

การวิเคราะห์และออกแบบโครงข่ายที่มีการใช้งานตัวกลางดาวเทียมและ ตัวกลาง
ภาคพื้นดินร่วมกัน

ANALYSIS AND DESIGN OF INTEGRATED TERRESTRIAL AND SATELLITE
CONNECTIVITY NETWORK



นายเสรี อัสวารักษ์

MR. SEREE ASSAWARUK

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2541

ISBN 974-622-101-9

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหน้า.....
เลขทะเบียน.....**29652**
วัน, เดือน, ปี...**5 ส.ค. 2541**

**ANALYSIS AND DESIGN OF INTEGRATED TERRESTRIAL AND SATELLITE
CONNECTIVITY NETWORK**

SEREE ASSAWARUK

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

1998

ISBN 974-622-101-9

หัวข้อวิทยานิพนธ์	การวิเคราะห์และออกแบบโครงข่ายที่มีการใช้ตัวกลางดาวเทียม และ ตัวกลางภาคพื้นดินร่วมกัน
ชื่อนักศึกษา	นายเสรี อัสวรักษ์
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ. ดร. สุวิพล สิริชีวะภาค
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2541

บทคัดย่อ

ระบบดาวเทียมที่มีการใช้งานอยู่ในปัจจุบันมีลักษณะที่สำคัญคือ ค่าหน่วยเวลาของการส่งข้อมูลมีค่าสูงในขณะที่สามารถใช้แถบความถี่กว้าง ดังนั้นเส้นทางของข้อมูลที่เชื่อมต่อกับระบบดาวเทียม จะมีผลกระทบอย่างมากต่อคุณสมบัติโดยรวมของโครงข่ายที่มีการใช้งานตัวกลางดาวเทียมและตัวกลางภาคพื้นดินร่วมกัน

วิทยานิพนธ์ฉบับนี้ได้ศึกษารูปแบบของการใช้งานร่วมกันระหว่างตัวกลางดาวเทียมที่สามารถรองรับข้อมูลความเร็วปานกลางกับระบบตัวกลางภาคพื้นดินที่สามารถรองรับข้อมูลในขนาดจำกัด ซึ่งระบบดาวเทียมที่นำมาเป็นต้นแบบในการพิจารณา มีลักษณะของการเข้าใช้ช่องสัญญาณเป็นแบบสลับโดยผู้ใช้หลายคนและเป็นอิสระต่อกัน รวมทั้งโปรโตคอลที่กำหนดเป็นแบบที่สามารถจองช่องสัญญาณล่วงหน้าได้ ระบบดาวเทียมที่มีลักษณะดังกล่าวมีการศึกษาและใช้งานหลายรูปแบบแตกต่างกัน ในงานวิจัยนี้ได้นำเสนอแบบจำลองระบบดาวเทียมที่สามารถพัฒนาเป็นโมดูลที่เชื่อมต่อกับโมดูลของโครงข่ายภาคพื้นดินได้โดยสะดวกโดยพฤติกรรมของระบบดาวเทียมจะแทนด้วยแบบจำลองการไหลของแพคเกจที่ไม่ซับซ้อน พารามิเตอร์สำหรับพิจารณาสมรรถนะได้จากการคำนวณที่สถานะสมดุลทำให้จำนวนของตัวแปรที่เกี่ยวข้องมีขนาดลดลง

การใช้สถานีดาวเทียมขนาดเล็ก(VSAT)ร่วมกับสถานีที่ใช้ตัวกลางภาคพื้นดินในลักษณะของโนดที่เป็นจุดผ่านทาง(Access Node) จำเป็นต้องคำนึงถึงสถานะของระบบช่องสัญญาณดาวเทียมที่สถานีดาวเทียมขนาดเล็กดังกล่าวเชื่อมต่ออยู่ ในงานวิจัยนี้ได้ศึกษาและแสดงแนวทางในการกำหนดจุดทำงานของระบบช่องสัญญาณดาวเทียม เพื่อการออกแบบที่เหมาะสม สำหรับโครงข่ายแบบผสม

อัลกอริทึมของการออกแบบโครงข่ายแบบผสมดังกล่าว สามารถพัฒนาหรือวางแนวทางในการออกแบบได้หลายรูปแบบตามลักษณะโครงสร้างของโครงข่ายที่แตกต่างกันรวมทั้งคุณสมบัติและพฤติกรรมของระบบดาวเทียมที่มีอยู่ในปัจจุบัน แต่โครงสร้างหรือหลักการสำคัญสามารถจำแนกออกได้เป็น 2 หลักการ คือ การกำหนดปริมาณของข้อมูลในแต่ละช่องสัญญาณ และการกำหนดเส้นทางส่งผ่านของข้อมูล วิทยานิพนธ์นี้จะใช้โครงสร้างของ 2 หลักการดังกล่าวรวมกันเป็นโครงสร้างหลักในการวิจัยและพัฒนาอัลกอริทึม สำหรับการออกแบบและคำนวณหาจุดทำงานที่เหมาะสมของโครงข่ายแบบผสม การวิเคราะห์ผลจะใช้การเปรียบเทียบผลที่ได้จากสมการโมเดลที่พัฒนาขึ้น กับโปรแกรมคอมพิวเตอร์ที่จำลองการทำงานของระบบตามเหตุการณ์ที่เกิดขึ้น การจำลองเหตุการณ์จะใช้วิธีคำนวณแบบวนซ้ำเพื่อให้ได้ผลลัพธ์เข้าใกล้จุดที่ต้องการ

Thesis Title	Analysis and Design of Integrated Terrestrial and Satellite Connectivity Network.
Student	Mr. Seree Assawaruk.
Thesis Advisor	Assist. Prof. Dr.Suvepon Sittichivapak.
Level of Student	Master of Engineering in Electrical Engineering. King Mongkut's Institute of Technology Ladkrabang.
Year	1998

ABSTRACT

Satellite links are characterized large delay and high bandwidth for transmission thus, they significantly affect the overall performance of the network which establishes both satellite and terrestrial connectivity. This cannot be seen obviously when there is only one type of media in the system.

This thesis studies an approach to combine the capabilities of satellite and terrestrial network. Its concept involves combining a medium-capacity satellite network with a relatively thin-line, low-to-medium capacity terrestrial network. This study focuses on the network model in which the satellite channel is used as a multiple access scheme using contention-based reservation protocol. This scheme has been determined earlier but we present herein a new model which employs a simple flow diagram to represent system behavior at the equilibrium point such that small dimension of problem can be attained.

To utilize small earth station or VSAT as the access node in the integrated system, ones must concern current state of the satellite access channel connected by those VSATs. This thesis has studied and analyzed the optimum point of its state that is apt to operate the VSATs in the integrated system.

The utilization of the integrated network can be designed and optimized by two major issues : Traffic assignment and Routing procedure. Based on the well-known algorithms of these issues, this research develops an algorithm in particular, which are essential to designing of

iterative numerical procedure. The method for performance evaluation and analyzation of the network models with and without satellite links can be principally achived by comparing the result of mathematic model and programming simulation which has been written based on event-based simulation concept and to follow the sequence of events constructed on the basis of actual process. This method is presented and the results on the system capability are compared.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จได้ด้วยดี โดยได้รับความกรุณาและคำแนะนำจากหลายฝ่าย ผู้วิจัย ขอขอบพระคุณอย่างสูงต่อทุกท่าน ดังต่อไปนี้

ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ ที่ให้โอกาสในการศึกษาและสนับสนุนให้กำลังใจ ตลอดเวลา อันเป็นแรงบันดาลใจให้ผู้วิจัยมีมานะจนถึงปัจจุบัน

ขอกราบขอบพระคุณ ผศ. ดร. สุวิพล สิริชีวะภาค อาจารย์ที่ปรึกษา ที่กรุณาให้คำปรึกษา แนะนำและให้ข้อมูลที่จำเป็นต่องานวิจัยนี้

ขอขอบพระคุณ ดร. จินดากร ตูจินดา และแผนกวิศวกรรมระบบดาวเทียม บริษัท ชินวัตร แซทเทลไลท์ ทุกๆท่าน ที่ให้คำปรึกษาและความรู้สำหรับระบบดาวเทียม

ขอกราบขอบพระคุณท่านอาจารย์ทุกๆท่าน ที่สั่งสอนและให้ความรู้แก่ผู้วิจัย

ขอขอบคุณ คุณกัลยา จันทร์ชานา และคุณกัลยา นริศรานุกูล ที่ให้ความช่วยเหลือและสนับสนุนอุปกรณ์ในการจัดทำวิทยานิพนธ์

ขอกราบขอบพระคุณทุกท่านที่ไม่ได้กล่าวถึงในที่นี้ที่ได้ให้ความช่วยเหลือและกำลังใจในการวิจัย

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

เสรี อัครวรักษ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	III
กิตติกรรมประกาศ.....	V
สารบัญ.....	VI
สารบัญตาราง.....	VIII
สารบัญภาพ.....	X
บทที่	
1. บทนำ.....	1
ความเป็นมาและขอบเขตของวิทยานิพนธ์.....	1
โครงร่างวิทยานิพนธ์.....	3
2. หลักการทั่วไปของการออกแบบโครงข่ายภาคพื้นดิน.....	4
บทนำ.....	4
หลักการและเงื่อนไขของการวิเคราะห์.....	5
3. การวิเคราะห์และสร้างโมเดลสำหรับระบบดาวเทียมที่ใช้ช่องสัญญาณแบบสุ่ม.....	11
บทนำ.....	11
แบบจำลองระบบดาวเทียม.....	12
การวิเคราะห์สมรรถนะของระบบจากแบบจำลอง.....	15
การวิเคราะห์ผล.....	22
4. การวิเคราะห์และสร้างโมเดลสำหรับสถานีที่ใช้ตัวกลางดาวเทียมในโครงข่ายผสม.....	32
บทนำ.....	33
สมการโมเดลสำหรับคำนวณค่าหน่วงเวลา.....	33
ผลการทดลอง.....	45
สรุปผลการทดลอง.....	46

สารบัญ(ต่อ)

บทที่	หน้า
5. การวิเคราะห์โครงข่ายแบบผสมโดยใช้หลักการพิจารณาปริมาณกราฟิกและเส้นทางผ่านของข้อมูล.....	51
บทนำ.....	51
การกำหนดเส้นทางส่งผ่านข้อมูล.....	51
การคำนวณค่าหน่วงเวลา.....	52
โครงสร้างและหลักการพิจารณาปริมาณกราฟิกภายในโครงข่ายแบบผสม.....	53
ผลการทดลอง.....	55
6. บทสรุป.....	82
บทนำ.....	82
การนำไปประยุกต์ใช้.....	83
แนวทางการศึกษาวิจัยต่อไป.....	83
บรรณานุกรม.....	85
7. ภาคผนวก.....	87
ภาคผนวก ก. ซอฟต์แวร์ระบบ.....	88
ภาคผนวก ข. ตารางแสดงผลการทดลองจากบทที่ 3 และ บทที่ 4.....	121
ภาคผนวก ค. ผลการของทำงานของโปรแกรม.MG_C.....	162
8. ประวัติผู้เขียน.....	172

สารบัญตาราง

ตารางที่	หน้า
1 แสดง เมตริกซ์ของ Cost ของช่องสัญญาณต่างๆ.....	58
2 แสดง ผลการทำงานของโปรแกรม MG_C สำหรับตัวอย่างที่ 5.2.....	67
3 แสดง ผลการทำงานของโปรแกรม MG_C สำหรับตัวอย่างที่ 5.2 ในกรณีที่โครงข่ายมีแต่เฉพาะ สถานีที่ใช้ตัวกลางภาคพื้นดิน.....	71
4 แสดง การเปรียบเทียบผลของช่องสัญญาณดาวเทียมที่ได้จากสมการ โมเดลและการจำลองระบบ โดยใช้โปรแกรมคอมพิวเตอร์ที่อัตราการมาถึงของข้อมูลต่างๆกันจาก บทที่ 3 ที่ค่า Max User = 50.....	121
5 แสดง การเปรียบเทียบผลของช่องสัญญาณดาวเทียมที่ได้จากสมการ โมเดลและการจำลองระบบ โดยใช้โปรแกรมคอมพิวเตอร์ที่อัตราการมาถึงของข้อมูลต่างๆกันจากบทที่ 3 ที่ค่า Max User = 75.....	122
6 แสดง การเปรียบเทียบผลของช่องสัญญาณดาวเทียมที่ได้จากสมการ โมเดลและการจำลองระบบโดยใช้โปรแกรมคอมพิวเตอร์ที่อัตราการมาถึงของข้อมูลต่างๆกันจากบทที่ 3 ที่ค่า Max User = 100.....	124
7 แสดง การเปรียบเทียบผลของช่องสัญญาณดาวเทียมที่ได้จากสมการ โมเดลและการจำลองระบบโดยใช้โปรแกรมคอมพิวเตอร์ที่อัตราการมาถึงของข้อมูลต่างๆกันจากบทที่ 3 ที่ค่า Max User = 125.....	125
8-20 แสดง ผลการจำลองในบทที่ 4 โดยใช้โปรแกรมเพื่อหาค่าหาช่วงเวลาโดยเฉลี่ยของข้อมูล ที่เกิดขึ้นในสถานีดาวเทียมที่ใช้ระบบดาวเทียมที่ User = 50 และที่อัตราการมาถึงของข้อมูลตั้งแต่ 0.005-0.08 แพคเกต/เฟรม.....	127-134
21-33 แสดง ผลการจำลองในบทที่ 4 โดยใช้โปรแกรมเพื่อหาค่าหาช่วงเวลาโดยเฉลี่ยของข้อมูล ที่เกิดขึ้นในสถานีดาวเทียมที่ใช้ระบบดาวเทียมที่ User = 75 และที่อัตราการมาถึงของข้อมูลตั้งแต่ 0.005-0.08 แพคเกต/เฟรม.....	134-141
34-47 แสดง ผลการจำลองในบทที่ 4 โดยใช้โปรแกรมเพื่อหาค่าหาช่วงเวลาโดยเฉลี่ยของข้อมูล ที่เกิดขึ้นในสถานีดาวเทียมที่ใช้ระบบดาวเทียมที่ User = 100 และที่อัตราการมาถึงของข้อมูลตั้งแต่ 0.005-0.08 แพคเกต/เฟรม.....	141-149

สารบัญตาราง(ต่อ)

ตารางที่

หน้า

48-62 แสดง ผลการจำลองในบทที่ 4 โดยใช้โปรแกรมเพื่อหาค่าหน่วยเวลาโดยเฉลี่ย ของ
ข้อมูล ที่เกิดขึ้นในสถานีดาวเทียมที่ใช้ระบบดาวเทียมที่ User = 125 และที่อัตราการ
มาถึงของข้อมูลตั้งแต่ 0.005-0.1 แพคเกจ/เฟรม.....150-167

สารบัญญภาพ

หน้า

1 แสดง Convexity Property.....	7
2 แสดงผลรวมของข้อมูลในเส้นทางที่กำหนด.....	7
3(ก) แสดง โครงข่ายวีแซท ช่องสัญญาณขาออก.....	13
3(ข) แสดง โครงข่ายวีแซท ช่องสัญญาณขาเข้า.....	13
4(ก) แสดง โครงสร้างของแต่ละเฟรมของช่องสัญญาณดาวเทียม.....	15
4(ข) แสดง รูปแบบการทำงานของเฟรมย่อยอะโลฮาและเฟรมย่อยสำหรับการจอง.....	15
5 แสดง แบบจำลองสถานะของระบบ.....	17
6 แสดง ช่วงความน่าจะเป็นสำหรับการส่งแพคเกจในแต่ละเฟรม.....	18
7 แสดง แผนภูมิของโปรแกรม SIMSAT.....	24
8 (ก) แสดง ค่าทฤษฎีเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=50$	25
8 (ข) แสดง ค่าทฤษฎีเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=75$	25
8 (ค) แสดง ค่าทฤษฎีเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=100$	26
8 (ง) แสดง ค่าทฤษฎีเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=125$	26
9 (ก) แสดง ค่าหน่วงเวลาเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=50$	27
9 (ข) แสดง ค่าหน่วงเวลาเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=75$	27
9 (ค) แสดง ค่าหน่วงเวลาเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=100$	28
9 (ง) แสดง ค่าหน่วงเวลาเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=125$	28
10(ก) แสดง ค่าเฉลี่ยของค่าหน่วงเวลาเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=50$	29
10(ข) แสดง ค่าเฉลี่ยของค่าหน่วงเวลาเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=75$	29
10(ค) แสดง ค่าเฉลี่ยของค่าหน่วงเวลาเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=100$	29
10(ง) แสดง ค่าเฉลี่ยของค่าหน่วงเวลาเปรียบเทียบกับ λ_d ที่ $w=3, L=24, A=0.5$ เมื่อ $N=125$	30

สารบัญภาพ(ต่อ)

หน้า

11 แสดง ค่าหน่วยเวลาที่อัตราส่วน(L_r)ต่างๆกันของสล็อตอะ โลฮา และสล็อตสำหรับการจอง ที่ $\lambda_d=0.05$	30
12 แสดง แผนภูมิจำลองการทำงานของโปรแกรม SIMSAT.....	47
13(ก) แสดง การเปรียบเทียบผลจากสมการโมเดลกับ โปรแกรมคอมพิวเตอร์ที่ภาระ(load) ต่างๆ กัน ที่ N=50.....	48
13(ข) แสดง การเปรียบเทียบผลจากสมการโมเดลกับ โปรแกรมคอมพิวเตอร์ที่ภาระ(load) ต่างๆ กัน ที่ N=75.....	48
13(ค) แสดง การเปรียบเทียบผลจากสมการโมเดลกับ โปรแกรมคอมพิวเตอร์ที่ภาระ(load) ต่างๆ กัน ที่ N=100.....	49
13(ง) แสดง การเปรียบเทียบผลจากสมการโมเดลกับ โปรแกรมคอมพิวเตอร์ที่ภาระ(load) ต่างๆ กัน ที่ N=125.....	49
14(ก), (ข) แสดง ค่าความแปรปรวนที่ได้จากโปรแกรม SIMSAT ที่อัตราการมาถึงของแพค เกต ต่างๆกัน.....	50
15 แสดง โครงข่ายสำหรับตัวอย่างที่ 5.1.....	57
16 แสดง โครงข่ายสำหรับตัวอย่างที่ 5.2.....	66
17 แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 8 และ 9 เทียบกับ Link Speed.....	70
18 แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 1 และ 18 เทียบกับ Link Speed.....	70
19 แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 5 และ 6 เทียบกับ Link Speed.....	71
20 แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 8 และ 9 เทียบกับ Link Speed.....	74
21 แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 1 และ 18 เทียบกับ Link Speed.....	75
22 แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 5 และ 6 เทียบกับ Link Speed.....	75
23 แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 8 และ 9 เทียบกับ Link Speed.....	76
24 แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 1 และ 18 เทียบกับ Link Speed.....	76
25 แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 5 และ 6 เทียบกับ Link Speed.....	77

บทที่ 1

บทนำ

ความเป็นมาและขอบเขตของวิทยานิพนธ์

ในปัจจุบันระบบดาวเทียมมีการใช้งานแพร่หลายมากขึ้นในประเทศไทย ทำให้ลัดข้อจำกัดทางด้านตัวกลางสำหรับการติดต่อไปยังสถานที่ต่างๆในประเทศ ส่งผลให้เกิดการขยายตัวของโครงข่ายทั้งที่มีอยู่เดิมและโครงข่ายที่เกิดขึ้นใหม่ รูปแบบของโครงข่ายมีความซับซ้อนเพิ่มขึ้น แต่งานวิจัยที่ศึกษาเกี่ยวกับการวิเคราะห์หรือออกแบบโครงข่ายทั้งภาคพื้นดินและดาวเทียม มักจะศึกษาและวิจัยแต่ละชนิดแยกต่างหากจากกัน หรือกำหนดให้ช่องสัญญาณที่เชื่อมต่อโดยตัวกลางดาวเทียมเป็นแบบสถานีถึงสถานี(Point-to-Point) ทั้งนี้เพื่อหลีกเลี่ยงข้อจำกัด ซึ่งจำกัดประสิทธิภาพโดยรวมของกันและกัน

เมื่อก้าวถึงโครงข่ายคอมพิวเตอร์จะหมายถึงการเชื่อมต่อสถานีหลายๆจุดเข้าด้วยกันด้วยเคเบิลชนิดต่างๆ ทั้งในรูปแบบของเครือข่ายสาธารณะ(Public Network) และคู่สายเช่า(Lease Circuit) โครงสร้างจะมีความซับซ้อนและประสิทธิภาพที่แตกต่างกันขึ้นอยู่กับโครงสร้างของโครงข่ายนั้นๆ เนื่องจากช่องสัญญาณระหว่างสถานีมีขนาดที่แตกต่างกันตามความต้องการและสถานีที่มีเป็นจำนวนมากในโครงข่าย ทำให้การจัดสรรตัวกลางในการส่งข้อมูลเพื่อรองรับองค์ประกอบอย่างเหมาะสมและมีประสิทธิภาพสูงสุดมีความจำเป็นอย่างยิ่ง โดยเฉพาะโครงข่ายที่จำเป็นต้องใช้ตัวกลางที่เป็นดาวเทียมร่วมอยู่ด้วย ทั้งนี้อาจใช้สำหรับแก้ปัญหาที่เคเบิลภาคพื้นดินไม่สามารถไปถึงได้ในบางพื้นที่ หรือต้องการใช้เส้นทางนั้นอย่างเร่งด่วน

การเลือกใช้ชนิดของช่องสัญญาณดาวเทียมสามารถเลือกได้หลายลักษณะที่เหมาะสมกับระบบของตน เช่น ในกรณีที่ระบบต้องการเวลาตอบรับ(Response)น้อยมาก และช่องสัญญาณต้องการความมั่นคงและเสถียรภาพ(Stability)ที่ค่อนข้างสูง สามารถเลือกใช้ระบบดาวเทียมแบบ Point-to-Point ซึ่งผู้ใช้เป็นเจ้าของช่องสัญญาณดาวเทียมดังกล่าว แต่ผู้เดียวทั้งในรูปของ ความถี่(Band width) หรือ ช่วงเวลา(Slot) ในขณะที่เดียวกัน ถ้าระบบของผู้ใช้มีผลกระทบจากคุณสมบัติดังกล่าวค่อนข้างน้อย อาจเลือกใช้ระบบที่มีลักษณะที่ ผู้ใช้สามารถเข้าใช้ช่องสัญญาณเดียวกันได้หลายคน

(Multi Access)

ในงานวิจัยนี้เป็นการวิเคราะห์เพื่อออกแบบโครงข่ายที่มีการใช้ตัวกลางที่เป็นเคเบิลภาคพื้นดินร่วมกับตัวกลางที่เป็นช่องสัญญาณดาวเทียม และกำหนดให้ช่องสัญญาณภาคพื้นดินที่อยู่ในโครงข่ายเป็นแบบวงจรเช่า(Lease Circuit)ที่อัตราเร็วต่างๆกัน ในขณะที่ช่องสัญญาณดาวเทียมเป็นแบบ Multiple Access โดยสมมติให้ผู้บริหารโครงข่ายที่ปัจจุบัน ภายในโครงข่ายมีเฉพาะสถานี(ในกรณีที่ต้องการกล่าวถึงสถานีหรืออุปกรณ์เฉพาะในสถานีนั้นๆ ที่เชื่อมต่อหรือแลกเปลี่ยนข้อมูลโดยตรงกับเครือข่ายและมีฟังก์ชันในระดับโปรโตคอล 1-4 จะเรียกในลักษณะทั่วไปว่า โหนด) ที่ใช้ช่องสัญญาณภาคพื้นดิน ต้องการสถานีที่ใช้ช่องสัญญาณดาวเทียมจำนวนหนึ่ง สำหรับแก้ปัญหาที่กล่าวถึงข้างต้น โดยระบบของตนไม่วิกฤตต่อเวลาตอบรับ(Response Time) และปริมาณทราฟฟิกต่อช่วงเวลาค่อนข้างต่ำ สถานีดังกล่าวเป็นสถานีที่ใช้ช่องสัญญาณดาวเทียมร่วมกับสถานีดาวเทียมอื่นๆ ซึ่งไม่เกี่ยวข้องข้อกับโครงข่ายแบบผสมที่กำลังพิจารณา สถานีที่ใช้ช่องสัญญาณดาวเทียมทั้งหมดถูกบริหารโดยผู้ให้บริการวีแซท(VSAT Operator) ที่ผู้บริหารโครงข่ายแบบผสมเลือกใช้บริการ ในประเทศไทยมีผู้ให้บริการวีแซทอยู่จำนวนหนึ่ง ระบบช่องสัญญาณดาวเทียมของแต่ละผู้ให้บริการมีรูปแบบพื้นฐานที่ใกล้เคียงกัน ผู้บริหารโครงข่ายควรจะต้องศึกษาแต่ละรูปแบบโดยคร่าว เพื่อให้สามารถเลือกใช้งานตามความเหมาะสมกับระบบของตน

