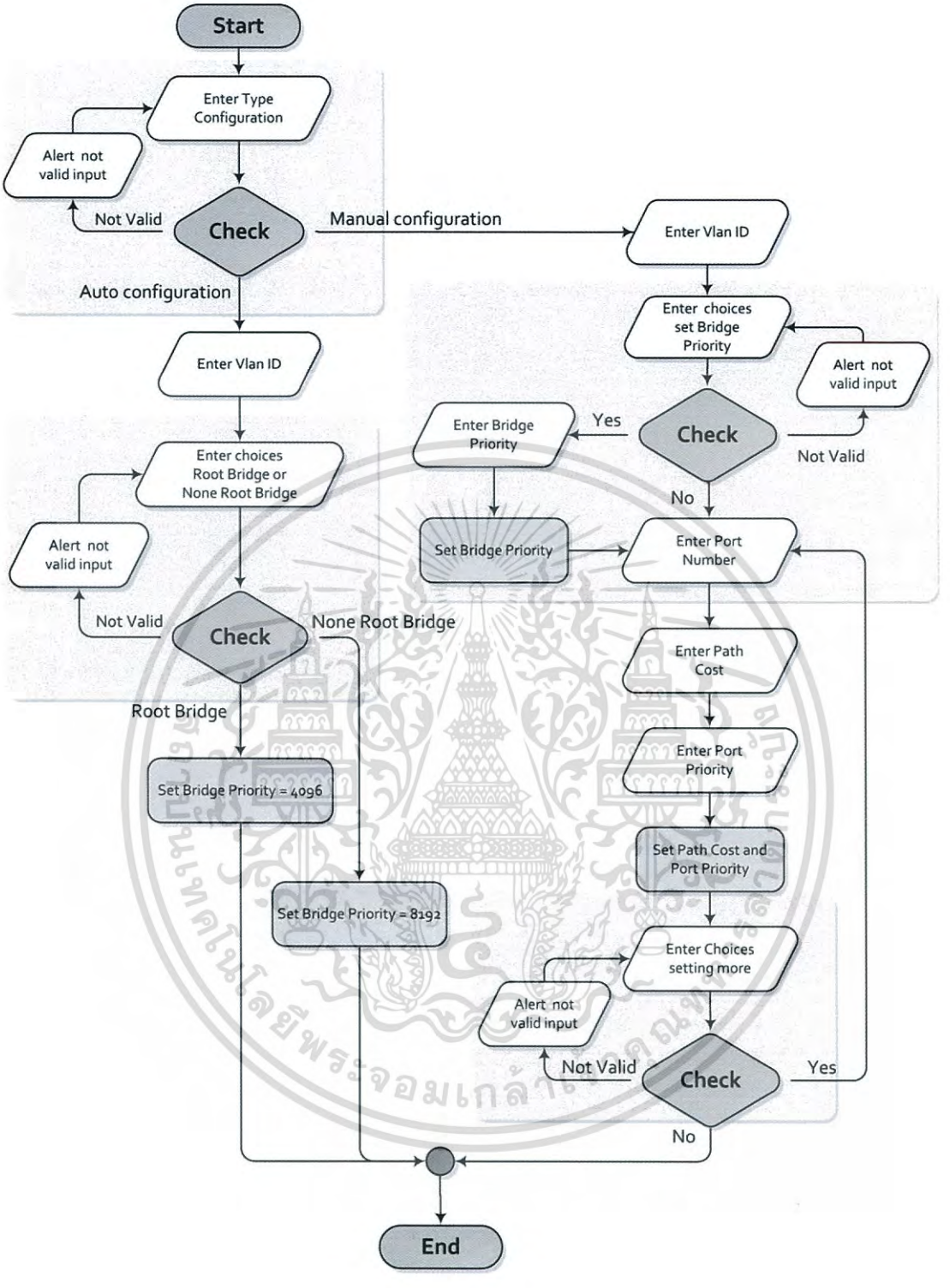
ในระบบนี้จะไม่มีการคอนฟิกค่า Port Priority ให้กับพอร์ตใดๆ ซึ่งแต่ละพอร์ตจะมีค่าเป็นค่าดีฟอลต์ หรือเท่ากับ 128

3.4 ขั้นตอนการออกแบบสคริปต์ช่วยในการพัฒนาระบบ

เนื่องจากการคอนฟิกแบบปกติจำเป็นต้องใส่คำสั่งเข้าทีละโหมด และกำหนดทีละพอร์ต รวมถึงถ้าผู้ใช้งานไม่มีความรู้พื้นฐานเกี่ยวกับคำสั่งในการคอนฟิก จำเป็นต้องใช้เวลาในการศึกษาเรียนรู้ค่อนข้างมาก ผู้จัดทำจึงได้นำ TCL เข้ามาใช้ในการพัฒนาสคริปต์เพื่อช่วยในการคอนฟิก เนื่องจากเป็นภาษาที่สามารถใช้งานได้บนอุปกรณ์สวิตช์ ซึ่งจำเป็นต้องคัดลอกไฟล์สคริปต์เข้าไปไว้ในหน่วยความจำแฟลชของอุปกรณ์สวิตช์ก่อน โดยการใช้โปรแกรม Tftpd32

การเรียกไฟล์สคริปต์จากหน่วยความจำแฟลชขึ้นมาทำงาน สามารถเข้าใช้งานได้ทั้งแบบ Serail โดยการนำสายคอนโซลต่อโดยตรง หรือทาง Telnet สามารถอธิบายการทำงานของสคริปต์ได้ดังรูปที่ 3.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 Data Flow Diagram แสดงการทำงานของสคริปต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้งานเรียกสคริปต์ขึ้นมาทำงานผ่านโปรแกรม Putty ระบบจะแสดงข้อความเพื่อให้ผู้ใช้งานเลือกโหมดการคอนฟิก ดังนี้

1) Auto Configuration
2) Manual Configuration

การรับค่าจากผู้ใช้งานระบบสามารถรับได้เพียง 2 ค่า คือ “1” หรือ “2” ในกรณีที่ผู้ใช้งานใส่ค่าที่นอกเหนือจากนี้ ระบบจะมีการแจ้งเตือนขึ้นมาว่าผู้ใช้งานใส่ค่าไม่ถูกต้องและให้ใส่ค่าใหม่อีกครั้ง

เมื่อผู้ใช้งานใส่ค่าถูกต้องแล้วระบบจะเปลี่ยนเข้าสู่โหมดนั้นๆที่ผู้ใช้เลือก ดังนี้

1) Auto Configuration

เป็นโหมดการคอนฟิคแบบอัตโนมัติ สำหรับผู้ใช้งานที่ต้องการคอนฟิกให้สวิตช์เป็น Root Bridge หรือ Non Root Bridge โดยการไม่มีการคอนฟิกค่าอื่นๆ มีการทำงานดังนี้

- ระบบแสดงข้อความเพื่อรับค่า Vlan ID จากผู้ใช้งาน เพื่อใช้ในการกำหนดขอบเขตการทำงาน
- ระบบแสดงข้อความเพื่อให้ผู้ใช้งานเลือกโหมดการคอนฟิกว่าต้องการคอนฟิกแบบใด ระหว่าง 1. Root Bridge หรือ 2. Non Root Bridge การรับค่าจากผู้ใช้งาน ระบบสามารถรับได้เพียง 2 ค่า คือ “1” หรือ “2”
- เมื่อผู้ใช้งานป้อนข้อมูลเข้ามา ระบบจะทำการตรวจสอบความถูกต้อง ในกรณีที่ผู้ใช้งานป้อนค่าที่นอกเหนือจาก 1 หรือ 2 ระบบจะแจ้งเตือนผู้ใช้งานว่าใส่ค่าไม่ถูกต้องและให้ใส่ค่าใหม่อีกครั้ง
- เมื่อผู้ใช้งานป้อนค่าที่ถูกต้องแล้ว ระบบจะทำการคอนฟิกค่า Bridge Priority ให้กับสวิตช์ ดังนี้

  1. Root Bridge

  ระบบจะคอนฟิกค่า Bridge Priority ให้กับสวิตช์เท่ากับ 4096 ซึ่งเป็นค่าที่ต่ำที่สุดที่สามารถคอนฟิกได้ เพื่อเป็นการรองรับในกรณีที่มีการนำสวิตช์อื่นมาเชื่อมต่อ ซึ่งจะไม่มีสวิตช์ตัวใดมีค่า Bridge Priority ที่ต่ำกว่าค่านี้แล้ว

  2. Non Root Bridge

  ระบบจะคอนฟิกค่า Bridge Priority ให้กับสวิตช์เท่ากับ 8192 ซึ่งเป็นค่าที่สูงกว่าค่าที่ระบบคอนฟิกให้กับ Root Bridge

- เมื่อระบบคอนฟิกค่า Bridge Priority ให้กับสวิตช์เสร็จเรียบร้อย ระบบจะออกจากโหมด Tcl เนื่องจากผู้ใช้งานเลือกโหมดการคอนฟิกแบบอัตโนมัติแล้ว ซึ่งไม่จำเป็นต้องคอนฟิกค่าใดเพิ่มอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Manual Configuration

เป็นโหมดการคอนฟิกแบบกำหนดค่าเอง มีการทำงานดังนี้

- ระบบแสดงข้อความเพื่อรับค่า Vlan ID จากผู้ใช้งาน
- ระบบแสดงข้อความถามผู้ใช้งานในการคอนฟิกค่า Bridge Priority ถ้าผู้ใช้งานต้องการคอนฟิกให้ป้อน “yes”หรือ“y” ถ้าหากไม่ต้องการให้ป้อน “no” หรือ “n”
- เมื่อผู้ใช้งานป้อนข้อมูลเข้ามา ระบบจะทำการตรวจสอบความถูกต้อง ในกรณีที่ผู้ใช้งานป้อนค่าที่นอกเหนือจาก “yes” ,” y” , “no” หรือ “n” ระบบจะแจ้งเตือนผู้ใช้งานว่าใส่ค่าไม่ถูกต้องและให้ใส่ค่าใหม่อีกครั้ง
- เมื่อผู้ใช้งานป้อนค่าที่ถูกต้องแล้ว ระบบจะตรวจสอบว่าค่าที่ผู้ใช้งานป้อนเข้ามาเป็นค่าใด ถ้าเป็น “yes” หรือ “y” ระบบจะเข้าสู่โหมดการคอนฟิกค่า Bridge Priority แต่ถ้าค่าที่ป้อนเข้ามาเป็น “no” หรือ “n” ระบบจะเข้าสู่โหมดการคอนฟิกพื้นฐาน การคอนฟิกในโหมด Manual Configuration สามารถแบ่งได้เป็น 2 ขั้นตอนดังนี้

1. การคอนฟิกค่า Bridge Priority
    - ระบบจะแสดงข้อความเพื่อรับค่า Bridge Priority จากผู้ใช้งาน
    - เมื่อผู้ใช้งานป้อนค่า Bridge Priority เข้ามาแล้ว ระบบจะคอนฟิกค่าดังกล่าวให้กับสวิตช์
    - เมื่อระบบคอนฟิกค่า Bridge Priority เสร็จเรียบร้อย ระบบจะเปลี่ยนไปสู่โหมดการคอนฟิกพื้นฐาน
2. การคอนฟิกพื้นฐาน
    - ระบบแสดงรายละเอียดและรูปแบบในการใส่ค่าให้กับผู้ใช้งานโดยมีรูปแบบดังนี้ PortNumber-PathCost-PortPriority
    - Port Number ผู้ใช้งานจำเป็นต้องป้อนชนิดของพอร์ตร่วมกับเลขพอร์ตด้วย เช่น Gigabitethernet 0/1, Fastethernet 0/2
    - Path Cost ผู้ใช้งานสามารถเลือกที่จะไม่ใส่ค่า Path Cost ได้ โดยการใส่เครื่องหมาย “x”
    - Port Priority ผู้ใช้งานจำเป็นต้องใส่ค่าให้ถูกต้องตามรูปแบบ โดยการใส่ค่าเริ่มจาก 0 และเพิ่มขึ้นทีละ 16 ผู้ใช้งานสามารถเลือกที่จะไม่ใส่ค่า Path Cost ได้ โดยการใส่เครื่องหมาย “x”
    - ระบบจะวนรับค่าจากผู้ใช้งานจนกว่าจะมีการป้อนค่า “exit” หรือ “ex” เข้ามา หลังจากนั้นระบบจะคอนฟิกค่า STP ให้กับสวิตช์และจะออกจากโหมด tcl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ขั้นตอนการออกแบบโปรแกรมช่วยในการพัฒนาระบบ

ในระบบเครือข่ายที่มีขนาดใหญ่อาจมีการใช้อุปกรณ์สวิตช์เป็นจำนวนมาก จึงส่งผลให้ต้องใช้เวลามากในการคอนฟิกแต่ละตัวอุปกรณ์ ผู้จัดทำจึงได้นำภาษา Java มาช่วยในการพัฒนาโปรแกรมเพื่อช่วยในการพัฒนาระบบ โดยผู้ใช้งานสามารถเก็บข้อมูลค่าที่ใช้ในการคอนฟิก STP ของแต่ละสวิตช์รวมไว้ในตารางเป็นไฟล์ .csv ซึ่งโปรแกรมจะนำข้อมูลจากไฟล์ .csv ไปประมวลผลโดยการนำค่าจากไฟล์ .csv มาแปลงให้อยู่ในรูปแบบของคำสั่ง Tcl เพื่อนำไปเรียกใช้ได้ทันทีบนสวิตช์ รูปแบบของตารางเพื่อเก็บค่าแสดงได้ดังตารางที่ 3.13

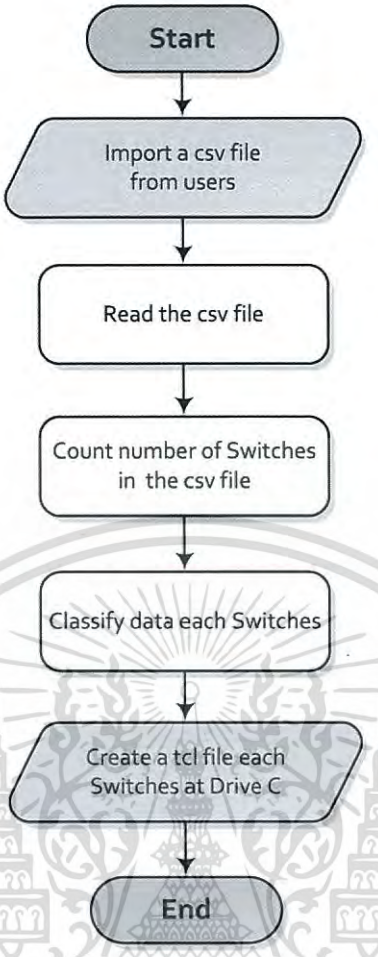
ตารางที่ 3.13 แสดงรูปแบบตารางเก็บค่าที่ใช้คำนวณ STP

Switch	Vlan	BridgePriority	PortNumber	PathCost	PortPriority

- คอลัมน์ที่ 1 เก็บชื่อของสวิตช์ที่ต้องการคอนฟิก
- คอลัมน์ที่ 2 เก็บค่า Vlan ID
- คอลัมน์ที่ 3 เก็บค่า Bridge Priority ของสวิตช์
- คอลัมน์ที่ 4 เก็บค่า Port Number
- คอลัมน์ที่ 5 เก็บค่า Path Cost
- คอลัมน์ที่ 6 เก็บค่า Port Priority

ขั้นตอนการทำงานของโปรแกรมแสดงได้ดังรูปที่ 3.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 27 Data Flow Diagram แสดงการทำงานของโปรแกรม

1) ผู้ใช้งานป้อนไฟล์ .csv ให้แก่ระบบ
2) ระบบจะทำการอ่านไฟล์ .csv ที่ผู้ใช้งานป้อนเข้ามา
3) ระบบจะตรวจสอบจำนวนอุปกรณ์สวิตช์ทั้งหมดที่ผู้ใช้งานต้องการคอนฟิกจากชื่อของสวิตช์ในคอลัมน์ที่ 1
4) ระบบจะแยกข้อมูลของคอลัมน์ที่ 2-6 ให้กับแต่ละสวิตช์
5) ระบบสร้างไฟล์ใหม่เป็น .tcl เก็บคำสั่งที่ใช้สำหรับคอนฟิกสวิตช์เป็นภาษา Tcl ซึ่งนำค่าจากการแยกข้อมูลในขั้นตอนที่ 4 มาใช้ โดยไฟล์ที่ได้จะมีจำนวนเท่ากับจำนวนของสวิตช์ที่ต้องการคอนฟิก โดยชื่อไฟล์จะเป็นชื่อของสวิตช์นั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการวิจัยและการอภิปรายผล

4.1 ผลการคอนฟิก

4.1.1 การแสดงผล Etherchannel

ในการแสดงรายละเอียดของ Etherchannel ที่ได้คอนฟิกไว้ สามารถแสดงได้จากคำสั่ง **show etherchannel summary** ซึ่งในระบบใหม่นี้มีการทำ EtherChannel ไว้ที่สวิตช์ CORE และ SW1

- CORE : ผลที่ได้เป็นดังรูปที่ 4.1

```
CORE#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

2      Po2(SU)           -      Gig0/1(P) Gig0/2(P)
```

รูปที่ 4.1 แสดงผล Etherchannel ของสวิตช์ CORE

จากผลที่แสดงของสวิตช์ CORE แสดงให้เห็นว่ามีการสร้าง Channel Group ขึ้นมา 1 Group คือ Group number 2 ที่พอร์ต Gi0/1 และ Gi0/2 ชื่อของ Port Channel คือ Po2

- SW1 : ผลที่ได้เป็นดังรูปที่ 4.2

```
SW1#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

2      Po2(SU)           -      Gig1/0/1(P) Gig1/0/2(P)
```

รูปที่ 4.2 แสดงผล Etherchannel ของสวิตช์ SW1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.2 แสดงให้เห็นว่าที่สวิตช์ CORE มีการสร้าง Channel Group ขึ้นมา 1 Group คือ Group number 2 ที่พอร์ต Gi1/0/1 และ Gi1/0/2 ชื่อของ Port Channel คือ Po2

4.1.2 การแสดงผล Trunk

ในการแสดงรายละเอียดของ Trunk ที่ได้คอนฟิกไว้ สามารถแสดงได้จากคำสั่ง **show interfaces trunk** ซึ่งในระบบใหม่นี้มีการทำ Trunk ไว้ที่ Channel Group 2 ของสวิตช์ CORE และ SW1

- **CORE : ผลที่ได้เป็นดังรูปที่ 4.3**

```
CORE#show interfaces trunk
Port        Mode         Encapsulation  Status        Native vlan
Po2         on           802.1q         trunking      1

Port        Vlans allowed on trunk
Po2         1-1005

Port        Vlans allowed and active in management domain
Po2         1

Port        Vlans in spanning tree forwarding state and not pruned
Po2         1
```

รูปที่ 4.3 แสดงผล Trunk ของสวิตช์ CORE

จากรูปที่ 4.3 แสดงให้เห็นว่าที่สวิตช์ CORE มีการสร้าง Trunk ขึ้นมาที่ Port Channael Po2 โดย Encapsulation เท่า 802.1q ซึ่งเป็นมาตรฐานการทำ Trunk ในการส่งข้อมูลหลายๆ Network หรือ Vlan

- **SW1 : ผลที่ได้เป็นดังรูปที่ 4.4**

```
SW1#show interfaces trunk
Port        Mode         Encapsulation  Status        Native vlan
Po2         on           802.1q         trunking      1

Port        Vlans allowed on trunk
Po2         1-1005

Port        Vlans allowed and active in management domain
Po2         1

Port        Vlans in spanning tree forwarding state and not pruned
Po2         1
```

รูปที่ 4.4 แสดงผล Trunk ของสวิตช์ SW1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การแสดงผล Vlan

