ส่วนติดต่อผู้ใช้ซอฟต์แวร์เอ็นเอส-ทูสำหรับโพรโทคอลทีซีพี

NS2- USER INTERFACE FOR TCP PROTOCOLS

กัญญวัฒน์ โชติพุทธิกุล

ปรัญญา หงส์ศิริวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2555

ส่วนติดต่อผู้ใช้ซอฟต์แวร์เอ็นเอส-ทูสำหรับโพรโทคอลทีซีพี

NS-2 USER INTERFACE FOR TCP PROTOCOLS

กัญญวัฒน์ โชติพุทธิกุล

ปรัญญา หงส์ศิริวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2555

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ส่วนติดต่อผู้ใช้ซอฟต์แวร์เอ็นเอส-ทูสำหรับโพรโทคอลทีซีพี

NS-2 USER INTERFACE FOR TCP PROTOCOLS

ผู้จัดทำ

1. นางสาวกัญญวัฒน์ โชติพุทธิกุล รหัสนักศึกษา 52010044
2. นางสาวปรัญญา หงส์ศิริวรรณ รหัสนักศึกษา 52010689

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. ศักดิ์ชัย ทิพย์จักษุรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนติดต่อผู้ใช้ซอฟต์แวร์เอ็นเอส-ทูสำหรับโพรโทคอลทีซีพี

นางสาว กัญญวัฒน์ โชติพุทธิกุล 52010044
นางสาว ปรัญญา หงส์ศิริวรรณ 52010689
ผศ.ดร. ศักดิ์ชัย ทิพย์จักษุรัตน์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2555

บทคัดย่อ

เน็ตเวิร์คซิมมูเลเตอร์ รุ่นที่ 2 (เอ็นเอส-ทู) เป็นซอฟต์แวร์จำลองการทำงานของเครือข่ายที่ถูกพัฒนาขึ้นจากภาษาซีพลัสพลัสบนระบบปฏิบัติการลินุกซ์ที่มีการใช้งานกันอย่างแพร่หลาย สามารถจำลองการทำงานของโพรโทคอลต่าง ๆ ไม่ว่าจะเป็นเครือข่ายแบบมีสายและเครือข่ายไร้สายตลอดจนเครือข่ายเคลื่อนที่ แต่เนื่องจากเอ็นเอส-ทูเป็นซอฟต์แวร์ที่ค่อนข้างมีความซับซ้อนในการใช้งาน การใช้งานเอ็นเอส-ทูต้องเขียนสคริปต์ด้วยภาษาทีซีแอลซึ่งผู้ใช้ต้องมีความรู้เกี่ยวกับภาษาทีซีแอล, โครงสร้างของไฟล์เทรซและสคริปต์อ๊อค ผลจากการจำลองจะได้กราฟเพียง 1 จุด ถ้าต้องการกราฟที่เป็นเส้นผู้ใช้ต้องทำการจำลองหลายครั้งทำให้ไม่สะดวกต่อการใช้งาน ดังนั้นเพื่อให้ผู้ใช้สามารถจำลองการทำงานเครือข่ายของโพรโทคอลทีซีพีได้ง่ายและสะดวกยิ่งขึ้นโดยที่ผู้ใช้ไม่จำเป็นต้องมีพื้นฐานดังกล่าวข้างต้น ในโครงงานนี้เราจึงนำเสนอการพัฒนาส่วนติดต่อผู้ใช้แบบกราฟฟิกสำหรับเอ็นเอส-ทูโดยใช้โพรโทคอลทีซีพีเป็นกรณีศึกษา เราเรียกซอฟต์แวร์ที่เราพัฒนาว่า ส่วนติดต่อผู้ใช้ซอฟต์แวร์เอ็นเอส-ทูสำหรับโพรโทคอลทีซีพี หรือเรียกสั้น ๆ ว่า “เอ็นเอส-ทูยูไอฟอร์ทีซีพี” ในการใช้งานผู้ใช้เพียงกำหนดจำนวนของโหนด โทโพโลยี ค่าพารามิเตอร์ต่าง ๆ โพรโทคอลที่ต้องการจำลองการทำงานและประสิทธิภาพที่ต้องการประเมิน หลังจากนั้นเอ็นเอส-ทูยูไอฟอร์ทีซีพีจะทำการจำลองการทำงานโดยอัตโนมัติและแสดงผลการจำลองการทำงานในรูปแบบของกราฟประสิทธิภาพต่าง ๆ และแอนิเมชันผ่านทางเน็ตเวิร์คแอนิเมเตอร์ นอกจากนี้เรายังสร้างส่วนแสดงเนื้อหาของโพรโทคอลทีซีพีและคู่มือการใช้งาน เพื่อให้ผู้ใช้สามารถศึกษาและทำความเข้าใจในแต่ละโพรโทคอลได้ผ่านซอฟต์แวร์ที่เราพัฒนาขึ้น และสามารถใช้งานซอฟต์แวร์ได้โดยง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NS-2 USER INTERFACE FOR TCP PROTOCOLS

Ms. Kanyawat	Chotiputhikul	52010044
Ms. Parunya	Hongsirivon	52010689
Asst. Prof. Dr. Sakchai	Thipchaksurat	Advisor

Academic Year 2012

ABSTRACT

Network Simulator version 2 (NS-2) is a simulation software for computer communication networks which is developed using C++ Language on Linux platform. NS-2 is an effective software and widely used in computer network communication researcher groups. However NS-2 is complicated. To use NS-2, users have to write TCL Script and understand structure of trace file and AWK Script as well. For the network simulation, users need to specify all parameters and environment and create related files or scripts to generate a single point of graph and the animation of one situation. Therefore NS-2 needs to run several times to create entire line graph and complete scene of animation.

In this project, we present the development of Graphic User Interface (GUI) for NS-2 which is more convenient to use for non-expert users. Our developed software is called "NS-2 USER INTERFACE FOR TCP PROTOCOLS" or "NS-2UI4TCP". To use NS-2UI4TCP, users only specify simulation parameters, TCP Protocols and performance metrics. Then it will automatically call NS-2 program to run simulation. The simulation results are shown in the terms of network performances which are represented in graphs and complete animations. Moreover, we provide the tutorial for TCP Protocols in wired networks in order to make users clearly understand the principle of TCP Protocols in wired networks. The user guide is also provided for understanding the steps of using NS-2UI4TCP.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานส่วนติดต่อผู้ใช้ซอฟต์แวร์เอ็นเอส-ทูสำหรับโพรโทคอลทีซีพี สามารถสำเร็จสมบูรณ์และลุล่วงไปได้ด้วยดีด้วยคำปรึกษาการพัฒนาโครงงานและการวางแผนการดำเนินการจาก ผศ.ดร. ศักดิ์ชัย ทิพย์จักษุรัตน์ ซึ่งเป็นอาจารย์ที่ปรึกษาของโครงงานนี้ที่ให้การดูแล ให้คำแนะนำ และมีการติดตามความคืบหน้าของโครงงานอย่างสม่ำเสมอ

ขอบพระคุณอาจารย์ทุกท่านในสาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้กรุณาช่วยวิจารณ์โครงงานเพื่อนำไปปรับปรุงให้ดียิ่งขึ้น

ขอขอบคุณรุ่นพี่ปริญญาโทและปริญญาเอกทุกท่าน รวมถึงเพื่อนๆที่คอยให้คำแนะนำการทำโครงงานในด้านต่างๆ

ขอบคุณห้องวิจัย Multi Media Laboratory ที่เป็นแหล่งสนับสนุนสถานที่และอุปกรณ์ในการพัฒนาโครงงานได้อย่างสะดวก

สุดท้ายนี้ขอกราบขอบพระคุณบิดามารดา ที่ให้กำลังใจและสนับสนุนในทุก ๆ ด้าน ด้วยคุณค่าและประโยชน์อันพึงมาจากโครงงานนี้ เราขอมอบแด่ผู้มีพระคุณทุกท่าน

นางสาว กัญญวัฒน์ โชติพุทธิกุล
นางสาว ปรัญญา หงส์ศิริวรรณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง IX
สารบัญรูป X

บทที่ 1 บทนำ 1
1.1 ความสำคัญและที่มาของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 1
1.4 วิธีการดำเนินการ 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
1.6 ส่วนประกอบของปริญญานิพนธ์ 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 5
2.1 แบบจำลอง OSI (OSI Model) 5
2.2 แบบจำลอง TCP/IP (TCP/IP Model) 10
2.2.1 การเชื่อมต่อของทีซีพี 13
2.2.1.1 การสร้างการเชื่อมต่อ (Connection Establishment) 13
2.2.1.2 การส่งข้อมูลระหว่างต้นทางและปลายทาง (Data Transfer) 14
2.2.1.3 การปิดการเชื่อมต่อ (Connection Termination) 16
2.3 โพรโทคอลทีซีพี 17
2.3.1 TCP Header 19
2.3.2 โพรโทคอลที่นำมาศึกษา 22
2.3.2.1 TCP Tahoe 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.2.2 TCP Reno 25
2.3.2.3 TCP New Reno 27
2.3.2.4 TCP Vegas 28
2.3.2.5 TCP Sack 30
2.4. การทำงานพื้นฐานของระบบการจำลองการทำงาน 34
2.4.1 พื้นฐานการจำลองการทำงานด้วยคอมพิวเตอร์ 34
2.4.2 การแบ่งประเภทของการจำลองการทำงานด้วยคอมพิวเตอร์ 36
2.4.3 การจำลองการทำงานของเหตุการณ์ที่มีความไม่ต่อเนื่องกัน (Discrete Event Simulation : DES) 38
2.5 Network Simulator Version 2 40
2.5.1 หลักการทำงานของเอ็นเอส-ทู 41
2.5.2 ภาษาที่ใช้ในเอ็นเอส-ทู 42
2.6 วิธีการใช้งานเอนเอส-ทูสำหรับโพรโทคอลทีซีพี 43
2.6.1 สคริปต์ทีซีแอล (TCL Script) 44
2.6.1.1 การติดตั้งค่าพารามิเตอร์ของการจำลองการทำงาน 44
2.6.1.2 การเริ่มต้นการทำงาน 44
2.6.1.3 การให้คำนิยามของโหนดต่างๆ 45
2.6.1.4 การให้คำนิยามของการเชื่อมต่อ 45
2.6.1.5 การให้คำนิยามของเอเจนต์ 46
2.6.1.6 การให้คำนิยามของแอพพลิเคชัน 46
2.6.1.7 การจบการทำงาน 46
2.6.2 Traffic File 47
2.6.3 Trace File 48
2.6.4 สคริปต์ AWK 50
2.6.5 NAM Trace File 53
2.6.6 Result File 62
2.7 Java Programming 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและพัฒนาซอฟต์แวร์ 66
3.1 แนวคิดในการพัฒนา 66
3.2 เครื่องมือที่ใช้ในการพัฒนา 71
3.2.1 ระบบปฏิบัติการ 71
3.2.2 ภาษาโปรแกรมที่ใช้ 71
3.2.3 เครื่องมือพัฒนาซอฟต์แวร์ (Integrated Development Environment) 71
3.2.4 JRE (Java Runtime Environment) 72
3.3 รายละเอียดของซอฟต์แวร์เชิงเทคนิค (Software Specification) 72
3.3.1 รายละเอียดส่วนนำเข้า (Input) 72
3.3.2 รายละเอียดส่วนนำออก (Output) 72
3.3.3 รายละเอียดฟังก์ชัน (Function) 73
3.4 โครงสร้างของซอฟต์แวร์ 73
3.4.1 ส่วนติดต่อผู้ใช้ (User Interface) 74
3.4.2 ส่วนการแปลงการจำลองเครือข่ายเป็นภาษาทีซีแอล (TCL Script Generator) 74
3.4.3 ส่วนการสร้างสคริปต์ AWK (AWK Script Generator) 74
3.4.4 ส่วนการสร้างกราฟ (Graph Generator) 74
3.4.5 ส่วนการจำลองเครือข่ายผ่านเนตเวิร์กแอนิเมเตอร์ (Network Animator) 75
3.5 แผนภาพ UML (Unified Modeling Language Diagram) 75
3.5.1 Use Case Diagram 75
3.5.2 คลาสไดอะแกรม (Class Diagram) 76
3.6 ขอบเขตและข้อจำกัดของซอฟต์แวร์ที่พัฒนา 78

บทที่ 4 การทดสอบซอฟต์แวร์ 80
4.1 การทดสอบส่วนการจำลองเครือข่าย (Network Simulation) 80
4.1.1 หน้าต่างหลักของซอฟต์แวร์ 80
4.2 การจำลองเครือข่ายในตัวอย่างสถานการจำลองต่างๆ 92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.2.1 สถานการณ์จำลองที่ 1 92

บทที่ 5 บทสรุป 95
5.1 ผลที่ได้จากการทำโครงงาน 94
5.2 ปัญหาอุปสรรคและแนวทางแก้ไข 95
5.3 แนวทางในการพัฒนาต่อ 96

บรรณานุกรม 97

ภาคผนวก ก 98
ก.1 การติดตั้งโปรแกรม NS-2.35 105
ก.2 การติดตั้งและการเรียกใช้งาน Netbeans IDE 7.0.1 107
ภาคผนวก ข 104
ข.1 รายละเอียดของคลาส UI 111
ข.2 รายละเอียดของคลาส Simulation 111
ข.3 รายละเอียดของคลาส Graph 111
ข.4 รายละเอียดของคลาส Parameters 113
ข.5 รายละเอียดของคลาส Nodes 115
ข.6 รายละเอียดของคลาส TCLScript 115
ข.7 รายละเอียดของคลาส AWKResult 115
ข.8 รายละเอียดของคลาส TrafficFile 116
ข.9 รายละเอียดของคลาส TraceFile 116
ข.10 รายละเอียดของคลาส AWKScript 116

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 1.1 วิธีดำเนินการในภาคเรียนที่ 1 10
ตารางที่ 1.2 วิธีดำเนินการในภาคเรียนที่ 2 11
ตารางที่ 2.1 TCP Header Flag 27
ตารางที่ 2.2 การเปรียบเทียบคุณสมบัติระหว่าง UDP และ TCP 28
ตารางที่ 2.3 Wire event 56
ตารางที่ 2.4 TCP Trace 57
ตารางที่ 2.5 NAM Trace File 60
ตารางที่ 2.6 Node Trace 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

รูปที่ 2.1 ชั้นต่าง ๆ ของแบบจำลอง OSI Layer 6
รูปที่ 2.2 ชั้นต่างๆของแบบจำลอง OSI และแบบจำลอง TCP/IP 10
รูปที่ 2.3 การสถาปนาการเชื่อมต่อ 13
รูปที่ 2.4 การส่งข้อมูลระหว่างต้นทางและปลายทาง 14
รูปที่ 2.5 การปิดการเชื่อมต่อ 16
รูปที่ 2.6 Half-Close 17
รูปที่ 2.6 TCP Segment 19
รูปที่ 2.7 กราฟแสดงการทำงานของอัลกอริทึม Slow Start และอัลกอริทึมการหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่าย 24
รูปที่ 2.8 กราฟแสดงการทำงานของ Fast Retransmit และ Fast Recovery 27
รูปที่ 2.9 กราฟแสดงการทำงานของ TCP Vegas 30
รูปที่ 2.10 ตัวอย่างของการกำหนดเงื่อนไข Sack - Permitted 32
รูปที่ 2.11 ตัวอย่างโครงสร้างของเงื่อนไข Sack 32
รูปที่ 2.12 การแบ่งประเภทของการจำลองการทำงานด้วยคอมพิวเตอร์ 36
รูปที่ 2.13 ลักษณะการทำงานแบบ Event stepped 37
รูปที่ 2.14 ลักษณะการทำงานแบบ Time stepped 38
รูปที่ 2.15 แผนผังการทำงานของการจำลองการทำงานของเหตุการณ์ที่มีความไม่ต่อเนื่องกัน 39
รูปที่ 2.16 ผังการทำงานโดยรวมของเอ็นเอส-ทู 41
รูปที่ 2.17 ขั้นตอนการจำลองการทำงานของเอ็นเอส-ทู 43
รูปที่ 3.1 ขั้นตอนจำลองการทำงานของเอ็นเอส-ทูและส่วนที่เกี่ยวข้อง 68
รูปที่ 3.2 ขั้นตอนการจำลองการทำงานของเอ็นเอสทูยูไอฟอร์ทีซีพี 70
รูปที่ 3.3 โครงสร้างและการทำงานของเอ็นเอส-ทูยูไอฟอร์ทีซีพี 73
รูปที่ 3.4 แผนภาพ Use Case ของเอ็นเอส-ทูยูไอฟอร์ทีซีพี 76
รูปที่ 3.5 แผนภาพคลาสไดอะแกรมของเอ็นเอส-ทูยูไอฟอร์ทีซีพี 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รูปที่ 4.1 หน้าต่างหลักของซอฟต์แวร์....80
รูปที่ 4.2 ไดอะล็อกตัวเลือกวิธีการสร้างเครือข่ายจำลอง....80
รูปที่ 4.3 หน้าเริ่มต้นของ Automatic Mode....81
รูปที่ 4.4 ส่วนของ Parameters Configuration....81
รูปที่ 4.5 ตั้งค่าพารามิเตอร์ของการจำลองเครือข่าย Link & Agent Configuration....82
รูปที่ 4.6 ตั้งค่าการแสดงผลกราฟในส่วนของ Performance metric Graph versus Number of Connections....83
รูปที่ 4.7 ตั้งค่าการแสดงผลกราฟในส่วนของ Performance metric....84
รูปที่ 4.8 การแสดงผลกราฟในส่วนของโทโพโลยี....85
รูปที่ 4.9 ผลลัพธ์จากการจำลองการทำงานในรูปแบบกราฟ....86
รูปที่ 4.10 ตัวอย่างผลลัพธ์ในรูปของแอนิเมชันที่ค่าคอนเน็คชัน 15 คอนเน็คชัน.... 87
รูปที่ 4.11 การเลือกโหมดที่ผู้ใช้กำหนดเองในการสร้างการจำลองเครือข่ายแบบมีสาย....87
รูปที่ 4.12 ตัวอย่างในการสร้างการจำลองเครือข่ายแบบมีสาย....88
รูปที่ 4.13 ตัวอย่างในการสร้างการจำลองเครือข่ายแบบมีสาย....89
รูปที่ 4.14 การตั้งค่า Link Setting....89
รูปที่ 4.15 การตั้งค่าพารามิเตอร์....90
รูปที่ 4.16 ผลลัพธ์จากการจำลองการทำงานในรูปแบบกราฟ....91
รูปที่ 4.17 ตัวอย่างผลลัพธ์ในรูปของแอนิเมชัน....92
รูปที่ 4.18 ผลลัพธ์จากการจำลองการทำงานในรูปแบบกราฟของสถานการณ์จำลองที่ 1....93
รูปที่ 4.19 ตัวอย่างผลลัพธ์ในรูปของอนิเมชันที่ค่าคอนเน็คชัน 15 คอนเน็คชัน....93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

เอ็นเอส-ทู (Network Simulator Version 2) เป็นซอฟต์แวร์จำลองการทำงานของเครือข่ายที่มีประสิทธิภาพ สามารถจำลองการทำงานของโพรโทคอลต่าง ๆ ไม่ว่าจะเป็นเครือข่ายแบบมีสายและเครือข่ายไร้สายตลอดจนเครือข่ายเคลื่อนที่ นอกจากนี้เอ็นเอส-ทูยังสามารถดาวน์โหลดมาใช้งานได้โดยไม่เสียค่าใช้จ่าย แต่เนื่องจากเอ็นเอส-ทูเป็นซอฟต์แวร์ที่มีความซับซ้อนในการใช้งาน โดยการใช้งานเอ็นเอส-ทูต้องเขียนสคริปต์ภาษาทีซีแอล ซึ่งผู้ใช้ต้องมีความรู้เกี่ยวกับภาษาทีซีแอลด้วยและโพรโทคอลทีซีพีมีหลายอัลกอริทึมและมีความซับซ้อน ดังนั้นเพื่อให้ผู้ใช้สามารถศึกษาจำลองการทำงานของเครือข่ายได้ง่ายโดยที่ผู้ใช้ไม่จำเป็นต้องมีพื้นฐานมาก่อนและผู้ใช้สามารถเข้าใจการทำงานของโพรโทคอลทีซีพี ซึ่งผู้ใช้สามารถเข้าใจและสั่งงานการรันการจำลองการทำงานได้ โครงงานนี้ได้นำกรณีศึกษาการพัฒนาส่วนติดต่อผู้ใช้ซอฟต์แวร์เอ็นเอส-ทูสำหรับโพรโทคอลเอโอดีวีและดีเอ็สอาร์ในเครือข่ายเคลื่อนที่แอดฮอคมาพัฒนาต่อ โดยผู้จัดทำได้พัฒนาส่วนติดต่อผู้ใช้สำหรับโพรโทคอลทีซีพีในเครือข่ายมีสาย ผู้ใช้สามารถดูผลการทำงานของการจำลองระบบเครือข่ายสื่อสารในรูปแบบของกราฟต่าง ๆ และสามารถดูการจราจรของเครือข่ายจำลองผ่านทางเน็ตเวิร์คแอนิเมเตอร์

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อให้ผู้ใช้งานสามารถใช้เอ็นเอส-ทูได้ง่ายและสะดวก โดยจะใช้งานผ่านซอฟต์แวร์ส่วนติดต่อผู้ใช้แบบกราฟิก (Graphic User Interface) ที่ถูกพัฒนาขึ้น
2. เพื่อให้ผู้ใช้งานสามารถดูผลประสิทธิภาพของโพรโทคอลทีซีพีในเครือข่ายมีสาย ในรูปแบบของกราฟต่าง ๆ ได้ และสามารถดูการจราจรของเครือข่ายจำลองในรูปแบบแอนิเมเตอร์
3. ศึกษาการทำงานของโพรโทคอลทีซีพีที่มีความซับซ้อนและเข้าใจได้ยาก

1.3 ขอบเขตของโครงงาน

1. ซอฟต์แวร์สามารถทำงานโดยเป็นเดสก์ท็อปแอพพลิเคชัน (Desktop Application)
2. ใช้ Netbeans IDE 6.9 เป็นเครื่องมือในการพัฒนาซอฟต์แวร์ด้วยภาษาจาวา (Java)
3. ซอฟต์แวร์สามารถทำงานได้บนระบบปฏิบัติการลินุกซ์ (Linux) โดยจะทำการทดสอบบนระบบปฏิบัติการอูบุนตู (Ubuntu) 10.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

4. ซอฟต์แวร์สามารถจำลองเฉพาะเครือข่ายมีสายโพรโทคอลทีซีพี โดยเลือกใช้ได้ 5 โพรโทคอล ได้แก่ TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas และ TCP Sack
5. ซอฟต์แวร์สามารถแสดงผลประสิทธิภาพการทำงานในรูปแบบกราฟได้แก่ Packet Delivery Fraction (PDF), Packets Loss, Delay และ Throughput และแสดงผลการจราจรของเครือข่ายจำลองในรูปแบบแอนิเมเตอร์
6. เนื่องจากฟังก์ชันในเอ็นเอส-ทูมีเป็นจำนวนมาก ผู้จัดทำจึงเลือกพัฒนาเฉพาะฟังก์ชันที่ใช้งานเป็นส่วนใหญ่

1.4 วิธีการดำเนินการ

เพื่อให้โครงงานนี้ดำเนินไปอย่างราบรื่น กระชับ และเสร็จสิ้นตามกำหนด จึงได้วางแผนการทำงานล่วงหน้าโดยกำหนดช่วงวันและหัวข้อของทั้งสองภาคเรียนในรูปแบบของ Gantt Chart ดังตารางที่ 1.1 และตารางที่ 1.2 ตามลำดับ

ตารางที่ 1.1 วิธีดำเนินการในภาคเรียนที่ 1

ลำดับที่	รายการ	มิ.ย. 55				ก.ค. 55				ส.ค. 55			
		1	2	3	4	1	2	3	4	1	2	3	4
1	ศึกษาการทำงานเอ็นเอส-ทู												
2	ศึกษาการทำงานของโพรโทคอลทีซีพี												
3	ศึกษาภาษาจาวาและสคริปต์ทีซีแอล												
4	ออกแบบส่วนติดต่อผู้ใช้ด้วยภาษาจาวา												

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.2 วิธีดำเนินการในภาคเรียนที่ 2

ลำดับที่	รายการ	พ.ย. 55				ธ.ค. 55				ม.ค. 56			
		1	2	3	4	1	2	3	4	1	2	3	4
1	นำส่วนติดต่อผู้ใช้มา												
	รวมกับทีซีแอลและ ทดสอบผล												
2	สร้างการจราจรของ เครือข่ายจำลองและ นำไปรวมกับทีซีแอล												
3	สร้างไฟล์ AWK และ นำไปทดสอบผลกราฟ												
4	สร้างแอนิเมเตอร์และ ทดสอบผล												
5	ทดสอบการทำงาน ของซอฟต์แวร์และ แก้ไข												

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้ใช้งานสามารถใช้งานเอ็นเอส-ทูได้ง่ายและสะดวกขึ้น ผ่านส่วนติดต่อผู้ใช้แบบกราฟิก (Graphic User Interface) ซึ่งเป็นซอฟต์แวร์ที่ทางผู้จัดทำได้พัฒนาขึ้นมา
2. ผู้ใช้งานสามารถดูผลการทำงานและประสิทธิภาพของการจำลองการทำงานโพรโทคอลทีซีพีบนเครือข่ายมีสายในรูปแบบของกราฟต่าง ๆ ได้
3. ผู้ใช้งานสามารถดูผลการจราจรของเครือข่ายจำลองในรูปแบบแอนิเมเตอร์
4. สามารถเป็นสื่อการสอนเรื่องโพรโทคอลทีซีพีได้

1.6 ส่วนประกอบของปริญญานิพนธ์

เนื้อหาของปริญญานิพนธ์ฉบับนี้ประกอบไปด้วย 5 บทด้วยกันก็คือ บทนำ ทฤษฎีที่เกี่ยวข้อง การออกแบบและพัฒนาซอฟต์แวร์ การทดลองและผลการทดลอง และบทสรุป โดยสามารถจำแนกรายละเอียดได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ กล่าวถึง ความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงานวิธีการดำเนินการ และ ประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึง แบบจำลอง OSI แบบจำลอง TCP/IP โพรโทคอล TCP การทำงานพื้นฐานของระบบจำลองเครือข่าย การทำงานของเอ็นเอส-ทู วิธีการใช้งานเอ็นเอส-ทู สำหรับโพรโทคอลทีซีพี และจาวาโปรแกรมมิ่ง

บทที่ 3 การออกแบบและพัฒนาซอฟต์แวร์ กล่าวถึง เครื่องมือที่ใช้ในการพัฒนา ภาพรวมของโปรแกรม กระบวนการทำงานของโปรแกรม

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึง วิธีการตั้งค่าของซอฟต์แวร์ในส่วนต่าง ๆ และผลลัพธ์จากการจำลองเครือข่ายในสถานการณ์ตัวอย่าง

บทที่ 5 บทสรุป กล่าวถึง ผลได้จากการทำโครงงาน ปัญหาอุปสรรคและแนวทางแก้ไข และแนวทางการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในการพัฒนาซอฟต์แวร์นั้นเราจำเป็นต้องเข้าใจทฤษฎีต่าง ๆ เป็นอย่างดีเพื่อที่จะสามารถจำลองการทำงานของเครือข่ายได้อย่างถูกต้อง ซึ่งทฤษฎีที่เกี่ยวข้องกับส่วนติดต่อผู้ใช้ซอฟต์แวร์เอ็นเอส-ทูสำหรับโพรโทคอลทีซีพี ได้แก่ แบบจำลอง OSI, แบบจำลอง TCP/IP, โพรโทคอลทีซีพีที่นำมาเป็นกรณีศึกษาทั้ง 5 โพรโทคอล, การทำงานพื้นฐานของระบบจำลองเครือข่าย, การทำงานของเอ็นเอส-ทู, วิธีการใช้งานเอ็นเอส-ทูสำหรับโพรโทคอลทีซีพี และจาวาโปรแกรมมิ่ง

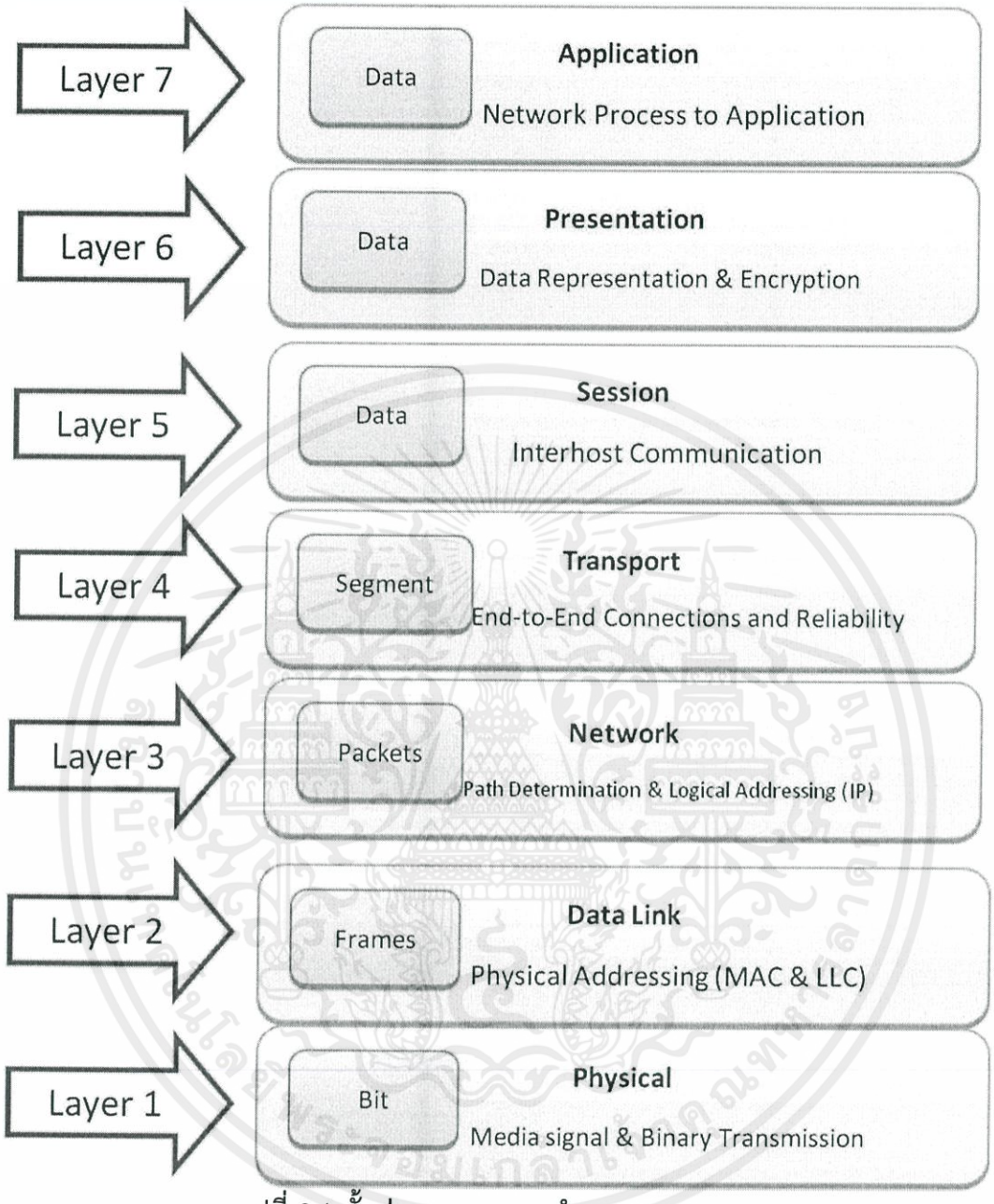
2.1 แบบจำลอง OSI (OSI Model)

การที่คอมพิวเตอร์เครื่องหนึ่งจะส่งข้อมูลไปยังคอมพิวเตอร์อีกเครื่องหนึ่งได้ ต้องอาศัยอุปกรณ์ต่าง ๆ ที่มีหน้าที่การทำงานต่างกันและเชื่อมต่อเป็นเครือข่ายเข้าด้วยกัน ปัญหาที่เกิดขึ้นคือการเชื่อมต่อมีความแตกต่าง ระหว่างระบบและอุปกรณ์ที่ผลิตจากบริษัทผู้ผลิตต่างกัน ทำให้เกิดความเข้ากันไม่ได้ของระบบจากบริษัทผู้ผลิตต่าง ๆ ทั่วโลก การสร้างระบบเครือข่ายเป็นเรื่องที่ทำได้ยากเนื่องจากขาดมาตรฐานกลางที่จำเป็นในการเชื่อมต่อ

หน่วยงานกำหนดมาตรฐานสากล คือ International Standards Organization (ISO) จึงทำการกำหนดโครงสร้างทั้งหมดที่จำเป็นต้องใช้ในการสื่อสารข้อมูล โดยเสนอและพัฒนาแบบจำลอง OSI (Open System Interconnection Reference Model) ที่บรรยายถึงโครงสร้างของสถาปัตยกรรมเครือข่ายในอุดมคติ ซึ่งระบบเครือข่ายที่เป็นไปตามสถาปัตยกรรมนี้จะเป็นระบบเครือข่ายแบบเปิด เพื่อให้ผู้ผลิตต่าง ๆ สามารถแยกผลิตอุปกรณ์ทางเครือข่ายในส่วนที่ตัวเองเชี่ยวชาญ แต่สามารถติดต่อกันได้และนำไปใช้งานร่วมกันได้ ระบบเครือข่ายคอมพิวเตอร์สมัยใหม่ถูกออกแบบให้มีโครงสร้างที่แน่นอน และเพื่อเป็นการลดความซับซ้อน มีการแยกการทำงานออกเป็นชั้น ๆ (Layer) โดยกำหนดหน้าที่ในแต่ละชั้นไว้อย่างชัดเจน

แบบจำลอง OSI มีการแบ่งการทำงานของระบบเครือข่ายออกเป็น 7 ชั้น ดังแสดงในรูปที่ 2.1 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ชั้นต่าง ๆ ของแบบจำลอง OSI Layer

1. ชั้นฟิสิคัล (Physical Layer)

ชั้นฟิสิคัลนี้เป็นการกล่าวถึงข้อกำหนดมาตรฐานคุณสมบัติทางกายภาพของฮาร์ดแวร์ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์ทั้ง 2 ระบบ สัญญาณทางไฟฟ้าและการเชื่อมต่อต่างๆของสายเคเบิล ตัวเชื่อมต่อต่าง ๆ เช่น สายที่ใช้รับส่งข้อมูล มาตรฐานของข้อต่อหรือปลั๊กที่ใช้ ขนาดความเข้มของแหล่งจ่ายไฟ ความเร็วในการรับส่งสัญญาณ มาตรฐานสัญญาณที่ใช้รับส่งข้อมูล เป็นต้น ชั้นฟิสิคัลนี้จะมองเห็นข้อมูลเป็นการรับ - ส่งทีละบิตเรียงต่อกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยไม่มีการพิจารณาเรื่องความหมายของข้อมูล การรับส่งจะอยู่ในรูป 0 หรือ 1 หากการรับส่งข้อมูลมีปัญหาเนื่องจากฮาร์ดแวร์จะตรวจสอบและแจ้งข้อผิดพลาดนั้นให้ชั้นอื่น ๆ ที่อยู่เหนือขึ้นไปทราบ

หน่วยของชั้นฟิสิคัลคือบิต (Bit) โพรโทคอลในชั้นนี้ ได้แก่ CAT5, CAT6, RJ-45, EIA/TIA-232 และ V.35Cable เป็นต้น

2. ชั้นดาต้าลิงค์ (Data Link Layer)

ชั้นดาต้าลิงค์ทำหน้าที่รับผิดชอบในการส่งข้อมูลบนเครือข่ายแต่ละประเภท เช่น Ethernet, Token Ring, FDDI หรือบน Wire Area Network (WAN) ต่าง ๆ โดยตรวจสอบความถูกต้องของข้อมูลแล้วส่งข้อมูลที่ปราศจากข้อผิดพลาดให้กับชั้นเน็ตเวิร์ก โดยปกติผู้ส่งข้อมูลจะแบ่งข้อมูลที่มีความยาวมากออกเป็นกลุ่มข้อมูลย่อย ๆ แต่ละส่วนย่อยเรียกว่า เฟรมข้อมูล (Data Frame) ซึ่งจะมีขนาดคงที่ประมาณสองหรือสามร้อยไบต์ (Byte) ชุดของเฟรมข้อมูลสำหรับข้อมูลที่ต้องการส่งไปให้ผู้รับจะถูกส่งไปทีละเฟรมจนครบ ทางฝ่ายผู้รับจะตอบโดยการส่งเฟรมตอบรับการร้องขอ (Acknowledgement Frame) ไปยังผู้ส่งเพื่อเป็นการบอกให้ผู้ส่งทราบว่าได้รับข้อมูลครบแล้ว กระบวนการรับ-ส่งข้อมูลนี้ก็จะเสร็จสมบูรณ์ การรับ - ส่งข้อมูลในชั้นฟิสิคัลจะไม่รับรู้ในโครงสร้างข้อมูล กล่าวคือจะมองเห็นข้อมูลเป็นบิต 0 หรือ 1 กลุ่มหรือเป็นชุดข้อมูลที่เรียงตามลำดับซึ่งเรียกว่า กระแสบิต (Bit Stream) จึงเป็นหน้าที่ของชั้นดาต้าลิงค์ที่ต้องทำการตรวจสอบความถูกต้องซึ่งทำได้โดยการเพิ่มข้อมูลสำหรับการตรวจสอบติดไว้กับข้อมูลทุกเฟรม เช่น การเพิ่มข้อมูลส่วนหัวและส่วนหางเข้าไปกับทุกเฟรมซึ่งจะใช้เป็นตัวกำหนดขอบเขตของเฟรมข้อมูลอีกด้วย การดูแลเรื่องการห่อหุ้มข้อมูลจากชั้นของ OSI ที่อยู่ถัดจากชั้นดาต้าลิงค์ ได้แก่ ชั้นเน็ตเวิร์ก, ชั้นทรานสปอร์ต, ชั้นเซสชัน, ชั้นพรีเซนเทชัน และชั้นแอพพลิเคชัน ตัวอย่างเช่น แพ็คเก็ต IP ที่เก็บภายในเฟรมข้อมูลและส่งจากต้นทางไปยังอุปกรณ์ตัวถัดไป

ชั้นดาต้าลิงค์นี้จะเข้าใจถึงกลไกและอัลกอริทึมรวมทั้งรูปแบบของเฟรมข้อมูลที่ต้องใช้ในเครือข่ายประเภทต่าง ๆ โดยในเครือข่ายบนชั้นอีเทอร์เน็ต (Ethernet Layer) จะมีการระบุหมายเลขที่อยู่ของอุปกรณ์คือหมายเลขประจำตัวเครื่องที่เรียกว่า MAC Address (Media Access Control Address) ซึ่งเป็นหมายเลขที่ฝังมากับอุปกรณ์นั้น ๆ ไม่สามารถเปลี่ยนแปลงได้โดย MAC Address จะเป็นตัวเลขขนาด 6 ไบต์ โดยหมายเลข 3 ไบต์แรกจะได้รับการจัดสรรโดยองค์กรกลางที่ชื่อว่า สถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ (Institute of Electrical and Electronics Engineers : IEEE) ให้กับผู้ผลิตแต่ละราย ส่วนหมายเลข 3 ไบต์หลังทางผู้ผลิตจะเป็นผู้กำหนดเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยของชั้นดาต้าลิงค์นี้คือเฟรม (Frame) โพรโทคอลในชั้นนี้ ได้แก่ Ethernet, Token Ring, IEEE 802.3/202.2, Frame Relay, FDDI, HDLC และ ATM เป็นต้น

3. ชั้นเน็ตเวิร์ก (Network Layer)

ชั้นเน็ตเวิร์กมีหน้าที่หลักคือการส่งแพ็คเก็ตจากเครื่องต้นทางให้ไปถึงปลายทางโดยค้นหาเส้นทางที่มีประสิทธิภาพดีที่สุด ซึ่งมีการกำหนดหาเส้นทางเดินของข้อมูลจากโหนดผู้ส่ง (Node) ไปตามโหนดต่าง ๆ จนถึงโหนดผู้รับข้อมูลในที่สุด โฮสบางกลุ่มจะกำหนดเส้นทางเดินข้อมูลโดยศึกษาระบบเครือข่ายแล้วสร้างตารางเส้นทางเดินข้อมูลแบบถาวร โฮสบางกลุ่มจะกำหนดเส้นทางเดินข้อมูลในตอนเริ่มต้นของการสื่อสาร ดังนั้นในการสื่อสารครั้งต่อไป อาจจะเปลี่ยนไปใช้เส้นทางอื่นได้ โฮสในกลุ่มที่มีวิธีการซับซ้อนมากจะกำหนดเส้นทางเดินข้อมูลคล้ายกับสภาวะการจราจรในชั่วโมงเร่งด่วน โฮสในกลุ่มนี้จะปรับเส้นทางของการเดินข้อมูลของแต่ละแพ็คเก็ตให้เหมาะสมกับสภาวะของระบบเครือข่ายตลอดเวลา ชั้นเน็ตเวิร์กนี้จะกำหนดให้มีการตั้งหมายเลขลอจิคัล (Logical Address) ขึ้นมาเพื่อใช้ในการระบุตัวตน เช่น โพรโทคอล IP หมายเลขลอจิคัลคือหมายเลข IP นั่นเอง

อุปกรณ์ฮาร์ดแวร์ที่ทำงานอยู่บนชั้นเน็ตเวิร์กนี้คือเราท์เตอร์ (Router) และโพรโทคอลที่ทำงานในชั้นเน็ตเวิร์กนี้จะไม่ทราบว่าแพ็คเก็ตไปถึงเครื่องปลายทางหรือไม่ เนื่องจากหน้าที่การยืนยันข้อมูลได้ไปถึงปลายทางจริง ๆ คือหน้าที่ของชั้นทรานสปอร์ต หน่วยของชั้นเน็ตเวิร์กนี้คือแพ็คเก็ต (Packet) และโพรโทคอลในชั้นนี้ ได้แก่ IP, IPX และ Apple talk เป็นต้น

4. ชั้นทรานสปอร์ต (Transport Layer)