วิธีการออกแบบและเลือกใช้นาความจุของช่องสัญญาณทั้งแบบดาวเทียมและภาคพื้นดิน ที่นำเสนอในวิทยานิพนธ์ฉบับนี้สามารถนำไปประยุกต์ใช้กับระบบดาวเทียมของผู้ให้บริการวีแซทแต่ละรายที่มีอยู่ในประเทศไทยโดยการพิจารณาโครงสร้างของการเข้าใช้(Access)ช่องสัญญาณดาวเทียมแต่ละแบบซึ่งมีโครงสร้างพื้นฐานเดียวกัน และดัดแปลงสมการโมเดลหรือโปรแกรมจำลองการทำงานให้สอดคล้องกับรูปแบบและฟังก์ชันของแต่ละชนิดของช่องสัญญาณดาวเทียมเพื่อการออกแบบที่เหมาะสมต่อไป

โครงร่างวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วย

บทคัดย่อและ Abstract

บทที่ 1 กล่าวถึงบทนำ

บทที่ 2 กล่าวถึงหลักการทั่วไปของการออกแบบ โครงข่ายภาคพื้นดิน

บทที่ 3 กล่าวถึงการวิเคราะห์และสร้าง โมเดลสำหรับระบบดาวเทียมที่ใช้ช่องสัญญาณแบบสุ่ม

บทที่ 4 กล่าวถึงการวิเคราะห์และสร้าง โมเดลสำหรับสถานีที่ใช้ตัวกลางดาวเทียมใน โครงข่ายแบบผสม

บทที่ 5 กล่าวถึงการวิเคราะห์โครงข่ายแบบผสม โดยใช้หลักการพิจารณาปริมาณกราฟฟิกและเส้นทางผ่านของข้อมูล

บทที่ 6 กล่าวถึงการสรุปผลการวิจัย การนำไปประยุกต์ใช้งาน ปัญหาที่พบ และแนวทางการวิจัยต่อ

บทที่ 2

หลักการทั่วไปของการออกแบบโครงข่ายภาคพื้นดิน

บทนำ

โดยทั่วไปเมื่อก้าวถึงโครงข่ายคอมพิวเตอร์จะหมายถึงการเชื่อมต่อคอมพิวเตอร์หลายๆ เครื่องเข้าด้วยกัน โดยมีโครงสร้าง ความซับซ้อนและประสิทธิภาพที่แตกต่างกัน ข้อมูลที่เดินทางอยู่ภายในโครงข่ายจะมีขนาดที่แตกต่างกันเช่น ข้อมูลคำสั่ง(Command) ข้อมูลเพื่อสอบถาม(Inquiry) หรือ ข้อมูลเพื่อร้องขอ(Request) ข้อมูลเหล่านี้มีขนาดของข้อมูลประมาณ 50-200 bytes และข้อมูลในลักษณะของ File Transfer ที่มีขนาดของข้อมูลมากกว่า 1000 bytes เป็นต้น เนื่องจากข้อมูลที่มีขนาดที่แตกต่างกันและคอมพิวเตอร์ที่มีจำนวนมากในโครงข่าย ทำให้การจัดสรรตัวกลางในการส่งข้อมูลเพื่อรองรับองค์ประกอบดังกล่าวอย่างเหมาะสมและมีประสิทธิภาพสูงสุดมีความจำเป็นอย่างยิ่ง โครงข่ายของการสื่อสารข้อมูลสามารถแบ่งออกเป็นสี่รูปแบบใหญ่ๆคือ เซอร์กิตสวิตซิง(Circuit Switching) เมสเสจสวิตซิง(Message Switching) แพคเกจสวิตซิง(Packet Switching) เซลล์สวิตซิง(Cell Switching)

เซอร์กิตสวิตซิง(Circuit Switching) เป็นการให้บริการโดยสร้างเส้นทางทั้งหมดที่ต้องเชื่อมต่อระหว่างต้นทาง(Source)และปลายทาง(Destination) ผู้ใช้จะเริ่มส่งข้อมูลระหว่างกันได้ก็ต่อเมื่อเส้นทางทั้งหมดได้มีการจัดตั้งอย่างสมบูรณ์และจะคงอยู่จนกระทั่งสิ้นสุดการใช้งาน

เมสเสจสวิตซิง(Message Switching) เป็นการให้บริการโดยการสร้างเส้นทางเชื่อมต่อระหว่างโนดหนึ่งไปยังอีกโนดหนึ่ง ข้อมูลที่ถูกส่งในเส้นทางดังกล่าวเมื่อไปถึงโนดถัดไปครบทั้งข้อมูล โหนดนั้นจะจัดส่งไปยังโนดต่อไปโดยเลือกเส้นทางที่ดีที่สุด ถ้าเส้นทางที่ถูกเลือกกำลังถูกใช้งานหรือไม่ว่าง ข้อมูลดังกล่าวจะถูกเก็บอยู่ในหน่วยความจำชั่วคราวเพื่อรอให้เส้นทางว่างและทำการส่งข้อมูลออกในลักษณะเรียงตามลำดับ(First In First Out : FIFO)ต่อไป . ขั้นตอนดังกล่าว

ดำเนินไปจนกระทั่งข้อมูลนั้นไปถึงปลายทางที่ต้องการ แพคเกจสวิตซิง(Packet Switching) จะได้กล่าวโดยละเอียดต่อไป เซลสวิตซิง(Cell Switching) ข้อมูลจะถูกแบ่งออกเป็นเซลล์ที่มีขนาดเล็กๆ ที่มีขนาดคงที่ เนื่องจากแต่ละเซลล์มีขนาดเล็กการจัดสรรองค์ประกอบของแต่ละช่องสัญญาณ เช่น อัตราเร็วของข้อมูลหรือช่องสัญญาณที่สร้างขึ้นชั่วคราว(Virtual Circuit) หรือถาวร ชนิดของข้อมูล สำหรับแต่ละรูปแบบ(Application)ของผู้ใช้บริการทำได้ดีกว่า ทำให้การใช้งานตัวกลางมีประสิทธิภาพที่สูงสุดแต่มีข้อเสียที่เด่นชัดคือ ข้อมูลที่ถูกเพิ่มเข้าไปเป็นส่วนหัว(Head)ในแต่ละเซลล์ต้องมีจำนวนที่น้อยที่สุด ทำให้ต้องลดรายละเอียดที่ต้องการสำหรับการให้บริการภายในโครงข่ายลง รายละเอียดดังกล่าวเช่น ข้อมูลกำหนดการไหล(Flow Control) ข้อมูลในการตรวจสอบความผิดพลาดเป็นต้น รวมทั้งต้องการตัวกลางที่มีขนาดความจุที่สูง ในขณะที่ต้องมีค่าความผิดพลาดของข้อมูล(Bit Error Rate)ต่ำ

หลักการและเงื่อนไขของการวิเคราะห์

ในงานวิจัยนี้ ได้ทำการวิเคราะห์โครงข่ายที่ใช้โปรโตคอลแบบ แพคเกจสวิตซิง(Packet Switching)ซึ่งเป็น โครงสร้างที่มีการใช้งานจริงอยู่ในปัจจุบันรวมทั้งในประเทศไทย วิธีการนี้กล่าวโดยย่อคือการนำข้อมูลมาทำเป็นแพคเกจที่มีขนาดคงที่ ข้อมูลของผู้ใช้(user)ที่มีขนาดเกินกว่าพื้นที่ข้อมูลในแต่ละแพคเกจจะถูกแบ่งออกเป็นส่วนๆ เพื่อให้สามารถจัดสรรลงในแต่ละแพคเกจได้ ข้อมูลในแต่ละแพคเกจจะถูกเพิ่มข้อมูลส่วนหัว(Head)และข้อมูลส่วนท้าย(Tail) ซึ่งบรรจุรายละเอียด ของสถานีต้นทาง สถานีปลายทาง รวมทั้งข้อมูลที่ใช้ตรวจสอบและแก้ไขข้อผิดพลาด ดังนั้นข้อมูลในลักษณะแพคเกจที่ถูกส่งไปยังปลายทางต่างๆกันสามารถใช้งานตัวกลางร่วมกันได้ สถานีหรือโหนดหนึ่งๆจะมีอุปกรณ์ที่เรียกว่า แพคเกจสวิตซิงโหนด ทำหน้าที่แปลงข้อมูลจากเทอร์มินอล ของผู้ใช้ให้อยู่ในรูปของแพคเกจ คัดเลือกแพคเกจที่มีเลขหมายปลายทางเป็นของตนเองและส่งผ่านแพคเกจที่มีเลขหมายอื่นไปยังเส้นทางที่เหมาะสมในโครงข่าย แต่ละแพคเกจที่มีต้นทางและปลายทางเดียวกันอาจเดินทางในเส้นทางที่แตกต่างกันได้มากกว่าหนึ่งเส้นทาง ทำให้การใช้ตัวกลางสำหรับการส่งข้อมูล (Transmission Link) เป็นไปอย่างมีประสิทธิภาพ

การสร้างโมเดลสำหรับใช้ในการวิเคราะห์โครงข่ายแบบแพคเกจสวิตซิง มีพื้นฐานขึ้นอยู่กับ การพิจารณาคุณสมบัติและฟังก์ชันในระบบ เช่น กำหนดช่วงเวลา ค่าทราฟฟิค ขนาดของบัฟเฟอร์ที่สอดคล้องกับกำหนดช่วงเวลาและค่าทราฟฟิค การจัดลำดับแพคเกจที่ได้รับที่ปลายทางที่ต้องการ ชนิดของตัวกลางที่ใช้ในแต่ละสถานี ความจุของแต่ละช่องสัญญาณที่มีอยู่ในแต่ละสถานี การแก้ไขค่าผิด

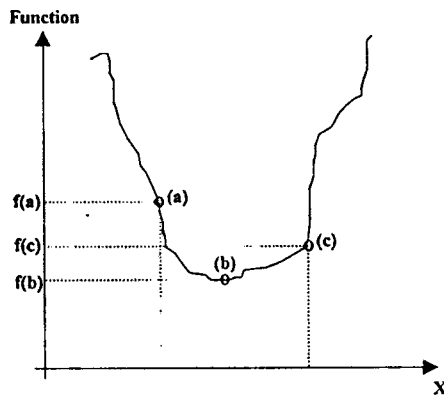
พลาดที่เกิดขึ้น การเชื่อมต่อเส้นทางจากโหนดหนึ่งไปยังอีกโหนดหนึ่ง การควบคุมการไหลของแพ็คเกจ ภายในโครงข่ายเป็นต้น ปัจจุบันได้มีการศึกษาในหลากหลายรูปแบบของการเชื่อมต่อเส้นทาง และวิธีคำนวณหาปริมาณการไหลของแพ็คเกจที่เหมาะสมในแต่ละเส้นทางในโครงข่าย ในงานวิจัยนี้ได้นำเสนอโครงสร้างของการเชื่อมต่อเส้นทางแบบ Dijkstra (Dijkstra's Shortest Path Algorithm) [Dimitri, 1992:401-403][Aaron, 1993:157-159] และการกำหนดปริมาณการส่งผ่านข้อมูล(Flow Diviation Algorithm)[Dimitri, 1992:458-464][Aaron, 1993:252-255] มาพัฒนาและสร้างเป็นโมเดลทางคณิตศาสตร์ที่เหมาะสมและสอดคล้องกับอัลกอริทึมทั้งสองเพื่อการคำนวณโครงข่ายที่มีการใช้ตัวกลางดาวเทียมและภาคพื้นดินร่วมกัน ทั้งนี้จะอธิบายโครงสร้าง เงื่อนไขและรูปแบบของอัลกอริทึมที่จะนำมาพัฒนาทั้งสองโดยย่อ ผู้สนใจสามารถค้นคว้าเพิ่มเติมได้จากเอกสารอ้างอิงที่ได้แสดงไว้

ภายในโครงข่ายที่ประกอบไปด้วยสถานีหลายแห่งที่ส่งข้อมูลติดต่อกัน พิงค์ชันและเหตุการณ์ถูกส่งผ่านจากสถานีหนึ่งไปยังอีกสถานีหนึ่งจนถึงปลายทาง กล่าวอีกนัยหนึ่งคือเหตุการณ์ที่เกิดในสถานีหนึ่งอาจส่งผลต่อไปยังอีกสถานีหนึ่งได้ในบางกรณี ดังนั้นเพื่อความสะดวกในการวิเคราะห์กำหนดช่วงเวลาและพารามิเตอร์อื่นๆที่เกี่ยวข้อง การสร้างโมเดลในงานวิจัยนี้จะอยู่ภายใต้เงื่อนไข ต่อไปนี้

- 1) สถานีทุกสถานีมีช่องสัญญาณหรือเส้นทางของข้อมูลทั้งขาเข้าและขาออกมากกว่าหนึ่งเส้นทาง
- 2) กำหนดให้ข้อมูลที่เกิดขึ้นหรือมาถึงในแต่ละสถานี มีรูปแบบของการเปลี่ยนแปลงเป็นไปตามการกระจายแบบปัวซอง(Poisson Distribution)[Klienrock,1975 : 60-64] สำหรับสถานีที่ใช้โครงข่ายภาคพื้นดินและสถานีที่ใช้โครงข่ายดาวเทียม
- 3) กำหนดให้เวลาของการให้บริการ(Service Time) มีการเปลี่ยนแปลงเป็นไปตามการกระจายแบบเอ็กโปเนนเชียล(Exponential Distribution)[Hisashi,1978 : 53-54] สำหรับสถานีที่ใช้ตัวกลางภาคพื้นดินและเปลี่ยนแปลงเป็นไปตามการกระจายแบบทั่วไป (General Distribution)[Hisashi,1978 : 188-193] สำหรับสถานีที่ใช้โครงข่ายดาวเทียม
- 4) กำหนดเวลาที่เกิดขึ้นในแต่ละสถานีกำหนดให้เป็นอิสระต่อกัน
- 5) ผลลัพธ์ที่ได้จากการคำนวณ ณ จุดพิจารณาใดๆ ภายในอาณาบริเวณที่กำหนด จะเป็นผลลัพธ์ของอาณาบริเวณทั้งหมดด้วย (Convexity Property) ดังตัวอย่างในรูปที่ 1 เมื่อทำการคำนวณเพื่อหาผลลัพธ์จากอาณาบริเวณหนึ่ง(Area $X:X=[a,b,c,\dots]$)โดยพิจารณาตามแนวโน้มที่ลดลงจากจุดใดจุดหนึ่งในขอบเขตที่กำหนด(จุด (a) ในรูปที่ 1)

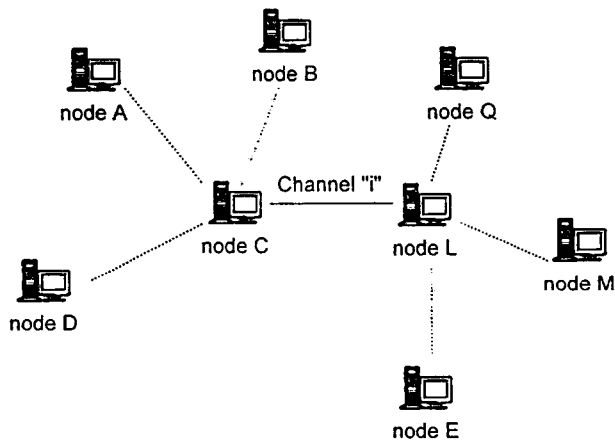
จนพบค่าต่ำสุด(จุด (b) ในรูปที่ 1) ค่าต่ำสุดดังกล่าวจะเป็นค่าต่ำสุดของอาณาบริเวณนั้นทั้งหมด ตัวอย่างของฟังก์ชันดังกล่าวเช่น a, b เป็นปริมาณกราฟฟิคที่มีปลายทางแตกต่างกันที่เกิดขึ้นหรือเดินทางผ่าน โหนดใดๆ ในเครือข่าย

ภาพที่ 1



แสดง Convexity Property

ภาพที่ 2



แสดง ช่องสัญญาณหรือเส้นทางผ่านของข้อมูลจาก โหนดต่างๆ ในโครงข่าย

- 6) กำหนดให้ให้ข้อมูลที่เกิดภายในโครงข่ายเช่นข้อมูลที่ใช้ในการตรวจสอบช่องสัญญาณ ข้อมูลที่ใช้ในการเชื่อมต่อช่องสัญญาณ ข้อมูลที่ใช้ในการตรวจสอบเส้นทาง ข้อมูลที่ใช้สำหรับโพลคอนโทรล มีจำนวนน้อยเมื่อเทียบกับจำนวนข้อมูลทั้งหมดที่ปรากฏในแต่ละช่องสัญญาณภายในโครงข่าย

การวิเคราะห์ค่าหน่วงเวลา

สมมติให้ $j = A, B$ และ $k = M, E, Q$ ตามลำดับ กำหนดให้ช่องสัญญาณ i เป็นช่องสัญญาณจากโนดหนึ่งใดๆไปยังอีกโนดหนึ่งใดๆที่อยู่ติดกัน ดังแสดงในภาพที่ 2 ดังนั้นอัตราการไหลของแพคเกจที่เข้าสู่ช่องสัญญาณ i ดังกล่าวจะเท่ากับ อัตราการไหลของข้อมูลทั้งหมดที่เข้าสู่โนด $j (j \in [A, B])$ และมีปลายทางไปยังโนด $k (k \in [M, E, Q])$ ซึ่งจะเรียกว่าเส้นทาง jk และช่องสัญญาณ i เป็นส่วนหนึ่งของเส้นทาง jk

กราฟฟิคที่เดินทางจากต้นทางไปยังปลายทางจะผ่านช่องสัญญาณ i ดังนั้นเมื่อพิจารณาที่ช่องสัญญาณ i อัตราโดยเฉลี่ยของแพคเกจที่ไหลผ่านช่องสัญญาณ i จะได้เท่ากับผลรวมของอัตราโดยเฉลี่ยของแพคเกจที่ไหลเข้าสู่เส้นทางทุกเส้นทางที่มีองค์ประกอบเป็นช่องสัญญาณ i อยู่ในเส้นทางนั้น กำหนดให้แทนด้วย R_i ถ้าให้ N คือสถานีหรือโนดใดๆในโครงข่าย เราจะได้

$$R_i = \sum_j \sum_k r_{jk} \quad (2.1)$$

โดยที่ $j, k \in N$ และ

r_{jk} : ปริมาณของแพคเกจที่เกิดจากโนดต้นทาง j และมีปลายทางเป็นโนด k

N : จำนวนของสวิตชิงโนดทั้งหมดในเครือข่าย

จากข้อกำหนดเราสามารถพิจารณาช่องสัญญาณใดๆในโครงข่ายด้วยโมเดลแบบ $M/M/1$ ทำให้ได้สมการในรูปของค่าหน่วงเวลา(t_j) ที่เกิดขึ้นเนื่องจากแพคเกจได้รับการบริการ(Service) ในคิวเท่ากับ

$$t_i = \frac{1}{uC_i - R_i} \quad (2.2)$$

โดยที่

C_i = อัตราเร็วของช่องสัญญาณ (bits/sec)

R_i = อัตราโดยเฉลี่ยของแพคเกจที่ไหลผ่านช่องสัญญาณ i (packets/sec)

u = จำนวนของบิตในแต่ละแพคเกจ

ค่าหน่วยเวลาโดยเฉลี่ยที่แพคเกจใดๆเดินทางอยู่ในโครงข่ายจะเท่ากับ

$$T = P_{pc} + \sum_{i=1}^M \frac{R_i}{\lambda} \left[t_i + \frac{1}{uC_i} + P_{pc} + P_d \right] \quad (2.3)$$

โดยที่

P_{pc} = เวลาที่ใช้ในการประมวลผล(Processing Time)

P_d = เวลาที่ใช้เดินทางในตัวกลาง(Propagation Delay)

λ = อัตราของทราฟฟิคจากภายนอกทั้งหมดที่เข้าสู่โครงข่าย

เพื่อความสะดวกในการคำนวณจะแยกตัวแปรที่คงที่ และไม่มีผลต่อสถานะของระบบเพื่อนำมารวม

ในภายหลังและจัดรูปแบบที่เหมาะสม ดังนั้นสมการที่(2.3) จะเปลี่ยนรูปแบบเป็น

$$T = \sum_{i=1}^M \frac{R_i}{\lambda} \left[\frac{1}{uC_i - R_i} \right] \quad (2.4)$$

ถ้าให้ $f = \frac{R_i}{u}$ จะได้สมการ(2.4) ในรูปของ

$$T = \sum_{i=1}^M \frac{1}{\lambda} \left[\frac{f_i}{C_i - f_i} \right] \quad (2.5)$$

สมการที่(2.5) จะอยู่ภายใต้เงื่อนไขที่ได้กล่าวมา โดยเฉพาะความเป็นอิสระของแต่ละสถานี และ Convexity Property ซึ่งสามารถนำมาใช้ในการพิจารณาจุดทำงานที่เหมาะสมของระบบได้จาก

คุณสมบัติดังกล่าว สามารถหาค่าหน่วยเวลาโดยเฉลี่ยของโครงข่ายที่จุดต่ำสุด(ในขณะที่อัตราของแพคเกจข้อมูลสอดคล้องกับความจุของช่องสัญญาณ) จากสมการ

$$\frac{dT}{df} = \frac{1}{\lambda} \cdot \left(\frac{C_i}{(C_i - f_i)^2} \right) \quad i = 1, 2, 3, \dots, n \quad (2.6)$$

สมการที่ (2.5) และ (2.6) สามารถใช้เป็นสมการเริ่มต้นรวมทั้งนำไปประยุกต์ใช้ สำหรับคำนวณค่าหน่วยเวลาที่เกิดขึ้นในระบบโครงข่ายแบบผสมที่จะได้กล่าวถึงโดยละเอียดในบทต่อไป

บทที่ 3

การวิเคราะห์และสร้างโมเดลสำหรับระบบดาวเทียมที่ใช้ช่องสัญญาณแบบสุ่ม

บทนำ

ในบทนี้ นำเสนอการสร้างแบบจำลองสำหรับการวิเคราะห์สมรรถนะของช่องสัญญาณดาวเทียมแบบสุ่มที่มีโปรโตคอลที่สามารถทำการจองช่องสัญญาณล่วงหน้าโดยผู้ใช้หลายคนได้ จุดประสงค์ของแบบจำลองนี้คือสามารถพัฒนาเป็นโมดูลที่เชื่อมต่อกับโมดูลของโครงข่ายภาคพื้นดินได้โดยสะดวกสำหรับการจำลองระบบโครงข่ายแบบผสม พฤติกรรมของระบบดาวเทียมจะแทนด้วยแบบจำลองการไหลของแพคเกจที่ไม่ซับซ้อน พารามิเตอร์สำหรับพิจารณาสมรรถนะได้จากการคำนวณที่ภาวะสมดุล การวิเคราะห์สมรรถนะ(Performance)พิจารณาจากค่าหนึ่งช่วงเวลาและทราฟฟิค(Throughput)โดยเปรียบเทียบระหว่างผลที่ได้จากการคำนวณกับผลที่ได้จากการจำลองการทำงานจากระบบด้วยโปรแกรมคอมพิวเตอร์เพื่อแสดงจุดการทำงานที่เหมาะสม