ในการแสดงรายละเอียดของ Vlan สามารถแสดงได้จากคำสั่ง **show vlan**

- CORE : **ผลที่ได้เป็นดังรูปที่ 4.5**

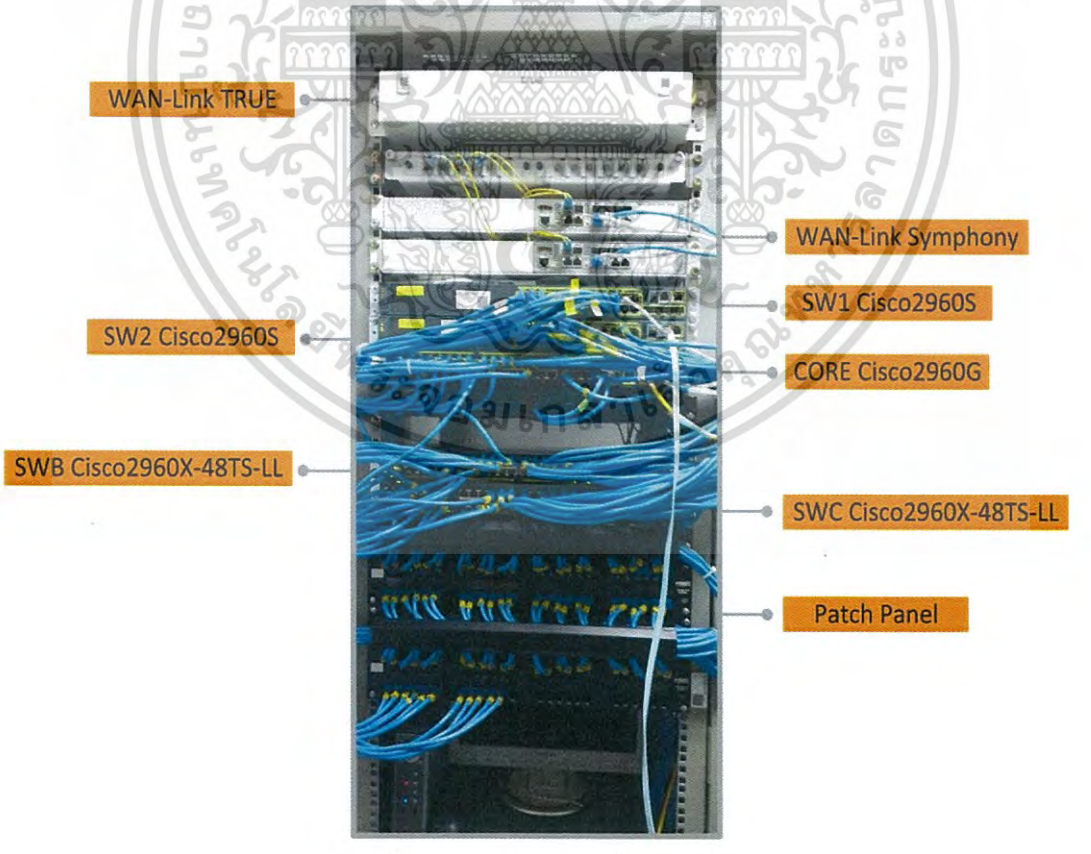
```
CORE#show vlan

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Gig0/1,Gig0/2,Gig0/3,Gig0/4
                                                 Gig0/5,Gig0/6,Gig0/7,Gig0/8
                                                 Gig0/9,Gig0/10,Gig0/11,Gig0/12
                                                 Gig0/13,Gig0/14,Gig0/15,Gig0/16
                                                 Gig0/17,Gig0/18,Gig0/19,Gig0/20
                                                 Gig0/21,Gig0/22,Gig0/23,Gig0/24
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
```

รูปที่ 4.5 แสดงผล Vlan ของสวิตช์ CORE

จากรูปที่ 4.5 แสดงให้เห็นว่าพอร์ตทุกพอร์ตบนสวิตช์ CORE เป็น Vlan 1 ซึ่งเป็นค่าดีฟอลต์

4.2 ผลการติดตั้งอุปกรณ์



รูปที่ 4.6 แสดงการติดตั้งอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบการเกิดลูปบนระบบเครือข่ายใหม่

ในขั้นตอนการทดสอบการเกิดลูปหลังจากการออกแบบและติดตั้งระบบเครือข่ายใหม่ จะทดสอบโดยการต่อสวิตช์แต่ละตัวเข้าหากันทางพอร์ตที่ไม่มีการใช้งาน ซึ่งจะทำการการทดสอบระหว่าง

- สวิตช์ SW1 เชื่อมต่อกับสวิตช์ SWB
- สวิตช์ SW1 เชื่อมต่อกับสวิตช์ SWC
- สวิตช์ SW1 เชื่อมต่อกับสวิตช์ SW2
- สวิตช์ SWB เชื่อมต่อกับสวิตช์ SWC
- สวิตช์ SWB เชื่อมต่อกับสวิตช์ SW2
- สวิตช์ SWC เชื่อมต่อกับสวิตช์ SW2
- สวิตช์ SWB เชื่อมต่อกับสวิตช์ CORE

คำสั่งในการแสดงผล STP ในโปรแกรม Putty สามารถตรวจสอบได้จากคำสั่ง **show spanning tree ?** ดังรูปที่ 4.7

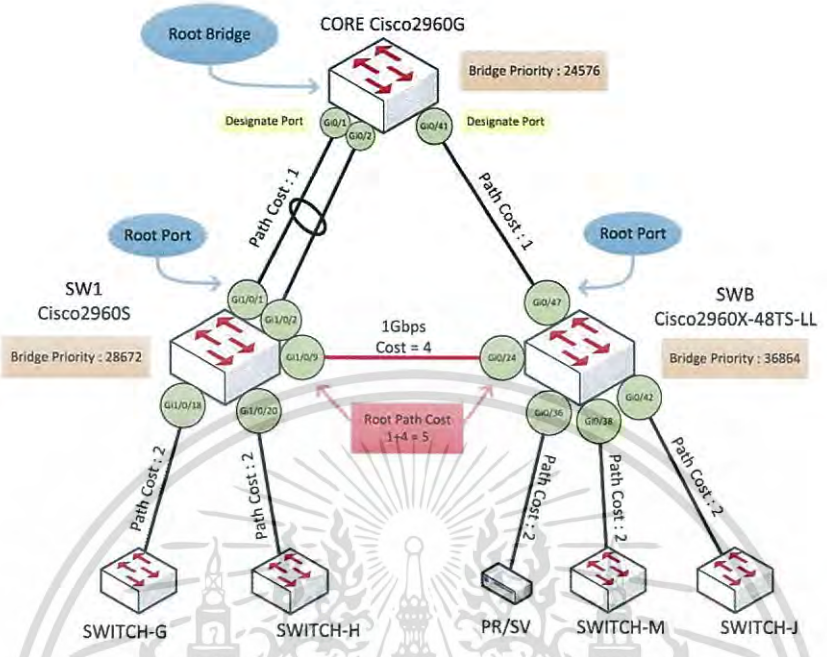
```
SWITCH#sh spanning-tree ?
  active          Report on active interfaces only
  backbonefast    Show spanning tree backbonefast status
  blockedports    Show blocked ports
  bridge          Status and configuration of this bridge
  detail          Detailed information
  inconsistentports Show inconsistent ports
  interface       Spanning Tree interface status and configuration
  mst             Multiple spanning trees
  pathcost        Show Spanning pathcost options
  root            Status and configuration of the root bridge
  summary         Summary of port states
  uplinkfast      Show spanning tree uplinkfast status
  vlan            VLAN Switch Spanning Trees
  |               Output modifiers
  <cr>
```

รูปที่ 4.7 แสดงคำสั่งที่ใช้ใน STP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 สวิตช์ SW1 เชื่อมต่อกับสวิตช์ SWB

พอร์ต Gi1/0/9 ของสวิตช์ SW1 เชื่อมต่อกับพอร์ต Gi0/24 ของสวิตช์ SWB



รูปที่ 4.8 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SW1 ต่อกับ SWB

พอร์ต Gi0/1 และ Gi0/2 ของสวิตช์ CORE เป็นพอร์ตที่มีการทำ Link Aggregation จะถูกมองเป็นพอร์ตเดียวกัน ซึ่งทั้งสองพอร์ตนี้จึงมีสถานะเหมือนกันเสมอ

พอร์ต Gi1/0/9 ของสวิตช์ SW1 และพอร์ต Gi0/24 ของสวิตช์ SWB มีค่า Path Cost เท่ากับ 4 เนื่องจากไม่ได้มีการคอนฟิก Path Cost ไว้ จึงขึ้นกับค่าแบนด์วิธของพอร์ตนั้นๆคือ 1 Gbps

- **การหา Root Bridge**

  อุปกรณ์ CORE มีค่า Bridge Priority สูงที่สุดจึงทำให้ได้เป็น Root Bridge โดยไม่มีการนำค่า MAC address มาพิจารณาต่อ และพอร์ตทุกพอร์ตของสวิตช์ CORE จึงกลายเป็น Designated Port โดยอัตโนมัติ

- **การหา Root Port จะพิจารณาที่ค่า Root Path Cost เป็นอันดับแรก**

  SW1 มีค่า Root Path Cost ประจำพอร์ตดังนี้

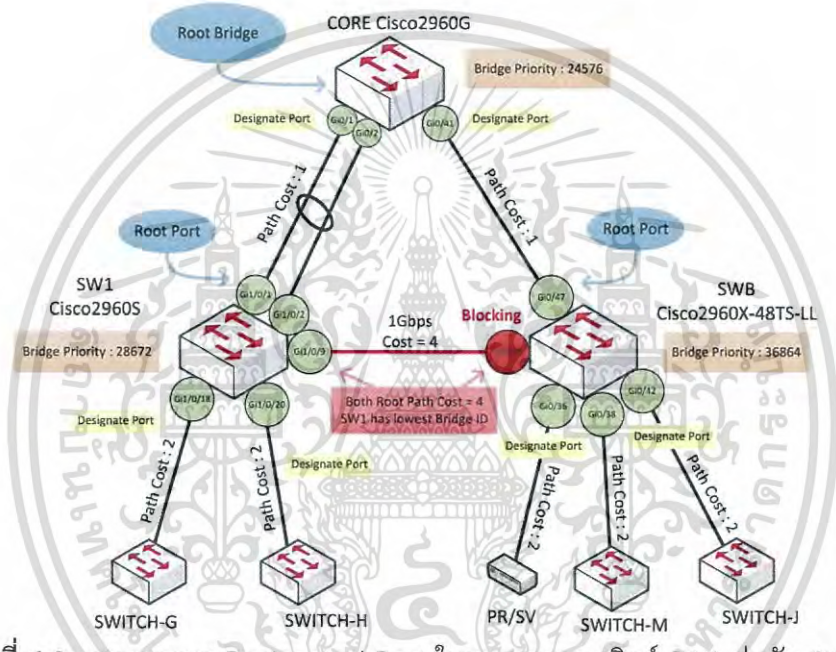
- พอร์ต Gi1/0/1 และ Gi1/0/2 ซึ่งทำเป็น Link Aggregation มีค่า Root Path Cost เท่ากับ 1 พอร์ตนี้จึงได้เป็น Root Port
- พอร์ต Gi1/0/9 ที่เชื่อมต่อเข้ากับสวิตช์ SW2 มีค่า Root Path Cost เท่ากับ 1+4 = 5
- พอร์ต Gi1/0/18 ที่เชื่อมต่อเข้ากับสวิตช์ SWITCH-G และ พอร์ต Gi1/0/20 ที่เชื่อมต่อเข้ากับ SWITCH-H มีค่า Root Path Cost ได้เท่ากับ 1+2 = 3 หรือ 1+1 = 2 นับจาก Root

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bridge ซึ่งไม่ว่าจะมาจากทางไหน STP จะไม่สนใจว่าตัวไหนมีค่า Root Path Cost มากที่สุด แต่จะสนใจที่ค่า Root Path Cost ที่น้อยที่สุด ซึ่งถึงจะมีค่ามากเพียงไหนก็ไม่มีผลในการเลือก Root Port

SWB มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/47 ที่เชื่อมต่อเข้ากับสวิตช์ CORE มีค่า Root Path Cost เท่ากับ 1 ซึ่งเป็นค่าที่ได้คอนฟิกไว้ พอร์ตนี้จึงได้เป็น Root Port
- พอร์ต Gi0/24 ที่เชื่อมต่อเข้ากับสวิตช์ SW1 มีค่า Root Path Cost เท่ากับ 1+4 = 5 นับจาก Root Bridge



รูปที่ 4.9 แสดงการหา Designated Port ในการทดสอบสวิตช์ SW1 ต่อกับ SWB

- **การหา Designated Port จะพิจารณาพอร์ตที่เหลือทีมีค่า Path Cost ที่น้อยที่สุด**

SW1 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/9 ที่เชื่อมต่อเข้ากับสวิตช์ SWB มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi1/0/18 ที่เชื่อมต่อเข้ากับสวิตช์ SWITCH-G และ พอร์ต Gi1/0/20 ที่ต่อเข้ากับ SWITCH-H มีค่า Root Path Cost เท่ากับ 2 จึงได้เป็น Designated Port

SWB มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/24 มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi1/0/9 ของสวิตช์ SW1 และ พอร์ต Gi0/24 ของสวิตช์ SW2 มีค่า Root Path Cost เท่ากัน จึงพิจารณาที่ค่า Bridge ID แทน ซึ่งค่า Bridge Priority ของสวิตช์ SW1 มีค่าน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กว่าของสวิตช์ SWB พอร์ต Gi1/0/9 ของสวิตช์ SW1 จึงได้เป็น Designated Port และพอร์ต Gi0/24 ของสวิตช์ SWB จึงกลายเป็น Block Port

คำสั่งที่ใช้ในการแสดงรายละเอียดของสถานะและค่าที่ใช้ในการคำนวณของแต่ละพอร์ตที่ใช้งานอยู่ ผ่านโปรแกรม Putty จะใช้คำสั่ง **show spanning-tree active**

- **CORE : ผลที่ได้เป็นดังรูปที่ 4.10**

```
CORE#sh spanning-tree active

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    24577  (priority 24576 sys-id-ext 1) <<<< Bridge Priority ของ CORE
              Address     001f.6d5c.3c00  <<<< Mac Address ของ CORE
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 15

Interface        Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน CORE
---------------- ---- --- --------- -------- --------------------------------
Gi0/23           Desg FWD 1         128.23   P2p   <<<< Connect to SWITCH-D
Gi0/34           Desg FWD 1         128.34   P2p   <<<< Connect to SW2
Gi0/40           Desg FWD 1         128.40   P2p   <<<< Connect to SWITCH-I
Gi0/41           Desg FWD 1         128.41   P2p   <<<< Connect to SWB
Gi0/42           Desg FWD 1         128.42   P2p   <<<< Connect to SWC
Gi0/46           Desg FWD 1         128.46   P2p   <<<< Connect to Firewall
Gi0/47           Desg FWD 1         128.47   P2p   <<<< Connect to WAN-Link Symphony
Gi0/48           Desg FWD 1         128.48   P2p   <<<< Connect to WAN-Link True
Po2              Desg FWD 1         128.64   P2p   <<<< Port Channel Connect to SW1
```

รูปที่ 4.10 แสดงผล STP บนสวิตช์ CORE ในการทดสอบสวิตช์ SW1 ต่อกับ SWB

ในส่วนของ Root ID ที่แสดงเป็นค่าของตัวสวิตช์ CORE เอง เพราะสวิตช์ CORE ได้เป็น Root Bridge ส่วนที่เป็นรายละเอียด Interface หรือพอร์ตแสดงให้เห็นว่าทุกพอร์ตบนสวิตช์ CORE เป็น Designated Port และมีสถานะเป็น Forwarding ทุกพอร์ต ค่า Cost ของแต่ละพอร์ตมีค่าเท่ากับ 1 ส่วนค่า Priority ของพอร์ตคือ 128 ซึ่งเป็นค่าดีฟอลต์เท่ากันหมด โดยพอร์ต Channel ที่ทำ EtherChannel ระหว่าง พอร์ต Gi0/1 และ Gi0/2 จะแสดงรวมกันเป็นพอร์ตเดียวในชื่อ Po2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SW1 : ผลที่ได้เป็นดังรูปที่ 4.11

```
SW1#sh spanning-tree active

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        232 (Port-channel2)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28673  (priority 28672 sys-id-ext 1) <<<< Bridge Priority ของ SW1
              Address     f4ea.6738.6400  <<<< Mac Address ของ SW1
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน SW1
------------------- ---- --- --------- -------- --------------------------------
Gi1/0/9             Desg FWD 4         128.9    P2p   <<<< Connect to SWB
Gi1/0/18            Desg FWD 2         128.18   P2p   <<<< Connect to SWITCH-G
Gi1/0/20            Desg FWD 2         128.20   P2p   <<<< Connect to SWITCH-H
Po2                 Root FWD 1         128.232  P2p   <<<< Port Channel Connect to CORE
```

รูปที่ 4.11 แสดงผล STP บนสวิตช์ SW1 ในการทดสอบสวิตช์ SW1 ต่อกับ SWB

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Po2 หรือพอร์ต Channel ของพอร์ต Gi1/0/1 และ พอร์ต Gi1/0/2 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ตที่เหลือจะเป็น Designated Port และทุกพอร์ตจะมีสถานะเป็น Forwarding ทุกพอร์ต

- SWB : ผลที่ได้เป็นดังรูปที่ 4.12

```
SWB#sh spanning-tree active

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        47 (GigabitEthernet0/47)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    36865  (priority 36864 sys-id-ext 1) <<<< Bridge Priority ของ SWB
              Address     005f.8688.e980 <<<< Mac Address ของ SWB
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน SWB
------------------- ---- --- --------- -------- --------------------------------
Gi0/24              Altn BLK 4         128.24   P2p   <<<< Connect to SW1
Gi0/36              Desg FWD 2         128.36   P2p   <<<< Connect to PR/SV
Gi0/38              Desg FWD 2         128.38   P2p   <<<< Connect to SWITCH-M
Gi0/42              Desg FWD 2         128.42   P2p   <<<< Connect to SWITCH-J
Gi0/47              Root FWD 1         128.47   P2p   <<<< Connect to SW2
```

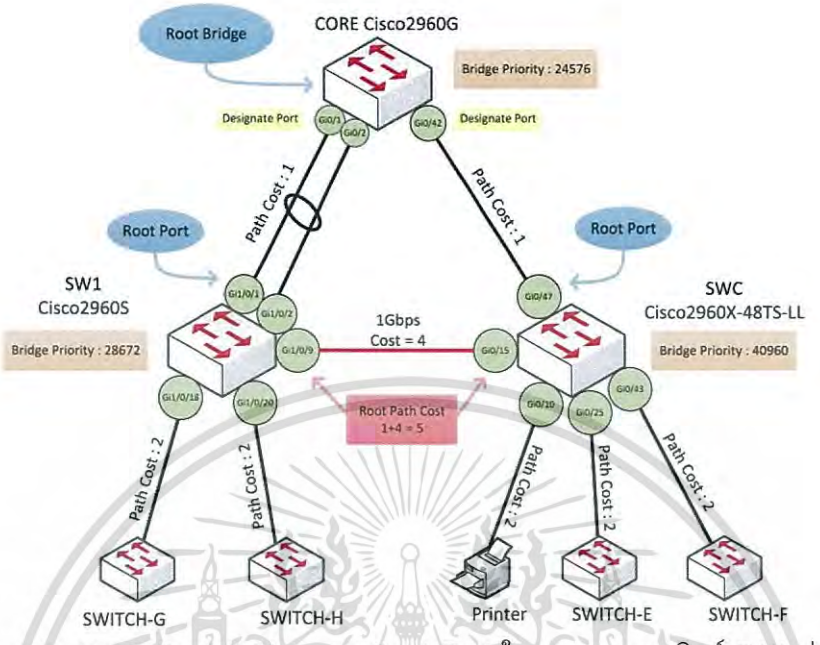
รูปที่ 4.12 แสดงผล STP บนสวิตช์ SWB ในการทดสอบสวิตช์ SW1 ต่อกับ SWB

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Gi0/47 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ต Gi0/24 ที่เชื่อมต่อกับ SW1 มีค่า Cost มากที่สุด จึงมีสถานะเป็น Blocking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 สวิตช์ SW1 เชื่อมต่อกับสวิตช์ SWC

พอร์ต Gi1/0/9 ของสวิตช์ SW1 เชื่อมต่อกับพอร์ต Gi0/15 ของสวิตช์ SWC



รูปที่ 4.13 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SW1 ต่อกับ SWC

พอร์ต Gi1/0/9 ของสวิตช์ SW1 และพอร์ต Gi0/15 ของสวิตช์ SWB มีค่า Path Cost เท่ากับ 4 เนื่องจากไม่ได้มีการคอนฟิกไว้ จึงขึ้นกับค่าแบนด์วิธของพอร์ตนั้นๆคือ 1 Gbps