ชั้นทรานสปอร์ตมีหน้าที่หลักคือการแบ่งข้อมูลชั้นของ OSI ที่อยู่ถัดจากชั้นทรานสปอร์ต ได้แก่ ชั้นเซสชัน, ชั้นพรีเซนเทชัน และชั้นแอพพลิเคชัน ให้พอเหมาะกับการจัดส่งไปในชั้นที่ต่ำกว่าชั้นทรานสปอร์ตของแบบจำลอง OSI ซึ่งได้แก่ ชั้นเน็ตเวิร์ก ชั้นดาต้าลิงค์ และชั้นฟิสิคัล ซึ่งการแบ่งส่วนข้อมูลนี้เรียกว่า เซกเมนต์เทชัน (Segmentation) ประกอบกับข้อมูลต่าง ๆ ที่ได้รับมาจากชั้นที่ต่ำกว่าชั้นทรานสปอร์ตของแบบจำลอง OSI ให้บริการตรวจสอบและแก้ไขปัญหาเมื่อเกิดข้อผิดพลาดขึ้นระหว่างการส่ง (Error Recovery) โดยทำหน้าที่ยืนยันว่าข้อมูลได้ถูกส่งไปถึงยังเครื่องปลายทางและได้รับข้อมูลถูกต้องเรียบร้อยแล้ว นอกจากนี้ชั้นทรานสปอร์ตจะทำหน้าที่ดูแลจัดการเรื่องของความผิดพลาดที่เกิดขึ้นจากการสื่อสาร ซึ่งการตรวจสอบความผิดพลาดนั้นจะพิจารณาจากข้อมูลส่วนที่เรียกว่า เช็คซัม (Checksum)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยของข้อมูลที่ถูกแบ่งแล้วนี้เรียกว่าเซกเมนต์ (Segment) โพรโทคอลในชั้นนี้ ได้แก่ TCP, UDP และ SPX เป็นต้น

5. ชั้นเซสชัน (Session Layer)

ชั้นเซสชันทำหน้าที่ควบคุมการสื่อสารจากต้นทางไปยังปลายทางแบบเอ็นทูเอ็น (End-to-End) และคอยควบคุมช่องทางการสื่อสารในกรณีที่มีหลาย ๆ โพรเซส (Process) ต้องการรับส่งข้อมูลพร้อม ๆ กันบนเครื่องเดียวกันคือทำงานคล้าย ๆ เป็นหน้าต่างคอยสลับเปิดให้ข้อมูลเข้าออกตามหมายเลขพอร์ต (Port) ที่กำหนดและยังให้อินเตอร์เฟซ (Interface) สำหรับชั้นแอพพลิเคชันในการควบคุมขั้นตอนการทำงานของโพรโทคอลในระดับชั้นทรานสปอร์ตและชั้นเน็ตเวิร์ก เช่น Socket ในระบบปฏิบัติการลินุกซ์ (Linux) หรือ Windows Socket ในระบบปฏิบัติการแบบวินโดว์ (Windows) ซึ่งได้ให้ส่วนต่อประสานกับผู้ใช้แบบแอพพลิเคชันโปรแกรมมิ่ง (Application Programming Interface : API) แก่ผู้พัฒนาซอฟต์แวร์ในระดับบนสำหรับการเขียนโปรแกรมเพื่อควบคุมการทำงานของโพรโทคอล TCP/IP ในชั้นที่ต่ำกว่าของแบบจำลอง OSI และทำหน้าที่ควบคุมจังหวะในการรับส่งข้อมูลของทั้ง 2 ด้านให้มีความสอดคล้องกัน

นอกจากนี้ยังกำหนดวิธีที่ใช้รับส่งข้อมูล เช่น อาจจะเป็นในลักษณะสลับกันส่ง (Half Duplex) หรือรับส่งไปพร้อมกันทั้ง 2 ด้าน (Full Duplex) ข้อมูลที่รับส่งกันในชั้นเซสชันนี้จะอยู่ในรูปของไดอะล็อก (Dialog) หรือประโยคข้อมูลที่สนทนาโต้ตอบกันระหว่างผู้รับและผู้ส่งข้อมูลไม่ได้มองเป็นแบบในชั้นพรีเซนเทชัน เช่น เมื่อผู้รับได้รับข้อมูลส่วนแรกจากผู้ส่งก็จะตอบกลับไปให้ผู้ส่งรู้ว่าได้รับข้อมูลส่วนแรกเรียบร้อยแล้วและพร้อมที่จะรับข้อมูลส่วนต่อไป คล้ายกับเป็นการสนทนาตอบโต้กันระหว่างผู้รับกับผู้ส่งนั่นเอง โพรโทคอลในชั้นนี้ ได้แก่ RPC, SQL, Netbios, Windows socket และ NFS เป็นต้น

6. ชั้นพรีเซนเทชัน (Presentation Layer)

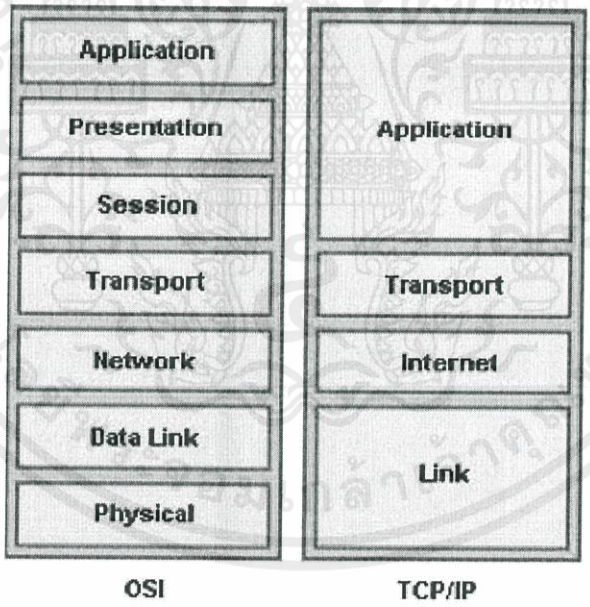
ชั้นพรีเซนเทชันทำหน้าที่ตกลงกับเครื่องโอสอีกฝั่งหนึ่งในชั้นเดียวกันในเรื่องของขั้นตอนและข้อบังคับในการรับส่งข้อมูลในระดับโปรแกรมประยุกต์จะมีขั้นตอน จุดประสงค์หลักของชั้นพรีเซนเทชันคือการกำหนดรูปแบบของการสื่อสาร เช่น ASCII Text, EBCDIC, Binary, JPEG รวมถึงการเข้ารหัส (Encryption) เช่น โปรแกรม FTP ต้องการรับส่งโอนย้ายไฟล์กับเครื่องเซิร์ฟเวอร์ปลายทางโพรโทคอล FTP จะอนุญาตให้ผู้ใช้ระบุรูปแบบของข้อมูลที่โอนย้ายกันได้ว่าเป็นแบบแอสกี (ASCII Text) หรือแบบ JPEG, ASCII, Binary, EBCDICTIFF, GIF และ MPEG เป็นต้น

7. ชั้นแอพพลิเคชัน (Application Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นแอพพลิเคชันเป็นชั้นที่อยู่บนสุดของการรับส่งข้อมูลในแบบจำลอง OSI ซึ่งทำหน้าที่ติดต่อกับผู้ใช้โดยจะรับคำสั่งต่าง ๆ จากผู้ใช้ส่งให้คอมพิวเตอร์แปลความหมาย การบริการต่าง ๆ ในชั้นแอพพลิเคชันจะเน้นการอำนวยความสะดวกแก่ผู้ใช้ด้วยโพรโทคอลต่าง ๆ ในชั้นแอพพลิเคชันทำให้ผู้ส่งสามารถทำการส่งไฟล์ อีเมล์หรืออื่น ๆ โดยไม่ต้องทำการเขียนโปรแกรมเพื่อกระทำดังกล่าว เช่น กรณีที่มีการถ่ายโอนข้อมูลระหว่างต้นทางและปลายทางก็ใช้บริการโพรโทคอล FTP เพื่อให้ทั้งสองฝั่งสามารถส่งไฟล์ระหว่างกันได้ การส่งอีเมล์โดยใช้บริการ SMTP หรือการท่องเว็บด้วยโพรโทคอล HTTP เป็นต้น ซึ่งการให้บริการเหล่านี้ถือเป็นการบริการทางเครือข่ายในรูปแบบแอพพลิเคชันของโพรโทคอลนั้น ๆ และทำงานตามคำสั่งที่ได้รับในระดับโปรแกรมประยุกต์ เช่น แปลความหมายของการกดปุ่มเมาส์ (Mouse) ให้เป็นคำสั่งในการคัดลอกไฟล์

2.2 แบบจำลอง TCP/IP (TCP/IP Model)



รูปที่ 2.2 ชั้นต่างๆของแบบจำลอง OSI และแบบจำลอง TCP/IP

TCP/IP (Transmission Control Protocol / Internet Protocol) เป็นระบบโพรโทคอลการสื่อสารพื้นฐานของระบบอินเตอร์เน็ตซึ่งสามารถใช้เป็นโพรโทคอลในการสื่อสารภายในเครือข่ายส่วนบุคคลที่เรียกว่า อินทราเน็ต (Intranet) และเอ็กทราเน็ต (Extranet) เมื่อมีการติดต่อโดยตรงกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเตอร์เน็ตเครื่องคอมพิวเตอร์จะใช้ TCP/IP เช่นเดียวกับคอมพิวเตอร์อื่น ๆ ทำให้สามารถส่งข้อความขอรับสารสนเทศได้

ซึ่งสถาปัตยกรรมรูปแบบ OSI มีคุณสมบัติไม่เพียงพอสำหรับการใช้งานในการสื่อสารข้อมูลในเครือข่ายจริง กระทรวงกลาโหมของสหรัฐอเมริกาหรือ DOD (The U.S. Department of Defend) จึงมีการกำหนดและเริ่มใช้รูปแบบสถาปัตยกรรมและโพรโทคอลสำหรับระบบเครือข่ายระดับ WAN (Wide Area Network) ที่ DOD ได้ทำการพัฒนาขึ้นมาคือสถาปัตยกรรม ARPANET และโพรโทคอล TCP/IP เพื่อใช้ในการเชื่อมโยงโฮสคอมพิวเตอร์ที่มีความหลากหลายแตกต่างกันและอยู่ห่างไกลกันให้สามารถติดต่อสื่อสารข้อมูลกัน แต่ ARPANET ยังถือว่าเป็นสถาปัตยกรรมเครือข่ายระดับ Wire Area Network มักใช้สำหรับการทหารมากกว่าที่จะใช้ในด้านธุรกิจหรือสาธารณะและเนื่องจากการเติบโตอย่างรวดเร็วของคอมพิวเตอร์ส่วนบุคคล (Personal Computer) ทำให้ DOD ต้องการพัฒนารูปแบบของสถาปัตยกรรมและโพรโทคอลสำหรับเครือข่ายระดับท้องถิ่น (Local Area Network : LAN) ขึ้นมาใช้เองด้วยเหตุผลหลัก 3 ประการคือ

1. DOD ต้องการรูปแบบสถาปัตยกรรมและโพรโทคอลที่สามารถใช้งานได้จริง
2. DOD มีข้อกำหนดลักษณะของเครือข่ายที่พิเศษเฉพาะออกไปจากเครือข่ายของระบบเปิด
3. DOD ต้องการรูปแบบสถาปัตยกรรมและโพรโทคอลของเครือข่ายที่ง่ายและไม่ซับซ้อน

โครงสร้างของสถาปัตยกรรมของชุดโพรโทคอล TCP/IP แบ่งออกเป็น 3 ส่วนหลัก ๆ คือส่วนกรรมวิธีปฏิบัติการหรือโพรเซส (Process) โฮส (Host) และเครือข่าย (Network) ส่วนของโพรเซสได้แก่ เอนทิตี้ (Entity) คือแอพพลิเคชันที่ต้องการติดต่อสื่อสารโดยทุกโพรเซสจะกระทำในเครื่องโฮสหรือสเตชัน (Station) ซึ่งแต่ละโฮสสามารถมีหลาย ๆ เอนทิตี้ได้พร้อมกัน การสื่อสารระหว่างเอนทิตี้ของโฮสเครื่องหนึ่งหรือหลายเครื่องจะกระทำโดยผ่านทางเครือข่ายที่โฮสเชื่อมต่ออยู่

การทำงานที่สัมพันธ์กันระหว่างโพรเซส โฮส และเครือข่ายของสถาปัตยกรรม TCP/IP ทำให้สามารถจัดรูปแบบของสถาปัตยกรรม TCP/IP ได้เป็น 4 ชั้นและสามารถกำหนดชนิดของโพรโทคอลที่ทำงานในแต่ละชั้นได้เป็น 4 โพรโทคอล

โพรโทคอล TCP/IP นั้นเอนทิตี้ของแต่ละชั้นอาจจะติดต่อสื่อสารข้อมูลโดยผ่านเอนทิตี้ในชั้นเดียวกันหรือเอนทิตี้ในชั้นที่ถัดลงไปของแบบจำลอง TCP/IP โดยไม่จำเป็นจะต้องเป็นชั้นที่ติดกัน ลำดับชั้นของโพรโทคอลในระบบอินเทอร์เน็ตมีลำดับชั้นที่น้อยกว่าโครงสร้างแบบจำลอง OSI โดยในแบบจำลอง OSI มีลำดับชั้นของโพรโทคอลทั้งหมด 7 ชั้นแต่ในระบบอินเทอร์เน็ตมีทั้งหมดเพียง 4 ชั้นเท่านั้น คือ

1. ชั้นลิงค์ (Link Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทียบได้กับชั้นฟิสิคัลและชั้นดาต้าลิงค์ในแบบจำลอง OSI เป็นลำดับชั้นที่ไม่ได้เกี่ยวข้องกับระบบอินเทอร์เน็ตโดยตรงแต่เป็นระบบพื้นฐานของการเชื่อมต่อที่ระบบอินเทอร์เน็ตใช้ส่งข้อมูลภายในเครือข่าย

หน้าที่คือการจัดเตรียมข้อมูลเพื่อให้เหมาะสมก่อนที่จะส่งไปตามสายส่งไปยังที่หมายปลายทาง ได้แก่ การจัดเตรียม Header การควบคุมระบบฮาร์ดแวร์และซอฟต์แวร์ที่จะใช้ในการจัดส่ง เช่น การเชื่อมต่อกับการ์ดเครือข่าย (Network card) และการใช้งานไดร์เวอร์ของอุปกรณ์ (Device driver) หน้าที่สำหรับการรับข้อมูลคือนำข้อมูลส่วนหัวออกมาและจัดเตรียมข้อมูลเพื่อส่งต่อไปยังชั้นถัดไปของแบบจำลอง TCP/IP ซึ่งก็คือชั้นอินเตอร์เน็ต ชั้นทรานสปอร์ต และชั้นแอพพลิเคชัน โพรโทคอลในชั้นนี้ ได้แก่ ATM, DTM, Ethernet, FDDI เป็นต้น

2. ชั้นอินเตอร์เน็ต (Internet Layer)

ชั้นอินเตอร์เน็ตสามารถเทียบได้กับชั้นเน็ตเวิร์กในแบบจำลอง OSI โดยชั้นอินเตอร์เน็ตเป็นชั้นที่มีหน้าที่ส่งข้อมูลจากจุดเริ่มต้นไปยังปลายทางโดยหาเส้นทางที่ข้อมูลจะใช้เดินทางผ่านเครือข่ายหนึ่งไปยังอีกเครือข่ายหนึ่งจนกระทั่งถึงปลายทาง โพรโทคอลที่ใช้ในชั้นเน็ตเวิร์กนี้คือ IP (Internet Protocol) ทำหน้าที่เปรียบเสมือนซองจดหมายซึ่งระบุถึงที่อยู่ของต้นทางและปลายทางโดยมีบุรุษไปรษณีย์ทำหน้าที่ส่งจดหมายนั้นผ่านกรมการไปรษณีย์ในพื้นที่ต่างๆจนถึงจุดหมายปลายทาง ซึ่งที่อยู่บนซองจดหมายของ IP เรียกว่าหมายเลข IP และที่ทำการไปรษณีย์คือเราท์เตอร์ (Router) ทำหน้าที่ค้นหาเส้นทางที่เหมาะสมเพื่อส่งข้อมูลไปตามสายส่งจนกระทั่งถึงปลายทาง โพรโทคอลในชั้นนี้ ได้แก่ IP, IPX, IGMP, ICMP, RSVP และ BGP เป็นต้น

3. ชั้นทรานสปอร์ต (Transport Layer)

ชั้นทรานสปอร์ตสามารถเทียบได้กับชั้นทรานสปอร์ตในแบบจำลอง OSI โดยชั้นทรานสปอร์ตมีหน้าที่ควบคุมการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์สองเครื่องที่ติดต่อกันซึ่งอาจแบ่งได้สองลักษณะคือบริการการส่งข้อมูลแบบที่สถาปนาการเชื่อมต่อและบริการการส่งข้อมูลแบบไม่สถาปนาการเชื่อมต่อและทำการจัดส่งข้อมูลไปยังแอพพลิเคชันที่ต้องการข้อมูล ซึ่งโพรโทคอลในชั้นนี้ ได้แก่ TCP, UDP, RTP เป็นต้น

4. ชั้นแอพพลิเคชัน (Application Layer)

ชั้นแอพพลิเคชันสามารถเทียบได้กับชั้นเซสชัน ชั้นพรีเซนเทชัน และชั้นแอพพลิเคชันในแบบจำลอง OSI โดยชั้นแอพพลิเคชันจะครอบคลุมบริการที่เกี่ยวข้องกับการรักษาความปลอดภัย การเข้ารหัส การเชื่อมต่อระหว่างโปรแกรมประยุกต์ และเป็นชั้นที่โปรแกรม

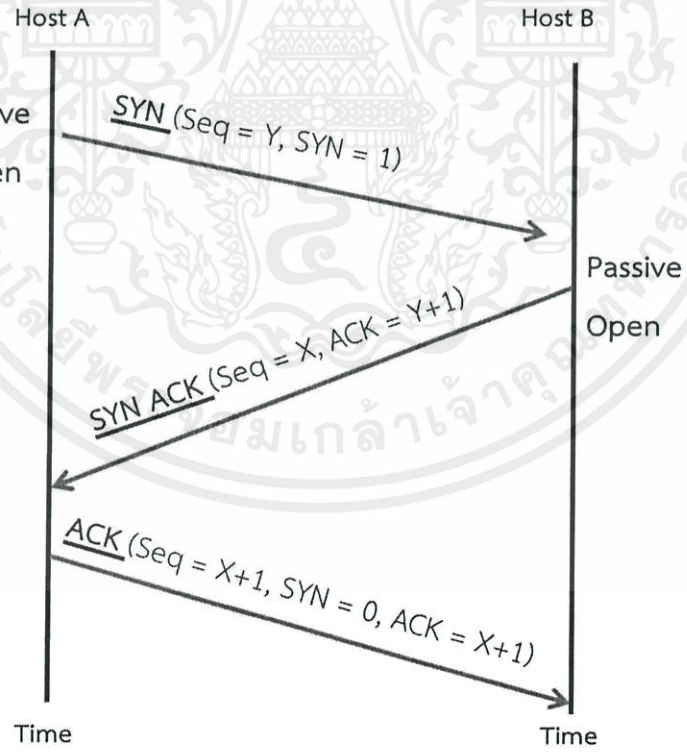
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประยุกต์ใช้งานโดยตรง โดยโพรโทคอลที่อยู่บนชั้นแอพพลิเคชันนี้ถูกออกแบบให้เหมาะสมกับประเภทของโปรแกรมประยุกต์เฉพาะทาง เช่น โปรแกรมอีเมล์ใช้โพรโทคอล SMTP สำหรับส่งอีเมล์ (E-mail) การรับและเรียกดูอีเมล์จะใช้โพรโทคอล POP3 ส่วนโปรแกรมเว็บเบราว์เซอร์ (Web browser) จะใช้โพรโทคอล HTTP สำหรับเรียกดูเว็บ เป็นต้น

2.2.1 การเชื่อมต่อของทีซีพี

ทีซีพีเป็นโพรโทคอลที่มีคุณสมบัติการเชื่อมต่อที่เชื่อถือได้ (Connection-oriented) ซึ่งจะมีการสร้างช่องทางการติดต่อเสมือนระหว่างต้นทางและปลายทาง เซกเมนต์ทุก ๆ เซกเมนต์จะถูกส่งผ่านทางช่องทางเสมือนนี้ ซึ่งมีเฟสทั้งหมด 3 เฟส ดังนี้

2.2.1.1 การสร้างการเชื่อมต่อ (Connection Establishment)



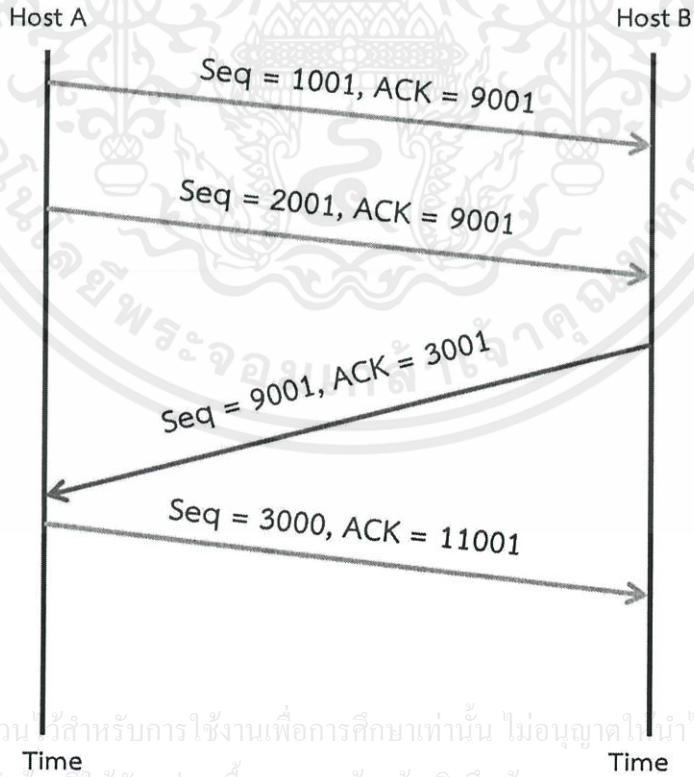
รูปที่ 2.3 การสถาปนาการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนที่จะเริ่มต้นการสื่อสารระหว่างผู้ส่งกับผู้รับจะต้องมีการส่งสัญญาณเพื่อบอกโฮสอีกฝั่งหนึ่งให้เตรียมตัวติดต่อ ซึ่งกระบวนการที่ใช้เรียกว่า 3-Way Handshake มีขั้นตอนคือ

1. โฮสต้นทางจะทำการส่งเซกเมนต์โดยเปิด SYN Flag ระบุหมายเลขพอร์ตที่ต้องการติดต่อบนโฮสปลายทางและระบุหมายเลขลำดับของข้อมูล (Initial Sequence Number)
2. โฮสปลายทางเมื่อได้รับข้อมูลเซกเมนต์จากโฮสต้นทางจะตอบกลับด้วยการเพิ่มค่าหมายเลข Sequence ที่ได้รับขึ้นอีก 1 พร้อมทั้งระบุหมายเลข Sequence ของตนเองและเปิด SYN กับ ACK Flag
3. โฮสต้นทางเมื่อได้รับการตอบกลับจากเครื่องโฮสปลายทางก็จะทำการตอบรับกลับไปโดยการเพิ่มค่าหมายเลข Sequence ของโฮสปลายทางขึ้นอีก 1 และเปิด ACK Flag เมื่อผ่านการสร้าง connection ทั้ง 3 ขั้นตอนแล้ว ทั้งโฮสต้นทางและโฮสปลายทางเปรียบเสมือนมีการเชื่อมต่อถึงกัน

2.2.1.2 การส่งข้อมูลระหว่างต้นทางและปลายทาง (Data Transfer)



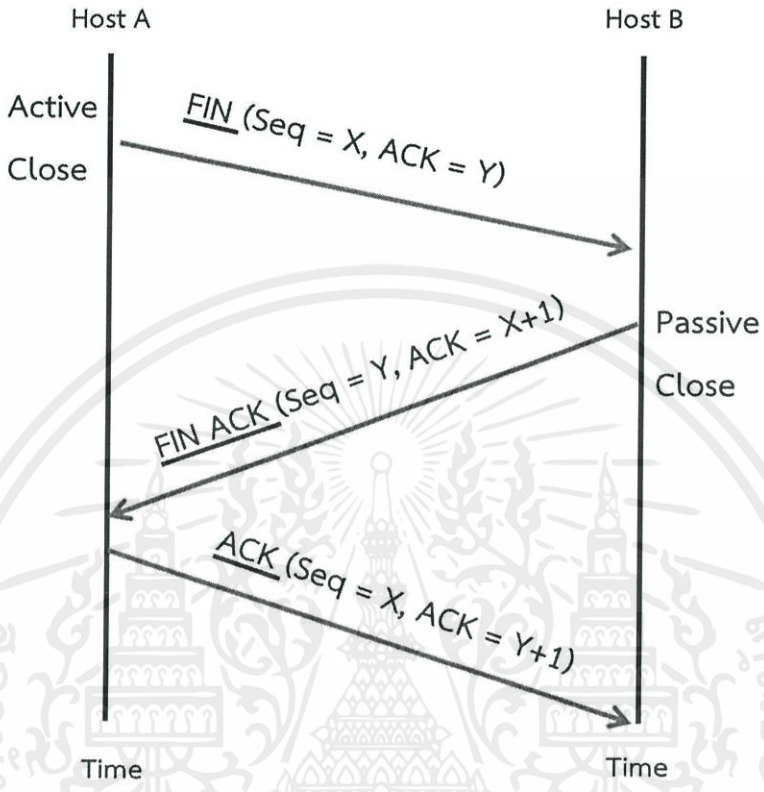
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4 การส่งข้อมูลระหว่างต้นทางและปลายทาง

จากรูปเป็นการส่งเซกเมนต์ระหว่างโฮสต์ต้นทางไปยังโฮสต์ปลายทางโดยกระบวนการนี้เกิดขึ้นหลังจากโฮสต์ต้นทางได้ทำการขอเริ่มต้นการเชื่อมต่อจากโฮสต์ปลายทางแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.3 การปิดการเชื่อมต่อ (Connection Termination)



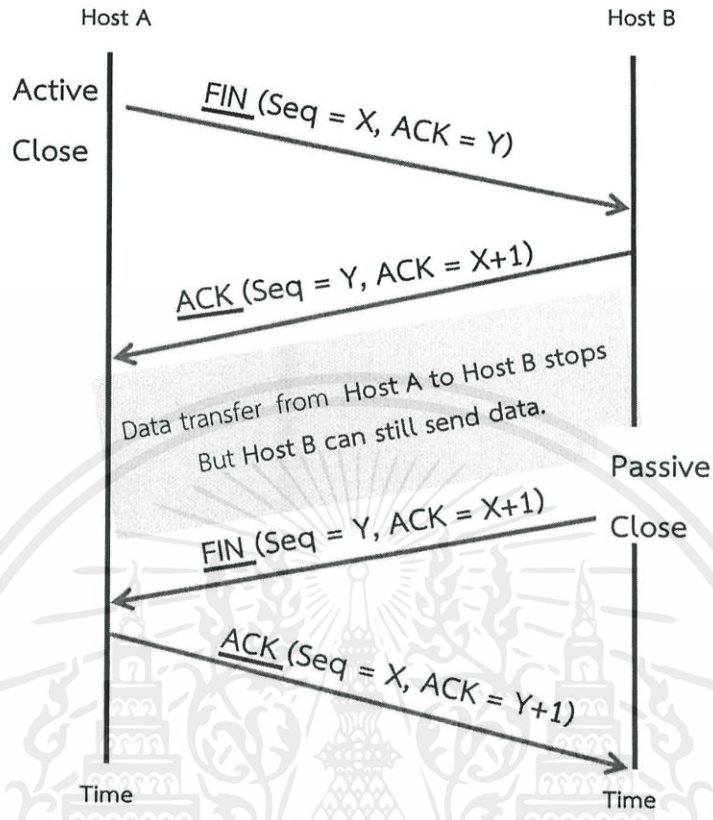
รูปที่ 2.5 การปิดการเชื่อมต่อ

เมื่อการสื่อสารของทั้งสองฝั่งจบลง และไม่ต้องการรับส่งข้อมูลอีกต่อไป จะต้องทำตามขั้นตอนการยุติการสื่อสารเพื่อให้การสื่อสารจบลงอย่างสมบูรณ์ ซึ่งมีอยู่ 3 ขั้นตอนคือ

1. โฮสต้นทางจะส่ง FIN เซกเมนต์ไปโดยมีหมายเลข Sequence พร้อมกับ FIN Flag ไปยังโฮสปลายทาง
2. โฮสปลายทางตอบกลับส่ง FIN + ACK เซกเมนต์กลับมาโดยตอบรับหมายเลข Sequence และส่ง FIN 1 ACK กลับไปยังโฮสต้นทาง
3. โฮสต้นทางตอบกลับโดยส่ง ACK เซกเมนต์ไป ถ้าไม่มีการส่งเซกเมนต์อีกก็จะไม่มีหมายเลข Sequence

ในทีซีพีสามารถหยุดการส่งเซกเมนต์ของโฮสต้นทางในขณะที่ยังคงสามารถรับเซกเมนต์ได้ เรียกว่า “Half-close” ซึ่งมีหลักการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 Half-Close

1. โฮสต้นทางส่งการร้องขอการเชื่อมต่อแบบ Half-close ไปยังโฮสปลายทางโดยการส่ง FIN เซกเมนต์ไป
2. โฮสปลายทางทำการตอบรับการร้องขอโดยการส่ง ACK เซกเมนต์กลับไปยังต้นทาง
3. การส่งเซกเมนต์จากโฮสต้นทางไปยังโฮสปลายทางถูกหยุดลง แต่โฮสปลายทางยังคงสามารถส่งเซกเมนต์ไปต้นทางได้
4. หลังจากส่งเซกเมนต์ทั้งหมดแล้ว โฮสปลายทางจะส่ง FIN เซกเมนต์ไปยังโฮสต้นทาง ซึ่งจะถูกตอบกลับจากโฮสต้นทางโดย ACK เซกเมนต์

2.3 โพรโทคอลทีซีพี

ทีซีพีเป็นโพรโทคอลที่ใช้สื่อสารระหว่างโฮสที่มีความน่าเชื่อถือ โพรโทคอลในระดับ IP หรือแม้กระทั่ง UDP (User Datagram Protocol) โดยจะสนใจข้อมูลเพียงหนึ่งเซกเมนต์ กลไกของโพรโทคอลจะมีหน้าที่ตรวจสอบความถูกต้องเพียงเฉพาะเซกเมนต์นั้น ๆ เมื่อจะทำการส่งเซกเมนต์ใหม่จะถือว่าเป็นข้อมูลชุดใหม่ที่ไม่มีความสัมพันธ์ใด ๆ กับข้อมูลเซกเมนต์อื่น ดังนั้นการสื่อสาร 1 ครั้งจึงใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพียง 1 เซกเมนต์ แต่สำหรับทีซีพีแล้วจะเห็นว่าข้อมูลนั้นเป็นกระแสข้อมูล (Stream) คือมีความสัมพันธ์ต่อเนื่องกัน มีกลไกในการตรวจสอบทั้งด้านผู้ส่งและด้านผู้รับเพื่อให้แน่ใจว่าสามารถสื่อสารกันได้จริงจึงจะมีการรับส่งข้อมูลเกิดขึ้น ตลอดจนการยกเลิกการติดต่อก็มีกลไกสำหรับแจ้งให้อีกฝั่งทราบ ทำให้การสื่อสารด้วยทีซีพีเสมือนว่าทั้ง 2 ฝั่งคือผู้รับและผู้ส่งได้มีการเชื่อมต่อกันตลอดเวลาที่การรับส่งข้อมูล จนกระทั่งการสื่อสารทั้งหมดเสร็จสิ้นจึงจะทำการยกเลิกการเชื่อมต่อนั้นเสีย

จุดเด่นที่สำคัญของทีซีพีคือความมีเสถียรภาพและความถูกต้องของการสื่อสารซึ่งมีความน่าเชื่อถือได้สูง คุณสมบัติที่ทำให้ทีซีพีมีข้อดีดังกล่าวคือ

1. ข้อมูลที่จะส่งผ่านทีซีพีจะถูกนำมาแตกย่อยออกเป็นส่วน ๆ ให้มีขนาดเหมาะสมสำหรับการส่งข้อมูล โดยทีซีพีมีกลไกในการพิจารณาว่าขนาดเท่าใดจะทำให้การรับ – ส่งนั้นมีประสิทธิภาพและน่าเชื่อถือสูงสุด โดยข้อมูลแต่ละชุดที่แบ่งออกและทำการส่งโดยทีซีพีแต่ละครั้งจะเรียกว่า ทีซีพีเซกเมนต์ (TCP Segment)
2. ในการส่งข้อมูลแต่ละครั้งทีซีพีจะมีการจับเวลาไว้เสมอเพื่อรอการตอบรับจากผู้รับว่าได้รับข้อมูลถูกต้อง หากหมดเวลาแล้วไม่มีการตอบรับทีซีพีจะถือว่าข้อมูลไปไม่ถึงและทำการแก้ปัญหาที่เกิดขึ้น เช่น ยกเลิกการติดต่อ หรือส่งข้อมูลซ้ำ เป็นต้น ทำให้ทราบสถานะการส่งข้อมูลตลอดเวลา
3. ทีซีพีมีเช็กซัม (Checksum) โดยเช็กซัมจะครอบคลุมทั้ง TCP Header และ TCP Data เพื่อเป็นการป้องกันและตรวจสอบว่าข้อมูล ที่ส่งมานั้นถูกต้องและไม่ได้ถูกแก้ไขระหว่างทาง หากทีซีพีได้รับข้อมูลที่ทำการตรวจสอบกับเช็กซัมแล้วปรากฏว่ามีความผิดพลาดเกิดขึ้นทีซีพีจะทิ้งข้อมูลที่ได้รับและจะไม่ทำการตอบรับข้อมูลนั้นกลับไปยังผู้ส่งเพื่อให้ทางฝ่ายผู้ส่งทำการส่งใหม่หรือหาข้อบกพร่องและพยายามแก้ไข
4. เนื่องจากทีซีพีอาศัย IP ในการส่งข้อมูลซึ่ง IP เองอาจจะถูกแยกส่วนข้อมูลได้และทำให้ข้อมูลที่ถูกแยกส่วนออกมานั้นส่งถึงปลายทางในลำดับที่ไม่ถูกต้องได้ หน้าที่ของทีซีพีเมื่อรับข้อมูลที่ แยกส่วนออกมานั้นจะต้องนำข้อมูลแต่ละส่วนมาประกอบรวมกันให้ถูกต้องสมบูรณ์ก่อนจะส่งไปยัง ชั้นแอพพลิเคชันต่อไป
5. อาจจะมีกรณีที่ไอพีดาต้าแกรม (IP Datagram) นั้นถูกส่งซ้ำขึ้นได้ ทีซีพีที่รับข้อมูลซ้ำดังกล่าวจะต้องทราบว่าเป็นไอพีดาต้าแกรมที่ซ้ำและไม่นำข้อมูลไปใช้งาน
6. ทีซีพีมีกลไกควบคุมการไหลของข้อมูลโดยการควบคุมนี้จะต้องอาศัยลำดับของการรับส่งที่ถูกต้องและสัมพันธ์กันทั้ง 2 ฝั่ง ในขณะเดียวกันข้อมูลที่ส่งนั้นจะต้องอาศัย IP หลายดาต้าแกรมจึงจะได้รับข้อมูลครบทั้งหมด ดังนั้นในการรับข้อมูลทางฝ่ายรับจึงต้องเตรียม

เอกสารนี้เป็นเอกสารที่สงวน ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่พักข้อมูล (Buffer) ไว้จำนวนหนึ่งเพื่อรอรับข้อมูลและรวบรวมข้อมูลทั้งชุดให้อยู่ในที่พักข้อมูลก่อนที่จะทำการจัดเรียงข้อมูล ตรวจสอบความถูกต้องแล้วจึงส่งต่อไป จะเห็นว่าขนาดของข้อมูลมิได้ถูกจำกัดที่ขนาดของดาต้าแกรมใด ๆ ข้อมูลที่ส่งอาจจะมีขนาดใหญ่มากอยู่ในหลายดาต้าแกรมก็เป็นได้ ดังนั้นเพื่อป้องกันการส่งข้อมูลขนาดใหญ่เร็วเกินไปจนทำให้ทางฝ่ายรับไม่มีหน่วยความจำเพียงพอที่จะเป็นที่พักข้อมูลการส่งข้อมูลจึงถูกจำกัดโดยจะอนุญาตให้ทำการส่งข้อมูลได้เท่าที่ฝ่ายรับมีที่พักข้อมูลเพียงพอเท่านั้น

2.3.1 TCP Header

<table>
<tr><td colspan="6">Source port address</td><td colspan="2">Destination port address</td></tr>
<tr><td colspan="8">Sequence Number</td></tr>
<tr><td colspan="8">Acknowledgement Number</td></tr>
<tr><td>Header Length</td><td>Reserve</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>RYN</td><td>FIN</td><td>Window size</td></tr>
<tr><td colspan="6">Checksum</td><td colspan="2">Urgent pointer</td></tr>
<tr><td colspan="8">Options</td></tr>
</table>

รูปที่ 2.6 TCP Segment

ส่วนประกอบของ TCP Segment มีดังต่อไปนี้

1. หมายเลขพอร์ตต้นทาง (Source Port Number) คือพอร์ตที่โฮสต้นทางใช้ในการสื่อสารกันและจะใช้พอร์ตนั้นไปตลอดที่การสื่อสารยังไม่ยุติลง โดยทั่วไปพอร์ตนี้จะเรียกว่าไคลเอนต์พอร์ต (Client Port) หมายถึงพอร์ตที่ไคลเอนต์เปิดขึ้นมาเพื่อรอการตอบรับจากเซิร์ฟเวอร์โดยพิจารณาจากทิศทางของแพ็คเก็ตที่ส่งมาจากไคลเอนต์ไปยังเซิร์ฟเวอร์ไคลเอนต์พอร์ตจะมีหมายเลขไม่แน่นอนและเปลี่ยนไปทุกครั้งที่มีการเริ่มการเชื่อมต่อใหม่ เป็นพอร์ตที่ถูกเปิดไว้ในระยะเวลาสั้น ๆ ซึ่งค่าที่เป็นไปได้ของพอร์ตนี้ขึ้นอยู่กับการจัดสรรของระบบปฏิบัติการ
2. หมายเลขพอร์ตปลายทาง (Destination Port Number) คือหมายเลขพอร์ตบนโฮสปลายทางที่โฮสต้นทางต้องการติดต่อด้วย อาจหมายถึงแอพพลิเคชั่นที่ให้บริการอยู่พอร์ตนั้นที่โฮสปลายทาง พอร์ตนี้เรียกอีกอย่างหนึ่งว่า เซิร์ฟเวอร์พอร์ต (Server Port)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลง เนื้อหาบางส่วนของเอกสารนี้ก่อนนำไปใช้

หมายเลขพอร์ตที่เปิดไว้จะขึ้นอยู่กับแอพพลิเคชันที่ให้บริการ โดยทั่วไปแอพพลิเคชันแต่ละประเภทจะมีหมายเลขพอร์ตเป็นมาตรฐาน

3. หมายเลข Sequence คือหมายเลขลำดับที่ใช้อ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อให้ทั้ง 2 ฝ่ายได้รับทราบตรงกันว่าเป็นข้อมูลของชุดใด การนำไปใช้งานจะได้ไม่ปะปนกันและมีลำดับที่ถูกต้อง เนื่องจากการสื่อสารข้อมูลผ่านทีซีพีนั้นจังหวะและลำดับเป็นส่วนสำคัญของโพรโทคอลเทียบเท่ากับข้อมูลใน TCP Header รวมไปถึงการที่ข้อมูลในแต่ละ TCP Segment อาจจะถูกทำการแยกส่วนทำให้ข้อมูลถูกแบ่งออกและส่งไปในลำดับที่ไม่เรียงกัน หากไม่มีจุดอ้างอิงของข้อมูลจะไม่สามารถอ่านข้อมูลกลับใหม่ได้อย่างสมบูรณ์และถูกต้อง
4. หมายเลขการตอบรับการร้องขอ (Acknowledge Number) คือหมายเลข Sequence ที่ตอบรับการร้องขอ เนื่องจากหมายเลข Sequence ที่ใช้ในการอ้างอิงนั้นผู้ที่เริ่มส่งข้อมูลจะเป็นผู้กำหนดหมายเลขขึ้นมาและส่งไปพร้อมกับการสร้างการเชื่อมต่อครั้งใหม่ แต่สำหรับฝ่ายที่ถูกติดต่อจำเป็นต้องกำหนดหมายเลขสำหรับใช้อ้างอิงในการตอบรับ
5. ความยาวของ Header โดยทั่วไปจะมีค่าเท่ากับ 20 ไบต์แต่ถ้าหากมีการเพิ่มเงื่อนไขอาจจะทำให้ขนาดของ Header ยาวขึ้นตามข้อมูลที่ต้องเพิ่มมาจากเงื่อนไขนั้น ๆ แต่ทั้งหมดจะไม่เกิน 60 ไบต์
6. Flag มีทั้งหมด 6 บิตแต่ละบิตมีชื่อและมีความหมาย ดังนี้

ตารางที่ 2.1 TCP Header Flag

Flag	ความหมาย
URG	บอกความหมายว่าเป็นข้อมูลด่วนและมีข้อมูลพิเศษมาด้วยซึ่งอยู่ใน Urgent Pointer
ACK	ข้อมูลในหมายเลขการตอบรับการร้องขอ อนุญาตให้นำมาใช้งานได้
DSH	แจ้งให้ผู้รับข้อมูลทราบว่า ควรจะส่งข้อมูลเซกเมนต์ที่แจ้งไปยังโพรเซสที่กำลังรออยู่ทันที
RST	เกิดความผิดพลาดขึ้นด้วยเหตุผลต่าง ๆ ให้เริ่มต้นการเชื่อมต่อใหม่
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่าขอยุติการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ขนาดของหน้าต่างข้อมูล (Window Size) คือขนาดของการรับ - ส่งข้อมูลในแต่ละครั้งที่ผู้รับสามารถรับได้ เนื่องจากในการรับข้อมูลนั้นทางผู้รับจะต้องจัดเตรียมหน่วยความจำในการพักข้อมูลที่มากจากทีซีพีและทำการกระจายข้อมูลออกมา หากไม่มีการตกลงขนาดหน้าต่างข้อมูลที่ผู้รับสามารถรับได้ก็จะทำให้การสื่อสารข้อมูลไม่สมดุลและฝ่ายรับอาจจะประมวลผลไม่ทัน ส่งผลให้ต้องส่งข้อมูลซ้ำกันหลายครั้ง
8. เช็กซัม (Checksum) คือสิ่งที่ใช้ในการตรวจสอบความถูกต้องของข้อมูลใน TCP Segment
9. Urgent Pointer คือจุดชี้ตำแหน่งไบต์ของข้อมูลที่ต้องดำเนินการเริ่มอย่างเร่งด่วน โดยจะมีการอ่านค่านี้ถ้า URG flag ถูกตั้งค่าเป็น 1

