

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การเพิ่มประสิทธิภาพของการส่งข้อมูล TCP
บนเครือข่ายไร้สายโดยการรีเซตไทม์เอาท์

PERFORMANCE ENHANCEMENT OF TCP PACKET
TRANSMISSION OVER WIRELESS NETWORK
BY TIMEOUT RESETTING



นริศร แสงคะนอง

NARISSORN SANGKANONG

เลขหม.....
เลขทะเบียน..... 47660
วัน, เดือน, ปี..... 21 ส.ค. 2546

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ISBN 974-324-501-4

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**PERFORMANCE ENHANCEMENT OF TCP PACKET
TRANSMISSION OVER WIRELESS NETWORK
BY TIMEOUT RESETTING**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2003

ISBN 974-324-501-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2003

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การเพิ่มประสิทธิภาพของการส่งข้อมูล TCP บนเครือข่ายไร้สาย โดยการรีเซตไทม์เอาต์
นักศึกษา	นาย นริศร แสงคะนอง
รหัสประจำตัว	41061138
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2546
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. รัตติกร วรากุลศิริพันธุ์

บทคัดย่อ

ปัจจุบันการสื่อสารข้อมูลระหว่างเครือข่ายมีสายและไร้สายได้มีการใช้งานร่วมกันมากขึ้น ซึ่ง Transmission Control Protocol (TCP) เป็นโปรโตคอลที่มีการใช้อย่างแพร่หลายในการควบคุมความน่าเชื่อถือของการส่งข้อมูล โดย TCP ถูกออกแบบสำหรับเครือข่ายมีสาย โดยสามารถที่จะนำมาใช้งานกับเครือข่ายไร้สายได้ แต่จะทำให้ประสิทธิภาพของ TCP ลดลง เนื่องจากการเกิดการส่งแพ็คเก็ตซ้ำซ้อน (Duplicated Packet) ซึ่งมีสาเหตุมาจากเวลาในการส่งข้อมูลจากสถานีฐาน (Base Station) ไปที่โฮสต์เคลื่อนที่ (Mobile Host) สูงกว่า TCP ไทม์เอาต์ (TCP's Timeout)

ในงานวิจัยได้เสนอวิธีการรีเซตไทม์เอาต์ (Timeout Resetting) เพื่อลดการเกิดการส่งแพ็คเก็ตซ้ำซ้อน โดยจะทำการตรวจสอบสถานะอัตราการผิดพลาดของเครือข่ายไร้สาย ถ้าอัตราการผิดพลาดมีค่าสูงเกินกว่าค่าที่กำหนด ก็จะทำการรีเซต TCP ไทม์เอาต์ จะเห็นว่าอัตราการผิดพลาดจะเป็นตัวกำหนดสถานะของเครือข่ายไร้สาย โดยในงานวิจัยนี้ได้ใช้แบบจำลองการเข้าคิว (Queueing Model) ในการวิเคราะห์และจำลองการทำงานของสถานีฐาน เพื่อหาค่าที่เหมาะสมของอัตราการผิดพลาด

งานวิจัยนี้ได้ใช้โปรแกรม OPNET ในการจำลองเครือข่าย ซึ่งผลปรากฏว่าได้ลดการส่งแพ็คเก็ตซ้ำซ้อนลง ทำให้ TCP มีประสิทธิภาพในการทำงานที่ดีขึ้น

Thesis Title	Performance Enhancement of TCP Packet Transmission over Wireless Network by Timeout Resetting
Student	Mr. Narissorn Sangkanong
Student ID.	41061138
Degree	Master of Engineering
Program	Electrical Engineering
Year	2003
Thesis Advisor	Assoc. Prof. Dr. Ruttikorn Varakulsiripunth

ABSTRACT

Currently, wired and wireless internetworking is increasingly used for data communication. The transmission control protocol (TCP) is one of the traditional protocols utilized to control the reliability of data transmission. This protocol is mainly designed for wired networks. Nevertheless, it can be applied to wireless networks, but its performance will be decreased because of duplicated packets. The duplicated packets are obstacle and are occurred when transmission time from base station to mobile host is greater than TCP's timeout.

In order to improve the TCP's performance over wireless networks, we propose TCP timeout resetting approach to reduce the duplicated packets. In this way, the status of error rate of wireless networks is firstly checked. If the error rate is greater than the presetting value, then the TCP timeout parameter is reset. The presetting value is significant for deciding the status of wireless network. In this paper, the queueing model is analyzed and implemented to the base station to obtain the suitable presetting value.

Finally, the simulation results by using OPNET simulation program are evident that duplicated packets are greatly reduced. This means that the TCP's performance is improved.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้เป็นอย่างดี เนื่องด้วยการสนับสนุนของบุคคลหลายฝ่าย ผู้ทำวิจัย

ขอขอบพระคุณ รศ.ดร.รัตติกร วรากุลศิริพันธุ์ ซึ่งเป็นอาจารย์ที่ปรึกษา ที่กรุณาช่วยให้แนวความคิด ให้คำปรึกษาในการทำวิจัย ตลอดจนการวิเคราะห์เพื่อแก้ปัญหาต่างๆ อันเป็นประโยชน์ต่อ งานวิจัย และได้กรุณาให้การสนับสนุน โอกาสในการศึกษาวิจัยในระดับต่อไป

ขอขอบพระคุณครูบาอาจารย์ทุกท่านที่ได้กรุณาสอนวิชาพื้นฐานที่จำเป็นต้องใช้ในการทำ วิจัย

ขอขอบพระคุณ ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โครงการสำนักวิจัยการสื่อสารและเทคโนโลยีสารสนเทศ (ReCCIT) และองค์การความร่วมมือระหว่างประเทศของญี่ปุ่น (JICA) ที่ได้ให้การสนับสนุนเครื่องมือและอุปกรณ์ใน การทำวิจัยนี้ในห้องปฏิบัติการโครงข่ายการสื่อสาร (Communication Network Laboratory)

ขอขอบพระคุณ ภาควิชาครุศาสตร์ไฟฟ้า คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยี พระจอมเกล้าพระนครเหนือ ที่ได้ให้การสนับสนุนทุนการศึกษา

ขอขอบคุณ อ.พารา ทิมมะณีประเสริฐ และเพื่อนๆในห้องปฏิบัติการโครงข่ายการสื่อสาร ที่ ช่วยแนะนำแนวทางในการทำวิจัย การเขียนและการนำเสนอบทความ การหาข้อมูลเกี่ยวกับงานวิจัย ให้คำปรึกษาที่ดี และความช่วยเหลือที่ดีตลอดมา

ขอขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ที่ช่วยให้คำปรึกษาและแก้ปัญหาต่างๆในงานวิจัย รวมทั้งทุกๆ คนที่มีได้เอ่ยนามในที่นี้

และสุดท้ายนี้ขอกราบขอบพระคุณคุณแม่ ผู้ซึ่งคอยอบรมเลี้ยงดู และทุกคนใน ครอบครัวที่ให้โอกาสในการศึกษา และสนับสนุนในทุกๆด้าน รวมถึงช่วยเป็นกำลังใจให้เสมอมา และตลอดไป อีกทั้งช่วยผลักดันให้วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดีเยี่ยม

นริศร แสงคะนอง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 ทฤษฎีหรือแนวความคิด.....	2
1.4 ขอบเขตของวิทยานิพนธ์.....	2
1.5 เนื้อหาภายในวิทยานิพนธ์.....	2
บทที่ 2 การใช้งาน TCP บนเครือข่ายไร้สาย และทฤษฎีการเข้าคิวแบบ M/M/1.....	4
2.1 การใช้งาน TCP บนเครือข่ายไร้สาย.....	4
2.1.1 Transmission Control Protocol (TCP).....	4
2.1.2 การใช้งาน TCP บนเครือข่ายไร้สาย.....	9
2.2 ทฤษฎีการเข้าคิวแบบ M/M/1.....	11
2.2.1 ค่าพารามิเตอร์และค่าการเริ่มต้นในการคำนวณสำหรับการเข้าคิวแบบ M/M/1.....	13
2.2.2 เวลาในระบบหรือเวลาการเข้าคิว.....	15
2.2.3 ความไม่เป็นเชิงเส้นของเวลาในระบบ.....	15
2.2.4 เวลาการรอ.....	16
2.2.5 การสร้างแบบจำลอง.....	17
บทที่ 3 แบบจำลองทางคณิตศาสตร์และวิธีการรีเซตไทม์เอาต์ (Timeout Resetting).....	19
3.1 แบบจำลองทางคณิตศาสตร์.....	19
3.2 สภาวะดีและไม่ดีในเครือข่ายไร้สาย.....	22
3.3 วิธีการรีเซตไทม์เอาต์.....	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.4 ผลของการวิเคราะห์ทางคณิตศาสตร์.....	31
บทที่ 4 การจำลองการทำงาน.....	34
4.1 ฟังก์ชันการจำลองของโปรแกรม.....	34
4.2 โปรแกรมจำลอง.....	37
4.2.1 OPNET.....	37
4.2.2 โปรแกรมจำลอง.....	40
บทที่ 5 ผลการจำลอง.....	46
5.1 ผลการจำลอง.....	46
5.2 การเปรียบเทียบผลลัพธ์ที่ได้จากการจำลองและจากการคำนวณทางคณิตศาสตร์.....	60
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....	67
เอกสารอ้างอิง.....	69
ภาคผนวก.....	70
ภาคผนวก ก. ผลงานวิจัยและบทความทางวิชาการที่ได้รับการตีพิมพ์.....	71
ประวัติผู้เขียน.....	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ความไม่เป็นเชิงเส้นของเวลาการเข้าคิว จาก 10 คนเป็น 12 คนต่อชั่วโมง.....	16
2.2 ความไม่เป็นเชิงเส้นของเวลาการเข้าคิว จาก 20 คนเป็น 22 คนต่อชั่วโมง.....	16
5.1 ค่าของการจำลอง ภาพที่ 5.1-5.3 และ ภาพที่ 5.5-5.7.....	46
5.2 จำนวนแพ็คเกตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.3$	47
5.3 จำนวนแพ็คเกตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.5$ และ $\rho = 0.3$	48
5.4 จำนวนแพ็คเกตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.7$ และ $\rho = 0.3$	49
5.5 จำนวนแพ็คเกตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.9$	51
5.6 จำนวนแพ็คเกตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.5$ และ $\rho = 0.9$	52
5.7 จำนวนแพ็คเกตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.7$ และ $\rho = 0.9$	53
5.8 S/G เมื่อ $P_s = 0.7$ และ $\rho = 0.3$ ที่ $1t-7t$	55
5.9 S/G เมื่อ $P_s = 0.5$ และ $\rho = 0.5$ ที่ $1t-7t$	57
5.10 S/G เมื่อ $P_s = 0.3$ และ $\rho = 0.9$ ที่ $1t-7t$	58
5.11 การเปรียบเทียบค่าคุณภาพ โดยการหาค่า MSE ของการจำลองและการคำนวณ.....	63
5.12 การเปรียบเทียบค่า S และ G โดยการหาค่า MSE ของการจำลองและการคำนวณ.....	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
2.1 การสื่อสารแบบปลายถึงปลายของ TCP	5
2.2 TCP ไทม์เอาต์และการส่งซ้ำ	6
2.3 ฟิลด์เบื้องต้นของ TCP เฮดเดอร์	7
2.4 การส่งข้อมูลพื้นฐานและ ACKs	7
2.5 การสูญหายและการส่งซ้ำข้อมูล	8
2.6 การเชื่อมต่อระหว่างเครือข่ายไร้สายและมีสาย	9
2.7 การทำงานของ TCP ระหว่างเครือข่ายมีสายและไร้สาย	10
2.8 การส่งข้อมูลที่ช้าซ้อนของ TCP บนเครือข่ายไร้สาย	12
2.9 รูปแบบการเข้าคิวแบบ M/M/1	13
2.10 ค่าพารามิเตอร์ของการเข้าคิวแบบ M/M/1	13
2.11 เวลาในระบบของการเข้าคิวแบบ M/M/1	15
2.12 วงรอบของการสร้างแบบจำลอง	17
3.1 แบบจำลองการเข้าคิวแบบ M/M/1 ของสถานีฐาน	20
3.2 ความสัมพันธ์ระหว่าง E และ T ที่ค่า ρ ต่างๆ	23
3.3 การเปลี่ยนสถานะระหว่างสถานะดีและไม่ดี	25
3.4 การทำงานของวิธีการรีเซตไทม์เอาต์	26
3.5 การส่งซ้ำข้อมูลเมื่อเครือข่ายไร้สายอยู่ในสถานะไม่ดี	27
3.6 แบบจำลองความน่าจะเป็นของเครือข่ายไร้สายในสถานะไม่ดี	28
3.7 S และ G ของวิธีการ TCP ทั่วไปและรีเซตไทม์เอาต์	31
3.8 คุณภาพของวิธีการ TCP ทั่วไปและรีเซตไทม์เอาต์	32
4.1 กระบวนการของโปรแกรมการจำลองการส่งข้อมูลระหว่างเครือข่ายมีสายและไร้สาย	36
4.2 โปรเจกต์เอคิเตอร์	37
4.3 โหนดเอคิเตอร์	38
4.4 โปรเซส โมเดลเอคิเตอร์และซอร์สโค้ด	39
4.5 แบบจำลองของเครือข่ายใน OPNET	40
4.6 โปรเซส โมเดลของ base	41
4.7 กระบวนการทำงานของ base	42
4.8 state variable	44

เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
4.9 โปรแกรมภายใต้ scv-start	44
4.10 โปรแกรมภายใต้ dest	45
5.1 จำนวนแพ็คเก็ตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.3$	47
5.2 จำนวนแพ็คเก็ตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.5$ และ $\rho = 0.3$	48
5.3 จำนวนแพ็คเก็ตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.7$ และ $\rho = 0.3$	49
5.4 จำนวนแพ็คเก็ตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3, 0.5$ และ 0.7 ที่ $\rho = 0.3$	50
5.5 จำนวนแพ็คเก็ตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.9$	51
5.6 จำนวนแพ็คเก็ตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.5$ และ $\rho = 0.9$	52
5.7 จำนวนแพ็คเก็ตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.7$ และ $\rho = 0.9$	53
5.8 จำนวนแพ็คเก็ตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3, 0.5$ และ 0.7 ที่ $\rho = 0.9$	54
5.9 S/G เมื่อ $P_s = 0.7$ และ $\rho = 0.3$ ที่ $1t-7t$	55
5.10 S/G เทียบกับวิธีการ TCP ทั่วไปและการรีเซตไทม์เอาต์ เมื่อ $P_s = 0.7$ และ $\rho = 0.3$	56
5.11 S/G เมื่อ $P_s = 0.5$ และ $\rho = 0.5$ ที่ $1t-7t$	56
5.12 S/G เทียบกับวิธีการ TCP ทั่วไปและการรีเซตไทม์เอาต์ เมื่อ $P_s = 0.5$ และ $\rho = 0.5$	57
5.13 S/G เมื่อ $P_s = 0.3$ และ $\rho = 0.9$ ที่ $1t-7t$	58
5.14 S/G เทียบกับวิธีการ TCP ทั่วไปและการรีเซตไทม์เอาต์ เมื่อ $P_s = 0.3$ และ $\rho = 0.9$	59
5.15 จำนวนแพ็คเก็ตที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.3$ ทำการจำลอง 10 นาที	60
5.16 S/G เมื่อ $P_s = 0.3$	61
5.17 S/G เมื่อ $P_s = 0.5$	62
5.18 S/G เมื่อ $P_s = 0.7$	62
5.19 S และ G เมื่อ $P_s = 0.3$	64
5.20 S และ G เมื่อ $P_s = 0.5$	64
5.21 S และ G เมื่อ $P_s = 0.7$	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเทคโนโลยีด้านเครือข่ายคอมพิวเตอร์ได้พัฒนาไปอย่างรวดเร็ว ในอดีตการเชื่อมต่อระหว่างเครือข่ายหรือเครื่องคอมพิวเตอร์จะใช้การเชื่อมต่อโดยใช้สายสัญญาณเป็นหลัก แต่ในปัจจุบันได้มีการใช้การเชื่อมต่อแบบไร้สายมากขึ้น เช่นระบบแลนไร้สาย เนื่องจากติดตั้งได้สะดวกและไม่ต้องเดินสายสัญญาณ แต่ทั้งนี้ทั้งสองระบบก็จำเป็นต้องใช้งานร่วมกัน เนื่องจากระบบส่วนใหญ่เป็นเครือข่ายแบบมีสาย ดังนั้นเมื่อมีการนำเครือข่ายไร้สายมาใช้ร่วมกับเครือข่ายมีสาย ทำให้ต้องมีการปรับปรุงฟังก์ชันการใช้งานเพื่อให้ทั้ง 2 ระบบทำงานสอดคล้องกัน แต่ฟังก์ชันบางอย่างก็สามารถนำมาใช้งานระหว่างทั้ง 2 ระบบได้ทันที

TCP (Transmission Control Protocol) เป็นโปรโตคอล (Protocol) ที่ใช้ควบคุมการส่งข้อมูลในเครือข่ายคอมพิวเตอร์ ทำหน้าที่ตรวจสอบว่าข้อมูลที่ส่งจากต้นทางถูกส่งไปถึงปลายทางอย่างถูกต้องสมบูรณ์หรือไม่โดยใช้การจับเวลาและการรอสัญญาณตอบกลับจากปลายทาง เมื่อนำ TCP มาใช้ในการสื่อสารข้อมูลระหว่างเครือข่ายไร้สายและมีสายจะทำให้ประสิทธิภาพในการทำงานของ TCP ลดลง เนื่องจาก TCP ถูกออกแบบเพื่อใช้กับเครือข่ายมีสายเป็นหลัก จึงทำให้ฟังก์ชันในการทำงานบางอย่างไม่เหมาะสมเมื่อนำมาใช้งานกับเครือข่ายไร้สาย เช่นฟังก์ชันที่มีชื่อว่าไทม์เอาต์ (Timeout) มีหน้าที่ในการจับเวลาของการส่งข้อมูล โดย TCP จะตัดสินใจว่าข้อมูลที่ส่งไปไม่ถึงปลายทางในเวลาที่กำหนดหรือเกิดการสูญหาย จะเกิดจากความแออัดบนเครือข่ายมีสายเป็นหลัก แต่ในเครือข่ายไร้สายจะเกิดจากความผิดพลาดของช่องทางการสื่อสารเป็นหลัก

ในการเชื่อมต่อระหว่างเครือข่ายไร้สายและมีสายจะมีสถานีฐาน (Base Station) ทำหน้าที่เชื่อมต่อระหว่างเครือข่ายไร้สายและมีสาย รวมทั้งตรวจสอบว่าข้อมูลที่ส่งจากสถานีฐานไปถึงปลายทางที่เครือข่ายไร้สายหรือไม่ โดยการตรวจสอบจะใช้การจับเวลาเช่นกัน เมื่อสถานีฐานไม่ได้รับสัญญาณตอบกลับภายในเวลาที่กำหนดก็จะทำการส่งข้อมูลซ้ำอีกครั้ง แต่ TCP ก็ส่งข้อมูลซ้ำเหมือนกันเมื่อไม่ได้รับสัญญาณตอบรับภายในเวลาที่กำหนด ซึ่งเป็นสาเหตุให้เกิดการส่งข้อมูลที่ซ้ำซ้อน (Duplicated Packet) จากต้นทาง ทำให้ประสิทธิภาพของ TCP ลดลง รวมทั้งมีการใช้ช่องสัญญาณมากขึ้นเนื่องจากการส่งข้อมูลที่ซ้ำซ้อน ดังนั้นหากสามารถลดการเกิดการส่งข้อมูลที่ซ้ำซ้อนได้ก็จะทำให้ลดการใช้ช่องสัญญาณลง จึงเป็นการเพิ่มประสิทธิภาพการใช้ช่องสัญญาณและทำให้ประสิทธิภาพของการทำงาน TCP ดีขึ้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาถึงปัญหาการส่งข้อมูลซ้ำซ้อนระหว่างเครือข่ายมีสายและไร้สาย เมื่อใช้ TCP เป็นโปรโตคอลบนเครือข่าย

1.2.2 ปรับปรุงการใช้งาน TCP บนเครือข่ายไร้สาย โดยลดการส่งข้อมูลที่ซ้ำซ้อนระหว่างผู้ส่งและผู้รับภายใต้ระบบการควบคุมการสื่อสารแบบ TCP

1.3 ทฤษฎีหรือแนวความคิด

งานวิจัยในวิทยานิพนธ์ จะใช้ทฤษฎีการเข้าคิวมาวิเคราะห์การทำงานของการทำงานของการส่งข้อมูลระหว่างเครือข่ายไร้สายและมีสาย เมื่อใช้ TCP เป็นโปรโตคอลควบคุมการส่งข้อมูล เพื่อที่จะสร้างแบบจำลองทางคณิตศาสตร์ที่ใช้ศึกษาพฤติกรรมของระบบ และวิเคราะห์ถึงปัญหาการส่งข้อมูลที่ซ้ำซ้อนระหว่างฝ่ายส่งที่อยู่กับที่และฝ่ายรับเคลื่อนที่ เมื่อสถานการณ์ใช้เวลาในการส่งข้อมูลเกินกว่าไทม์เอาต์พร้อมทั้งเสนอวิธีการลดการส่งข้อมูลซ้ำซ้อนโดยการตรวจสอบสถานะของเครือข่ายไร้สาย โดยถ้ารู้สถานะของเครือข่ายไร้สาย ก็จะทำให้ทราบถึงเวลาที่จำเป็นต้องใช้ในการส่งข้อมูลจากสถานการณ์ไปที่โฮสต์เคลื่อนที่

1.4 ขอบเขตของวิทยานิพนธ์

งานวิจัยนี้ทำการศึกษาปัญหาการส่งข้อมูลซ้ำซ้อนระหว่างเครือข่ายมีสายและไร้สาย เมื่อใช้ TCP เป็นโปรโตคอลควบคุมการส่งข้อมูล รวมทั้งใช้ทฤษฎีการเข้าคิวมาทำการวิเคราะห์ระบบเพื่อสร้างแบบจำลองทางคณิตศาสตร์ของเครือข่าย และทำการแก้ปัญหาการส่งข้อมูลที่ซ้ำซ้อนระหว่าง TCP ต้นทางและสถานการณ์ รวมทั้งสร้างโปรแกรมจำลองการทำงานโดยใช้ OPNET เพื่อศึกษาพฤติกรรมของระบบ

1.5 เนื้อหาภายในวิทยานิพนธ์

ในวิทยานิพนธ์นี้เป็นการเสนอการปรับปรุงประสิทธิภาพของการส่งข้อมูล TCP บนเครือข่ายไร้สายโดยการรีเซ็ตไทม์เอาต์ (Timeout Resetting) คือการกำหนดให้ผู้ส่งภายในเครือข่ายมีสายที่ใช้โปรโตคอล TCP เริ่มต้นจับเวลาการส่งข้อมูลใหม่ เมื่อได้รับรายงานถึงสภาพแวดล้อมที่ไม่ดีภายในเครือข่ายไร้สาย จากการเริ่มต้นจับเวลาใหม่ (หรือการเริ่มต้นไทม์เอาต์ใหม่) จะช่วยลดการเกิดการส่งข้อมูลที่ซ้ำซ้อนระหว่างเครือข่ายมีสายและไร้สาย ซึ่งทำการวิเคราะห์โดยการสร้างแบบจำลองทางคณิตศาสตร์ และโปรแกรมจำลอง เพื่อเปรียบเทียบการทำงานระหว่างวิธีการ TCP ทั่วไปกับวิธีการที่นำเสนอ โดยเนื้อหาภายในวิทยานิพนธ์มีดังนี้

บทที่ 2 อธิบายถึงหลักการการทำงานของ TCP ซึ่งเป็นโปรโตคอลที่ทำการพิจารณาทั้งบนเครือข่ายไร้สายและมีสาย รวมถึงปัญหาที่เกิดขึ้นเมื่อนำมาใช้งานบนเครือข่ายไร้สาย ในส่วนหลังจะกล่าวถึงทฤษฎีการเข้าคิวแบบ M/M/1 ซึ่งใช้ในการวิเคราะห์เครือข่าย

บทที่ 3 อธิบายถึงการสร้างแบบจำลองของสถานีฐานที่เชื่อมต่อกันระหว่างเครือข่ายมีสายและไร้สาย รวมถึงแบบจำลองทางคณิตศาสตร์ที่สร้างขึ้นมาโดยใช้ทฤษฎีการเข้าคิวแบบ M/M/1 และในที่สุดท้ายจะนำเสนอวิธีการรีเซตไทม์เอาต์และผลของการวิเคราะห์ทางคณิตศาสตร์ ซึ่งเปรียบเทียบระหว่างวิธีการ TCP ทั่วไป และวิธีที่นำเสนอ

บทที่ 4 อธิบายถึงการใช้งานโปรแกรม OPNET และโปรแกรมจำลองการทำงานระหว่างเครือข่ายมีสายและไร้สาย

บทที่ 5 แสดงผลการจำลองที่ได้จากโปรแกรมจำลอง และสรุปผล

บทที่ 6 กล่าวสรุปถึงงานโดยรวม และข้อเสนอแนะในการพัฒนางานวิจัยนี้

บทที่ 2

การใช้งาน TCP บนเครือข่ายไร้สาย และทฤษฎีการเข้าคิวแบบ M/M/1

ในการศึกษาเกี่ยวกับงานวิจัยเรื่องนี้ สิ่งแรกคือการทำความเข้าใจเกี่ยวกับทฤษฎีที่เป็นพื้นฐานในงานวิจัยนี้ นั่นคือเรื่องเกี่ยวกับการนำ TCP ไปใช้งานบนเครือข่ายไม่ว่าจะเป็นเครือข่ายมีสายหรือไร้สายก็ตาม จากนั้นจึงพิจารณาถึงปัญหาที่เกิดขึ้น เมื่อนำ TCP มาใช้งานบนเครือข่ายไร้สายและทำการศึกษาเกี่ยวกับทฤษฎีการเข้าคิวซึ่งถูกนำมาใช้ในการสร้างแบบจำลองของระบบที่ทำการพิจารณา จึงมีความจำเป็นที่ต้องศึกษาทฤษฎีพื้นฐานเหล่านี้

2.1 การใช้งาน TCP บนเครือข่ายไร้สาย

เนื่องจาก TCP ถูกออกแบบมาเพื่อใช้งานบนเครือข่ายมีสายเป็นหลัก ดังนั้นเมื่อมีการนำไปใช้งานบนเครือข่ายไร้สาย จึงทำให้ประสิทธิภาพในการใช้งานบางอย่างของ TCP ลดลง เนื่องจากมีความไม่เหมาะสมในการนำไปใช้งาน จึงต้องทำการศึกษาลักษณะทั่วไปของ TCP และปัญหาที่เกิดขึ้นเมื่อมีการนำไปใช้งานบนเครือข่ายไร้สาย

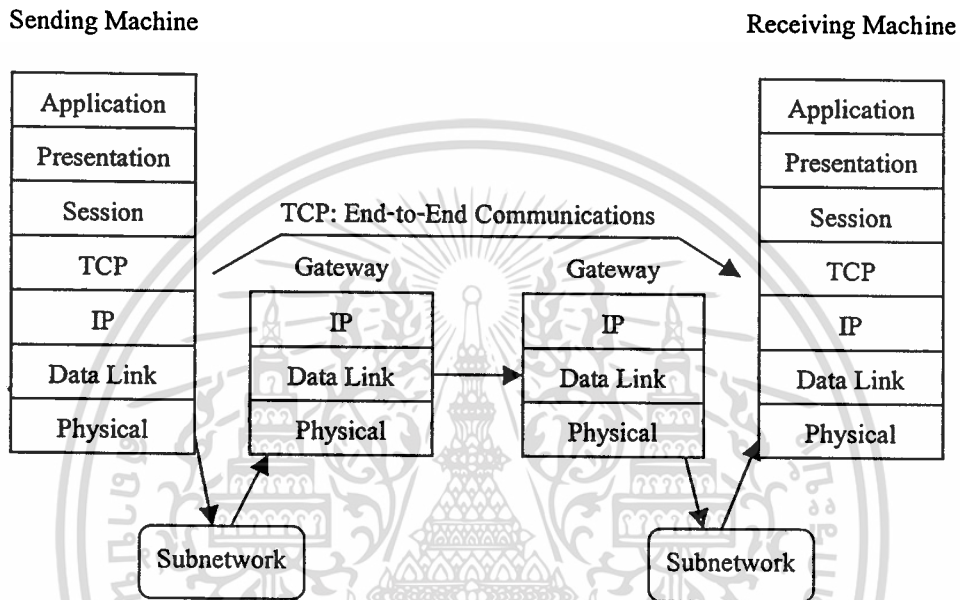
2.1.1 ทรานสมิชันคอนโทรลโพรโตคอล (Transmission Control Protocol :TCP)

TCP เป็น โพรโตคอลที่ได้เตรียมการบริการต่างๆ ไว้เพื่อรองรับการใช้งานในเลเยอร์ (Layer) ที่สูงกว่าหรือต่ำกว่า เช่น IP (Internet Protocol) เลเยอร์ โดย TCP เป็นโพรโตคอลแบบคอนเนคชันโอเรียนเตด (Connection-Oriented Protocol) ซึ่งทำให้การส่งข้อมูลบนเครือข่ายมีความน่าเชื่อถือและทำให้แน่ใจว่า สามารถที่จะส่งข้อมูลถึงปลายทางได้อย่างถูกต้องสมบูรณ์ โดย TCP จะทำหน้าที่คล้ายกับเป็นโพรโตคอลที่มีหน้าที่ตรวจสอบข้อความว่าถูกต้องหรือไม่ โดยถ้าข้อมูลที่ส่งไปที่ปลายทางมีการสูญหายหรือไม่ครบถ้วน TCP ก็จะทำการส่งซ้ำข้อมูลที่สูญหาย ซึ่งในเลเยอร์อื่น ๆ จะไม่มีการทำงานในลักษณะนี้ [1]

ภาพที่ 2.1 แสดงตำแหน่งของ TCP ในมาตรฐานการสื่อสารการติดต่อในระบบเปิด (Open Systems Interconnection : OSI) จะเห็นว่า TCP อยู่ในชั้นของ ทรานสปอร์ตเลเยอร์ และข้อมูลที่ส่งผ่าน TCP ถูกเรียกว่าดาตาแกรม (Datagram) โดย TCP จะติดตั้งอยู่ที่อุปกรณ์ที่มีการจัดการเกี่ยวกับดาตาแกรมเท่านั้น เพื่อทำให้มั่นใจว่าดาตาแกรมสามารถที่จะถูกส่งจากต้นทางไปถึงปลายทางได้อย่างถูกต้องสมบูรณ์ จากภาพที่ 2.1 จะเห็นว่า TCP ไม่ได้ติดตั้งอยู่ในระดับเกตเวย์ (Gateway) ซึ่งมีหน้าที่หาเส้นทางให้ดาตาแกรมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก TCP เป็นโปรโตคอลแบบคอนเนกชันโอเรียนเตด เพื่อให้มั่นใจว่าสามารถส่งข้อมูลจากต้นทางไปถึงปลายทางได้อย่างถูกต้อง โดย TCP จะได้รับข้อความจากปลายทาง ซึ่งจะแจ้งให้ต้นทางทราบว่าข้อมูลที่ส่งออกไปได้ไปถึงปลายทางอย่างถูกต้องแล้ว โดยการสื่อสารระหว่างต้นทางกับปลายทางของ TCP จะมีการสร้างวงจรเสมือน (Virtual Circuit) ขึ้นมาก่อน ซึ่งทำให้ง่ายในการสร้างสัญญาณตอบรับหรือตัวเลขลำดับของดาตาแกรม



ภาพที่ 2.1 การสื่อสารแบบปลายถึงปลายของ TCP

2.1.1.1 วิธีการสร้างความน่าเชื่อถือของ TCP

ในส่วนนี้ จะกล่าวถึงวิธีการที่ TCP ใช้สร้างความน่าเชื่อถือในการส่งข้อมูล ซึ่งทำให้ไม่เกิดการสูญหายและการส่งซ้ำข้อมูล

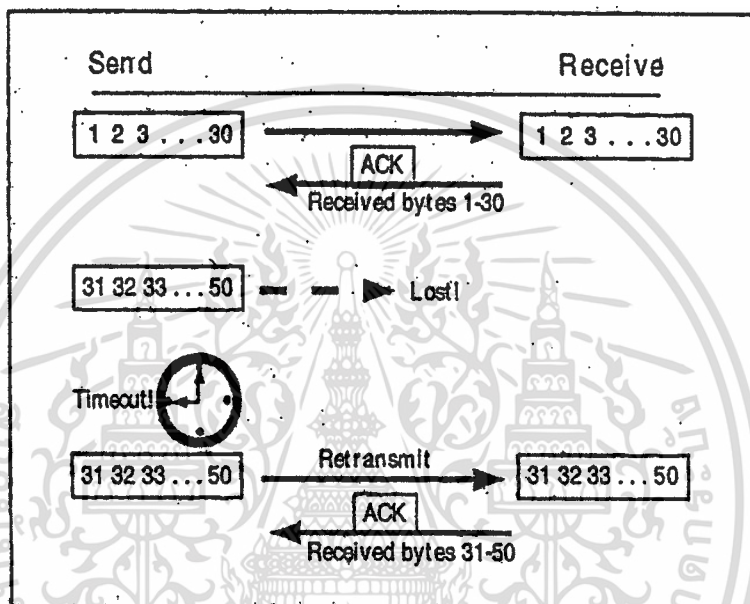
2.1.1.1.1 เลขลำดับและการตอบรับ

TCP จะใช้เลขลำดับและการตอบรับ (ACK) ในการสร้างความน่าเชื่อถือในการส่งข้อมูล โดยทุกไบต์ข้อมูลที่ถูกส่งในการเชื่อมต่อของ TCP จะถูกมองเป็นลำดับตัวเลขที่เรียงต่อกัน ซึ่งในส่วนเฮดเดอร์ (Header) ของ TCP จะบรรจุตัวเลขที่เป็นลำดับเรียงกันของข้อมูลไบต์แรกไว้ในเซกเมนต์นั้น [2]