- **การหา Root Bridge**

  อุปกรณ์ CORE มีค่า Bridge Priority สูงที่สุดจึงทำให้ได้เป็น Root

- **การหา Root Port** จะพิจารณาที่ค่า Root Path Cost เป็นอันดับแรก

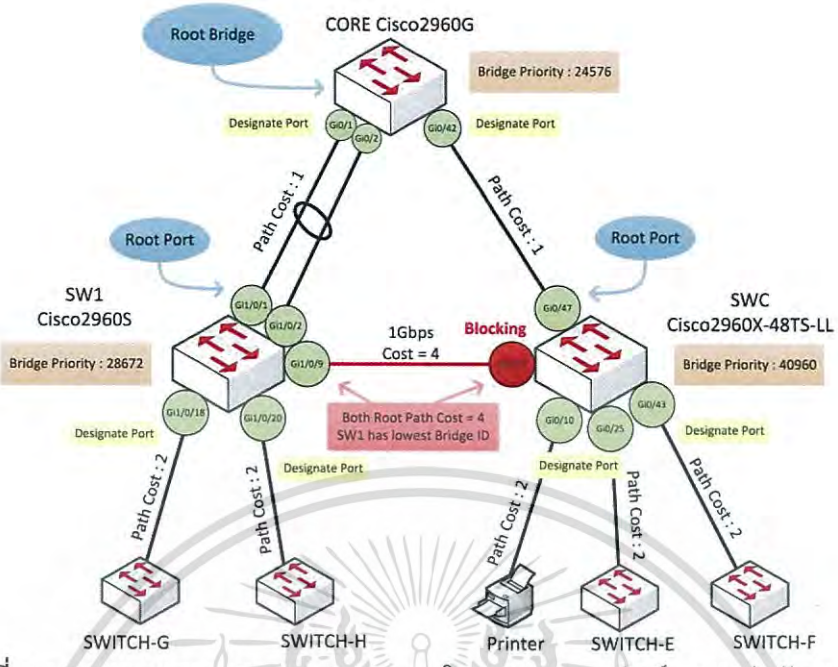
  SW1 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/1 และ Gi1/0/2 มีค่า Root Path Cost เท่ากับ 1 จึงได้เป็น Root Port
- พอร์ต Gi1/0/9 ที่เชื่อมต่อเข้ากับสวิตช์ SWC มีค่า Root Path Cost เท่ากับ 1+4 = 5
- พอร์ต Gi1/0/18 ที่เชื่อมต่อเข้ากับสวิตช์ SWITCH-G และ พอร์ต Gi1/0/20 ที่เชื่อมต่อเข้ากับ SWITCH-H มีค่า Root Path Cost เท่ากับ 1+2 = 3

  SWC มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/47 มีค่า Root Path Cost เท่ากับ 1 จึงได้เป็น Root Port
- พอร์ต Gi0/15 ที่ต่อเข้ากับสวิตช์ SW1 มีค่า Root Path Cost เท่ากับ 1+4 = 5
- พอร์ต Gi0/10 ที่เชื่อมต่อเข้ากับ Printer พอร์ต Gi0/25 ที่เชื่อมต่อเข้ากับ SWITCH-E และพอร์ต Gi0/43 ที่เชื่อมต่อเข้ากับ SWITCH-F มีค่า Root Path Cost เท่ากับ 1+2 = 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงการหา Desiganted Port ในการทดสอบสวิตช์ SW1 ต่อกับ SWC

- **การหา Designated Port** จะพิจารณาพอร์ตที่เหลือที่มีค่า Path Cost ที่น้อยที่สุด

SW1 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/9 มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi1/0/18 และ Gi1/0/20 มีค่า Root Path Cost เท่ากับ 2 จึงได้เป็น Designated Port

SWC มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/15 มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi0/10, Gi0/25 และ Gi0/43 มีค่า Root Path Cost เท่ากับ 2 จึงได้เป็น Designated Port

พอร์ต Gi1/0/9 ของสวิตช์ SW1 และ พอร์ต Gi0/15 ของสวิตช์ SWC มีค่า Root Path Cost เท่ากัน จึงพิจารณาที่ค่า Bridge ID แทน ซึ่งค่า Bridge Priority ของอุปกรณ์ SW1 มีค่าน้อยกว่าของสวิตช์ SWC พอร์ต Gi1/0/9 ของสวิตช์ SW1 จึงได้เป็น Designated Port และพอร์ต Gi0/15 ของสวิตช์ SWC จึงกลายเป็น Block Port

รายละเอียดของสถานะและค่าที่ใช้ในการคำนวณของแต่ละพอร์ตที่ใช้งานอยู่ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **SW1 : ผลที่ได้เป็นดังรูปที่ 4.15**

```
SW1#sh spanning-tree active
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        232 (Port-channel2)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28673  (priority 28672 sys-id-ext 1) <<<< Bridge Priority ของ SW1
              Address     f4ea.6738.6400  <<<< Mac Address ของ SW1
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน SW1
------------------- ---- --- --------- -------- --------------------------------
Gi1/0/9             Desg FWD 4         128.9    P2p   <<<< Connect to SWC
Gi1/0/18            Desg FWD 2         128.18   P2p   <<<< Connect to SWITCH-G
Gi1/0/20            Desg FWD 2         128.20   P2p   <<<< Connect to SWITCH-H
Po2                 Root FWD 1         128.232  P2p   <<<< Port Channel Connect to CORE
```

รูปที่ 4.15 แสดงผล STP บนสวิตช์ SW1 ในการทดสอบสวิตช์ SW1 ต่อกับสวิตช์ SWC

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Po2 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ตที่เหลือจะเป็น Designated Port และทุกพอร์ตจะมีสถานะเป็น Forwarding ทุกพอร์ต

- **SWC : ผลที่ได้เป็นดังรูปที่ 4.16**

```
SWC#sh spanning-tree active

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        47 (GigabitEthernet0/47)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    40961  (priority 40960 sys-id-ext 1)  <<<< Bridge Priority ของ SWC
              Address     005f.86aa.f000  <<<< Mac Address ของ SWC
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน SWC
------------------- ---- --- --------- -------- --------------------------------
Gi0/10              Desg FWD 2         128.10   P2p   <<<< Connect to Printer
Gi0/15              Altn BLK 4         128.15   P2p   <<<< Connect to SW1
Gi0/25              Desg FWD 2         128.25   P2p   <<<< Connect to SWITCH-E
Gi0/43              Desg FWD 2         128.43   P2p   <<<< Connect to SWITCH-F
Gi0/47              Root FWD 1         128.47   P2p   <<<< Connect to CORE
```

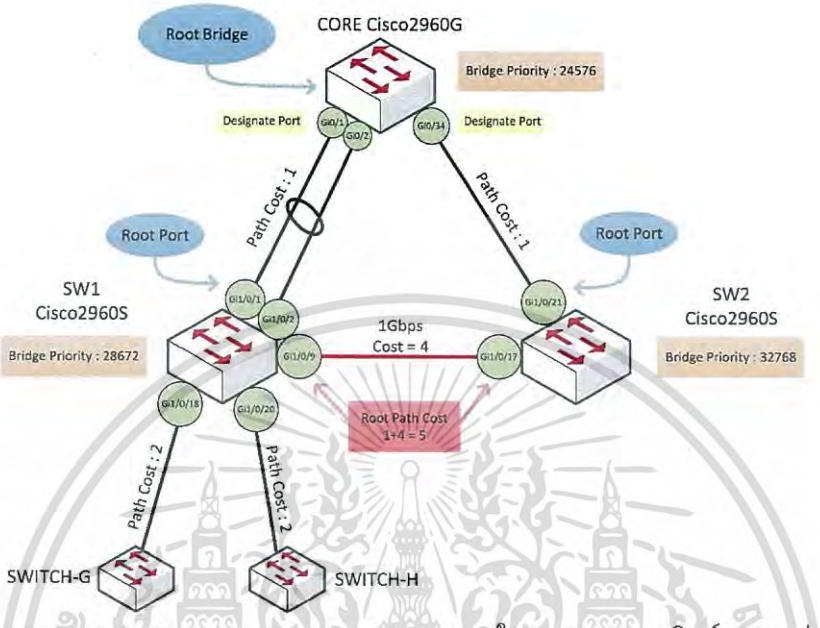
รูปที่ 4.16 แสดงผล STP บนสวิตช์ SWC ในการทดสอบสวิตช์ SW1 ต่อกับสวิตช์ SWC

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Gi0/47 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ต Gi0/15 มีค่า Cost มากที่สุด จึงมีสถานะเป็น Blocking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 สวิตช์ SW1 เชื่อมต่อกับสวิตช์ SW2

ในการต่อสวิตช์เข้าหากัน จะเชื่อมต่อกันที่พอร์ต Gi1/0/9 ของสวิตช์ SW1 และที่พอร์ต Gi0/17 ของสวิตช์ SW2



รูปที่ 4.17 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SW1 ต่อกับ SW2

พอร์ต Gi1/0/9 ของสวิตช์ SW1 และที่พอร์ต Gi0/17 ของสวิตช์ SW2 มีค่า Path Cost เท่ากับ 4 เนื่องจากไม่ได้มีการคอนฟิกไว้ จึงขึ้นกับค่าแบนด์วิธของพอร์ตนั้นๆคือ 1 Gbps

- **การหา Root Bridge**

  อุปกรณ์ CORE มีค่า Bridge Priority สูงที่สุดจึงทำให้ได้เป็น Root Bridge

- **การหา Root Port** จะพิจารณาที่ค่า Root Path Cost เป็นอันดับแรก

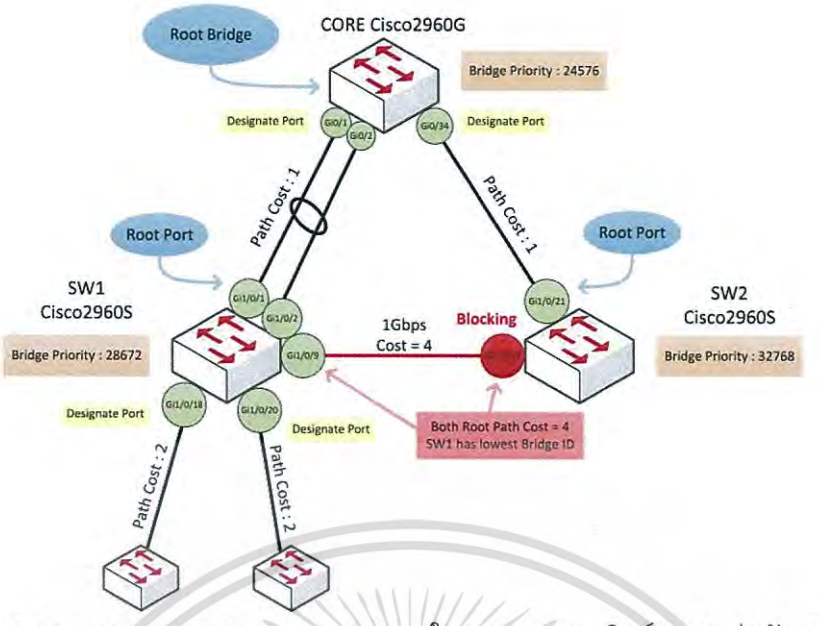
  SW1 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/1 และ Gi1/0/2 มีค่า Root Path Cost เท่ากับ 1 จึงได้เป็น Root Port
- พอร์ต Gi1/0/9 ที่เชื่อมต่อเข้ากับสวิตช์ SW2 มีค่า Root Path Cost เท่ากับ 1+4 = 5
- พอร์ต Gi1/0/18 ที่เชื่อมต่อเข้ากับ SWITCH-G และพอร์ต Gi1/0/20 ที่เชื่อมต่อเข้ากับ SWITCH-H มีค่า Root Path Cost เท่ากับ 1+2 = 3

  SW2 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/21 มีค่า Root Path Cost เท่ากับ 1 จึงได้เป็น Root Port
- พอร์ต Gi1/0/17 ที่ต่อเข้ากับสวิตช์ SW1 มีค่า Root Path Cost เท่ากับ 1+4 = 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 แสดงการหา Designated Port ในการทดสอบสวิตช์ SW1 ต่อกับ SW2

- **การหา Designated Port** จะพิจารณาพอร์ตที่เหลือที่มีค่า Path Cost ที่น้อยที่สุด

SW1 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/9 มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi1/0/18 และ Gi1/0/20 มีค่า Root Path Cost เท่ากับ 2 จึงได้เป็น Designated Port

SW2 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/17 มีค่า Root Path Cost เท่ากับ 4

พอร์ต Gi1/0/9 ของสวิตช์ SW1 และ พอร์ต Gi1/0/17 ของสวิตช์ SW2 มีค่า Root Path Cost เท่ากัน จึงพิจารณาที่ค่า Bridge ID แทน ซึ่งค่า Bridge Priority ของอุปกรณ์ SW1 มีค่าน้อยกว่าของสวิตช์ SW2 พอร์ต Gi1/0/9 ของสวิตช์ SW1 จึงได้เป็น Designated Port และ พอร์ต Gi1/0/17 ของสวิตช์ SW2 จึงกลายเป็น Block Port

รายละเอียดของสถานะและค่าที่ใช้ในการคำนวณของแต่ละพอร์ตที่ใช้งานอยู่ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SW1 : ผลที่ได้เป็นดังรูปที่ 4.19

```
SW1#sh spanning-tree active

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        232 (Port-channel2)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28673  (priority 28672 sys-id-ext 1) <<<< Bridge Priority ของ SW1
              Address     f4ea.6738.6400 <<<< Mac Address ของ SW1
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type <<<< แสดงข้อมูลของแต่ละพอร์ตบน SW1
------------------- ---- --- --------- -------- --------------------------------
Gi1/0/9             Desg FWD 4         128.9    P2p  <<<< Connect to SW2
Gi1/0/18            Desg FWD 2         128.18   P2p  <<<< Connect to SWITCH-G
Gi1/0/20            Desg FWD 2         128.20   P2p  <<<< Connect to SWITCH-H
Po2                 Root FWD 1         128.232  P2p  <<<< Port Channel Connect to CORE
```

รูปที่ 4.19 แสดงผล STP บนสวิตช์ SW1 ในการทดสอบสวิตช์ SW1 ต่อกับ SW2

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Po2 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ตที่เหลือจะเป็น Designated Port และทุกพอร์ตจะมีสถานะเป็น Forwarding

- SW2 : ผลที่ได้เป็นดังรูปที่ 4.20

```
SW2#sh spanning-tree active

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        21 (GigabitEthernet1/0/21)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1) <<<< Bridge Priority ของ SW2
              Address     f47f.3562.3a00 <<<< Mac Address ของ SW2
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type <<<< แสดงข้อมูลของแต่ละพอร์ตบน SW2
------------------- ---- --- --------- -------- --------------------------------
Gi1/0/17            Altn BLK 4         128.17   P2p  <<<< Connect to SW1
Gi1/0/21            Root FWD 1         128.21   P2p  <<<< Connect to CORE
```

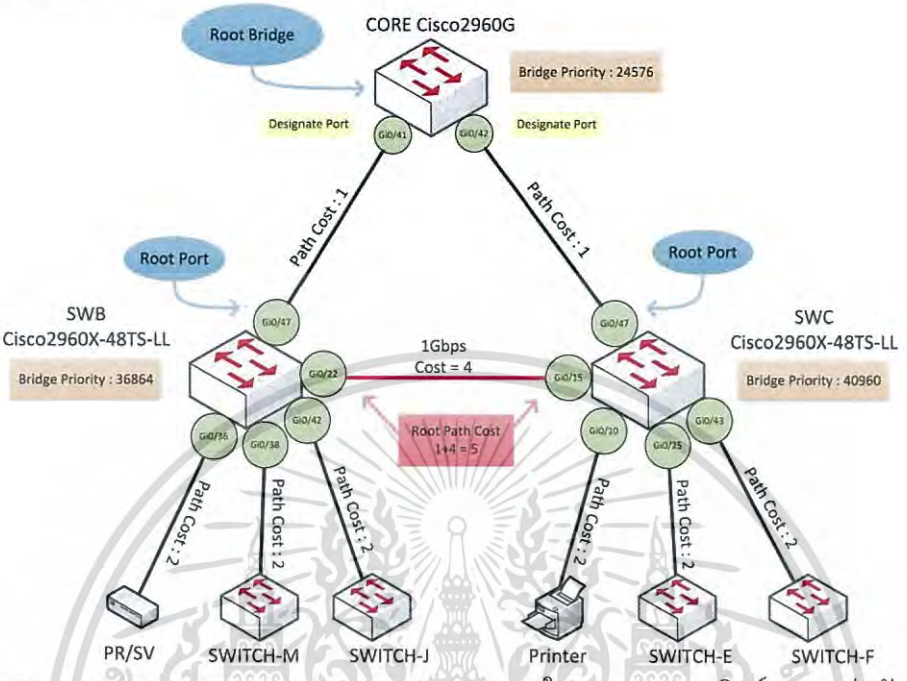
รูปที่ 4.20 แสดงผล STP บนสวิตช์ SW2 ในการทดสอบสวิตช์ SW1 ต่อกับ SW2

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Gi1/0/21 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ต Gi1/0/17 มีค่า Cost มากกว่า จึงมีสถานะเป็น Blocking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 สวิตช์ SWB เชื่อมต่อกับสวิตช์ SWC

ในการต่อสวิตช์เข้าหากัน จะเชื่อมต่อกันที่พอร์ต Gi0/22 ของสวิตช์ SWB และที่พอร์ต Gi0/15 ของสวิตช์ SWC



รูปที่ 4.21 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SWB ต่อกับ SWC

พอร์ต Gi0/22 ของสวิตช์ SWB และที่พอร์ต Gi0/15 ของสวิตช์ SWC มีค่า Path Cost เท่ากับ 4 เนื่องจากพอร์ตไม่ได้มีการคอนฟิกค่า Path Cost เอาไว้ จึงขึ้นกับค่าแบนด์วิธของพอร์ตนั้นๆคือ 1 Gbps

- **การหา Root Bridge**

  อุปกรณ์ CORE มีค่า Bridge Priority สูงที่สุดจึงทำให้ได้เป็น Root Bridge

- **การหา Root Port** จะพิจารณาที่ค่า Root Path Cost เป็นอันดับแรก

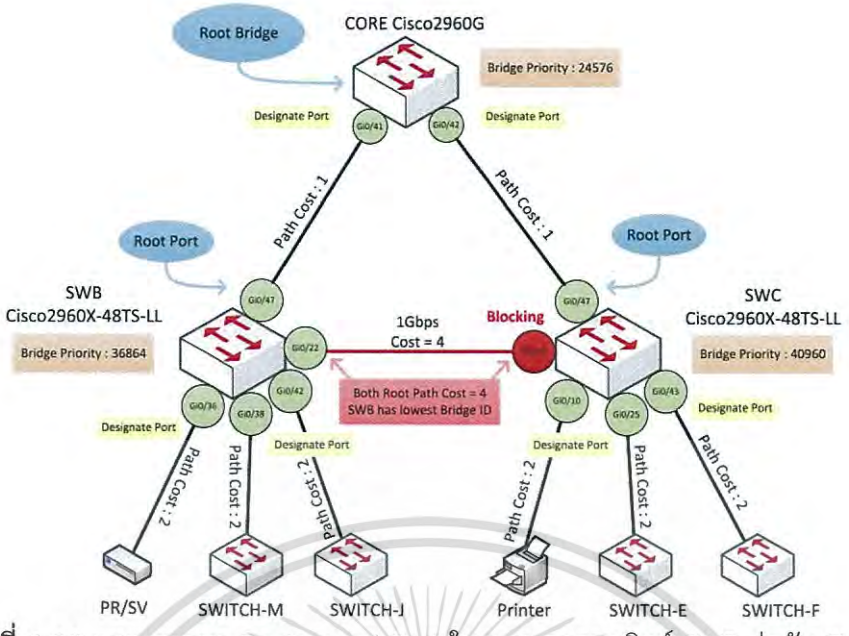
  SWB มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/47 มีค่า Root Path Cost เท่ากับ 1จึงได้เป็น Root Port
- พอร์ต Gi0/22 ที่เชื่อมต่อเข้ากับสวิตช์ SWC มีค่า Root Path Cost เท่ากับ 1+4 = 5
- พอร์ต Gi0/36 ที่เชื่อมต่อเข้ากับสวิตช์ PR/SV, พอร์ต Gi0/38 ที่เชื่อมต่อเข้ากับ SWITCH-M และพอร์ต Gi0/42 ที่เชื่อมต่อเข้ากับ SWITCH-J มีค่า Root Path Cost เท่ากับ 1+2 = 3

  SWC มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/47 มีค่า Root Path Cost เท่ากับ 1 จึงได้เป็น Root Port
- พอร์ต Gi0/15 ที่ต่อเข้ากับสวิตช์ SWB มีค่า Root Path Cost เท่ากับ 1+4 = 5
- พอร์ต Gi0/10, Gi0/25 และ Gi0/43 มีค่า Root Path Cost เท่ากับ 1+2 = 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 แสดงการหา Designated Port ในการทดสอบสวิตช์ SWB ต่อกับ SWC

- **การหา Designated Port** จะพิจารณาพอร์ตที่เหลือที่มีค่า Path Cost ที่น้อยที่สุด