ตารางที่ 2.2 การเปรียบเทียบคุณสมบัติระหว่าง UDP และ TCP

คุณสมบัติ	UDP	TCP
การสร้างการเชื่อมต่อ	- การส่งข้อมูลเป็นแบบ Connectionless	- การส่งข้อมูลเป็นแบบ Connection-oriented
การรับรองความถูกต้อง	- ไม่รับรองความถูกต้องของข้อมูล	- รับรองความถูกต้องของข้อมูล
ความเร็วในการติดต่อสื่อสาร	- ติดต่อสื่อสารได้เร็วกว่า TCP	- มีการติดต่อสื่อสารที่ช้ากว่า UDP
ลักษณะของการส่งข้อมูล	- ได้เปรียบในการส่งข้อมูลได้ทั้งแบบ Unicast, Multicast และ Broadcast	- ข้อมูลมีความยาวมากกว่า และมีความซับซ้อนมากกว่า UDP
การใช้งาน	- เหมาะสำหรับแอพพลิเคชันที่ต้องการแค่รับส่งข้อมูลเท่านั้น เช่น SNMP ,DHCP ,TFTP, DNS RPC, NFS เป็นต้น	- เหมาะสำหรับข้อมูลที่สำคัญมาก ๆ เช่น FTP, Telnet, HTTP และ SMTP เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 โพรโทคอลที่นำมาศึกษา

เนื่องจากโพรโทคอลทีซีพีมีหลากหลายชนิด โครงงานนี้ทางผู้จัดทำจึงเลือก 5 โพรโทคอลที่น่าสนใจนำมาศึกษาคือ

2.3.2.1 TCP Tahoe

Tahoe เป็นขั้นตอนวิธีการควบคุมความคับคั่งของทีซีพีโพรโทคอลถูกเสนอโดย Van Jacobson ซึ่งจะใช้อัลกอริทึม Slow Start ก่อนที่ค่าปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้ (Congestion window : cwnd) จะเพิ่มขึ้นจนถึงค่า Slow start Threshold (ssthresh) และมีการนำหลักการของ Congestion avoidance มาใช้โดยจะทำให้ค่าปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้ (cwnd) มีค่าเพิ่มขึ้นในลักษณะเป็นแนวเส้นตรง อีกทั้งยังเพิ่มประสิทธิภาพการทำงานโดยนำหลักการของการส่งข้อมูลกลับไปใหม่อย่างรวดเร็ว (Fast Retransmit) มาใช้ซึ่งรวมไปถึงการปรับปรุงการหาค่าคาบเวลาหนึ่งรอบในการส่งข้อมูล (Round-trip Time) เพื่อใช้ในการตั้งค่าการกำหนดการหมดเวลารอของการส่งแพ็คเก็ตกลับไปใหม่อีกครั้งที่เหมาะสม

Tahoe ยังนำหลักการของการสื่อสารระหว่างการส่งแพ็คเก็ตมาใช้ เช่น หากการเชื่อมต่อการใช้งานมีความจุความกว้างของช่องทางในการส่งข้อมูล (Bandwidth) ที่สามารถใช้ได้จำกัดแล้ว แพ็คเก็ตจะไม่ถูกส่งไปภายในเครือข่ายเว้นแต่ว่าแพ็คเก็ตจะถูกส่งออกไปอย่างมีประสิทธิภาพ โดยฝั่งผู้รับจะใช้หลักการนี้ในการตอบกลับตามเวลาที่ถูกส่งออกไปของแพ็คเก็ตและใช้ค่าปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้ (cwnd) ที่บ่งบอกถึงความคับคั่งของเครือข่ายในขณะนั้นด้วย อย่างไรก็ตามยังมีปัญหาที่ควรได้รับการแก้ไข ดังนี้

1. การพิจารณาหาปริมาณความกว้างของช่องทางในการส่งข้อมูลที่สามารถใช้ได้
2. การตรวจสอบให้แน่ใจว่ามีการรักษาความสมดุลอยู่เสมอ
3. วิธีตอบสนองต่อความคับคั่งของข้อมูล

อัลกอริทึมใน Tahoe มีดังนี้

1. อัลกอริทึม Slow Start

อัลกอริทึม Slow Start เป็นอัลกอริทึมที่ใช้โดยฝั่งผู้ส่งข้อมูลเพื่อควบคุมปริมาณข้อมูลที่ถูกส่งภายในเครือข่ายไม่ให้มีมากเกินไป ผู้ส่งจะทำการส่งแพ็คเก็ต เมื่อได้รับสัญญาณการร้องขอการขอเริ่มการเชื่อมต่อ (Acknowledgement : ACK) ต้องมีการตอบกลับของข้อมูลในเครือข่ายเพื่อทำการส่งข้อมูลเข้าไปในเครือข่าย แสดงว่าเมื่อใดก็ตามที่ทีซีพีจะเริ่มต้นการเชื่อมต่อหรือเริ่มต้นการส่งแพ็คเก็ตใหม่อีกครั้งหลังจากแพ็คเก็ตมีความเสียหายจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกิดกระบวนการที่เรียกว่า Slow Start ซึ่งมีการนำปริมาณของข้อมูลที่ฝั่งผู้ส่งสามารถส่งไปภายในเครือข่าย (cwnd) ก่อนที่จะได้รับการตอบรับการร้องขอมาใช้ในกระบวนการด้วย

หลักการทำงานของอัลกอริทึม Slow Start คือฝั่งผู้ส่งจะตั้งค่า cwnd เป็น 1 ซึ่งเป็นค่าเริ่มต้นและถ้าฝั่งผู้รับได้รับแพ็คเก็ตโดยมีหมายเลข Sequence เป็น n ฝั่งผู้รับจะส่งแพ็คเก็ต ACK เพื่อยืนยันการรับแพ็คเก็ตพร้อมทั้งส่งหมายเลข Sequence ของแพ็คเก็ตต่อไปเท่ากับ (n+1) ที่ผู้ส่งจะต้องส่งมาให้ จากนั้นทางฝั่งผู้ส่งจะเพิ่ม cwnd หลังจากส่ง ACK ของแพ็คเก็ตไปและเมื่อได้รับการตอบรับการร้องขอแต่ละครั้งให้เพิ่ม cwnd อีกเป็นสองเท่า โดยค่า cwnd แสดงจำนวนแพ็คเก็ตที่ผู้ส่งสามารถส่งไปให้ผู้รับโดยไม่ต้องรอการยืนยัน แต่การเพิ่มขึ้นของจำนวนแพ็คเก็ตที่ผู้ส่งสามารถส่งไปให้ผู้รับมีความจำกัดเพราะถูกทำให้ลดลงเมื่อเกิดการส่งแพ็คเก็ตที่ไม่สมบูรณ์ซึ่งอาจเกิดจาก Timeout หรือเกิด Duplicate ACK ตอบกลับมาในแพ็คเก็ตเดียวกัน

ตัวอย่างการเกิด Slow start มีดังนี้คือรอบแรกผู้ส่งจะส่ง 1 แพ็คเก็ตไปฝั่งผู้รับ รอบสองผู้ส่งจะส่ง 2 แพ็คเก็ต รอบสามผู้ส่งจะส่ง 4 แพ็คเก็ต จะเห็นว่ามีการเพิ่มขึ้นเป็นแบบ Exponential ไปจนกระทั่งแพ็คเก็ตเกิดการเสียหายหรือสูญหาย ซึ่งก็คือเริ่มที่จะมีการคับคั่งภายในเครือข่าย จากนั้นผู้ส่งจะทำการลดอัตราการส่งและลดความคับคั่งภายในเครือข่ายลงโดยให้ค่า cwnd เป็น 1 และเริ่มต้นอัลกอริทึม Slow Start ใหม่อีกครั้งหนึ่ง

2. อัลกอริทึมการหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่าย (Congestion Avoidance)

การหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่ายเป็นการแก้ไขปัญหาการสูญเสียของแพ็คเก็ตโดยควบคุมการไหลของข้อมูลหนึ่งชุดทั้งฝั่งผู้ส่งและผู้รับ เมื่ออัตราการส่งแพ็คเก็ตที่เข้าสู่เราท์เตอร์มีความเสียหาย เช่น การทำเช็กซัมเกิดความผิดพลาด เป็นต้น ซึ่งจะทำให้อัตราการส่งแพ็คเก็ต ลดลงทันที

อย่างไรก็ตามอัลกอริทึมการหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่ายและอัลกอริทึม Slow Start เป็นอัลกอริทึมที่มีจุดประสงค์ในการควบคุมความคับคั่งของข้อมูลที่แตกต่างกันแต่ประสิทธิภาพของทั้งสองอัลกอริทึมจะขึ้นตรงต่อกันโดยการนำทั้งสองอัลกอริทึมมาผสมผสานกันนั้นจะใช้ cwnd และขนาด Slow Start Threshold (ssthresh) ในการระบุจำนวนที่แพ็คเก็ตสามารถส่งไปได้ ซึ่งเป็นไปตามขนาดของหน้าต่างที่ได้รับมาโดยจะมีขนาดเท่ากับที่พักข้อมูลของหน้าต่างผู้รับ โดยหลังจากการเชื่อมต่อทีซีพีจะตั้งค่าให้ขนาดของ cwnd เท่ากับ 1 และเริ่มอัลกอริทึม Slow start เมื่อเกิดความคับคั่งของข้อมูลขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครึ่งหนึ่งของหน้าต่างปัจจุบันจะถูกเก็บลงใน ssthresh และ cwnd จะถูกตั้งค่าให้เท่ากับ 1 อีกครั้ง

ทีซีพีสามารถทำอัลกอริทึม Slow Start ได้นานจนกระทั่งขนาดของ cwnd มีค่าน้อยกว่า ssthresh เพราะหลังจาก cwnd มีขนาดมากกว่า ssthresh ทีซีพีจะทำการเริ่มอัลกอริทึมการหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่ายอีกครั้งหนึ่งและเพิ่มขนาดของ cwnd ใหม่แบบเส้นตรงโดยมีค่าตามสมการดังนี้

$$\text{cwnd new} = \text{SegmentSize2} / \text{cwnd old} \qquad (2.1)$$

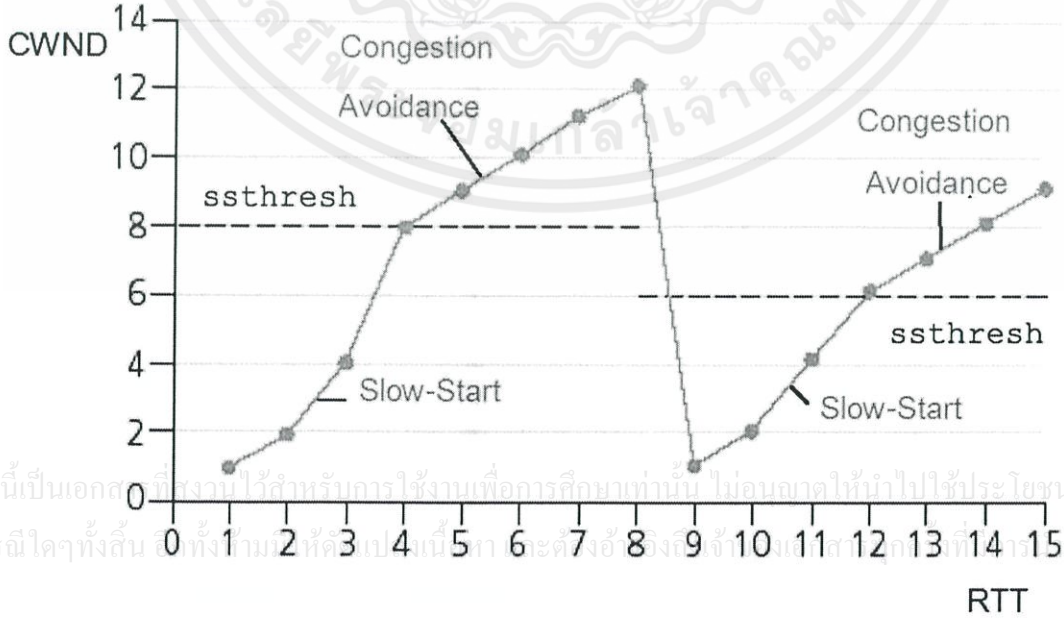
เมื่อ cwnd new คือปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้ครั้งใหม่

SegmentSize คือจำนวนแพ็คเก็ตที่ผู้ส่งสามารถส่งไปได้

cwnd old คือปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้ครั้งเดิม

จากสมการจะเห็นได้ว่าค่า cwnd new จะมีการเพิ่มขึ้นอย่างช้า ๆ ตามความจุของความกว้างของช่องทางในการส่งข้อมูล

ปัญหาคือ Tahoe ใช้ขนาดของ Timeout ในการตรวจพบการสูญเสียของแพ็คเก็ต ซึ่งใช้เวลานาน แทนที่จะตรวจพบได้ทันที แต่ Tahoe ใช้การตอบรับการร้องขอแบบการสะสม (Cumulative Acknowledment) ซึ่งมาจากการเข้าถึงโดยใช้วิธี Go Back N ดังนั้นเมื่อเกิดแพ็คเก็ตที่สูญหาย Tahoe จะทำการรอ Timeout



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 กราฟแสดงการทำงานของอัลกอริทึม Slow Start และอัลกอริทึมการหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่าย

3. อัลกอริทึม Fast Retransmit

อัลกอริทึม Fast Retransmit เป็นอัลกอริทึมที่เกิดขึ้นหลังจากที่ผู้ส่งได้ส่งแพ็คเก็ตไปแล้ว ผู้รับรอรับแพ็คเก็ตในช่วงเวลาหนึ่ง (A Period of Time : RTO) ในช่วงเวลาดังกล่าวนั้นถ้าผู้รับได้รับแพ็คเก็ตที่ถูกต้อง ผู้รับจะส่ง ACK ตอบกลับไป แต่ถ้าแพ็คเก็ตที่ได้รับมีหมายเลข Sequence ที่ไม่ถูกต้องตรงตามที่ร้องขอไป ผู้รับจะทำการส่ง ACK ติดต่อกันสามครั้งหรือมากกว่านั้นเรียกว่า Duplicate ACKs ซึ่งอาจหมายความได้ว่าแพ็คเก็ตมีความสูญเสียหรือเกิดการสูญหาย ทีซีพีจะทำการส่งแพ็คเก็ตที่สูญเสียหรือสูญหายนั้นไปใหม่ทันทีโดยไม่รอให้หมดช่วงเวลาที่กำหนดไว้ (RTO)

2.3.2.2 TCP Reno

Reno มีหลักการพื้นฐานของ Tahoe และมีการเพิ่มความสามารถคืออัลกอริทึม Fast recovery โดยหากมีแพ็คเก็ตที่สูญหายก็จะถูกตรวจพบก่อนโดยไม่ต้องรอ Timeout และไปป์ไลน์ (Pipeline) จะไม่ถูกทำให้ว่างเมื่อพบแพ็คเก็ตที่สูญหาย

Reno ต้องการให้ผู้ส่งได้รับทราบทันทีเมื่อแพ็คเก็ตสูญหาโดยใช้อัลกอริทึม Fast Retransmit คือเมื่อใดก็ตามที่ผู้ส่งได้รับ ACK ซ้ำ 3 ครั้งของแพ็คเก็ตเดียวกันให้ถือว่าแพ็คเก็ตสูญหายหรือมีความเสียหาย จากนั้นให้ผู้ส่งทำการส่งแพ็คเก็ตใหม่อีกครั้งโดยไม่ต้องรอ Timeout

ประสิทธิภาพการทำงานของ Reno ดีกว่า Tahoe เมื่อมีแพ็คเก็ตที่สูญหายหรือมีความเสียหายแค่ 1 แพ็คเก็ต แต่ถ้าหากมีการสูญหายของแพ็คเก็ตจำนวนมาก Reno จะมีประสิทธิภาพแทบจะเหมือนกับ Tahoe เนื่องจาก Reno สามารถตรวจสอบการสูญหายของแพ็คเก็ตได้เพียง 1 แพ็คเก็ต

อัลกอริทึมใน Reno มีดังนี้

1. อัลกอริทึม Fast Recovery

หลังจากเสร็จสิ้นอัลกอริทึม Fast Retransmit ซึ่งได้ส่งเซกเมนต์ที่หายไปถือเป็นการแสดงถึงการหลีกเลี่ยงความคับคั่งของข้อมูลแต่ไม่ได้แสดงถึงการทำงานของอัลกอริทึม Slow start อัลกอริทึม Fast recovery จะทำการปรับปรุงให้มีปริมาณงานที่ทำในช่วงเวลาหนึ่งของการส่งข้อมูล (Throughput) ระดับสูงภายใต้การควบคุมความคับคั่งภายในเครือข่ายของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างข้อมูลขนาดใหญ่ เหตุที่อัลกอริทึม Fast Retransmit ไม่แสดงถึงการทำงานของอัลกอริทึม Slow Start เพราะการได้รับ Duplicate ACKs บ่งบอกถึงทีซีพีมากกว่าที่จะบอกแค่ว่าแพ็คเก็ตเกิดการสูญหาย ถ้าผู้รับสร้าง Duplicate ACKs ในขณะที่รับเซกเมนต์อื่น เซกเมนต์นั้นจะออกจากเครือข่ายไปอยู่ในที่พักข้อมูลของฝั่งผู้รับแทน ซึ่งหมายความว่าข้อมูลยังคงไหลอยู่ระหว่างสองฝั่งผู้รับและผู้ส่งและทีซีพีไม่ต้องการที่จะลดอัตราการไหลอย่างรวดเร็วผ่านอัลกอริทึม Slow Start

2. อัลกอริทึม Fast Retransmit และ Fast Recovery มักจะทำงานร่วมกัน ดังนี้

เมื่อฝั่งผู้ส่งได้รับ Duplicate ACKs ซ้ำกันสามครั้งจะทำการตั้งค่า ssthresh ให้เหลือครึ่งหนึ่งของขนาดหน้าต่าง cwnd ในปัจจุบันแต่ไม่น้อยกว่า 2 เซกเมนต์ จากนั้นผู้ส่งจะส่งเซกเมนต์ที่เกิดความผิดพลาดไปใหม่อีกครั้งและขยายขนาดของ cwnd ไปจนถึงค่า ssthresh+3 การเพิ่มขนาดของปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้โดยอ้างอิงจากจำนวนของเซกเมนต์ที่ถูกนำออกไปจากเครือข่ายที่นำไปเก็บไว้

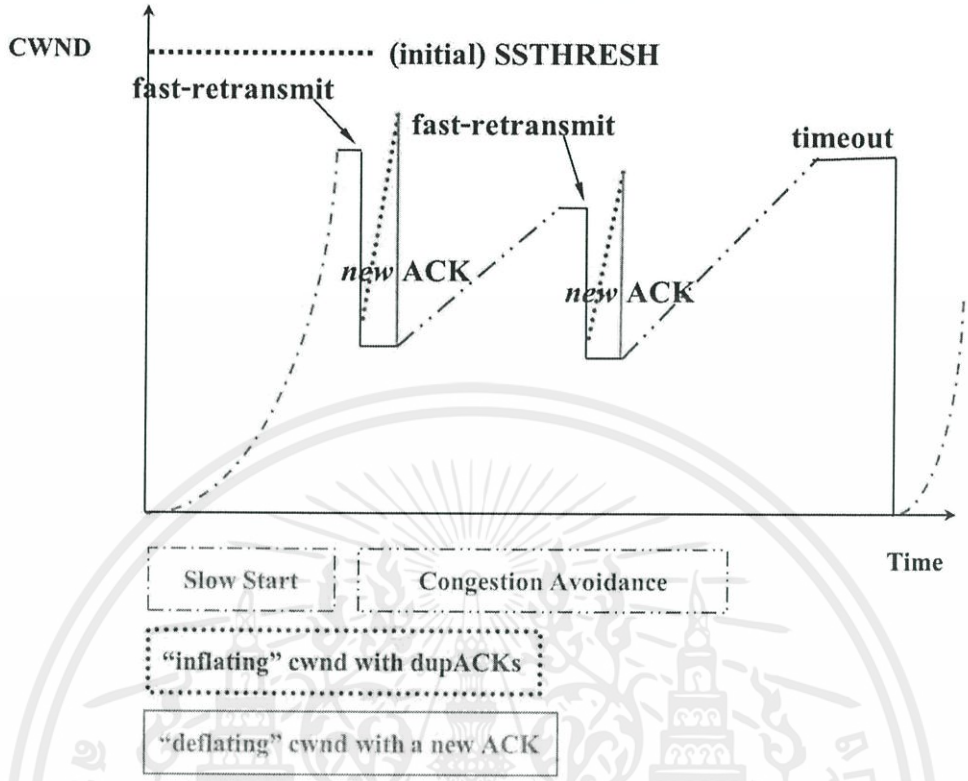
แต่ละครั้งที่ Duplicate ACKs อื่น ๆ มาถึงฝั่งผู้ส่งจะมีการเพิ่ม cwnd ตามขนาดของเซกเมนต์ซึ่งถือเป็นการขยายขนาดของปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้เพื่อรองรับเซกเมนต์อื่น ๆ ที่จะถูกนำออกไปจากเครือข่าย จากนั้นผู้ส่งจะส่งแพ็คเก็ตเมื่อได้รับอนุญาตจากค่า ปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้ใหม่

ถ้าหากเกิด Timeout จะกลับไปเริ่มขั้นตอนของอัลกอริทึม Slow Start ใหม่อีกครั้ง โดยตั้งค่าให้เป็นไปตามสมการและรูปที่ 2.7 ดังนี้

$$ssthresh = cwnd / 2 \tag{2.2}$$

และให้ค่า cwnd มีค่าเท่ากับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 กราฟแสดงการทำงานของ Fast Retransmit และ Fast Recovery

2.3.2.3 TCP New Reno

New Reno เป็นการนำหลักการของ Reno มาปรับปรุงโดยสามารถป้องกันการสูญเสียของแพ็คเก็ตได้เป็นจำนวนมากและมีประสิทธิภาพมากกว่า Reno โดยสามารถทำอัลกอริทึม Fast Retransmit เมื่อได้รับแพ็คเก็ต Duplicate เป็นจำนวนมาก นอกจากนี้ New Reno ยังมีความแตกต่างกับ Reno ตรงที่ New Reno จะไม่ทำอัลกอริทึม Fast Recovery กับข้อมูลทั้งหมดแต่ข้อมูลจะถูกดึงออกมาในขณะนั้นโดยการถูก ACK ซึ่งหมายความว่า New Reno จะช่วยลดปัญหาของ Reno ในเรื่องการลดอัตราการเพิ่มของปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้ (cwnd) แต่ส่วนของการส่งข้อมูลอย่างรวดเร็วจะมีลักษณะเหมือน Reno

โดยการทำงานของ Reno จะมีลักษณะโดยผู้ส่งจะส่งข้อมูลกลับไปใหม่หลังจากหมดเวลาการส่งข้อมูลไปใหม่หรือเกิด Duplicate ACKs ขึ้นสามครั้ง จากนั้นจะเข้าสู่การทำอัลกอริทึม Fast Retransmit โดยเมื่อหมดเวลาการส่งข้อมูลจะสามารถทำได้ทีละหลาย ๆ แพ็คเก็ตแต่ Fast Retransmit จะสามารถส่งแพ็คเก็ตกลับไปใหม่ได้เพียงครั้งละ 1 แพ็คเก็ตเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

New Reno ปรับปรุงโดยจะมีการรอแพ็คเก็ตที่ส่งมาใหม่โดยขึ้นอยู่กับการส่งการร้องขอมา (Partial ACK) ในขั้นตอนของอัลกอริทึม Fast Recovery ของผู้ส่งโดยเมื่อได้รับ Partial ACK จะสรุปว่าแพ็คเก็ตสูญหายและจะทำการส่งแพ็คเก็ตนั้นไปใหม่อีกครั้งหนึ่ง

ขั้นตอนของ New Reno มีดังนี้

1. เมื่อได้รับ Duplicate ACKs สามครั้งและผู้ส่งไม่อยู่ในอัลกอริทึม Fast Recovery จะมีการตั้งค่า ssthresh ตามสมการ

$$ssthresh = \max(FlightSize / 2, 2*mss) \quad (2.3)$$

เมื่อ FlightSize คือ ข้อมูลทั้งหมดที่ส่งไปแล้วแต่ยังไม่ได้รับ ACK

mss คือ Maximum Segment Size

จากนั้นทำการบันทึกหมายเลข Sequence ที่สูงที่สุด ที่ส่งไปในตัวแปร “recover”

2. ส่งเซกเมนต์ที่เกิดความเสียหายไปใหม่อีกครั้งหนึ่งและตั้งค่า cwnd ให้มีค่าเท่ากับ

$$cwnd = ssthresh + (3*mss) \quad (2.4)$$

3. เมื่อได้รับ Duplicate ACK แต่ละครั้งจะเพิ่มค่า cwnd และ mss
4. ถ้าได้รับการร้องขอใหม่ (ACK) การร้องขอนั้นอาจมาจากการส่งเซกเมนต์ไปใหม่อีกครั้งในขั้นตอนที่ 2 หรือมาจากการส่งเซกเมนต์ไปใหม่อีกครั้งในครั้งถัดไปก็ได้

2.3.2.4 TCP Vegas

Vegas เป็นการใช้งานทีซีพีโดยมีการปรับปรุงมาจาก Reno ซึ่งพยายามหลีกเลี่ยงปัญหา Timeout โดยใช้อัลกอริทึมตรวจสอบ Timeout ทุก ๆ รอบการส่งข้อมูลและมีการแก้ไขปัญหา Duplicate ACKs เพื่อตรวจสอบแพ็คเก็ตที่สูญหาย นอกจากนี้ยังปรับปรุงแก้ไขอัลกอริทึม Slow Start เพื่อป้องกันไม่ให้เกิดความคับคั่งของข้อมูลในเครือข่ายที่ไม่ได้ขึ้นอยู่กับการสูญเสียแพ็คเก็ตเพียงอย่างเดียวและสามารถตรวจจับความคับคั่งของข้อมูลได้ก่อนที่แพ็คเก็ตจะสูญหาย ซึ่งยังคงมีกลไกการดูแลรักษาอื่น ๆ ของ Reno และ Tahoe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึมของ Vegas มีดังนี้

1. New Retransmission Mechanism

Vegas มีการพัฒนาจากกลไกการส่งเซกเมนต์กลับไปใหม่อีกครั้งของ Reno โดยติดตามการส่งข้อมูลในแต่ละเซกเมนต์และคำนวณประมาณค่าของรอบเวลาการส่งเซกเมนต์ (Round Trip Time : RTT) ซึ่งดูจากเวลาที่ได้รับการร้องขอ ถ้าได้รับ Duplicate ACK จะตรวจสอบว่าเวลาของการส่งเซกเมนต์ปัจจุบันมีค่ามากกว่าค่าประมาณรอบเวลาหรือไม่ หากมีค่ามากกว่าจะทำการส่งเซกเมนต์กลับไปใหม่ทันทีโดยไม่ต้องรอการตอบกลับของ Duplicate ACK ครบสามครั้งหรือรอหมดเวลา Timeout และแก้ปัญหาที่ Reno ซึ่งไม่สามารถตรวจจับแพ็คเก็ตที่สูญหายเมื่อหน้าต่างมีขนาดเล็กโดยมีกระบวนการการทำงานต่าง ๆ ดังนี้

วัดค่ารอบเวลาในการส่งเซกเมนต์ทุก ๆ เซกเมนต์ที่ผู้ส่งส่งออกไป จากนั้นคำนวณหาคาบเวลา Timeout และเมื่อได้รับ Duplicate ACK จะตรวจสอบค่าเวลา Timeout หากหมดเวลาจะทำการส่งเซกเมนต์กลับไปใหม่ทันทีโดยไม่รอการตอบกลับของ Duplicate ACK

หากมีการสูญเสียแพ็คเก็ตเป็นจำนวนมากและมีการทำ Fast Retransmit มากกว่าหนึ่งครั้ง Vegas จะทำการลดปริมาณขนาดหน้าต่างของข้อมูลที่ผู้ส่งสามารถส่งไปได้ (cwnd) เฉพาะรอบการทำอัลกอริทึม Fast Retransmit ในครั้งแรก

2. อัลกอริทึมการหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่าย (Congestion Avoidance)

Vegas จะแตกต่างกับการทำงานอื่น ๆ ในอัลกอริทึมการหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่ายโดย Vegas ไม่ได้ใช้หลักการของการที่เมื่อแพ็คเก็ตสูญหายให้สรุปว่ามีความคับคั่งของข้อมูลภายในเครือข่ายแต่ให้อัตราการส่งที่ลดลงเป็นตัวกำหนดความคับคั่งของข้อมูลเมื่อเทียบกับอัตราที่คาดหวัง เป็นการสร้างคิว (Queue) ขนาดใหญ่ในเราท์เตอร์ โดยจะใช้รูปแบบที่ Wang และ Crowcroft เสนอ

เมื่อใดก็ตามที่อัตราที่คำนวณเกินจากอัตราที่คาดหวังไว้ Vegas จะใช้ขนาดความกว้างของช่องทางในการส่งข้อมูลที่มีการเพิ่มอัตราการส่งและเมื่อใดก็ตามที่อัตราการคำนวณใกล้เคียงกับอัตราที่คาดหวังไว้ การส่งก็จะลดลงเพื่อป้องกันไม่ให้เกินขนาดความกว้างของช่องทางในการส่งข้อมูล Vegas จึงมีประสิทธิภาพในการรับมือกับความคับคั่งของข้อมูลและไม่เสียความกว้างของช่องทางในการส่งข้อมูลในการส่งที่สูงเกินอัตราการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

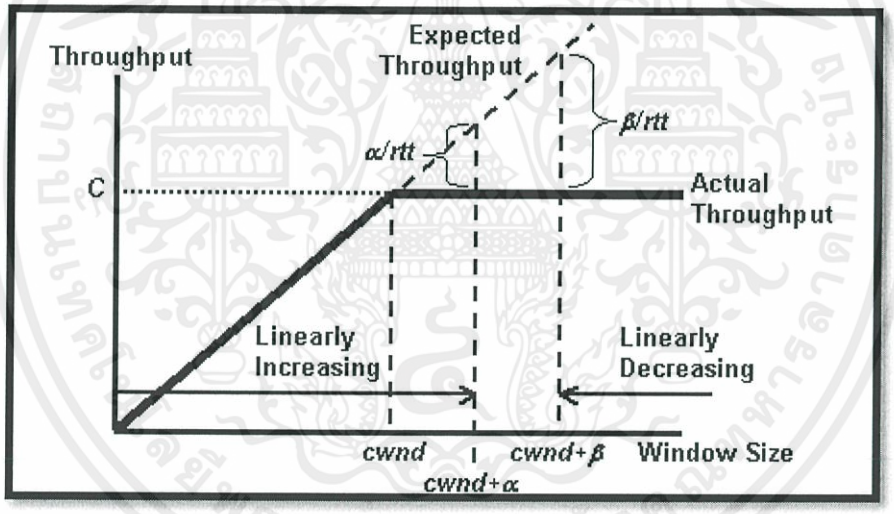
Vegas จะทำการเพิ่ม cwnd อย่างต่อเนื่องในอัลกอริทึมการหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่ายโดยจะมีการตรวจสอบความคับคั่งของข้อมูลจากการเปรียบเทียบ

ปริมาณการทำงานของทีซีพีในช่วงเวลาหนึ่ง (Throughput) โดยหากค่าปริมาณการทำงานของทีซีพีในช่วงเวลาหนึ่งที่พิจารณา มีค่าน้อยกว่าค่าปริมาณการทำงานของทีซีพีในช่วงเวลาหนึ่งที่คาดหวัง Vegas จะทำการลดขนาดของหน้าต่างผู้รับลง แต่หากค่าปริมาณการทำงานของทีซีพีในช่วงเวลาหนึ่งที่พิจารณามีค่าใกล้เคียงกับค่าที่คาดหวังไว้ จะเพิ่มขนาดของหน้าต่างโดยมีการคำนวณค่าต่าง ๆ ตามสมการและรูปที่ 2.9

$$\text{Diff} = (\text{Expected} - \text{Actual})\ (\text{BaseRTT} / \text{MinimumRTT}) \quad (2.3)$$

$$\text{Expected flow rate} = \text{CWND} / \text{BaseRTT} \quad (2.4)$$

$$\text{Actual flow rate} = \text{CWND} / \text{ActualRTT} \quad (2.5)$$



รูปที่ 2.9 กราฟแสดงการทำงานของ TCP Vegas

3. Modified Slow-start

เป็นการตรวจสอบค่ารอบการทำงานของทีซีพีในคาบเวลาการส่งข้อมูล ว่าเมื่อใดควรจะหยุดการเพิ่มค่า cwnd แบบ Exponential และให้เข้าสู่อัลกอริทึมการหลีกเลี่ยงความคับคั่งของข้อมูลภายในเครือข่าย (Congestion Avoidance)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.5 TCP Sack

การเลือกการตอบรับ (Selective acknowledgments : Sack) สามารถตรวจจับเซกเมนต์ที่สูญหายได้หลายเซกเมนต์และสามารถส่งข้อมูลกลับไปใหม่ได้มากกว่าหนึ่งเซกเมนต์ที่สูญหายต่อหนึ่งคาบเวลาในการส่งข้อมูล โดย SACK ยังคงมี คุณสมบัติของอัลกอริทึม Slow Start และ Fast Retransmit เหมือน Reno และมีการตรวจสอบ Timeout เหมือน Tahoe

ทีซีพีมีข้อจำกัดในเรื่องของ ACK ที่เกิดการสะสมเป็นจำนวนมากทำให้ผู้ส่งรู้ถึงการสูญเสียของเซกเมนต์แค่หนึ่งแพ็คเก็ตที่เสียหายต่อหนึ่งคาบเวลาในการส่งข้อมูลและมีข้อจำกัดในเรื่องที่ผู้ส่งส่งเซกเมนต์ไปให้ฝั่งผู้รับใหม่อีกครั้งแต่เซกเมนต์ที่ส่งไปใหม่นั้นอาจถึงผู้รับไปก่อนแล้ว

กลไกของ SACK เป็นไปตามหลักการทำงานของการเลือกส่งข้อมูลซ้ำอีกครั้งหนึ่ง (Selective Repeat Retransmission) ทำให้ผู้รับสามารถส่งเซกเมนต์ Sack เพื่อแจ้งให้ผู้ส่งทราบว่าเซกเมนต์ไหนบ้างที่ได้รับแล้วผู้ส่งจึงส่งแค่เซกเมนต์ที่หายไปใหม่อีกครั้งหนึ่ง

เซกเมนต์ที่สูญเสียไปจำนวนมากส่งผลกระทบที่รุนแรงต่อปริมาณการใช้งานของทีซีพีในช่วงเวลาหนึ่ง ทีซีพีจึงใช้การตอบรับการร้องขอที่สะสมกันเป็นจำนวนมากเพื่อวางแผนสำหรับเซกเมนต์ที่ได้รับแล้ว แต่ไม่ใช่เซกเมนต์ขอบซ้ายของหน้าต่างที่ได้รับ ไม่ให้ถูกส่งการร้องขอไปอีกซึ่งถือเป็นการบังคับผู้ส่งให้รอภายในคาบเวลาการส่งข้อมูล (RTT) เพื่อค้นหาเซกเมนต์ที่สูญเสียแต่ละเซกเมนต์หรือเพื่อค้นหาเซกเมนต์ที่ถูกส่งไปซ้ำโดยไม่จำเป็นเพราะเซกเมนต์นั้นได้ถูกรับอย่างถูกต้องแล้ว

SACK เป็นกลยุทธ์ที่แก้ไขปรับปรุงประสิทธิภาพเมื่อเกิดการสูญเสียเซกเมนต์เป็นจำนวนมากโดยการเลือกการส่ง ACK ของเซกเมนต์ซึ่งผู้รับจะบอกให้ผู้ส่งทราบถึงเซกเมนต์ที่ได้รับเรียบร้อยแล้วดังนั้นผู้ส่งจึงส่งเซกเมนต์ซ้ำอีกครั้งแค่เซกเมนต์ที่สูญหาย

โพรโทคอลทรานสปอร์ตจำนวนมาก เช่น NETBLT, XTP, BDP, NADIR และ VMTP เป็นต้น มีการนำ SACK มาใช้โดย SACK จะนำหลักการของทีซีพีมาใช้ 2 เงื่อนไข (Options) เงื่อนไขแรกคือเงื่อนไข SACK - Permitted เป็นการส่งเซกเมนต์ SYN ไปเพื่อบอกให้อีกฝ่ายทราบว่ามีการใช้เงื่อนไข SACK ในการเชื่อมต่อ และเงื่อนไขที่สองคือเงื่อนไข SACK ซึ่งเป็นการส่งการเชื่อมต่อที่ถูกอนุญาตภายใต้เงื่อนไข SACK - Permitted

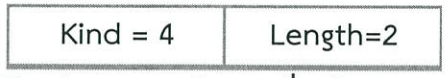
เงื่อนไข SACK ได้ถูกรวมไปในบทบาทของการส่งเซกเมนต์ของผู้ส่งไปยังผู้รับโดยจะใช้หลักการของการไหลของข้อมูลแบบทิศทางเดียว (Simplex)

1. เงื่อนไข Sack-Permitted

เงื่อนไข Sack–Permitted มีทั้งหมด 2 ไบต์ โดยสองไบต์นี้จะถูกส่งโดยเซกเมนต์ SYN ของทีซีพี จากนั้นการเชื่อมต่อของเงื่อนไข SACK จะถูกเปิดการเชื่อมต่อแต่ยังไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

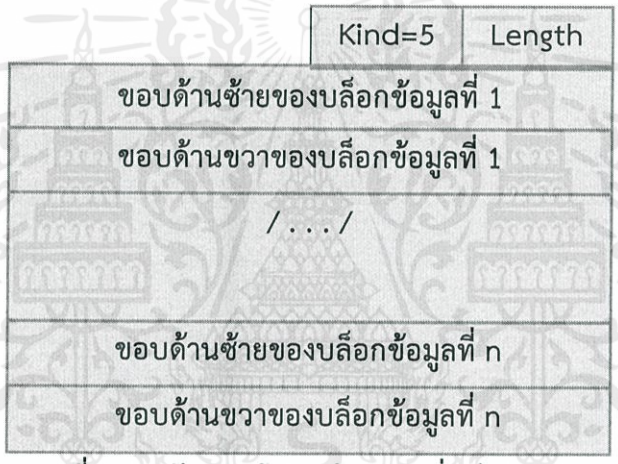
อนุญาตให้ส่งเซกเมนต์ที่ไม่ใช่เซกเมนต์ SYN ตัวอย่างเงื่อนไข Sack–Permitted เป็นไปดังแสดงในรูปที่ 2.10



รูปที่ 2.10 ตัวอย่างของการกำหนดเงื่อนไข Sack-Permitted

2. เงื่อนไข Sack

เงื่อนไข Sack จะถูกใช้ในการส่งข้อมูลการร้องขอจากผู้รับไปถึงผู้ส่งซึ่งก็คือเซกเมนต์ ACK โดยอยู่ในการเชื่อมต่อของทีซีพี ตัวอย่างเงื่อนไข Sack เป็นไปดังแสดงในรูปที่ 2.11



รูปที่ 2.11 ตัวอย่างโครงสร้างของเงื่อนไข Sack

ผู้ส่งจะส่งเงื่อนไข Sack ไปแจ้งผู้รับด้วยบล็อกของข้อมูลที่ไม่ติดกัน ซึ่งผู้รับที่ได้รับจะเก็บไว้ในคิว ทางฝั่งผู้รับจะรอการรับข้อมูลเซกเมนต์ ACKที่มาจากผู้ส่งเพื่อมาเติมเต็มช่องว่างที่เก็บลำดับของข้อมูลซึ่งอยู่ระหว่างบล็อกที่ได้รับ

เมื่อผู้รับได้รับเซกเมนต์ที่ขาดหายไปฝั่งผู้รับจะทำการส่งการร้องขอข้อมูลปกติ โดยเข้าไปที่ Acknowledgement Number Field ของ TCP header ซึ่งอยู่ภายในขอบซ้ายของหน้าต่าง

จะเห็นว่าเงื่อนไข Sack จะไม่ทำการเปลี่ยนแปลงใด ๆ กับความหมายของ Acknowledgement Number Field และเงื่อนไข Sack จะประกอบไปด้วยบล็อกบางบล็อกของช่องว่างที่เก็บลำดับของข้อมูลที่มีความต่อเนื่องกันที่ผู้รับได้รับและเก็บไว้ในคิวของหน้าต่าง โดยแต่ละบล็อกที่เก็บไว้ในคิวของผู้รับจะมีการกำหนดรูปแบบของเงื่อนไข Sack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นแบบเลขจำนวนเต็มแบบไม่กำหนดสัญลักษณ์ที่มีขนาด 32 บิตตามลำดับของ ชนิดของเครือข่ายดังนี้

ขอบด้านซ้ายของบล็อกข้อมูลคือหมายเลขลำดับแรกของบล็อก

ขอบด้านขวาของบล็อกข้อมูลคือหมายเลขลำดับที่อยู่ถัดจากหมายเลขลำดับสุดท้ายของบล็อก

เงื่อนไข Sack ที่มีจำนวน n บล็อกจะมีความยาวเป็น 8*n+2 ไบต์ ดังนั้นเงื่อนไขของทีซีพีที่มีความยาวได้สูงสุด 40 ไบต์ จึงมีจำนวนบล็อกได้ทั้งหมด 4 บล็อก ขณะที่เงื่อนไข Sack แจ้งเตือนไปที่ผู้ส่งว่าฝั่งผู้รับได้รับเซกเมนต์ที่ต้องการแล้ว ผู้รับจะอนุญาตให้นำข้อมูลที่ตามมาทิ้งไป ซึ่งจะทำการรายงานไปยังเงื่อนไข Sack ด้วย

ข้อกำหนดต่างๆของเงื่อนไข Sack มีดังนี้

1. การสร้างเงื่อนไข Sack จากฝั่งผู้รับข้อมูล

เมื่อผู้รับได้รับเซกเมนต์ SYN จากเงื่อนไข Sack – Permitted สำหรับการเชื่อมต่อแล้วผู้รับจะสร้างเงื่อนไข Sack ภายใต้การอนุญาตของเงื่อนไข Sack-Permitted โดยถ้ามีการส่งเงื่อนไข Sack จะถูกรวมเข้ากับเซกเมนต์ ACK ทั้งหมดที่ไม่ใช่เซกเมนต์ ACK ของหมายเลขลำดับข้อมูลที่สูงสุดที่อยู่ภายในคิวของฝั่งผู้รับ

ในสถานการณ์ที่เครือข่ายเกิดความขัดข้องหรือข้อมูลสูญเสียการเรียงลำดับ เช่น มีข้อมูลที่ไม่ต่อเนื่องกันภายในคิวของผู้รับ ฝั่งผู้รับควรจะส่งเซกเมนต์ ACK ไปทุก ๆ เซกเมนต์ที่มาพร้อมกับข้อมูลใหม่และแต่ละเซกเมนต์ของ Duplicate ACK ควรจะสนับสนุนเงื่อนไข Sack ด้วย ถ้าผู้รับเลือกที่จะส่งเงื่อนไข Sack ต้องทำตามกฎข้อบังคับดังนี้