การวิเคราะห์โครงข่ายที่มีโปรโตคอลหลายๆแบบและใช้ตัวกลางทั้งดาวเทียมร่วมกับเคเบิลภาคพื้นดิน จำเป็นต้องคำนึงถึงผลที่เกิดจากการส่งข้อมูลผ่านดาวเทียม เนื่องจากตัวกลางดาวเทียมมีข้อจำกัดของค่าหนึ่งช่วงเวลาและแบนด์วิดท์(Bandwidth)ที่จะต้องออกแบบให้เหมาะสมต่อระบบโดยรวมและค่าใช้จ่ายเมื่อเทียบกับสมรรถนะ(Performance)ที่ได้รับ ข้อมูลที่ได้จากการวิเคราะห์เหล่านี้ นอกเหนือจากการใช้สำหรับออกแบบ ยังสามารถใช้เป็นข้อมูลสำหรับการจัดการ(System Management) ภายหลังเมื่อมีการใช้งานในอนาคต

โปรโตคอลของระบบดาวเทียมที่นำมาเป็นต้นแบบของการพิจารณาผลกระทบของระบบดาวเทียมที่มีต่อระบบโดยรวมนี้ได้มีการแนะนำวิธีการทำงานโดยย่อใน[A. Fujii, 1986:1464-1499] และ X.Wang[1991] ได้ทำการศึกษาและวิเคราะห์การทำงานของโมเดลดังกล่าวด้วย Catastrophe Theory (กำหนดลักษณะของพฤติกรรมของระบบด้วย Cusp Catastrophe) Wang ทำการวิเคราะห์ระบบโดยใช้ค่าทราฟฟิค(Throughput)ที่อัตราการส่งและเงื่อนไขที่สภาพแวดล้อมต่างๆกันเป็นตัวแปร(Parameter)หลักในการพิจารณา แบบจำลองพฤติกรรมของระบบที่สร้างจาก Catastrophe Theory จะประกอบไปด้วยตัวแปรที่เกี่ยวข้องทุกตัวแปรที่มีอยู่ในระบบ รวมทั้งกำหนดจุดทำงาน

ของตัวแปรที่ไม่เป็นอิสระต่อกัน ดังนั้นเพื่อลดความซับซ้อนและใช้ตัวแปรหลักที่เหมาะสม ในบทนี้จะนำเสนอแบบจำลองอีกลักษณะหนึ่งภายใต้ข้อกำหนดของ Fujii และ Wang แต่จะใช้ค่าหน่วยเวลา ซึ่งจำเป็นสำหรับการออกแบบโครงข่ายแบบผสมเป็นตัวแปรหลักในการพิจารณา และได้นำหลักการเบื้องต้นของขบวนการเกิดและตาย(Birth-and-Death Process)[Hisahsi1978:129-144] สร้างเป็นไดอะแกรมจำลองการไหลของแพคเกจ

การวิเคราะห์สมรรถนะของระบบดาวเทียมทำได้หลายวิธีเช่นการใช้ช่วงเวลาที่ระบบปรับตัวจากสถานะโอเวอร์โหลดกลับเป็นสภาวะปกติเป็นพารามิเตอร์ในการพิจารณา[D.Raychaudhuri 1990]และการเปรียบเทียบระหว่างจำนวนสถานีที่ต้องส่งซ้ำในขณะนั้นกับค่าทฤษฎีเป็นต้น แต่ในบทวิจัยนี้ใช้การพิจารณาค่าหน่วยเวลาเปรียบเทียบกับทฤษฎีที่จำนวนสถานีต่าง ๆ กัน

แบบจำลองระบบดาวเทียม

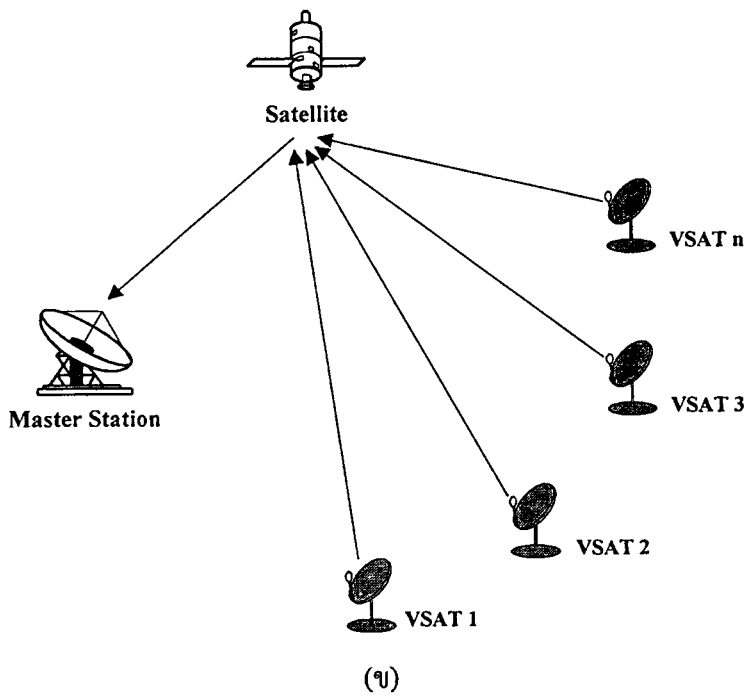
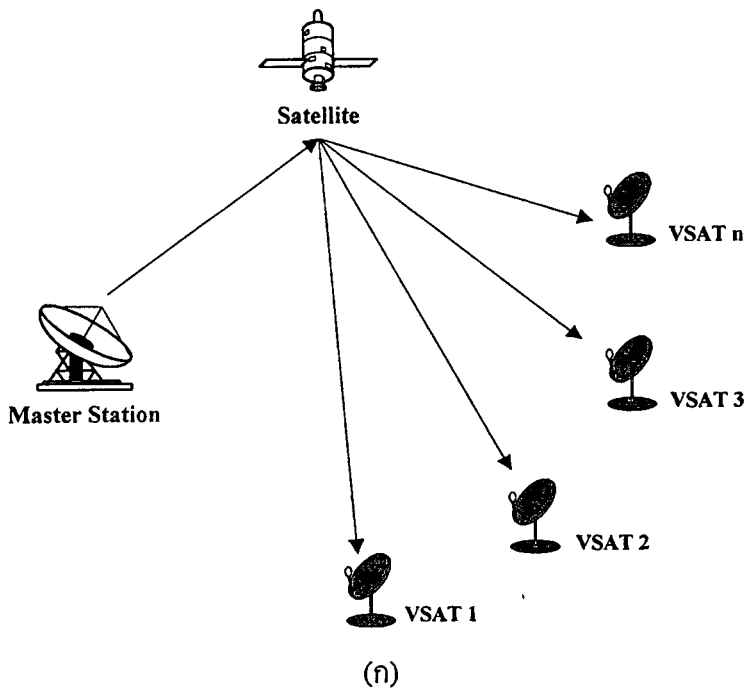
(ใช้ข้อกำหนดของ Fujii และ Wang เป็นข้อกำหนดในการสร้างแบบจำลอง ข้อกำหนดดังกล่าวกำหนดลักษณะของช่องสัญญาณและการติดต่อระหว่างสถานีแม่ข่ายกับสถานีโครงข่าย)

โครงข่ายดาวเทียมประกอบด้วยสถานีแม่ข่าย(HUB)และสถานีดาวเทียมขนาดเล็กซึ่งเป็นสถานีโครงข่ายที่มีการใช้งานสายอากาศขนาดเล็ก ขนาดไม่เกิน 3.8 เมตร(VSAT) จำนวน N สถานี ดังตัวอย่างในภาพที่ 3 แต่ละ VSAT ติดต่อกับสถานีแม่ข่ายได้โดยตรงหรือติดต่อระหว่างกันโดยผ่านสถานีแม่ข่ายเท่านั้น ช่องสัญญาณที่ VSAT ทุกสถานีใช้ติดต่อกับสถานีแม่ข่ายเรียกว่าช่องสัญญาณขาเข้า(Inbound Channel) ลักษณะการเข้าสู่ช่องสัญญาณเป็นแบบสุ่ม(Random Access) และช่องสัญญาณที่สถานีแม่ข่ายใช้ติดต่อกับ VSAT ทุกสถานีเรียกว่าช่องสัญญาณขาออก(outbound channel)ในลักษณะกระจายข้อมูล(Data Broadcast) โดย VSAT ทุกสถานีจะได้รับข้อมูลเหมือนกันหมด แต่ข้อมูลจะถูกถอดเพื่อนำไปใช้งานตามหมายเลข(address) ของสถานีนั้นๆ

สมมติให้ช่องสัญญาณทั้งขาเข้าและออกมีผลกระทบจากสัญญาณรบกวนน้อย รวมทั้งมีระบบที่สามารถแก้ไขปัญหาบิตผิดพลาดที่อาจส่งผลกระทบต่อกระบวนการในโปรโตคอล ชั้น 2, 3(layer2,3)ร่วมด้วย เช่นกระบวนการโพลคอนโทรล[สุวิพล : 1995] ทำให้อัตราบิตที่ผิดพลาด (Bit Error Rate)อยู่ในระดับที่ยอมรับได้ตลอดช่วงของการทดสอบ

ช่องสัญญาณดาวเทียมมีการจัดสรรในรูปของช่วงเวลาโดยแบ่งออกเป็นเฟรม(Frame) ดังแสดงในภาพที่ 4(ก) สมมติให้แต่ละเฟรมมีจำนวนสล็อตเท่ากับ L สล็อต(Time Slot) และแต่ละสล็อตมีขนาดเท่ากับ 0.035 วินาที กำหนดให้ขนาดของแพคเกจที่ส่งจากแต่ละสถานี มีขนาดเท่ากับหนึ่งสล็อต(ขนาดของช่วงว่างระหว่างเฟรมและช่วงว่างของแต่ละสล็อต(Slot)ที่เหมาะสม มี

ภาพที่ 3



แสดง โครงข่ายวีแซท (ก).ช่องสัญญาณขาออก (ข).ช่องสัญญาณขาเข้า

ความจำเป็นอย่างยิ่งสำหรับการกำหนดจุดเริ่มต้นของเฟรมและสล็อต) จำนวนบิตในหนึ่งสล็อตจะขึ้นอยู่กับอัตราเร็วของช่องสัญญาณขาเข้า แต่ละเฟรมกำหนดให้ประกอบไปด้วยสองเฟรมย่อย (Subframe) คือเฟรมย่อยอะโลฮา(Aloha Subframe) และเฟรมย่อยสำหรับการจอง (Reservation Subframe)

กำหนดให้เฟรมย่อยอะโลฮามีอัตราส่วนของจำนวนสล็อตในหนึ่งเฟรมเท่ากับ AL ; $0 < A < 1$ การทำงานของเฟรมย่อยนี้จะเป็นแบบสุ่มในลักษณะ สล็อตอะโลฮา(Random Slotted-Aloha)[Tri.T.Ha,1990:366-378]

กำหนดให้เฟรมย่อยสำหรับการจองมีอัตราส่วนของจำนวนสล็อต(Slot)ในหนึ่งเฟรมเท่ากับ $(1-A)L$ เฟรมย่อยนี้ใช้สำหรับส่งข้อมูลที่ได้รับการจองสำเร็จ เรียงตามลำดับของสถานีที่จอง

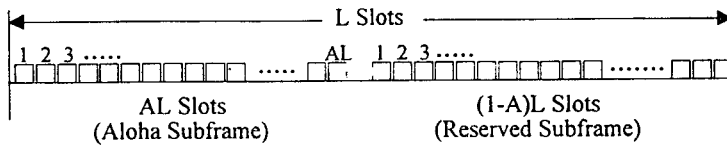
แพ็คเกจใหม่หรือแพ็คเกจที่ต้องส่งซ้ำเนื่องจากการชนจะถูกส่งในสล็อตอะโลฮาในเฟรมย่อยอะโลฮาเท่านั้น ข้อมูลจากสถานีในโครงข่ายที่มาถึง VSAT แบ่งออกเป็นข้อมูลที่มีขนาดหนึ่งแพ็คเกจ และข้อมูลที่มีขนาดหลายแพ็คเกจติดต่อกัน

ข้อมูลที่มีขนาดหนึ่งแพ็คเกจส่วนใหญ่เป็นข้อมูลในลักษณะถามและตอบเช่นข้อมูลจากการใช้เครื่องเบิกเงินอัตโนมัติ ข้อมูลการสั่งซื้อขายในตลาดหลักทรัพย์ ข้อมูลการร้องขอ(Browse)ในวงจรรินเตอร์เน็ต เป็นต้น ข้อมูลนี้ต้องมีขนาดที่สามารถบรรจุในหนึ่งแพ็คเกจได้และส่งในเฟรมย่อยอะโลฮาเท่านั้น

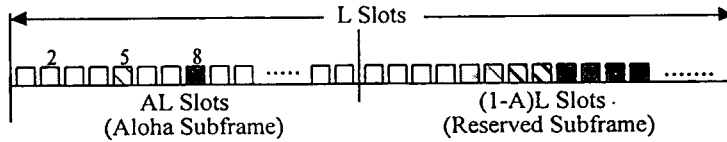
ข้อมูลขนาดหลายแพ็คเกจติดต่อกัน(Batch Data) ที่มาจากสถานีในโครงข่ายจะถูกแบ่งเป็นหลายเมสเสจ โดยโปรโตคอลที่เชื่อมต่อระหว่างสถานีในโครงข่ายและVSAT หรือจากโปรโตคอล ภายในระบบดาวเทียมเอง ในกรณีที่ต่อโดยตรงกับเทอร์มินอล กำหนดให้อัตราส่วนของการเกิดข้อมูลที่มีขนาดเพียงหนึ่งแพ็คเกจและข้อมูลที่มีขนาดหลายแพ็คเกจที่เกิดขึ้นที่แต่ละ VSAT มีค่าเท่ากับ l ,

เมื่อ VSAT พบว่ามีข้อมูลขนาดหลายแพ็คเกจอยู่ในหน่วยความจำ หรือคิวของตน VSAT จะทำการส่งเฉพาะแพ็คเกจที่บรรจุข้อมูลส่วนแรกของข้อมูลขนาดหลายแพ็คเกจดังกล่าวนี้ไปยังช่องสัญญาณดาวเทียม(เฉพาะช่วงของเฟรมย่อยอะโลฮาเท่านั้น) เพื่อทำการจองสล็อต(จำนวนเท่ากับแพ็คเกจที่เหลืออยู่ในคิว) ในเฟรมย่อยสำหรับการจองเรียงตามตำแหน่งของสล็อตที่แพ็คเกจส่งได้สำเร็จดังตัวอย่างที่แสดงในรูปที่ 4(ข) สมมติให้ สล็อตที่ 2 สล็อตที่ 5 และ สล็อตที่ 8 เป็นแพ็คเกจที่บรรจุข้อมูลส่วนแรกของข้อมูล ขนาด 6 แพ็คเกจ 4 แพ็คเกจ และ 5 แพ็คเกจ ตามลำดับ และทำการจองสำเร็จในเฟรมย่อยอะโลฮาตามรูปแบบและข้อกำหนดของช่องสัญญาณ ทำให้สถานีที่ส่งแพ็คเกจในสล็อตที่ 2 สล็อต 5 และ สล็อต 8 ได้รับอนุญาตให้ส่งข้อมูลส่วนที่เหลือ(5 แพ็คเกจ 3 แพ็คเกจ และ 4 แพ็คเกจ ตามลำดับ) ในเฟรมย่อยสำหรับการจองเรียงตามลำดับ ดังนั้นทำให้สล็อตที่ถูกใช้ในเฟรมย่อยสำหรับการจอง เท่ากับ ผลรวมของจำนวนแพ็คเกจที่บรรจุเมสเสจแรก

ภาพที่ 4



(ก)



(ข)

แสดง (ก) โครงสร้างของแต่ละเฟรมของช่องสัญญาณดาวเทียม (ข) รูปแบบการทำงานของเฟรมย่อยอะโลฮาและเฟรมย่อยสำหรับการจอง

ที่ส่งสำเร็จในเฟรมย่อยอะโลฮาและจำนวนเมสเสจที่อยู่ในแต่ละคิว แต่ถ้าแพ็คเกจที่อยู่ในคิวของสถานีที่ทำการจองสำเร็จทั้งหมดมีจำนวนมากกว่า $(1-A)L$ สถานีที่มีคิวส่วนที่เกินจะถูกยกเลิกการจองและต้องทำการส่งใหม่ในเฟรมถัดไป

การวิเคราะห์สมรรถนะของระบบจากแบบจำลอง

1. การวิเคราะห์ค่าทราฟฟิค(Throughput : S)

กำหนดให้แต่ละ VSAT ส่งแพ็คเกจใหม่(แพ็คเกจที่บรรจุเมสเสจแรกของกลุ่มเมสเสจ หรือแพ็คเกจที่บรรจุเมสเสจเดียว)ตามรูปแบบของปัวซอง(Poisson Process) ด้วยค่าอัตราเฉลี่ย λ แพ็คเกจ/เฟรม ในกรณีที่ VSAT ส่งแพ็คเกจใหม่ขึ้นไปและเกิดการชนจะทำการส่งซ้ำจนกว่าจะ สำเร็จ สภาวะที่ VSAT พบว่าแพ็คเกจที่ส่งขึ้นไปเกิดการชนจนกระทั่งส่งซ้ำจนสำเร็จเรียกว่าสถานีอยู่ในสภาวะแบคล็อก(Backlog State) และสถานีที่อยู่ในสภาวะแบคล็อกจะเรียกว่า สถานีแบคล็อก(Backlog Station) แทนด้วย n_{rx} ซึ่ง VSAT ที่อยู่ในสภาวะแบคล็อก ก่อนทำการส่งซ้ำจะหน่วงเวลา(Delay)ด้วยค่าเวลาโดยเฉลี่ยเท่ากับ d_r สล็อต(สมมติให้การหน่วงเวลาดังกล่าวมีการกระจายของเวลาเป็นแบบเอ็กซ์โปเนนเชียลที่ค่าเฉลี่ย $d_r = 1 / \lambda_r$)

เมื่อพิจารณาการมาถึงของแพคเกจใหม่โดยเปรียบเทียบเวลาจากดาวเทียม สามารถแยกพิจารณาได้ดังนี้

1.1 เมื่อ VSAT มีแพคเกจใหม่มาถึงที่เฟรมย่อยสำหรับการจองของช่องสัญญาณดาวเทียม(VSAT) คำนวณแล้วพบว่าเมื่อส่งแพคเกจออกไป แพคเกจจะถึงช่องสัญญาณดาวเทียมในขณะที่ช่องสัญญาณดาวเทียมอยู่ในเฟรมย่อยที่สล็อตถูกจองการใช้งาน) แพคเกจดังกล่าวจะไม่สามารถทำการส่งได้ทันที ดังนั้นแต่ละสล็อตในเฟรมย่อยที่สล็อตถูกจองการใช้งานจะมีจำนวนโดยเฉลี่ยของ VSAT ที่ต้องการส่งแพคเกจ คือ

$$(N - n_{rx}) * \lambda_a \quad (3.1)$$

จำนวน VSAT ที่อยู่ในสถานะ(state) ดังกล่าวในแต่ละเฟรมจะเท่ากับ

$$n_{rv} = \text{int}[(N - n_{rx}) * \lambda_a * (1 - A)L] \quad (3.2)$$

; int [] : เป็นจำนวนเต็มที่ได้จากการปัดเศษขึ้น

VSAT จะเริ่มต้นส่งแพคเกจในเฟรมย่อยอะโลฮาโดยเริ่มจากเฟรมถัดไปด้วยการเลือกช่วงเวลา(Slot)แบบสุ่มและเป็นอิสระต่อกัน ทำให้การส่งแพคเกจใหม่ของแต่ละสถานีมีการกระจายแบบยูนิฟอร์ม(Uniform Distribution) ด้วยจำนวนเท่ากับ n_{rv} / A

1.2 เมื่อ VSAT มีแพคเกจใหม่มาถึงที่เฟรมย่อยอะโลฮาของช่องสัญญาณดาวเทียม VSAT จะอยู่ในสถานะที่สล็อตมีการใช้งานเป็นแบบสุ่ม(Aloha State) และสามารถทำการส่งได้ทันทีด้วยอัตราการส่ง λ_a แพคเกจ/เฟรม

การทำงานและสถานะของช่องสัญญาณดาวเทียม สามารถแสดงโดยการเปรียบเทียบกับกราฟไหลเข้าและออกจากระบบของแพคเกจ ดังในภาพที่ 5 ซึ่งค่าพารามิเตอร์ Q_{in}^* , Q_{out}^* ในภาพที่ 5 แสดงในรูปของสมการได้เท่ากับ

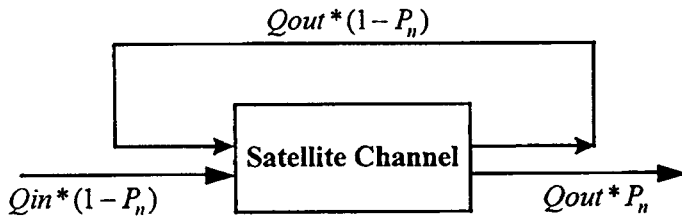
$$Q_{in}^* = (N - n_{rx} - n_{rv})\lambda_a + (n_{rv} / A) \quad (3.3)$$

$$Q_{out}^* = n_{rx}\lambda_r p_{rx} \quad (3.4)$$

อัตราการไหลของแพคเกจจะเท่ากับอัตราการเปลี่ยนแปลงของ n_{rx} ที่มีอยู่ในระบบขณะนั้น และค่าความเป็นไปได้ที่การส่งแพคเกจในแต่ละสล็อตจะสำเร็จ (P_s) ขึ้นอยู่กับ จำนวนของ n_{rx} ปริมาณของแพคเกจที่เข้าสู่ระบบ รวมทั้งอัตราการส่งซ้ำของแพคเกจที่เกิดการชนขึ้น เพื่อให้สะดวก

ต่อการคำนวณจะทำการพิจารณา สมรรถนะ(Performance)ของระบบที่สภาวะคงตัว โดยที่สภาวะดังกล่าวอัตราการไหลเข้าและออกจากระบบของ n_{rx} โดยเฉลี่ยจะเท่ากัน

ภาพที่ 5



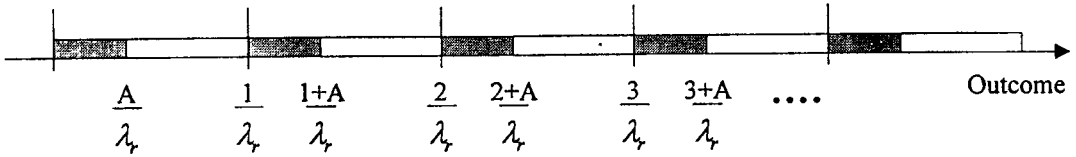
แสดง แบบจำลองสถานะของระบบ

เนื่องจากการกำหนดให้แพ็คเกจสามารถส่งได้เฉพาะภายในช่วงเวลาของสล็อต ทำให้การชนกันของแพ็คเกจจะเกิดขึ้น เมื่อมีจำนวนแพ็คเกจที่ส่งมายังสล็อตเดียวกันมากกว่าหนึ่งตัว ถ้าสมมติให้การส่งแพ็คเกจจาก VSAT ทั้งแพ็คเกจใหม่และแพ็คเกจที่เกิดการชนในการส่งครั้งก่อนเป็นไปตามรูปแบบของปัวซอง(Poisson Process) จะได้ค่าความน่าจะเป็นที่ส่งแพ็คเกจสำเร็จ (P_n) เท่ากับ $\exp(-G_s)$ [Tri.T.Ha: 1990] โดยที่

$$G_s = (N - n_{rx} - n_{sv})\lambda_u + (n_{sv} / A) + (n_{rx}\lambda_r p_{rx}) \quad (3.5)$$