SWB มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/22 มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi0/36, Gi0/38 และ Gi0/42 มีค่า Root Path Cost เท่ากับ 2

SWC มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/15 มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi0/10, Gi0/25 และ Gi0/43 มีค่า Root Path Cost เท่ากับ 2 จึงได้เป็น Designated Port

พอร์ต Gi0/22 ของสวิตช์ SWB และ พอร์ต Gi0/15 ของสวิตช์ SWC มีค่า Root Path Cost เท่ากัน จึงพิจารณาที่ค่า Bridge ID แทน ซึ่งค่า Bridge Priority ของสวิตช์ SWB มีค่าน้อยกว่าของสวิตช์ SWC พอร์ต Gi0/22 ของสวิตช์ SWB จึงได้เป็น Designated Port และพอร์ต Gi0/15 ของสวิตช์ SWC จึงกลายเป็น Block Port

รายละเอียดของสถานะและค่าที่ใช้ในการคำนวณของแต่ละพอร์ตที่ใช้งานอยู่ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SWB : ผลที่ได้เป็นดังรูปที่ 4.23

```
SWB#sh spanning-tree active

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        47 (GigabitEthernet0/47)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    36865  (priority 36864 sys-id-ext 1) <<<< Bridge Priority ของ SWB
              Address     005f.8688.e980 <<<< Mac Address ของ SWB
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type <<<< แสดงข้อมูลของแต่ละพอร์ตบน SWB
------------------- ---- --- --------- -------- --------------------------------
Gi0/22              Desg FWD 4         128.22   P2p  <<<< Connect to SWC
Gi0/36              Desg FWD 2         128.36   P2p  <<<< Connect to PR/SV
Gi0/38              Desg FWD 2         128.38   P2p  <<<< Connect to SWITCH-M
Gi0/42              Desg FWD 2         128.42   P2p  <<<< Connect to SWITCH-J
Gi0/47              Root FWD 1         128.47   P2p  <<<< Connect to CORE
```

รูปที่ 4.23 แสดงผล STP บนสวิตช์ SWB ในการทดสอบสวิตช์ SWB ต่อกับ SWC

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Gi0/47 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ตที่เหลือจะเป็น Designated Port และทุกพอร์ตจะมีสถานะเป็น Forwarding

- SWC : ผลที่ได้เป็นดังรูปที่ 4.24

```
SWC#sh spanning-tree active
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        47 (GigabitEthernet0/47)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    40961  (priority 40960 sys-id-ext 1)  <<<< Bridge Priority ของ SWC
              Address     005f.86aa.f000  <<<< Mac Address ของ SWC
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน SWC
------------------- ---- --- --------- -------- --------------------------------
Gi0/10              Desg FWD 2         128.10   P2p  <<<< Connect to Printer
Gi0/15              Altn BLK 4         128.15   P2p  <<<< Connect to SWB
Gi0/25              Desg FWD 2         128.25   P2p  <<<< Connect to SWITCH-E
Gi0/43              Desg FWD 2         128.43   P2p  <<<< Connect to SWITCH-F
Gi0/47              Root FWD 1         128.47   P2p  <<<< Connect to CORE
```

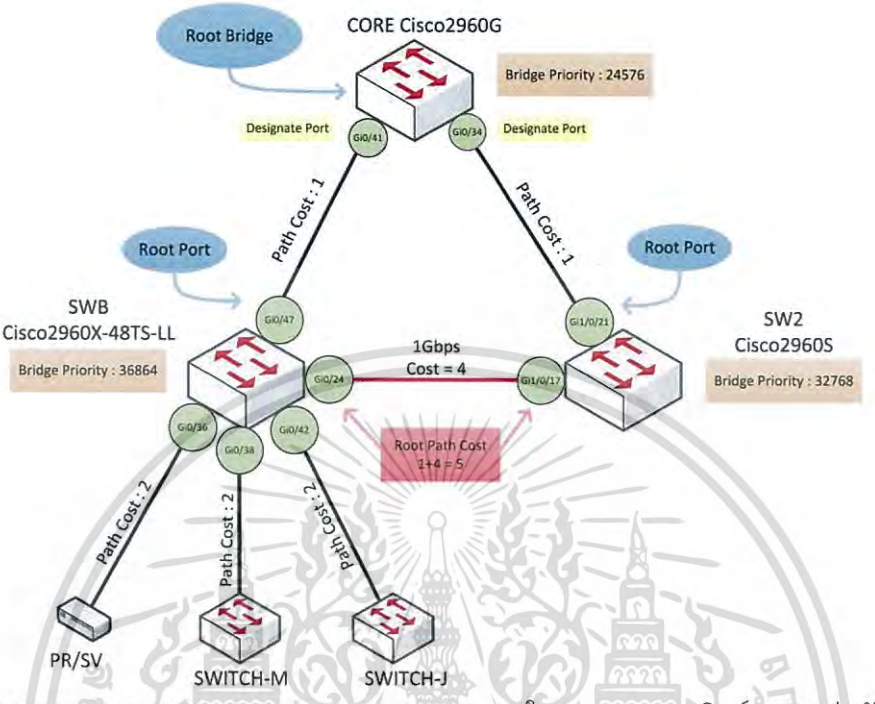
รูปที่ 4.24 แสดงผล STP บนสวิตช์ SWC ในการทดสอบสวิตช์ SWB ต่อกับ SWC

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Gi0/47 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ต Gi0/15 มีค่า Cost มากที่สุด จึงมีสถานะเป็น Blocking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 สวิตช์ SWB เชื่อมต่อกับสวิตช์ SW2

พอร์ต Gi0/24 ของสวิตช์ SWB เชื่อมต่อกับพอร์ต Gi1/0/17 ของสวิตช์ SW2



รูปที่ 4.25 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SWB ต่อกับ SW2

พอร์ต Gi0/24 ของสวิตช์ SWB และที่พอร์ต Gi1/0/17 ของสวิตช์ SW2 มีค่า Path Cost เท่ากับ 4 เนื่องจากพอร์ตไม่ได้มีการคอนฟิกค่า Path Cost เอาไว้ จึงขึ้นกับค่าแบนด์วิธของพอร์ตนั้นๆคือ 1 Gbps

- **การหา Root Bridge**

  อุปกรณ์ CORE มีค่า Bridge Priority สูงที่สุดจึงทำให้ได้เป็น Root Bridge

- **การหา Root Port** จะพิจารณาที่ค่า Root Path Cost เป็นอันดับแรก

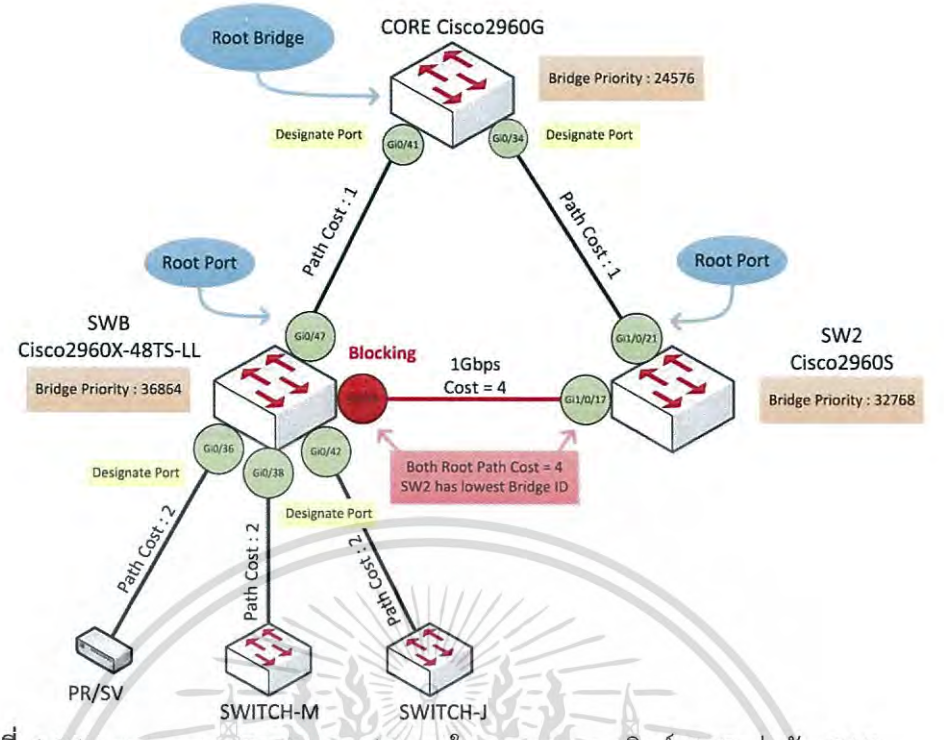
  SWB มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/47 มีค่า Root Path Cost เท่ากับ 1 จึงได้เป็น Root Port
- พอร์ต Gi0/24 ที่เชื่อมต่อเข้ากับสวิตช์ SW2 มีค่า Root Path Cost เท่ากับ 1+4 = 5
- พอร์ต Gi0/36, Gi0/38 และ Gi0/42 มีค่า Root Path Cost เท่ากับ 1+2 = 3

  SW2 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/21 มีค่า Root Path Cost เท่ากับ 1 จึงได้เป็น Root Port
- พอร์ต Gi1/0/17 ที่เชื่อมต่อเข้ากับสวิตช์ SWB มีค่า Root Path Cost เท่ากับ 1+4 = 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 แสดงการหา Designated Port ในการทดสอบสวิตช์ SWB ต่อกับ SW2

- **การหา Designated Port** จะพิจารณาพอร์ตที่เหลือที่มีค่า Path Cost ที่น้อยที่สุด

SWB มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/24 มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi0/36, Gi0/38 และ Gi0/42 มีค่า Root Path Cost เท่ากับ 2 จึงได้เป็น Designated Port

SW2 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/17 มีค่า Root Path Cost เท่ากับ 4

พอร์ต Gi0/24 ของอุปกรณ์ SWB และ พอร์ต Gi1/0/17 ของสวิตช์ SW2 มีค่า Root Path Cost เท่ากัน จึงพิจารณาที่ค่า Bridge ID แทน ซึ่งค่า Bridge Priority ของสวิตช์ SW2 มีค่าน้อยกว่าของสวิตช์ SWB พอร์ต Gi1/0/17 ของสวิตช์ SW2 จึงได้เป็น Designated Port และ พอร์ต Gi0/24 ของสวิตช์ SWB จึงกลายเป็น Block Port

รายละเอียดของสถานะและค่าที่ใช้ในการคำนวณของแต่ละพอร์ตที่ใช้งานอยู่ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SWB : ผลที่ได้เป็นดังรูปที่ 4.27

```
SWB#sh spanning-tree active
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        47 (GigabitEthernet0/47)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    36865  (priority 36864 sys-id-ext 1) <<<< Bridge Priority ของ SWB
              Address     005f.8688.e980 <<<< Mac Address ของ SWB
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type <<<< แสดงข้อมูลของแต่ละพอร์ตบน SWB
------------------- ---- --- --------- -------- --------------------------------
Gi0/24              Altn BLK 4         128.24   P2p  <<<< Connect to SW2
Gi0/36              Desg FWD 2         128.36   P2p  <<<< Connect to PR/SV
Gi0/38              Desg FWD 2         128.38   P2p  <<<< Connect to SWITCH-M
Gi0/42              Desg FWD 2         128.42   P2p  <<<< Connect to SWITCH-J
Gi0/47              Root FWD 1         128.47   P2p  <<<< Connect to CORE
```

รูปที่ 4.27 แสดงผล STP บนสวิตช์ SWB ในการทดสอบสวิตช์ SWB ต่อกับ SW2

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Gi0/47 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ต Gi0/24 มีค่า Cost มากกว่า จึงมีสถานะเป็น Blocking

- SW2 : ผลที่ได้เป็นดังรูปที่ 4.28

```
SW2#sh spanning-tree active

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        21 (GigabitEthernet1/0/21)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)  <<<< Bridge Priority ของ SW2
              Address     f47f.3562.3a00  <<<< Mac Address ของ SW2
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน SW2
------------------- ---- --- --------- -------- --------------------------------
Gi1/0/17            Desg LRN 4         128.17   P2p  <<<< Connect to SWB
Gi1/0/21            Root FWD 1         128.21   P2p  <<<< Connect to CORE
```

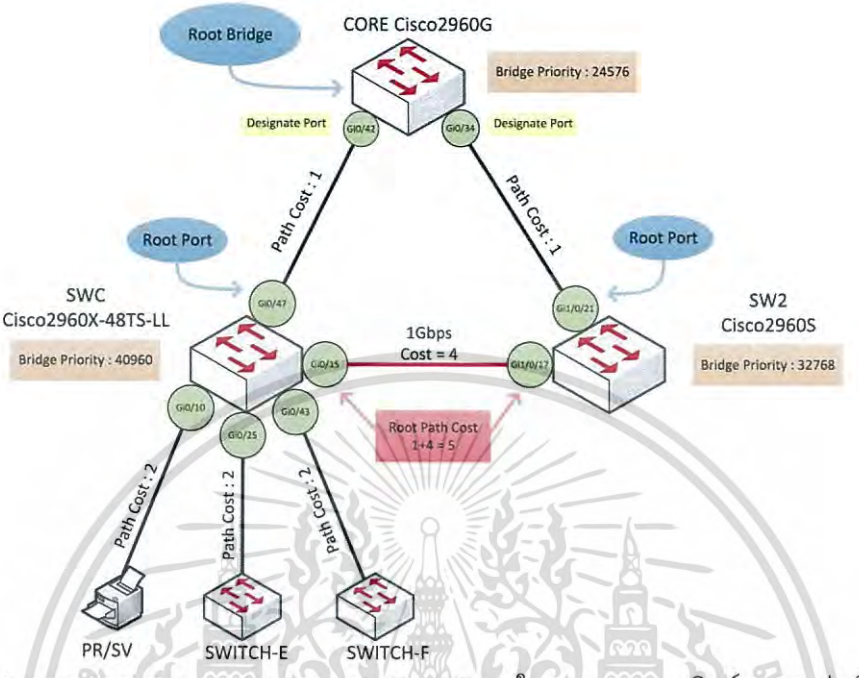
รูปที่ 4.28 แสดงผล STP บนสวิตช์ SW2 ในการทดสอบสวิตช์ SWB ต่อกับ SW2

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Gi1/0/21 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ต Gi1/0/17 ได้เป็น Designated Port และทุกพอร์ตจะมีสถานะเป็น Forwarding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6 สวิตช์ SWC เชื่อมต่อกับสวิตช์ SW2

พอร์ต Gi0/15 ของสวิตช์ SWC เชื่อมต่อกับพอร์ต Gi1/0/17 ของสวิตช์ SW2



รูปที่ 4.29 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ SWC ต่อกับ SW2

พอร์ต Gi0/15 ของสวิตช์ SWC และที่พอร์ต Gi1/0/17 ของสวิตช์ SW2 มีค่า Path Cost เท่ากับ 4 เนื่องจากไม่ได้มีการคอนฟิกค่า Path Cost ไว้ จึงขึ้นกับค่าแบนด์วิธของพอร์ตนั้นๆคือ 1 Gbps

- **การหา Root Bridge**

  อุปกรณ์ CORE มีค่า Bridge Priority สูงที่สุดจึงทำให้ได้เป็น Root Bridge

- **การหา Root Port** จะพิจารณาที่ค่า Root Path Cost เป็นอันดับแรก

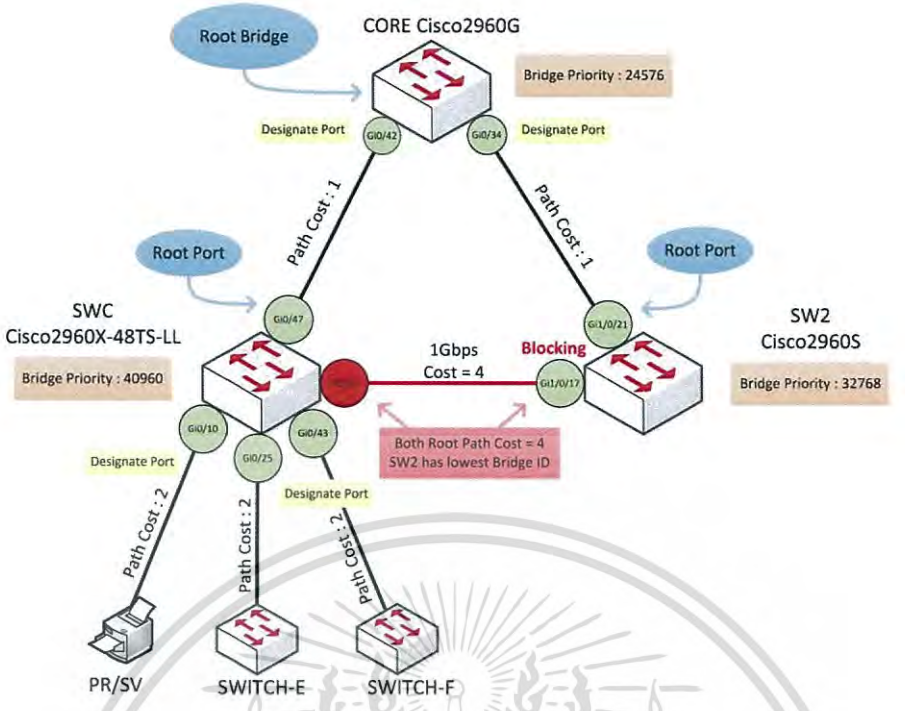
  SWC มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/47 มีค่า Root Path Cost เท่ากับ 1 จึงได้เป็น Root Port
- พอร์ต Gi0/15 ที่ต่อเข้ากับสวิตช์ SW2 มีค่า Root Path Cost เท่ากับ 1+4 = 5
- พอร์ต Gi0/10, Gi0/25 และ Gi0/43 มีค่า Root Path Cost เท่ากับ 1+2 = 3

  SW2 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/21 มีค่า Root Path Cost เท่ากับ 1 จึงได้เป็น Root Port
- พอร์ต Gi1/0/17 ที่ต่อเข้ากับสวิตช์ SWC มีค่า Root Path Cost เท่ากับ 1+4 = 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 แสดงการหา Designated Port ในการทดสอบสวิตช์ SWC ต่อกับ SW2

- **การหา Designated Port** จะพิจารณาพอร์ตที่เหลือที่มีค่า Path Cost ที่น้อยที่สุด

SWC มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/15 มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi0/10, Gi0/25 และ Gi0/43 มีค่า Root Path Cost เท่ากับ 2 จึงได้เป็น Designated Port

SW2 มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi1/0/17 มีค่า Root Path Cost เท่ากับ 4

พอร์ต Gi0/15 ของสวิตช์ SWC และ พอร์ต Gi1/0/17 ของสวิตช์ SW2 มีค่า Root Path Cost เท่ากัน จึงพิจารณาที่ค่า Bridge ID แทน ซึ่งค่า Bridge Priority ของสวิตช์ SW2 มีค่าน้อยกว่าของสวิตช์ SWC พอร์ต Gi1/0/17 ของสวิตช์ SW2 จึงได้เป็น Designate Port และ พอร์ต Gi0/15 ของสวิตช์ SWC จึงกลายเป็น Block Port

รายละเอียดของสถานะและค่าที่ใช้ในการคำนวณของแต่ละพอร์ตที่ใช้งานอยู่ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SWC : ผลที่ได้เป็นดังรูปที่ 4.31

```
SWC#sh spanning-tree active
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        47 (GigabitEthernet0/47)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    40961  (priority 40960 sys-id-ext 1)  <<<< Bridge Priority ของ SWC
              Address     005f.86aa.f000  <<<< Mac Address ของ SWC
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน SWC
------------------- ---- --- --------- -------- --------------------------------
Gi0/10              Desg FWD 2         128.10   P2p   <<<< Connect to Printer
Gi0/15              Altn BLK 4         128.15   P2p   <<<< Connect to SW2
Gi0/25              Desg FWD 2         128.25   P2p   <<<< Connect to SWITCH-E
Gi0/43              Desg FWD 2         128.43   P2p   <<<< Connect to SWITCH-F
Gi0/47              Root FWD 1         128.47   P2p   <<<< Connect to CORE
```

รูปที่ 4.31 แสดงผล STP บนสวิตช์ SWC ในการทดสอบสวิตช์ SWC ต่อกับ SW2

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Gi0/47 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ต Gi0/15 มีค่า Cost มากที่สุด จึงมีสถานะเป็น Blocking