บล็อกแรกของ Sack ต้องระบุบล็อกที่ต่อเนื่องกันของข้อมูลที่ประกอบไปด้วยเซกเมนต์ที่ทำให้เกิดเซกเมนต์ ACK ขึ้น ถ้าเซกเมนต์นั้นไม่เข้าไปใน Acknowledgement Number Field ภายใน Header ที่สามารถรับรองได้ว่าเซกเมนต์ ACK ที่มากับเงื่อนไข Sack จะส่งผลให้เกิดการเปลี่ยนแปลงภายในที่พักข้อมูลของฝั่งผู้รับอย่างรวดเร็วที่สุด นอกจากนี้ผู้รับควรจะรวมบล็อกของ Sack ที่มีความแตกต่างกันชัดเจนภายในเงื่อนไข Sack โดยมีข้อจำกัดว่าค่าที่มากที่สุดที่เป็นไปได้ของพื้นที่ว่างอาจไม่เพียงพอที่จะรายงานบล็อกทั้งหมดที่อยู่ในคิวของผู้รับ และเงื่อนไข Sack ควรจะถูกขยายออกโดยการรายงานของบล็อกของ Sack ซ้ำ ๆ อย่างรวดเร็วรวมถึงการที่เงื่อนไข Sack ถูกสร้างขึ้นด้วย

ในกระบวนการที่เป็นไปอย่างปกติผู้รับจะรายงานเซกเมนต์ใด ๆ ที่เป็นส่วนหนึ่งของบล็อกข้อมูลที่ไม่ต่อเนื่องกันอย่างน้อย 3 เงื่อนไข Sack ที่สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่เงื่อนไข Sack มีการรายงานบล็อกของเซกเมนต์ที่ได้รับมาอย่างรวดเร็วที่สุดถือเป็นเรื่องสำคัญเป็นอย่างมากเพราะจะทำให้ผู้ส่งจัดลำดับข้อมูลที่หมดอายุภายในเครือข่ายและผู้รับสามารถจัดคิวได้อย่างถูกต้อง

2. กระบวนการของเงื่อนไข Sack และการส่งข้อมูลกลับไปใหม่อีกครั้งหนึ่ง

เมื่อผู้ส่งได้รับเซกเมนต์ ACK ที่มีเงื่อนไข Sack อยู่ด้วย ผู้ส่งจะบันทึกการเลือกการส่งการร้องขอข้อมูล (Selective Acknowledgement) ไว้สำหรับการอ้างอิงในอนาคตและจะมีคิวที่เก็บการส่งเซกเมนต์กลับไปใหม่อีกครั้ง ซึ่งมีเซกเมนต์ที่ถูกส่งมาแล้วแต่ยังไม่ถูกส่งเซกเมนต์ ACK กลับไปโดยเรียงตามหมายเลขลำดับข้อมูล ถ้าผู้ส่งทำการส่งเซกเมนต์กลับไปใหม่ก่อนที่จะทำการส่งขอบเขตของบล็อกภายในเงื่อนไข Sack ไปใหม่ให้ฝั่งผู้ส่ง จะไม่ชนกับขอบเขตบล็อกของเซกเมนต์ในคิวที่เก็บการส่งเซกเมนต์กลับไปใหม่อีกครั้ง

ดังนั้นแต่ละเซกเมนต์ในคิวที่เก็บการส่งเซกเมนต์กลับไปใหม่อีกครั้ง จะมีบิต Flag ที่ถูกตั้งค่าใหม่ซึ่งเรียกว่าบิต SACKed เพื่อบอกว่าเซกเมนต์นั้นเป็นเงื่อนไข Sack

เมื่อผู้ส่งได้รับเซกเมนต์ที่บรรจุเงื่อนไข Sack แล้วผู้ส่งจะตั้งค่าบิต SACKed สำหรับเซกเมนต์ที่ถูกเลือกให้ส่งเซกเมนต์ ACK กลับไป นอกจากนี้แต่ละบล็อกที่อยู่ในเงื่อนไข Sack ผู้ส่งจะตั้งค่า SACKed Flag ให้ทำงานสำหรับเซกเมนต์ที่อยู่ภายใน คิวที่เก็บการส่งเซกเมนต์กลับไปใหม่อีกครั้ง ซึ่งจะรวมไปในบล็อกนั้น ๆ หลังจากบิต SACKed ถูกเปิด ผู้ส่งจะข้ามเซกเมนต์นั้น ๆ ไประหว่างการส่งเซกเมนต์กลับไปใหม่ที่เกิดทีหลัง

หลังจากหมดเวลาการส่งเซกเมนต์ไปใหม่อีกครั้ง ผู้ส่งจะปิดการใช้งานของบิต SACKed ทั้งหมดและส่งเซกเมนต์ที่อยู่ตรงขอบซ้ายของหน้าต่างไปใหม่ อย่างไรก็ตามบิต SACKed จะยังคงเปิดอยู่ในเซกเมนต์นั้น เซกเมนต์จะไม่ถูกนำออกจากคิวและที่พักข้อมูลจะยังคงว่างจนกระทั่งขอบซ้ายของหน้าต่างได้เคลื่อนไปข้างหน้า

2.4 การทำงานพื้นฐานของระบบการจำลองการทำงาน

การจำลองการทำงานคือการนำเสนอหรือการจำลองลักษณะของระบบอื่น ๆ ตลอดช่วงเวลาที่สนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 พื้นฐานการจำลองการทำงานด้วยคอมพิวเตอร์

1. การจำลองการทำงานด้วยคอมพิวเตอร์ คือโปรแกรมคอมพิวเตอร์ที่จำลองลักษณะการทำงานของโปรแกรมที่สนใจตลอดเวลาหรือเฉพาะช่วงเวลาที่ให้ความสนใจ
2. ตัวแปรของโปรแกรมหรือสถานะของตัวแปร เป็นตัวแปรซึ่งใช้เป็นสื่อกลางในการแสดงถึงสถานะปัจจุบันของระบบที่จำลอง
3. โปรแกรมการจำลองการทำงานจะปรับเปลี่ยนค่าของสถานะของตัวแปรเพื่อที่จะพัฒนาแบบจำลองไปตามช่วงเวลาที่เปลี่ยนแปลงไปเรื่อยๆ
4. รูปแบบของเวลาที่มีการใช้งานในระบบการจำลองการทำงาน
   4.1 เวลาที่ระบบต้นแบบใช้ในการทำงาน (Physical time) คือเวลาที่ระบบต้นแบบหรือระบบจริงใช้ในการทำงานดังกล่าวที่จะนำมาทำเป็นแบบจำลองจริง เช่น การทำแบบจำลองเพื่อแสดงการขึ้น-ลงของเครื่องบินตั้งแต่เวลาเที่ยงวันที่ 14 มกราคม 2555 จนกระทั่งถึงเวลาเที่ยงของวันที่ 15 มกราคม 2555 ซึ่งหมายความว่าช่วงเวลาตั้งแต่เวลาเที่ยงวันที่ 14 มกราคม 2555 จนกระทั่งถึงเวลาเที่ยงของวันที่ 15 มกราคม 2555 เป็นเวลาที่ระบบต้นแบบใช้ในการทำงานจริง (Physical Time)
   4.2 เวลาที่เลื่อนไปในแบบจำลอง คือเวลาที่แบบจำลองใช้ในการทำงานเพื่อจำลองระบบซึ่งเวลาดังกล่าวจะมีความสัมพันธ์กับเวลาที่ระบบต้นแบบใช้ในการทำงานจริง เช่น เวลาในแบบจำลองเป็น 9.0 จะหมายถึงเวลา 9.00 น. ในระบบจริงและหนึ่งหน่วยของเวลาที่เลื่อนไปในแบบจำลอง (1.0) จะหมายถึงเวลาหนึ่งชั่วโมงที่เลื่อนไปในระบบจริง
   4.3 เวลาที่เลื่อนไปในความเป็นจริง (Wallclock Time) คือเวลาที่เลื่อนไปในความเป็นจริงซึ่งเหมือนกับเวลาที่แสดงโดยนาฬิกาในระหว่างที่แบบจำลองมีกำลังทำงาน เช่น แบบจำลองเริ่มต้นทำงานที่เวลา 10.00 น. จนกระทั่งถึงเวลา 12.00 น. ในวันที่ 10 มกราคม 2555
5. ลักษณะการทำงานของระบบการจำลองการทำงานเมื่อแบ่งตามเวลาในการทำงานจะมี 3 รูปแบบ คือ
   5.1 As-Fast-As-Possible Execution คือการทำงานของแบบจำลองจะทำงานไปด้วยความเร็วสูงสุดเท่าที่จะเป็นไปได้โดยไม่สัมพันธ์เวลาที่เลื่อนไปในโลกของความเป็นจริง หรือการทำงานจะไม่ขึ้นกับเวลาที่ใช้ในการทำงานแต่ยังคงสัมพันธ์กับเวลาที่ระบบต้นแบบใช้ในการทำงาน
   5.2 Real Time Execution คือการทำงานของแบบจำลองจะมีความสัมพันธ์กับเวลาจริงที่ใช้ในการทำงานของแบบจำลอง เช่น โปรแกรมหรือแบบจำลองใช้เวลาในการ

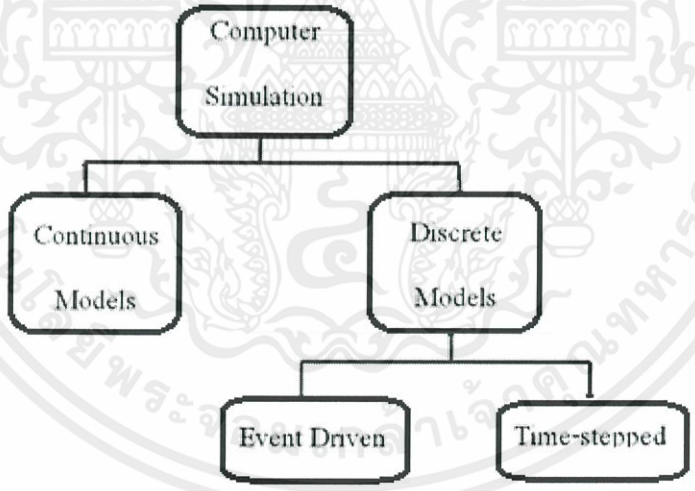
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานตามความเป็นจริงไปเป็นเวลา 15 นาที ดังนั้นเวลาที่เลื่อนไปในแบบจำลองก็จะเท่ากับ 15 นาทีด้วย

5.3 Scalable Real-Time Execution คือการทำงานจะมีความสัมพันธ์กับเวลาจริงที่ใช้ในการทำงานในลักษณะที่เป็นอัตราส่วนต่อกัน ในกรณีที่อัตราส่วนมีค่ามากกว่าหนึ่งแบบจำลองก็จะทำงานได้เร็วกว่าระบบจริง ในกรณีที่อัตราส่วนมีค่าน้อยกว่าหนึ่งแต่จะไม่ต่ำกว่า 0 แบบจำลองจะทำงานได้ช้ากว่าระบบจริงและในกรณีที่อัตราส่วนเท่ากับหนึ่งระบบจะทำงานแบบ Real-Time Execution เช่น อัตราส่วนในการทำงานเท่ากับ 2 ดังนั้นเมื่อแบบจำลองทำงานไปเป็นเวลา 15 นาที เวลาที่ถูกเลื่อนไปในแบบจำลองก็จะเท่ากับ 2 x 15 หรือ 30 นาที เป็นต้น

2.4.2 การแบ่งประเภทของการจำลองการทำงานด้วยคอมพิวเตอร์

เป็นการกล่าวถึงลักษณะการทำงานของการจำลองการทำงานด้วยคอมพิวเตอร์ซึ่งสามารถที่จะแบ่งการทำงานออกมาเป็นลักษณะหรือวิธีการทำงานย่อยได้ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 การแบ่งประเภทของการจำลองการทำงานด้วยคอมพิวเตอร์

ส่วนต่าง ๆ สามารถอธิบายได้ดังนี้คือ

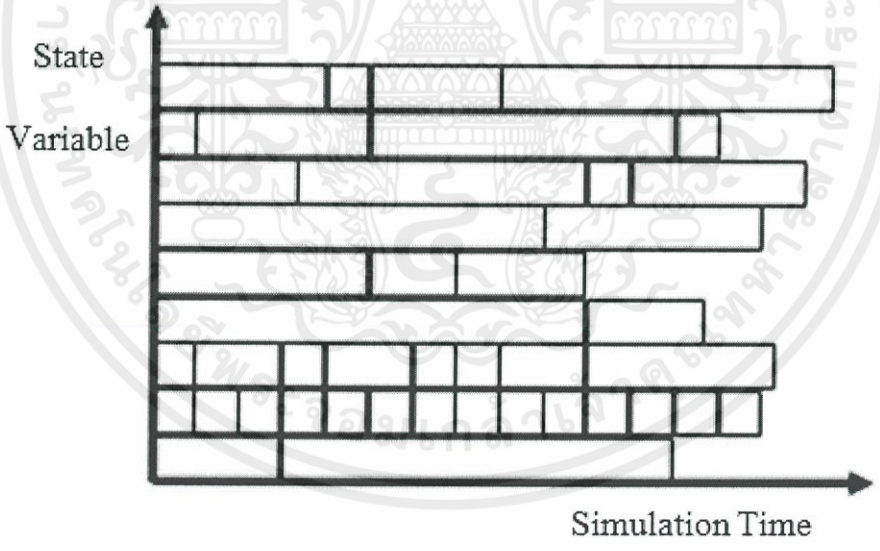
1. การจำลองการทำงานด้วยเวลาที่มีความต่อเนื่องกัน (Continuous Time Simulation) เป็นระบบที่จำลองโดยการให้ความสนใจการเปลี่ยนแปลงในแบบจำลองตลอดเวลาตั้งแต่เริ่มต้นทำงานจนสิ้นสุดการทำงาน ไม่ว่าระหว่างการทำงานของแบบจำลองจะมีการเปลี่ยนแปลงหรือไม่มีการเปลี่ยนแปลงใด ๆ เกิดขึ้นก็ตาม ซึ่งลักษณะของแบบจำลองที่

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณีใดๆทั้งสิ้น...

ได้มักจะขึ้นกับสมการที่จะนำมาใช้ในการอธิบายแบบจำลอง แบบจำลองที่เหมาะสมกับการทำงานแบบนี้มักจะเป็นแบบจำลองที่ต้องการความต่อเนื่องในการทำงาน เช่น แบบจำลองของเครื่องบินที่ใช้แสดงการบินของเครื่องบิน

2. การจำลองการทำงานที่มีความไม่ต่อเนื่องกันของเวลา (Discrete Time Simulation) เป็นระบบที่การจำลองโดยการให้ความสนใจการเปลี่ยนแปลงในลักษณะเป็นช่วงของเวลาที่ไม่ต่อเนื่อง กล่าวคือไม่ต้องเก็บข้อมูลตลอดเวลาที่จำลองการทำงาน ซึ่งสามารถแบ่งเป็นรูปแบบย่อย ๆ ได้ 2 รูปแบบ คือ

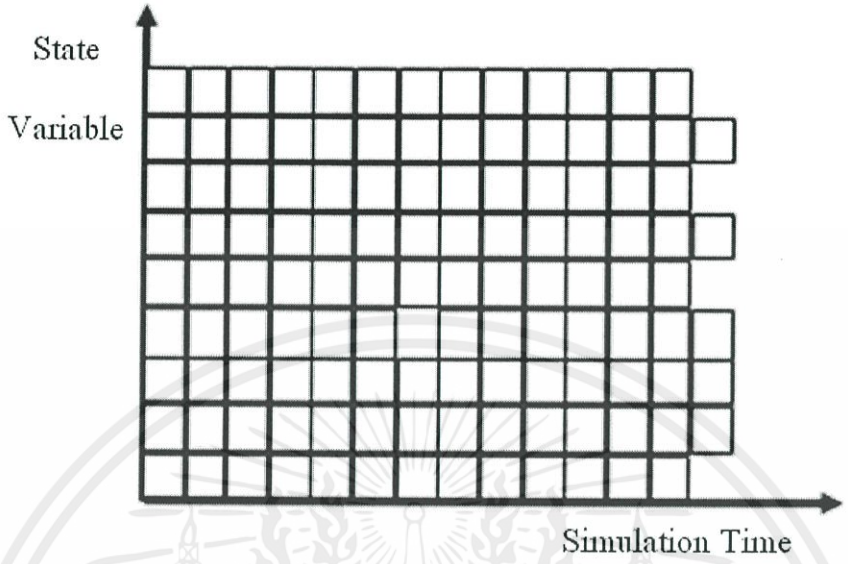
   2.1 Event Stepped จะเป็นแบบจำลองที่จะสนใจต่อเหตุการณ์ที่มีการเปลี่ยนแปลงไป ดังนั้นการเลื่อนไปของเวลาที่ใช้ในแบบจำลองจะขึ้นอยู่กับการเกิดเหตุการณ์ต่าง ๆ หากมีเหตุการณ์ใดเหตุการณ์หนึ่งเกิดขึ้นจึงจะมีการเลื่อนไปของเวลา แบบจำลองที่มีความเหมาะสม เช่น แบบจำลองของสนามบินอย่างง่ายซึ่งจะสนใจการขึ้นและลงของเครื่องบิน นั้นคือจะมีการเปลี่ยนแปลงเมื่อเครื่องบันมีการบินขึ้นและบินลงในสนามบิน ดังแสดงในรูปที่ 2.13



รูปที่ 2.13 ลักษณะการทำงานแบบ Event stepped

   2.2 Time Stepped เป็นแบบจำลองที่จะเก็บข้อมูลเป็นช่วง ๆ ของเวลา ดังนั้นเวลาที่จะใช้ในการทำงานจะมีการเลื่อนไปด้วยอัตราคงที่ซึ่งจะขึ้นอยู่กับการกำหนดของแบบจำลอง เช่น แบบจำลองการเปลี่ยนแปลงระดับน้ำของทะเลที่เก็บการเปลี่ยนแปลงทุก ๆ ครึ่งชั่วโมง ดังรูป 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ลักษณะการทำงานแบบ Time stepped

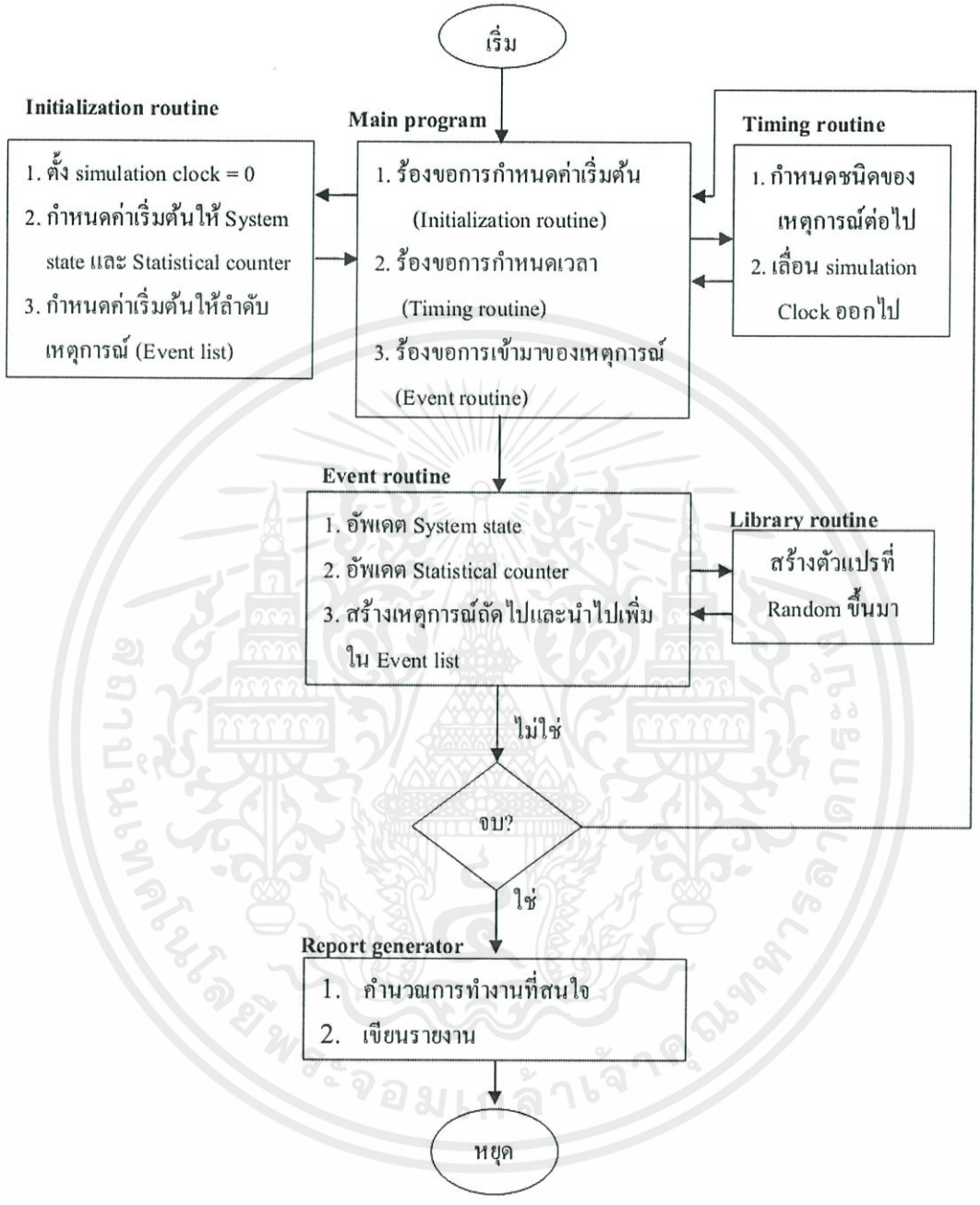
2.4.3 การจำลองการทำงานของเหตุการณ์ที่มีความไม่ต่อเนื่องกัน (Discrete Event Simulation : DES)

การจำลองการทำงานของเหตุการณ์ที่มีความไม่ต่อเนื่องกันเป็นโมเดลทางคอมพิวเตอร์สำหรับระบบใด ๆ การที่เปลี่ยนแปลงทางสถานะของระบบเกิดขึ้น ณ จุดที่ไม่ต่อเนื่องกัน และไม่มีความสัมพันธ์กันบนแกนเวลาของการจำลอง แนวคิดพื้นฐานคือระบบที่ประกอบไปด้วย

1. สถานะของระบบ (System State) ในระบบจำลองสถานะตัวแปรสถานะ
2. การเปลี่ยนแปลงสถานะในระบบจำลอง (State transitions) แทนด้วย เหตุการณ์ (Event)

สรุปคือเหตุการณ์ที่เกิดขึ้นส่งผลให้สถานะของระบบเปลี่ยนแปลง หากไม่มีเหตุการณ์ใด ๆ เกิดขึ้นสถานะของระบบในการจำลองการทำงานของเหตุการณ์ที่มีความไม่ต่อเนื่องกันก็จะไม่เปลี่ยนแปลง การประมวลผลหรือการคำนวณในการจำลองการทำงานของเหตุการณ์ที่มีความไม่ต่อเนื่องกันสามารถเปรียบได้กับลำดับของการประมวลผลเหตุการณ์ต่าง ๆ หรือลำดับของการเกิดเหตุการณ์ต่าง ๆ โดยที่แต่ละเหตุการณ์จะถูกกำหนดให้เกิดขึ้น ณ เวลาใดเวลาหนึ่งในเวลาจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แผนผังการทำงานของการจำลองการทำงานของเหตุการณ์ที่มีความไม่ต่อเนื่องกัน

1. System State คือสถานะของตัวแปรทั้งหลายที่จำเป็นต่อระบบในเวลาต่าง ๆ
2. Simulation Clock คือตัวแปรที่เก็บเวลาของการจำลองการทำงาน
3. Event List คือรายการที่เก็บเวลาถัดไปเมื่อมีเหตุการณ์เกิดขึ้น
4. Statistical Counters คือตัวแปรที่ใช้เก็บรายละเอียดสถิติเกี่ยวกับคุณสมบัติของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Initialization Routine คือโปรแกรมย่อยที่กำหนดค่าเริ่มต้นให้การจำลองการทำงานที่เวลาเท่ากับ 0
6. Timing Routine คือโปรแกรมย่อยที่กำหนดเหตุการณ์ที่จะเกิดต่อไปโดยอ้างจากรายชื่อเหตุการณ์ จากนั้นเมื่อถึงเวลาที่กำหนดไว้เหตุการณ์นั้นก็จะเริ่มทำงาน
7. Event Routine คือโปรแกรมย่อยที่อัพเดตสถานะของระบบเมื่อมีเหตุการณ์ชนิดต่าง ๆ เกิดขึ้น
8. Library Routine คือไลบรารีที่ใช้กำหนดค่าให้โปรแกรมย่อยโดยการสุ่มจากกฎของความน่าจะเป็นเพื่อใช้กำหนดเส้นทางของระบบการจำลองการทำงาน
9. Report Generator คือโปรแกรมย่อยที่คำนวณและแสดงผลค่าต่าง ๆ ที่ต้องการเมื่อการจำลองการทำงานเสร็จสิ้นลง
10. Main Program คือโปรแกรมย่อยที่ต้องการใช้ Timing Routine ในการกำหนดเหตุการณ์ต่อไปและต่อจากนั้นก็ส่งการควบคุมไปยัง Event Routine เพื่ออัพเดตข้อมูลที่เกี่ยวข้องของโดยสถานะของระบบในโปรแกรมหลักจะเป็นตัวตรวจสอบการสิ้นสุดของการจำลองการทำงานและร้องขอไปยังตัวสร้างรายงานให้แสดงรายงานเมื่อสิ้นสุดการทำงาน

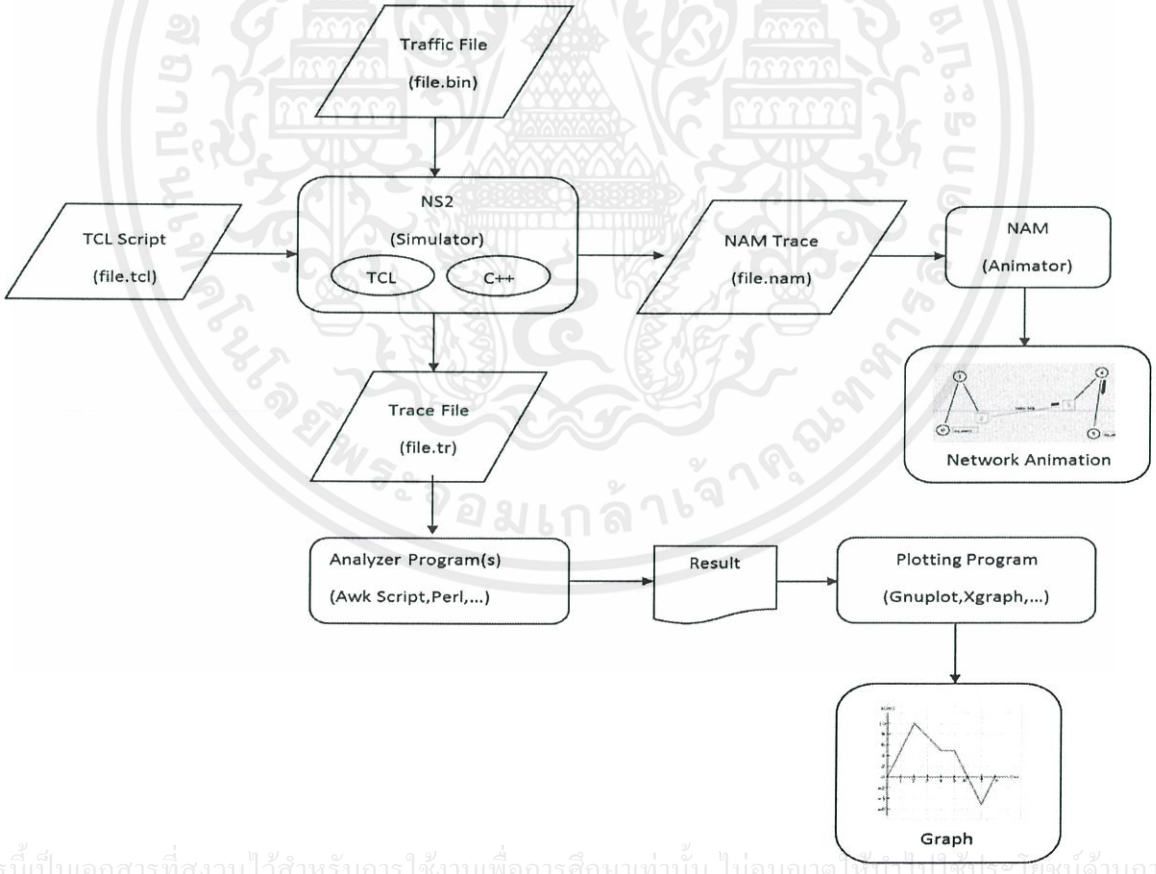
2.5 Network Simulator Version 2

เอ็นเอส-ทู (Network simulator version 2 : NS-2) เป็นซอฟต์แวร์โอเพนซอร์สที่ถูกพัฒนาขึ้นโดย ISI (Information Sciences Institute) ซึ่งรองรับระบบปฏิบัติการที่หลากหลาย ได้แก่ Linux , FreeBSD, SunOS, Solaris และ Windows เป็นต้น ซึ่งเอ็นเอส-ทูเป็นโปรแกรมที่ใช้ในการจำลองการทำงานของระบบเครือข่ายในแบบไม่ต่อเนื่องที่สนับสนุนการจำลองการเลือกเส้นทางในการขนส่งแพ็คเก็ต จำลองการทำงานของโพรโทคอลแบบมัลติคาสต์ (Multicast) และโพรโทคอล IP เช่น UDP, TCP, RTP และ SRM ที่อยู่บนเครือข่ายประเภทมีสายและไร้สาย เห็นได้ว่าเอ็นเอส-ทูเป็นซอฟต์แวร์ที่มีประโยชน์มาก ทั้งยังสนับสนุนโพรโทคอลที่หลากหลาย ใช้ในการวัดทางสถิติ เช่น ค่า Throughput, Delay, Jitter, Queue Monitoring, Drops Queues สามารถแสดงรายละเอียดของข้อมูลการจราจรในเครือข่าย (Network Traffic) ออกมาในรูปแบบของภาพเคลื่อนไหว (Network Animator) รวมทั้งยังสนับสนุนอัลกอริทึมในการค้นหาเส้นทางและจัดลำดับของคิว เช่น FIFO และ Round-Robin เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 หลักการทำงานของเอ็นเอส-ทู

ผู้ใช้งานสามารถสร้างสคริปต์ Otcl ตัวแปรทางเครือข่ายโดยภาษาซีพลัสพลัส (C++ Network Elements) และสร้างสคริปต์ Perl เพื่อที่จะนำมาใช้ในการออกแบบเพื่อจำลองการทำงานของเครือข่าย ซึ่งผู้ใช้งานสร้างตัวแปรทางเครือข่ายโดยภาษาซีพลัสพลัสขึ้นมาเพื่อให้เป็นออปเจ็ค (Object) ที่อยู่ในไลบรารีที่สามารถจะเรียกใช้ได้โดยเรียกใช้ผ่านการใช้คำสั่งของสคริปต์ Otcl ซึ่งจะมีตัวเชื่อมต่อ Otcl (Otcl Linkage) เป็นตัวที่จะเรียกออปเจ็คในไลบรารีขึ้นมาใช้งาน เมื่อสคริปต์ Otcl ถูกประมวลผลสคริปต์ Otcl จะสร้าง Nam Trace File และ NS Trace File ซึ่งตัว NAM Trace File ที่ได้นั้นจะถูกเรียกใช้โดย NAM (Network Animator) เพื่อที่จะนำไปประมวลผลแล้วแสดงผลออกมาในรูปกราฟิก ในส่วนของสคริปต์ Perl นั้นจะเรียกใช้ NS Trace File เพื่อนำ NS Trace File มาคัดกรองให้ได้ผลลัพธ์ออกมาเป็นผลลัพธ์ที่จะถูกเรียกใชโดยเครื่องมือที่แสดงผลให้เห็นภาพ เช่น Xgraph เพื่อที่จะนำมาใช้แสดงกราฟที่แสดงผลการทำงานของการจำลองเครือข่าย ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 ผังการทำงานโดยรวมของเอ็นเอส-ทู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ภาษาที่ใช้ในเอ็นเอส-ทู

ภาษาที่ใช้ในเอ็นเอส-ทูมีอยู่ 2 ภาษา เนื่องจากการจำลองการทำงานมีลักษณะการทำงาน 2 อย่างที่แตกต่างกัน ส่วนแรกจะเป็นการจำลองรายละเอียดต่าง ๆ เกี่ยวกับโพรโทคอลจึงจำเป็นต้องใช ภาษา System Programming ซึ่งมีประสิทธิภาพในการจัดการข้อมูลที่มีหน่วยของขนาดข้อมูลเป็นไบต์ Packet Header และสร้างอัลกอริทึมที่สามารถประมวลผลข้อมูลที่มีขนาดใหญ่ได้ โดยส่วนนี้จะเน้นและให้ความสำคัญในเรื่องของความเร็วในการรัน (Runtime Speed) ส่วนเรื่องของเวลาในการหมุนรอบ (Turn-Around Time) เช่น การค้นหาข้อผิดพลาด (Bug) การรันซ้ำหรือการคอมไพล์ซ้ำนั้นจะมีความสำคัญน้อยกว่า

งานในส่วนที่ 2 ส่วนใหญ่จะเป็นเรื่องของการทำวิจัยเกี่ยวกับเครือข่ายซึ่งจะต้องเกี่ยวข้องกับค่าพารามิเตอร์หรือการตั้งค่าต่าง ๆ (Configuration) ซึ่งเวลาในการทำซ้ำ (Iteration Time) เช่น การเปลี่ยนแปลงรูปแบบและการรันซ้ำ นั้นเป็นสิ่งสำคัญ ดังนั้นจึงมีความจำเป็นที่เอ็นเอส-ทูจะใช้ 2 ภาษาคือ

1. ภาษาซีพลัสพลัส (C++)

เป็นลักษณะของระดับที่ต่ำที่สามารถกำหนดอัลกอริทึมต่าง ๆ หรือสร้างโพรโทคอลใหม่ขึ้นมาได้เอง ใช้ในการสร้างโพรโทคอลและตัวแทนของแอพพลิเคชัน ใช้กำหนดคุณสมบัติของกระบวนการทำงานของแต่ละแพ็คเก็ตตามที่ผู้ใช้ต้องการ และยังใช้เปลี่ยนการทำงานของออปเจ็คในซีพลัสพลัสเพื่อเอาไว้ปรับใช้กับนโยบายที่ยังไม่มีมาตรฐาน ข้อดีของภาษาซีพลัสพลัสคือมีการประมวลผลที่เร็ว แต่ก็มีข้อเสียคือจะมีการทำงานช้าหามีการรันการจำลองการทำงาน ค้นหาและแก้ไขข้อผิดพลาด การรันซ้ำหรือการคอมไพล์ซ้ำ

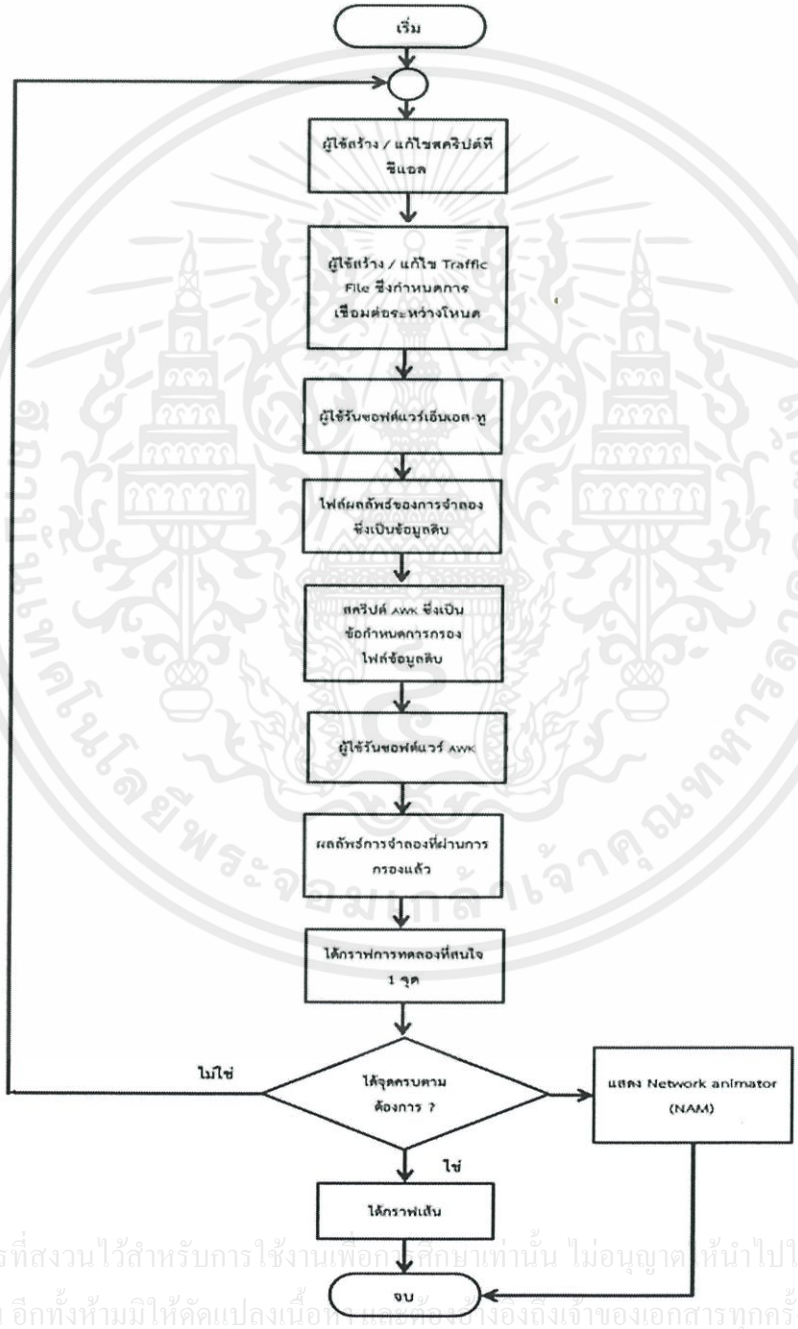
2. ภาษา Otcl (Object tool command language)

เป็นลักษณะฟอนต์เอน (Front-End) ใช้ในการเขียนการจำลองการทำงานของสคริปต์ การตั้งค่า และการติดตั้งเครือข่ายต่างๆโดยสามารถสร้างโหนด กำหนดคุณลักษณะต่าง ๆ ของโหนด กำหนดอัลกอริทึมในการค้นหาเส้นทางที่มีมาให้อยู่แล้ว เป็นต้น มีการทดสอบผลของค่าพารามิเตอร์ภายในเครือข่ายโดยทำการเปลี่ยนแปลงค่าในออปเจ็คของซีพลัสพลัส วิเคราะห์ผลของการเปลี่ยนแปลงพารามิเตอร์โดยใช้การ Trace file และแสดงผลทางหน้าจอ ข้อดีของภาษา Otcl คือสามารถทำงานได้อย่างรวดเร็วเมื่อมีการเปลี่ยนแปลง แต่มีข้อเสียคือประมวลผลได้ช้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 วิธีการใช้งานเอนเอส-ทูสำหรับโพรโทคอลทีซีพี

การใช้เอ็นเอส-ทูในการทำงานวิจัยทางด้านเครือข่ายมีสายสำหรับโพรโทคอลทีซีพีนั้นเป็นการศึกษาผลลัพธ์ซึ่งแสดงในรูปแบบของกราฟที่แสดงแนวโน้มของโพรโทคอลที่ทำการจำลองนั้น ๆ และส่วนของเน็ตเวิร์กแอนิเมเตอร์ซึ่งจะเป็นส่วนที่ทำการแสดงการเคลื่อนไหวและการจราจรในเครือข่ายที่เวลาต่าง ๆ ดังแสดงในรูปที่ 2.17



รูปที่ 2.17 ขั้นตอนการจำลองการทำงานของเอ็นเอส-ทู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของเอ็นเอส-ทูในรูป ซึ่งประกอบด้วยส่วนต่าง ๆ มีรายละเอียด ดังนี้

2.6.1 สคริปต์ทีซีแอล (TCL Script)

สคริปต์ทีซีแอลเป็นไฟล์ซึ่งกำหนดสภาวะการจำลองโดยรวมของเครือข่ายโดยในเครือข่ายมีสายจะประกอบด้วย 7 ส่วนย่อย ๆ คือ

1. การติดตั้งค่าพารามิเตอร์ของการจำลองการทำงาน (Simulation Parameters Setup)
2. การเริ่มต้นการทำงาน (Initialization)
3. การให้คำนิยามของโหนดต่าง ๆ (Nodes Definition)
4. การให้คำนิยามของการเชื่อมต่อ (Links Definition)
5. การให้คำนิยามของเอเจนต์ (Agents Definition)
6. การให้คำนิยามของแอพพลิเคชัน (Applications Definition)
7. การจบการทำงาน (Termination)

2.6.1.1 การติดตั้งค่าพารามิเตอร์ของการจำลองการทำงาน

ส่วนแรกของการจำลองการทำงานแบบมีสายเป็นส่วนที่ต้องประกาศรายละเอียดต่าง ๆ ของระบบเครือข่ายที่ต้องการจำลองขึ้น ได้แก่

val() เป็นตัวแปรที่ประกาศขึ้นมาเก็บค่ารายละเอียดของค่านั้น ๆ

set val(stop) เวลาที่ใช้ในการจำลอง; จะนำไปใช้ต่อในส่วนการจบการทำงาน

โปรแกรมที่ 2.1 Simulation Parameters Setup

```
#=================================
#     Simulation parameters setup
#=================================
set val(stop)   10.0;          # time of simulation end
```

2.6.1.2 การเริ่มต้นการทำงาน

การเริ่มต้นการทำงานเป็นส่วนที่เริ่มสร้างเครือข่ายจำลอง กำหนดที่เก็บ NS Trace File และ NAM Trace File ซึ่งโปรแกรมที่ 2.2 เป็นตัวอย่างการกำหนดค่าในส่วนการเริ่มต้นการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 2.2 Initialization

```
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
```

2.6.1.3 การให้คำนิยามของโหนดต่างๆ

เป็นส่วนที่สร้างโหนดซึ่งโปรแกรมที่ 2.3 เป็นตัวอย่างการกำหนดค่าในส่วนนี้

โปรแกรมที่ 2.3 Nodes Definition

```
#=================================
#        Nodes Definition
#=================================
#Create 2 nodes
set n0 [$ns node]
set n1 [$ns node]
```

2.6.1.4 การให้คำนิยามของการเชื่อมต่อ

เป็นส่วนที่สร้างการเชื่อมต่อ ซึ่งโปรแกรมที่ 2.4 เป็นตัวอย่างการกำหนดค่าในส่วนการให้คำนิยามของการเชื่อมต่อ

โปรแกรมที่ 2.4 Links Definition

```
#=================================
#        Links Definition
#=================================
#Create links between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50

#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1.5 การให้คำนิยามของเอเจนต์

เป็นส่วนที่กำหนดโพรโทคอลในชั้นทรานสปอร์ตโดยจะเลือกเป็นโพรโทคอลทีซีพีและสามารถระบุได้ว่าต้องการชนิดของทีซีพีแบบใด ได้แก่ Tahoe, Reno, New Reno, Vegas และ Sack โดยจะกำหนดโหนดต้นทางและโหนดปลายทาง (Sink) ซึ่งตอนท้ายสามารถกำหนดขนาดของแพ็คเก็ตได้อีกด้วย โดยโปรแกรมที่ 2.5 เป็นตัวอย่างการกำหนดค่าในส่วนการให้คำนิยามของเอเจนต์

โปรแกรมที่ 2.5 Agents Definition