VSAT ที่ตกอยู่ในสภาวะแบคค็อก(Backlog)นั้น ก่อนที่จะทำการส่งซ้ำจะหน่วงเวลา(delay)ด้วยค่าเวลาโดยเฉลี่ยเท่ากับ d_r และเช่นเดียวกันกับ VSAT ในสภาวะปกติ VSAT ในสภาวะแบคค็อกไม่สามารถส่งซ้ำแพ็คเกจบนเฟรมย่อยสำหรับการจองได้ตามข้อกำหนด ทำให้ VSAT ในสภาวะแบคค็อกต้องส่งแพ็คเกจของตนในเฟรมย่อยอะโลฮาด้วยเช่นกัน ภาพที่ 6 แสดงช่วงของความน่าจะเป็นโดยเฉลี่ยของการส่งแพ็คเกจของแต่ละสถานีแบคค็อกในแต่ละเฟรม โดยเริ่มพิจารณาจากเฟรมถัดไปหลังจากที่วิเคราะห์ได้ว่าแพ็คเกจของตนเกิดการชนหรือส่งไม่สำเร็จ การส่งแพ็คเกจเข้าสู่เฟรมย่อยอะโลฮาของสถานีที่อยู่ในสภาวะแบคค็อกต้องสอดคล้องกับอัตราการส่งที่กำหนดไว้สำหรับสถานีแบคค็อก

ภาพที่ 6



แสดง ช่วงความน่าจะเป็นสำหรับการส่งแพคเกจในแต่ละเฟรม

ถ้าให้ p_{rx} เป็นความน่าจะเป็นสะสม(CDF) ที่แต่ละสถานีแบคคี่อกจะส่งแพคเกจในเฟรมย่อยอะโลฮา เท่ากับ

$$\begin{aligned}
 P_{rx} &= [(1 - e^{-A\lambda_r}) - (1 - e^0)] + [(1 - e^{-(1+A)\lambda_r}) - (1 - e^{-\lambda_r})] \\
 &\quad + \dots + [(1 - e^{-(i+A)\lambda_r}) - (1 - e^{-i\lambda_r})] + \dots \\
 &= \sum_{i=0}^{\infty} [(1 - \exp(-i\lambda_r - A\lambda_r)) - (1 - \exp(-i\lambda_r))] \\
 &= \frac{1 - \exp(-A\lambda_r)}{1 - \exp(-\lambda_r)} \tag{3.6}
 \end{aligned}$$

จากสมการที่จุดสมดุล(Equilibrium Point) $Q_{in}^*(1-P_n) = Q_{out}^*(P_n)$ และสมการที่(3.3), (3.4) คำนวณแบบวนซ้ำ(Iteration)โดยเพิ่มค่า n_{rx} ที่ละน้อยจนกระทั่งความแตกต่างของ $Q_{in}^*(1-P_n)$ และ $Q_{out}^*(P_n)$ น้อยกว่า 0.001 จะได้ n_{rx} (เป็นค่าจำนวนเต็ม)ที่จุดสมดุล ดังนั้นค่าทราฟฟิค(Throughput) โดยเฉลี่ยที่จุดสมดุล จะเท่ากับ

$$S_n = [(N - n_{rx} - n_{rxv})\lambda_a + (n_{rxv} / A)] \exp(-G_s) \tag{3.7}$$

เมื่อกำหนดที่จุดสมดุลจะได้จำนวนแพคเกจโดยเฉลี่ยที่บรรจุเมสเสจแรกที่ทำกรจองได้สำเร็จในแต่ละเฟรม(เท่ากับจำนวนของ VSAT ที่มีคิวอยู่) เท่ากับ W_q สล็อต โดยที่

$$W_q = AL(1-l_r)G_s P_n \tag{3.8}$$

ในเฟรมย่อยสำหรับการจอง สล็อตจะจำกัดอยู่ที่ค่า $(1-A)L$ ดังนั้นถ้าจำนวนแพคเกจในคิวทั้งหมดเกินกว่า $(1-A)L$ คิวที่เกินจะถูกตัดการจองทันที สมมติให้คิวของแต่ละ VSAT ที่ช่วงเวลาใดๆเป็นอิสระต่อกันและจำนวนของแพคเกจในคิวมีการกระจายเป็นแบบเอ็กซ์โปเนนเชียล ถ้ากำหนดให้จำนวนแพคเกจโดยเฉลี่ยที่เกิดขึ้นต่อหนึ่งคิว เท่ากับ w แพคเกจ และ p_{sum} เป็นความน่าจะเป็นที่คิวไม่ถูกตัดการจอง เราสามารถสร้างสมการความน่าจะเป็นสำหรับแทนเหตุการณ์ที่ VSAT จะถูกตัดการจองได้ดังนี้

ถ้าให้ $P(N_x > r)$ เป็นจำนวน r หรือ มากกว่าที่เกิดขึ้นในช่วงในช่วง $[0, x]$ เราจะได้

$$\begin{aligned} 1 - P(N_x < r) &= 1 - P(\text{จำนวนที่น้อยกว่า } r \text{ ที่เกิดในช่วง } [0, x]) \\ &= 1 - P(\text{เกิดจำนวนน้อยกว่าหรือเท่ากับ } r \text{ คิวที่ทำให้เกิดแพคเกจเท่ากับ } x) \end{aligned}$$

ทำให้ความน่าจะเป็นที่จะเกิดน้อยกว่าหรือเท่ากับ r คิวและได้แพคเกจเท่ากับ x คือ

$$\begin{aligned} &= \text{CDF ที่จะเกิด 0 คิวและได้แพคเกจเท่ากับ } x + \text{CDF ที่จะเกิด 1 คิวและได้แพคเกจ เท่ากับ } \\ &x + \text{CDF ที่จะเกิด 2 คิวและได้แพคเกจเท่ากับ } x + \text{CDF ที่จะเกิด 3 คิวและได้แพคเกจเท่ากับ } x + \dots + \\ &\text{CDF ที่จะเกิด } x \text{ คิวและได้แพคเกจเท่ากับ } x \end{aligned}$$

$$= \frac{\exp^{-\frac{x}{w}} \cdot (x/w)^0}{0!} + \frac{\exp^{-\frac{x}{w}} \cdot (x/w)^1}{1!} + \frac{\exp^{-\frac{x}{w}} \cdot (x/w)^2}{2!} + \dots + \frac{\exp^{-\frac{x}{w}} \cdot (x/w)^x}{x!}$$

$$= \sum_{i=0}^x \frac{\exp^{-\frac{x}{w}} \cdot (\frac{x}{w})^i}{i!}$$

และความน่าจะเป็นที่จะเกิด i คิวจาก i สถานีในแต่ละเฟรมเท่ากับ

$$= \frac{\exp^{-W_q} \cdot (W_q)^i}{i!}$$

ถ้าให้ $x = (1-A)L$ เราจะได้

$$p_{sum} = \sum_{i=0}^{(1-A)L} \frac{\exp^{-\frac{(1-A)L}{w}} \cdot (\frac{(1-A)L}{w})^i}{i!} \cdot \frac{\exp^{-W_q} \cdot (W_q)^i}{i!}$$

$$= \sum_{i=0}^{(1-A)L} \frac{\exp(-W_q - \frac{(1-A)L}{w}) \cdot (\frac{(1-A)L}{w})^i \cdot W_q^i}{(i!)^2} \quad (3.9)$$

ดังนั้น ค่าทรูพาทโดยเฉลี่ยของเฟรมย่อยสำหรับการจองในแต่ละเฟรมจะเท่ากับ $W_q P_{sum} w$

2. การวิเคราะห์ค่าหน่วงเวลา(Delay : D)

VSAT ที่ส่งแต่ละแพคเกจออกไปจะพบว่าเกิดค่าหน่วงเวลาก่อนที่สถานีแม่ข่ายจะส่งข้อมูลตอบรับ(ack)ว่าได้รับแพคเกจดังกล่าวอย่างสมบูรณ์ ค่าหน่วงเวลาดังกล่าวจะประกอบไปด้วย ค่าหน่วงเวลาที่เกิดจากการเดินทางของคลื่น(Round-trip Propagation Delay: R) เท่ากับ 0.27msec, ค่าหน่วงเวลาที่เกิดจากการทำงานของโปรเซสเซอร์(Processing Time: T_{proc}) ทั้งที่สถานีแม่ข่ายและ VSAT, ค่าหน่วงเวลาที่เกิดจากส่งซ้ำจนกว่าจะสำเร็จ(Retransmission Delay), ค่าหน่วงเวลาที่เกิดจากการถูกยกเลิกการจองในกรณีที่ยังไม่สำเร็จเนื่องจากมีแพคเกจเกินกว่าที่กำหนด

การพิจารณาจะแยกเป็นค่าหน่วงเวลาที่เกิดในสองสถานะคือสถานะอะโลฮาและสถานะที่เกิดการจอง

2.1. เมื่อ VSAT มีแพคเกจใหม่มาถึงในขณะที่ช่องสัญญาณดาวเทียมอยู่ในสถานะอะโลฮาแบ่งได้เป็นสองกรณีคือ

2.1.1 การส่งครั้งแรกสำเร็จ

$$D_1 = 1.5 + 2R + 2T_{proc} \quad (3.10)$$

2.1.2 การส่งครั้งแรกเกิดการชนกับแพคเกจของสถานีอื่น

$$D_2 = 1.5 + 2R + 2T_{proc} + E[r](D_r) \quad (3.11)$$

ค่า $E[r]$ ในสมการที่(3.11) เป็นจำนวนครั้งโดยเฉลี่ยของการส่งซ้ำ ซึ่งมีค่าเท่ากับ $1 - q/q'$ [Tri.T.Ha 1990:365-367] โดยที่ q เป็นความน่าจะเป็นที่จะส่งแพคเกจได้สำเร็จ(P_n) และ q' เป็นความน่าจะเป็นที่จะส่งแพคเกจสำเร็จเท่ากับ $P_n P_{rx}$ VSAT ที่อยู่ในสถานะแบคค็อกต้องทำการส่งซ้ำในเฟรมย่อยอะโลฮาตามข้อกำหนด แต่ในการส่งซ้ำบางครั้งอาจตรงกับเฟรมย่อยสำหรับการจอง เนื่องจากการหน่วงเวลาก่อนการส่งซ้ำแต่ละครั้งมีการกระจายของช่วงเวลาเป็นแบบเอ็กโปเนนเชียล

(Exponential Process) ดังนั้นจำนวนครั้งของการส่งให้ตรงกับเฟรมย่อยอะ โยฮาคำนวณได้จาก

$$m = \sum_{k=1}^{\infty} k(p_{rx})(1-p_{rx})^{k-1} = \frac{1}{p_{rx}} \quad (3.12)$$

จะได้

$$D_r = \frac{L}{\lambda_r} + 2R + 2T_{proc} + m \frac{L}{\lambda_r} \quad (3.13)$$

2.2. เมื่อ VSAT มีแพคเกจใหม่มาถึงในขณะที่ช่องสัญญาณดาวเทียมอยู่ในสถานะที่เกิดการจอง

2.2.1 การส่งครั้งแรกสำเร็จ

$$D_3 = \frac{(1-A)L}{2} + \frac{AL}{2} + 2R + 2T_{proc} \quad (3.14)$$

2.2.2 การส่งครั้งแรกเกิดการชนกับแพคเกจของสถานีอื่น

$$D_4 = \frac{(1-A)L}{2} + \frac{AL}{2} + 2R + 2T_{proc} + E[r]D_r \quad (3.15)$$

จำนวนของคิวหรือ VSAT ที่ถูกตัดการจองจะทำการส่งใหม่ในเฟรมย่อยสำหรับการจองในเฟรมถัดไป ดังนั้น คำนวณเวลาโดยเฉลี่ยของระบบจะเท่ากับ

$$D_i = l_r(D_5) + (1-l_r)[(p_{sum}D_5) + ((1-p_{sum}) * (D_5 + (1-A)L))] \quad (3.16)$$

โดยที่

$$D_5 = \exp(-G_s)[AD_1 + (1-A)D_3] + (1-\exp(-G_s))[AD_2 + (1-A)D_4] \quad (3.17)$$

การวิเคราะห์ผล

โดยทั่วไประบบดาวเทียมที่มีการเข้าใช้แบบสล็อต(Slotted-Aloha) ระบบจะเริ่มเข้าสู่สถานะที่ไม่เสถียรในกรณีที่ที่มีจำนวนแพคเกจเข้าสู่ช่องสัญญาณมากเกินไป(Overload) และสถานะที่เปลี่ยนแปลงขึ้นอยู่กับพารามิเตอร์ที่ใช้ในการทดสอบ ตัวอย่างการวิเคราะห์ผลในที่นี่จะพิจารณาสมรรถนะของระบบจากค่าทฤษฎี(สมการที่ 3.7) และค่าหน่วยเวลา(สมการที่ 3.17) เทียบกับภาระ (load) ที่ $w=3, L=24, A=0.5, R=6, I_r=0.54, \lambda_r=0.5$ กำหนดให้แต่ละสล็อตเท่ากับ 0.035 วินาที ที่ความเร็วของช่องสัญญาณ เท่ากับ 64 kbps

การทดสอบโดยการจำลอง(Simulation) เป็นการจำลองการทำงานของระบบจริงด้วยโปรแกรมคอมพิวเตอร์ภาษา C ที่เวลาทั้งหมด 20,000 สล็อต การจำลองดังกล่าวอยู่บนพื้นฐานของเหตุการณ์ที่เกิดขึ้นจริงในระบบ(Event-Based Simulation) อธิบายในลักษณะของอัลกอริทึม ได้ดังนี้

อัลกอริทึมของโปรแกรมจำลองการทำงานของช่องสัญญาณดาวเทียม

1. กำหนดจำนวนผู้ใช้(User) ทั้งหมดในระบบ อัตราการเกิดของแพคเกจ และสร้างตัวแปรอาร์เรย์ เพื่อเก็บเหตุการณ์ที่เกิดขึ้น(array user) ประกอบไปด้วย
 - ตัวแปรที่เก็บเหตุการณ์เวลาของการเกิดแพคเกจใดๆ ในแต่ละสถานี
 - ตัวแปรที่เก็บเหตุการณ์เวลาของการเกิดแพคเกจในรูปของสล็อต
 - ตัวแปรที่เก็บเหตุการณ์เป็นสถานะของสถานีที่เป็นผู้ส่งแพคเกจดังกล่าว
 - ตัวแปรที่เก็บเวลาในรูปของสล็อต ที่จะเกิดเหตุการณ์ในลำดับถัดไป
 - ตัวแปรที่เก็บค่าเวลาทั้งหมดที่เก็บแพคเกจได้รับก่อนจะส่งสำเร็จ และออกจากระบบ
 - ตัวแปรที่เก็บจำนวนครั้งของการส่งใหม่เนื่องจากแพคเกจในคิวมีจำนวนเกินกว่าที่จะส่งได้ในเฟรมย่อยอะโลฮา
2. สร้างแพคเกจให้เกิดขึ้นในแต่ละสถานีทุกสถานีที่เวลาต่างๆ กัน ด้วยฟังก์ชันย่อย ARRTIME() โดยสอดคล้องกับช่วงเวลาที่อยู่ในรูปของสล็อต(ฟังก์ชันย่อย MKSLOT()) และปรับให้อยู่ในช่วงของเฟรมย่อยอะโลฮาเท่านั้น
3. ตรวจสอบสล็อตที่ 1 ว่าอยู่ในช่วงของเฟรมย่อยอะโลฮาหรือไม่ ถ้าอยู่ในช่วงดังกล่าวตรวจสอบต่อไปว่า
 - 3.1 ถ้ามีแพคเกจที่เกิดขึ้นเพียงแพคเกจเดียว กำหนดให้แพคเกจของสถานีนั้นทำการส่งสำเร็จ หลังจากนั้นตรวจสอบว่าเป็นแพคเกจแรกของข้อมูลที่ มีขนาดยาวๆ หรือไม่ผลที่ได้แบ่งเป็นสองกรณีเช่นกันคือ
 - 3.1.1 กรณีที่เป็นแพคเกจแรกของข้อมูลที่มีขนาดยาว จะส่งแพคเกจที่เหลือใน

คิวของตนได้เมื่อเฟรมย่อยที่ถูกจองมีจำนวนสล็อตเพียงพอเท่านั้นซึ่งถ้าพบว่าแพคเกจในคิวมีขนาดมากกว่าจำนวนสล็อตที่เหลือในเฟรมย่อยสำหรับจำลอง ทำการส่งใหม่ในเฟรมย่อยอะโลฮา (ฟังก์ชันย่อย Aloha ())

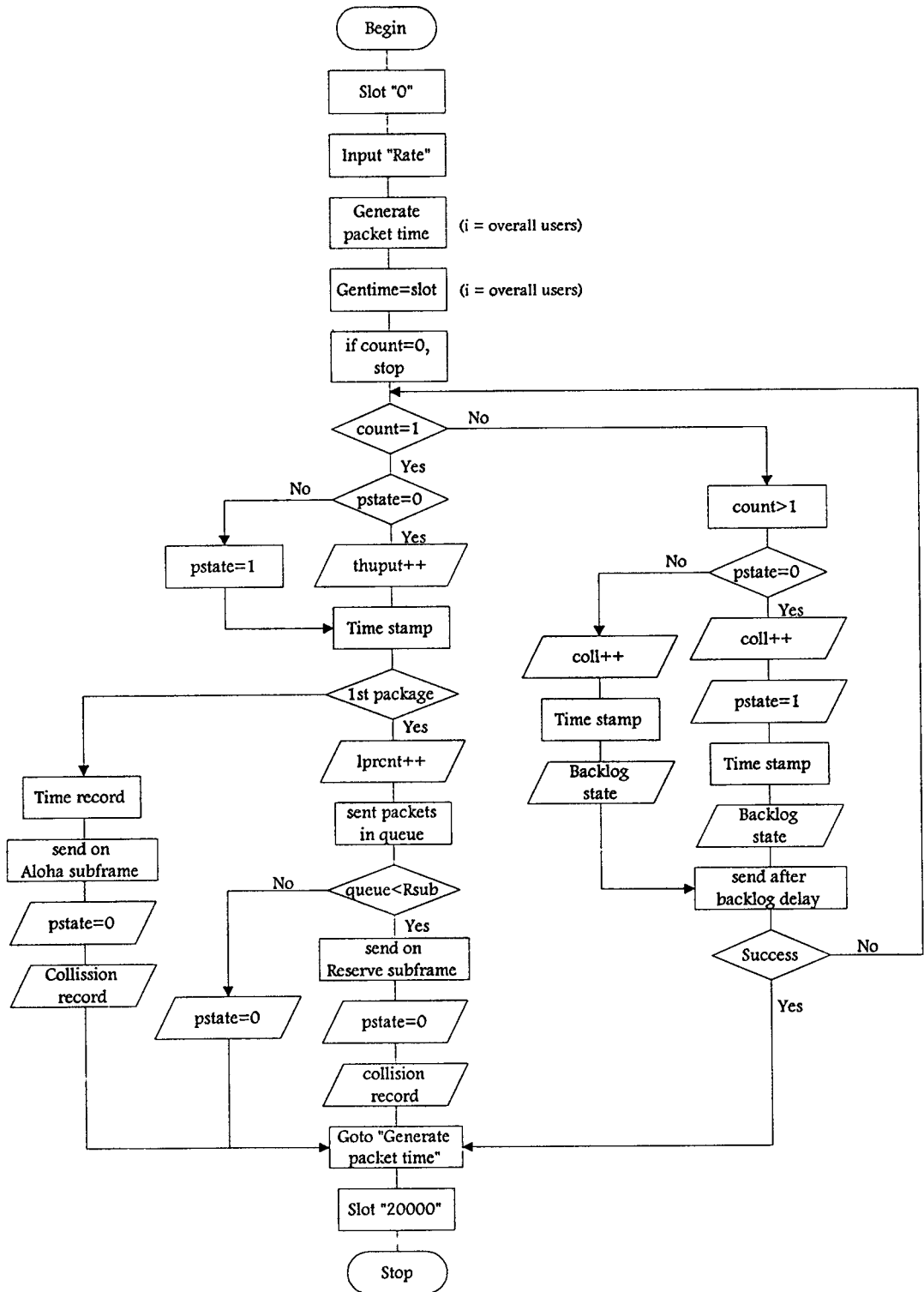
3.1.2 ในกรณีที่แพคเกจของข้อมูลเดียว เก็บค่าเวลาที่ส่งสำเร็จ, เวลาที่เกิด, จำนวนครั้งของการชน หลังจากนั้น เสร็จสิ้นการส่งแพคเกจนั้น

3.2 ถ้ามีแพคเกจหลายแพคเกจส่งจากหลายสถานีเกิดขึ้นในสล็อตนี้ แสดงว่าเกิดการชนขึ้น ทำการปรับสถานะของสถานีที่เป็นต้นกำเนิดของแพคเกจเหล่านั้นทั้งหมด ให้มีสถานะแบคล็อก(State = 1) และทำการส่งใหม่ตามขั้นตอนและข้อกำหนดของการส่งซ้ำ(Backlog State) จนกว่าจะสำเร็จ

4. ทำการวนลูปโดยตรวจสอบสล็อตถัดไปจนกว่าจะครบจำนวนสล็อตตามที่กำหนด

อัลกอริทึมดังกล่าว สามารถเขียนให้อยู่ในรูปของแผนภูมิโปรแกรม(Flow chart) ดังแสดงในภาพในหน้าถัดไป

ภาพที่ 7

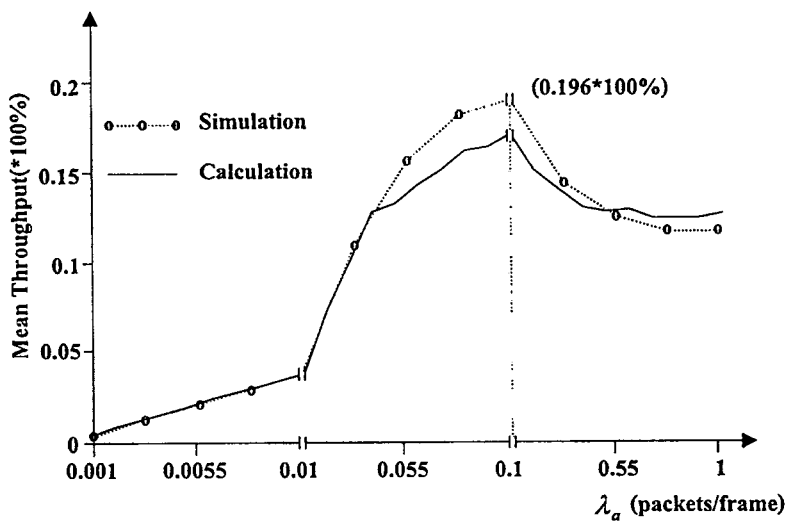


แสดง แผนภูมิโปรแกรม SIMSAT

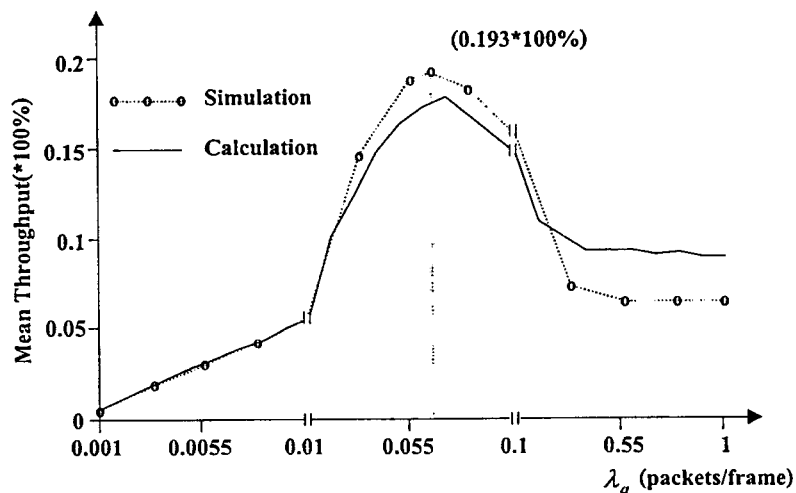
จากอัลกอริทึมและแผนภูมิโปรแกรมที่ได้แสดงไว้ สร้างเป็นตัวโปรแกรมภาษา C ที่ชื่อ SIMSAT แสดงในภาคผนวก ก.