ทางฝั่งผู้รับจะได้รับการคาดหวังว่า จะต้องส่ง ACK ของข้อมูลที่ได้รับไปแล้วไปให้ทางฝั่งผู้ส่ง โดยถ้าสัญญาณ ACK ไม่ได้ถูกส่งไปถึงต้นทางภายในระยะเวลาที่กำหนดหรือไทม์เอาท์ ข้อมูลเดิม

ก็จะถูกส่งมาใหม่อีกครั้งหนึ่ง ซึ่งวิธีการเช่นนี้เรียกว่า การตอบรับทางบวกกับการส่งซ้ำ (Positive Acknowledgement with Retransmission)

TCP ทางฝั่งผู้รับจะคอยดูว่าลำดับตัวเลขของข้อมูลที่มาถึงมีลำดับที่ถูกต้องหรือไม่ เพื่อให้แน่ใจว่าไม่มีความผิดพลาดในการรับส่งข้อมูล แต่ในบางครั้งสัญญาณ ACK ก็อาจจะเกิดการสูญหายหรือไปถึงช้า จึงอาจทำให้เกิดการส่งข้อมูลซ้ำซ้อนมาจากฝั่งตัวส่งได้เช่นกัน ซึ่งตัวเลขที่เรียงลำดับกันช่วยให้รู้ว่าข้อมูลเกิดการซ้ำซ้อนและสามารถจัดทิ้งไปได้ รายละเอียดที่กล่าวมาแสดงดังภาพที่ 2.2

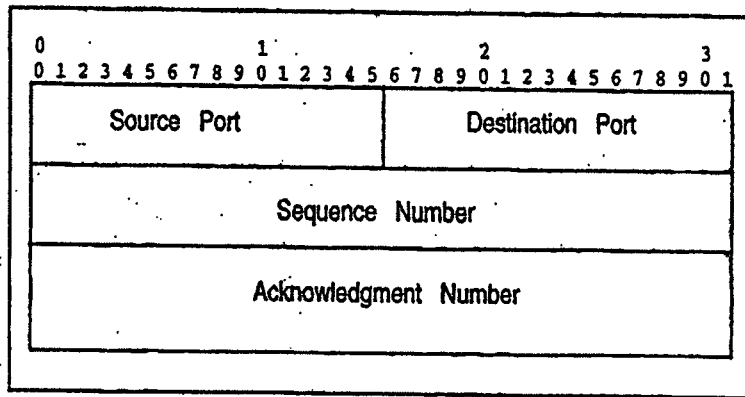


ภาพที่ 2.2 TCP ใหม่อาทและการส่งซ้ำ

2.1.1.1.2 TCP เฮดเดอร์ฟิลด์ (TCP Header Fields)

ภาพที่ 2.3 แสดง TCP เฮดเดอร์ฟิลด์หรือส่วนแรกที่ใช้เก็บรายละเอียดในการส่งข้อมูลของ TCP จะเห็นได้ว่าในฟิลด์แรกของ TCP เฮดเดอร์ได้จัดเตรียมพื้นที่สำหรับพอร์ตต้นทางและพอร์ตปลายทาง ตัวเลขที่เรียงลำดับของไบต์แรกของข้อมูลที่ส่งมาด้วยและตัวเลขของสัญญาณตอบรับ ซึ่งจะบรรจุเลขลำดับของไบต์ถัดไปที่คาดว่าจะส่งต่อไป ยกตัวอย่างเช่น ถ้า TCP ได้รับข้อมูลถึงไบต์ที่ 30 ก็อาจจะเขียน 31 ลงไปในฟิลด์การตอบรับ (Acknowledgement Field)

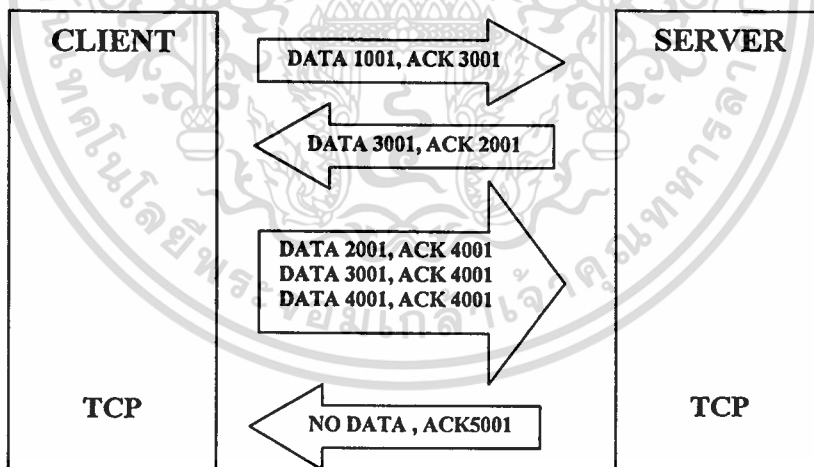
มีสิ่งหนึ่งที่น่าสนใจ สมมติว่า TCP ส่งข้อมูลไบต์ที่ 1 ถึงไบต์ที่ 50 แล้วไม่มีข้อมูลที่ต้องการจะส่งในกรณีเช่นนี้เมื่อ TCP ได้รับ ACK จากฝั่งตรงข้ามก็จะส่งเฉพาะ TCP เฮดเดอร์ไปโดยไม่มีข้อมูล โดยปกติในเฮดเดอร์จะบรรจุค่าของสัญญาณ ACK ซึ่งในที่นี้ก็คือ 51 เพื่อเป็นการบอกให้ฝั่งตรงข้ามรู้ว่าข้อมูลที่จะส่งในครั้งต่อไปก็คือ 51



ภาพที่ 2.3 ฟิวด์เบื่องต้นของ TCP เซคเตอร์

2.1.1.2 การส่งข้อมูล

ภาพที่ 2.4 แสดงการส่งข้อมูลระหว่าง TCP ซึ่งในตัวอย่างจะใช้ข้อมูลขนาด 1,000 ไบต์ และในแต่ละ TCP เซคเตอร์เซกเมนต์จะบรรจุด้วย ACK ฟิวด์ ซึ่งแสดงตัวเลขลำดับของไบต์ถัดไปที่คาดว่าจะส่งมาจากอีกฝ่ายหนึ่ง



ภาพที่ 2.4 การส่งข้อมูลพื้นฐานและ ACKs

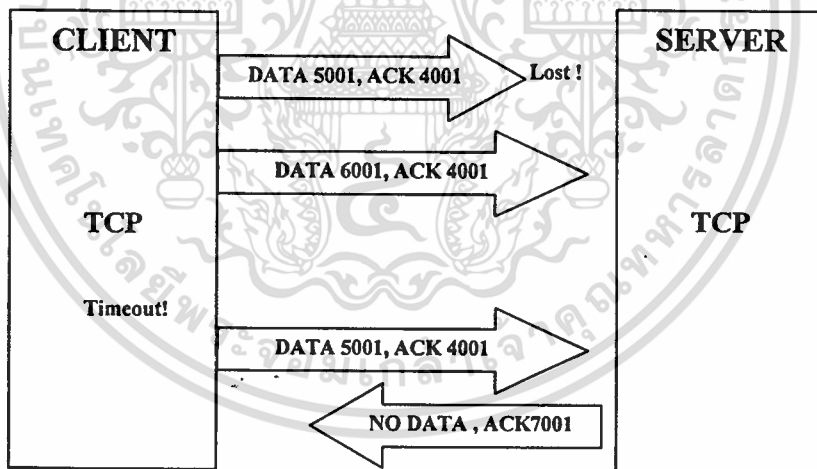
เซกเมนต์แรกจะส่งจากทางฝั่งไคลเอนต์โดยมีไบต์ที่ 1001 ถึง 2000 โดย ACK ฟิวด์จะเป็น 3001 ซึ่งคาดว่าจะจะเป็นไบต์ถัดไปที่ฝั่งเซิร์ฟเวอร์จะส่งมา

เซิร์ฟเวอร์จะตอบกลับมาด้วยเซกเมนต์ซึ่งมีข้อมูล 1,000 ไบต์ (เริ่มต้นที่ 3001) ซึ่ง ACK ฟิลด์ ใน TCP เซกเตอร์จะแสดงให้เห็นว่าไบต์ที่ 1001 ถึง 2001 ที่ฝั่งเซิร์ฟเวอร์ได้รับข้อมูลอย่างสมบูรณ์แล้ว จะเห็นว่าเลขลำดับของไบต์ถัดไปที่คาดว่าจะได้รับข้อมูลจากฝั่งไคลเอนต์คือไบต์ที่ 2001

ถัดมาไคลเอนต์จะส่งเซกเมนต์ที่เริ่มต้นที่ไบต์ 2001 3001 และ 4001 โดยส่งอย่างต่อเนื่อง ซึ่งฝั่งไคลเอนต์ไม่จำเป็นที่จะต้องรอสัญญาณ ACK ของแต่ละเซกเมนต์ก่อนในการส่งแต่ละเซกเมนต์ โดยสามารถส่งได้จนกว่าบัฟเฟอร์ฝั่งตรงข้ามจะเต็ม

ในทางฝั่งเซิร์ฟเวอร์จะทำการส่งสัญญาณ ACK มาเพียงสัญญาณเดียว ก็สามารถที่จะแสดงให้เห็นว่าข้อมูลทั้งหมดได้รับอย่างถูกต้องสมบูรณ์ ทั้งที่ส่งมาเพียง ACK เดียวนั้น ทั้งนี้เพื่อเป็นการประหยัดแบนด์วิดท์

ภาพที่ 2.5 แสดงการส่งข้อมูลในกรณีที่เซกเมนต์แรกเกิดการสูญหาย ซึ่งหลังจากเกินช่วงเวลาไทม์เอาต์ เซกเมนต์จะถูกส่งซ้ำอีกครึ่งหนึ่ง จะเห็นว่าเมื่อเซกเมนต์ที่ส่งไม่สำเร็จในครั้งแรกมาถึงฝั่งผู้รับก็สามารถที่จะส่งสัญญาณ ACK เพียงครั้งเดียว ก็สามารถที่จะบอกได้ว่าทั้ง 2 เซกเมนต์ได้รับอย่างถูกต้อง



ภาพที่ 2.5 การสูญหายและการส่งซ้ำข้อมูล

2.1.1.3 ตัวจับเวลาของ TCP

TCP ใช้การจับเวลาหลายแบบเพื่อให้แน่ใจว่าจะไม่เกิดดีเลย์มากเกินไปในขณะการสื่อสาร ซึ่งตัวจับเวลาเหล่านี้จะจัดการแก้ปัญหาต่างๆ ได้เป็นอย่างดี แต่อาจจะยังไม่สามารถเห็นได้อย่างชัดเจนในการวิเคราะห์ในครั้งแรก ซึ่งในส่วนนี้จะแสดงให้เห็นว่าตัวจับเวลาของ TCP ทำหน้าที่ส่งข้อมูลให้ไปถึงปลายทางได้อย่างถูกต้อง

2.1.1.3.1 ตัวจับเวลาการส่งซ้ำ

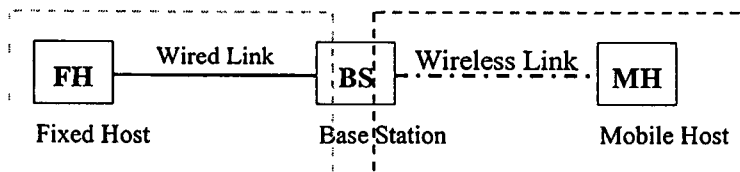
ตัวจับเวลาการส่งซ้ำจะทำหน้าที่จัดการกับไทม์เอาต์การส่งซ้ำ (Retransmission Timeouts : RTOs) ซึ่งเกิดขึ้นเมื่อเวลาที่ใช้ในการส่งข้อมูลไปยังปลายทางและสัญญาณ ACK มาถึงต้นทางมากเกินไปกว่าค่าที่กำหนด โดยค่าไทม์เอาต์จะเปลี่ยนแปลงไปตามชนิดของเครือข่ายซึ่งมีความเร็วที่ต่างกัน เมื่อตัวจับเวลาถึงค่าที่กำหนดข้อมูลจะถูกส่งซ้ำและทำการปรับค่า RTO ซึ่งโดยปกติจะทำการปรับค่าแบบเอ็กซ์โพเนนเชียลไปจนกระทั่งถึงค่าสูงสุดที่ตั้งไว้ โดยถ้าค่าสูงสุดเกิดขึ้นแสดงว่าการเชื่อมต่อล้มเหลว โดยค่าสำหรับไทม์เอาต์จะถูกพิจารณาโดยการวัดเวลาเฉลี่ยของข้อมูลที่ถูกส่งออกไปและสัญญาณ ACK ที่ตอบกลับมา ซึ่งเรียกว่าเวลาการเดินทางวนรอบ (Round Trip Time : RTT)

2.1.1.3.2 ตัวจับเวลารักษาการเชื่อมต่อและตัวจับเวลารอการทำงาน (Keep - Alive Timer and Idle Timer)

ทั้งตัวจับเวลารักษาการเชื่อมต่อและรอการทำงานเป็นฟังก์ชันที่ถูกเพิ่มเข้ามาใน TCP ภายหลัง โดยตัวจับเวลารักษาการเชื่อมต่อจะส่งแพ็คเก็ตว่างในช่วงเวลาที่กำหนด เพื่อให้แน่ใจว่าการสื่อสารระหว่างต้นทางและปลายทางจะยังคงทำงานอยู่ แต่ถ้าไม่มีการตอบสนองภายหลังจากการส่งข้อความตามช่วงเวลาตัวจับเวลารอการทำงานก็จะหมดเวลาทำให้สันนิษฐานได้ว่าการสื่อสารยกเลิก

ค่าของตัวจับเวลารักษาการเชื่อมต่อโดยปกติจะตั้งค่าโดยแอปพลิเคชันต่างๆ ซึ่งมีค่าอยู่ระหว่าง 4 ถึง 45 วินาที และค่าตัวจับเวลารอการทำงานโดยปกติคือ 360 วินาที

2.1.2 การใช้งาน TCP บนเครือข่ายไร้สาย



ภาพที่ 2.6 การเชื่อมต่อระหว่างเครือข่ายไร้สายและมีสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

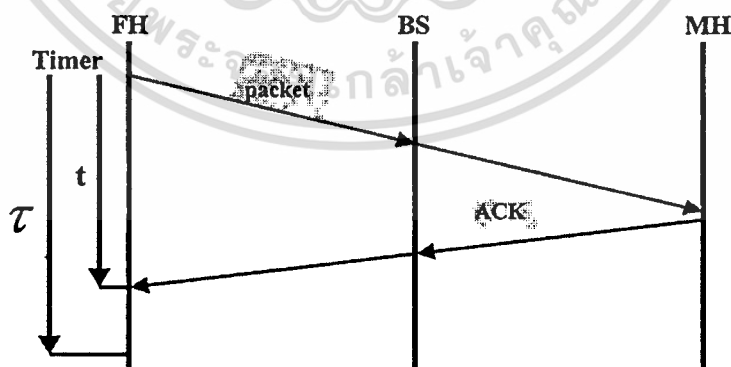
จากภาพ 2.6 จะเห็นว่าเมื่อนำเครือข่ายไร้สายมาต่อเข้ากับเครือข่ายมีสายจะประกอบไปด้วยเครือข่าย 3 ส่วน ส่วนแรกคือ โฮสต์อยู่กับที่ (Fixed Host : FH) ส่วนที่สองสถานีฐาน (Base Station : BS) และส่วนที่สามโฮสต์เคลื่อนที่ (Mobile Host : MH) โดย FH จะเชื่อมต่อไปยัง BS ด้วยเครือข่ายมีสาย และ MH จะเชื่อมต่อไปยัง BS ด้วยเครือข่ายไร้สายในโครงสร้างนี้ BS จะทำหน้าที่เป็นประตูหรือเกตเวย์ที่ใช้เชื่อมต่อระหว่างเครือข่ายไร้สายและมีสาย รวมทั้งทำการส่งข้อมูลและแก้ไขความผิดพลาดในการส่งข้อมูลไปยัง MH ด้วย โดย BS จะมีตัวจับเวลาในส่งข้อมูล ถ้าข้อมูลที่ส่งไปยัง MH ไม่ได้รับสัญญาณตอบกลับในเวลาที่กำหนด ก็จะทำการส่งซ้ำข้อมูลไปยัง MH [3]

จะเห็นว่าเมื่อมีการนำเอา TCP มาใช้งานบนเครือข่ายไร้สาย ก็ได้นำมาติดตั้งที่ต้นทางและปลายทางเหมือนเดิม ก็คือที่ FH และ MH แต่จะมีส่วนที่เพิ่มขึ้นมาต่างจากเครือข่ายมีสายปกติก็คือ BS ซึ่งการทำงานของ TCP บนเครือข่ายไร้สายแสดงได้ดังภาพที่ 2.7

ซึ่งจะเห็นว่าทำงานเหมือนกับ TCP ที่ใช้ในเครือข่ายแบบมีสายคือ ข้อมูลจะถูกส่งจาก FH ไปยัง BS ผ่านเครือข่ายมีสาย จากนั้น BS ก็จะทำการส่งข้อมูลต่อไปยัง MH โดยผ่านเครือข่ายไร้สาย และ MH ก็ส่งสัญญาณตอบกลับหรือ ACK ไปยัง BS จากนั้น BS ก็จะทำการส่งต่อไปยังโฮสต์อยู่กับที่ผ่านเครือข่ายมีสายโดย

- t คือ เวลาที่ใช้ในการส่งข้อมูลจากต้นทางไปถึงปลายทางรวมเวลาในการส่งสัญญาณ ACK
- τ คือ ค่าเวลาไทม์เอาท์

จะเห็นว่าจากภาพที่ 2.7 ข้อมูลสามารถส่งสำเร็จได้ภายในเวลาไทม์เอาท์



ภาพที่ 2.7 การทำงานของ TCP ระหว่างเครือข่ายมีสายและไร้สาย

2.1.2.1 ปัญหาเมื่อใช้งาน TCP บนเครือข่ายไร้สาย

จะเห็นว่า TCP ได้ถูกออกแบบมาเพื่อให้เหมาะสมกับการใช้งานบนเครือข่ายมีสาย ซึ่งปัญหาที่เกิดขึ้นเมื่อข้อมูลไม่สามารถไปถึงปลายทาง มักจะเกิดจากความแออัดของการส่งข้อมูลเป็นหลัก แต่เมื่อมีการนำมาใช้งานบนเครือข่ายไร้สายปัญหาของการส่งข้อมูลบนเครือข่ายไร้สาย จะเกิดจากการผิดพลาดของตัวช่องทางในการส่งข้อมูลนั่นเอง ซึ่งปกติจะมีสัญญาณรบกวนสูงและผิดพลาดได้ง่ายจะเห็นว่าสาเหตุของความผิดพลาดในการส่งข้อมูลของ 2 ระบบนี้ต่างกัน

ภาพที่ 2.8 แสดงให้เห็นถึงปัญหาการส่งข้อมูลที่ซ้ำซ้อนเมื่อนำ TCP มาใช้งานบนเครือข่ายไร้สาย ซึ่งจะพบว่าในขณะที่มีการส่งข้อมูลจาก BS ไปยัง MH (ภาพที่ 2.8(ก)) หรือจาก MH ไปยัง BS นั้น (ภาพที่ 2.8(ข)) ในบางครั้งอาจเกิดการสูญหายหรือส่งไม่สำเร็จ ซึ่ง BS จะทำการแก้ไขโดยทำการส่งซ้ำ ซึ่งในระหว่างนั้นบางครั้งก็เกิดไทม์เอาต์ที่ขึ้นที่ FH ทำให้เกิดการส่งข้อมูลซ้ำออกมา แต่ในระหว่างนั้นข้อมูลจาก BS ก็อาจจะส่งสำเร็จไปยัง MH และกำลังส่งสัญญาณ ACK กลับไปยัง FH ซึ่งจะทำให้เกิดการส่งข้อมูลที่ซ้ำซ้อน (Duplicated Packet) ซึ่งทำให้ประสิทธิภาพการทำงานของ TCP ลดลง

จะเห็นว่าสิ่งหนึ่งที่แตกต่างกันของการใช้งาน TCP บนเครือข่ายมีสายและเครือข่ายไร้สายก็คือ สถานีฐาน เนื่องจากสถานีฐานมีขั้นตอนการส่งซ้ำข้อมูลไปยังโฮสต์เคลื่อนที่ซึ่งทำให้เกิดการทำงานที่ซ้ำซ้อนกันระหว่าง TCP กับสถานีฐาน ซึ่งในงานวิจัยนี้ได้พยายามแก้ปัญหาการส่งข้อมูลที่ซ้ำซ้อนในจุดนี้

2.2 ทฤษฎีการเข้าคิวแบบ M/M/1

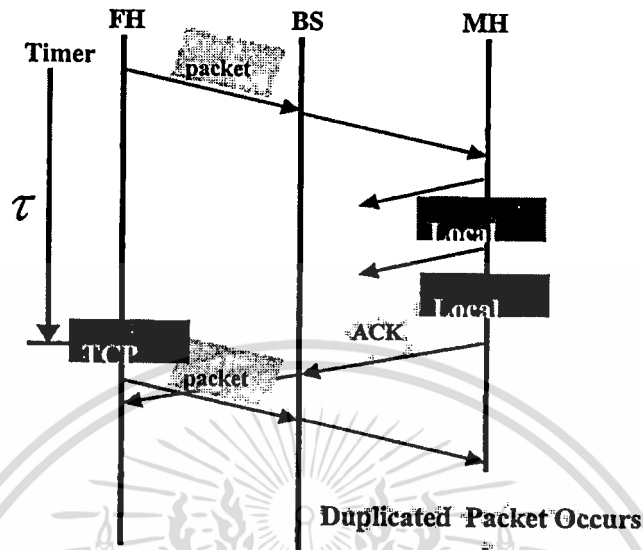
แบบจำลองของระบบการเข้าคิวแบบ M/M/1 เป็นแบบจำลองที่มีผู้ใช้กันมาก นอกจากนี้ยังใช้เป็นระบบที่เริ่มต้นในการศึกษาเกี่ยวกับการเข้าคิว ความหมายของ M/M/1 คือ [4]

M ตัวแรก หมายถึง รูปแบบการมาถึงในที่นี้ก็คือเวลาระหว่างการมาถึงเป็นการแจกแจงแบบเอ็กโปเนนเชียล (Exponential Distribution) และมีรูปแบบการมาถึงเป็นแบบสุ่มโดยมีการแจกแจงแบบพัชซอง (Poisson Distribution)

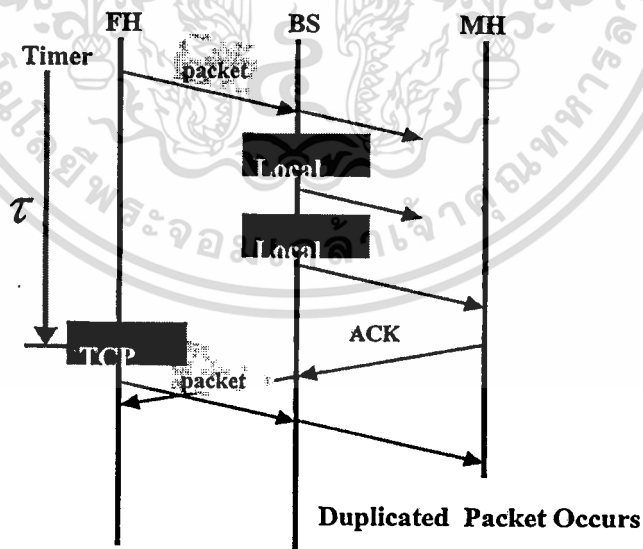
M ตัวที่สอง หมายถึง การแจกแจงเวลาการบริการในที่นี้ก็คือการแจกแจงแบบเอ็กซ์โปเนนเชียล

1 หมายถึง จำนวนเซิร์ฟเวอร์ ซึ่งก็คือ 1 เซิร์ฟเวอร์.

ลักษณะของระบบการเข้าคิวแบบ M/M/1 แสดงดังภาพที่ 2.9



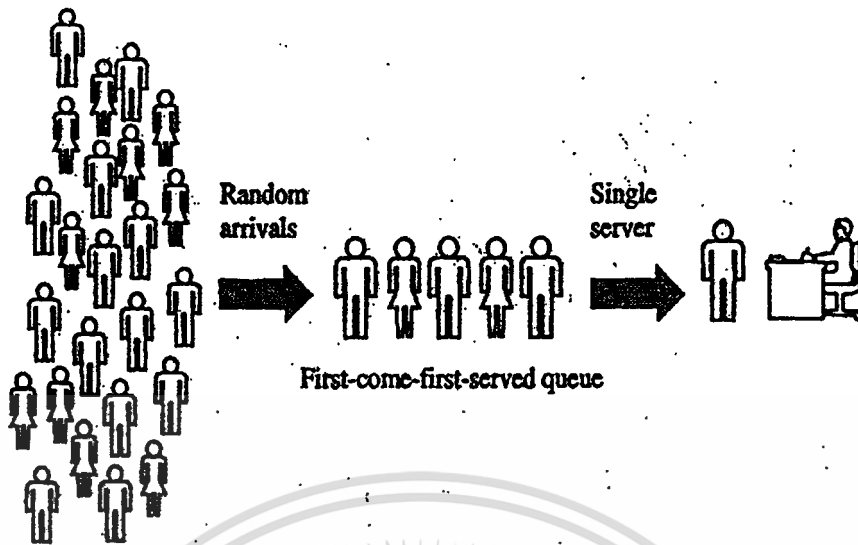
(ก) การส่งจาก BS ไป MH



(ข) การส่งจาก MH ไป BS

ภาพที่ 2.8 การส่งข้อมูลที่ซ้ำซ้อนของ TCP บนเครือข่ายไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



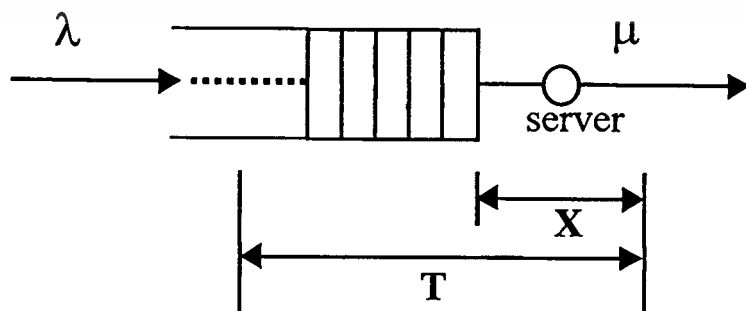
ภาพที่ 2.9 รูปแบบการเข้าคิวแบบ M/M/1

โดยแบบจำลอง M/M/1 จะมีรูปแบบที่เป็นข้อกำหนดดังนี้ [4]

- อัตราการมาถึงจะเป็นแบบสุ่ม โดยมีการแจกแจงแบบพัวซอง ซึ่งก็คือมีรูปแบบการมาที่ไม่พร้อมกัน และระยะเวลาในการมาถึงไม่แน่นอน
- เวลาในการบริการเป็นการแจกแจงแบบเอ็กซ์โพเนนเชียล
- ผู้ใช้บริการที่เป็นไปได้มีขนาดไม่จำกัด ซึ่งก็คืออัตราที่ผู้ให้บริการมาถึงไม่มีผลต่อจำนวนผู้ให้บริการที่รออยู่ในคิวและกำลังได้รับบริการอยู่ในขณะนั้น
- แบบแผนการเข้าคิวจะเป็นแบบมาถึงก่อนบริการก่อน (First Come First Served : FCFS)

2.2.1 ค่าพารามิเตอร์และค่าการเริ่มต้นในการคำนวณสำหรับการเข้าคิวแบบ M/M/1

ค่าพารามิเตอร์ที่จำเป็นสำหรับการเข้าคิวแบบ M/M/1 แสดงดังภาพที่ 2.10 มีดังนี้



ภาพที่ 2.10 ค่าพารามิเตอร์ของการเข้าคิวแบบ M/M/1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- อัตราการมาถึงของผู้ใช้บริการเฉลี่ย λ เช่น ผู้ใช้บริการต่อนาที
- เวลาในการบริการเฉลี่ย X เช่น 2 นาที โดย X จะมีค่าเท่ากับ $1/\mu$

จะพบว่าการระบุหน่วยเวลาจะไม่ค่อยมีความสำคัญนัก เมื่ออัตราการมาถึงและอัตราการบริการใช้หน่วยเวลาเดียวกัน ในบางครั้งเพื่อความง่ายในการดูความสัมพันธ์ในระบบการเข้าคิว จะทำการสมมติให้ $X = 1$ หน่วยเวลา ซึ่งทำให้เหลือเพียงแค่การเปลี่ยนแปลงอัตราการมาถึงทำให้มองได้เทียบเท่ากับประสิทธิภาพของเซิร์ฟเวอร์

เมื่อมีการกำหนดค่าให้กับ λ และ X สิ่งต่อไปที่จะทำการคำนวณก็คือ ประสิทธิภาพของเซิร์ฟเวอร์ โดยค่าหรือช่วงประสิทธิภาพของเซิร์ฟเวอร์ก็คือ 0 ถึง 1

- 0 หมายถึง เซิร์ฟเวอร์ว่างตลอดเวลา คือ ไม่มีผู้เข้ามาใช้บริการ
- 1 หมายถึง เซิร์ฟเวอร์มีการบริการตลอดเวลา

ในบางครั้งก็มีการแสดงค่าประสิทธิภาพของเซิร์ฟเวอร์ในรูปแบบของเปอร์เซ็นต์เช่นกัน แต่โดยปกติเวลาในการคำนวณจะใช้ค่า 0 ถึง 1 แล้วจึงทำการแสดงผลในตอนสุดท้ายเป็นเปอร์เซ็นต์ ซึ่งสูตรการคำนวณค่าประสิทธิภาพของเซิร์ฟเวอร์ (Server Utilization) แสดงดังนี้

$$\rho = \lambda X \quad (2.1)$$

เมื่อ ρ คือค่าประสิทธิภาพของเซิร์ฟเวอร์

สิ่งสำคัญที่ต้องทำการตรวจสอบสำหรับค่าประสิทธิภาพของเซิร์ฟเวอร์คือ $\rho < 1$ ซึ่งก็คือเซิร์ฟเวอร์จะต้องไม่เกิดอาการภาระเกิน (Overload) ซึ่งจะทำให้คิวหรือจำนวนของผู้ใช้บริการที่รอการบริการอยู่เพิ่มขึ้นไปเรื่อยโดยไม่มีข้อจำกัด ทั้งนี้เป็นไปตามข้อกำหนดของระบบที่ว่าความนิยมของผู้ใช้บริการไม่จำกัดและแบบแผนการเข้ามาถึงไม่มีผลกระทบอันเนื่องมาจากสถานะของคิว จากจุดนี้จะพบว่าผู้บริการจะยังคงเข้ามาที่คิวตามอัตราการมาถึงเช่นเดิม แม้ว่าเซิร์ฟเวอร์จะไม่สามารถบริการได้ตามอัตรานั้นก็ตาม ในส่วนที่ค่า $\rho = 1$ ก็ไม่ค่อยจะมีความเสถียรเช่นกัน

2.2.2 เวลาในระบบหรือเวลาการเข้าคิว (Time-in-System or Queueing Time)