```
#====================================
#        Agents Definition
#====================================
#Setup a TCP Newreno connection
set tcp0 [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500
```

2.6.1.6 การให้คำนิยามของแอพพลิเคชัน

ส่วนนี้จะเป็นส่วนที่เลือกแอพพลิเคชัน โดยที่นิยมมี 2 ชนิดคือ FTP และ Telnet โดยแอพพลิเคชันที่สร้างจะต้องกำหนดเอเจนต์ที่จะรองรับสามารถกำหนดเวลาเริ่มหยุดของ Application ที่สร้างขึ้นมาได้อีกด้วย ซึ่งโปรแกรม 2.6 เป็นตัวอย่างการกำหนดค่าส่วนการให้คำนิยามของแอพพลิเคชัน

โปรแกรมที่ 2.6 Application Definition

```
#====================================
#        Applications Definition
#====================================
#Setup a FTP Application over TCP/Newreno connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"
```

2.6.1.7 การจบการทำงาน

เมื่อถึงเวลาตามที่ $val(stop) ตั้งค่าไว้ ระบบจะหยุดทำงานและโหนดจะถูกรีเซตให้กลับสู่สภาวะเริ่มต้น หลังจากนั้น proc finish จะถูกเรียกขึ้นมาเพื่อ ทำหน้าที่สร้าง NS Trace File และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

Nam Trace File จากนั้นจึงแสดงข้อความบอกลาผู้ใช้ ซึ่งโปรแกรมที่ 2.7 เป็นตัวอย่างการกำหนดค่าในส่วนการจบการทำงาน

โปรแกรมที่ 2.7 Termination

```
#===================================
#        Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

2.6.2 Traffic File

Traffic File เป็นไฟล์ซึ่งเก็บลักษณะการจราจรในเครือข่าย ผู้ใช้สามารถสร้างเองโดยใช้ซอฟต์แวร์ cbrgen.tcl ที่ติดตั้งมาพร้อมกับเอ็นเอส-ทูอยู่แล้ว โดยการเข้าไปที่สารบบตามโปรแกรมที่ 2.8

โปรแกรมที่ 2.8 คำสั่งการติดตั้งซอฟต์แวร์ cbrgen.tcl

```
~/home/ns-allinone-2.34/ns-2.34/indept-utils/cmu-scen-gen
```

หลักจากนั้นใช้คำสั่ง

โปรแกรมที่ 2.9 คำสั่งการติดตั้งซอฟต์แวร์ cbrgen.tcl (ต่อ)

```
ns cbrgen.tcl [-type tcp] [-nn nodes] [-seed seed] [-mc
connection] [-rate rate]
```

โดยที่

Type คือการเลือกรูปแบบการส่ง TCP

Nodes คือจำนวนโหนดในการจำลอง หน่วยเป็น โหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Seed คือค่าจำนวนการส่ง

Connection คือจำนวนการเชื่อมต่อ หน่วยเป็น Connection

Rate คืออัตราการส่ง

โปรแกรมที่ 2.10 ตัวอย่างคำสั่ง

```
ns tcpgen.tcl  -type tcp -nn 25 -seed 0.0 -mc 8  -rate 4.0
```

ซึ่งโปรแกรม 2.11 เป็นตัวอย่างรายละเอียดของ Traffic File ที่ได้จากการใช้คำสั่งดังกล่าว

โปรแกรมที่ 2.11 Traffic file

```
#
# 5 connecting to 7 at time 163.0399642433226
#
set tcp_(1) [$ns_ create-connection  TCP $node_(5) TCPSink
$node_(7) 0]
$tcp_(1) set window_ 32
$tcp_(1) set packetSize_ 512
set ftp_(1) [$tcp_(1) attach-source FTP]
$ns_ at 163.0399642433226 "$ftp_(1) start"
```

2.6.3 Trace File

Trace File เป็นข้อมูลดิบซึ่งเป็นผลลัพธ์ของการจำลองที่แสดงรายละเอียดการเชื่อมต่อของแต่ละโหนดที่เวลาต่าง ๆ โดยจะประกอบด้วย 3 ส่วนคือ Wire_Event, IP_Trace และ Protocol

โปรแกรมที่ 2.12 รูปแบบของ Trace File

```
Wire_Event ------- TCP_trace
```

ซึ่งตารางที่ 2.3 และตารางที่ 2.4 เป็นรายละเอียดในส่วน Wire_Event และ TCP_trace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 Wire event

Event	Abbreviation	Type	Value
Wire Event	r : Receive d : Drop e : Error + : Enqueue - : Dequeue	%g %d %d %s %d %s %d %d.%d %d.%d %d %d	
		double	Time
		int	(Link-layer) Source Node
		int	(Link-layer) Destination Node
		string	Packet Name
		int	Packet Size
		string	Flags
		int	Flow ID
		int	(Network-layer) Source Address
		int	Source Port
		int	(Network-layer) Destination Address
		int	Destination Port
		int	Sequence Number
		int	Unique Packet ID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 TCP Trace

Event	Type	Value
TCP Trace	%d 0x%x %d %d	
	int	Ack Number
	hexadecimal	Flags (Used by FullTCP) FIN=0x01, SYN=02, PUSH=08, ACK=10, ECE=40, CWR=80
	int	Header Length
	int	Socket Address Length

ตัวอย่างการเรียงคอลัมน์ต่าง ๆ ของ Trace File ดังแสดงในรูปที่ 2.18

Type	Time	Source Node	Destination Node	Packet Name	Packet Size (Byte)	Flags	Flow ID	Source adress	Destination Address	Sequence Number	Packet ID
+	0.94176	2	3	tcp	1000	------	0	0.0	3.0	25	40
+	0.94176	2	3	tcp	1000	------	0	0.0	3.0	26	41
d	0.94176	2	3	tcp	1000	------	0	0.0	3.0	26	41
+	0.95072	2	0	ack	40	------	0	3.0	0.0	14	29
-	0.95072	2	0	ack	40	------	0	3.0	0.0	14	29
-	0.95176	2	3	tcp	1000	------	0	0.0	3.0	21	36
+	0.95176	2	3	tcp	1000	------	0	0.0	3.0	27	42

รูปที่ 2.18 คอลัมน์ต่างๆของ Trace File

2.6.4 สคริปต์ AWK

สคริปต์ AWK เป็นไฟล์ซึ่งเขียนด้วยภาษา AWK ทำหน้าที่เป็นข้อกำหนดในการคัดกรองแยกแยะ Trace File เพื่อหาข้อประสิทธิภาพต่าง ๆ ของการจำลอง เช่น PDF, Throughput, Delay และ Loss โดยที่

PDF (Packet Delivery Fraction) คือค่าสัดส่วนของปริมาณข้อมูลที่ปลายทางรับได้สำเร็จต่อปริมาณข้อมูลที่ส่งจากต้นทาง

Delay คือเวลาที่เสียไปในขณะที่ข้อมูลเดินทางจากต้นทางไปถึงปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Loss คือปริมาณข้อมูลที่หายไปในระหว่างการส่งข้อมูล

Throughput คือสัดส่วนของปริมาณข้อมูลที่ปลายทางรับได้สำเร็จต่อปริมาณข้อมูลที่ทุกโหนดในเครือข่ายส่งออกมา

ซึ่งโปรแกรมที่ 2.13 จะเป็นตัวอย่างสคริปต์ AWK สำหรับคัดกรอง TCP trace file

โปรแกรมที่ 2.13 สคริปต์ AWK สำหรับคัดกรอง TCP Trace File

```
BEGIN {
        startTime=0;
        stopTime=0;
        sends=0;
        recvs=0;
        droppedBytes=0;
        droppedPackets=0;
        highest_packet_id =0;
        sum=0;
        recvnum=0;
        recvdSize = 0
}
  {
  time = $2;
  packet_id = $6;
  pkt_size  = $8;

  # CALCULATE PACKET DELIVERY FRACTION
  if (($1=="s")&&($7=="tcp")&&($4=="AGT")&&(pkt_size>=512))
      {sends++; }
  if (($1=="r")&&($7=="tcp")&&( $4=="AGT")&&(pkt_size>=512))
      {recvs++; }
  # STORE START TIME
  if($4 == "AGT"&&($1 == "+"||$1 == "s")&&$8 >= 512)
  {
    if($2<startTime){
   startTime = $2
    }
  }
  # STORE STOP TIME
  if ($4 == "AGT" && $1 == "r" && $8 >= 512)
  {
    if ($2 > stopTime)
    {
   stopTime = $2
    }

    # For throughput
    # Rip off the header
    pkt_size = $8;
    hdr_size = pkt_size % 512;
    pkt_size -= hdr_size;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 2.13 สคริปต์ AWK สำหรับคัดกรอง TCP Trace File (ต่อ)

```
 # Store received packet's size
    recvdSize += pkt_size;
}
  # CALCULATE DELAY
  if (start_time[packet_id]==0)
        start_time[packet_id]=time;
  if (($1=="r")&&($7=="tcp")&&($4=="AGT")&&($8>=512))
          {
             end_time[packet_id] = time;
          }
        else {
             end_time[packet_id] = -1;
             }
  # DROPPED DSR PACKETS
  if ((($1=="D")||($1=="d"))&&($7=="tcp")&&($2>0))
       {
             droppedBytes=droppedBytes+$8;
             droppedPackets=droppedPackets+1;
       }

 #find the number of packets in the simulation
 if (packet_id > highest_packet_id)
   highest_packet_id = packet_id;

 }
 END {
  for ( i in end_time ) {
     start = start_time[i];
     end = end_time[i];
     packet_duration = end - start;
     if ( packet_duration > 0 )
         {    sum += packet_duration;
         recvnum++;
         }
  }
     delay=sum/recvnum;
     simTime=stopTime-startTime;
     PDF=(recvs/sends)*100; #Packet Delivery Ratio(fraction)
     PDR=recvs/simTime;          # Packet Delivery Rate
     PLR=droppedPackets/simTime;    # Packet Loss Rate
     throughput= recvdSize/simTime)*(8/1024);

     printf("%.2f %.2f %d %.2f", PDF, delay*1000,
droppedPackets, throughput);
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5 NAM Trace File

NAM Trace File เป็นข้อมูลดิบซึ่งเป็นผลลัพธ์ของการจำลองจะถูกเรียกใช้โดย NAM ที่จะนำไปประมวลผลและแสดงผลออกมาในรูปกราฟิกที่เวลาต่าง ๆ จะประกอบด้วย 2 ส่วนคือ Nam_Event และ Node_Trace

โปรแกรมที่ 2.14 รูปแบบของ NAM Trace File

```
Nam_Event ------- Node_trace
```

ซึ่งตารางที่ 2.5 และตารางที่ 2.6 เป็นรายละเอียดในส่วน NAM Trace File และ Node trace

ตารางที่ 2.5 รูปแบบของ NAM Trace File

Event	Abbreviation	Flag	Type	Value
Comment -- this line is ignored	#			
Dummy event to be used in time synchronization	T	-t	time	Time
Node	n	-t	time	Time
		-s	int	Node ID
		-u	double	X Velocity
		-U	double	X Velocity
		-V	double	Y Velocity
		-v	shape	Shape (circle, box, hexagon)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 รูปแบบของ NAM Trace File (ต่อ)

Event	Abbreviation	Flag	Type	Value
		-c	color	Color
		-z	double	Size Of Node
		-a	int	Address
		-x	double	X Location
Node	n	-y	double	Y Location
		-Z	double	Z Location (Not Supported)
		-i	color	Label Color
		-b	string	Label
		-l	string	Label
		-o	color	Previous Color
		-S	string	State (UP, DOWN, COLOR)
		-L	string	Previous Label
		-p	string	Label Location
		-P	string	Previous Label Location
		-i	color	Inside Label Color
		-I	color	Previous Inside Label Color

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 รูปแบบของ NAM Trace File (ต่อ)

Event	Abbreviation	Flag	Type	Value
		-e	color	Label Color
		-E	color	Previous Label Color
		-T	double	Duration Of Movement
		-w	flag	Wireless Node
		-t	time	Time
Link	l h: Hop	-s	int	Source ID
		-d	int	Destination ID
	r: Receive d: Drop Line +: Enqueue -: Dequeue	-r	double	Transmission Rate
		-D	double	Delay
		-h	double	Length
		-O	orientation	Orientation
		-b	string	Label
		-c	color	Color
		-o	color	Previous Color
		-S	string	State (UP, DOWN)
		-l	string	Label

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 รูปแบบของ NAM Trace File (ต่อ)

Event	Abbreviation	Flag	Type	Value
		-L	string	Previous Label
		-e	color	Label Color
		-E	color	Previous Label Color
		-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-e	int	Extent
		-a	int	Packet Color Attribute ID
		-i	int	ID
		-l	int	Energy
		-c	string	Conversation
		-x	comment	Comment
		-p	string	Packet Type
Packet	h: Hop r: Receive d: Drop Line	-k	string	Packet Type
		-y	comment	
Packet	+: Enqueue -: Dequeue E: Enqueue	-S	int	
		-m	int	
	D: Dequeue P: Drop	-f	int	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 รูปแบบของ NAM Trace File (ต่อ)

Event	Abbreviation	Flag	Type	Value
		-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-e	int	Extent
		-a	int	Attribute
		-i	int	ID
		-l	int	Energy
		-c	string	Conversation
Session	E: Enqueue D: Dequeue P: Drop	-x	comment	Comment
		-p	string	Packet Type
		-k	string	Packet Type
		-t	time	Time
Session Agent	E: Enqueue D: Dequeue P: Drop a f	-s	int	Source ID
		-d	int	Destination ID
		-x	flag	Remove Agent
		-n	string	Agent Name
		-t	time	Time
		-s	int	Source ID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 รูปแบบของ NAM Trace File (ต่อ)

Event	Abbreviation	Flag	Type	Value
Agent	a	-x	flag	Remove Feature
		-T	char	Type
		-n	string	Name
		-a	string	Agent
Feature	f	-v	string	Value
Event	G	-o	string	Previous Value
Group	G	-t	time	Time
	L	-n	string	Name
		-i	int	Node ID
		-a	int	Group ID
		-s	int	Source ID
Group	G	-d	int	Destination ID
Lan link	L	-o	orientation	Orientation
	L	-O	orientation	Orientation
	m	-t	time	Time
		-n	string	Name
		-s	int	Node ID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 รูปแบบของ NAM Trace File (ต่อ)

Event	Abbreviation	Flag	Type	Value
Lan link	L	-c	string	Color
Mark node	m	-h	string	Shape (circle, square, hexagon)
	m			
	R	-X	flag	Remove Mark
		-t	time	Time
		-s	int	Source ID
Mark node	m	-d	int	Destination ID
Routing event	R	-g	int	Multicast Group
	R	-p	packet source	Packet Source ID Or *
	v			
		-n	flag	Negative Cache
		-T	double	Timeout
Routing event	R	-m	string	Mode (IIF Or OIF)
Execute tcl expression	v			
	v			
Set trace file version	V			
	V			
	N			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 รูปแบบของ NAM Trace File (ต่อ)

Event	Abbreviation	Flag	Type	Value
Routing event	R	-t	time	Time
Execute tcl expression	v	-e	tcl expression	Tcl Script
Set trace file version	v			
Use nam graph	V	-t	time	Time
Energy status -- for future use	V	-v	string	Version
	N	-a	int	Attribute
	g	-t	time	Time
	A	-t	time	Time
		-n	int	Hierarchy
Use nam graph	g	-n	int	Hierarchy
Energy status -- for future use	A			
Hierarchical address space configuration -- initialization only	A	-p	int	Port Shift
	c	-o	hexadecimal	Port Mask
		-c	int	Multicast Shift
		-a	int	Multicast Mask
		-h	int	Hierarchy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 รูปแบบของ NAM Trace File (ต่อ)

Event	Abbreviation	Flag	Type	Value
		-m	int	Node Shift
		-s	int	Node Mask
		-t	time	Time
Hierarchical address space configuration -- initialization only		-i	int	ID
Color table configuration -- initialization only	c	-n	string	Color
	q	-t	time	Time
		-s	int	Source ID
Create packet queue -- initialization only	q	-d	int	Destination ID
	X	-a	orientation	Orientation
		-t	time	Time
		-n	string	Name
Layout lan	X	-r	double	Rate
		-D	double	Delay
		-o	orientation	Orientation
		-O	orientation	Orientation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 Node Trace

Event	Type	Value
Node Trace	{%s.%s %s.%s %d %s %s}	
	string	Source Node Address
	string	Source Node Port
	string	Destination Node Address
	string	Destination Node Port
	int	Sequence Number
	string	Flags
	string	Packet Name

2.6.6 Result File

Result File เป็นข้อมูลสรุปที่ได้จากการนำ Trace File มาคัดกรองด้วยสคริปต์ AWK และสามารถนำไปเขียนกราฟของ PDF, Delay, Loss และ Throughput ได้อย่างละ 1 จุด โดยรูปแบบและตัวอย่างการแสดงของ Result File เป็นไปตามโปรแกรมที่ 2.15 และโปรแกรมที่ 2.16

โปรแกรมที่ 2.15 รูปแบบที่แสดงของ Result File

```
PDF     Delay     Loss  Throughput
```

โปรแกรมที่ 2.16 ตัวอย่างผลลัพธ์การแสดง Result File

```
87.93   20.72  0.4              5.98
```

สำหรับการทำงานจะเริ่มจากการจำลองผ่านเอ็นเอส-ทูโดยใช้สคริปต์ TCL และ Traffic File เป็นข้อกำหนดสภาวะในการจำลองครั้งนั้น ผลลัพธ์ออกมาจะได้ Trace File ที่เป็นข้อมูลดิบซึ่งเข้าใจได้ยาก ผู้ใช้จึงต้องเขียนสคริปต์ AWK ในการคัดกรองหาข้อสรุปออกมาโดยมีข้อกำหนดต่าง ๆ อยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สคริปต์ AWK ผลลัพธ์จะได้ Result File จากนั้นผู้ใช้จะนำข้อมูลผลสรุปประสิทธิภาพเครือข่ายใน Result File ไปเขียนกราฟต่อไป

จะเห็นได้ว่าขั้นตอนต่าง ๆ มีความซับซ้อนคือการเขียนสคริปต์และไฟล์ต่าง ๆ นั้นผู้ใช้ต้องเขียนเองและสั่งให้ซอฟต์แวร์ต่าง ๆ ทำงานโดยใช้คำสั่งภาษายูนิกซ์ซึ่งการจำลองแต่ละรอบยังไม่เหมือนกันอีกด้วย อาจทำให้เกิดการหลงลืม สับสน เมื่อได้ Trace File ออกมาผู้ใช้ยังต้องเขียนสคริปต์ AWK เพิ่มอีก ในการคัดกรอง Trace File ซึ่ง Trace File ของแต่ละโพรโทคอลยังจัดเรียงมาไม่เหมือนกันอีกด้วย ทำให้ผู้ใช้ต้องศึกษาลักษณะของ Trace File เพิ่มจากนั้นจึงนำ Result File ที่ได้ไปเขียนกราฟได้ 1 จุด จะเห็นได้ว่าการที่จะเห็นแนวโน้มของโพรโทคอลที่สภาวะต่าง ๆ ต้องดูจากกราฟที่เป็นเส้นคือมีมากกว่า 2 จุดขึ้นไปเช่นนั้นผู้ใช้งานต้องทำกระบวนการเดิมซ้ำใหม่อีกเรื่อย ๆ จนกว่าจะเห็นแนวโน้มของกราฟตามที่ตั้งสมติฐานไว้ในตอนแรก

2.7 Java Programming

จาวาเป็นภาษาคอมพิวเตอร์ที่ได้รับการออกแบบสำหรับการใช้บนอินเตอร์เน็ตโดยมีส่วนของการมองเห็นและความรู้สึกแบบภาษาซีพลัสพลัสแต่ง่ายกว่าการใช้ซีพลัสพลัสและสามารถสร้างมุมมองโดยโปรแกรมได้ จาวาสามารถใช้ในการสร้างการประยุกต์แบบสมบูรณ์ซึ่งสามารถเรียกใช้ได้เฉพาะเครื่องคอมพิวเตอร์หรือการกระจายระหว่างเครื่องแม่ข่ายกับลูกข่ายในระบบเครือข่ายและสามารถสร้างโมดูลการประยุกต์ขนาดเล็กหรือแอพเลต (Applet) สำหรับเป็นส่วนของเว็บเพจแอพเลตทำให้มีความเป็นได้ในด้านการตอบสนองของผู้ใช้กับเว็บเพจ

คุณลักษณะคือ

1. จาวาเป็นโปรแกรมที่มีขนาดเล็กในระบบเครือข่าย การคอมไพล์จะแปลงโปรแกรมเป็น Java Bytecode ซึ่งสามารถเรียกใช้งานได้ทุกที่ภายในกลไกการจำลองเครือข่ายของจาวา (Java Virtual Machine) ตัวแปร Bytecode เป็นโปรแกรมเพื่อใช้งานบนเครื่องคอมพิวเตอร์มีความหมายว่าโปรแกรมนี้สามารถใช้ได้ในแพลตฟอร์มที่แตกต่างกันของคอมพิวเตอร์
2. คำสั่งเป็นแบบแข็งแรง มีความหมายว่าออปเจ็คของจาวาไม่มีการอ้างอิงข้อมูลหรือออปเจ็คจากภายนอกซึ่งแตกต่างจากซีพลัสพลัสและภาษาอื่น ๆ เป็นการทำให้มั่นใจว่าจะไม่มีการเก็บตำแหน่งของข้อมูลในโปรแกรมประยุกต์อื่นหรือในระบบปฏิบัติการที่ทำให้โปรแกรมไม่ทำงาน กลไกการจำลองเครือข่ายของจาวาจะทำการตรวจสอบแต่ละออปเจ็คที่ใช้ในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จาวาเป็นเหมือนกับออปเจ็คหนึ่งที่สามารถได้รับประโยชน์จากคลาสหรือคำสั่ง เนื่องจากออปเจ็คมีคุณสมบัติเป็นนามซึ่งทำให้ติดต่อกับผู้ใช้ได้ ขณะที่ภาษาดั้งเดิมมีคุณสมบัติเป็นกริยา ดังนั้น วิธีที่จะรับรู้เป็นความสามารถของออปเจ็คหรือพฤติกรรม
4. การประมวลผลทำที่เครื่องลูกข่าย ดังนั้นจาวาแอพเลตมีคุณลักษณะในการออกแบบให้ทำงานได้เร็ว
5. จาวาง่ายกว่าซีพลัสพลัสโดยเปรียบเทียบ
6. จาวาได้รับการแนะนำโดยบริษัท Sun Microsystems ในปี 1995 และทำให้เกิดทัศนคติการตอบสนองของเว็บทำให้ผู้ผลิตเว็บเบราว์เซอร์รายหลักได้รวมกลไกการจำลองเครือข่ายของจาวาเป็นส่วนหนึ่งของเบราว์เซอร์ ผู้พัฒนาระบบปฏิบัติเกือบทั้งหมดได้รวมจาวาคอมไพเลอร์เป็นส่วนหนึ่งของผลิตภัณฑ์

กลไกการจำลองเครือข่ายของจาวา (Java Virtual Machine) รวมถึงตัวเลือกคือคอมไพเลอร์แบบทันเวลาพอดี (Just-In-Time Complier) ซึ่งเป็นคอมไพเลอร์แบบไดนามิคในการคอมไพล์ Bytecode เป็นคำสั่งที่ประมวลผลได้ เป็นตัวเลือกในการแปร Bytecode ในหลาย ๆ กรณี Dynamic Just-In-Time สามารถคอมไพล์ได้เรียกว่าการแปรของกลไกการจำลองเครือข่ายของจาวา

จาวาสคริปต์ (JavaScript) เป็นภาษาที่พัฒนาโดย Netscape ซึ่งเป็นตัวแปร (Interpreter) ภาษาระดับสูงและง่ายกว่าการเขียนด้วยจาวา แต่ขาดความกะทัดรัดเหมือนจาวาและมีความเร็วไม่มาก เนื่องจากจาวาแอพเลตสามารถใช้งานได้รับเกือบทุกระบบปฏิบัติการโดยไม่ต้องคอมไพล์ใหม่ และจาวาไม่ใช้ส่วนขยายของระบบปฏิบัติการหรือตัวแปร ดังนั้นจาวาจึงได้รับพิจารณาเป็นภาษาหลักในการพัฒนาการประยุกต์บนเว็บ

Java Software Development Kit (J2SDK) เป็นชุดพัฒนาโปรแกรมด้วยภาษาจาวาซึ่งมีเครื่องมือช่วยสำหรับนักพัฒนาโปรแกรม เช่น คอมไพเลอร์ ดีบักเกอร์ รันไทม์(JRE) ไลบรารี และโปรแกรมสนับสนุนอื่น ๆ อีกมากมาย บริษัท Sun Microsystems ได้แบ่งรูปแบบการทำงานของโปรแกรมที่สร้างจากภาษาจาวาออกเป็น 3 ส่วน คือ

J2ME (Java 2 Micro Edition) หมายถึงรูปแบบสภาวะการทำงานของโปรแกรมที่สร้างจากภาษาจาวาแต่เหมาะสมสำหรับการใช้งานบนอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก เช่น โทรศัพท์มือถือ PDA เครื่องใช้ไฟฟ้า หรืออุปกรณ์ไร้สาย เป็นต้น

J2SE (Java 2 Standard Edition) หมายถึงรูปแบบสภาวะการทำงานของโปรแกรมที่สร้างจากภาษาจาวาที่ทำงานบนคอมพิวเตอร์ทั่วไปในลักษณะสแตนอโลน (Standalone) ประกอบไปด้วย

1. จาวาแอพพลิเคชัน (Java Application) คือลักษณะการทำงานในแบบทั่วไปสามารถติดตั้งและทำงานในทุกเครื่องที่ถูกติดตั้ง JVM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จาวาแอพเลต (Java Applet) คือลักษณะการทำงานของโปรแกรมที่ต้องทำงานบนพื้นที่ใช้งานเบราวเซอร์อีกที ดังนั้นเบราว์เซอร์จะต้องติดตั้ง JVM หรือ Java Plug-in ไว้รองรับการทำงาน
3. จาวาบีน (JavaBean) คือรูปแบบของชิ้นส่วนโปรแกรมที่สร้างขึ้นมาเพื่อนำไปใช้พัฒนาโปรแกรมหรือใช้เพื่อทำงานร่วมภายในโปรแกรมอื่น ๆ อีกทีหนึ่ง

J2EE (Java 2 Enterprise Edition) หมายถึงรูปแบบสภาวะการทำงานของโปรแกรมที่สร้างจากภาษาจาวาที่ทำงานบนคอมพิวเตอร์ระดับเซิร์ฟเวอร์เพื่อรองรับการใช้งานจากผู้ใช้จำนวนมาก ๆ ประกอบไปด้วย

1. จาวาเซิร์ฟเลต (Java Servlet) คือโปรแกรมสร้างจากภาษาจาวาตามรูปแบบการสร้างเซิร์ฟเลต การใช้งานต้องติดตั้งบนเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงานเมื่อเซิร์ฟเลตถูกเรียกใช้จากเบราวเซอร์ เซิร์ฟเลตจะทำงานโดยอาศัย JVM บนเครื่องเว็บเซิร์ฟเวอร์และประมวลผลการทำงานให้ได้ข้อมูลเพื่อจัดส่งไปให้ผู้เรียกใช้จากบราวเซอร์ต่อไป เซิร์ฟเลตถูกเรียกใช้จากผู้ใช้ผ่านทางโพรโทคอล HTTP และส่งผลลัพธ์เป็นข้อมูลเว็บไปยังเบราวเซอร์ของผู้ใช้อีกที
2. JSP (Java Server Page) คือการใช้ภาษาจาวาและลักษณะสคริปต์ โดยเขียนไว้เป็นส่วนหนึ่งของภาษา HTML และจัดเก็บไว้บนเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงาน JSP เมื่อผู้ใช้เรียกใช้งานเว็บเพจดังกล่าว ส่วนที่เป็น JSP ในเว็บเพจจะถูกคอมไพล์และประมวลผลผ่าน JVM ของเว็บเซิร์ฟเวอร์จากนั้นจะแทรกผลลัพธ์ที่ได้จากการประมวลผลลงในเว็บก่อนจัดส่งให้กับผู้เรียกดูต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนาซอฟต์แวร์

ในการพัฒนาโปรแกรมจำเป็นต้องมีการวางแผนและออกแบบโปรแกรมไว้ล่วงหน้า ซึ่งบทนี้จะอธิบายถึงแนวคิดในการพัฒนาซอฟต์แวร์ เครื่องมือที่และภาษาที่ใช้ในการพัฒนาซอฟต์แวร์ รายละเอียดของซอฟต์แวร์เชิงเทคนิค (Software Specification) โครงสร้างหลักของซอฟต์แวร์และกระบวนการทำงานภายในส่วนต่าง ๆ ภาพรวมของโปรแกรม ขอบเขตและข้อจำกัดของซอฟต์แวร์ที่พัฒนา

3.1 แนวคิดในการพัฒนา

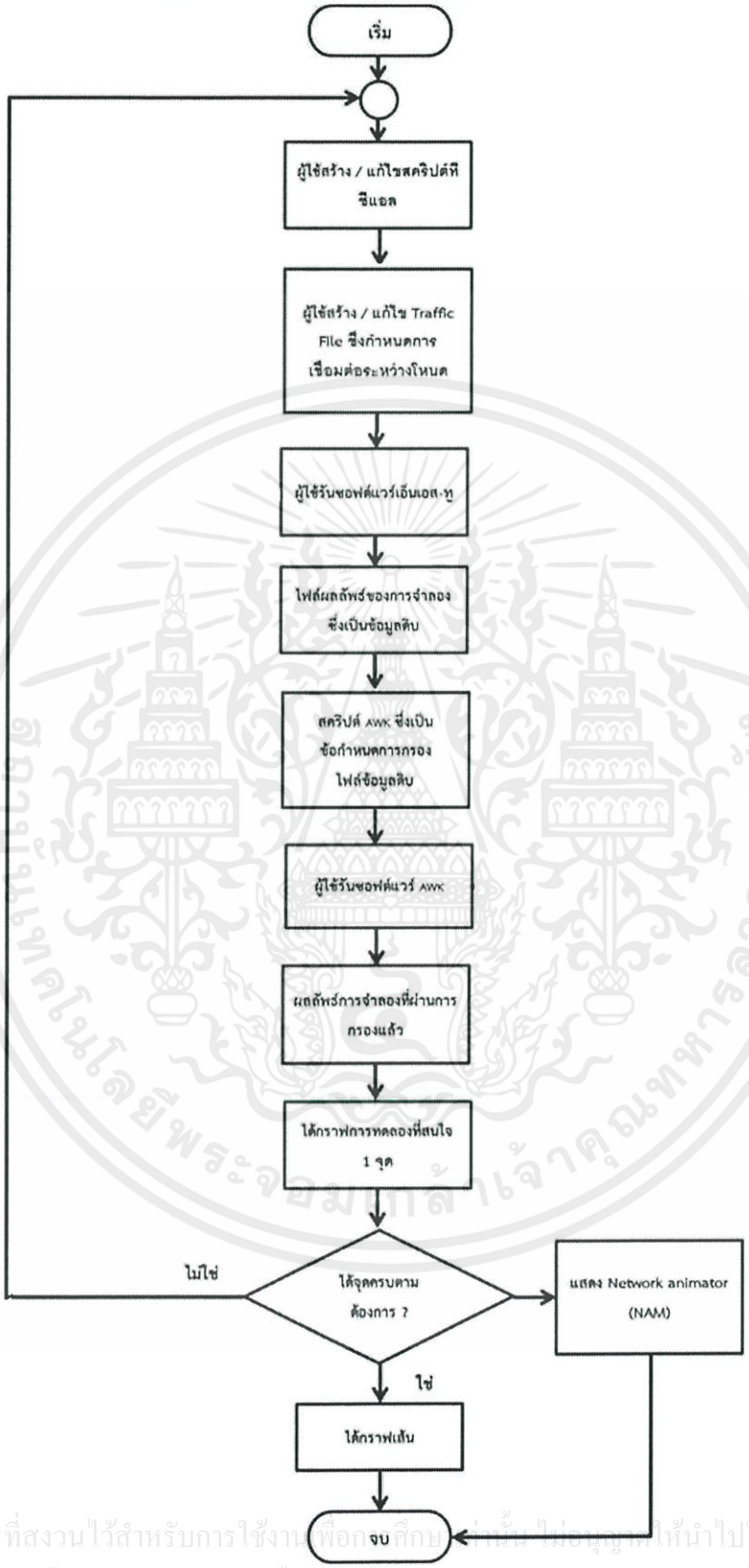
เอ็นเอส-ทู (Network Simulator 2 : NS-2) เป็นซอฟต์แวร์จำลองการทำงานของเครือข่ายที่นิยมใช้กันอย่างแพร่หลาย เนื่องจากมีความสามารถและประสิทธิภาพสูงในการจำลองเครือข่ายที่ระดับเลเยอร์ต่าง ๆ และยังมีการพัฒนาความสามารถเพิ่มขึ้นเรื่อย ๆ ทำให้เป็นประโยชน์อย่างยิ่งในการศึกษาพฤติกรรมต่าง ๆ ของระบบเครือข่าย สำหรับการใช้งานเอ็นเอส-ทูสำหรับจำลองเครือข่ายเคลื่อนที่จะเริ่มจากผู้ใช้ต้องสร้างสคริปต์ทีซีแอล (TCL Script) ซึ่งเป็นส่วนกำหนดสภาพแวดล้อมพื้นฐานของการจำลองและ Traffic File คือไฟล์ที่กำหนดจำนวนการเชื่อมระหว่างโหนด เมื่อผู้ใช้สร้างทั้ง 2 ไฟล์เสร็จแล้ว ต่อไปผู้ใช้ต้องทำการรันซอฟต์แวร์เอ็นเอส-ทูโดยใช้สองไฟล์ที่สร้างมาเมื่อข้างต้นเป็นข้อมูลนำเข้า ผลลัพธ์จะได้ออกมาเป็นไฟล์ผลลัพธ์ของการจำลองซึ่งเป็นข้อมูลดิบ (Trace File) ไฟล์ข้อมูลดิบที่ได้นี้ยังไม่สามารถนำไปสรุปผลได้ ผู้ใช้จะต้องสร้างสคริปต์ AWK ในการคัดกรองสรุปหาค่าประสิทธิภาพการสื่อสารได้แก่ PDF, Delay, Loss และ Throughput จากนั้นผู้ใช้ต้องทำการรันซอฟต์แวร์ AWK โดยให้สคริปต์ AWK และไฟล์ข้อมูลดิบเป็นข้อมูลนำเข้าจะได้ผลลัพธ์ออกมาเป็นผลลัพธ์ในเทอมของประสิทธิภาพเครือข่ายที่จำลองที่ผ่านการกรองแล้ว (Result File) ในขั้นตอนสุดท้ายของการหาผลครั้งนี้ผู้ใช้จะต้องนำผลลัพธ์ของการจำลองที่ผ่านการกรองแล้วนั้นไปเขียนกราฟของการทดลองได้ 1 จุด เมื่อยังไม่ได้จุดละเอียดตามต้องการผู้ใช้ต้องย้อนกลับไปยังขั้นตอนการแก้ไขสคริปต์ทีซีแอลและ Traffic File ใหม่และทำแบบวนรอบจนกว่าจะได้กราฟเส้นที่ละเอียดตามต้องการ

จากการใช้งานข้างต้นจะเห็นได้ว่าการใช้งานเอ็นเอส-ทูมีความซับซ้อนและเริ่มต้นใช้งานยากสำหรับผู้ที่ไม่คุ้นเคย โดยที่สคริปต์และไฟล์ต่าง ๆ นั้นผู้ใช้ต้องเขียนเองและสั่งให้ซอฟต์แวร์ต่าง ๆ ทำงานโดยใช้คำสั่งภาษายูนิกซ์ซึ่งการจำลองแต่ละรอบนั้นไม่เหมือนกันอาจทำให้เกิดการหลงลืมสับสน เมื่อได้ไฟล์ผลลัพธ์ของการจำลองซึ่งเป็นข้อมูลดิบออกมาผู้ใช้ยังต้องเขียนสคริปต์ AWK เพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อคัดกรองข้อมูลดิบซึ่งแต่ละไฟล์ข้อมูลดิบของแต่ละโพรโทคอลยังจัดเรียงมาไม่เหมือนกันอีกด้วย ทำให้ผู้ใช้ต้องศึกษาลักษณะของไฟล์ข้อมูลดิบเพิ่มอีกจากนั้นจึงนำผลลัพธ์ของการจำลองที่ผ่านการกรองแล้วที่ได้ไปเขียนกราฟได้ 1 จุด จะเห็นได้ว่าการที่จะเห็นแนวโน้มของโพรโทคอลที่สภาวะต่าง ๆ ต้องดูจากกราฟที่เป็นเส้นคือมีมากกว่า 2 จุดขึ้นไป เช่นนั้นผู้ใช้งานต้องทำการบวนการเดิมซ้ำใหม่อีกเรื่อย ๆ จนกว่าจะเห็นแนวโน้มของกราฟตามที่ตั้งสมติฐานไว้ในตอนแรก ซึ่งแผนผังการทำงานได้แสดงดังแสดงในรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

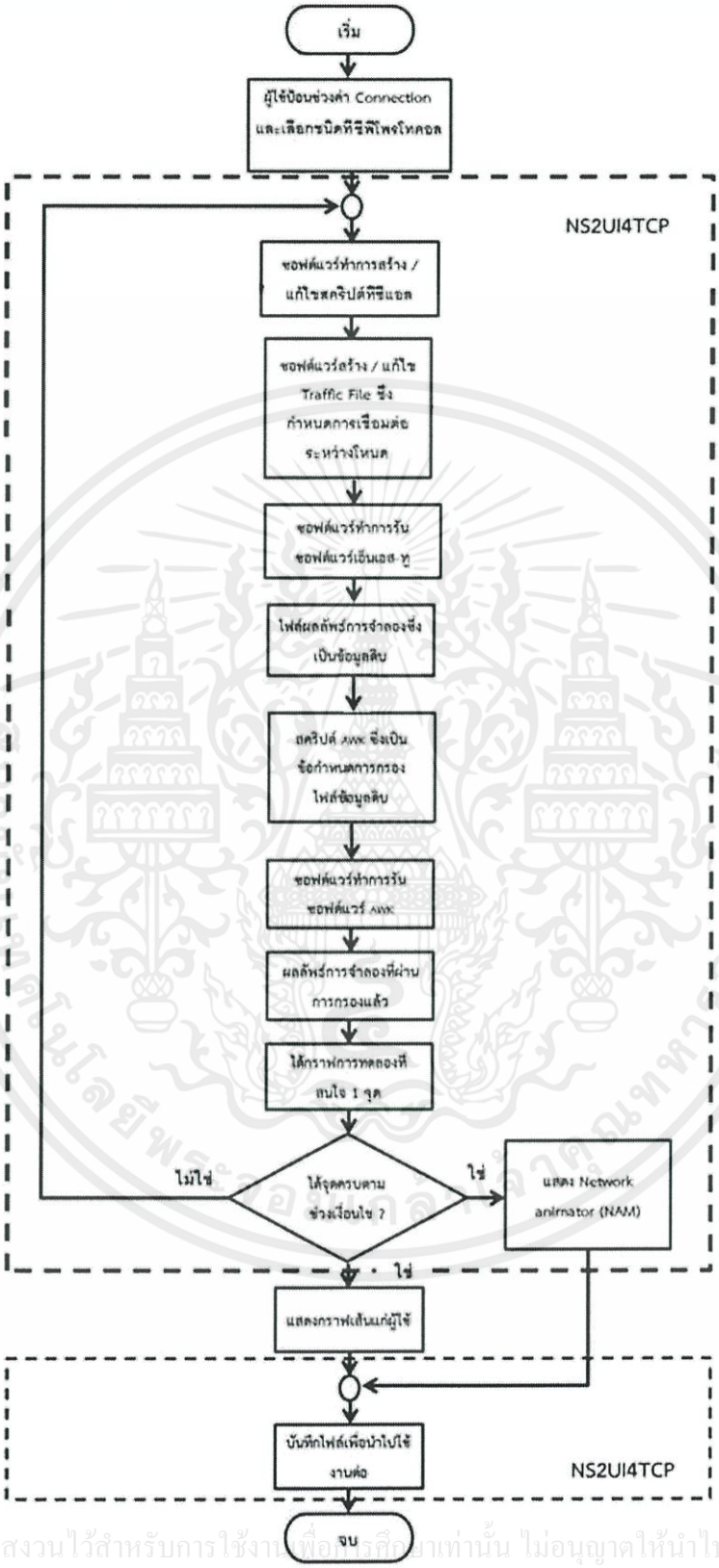


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 ขั้นตอนจำลองการทำงานของเอ็นเอส-ทูและส่วนที่เกี่ยวข้อง

จากความซับซ้อนของการใช้เอ็นเอส-ทูในการจำลองเครือข่ายงานแบบมีสายแบบเฉพาะกิจ ทางผู้จัดทำจึงเสนอซอฟต์แวร์ซึ่งทำครอบในส่วนที่ซับซ้อนไว้เพียงแค่ให้ผู้ใช้งานป้อนข้อมูลการจำลองเฉพาะที่สำคัญ เช่น ค่า Traffic และเลือกโพรโทคอลทีซีพีที่ต้องการ เป็นต้น ผ่านทางส่วนติดต่อผู้ใช้ที่ได้พัฒนาขึ้น จากนั้นก็เป็นหน้าที่ของซอฟต์แวร์เอ็นเอส-ทูยูไอฟอร์ทีซีพีในการวนรอบซึ่งจะเริ่มตั้งแต่สร้างสคริปต์ทีซีแอล, Traffic File รันผ่านเอ็นเอส-ทู กรองผลลัพธ์ผ่าน AWK ซึ่งจะได้ข้อมูลประสิทธิภาพของเครือข่ายสำหรับ 1 จุด จากนั้นซอฟต์แวร์จะตรวจสอบเงื่อนไขว่าได้จุดที่ละเอียดเพียงพอจะเป็นกราฟเส้นหรือไม่ ถ้าไม่เพียงพอซอฟต์แวร์จะกลับไปเริ่มขั้นตอนตั้งแต่แก้ไขสคริปต์ทีซีแอลและ Traffic File ใหม่จนสุดท้ายเมื่อวนรอบจนได้กราฟเส้นที่มีจุดละเอียดตามต้องการแล้ว ซอฟต์แวร์จะนำจุดที่ได้จากการวนรอบทั้งหมดเขียนเป็นกราฟแสดงประสิทธิภาพของเครือข่ายทั้ง 4 กราฟคือ PDF, Delay, Loss และ Throughput พร้อมแสดงแก่ผู้ใช้งานและหากผู้ใช้งานต้องการเห็นการทำงานของแต่ละโหนดก็สามารถสั่งให้แสดงผ่านเน็ตเวิร์กแอนิเมเตอร์ (Network Animator : NAM) ได้ซึ่งจะทำให้ผู้ใช้เห็นภาพและมีความเข้าใจในการทำงานของโหนดที่จำลองมากขึ้นและผู้ใช้สามารถบันทึกไฟล์ผลลัพธ์ออกไปเพื่อไปใช้งานต่อยอด ใช้ประโยชน์หรือเพื่อศึกษาต่อได้ เป็นอันเสร็จสิ้นขั้นตอน ซึ่งการทำงานที่กล่าวมานี้อธิบายเป็นแผนผังการทำงานได้ดังแสดงในรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 ขั้นตอนการจำลองการทำงานของเอ็นเอสทูยูไอฟอร์ทีซีพี