- SW2 : ผลที่ได้เป็นดังรูปที่ 4.32

```
SW2#sh spanning-tree active
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        21 (GigabitEthernet1/0/21)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)  <<<< Bridge Priority ของ SW2
              Address     f47f.3562.3a00  <<<< Mac Address ของ SW2
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน SW2
------------------- ---- --- --------- -------- --------------------------------
Gi1/0/17            Desg LRN 4         128.17   P2p   <<<< Connect to SWC
Gi1/0/21            Root FWD 1         128.21   P2p   <<<< Connect to CORE
```

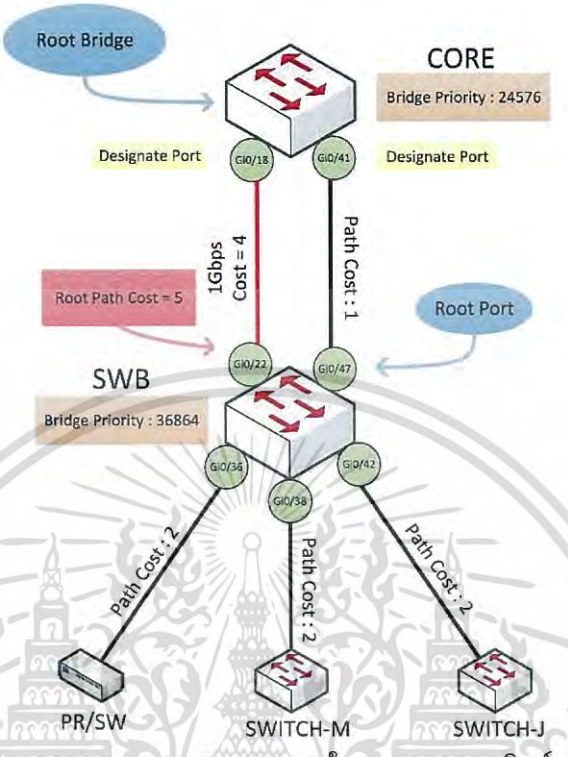
รูปที่ 4.32 แสดงผล STP บนสวิตช์ SW2 ในการทดสอบสวิตช์ SWC ต่อกับ SW2

รายละเอียด Interface แสดงให้เห็นว่าพอร์ต Gi1/0/21 ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำที่สุดเท่ากับ 1 ส่วนพอร์ต Gi1/0/17 จะเป็น Designate Port และทุกพอร์ตจะมีสถานะเป็น Forwarding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.7 สวิตช์ SWB เชื่อมต่อกับสวิตช์ CORE

พอร์ต Gi0/18 ของสวิตช์ CORE เชื่อมต่อกับพอร์ต Gi0/22 ของสวิตช์ SWB



รูปที่ 4.33 แสดงการหา Root Bridge และ Root Port ในการทดสอบสวิตช์ CORE ต่อกับ SWB

พอร์ต Gi0/18 ของสวิตช์ CORE และพอร์ต Gi0/22 ของสวิตช์ SWB จะมีค่า Path Cost เท่ากับ 4 เนื่องจากพอร์ตไม่ได้มีการคอนฟิกค่า Path Cost เอาไว้ จึงขึ้นกับค่าแบนด์วิธของพอร์ตนั้นๆ คือ 1 Gbps

- **การหา Root Bridge**

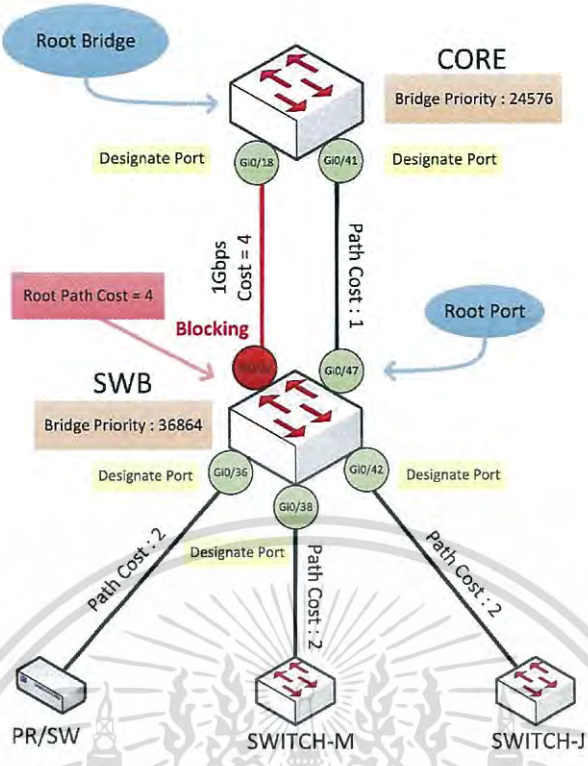
  อุปกรณ์ CORE มีค่า Bridge Priority สูงที่สุดจึงทำให้ได้เป็น Root Bridge

- **การหา Root Port** จะพิจารณาที่ค่า Root Path Cost เป็นอันดับแรก

  SWB มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/47 มีค่า Root Path Cost เท่ากับ 1 พอร์ตนี้จึงได้เป็น Root Port
- พอร์ต Gi0/22 ที่เชื่อมต่อเพิ่มเข้ากับสวิตช์ CORE มีค่า Root Path Cost เท่ากับ 1+4 = 5 นับจาก Root Bridge
- พอร์ต Gi0/36, Gi0/38 และ Gi0/42 มีค่า Root Path Costเท่ากับ 1+2 = 3 หรือ 1+4 =5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.34 แสดงการหา Designated Port ในการทดสอบสวิตช์ CORE ต่อกับ SWB

- **การหา Designate Port** จะพิจารณาพอร์ตที่เหลือที่มีค่า Path Cost ที่น้อยที่สุด

SWB มีค่า Root Path Cost ประจำพอร์ตดังนี้

- พอร์ต Gi0/22 มีค่า Root Path Cost เท่ากับ 4
- พอร์ต Gi0/36, Gi0/38, Gi0/42 มีค่า Root Path Cost เท่ากับ 2 จึงได้เป็น Designated Port

ส่วนพอร์ต Gi0/18 ของสวิตช์ CORE ได้ถูกกำหนดให้เป็น Designated Port ตั้งแต่ในขั้นตอนการหา Root Bridge แล้ว ดังนั้น พอร์ต Gi0/22 ของสวิตช์ SWB จึงเป็น Block Port

รายละเอียดของสถานะและค่าที่ใช้ในการคำนวณของแต่ละพอร์ตที่ใช้งานอยู่ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CORE : ผลที่ได้เป็นดังรูปที่ 4.35

```
CORE#sh spanning-tree active
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    24577  (priority 24576 sys-id-ext 1) <<<< Bridge Priority ของ CORE
              Address     001f.6d5c.3c00  <<<< Mac Address ของ CORE
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 15

Interface        Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน CORE
---------------- ---- --- --------- -------- --------------------------------
Gi0/18           Desg FWD 4         128.18   P2p   <<<< Connect to SWB
Gi0/23           Desg FWD 1         128.23   P2p   <<<< Connect to SWITCH-D
Gi0/34           Desg FWD 1         128.34   P2p   <<<< Connect to SW2
Gi0/40           Desg FWD 1         128.40   P2p   <<<< Connect to SWITCH-I
Gi0/41           Desg FWD 1         128.41   P2p   <<<< Connect to SWB
Gi0/42           Desg FWD 1         128.42   P2p   <<<< Connect to SWC
Gi0/46           Desg FWD 1         128.46   P2p   <<<< Connect to Firewall
Gi0/47           Desg FWD 1         128.47   P2p   <<<< Connect to WAN-Link Symphony
Gi0/48           Desg FWD 1         128.48   P2p   <<<< Connect to WAN-Link True
Po2              Desg FWD 1         128.64   P2p   <<<< Port Channel Connect to SW1
```

รูปที่ 4.35 แสดงผล STP บนสวิตช์ CORE ในการทดสอบสวิตช์ CORE ต่อกับ SWB

รายละเอียด Interface แสดงให้เห็นว่าทุกพอร์ตบนสวิตช์ CORE เป็น Designated Port และมีสถานะเป็น Forwarding ทุกพอร์ต ค่า Cost ของแต่ละพอร์ตมีค่าเท่ากับ 1 ยกเว้นพอร์ต Gi0/18 ที่ต่อเพิ่มเข้ากับสวิตช์ SWB

- SWB : ผลที่ได้เป็นดังรูปที่ 4.36

```
SWB#sh spanning-tree active

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577          <<<< Bridge Priority ของตัวที่เป็น Root Bridge
              Address     001f.6d5c.3c00 <<<< Mac Address ของตัวที่เป็น Root Bridge
              Cost        1              <<<< Cost ของตัวที่เป็น Root Bridge
              Port        47 (GigabitEthernet0/47)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    36865  (priority 36864 sys-id-ext 1) <<<< Bridge Priority ของ SWB
              Address     005f.8688.e980 <<<< Mac Address ของ SW2
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type  <<<< แสดงข้อมูลของแต่ละพอร์ตบน SWB
------------------- ---- --- --------- -------- --------------------------------
Gi0/22              Altn BLK 4         128.22   P2p   <<<< Connect to CORE
Gi0/36              Desg FWD 2         128.36   P2p   <<<< Connect to PR/SV
Gi0/38              Desg FWD 2         128.38   P2p   <<<< Connect to SWITCH-M
Gi0/42              Desg FWD 2         128.42   P2p   <<<< Connect to SWITCH-J
Gi0/47              Root FWD 1         128.47   P2p   <<<< Connect to CORE
```

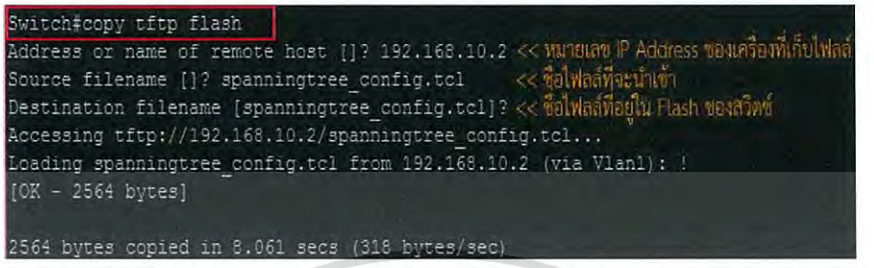
รูปที่ 4.36 แสดงผล STP บนสวิตช์ SWB ในการทดสอบสวิตช์ CORE ต่อกับ SWB

พอร์ต Gi0/47 ที่เป็นเส้นทางหลักในการเชื่อมต่อกับสวิตช์ CORE ได้เป็น Root Port เนื่องจากมีค่า Cost ต่ำสุด พอร์ต Gi0/36, Gi0/38, Gi0/42 มีค่า Cost เท่ากับ 2 จึงได้เป็น Designated Port ส่วนพอร์ต Gi0/22 ที่เชื่อมต่อเข้ากับสวิตช์ CORE ซึ่งไม่ใช่เส้นทางหลักมีค่า Cost มากที่สุดเท่ากับ 4 จึงมีสถานะเป็น Blocking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

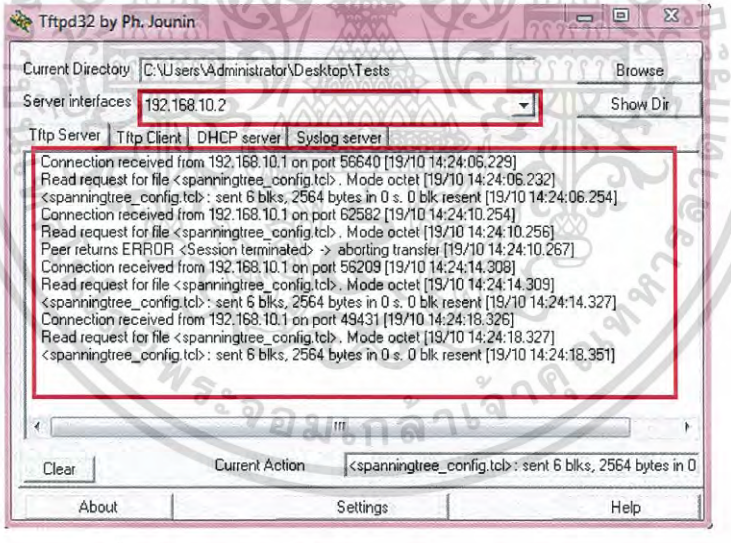
4.4 การทำงานของสคริปต์ช่วยในการพัฒนาระบบ

การนำสคริปต์ที่พัฒนาขึ้นเข้ามาใช้ในสวิตช์จำเป็นต้องคัดลอกจากคอมพิวเตอร์ผ่านโปรแกรม tftpd32d โดยการใช้คำสั่ง **copy tftp flash** บนโปรแกรม Putty ซึ่งเป็นคำสั่งในการคัดลอกไฟล์ที่อยู่บนคอมพิวเตอร์เข้ามาในหน่วยความจำแฟลชของสวิตช์



รูปที่ 4.37 แสดงการคัดลอกไฟล์สคริปต์บนสวิตช์จากคอมพิวเตอร์

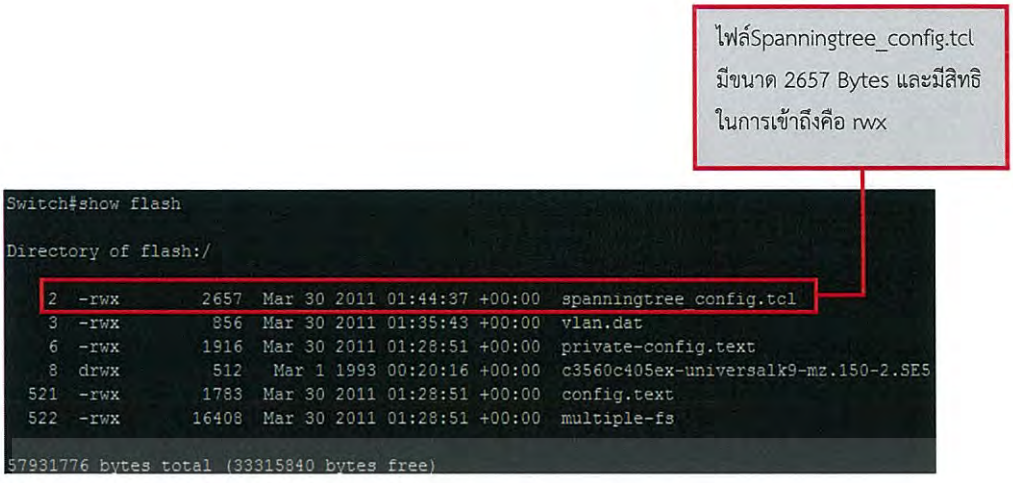
โปรแกรม Tftpd32 ทำงานบนคอมพิวเตอร์ เมื่อมีการคัดลอกไฟล์จากคอมพิวเตอร์เข้าสู่สวิตช์ โปรแกรม Tftpd32 จะสื่อสารกับสวิตช์โดยอาศัย IP Address ของคอมพิวเตอร์และสวิตช์ โดยผ่านการเชื่อมต่อของสาย UTP แสดงรายละเอียดการทำงานของโปรแกรม Tftpd32 ดังรูปที่ 4.38



รูปที่ 4.38 แสดงการทำงานของโปรแกรม Tftpd 32

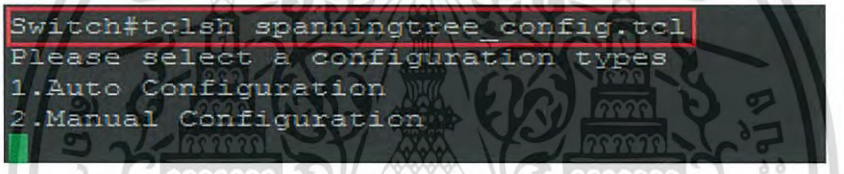
หลังจากคัดลอกไฟล์มาใส่ในหน่วยความจำแฟลช สามารถตรวจสอบรายละเอียดไฟล์บนสวิตช์ได้โดยการใช้คำสั่ง **show flash** บนโปรแกรม Putty แสดงดังรูปที่ 4.39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



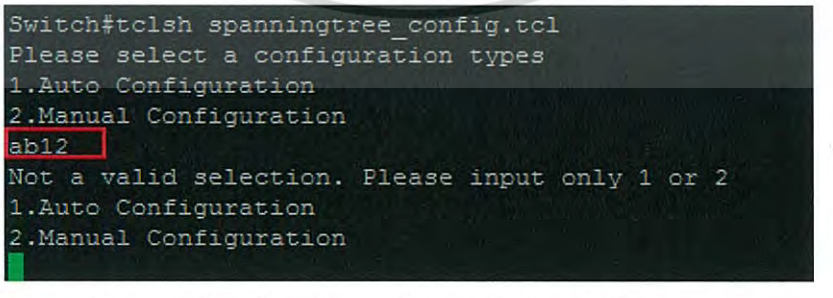
รูปที่ 4.39 แสดงรายละเอียดไฟล์ในหน่วยความจำแฟลช

ในการเรียกไฟล์สคริปต์จากหน่วยความจำแฟลชที่คัดลอกมาจากคอมพิวเตอร์ขึ้นมาทำงานบนสวิตช์จะใช้คำสั่ง **tclsh spanningtree_config.tcl** ดังรูปที่ 4.40



รูปที่ 4.40 แสดงการเรียกไฟล์สคริปต์ขึ้นมาทำงาน

ขั้นตอนแรกหลังจากใช้คำสั่งเรียกไฟล์สคริปต์ขึ้นมาทำงาน ระบบจะเข้าสู่โหมดการเลือก Configuration Types ซึ่งในโหมดนี้มีตัวเลือก 2 ตัวเลือกคือ Auto Configuration และ Manual Configuration ในการรับค่าจากผู้ใช้งานสามารถรับได้เพียง 2 ค่า คือ 1 และ 2 ในกรณีที่ผู้ใช้งานใส่ค่าที่นอกเหนือจากนี้ ระบบจะมีการแจ้งเตือนผู้ใช้งานว่าใส่ค่าไม่ถูกต้อง และให้ใส่ค่าใหม่อีกครั้ง แสดงดังรูปที่ 4.41

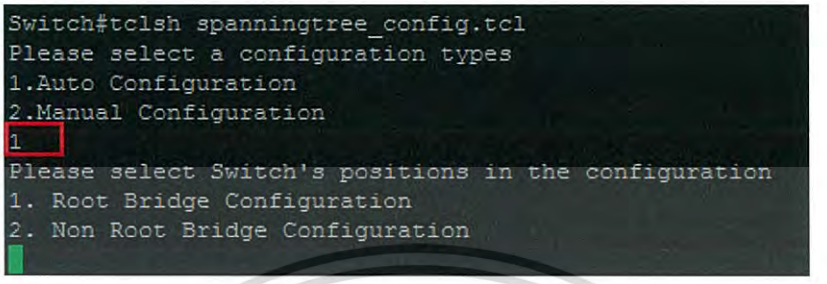


รูปที่ 4.41 แสดงข้อความเตือนเมื่อผู้ใช้งานใส่ค่าผิดในโหมดการเลือก Configuration Types

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) Auto Configuration

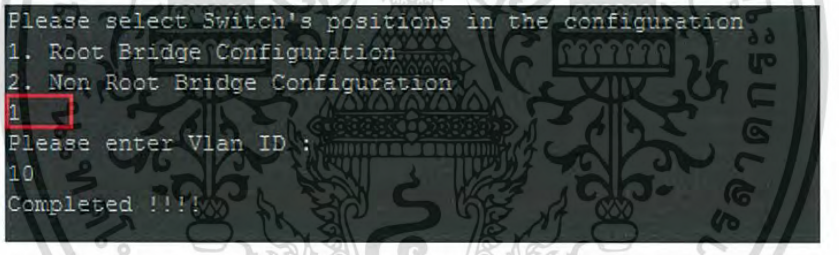
เมื่อมีการรับค่า 1 เข้ามา ระบบจะเข้าสู่โหมด Auto Configuration ซึ่งเป็นการคอนฟิกแบบอัตโนมัติ โดยระบบจะแสดงตัวเลือกให้ผู้ใช้งานเลือกว่าจะให้สวิตช์ที่คอนฟิกอยู่นั้นเป็น Root Bridge หรือเป็น Non Root Bridge โดยการป้อนค่า “1” หรือ “2” แสดงดังรูปที่ 4.42



รูปที่ 4.42 แสดงการรับข้อมูลในโหมด Auto Configuration

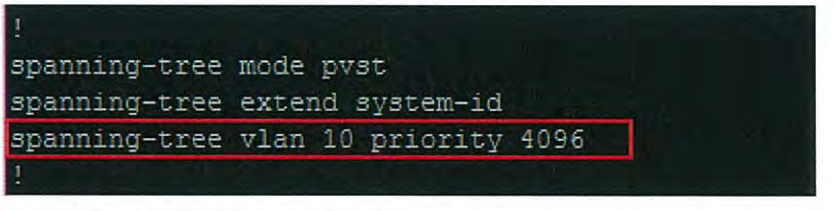
1.1) Root Bridge Configuration

เมื่อมีการรับค่า 1 เข้ามา ระบบจะเข้าสู่โหมด Root Bridge Configuration และจะแสดงข้อความเพื่อรับค่า Vlan ID จากผู้ใช้งาน แสดงดังรูปที่ 4.43