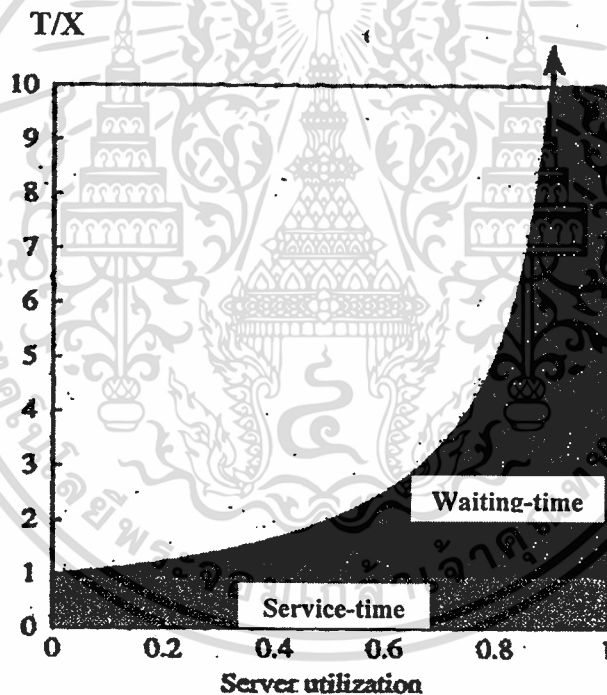
โดยทั่ว ๆ ไปในการวัดผลที่ดีที่สุดโดยรวมของการบริการก็คือเวลาเฉลี่ยที่ถูกใช้ในระบบ เมื่อผู้ใช้บริการมารับบริการตั้งแต่เข้าสู่ระบบจนถึงที่สุด ซึ่งสำหรับการเข้าคิวแบบ M/M/1 สามารถคำนวณ ได้ดังนี้

$$T = \frac{X}{1-\rho} \quad (2.2)$$

เมื่อ T คือเวลาในระบบหรือเวลาในการเข้าคิว

X คือเวลาในการบริการเฉลี่ย

โดยภาพที่ 2.11 แสดงลักษณะความสัมพันธ์ของสมการ



ภาพที่ 2.11 เวลาในระบบของการเข้าคิวแบบ M/M/1

2.2.3 ความไม่เป็นเชิงเส้นของเวลาในระบบ

ภาพที่ 2.11 อาจจะเป็นภาพที่รู้จักกันดีที่สุดในทฤษฎีการเข้าคิว โดยลักษณะหนึ่งที่เราเห็นได้ชัดเจนก็คือ T จะเพิ่มขึ้นอย่างรวดเร็วโดยเฉพาะเมื่อค่าประสิทธิภาพของเซิร์ฟเวอร์เข้าใกล้ 1 ซึ่งก็เป็นลักษณะพื้นฐานของระบบการเข้าคิวอีกหลายชนิด เมื่อค่าประสิทธิภาพของเซิร์ฟเวอร์เพิ่มขึ้นก็จะ

ทำให้เวลาการเข้าคิวเพิ่มขึ้นเร็วมากขึ้นเรื่อย ๆ เว้นแต่จะมีวิธีการที่จะมาระงับการมาถึงของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้กับการแข่งขันเพื่อการศึกษาเท่านั้น เมื่อผู้เผยแพร่เห็นประโยชน์ทางวิชาการ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริการเมื่อระบบอยู่ในสถานะแอ็ด ซึ่งสามารถแสดงตัวอย่างการไม่เป็นเชิงเส้นของเวลาที่ใช้ใน ระบบได้ง่าย ๆ ดังนี้ สมมติให้ เวลาในการบริการเฉลี่ยคือ 2 นาที 24 วินาที สิ่งที่จะเกิดขึ้นเมื่ออัตราการมาถึงเฉลี่ยเพิ่มขึ้นจาก 10 ผู้ใช้บริการต่อชั่วโมง (Case A1) ไปเป็น 12 ผู้ใช้บริการต่อชั่วโมง (Case A2) แสดงในตารางที่ 2.1 ซึ่งได้จากการคำนวณ

ตารางที่ 2.1 ความไม่เป็นเชิงเส้นของเวลาการเข้าคิว จาก 10 คน เป็น 12 คนต่อชั่วโมง

	Case A1	Case A2	Increase
λ	10/hr	12/hr	20 %
ρ	0.40	0.48	
$T = X/(1-\rho)$	240 secs	277 secs	15.4 %

จะเห็นว่าเมื่อเพิ่มจาก 10 ไปเป็น 12 ผู้ใช้บริการต่อชั่วโมงแสดงให้เห็นถึงการเพิ่มขึ้นถึง 20 % ในอัตราการมาถึง ซึ่งก็จะสอดคล้องกับการเพิ่มในเวลาการเข้าคิวที่เพิ่มจาก 240 ไปถึง 277 วินาที ซึ่งเพิ่ม 15.4 % จากนั้นมาลองสังเกตดูว่าถ้าอัตราการมาถึงเพิ่มจาก 20 (Case B1) ไปเป็น 22 ผู้ใช้บริการต่อชั่วโมง (Case B2) ซึ่งจะให้ผลการคำนวณดังตารางที่ 2.2

ตารางที่ 2.2 ความไม่เป็นเชิงเส้นของเวลาการเข้าคิว จาก 20 คน เป็น 22 คนต่อชั่วโมง

	Case B1	Case B2	Increase
λ	20/hr	22/hr	10 %
ρ	0.80	0.88	
$T = X/(1-\rho)$	720 secs	1200 secs	67 %

การเพิ่มขึ้นของอัตราการมาถึงจาก 20 ไปเป็น 22 ผู้ใช้บริการ 1 ชั่วโมงเป็นการเพิ่มถึง 10 % ของอัตราการมาถึง ซึ่งผลที่ได้ของเวลาการเข้าคิวก็คือเพิ่มขึ้น 67 % ซึ่งจะมีการเพิ่มที่รวดเร็วกว่ากรณีแรก

2.2.4 เวลาการรอ (Waiting Time)

ในบางครั้งจะมีการพิจารณาในเรื่องเวลาในการรอมากกว่าเวลาในระบบ โดยเวลาในการรอเฉลี่ยจะได้จากสมการดังนี้

$$W = \frac{\rho X}{1-\rho} \quad (2.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ W คือเวลาการรอ

ซึ่งเวลาในระบบเฉลี่ย เวลาการรอเฉลี่ย และเวลาการบริการเฉลี่ยจะมีความสัมพันธ์ดังนี้

$$T = W + X \tag{2.4}$$

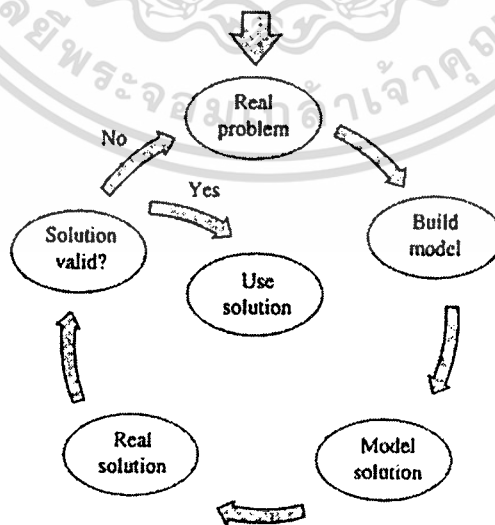
ซึ่งสามารถเขียนได้เป็น

$$T = \frac{\rho X}{1 - \rho} + X = \frac{X}{1 - \rho} \tag{2.5}$$

2.2.5 การสร้างแบบจำลอง

เมื่อต้องการที่จะใช้ทฤษฎีการเข้าคิวมาสร้างแบบจำลองของระบบที่สนใจขึ้นมาจะต้องทำความเข้าใจให้ชัดเจนถึงความหมายของคำว่าแบบจำลองก่อน โดยแบบจำลองคือสิ่งที่ใช้แสดงบางสิ่งบางอย่างในโลกที่เป็นจริง ในทางด้านความคิดหรือนามธรรมหรือบางส่วนของลักษณะหรือพฤติกรรมที่ต้องการติดตามเพื่อทำความเข้าใจ ซึ่งแบบจำลองก็มีอยู่หลายแบบ ขึ้นอยู่กับสถานะความเป็นจริงที่เกี่ยวข้อง เช่น การสร้างรถใหม่ก็อาจจะสร้างแบบจำลองที่มีขนาดเท่าของจริงจากไม้หรืออาจใช้โปรแกรมคอมพิวเตอร์เพื่อเลียนแบบการทำงานของระบบตรงตัวก่อนที่จะนำไปสร้างจริง ฉะนั้นจะเห็นว่าแบบจำลองจะถูกใช้ในหลายหน้าที่และรูปแบบ

2.2.5.1 วงรอบของการสร้างแบบจำลอง (Modelling Cycle)



ภาพที่ 2.12 วงรอบของการสร้างแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการสร้างแบบจำลองแสดงดังภาพที่ 2.12 ซึ่งเริ่มต้นจากด้านบนจะพบว่าจะต้องทำความเข้าใจกับปัญหาที่เป็นจริงก่อน เมื่อมีความรู้และความเข้าใจที่ดีเพียงพอจึงจะเริ่มเข้าสู่ขั้นตอนการสร้างแบบจำลอง จากนั้นจึงทำการตัดสินใจว่าด้านไหนของปัญหาหรือระบบที่แท้จริง ซึ่งจะถูกบรรจุเข้าไปในแบบจำลอง จากนั้นก็มาทำการพิจารณาถึงลักษณะเฉพาะตัวของแบบจำลอง เช่น ทฤษฎีการเข้าคิวหรือใช้สมการทางคณิตศาสตร์ในการเขียนแบบเป็นต้น รวมทั้งระดับของรายละเอียดที่จำเป็นต้องใช้ ซึ่งในขั้นตอนแรกนี้จะต้องทำการกระจายปัญหาจริงออกมาแล้วทำการหาแพ็คเกจที่สำคัญหรือน่าสนใจของปัญหา ซึ่งในการที่หาแพ็คเกจนี้เป็นทั้งศิลปะและทักษะในการสร้างแบบจำลอง ในขั้นตอนถัดไปทักษะทางด้านเทคนิคจะเป็นสิ่งที่จำเป็น แต่ถ้าทำขั้นตอนแรกได้ถูกต้องก็จะทำให้ได้ลักษณะพื้นฐานของแบบจำลองที่ดี ซึ่งเป็นประสบการณ์ที่มีคุณค่ามาก

ในการสร้างแบบจำลองที่ต้องการจะแก้ไขปัญหา ซึ่งในที่นี้จะใช้ทฤษฎีการเข้าคิวหมายความว่า จะต้องทำการหาสมการทางคณิตศาสตร์เพื่อใช้หาค่าเวลาในการรอคอยและความยาวของคิว รวมทั้งลักษณะอื่นๆ ที่น่าสนใจ ซึ่งในความเป็นจริงก็อาจจะไม่ต้องทำการหาสมการทางคณิตศาสตร์ แต่อาจจะแก้ปัญหามาตรฐานแบบจำลองโดยการใช้เทคนิคทางตัวเลขหรือการประมาณ ซึ่งเป็นขั้นตอนที่ผู้เชี่ยวชาญทั่วไปใช้กันอยู่เนื่องจากว่าในบางครั้งเป็นการยากในเชิงเทคนิคหรือมีความสนใจแค่เพียงบางส่วนของแบบที่กำลังศึกษาอยู่เท่านั้น แต่มีสิ่งหนึ่งที่ต้องทำความเข้าใจคือในขั้นตอนของการสร้างแบบจำลองเป็นแค่เพียง 1 ขั้นตอนในการใช้งานแบบจำลองเท่านั้น

ในขั้นตอนถัดมา 2 ขั้นตอน ในบางครั้งก็เป็นสิ่งที่ยากในการวินิจฉัยหรือตีความ เนื่องจากในขณะนี้เรามีแบบจำลองและวิธีการแก้ปัญหาเพื่อตอบคำถามและจากนั้นก็แปลความหมายของคำตอบไปสู่ระบบที่มีอยู่จริง และทำการฟังคำแนะนำที่มีเหตุผลเพื่อนำมาตรวจสอบ ซึ่งบางครั้งแบบจำลองจะมีความเป็นนามธรรมสูงหรือมีการละเลยข้อกำหนดที่มีอยู่ในระบบจริง ซึ่งต้องทำการตรวจสอบเพื่อให้แบบจำลองมีพฤติกรรมเหมือนระบบที่เป็นจริงอยู่จริง ในขั้นตอนของการปรับแต่งหรือตรวจสอบว่าแบบจำลองสามารถอธิบายการทำงานของระบบที่มีอยู่ได้ เป็นขั้นตอนที่มีความจำเป็นในการสร้างแบบจำลองให้มีความน่าเชื่อถือ

ส่วนมากจะต้องทำวงรอบของการสร้างแบบจำลองมากกว่า 1 ครั้งเสมอเนื่องจากความพยายามในการสร้างแบบจำลองเกิดจากความไม่เข้าใจหรือไม่สามารถเห็นได้ในระบบที่เป็นจริงที่กำลังศึกษาอยู่ เพราะฉะนั้นการสร้างแบบจำลองจึงเป็นวิธีการทางการศึกษาที่มีประสิทธิภาพสูง และจะเป็นผลเมื่อการวิเคราะห์มีความเข้าใจอย่างลึกซึ้งในระบบที่มีอยู่จริง แบบจำลองจึงจะสามารถทำนายถึงความเป็นไปในการปรับแต่งหรือพฤติกรรมของระบบได้

บทที่ 3

แบบจำลองทางคณิตศาสตร์ และวิธีการรีเซตไทม์เอาต์ (Timeout Resetting)

ในบทนี้จะกล่าวถึงแบบจำลองทางคณิตศาสตร์ของการเชื่อมต่อระหว่างเครือข่ายไร้สาย และมีสาย โดยใช้ทฤษฎีการเข้าคิวมาวิเคราะห์การทำงานของสถานีฐานที่ทำหน้าที่เชื่อมต่อระหว่างเครือข่ายมีสายและไร้สาย และทำการสร้างเป็นแบบจำลองทางคณิตศาสตร์ของสถานีฐานขึ้นมาซึ่งสามารถใช้สมการทางคณิตศาสตร์ของทฤษฎีการเข้าคิว มาวิเคราะห์แบบจำลองที่สร้างขึ้นมา ทำให้สามารถหาค่าต่างๆ และวิเคราะห์การทำงานของสถานีฐานได้ จากนั้นก็ทำการพิจารณาการทำงานของเครือข่ายในช่วงเวลาที่สถานีฐานใช้ในการส่งข้อมูลไปยังโฮสต์เคลื่อนที่มีค่าสูงเกินกว่าค่าไทม์เอาต์ที่ TCP ต้นทาง ซึ่งทำให้เกิดการส่งข้อมูลซ้ำซ้อนจากการพิจารณาปัญหาที่แบบจำลองได้นำเสนอวิธีการรีเซตไทม์เอาต์เพื่อลดการเกิดการส่งข้อมูลซ้ำซ้อน ในขณะที่สถานีฐานใช้เวลาในการส่งข้อมูลไปยังโฮสต์เคลื่อนที่สูงกว่าเวลาไทม์เอาต์ของ TCP ต้นทาง

3.1 แบบจำลองทางคณิตศาสตร์

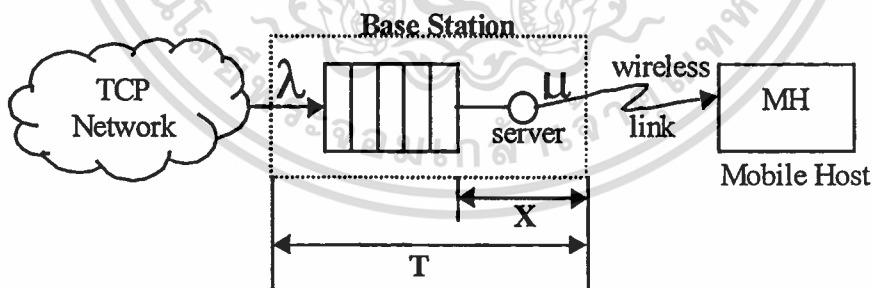
ในการสร้างแบบจำลองนั้นจำเป็นต้องศึกษาถึงพฤติกรรมและข้อมูลที่เกี่ยวข้องกับระบบนั้น ๆ ซึ่งสิ่งที่พิจารณาในงานวิจัยนี้ก็คือการพิจารณาระบบการเชื่อมต่อระหว่างเครือข่ายไร้สาย ดังแสดงในภาพที่ 2.6

เมื่อทำการพิจารณาจะพบว่า ข้อมูลที่ถูกส่งจากโฮสต์อยู่กับที่มายังสถานีฐานก็คือผู้ใช้บริการในระบบการเข้าคิวนั่นเอง โดยการมาถึงของข้อมูลแต่ละแพ็คเก็ตก็คือผู้ใช้บริการ 1 คนนั่นเอง จากนั้นสถานีฐานจะเก็บข้อมูลไว้ในบัฟเฟอร์เพื่อเตรียมที่จะส่งต่อไปซึ่งเปรียบเสมือนกับการเข้าคิวรอในระบบการเข้าคิวเช่นกัน และเวลาที่สถานีฐานใช้ในการส่งข้อมูลไปยังโฮสต์เคลื่อนที่และรอตรวจสอบว่าสามารถส่งได้สำเร็จหรือไม่ก็คือช่วงเวลาของการบริการ จะเห็นว่าสถานีฐานจะทำหน้าที่เสมือนกับเป็นเซิร์ฟเวอร์ในระบบการเข้าคิวจึงทำให้สามารถที่จะมองระบบการเชื่อมต่อระหว่างเครือข่ายไร้สายและมีสายเป็นระบบการเข้าคิวได้ โดยมีสถานีฐานทำหน้าที่เป็นเซิร์ฟเวอร์ ซึ่งได้ใช้หลักการนี้มาสร้างแบบจำลองขึ้นมา

โดยได้ทำการกำหนดเงื่อนไขของสภาวะการเชื่อมต่อระหว่างเครือข่ายไร้สายและมีสายดังนี้

- ขนาดของบัฟเฟอร์ที่สถานีฐานมีขนาดไม่จำกัด ซึ่งก็คือไม่สามารถจำกัดขนาดของคิวนั่นเอง ข้อมูลหรือแพ็คเกจที่ถูกส่งมาจาก TCP ต้นทางหรือเครือข่าย TCP จะถูกรับเข้าบัฟเฟอร์เสมอ และไม่มีกฏการปฏิเสธการมาถึงของข้อมูล
- ข้อมูลที่คาดว่าจะส่งมาจาก TCP ต้นทางมีจำนวนไม่จำกัด ซึ่งก็คือความนิยมของผู้ใช้บริการ ไม่มีการจำกัดนั่นเอง
- สถานีฐานซึ่งทำการเชื่อมต่อระหว่างเครือข่ายไร้สายและมีสายจะทำการส่งต่อข้อมูลแบบมาถึงก่อนส่งออกก่อน ซึ่งเป็นลักษณะการทำงานปกติของสถานีฐาน หรือก็คือมีแบบแผนการเข้าคิวแบบมาก่อนถึงก่อนบริการก่อนนั่นเอง
- อัตราการส่งข้อมูลเฉลี่ยจาก TCP ต้นทางมายังสถานีฐานมีอัตรา λ แพ็คเกจ/วินาที โดยมีการมาถึงเป็นแบบสุ่มมีการแจกแจงแบบพัวซอง (Poisson Distribution)
- อัตราการบริการเฉลี่ยของสถานีฐานในการส่งข้อมูลจากเครือข่ายมีสายไปเครือข่ายไร้สายมีอัตรา μ แพ็คเกจ/วินาที โดยเวลาในการบริการต่อแพ็คเกจมีการแจกแจงแบบเอ็กซ์โพเนนเชียล (Exponential Distribution)

จากเงื่อนไขที่ได้กล่าวมาแล้ว ทำให้ได้แบบจำลองทางสถานีฐานในการเชื่อมต่อระหว่างเครือข่ายไร้สายและมีสายที่อยู่ในรูปแบบของระบบการเข้าคิวแบบ M/M/1 ตามที่ได้กล่าวไปแล้วในบทที่ 2 ซึ่งแสดงได้ดังภาพที่ 3.1



ภาพที่ 3.1 แบบจำลองการเข้าคิวแบบ M/M/1 ของสถานีฐาน

จากภาพที่ 3.1 ถ้า T เป็นเวลาโดยรวมของระบบที่แต่ละแพ็คเกจใช้บนเครือข่ายไร้สาย ก่อนที่จะไปถึงปลายทางหรือโฮสต์เคลื่อนที่ จากทฤษฎีการเข้าคิวแบบ $M/M/1$ จะทำให้ได้

$$T = \frac{1/\mu}{1-\rho} \quad (3.1)$$

เมื่อ

$$\rho = \frac{\lambda}{\mu} \quad (3.2)$$

ρ คือค่าประสิทธิภาพการทำงานของสถานีฐาน โดยปกติจะถูกออกแบบให้มีค่าน้อยกว่า 1 เสมอ เพราะถ้า ρ มีค่ามากกว่า 1 หมายความว่าอัตราการมาถึงมีค่ามากกว่าอัตราการให้บริการของเซิร์ฟเวอร์หรือ ($\lambda > \mu$) จะทำให้เกิดเป็นโอเวอร์โฟลว์หรือสถานีฐานไม่สามารถบริการได้ทันกับจำนวนข้อมูลที่เข้ามาจะทำให้เกิดการสะสมข้อมูลในบัฟเฟอร์ของสถานีฐานมากขึ้นไปเรื่อยๆ

เมื่อมีการกำหนดให้ X เป็นเวลาการบริการเฉลี่ยของสถานีฐาน เมื่อระบบอยู่ในภาวะปกติคือไม่มีความผิดพลาดเกิดขึ้นในเครือข่ายไร้สาย ซึ่งเวลาการบริการเฉลี่ยของสถานีฐานจะประกอบไปด้วยเวลาในการส่งข้อมูลจากสถานีฐานไปยังโฮสต์เคลื่อนที่และการตอบกลับจากโฮสต์เคลื่อนที่ไปยัง TCP ต้นทางที่ส่งข้อมูลรวมกัน จากสมการที่ (3.1) สามารถที่จะหาเวลาของระบบโดยรวมในเครือข่ายไร้สายของข้อมูลแต่ละแพ็คเกจที่ส่งในสภาวะไม่มีความผิดพลาดเกิดในเครือข่ายไร้สายได้ดังนี้

$$X = \frac{1}{\mu} \quad (3.3)$$

$$t = \frac{X}{1-\rho} \quad (3.4)$$

เมื่อ

t เป็นเวลาโดยรวมของระบบที่แต่ละแพ็คเกจใช้บนเครือข่ายไร้สาย ภายใต้สภาวะไม่มีความผิดพลาดในเครือข่ายไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ρ เป็นค่าคงที่ ซึ่งมีค่ามากกว่า 0 แต่มีค่าน้อยกว่า 1 รวมทั้ง X และ μ ภายใต้สภาวะไม่มีความผิดพลาดในเครือข่ายไร้สาย

จะพบว่าโดยปกติ เครือข่ายไร้สายจะเป็นระบบที่ช่องทางการสื่อสารมักจะมีสัญญาณรบกวนหรือความผิดพลาดเกิดขึ้นได้ง่ายอันเนื่องมาจากลักษณะของช่องทางการสื่อสารของระบบเอง

ถ้าให้ ε คือค่าความน่าจะเป็นของความผิดพลาดที่เกิดขึ้นในปัจจุบันของเครือข่ายไร้สาย ซึ่งเมื่อพิจารณาแล้วจะพบว่าเมื่อมีความผิดพลาดเกิดขึ้นในปัจจุบันของเครือข่ายไร้สายจะทำให้เวลาที่ใช้ในการส่งข้อมูลจากสถานีฐานไปเครือข่ายไร้สายมีค่ามากขึ้น หรือเวลาการบริการที่สถานีฐานมากขึ้นนั่นเอง ซึ่งจะพบว่าความผิดพลาดที่เกิดขึ้นจะมีความสัมพันธ์กับ t กล่าวคือเมื่อมีความผิดพลาดเกิดขึ้น ก็จะทำให้ t เพิ่มขึ้นด้วย ซึ่งจากสาเหตุที่กล่าวมาสามารถที่จะตัดแปลง T หรือเวลาโดยรวมของระบบที่แต่ละแพ็คเกจใช้บนเครือข่ายไร้สาย จากสมการที่ (3.1) ในรูปแบบของ t ในสมการที่ (3.4) ภายใต้สภาวะที่มีความผิดพลาดในเครือข่ายไร้สายได้ดังนี้

$$\begin{aligned} T &= \varepsilon + t \\ &= (\varepsilon + 1)t \end{aligned} \quad (3.5)$$

เมื่อแทนค่า t จากสมการที่ (3.4) ในสมการที่ (3.5) จะได้

$$T = (\varepsilon + 1) \frac{X}{1 - \rho} \quad (3.6)$$

จะพบว่าเมื่อ ε มีค่าเท่ากับ 0 หรือไม่มีความผิดพลาดเกิดขึ้นในเครือข่ายไร้สาย ค่าของ T จะเท่ากับ t

3.2 สภาวะดีและไม่ดีในเครือข่ายไร้สาย

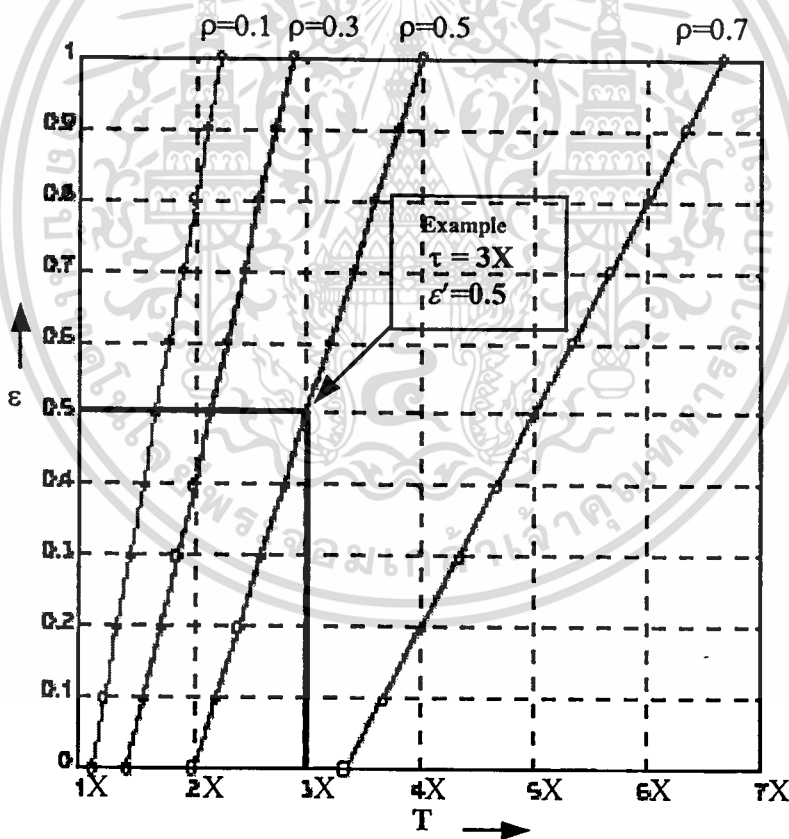
ในส่วนแรกได้กล่าวถึงวิธีการสร้างแบบจำลองของสถานีฐาน และได้แบบจำลองทางคณิตศาสตร์ซึ่งสามารถใช้พิจารณาหาค่าต่างๆ ในส่วนนี้จะทำการนำแบบจำลองนั้นมาเพื่อพิจารณาหาสภาวะดีและไม่ดีในเครือข่ายไร้สาย

เมื่อทำการกำหนดให้ τ คือค่า TCP ไทม์เอาต์ที่ใช้ในการตรวจสอบการส่งข้อมูลจากเครือข่ายมีสายไปเครือข่ายไร้สาย จะพบว่าสภาวะดีและไม่ดีในการเชื่อมต่อไร้สายสามารถสรุปออกมาได้ดังนี้

3.2.1 ถ้า $T \leq \tau$ เครือข่ายไร้สายอยู่ในสถานะดี (Good) นั่นคือเวลาโดยรวมของระบบที่ใช้ส่งข้อมูลมีค่าน้อยกว่าหรือเท่ากับ TCP ไทม์เอาท์ จะพบว่าการส่งข้อมูลระหว่าง TCP ดันทางหรือโฮสต์ที่อยู่กับที่ไปที่ปลายทางซึ่งเป็นโฮสต์เคลื่อนที่ สามารถที่จะทำได้สำเร็จได้ภายในเวลา TCP ไทม์เอาท์ ซึ่งจะไม่ทำให้เกิดการส่งที่ซ้ำซ้อนขึ้นนั่นเอง

3.2.2 ถ้า $T > \tau$ เครือข่ายไร้สายอยู่ในสถานะไม่ดี (Bad) นั่นคือเวลาโดยรวมของระบบที่ใช้ส่งข้อมูลมีค่ามากกว่า TCP ไทม์เอาท์ จะพบว่าการส่งข้อมูลระหว่าง TCP ดันทางหรือโฮสต์ที่อยู่กับที่ไปที่ปลายทางซึ่งเป็นโฮสต์เคลื่อนที่ ไม่สามารถที่จะทำได้สำเร็จได้ภายในเวลา TCP ไทม์เอาท์ ซึ่งจะทำให้เกิดการส่งซ้ำข้อมูลจาก TCP ดันทางและอาจจะเป็นสาเหตุให้เกิดการส่งข้อมูลที่ซ้ำซ้อน

เมื่อนำสมการที่ (3.6) มาพล็อตค่าเพื่อดูความสัมพันธ์ระหว่าง T กับ ε ที่ค่า ρ ต่างๆ โดยค่าของ T จะแสดงในรูปจำนวนเท่าของ X ก็จะได้ลักษณะที่แสดงออกมาในภาพที่ 3.2



ภาพที่ 3.2 ความสัมพันธ์ระหว่าง ε และ T ที่ค่า ρ ต่างๆ

โดยแกนตั้งจะเป็นแกนของ \mathcal{E} หรือความผิดพลาดในเครือข่ายไร้สายซึ่งมีค่าจาก 0 - 1 แกนนอนจะเป็นแกนของ T หรือเวลาของระบบโดยรวมซึ่งเขียนอยู่ในรูปแบบจำนวนเท่าของ X ซึ่งจะพบว่าเมื่อ \mathcal{E} มีค่ามากขึ้นก็จะทำให้ค่าของ T มีค่ามากขึ้นด้วยเช่นกัน นอกจากนี้ค่า ρ หรือค่าประสิทธิภาพยิ่งมีค่าสูงก็จะทำให้ค่า T มีค่าสูงขึ้นด้วยเช่นกันเมื่อเทียบกับ \mathcal{E} ที่เท่ากัน

จากภาพที่ 3.2 จะพบว่าที่จุดที่ค่าของ \mathcal{E} ที่ T มีค่าเท่ากับ τ หรือ TCP ใหม่เอาท์จะเป็นจุดวิกฤติที่มีการเปลี่ยนสถานะของเครือข่ายไร้สายจากสถานะดีไปไม่ดีหรือไม่ดีไปดีคือถ้า \mathcal{E} มีค่ามากเกินไปจุดนี้เครือข่ายไร้สายก็อยู่ในสถานะไม่ดี แต่ถ้าค่า \mathcal{E} มีค่าต่ำกว่าหรือเท่ากับจุดนี้เครือข่ายไร้สายก็จะอยู่ในสถานะดี

ถ้ากำหนดให้ \mathcal{E}' เป็นค่าวิกฤติของ \mathcal{E} ในการเปลี่ยนสถานะ จากสมการที่ (3.6) เมื่อ $T = \tau$ จะได้ว่า

$$\tau = (\mathcal{E}' + 1) \frac{X}{1 - \rho} \quad (3.7)$$

และ

$$\mathcal{E}' = \frac{\tau(1 - \rho)}{X} - 1 \quad (3.8)$$

จากสมการที่ (3.8) ทำให้สามารถที่จะหาค่า \mathcal{E}' หรือค่าความผิดพลาดวิกฤติในการเปลี่ยนสถานะได้ ถ้ารู้ค่าของ τ และ ρ ดังตัวอย่าง

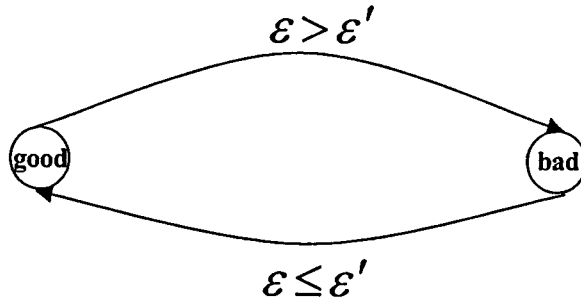
กำหนดให้ $\tau = 3X$ และ $\rho = 0.5$ ดังภาพที่ 3.2 จากสมการ (3.8)

$$\mathcal{E}' = \frac{3X(1 - 0.5)}{X} - 1$$

$$= 3(0.5) - 1$$

$$\mathcal{E}' = 0.5$$

ซึ่งก็คือเครือข่ายไร้สายจะอยู่ในสถานะไม่ดี เมื่อ \mathcal{E} มีค่ามากกว่า 0.5 ซึ่งจะทำให้ T มีค่ามากกว่า τ หรือ $3X$ ซึ่งสามารถแสดงเป็นสถานะของการเปลี่ยนแปลงได้ดังภาพที่ 3.3 ซึ่งมีอยู่ 2 เงื่อนไขคือ



ภาพที่ 3.3 การเปลี่ยนสถานะระหว่างสถานะดีและไม่ดี

- เมื่อ $\varepsilon \leq \varepsilon'$ ซึ่งก็จะพบว่า $T \leq \tau$ ก็คือเครือข่ายไร้สายอยู่ในสถานะดี (Good)
- เมื่อ $\varepsilon > \varepsilon'$ ซึ่งก็จะพบว่า $T > \tau$ ก็คือเครือข่ายไร้สายอยู่ในสถานะไม่ดี (Bad)

จากตัวอย่างที่แสดงในรูป 3.2 เมื่อกำหนดให้ $\tau = 3X$ และ $\varepsilon' = 0.5$ จะพบว่าสามารถที่จะแบ่งช่องของความผิดพลาดในปัจจุบันหรือ ε ได้เป็น 2 ส่วนคือ