พิจารณาภาพที่ 8 (ก),(ข),(ค)และ(ง) ซึ่งเป็นกราฟแสดงค่าทรูพุทโดยเฉลี่ยของระบบที่ได้จากการทดลอง run โปรแกรม SIMSAT แปรค่าตามอัตราการส่งแพคเกจโดยเฉลี่ยที่ค่าต่างๆ ที่ $N=50,75,100$ และ 125 users ตามลำดับ

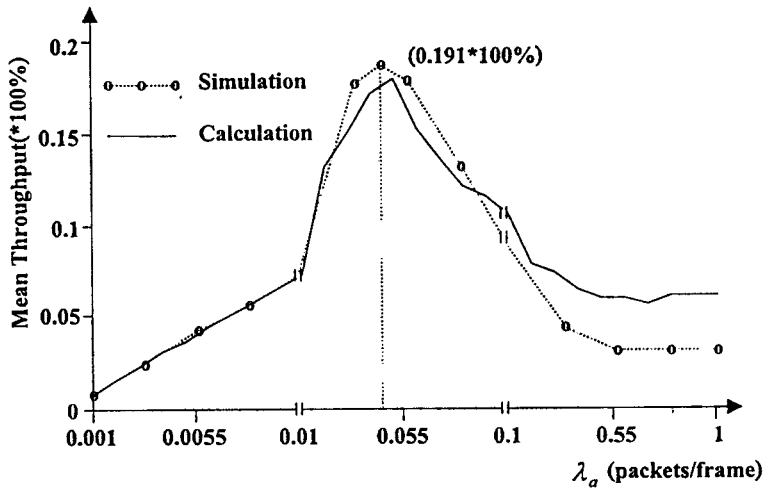
ภาพที่ 8



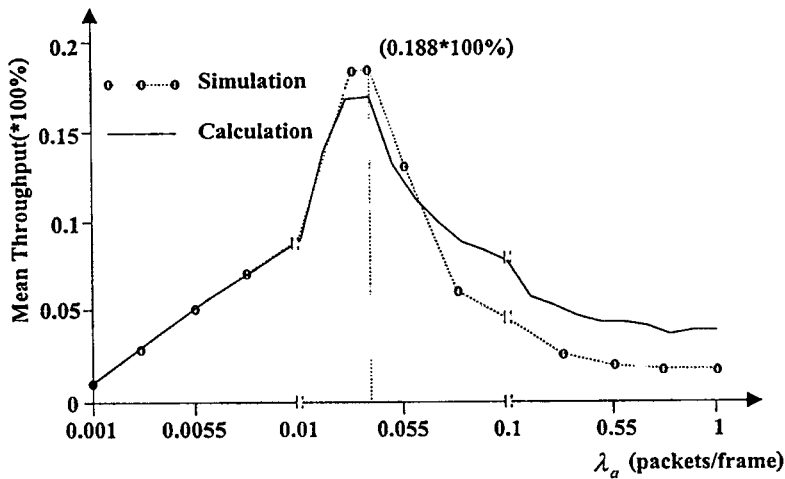
(ก)



(ข)



(ค)



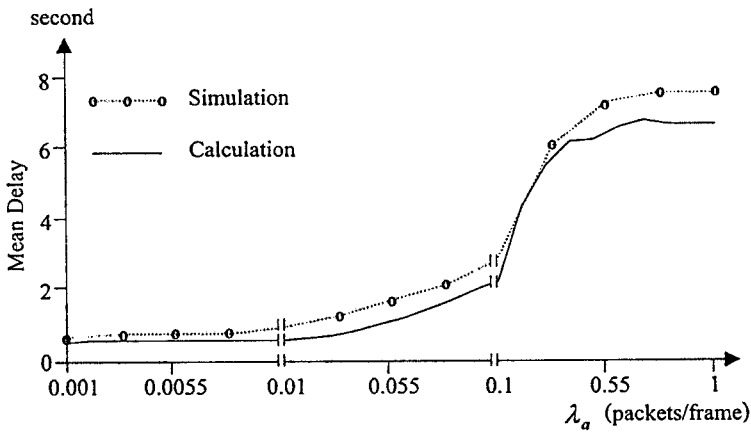
(ง)

แสดง ค่าทรูพุทเปรียบเทียบกับ λ_a ที่ $w=3, L=24, A=0.5$ เมื่อ (ก) $N=50$, (ข) $N=75$, (ค) $N=100$,
(ง) $N=125$.

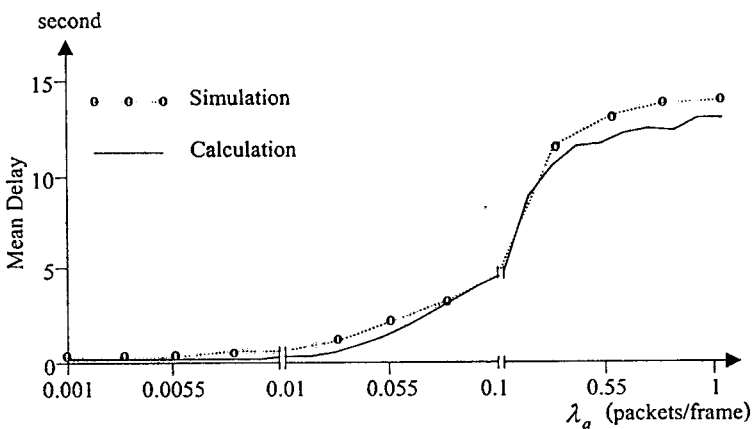
จากรูปสามารถสังเกตได้ว่า ค่าทรูพุทสูงสุดที่ได้จากการคำนวณและการจำลองจะอยู่ที่ค่าประมาณ 18% (สมการที่ 3.7) แต่จะได้ค่าดังกล่าวที่อัตราการส่งต่างกันเมื่อเปลี่ยนค่า N โดยอัตราการส่งที่สูงขึ้นหมายถึงโหนดที่มากขึ้น ระบบเริ่มอึดตัวและจะเข้าสู่สถานะที่ไม่เสถียรเร็วขึ้นเมื่ออัตราการส่งสูงขึ้น ที่สถานะไม่เสถียรจะมีสถานีที่ต้องการส่งซ้ำจำนวนมาก แต่เมื่อพิจารณาเปรียบเทียบสถานะโอเวอร์โหนดจากรูปทั้งสามพบว่าที่ $N=50$ มีทรูพุทที่สูงกว่ารูปอื่น เนื่องจากมีสถานีที่ต้องการส่งซ้ำน้อยกว่าที่ค่า $N=75, N=100$ และ 125 ทำให้ค่าความน่าจะเป็นที่แพคเกจใหม่จะส่งสำเร็จในระบบที่ $N=50$ มีค่าสูงกว่าระบบอื่นๆ (ที่ค่า λ_a เดียวกัน) ที่สถานะโอเวอร์โหนด

เมื่อพิจารณาจากกราฟพบว่าเกิดความแตกต่างประมาณ 3% ระหว่างการคำนวณและการจำลองหลังจากที่ระบบเข้าสู่สภาวะโอเวอร์โหลด ซึ่งเป็นผลจากการคำนวณเพื่อคำนวณหา n_{tx} และจำนวนสล็อตที่จะต้องทำการส่งใหม่ในกรณีที่พบว่าเกิดบนเฟรมย่อยสำหรับการจอง ค่าดังกล่าวที่ได้จากการคำนวณจะมีความต่อเนื่อง(Continuous) มากกว่าการจำลอง ทำให้ความน่าจะเป็นที่จะเกิดแพคเกจที่ต้องการส่งในครั้งต่อไปเทียบกับเวลา(Slot)มีค่าต่ำกว่า ซึ่งถ้าแก้ไขโดยการปิดเศษขึ้นทุกครั้งจะทำให้มีการสะสมของค่าผิดพลาดที่เกิดขึ้นแต่ละครั้งของการคำนวณ

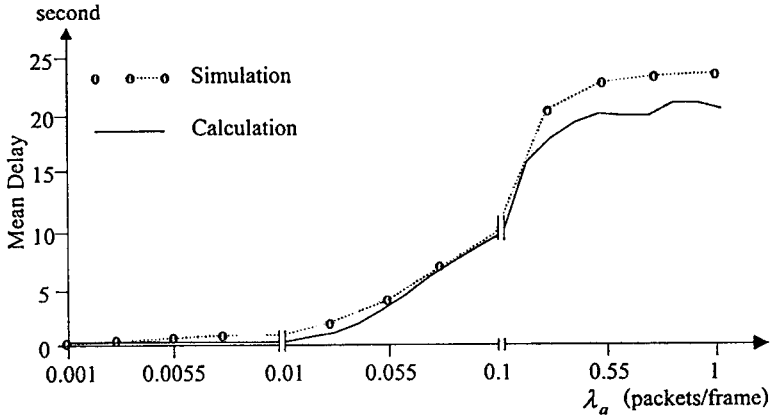
ภาพที่ 9



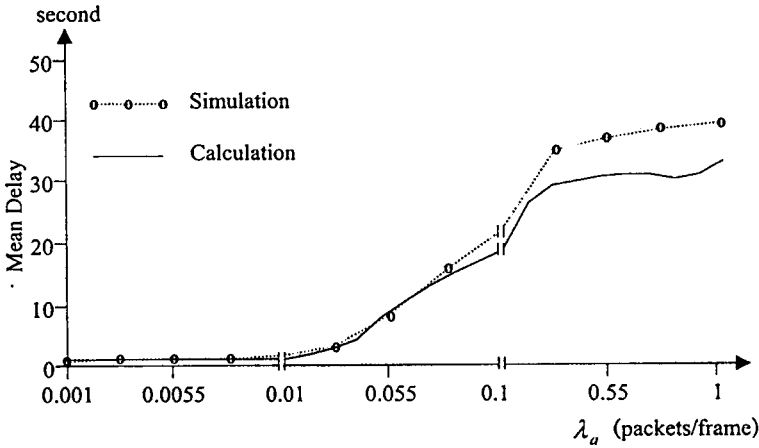
(ก)



(ข)



(ค)



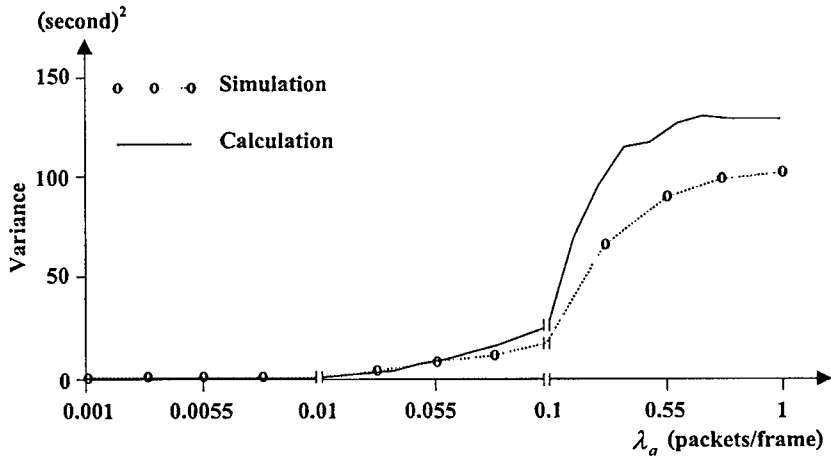
(ง)

แสดง ค่าหน่วยเวลาเปรียบเทียบกับ λ_a ที่ $w=3, L=24, A=0.5$ เมื่อ (ก) $N=50$, (ข) $N=75$,
 (ค) $N=100$, (ง) $N=125$.

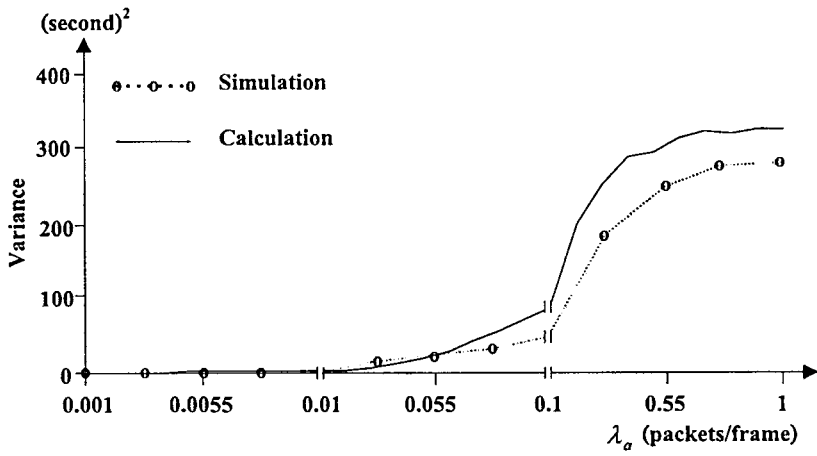
ภาพที่ 9 และ 10 (ก),(ข),(ค) และ (ง) แสดงค่าหน่วยเวลาเป็นวินาที(ไม่รวมเวลาที่ใช้ในการส่งทุกแพคเกจในคิว)และค่าความแปรปรวนของค่าหน่วยเวลาแปรค่าตามอัตราการส่งแพคเกจโดยเฉลี่ยที่ค่าต่างๆ ที่ $N=50,75,100$ และ 125 ตามลำดับ จากรูปพบว่าค่าหน่วยเวลาโดยเฉลี่ย(Mean Delay)ของระบบจะเปลี่ยนแปลงในลักษณะเพิ่มขึ้นเมื่อระบบเข้าสู่สถานะอิมตัวเร็วขึ้นตามจำนวนสถานีที่มากขึ้นส่งผลต่อเนื่องไปยังค่าความแปรปรวนทำให้เกิดค่าที่เปลี่ยนแปลงมากขึ้นในขณะนั้น

เมื่อระบบเข้าสู่สถานะไม่เสถียร จำนวนสถานีแบคคัล็อกจะเพิ่มสูงขึ้นทำให้จำนวนครั้งที่ VSAT ต้องส่งแพคเกจซ้ำสูงขึ้น ส่งผลให้ค่าหน่วยเวลาสูงขึ้นตามลำดับ(ค่าหน่วยเวลาที่เกิดจากคิวที่โค่นตัดการจองมีผลค่อนข้างน้อยเมื่อเทียบกับผลของการส่งซ้ำ)

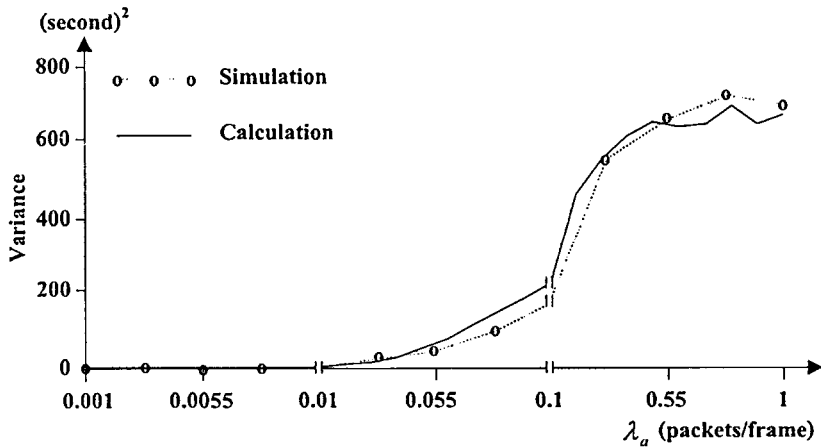
ภาพที่ 10



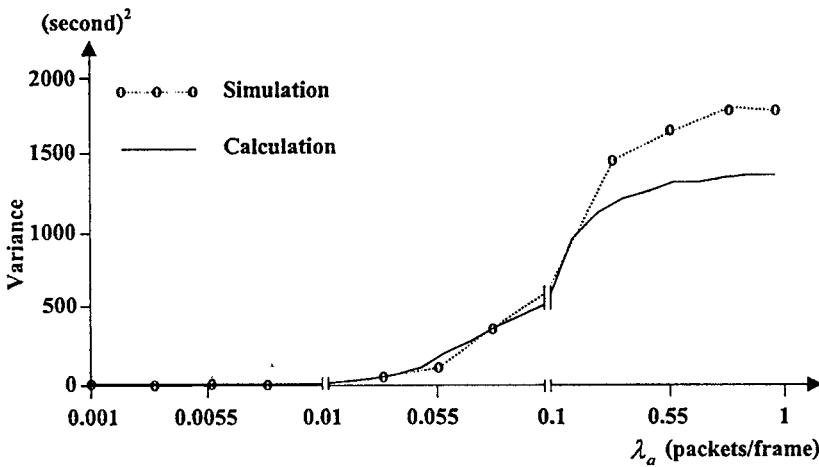
(f)



(g)



(h)

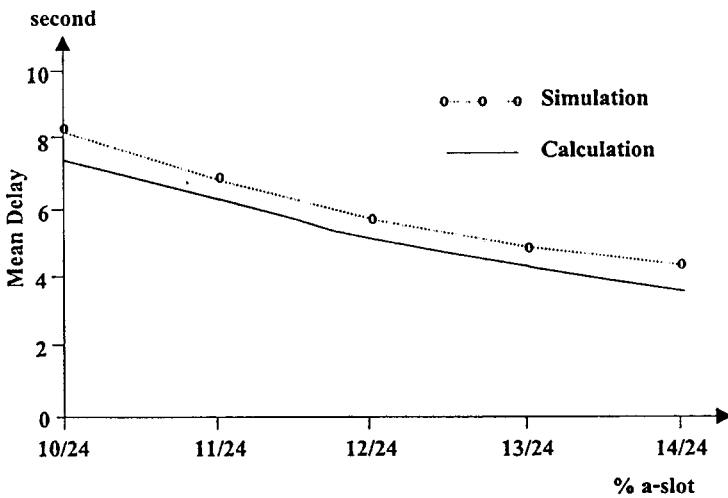


(ง)

แสดง ค่าเฉลี่ยของค่าหน่วยเวลาเปรียบเทียบกับ λ_a ที่ $w=3, L=24, A=0.5$ เมื่อ (ก) $N=50$,
 (ข) $N=75$, (ค) $N=100$, (ง) $N=125$.

เมื่อทดลองเปลี่ยนค่าอัตราส่วนของจำนวนสล็อตในเฟรมย่อยอะโลฮา และเฟรมย่อย สำหรับการจอง ค่าหน่วยเวลาโดยเฉลี่ยของระบบจะลดลงเมื่อจำนวนสล็อตในเฟรมย่อยอะโลฮา เพิ่มขึ้นดังภาพที่ 11 แสดงให้เห็นว่า ผลของค่าหน่วยเวลาส่วนใหญ่ขึ้นอยู่กับขนาดของเฟรมย่อย อะโลฮาเนื่องจากกำหนดให้แพ็คเกจใหม่และแพ็คเกจที่ต้องส่งซ้ำเนื่องจากการชนต้องส่งในเฟรมย่อยอะโลฮาเท่านั้น หรืออีกนัยหนึ่งการชนของแพ็คเกจเป็นแฟคเตอร์สำคัญต่อค่าหน่วยเวลาโดยรวมของระบบ

ภาพที่ 11



แสดง ค่าหน่วยเวลาที่อัตราส่วน (L_r) ต่างๆกันของสล็อตอะโลฮาและสล็อตสำหรับการจอง ที่ $\lambda_a=0.05$ packets/frame

จากภาพที่ 9, 10 และ 11 แสดงให้เห็นว่าการเลือกจุดทำงานควรมีขั้นตอนคือ

- ประเมินอัตราการส่งข้อมูลโดยเฉลี่ยของ VSAT ที่คาดว่าจะมีในระบบ
- พิจารณาค่าเวลาที่เหมาะสมสำหรับชั้นที่ 2,3(layer 2,3) ของโปรโตคอลที่ใช้ในการเชื่อม

ต่อที่สถานีโครงข่าย

- เลือกค่าทรูพุทที่ N ที่เหมาะสมกับข้อมูลสองข้อข้างต้น เช่น $N=100$ ในตัวอย่างข้างต้น
- ประเมินอัตราส่วนของการเกิดข้อมูลขนาดหนึ่งแพคเกจและข้อมูลที่มีขนาดหลายแพคเกจ

ติดต่อกันที่จะเกิดขึ้นที่แต่ละ VSAT

- เลือกขนาดของเฟรมย่อยอะโลฮา(Aloha Subframe)และเฟรมย่อยสำหรับการจอง(Reservation Subframe) ในหนึ่งเฟรม

วิธีการเลือกจุดทำงานนี้ สามารถนำไปใช้เพื่อการคำนวณหาจุดทำงานที่เหมาะสมสำหรับโครงข่ายแบบผสมในบทที่ 5 ต่อไป

บทที่ 4

การวิเคราะห์และสร้างโมเดลสำหรับสถานีที่ใช้ตัวกลางดาวเทียมใน โครงข่ายแบบผสม

บทนำ

ในบทนี้ เป็นการวิเคราะห์และศึกษารูปแบบของเหตุการณ์ที่มีความเป็นไปได้ที่จะเกิดขึ้นจริงในโครงข่ายที่มีการใช้ตัวกลางร่วมกัน โดยการสร้างแบบจำลอง(Simulation Model)ที่สามารถจำลองการทำงานของระบบสถานีที่ใช้ตัวกลางเป็นดาวเทียมด้วยโครงสร้างทางคณิตศาสตร์และเปรียบเทียบผลที่ได้ด้วยการจำลองการทำงานของระบบด้วยโปรแกรมคอมพิวเตอร์เช่นเดียวกับการวิเคราะห์ในบทที่ 3 แบบจำลองที่สร้างขึ้นจำลองการเกิดเหตุการณ์ในรูปแบบที่แตกต่างกันและเปรียบเทียบผลที่ได้จากการจำลองทั้งสองแบบ

เมื่อสถานีแต่ละสถานีเป็นอิสระต่อกันรวมทั้งพิจารณาจากสมการที่ 2.2 และสมการที่ 2.3, เราสามารถแยกพิจารณาเพื่อหาค่าพารามิเตอร์ที่แสดงประสิทธิภาพของระบบภายในแต่ละสถานี กล่าวคือ สถานีที่ใช้ตัวกลางเป็นสายภาคพื้นดินจะมีค่าหน่วยเวลาโดยเฉลี่ยภายในสถานี ดังสมการที่ 2.2 และมีจำนวนแพคเกจที่อยู่ภายในคิวเท่ากับ

$$N_i = \frac{R_i}{uC_i - R_i} \quad (4.1)$$

แต่สำหรับสถานีที่ใช้ตัวกลางเป็นดาวเทียมที่มีลักษณะการเข้าสู่ช่องสัญญาณเป็นแบบสุ่ม(Random Access)นั้น เมื่อทำการส่งแพคเกจของข้อมูลเข้าสู่ระบบดาวเทียมเพื่อไปยังสถานีที่เชื่อมต่อปลายทาง ค่าหน่วยเวลาที่เกิดขึ้น ณ จุดนี้จะแตกต่างจากค่าหน่วยเวลาที่พิจารณาจากสมการที่ 2.2 เนื่องจากผลของลักษณะการเข้าสู่ช่องสัญญาณดาวเทียมและจำนวนของแพคเกจที่อยู่ภายในคิวแปรเปลี่ยนไปตามค่าหน่วยเวลาที่ข้อมูลเดินทางเข้าสู่ช่องสัญญาณดาวเทียมดังกล่าว เมื่อพิจารณาผลกระทบของค่าหน่วยเวลาที่เกิดจากระบบดาวเทียม(วิเคราะห์ในบทที่ 3) ที่มีต่อการส่งแพคเกจของข้อมูลแต่