รูปที่ 4.43 แสดงการรับข้อมูลในโหมด Root Bridge Configuration

เมื่อผู้ใช้งานใส่ค่า Vlan ID ระบบจะทำการคอนฟิกค่า Bridge Priority ให้กับสวิตช์ โดยกำหนดค่าให้อยู่ใน Vlan ที่ผู้ใช้งานกำหนดเข้ามา ซึ่งค่า Bridge Priority ที่ระบบคอนฟิกเท่ากับ 4096 และจะแสดงข้อความ “Completed” เพื่อแสดงว่าการคอนฟิกสำเร็จแล้ว หลังจากนั้นจะทำการออกจากโหมด Tcl ผลการคอนฟิกสามารถแสดงได้จากคำสั่ง **show run** แสดงดังรูปที่ 4.44



รูปที่ 4.44 ผลการคอนฟิกจากโหมด Root Bridge Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2) Non Root Bridge Configuration

เมื่อมีการรับค่า 2 เข้ามา ระบบจะเข้าสู่โหมด Non Root Bridge Configuration และระบบจะแสดงข้อความเพื่อรับค่า Vlan ID จากผู้ใช้งาน แสดงดังรูปที่ 4.45

```
Please select Switch's positions in the configuration
1. Root Bridge Configuration
2. Non Root Bridge Configuration
2
Please enter Vlan ID :
10
Completed !!!!
```

รูปที่ 4.45 แสดงการรับข้อมูลในโหมด Non Root Bridge Configuration

เมื่อผู้ใช้งานป้อนค่า Vlan ID ระบบจะทำการคอนฟิกค่า Bridge Priority ให้กับสวิตช์ ซึ่งค่า Bridge Priority ที่ระบบคอนฟิกให้กับสวิตช์ที่เป็น Non Root Bridge เท่ากับ 8129 และจะแสดงข้อความ “Completed” เพื่อแสดงว่าการคอนฟิกสำเร็จแล้ว หลังจากนั้นจะทำการออกจากโหมด Tcl ผลการคอนฟิกสามารถแสดงดังรูปที่ 4.46

```
!
spanning-tree mode pvst
spanning-tree extend system-id
spanning-tree vlan 10 priority 8192
!
```

รูปที่ 4.46 ผลการคอนฟิกจากโหมด Non Root Bridge Configuration

2) Manual Configuration

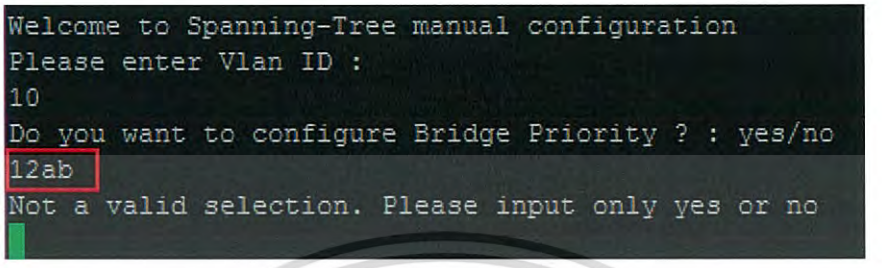
เมื่อมีการรับค่า 2 เข้ามาระบบจะเข้าสู่โหมด Manual Configuration หลังจากนั้นระบบจะแสดงข้อความเพื่อรับค่า Vlan ID จากผู้ใช้งานแสดงดังรูปที่ 4.47

```
Switch#tclsh spanningtree_config.tcl
Please select a configuration types
1.Auto configuration
2.Manual Configuration
2
Welcome to Spanning-Tree manual configuration
Please enter Vlan ID :
10
Do you want to configure Bridge Priority ? : yes/no
```

รูปที่ 4.47 แสดงการรับข้อมูลในโหมด Manual Configuration

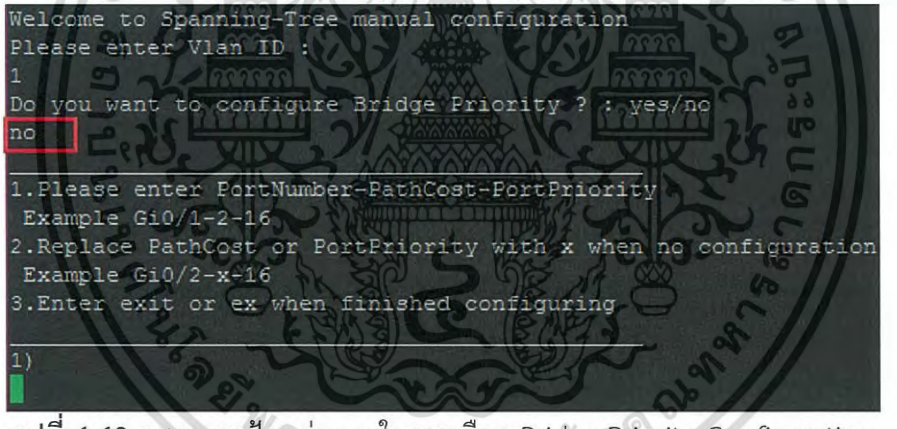
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้งานป้อนค่า Vlan ID ระบบจะแสดงข้อความถามผู้ใช้งานว่าต้องการคอนฟิกค่า Bridge Priority หรือไม่ ซึ่งในการรับค่าจากผู้ใช้งานสามารถรับได้เพียง “yes”, “y”, “no” และ “n” ในกรณีที่ผู้ใช้งานป้อนค่าที่นอกเหนือจากนี้ ระบบจะมีการแจ้งเตือนผู้ใช้งานว่าใส่ค่าไม่ถูกต้อง และให้ใส่ค่าใหม่อีกครั้ง แสดงดังรูปที่ 4.48



รูปที่ 4.48 แสดงข้อความเตือนเมื่อผู้ใช้งานใส่ค่าผิดในการเลือก Bridge Priority Configuration

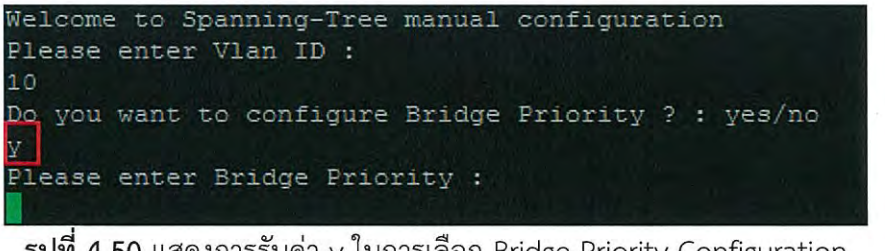
ถ้าผู้ใช้งานป้อนค่า “no” หรือ “n” ระบบจะข้ามขั้นตอนการคอนฟิก Bridge Priority และเข้าสู่โหมดการคอนฟิกพื้นฐานทันที แสดงดังรูปที่ 4.49



รูปที่ 4.49 แสดงการป้อนค่า no ในการเลือก Bridge Priority Configuration

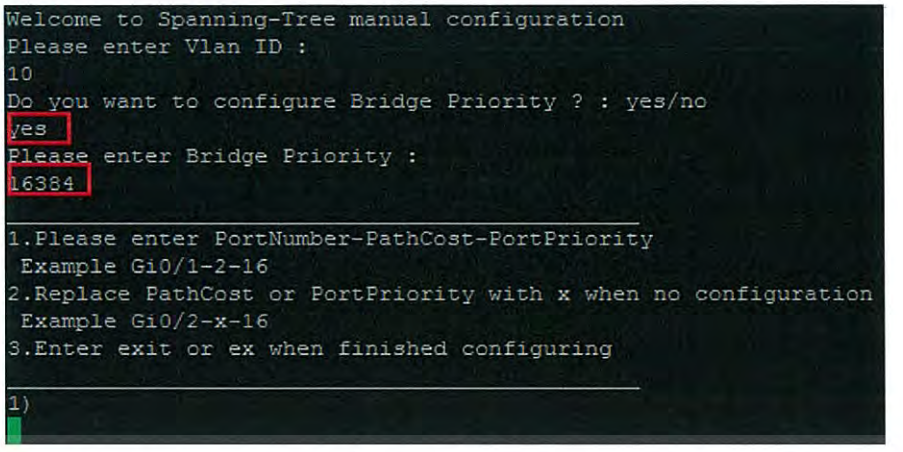
2.1) Bridge Priority Configuration

เมื่อผู้ใช้งานป้อนค่า “yes” หรือ “y” ระบบจะเข้าสู่โหมด Bridge Priority Configuration และจะแสดงข้อความรับค่า Bridge Priority จากผู้ใช้งาน แสดงดังรูปที่ 4.50 และ 4.51



รูปที่ 4.50 แสดงการรับค่า y ในการเลือก Bridge Priority Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

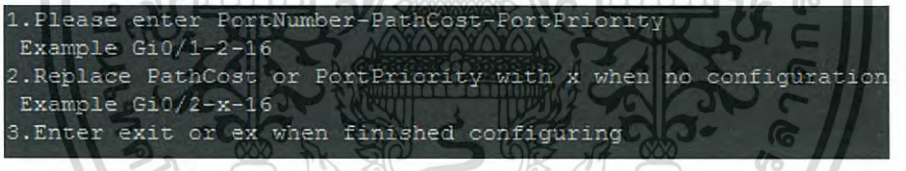


รูปที่ 4.51 แสดงการรับข้อมูลในโหมด Bridge Priority Configuration

จากรูปที่ 4.51 เมื่อผู้ใช้งานป้อนค่า Bridge Priority เรียบร้อย ระบบจะทำการคอนฟิกค่า ที่ผู้ใช้งานป้อนเข้ามาให้กับสวิตช์ หลังจากนั้นระบบจะเข้าสู่โหมดคอนฟิกพื้นฐาน

2.2) การคอนฟิกค่าพื้นฐาน

เป็นการคอนฟิกค่าหลักที่ใช้ในการคำนวณ STP ให้กับแต่ละพอร์ตบนสวิตช์ ระบบจะแสดงรายละเอียดรูปแบบในการป้อนค่าให้กับผู้ใช้งาน แสดงได้ดังรูปที่ 4.52



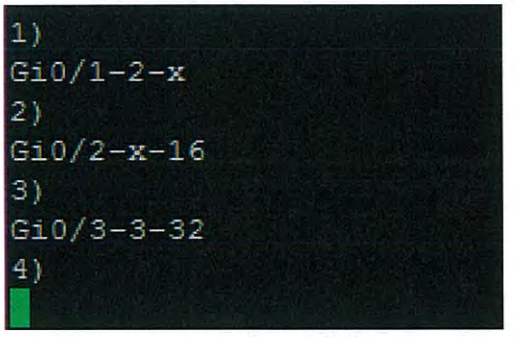
รูปที่ 4.52 แสดงรายละเอียดและรูปแบบในการป้อนค่า

- ผู้ใช้งานจะต้องป้อนค่าตามรูปแบบดังนี้
  PortNumber – PathCost – PortPriority
- กรณีที่ผู้ใช้งานไม่ต้องการคอนฟิกค่าใดค่าหนึ่ง ให้ผู้ใช้งานป้อนค่า “x” แทนค่านั้นๆ
- เมื่อผู้ใช้งานต้องการออกจากระบบ ให้ผู้ใช้งานป้อนค่า “exit” หรือ “ex”

ทดสอบการรับค่าจากผู้ใช้งาน แบ่งได้เป็น 3 กรณีดังนี้

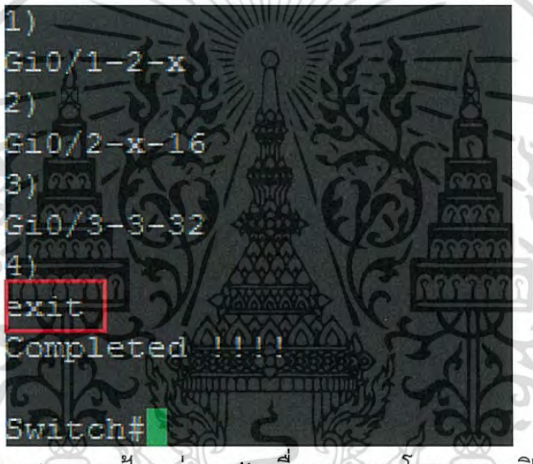
- กรณีที่ 1 ผู้ใช้งานป้อนเฉพาะค่า Path Cost
- กรณีที่ 2 ผู้ใช้งานป้อนเฉพาะค่า Port Priority
- กรณีที่ 3 ผู้ใช้งานป้อนทั้งค่า Path Cost และ Port Priority

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

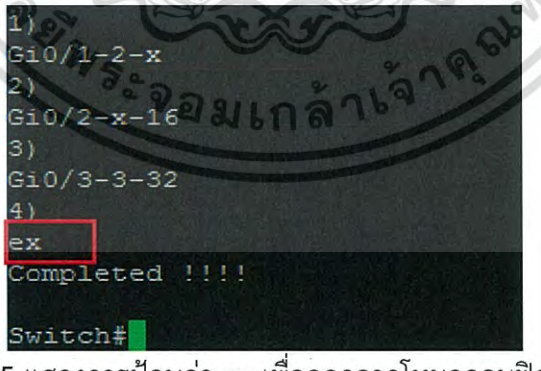


รูปที่ 4.53 แสดงการทดสอบการป้อนค่าในโหมดการคอนฟิกพื้นฐาน

เมื่อผู้ใช้งานป้อนค่า “exit” หรือ “ex” เข้ามา ระบบจะสิ้นสุดการรับค่าจากผู้ใช้งาน โดยจะแสดงข้อความ Completed เพื่อแสดงว่าการคอนฟิกค่าสำเร็จแล้ว หลังจากนั้นระบบจะออกจากโหมด tcl แสดงดังรูปที่ 4.54 และ 4.55



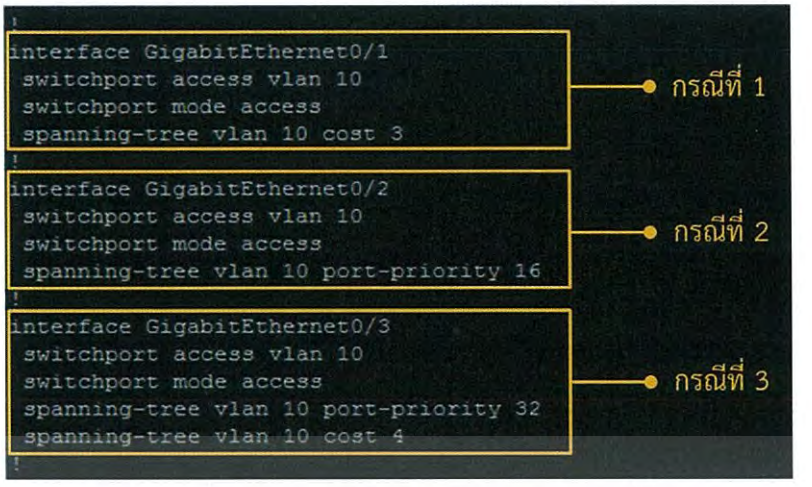
รูปที่ 4.54 แสดงการป้อนค่า exit เพื่อออกจากโหมดคอนฟิกพื้นฐาน



รูปที่ 4.55 แสดงการป้อนค่า ex เพื่อออกจากโหมดคอนฟิกพื้นฐาน

ผลการคอนฟิกสามารถแสดงได้จากคำสั่ง **show run** แสดงได้ดังรูปที่ 4.56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.56 ผลการคอนฟิกจากโหมดการคอนฟิกพื้นฐาน

4.5 การทำงานของโปรแกรมช่วยในการพัฒนาระบบ

1) ไฟล์ .csv

ผู้ใช้งานสร้างไฟล์ .csv จากโปรแกรม Excel โดยเก็บค่าที่ใช้ในการคำนวณ STP ตัวอย่างแสดงดังตารางที่ 4.1

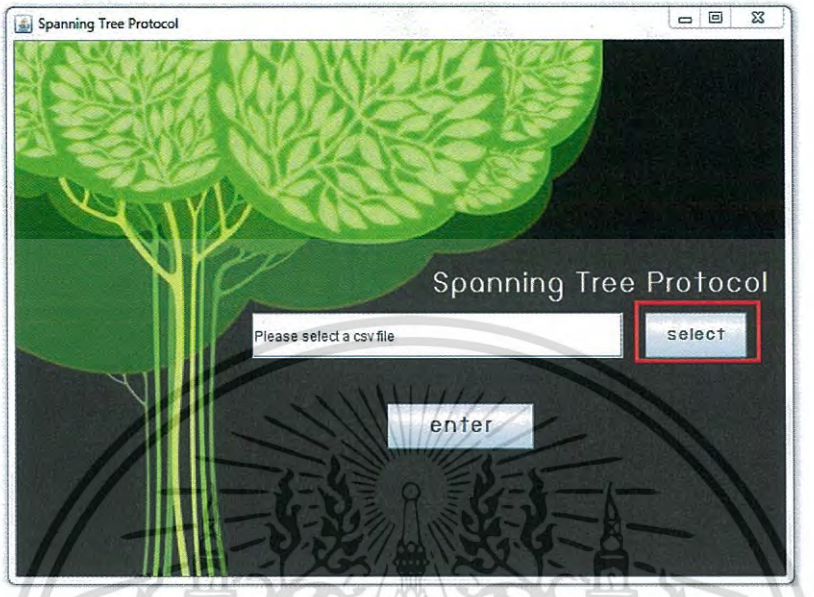
ตารางที่ 4.1 แสดงตัวอย่างการเก็บข้อมูลเพื่อใช้ในโปรแกรม

Switch	Vlan	BridgePriority	PortNumber	PathCost	PortPriority
CORE	1	4096	gi0/1	1	16
CORE	1		gi0/2	1	16
CORE	1		gi0/23	1	16
CORE	1		gi0/34	1	16
CORE	1		gi0/40	2	16
CORE	1		gi0/41	1	16
CORE	1		gi0/42	1	16
CORE	1		gi0/46	1	16
CORE	1		gi0/47	1	16
CORE	1		gi0/48	1	16
SW1	1	28672	gi1/0/1	1	-
SW1	1		gi1/0/2	1	32
SW1	1		gi1/0/18	2	-
SW1	1		gi1/0/20	2	-
SWB	1	36864	gi0/36	2	32
SWB	1		gi0/38	2	32
SWB	1		gi0/42	2	32
SWB	1		gi0/47	1	16
SWC	1	40960	gi0/10	2	16
SWC	1		gi0/25	2	-
SWC	1		gi0/43	2	-
SWC	1		gi0/47	1	-
SW2	1	32768	gi1/0/21	1	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

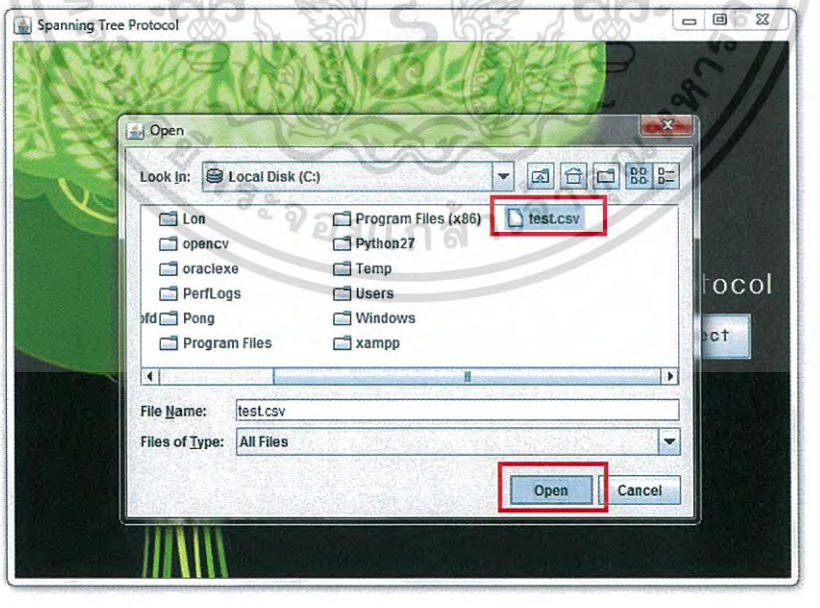
2) หน้าจอโปรแกรม

เมื่อผู้ใช้งานเปิดโปรแกรม STPconfig.exe ขึ้นมา ระบบจะแสดงหน้าจอเพื่อให้ผู้ใช้งานเลือกไฟล์ .csv ที่ได้ใส่ข้อมูลในการคอนฟิก STP ไว้เรียบร้อยแล้ว โดยการกดที่ปุ่ม “select”