- $0 \leq \varepsilon \leq \varepsilon'$ เป็นช่วงสถานะดีของเครือข่ายไร้สาย
- $\varepsilon' < \varepsilon \leq 1$ เป็นช่วงสถานะไม่ดีของเครือข่ายไร้สาย

จะเห็นว่า $1 - \varepsilon'$ ทั้งช่วงนี้จะเป็นส่วนที่ ε ทำให้เครือข่ายไร้สายอยู่ในสถานะไม่ดี และเมื่อ ε มีค่าสูงกว่า ε' ก็จะทำให้อัตราการส่งข้อมูลไม่สำเร็จผ่านเครือข่ายไร้สายมีค่าสูงขึ้นด้วย ซึ่งจะทำให้สามารถหาความน่าจะเป็นของการส่งข้อมูลไม่สำเร็จหรือ P_u จากช่วงของ ε ที่อยู่ในสถานะไม่ดีได้ดังนี้

$$P_u = \frac{\varepsilon - \varepsilon'}{1 - \varepsilon'} \quad \text{เมื่อ } \varepsilon > \varepsilon' \quad (3.9)$$

ซึ่งเมื่อหาความน่าจะเป็นของการส่งข้อมูลไม่สำเร็จได้ ก็จะทำให้สามารถที่จะหาความน่าจะเป็นของการส่งข้อมูลสำเร็จในเครือข่ายไร้สายหรือ P_s ได้ดังนี้

$$P_s = 1 - P_u$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ $1 - \frac{\varepsilon - \varepsilon'}{1 - \varepsilon'}$ เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา $1 - \frac{\varepsilon - \varepsilon'}{1 - \varepsilon'}$ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

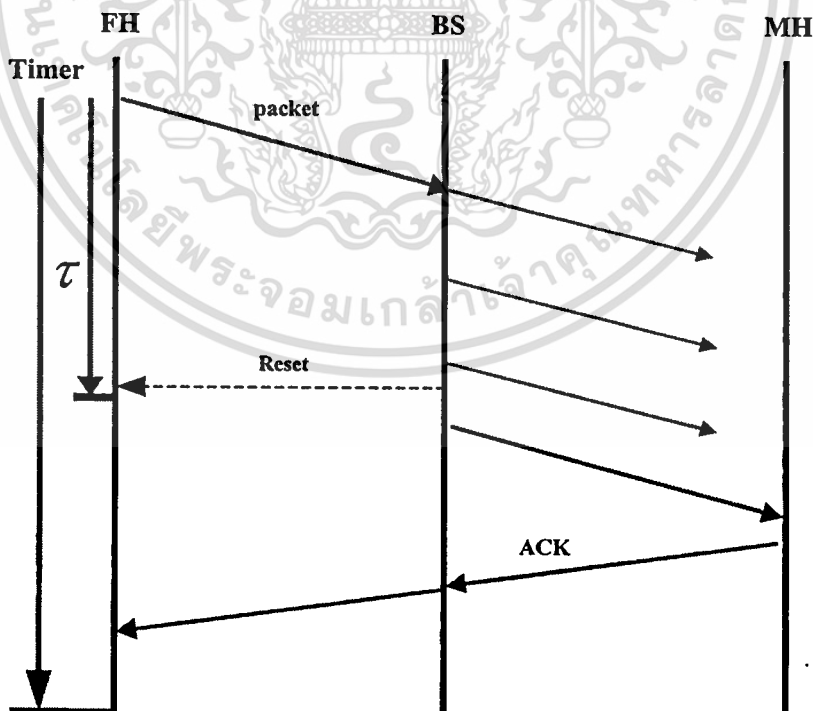
(3.10)

เมื่อ $\epsilon > \epsilon'$

3.3 วิธีการรีเซตไทม์เอาท์

การส่งข้อมูลระหว่างเครือข่ายมีสายและไร้สายโดยใช้ TCP จะมีการเกิดการส่งข้อมูลที่ซ้ำซ้อนเมื่อสถานการณ์ใช้เวลาในการส่งข้อมูลไปที่โฮสต์เคลื่อนที่มากกว่าค่า TCP ไทม์เอาท์ ซึ่งทำให้ TCP ต้นทางส่งซ้ำข้อมูลออกมา จากการพิจารณาแบบจำลองที่ได้กล่าวแล้วในข้างต้นจะพบว่า ถ้า $T > \tau$ หรือ $\epsilon > \epsilon'$ นั่นคือเครือข่ายไร้สายอยู่ในสถานะไม่ดี ซึ่งก็就会导致เกิดการส่งซ้ำข้อมูลจาก TCP ต้นทาง จากจุดนี้จะพบว่าถ้ารู้สถานะความผิดพลาดปัจจุบันในเครือข่ายไร้สายก็สามารถที่จะรู้สถานะของเครือข่ายปัจจุบันได้ว่าเป็นเช่นไร

ในงานวิจัยนี้จึงได้ทำการเสนอการแก้ปัญหาการเกิดการส่งซ้ำข้อมูล โดยวิธีการยืดเวลา TCP ไทม์เอาท์หรือการรีเซตไทม์เอาท์ ซึ่งไม่ต้องมีการดัดแปลง TCP ที่มีอยู่ในปัจจุบัน และไม่เป็นการขัดต่อหลักการควบคุมจากปลายถึงปลายของ TCP โดยหลักการคือจะทำการรีเซตค่า TCP ไทม์เอาท์เมื่อตรวจพบว่าเครือข่ายไร้สายมีค่า $\epsilon > \epsilon'$ ดังที่ได้กล่าวมาแล้วในข้างต้น นั่นคือเครือข่ายไร้สายอยู่ในสถานะไม่ดีนั่นเอง โดยหลักการทำงานแสดงดังภาพที่ 3.4



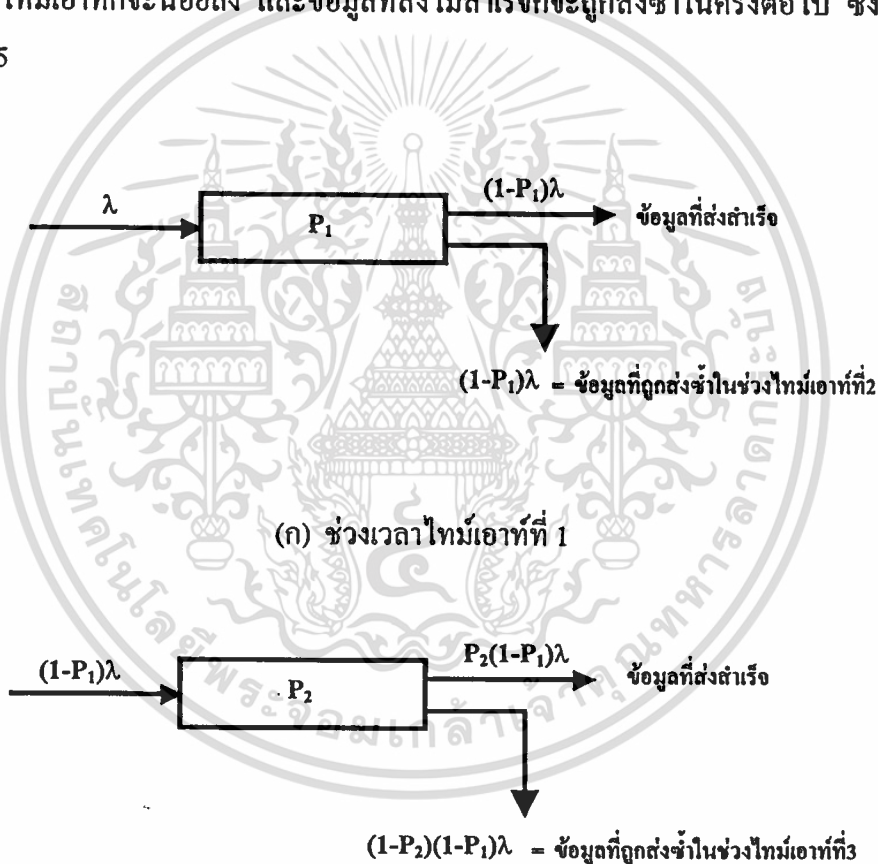
ภาพที่ 3.4 การทำงานของวิธีการรีเซตไทม์เอาท์

จากภาพที่ 3.4 จะพบว่า TCP ปกติเมื่อเครือข่ายไร้สายอยู่ในสถานะไม่ดีคือไม่สามารถส่งข้อมูลจากสถานีฐานไปโฮสต์เคลื่อนที่ได้ภายในเวลา TCP ไทม์เอาท์ TCP ต้นทางก็จะทำการส่งซ้ำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในงานวิจัยเท่านั้น เมื่อผู้ใดเห็นประโยชน์ในเอกสารนี้ กรุณาอย่าเผยแพร่โดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่เกิดไทม์เอาท์ ซึ่งอาจทำให้เกิดการส่งข้อมูลที่ซ้ำซ้อนในขณะที่สถานีฐานกำลังทำงานอยู่ แต่วิธีการที่ได้นำเสนอจะพบว่าเมื่อใกล้จะถึงช่วงเวลาไทม์เอาท์และรู้ว่าเครือข่ายอยู่ในสถานะไม่ดี คือ $\epsilon > \epsilon'$ ซึ่งไม่สามารถส่งข้อมูลจากสถานีฐานไปยังโฮสต์เคลื่อนที่ภายในเวลา TCP ไทม์เอาท์ได้ ก็จะส่งสัญญาณไปเพื่อทำการรีเซตค่า TCP ไทม์เอาท์หรือเริ่มจับเวลาใหม่นั้นเอง ซึ่งจะทำให้โอกาสในการส่งข้อมูลระหว่างสถานีฐานไปยังโฮสต์เคลื่อนที่มีอัตราความสำเร็จเพิ่มขึ้น รวมทั้งเป็นการลดการส่งข้อมูลที่ซ้ำซ้อนกันระหว่างสถานีฐานและ TCP ดันทาง ซึ่งจะทำให้เกิดการส่งข้อมูลที่ซ้ำซ้อนจากดันทางไปปลายทางลดลง

เมื่อทำการพิจารณาการทำงานการส่งข้อมูลของเครือข่ายไร้สายในสถานะไม่ดีจะพบว่า การส่งข้อมูลระหว่างสถานีฐานไปที่โฮสต์เคลื่อนที่ผ่านเครือข่ายไร้สายก็ต้องใช้เวลานานขึ้น โอกาสสำเร็จภายในเวลาไทม์เอาท์ก็จะน้อยลง และข้อมูลที่ส่งไม่สำเร็จก็จะถูกส่งซ้ำในครั้งต่อไป ซึ่งแสดงได้ดังภาพที่ 3.5



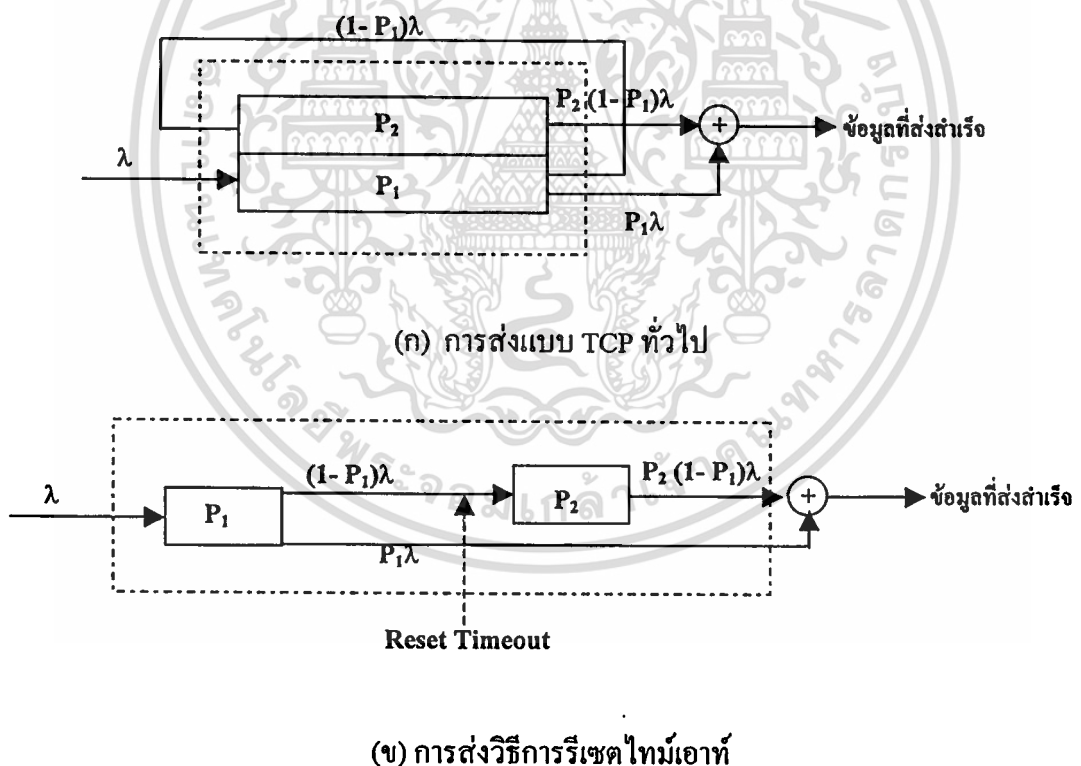
ภาพที่ 3.5 การส่งซ้ำข้อมูลเมื่อเครือข่ายไร้สายอยู่ในสถานะไม่ดี

จากภาพที่ 3.5 (ก) P_1 คือค่าของความสำเร็จจะเป็นในการส่งข้อมูลสำเร็จในช่วงเวลาไทม์เอาท์ที่ 1 หรือไทม์เอาท์แรก จะพบว่าเมื่อข้อมูลถูกส่งผ่านเข้าไปในเครือข่ายไร้สายก็จะมีข้อมูลส่วนหนึ่งที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งไปถึงปลายทางได้ภายในเวลาที่กำหนด ส่วนข้อมูลที่เหลือก็จะส่งไม่สำเร็จคือใช้เวลาในการส่งมากกว่าไทม์เอาต์ ซึ่งในระบบการทำงานของ TCP ก็จะทำการส่งซ้ำข้อมูลที่ไม่สำเร็จเหล่านี้ อีก ครั้ง ในช่วงเวลาไทม์เอาต์ที่ 2 ดังแสดงในภาพที่ 3.5 (ข) โดย P_2 คือค่าของความน่าจะเป็นในการส่งข้อมูลสำเร็จในช่วงเวลาไทม์เอาต์ที่ 2 และบางส่วนของข้อมูลที่ส่งไม่สำเร็จก็จะถูกส่งซ้ำในช่วงเวลาไทม์เอาต์ต่อไป ซึ่งการทำงานจะเป็นเช่นนี้ไปเรื่อย ๆ จนกว่าข้อมูลจะถูกส่งถึงปลายทางจนหมดหรือ TCP ยกเลิกการสื่อสาร

ในส่วนนี้จะลองทำการพิจารณาเปรียบเทียบระหว่างวิธีการ TCP ทั่วไปกับวิธีการที่นำเสนอในช่วง 2 ไทม์เอาต์ โดยใช้ค่าความน่าจะเป็นสร้างเป็นแบบจำลองในการทำงานของทั้ง 2 แบบ และพิจารณาที่ 2 ไทม์เอาต์ เพื่อดูว่าผลที่ได้จะแตกต่างกันเช่นไร โดยกำหนดให้ P_1 และ P_2 เป็นค่าของความน่าจะเป็นของการส่งข้อมูลสำเร็จในเครือข่ายไร้สายหรือ P_s ในช่วงไทม์เอาต์ที่ 1 และ 2 ตามลำดับ ซึ่งก็จะทำให้ได้แบบจำลองความน่าจะเป็นของเครือข่ายไร้สายในการส่งแบบ TCP ทั่วไป และส่งโดยวิธีการรีเซตไทม์เอาต์แสดงดังภาพที่ 3.6



ภาพที่ 3.6 แบบจำลองความน่าจะเป็นของเครือข่ายไร้สายในสถานะไม่ตี

ภาพที่ 3.6 (ก) แสดงแบบจำลองของวิธีการส่งแบบ TCP ทั่วไป ซึ่งจะพบว่าจะมีลักษณะการทำงานเหมือนกับในภาพที่ 3.5 คือ เมื่อข้อมูลจากต้นทางถูกส่งเข้าไปในระบบเครือข่ายไร้สายเมื่อเครือข่ายไร้สายอยู่ในสถานะไม่ดี ในช่วงเวลาไทม์เอาต์แรกก็จะมีข้อมูลส่วนหนึ่งที่ส่งข้อมูลสำเร็จไปถึงปลายทางได้ภายในเวลาที่กำหนด ส่วนข้อมูลที่เหลือก็จะถูกส่งเข้ามาจากต้นทางหรือโฮสต์เคลื่อนที่อีกครั้งในช่วงไทม์เอาต์ที่ 2 ซึ่งสามารถหาสมการ ข้อมูลที่ถูกส่งมาจากต้นทางและข้อมูลไปถึงปลายทางโดยวิธีการ TCP ทั่วไปในช่วงเวลา 2 ไทม์เอาต์ได้ดังนี้

$$G = \lambda + (1 - P_1)\lambda \quad (3.11)$$

$$S = P_1\lambda + P_2(1 - P_1)\lambda \quad (3.12)$$

$$\begin{aligned} \text{goodput} &= \frac{S}{G} \\ &= \frac{P_1\lambda + P_2(1 - P_1)\lambda}{\lambda + (1 - P_1)\lambda} \end{aligned} \quad (3.13)$$

เมื่อ

G คือ ข้อมูลทั้งหมดที่ถูกส่งจาก TCP ต้นทาง ไปยังเครือข่ายไร้สายรวมทั้งข้อมูลที่ถูกส่งซ้ำ

S คือ ข้อมูลทั้งหมดที่ไปถึงปลายทางในเครือข่ายไร้สายได้สำเร็จ ไม่รวมข้อมูลที่ซ้ำซ้อน (Throughput)

goodput คือ อัตราส่วนระหว่างข้อมูลไปถึงปลายทางเทียบกับข้อมูลที่ถูกส่งมาจากต้นทาง โดยปกติจะมีค่าน้อยกว่าหรือเท่ากับ 1

ภาพที่ 3.6 (ข) แสดงแบบจำลองของวิธีการส่งแบบรีเซตไทม์เอาต์ ซึ่งเมื่อเริ่มการทำงานในช่วงเวลาไทม์เอาต์แรกก็จะมีการทำงานเหมือนกับวิธีการ TCP ทั่วไปคือ จะมีข้อมูลส่วนหนึ่งที่สามารถส่งสำเร็จได้ภายในเวลาไทม์เอาต์ และข้อมูลอีกส่วนหนึ่งซึ่งไม่สามารถส่งสำเร็จได้ภายในเวลาไทม์เอาต์แรก แต่การทำงานในช่วงเวลาไทม์เอาต์ที่ 2 จะแตกต่างกันเนื่องจากวิธีการ TCP ทั่วไปจะส่งข้อมูลจาก TCP ต้นทางเมื่อข้อมูลไม่สามารถส่งสำเร็จได้ภายในเวลาไทม์เอาต์ แต่จากภาพที่ 3.6 (ข) จะเห็นว่าวิธีการที่นำเสนอ ในช่วงเวลาไทม์เอาต์ที่ 2 ข้อมูลที่ไม่สามารถส่งสำเร็จได้ภายในเวลาไทม์เอาต์ที่ 1 จะไม่ถูกส่งมาจาก TCP ต้นทาง เนื่องจากได้มีการรีเซตค่าไทม์เอาต์ของ TCP ต้นทาง จึงทำให้ข้อมูลที่ส่งไม่สำเร็จในช่วงไทม์เอาต์แรกทำการส่งจากสถานีฐานต่อไป ซึ่งจะเห็นได้ว่าวิธีการนี้จะทำให้การส่งข้อมูลที่ซ้ำซ้อนระหว่าง TCP ต้นทางและสถานีฐานลดลง ซึ่งต่างกับวิธีการ TCP ปกติ ซึ่งในช่วงเวลาไทม์เอาต์ที่ 2 จะมีการส่งซ้ำข้อมูลที่ไมสำเร็จมาจากต้นทาง แม้ว่าสถานีฐานเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำการส่งซ้ำข้อมูลก็ตาม ซึ่งข้อมูลที่ถูกส่งมาจากต้นทาง และข้อมูล que ไปถึงปลายทาง โดยวิธีการรีเซต
 ใหม่อะไรสามารถหาได้โดย สมการดังต่อไปนี้

$$G = \lambda \quad (3.14)$$

$$S = P_1\lambda + P_2(1 - P_1)\lambda \quad (3.15)$$

$$goodput = \frac{P_1\lambda + P_2(1 - P_1)\lambda}{\lambda} \quad (3.16)$$

เมื่อ $P_1, P_2 \in P,$

จะเห็นว่าทั้ง 2 วิธีการเมื่อพิจารณาจากสมการทางความน่าจะเป็นจะมีข้อมูลที่มาถึงปลายทาง
 หรือ S เท่ากัน แต่จะมีจำนวนข้อมูลที่ถูกส่งมาจากต้นทางหรือ G ต่างกัน ซึ่งก็เนื่องมาจากวิธีการส่ง
 ข้อมูลแบบ TCP ทั่วไปทุกครั้งที่มีการเกิดใหม่เอ่อข้อมูลจะถูกส่งมาจาก TCP ต้นทางเสมอ แต่ใน
 วิธีการรีเซตใหม่เอ่อข้อมูลจะถูกส่งมาจากสถานีฐานในขณะที่มีการเกิดใหม่เอ่อ ซึ่งเมื่อพิจารณาที่
 ข้อมูลต้นทางที่ถูกส่งมายังปลายทาง จะเห็นได้อย่างชัดเจนว่าวิธีการรีเซตใหม่เอ่อจะมีการส่งข้อมูลมา
 จากต้นทางน้อยกว่า แต่มีข้อมูล que ถึงปลายทางเท่ากัน

สมการที่ (3.11) และ (3.12) เมื่อทำการพิจารณาเป็นจำนวน n ใหม่เอ่อจะได้

$$G = \left[1 + \sum_{i=2}^n \prod_{j=2}^i (1 - P_{j-1}) \right] \lambda \quad (3.17)$$

$$S = \left[P_1 + \sum_{i=2}^n P_i \prod_{j=2}^i (1 - P_{j-1}) \right] \lambda \quad (3.18)$$

เมื่อ $i = 2 \dots n$; n คือ จำนวนใหม่เอ่อที่ทำการพิจารณา

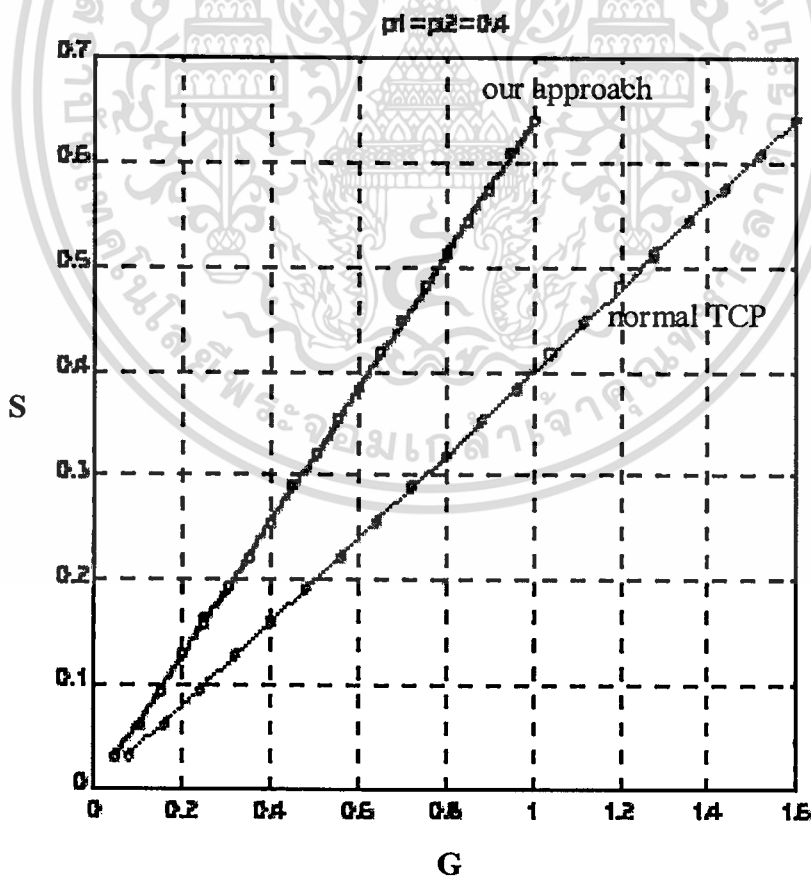
3.4 ผลของการวิเคราะห์ทางคณิตศาสตร์

ในหัวข้อที่ผ่านมาได้นำเสนอแบบจำลองทางความน่าจะเป็นระหว่างวิธีการส่งข้อมูลแบบ TCP ทั่วไป และวิธีการรีเซตไทม์เอาต์ทำให้ได้สมการในการหาค่าข้อมูลที่ถูกลบออกจากต้นทางและปลายทางของทั้ง 2 ระบบ ในส่วนนี้จะได้นำสมการในแบบจำลองความน่าจะเป็นในการส่งข้อมูลทั้ง 2 วิธี มาแสดงค่าเพื่อดูผลในทางคณิตศาสตร์ว่ามีพฤติกรรมเช่นไร โดยใช้การกำหนดค่าจากตัวอย่างในหัวข้อ 3.2 ซึ่งกำหนดให้ $\epsilon' = 0.5$ และ $\epsilon = 0.8$ เมื่อแทนค่าให้สมการ (3.10) จะได้ P_s ดังนี้

$$P_s = \frac{1-0.8}{1-0.5}$$

$$= 0.4$$

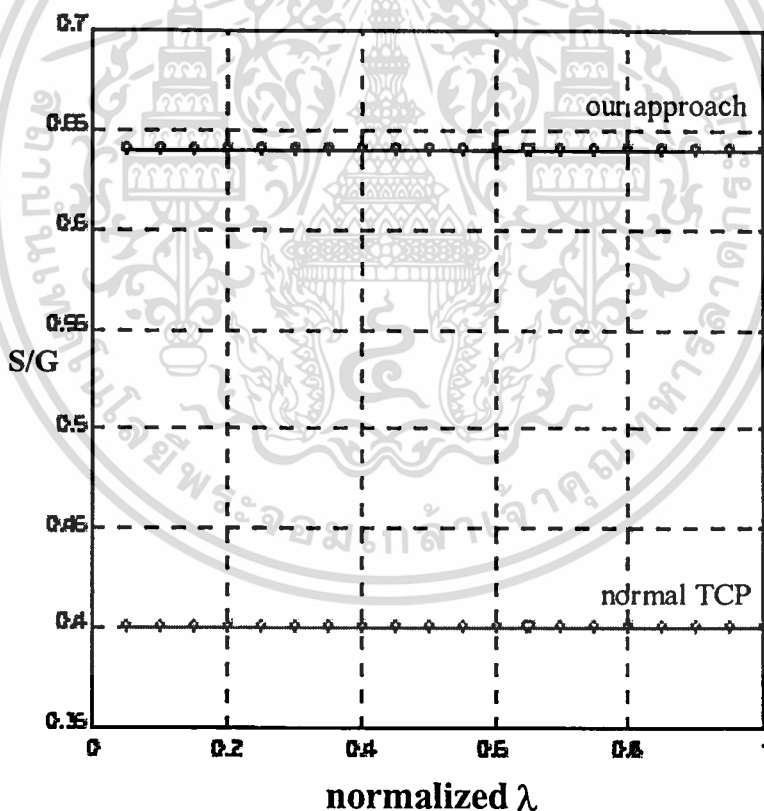
โดยกำหนดให้ P_1 หรือค่าความน่าจะเป็นในการส่งข้อมูลสำเร็จในช่วงเวลาไทม์เอาต์แรก และ P_2 หรือค่าความน่าจะเป็นในการส่งข้อมูลสำเร็จในช่วงเวลาไทม์เอาต์ที่สองมีค่าเท่ากัน คือเท่ากับ P_s เมื่อนำค่าที่ได้มาแทนค่าสมการเพื่อทำการวิเคราะห์ทางคณิตศาสตร์ จะแสดงได้ดังภาพที่ 3.7 และ 3.8



ภาพที่ 3.7 S และ G ของวิธีการ TCP ทั่วไปและรีเซตไทม์เอาต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

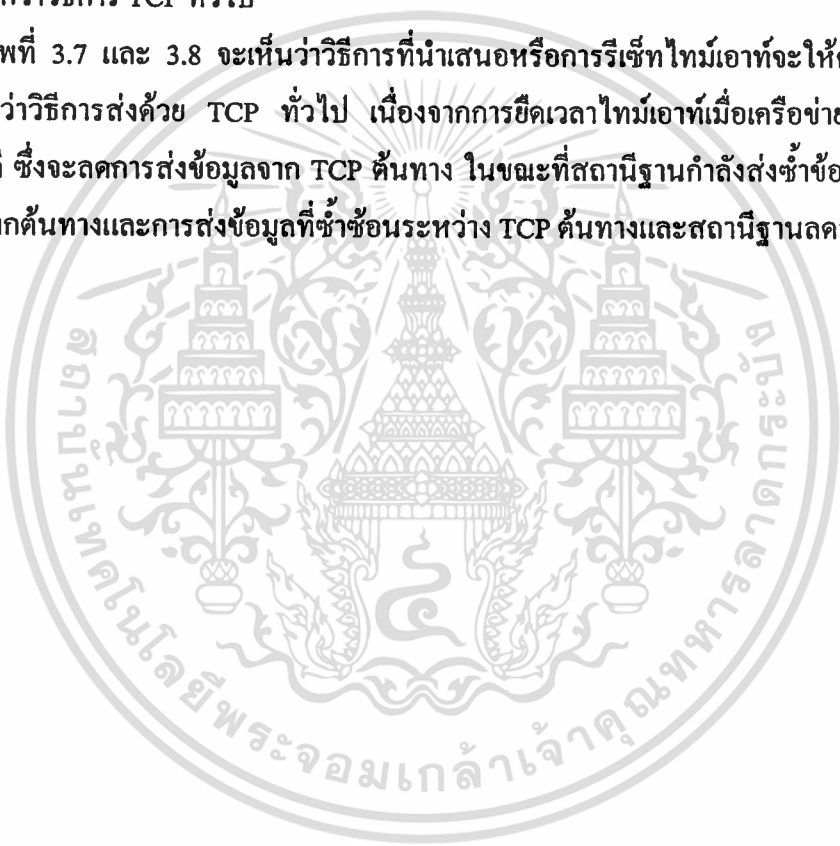
ภาพที่ 3.7 แกนตั้ง คือ S หรือจำนวนข้อมูลที่ไปถึงปลายทางได้สำเร็จบนเครือข่ายไร้สายแกน นอนคือ G หรือจำนวนข้อมูลที่ส่งออกมาจากต้นทาง จากผลการวิเคราะห์ทางคณิตศาสตร์จะพบว่า เมื่อเทียบที่จำนวนข้อมูลที่ส่งออกมาจากต้นทางเท่ากัน วิธีการรีเซตไทม์เอาต์ที่นำเสนอจะให้ข้อมูลที่ถึงปลายทางมากกว่าวิธี TCP ทั่วไป หรือให้ทรูพุดที่สูงกว่า ที่ผลเป็นเช่นนี้ก็เนื่องจากวิธีการ TCP ทั่วไป เมื่อข้อมูลไม่สามารถส่งได้สำเร็จภายในเวลาไทม์เอาต์ข้อมูลจะถูกส่งซ้ำจากต้นทางไปยังโฮสต์เคลื่อนที่ทันที แต่วิธีการรีเซตไทม์เอาต์จะพบว่าเมื่อข้อมูลไม่สามารถส่งสำเร็จได้ภายในเวลาไทม์เอาต์ ข้อมูลจะยังไม่ถูกส่งซ้ำจาก TCP ต้นทาง แต่จะเข้ามาจากสถานีฐานแทน ทำให้ลดการส่งข้อมูลจาก TCP ต้นทาง ดังนั้นเมื่อเทียบปริมาณข้อมูลที่มาถึงปลายทางเท่ากัน จึงทำให้วิธีการรีเซตไทม์เอาต์มีค่าทรูพุด (Throughput) ที่สูงกว่า เมื่อมีการส่งข้อมูลมาจากต้นทางเท่ากัน เมื่อเครือข่ายไร้สายอยู่ในสภาวะไม่ดี