ละครึ่ง พบว่าค่าหน่วยเวลาที่เกิดขึ้นไม่สามารถแทนด้วยรูปแบบการกระจายแบบ เอ็กโปเนนเชียล (Exponential Distribution) ทั้งนี้เนื่องจากจำนวนแพคเกจที่เกิดขึ้นในคิวจะเป็นผลสืบเนื่องมาจากค่าหน่วยเวลาที่เกิดจากระบบดาวเทียม ทำให้การกระจายของเวลาในการให้บริการ (Service Time Distribution) ไม่สอดคล้องกับคุณสมบัติของการไม่เกี่ยวข้องกันของสองเหตุการณ์ที่เกิดขึ้นต่อเนื่องกัน (Markovian Property) [Hisashi 1978:82] ในขณะที่ขบวนการของการมาถึงแต่ละสถานียังคงสอดคล้องกับคุณสมบัติดังกล่าว จากโครงสร้างดังที่กล่าวมาสอดคล้องกับโมเดลแบบ $M/G/1$ [Hisashi 1978:191-195] ยกเว้นในกรณีของจำนวนแพคเกจที่มาถึงในขณะที่แพคเกจใด ๆ กำลังได้รับการบริการ ซึ่งโดยทั่วไปโมเดลแบบ $M/G/1$ จะทำการคำนวณที่ช่วงเวลาทั้งหมดของการให้บริการ ทำให้ค่าหน่วยเวลาที่เกิดขึ้นภายในสถานีผิดจากความเป็นจริงเมื่อแพคเกจที่ส่งเข้าสู่ช่องสัญญาณดาวเทียมเกิดการชนมากกว่าหนึ่งครั้ง ในงานวิจัยนี้ได้ทำการคำนวณค่าหน่วยเวลาที่เกิดขึ้นเนื่องจากแพคเกจที่อยู่ในคิวเป็นค่าเฉลี่ย และเปรียบเทียบผลที่ได้กับการจำลองการทำงานของระบบด้วยโปรแกรมคอมพิวเตอร์ที่จำลองขั้นตอนการทำงานตามสภาพที่เป็นจริง

สมการโมเดลสำหรับคำนวณค่าหน่วยเวลา

โมเดลที่ใช้ระบบคิวแบบ $M/G/1$ สามารถทำการคำนวณได้สองวิธี วิธีแรกเป็นการแบ่งสแตต (State) ของการให้บริการออกเป็นหลายๆสแตต แต่ละสแตตจะเสมือนว่าเป็นอิสระต่อกันและใช้สมการเอ็กโปเนนเชียลในการคำนวณ ส่วนอีกวิธีหนึ่งคือใช้การพิจารณาการเปลี่ยนสถานะของระบบในแต่ละช่วงเวลา (The Imbedded Markov Chain Technique) สำหรับวิธีที่นำมาพิจารณาและแก้ไขในบทนี้เป็นวิธีแบบที่สอง สามารถแสดงการคำนวณโดยละเอียดได้ดังนี้

กำหนดให้ ช่วงเวลาแบ่งเป็นสล็อต (Slot) ที่มีขนาดเท่ากันเช่นเดียวกับในบทที่สาม แต่ละแพคเกจที่มาถึงระบบจะเกิดค่าหน่วยเวลาก่อนที่ออกจากระบบ เนื่องจากสาเหตุดังต่อไปนี้

- ค่าหน่วยเวลาเนื่องจากแพคเกจในกลุ่มเดียวกัน (X_k) ที่มาถึงคิวของสถานีก่อนหน้าแพคเกจที่กำลังพิจารณา
- ค่าหน่วยเวลาเนื่องจากแพคเกจที่มีอยู่ในคิว ก่อนที่แพคเกจที่ k เข้ารับบริการ
- ค่าหน่วยเวลาเนื่องจากการโปรเซส (Process) ในช่องสัญญาณดาวเทียม
- ค่าหน่วยเวลาที่เกิดจากการส่งแพคเกจออกจากระบบ

ถ้าให้ N_k เป็นจำนวนของแพคเกจที่พบอยู่ในระบบ หลังจากทีแพคเกจที่ K ได้รับการบริการจนเสร็จสิ้น และ X_k เป็นจำนวนของแพคเกจที่มาถึงในขณะที่แพคเกจที่ K กำลังได้รับการบริการ โดยที่ X_1, X_2, X_3, \dots เป็นอิสระต่อกัน (Independent) และมีการกระจายแบบเดียวกัน นอก

จากนี้ ตามโครงสร้างของลำดับการให้บริการแบบตามลำดับ (First Come First Serve) ทำให้ X_k เป็นอิสระต่อ $N_1, N_2, N_3 \dots N_{k-1}$ แต่ปริมาณของ N_k และ X_k ไม่ใช่ตัวแปรที่เป็นอิสระต่อกัน แสดงได้ตามสมการดังนี้

$$N_k = \begin{cases} X_k & \text{if } N_{k-1} = 0 \\ N_{k-1} + X_k - 1 & \text{if } N_{k-1} > 0 \end{cases}$$

เมื่อ $N_{k-1} = 0$ หมายความว่าแพคเกจที่ K ได้รับการบริการจนเสร็จสิ้น และออกจากระบบไปโดยไม่มีแพคเกจเหลืออยู่ในระบบนอกจากแพคเกจใหม่ที่มาถึงในขณะที่แพคเกจที่ K เริ่มเข้ารับบริการ ซึ่งกำหนดให้เป็นจำนวนเฉลี่ย เท่ากับ X_k และเมื่อ $N_{k-1} > 0$ พบว่าก่อนที่แพคเกจที่ K เริ่มเข้ารับบริการมีแพคเกจอยู่ในคิวจำนวน N_{k-1} และเมื่อแพคเกจที่ K ได้รับการบริการจนเสร็จสิ้นจะมีแพคเกจเพิ่มขึ้นอีก เป็นจำนวนเฉลี่ยเท่ากับ X_k กำหนดให้

$$U(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

รวมสมการทั้งสองเข้าด้วยกันจะได้

$$N_k = N_{k-1} - U(N_{k-1}) + X_k \quad (4.1)$$

กำหนดให้ p_n เป็นความน่าจะเป็นที่จะพบจำนวน N_k แพคเกจในระบบ หลังจากที่แพคเกจที่ K ได้รับการบริการจนเสร็จสิ้นและจากฟังก์ชันกำหนดความน่าจะเป็น (Probability Generating Function) เราจะได้

$$P(z) = \sum_{n=0}^{\infty} p_n z^n$$

สามารถสร้างฟังก์ชันกำหนดความน่าจะเป็น (Probability Generating Function) จากสมการที่ (4.1) ได้ จากสมการ

$$P_N(z) = \frac{\rho_0 \cdot (z-1) \cdot A(z)}{z - A(z)} \quad (4.2)$$

โดยที่ $A(z)$ เป็นฟังก์ชันกำหนดความน่าจะเป็นของ X_k (ในบทวิจัยนี้จะศึกษา X_k ในรูปของช่วงเวลา(สล็อต)สำหรับการเดินทางจากสถานีต้นทางไปยังสถานีปลายทางผ่านดาวเทียมของแต่ละแพคเกจ ช่วงเวลาดังกล่าวเป็นค่าเฉลี่ยที่ได้จากค่าหน่วงเวลาที่เกิดในช่องสัญญาณดาวเทียม)

จากสมการพบว่า N_k มีการเปลี่ยนแปลงในลักษณะของ ลูกโซ่มาร์คอฟ(Markov Chain) และเป็นสมการเริ่มต้นของการพิจารณากระบวนการคิว แบบ M/G/1 ในงานวิจัยนี้ใช้โครงสร้างเริ่มต้นเดียวกันจากสมการที่(4.1) แต่จะแยกพิจารณาจำนวนเฉลี่ย X_k เป็นอีกลักษณะหนึ่ง เนื่องจากโมเดลคำนวณค่าเวลาที่แพคเกจใดๆอยู่ในระบบที่ใช้แทนการทำงานของระบบดาวเทียมขึ้นอยู่กับสถานะของระบบดาวเทียมในขณะนั้น ทำให้ไม่สามารถใช้จำนวนแพคเกจที่เกิดขึ้นทั้งหมดในช่วงเวลาการให้บริการสำหรับการพิจารณาจำนวน โดยเฉลี่ยของแพคเกจและค่าหน่วงเวลาที่เกิดในสถานีดาวเทียมในขณะใดๆได้

เมื่อใช้คุณสมบัติที่ X_k เป็นอิสระต่อ $N_1, N_2, N_3 \dots N_{k-1}$ ในทุกๆเหตุการณ์ และกำหนดสมการเริ่มต้นโดยสมมติให้แพคเกจที่เราต้องการพิจารณาเกิดขึ้นในขณะที่แพคเกจที่ K กำลังได้รับการบริการ ช่วงขณะที่แพคเกจที่ K กำลังได้รับการบริการ พบว่ามีแพคเกจจำนวนหนึ่งอยู่ในระบบแพคเกจจำนวนดังกล่าวประกอบไปด้วยแพคเกจชุดที่เหลืออยู่ก่อนที่แพคเกจที่ K จะเข้ารับการให้บริการ(อาจจะไม่พบแพคเกจจำนวนนี้)และแพคเกจที่กำลังได้รับการบริการ แทนเหตุการณ์ที่มีแพคเกจ จำนวนเท่ากับ W_D อยู่ในระบบขณะที่แพคเกจที่ K ออกจากระบบด้วยสมการ

$$W_D = N_{k-1} - U(N_{k-1}) \quad (4.3)$$

สร้างฟังก์ชันกำหนดความน่าจะเป็นจากสมการที่ (4.3) ได้

$$\begin{aligned} P_W(z) &= p_0 + \sum_{n=1}^{\infty} p_n z^{n-1} \\ &= p_0 + z^{-1} (P_N(z) - p_0) \end{aligned} \quad (4.4)$$

โดยที่

p_0 : ความน่าจะเป็นที่แพคเกจใหม่ที่มาถึงจะพบว่าไม่มีแพคเกจอื่นอยู่ในคิวหรือกล่าวได้ว่าผู้ให้บริการไม่ได้ให้บริการ มีค่าเท่ากับ $P[N_{k-1} = 0] = 1 - P[N_{k-1} > 0]$

ในกรณีที่เรากำลังพิจารณาช่วงเวลาในแต่ละแพ็คเกจได้รับการบริการ อัตราการใช้งานของระบบ (Utilization factor: ρ) จะเท่ากับ $\lambda\mu$ [Leonard Kleinrock 1975:18-19] ซึ่งเท่ากับจำนวนโดยเฉลี่ยของแพ็คเกจที่มาถึงและยังคงอยู่ในระบบในแต่ละช่วงเวลาของการให้บริการ (u_k) ดังนั้นสรุปได้ว่า $1 - P[N_{k-1} > 0] = 1 - \rho = 1 - u_n$.

สำหรับอีกเหตุการณ์หนึ่ง(เปรียบเทียบกับการศึกษาแพ็คเกจจำนวนหนึ่งที่มาถึงในขณะที่แพ็คเกจที่ K กำลังได้รับการบริการ)เป็นการพิจารณาเหตุการณ์ที่มีแพ็คเกจจำนวนหนึ่งมาถึงก่อนหน้าที่แพ็คเกจที่กำหนดจะมาถึงระบบจากจำนวนแพ็คเกจทั้งหมดที่มาถึงในช่วงเวลาหนึ่งสล็อต ในกรณีนี้ ถ้าสมมติให้แพ็คเกจที่กำหนดเป็นตำแหน่งที่ $d+1$ ในกลุ่มของแพ็คเกจเดียวกัน (X_k) ที่มาถึงในช่วงเวลาหนึ่งสล็อต ดังนั้นกล่าวได้ว่า X_k ต้องมีจำนวนมากกว่า d

ถ้าให้ $P[X_k = d + 1] = P[X_d]$ เราสามารถแทนเหตุการณ์ที่แพ็คเกจที่กำหนดเข้ารับการบริการเป็นแพ็คเกจที่ $d+1, d+2, d+3, \dots$ ด้วยค่าความน่าจะเป็นที่มีการกระจายเท่ากับ

$$P[X_d] = \frac{1 - (P[X_0] + P[X_1] + P[X_2] + \dots + P[X_d])}{P[X > 0]} \quad (4.5)$$

$$P(X_d) = \frac{1 - \sum_{k=0}^d P[X_k]}{u_n} \quad (4.6)$$

จากสมการที่(4.4) และ (4.6) เราได้โมเดลพื้นฐานในการคำนวณจำนวนโดยเฉลี่ยของแพ็คเกจ ที่อยู่ในคิวก่อนลำดับของแพ็คเกจที่กำหนดในขณะใดขณะหนึ่ง ถ้าพิจารณาค่าเวลาของการให้บริการ(Service Time) สำหรับสถานีที่ใช้ตัวกลางเป็นดาวเทียมที่มีโครงสร้างและข้อกำหนดตามรูปแบบที่ได้แสดงในบทที่สาม พบว่าค่าเวลาของการให้บริการของแต่ละแพ็คเกจหมายถึงเวลาที่แพ็คเกจดังกล่าวส่งผ่านช่องสัญญาณดาวเทียมได้สำเร็จ(การพิจารณาการทำงานของระบบ หรือขั้นตอนการปฏิบัติการ(Process)กับแพ็คเกจใดๆจะอยู่ภายในโปรโตคอลระดับที่สาม(Network layer) แต่ละแพ็คเกจที่ถูกส่งขึ้นดาวเทียมมีความเป็นอิสระต่อกัน ถึงแม้ว่าจะเป็นเพียงส่วนหนึ่งของข้อมูลที่ได้นำมาประกอบเข้าด้วยกันบนโปรโตคอลระดับที่สี่(Transport Layer))

กำหนดให้ช่วงเวลาแบ่งเป็นสล็อต(Slot)ที่มีขนาดเท่ากันดังเช่นในบทที่สาม แต่ละแพ็คเกจได้รับการบริการเป็นจำนวนสล็อตโดยเฉลี่ยแตกต่างกันตามสถานะของระบบดาวเทียม ถ้าจำนวนสล็อตสำหรับการเดินทางจากสถานีต้นทางไปยังสถานีปลายทางผ่านดาวเทียมเท่ากับ S สล็อต แพ็คเกจ จะได้รับการบริการเป็นจำนวนสล็อตโดยเฉลี่ยเท่ากับ nS กรณีนี้พบว่าเราสามารถแทนแต่ละแพ็คเกจในรูปของแต่ละ nS ยกตัวอย่างเช่นแพ็คเกจที่มาถึงระบบในขณะที่แพ็คเกจที่ K กำลังได้รับ

การบริการมีจำนวนเท่ากับ 4 แพคเกจ จะเท่ากับจำนวนสล็อตโดยรวมที่ทั้ง 4 แพคเกจมาถึง เท่ากับ $4 * nS$ (เพื่อความสะดวกในการคำนวณ กำหนดให้ $nS = s$ โดยที่ n เป็นค่าเฉลี่ยที่ได้จากค่าหน่วยเวลาที่เกิดในช่องสัญญาณดาวเทียม) ในบทวิจัยนี้ได้ใช้คุณสมบัติที่สอดคล้องกับการคำนวณผลรวมของตัวแปรเชิงสุ่มที่เป็นอิสระต่อกันในส่วนของ การคำนวณค่าคาดหวังแบบมีเงื่อนไข (Conditional Expectation and Prediction : Sum of Random Number of Random variable)[Jack R. Benjamin, C. Allin Cornell 1981: 175-180] ซึ่งอธิบายโดยย่อได้ดังนี้

สมมติให้ R เป็นผลรวมของเหตุการณ์ r เหตุการณ์ ที่มีรูปแบบเดียวกัน เป็นอิสระต่อกัน (Independent Random Variable) และมีค่าเฉลี่ยโดยรวมร่วมกัน

$$\begin{aligned} R &= r_0 + r_1 + r_2 + r_3 + \dots + r_n \\ &= \sum_{i=0}^n r_i \end{aligned} \quad (4.7)$$

$$; r \in \{0,1,2,3,\dots\}$$

ถ้าค่า n ถูกกำหนดให้เป็นค่าคงที่เท่ากับ l และจากคุณสมบัติของค่าเฉลี่ย(ค่าคาดหวัง)ของตัวแปรเชิงสุ่มใดๆ เราจะได้ค่าเฉลี่ยและค่าความแปรปรวน(Variance)ของ R ในรูป

$$\begin{aligned} E[R / n = l] &= E\left[\sum_{i=0}^l r_i\right] \\ &= \sum_{i=0}^l E[r_i] = lE[r] \end{aligned} \quad (4.8)$$

และ

$$\begin{aligned} Var[R / n = l] &= E[R^2] - E^2[R] \\ &= E\left[\left(\sum_{i=0}^l r_i\right)^2\right] - E^2\left[\sum_{i=0}^l r_i\right] \\ &= E\left[\left(\sum_{i=0}^l r_i\right)^2\right] - lE[r] \end{aligned} \quad (4.9)$$

เมื่อพิจารณาค่า $Var[R / n = l]$ เราสามารถใช้คุณสมบัติของค่าความแปรปรวน ที่แสดงในรูป

$$Var[R / n = l] = Var\left[\sum_{i=0}^l r_i\right]$$

$$= \sum_{i=1}^l \text{Var}[r_i] = I\text{Var}[r] \quad (4.10)$$

เราจะได้

$$E\left[\left(\sum_{i=0}^l r_i\right)^2\right] = I\text{Var}[r] + (IE[r])^2 \quad (4.11)$$

ถ้าค่า n คงที่เราต้องการค่า $\text{Var}[r]$ และ $E[r]$ ของ r_i ในการคำนวณเท่านั้น ในกรณีที่ค่า n ในสมการเป็นตัวแปรเชิงสุ่มอีกตัวหนึ่งที่เราต้องการพิจารณา จะได้

$$\begin{aligned} E[R / N = n] &= E[R] = E[n \cdot (E[r])] \\ &= E[r] \cdot E[n] \end{aligned} \quad (4.12)$$

และ

$$\text{Var}[R / n] = E^2[R] - E[R^2] \quad (4.13)$$

ค่า $E[R^2]$ หาได้จากสมการ

$$\begin{aligned} E[R^2] &= E[n \cdot \text{Var}[r] + n^2 \cdot E[r]^2] \\ &= \text{Var}[r] \cdot E[n] + E[r]^2 \cdot E[n^2] \\ &= \text{Var}[r] \cdot E[n] + E^2[r] \cdot \text{Var}[n] + E^2[r] \cdot E^2[n] \end{aligned} \quad (4.14)$$

ให้ $\text{Var}[R / N = n] = \text{Var}[R]$ และแทนค่าสมการที่(4.12) และ (4.14) ลงในสมการที่ (4.13) จะได้

$$\text{Var}[R] = \text{Var}[r] \cdot E[n] + E^2[r] \cdot \text{Var}[n] \quad (4.15)$$

จากสมการที่ (4.7) กำหนดให้ r และ R มีฟังก์ชันกำหนดความน่าจะเป็นเท่ากับ $G_r(z)$ และ $G_R(z)$ ตามลำดับ โดยที่ N เป็นจำนวนหรือปริมาณของ r เหตุการณ์ ที่พบในการทดลองแต่ละครั้ง และ $G_R(z) = G_N(G_r(z))$ [Geoferey and Dominic : 51] พิสูจน์โดยย่อได้ดังนี้

$$\begin{aligned} G_R(z) &= E(z^{r_1 + \dots + r_n}) \\ &= \sum_{n=0}^{\infty} E(z^{r_1 + \dots + r_n} / N = n) P(N = n) \end{aligned}$$

$$\begin{aligned}
&= \sum_{n=0}^{\infty} (G_r(z))^n P(N = n) \\
&= G_N(G_r(z))
\end{aligned} \tag{4.16}$$

เมื่อทำการเปรียบเทียบการคำนวณผลรวมของตัวแปรเชิงสุ่มแบบมีเงื่อนไขที่เป็นอิสระต่อกัน (Conditional Expectation and Prediction : Sum of Random Number of Random variable) กับเหตุการณ์ที่กำลังพิจารณา เราจะได้

R = ผลรวมของเวลาการให้บริการ(สลีต)แต่ละแพคเกจที่มาถึงในแต่ละช่วงเวลาของการให้บริการ

r = เวลาของการให้บริการแต่ละแพคเกจ(ผลจากระบบคิวเทียม)

u_r = เวลาโดยเฉลี่ยของการให้บริการแต่ละแพคเกจ

N = จำนวนของแพคเกจที่ปรากฏในคิวที่เวลาใดๆ

u_n = จำนวนของแพคเกจโดยเฉลี่ยที่มาถึงในแต่ละครั้งของการให้บริการ

u_R = จำนวนเวลา(สลีต)โดยเฉลี่ยสำหรับ R แพคเกจ

$$= u_n u_r$$

$Var[R]$ = ค่าความแปรปรวนของจำนวนเวลา(สลีต)สำหรับ R แพคเกจ

$$= u_n \cdot Var[r] + (u_r)^2 \cdot Var[n]$$

$G_R(z)$ = ฟังก์ชันกำหนดความน่าจะเป็นของ R

$$= G_N(G_r(z))$$

ค่า $Var[R]$, $E[R]$ และ $G_N(G_r(z))$ เป็นพารามิเตอร์ที่จำเป็นสำหรับการสร้างสมการโมเดลที่จะกล่าวถึงต่อไป ค่า $Var[R]$ และ $E[R]$ สามารถคำนวณได้จากฟังก์ชันกำหนดความน่าจะเป็นอีกวิธีหนึ่ง[Chu and Alan : 1972]] โดยพิจารณาสมการ

$$P(z) = \sum_{n=0}^{\infty} p_n z^n$$

โดยที่ $p_n \geq 0$ และทำการ Differentiate เทียบกับ z จะได้

$$\frac{dP(z)}{dz} = \sum_{n=0}^{\infty} n p_n \cdot z^{n-1}$$

$$P'(z) = E[n] \quad ; z=1 \quad (4.17)$$

และ

$$\frac{d^2 P(z)}{dz^2} = \sum_{n=0}^{\infty} n^2 p_n \cdot z^{n-2}$$

$$P''(z) = E[n^2] - E[n] \quad ; z=1 \quad (4.18)$$

ทำให้เราได้ค่า $Var[x]$ ทันที เท่ากับ

$$Var[x] = P''(1) + P'(1) - (P'(1))^2 \quad (4.19)$$

พิจารณา $A(z)$ ในสมการที่(4.2) โดยที่ $A(z)$ เป็นฟังก์ชันกำหนดความน่าจะเป็นของ X_k ถ้าเราพิจารณา X_k ในรูปของช่วงเวลา(สล็อต)ของการให้บริการตามข้อกำหนด เราสามารถแสดง $A(z)$ ในรูปของตัวแปรเชิงสุ่มแบบมีเงื่อนไขที่มีรูปใหม่คือ $A_N(G_r(z))$ แทนลงในสมการที่ (4.2) ได้สมการใหม่ในรูปของ $G_r(z)$ คือ

$$P_N(z) = \frac{(1 - u_k u_r) \cdot (z-1) \cdot A_N(G_r(z))}{z - A_N(G_r(z))} \quad (4.20)$$

แทนสมการที่ (4.20) ลงในสมการที่ (4.4)

$$P_w(z) = p_0 + z^{-1}(P_N(z) - p_0)$$

จัดรูปใหม่จะได้

$$P_w(z) = \frac{(z-1) \cdot (1 - u_k u_r)}{z - A_N(G_r(z))} \quad (4.21)$$

(เพื่อความสะดวกในการพิจารณาจะแทน $A_N(G_r(z))$ ด้วย “A ”)

ทำการ Differentiate เทียบกับ z

$$T_w(z) = \frac{(1 - u_n u_r) \cdot [(z - A) - (z - 1)(1 - A')]}{(z - A)^2} \quad (4.21.1)$$

เมื่อแทน $z=1$ จะได้ $\frac{0}{0}$ ซึ่งตามกฎหมายของ L'Hopital ที่กำหนดการ Differentiate แยกทั้งเศษและส่วนจะได้

$$T_w(z) = \frac{(1 - u_n u_r) \cdot [1 - A' - 1 + zA'' + A' - A'']}{2(z - A)(1 - A')} \quad (4.21.2)$$

$$= \frac{(1 - u_n u_r) \cdot [zA'' + A'']}{2(z - zA' - A + A'A)} \quad (4.21.3)$$

$$= \frac{(1 - u_n u_r) \cdot [A'']}{2(1 - A')^2} \quad ; z=1 \quad (4.21.4)$$

พิจารณาการกระจายของ X_d จากสมการที่ (4.6) :

$$P[X_d] = \frac{1 - (P[X_0] + P[X_1] + P[X_2] + \dots + P[X_d])}{P[X > 0]}$$

$$= \frac{1 - \sum_{k=0}^d (P[x = k] \cdot P[l = r])}{u_n} \quad (4.22)$$

จัดสมการให้อยู่ในรูปของฟังก์ชันกำหนดความน่าจะเป็น (Probability Generating Function) ในรูปของ $G_r(z)$ โดยใช้คุณสมบัติของ z-transform [Kleinrock 1975 : 330] ได้

$$P_x(z) = \frac{1}{u_n} \left(\frac{1}{1 - G_r(z)} - \frac{A_n(G_r(z))}{1 - G_r(z)} \right) \quad (4.23)$$