3.2 เครื่องมือที่ใช้ในการพัฒนา

ในการพัฒนาซอฟต์แวร์ได้มีการกำหนดเครื่องมือและภาษาที่ใช้ในการพัฒนาเพื่อให้มีความเหมาะสมกับกับระบบและความต้องการใช้งาน ดังนี้

3.2.1 ระบบปฏิบัติการ

โปรแกรมเอ็นเอส-ทูเป็นโปรแกรมที่สามารถติดตั้งเพื่อใช้งานได้เฉพาะระบบปฏิบัติการลินุกซ์เท่านั้น ดังนั้นโครงงานนี้จึงเลือกใช้ระบบปฏิบัติการลินุกซ์ในการพัฒนาและทดสอบ โดยระบบปฏิบัติการลินุกซ์ที่เลือกใช้งานคืออูบุนตู 10.10 เนื่องจากสามารถติดตั้งเอ็นเอส-ทูได้ง่ายและสมบูรณ์ อีกทั้งเป็นระบบปฏิบัติการที่ค่อนข้างมีข้อผิดพลาดน้อย แต่ในการใช้งานจริงผู้ใช้สามารถใช้ระบบปฏิบัติการลินุกซ์ตัวใดก็ได้ที่สนับสนุนการทำงานของเอ็นเอส-ทูและในการพัฒนาโปรแกรมจะพัฒนาในระบบปฏิบัติการวินโดวส์ 7

3.2.2 ภาษาโปรแกรมที่ใช้

ภาษาที่ใช้ในการพัฒนาซอฟต์แวร์คือภาษาจาวาซึ่งเป็นภาษาเชิงวัตถุ (Object Oriented Programming) เหมาะกับระบบที่มีความซับซ้อนและใช้ได้หลายแพลตฟอร์มโดยไม่จำเป็นต้องแก้ไขหรือคอมไพล์ใหม่และภาษาจาวาเป็นภาษาที่ได้รับความนิยมสูงจึงทำให้มีผู้สร้างไลบรารีมาให้ใช้เป็นจำนวนมากได้โดยไม่เสียค่าใช้จ่าย ทำให้ช่วยลดระยะเวลาและค่าใช้จ่ายในการพัฒนาโปรแกรม รวมถึงภาษาจาวามีไวยากรณ์ของภาษาที่ถูกออกแบบมาเป็นอย่างดีอีกด้วย

3.2.3 เครื่องมือพัฒนาซอฟต์แวร์ (Integrated Development Environment)

เครื่องมือพัฒนาซอฟต์แวร์ (IDE) จะช่วยพัฒนาซอฟต์แวร์โดยมีสิ่งอำนวยความสะดวกต่างๆ ซึ่งจะรวมคำสั่ง เมนู และส่วนติดต่อผู้ใช้งานแบบกราฟิก (Graphical User Interface) ต่าง ๆ มาสร้างเป็นซอฟต์แวร์ที่รวมคอมไพเลอร์ไปในตัว หรือก็คือเป็นซอฟต์แวร์ที่จะสร้างสภาพแวดล้อม (Environment) ให้เหมาะแก่กาพัฒนาซอฟต์แวร์ เพื่อเสริมให้เกิดความถูกต้องแม่นยำ มีผลให้สามารถพัฒนาได้สะดวกและรวดเร็วมากยิ่งขึ้นซึ่งเครื่องมือพัฒนาซอฟต์แวร์ที่ผู้จัดทำเลือกใช้คือ Netbeans IDE 6.9.1

NetBeans IDE เป็นเครื่องมือพัฒนาซอฟต์แวร์ที่มีประสิทธิภาพสูงสามารถใช้งานง่ายและเป็นเครื่องมือที่นำมาใช้งานได้โดยไม่เสียค่าใช้จ่าย นอกจากนี้ Netbeans IDE ยังสามารถพัฒนาซอฟต์แวร์ด้วยภาษาอื่น ๆ ได้อีกมากมาย ทำได้โดยติดตั้งโปรแกรมเสริม (Add-on) จากเว็บไซต์หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านตัวอัปเดตเซนเตอร์ (Update Center) ของ NetBeans IDE เช่น ภาษา C, C++ Ruby, UML, SOA เป็นต้น

3.2.4 JRE (Java Runtime Environment)

เป็นเทคโนโลยีจาวาที่ใช้ในการรันซอฟต์แวร์ภาษาจาวาที่จะรวบรวมคลาสแลอินเตอร์เฟสต่าง ๆ ที่มีความจำเป็นต่อการใช้งานของซอฟต์แวร์ภาษาจาวา เพื่อให้ผู้ใช้สามารถใช้งานซอฟต์แวร์ภาษาจาวาบนระบบปฏิบัติการนั้น ๆ ได้ โดยโครงงานนี้จะใช้ JRE 6 ซึ่งสามารถดาวน์โหลดได้โดยไม่เสียค่าใช้จ่ายจากเว็บไซต์ของจาวา

3.3 รายละเอียดของซอฟต์แวร์เชิงเทคนิค (Software Specification)

3.3.1 รายละเอียดส่วนนำเข้า (Input)

1. ผู้ใช้เรียกใช้งานซอฟต์แวร์ผ่านระบบปฏิบัติการลินุกซ์โดยจะเป็นซอฟต์แวร์ประเภท Java Desktop Application มีลักษณะเป็นการจัดรูปแบบเอกสาร (Document-Based) ที่เน้นการเพิ่ม ลบ แก้ไข รวมถึงเชื่อมโยงเอกสารต่าง ๆ ที่มีอยู่แล้ว
2. ผู้ใช้สามารถสร้างรูปแบบการจำลองเครือข่าย (Network Simulation) ขึ้นมาในรูปแบบของกราฟิกและกำหนดค่าพารามิเตอร์ต่าง ๆ ของเครือข่ายได้
3. ผู้ใช้สามารถเปิดไฟล์ที่เคยสร้างขึ้นไว้ก่อนหน้าขึ้นมาทำงานต่อในซอฟต์แวร์ได้

3.3.2 รายละเอียดส่วนนำออก (Output)

1. ซอฟต์แวร์สามารถแปลงเครือข่ายจำลองที่ผู้ใช้ได้สร้างขึ้นมาเป็น TCL Script ได้
2. ซอฟต์แวร์สามารถสร้างไฟล์ Traffic และสคริปต์ AWK จากเครือข่ายจำลองได้
3. ซอฟต์แวร์สามารถสร้างกราฟแสดงประสิทธิภาพของเครือข่ายตามความต้องการของผู้ใช้งานได้
4. ซอฟต์แวร์สามารถสร้างแอนิเมชันผ่านการทำงานของเนตเวิร์กแอนิเมเตอร์ตามความต้องการของผู้ใช้งานได้
5. ผู้ใช้สามารถบันทึกไฟล์เพื่อนำไฟล์การจำลองเครือข่ายไปใช้งานต่อได้
6. ซอฟต์แวร์สามารถให้คำแนะนำ ความรู้ของโพรโทคอลทีซีพีแต่ละโพรโทคอลให้กับผู้ใช้งานได้
7. ซอฟต์แวร์ให้คำแนะนำเบื้องต้นในการใช้งานของซอฟต์แวร์ที่พัฒนาแก่ผู้ใช้งานได้

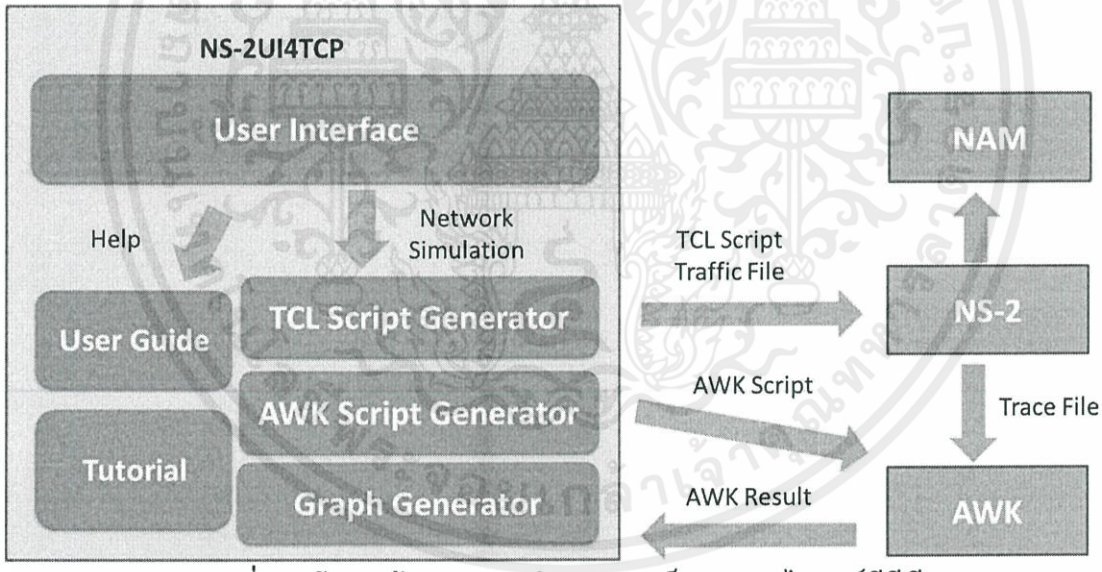
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 รายละเอียดฟังก์ชัน (Function)

1. ซอฟต์แวร์ทำงานได้โดยขึ้นอยู่กับระบบปฏิบัติการลินุกซ์เท่านั้น
2. ซอฟต์แวร์สามารถแปลงเครือข่ายจำลองให้กลายเป็น TCL Script และส่งไปยังซอฟต์แวร์เอ็นเอส-ทูเพื่อให้นำไปประมวลผลต่อ
3. ซอฟต์แวร์สามารถสร้างกราฟจากไฟล์ Traffic และ AWK Script ที่สร้างขึ้นมาเองและ Trace File จากผลลัพธ์ที่ได้จากการรันเอ็นเอส-ทูมาสร้างเป็นกราฟแสดงประสิทธิภาพตามที่ผู้ใช้ต้องการได้
4. ซอฟต์แวร์สามารถเชื่อมต่อโหนดต่าง ๆ ตรงตามกฎเกณฑ์ของซอฟต์แวร์เอ็นเอส-ทู

3.4 โครงสร้างของซอฟต์แวร์

โครงสร้างหลักของซอฟต์แวร์และกระบวนการทำงานภายในส่วนต่าง ๆ โดยภาพรวมสามารถแสดงให้เห็นดังแสดงในรูปที่ 3.3



รูปที่ 3.3 โครงสร้างและการทำงานของเอ็นเอส-ทูยูไอฟอร์ทีซีพี

จากรูปจะเห็นว่าซอฟต์แวร์ได้ถูกทดสอบการใช้งานผ่านระบบปฏิบัติการอูบุนตู 10.10 โดยส่วนของซอฟต์แวร์ที่ทางผู้จัดทำได้พัฒนาขึ้นจะอยู่ทางด้านซ้ายมือ ส่วนทางขวามือเป็นซอฟต์แวร์เอ็นเอส-ทูที่ถูกติดตั้งลงบนระบบปฏิบัติการแล้ว ซอฟต์แวร์ที่ผู้จัดทำได้พัฒนามีการแบ่งการทำงานเป็นห้าส่วนหลัก ๆ ได้แก่ ส่วนติดต่อผู้ใช้ (User Interface) ส่วนการแปลงการจำลองเครือข่ายเป็นภาษา TCL (TCL Script Generator) ส่วนการสร้างสคริปต์ AWK (AWK Script Generator) ส่วนการสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟ (Graph Generator) และส่วนการจำลองเครือข่ายผ่านเนตเวิร์กแอนิเมเตอร์ (Network Animator) แต่ละส่วนจะมีรายละเอียดการทำงาน ดังนี้

3.4.1 ส่วนติดต่อผู้ใช้ (User Interface)

ส่วนแรกนี้จะเป็นส่วนที่ใช้ในการสร้างการจำลองเครือข่ายแบบมีสาย (Wired Network Simulation) โดยผู้ใช้สามารถกำหนดการทำงานด้วยการ เพิ่ม ลบ หรือเคลื่อนย้ายโหนด เป็นต้น รวมถึงการตั้งค่าพารามิเตอร์ต่าง ๆ ที่เกี่ยวข้องได้อย่างสะดวกและง่ายผ่านส่วนติดต่อผู้ใช้งานของซอฟต์แวร์ นอกจากนี้ผู้ใช้ยังสามารถจัดเก็บไฟล์การจำลองเครือข่ายไว้ในฮาร์ดดิสก์แล้วนำมาเปิดเพื่อใช้งานต่อในซอฟต์แวร์ได้

3.4.2 ส่วนการแปลงการจำลองเครือข่ายเป็นภาษาทีซีแอล (TCL Script Generator)

เมื่อผู้ใช้ได้สร้างรูปแบบของการเชื่อมโยงเครือข่าย (Topology) ของการจำลองเครือข่ายเรียบร้อยแล้ว การทำงานส่วนนี้จะทำการแปลงการจำลองเครือข่ายให้กลายเป็นไฟล์สคริปต์ทีซีแอล เพื่อให้ซอฟต์แวร์เอ็นเอส-ทูนำไปรันและแสดงผลได้ถูกต้องตามเครือข่ายจำลองที่ได้กำหนดไว้ โดยผู้ใช้สามารถจัดเก็บไฟล์เพื่อนำไปรันกับซอฟต์แวร์เอ็นเอส-ทูเองหรือรันผ่านซอฟต์แวร์ที่ผู้จัดทำได้พัฒนาขึ้นซึ่งการทำงานจะอยู่ในส่วนถัดไปที่จะกล่าวถึง

3.4.3 ส่วนการสร้างสคริปต์ AWK (AWK Script Generator)

เมื่อไฟล์สคริปต์ทีซีแอลผ่านการประมวลผลจากซอฟต์แวร์เอ็นเอสทูแล้ว จากนั้นซอฟต์แวร์เอ็นเอส-ทูจะสร้าง Trace File เพื่อที่ผู้ใช้สามารถนำไฟล์นี้ไปใช้คำนวณหาจุดแต่ละจุดของกราฟแสดงประสิทธิภาพเครือข่ายซึ่งมีการใช้งานร่วมกับสคริปต์ AWK ผ่าน AWK โดยส่วนนี้จะสร้างสคริปต์ AWK ขึ้นมาเพื่อใช้ในการสร้างกราฟ

3.4.4 ส่วนการสร้างกราฟ (Graph Generator)

หลังจากที่ AWK นำ Trace File และสคริปต์ AWK ไปประมวลผลให้ได้จุดหนึ่งจุดของกราฟแล้วจะส่งผลลัพธ์กลับมายังส่วนการสร้างกราฟเพื่อให้ซอฟต์แวร์ทำการวาดกราฟขึ้น การทำงานในส่วนนี้จะมีลักษณะแบบวนเป็นลูป (Loop) เพื่อให้ได้กราฟที่เป็นเส้น (Line graph) ตามที่ผู้ใช้งานต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 ส่วนการจำลองเครือข่ายผ่านเนตเวิร์กแอนิเมเตอร์ (Network Animator)

ส่วนนี้จะทำงานพร้อม ๆ กับส่วนการสร้างกราฟซึ่งหลังจากที่ AWK นำ Trace File และสคริปต์ AWK ไปประมวลผลได้จุดหนึ่งจุดของกราฟแล้วก็จะส่งผลลัพธ์มายังส่วนนี้เช่นกัน เพื่อให้ซอฟต์แวร์ทำการสร้างแอนิมชัน (Animation) ของการจำลองเครือข่ายที่ผู้ใช้ได้สร้างขึ้นโดยการทำงานจะมีลักษณะแบบวนลูปเช่นกันเพื่อนำผลลัพธ์สุดท้ายที่ได้ไปแสดงแก่ผู้ใช้งานตามที่ต้องการได้

3.5 แผนภาพ UML (Unified Modeling Language Diagram)

3.5.1 Use Case Diagram

ระบบของ NS2UI4TCP จะมีผู้ใช้ระบบที่เป็นคน (User) และมีระบบย่อยซึ่งแยกออกมาตามกรณีการใช้งานระบบอย่างชัดเจนดังแสดงในรูปที่ 3.4 จะเห็นว่ามีระบบหลักทั้งหมดสี่ส่วน ได้แก่

1. สร้างการจำลอง (Create Simulation)
2. ส่วนเรียกแสดงกราฟ (View Graph)
3. ส่วนเรียกแสดงเน็ตเวิร์กแอนิเมเตอร์ (View Network Animator)
4. ส่วนความรู้เกี่ยวกับโพรโทคอลทีซีพี (View TCP Protocol Tutorial)
5. ส่วนคู่มือการใช้งานซอฟต์แวร์ (View Manual)

โดยทั้งสามระบบหลักแรกนี้จะมีการเรียกใช้งาน (include) ระบบย่อยอื่น ๆ อีกซึ่งได้แก่ configure parameters, configure graph, run AWK, run NS-2, create AWK script, create TCL script, create trace file, create traffic file, create NAM file และ run NAM โดยมีรายละเอียดการทำงานดังนี้

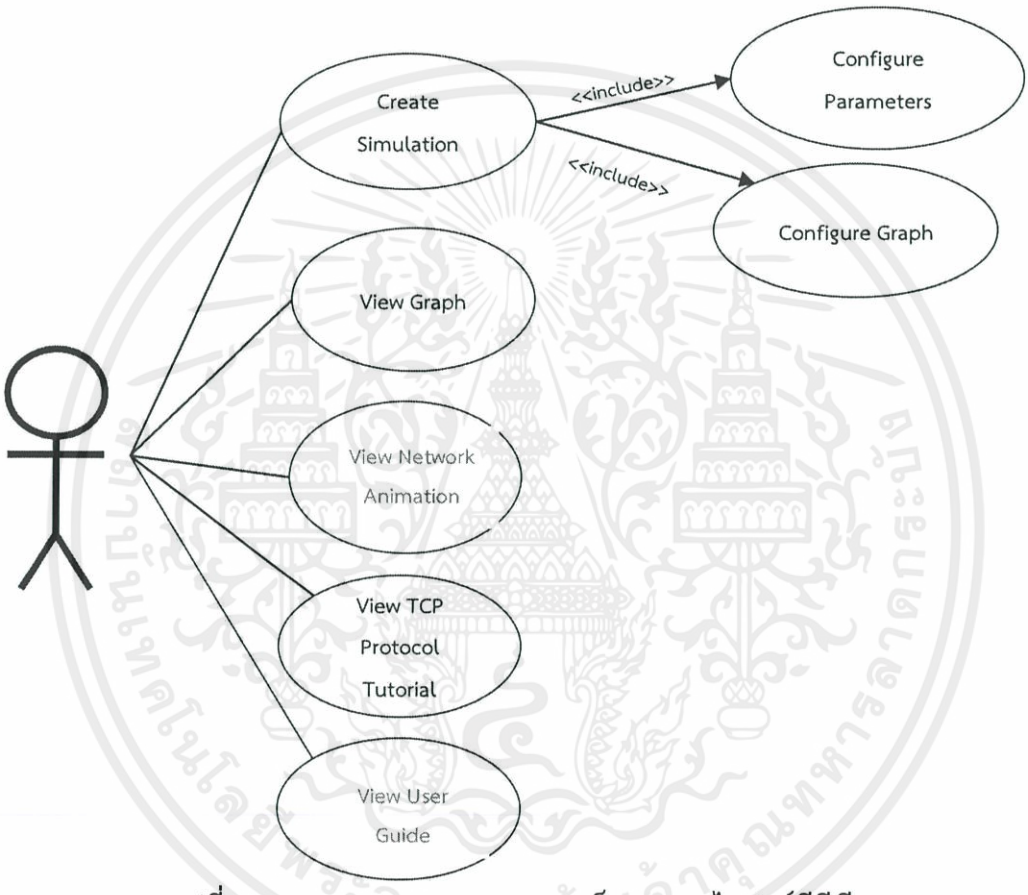
การสร้างการจำลอง (create simulation) เป็นการสร้างเครือข่ายจำลองขึ้นมา โดยจะมีการเรียกใช้การตั้งค่าพารามิเตอร์ (configure parameters) และการตั้งค่ากราฟ (configure graph) เพื่อตั้งค่าที่จำเป็นสำหรับการจำลองเครือข่ายและการแสดงผลในรูปแบบของกราฟ ซึ่งส่งผลในส่วนของการแสดงรูปแบบแอนิเมชันด้วยเช่นกัน

ในการแสดงผลของกราฟให้ผู้ใช้งานในระบบได้เห็น ผู้ใช้จะมีการเรียกใช้เรียกแสดงกราฟ (view graph) ซึ่งการทำงานเหล่านี้จะต้องรันเอ็นเอส-ทูเพื่อสร้าง trace file ซึ่งเอ็นเอส-ทูจะต้องมีการเรียกใช้ TCL script และ traffic file จากนั้นจึงส่ง trace file ไปยัง AWK รวมถึงมีการสร้าง AWK script ส่งไปยัง AWK เพื่อให้ได้ผลลัพธ์ที่สามารถนำไปสร้างกราฟได้ นอกจากนี้เอ็นเอส-ทูยังส่ง trace file ไปที่ NAM เพื่อสร้างเนตเวิกอนิเมเตอ (create NAM file)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของความรู้เกี่ยวกับโพรโทคอลทีซีพี จะเป็นรายละเอียดกระบวนการทำงานต่าง ๆ ของแต่ละโพรโทคอลทีซีพีและเนื้อหาความรู้อื่น ๆ ที่เกี่ยวข้อง

และในส่วนของคู่มือการใช้งานซอฟต์แวร์ ผู้ใช้สามารถดูรายละเอียดการใช้งานต่าง ๆ ของซอฟต์แวร์ ความหมายของพารามิเตอร์ การตั้งค่าพารามิเตอร์ต่าง ๆ หรือวิธีการใช้งานอื่น ๆ ที่เกี่ยวข้อง



รูปที่ 3.4 แผนภาพ Use Case ของเอ็นเอส-ทูยูไอฟอร์ทีซีพี

3.5.2 คลาสไดอะแกรม (Class Diagram)

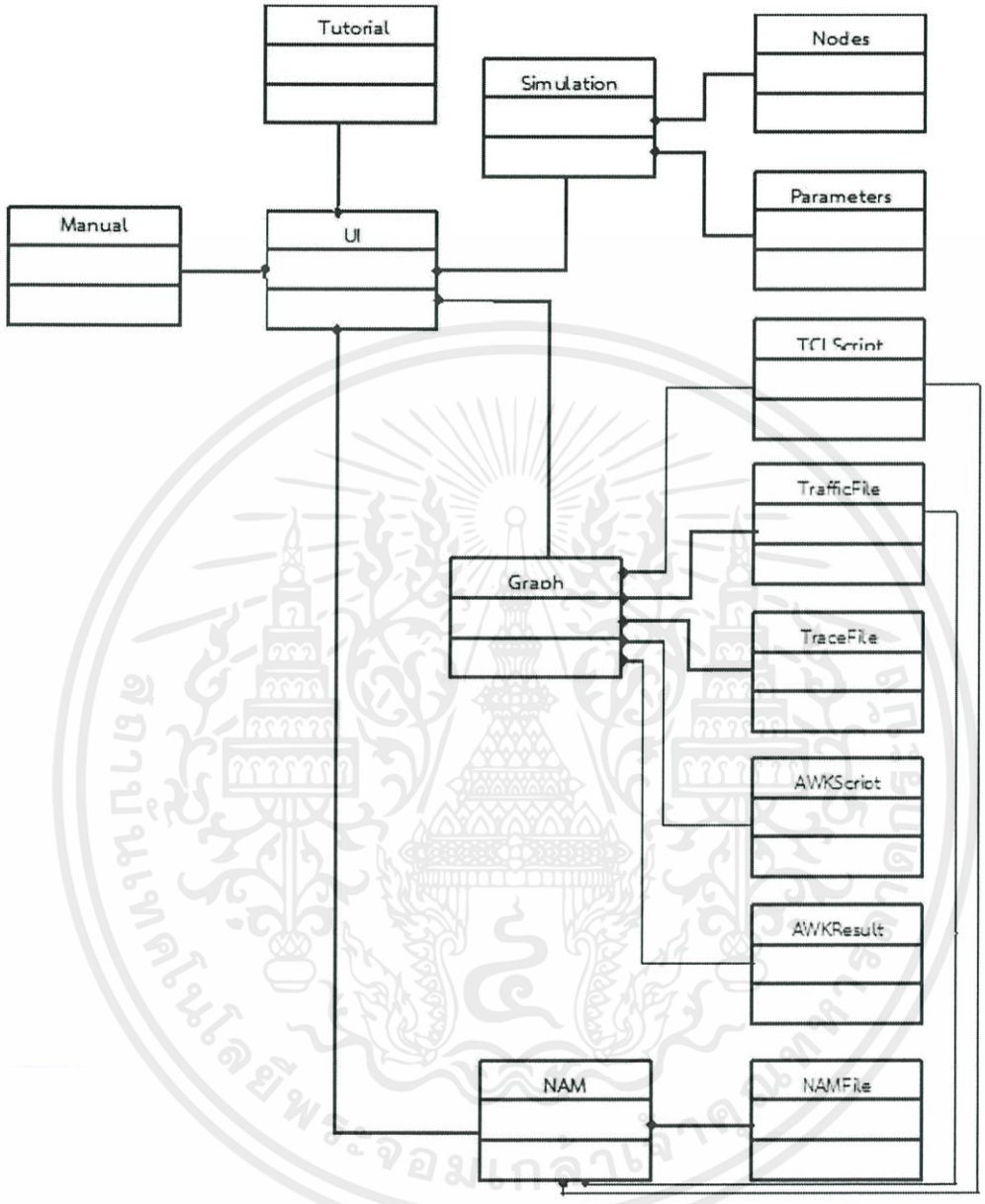
ประกอบไปด้วยคลาสและออปเจ็คต่าง ๆ ดังแสดงในรูปที่ 3.5 รายละเอียดโดยย่อจะมีดังนี้

1. Class UI ส่วนติดต่อผู้ใช้แบบกราฟิก
2. Class Simulation การสร้างการจำลองเครือข่ายโดยผ่านส่วนติดต่อผู้ใช้แบบกราฟิก
3. Class Nodes รายละเอียดของโหนดต่าง ๆ ที่ได้สร้างขึ้นในเครือข่ายจำลอง
4. Class Parameters รายละเอียดของเครือข่ายจำลอง
5. Class Graph กราฟแสดงประสิทธิภาพของเครือข่ายจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Class TCLScript รายละเอียดของ TCL Script
7. Class TrafficFile รายละเอียดของ Traffic File
8. Class TraceFile รายละเอียดของ Trace File ที่ได้จากการรันเอ็นเอส-ทู
9. Class AWKScript รายละเอียดของ AWK Script
10.Class AWKResult รายละเอียดของผลลัพธ์ที่ได้จากการรัน AWK
11. Class NAM เนตเวิกอนิเมเตอแสดงการจำลองเครือข่ายในรูปแบบอนิเมชัน
12. Class NAMFile รายละเอียดของ NAM File
13. Class Manual รายละเอียดต่าง ๆ เกี่ยวกับการใช้งานซอฟต์แวร์
14. Class Tutorial รายละเอียดเนื้อหาเกี่ยวกับโพรโทคอลทีซีพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แผนภาพคลาสไดอะแกรมของเอ็นเอส-ทูยูไอฟอร์ทีซีพี

3.6 ขอบเขตและข้อจำกัดของซอฟต์แวร์ที่พัฒนา

1. ซอฟต์แวร์ทำงานเป็นลักษณะของเดสก์ทอปแอพพลิเคชัน (Desktop Application)
2. ซอฟต์แวร์สามารถใช้งานได้เฉพาะบนระบบปฏิบัติการลินุกซ์ที่ติดตั้งซอฟต์แวร์เอ็นเอส-ทูแล้วเท่านั้น
3. ซอฟต์แวร์สามารถจำลองเฉพาะเครือข่ายแบบมีสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ซอฟต์แวร์ไม่สามารถตรวจสอบความผิดพลาดทางตรรกะ (Logic Error) สำหรับการจำลองเครือข่ายที่สร้างจากผู้ใช้งานได้ เช่น สร้างการเชื่อมต่อไม่ครบ เป็นต้น
5. ผู้ใช้งานจำเป็นต้องมีความรู้ความเข้าใจพื้นฐานในระบบเครือข่ายแบบมีสายและ TCP/IP มาก่อนการใช้งานซอฟต์แวร์
6. เนื่องจากฟังก์ชันต่าง ๆ ในเอ็นเอส-ทูมีความหลากหลายมาก ผู้จัดทำจึงได้นำมาพัฒนาเป็นเพียงบางส่วนเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

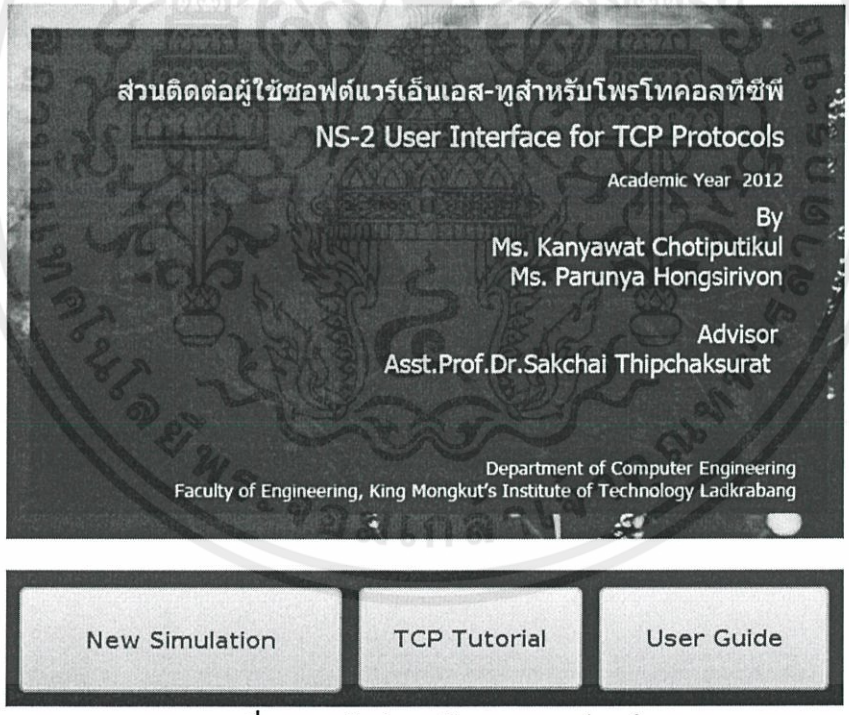
บทที่ 4
การทดสอบซอฟต์แวร์

ในบทนี้จะกล่าวถึงการทดสอบการจำลองเครือข่าย โดยนำซอฟแวร์ไปใช้งานจริงด้วยการป้อนค่าพารามิเตอร์ต่างๆ และจำลองเครือข่ายในตัวอย่างสถานการจำลองต่าง ๆ โดยซอฟต์แวร์สามารถจำลองเครือข่ายได้อย่างถูกต้อง

4.1 การทดสอบส่วนการจำลองเครือข่าย (Network Simulation)

4.1.1 หน้าต่างหลักของซอฟต์แวร์

เมื่อผู้ใช้งานเปิดใช้ซอฟต์แวร์ จะปรากฏหน้าต่างหลักของซอฟต์แวร์ขึ้นมา ดังแสดงในรูปที่ 4.1

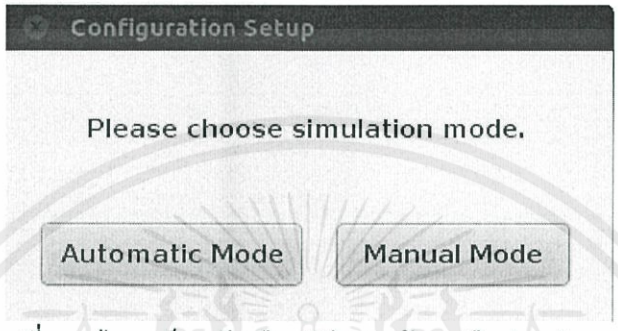


รูปที่ 4.1 หน้าต่างหลักของซอฟต์แวร์

จากรูปที่ 4.1 เป็นหน้าต่างหลักของซอฟต์แวร์ จะมีให้เลือกสร้างการจำลองเครือข่ายแบบมีสายใหม่ (New Simulation) การให้ความรู้เกี่ยวกับโพรโทคอลทีซีพี(TCP Protocols Tutorials) และในส่วนของคู่มือการใช้งาน (User Guide)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

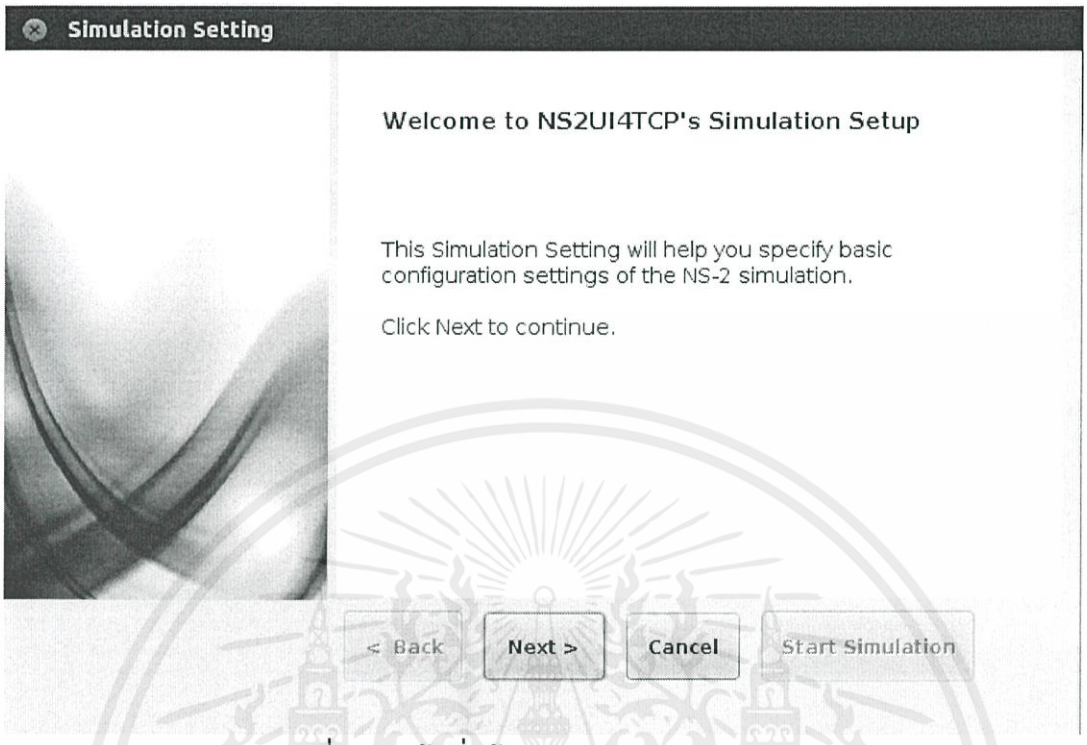
ผู้ใช้งานสามารถสร้างการจำลองเครือข่ายแบบจากปุ่ม New Simulation จะมีหน้าต่างถามรูปแบบการสร้างว่าจะสร้างแบบ Automatic Mode คือการสร้างการจำลองโดยทำตามขั้นตอนตามที่ซอฟต์แวร์แนะนำหรือเป็นแบบ Manual Mode จะเป็นการสร้างการจำลองโดยผู้ใช้เป็นผู้กำหนดโครงสร้างและพารามิเตอร์ในการจำลองต่าง ๆ ดังแสดงในรูปที่ 4.2



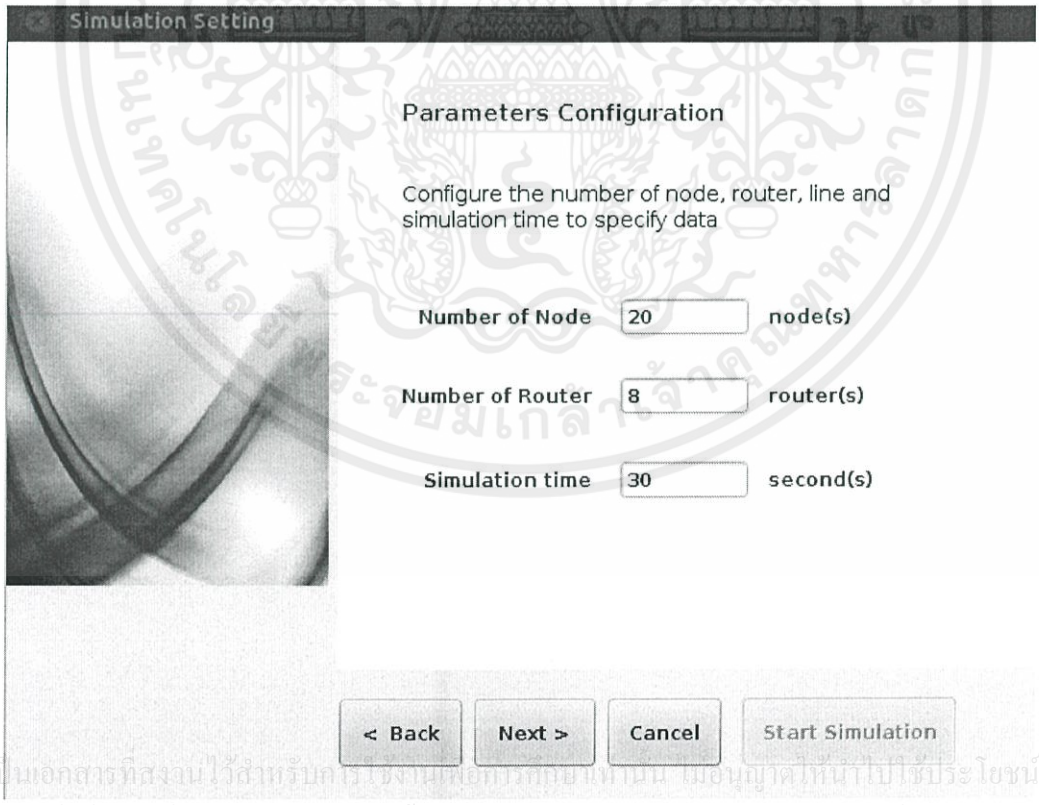
รูปที่ 4.2 ไดอะล็อกตัวเลือกวิธีการสร้างเครือข่ายจำลอง

ถ้าหากผู้ใช้งานต้องการใช้ Automatic Mode ซึ่งเป็นโหมดที่มีการตั้งค่าต่างๆพื้นฐานไว้อยู่แล้ว และมีการดำเนินการเป็นขั้นเป็นตอนช่วยให้ผู้ใช้เข้าใจกระบวนการการทำงานของโปรแกรม ทำให้สร้างแบบจำลองเครือข่ายได้สะดวกยิ่งขึ้น ในรูปที่ 4.3 จะเป็นหน้าเริ่มต้นของ Automatic Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



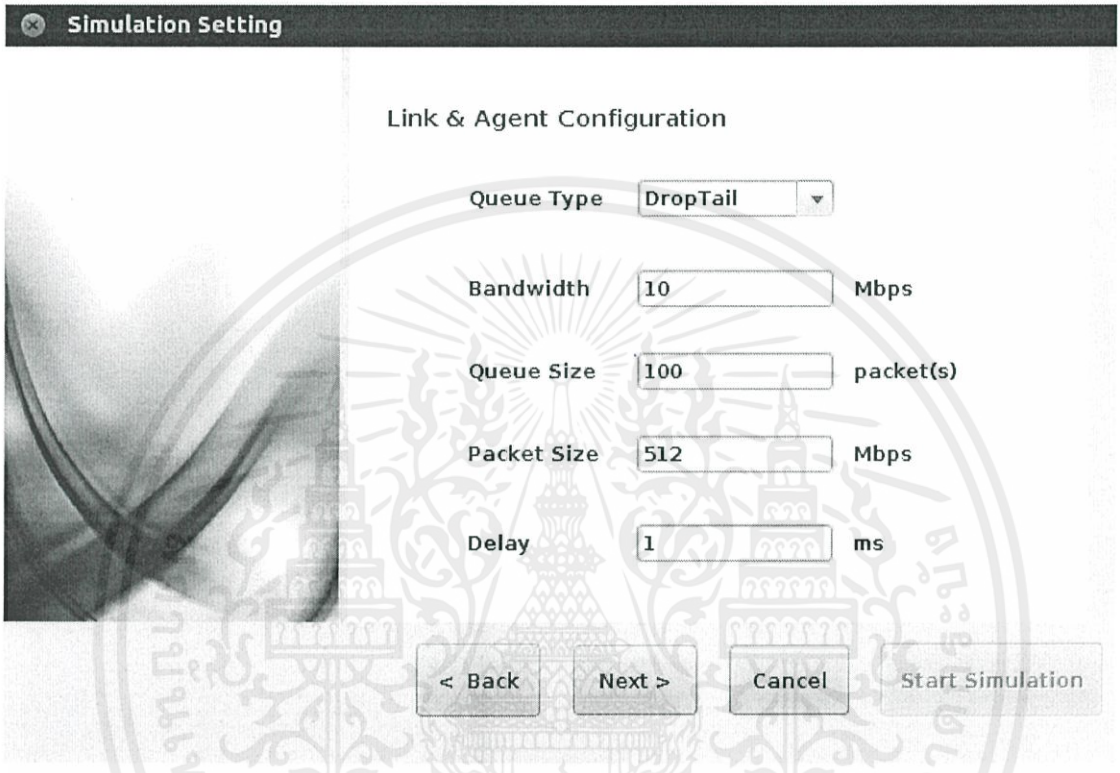
รูปที่ 4.3 หน้าเริ่มต้นของ Automatic Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 ส่วนของ Parameters Configuration