ภาพที่ 3.8 ทรูพุดของวิธีการ TCP ทั่วไปและรีเซตไทม์เอาต์

ภาพที่ 3.8 แกนตั้ง คือ กุดพุด (Goodput) ส่วนแกนนอน คือข้อมูลที่ส่งมาจากต้นทางที่ถูกลดโมดไลซ์ (normalized λ) ซึ่งจะพบว่าวิธีการที่นำเสนอจะให้อัตราส่วนระหว่างข้อมูลที่ส่งสำเร็จต่อข้อมูลที่ถูกลดส่งออกจากต้นทางสูงกว่าวิธีการ TCP ทั่วไป ทั้งนี้ก็เนื่องมาจากเหตุผลที่ได้กล่าวไปแล้วข้างต้นคือ เมื่อเครือข่ายไร้สายอยู่ในสถานะไม่ดี ซึ่งจะทำให้เกิดการส่งซ้ำข้อมูลจาก TCP ต้นทางมายังปลายทางในวิธีการ TCP ทั่วไป แต่วิธีการรีเซตไทม์เอาต์สถานะพื้นฐานจะทำการรีเซตค่าไทม์เอาต์ที่ TCP ต้นทางเมื่อใกล้จะถึงเวลาไทม์เอาต์และทำการส่งซ้ำข้อมูลจากสถานะพื้นฐานต่อไป ทำให้ไม่ต้องส่งข้อมูลมาจากต้นทาง ซึ่งจะเห็นว่าถึงแม้จะมีข้อมูลมาถึงปลายทางเท่ากัน แต่จำนวนข้อมูลที่ถูกลดส่งมาจากต้นทางต่างกันคือวิธีการรีเซตไทม์เอาต์จะลดการส่งข้อมูลจากต้นทางลง จึงทำให้ได้ค่ากุดพุดที่สูงกว่าวิธีการ TCP ทั่วไป

จากภาพที่ 3.7 และ 3.8 จะเห็นว่าวิธีการที่นำเสนอหรือการรีเซตไทม์เอาต์จะให้ค่าทฤษฎีและกุดพุดสูงกว่าวิธีการส่งด้วย TCP ทั่วไป เนื่องจากการยึดเวลาไทม์เอาต์เมื่อเครือข่ายไร้สายอยู่ในสถานะไม่ดี ซึ่งจะลดการส่งข้อมูลจาก TCP ต้นทาง ในขณะที่สถานะพื้นฐานกำลังส่งซ้ำข้อมูลทำให้การส่งข้อมูลจากต้นทางและการส่งข้อมูลที่ซ้ำซ้อนระหว่าง TCP ต้นทางและสถานะพื้นฐานลดลง



บทที่ 4

การจำลองการทำงาน

ในบทที่ 3 ได้กล่าวถึงการสร้างแบบจำลองทางคณิตศาสตร์ของสถานีฐานโดยใช้ทฤษฎีการเข้าคิวแบบ M/M/1 ซึ่งในบทนี้จะทำการจำลองการทำงาน (Simulation) ของการเชื่อมต่อระหว่างเครือข่ายมีสายและไร้สายโดยใช้ TCP รวมทั้งจำลองการทำงานของสถานีฐานตามแบบจำลองที่ได้นำเสนอ เพื่อทำการสร้างเครื่องมือที่จะใช้ในการศึกษาถึงพฤติกรรมของการส่งข้อมูลผ่านเครือข่ายไร้สายโดยวิธีการ TCP ทั่วไป และวิธีการรีเซตค่า TCP ซึ่งผลของการทำจำลองการทำงานจะได้นำเสนอในบทถัดไป และโปรแกรมจำลองการทำงานจะใช้โปรแกรมการจำลองเครือข่ายที่ชื่อว่า OPNET

4.1 ฟังก์ชันการจำลองของโปรแกรม

ในส่วนนี้จะกล่าวถึงการทำงานโดยรวมของ โปรแกรมการจำลองการทำงานการเชื่อมต่อระหว่างเครือข่ายมีสายและไร้สายโดยวิธีการ TCP ทั่วไป และวิธีการรีเซตค่าของ TCP ซึ่งทั้ง 2 วิธีการนี้สามารถจำลองได้โดยใช้ฟังก์ชันเดียวกัน โดยวิธีการรีเซตค่าได้ทำการออกแบบโปรแกรมเพื่อศึกษาถึงผลของการรีเซตค่าได้ตั้งแต่ 2 ไซม์เฮนส์ถึง 7 ไซม์เฮนส์ ซึ่งสามารถแสดงการทำงานของโปรแกรมได้ดังกระบวนการต่อไปนี้

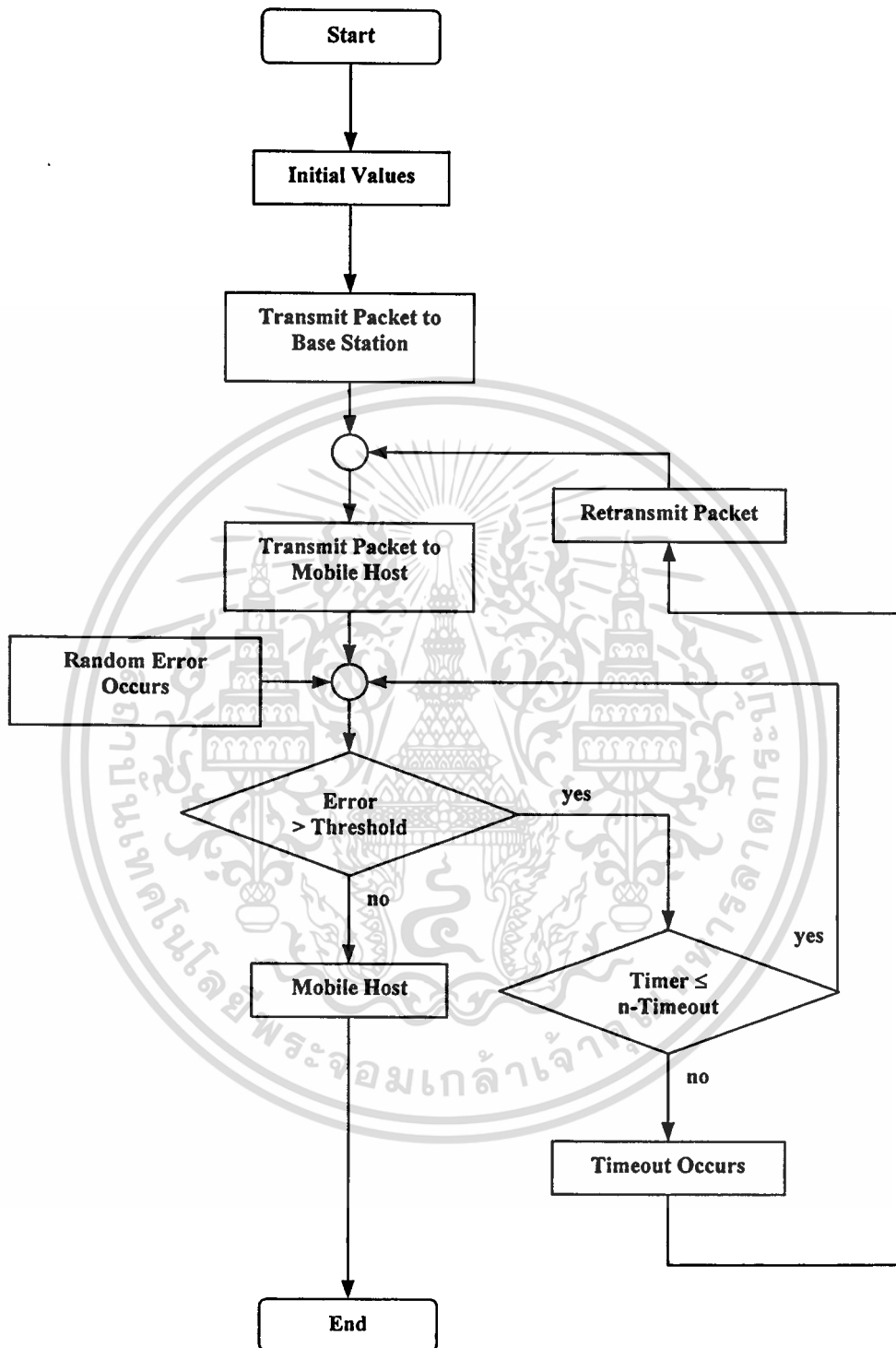
จากกระบวนการในส่วนเริ่มต้นของโปรแกรมคือการตั้งค่าต่าง ๆ ก่อนที่โปรแกรมจะเริ่มทำงาน ซึ่งประกอบไปด้วยขนาดของแพ็คเก็ต ความเร็วในการส่งข้อมูลหรือ λ โดยมีการแจกแจงเป็นแบบเอ็กซ์โพเนนเชียลหรือพัวซง ค่าความผิดพลาดวิกฤติของเครือข่ายหรือ ϵ' ซึ่งสามารถแปลงเป็น P_s ได้ ความเร็วในการส่งข้อมูลของสถานีฐานหรือ μ ซึ่งมีเวลาในการส่งข้อมูลจากสถานีฐานมีรูปแบบการแจกแจงแบบเอ็กซ์โพเนนเชียล ซึ่งเป็นการปรับค่าให้ตรงกับรูปแบบของระบบการเข้าคิวแบบ M/M/1 ตามแบบจำลองดังที่ได้นำเสนอในบทที่ผ่านมารวมทั้งการปรับค่าจำนวนครั้งในการรีเซตค่า

ในส่วนถัดมาก็จะทำการสร้างแพ็คเก็ตให้มีขนาดตามที่กำหนด และทำการส่งไปสถานีฐานตามความเร็วที่กำหนดซึ่งก็คือขั้นตอนการส่งข้อมูลจากสถานีฐานหรือ TCP ต้นทางไปสถานีฐานนั่นเอง เมื่อส่งแพ็คเก็ตหรือข้อมูลไปที่สถานีฐานแล้ว สถานีฐานจะรับข้อมูลเข้ามาเก็บไว้ที่บัฟเฟอร์หรือคิวแล้วทำการคำนวณเวลาที่ใช้ในการส่งตามความเร็วในการส่งข้อมูลที่ได้ทำการตั้งไว้ในตอนเริ่มโปรแกรมหรือก็คือเวลาที่ใช้ในการส่งข้อมูลจากสถานีฐานไปยังโฮสต์เคลื่อนที่ โดยผ่านเครือข่ายไร้สาย ซึ่งก็คือ X นั่นเอง จากนั้นจะทำการตรวจสอบความผิดพลาดของเครือข่ายไร้สายหรือ ϵ ว่ามีค่าสูงเกินกว่า ϵ' หรือไม่ ซึ่งค่าความผิดพลาดที่เกิดขึ้นในเครือข่ายจะใช้การสร้างโดยใช้การแจก

แจกแบบยูนิฟอร์ม (Uniform Distribution) ซึ่งได้ทำการปรับให้อยู่ในรูปแบบของ P_s ซึ่งจะพบว่า จากบทที่ 3 ที่ผ่านมา สามารถที่จะคำนวณจาก \mathcal{E} และ \mathcal{E}' ไปสู่ P_s ได้ เมื่อทำการตรวจสอบแล้วพบว่า $\mathcal{E} \leq \mathcal{E}'$ หรือค่าความผิดพลาดมีค่าน้อยกว่าค่าที่กำหนด ซึ่งก็คือเครือข่ายไร้สายอยู่ในสภาวะดี แสดงว่าข้อมูลสามารถส่งจากสถานีฐานผ่านเครือข่ายไร้สายไปถึง โฮสต์เคลื่อนที่ได้ภายในเวลาที่กำหนด ซึ่งโฮสต์เคลื่อนที่ก็จะทำการรับข้อมูลถือว่าเป็นการสิ้นสุดควมรอบในการจำลอง

จากที่กล่าวมาเบื้องต้น เป็นการทำงานเมื่อข้อมูลสามารถส่งสำเร็จได้ภายในเวลาไทม์เอาต์ แต่ในทางกลับกันถ้าข้อมูลไม่สามารถส่งสำเร็จได้ภายในเวลาไทม์เอาต์ ก็คือค่าความผิดพลาดในเครือข่ายไร้สายหรือ \mathcal{E} มีค่าสูงเกินกว่า \mathcal{E}' หรือค่าความผิดพลาดที่กำหนดไว้ก็คือเครือข่ายไร้สายจะอยู่ในสภาวะไม่ดี โปรแกรมจะทำการตรวจสอบว่ามีการตั้งการรีเซตไทม์เอาต์ไว้หรือไม่ ถ้ามีการตั้งการรีเซตไทม์เอาต์ไว้ ก็จะทำการตรวจสอบต่อไปว่าได้ทำรีเซตไปกี่ครั้ง ถึงจำนวนที่ตั้งไว้หรือไม่ ถ้ายังไม่ถึงจำนวนครั้งในการรีเซตไทม์เอาต์ที่ตั้งไว้ ก็จะทำการรีเซตค่าไทม์เอาต์แล้วทำการส่งข้อมูลจากสถานีฐานไปโฮสต์เคลื่อนที่อีกครั้ง แต่ถ้าจำนวนในการรีเซตไทม์เอาต์ถึงค่าที่กำหนดไว้ ข้อมูลก็จะถูกส่งเข้ามาจากต้นทางในขณะที่สถานีฐานที่กำลังพยายามจะส่งไปที่โฮสต์เคลื่อนที่เช่นกัน

จากที่กล่าวมาในเบื้องต้น โปรแกรมนี้สามารถใช้ได้ทั้งวิธีการ TCP ทั่วไป และการรีเซตไทม์เอาต์ เนื่องจากทั้ง 2 แบบสามารถใช้โปรแกรมร่วมกันในการจำลอง โดยใช้การปรับค่าจำนวนไทม์เอาต์ในการรีเซต ถ้าเป็น 1 ไทม์เอาต์หมายความว่าการทำงานในรูปแบบปกติหรือ TCP ทั่วไป แต่ถ้าทำการปรับค่าไทม์เอาต์ในการตรวจสอบให้มากกว่า 1 ก็จะเป็นวิธีการรีเซตไทม์เอาต์



ภาพที่ 4.1 กระบวนการของโปรแกรมการจำลองการส่งข้อมูลระหว่างเครือข่ายมีสายและไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

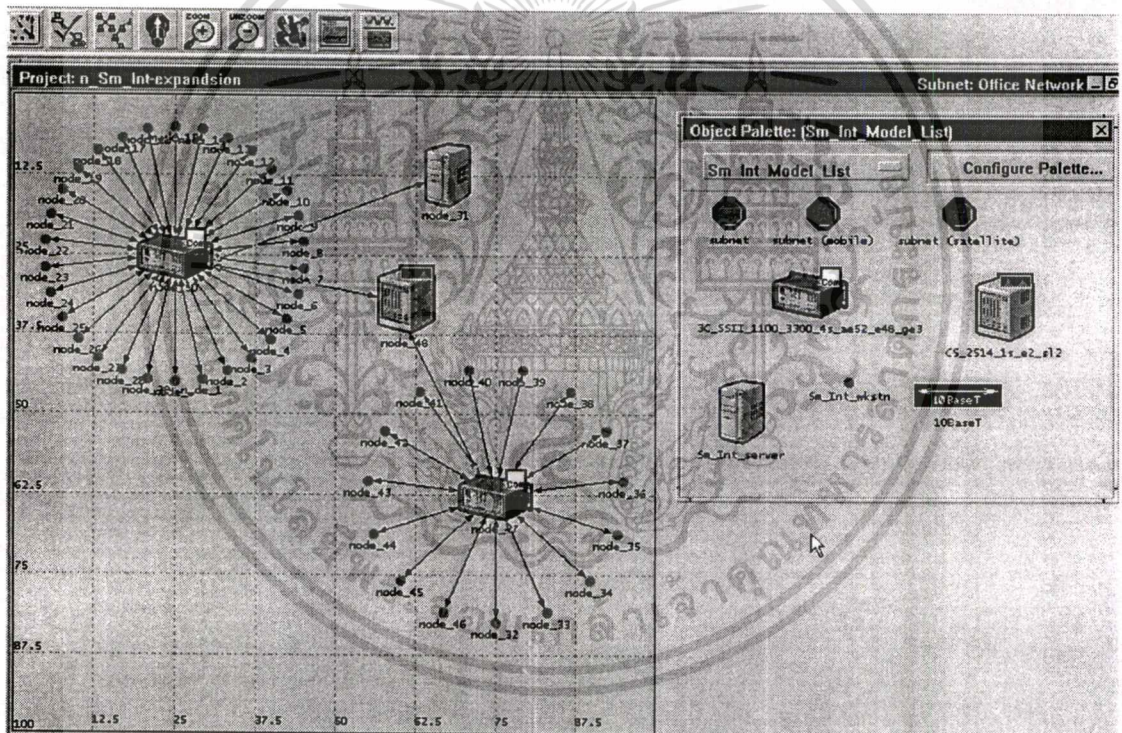
4.2 โปรแกรมจำลอง

ในการสร้างโปรแกรมในการจำลอง จะใช้โปรแกรมจำลองเครือข่ายที่ชื่อว่า OPNET

4.2.1 OPNET

OPNET เป็นโปรแกรมที่ใช้ในการจำลองเครือข่าย ซึ่งสามารถที่จะสร้างแบบจำลองเครือข่าย โดยใช้ออบเจกต์ที่มีอยู่ คัดแปลง หรือทำการสร้างขึ้นมาใหม่ก็ได้และทำการปรับแต่งค่า เก็บค่าทางสถิติของแต่ละส่วนของออบเจกต์หรือทั้งระบบที่ทำการเลียนแบบรวมทั้งสามารถที่จะนำข้อมูลที่เก็บไว้มาวิเคราะห์ ซึ่งประกอบด้วยส่วนต่างๆ ดังนี้

4.2.1.1 โปรเจกต์เอดิเตอร์ (Project Editor)



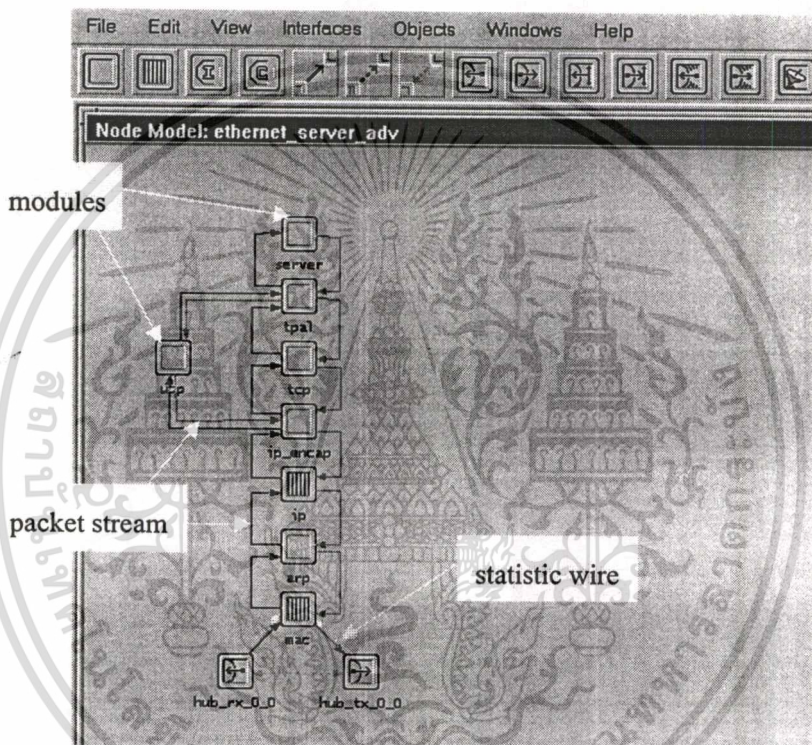
ภาพที่ 4.2 โปรเจกต์เอดิเตอร์

ภาพที่ 4.2 เป็นพื้นที่หลักในการจำลองและสร้างเครือข่าย ซึ่งสามารถที่จะใช้วิธีการลากและวางออบเจกต์ (Object) มาจากออบเจกต์พาเลต (Object Palette) แล้วนำมาวางในพื้นที่งานและปรับแต่งค่าต่างๆ รวมทั้งเริ่มทำการจำลองเพื่อแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.2 โหนดเอดิเตอร์ (Node Editor)

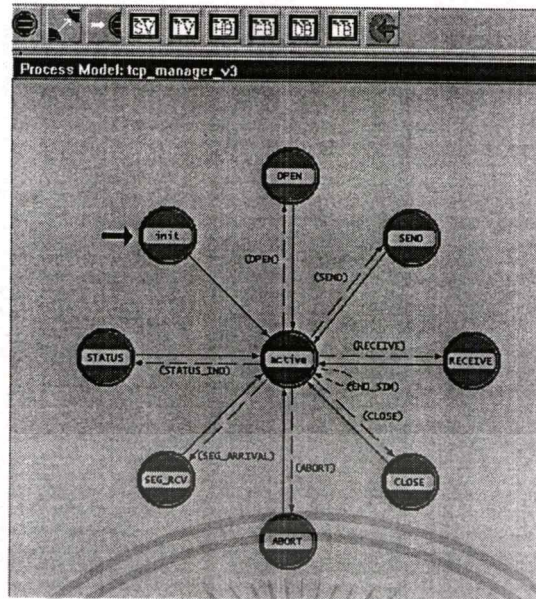
ภายใต้แต่ละออบเจกต์หรือโหนดของ OPNET เมื่อดับเบิลคลิกจะพบโมดูลหลายๆ โมดูลที่ประกอบขึ้นมาเป็นออบเจกต์ดังภาพที่ 4.3 ซึ่งแต่ละโมดูลภายใต้ออบเจกต์จะเป็นตัวกำหนดพฤติกรรมหรือการทำงานของออบเจกต์นั้นๆ ซึ่งในการเชื่อมต่อของแต่ละโมดูลจะมีการเชื่อมต่ออยู่ 2 แบบคือการเชื่อมต่อทางสถิติซึ่งจะใช้ในการส่งข้อมูลทางสถิติระหว่างโมดูลและการเชื่อมต่อทางแพ็คเกจซึ่งจะใช้ในการส่งแพ็คเกจหรือข้อมูลเหมือนกันการส่งข้อมูลในเครือข่าย



ภาพที่ 4.3 โหนดเอดิเตอร์

4.2.1.3 โพรเซสโมเดลเอดิเตอร์ (Process Model Editor)

ภายใต้ของแต่ละโมดูล เมื่อดับเบิลคลิกก็จะพบกับโพรเซสโมเดลดังภาพที่ 4.4 ซึ่งจะทำหน้าที่ควบคุมฟังก์ชันต่างๆ ของโหนดโมดูล ซึ่งจะใช้หลักการของไฟไนต์สเตตแมชชีนในการทำงาน โดยไอคอนจะแทนแต่ละสถานะ และเส้นจากการเปลี่ยนสถานะระหว่างสถานะซึ่งในการทำงานของแต่ละสถานะจะเขียนโดยใช้โปรแกรมภาษา C หรือ C++



(ก) โปรเซสโมเดลเอ็ดิเตอร์

```

init : Enter Exces
/* Initialize the state variables and notification log handle
tcp_mgr_sv_init ();
tcp_notification_log_init ();

/* If not already done, define the tcp2 event pooled memory
if (ev_defined == OPC_FALSE)
{
    ev_defined = OPC_TRUE;
    ev_pmh = op_prg_pmo_define ("tcp2 event", sizeof (TcpT_
}

ev_ptr = (TcpT_Event*) op_prg_pmo_alloc (ev_pmh);
/* Without memory to hold the event, this model will be di
if (ev_ptr == OPC_NIL)
{
    op_prg_log_entry_write (ll_loghndl,
        "Unable to create event structure.");
    op_sim_end ("Please check simulation log for simulation
  
```

(ข) ซอร์สโค้ด (Source Code) ภายใต้อิโพรเซส (Process)

ภาพที่ 4.4 โปรเซสโมเดลเอ็ดิเตอร์และซอร์สโค้ด

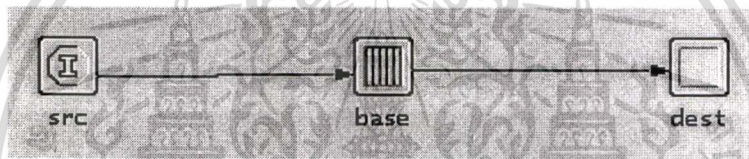
ในการเขียนโปรแกรมใน OPNET จะใช้การอินเทอร์พเพื่อเป็นจังหวะในการทำงานซึ่งจะมีคำสั่งหลายๆ คำสั่งซึ่งเป็นคำสั่งเฉพาะใน OPNET เพื่อทำให้เกิดการอินเทอร์พชนิดต่างๆ ซึ่งเมื่อทำการตรวจสอบชนิดของอินเทอร์พที่เกิดขึ้นก็จะทำให้ทราบว่าเป็นอินเทอร์พชนิดใด เช่น อินเทอร์พจากการส่งข้อมูลก็หมายความว่ามีการส่งข้อมูลมาที่โปรเซสนั้นถ้าในโปรเซสนั้นต้องการรับข้อมูลเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็จะต้องทำการเขียนโปรแกรมเพื่อทำการตรวจสอบการเกิดอินเทอร์เน็ตว่าเป็นชนิดใด โดยถ้าเป็นชนิดที่เกิดจากการส่งข้อมูลก็ให้ทำการรับข้อมูลเข้ามา ซึ่งในการเขียนโปรแกรมใน OPNET ในระดับที่ลึกซึ้งหรือในระดับโปรเซสโมเดลจำเป็นต้องทำความเข้าใจในส่วนนี้ให้ชัดเจน จึงจะสามารถเขียนโปรแกรมได้ถูกต้อง

ซึ่งในการสร้างโปรแกรมการจำลองในงานวิจัยนี้ จะทำการดัดแปลงและสร้างโปรเซสในการทำงานขึ้นมาใหม่ โดยใช้ OPNET

4.2.2 โปรแกรมจำลอง

จากการพิจารณากระบวนการของการเชื่อมต่อระหว่างเครือข่ายไร้สายและมีสายโดยใช้ TCP ดังที่ได้เสนอในส่วนที่ผ่านมา นั้น เมื่อนำมาสร้างเป็นแบบจำลองใน OPNET จะแสดงได้ดังภาพที่ 4.5



ภาพที่ 4.5 แบบจำลองของเครือข่ายใน OPNET

ประกอบด้วย 3 ส่วนคือ src, base และ dest ซึ่งเมื่อเทียบกับเครือข่ายจริงก็คือต้นทางสถานีฐาน และ โฮสต์เคลื่อนที่ซึ่งจะได้อธิบายการทำงานของแต่ละส่วนในส่วนถัดไป

4.2.2.1 src (Source)

src เป็นส่วนที่ใช้สร้างแพ็คเก็ตขึ้นมากับเครือข่าย ซึ่งก็เปรียบเสมือนโฮสต์อยู่ที่นั่นเอง ซึ่งต้องทำการปรับค่าเพื่อให้มีลักษณะการทำงานเหมือนกับภาวะที่กำหนดให้กับแบบจำลองทางคณิตศาสตร์ที่ได้กล่าวถึงในบทที่ผ่านมา ซึ่งค่าต่างๆ ที่ทำการปรับมีดังนี้

- interarrival pdf (Interarrival Probability Density Function) เป็นเป็นช่วงเวลาการมาถึงของแพ็คเก็ต ซึ่งในที่นี้จะทำการปรับค่าให้เป็นการแจกแจงแบบเอ็กซ์โพเนนเชียล หรือให้อัตราการมาถึง หรือ λ อยู่ในรูปแบบของตัวของนั่นเอง

- interarrival args เป็นการปรับค่าช่วงเวลาการมาถึงของแพ็คเก็ตเฉลี่ยหรือค่าเฉลี่ยของ λ

- pk size pdf (Packet Size Probability Density Function) เป็นการปรับค่าการแจกแจงขนาดของแพ็คเก็ตที่สร้างขึ้นมา โดยในที่นี้ทำการปรับค่าเป็นการแจกแจงแบบเอ็กซ์โพเนนเชียล

- pk size arg เป็นการปรับค่าขนาดของแพ็คเก็ตโดยเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

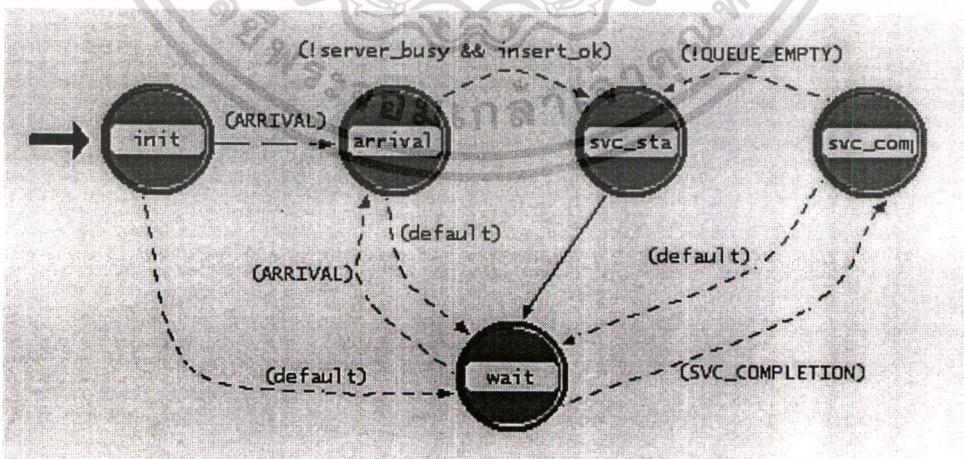
src จะทำหน้าที่สร้างแพ็คเก็ตขึ้นมา และส่งไปตู้เครือข่ายเพื่อไปยังส่วนถัดไปคือ base โดยจะทำการส่งตามค่าที่ได้ปรับแต่ง ในส่วนของรูปแบบแพ็คเก็ตนั้นจะมีเฮดเดอร์ที่ได้ตัวเลขของข้อมูล ซึ่งเป็นรูปแบบแพ็คเก็ตที่สร้างขึ้นมามาเพื่อใช้ในงานจำลองนี้ โดย src จะทำการส่งแพ็คเก็ตจนกว่าการจำลองจะสิ้นสุด

4.2.2.2 base

base ในที่นี้จะทำหน้าที่เหมือนสถานีฐานคือทำหน้าที่ส่งข้อมูลระหว่างต้นทางและปลายทาง ซึ่งที่ base นี้ จะทำงานเหมือนกับแบบจำลองของสถานีฐานที่ได้สร้างขึ้นมามาโดยใช้ทฤษฎีการเข้าคิวแบบ M/M/1 เพราะฉะนั้น base ก็คือระบบการเข้าคิวแบบ M/M/1 นั่นเอง คือมีบัฟเฟอร์ไม่จำกัด และมีแบบแผนการบริการแบบมาก่อนบริการก่อน ซึ่งค่าต่างๆ ที่ทำการปรับแต่งมีดังนี้

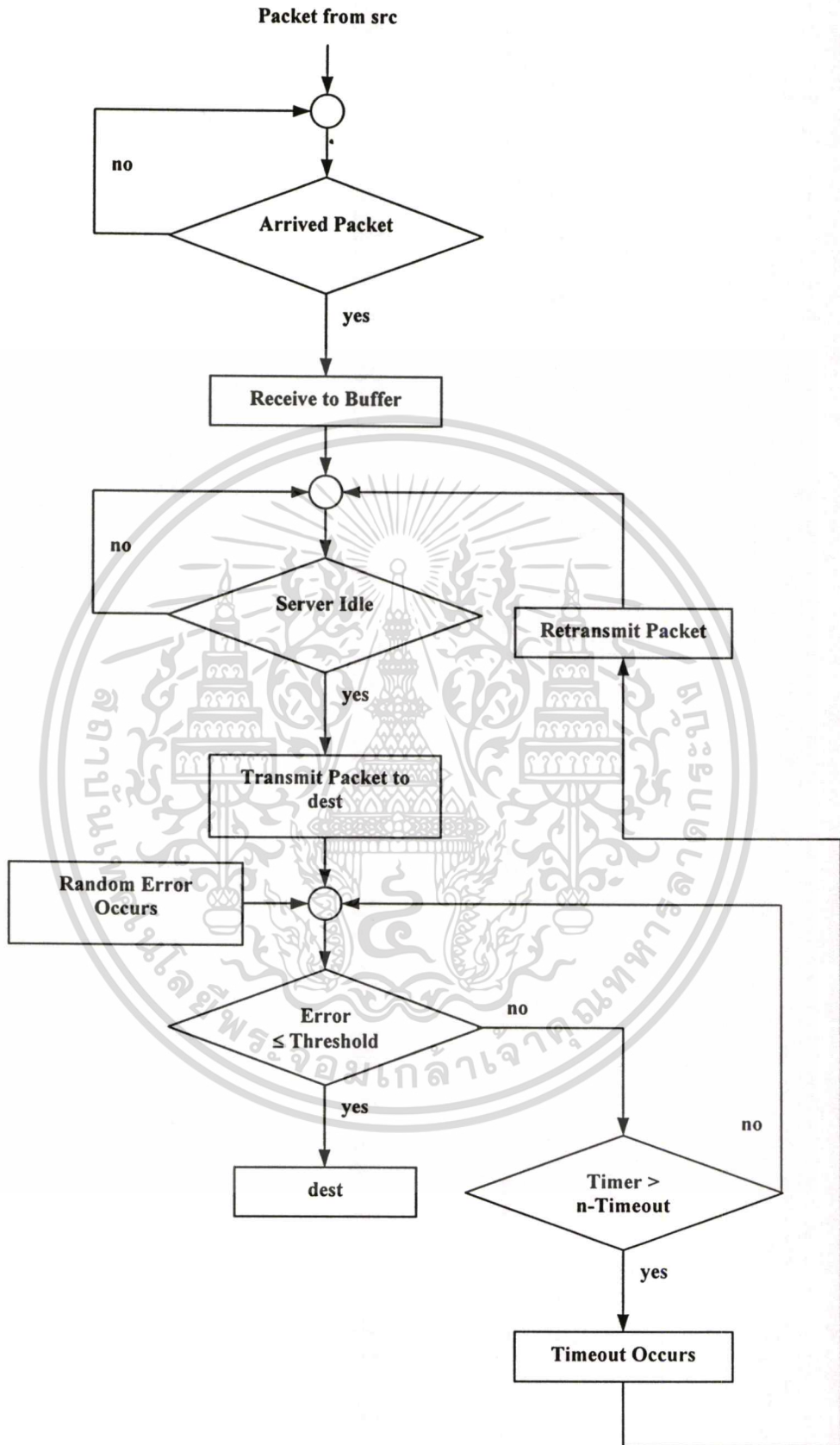
- process model จะถูกปรับแต่งเป็น acb-fifo ก็คือให้มีแบบแผนการเข้าคิวเป็นแบบมาก่อนถึงก่อนบริการก่อนนั่นเอง
- service rate คืออัตราในการบริการเฉลี่ยหรือ X ซึ่งก็คือความเร็วในการส่งข้อมูลจากสถานีฐานไปยังปลายทางนั่นเอง ซึ่งในที่นี้จะปรับไว้ที่ 1 Mb/S ซึ่งเป็นความเร็วขั้นต่ำตามมาตรฐาน IEEE 802.11