แทน $A_n(G_r(z))$ ด้วย "A" และ แทน $G_r(z)$ ด้วย "G" ได้สมการที่ (4.23) ในรูปใหม่ คือ

$$P_x(z) = \frac{1}{u_n} \left(\frac{1 - A}{1 - G} \right) \quad (4.24)$$

ใช้วิธีการเดียวกันกับการแก้สมการที่ (4.21) จะได้สมการสุดท้ายคือ

$$P_x(z) = \frac{1}{u_n} \left(\frac{A''G' - G''A'}{2(G')^2} \right) ; z=1 \quad (4.25)$$

สมการที่ (4.21.4) และ (4.25) เป็นสมการโมเดลสำหรับองค์ประกอบของค่าหน่วยเวลา ในรูปของฟังก์ชันกำหนดความน่าจะเป็น เมื่อนำสมการทั้งสองและค่าเวลาของการให้บริการของแพคเกจที่กำลังพิจารณาารวมกัน จะได้เวลาเฉลี่ยทั้งหมดที่แพคเกจใดๆ ได้รับก่อนที่จะออกจากระบบ เท่ากับ

$$T = \frac{A''G'A' + A''G'(1-A') - G''A'(1-A') + 2(G')^2 A'(1-A')}{2G'A'(1-A')} \quad (4.26)$$

จากสมการที่(4.19)ร่วมกับคุณสมบัติทั่วไปของการคำนวณผลรวมของตัวแปรเชิงสุ่มแบบมีเงื่อนไข เราจะได้สมการที่ (4.19) ในรูปใหม่ และค่าเฉลี่ยของตัวแปรดังกล่าวเท่ากับ

$$Var[i,r] = A'' + A' - (A')^2 ; z=1 \quad (4.27)$$

$$A' = u_{n,r} = u_n u_r ; z=1 \quad (4.28)$$

แทนค่าสมการที่(4.27) ลงในสมการที่(4.26) จะได้

$$T = \frac{G'A'(Var(n,r)) - G'(A')^2(1-A') + G'(Var(n,r)) - G'A'(Var(n,r)) - A'(Var(n,r)) + (A')^2(Var(n,r)) - A'(G')^2 + (A')^2(G')^2 + G'(A')^2 - G'A'(A')^2 + 2(1-A')(G')^2 A'}{2(1-A')G'A'} \quad (4.29)$$

แทนค่าสมการที่(4.28) ลงในสมการที่(4.29) จะได้

$$T = \frac{(u_r)^2 (u_n)^2 \text{Var}(r) + (u_r)^4 u_n \text{Var}(n) - (u_r)^3 (u_n)^2 + (u_r)^4 (u_n)^3 + u_r u_n \text{Var}(r) + (u_r)^3 \text{Var}(n) - (u_r)^2 (u_n)^2 \text{Var}(r) - (u_r)^4 u_n \text{Var}(n) - u_r u_n \text{Var}(r) + (u_r)^2 (u_n)^2 \text{Var}(r) - u_n (u_r)^3 + (u_r)^4 (u_n)^2 + (u_r)^3 (u_n)^2 - (u_r)^4 (u_n)^3 + 2(u_r)^3 u_n - 2(u_r)^4 (u_n)^2}{2(1 - u_r u_n) u_n (u_r)^2}$$

ได้สมการสุดท้ายคือ

$$T = \frac{u_r u_n - (u_r)^2 (u_n)^2 + u_r \text{Var}(n) + (u_n)^2 \text{Var}(n)}{2(1 - u_r u_n) u_n} \quad (4.30)$$

จากสมการ ตัวแปรที่ยังไม่สามารถหาค่าได้คือ $\text{Var}(n)$

พิจารณา $A(z)$ ซึ่งเป็นฟังก์ชันกำหนดความน่าจะเป็นของ X_k [Hisashi 1978 : 191-200] ภายใต้เงื่อนไขของความน่าจะเป็นที่มีการกระจายแบบปัวซอง (Poisson Distribution) โดยที่ X_k ขึ้นกับช่วงเวลาของการให้บริการ ในรูปของสมการ

$$\begin{aligned} A(z) &= \sum_{i=0}^{\infty} \int_0^{\infty} \frac{e^{-\lambda x} (\lambda x)^i}{i!} f_S(x) dx \\ &= \int_0^{\infty} e^{-\lambda x} e^{-\lambda z x} f_S(x) dx = A(\lambda - \lambda z) \end{aligned}$$

ฟังก์ชัน $A(\lambda - \lambda z)$ เป็นสมการโมเดลที่แสดงพฤติกรรมการมาถึงระบบของแพคเกจ ซึ่งมีความสัมพันธ์โดยตรงกับเวลาของการให้บริการ Leonard Kleinrock [Kleinrock : 1975] ได้แสดงความสัมพันธ์ของ $\text{Var}[n]$ และ $A(z)$ ในรูปของค่าเฉลี่ย และความแปรปรวนของเวลาการให้บริการ ซึ่งจะแสดงโดยย่อในที่นี้เท่านั้น ผู้สนใจสามารถค้นคว้าเพิ่มเติมได้จากหนังสืออ้างอิงดังกล่าว กำหนดให้

$$\begin{aligned} \frac{dA(\lambda - \lambda z)}{dz} &= \left(\frac{dA(\lambda - \lambda z)}{d(\lambda - \lambda z)} \right) \cdot \left(\frac{d(\lambda - \lambda z)}{dz} \right) \\ \frac{dA(\lambda - \lambda z)}{dz} &= -\lambda \frac{dA(\lambda - \lambda z)}{d(\lambda - \lambda z)} \end{aligned} \quad (4.31)$$

ในขณะเดียวกัน

$$\frac{d^2 A(\lambda - \lambda z)}{dz^2} = \frac{d[-\lambda \cdot (\frac{dA(\lambda - \lambda z)}{d(\lambda - \lambda z)})]}{dz}$$

$$\frac{d^2 A(\lambda - \lambda z)}{dz^2} = \lambda^2 \cdot (\frac{d^2 A(\lambda - \lambda z)}{d(\lambda - \lambda z)^2}) \quad (4.32)$$

พิจารณาสมการทั่วไป

$$\frac{d^k A(\lambda - \lambda z)}{d(\lambda - \lambda z)^k} = (-1)^k E[x^k] \quad (4.33)$$

และจากสมการที่ (4.17) และ (4.18)

$$P'(z) = E[n] \quad ; z=1 \quad (4.17)$$

$$P''(z) = E[n^2] - E[n] \quad ; z=1 \quad (4.18)$$

แทนสมการที่ (4.17) ลงในสมการที่ (4.31) ได้

$$E[A] = -\lambda \cdot (-E[r]) \quad (4.34)$$

แทนสมการที่ (4.18) ลงในสมการที่ (4.32) ได้

$$E[A^2] - E[A] = \lambda^2 \cdot E[r^2] \quad (4.35)$$

เมื่อ $u_n = \lambda \cdot E[x]$ และ $E[x^2] = Var[n] + E^2[x]$ แทนค่าในสมการที่ (4.35) ได้

$$E[A^2] - E[A] = (u_n)^2 \cdot (1 + \frac{Var[r]}{E^2[r]}) \quad (4.36)$$

ทำให้ได้อยู่ในรูปของค่าความแปรปรวนของของจำนวนแพคเกจที่มาถึงในช่วงเวลาของการให้บริการเท่ากับ

$$E[A^2] - E^2[A] = (u_n)^2 \cdot \left(1 + \frac{\text{Var}[r]}{E^2[r]}\right) + u_n - (u_n)^2$$

$$\text{Var}[A] = \text{Var}[n] = (u_n)^2 \cdot \left(1 + \frac{\text{Var}[r]}{E^2[r]}\right) + u_n - (u_n)^2$$

ได้สมการสุดท้ายเท่ากับ

$$T = \frac{u_r}{2} + \frac{(u_r)^2 + u_n \cdot \text{Var}[r] \cdot (1 + u_r)}{2 \cdot u_r \cdot (1 - u_r u_n)} \quad (4.37)$$

สมการที่(4.37) อยู่ภายใต้เงื่อนไขที่กำหนด โดยเฉพาะความเป็นอิสระของแต่ละสถานีและ Convexity Property เราสามารถใช้สมการที่ (4.37) เป็นโมเดลทางคณิตศาสตร์สำหรับการจำลองเหตุการณ์ภายในสถานีที่ใช้ตัวกลางดาวเทียม ซึ่งจะได้ทำการวิเคราะห์และเปรียบเทียบผลการทดลองที่ได้จากโมเดลทางคณิตศาสตร์กับผลที่ได้จากการจำลองเหตุการณ์โดยโมเดลที่สร้างจากโปรแกรมคอมพิวเตอร์ต่อไป

สมการที่ (4.37) สามารถใช้เป็นสมการโมเดลสำหรับคำนวณค่าหน่วงเวลาที่เกิดขึ้นในสถานีที่ใช้ตัวกลางดาวเทียมในระบบโครงข่ายแบบผสม แสดงโดยละเอียดในบทต่อไป

ผลการทดลอง

การตรวจสอบผลที่ได้จากโมเดลทางคณิตศาสตร์จะทำการจำลองเหตุการณ์โดยโมเดลที่สร้างจากโปรแกรมคอมพิวเตอร์เช่นเดียวกับการทดลองในบทที่สาม การพิจารณาเหตุการณ์ที่เกิดขึ้นในแต่ละสถานี เป็นการพิจารณาผลที่เป็นอิสระต่อกันตามข้อกำหนดที่ได้แสดงไว้ในบทที่สอง

กำหนดให้เหตุการณ์ที่เกิดขึ้นมีผลร่วมกันและเป็นผลต่อเนื่องกันกับเหตุการณ์ที่เกิดขึ้นในช่องสัญญาณดาวเทียม ทั้งนี้ความหมายของการให้บริการในสถานีที่ใช้ตัวกลางดาวเทียม หมายถึง เวลาที่ใช้สำหรับส่งผ่านแพคเกจบนช่องสัญญาณดาวเทียมจนสำเร็จ(พิจารณาโดยละเอียดในบทที่สาม) ทำให้แต่ละแพคเกจใช้เวลาอยู่ในสถานีดาวเทียมหนึ่งๆ เท่ากับเวลาของการให้บริการรวมกับเวลาที่ต้องรออยู่ในคิวก่อนที่จะเข้ารับการบริการ(แทนด้วยสมการที่ 4.37) จากแผนภูมิโปรแกรมที่จำลองเหตุการณ์ในช่องสัญญาณดาวเทียม(ภาพที่ 7 ในบทที่3) เพิ่มเหตุการณ์ที่เกิดขึ้นบนสถานีที่กำหนดให้เป็นสถานีที่ใช้ในการพิจารณา โดยการจำลองคิวและอัตราการมาถึงของข้อมูลของสถานี

ดังกล่าว ทำให้ได้แผนภูมิโปรแกรม(Flowchart) ดังแสดงในภาพที่ 12 ผลที่ได้จากโปรแกรมที่มีโครงสร้างตามแผนภูมิ(โปรแกรม Simsat ในภาคผนวก ก) แสดงในตารางที่ 2.1 - 2.4 ในภาคผนวก ข.

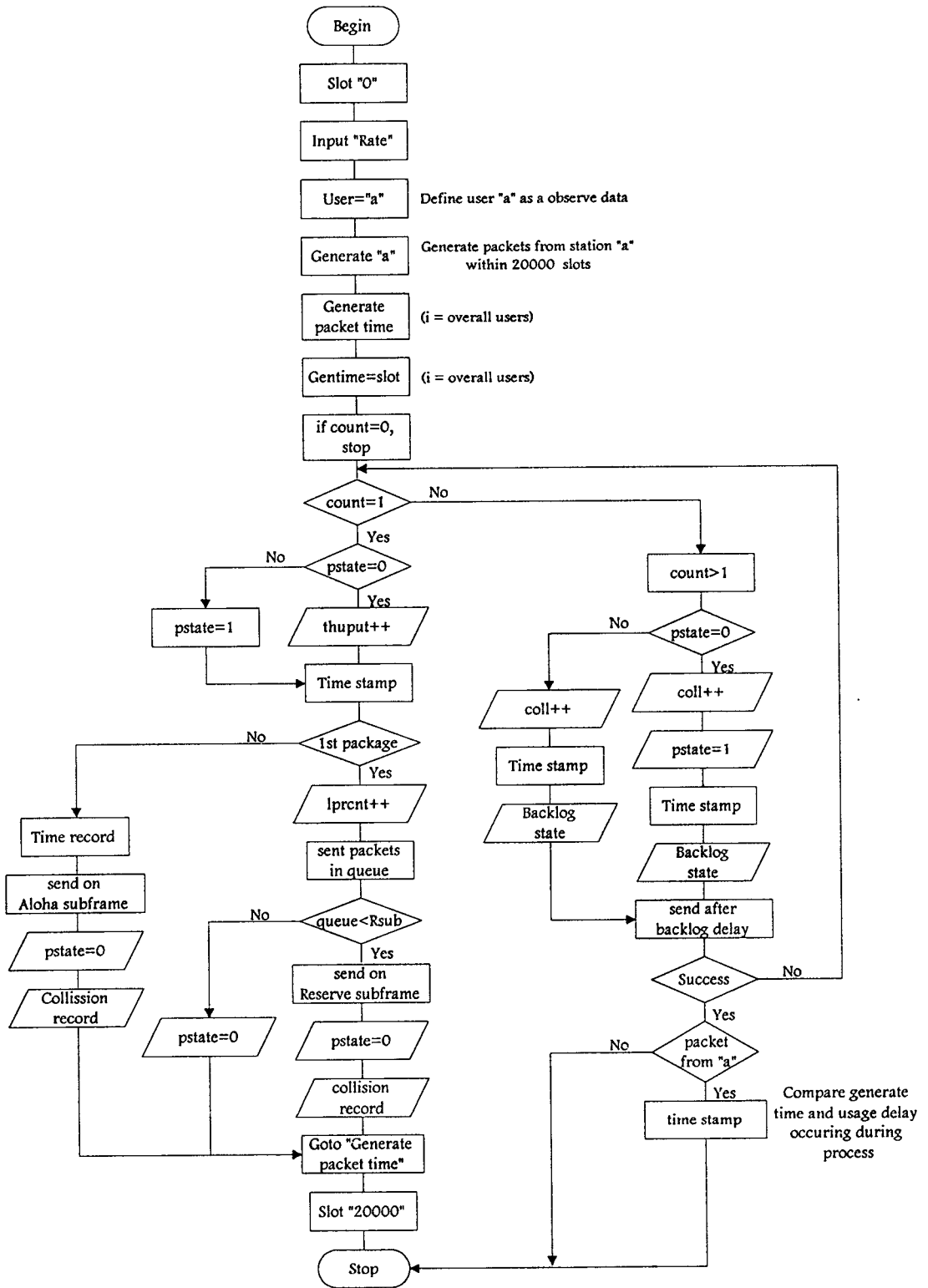
ตัวอย่างการเปรียบเทียบและวิเคราะห์ผลที่ได้จาก สมการที่ 4.37(Calculation) กับผลจากโปรแกรมคอมพิวเตอร์(Simulation) เทียบกับภาระ(load) ที่ค่าต่างๆ กัน ที่ $w=3, L=24, A=0.5, R=6, I_r=0.54, \lambda_r=0.5$ ได้แสดงในรูปของกราฟในภาพที่ 13 และ ค่าความแปรปรวนเทียบกับภาระในภาพที่ 14 โดยให้แต่ละสล็อตเท่ากับ 0.035 วินาทีที่ความเร็วของช่องสัญญาณเท่ากับ 64 kbps เวลาที่ใช้ในการทดสอบเท่ากับ 20,000 สล็อต

สรุปผลการทดลอง

เมื่อพิจารณาผลที่แสดงในกราฟและตารางในภาคผนวกพบว่า ค่าหน่วยเวลาจะเพิ่มสูงขึ้นอย่างรวดเร็ว เมื่อระบบเข้าสู่จุดที่มีภาระ(Load) มากเกินกว่าที่ขนาดของช่องสัญญาณจะรับได้ หรือกล่าวอีกนัยหนึ่งคือ ทราฟฟิกมีขนาดสูงกว่าความจุของช่องสัญญาณ ในกรณีของสมการที่ 4.37 เมื่อเกิดเหตุการณ์ดังกล่าว ค่าที่ได้เกิดจะกลายเป็นค่าที่ติดลบและไม่สามารถคำนวณต่อได้ดังตัวอย่างในภาพที่ 13 ค่าสูงสุดที่สามารถคำนวณได้เป็นค่าสุดท้ายจะอยู่ที่ 0.07 แพคเกจต่อเฟรมสำหรับกราฟที่มีผู้ใช้(User)เท่ากับ 75 และ 0.05 แพคเกจต่อเฟรมสำหรับ กราฟที่มีผู้ใช้เท่ากับ 100

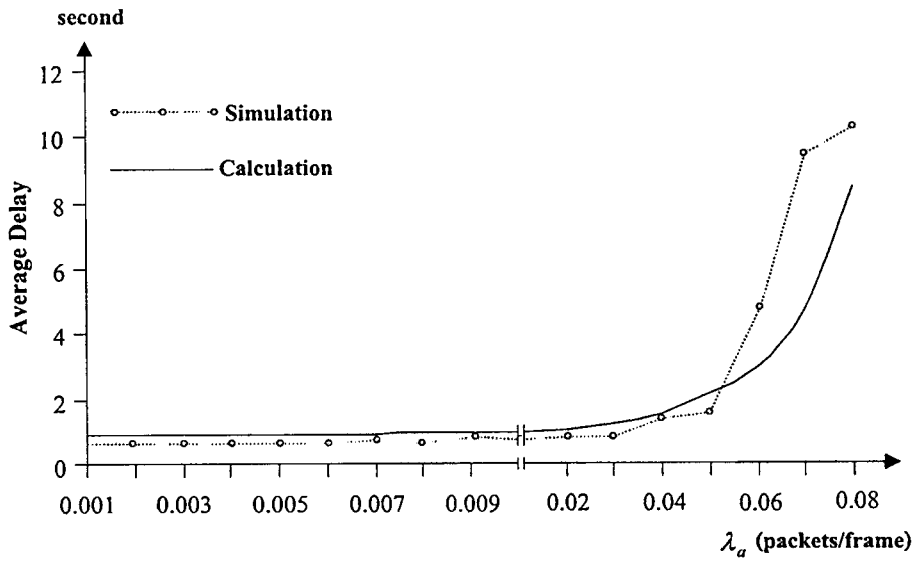
จากผลการทดลองที่ปรากฏดังกล่าว ทำให้สามารถสรุปได้ว่า ผลที่ได้จากโปรแกรมและสมการ โมเดลมีค่าที่ใกล้เคียงกันในช่วงก่อนที่จะเกิดการโอเวอร์โหลด ดังนั้นเพื่อให้สอดคล้องกับจุดประสงค์ในการสร้างโปรแกรมเพื่อใช้ในการออกแบบโครงข่ายแบบผสม ช่วงที่จะนำไปใช้เป็นแนวทางในการพัฒนาตามวัตถุประสงค์ดังกล่าวควรเป็นช่วงก่อนหรือใกล้ที่จะเกิดการโอเวอร์โหลด เช่นสำหรับระบบที่มีผู้ใช้ 75 ราย ช่วงที่เหมาะสมคือ 0.02-0.05 แพคเกจต่อเฟรม และ 0.01-0.04 แพคเกจต่อเฟรม สำหรับระบบที่มีผู้ใช้เท่ากับ 100 ราย เป็นต้น ค่าดังกล่าวจะนำไปใช้เป็นข้อมูลในการคำนวณในบทที่ 5 ต่อไป

ภาพที่ 12

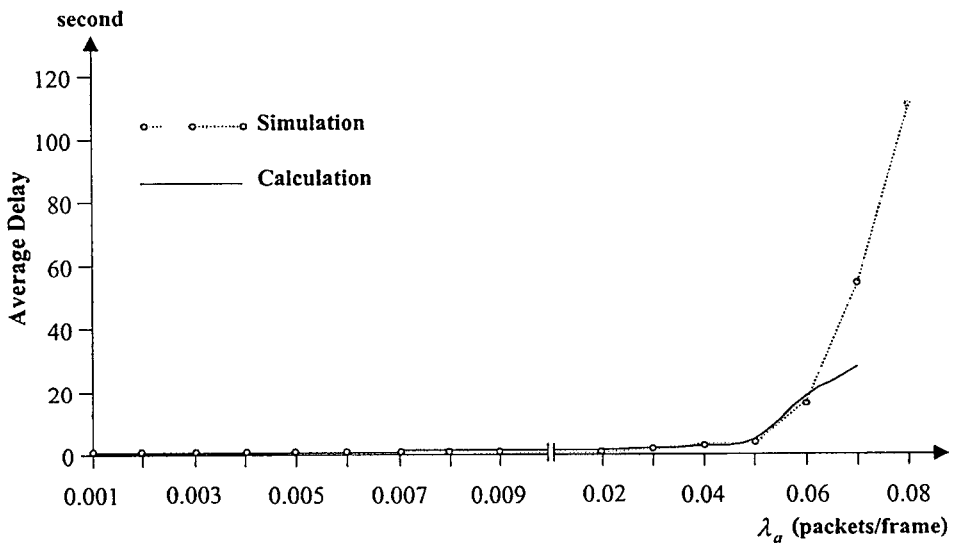


แสดง แผนภูมิจำลองการทำงานของระบบ

ภาพที่ 13

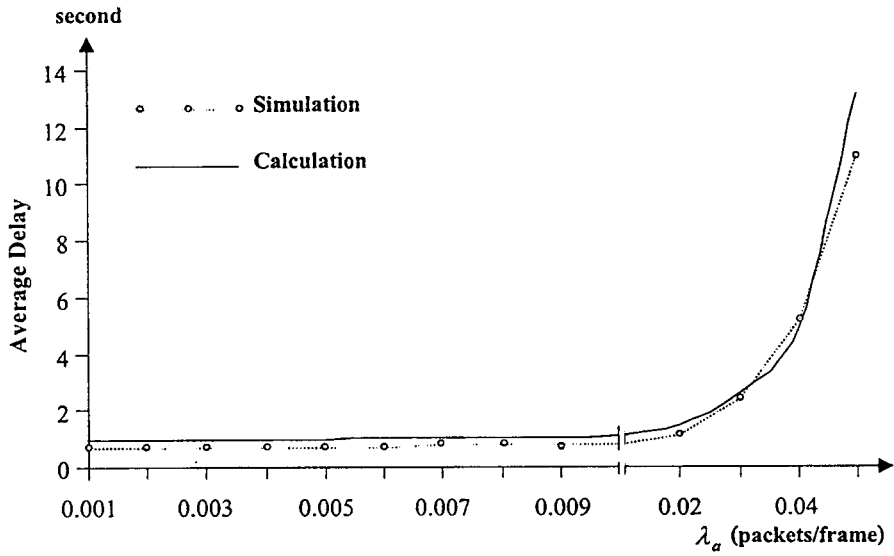


(ก)

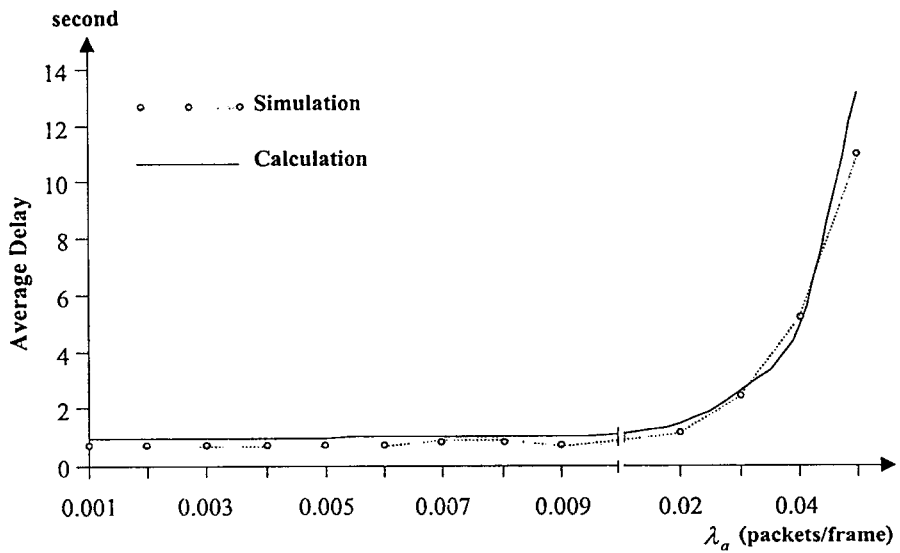


(ข)