รูปที่ 4. 57 แสดงหน้าจอเมื่อเปิดโปรแกรม

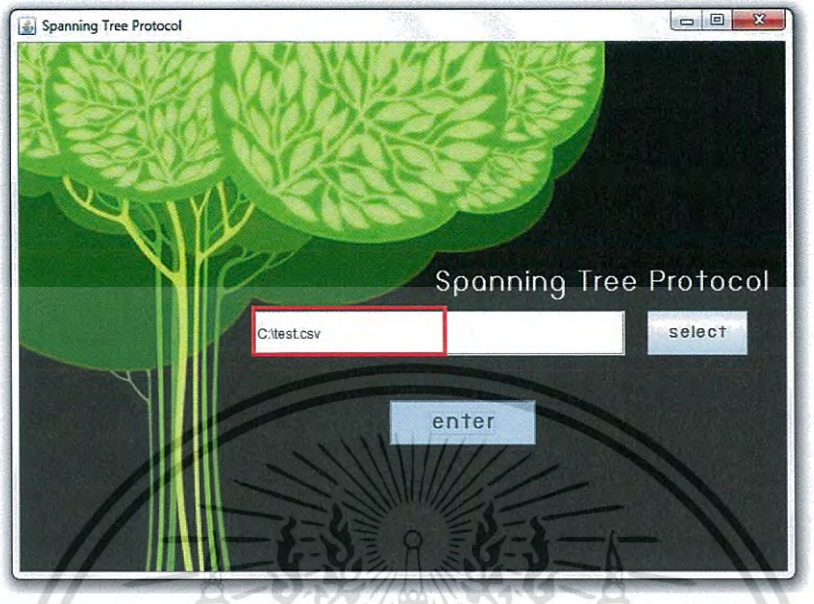
ระบบจะแสดงรายละเอียดของไฟล์ที่จัดเก็บในเครื่องคอมพิวเตอร์เพื่อให้ผู้ใช้งานเลือก เมื่อผู้ใช้งานเลือกไฟล์ .csv เสร็จเรียบร้อบ ให้กดที่ปุ่ม “Open”



รูปที่ 4.58 แสดงหน้าจอสำหรับให้ผู้ใช้งานเลือกไฟล์ .csv

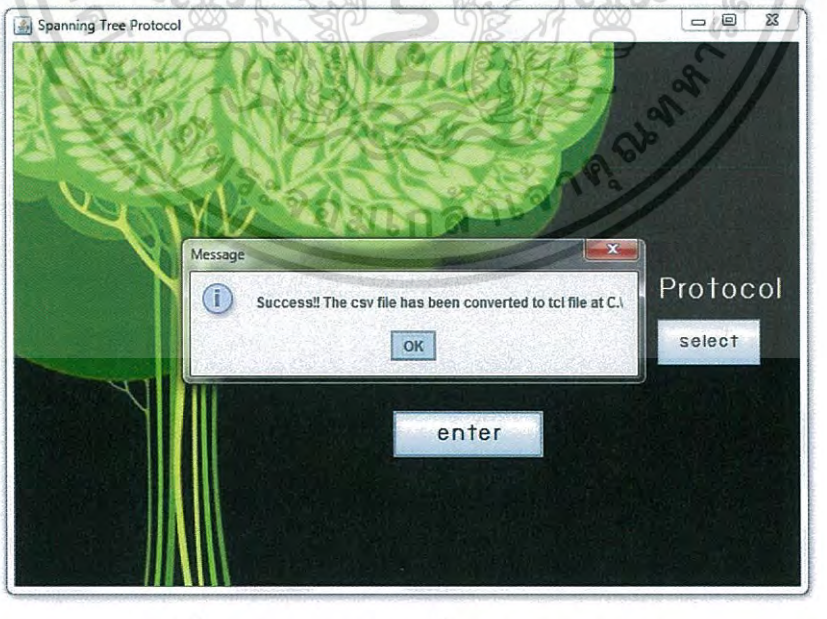
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้งานเลือกไฟล์เสร็จเรียบร้อย ระบบจะแสดง Path ของไฟล์ที่ผู้ใช้งานเลือกในช่อง TextBox แสดงดังรูปที่ 4.59



รูปที่ 4.59 แสดงหน้าจอเมื่อผู้ใช้งานเลือกไฟล์ .csv แล้ว

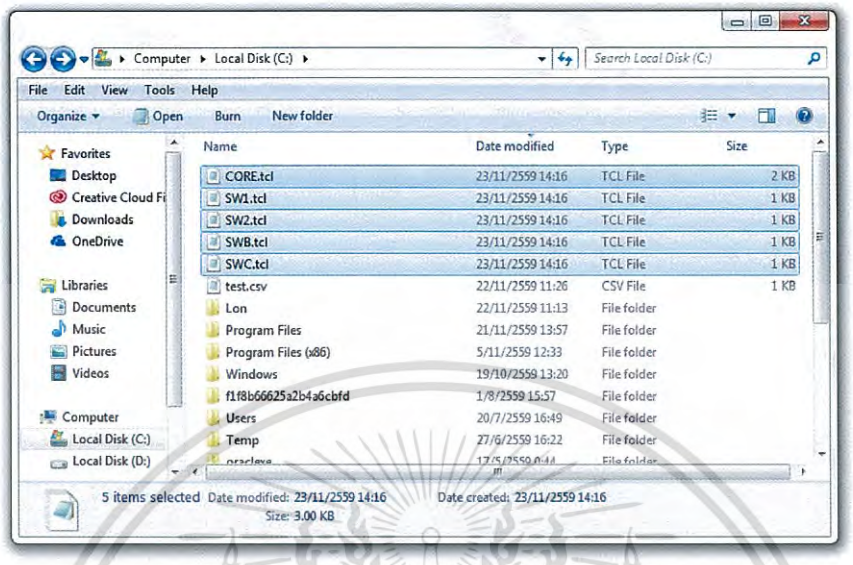
เมื่อผู้ใช้งานกดปุ่ม “enter” ระบบจะทำการอ่านไฟล์ .csv ที่ผู้ใช้งานเลือกเข้ามา และสร้างเป็นไฟล์ .tcl เพื่อให้ผู้ใช้งานนำไปเรียกใช้บนสวิตช์ เมื่อระบบทำงานเสร็จจะแสดงข้อความแก่ผู้ใช้งาน แสดงดังรูปที่ 4.60



รูปที่ 4.60 แสดงหน้าจอเมื่อระบบทำงานเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

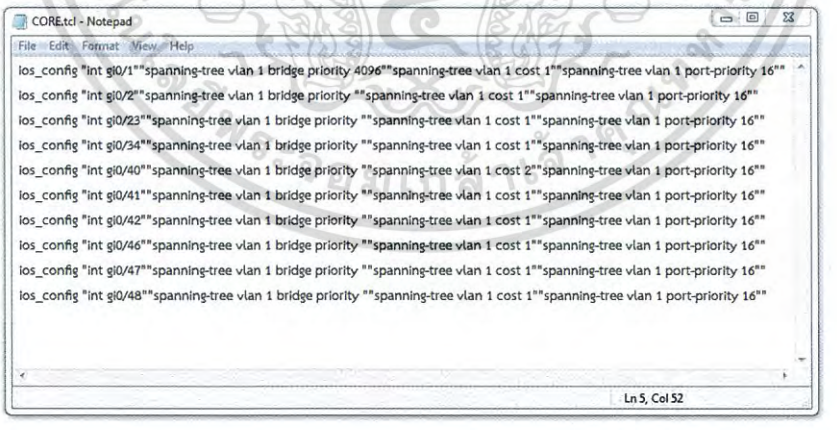
ไฟล์ .tcl ที่ระบบสร้างขึ้นมาใหม่จะถูกเก็บไว้ที่ Drive C บนคอมพิวเตอร์ของผู้ใช้งาน แสดงดังรูปที่ 4.61



รูปที่ 4.61 แสดงที่อยู่ของไฟล์ .tcl

3) ไฟล์ .tcl

ระบบจะนำข้อมูลจากตารางที่ 4.1 ซึ่งเป็นไฟล์ .csv ที่ผู้ใช้งานเลือกเข้ามาแปลงเป็นคำสั่งในการคอนฟิก สำหรับนำไปเรียกใช้บนสวิตช์ ซึ่งคำสั่งนี้เป็นคำสั่งที่ใช้ภาษา tcl ดังนั้นไฟล์ที่ได้ใหม่จึงเป็นไฟล์ .tcl แสดงดังรูปที่ 4.62



รูปที่ 4.62 แสดงคำสั่งในไฟล์ .tcl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

1) สวิตช์สามารถส่งข้อมูลสื่อสารระหว่างกันได้ในขณะที่มีการต่อระบบเครือข่ายในรูปแบบวนลูป
2) สวิตช์สามารถเรียนรู้และเลือก Block พอร์ตที่มีปัญหาเพื่อให้พอร์ตนั้นหยุดการรับส่งข้อมูลชั่วคราวได้
3) สวิตช์สามารถหาเส้นทางใหม่ได้เมื่อเส้นทางหลักในการสื่อสารเกิดมีปัญหา
4) สวิตช์สามารถรับและคอนฟิกค่าที่ใช้ในการคำนวณ STP จากผู้ใช้งานผ่านการใช้สคริปต์ได้
5) โปรแกรมสามารถนำข้อมูลจากไฟล์ .csv ที่เก็บค่าที่ใช้ในการคำนวณ STP มาแปลงเป็นคำสั่งเพื่อใช้คอนฟิกสวิตช์เป็นภาษา Tcl ได้
6) สามารถใช้สคริปต์และโปรแกรมช่วยลดระยะเวลาในการคอนฟิกบนสวิตช์ได้

5.2 ข้อจำกัดในการพัฒนาระบบ

1) ระบบเครือข่ายเดิมของลูกค้ามีการเชื่อมต่อสายที่ไม่เป็นระเบียบ จึงส่งผลให้เสียเวลาและเกิดความผิดพลาดในการ Migrate เนื่องจากข้อมูลการเชื่อมต่อไม่ตรงกับที่ออกแบบไว้
2) เวลาในการทำงานเป็นอุปสรรคต่อการทำงาน เช่น ไม่สามารถเข้าไปปิดตัวอุปกรณ์หลักเพื่อทดสอบปัญหาที่เกิดขึ้นได้ขณะที่มีพนักงานทำงานอยู่
3) การคอนฟิกตัวอุปกรณ์ในแบบปกติทำได้อย่างล่าช้าเนื่องจากเป็นคำสั่งใหม่ที่ผู้จัดทำยังไม่เข้าใจ จึงต้องใช้เวลาในการศึกษา
4) ภาษา Tcl ที่ใช้ในการพัฒนาสคริปต์มีข้อจำกัดค่อนข้างมาก เช่น สวิตช์ที่รองรับภาษา Tcl จะมีเฉพาะ version 12.2(25) ขึ้นไปเท่านั้น จึงทำให้ไม่สามารถพัฒนาให้ได้มีประสิทธิภาพเท่าที่ควร
5) การคัดลอกไฟล์สคริปต์ลงสวิตช์สามารถใช้ได้กับสวิตช์บางรุ่นเท่านั้น เนื่องจากรุ่นเก่ามีข้อจำกัดในเรื่องหน่วยความจำแฟลชที่ไม่สามารถเข้าถึงหรือเปลี่ยนแปลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

1) ในการวิเคราะห์ระบบต้องศึกษาข้อมูลให้ครอบคลุมทั้งด้านปัญหาและความต้องการการใช้งานระบบ เพื่อให้การออกแบบระบบงานครบถ้วนตามความต้องการและเพื่อไม่ให้เกิดการแก้ไขระบบงานตามมาภายหลัง
2) โครงสร้างการทำงานของสคริปต์ควรมีรูปแบบที่ง่ายและรวดเร็วในขั้นตอนการรับข้อมูลจากผู้ใช้งาน เพื่อไม่ให้เสียเวลาและยุ่งยากเกิน
3) ไฟล์สคริปต์ที่พัฒนาขึ้นควรทำงานได้กับสวิตช์ทุกรุ่นและทุกยี่ห้อ
4) ไฟลล์สคริปต์ที่พัฒนาขึ้นควรครอบคลุมการคอนฟิกรูปแบบอื่นๆด้วย เช่น การกำหนด IP Address ให้กับอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] เอกสิทธิ์ วิริยจารี. 2548. **เรียนรู้ระบบเน็ตเวิร์กจากอุปกรณ์ของ Cisco ภาคปฏิบัติ**. ครั้งที่พิมพ์ 9. กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

[2] ขัตติยา ฮิวซ์ส. 2556. **ความหมายของ LAN MAN WAN**. [Online]. Available : http://khattiyalovely.blogspot.com/2013/11/lan-man-wan.html

[3] Fon Jay Park. 2556. **สาย UTP คืออะไร**. [Online]. Available : http://comnetwork2013.blogspot.com/2013/11/utp.html

[4] Thananchai Panyatavahirun. 2556. **EtherChannel**. [Online]. Available : http://running-config.blogspot.com/2011/03/etherchannel-cisco-catalyst-switch.html

[5] Ice Suntisuk. 2556. **การทำงาน ARP Protocol (Address Resolution Protocol)**. [Online]. Available : http://icesuntisuk.blogspot.com/2014/06/arp-protocol-address-resolution-protocol.html

[6] ชวลิต ทินกรสูตินุตร. 2556. **ICMP**. [Online]. Available : http://www.tnetsecurity.com/content_basic/tcp_ip_knowledge.php

[7] TCL. 2556. **Tool Command Language**. สื่อการเรียนการสอน Word. [Online]. Available : staff.cs.psu.ac.th/noi/cs323-554/ปโท/g4322027/TclTk.doc

[8] ICMP. 2556. **หน้าที่หลักของโปรโตคอล ICMP**. [Online]. Available : http://amaneamisa.siam2web.com//?cid=1084628

[9] ComponentBases584. 2556. **วิธีใช้ NetBeasns เบื้องต้น**. [Online]. Available : http://component584.blogspot.com/2011/06/netbeans-ide-1.html

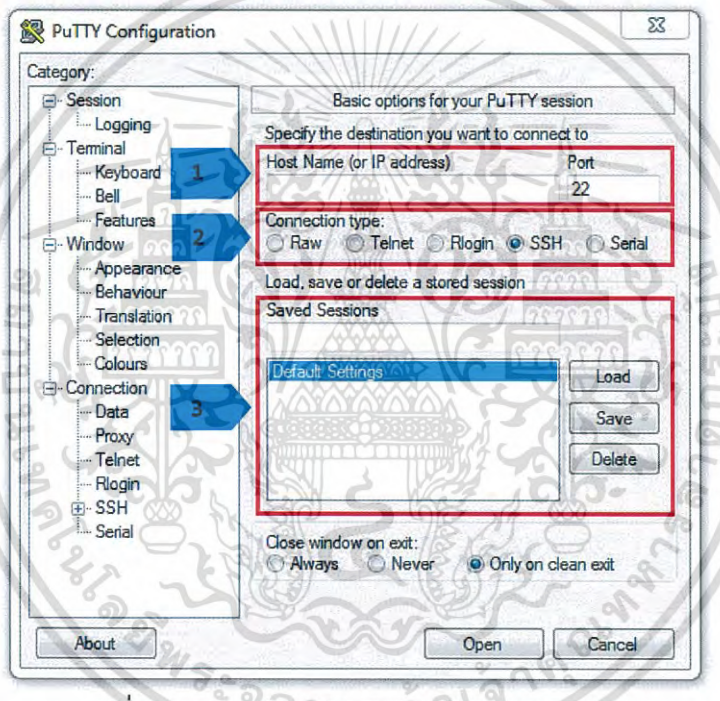
[10] THITI YAMSUNG. 2556. 2556. **การใช้งาน putty เบื้องต้น**. [Online]. Available : http://www.thitiblog.com/blog/1733

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
วิธีการใช้พื้นฐานโปรแกรม Putty

รูปที่ ก.1 โปรแกรม putty

เปิดไฟล์ putty.exe เพื่อเรียกใช้โปรแกรม putty หน้าจอโปรแกรม แสดงดังรูปที่ ก.2



รูปที่ ก.2 แสดงส่วนประกอบของโปรแกรม putty

1) ส่วนของการ Config ว่าเราจะเชื่อมต่อไปยัง Server IP และ Port อะไร
2) ส่วนของการเลือกประเภทของการเชื่อมต่อ
3) ส่วนของการบันทึกค่าที่ Config ไว้ เพื่อเพิ่มความสะดวกเมื่อใช้งานครั้งต่อไป

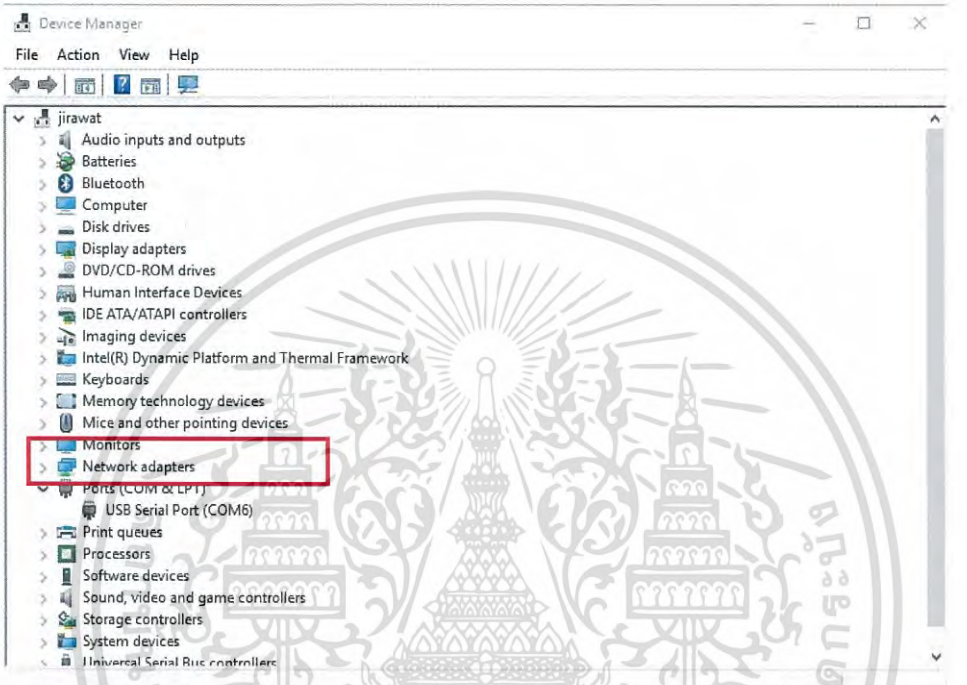
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เชื่อมต่อประเภท Serial

การเชื่อมต่อประเภทนี้เป็นการเชื่อมต่อโดยใช้สาย Console ต่อโดยตรงกับอุปกรณ์ สามารถตรวจสอบ Serial line ได้โดยการคลิกขวาที่

My Computer >> Properties >> Device Manager

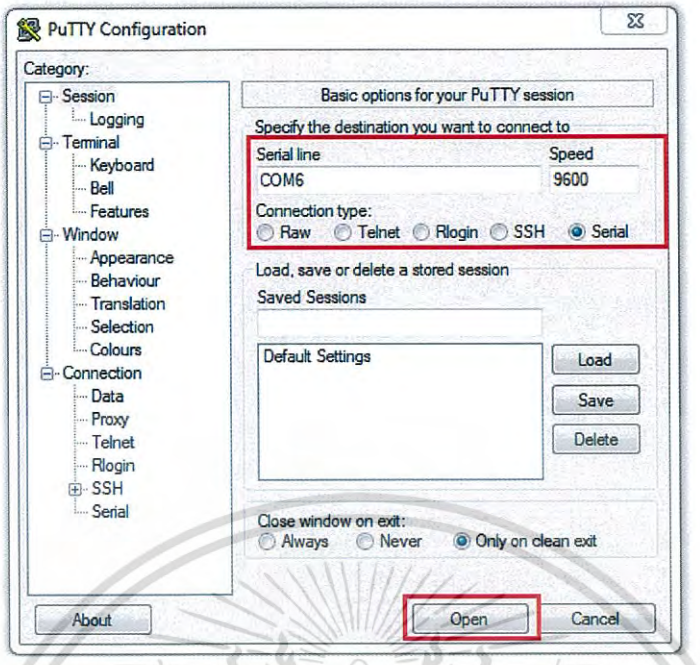
Ports(COM & LPT) จะแสดงว่า COM (Communications Port) ที่เท่าไร