ภาพที่ 4.6 แสดงโปรเซสโมเดลของ base ที่สร้างขึ้นในโปรแกรม OPNET และกระบวนการของ base แสดงดังภาพที่ 4.7 ในการอธิบายการทำงานของ base จะอธิบายจากโปรเซสโมเดลเป็นหลัก โดยใช้กระบวนการประกอบในบางส่วน



ภาพที่ 4.6 โปรเซสโมเดลของ base

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.7 กระบวนการทำงานของ base

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรม OPNET จะใช้จังหวะการอินเทอร์พซึ่งเกิดจากเหตุการณ์ต่าง ๆ ในการจำลอง เพื่อใช้ตรวจสอบเงื่อนไขในการเปลี่ยนสถานะ จากภาพที่ 4.7 ขั้นตอนแรกคือ init ซึ่งก็คือเป็นการปรับค่าเริ่มต้นต่างๆ ของโปรแกรมที่ทำการปรับไว้นั่นเอง จากนั้นจะทำการตรวจสอบเงื่อนไขว่าจะทำอะไรต่อไป ซึ่งในที่นี้จะมี 2 เงื่อนไขคือ ARRIVAL กับ default ในความหมายคำว่า default ก็คือเงื่อนไขอื่นไม่เป็นจริงจะมาทำที่เงื่อนไขนี้ จะพบว่าถ้าเงื่อนไข ARRIVAL ไม่เป็นจริงการทำงานก็จะมารอที่ขั้นตอน WAIT เพื่อรอทำงานในจังหวะต่อไปที่มีอินเทอร์พเกิดขึ้น แต่ถ้าเงื่อนไข ARRIVAL เป็นจริงซึ่งก็คือมีแพ็คเกตเข้ามา ซึ่งก็ตรงกับกระบวนการงานในส่วนของตรวจสอบการมาถึงของแพ็คเกต ก็จะไปทำงานที่ขั้นตอน arrival ซึ่งขั้นตอน arrival จะมีหน้าที่นำแพ็คเกตเก็บเข้าบัฟเฟอร์ของคิว หลังจากนั้นจะทำการตรวจสอบว่าเซิร์ฟเวอร์ว่างหรือไม่ ถ้าว่างก็จะเปลี่ยนไปทำงานที่ขั้นตอน svc-start ถ้าไม่ว่างก็จะไปที่ wait ที่ขั้นตอน svc-start จะทำการคำนวณเวลาของการบริการหรือเวลาที่แพ็คเกตใช้งานอยู่บนเครือข่ายไร้สายนั่นเองรวมทั้งทำการปรับค่าความผิดพลาดที่เกิดขึ้นบนเครือข่ายไร้สายโดยใช้เป็นการสุ่มแบบยูนิฟอร์ม ซึ่งก็เปรียบกับกระบวนการงานในส่วนของส่งข้อมูลไปที่ dest และการเกิดความผิดพลาดจากนั้นก็เปลี่ยนไปที่สถานะ wait ทันทีเมื่อเงื่อนไข svc-complete เป็นจริง ซึ่งจะเกิดขึ้นเมื่อถึงเวลาที่ข้อมูลส่งเสร็จหรือไทม์เอาต์นั่นเอง โดยจะเกิดจากการทำงานของ svc-start ก็จะมีการเปลี่ยนไปทำงานที่ขั้นตอน svc-complete ซึ่งจะทำหน้าที่ตรวจสอบว่าค่าความผิดพลาดของเครือข่ายมีค่าสูงเกินกว่าค่าที่กำหนดหรือไม่ ถ้าไม่เกินก็จะทำการนำข้อมูลออกจากบัฟเฟอร์และส่งไปที่ปลายทาง แต่ถ้าค่าความผิดพลาดของเครือข่ายมีค่าเกินกว่าค่าที่กำหนดไว้ก็จะทำการตรวจสอบว่ามีค่าเกินจำนวนไทม์เอาต์ที่ทำการรีเซตไว้หรือไม่ ถ้าเกินก็จะสั่งให้ส่งข้อมูลมาใหม่จากต้นทางหรือให้เกิดการส่งซ้ำนั่นเอง แต่ถ้าไม่เกินก็จะทำการส่งใหม่โดยใช้ขั้นตอน svc-start ซึ่งจะเห็นว่ามีเงื่อนไขในการเปลี่ยนขั้นตอนการทำงานจาก svc-complete ไปที่ svc-start ก็คือเมื่อบัฟเฟอร์หรือคิวมีข้อมูลอยู่ภายใน แต่ถ้าบัฟเฟอร์ว่างก็จะไปอยู่ที่ขั้นตอน wait ซึ่งก็จะเทียบขั้นตอนของ svc-complete กับกระบวนการงานก็คือในส่วนของตรวจสอบความผิดพลาดและจำนวนการรีเซตไทม์เอาต์ ซึ่งในการทำงานของโปรแกรมนั้น จะทำการเป็นวนรอบไปเรื่อยๆ จนกว่าเวลาในการจำลองจะจบลงซึ่งตัวอย่างของโปรแกรมบางส่วนแสดงดังภาพที่ 4.8 และ 4.9

ภาพที่ 4.8 แสดงการเซตค่าตัวแปรบางส่วนที่ใช้ในการเขียนโปรแกรมแบบจำลอง ซึ่งเป็นตัวแปรชนิดที่ใช้ร่วมกันทุกโปรเซส ส่วนภาพที่ 4.9 เป็นโปรแกรมที่เขียนอยู่ภายใต้โปรเซส svc-start : ซึ่งแสดงเพียงบางส่วนเท่านั้น ซึ่งจะเห็นว่ามิลักษณะเหมือนกันภาษา C และ C++

state variables		
Type	Name	Comments
int	server_busy	
double	service_rate	
Objid	own_id	
int	pack_num	
Stathandle	pack_num_stathandle	
Distribution *	gen_error	
int	err_point	
int	check_st	
int	err_num	
Stathandle	error_stathandle	
Stathandle	chk_st_stathandle	
int	number_time_set	

ภาพที่ 4.8 state variable

```

svc_start: Enter Execs
/* get a handle on packet at head of subqueue 0 */
/* (this does not remove the packet) */
pkptr = op_subq_pk_access (0, OPC_QPOS_HEAD);

/* determine the packets length (in bits) */
pk_len = op_pk_total_size_get (pkptr);

/* determine the time required to complete */
/* service of the packet */
pk_svc_time = pk_len / service_rate;

/* schedule an interrupt for this process */
/* at the time where service ends. */
op_intrpt_schedule_self (op_sim_time () + pk_svc_time, 0);

/* the server is now busy. */
server_busy = 1;

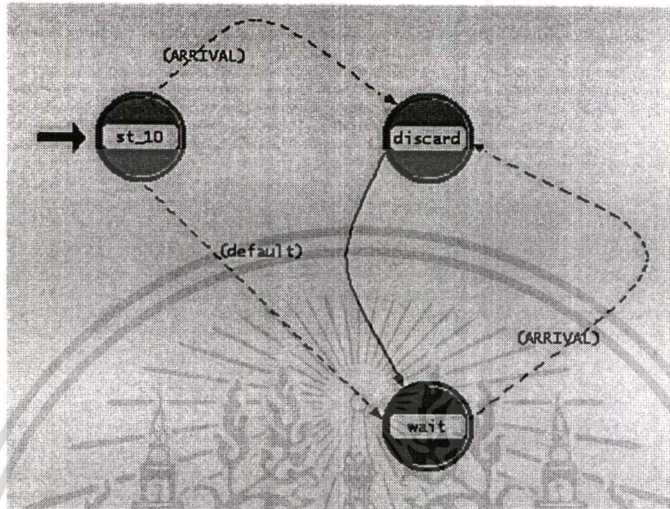
```

ภาพที่ 4.9 โปรแกรมภายใต้ svc-start

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.3 dest

dest ในที่นี้จะทำหน้าที่เป็นปลายทางหรือโหนดเคลื่อนที่นั่นเอง และจะทำการเก็บค่าต่างๆ ซึ่งภาพของโปรเซสโมเดล แสดงดังภาพที่ 4.10



ภาพที่ 4.10 โปรแกรมภายใต้ dest

ที่ขั้นตอน st_10 จะเป็นการปรับค่าเริ่มต้นเมื่อเริ่มโปรแกรม และตรวจสอบว่ามีแพ็คเกจส่งเข้ามาหรือไม่ ถ้ามีแพ็คเกจส่งเข้ามาเงื่อนไข ARRIVAL ก็จะเป็นจริง ก็จะเปลี่ยนสถานะไปที่ขั้นตอน discard แต่ถ้าไม่เป็นจริงก็จะไปที่ wait และรอทำการตรวจสอบว่ามีแพ็คเกจเข้ามาหรือไม่ต่อไป ที่ขั้นตอน discard จะเก็บค่าต่างๆ ในเครือข่าย รวมทั้งทำการกำจัดแพ็คเกจทิ้ง เนื่องจากถ้าไม่กำจัดจะเป็นการเปลืองหน่วยความจำของโปรแกรม OPNET

บทที่ 5

ผลการจำลอง

ในบทนี้จะเป็นกล่าวถึงผลของโปรแกรมจำลองการเชื่อมต่อระหว่างเครือข่ายไร้สายและมีสายตามแบบจำลองที่ได้สร้างขึ้นในบทที่ 3 และทำการเปรียบเทียบการส่งข้อมูลระหว่างวิธีการ TCP ทั่วไป และวิธีการรีเซตไทม์เอาต์ ดังที่ได้กล่าวในบทที่ผ่านมา โดยจะมีการปรับค่าและตัวแปรที่ใช้ในการจำลองดังนี้

- ขนาดของแพ็คเกจ จะมีขนาดคงที่ คือ 1460 ไบต์ หรือ 11,680 บิต
- ความเร็วในการส่งข้อมูลของสถานีฐาน 1 เมกกะบิตต่อวินาที
- ความเร็วในการส่งข้อมูลจากโฮสต์อยู่กับที่ จะใช้การปรับช่วงเวลาระหว่างแพ็คเกจ ซึ่งจะใช้ในการปรับค่า ρ เช่นกัน
- P_s คือค่าความน่าจะเป็นในการส่งสำเร็จ หรือ ค่าความผิดพลาดในเครือข่ายไร้สาย
- nt คือจำนวนครั้งในการรีเซตไทม์เอาต์ เช่น $2t$ หมายถึงสามารถรีเซตได้ 1 ครั้ง ถ้า $1t$ หมายถึงวิธีการ TCP ปกติ

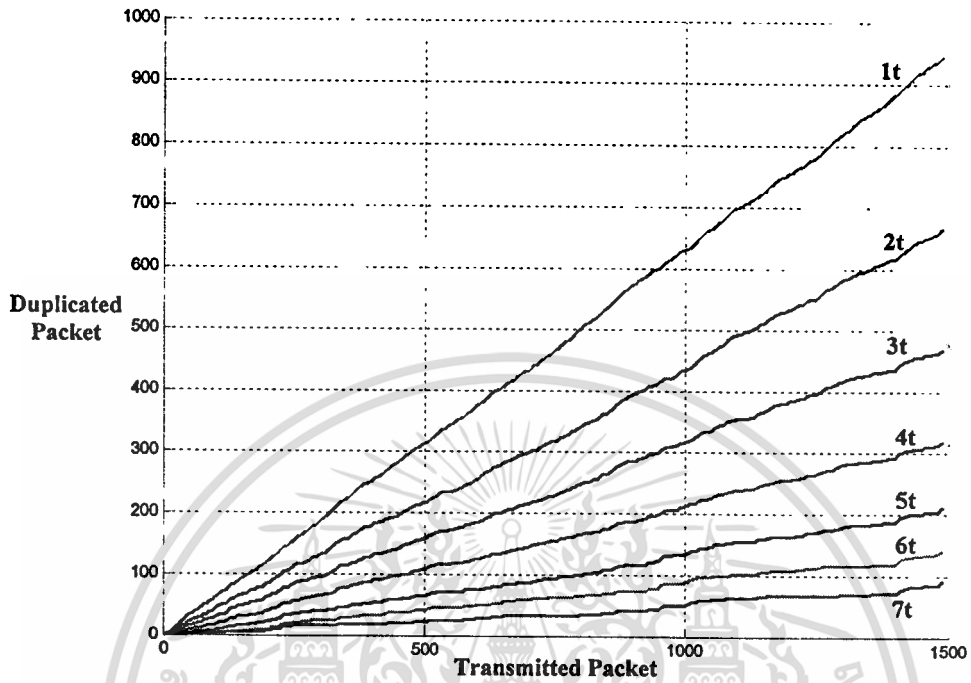
ในการจำลองแต่ละครั้ง จะใช้เวลาการจำลอง 60 วินาที ซึ่งผลการจำลองแสดงได้ดังนี้

5.1 ผลการจำลอง

ภาพที่ 5.1 – ภาพที่ 5.8 จะแสดงการทดลองเพื่อดูผลของการทำงานแบบ TCP ทั่วไป และวิธีการรีเซตไทม์เอาต์ที่ค่าต่างๆ โดยในที่นี้จะทำการรีเซตเป็นจำนวน 2 ไทม์เอาต์ - 7 ไทม์เอาต์ต่อแต่ละสถานการณ์ เพื่อดูผลของข้อมูลช้าช้าขึ้นที่เกิดขึ้น ซึ่งการปรับค่าต่างๆแสดงดัง ตารางที่ 5.1

ตารางที่ 5.1 ค่าของการจำลอง ภาพที่ 5.1 – ภาพที่ 5.3 และภาพที่ 5.5 – ภาพที่ 5.7

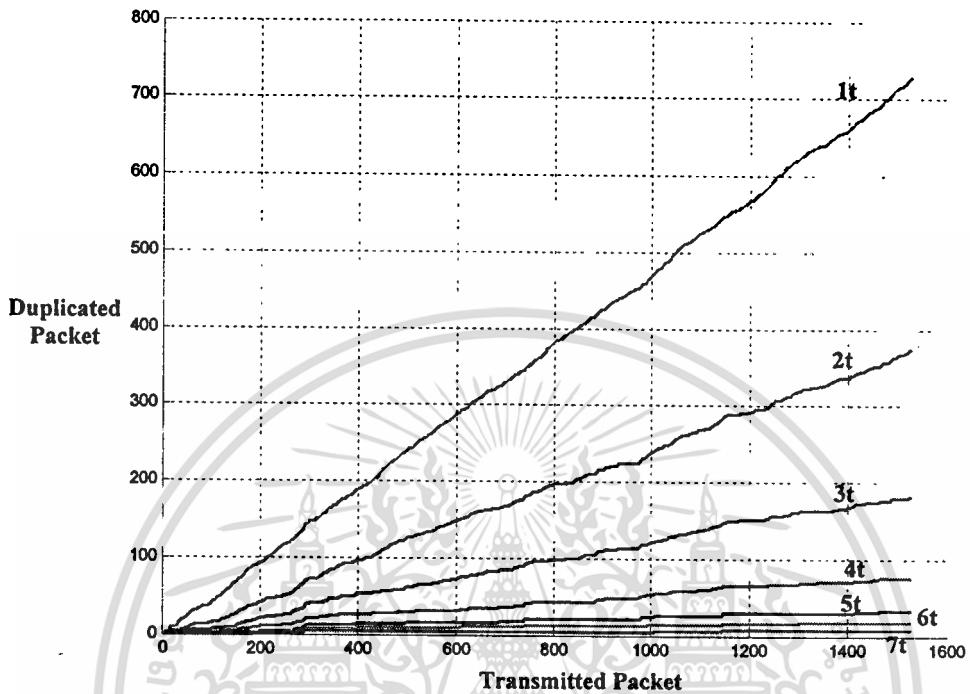
ρ	P_s
0.3	0.3
0.3	0.5
0.3	0.7
0.9	0.3
0.9	0.5
0.9	0.7



ภาพที่ 5.1 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.3$

ตารางที่ 5.2 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.3$

การรีเซตไทม์เอาท์	จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อน
1t	945
2t	667
3t	470
4t	317
5t	209
6t	140
7t	90

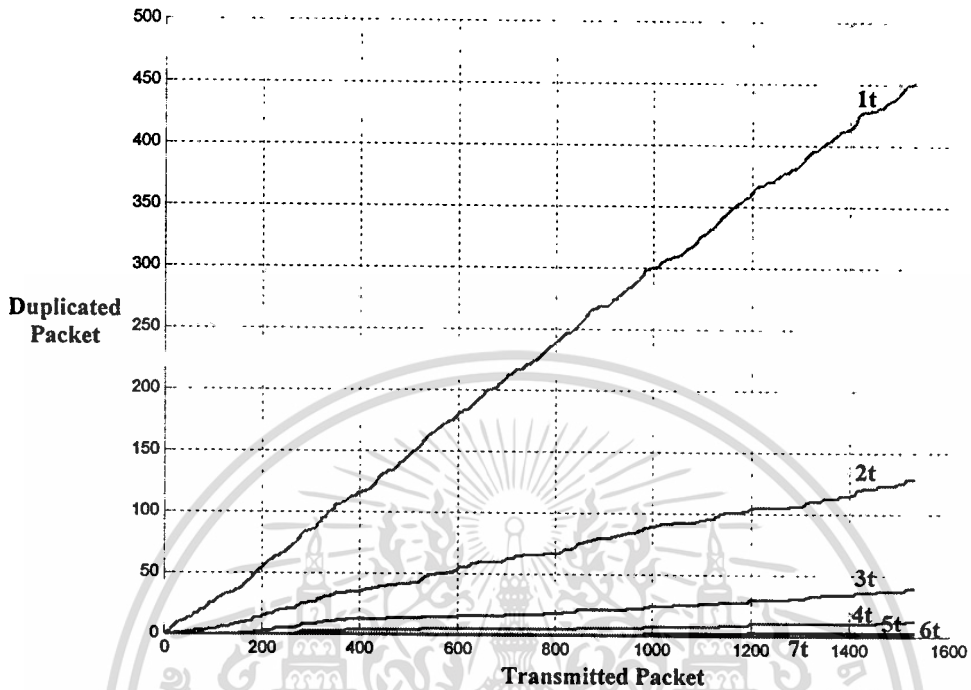


ภาพที่ 5.2 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.5$ และ $\rho = 0.3$

ตารางที่ 5.3 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.5$ และ $\rho = 0.3$

การรีเซต ไทม์เอาท์	จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อน
1t	729
2t	373
3t	181
4t	76
5t	33
6t	18
7t	8

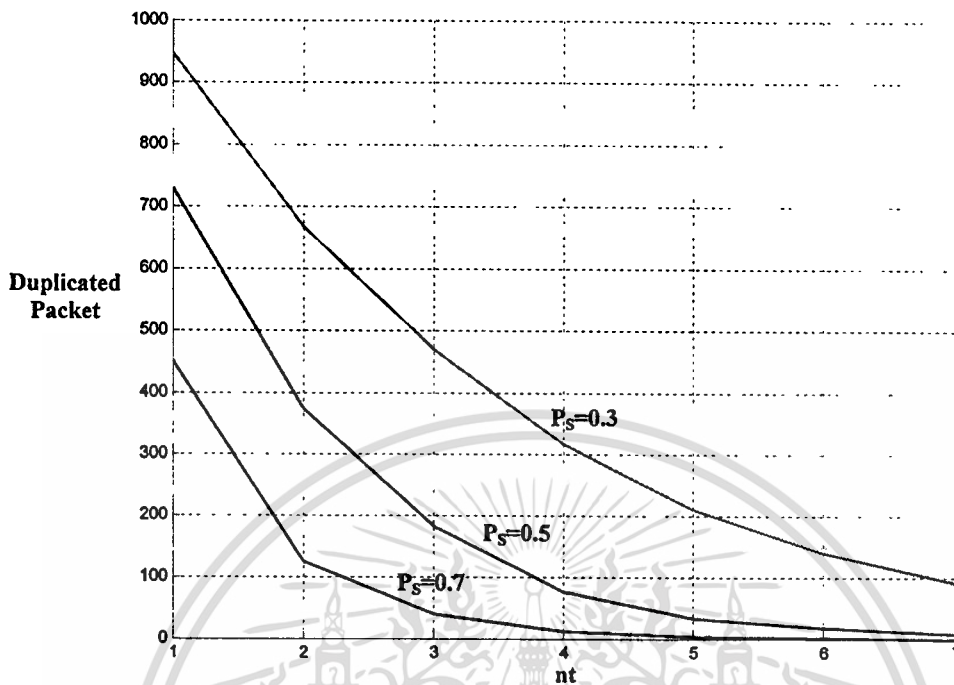
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.3 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.7$ และ $\rho = 0.3$

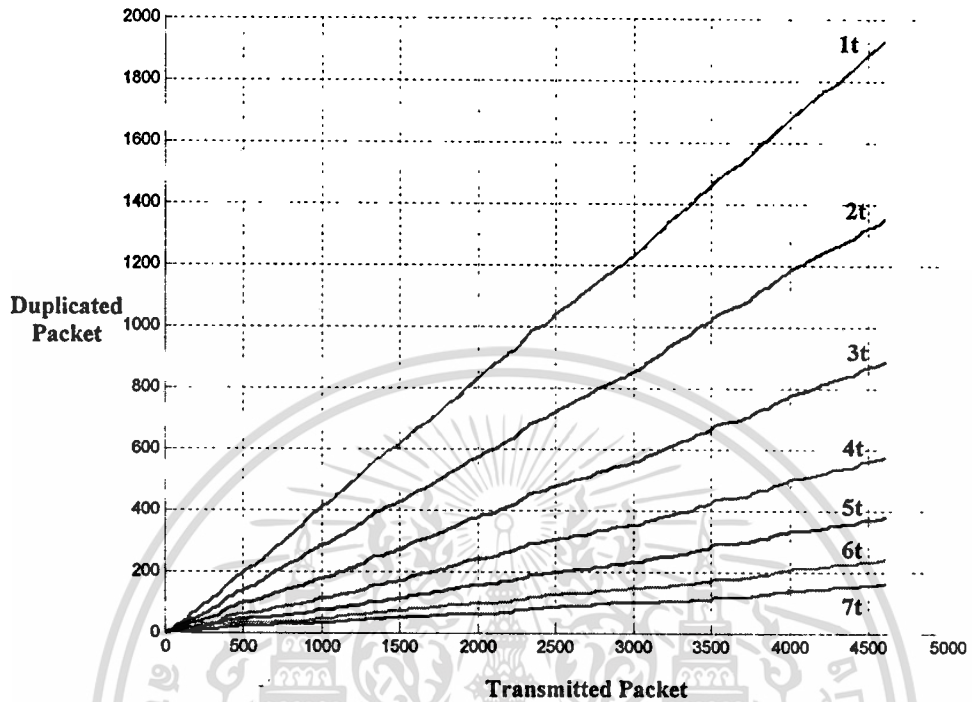
ตารางที่ 5.4 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.7$ และ $\rho = 0.3$

การรีเซตไทม์เอาท์	จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อน
1t	451
2t	125
3t	39
4t	12
5t	3
6t	1
7t	0



ภาพที่ 5.4 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3, 0.5$ และ 0.7 ที่ $p = 0.3$

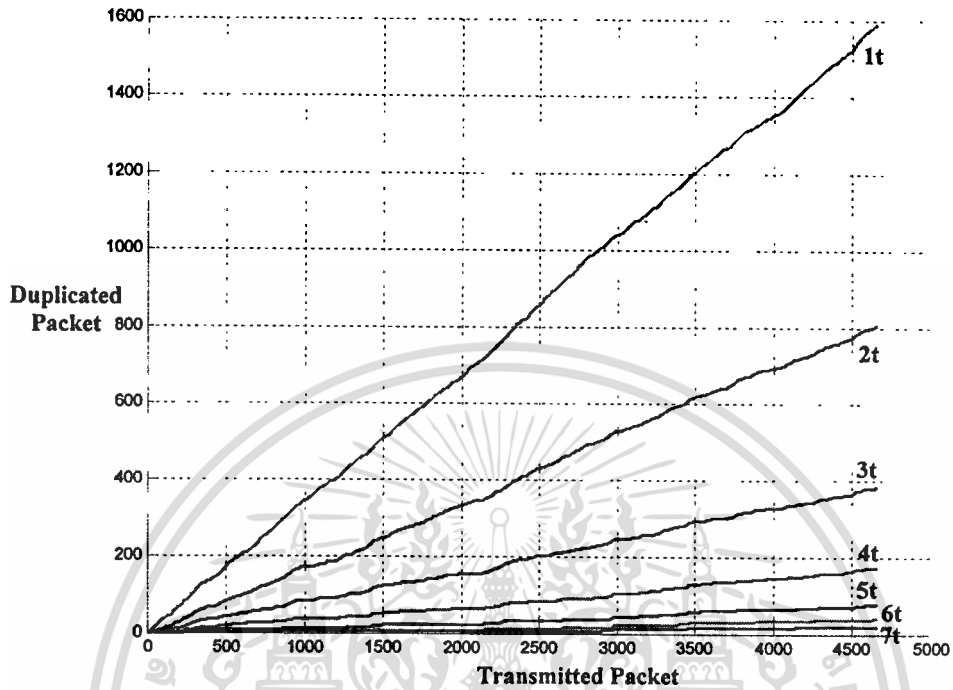
ภาพที่ 5.1 – ภาพที่ 5.3 เป็นการแสดงให้เห็นถึงจำนวนข้อมูลที่ถูกส่งซ้ำมาจากต้นทาง เมื่อมีการเปลี่ยนแปลงสถานะของความผิดพลาดในเครือข่ายซึ่งก็คือค่า P_s โดยมีค่า 0.3 0.5 และ 0.7 ตามลำดับ โดยมีค่า p คงที่เท่ากับ 0.3 ซึ่งผลที่ได้จากการจำลองแสดงให้เห็นว่า เมื่อทำการเพิ่มจำนวนการรีเซตไทม์เอาต์จะทำให้การส่งข้อมูลที่ซ้ำซ้อนลดลงตามลำดับ และจะเห็นได้อย่างชัดเจนเมื่อเทียบกับวิธีการ TCP ทั่วไป โดยตารางที่ 5.1- ตารางที่ 5.3 จะแสดงค่าของข้อมูลที่ถูกส่งซ้ำซ้อน เมื่อเวลาในการจำลองผ่านไป 60 วินาทีที่เวลาในการรีเซตไทม์เอาต์ค่าต่างๆของแต่ละกรณี เมื่อนำค่าจากตารางที่ 5.1 – ตารางที่ 5.3 มาแสดงค่าในภาพที่ 5.4 ในการเปรียบเทียบจำนวนข้อมูลที่ถูกส่งซ้ำมาจากต้นทางกับการรีเซตไทม์เอาต์ค่าต่างๆ จะแสดงได้อย่างชัดเจนว่า ทุกกรณี วิธีการรีเซตไทม์เอาต์จะให้ผลลัพธ์ที่ดีกว่าคือมีจำนวนข้อมูลที่ถูกส่งซ้ำมาจากต้นทางน้อยกว่า และเมื่อเปรียบเทียบที่ค่า P_s จะเห็นว่าค่า P_s มีค่าสูงจะเกิดข้อมูลที่ส่งซ้ำมาจากต้นทางน้อยกว่าค่า P_s ที่มีค่าต่ำ ซึ่งก็เป็นไปตามทฤษฎี



ภาพที่ 5.5 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.9$

ตารางที่ 5.5 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.9$

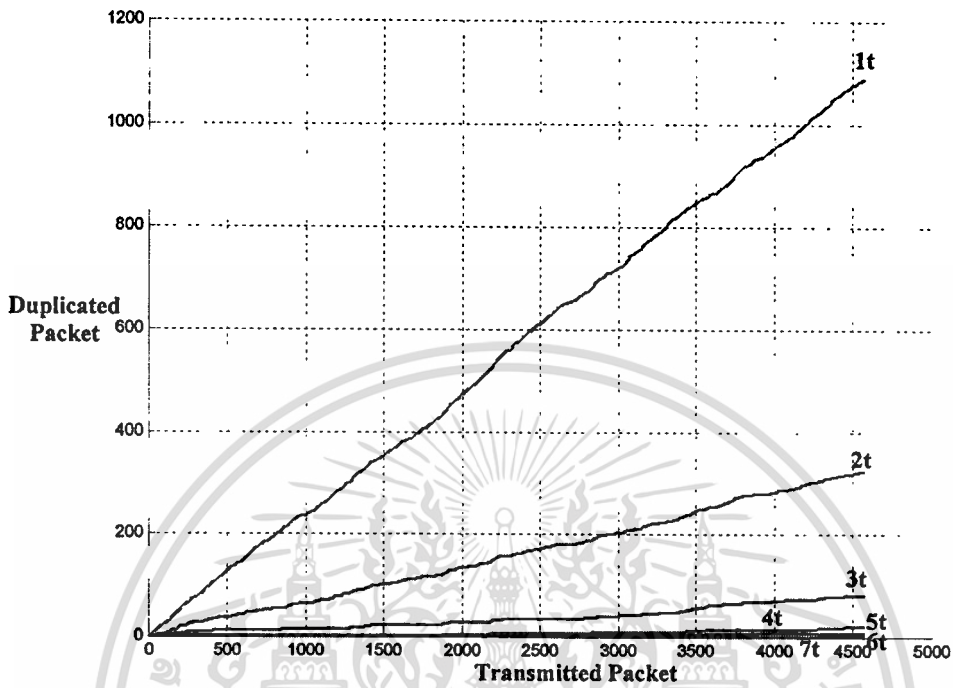
การรีเซตไทม์เอาท์	จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อน
1t	1930
2t	1353
3t	888
4t	577
5t	381
6t	243
7t	163



ภาพที่ 5.6 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.5$ และ $\rho = 0.9$

ตารางที่ 5.6 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.5$ และ $\rho = 0.9$

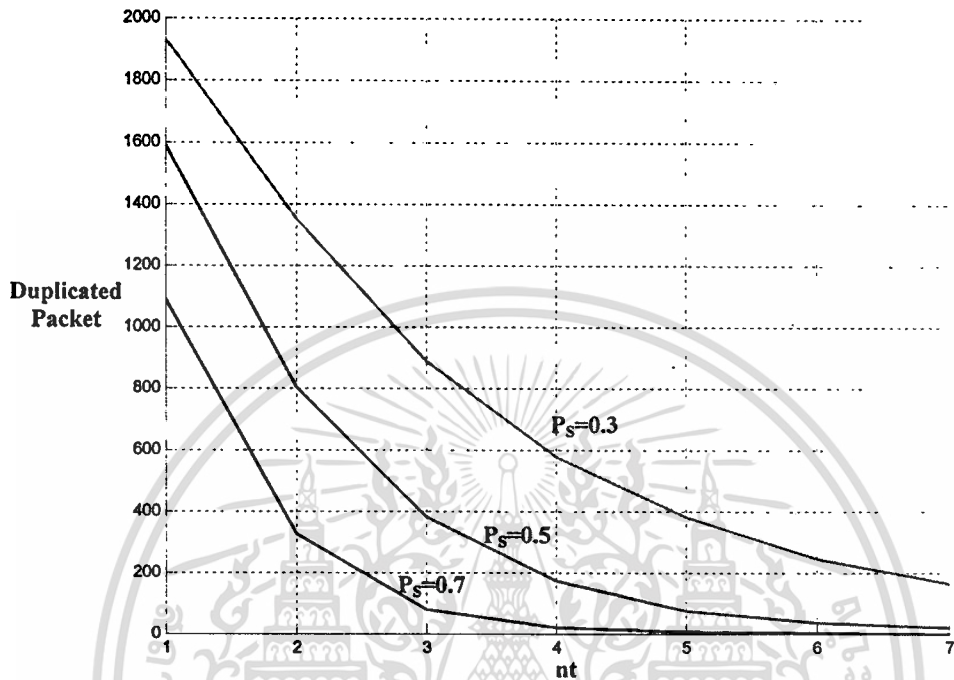
การรีเซตไทม์เอาท์	จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อน
1t	1589
2t	804
3t	382
4t	173
5t	76
6t	38
7t	20



ภาพที่ 5.7 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.7$ และ $\rho = 0.9$

ตารางที่ 5.7 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.7$ และ $\rho = 0.9$

การรีเซตไทม์เอาท์	จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อน
1t	1088
2t	324
3t	80
4t	18
5t	7
6t	1
7t	0