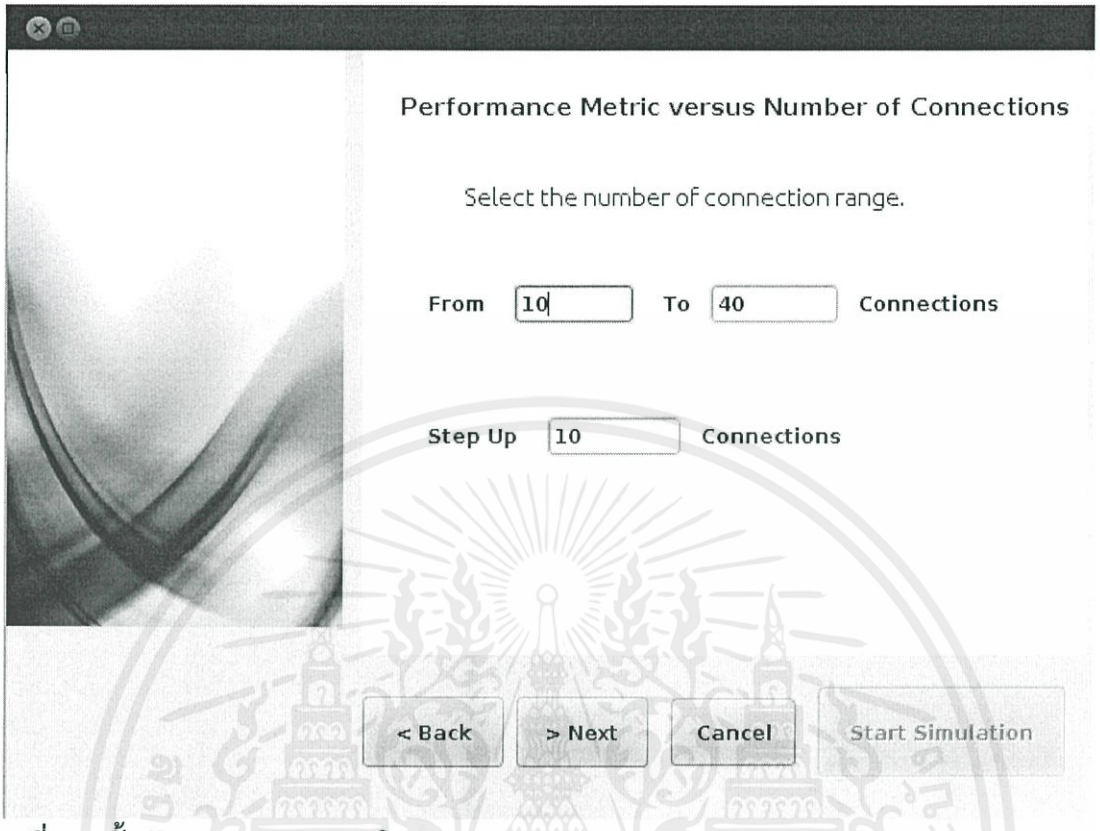
และหน้าต่าง Parameters Configuration ซึ่งไว้สำหรับกำหนดค่าต่าง ๆ ของโหนดแบบตั้งค่าการจำลองเบื้องต้น โดยจะมีให้กำหนดจำนวนโหนด, จำนวนเราท์เตอร์ทั้งหมดที่ใช้ในการจำลองเครือข่าย, เวลาที่ใช้ในการจำลอง ดังแสดงในรูปที่ 4.4



รูปที่ 4.5 ตั้งค่าพารามิเตอร์ของการจำลองเครือข่าย Link & Agent Configuration

ขั้นตอนต่อไปจะเป็นการตั้งค่าพารามิเตอร์ของการจำลองเครือข่ายในส่วนของ Link & Agent Configuration ดังแสดงในรูปที่ 4.5

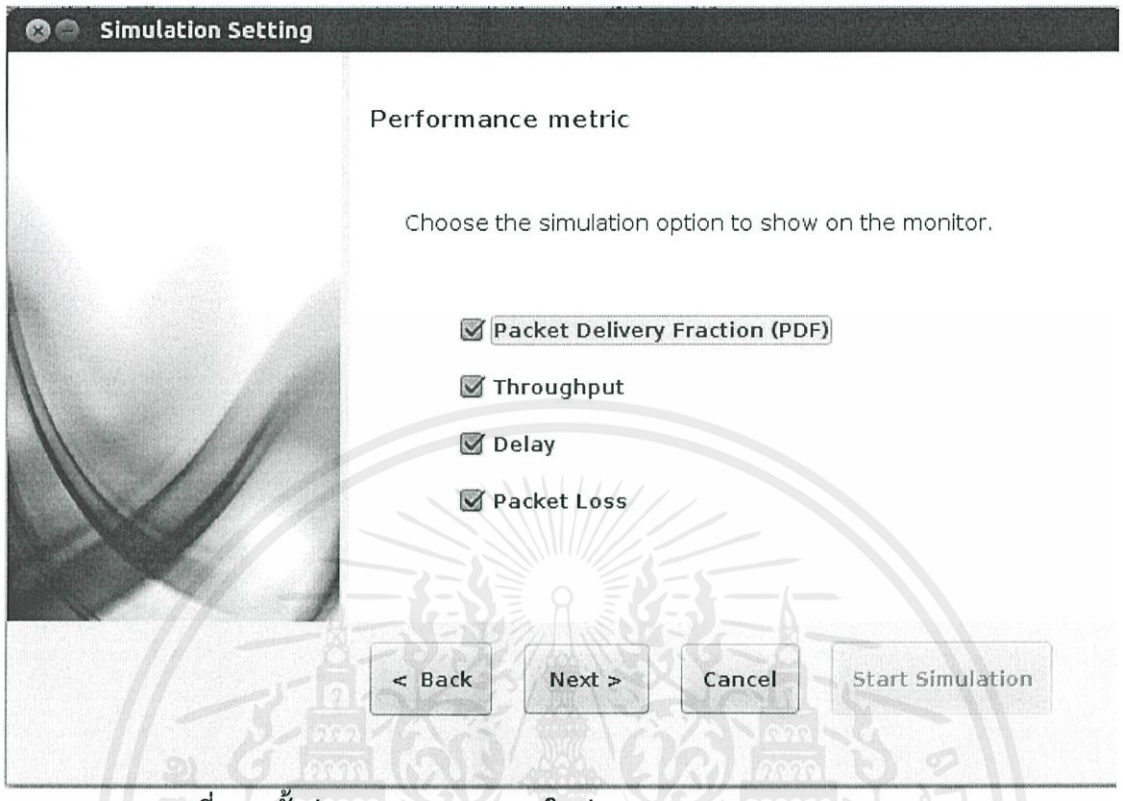
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 ตั้งค่าการแสดงผลกราฟในส่วนของ Performance metric Graph versus Number of Connections

ในส่วนของ Performance Metric versus Number of Connections คือเป็นการเปรียบเทียบประสิทธิภาพต่างๆกับจำนวนของคอนเน็คชัน และในส่วนของ Select the number of connection range เป็นการเลือกช่วงของค่าคอนเน็คชันที่ต้องการ จากรูปคือค่าคอนเน็คชันเริ่มจาก 10 คอนเน็คชันจนถึง 40 คอนเน็คชัน โดยมีการเพิ่มค่าทีละ 10 คอนเน็คชัน ดังแสดงในรูปที่ 4.6

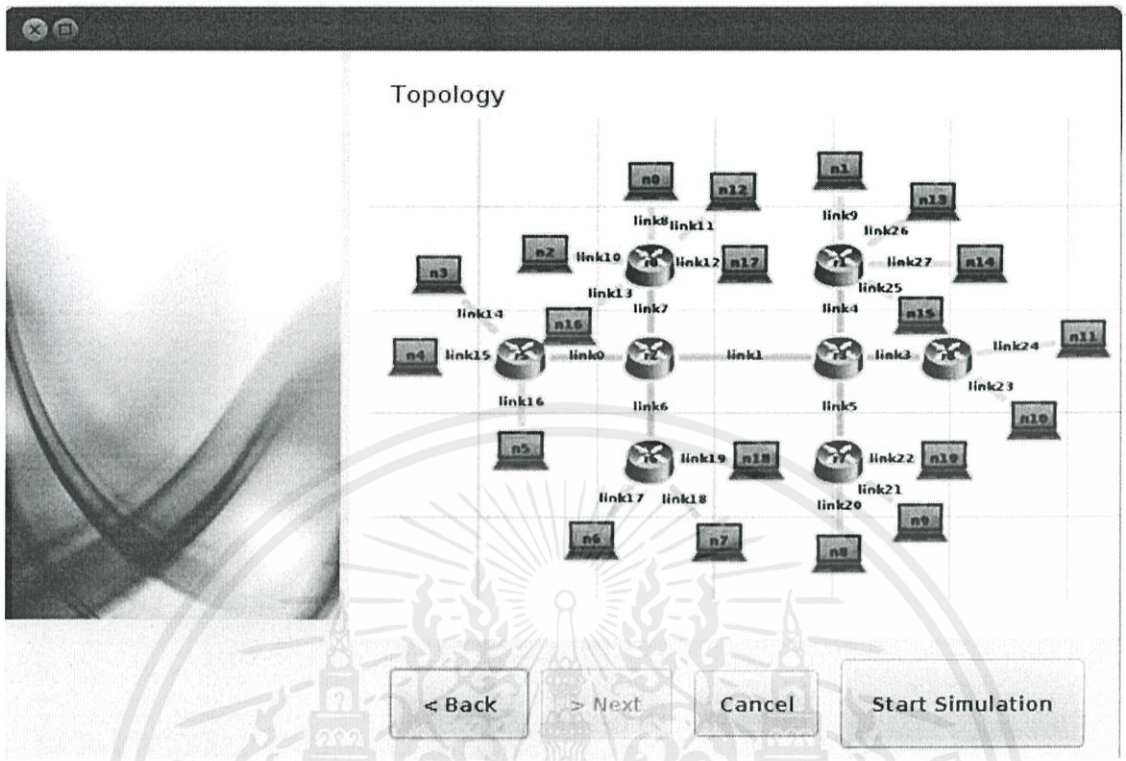
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ตั้งค่าการแสดงผลกราฟในส่วนของ Performance metric

ขั้นตอนถัดไปคือการตั้งค่าการแสดงผลของกราฟ Performance metric จะให้ผู้ใช้เลือกว่าต้องการดูประสิทธิภาพของกราฟในรูปแบบของพารามิเตอร์ใดบ้างโดยมีให้เลือกดังนี้คือ Packet Delivery Fraction (PDF), End to End Delay, Packet Loss และ Throughput ดังแสดงในรูปที่ 4.7

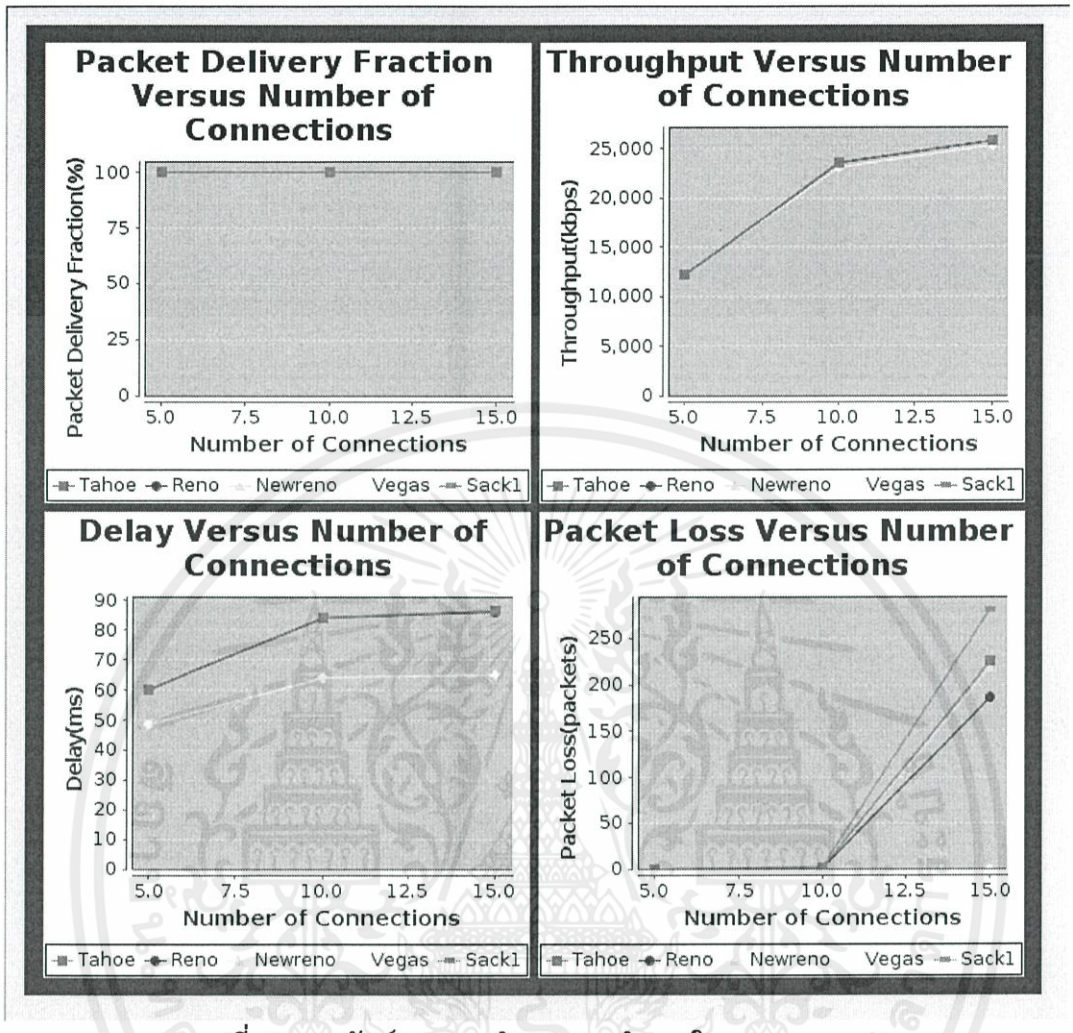
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 การแสดงผลกราฟในส่วนของโทโพโลยี

ขั้นตอนต่อไปเป็นส่วนของการแสดงผลโทโพโลยีที่ได้จากการตั้งค่าการจำลองเครือข่ายแบบมีสาย ดังแสดงในรูปที่ 4.8

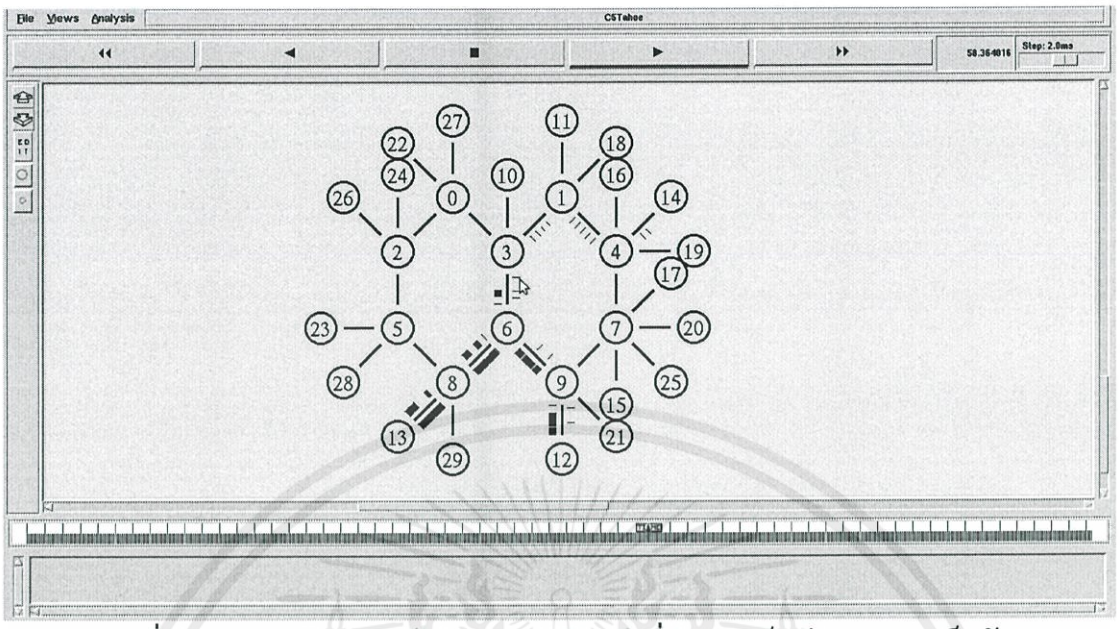
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผลลัพธ์จากการจำลองการทำงานในรูปแบบกราฟ

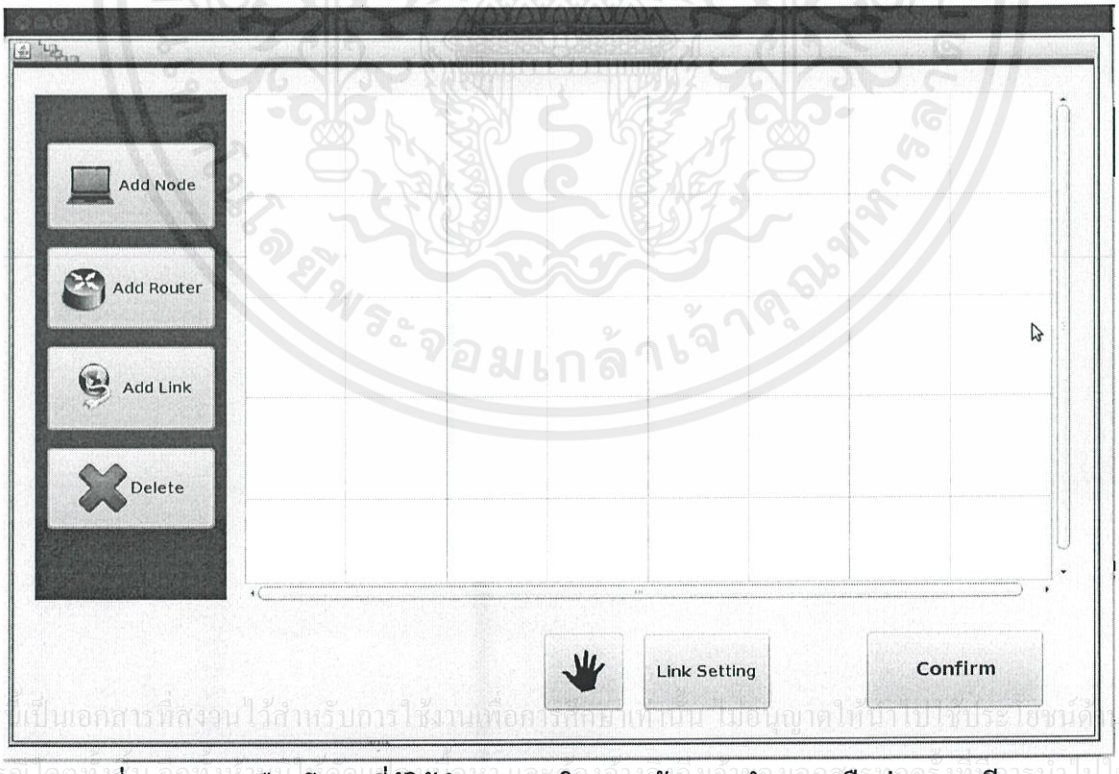
ซึ่งจากรูปจะมีรูปกราฟทั้งหมด 4 รูปคือเปรียบเทียบประสิทธิภาพทั้ง Packet Delivery Fraction (PDF), End to End Delay, Packet Loss และ Throughput ซึ่งในแต่ละรูปจะมีกราฟเส้นทั้งหมด 5 เส้น แสดงผลประสิทธิภาพของโพรโทคอล 5 โพรโทคอล ได้แก่ TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas และ TCP Sack และในแทบด้านขวาจะแสดงลิสต์ของอนิเมชันที่สามารถเลือกรันได้ของแต่ละโพรโทคอล ในแต่ละพารามิเตอร์ที่ต้องการเลือกแสดงโดยการดับเบิ้ลคลิกไปที่รายการนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ตัวอย่างผลลัพธ์ในรูปของแอนิเมชันที่ค่าคอนเน็คชัน 15 คอนเน็คชัน

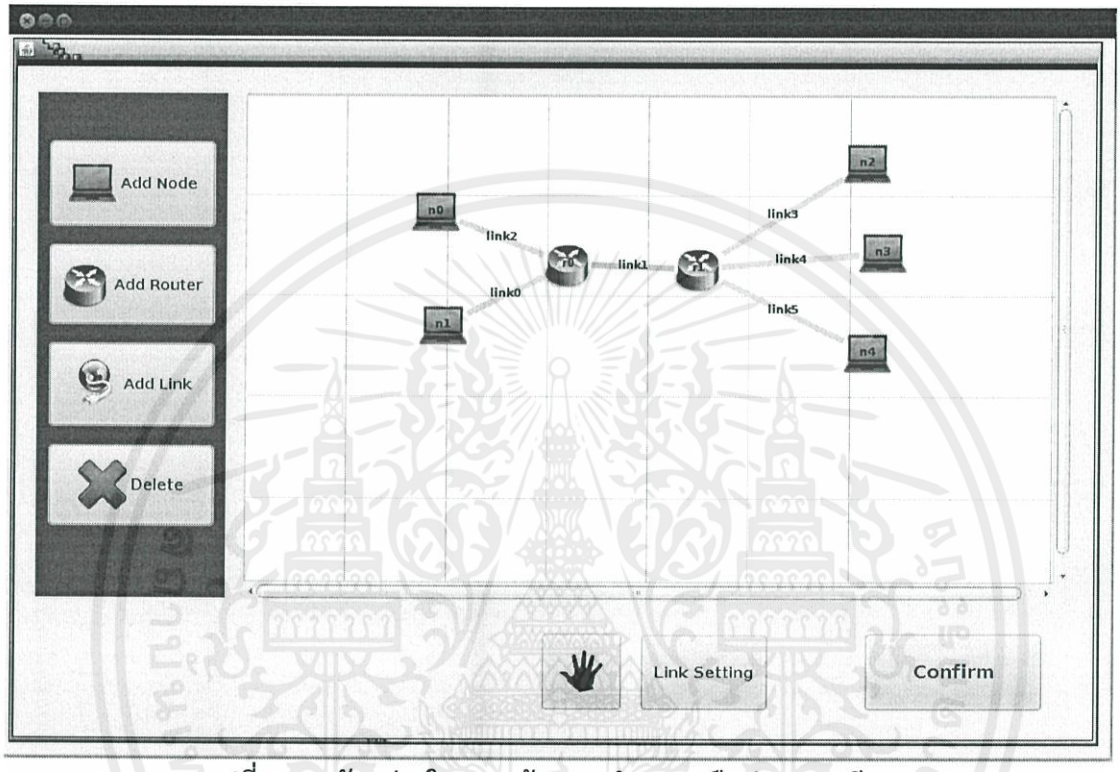
จากรูปที่ 4.10 เป็นตัวอย่างการแสดงผลแบบอนิเมชันที่ค่าคอนเน็คชันเป็น 15 คอนเน็คชัน เพื่อดูการรับส่งแพ็คเก็ตข้อมูลต่างๆ ว่ามีลักษณะอย่างไร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.11 การเลือกโหมดที่ผู้ใช้กำหนดเองในการสร้างการจำลองเครือข่ายแบบมีสาย

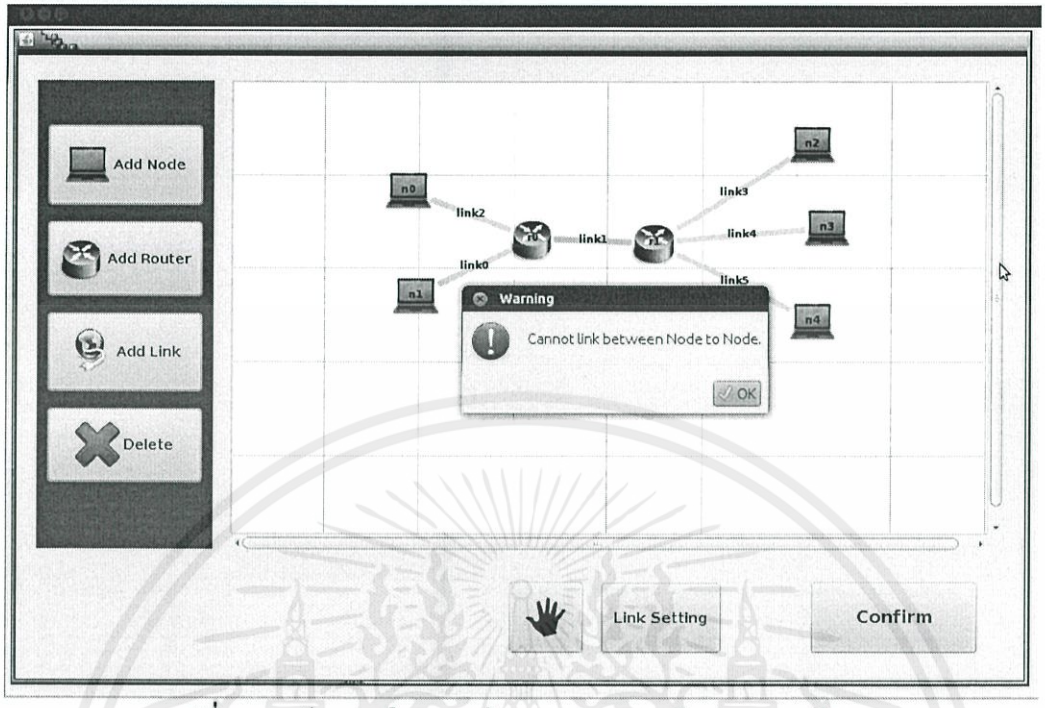
จากรูปที่ 4.11 เลือกโหมดทั่วไป (Manual Mode) ในการจำลองเครือข่ายแบบมีสาย ผู้ใช้งานสามารถสร้างโหนดโดยเลือกปุ่ม Add Node สร้างเราท์เตอร์โดยเลือกปุ่ม Add Router ลากสายโดยเลือกปุ่ม Add Link สามารถเปลี่ยนตำแหน่งโดยเลือกปุ่มมือ และลบโดยเลือกปุ่ม Delete



รูปที่ 4.12 ตัวอย่างในการสร้างการจำลองเครือข่ายแบบมีสาย

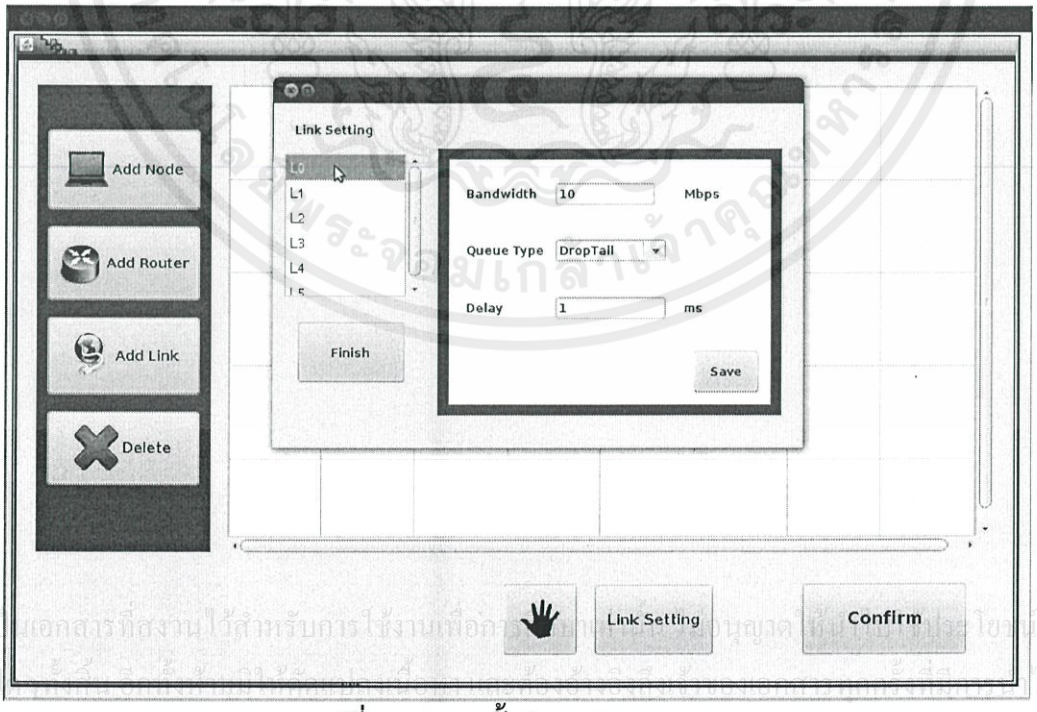
จากรูปที่ 4.12 เป็นตัวอย่างในการเลือกโหมดทั่วไป (Manual Mode) ในการจำลองเครือข่ายแบบมีสาย โดยสร้างเราท์เตอร์ขึ้นมา 2 เราท์เตอร์ สร้างโหนด 5 โหนดและลากสายเชื่อมต่อเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ตัวอย่างในการสร้างการจำลองเครือข่ายแบบมีสาย

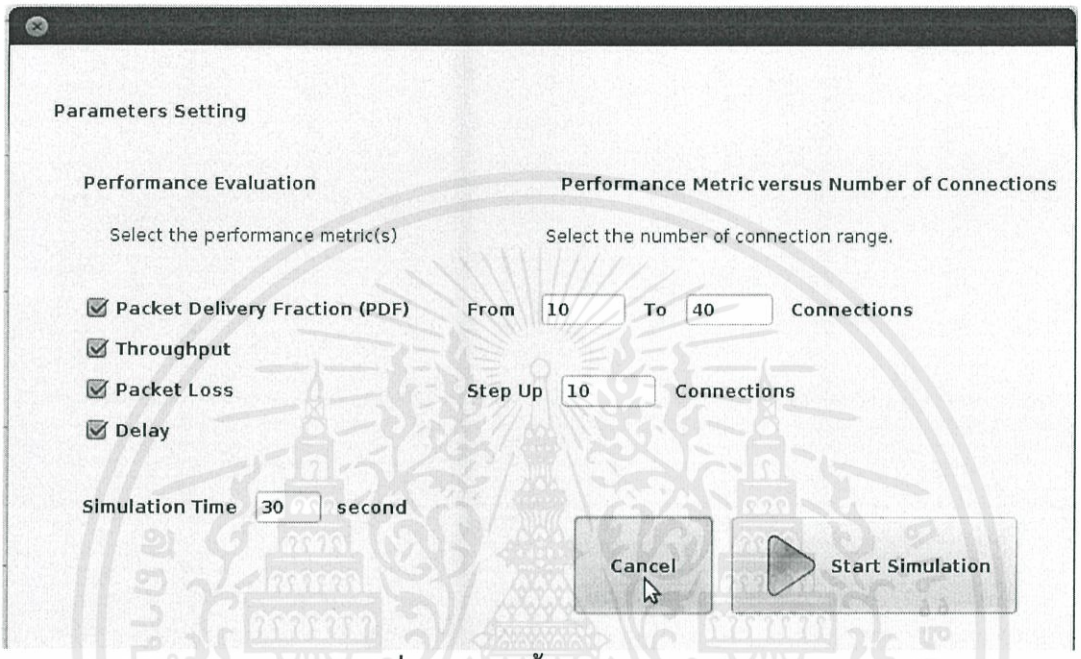
ในการจำลองเครือข่าย ผู้ใช้งานไม่สามารถสร้างสายระหว่างโหนดได้ จะมีหน้าต่างขึ้นมาแจ้งเตือน ดังแสดงในรูปที่ 4.13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

รูปที่ 4.14 การตั้งค่า Link Setting

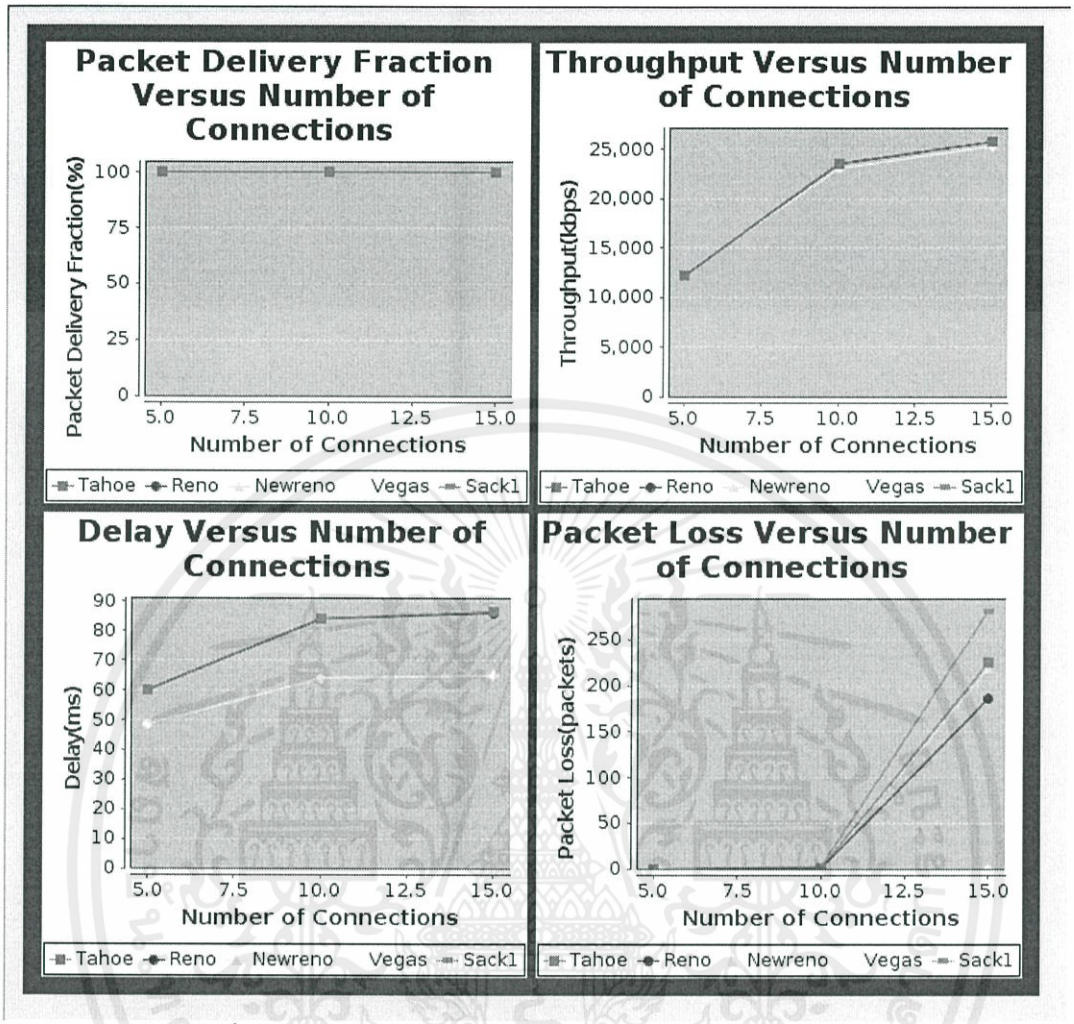
จากรูปที่ 4.14 เมื่อคลิดปุ่ม Link Setting จะเป็นการตั้งค่าพารามิเตอร์ในสายแต่ละเส้นที่ทำการสร้างขึ้นมา



รูปที่ 4.15 การตั้งค่าพารามิเตอร์

จากรูปที่ 4.14 เมื่อคลิกปุ่ม Confirm จะเป็นการตั้งค่าพารามิเตอร์ โดยส่วนของ Performance Evaluation จะให้ผู้ใช้เลือกว่าต้องการดูประสิทธิภาพของกราฟในรูปแบบของพารามิเตอร์ใดบ้างโดยมีให้เลือกดังนี้คือ Packet Delivery Fraction (PDF), End to End Delay, Packet Loss และ Throughput ส่วนของ Performance Metric versus Number of Connections เป็นการเปรียบเทียบประสิทธิภาพต่างๆกับจำนวนของคอนเน็คชัน และในส่วนของ Select the number of connection range เป็นการเลือกช่วงของค่าคอนเน็คชันที่ต้องการ จากรูปคือค่าคอนเน็คชันเริ่มจาก 10 คอนเน็คชันจนถึง 40 คอนเน็คชัน โดยมีการเพิ่มค่าทีละ 10 คอนเน็คชัน ดังรูป 4.15 และมีการตั้งค่าเวลาในการจำลอง

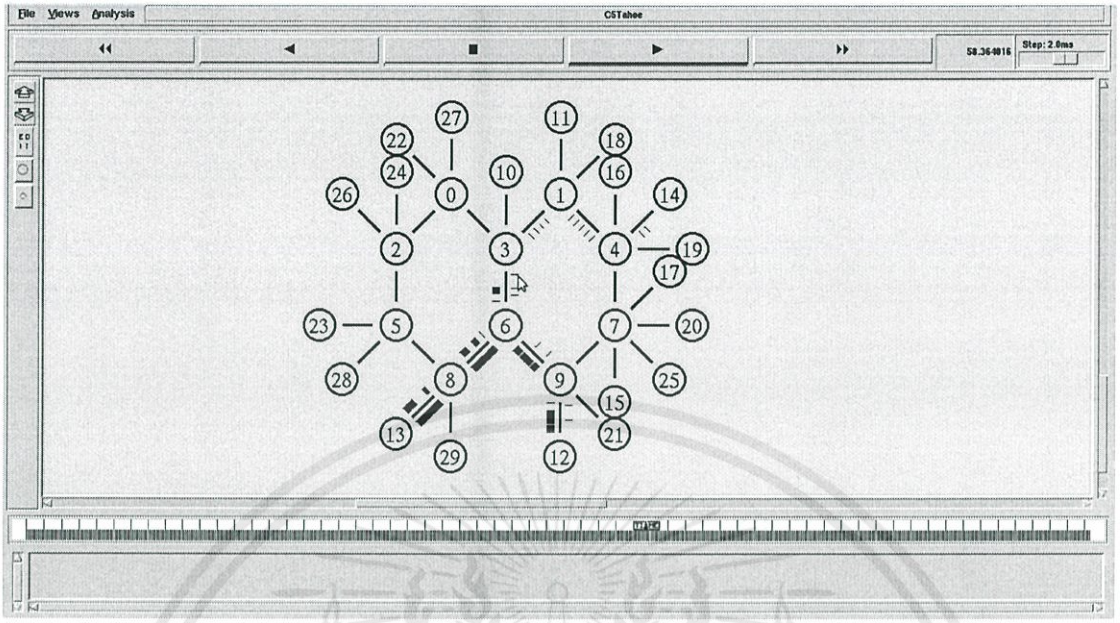
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ผลลัพธ์จากการจำลองการทำงานในรูปแบบกราฟ

ซึ่งจากรูปที่ 4.16 จะมีรูปกราฟทั้งหมด 4 รูปคือเปรียบเทียบประสิทธิภาพทั้ง Packet Delivery Fraction (PDF), End to End Delay, Packet Loss และ Throughput ซึ่งในแต่ละรูปจะมีกราฟเส้นทั้งหมด 5 เส้น แสดงผลประสิทธิภาพของโพรโทคอล 5 โพรโทคอล ได้แก่ TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas และ TCP Sack และในแทบด้านขวาจะแสดงลิสต์ของอนิเมชันที่สามารถเลือกรันได้ของแต่ละโพรโทคอล ในแต่ละพารามิเตอร์ที่ต้องการเลือกแสดงโดยการดับเบิ้ลคลิกไปที่รายการนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 ตัวอย่างผลลัพธ์ในรูปของแอนิเมชัน

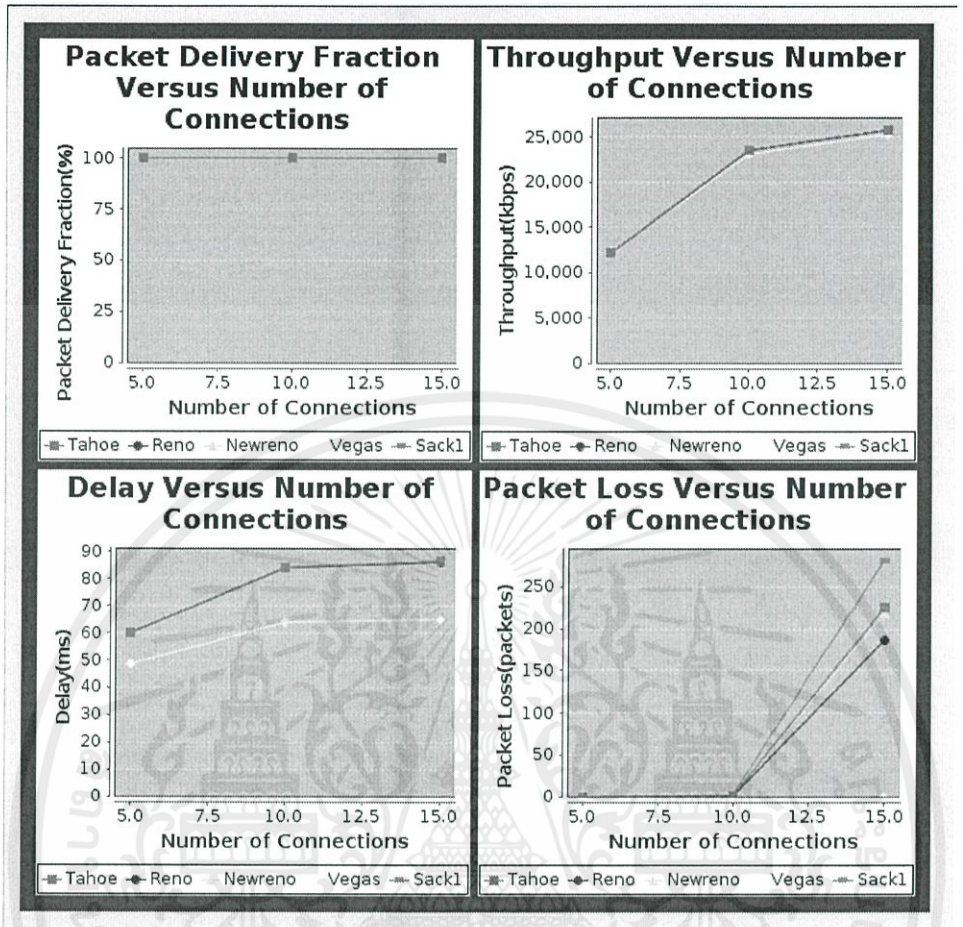
จากรูปที่ 4.17 เป็นตัวอย่างการแสดงผลแบบอนิเมชัน เพื่อดูการรับส่งแพ็คเก็ตข้อมูลต่างๆ ว่ามีลักษณะอย่างไร