รูปที่ ก.3 การหา Serial line

ที่หน้าจอ Putty ให้คลิกเลือก “Serial” ในส่วนของ Connection type และนำค่า Serial line ที่ได้จากภาพที่ ก.3 มาใส่ ส่วนค่าดีฟอลต์ Speed เท่ากับ 9600 หลังจากนั้นให้คลิกที่ปุ่ม “Open” เพื่อเปิดหน้าจอพรอมต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

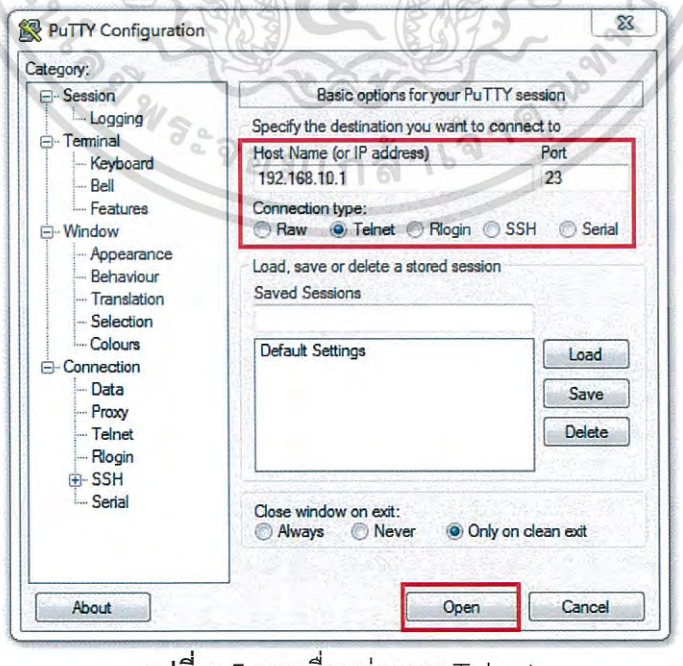


รูปที่ ก.4 การเชื่อมต่อแบบ Serial

2) **เชื่อมต่อประเภท Telnet**

เป็นการเชื่อมต่อกับคอมพิวเตอร์อีกเครื่องหนึ่งโดยการใช้อินเตอร์เน็ต ซึ่งจะต้องได้รับสิทธิ์ในการเชื่อมต่อกับเครื่องคอมพิวเตอร์เครื่องอื่นจากเจ้าของเครื่องหรือผู้ดูแล

ปัจจุบันการใช้โปรแกรมนี้เริ่มลดลง เพราะมีจุดบกพร่องเรื่องความปลอดภัย เนื่องจากอาจมีผู้ไม่หวังดี ใช้ Telnet แล้วแอบเก็บข้อมูลของเครื่องอื่นไปใช้ในทางที่ไม่ดีได้



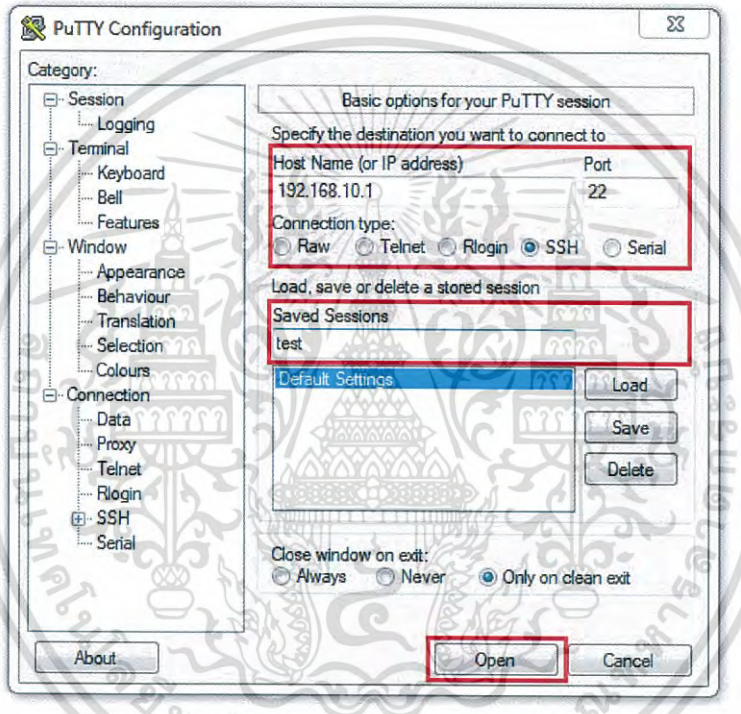
รูปที่ ก.5 การเชื่อมต่อแบบ Telnet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Host Name (or IP address) ให้ใส่ Host Name หรือ IP ที่ต้องการจะเชื่อมต่อ
- Connection Type ให้เลือกที่ “Telnet”
- กดปุ่ม “Open” เพื่อเปิดหน้าจอพรอมต์

3) เชื่อมต่อประเภท SSH

SSH หรือ Secure Shell ออกแบบมาเพื่อป้องกันการขโมยข้อมูลผ่าน Telnet โดยจะเข้ารหัสข้อมูลก่อนส่ง ทำให้ผู้ลักลอบไม่สามารถเห็นข้อมูลที่แท้จริง ปัจจุบันระบบปฏิบัติการ Unix หรือ Linux จะมีบริการ SSH เสมอ



รูปที่ ก.6 การเชื่อมต่อแบบ SSH

- Host Name (or IP address) คือ ให้ใส่ Host Name หรือ IP ที่เราจะเชื่อมต่อ
- Connection Type ให้เลือกที่ “SSH”
- Saved Sessions คือ ชื่อที่จะบันทึกการตั้งค่าไว้
- กดปุ่ม “Open” เพื่อเปิดหน้าจอพรอมต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
วิธีการใช้พื้นฐานโปรแกรม NetBeans

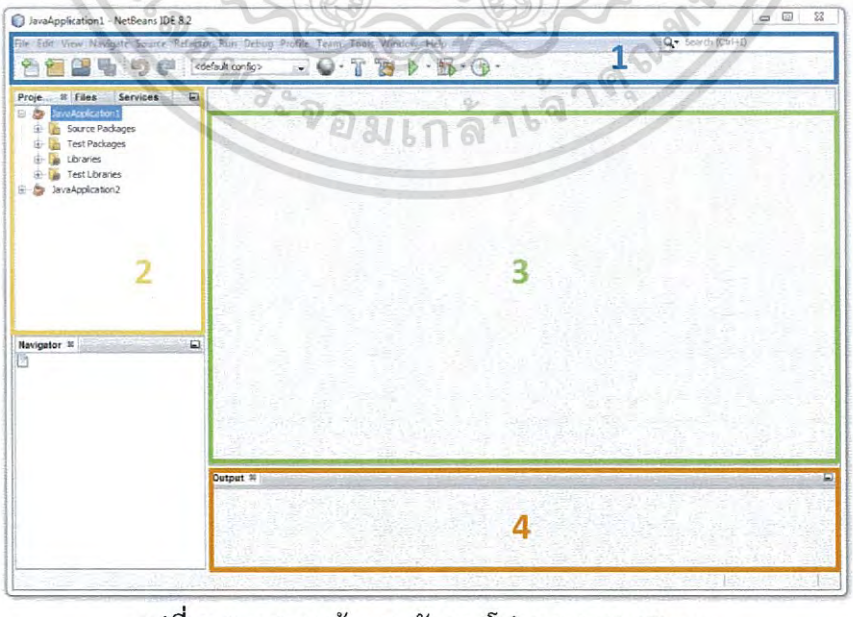
ในการพัฒนาโปรแกรมด้วยภาษา Java จะต้องทำการ download ตัว JDK หรือ Java Development Kit ก่อน หลังจากนั้นทำการ Set class path เพื่อให้สามารถ Compile และ Run Java ได้

รูปที่ ข.1 JDK

1) เปิดไฟล์ NetBeans.exe

รูปที่ ข.2 โปรแกรม NetBeans

หน้าจอหลักของ NetBeans จะแบ่งออกเป็น 4 ส่วนใหญ่ๆแสดงดังรูปที่ ข.3

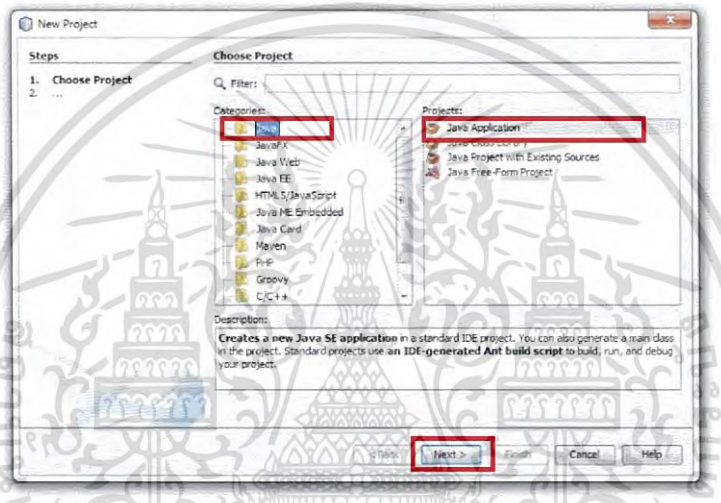


รูปที่ ข.3 แสดงหน้าจอหลักของโปรแกรม NetBeans

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

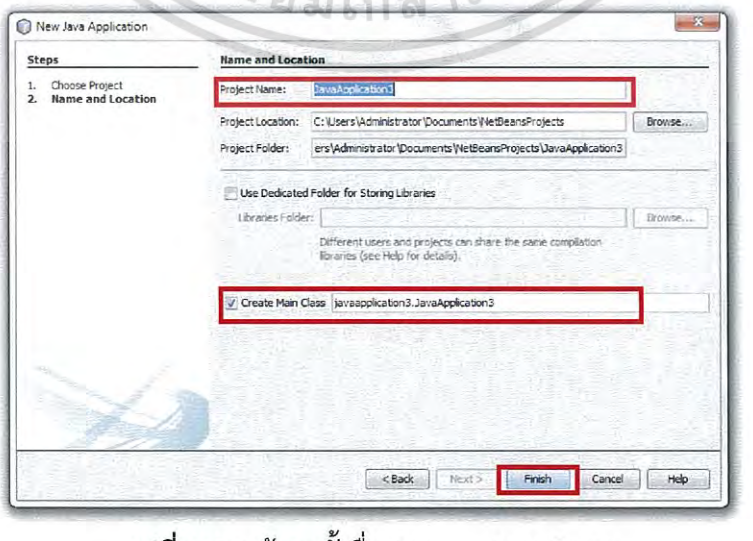
1) เป็นส่วนของแถบเมนู และ Tool ต่างๆ
2) เป็นส่วนที่แสดงและจัดการ Project ต่างๆ เหมือนเป็นการ Browse ดู Project หรือ File ต่างๆ
3) เป็นส่วนที่ใช้ในการเขียน Code
4) เป็นส่วนที่ใช้ในการแสดงผลการทำงานต่าง เมื่อทำการ Run โปรแกรม และยังมีส่วนของการ debug โปรแกรมด้วย

2) สร้าง New Project ใหม่ขึ้นมา โดยการเลือกที่ File >> New Project จะมีหน้าต่างขึ้นมาให้เราเลือก โดยให้เราทำการเลือก Java และ Java Application แล้วกด “Next”



รูปที่ ข. 4 หน้าจอสร้าง Project

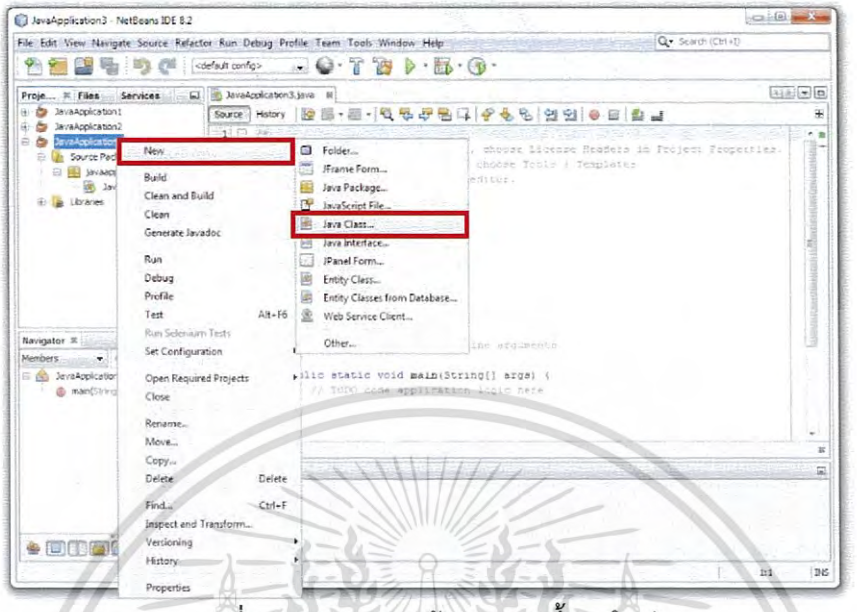
3) จะมีหน้าต่างให้กำหนดค่ารายละเอียดต่างๆของ Project โดยจะให้ใส่ Project Name และชื่อ Main Class เมื่อกำหนดค่ารายละเอียดต่างๆแล้ว กด “Finish” จะได้ Project ชื่อ JavaApplication3 และ File ชื่อ Javaapplication3.JavaApplication3.java



รูปที่ ข.5 หน้าจอตั้งชื่อ Project และ Class

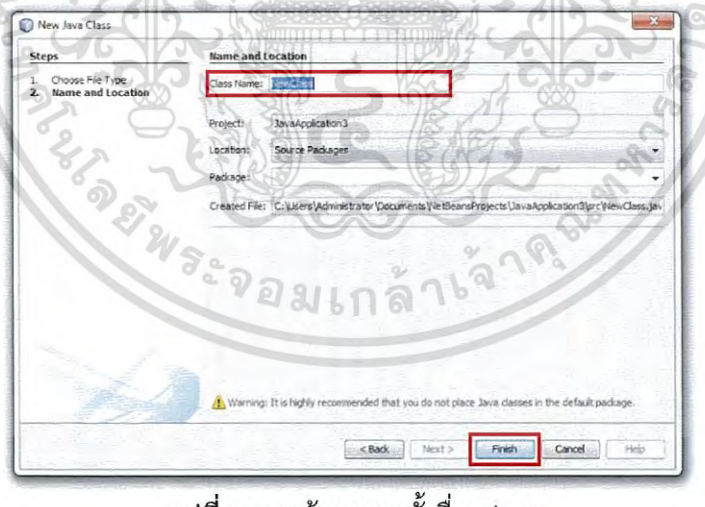
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) จากนั้นสร้าง class ขึ้นมาใหม่ โดยคลิกขวาที่ชื่อ Project แล้วเลือก New >> Java Class



รูปที่ ข.6 แสดงการสร้าง class ขึ้นมาใหม่

5) หลังจากนั้นโปรแกรมจะปรากฏหน้าต่างขึ้นมาให้กำหนดรายละเอียด โดยให้ตั้งชื่อ Class แล้วกด “Finish”

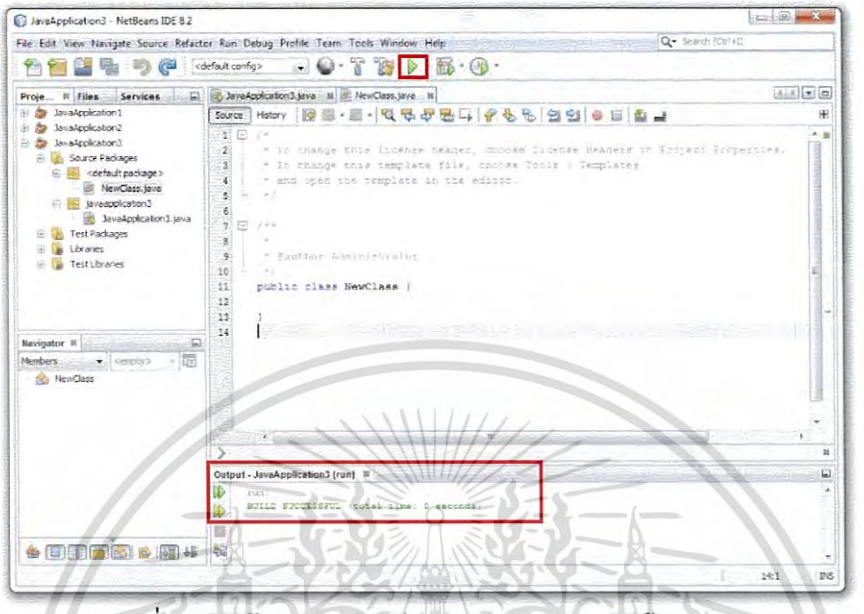


รูปที่ ข.7 หน้าจอการตั้งชื่อ class

ในการ code คำสั่งต่างๆ เราสามารถกด CTRL+SpaceBar เพื่อเรียกดูได้ว่าสามารถใช้ method หรือเรียก Attribute ใดมาใช้ได้บ้าง หรือบางที NetBeans ก็จะขึ้นส่วนนี้มาให้เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) หลังจากนั้นจะแสดงหน้าให้เริ่มเขียน Code เมื่อทำการแก้ไข Code เสร็จเรียบร้อย จะทำการ Run โปรแกรมได้โดยการกดปุ่มสีเขียวที่อยู่บนแถบเมนู



รูปที่ ข.8 หน้าจอสำหรับเขียน Code และ Run โปรแกรม

โดย NetBeans จะทำการ Save ให้อัตโนมัติทุกครั้งที่จะทำการ Run ถ้า NetBeans ทำการ compile แล้วไม่พบ error จะแสดงในส่วน Output ด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

Core Tcl Commands

Command	Description
after	Schedule a Tcl command for later execution.
append	Append arguments to a variable's value. No spaces added.
array	Query array state and search through elements.
binary	Convert between strings and binary data.
break	Premature loop exit.
catch	Trap errors.
cd	Change working directory.
clock	Get the time and format data strings.
close	Close an open I\O stream.
concat	Concatenate arguments with spaces between Splices lists.
console	Control the console used to enter commands interactively.
continue	Continue with next loop iteration.
error	Raise an error.
eof	Check for end of file.
eval	Concatenate arguments and evaluate them as a command.
exec	Fork and execute a UNIX program.
expr	Evaluate a math expression.
fblocked	Poll an I/O channel to see if data is ready.
fconfigure	Set and query I/O channel properties.
fcopy	Copy from one I/O channel to another.
file	Query the file system.
fileevent	Register callback for event-driven I/O.
flush	Flush output from an I/O stream's internal buffers.
for	Loop construct similar to C for statement.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command	Description
foreach	Loop construct over a list, or lists, of values.
format	Format a string similar to c sprintf.
gets	Read a line of input from an I/O stream.
glob	Expand a pattern to matching file names.
global	Declare global variables.
history	Command-line history control.
if	Conditional command. Allows else and elseif clause.
incr	Increment a variable by an integer amount.
info	Query the state of the Tcl intepreter.
interp	Create additional Tcl interpreters.
join	Concatenate list elements with a given separator string.
lappend	Add elements to the end of a list.
lindex	Fetch an element of a list.
linsert	Insert elements of a list.
list	Create a list out of the arguments.
llength	Return the number of elements in a list.
load	Load shared libraries that define Tcl commands.
lrange	Return a range of list elements.
ireplace	Replace elements of a list.
lsearch	Search for an element of a list that matches a pattern.
lsort	Sort a list.
namespace	Create and manipulate namespaces.
open	Open a file or process pipeline for I/O.
package	Provide or require code packages.
pid	Return the process ID.
proc	Define a Tcl procedure.
puts	Output a string to an I/O stream.
pwd	Return the current working directory.
read	Read blocks of character from an I/O stream.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command	Description
regexp	Regular expression matching.
regsub	Substitutions based on regular expressions.
rename	Change the name of a Tcl command.
return	Return a value from a procedure.
scan	Parses a string according to a format specification.
seek	Set the seek offset of an I/O stream.
set	Assign a value to a variable.
socket	Open a TCP/IP network connection.
source	Evaluate the Tcl commands in a file.
split	Chop a string up into list elements.
string	Operate on strings.
subset	Substitutions with out command evaluation.
switch	Multi-way branch.
tell	Return the current seek offset of an I/O stream.
time	Measure the execution time of a command.
trace	Monitor variable assignments.
unknown	Unknown command handler.
unset	Delete variable.
uplevel	Execute a command in different scope.
upvar	Reference a variable in a different scope.
variable	Declare namespace variables.
vwait	Wait for a variable to be modified.
while	A loop construct.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้