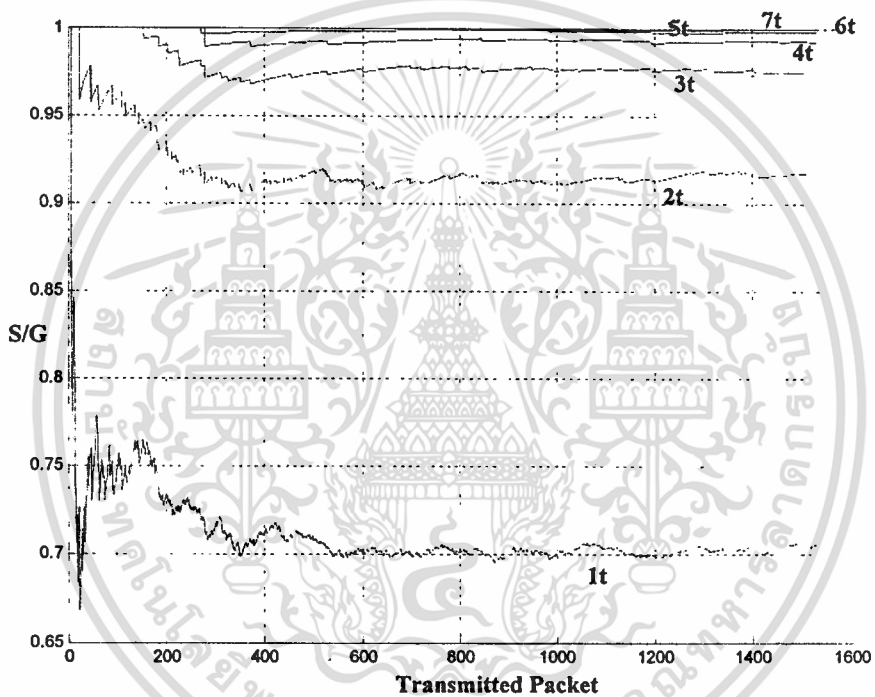


ภาพที่ 5.8 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3, 0.5$ และ 0.7 ที่ $\rho = 0.9$

ภาพที่ 5.5 – ภาพที่ 5.7 เป็นการแสดงให้เห็นถึงจำนวนข้อมูลที่ถูกส่งซ้ำมาจากต้นทาง เมื่อมีการเปลี่ยนแปลงสถานะของความผิดพลาดในเครือข่ายซึ่งก็คือค่า P_s โดยมีค่า 0.3 0.5 และ 0.7 ตามลำดับ โดยมีค่า ρ คงที่เท่ากับ 0.9 ซึ่งผลที่ได้จากการจำลองแสดงให้เห็นว่า เมื่อทำการเพิ่มจำนวนการรีเซตไทม์เอาต์จะทำให้การส่งข้อมูลที่ซ้ำซ้อนลดลงตามลำดับ และจะเห็นได้อย่างชัดเจนเมื่อเทียบกับวิธีการ TCP ทั่วไป โดยตารางที่ 5.5- ตารางที่ 5.7 จะแสดงค่าของข้อมูลที่ถูกส่งซ้ำซ้อน เมื่อเวลาในการจำลองผ่านไป 60 วินาทีที่เวลาในการรีเซตไทม์เอาต์ค่าต่างๆของแต่ละกรณี เมื่อนำค่าจากตารางที่ 5.5 – ตารางที่ 5.7 มาแสดงค่าในภาพที่ 5.8 ในการเปรียบเทียบจำนวนข้อมูลที่ถูกส่งซ้ำมาจากต้นทางกับการรีเซตไทม์เอาต์ค่าต่างๆ จะแสดงได้อย่างชัดเจนว่า ทุกกรณี วิธีการรีเซตไทม์เอาต์จะให้ผลลัพธ์ที่ดีกว่าคือมีจำนวนข้อมูลที่ถูกส่งซ้ำมาจากต้นทางน้อยกว่า และเมื่อเปรียบเทียบที่ค่า P_s จะเห็นว่าค่า P_s มีค่าสูงจะเกิดข้อมูลที่ส่งซ้ำมาจากต้นทางน้อยกว่าค่า P_s ที่มีค่าต่ำ ซึ่งก็เป็นไปตามทฤษฎี

จากผลการจำลองที่ได้กล่าวไปแล้ว แสดงให้เห็นว่า เมื่อมีการเปลี่ยนแปลงค่า ρ หรือค่า P_s วิธีการรีเซตไทม์เอาต์สามารถที่จะลดจำนวนข้อมูลที่ถูกลงส่งซ้ำซ้อนมาจากคั่นทางได้ทุกกรณีเมื่อเทียบกับวิธีการ TCP ทั่วไป

ภาพที่ 5.9 – ภาพที่ 5.14 แสดงค่ากวดทุบเปรียบเทียบระหว่างวิธีการรีเซตไทม์เอาต์ที่ค่าต่างๆ และวิธีการ TCP ทั่วไป รวมทั้งภาวะที่เปลี่ยนแปลงเมื่อมีการเปลี่ยนแปลงค่า ρ และ P_s ที่ค่าต่างๆ เพื่อดูสถานะของระบบ โดยภาพที่ 5.9 เป็นสถานะที่มีการส่งข้อมูลช้าสุดและมีความผิดพลาดในการส่งข้อมูลต่ำสุด ส่วนภาพที่ 5.13 เป็นภาวะที่มีการส่งข้อมูลเร็วสุด และมีความผิดพลาดเกิดขึ้นในเครือข่ายไร้สายสูงสุด



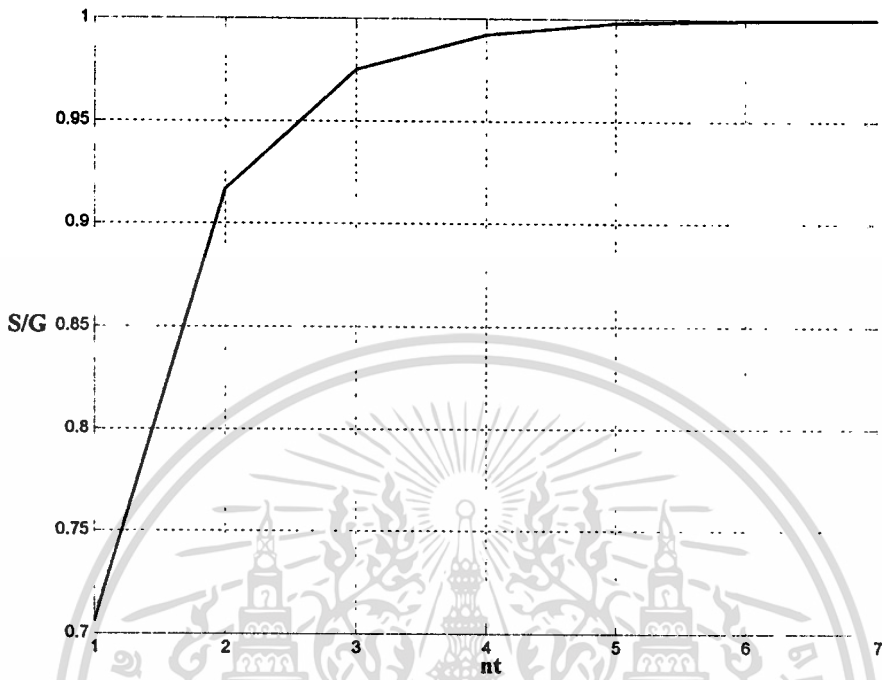
ภาพที่ 5.9 S/G เมื่อ $P_s = 0.7$ และ $\rho = 0.3$ ที่ $1t - 7t$

ตารางที่ 5.8 S/G เมื่อ $P_s = 0.7$ และ $\rho = 0.3$ ที่ $1t - 7t$

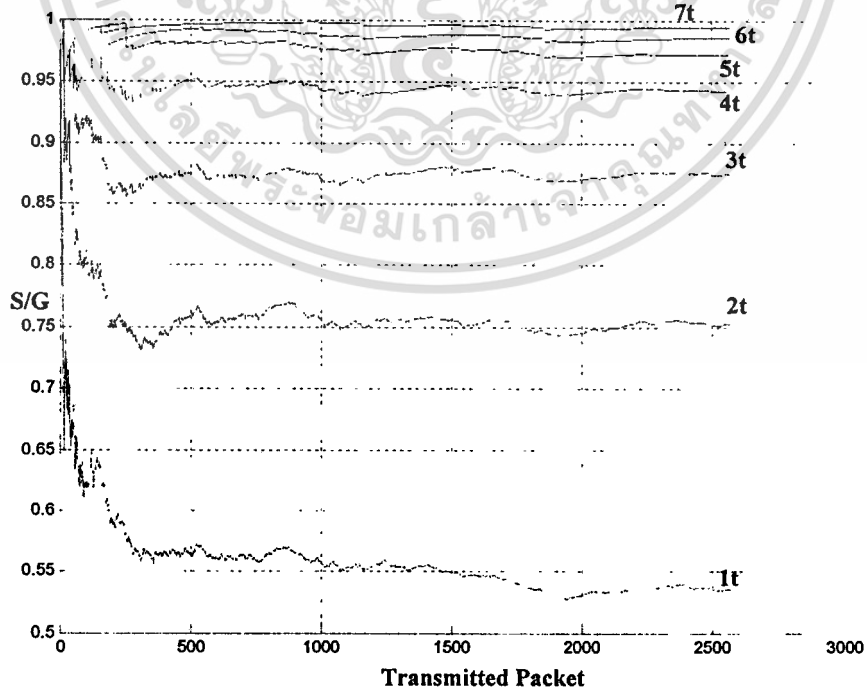
การรีเซตไทม์เอาต์	S/G
1t	.7056
2t	.9164
3t	.9745
4t	.9922
5t	.9980
6t	.9993
7t	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะที่ออกเรียนเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.10 S/G เทียบกับวิธีการ TCP ทั่วไปและการรีเซตไทม์เอาท์ เมื่อ $P_s = 0.7$ และ $\rho = 0.3$

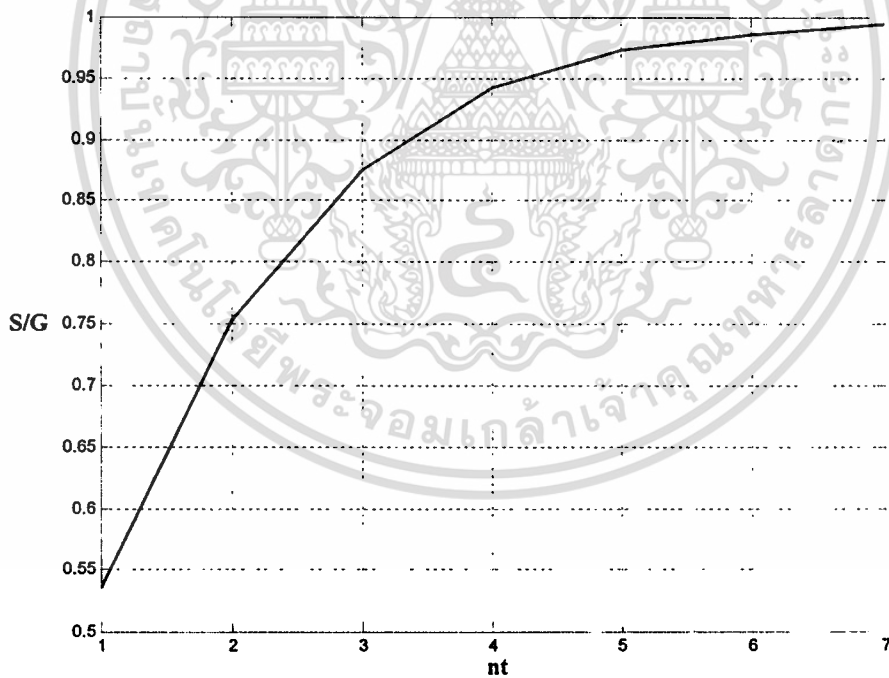


ภาพที่ 5.11 S/G เมื่อ $P_s = 0.5$ และ $\rho = 0.5$ ที่ 1t – 7t

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

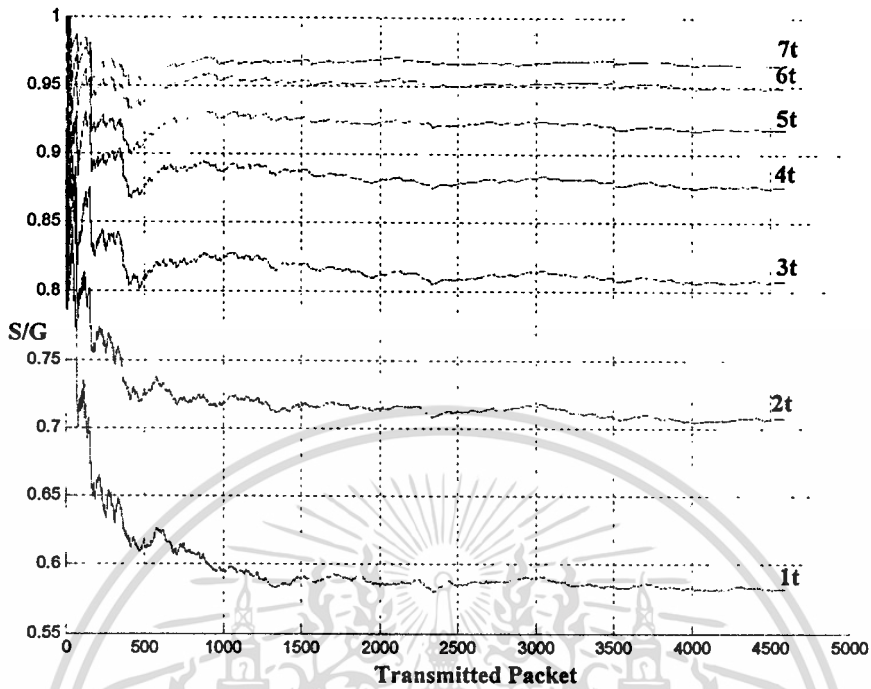
ตารางที่ 5.9 S/G เมื่อ $P_s = 0.5$ และ $\rho = 0.5$ ที่ $1t - 7t$

การรีเซตไทม์เอาท์	S/G
1t	.5354
2t	.7527
3t	.8750
4t	.9424
5t	.9724
6t	.9860
7t	.9945



ภาพที่ 5.12 S/G เทียบกับวิธีการ TCP ทัวไปและการรีเซตไทม์เอาท์ เมื่อ $P_s = 0.5$ และ $\rho = 0.5$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

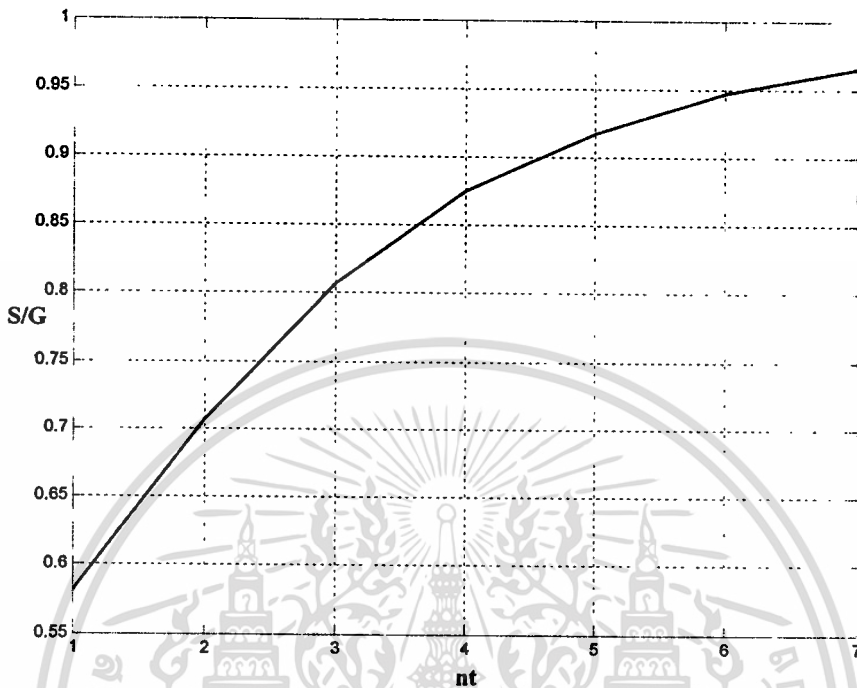


ภาพที่ 5.13 S/G เมื่อ $P_s = 0.3$ และ $\rho = 0.9$ ที่ 1t – 7t

ตารางที่ 5.10 S/G เมื่อ $P_s = 0.3$ และ $\rho = 0.9$ ที่ 1t – 7t

การรีเซตไทม์เอาท์	S/G
1t	.5813
2t	.7069
3t	.8073
4t	.8748
5t	.9173
6t	.9473
7t	.9646

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

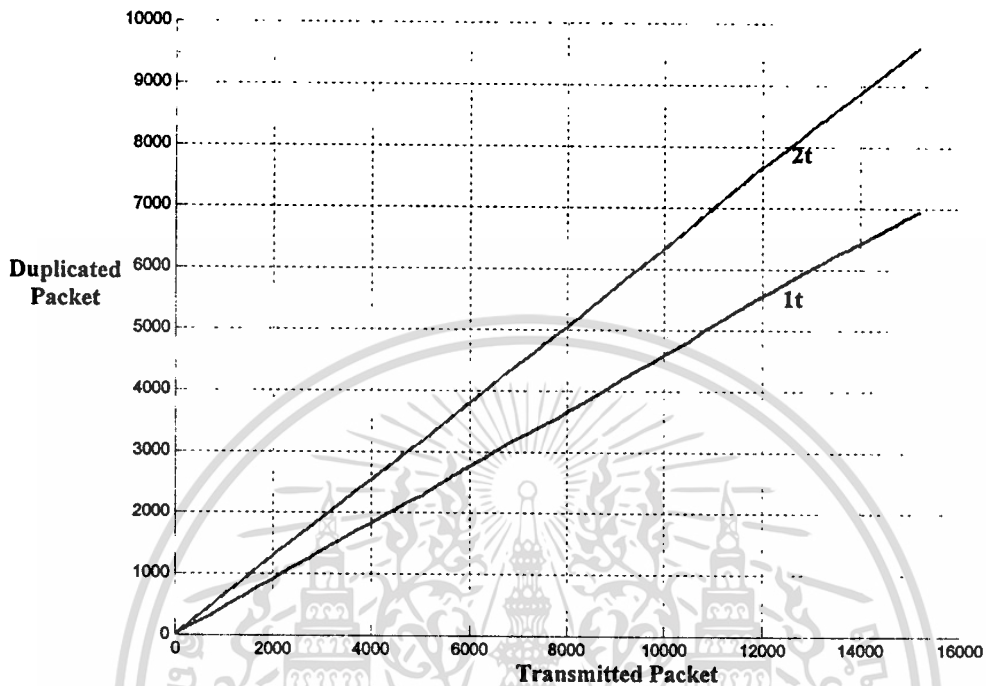


ภาพที่ 5.14 S/G เทียบกับวิธีการ TCP ทั่วไปและการรีเซตไทม์เอาต์ เมื่อ $P_s = 0.3$ และ $\rho = 0.9$

ค่าที่ได้เมื่อทำการเปลี่ยนสถานะต่างๆจากภาพที่ 5.9 – ภาพที่ 5.14 จะแสดงให้เห็นถึงค่ากวดพูนของวิธีการรีเซตไทม์เอาต์ ที่สูงกว่าในทุกกรณี โดยเฉพาะอย่างยิ่งเมื่อค่า P_s มีค่าต่ำ ดังแสดงในภาพที่ 5.9 คือมีค่าเท่ากับ 0.3 และจากตารางที่ 5.8 จะแสดงค่ากวดพูนที่เกิดขึ้นระหว่างวิธีการ TCP ทั่วไปและการรีเซตไทม์เอาต์ ซึ่งเมื่อนำค่าที่ได้มาแสดงด้วยกราฟดังภาพที่ 5.10 จะแสดงให้เห็นได้ชัดเจนถึงผลต่างของค่ากวดพูนที่เกิดขึ้นระหว่างวิธีการ TCP ทั่วไป และการรีเซตไทม์เอาต์ ที่เป็นเช่นนี้ก็เนื่องจากสถานะนี้เป็นสถานะที่เครือข่ายมีค่าความผิดพลาดสูง ทำให้ข้อมูลที่ส่งสำเร็จภายในเวลาไทม์เอาต์มีจำนวนน้อยเมื่อเทียบกับภาวะอื่นๆ แต่เมื่อทำการรีเซตไทม์เอาต์หรือเพิ่มเวลาในการส่งก็จะทำให้ข้อมูลสามารถส่งสำเร็จได้มากขึ้น และเมื่อนำไปเทียบกับผลที่เกิดขึ้นที่ค่า P_s เท่ากับ 0.5 และ 0.7 ซึ่งแสดงค่าโดยตารางที่ 5.9 และ 5.10 ตามลำดับ ก็จะมีแนวโน้มเช่นเดียวกันคือ เมื่อทำการรีเซตค่าไทม์เอาต์สูงขึ้นเรื่อยๆก็จะให้ค่ากวดพูนที่สูงขึ้นตาม ซึ่งก็คือจะลดการเกิดการส่งที่ซ้ำซ้อนนั่นเอง

เมื่อทำการพิจารณาที่ภาพ 5.13 และตารางที่ 5.10 เมื่อ P_s เท่ากับ 0.3 จะพบว่าเมื่อทำการเพิ่มค่ารีเซตไทม์เอาต์ ผลที่ได้ในช่วงการรีเซตไทม์เอาต์ที่ $4t$ $5t$ $6t$ และ $7t$ จะมีค่าใกล้เคียงกันคือ .9922 ถึง

1 ซึ่งเป็นค่าที่แตกต่างกันน้อยมาก ซึ่งจะเห็นได้ว่าการรีเซตไทม์เอาต์ไม่จำเป็นที่จะต้องใช้จำนวนครั้งในการรีเซตให้มากเสมอไป แต่ขึ้นอยู่กับสถานะของเครือข่ายเป็นหลัก



ภาพที่ 5.15 จำนวนแพ็คเกจที่เกิดการส่งซ้ำซ้อนเมื่อ $P_s = 0.3$ และ $\rho = 0.3$ ทำการจำลอง 10 นาที

ภาพที่ 5.15 แสดงการจำลองเมื่อเพิ่มเวลาจากการจำลองที่ 60 วินาทีหรือ 1 นาทีเป็น 10 นาที ซึ่งจะสังเกตว่า กราฟที่ได้จะมีลักษณะเพิ่มขึ้นอย่างคงที่ ซึ่งแสดงให้เห็นว่า เวลาในการจำลองไม่มีผลต่อแนวโน้มของผลที่ได้จากการจำลอง

5.2 การเปรียบเทียบผลลัพธ์ที่ได้จากการจำลองและจากการคำนวณทางคณิตศาสตร์

งานวิจัยนี้ ได้เสนอวิธีการแก้ปัญหาการส่งข้อมูลซ้ำ (Duplicated Packet) ในกรณีที่มีการส่งข้อมูลจากเครือข่ายมีสาย (Wire Network) ไปยังผู้รับที่อยู่ในเครือข่ายไร้สาย (Wireless Network) ภายใต้การสื่อสารแบบ TCP ด้วยการใช้ค่าไทม์เอาต์เมื่อสภาพแวดล้อมภายในเครือข่ายไร้สายไม่ดี

งานวิจัยนี้ได้วิเคราะห์วิธีการที่นำเสนอในทางคณิตศาสตร์ (Numerical Analysis) ที่ได้กล่าวมาแล้วในบทที่ 3 และการวิเคราะห์ในทางการจำลอง (Simulation Analysis) ที่กล่าวในบทที่ 5 นี้

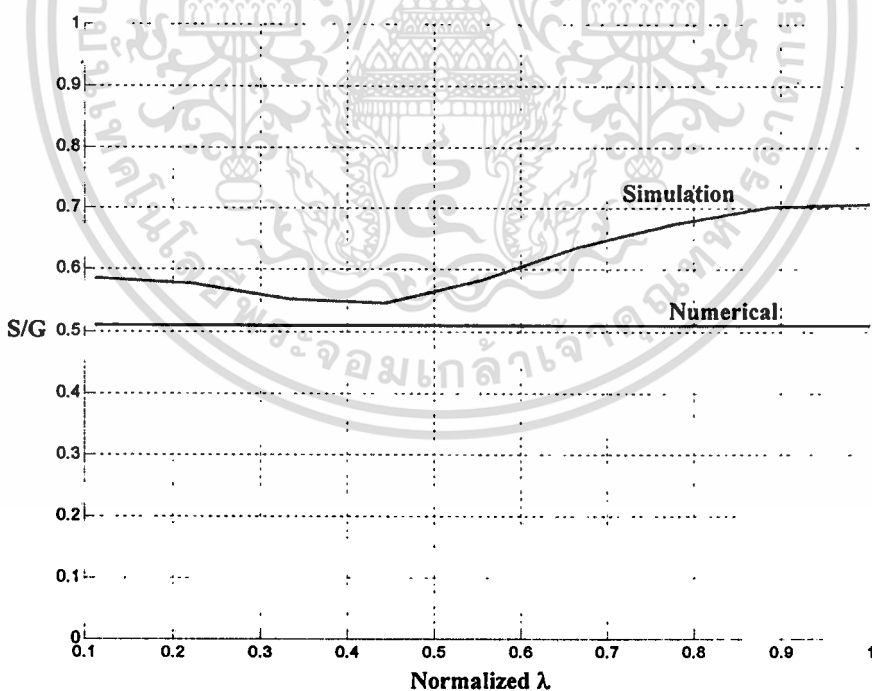
ผลของการวิเคราะห์ทั้งในด้านคณิตศาสตร์ และทางด้านการจำลอง ซึ่งให้เห็นได้อย่างชัดเจนว่า ด้วยวิธีการรีเซตค่าไทม์เอาต์ที่ต้นทางการส่งข้อมูลในเครือข่ายมีสาย TCP ทุกครั้งที่สภาพแวดล้อมภายในเครือข่ายไร้สายไม่ดี จะทำให้ค่าเฉลี่ยของปริมาณข้อมูลที่ส่งสำเร็จ (คือค่า ทูทูป :

Throughput) สูงขึ้น ทั้งนี้เพราะการรีเซ็ทค่าไทม์เอาต์ เป็นการเริ่มต้นจับค่าเวลาของการส่งข้อมูล ภายใต้ระบบควบคุม TCP ใหม่ เหมือนกับการยึดเวลาการรอสัญญาณตอบรับจากฝ่ายรับให้ยาวขึ้น เป็นการเปิดโอกาสให้ สถานีฐานมีเวลาในการส่งต่อข้อมูลมากขึ้น และในทางตรงกันข้าม จะเป็นการชะลอการส่งข้อมูลซ้ำ (Retransmission)

ในที่นี้จะทำการเปรียบเทียบผลลัพธ์ที่ได้จากการจำลองและจากการคำนวณทางคณิตศาสตร์ ในรูปของค่า ทฤษฎี (S) และ ค่า กูตพุท (S/G)

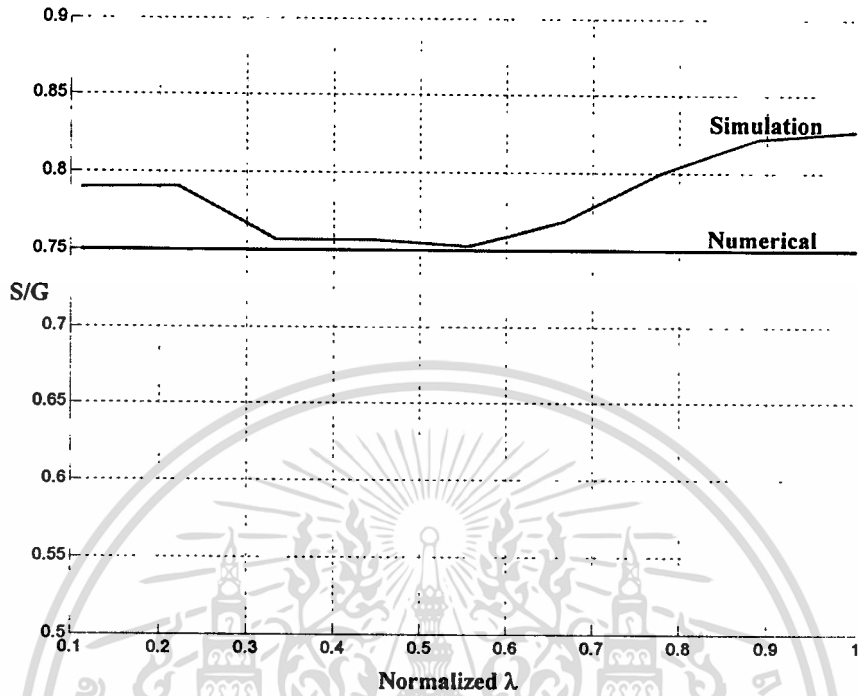
ภาพที่ 5.16, 5.17 และ 5.18 จะแสดงผลการเปรียบเทียบค่า S/G หรือกูตพุท ระหว่างค่าที่ได้จากการวิเคราะห์ทางคณิตศาสตร์ในบทที่ 3 กับค่าที่ได้จากโปรแกรมจำลองการทำงาน โดยวิธีการรีเซ็ทไทม์เอาต์จำนวน 2 ไทม์เอาต์ เมื่อกำหนดค่า $P_s = 0.3$ 0.5 และ 0.7 ตามลำดับ

เมื่อพิจารณาจากภาพที่ 5.16, 5.17 และ 5.18 จะเห็นได้ชัดเจนว่า ผลของค่า S/G ที่ได้จากการจำลอง และที่ได้จากการคำนวณทางคณิตศาสตร์ มีค่าใกล้เคียงกัน และเปลี่ยนแปลงตามค่า Normalized λ ไปในทิศทางหรือแนวโน้มเดียวกัน

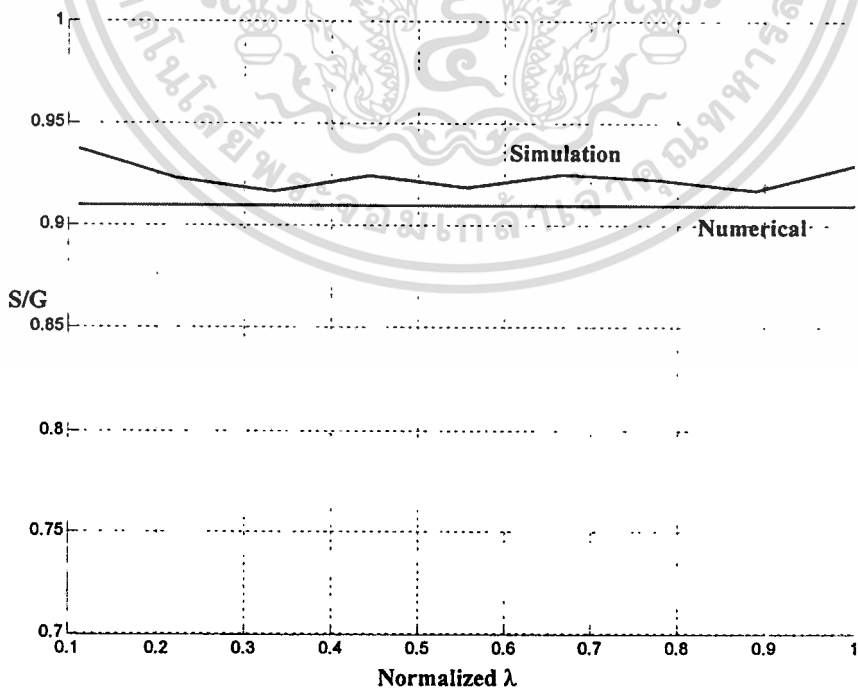


ภาพที่ 5.16 S/G เมื่อ $P_s = 0.3$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.17 S/G เมื่อ $P_s = 0.5$



ภาพที่ 5.18 S/G เมื่อ $P_s = 0.7$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อเป็นการชี้ให้เห็นชัดเจนขึ้น ในที่นี้จะทำการวัดความคลาดเคลื่อนของผลลัพธ์ที่ได้จากการจำลอง และผลลัพธ์ที่ได้จากการคำนวณในรูปแบบของ MSE (Mean Square Error) โดยนำค่า S/G มาเปรียบเทียบกับกัน ตามสมการที่ (5.1)

$$\text{MeanSquareError} = \text{MSE} = \frac{\sum_{i=1}^n (X_i - Y_i)^2}{n} \quad (5.1)$$

เมื่อกำหนดให้

X_i : เป็นค่าของ S/G ที่ได้จากการจำลอง ณ ตำแหน่ง i ($i = 1, 2, \dots, n$)