4.2 การจำลองเครือข่ายในตัวอย่างสถานการจำลองต่างๆ

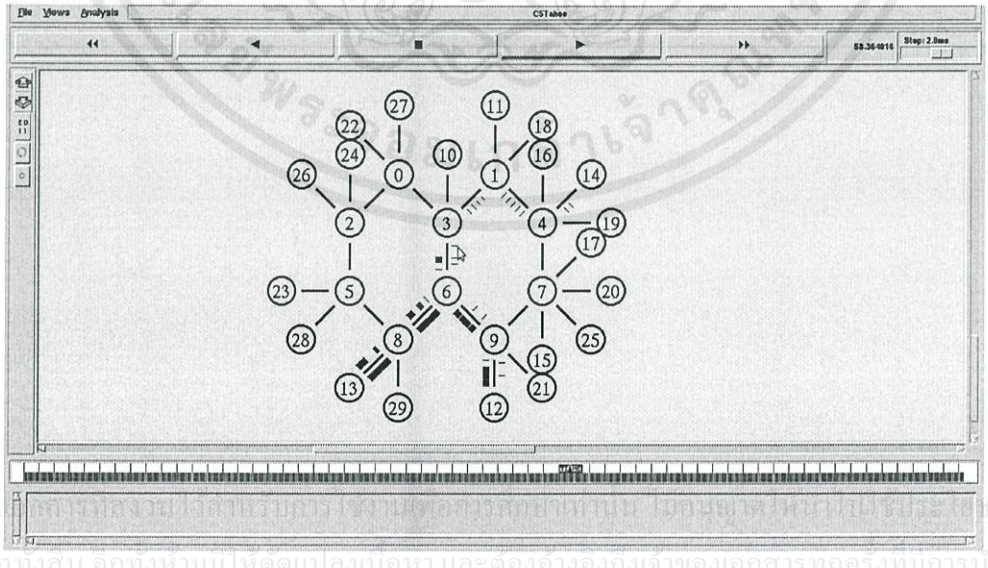
4.2.1 สถานการณ์จำลองที่ 1

เมื่อต้องการแสดงกราฟตามแกนการเชื่อมต่อ โดยเริ่มตั้งแต่ 10 ถึง 40 คอนเน็คชันของทั้งเครือข่ายมีโหนดในการจำลอง 20 โหนด และเราท์เตอร์ 10 เราท์เตอร์ โดยเปรียบเทียบระหว่างโพรโทคอลได้แก่ TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas และ TCP Sack ใช้เวลาในการจำลอง 100 วินาที ซึ่งแนวโน้มกราฟ PDF, Delay, Loss และ Throughput แสดงตามรูปที่ 4.18 และมีการแสดงผลแบบแอนิเมชันที่ค่าคอนเน็คชันเป็น 15 คอนเน็คชัน เพื่อดูการรับส่งแพ็คเก็ตข้อมูลต่างๆ ว่ามีลักษณะอย่างไร ดังแสดงในรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 ผลลัพธ์จากการจำลองการทำงานในรูปแบบกราฟ
ของสถานการณ์จำลองที่ 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.19 ตัวอย่างผลลัพธ์ในรูปของแอนิเมชันที่ค่าคอนเน็คชัน 15 คอนเน็คชัน

บทที่ 5
บทสรุป

เน็ตเวิร์คซิมมูเลเตอร์รุ่นที่ 2 (เอ็นเอส-ทู) เป็นซอฟต์แวร์จำลองการทำงานของเครือข่ายที่มีประสิทธิภาพมากแต่เอ็นเอส-ทูเป็นซอฟต์แวร์ที่มีความซับซ้อนมาก การใช้งานเอ็นเอส-ทูต้องเขียนสคริปต์ด้วยภาษาทีซีแอล ผู้ใช้ต้องมีความเข้าใจถึงโครงสร้างของฟังก์ชันต่าง ๆ ที่เกียวข้องที่เป็นภาษาซีพลัสพลัส และต้องมีความรู้เกี่ยวกับภาษาทีซีแอล ในการจำลองการทำงานผู้ใช้ต้องกำหนดพารามิเตอร์และสภาพแวดล้อมต่าง ๆ เพื่อทำการรันเอ็นเอส-ทู ผลจากการรันจะได้กราฟเพียง 1 จุด ถ้าต้องการกราฟที่เป็นเส้น ผู้ใช้ต้องทำการรันหลายครั้งทำให้ไม่สะดวกต่อการใช้งาน ดังนั้นเพื่อให้ผู้ใช้สามารถจำลองการทำงานของเครือข่ายได้ง่ายและสะดวกยิ่งขึ้นโดยที่ผู้ใช้ไม่จำเป็นต้องมีพื้นฐานภาษาทีซีแอลและโครงสร้างของฟังก์ชันของเอ็นเอส-ทู ทางผู้จัดทำจึงได้พัฒนาส่วนติดต่อผู้ใช้สำหรับซอฟต์แวร์เน็ตเวิร์คซิมมูเลเตอร์โดยเรียกซอฟต์แวร์ที่ทางผู้จัดทำพัฒนาว่า “เอ็นเอส-ทูยูไอฟอร์ทีซีพี” ในการจำลองการทำงานของเครือข่ายด้วยเอ็นเอส-ทู ผู้ใช้งานเพียงแต่ทำการกำหนดสภาพแวดล้อมต่าง ๆ ที่ต้องการผ่านเอ็นเอส-ทูยูไอฟอร์ทีซีพี หลังจากนั้นเอ็นเอส-ทูยูไอฟอร์ทีซีพีจะทำการรันเอ็นเอส-ทูให้โดยอัตโนมัติและแสดงผลการจำลองการทำงานในเทอมของประสิทธิภาพของเครือข่ายสื่อสารในรูปของกราฟต่าง ๆ และผู้ใช้งานยังสามารถศึกษาและดูการจำลองเครือข่ายแบบอนิเมชันผ่านเน็ตเวิร์กแอนิเมเตอร์ เพื่อให้ผู้ใช้งานเห็นภาพและเข้าใจในเครือข่ายจำลองมากขึ้น

5.1 ผลที่ได้จากการทำโครงงาน

1. มีการออกแบบส่วนติดต่อผู้ใช้ของซอฟต์แวร์ที่โดดเด่นสวยงาม และใช้งานได้ง่าย โดยมีการออกแบบอย่างเป็นระบบ และออกแบบให้ผู้ใช้สามารถใช้งานได้อย่างเป็นขั้นเป็นตอน ทำให้ผู้ใช้งานเข้าใจในกระบวนการทำงานของโปรแกรม
2. มีคู่มือการใช้งานและรายละเอียดต่างๆที่เกี่ยวข้องกับการใช้งานโปรแกรม ช่วยให้ผู้ใช้งานสามารถใช้งานโปรแกรมได้ง่าย
3. มีส่วนของทฤษฎีและความรู้ต่าง ๆ เกี่ยวกับโพรโทรคอลทีซีพี ทำให้ผู้ใช้งานสามารถเข้าใจเนื้อหาเกี่ยวกับโพรโทรคอลทีซีพีได้มากยิ่งขึ้น
4. ผู้ใช้งานสามารถเลือกดูการแสดงผลที่เป็นกราฟ หรือแอนิเมชัน เฉพาะที่พารามิเตอร์หรือโพรโทคอลที่ผู้ใช้ต้องการดูได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ซอฟต์แวร์มีการเก็บรูปภาพและไฟล์ต่างๆอย่างเป็นระเบียบแยกตามโพรโทคอล และการตั้งชื่อไฟล์ต่างๆที่สอดคล้องกับการจำลองเครือข่าย ทำให้ผู้ใช้งานสามารถเรียกดูได้อย่างสะดวก
6. ซอฟต์แวร์มีโหมดการทำงานให้เลือกสองโหมดคือแบบผู้ใช้กำหนดเอง และแบบอัตโนมัติเพื่อเป็นทางเลือกสำหรับผู้ใช้งาน ทำให้สามารถใช้งานได้ง่าย
7. ผู้จัดทำได้รับความรู้ความเข้าใจในโพรโทคอลทีซีพีมากยิ่งขึ้น นอกจากนี้ยังเข้าใจในการทำงานต่างๆของซอฟต์แวร์เอ็นเอส-ทู และสามารถใช้งานซอฟต์แวร์เอ็นเอส-ทูได้อย่างดี รวมไปถึงทักษะประสบการณ์ในการเขียนโปรแกรมโดยใช้ภาษาจาวา เพื่อใช้ในการพัฒนาซอฟต์แวร์ส่วนติดต่อผู้ใช้เอ็นเอส-ทูสำหรับโพรโทคอลทีซีพีและการออกแบบส่วนติดต่อผู้ใช้ที่เหมาะสม

5.2 ปัญหาอุปสรรคและแนวทางแก้ไข

1. เอ็นเอส-ทูมีฟังก์ชันการทำงานต่าง ๆ เป็นจำนวนมากถึงแม้จะเป็นแค่ในส่วนของเครือข่ายแบบมีสาย และบางฟังก์ชันที่มีก็ไม่มีคำอธิบายวิธีการใช้งานและผลลัพธ์มาให้ ผู้จัดทำจึงจำเป็นต้องหาแหล่งศึกษาและทดลองการทำงานเองจนกว่าจะได้ข้อสรุป
2. การศึกษาโครงสร้างของสคริปต์ทีซีแอลและสคริปต์ AWK ใช้เวลานานเนื่องจากผู้จัดทำไม่มีความรู้และพื้นฐานเกี่ยวกับสคริปต์ทีซีแอลและสคริปต์ AWK มาก่อนจึงทำให้เกิดความล่าช้าในการพัฒนาซอฟต์แวร์
3. แนวทางการพัฒนาซอฟต์แวร์ในลักษณะนี้ออกแบบและทำได้ค่อนข้างยากและโครงงานของรุ่นพี่ที่ทำไว้มีข้อบกพร่องหลายแห่ง ทำให้ผู้จัดทำต้องหาวิธีพัฒนาซอฟต์แวร์เพิ่มเติมและแนวทางแก้ไขปัญหาข้อบกพร่องของโครงงานที่รุ่นพี่ทำไว้มากขึ้น
4. เนื่องจากข้อจำกัดของการใช้งานซอฟต์แวร์เอ็นเอส-ทูทำให้ซอฟต์แวร์สามารถทำงานได้เฉพาะบนระบบปฏิบัติการลินุกซ์เท่านั้น แต่ผู้จัดทำได้เลือกพัฒนาซอฟต์แวร์โดยภาษาจาวาที่สามารถรันได้ทุกแพลตฟอร์ม ดังนั้นการพัฒนาซอฟต์แวร์ต่อจากโครงงานนี้จึงสามารถแก้ไขเฉพาะส่วนของการใช้งานร่วมกับเอ็นเอส-ทูโดยไม่ต้องแก้ไขโค้ดของซอฟต์แวร์ใหม่ทั้งหมดนอกจากว่าจะต้องการพัฒนาเพื่อต่อยอดความสามารถของซอฟต์แวร์ให้ดียิ่งกว่าเดิม
5. ขอบเขตโพรโทคอลทีซีพีในการทำโครงงานนี้ บางโพรโทคอลเป็นโพรโทคอลที่ผู้จัดทำไม่มีความรู้และพื้นฐานมาก่อน ทำให้ผู้จัดทำต้องศึกษา หาความรู้เพิ่มเติมและทำความเข้าใจเกี่ยวกับโพรโทคอลอย่างละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ในส่วนของเน็ตเวิร์คแอนิเมเตอร์แบบมีสายการกำหนดทิศทางของโหนดและเชื่อมสายต่าง ๆ เข้าด้วยกันแบบสุ่ม จะได้โทโพโลยีที่ซับซ้อนมาก จึงได้มีการกำหนดรูปแบบของเราท์เตอร์ในส่วนของการทำงานแบบสุ่ม
7. ในการจำลองโพรโทคอลทีซีพีทั้ง 5 โพรโทคอล ใช้เวลานานมากในการซิมมูเลเตอร์เนื่องจากรันบนระบบปฏิบัติการ Ubuntu ผ่านโปรแกรม VMware Workstation

5.3 แนวทางในการพัฒนาต่อ

1. เพิ่มความสามารถในการจำลองเครือข่ายสำหรับโหนดไร้สายที่สามารถเชื่อมต่อกับเครือข่ายมีสายได้
2. เพิ่มในส่วนการแสดงผลให้ผู้ใช้สามารถรู้การทำงานว่าต้องใช้เวลาเท่าไรในการจำลองเครือข่าย เนื่องจากในการจำลองแต่ละครั้งใช้เวลานานมาก
3. ปรับปรุงเน็ตเวิร์กแอนิเมเตอร์ให้มีความสวยงามมากยิ่งขึ้น เพื่อช่วยให้ผู้ใช้งานสามารถดูกระบวนการทำงานของเครือข่ายได้อย่างชัดเจนมากยิ่งขึ้น
4. นำโพรโทรคอลอื่นๆนอกเหนือจากโพรโทคอลทีซีพีมาพัฒนาต่อ
5. เพิ่มความสามารถซอฟต์แวร์ให้กลายเป็นชุดติดตั้ง (Setup software) ทำให้ผู้ใช้ไม่ต้องติดตั้งเอ็นเอส-ทู ด้วยตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] วจิตติมา นิตยวรรณ, โสพส ชคัตตรยาพงษ์ และอุไรรัตน พึ่งสุนทรบัตร์. "**Network Simulator (NS2).**" ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. 2550.

[2] วิกิพีเดีย สารานุกรมเสรี. "**ภาษาจาวา.**" [Online]. Available : http://th.wikipedia.org/wiki/ภาษาจาวา. 2554.

[3] วิกิพีเดีย สารานุกรมเสรี. "**โมเดลโอเอสไอ.**" [Online]. Available : http://th.wikipedia.org/wiki/โมเดลโอเอสไอ. 2554.

[4] UltraMod(นามแฝง). "**ความรู้เบื้องต้นของภาษาจาวา.**" [Online]. Available : http://happyeverytime.exteen.com/20090608/entry. 2552.

[5] mindphp.com. "**TCP/IP คืออะไร.**" [Online]. Available : http://www.mindphp.com/คู่มือ/73-คืออะไร/2042-tcp-ip-คืออะไร.html. 2554.

[6] Mathis M., Mahdavi J. "**TCP Selective Acknowledgment Options.**" Request for Comments 2018. 1996.

[7] Floyd S. "**The NewReno Modification to TCP's Fast Recovery Algorithm.**" Request for Comments 3782. 2004.

[8] The Ubuntu Manual Team. "**Ubuntu Manual Project.**" [Online]. Available : http://ubuntu-manual.org. 2012.

[9] Wikipedia. "**TCP congestion avoidance algorithm.**" [Online]. Available : http://en.wikipedia.org/wiki/TCP_congestion_avoidance_algorithm. 2012.

[10] Jeonghoon M., Richard J. "**Analysis and Comparison of TCP Reno and Vegas.**" [Online]. Available : http://netlab.caltech.edu/FAST/references/Mo_comparisonwithTCPReno.pdf. 1998.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
การติดตั้งโปรแกรม

ก.1 การติดตั้งโปรแกรม NS-2.35

NS-2.35 เมื่อติดตั้งเสร็จสมบูรณ์แล้วจะสามารถใช้งานได้ดังรูป ก.1 ซึ่งมีขั้นตอนการติดตั้งดังนี้

1. ดาวน์โหลดและติดตั้งแพ็คเก็ตที่จำเป็น โดยพิมพ์คำสั่งดังต่อไปนี้

   $ sudo apt-get update

   $ sudo apt-get install build-essential autoconf automake libxmu-dev

2. ดาวน์โหลดตัวโปรแกรม NS-2.35 จาก

   http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/ns-allinone-2.35.tar.gz/download

3. แตกไฟล์ ns-allinone-2.35.tar.gz โดยพิมพ์คำสั่งต่อไปนี้

   $ tar -zxvf ns-allinone-2.35.tar.gz

4. ติดตั้งโปรแกรม NS-2.35 โดยพิมพ์คำสั่งต่อไปนี้

   $ cd ns-allinone-2.35

   $ ./install

5. กำหนด Path ของโปรแกรม NS-2.35 โดยพิมพ์คำสั่งต่อไปนี้

   $ gedit ~/.bashrc

   และแก้ไข ดังต่อไปนี้

   # LD_LIBRARY_PATH

   OTCL_LIB=/your/path/ns-allinone-2.35/otcl-1.14

   NS2_LIB=/your/path/ns-allinone-2.35/lib

   X11_LIB=/usr/X11R6/lib

   USR_LOCAL_LIB=/usr/local/lib

   Export

   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_LIB

   # TCL_LIBRARY

   TCL_LIB=/your/path/ns-allinone-2.35/tcl8.5.10/library

   USR_LIB=/usr/lib

   export TCL_LIBRARY=$TCL_LIB:$USR_LIB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# PATH
XGRAPH=/your/path/ns-allinone-2.35/bin:/your/path/ns-allinone-
2.35/tcl8.5.10/unix:/your/path/ns-allinone-2.35/tk8.5.10/unix
NS=/your/path/ns-allinone-2.35/ns-2.35/
NAM=/your/path/ns-allinone-2.35/nam-1.15/
export PATH=$PATH:$XGRAPH:$NS:$NAM
```

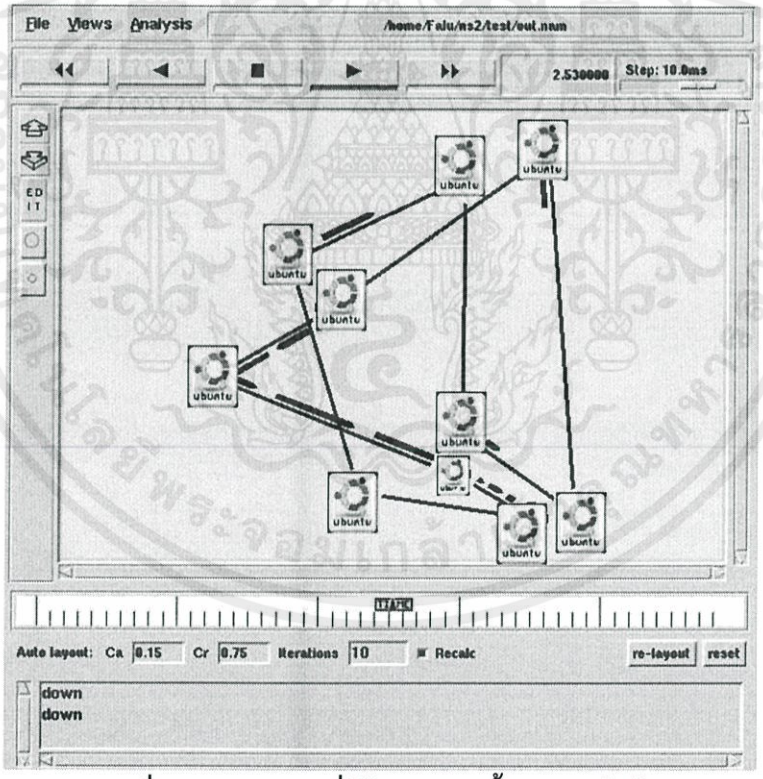
6. หลังจากแก้ไข Path เรียบร้อยแล้ว ให้พิมพ์คำสั่งเพิ่มบันทึกข้อมูลที่แก้ไข

   ```
   $ source ~/.bashrc
   ```

7. ตรวจสอบความถูกต้องของการติดตั้งโปรแกรม NS-2.35 โดยพิมพ์คำสั่งต่อไปนี้

   ```
   $ cd ns-2.35
   $ ./validate
   ```

8. ถ้า validate ผ่านแสดงว่าติดตั้งโปรแกรม NS-2.35 สมบูรณ์ และสามารถเปิดโปรแกรมแสดงได้ดังแสดงในรูปที่ ก.1

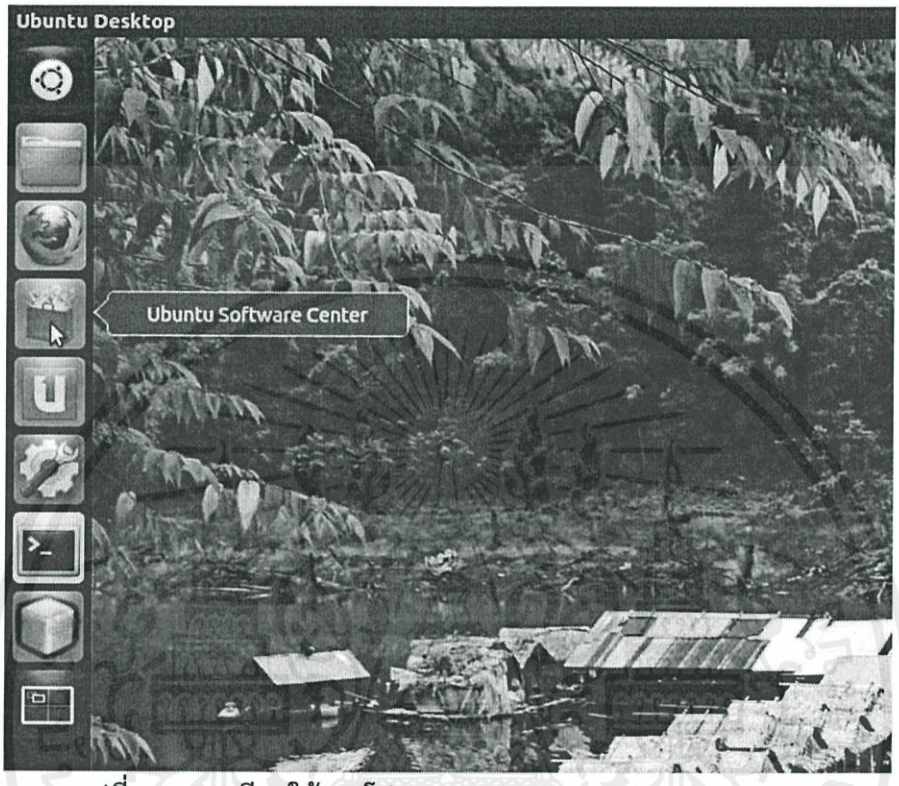


รูปที่ ก.1 NS-2.35 ที่ได้ทำการติดตั้งสมบูรณ์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 การติดตั้งและการเรียกใช้งาน Netbeans IDE 7.0.1

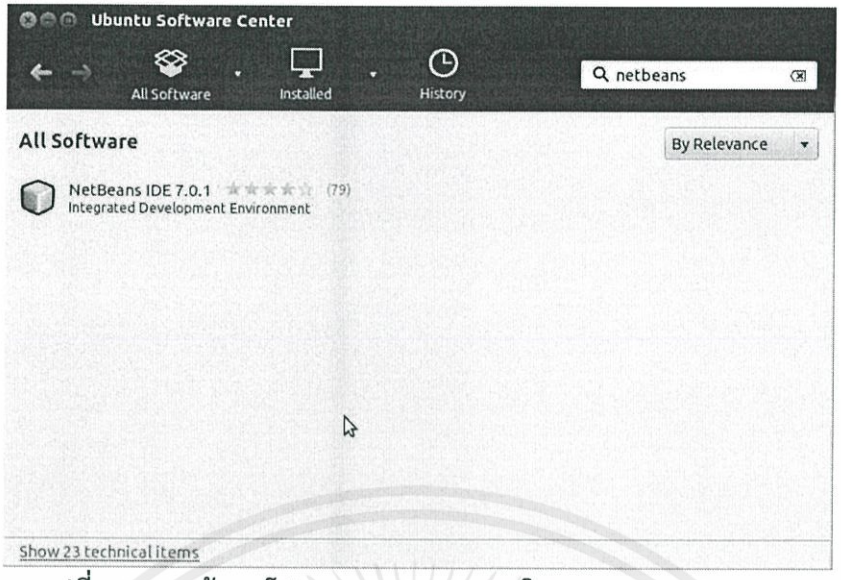
1. เปิดโปรแกรม Ubuntu Software Center ใน Taskbar Menu ดังแสดงในรูปที่ ก.2



รูปที่ ก.2 การเรียกใช้งานโปรแกรม Ubuntu Software Center

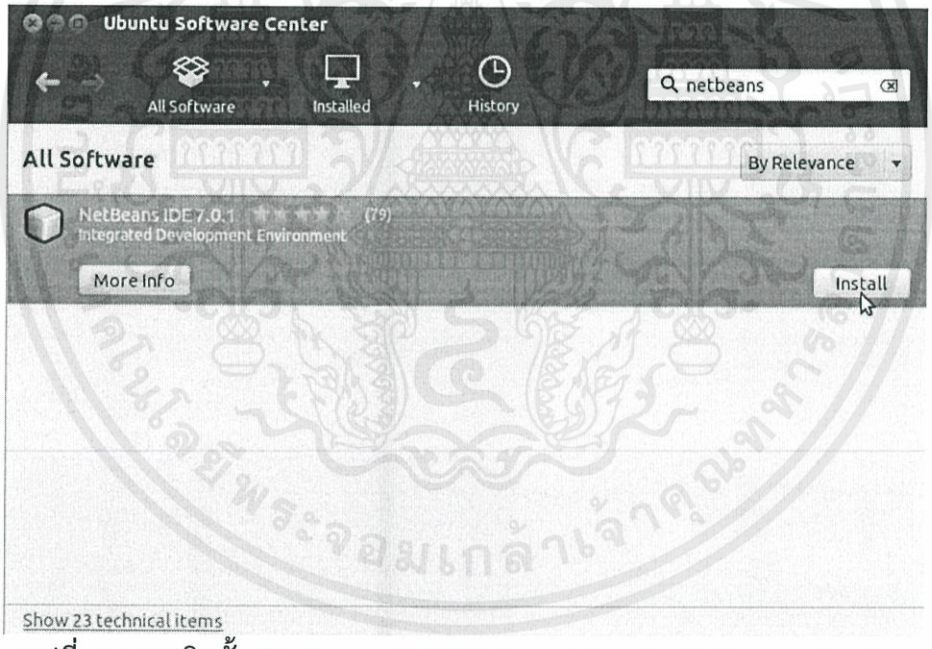
2. พิมพ์คำว่า netbeans ในช่องค้นหาที่โปรแกรม Ubuntu Software Center ดังแสดงในรูปที่ ก.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 การค้นหาโปรแกรม netbeans ใน Ubuntu Software

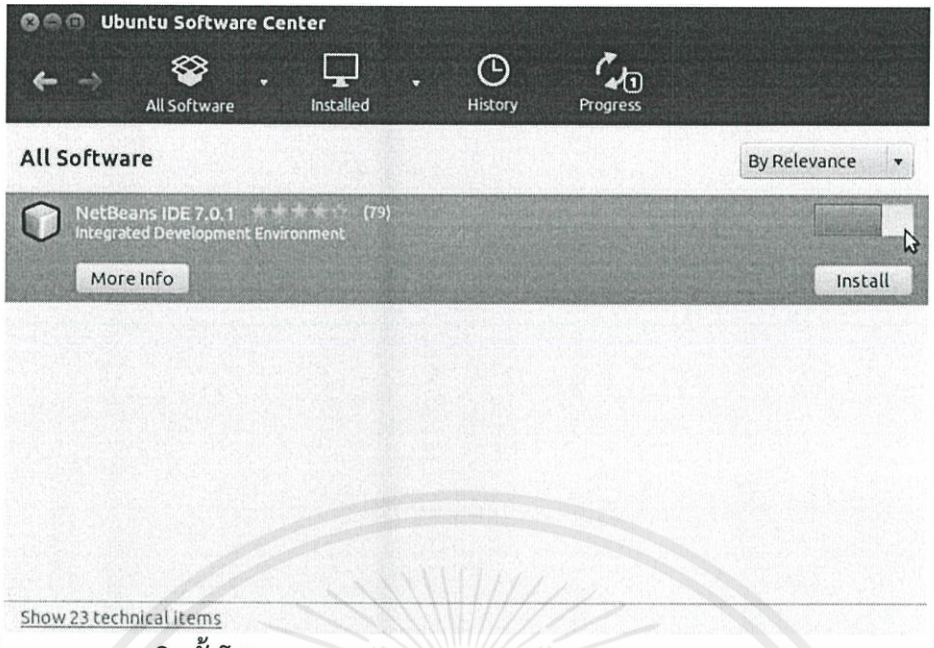
3. เลือกที่รายการ NetBeans IDE 7.0.1 แล้วกดปุ่ม Install ทางด้านขวามือสุด ดังแสดงในรูปที่ ก.4



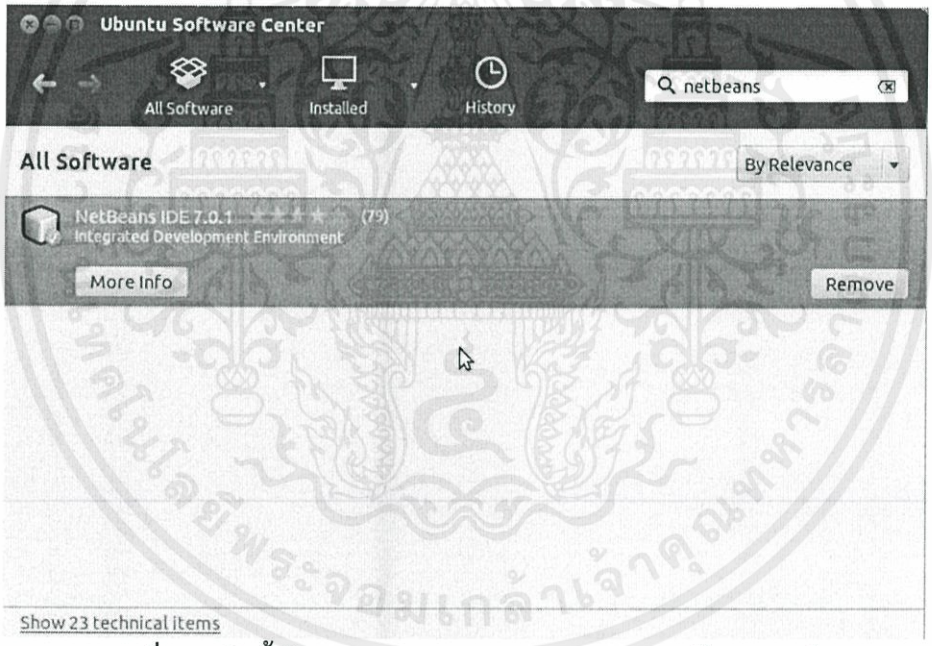
รูปที่ ก.4 การติดตั้ง NetBeans IDE 7.0.1 บน Ubuntu Software Center

4. หลังจากกดปุ่ม Install แล้ว ให้รอแถบสีส้มข้างบนปุ่ม Install เต็ม ดังแสดงในรูปที่ ก.5 การติดตั้งโปรแกรม NetBeans IDE 7.0.1 จึงจะเสร็จสมบูรณ์ แสดงดังแสดงในรูปที่ ก.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



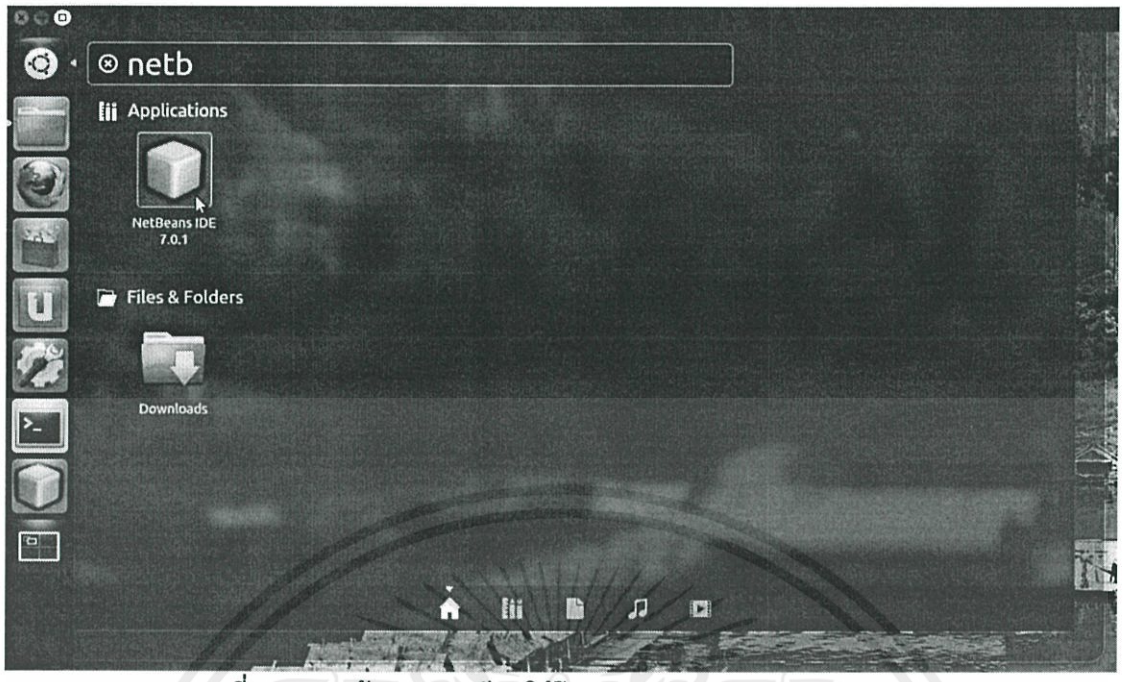
รูปที่ ก.5 แถบการติดตั้งโปรแกรม NetBeans IDE 7.0.1 บน Ubuntu Software Center



รูปที่ ก.6 ติดตั้งโปรแกรม NetBeans IDE 7.0.1 เสร็จสมบูรณ์

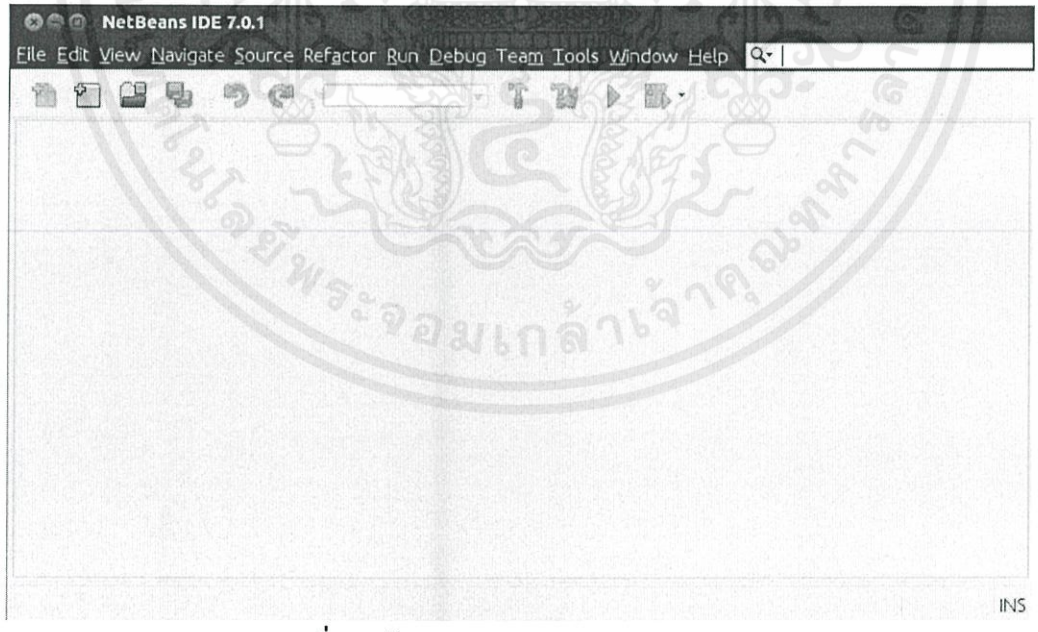
5. การเรียกใช้โปรแกรม NetBeans IDE 7.0.1 บน Ubuntu 12.04 จะเริ่มจากการพิมพ์ค้นหาคำว่า netbeans แล้วจะพบ icon ดังแสดงในรูปที่ ก.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.7 การค้นหาและเรียกใช้โปรแกรม NetBeans IDE 7.0.1

6. โปรแกรม NetBeans IDE 7.0.1 ที่สมบูรณ์แล้วจะแสดงดังแสดงในรูปที่ ก.8



รูปที่ ก.8 โปรแกรม NetBeans IDE 7.0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รายละเอียดแผนผังคลาส (Class Diagram)

ข.1 รายละเอียดของคลาส UI

รายละเอียดต่างๆของคลาส UI เป็นดังโปรแกรมที่ ข.1

โปรแกรมที่ ข.1 Attributes และ Methods ของคลาส UI

```
Class UI {
    Simulation sim;
    Graph g;

    //METHODS
    public void main();
    public void newSimulation();
    public void newGraph();
}
```

ข.2 รายละเอียดของคลาส Simulation

รายละเอียดต่างๆของคลาส Simulation เป็นดังโปรแกรมที่ ข.2

โปรแกรมที่ ข.2 Attributes และ Methods ของคลาส Simulation

```
Class Simulation {
    Parameters param;
    Nodes node;

    //METHODS
    public void main();
    public void newScenario();
    public void newWizard();
    public void genGraph();
    public void gridScale();
    public void setDimension();
}
```

ข.3 รายละเอียดของคลาส Graph

รายละเอียดต่างๆของคลาส Graph เป็นดังโปรแกรมที่ ข.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ ข.3 Attributes และ Methods ของคลาส Graph

```
      static String ifq ;
      //---------------------------------
      static int cbr_flag  ;
      static int tcp_flag  ;
      //---------------------------------
      static int traffic_flag  ;
      static int c_first  ;
      static int c_last  ;
      static int c_step  ;
      static int[] traffic_c_list;
      static int[] traffic_s_list = new int[6];
      //--------------
      static int mobility_flag  ;
      static int s_first  ;
      static int s_last  ;
      static int s_step  ;
      static int[] mobility_c_list = new int[6];
      static int[] mobility_s_list;
      //---------------------------------
      static int node  ;
      static double sim_time  ;
      static int sim_round  ;
      static int topoX;
      static int topoY;
      //--------- Couple Node ------------
      static int node1  ;
      static int node2  ;
      //---------------------------------
      static int pdf_flag;
      static int delay_flag;
      static int loss_flag;
      static int throughput_flag;

      //node
      static ArrayList<NodeDetail> allNode = new
  ArrayList<NodeDetail>();

      static String transport_protocol = "" ;
      static String routing_protocol = "" ;

      static DefaultCategoryDataset pdf_dataset = new
DefaultCategoryDataset();
      static DefaultCategoryDataset delay_dataset = new
DefaultCategoryDataset();
      static DefaultCategoryDataset loss_dataset = new
DefaultCategoryDataset();
      static DefaultCategoryDataset throughput_dataset = new
DefaultCategoryDataset();

      static ChartPanel CP_pdf;
      static ChartPanel CP_delay;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ ข.3 Attributes และ Methods ของคลาส Graph (ต่อ)

```
   static ChartPanel CP_loss;
      static ChartPanel CP_throughput;

   //METHODS
   public static void main();
   public static void generate_cbr_awk();
   public static void generate_tcp_awk();
   public static void generate_filter_awk();
   public static void generate_filter_pdf_awk();
   public static void generate_filter_delay_awk();
   public static void generate_filter_loss_awk();
   public static void generate_filter_throughput_awk();

   public static void select_based_type();
   public static void generate_traffic(String
transport_protocol, int node, int connection, String
traffic_out_file);
   public static void generate_mobility(int node, int speed,
double sim_time, String mobility_out_file)
   public static void generate_tcl(int node, double sim_time,
String val_rp, String tracefd, String traffic_file,  String
movement_file, String tcl_out_file);

   public static void simulation(String tcl_out_file);
   public static void filter_trace(String awk_file, String
tracefd, String result_file);
   public static void add_graph_value();

   public static void generate_graph();
   public static void create_pdf_graph(String chart_title,
String domain_axis_label);
   public static void create_delay_graph(String chart_title,
String domain_axis_label);
   public static void create_loss_graph(String chart_title,
String domain_axis_label);
   public static void create_throughput_graph(String
chart_title, String domain_axis_label);
   public static void remove_garbage();
   public static void DefaultParameters();
   public static void getParameters(Parameters p,
ArrayList<NodeDetail> n);
}
```

ข.4 รายละเอียดของคลาส Parameters

รายละเอียดต่างๆของคลาส Parameters เป็นดังโปรแกรมที่ ข.4

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ ข.4 Attributes และ Methods ของคลาส Parameters

```
Class Parameters {
        //---------- Simulation ----------//
        public double param_SimTime;
        public String param_TraceFile;
        public String param_NamFile;
        public int param_SimRound;
        //---------- Area and Nodes ----------//
        public int param_AreaX;
        public int param_AreaY;
        //public String param_NumNodes;
        public boolean randFlag;
        //---------- Wireless Properties ----------//
        public String param_ChannelType;
        public String param_PropogationModel;
        public String param_NWIntType;
        public String param_MacProtocol;
        public String param_IntQueue;
        public String param_LinkLayer;
        public String param_AntennaModel;
        public String param_MaxPacket;
        public String param_RoutingProtocol;
        public String param_RouterTrace;
        public String param_AgentTrace;
        public String param_MacTrace;
        public String param_MovementTrace;
        //---------- Nodes Definition ----------//
        public int numNode;
        //---------- Agent Definition ----------//
        public String agentP;
        public int pacSize;
        //---------- App Definition ----------//
        public String appP;
        //---------- Graph ----------//
        public int random_flag;
        public int manual_flag;
        public int aodv_flag;
        public int dsr_flag;
        public int cbr_flag;
        public int tcp_flag;
        public int traffic_flag;
        public int c_first;
        public int c_last;
        public int c_step;
        public int[] traffic_c_list;
        public int[] traffic_s_list;
        public int mobility_flag;
        public int s_first;
        public int s_last;
        public int s_step;
        public int[] mobility_c_list;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ ข.4 Attributes และ Methods ของคลาส Parameters (ต่อ)

```
        public int[] mobility_s_list;
        public int pdf_flag;
        public int delay_flag;
        public int loss_flag;
        public int throughput_flag;

        public Node node1 = new Node();
        public Node node2 = new Node ();
}
```

ข.5 รายละเอียดของคลาส Nodes

รายละเอียดต่างๆของคลาส Nodes เป็นดังโปรแกรมที่ ข.5

โปรแกรมที่ ข.5 Attributes ของคลาส Nodes

```
Class Nodes {
   int node_id;
   String name;
   int posX;
   int posY;
}
```

ข.6 รายละเอียดของคลาส TCLScript

รายละเอียดต่างๆของคลาส TCLScript เป็นดังโปรแกรมที่ ข.6

โปรแกรมที่ ข.6 Attributes ของคลาส TCLScript

```
Class TCLScript {
   int node;
   int sim_time;
   String val_rp;
   String tracefd;
   String traffic_file;
   String movement_file;
   String tcl_out_file;
   String ifq;
}
```

ข.7 รายละเอียดของคลาส AWKResult

รายละเอียดต่างๆของคลาส AWKResult เป็นดังโปรแกรมที่ ข.7

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ ข.7 Attributes ของคลาส AWKResult

```
Class AWKResult {
    float pdf;
    float delay;
    float loss;
    float throughput; }
```

ข.8 รายละเอียดของคลาส TrafficFile

รายละเอียดต่างๆของคลาส TrafficFile เป็นดังโปรแกรมที่ ข.8

โปรแกรมที่ ข.8 Attributes ของคลาส TrafficFile

```
Class TrafficFile {
    String transport_protocol;
    int node;
    int connection;
    String traffic_out_file;
}
```

ข.9 รายละเอียดของคลาส TraceFile

รายละเอียดต่างๆของคลาส TraceFile เป็นดังโปรแกรมที่ ข.9

โปรแกรมที่ ข.9 Attributes ของคลาส TraceFile

```
Class TraceFile {
    int traffic_c_list;
    int traffic_s_list;
    String routing_protocol;
    String transport_protocol;
}
```

ข.10 รายละเอียดของคลาส AWKScript

รายละเอียดต่างๆของคลาส AWKScript เป็นดังโปรแกรมที่ ข.10

โปรแกรมที่ ข.10 Attributes ของคลาส AWKScript

```
Class AWKScript {
    int random_flag;
    int manual_flag;
    int node1;
    int node2; }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้