Y_i : เป็นค่าของ S/G ที่ได้จากการคำนวณทางคณิตศาสตร์ ณ ตำแหน่ง i ($i = 1, 2, \dots, n$)

i : ตำแหน่งของการสุ่มตัวอย่าง (Sampling) ค่า S/G ณ จุดที่มีค่า Normalized λ เท่ากัน ในภาพที่ 5.16, 5.17 และ 5.18

n : จำนวนการสุ่มตัวอย่าง

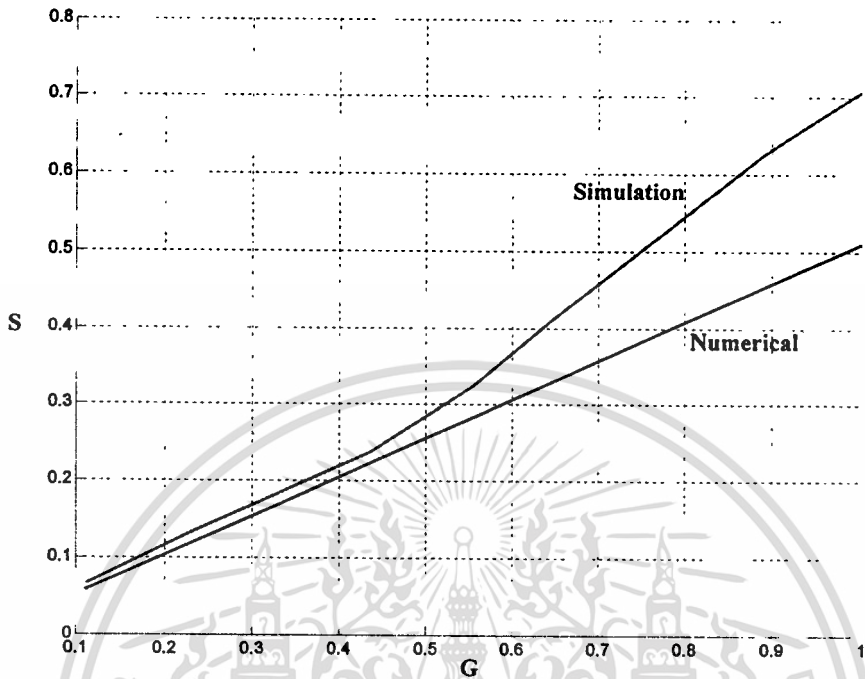
เมื่อทำการสุ่มตัวอย่าง ค่า X_i, Y_i ของ S/G จากกราฟในภาพที่ 5.9, 5.10 และ 5.11 ณ ตำแหน่งที่มีค่า Normalized λ เท่ากัน แล้วนำไปคำนวณค่า MSE ในสมการที่ (5.1) จะได้ผลตามตารางที่ 5.10 เมื่อ $P_s = 0.3, 0.5$ และ 0.7 ตามลำดับ

ตารางที่ 5.11 การเปรียบเทียบค่าจุดฟูทโดยการหาค่า MSE ของการจำลองและการคำนวณ

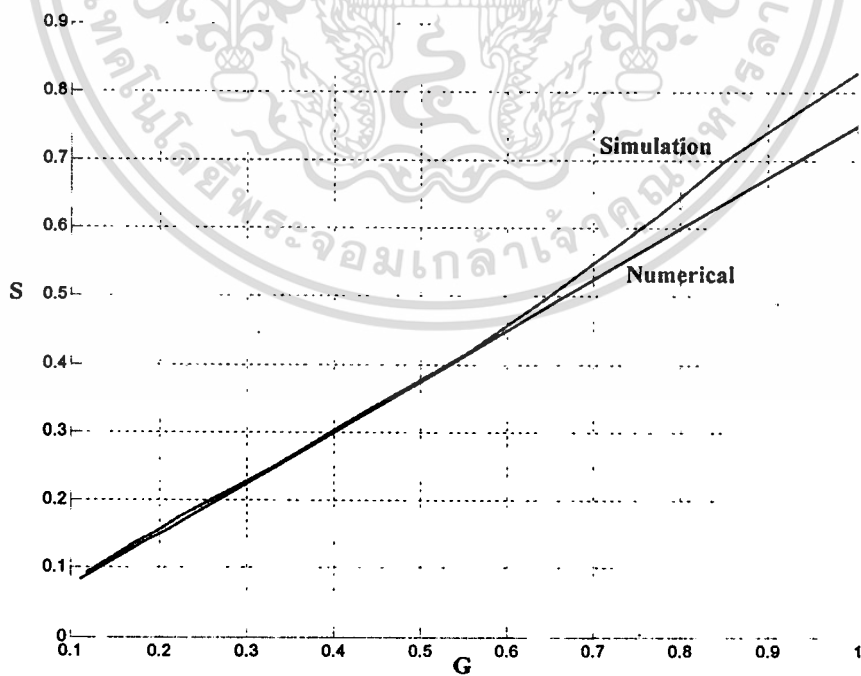
P_s	MSE
0.3	0.0138
0.5	0.0017
0.7	2.038×10^{-4}

จากตารางที่ 5.11 ค่าของ MSE ในแต่ละกรณี มีค่าน้อย ถึง น้อยมาก เป็นการแสดงให้เห็นว่า ค่า S/G ที่ได้จากการจำลองและจากการคำนวณทางคณิตศาสตร์ มีแนวโน้มไปในทิศทางเดียวกัน และมีค่าแตกต่างกันเพียงเล็กน้อย หรือมีค่าที่ใกล้เคียงกัน

ในทำนองเดียวกัน ภาพที่ 5.19, 5.20 และ 5.21 แสดงการเปรียบเทียบค่าทรูพุท (S) ที่ได้จากการจำลอง และจากการคำนวณทางคณิตศาสตร์ เมื่อทำการเปลี่ยนค่า G ภายใต้เงื่อนไขของความน่าจะเป็นในการส่งข้อมูลสำเร็จ 3 ค่า ได้แก่ $P_s = 0.3, 0.5$ และ 0.7 ตามลำดับ และ λ มีค่าเท่ากัน

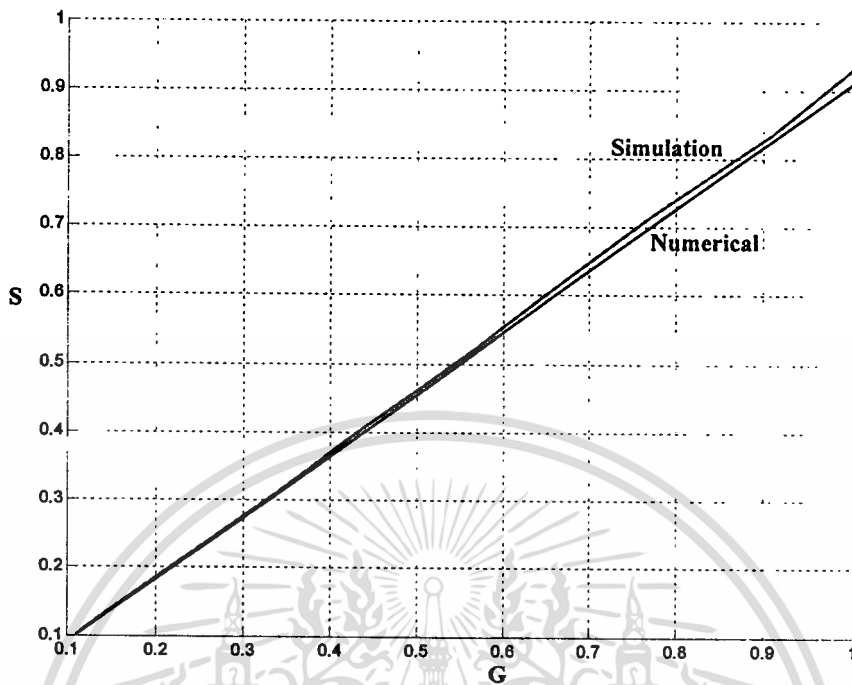


ภาพที่ 5.19 S และ G เมื่อ $P_s = 0.3$



ภาพที่ 5.20 S และ G เมื่อ $P_s = 0.5$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.21 S และ G เมื่อ $P_s = 0.7$

ในทำนองเดียวกันกับกรณีของจุดพืด (S/G) จากภาพที่ 5.16, 5.17 และ 5.18 จะเห็นได้ว่า ค่าของ S ที่ได้จากการจำลอง และที่ได้จากการคำนวณทางคณิตศาสตร์ มีค่าใกล้เคียง

เพื่อเป็นการชี้ให้เห็นชัดเจนขึ้น ในที่นี้จะทำการวัดความคลาดเคลื่อนของผลลัพธ์ที่ได้จากการจำลอง และผลลัพธ์ที่ได้จากการคำนวณในรูปของ MSE (Mean Square Error) โดยนำค่า S มาเปรียบเทียบกับ ตามสมการที่ (5.1)

เมื่อกำหนดให้

X_i : เป็นค่าของ S ที่ได้จากการจำลอง ณ ตำแหน่ง i ($i = 1, 2, \dots, n$)

Y_i : เป็นค่าของ S ที่ได้จากการคำนวณทางคณิตศาสตร์ ณ ตำแหน่ง i ($i = 1, 2, \dots, n$)

i : ตำแหน่งของการสุ่มตัวอย่าง (Sampling) ค่า S ณ จุดที่มีค่า G เท่ากัน ในภาพที่ 5.19, 5.20 และ 5.21

n : จำนวนการสุ่มตัวอย่าง

เมื่อทำการสุ่มตัวอย่างค่า X_i, Y_i ของ S จากกราฟในภาพที่ 5.19, 5.20 และ 5.21 ณ ตำแหน่งที่มีค่า G เท่ากัน แล้วนำไปคำนวณค่า MSE ในสมการที่ (5.1) จะได้ผลตามตารางที่ 5.12 เมื่อ $P_s = 0.3, 0.5$ และ 0.7 ตามลำดับ

ตารางที่ 5.12 การเปรียบเทียบค่า S และ G โดยการหาค่า MSE ของการจำลองและการคำนวณ

P_s	MSE
0.3	0.0096
0.5	9.5306×10^{-4}
0.7	2.1308×10^{-4}

จากตารางที่ 5.12 ค่าของ MSE ในแต่ละกรณี มีค่าน้อย ถึง น้อยมาก เป็นการแสดงให้เห็นว่า ค่า S ที่ได้จากการจำลองและจากการคำนวณทางคณิตศาสตร์ มีแนวโน้มไปในทิศทางเดียวกัน และมีค่าแตกต่างกันเพียงเล็กน้อย หรือมีค่าที่ใกล้เคียงกัน

จากการวิเคราะห์ในทั้งสองรูปแบบที่กล่าวมา คือ การหาค่า S/G และ S กับ G เมื่อทำการเปลี่ยนค่า P_s จะเห็นว่าค่าที่ได้จากการคำนวณ และการจำลอง จะมีแนวโน้มไปในทิศทางเดียวกัน และมีค่าความแตกต่างอยู่ในระดับที่ยอมรับได้ ซึ่งแสดงให้เห็นว่า แบบจำลองทางคณิตศาสตร์ที่สร้างขึ้นมากับโปรแกรมการจำลองมีการทำงานที่สอดคล้องกัน

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

TCP เป็นโปรโตคอลที่ถูกออกแบบมาเพื่อใช้กับเครือข่ายมีสายเป็นหลัก เมื่อมีการนำมาใช้งานร่วมกับเครือข่ายแบบไร้สายซึ่งมีระบบที่แตกต่างกันจึงทำให้เกิดปัญหาในการใช้งาน ซึ่งปัญหาที่ศึกษาในงานวิจัยนี้ก็คือการเกิดการส่งข้อมูลที่ซ้ำซ้อนระหว่างโฮสต์อยู่กับที่และโฮสต์เคลื่อนที่ เมื่อสถานีฐานใช้เวลาในการส่งข้อมูลไปโฮสต์เคลื่อนที่มากกว่าค่า TCP ไทม์เอาต์ ซึ่งเป็นการทำงานที่ซ้ำซ้อนระหว่างสถานีฐานและ TCP ต้นทาง จึงได้ทำการวิเคราะห์การทำงานของสถานีฐาน และสร้างเป็นแบบจำลองทางคณิตศาสตร์ของสถานีฐาน โดยใช้ทฤษฎีการเข้าคิวแบบ M/M/1 ในการสร้างแบบจำลอง

การวิเคราะห์แบบจำลอง ทำให้สามารถสร้างสมการซึ่งสามารถวิเคราะห์สถานะดีหรือไม่ดีของเครือข่ายไร้สายได้ ซึ่งก็คือ ถ้าเครือข่ายไร้สายอยู่ในสถานะดีคือการส่งข้อมูลจากสถานีฐานไปยังโฮสต์เคลื่อนที่ก็จะสามารถส่งสำเร็จได้ในเวลา TCP ไทม์เอาต์ แต่ถ้าเครือข่ายไร้สายอยู่ในภาวะไม่ดีคือข้อมูลก็ไม่สามารถส่งสำเร็จได้ภายในเวลาที่กำหนด หรือ TCP ไทม์เอาต์ ซึ่งจะทำให้เกิดการส่งข้อมูลที่ซ้ำซ้อนขึ้น จึงได้ทำการเสนอการแก้ปัญหาโดยวิธีการรีเซตไทม์เอาต์ โดยจะทำการรีเซตค่าไทม์เอาต์ของ TCP เมื่อเครือข่ายไร้สายอยู่ในภาวะไม่ดี ทำให้สถานีฐานมีเวลาในการส่งข้อมูลไปที่โฮสต์เคลื่อนที่ได้มากขึ้น และเป็นการลดการส่งข้อมูลซ้ำซ้อนระหว่างสถานีฐานและ TCP ต้นทาง ซึ่งผลจากการวิเคราะห์ทางคณิตศาสตร์เปรียบเทียบระหว่างการส่งข้อมูลแบบ TCP ทั่วไป และการรีเซตไทม์เอาต์ วิธีการรีเซตไทม์เอาต์จะให้ประสิทธิภาพที่ดีกว่า

จากแบบจำลองและวิธีการที่นำเสนอจึงได้สร้างโปรแกรมจำลองการทำงานของแบบจำลองโดยใช้โปรแกรมจำลองเครือข่ายที่มีชื่อว่า OPNET ซึ่งผลการจำลองที่ได้มีความสอดคล้องกับแบบจำลองการเข้าคิวที่ได้นำเสนอ โดยเมื่อทำการจำลองโดยใช้ค่าเดียวกับการจำลองทางคณิตศาสตร์ผลที่ได้จะต่างกันเพียงเล็กน้อย โดยการจำลองจะทำการจำลองในภาวะความผิดพลาดที่ค่าต่างๆ และค่าประสิทธิภาพ หรือ p ที่ค่าต่างๆ (0.3, 0.5, 0.7) โดยจะทำการจำลองเปรียบเทียบวิธี TCP ทั่วไป กับการรีเซตไทม์เอาต์ โดยในการรีเซตไทม์เอาต์จะทำการทดลองรีเซตค่าไทม์เอาต์ 2 - 7 ไทม์เอาต์ เพื่อดูผลที่เกิดขึ้น ซึ่งผลที่ได้แสดงให้เห็นว่าวิธีการที่นำเสนอจะเกิดแพ็กเก็ตที่ซ้ำซ้อนน้อยกว่าวิธีการ TCP ทั่วไป รวมทั้งมีจุดพุงที่ดีกว่า โดยจะเห็นว่า เมื่อใช้การรีเซตไทม์เอาต์สูงขึ้น แพ็กเก็ตที่ซ้ำซ้อนจะลดลง แต่ในบางสถานะ จำนวนครั้งในการ รีเซตไทม์เอาต์ที่มีจำนวนรอบสูงๆ ก็จะมีค่าใกล้เคียงกัน

ข้อเสนอแนะ

ในงานวิจัยนี้จะใช้วิธีการรีเซตไทม์เอาท์ โดยการทดลองใช้การรีเซตหลายๆค่า ซึ่งพบว่าในบางครั้งเวลาในการรีเซตที่มากกว่าก็ให้ค่าคุณภาพหรือการลดแพ้กเขตที่ช้าซ้อนได้เท่ากับเวลาในการรีเซตที่น้อยกว่า ถ้าสามารถหาค่าที่เหมาะสมในการรีเซตหรือย้คเวลาได้ ก็จะทำให้ระบบมีการทำงานดีขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- [1] Tim Parker. **Teach Yourself TCP/IP in 14 Days**. 2nd Ed. : SAMS Publishing. 1996.
- [2] Dr. Sidnie Feit. **TCP/IP Architecture, Protocol, and Implementation with Ipv6 and IP Security**. 2nd Ed. : McGraw-Hill. 1996.
- [3] William Stallings. **Wireless Communication AND Networking**. : Prentice-Hall International, Inc. 2002.
- [4] Mike Tanner. **Practical Queueing Analysis**. : McGraw-Hill International (UK) Ltd. 1995.
- [5] Leonard Kleinrock. **QUEUEING SYSTEMS VALUME 1: THEORY**. John Wiley & Sons, Inc. 1975.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ผลงานวิจัยและบทความทางวิชาการที่ได้รับการตีพิมพ์

1. Narissorn Sangkanong and Ruttikorn Varakulsiripunth, "Performance Improving on Transmission TCP 's Message over Wireless Networks". The Third International Symposium on Wireless Personal Multimedia Communications. (WPMC '00). Bangkok, Thailand, November 12 - 15, 2000.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Performance Improving on Transmission TCP 's Message over Wireless Networks

Narissorn Sangkanong[†] and Ruttikorn Varakulsiripunth^{††}

Faculty of Engineering and Research Center for Communications and Information Technology(ReCCIT)
King Mongkut 's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
Email: ^{††}kvruttik@kmitl.ac.th , [†]nss@kmitnb.ac.th

Abstract - When a TCP 's message was sent to a mobile host in wireless network, it had to be transmitted via a base station. If the environment in wireless network at that moment is not good, i.e., error rate is too high, then the time to transfer this message to reach its destination mobile host successfully may longer than TCP 's timeout. Under this circumstance, TCP will retransmit same message, hence, duplicated message will occur and system performance will decrease.

This paper proposes a mechanism to avoid this problem and also to improve the system performance by defining bad and good states in according to error rate of wireless environment. If wireless network is in bad state at any moment, TCP network will be informed by base station to extend timeout of that message, i.e., to reset timeout. The definition of bad and good states is analyzed in term of error rate and queueing model of base station is introduced. Numerical results show very good performance on our mechanism.

Keywords: TCP, Queueing theory, Wireless network

I. INTRODUCTION

Transmission Control Protocol (TCP) is a reliable transmission protocol. It is designed to be suitable for wire network. TCP detects message losses by using timer at TCP sender. If TCP does not receive any response from destination before timeout will occur, it assumes that the message has lost and retransmits same message again. When the TCP 's message is sent to a mobile host (MH) via wireless links, the message loss mainly occurs due to the density of error rate of wireless channel. Normally, the time to transmit a message from base station (BS) to a MH and response to TCP sender under good circumstance, i.e., error is low, should be shorter than TCP 's timeout. However, in a bad circumstance, i.e., error is too high, this time will be longer than TCP 's timeout. Therefore, if the time that BS still try to transmit a message in according to data recovery procedure in data link layer is longer than TCP 's timeout, then TCP retransmit the same message again to BS, and this will cause duplicated message. This problem will degrade the TCP performance.

In this paper, we propose a mechanism to avoid this problem and also to improve the system performance by defining bad and good states in according to error rate of wireless environment. If wireless network is in bad state at any moment, TCP network will be informed by

concerned BS to extend timeout of that message, i.e., to reset timeout. The definition of bad and good states is analyzed in term of error rate and queueing model of BS is introduced. Numerical results show very good performance on our mechanism.

II. SOME OTHER RELATED WORKS

There are some other related works to our approach [1,2,3,4]. Paper [1,2] proposed an approach to split the connection between a fixed host in TCP network and MH in wireless network into two sections. The first section is the connection between TCP 's fixed host and BS over wired medium. The second section is the connection between BS and MH over wireless medium. This approach improves TCP performance by separating the flow control and congestion control of the wireless link from the TCP network. However, it violates the semantics of end-to-end reliability. Because it is not the end-to-end control mechanism. Therefore, the acknowledgement from BS can arrive at the source (a fixed host) before the message actually reaches the intended destination MH. Furthermore, this approach requires a lot of state maintenance at the BS.

Normally, the message size on typical wireless network is smaller than the message size on wired network. Paper [3] presented a simulation study on the suitable message size that should send from wired to wireless network. The maximum message size was changed in order to prevent excessive transmission error. However, this approach increases transmission overhead because the total message of fixed host has to be segmented into individual blocks within defined maximum message size before transmission to wireless network.

Paper [4] proposed a snoop agent at the BS to act on behalf of sending source. Snoop agent will perform local retransmission inside wireless network whenever it received an acknowledge of message loss from MH, or when the local timeout occurred. However, the sending source will not be informed the situation during the snoop agent perform this procedure. Therefore, the duplicated message problem can occur in both [3] and [4].

In order to avoid above-mentioned problems, we have considered on the end-to-end control from fixed host (sending source) in TCP network to destination MH

in wireless network. The sending source will be informed by BS about situation during the local transmission in wireless network. Whenever the bad transmission environment in wireless network is detected by BS, that BS will inform sending source in TCP network and ask it to restart the TCP's timeout.

III. BAD AND GOOD STATES IN WIRELESS NETWORKS

The bad and good states in wireless network will be defined in according to the time that spent for a message at BS. And this time varies with the error rate occurred in wireless links. Let us assume that the message sent from TCP network to BS has a rate λ m/s, and it is Poisson distribution. We also assume that BS can service μ m/s and the time of service each message is exponential distribution. Then, the model of BS is M/M/1 queueing system as shown in Figure 1.

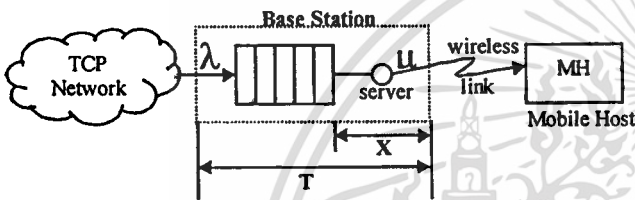


Figure 1. M/M/1 queueing model of BS.

As shown in Figure 1, let T be the total system time for a message that has to spent in a wireless network before arriving its destination MH, then according to [5], it will be given by

$$T = \frac{1/\mu}{1-\rho} \quad (1)$$

where $\rho = \frac{\lambda}{\mu} < 1$ is a utilization factor of BS.

Now, let X be an average service time of BS under no error condition, including transmission time from BS to MH and response time from MH to TCP sender. Then from (1), system time in wireless network for a message under no error environment can be obtained as below.

$$t = \frac{X}{1-\rho} \quad (2)$$

Here, ρ is a constant ($0 < \rho < 1$) and $X = 1/\mu$, under no error environment. Now, let ε be a probability of current error that may occur in wireless link. It is clear that as much as error occur, the time that a message has to spent in BS will increase related with t . Hence, we have to modify the value of T from (1) in term of t from (2) under the error environment as follow.

$$\begin{aligned} T &= \varepsilon t + t \\ &= (\varepsilon + 1)t \end{aligned} \quad (3)$$

Finally, it can be obtained

$$T = (\varepsilon + 1) \frac{X}{1-\rho} \quad (4)$$

Here, $\varepsilon = 0$, means no error in wireless link, then $T = t$. Now, let us define τ to be TCP's timeout. The definition of bad and good states of wireless link is determined as follow.

- If $T \leq \tau$, it is a good state. This condition means, that the message transmission between TCP sender and MH can complete within a TCP's timeout.
- If $T > \tau$, it is a bad state. This condition means that the message transmission can not complete within a TCP's timeout and retransmission will occur which may cause duplicating of message.

Figure 2 shows the relation of T and ε . The value of T is scaled in term of time X .

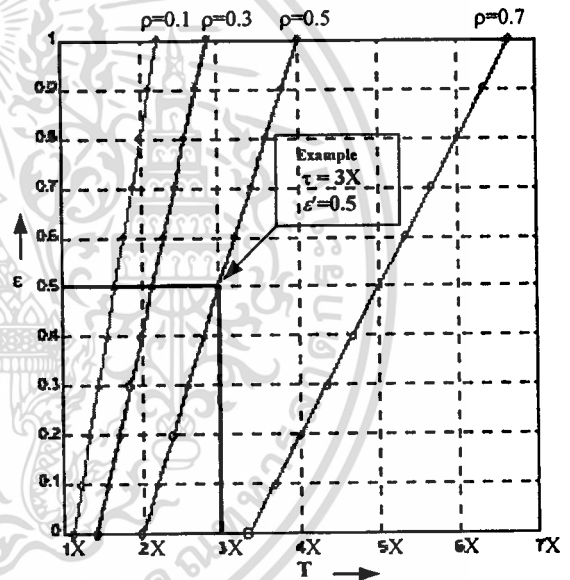


Figure 2. Relation of ε and T with different ρ .

The value of ε at $T = \tau$ is a critical value to change the state from bad to good, or good to bad. Let ε' be the critical value of ε , then from (4) when $T = \tau$, we obtain

$$\begin{aligned} \tau &= (\varepsilon' + 1) \frac{X}{1-\rho} \\ \text{and} \quad \varepsilon' &= \frac{\tau(1-\rho)}{X} - 1 \end{aligned} \quad (5)$$

For example, if $\tau = 3X$ and $\rho = 0.5$ as shown in Figure 2, then $\varepsilon' = 0.5$. So, if ε is more than 0.5, then T is more than $3X$ or $T > \tau$, the wireless network is in bad state.

Therefore, in general, as shown in Figure 3, it is clear that,

- when $\varepsilon \leq \varepsilon'$, then $T \leq \tau$, the wireless network is in good state, and
- when $\varepsilon > \varepsilon'$, then $T > \tau$, the wireless network is in bad state.

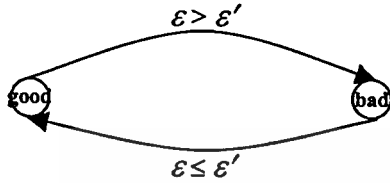


Figure 3. Transition between bad and good states.

From the example shown in Figure 2, when $\tau = 3X$ and $\varepsilon' = 0.5$, we can define the range of current error ε into two parts. Such as, $0 \leq \varepsilon \leq \varepsilon'$ is the range of good state and $\varepsilon' < \varepsilon \leq 1$ is the range of bad state. Therefore, $1 - \varepsilon'$ will be a full range of value of ε that causes wireless network fall into bad state. As ε becomes bigger than ε' , the failure rate of transmitting a message via wireless network will grow higher. Thus, we can yield the unsuccessful transmission probability (P_u) as follow.

$$P_u = \frac{\varepsilon - \varepsilon'}{1 - \varepsilon'} \quad , \quad \varepsilon > \varepsilon' \quad (6)$$

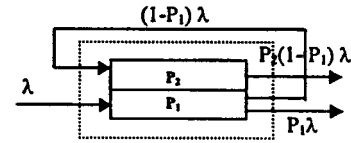
Let us define P_s as a successful probability of transmitting a message in wireless network when it is in bad state. Then from (6), we can obtain

$$\begin{aligned} P_s &= 1 - P_u \\ P_s &= 1 - \frac{\varepsilon - \varepsilon'}{1 - \varepsilon'} \\ &= \frac{1 - \varepsilon}{1 - \varepsilon'} \quad , \quad \varepsilon > \varepsilon' \quad (7) \end{aligned}$$

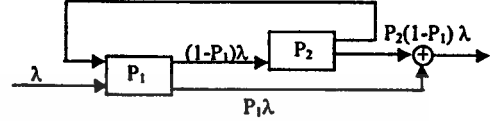
IV. TCP'S TIMEOUT RESETTING

As we have described in section III that when $T > \tau$ or $\varepsilon > \varepsilon'$, the wireless network is in bad state. Then our approach is to extend the TCP's timeout or reset TCP's timeout whenever it is found that $\varepsilon > \varepsilon'$. In this paper we will consider only the condition of wireless network in the duration of two timeout. Let P_1 and P_2 be the value of P_s in the first and second timeout respectively. Then the probability model of wireless network in normal TCP

transmission and our approach transmission can be described as shown in Figure 4 (a) and (b) respectively.



4 (a) Normal TCP transmission.



4 (b) Our approach transmission.

Figure 4. Probability model of wireless network in bad state.

Let G = total message sent from TCP to wireless network including retransmission.
 S = total message that reached destination in wireless network successfully (throughput).

From Figure 4 (a), we can obtain the following equations for normal TCP transmission.

$$G = \lambda + (1 - P_1)\lambda \quad (8)$$

$$S = P_1\lambda + P_2(1 - P_1)\lambda \quad (9)$$

$$\text{goodput} = \frac{S}{G} = \frac{P_1\lambda + P_2(1 - P_1)\lambda}{\lambda + (1 - P_1)\lambda} \quad (10)$$

And from Figure 4 (b), we can obtain the following equations for our approach transmission.

$$G = \lambda \quad (11)$$

$$S = P_1\lambda + P_2(1 - P_1)\lambda \quad (12)$$

$$\text{goodput} = \frac{P_1\lambda + P_2(1 - P_1)\lambda}{\lambda} \quad (13)$$

Where $P_1, P_2 \in P_s$

V. NUMERICAL RESULTS

From the example in section III, let us assume $\varepsilon' = 0.5$ and $\varepsilon = 0.8$, then from (7), $P_s = 0.4$. In order to compare the result between normal TCP and our procedure, let us also assume that $P_s = P_1 = P_2$ in this numerical analysis as shown in Figure 5 and Figure 6.

The numerical results in Figure 5 and Figure 6 demonstrate that our approach obtains throughput higher than normal TCP. This is because of the extension of

timeout when wireless network was in bad state. Then the number of retransmission is reduced.

timeout whenever the wireless network is in bad state, the performance, in term of throughput, can be improved.

VII. ACKNOWLEDGEMENT

We would like to gratitude Japan International Cooperation Agency (JICA) for research supporting.

VIII. REFERENCES

- [1] R Yavatkar and N. Bhagwat, "Improving end-to-end performance over mobile internetworks," Proc. Of workshop on Mobile Computing Systems and Applications, Dec., 1994.
- [2] A. Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for mobile host," ICDCS, Oct., 1994.
- [3] Bikram S. Bakshi, P. Krishna, N. H. Vaidya and D. K. Pradhan, "Improving Performance of TCP over Wireless Networks," Technical Report TR-96-014, May 1, 1996.
- [4] H. Balakrishnan, S. Seshan, E. Amir and R. H. Katz, "Improving TCP/IP performance over wireless networks," Proc. 1st ACM Conf. On Mobile Computing and Networking, November 1995.
- [5] Leonard Kleinrock, "Queueing Systems," John Wiley & Son, 1975.

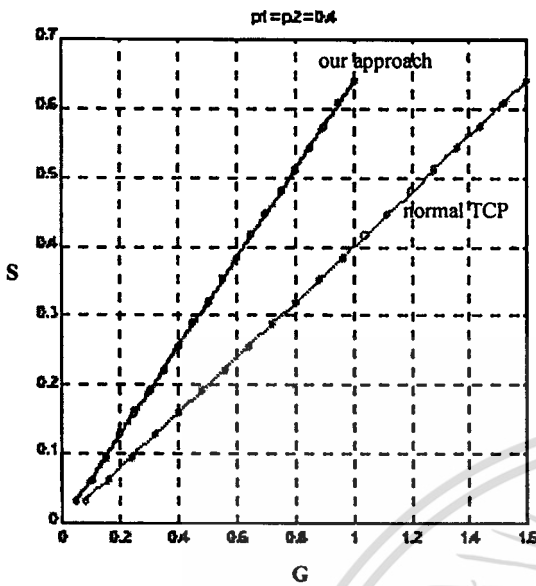


Figure 5. S and G in normal TCP and our approach.

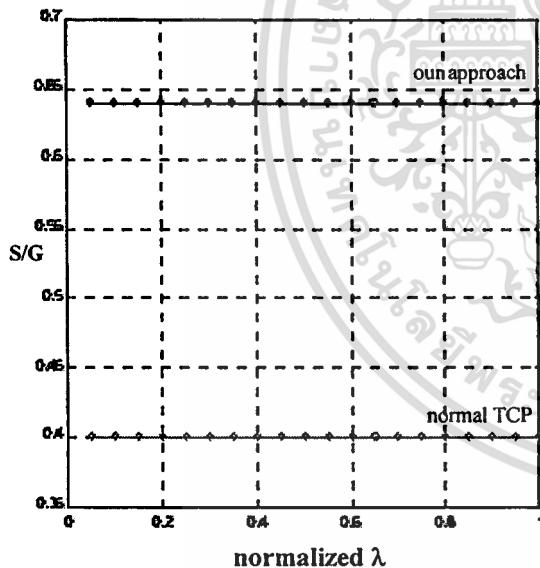


Figure 6. Goodput in normal TCP and our approach.

VI. CONCLUSIONS

In this paper we proposed the TCP's timeout resetting mechanism whenever the time to transmit a message via wireless network will be longer than TCP's timeout. The base station of wireless network can observe this bad situation by estimating the current error rate and informs TCP network. By extending the TCP's

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายนิศร แสงคนอง
วันเดือนปีเกิด	8 มกราคม 2518
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	สำเร็จการศึกษาระดับปริญญาตรี สาขาวิศวกรรมไฟฟ้า
สถานที่สำเร็จการศึกษา	สถาบันเทคโนโลยีพระจอมเกล้า พระนครเหนือ
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2538
ผลงานวิจัยที่ได้รับการตีพิมพ์	

1. Narissorn Sangkanong and Ruttikom Varakulsiripunth. **Performance Improving on Transmission TCP 's Message over Wireless Networks.** The Third International Symposium on Wireless Personal Multimedia Communications. (WPMC '00). Bangkok, Thailand, November 12 - 15, 2000. pp. 252 - 255