ภาพที่ 13 (ต่อ)



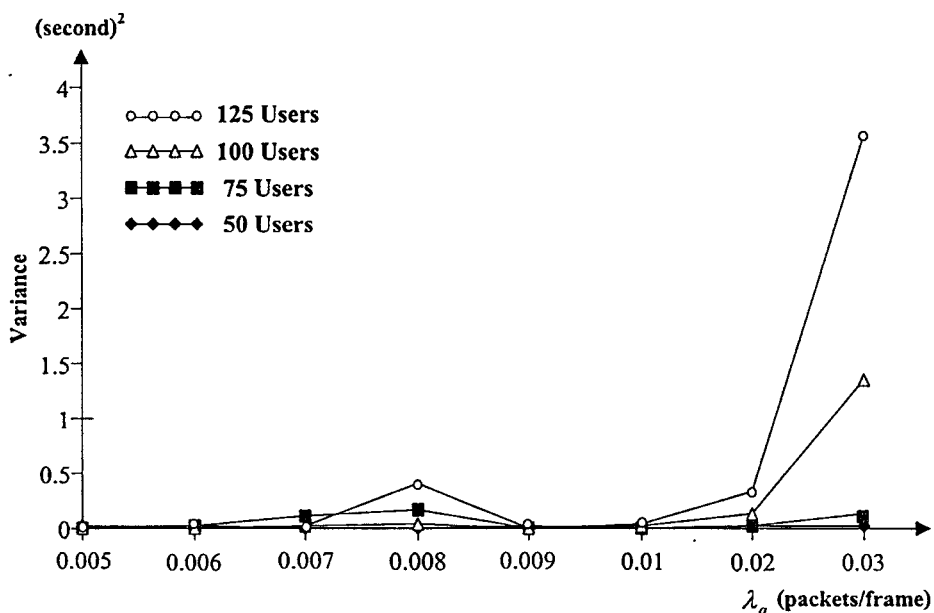
(ค)



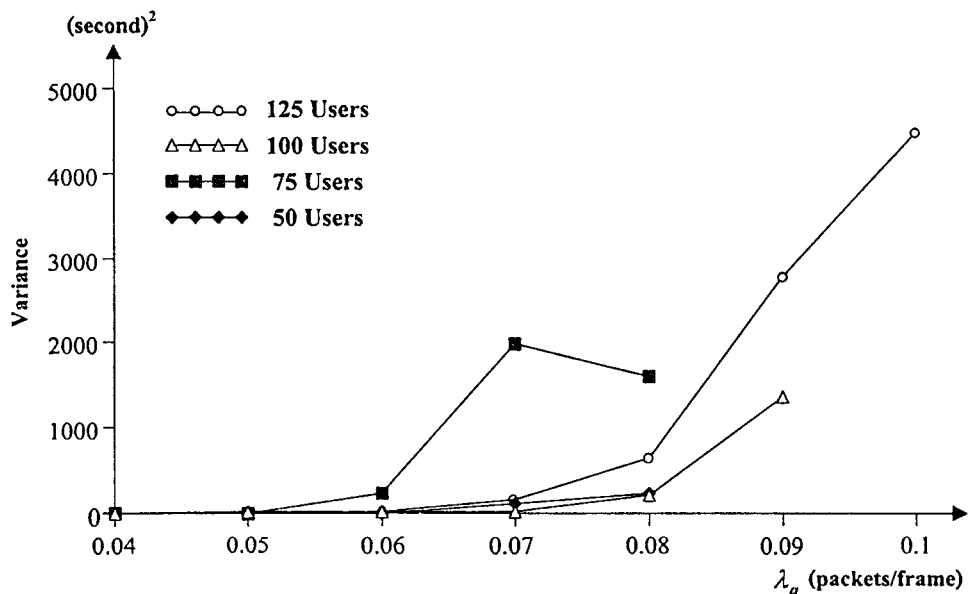
(ง)

แสดง การเปรียบเทียบผลจากสมการ โมเดลกับโปรแกรมคอมพิวเตอร์เทียบกับภาระ(load) ที่ค่า
 ต่างๆ กัน และที่จำนวน user (ก) $N=50$ (ข) $N=75$ (ค) $N=100$ (ง) $N=125$

ภาพที่ 14



(ก)



(ข)

แสดง ค่าความแปรปรวนของค่าหน่วยเวลาที่ได้จากโปรแกรมคอมพิวเตอร์ SIMSAT ที่ค่าอัตรา
มาถึงของแพคเกจต่าง ๆ กัน

บทที่ 5

การวิเคราะห์โครงข่ายแบบผสม โดยใช้หลักการพิจารณาปริมาณกราฟฟิกและเส้นทางผ่านของข้อมูล

คำนำ

ในบทที่สองและบทที่สี่ได้กล่าวถึงการคำนวณค่าหน่วยเวลาโดยเฉลี่ยของแต่ละแพคเกจ ที่เกิดขึ้นในสถานีภาคพื้นดินและดาวเทียมตามลำดับ รวมทั้งการคำนวณเพื่อหาความสัมพันธ์ระหว่างค่าความจุของช่องสัญญาณ(Channel Capacity)และอัตราการมาถึงของแพคเกจ เราสามารถประยุกต์ใช้สมการโมเดลที่สร้างขึ้นสำหรับการคำนวณหาจุดทำงานที่เหมาะสมของทั้งโครงข่ายได้ จุดทำงานที่เหมาะสมดังกล่าวหมายถึง การที่โครงข่ายมีเสถียรภาพ(Reliability) มีการจัดสรรและใช้งานทรัพยากร(ความจุของช่องสัญญาณ ขนาดของข้อมูล อัตราการมาถึงของข้อมูล ฯลฯ)อย่างเหมาะสม และไม่เกิดการแออัดของข้อมูลภายในเส้นทางต่างๆ โดยเฉพาะค่าหน่วยเวลาที่แพคเกจหนึ่งๆที่เกิดจากสถานีใดๆ ได้รับขณะอยู่ในโครงข่ายแบบผสมควรมีค่าที่ไม่แตกต่างกันมากนักเมื่อเดินทางในแต่ละเส้นทาง

จากวัตถุประสงค์ดังกล่าว ทำให้เกิดการพัฒนาวិธีการคำนวณและออกแบบขึ้นในหลายรูปแบบ ในบทวิจัยนี้นำเอาวิธีกำหนดการส่งผ่านของข้อมูล(Flow Deviation Method)ที่มีการใช้งานอยู่อย่างแพร่หลายในปัจจุบันสำหรับการออกแบบโครงข่ายภาคพื้นดิน มาทำการเพิ่มเติมและประยุกต์ให้สามารถพิจารณาโครงข่ายแบบผสมภายใต้ข้อกำหนดที่ได้กำหนดไว้ในบทที่สอง

การกำหนดเส้นทางส่งผ่านข้อมูล

ในการออกแบบและบริหารเครือข่าย เส้นใยที่สำคัญและจำเป็นอย่างยิ่งคือ การกำหนดเส้นทางส่งผ่านข้อมูลทั้งในขณะที่ระบบอยู่ในสภาวะปกติและระบบอยู่ในสภาวะที่บางเส้นทางไม่สามารถให้บริการได้ตามปรกติรวมทั้งเป็นการเพิ่มประสิทธิภาพให้กับโครงข่ายโดยรวมโดยการลด

คำนวณเวลาที่ข้อมูลอยู่ในโครงข่าย

การหาเส้นทางที่สั้นที่สุด(Shortest Path Method) เป็นวิธีการหนึ่งที่สอดคล้องกับวัตถุประสงค์ดังกล่าวและสามารถแบ่งออกได้ในหลายลักษณะ[นรฤทธิ์ : 2539] โดยใช้ข้อกำหนดที่แตกต่างกัน ในบทนี้จะใช้อัลกอริทึมของ Dijkstra (Dijkstra's Algorithm)[Dimitri and Robert 1992 : 401-403]เป็นเครื่องมือสำหรับการหาเส้นทางที่สั้นที่สุดในโปรแกรมสำหรับการออกแบบและวิเคราะห์เครือข่ายแบบผสม จากอัลกอริทึมดังกล่าวสามารถสร้างเป็น โปรแกรมย่อย dijkst(int n) ในโปรแกรมหลัก MG_C (แสดงในภาคผนวก ก.)

การคำนวณคำนวณเวลา

จากบทที่สองและบทที่สี่ สามารถสรุปโมเดลทางคณิตศาสตร์ที่จะนำมาใช้ได้ดังนี้

5.2.1 คำนวณเวลาโดยเฉลี่ยที่แพคเกจเดินทางอยู่ในสถานีภาคพื้นดินภายในโครงข่าย

$$T = \sum_{i=1}^M \frac{1}{\lambda} \left[\frac{f_i}{C_i - f_i} \right] \quad (2.5)$$

โดยที่

$$f = \frac{R_i}{u}$$

C_i = อัตราเร็วของช่องสัญญาณ (bits/sec)

R_i = อัตราโดยเฉลี่ยของแพคเกจที่ผ่านช่องสัญญาณ i (packets/sec)

u = จำนวนของบิตในแต่ละแพคเกจ

λ = อัตราของทราฟฟิคจากภายนอกทั้งหมดที่เข้าสู่โครงข่าย

พิจารณาแนวโน้มของการเปลี่ยนแปลงค่าเวลาในแต่ละสถานีได้จาก

$$\frac{dT}{df} = \frac{1}{\lambda} \left(\frac{C_i}{(C_i - f_i)^2} \right) \quad i = 1, 2, 3, \dots, n \quad (2.6)$$

5.2.2 ค่าหน่วยเวลาโดยเฉลี่ยที่แพคเกจเดินทางอยู่ในสถานีที่ใช้ตัวกลางคววมเทียมภายใน
โครงข่าย

$$T = \frac{u_r}{2} + \frac{(u_r)^2 + u_n \cdot \text{Var}[r] \cdot (1 + u_r)}{2 \cdot u_r \cdot (1 - u_r \cdot u_n)} \quad (5.37)$$

โดยที่

r = เวลาของการให้บริการแต่ละแพคเกจ(ผลจากระบบคววมเทียม)

u_r = เวลาโดยเฉลี่ยของการให้บริการแต่ละแพคเกจ

u_n = จำนวนของแพคเกจโดยเฉลี่ยที่มาถึงในแต่ละครั้งของการให้บริการ

$\text{Var}[R]$ = ค่าความแปรปรวนของจำนวนเวลา(สล็ต)สำหรับ R แพคเกจ
 $= u_n \cdot \text{Var}[r] + (u_r)^2 \cdot \text{Var}[n]$

โมเดลทางคณิตศาสตร์ดังกล่าวจะนำมาใช้เป็นเครื่องมือในการวิเคราะห์และออกแบบโครง
ข่าย ร่วมกับวิธีการหาเส้นทางที่ดีที่สุด(Shortest Path Method) ต่อไป

โครงสร้างและหลักการพิจารณาปริมาณทราฟฟิกภายในโครงข่ายแบบผสม

การพิจารณาและกำหนดปริมาณการส่งผ่านของข้อมูลหรือแพคเกจ ใช้การเคลื่อนย้าย
ปริมาณข้อมูลจำนวนเล็กๆไปยังเส้นทางต่างๆ เพื่อเป็นการกระจายความแออัดของปริมาณข้อมูล
โดยเฉพาะในสถานีที่ใช้ตัวกลางเป็นคววมเทียม และใช้วิธีการหาเส้นทางที่ดีที่สุด(Shortest Path
Method)สร้างเส้นทางที่สอดคล้องกับ ขนาดการส่งผ่านของข้อมูลทุกครั้ง หลังจากที่มีการเคลื่อน
ย้ายปริมาณของข้อมูลรวมทั้งตรวจสอบรูปแบบการเพิ่มของค่าหน่วยเวลาทุกครั้งที่ทำให้การเคลื่อนย้าย
เพื่อเป็นการพิจารณาว่าเป็นไปในทิศทางใด กระบวนการดังกล่าวจะทำให้ค่าต่ำสุด แต่
เนื่องจากช่องสัญญาณคววมเทียมที่อยู่ในโครงข่ายแบบผสมถูกกำหนดให้เป็นสถานีที่ใช้ช่องสัญญาณ
ร่วมกับสถานีคววมเทียมอื่นๆ ที่ไม่เกี่ยวข้องกับโครงข่ายแบบผสมที่กำลังพิจารณา ทำให้การกำหนด
ความจุของช่องสัญญาณหรือ Link Speed ต้องคำนึงถึงผลกระทบที่มีต่อระบบคววมเทียมโดยรวม ใน
บทที่ 3 ได้แสดงให้เห็นถึงความแตกต่างระหว่างความจุของช่องสัญญาณกับเวลาการให้บริการ(รวม
เวลาที่อยู่ในคิว)รวมทั้งผลกระทบที่มีต่อกัน และในบทนี้ได้ทำการวิเคราะห์โดยใช้ความจุของช่อง
สัญญาณคววมเทียมเท่ากับอัตราการมาถึงของแพคเกจที่สภาวะปรกติ(ก่อนเข้าสู่จุดโอเวอร์โหลด) ค่า
ความจุดังกล่าวจะถูกกำหนดให้เป็น อัตราที่สูงสุดของการมาถึงของแพคเกจ ที่สถานีคววมเทียม

สามารถรองรับได้ กำหนดให้เท่ากับ 0.03, 0.04 และ 0.05 แพลกเกตต่อเฟรม ที่จำนวนผู้ใช้ช่องสัญญาณความถี่ 75 และ 100 ราย(เวลาการให้บริการเท่ากับ 1.366, 2.713 และ 2.1945 วินาที ตามลำดับ สำหรับผู้ใช้ 75 ราย และ 1.8913, 2.59 และ 3.4437 วินาที ตามลำดับ สำหรับผู้ใช้ 100 ราย) จากโครงสร้างและข้อกำหนดที่กล่าวมา สามารถนำมาเขียนเป็นอัลกอริทึมได้ดังนี้

อัลกอริทึมของการพิจารณาและกำหนดปริมาณการส่งผ่านของข้อมูล

- 1) กำหนดค่าพารามิเตอร์ของแต่ละ Link เช่น จำนวนสถานี เส้นทาง ต้นทางและปลายทาง อัตราเร็วและชนิดของแต่ละ Link และคำนวณแนวโน้มของการเพิ่มของค่าหน่วยเวลา(Cost)ในแต่ละเส้นทางในขณะที่ยังไม่มีทราฟฟิกในทุกๆ เส้นทาง
- 2) กำหนดค่าทราฟฟิกที่ต้องการสำหรับในแต่ละเส้นทางที่กำหนด
- 3) คำนวณหาเส้นทางที่ดีที่สุด โดยใช้ Cost ของแต่ละเส้นทางดังกล่าวเป็นเงื่อนไขในการพิจารณา และตรวจสอบทราฟฟิกที่เกิดขึ้นในแต่ละเส้นทางเพื่อนำมารวมกัน
- 4) กำหนด(สมมติ) Link Speed ของแต่ละช่องสัญญาณให้เหมาะสมกับอัตราการเกิดของข้อมูล โดยเฉพาะ Link Speed (ความจุของช่องสัญญาณ)ของสถานีที่ถูกกำหนดให้ใช้ตัวกลางดาวเทียม
- 5) คำนวณค่าหน่วยเวลาของระบบทั้งหมดโดยใช้สมการ โมเดลตามชนิดของตัวกลางและเก็บเป็นค่าเปรียบเทียบกับไป

(เริ่มต้นการวนซ้ำ)

- 6) คำนวณแนวโน้มของการเพิ่มของค่าหน่วยเวลา(Cost)ในแต่ละเส้นทาง
- 7) คำนวณหาเส้นทางที่ดีที่สุดสำหรับค่าหน่วยเวลา โดยใช้ Cost ของแต่ละเส้นทางที่ได้จากขั้นตอนที่ 6 และตรวจสอบทราฟฟิกที่เกิดขึ้นในแต่ละเส้นทางเพื่อนำมารวมกัน
- 8) เคลื่อนย้ายข้อมูลบางส่วนของเส้นทาง ที่มีค่าหน่วยเวลาที่ต่ำกว่าไปยังเส้นทางที่มีค่าหน่วยเวลาที่สูงกว่า
- 9) กำหนด(สมมติ) Link Speed หรือ ความจุช่องสัญญาณ ของแต่ละช่องสัญญาณให้เหมาะสม กับอัตราการเกิดของแพกเกตค่าใหม่ในแต่ละ Link
- 10) คำนวณค่าหน่วยเวลาของระบบทั้งหมดและทำการเปรียบเทียบกับค่าหน่วยเวลาเดิมซึ่งถ้า

10.1) ค่าหน่วยเวลาใหม่มีค่ามากกว่าค่าหน่วยเวลาเก่า ให้นำวนกลับไปทำข้อ 6) อีกครั้ง

10.2) ค่าหน่วยเวลาใหม่มีค่าน้อยกว่าหน่วยเวลาเก่า โดยน้อยกว่าเป็นจำนวนที่อยู่ภายนอกช่วงของค่าความมั่นใจ(Confident Interval) ให้นำวนกลับไปทำข้อ 6)

อีกครั้ง

- 10.3) ค่าหน่วยเวลาใหม่มีค่าน้อยกว่าหน่วยเวลาเก่าโดยน้อยกว่าเป็นจำนวนที่อยู่ในช่วงของค่าความมั่นใจ(Confident Interval) แสดงว่าได้ค่าหน่วยเวลาโดยเฉลี่ยที่ต่ำสุดของระบบตามที่ต้องการ : $(\text{Previous delay} - \text{Current delay} > \text{err})$

สำหรับการเคลื่อนย้ายข้อมูลบางส่วนของเส้นทางที่มีค่าหน่วยเวลาที่ต่ำกว่า ไปยังเส้นทางที่มีค่าหน่วยเวลาที่สูงกว่าในข้อ 8) นั้น สามารถทำได้หลายวิธี วิธีหนึ่งที่ทำได้ง่ายและสะดวกคือการหาค่าที่ต่ำสุดของจุดสามจุดที่อยู่ใกล้กัน ก่อนที่จะย้ายจุดไปแทนจุดที่มีค่าสูงเป็นลำดับที่สอง(ค่าดังกล่าวต้องไม่ทำให้การเบี่ยงเบนเข้าสู่จุดต่ำสุดเสียหายไป) และเริ่มกระบวนการใหม่จนกว่าจะได้จุดต่ำสุดที่อยู่ระหว่างสองจุดนั้น[Aaron 1993 : 263-264] โครงสร้างดังกล่าวแสดงในรูปของโปรแกรมย่อย void adjf() ในโปรแกรมหลัก MG_C

จากอัลกอริทึมของการกำหนดปริมาณการไหลของข้อมูลและโปรแกรมย่อย(Function) ที่กล่าวถึงดังกล่าว สามารถสร้างเป็นโปรแกรมสำหรับการออกแบบโครงข่ายแบบผสม(Program MG_C) ดังแสดงในภาคผนวก ก.

ผลการทดลอง

กำหนดให้แต่ละแพคเกจที่เดินทางอยู่ในโครงข่ายมีขนาดโดยเฉลี่ยเท่ากับ 840 bytes คำนวณความจุของช่องสัญญาณโดยเฉลี่ย(Link Speed)ของช่องสัญญาณที่รองรับแพคเกจขนาดดังกล่าวก่อนที่ เท่ากับ

- อัตราเร็ว 14.4 kbps เท่ากับ 128.571 packets/min
- อัตราเร็ว 9600 bps เท่ากับ 85.714 packets/min
- อัตราเร็ว 7200 bps เท่ากับ 64.286 packets/min
- อัตราเร็ว 2400 bps เท่ากับ 21.428 packets/min

สำหรับช่องสัญญาณดาวเทียม กำหนดให้

- แต่ละสล็อตมีขนาดเท่ากับ 0.035 วินาที
- ข้อมูลแต่ละทรานแซกชัน(Transaction)ที่มีขนาดเกินกว่าหนึ่งสล็อตมีขนาดโดยเฉลี่ยเท่ากับ 840 bytes
- อัตราเร็วของแพคเกจหรือทรานแซกชันที่มาถึงแต่ละสถานีเท่ากับ $\lambda_{u,60}/\text{frame}$ แพคเกจ

ต่อมาที่

เนื่องจากระบบดาวเทียมที่นำมาทดลองวิเคราะห์เป็นแบบที่ผู้ใช้สามารถเข้าใช้ช่องสัญญาณเดียวกันได้หลายคน(Multiple Access) ดังนั้นอัตราเร็วของแพคเกจที่มาถึงแต่ละสถานีดาวเทียมที่ใช้ช่องสัญญาณดาวเทียมร่วมกันจะมีผลต่อค่าหน่วยเวลาโดยเฉลี่ยของระบบทั้งหมด ตามที่ได้แสดงไว้ในบทที่ 3 ทำให้การกำหนดอัตราเร็วของแพคเกจที่มาถึงของแต่ละสถานีดาวเทียมมีความจำเป็นและต้องสอดคล้องกับโครงข่ายโดยรวมที่สถานีดังกล่าวประกอบอยู่เช่นกัน

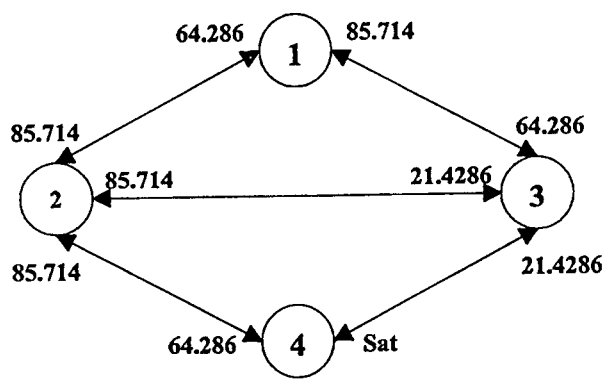
ตัวอย่างที่ 5.1

สมมติให้โครงข่ายมีลักษณะดังภาพที่ 14 แต่ละช่องสัญญาณ(Link)เป็นแบบสองทิศทาง(Undirected Route)และโครงข่ายมีช่องสัญญาณดาวเทียมอยู่หนึ่งช่องสัญญาณคือ Link#9 โดยพารามิเตอร์ของระบบดาวเทียมที่สถานีดังกล่าวรวมอยู่ จะใช้ค่าที่ได้จากการพิจารณาในบทที่ 3 ซึ่งในการทดลองนี้ สมมติให้สถานีดาวเทียมที่เรากำลังพิจารณาอยู่เป็นระบบดาวเทียมที่มีช่องสัญญาณแบบที่ผู้ใช้สามารถใช้งานพร้อมกันได้หลายคนและมีจำนวนผู้ใช้บริการระบบดาวเทียมดังกล่าวเท่ากับ 75 users (กรุณาดูบทที่ 3) พารามิเตอร์ที่ใช้คือ

- ค่าเวลาการให้บริการ(Service time) โดยเฉลี่ย ต่อหนึ่งแพคเกจเท่ากับ 1.7325 วินาที หรือ 0.028875 นาที
- ค่าแวลูชันท์ของเวลาการให้บริการเท่ากับ 14.18 วินาที หรือ 0.23633 นาที
- อัตราเร็วของแพคเกจที่มาถึงแต่ละสถานีดาวเทียมเท่ากับ 0.04762 แพคเกจ ต่อ วินาที หรือ 2.857 แพคเกจ ต่อ นาที

เนื่องจากอัตราการมาถึงของแพคเกจของแต่ละสถานีดาวเทียม มีผลกระทบต่อระบบดาวเทียมโดยรวม ดังที่ได้แสดงไว้ในบทที่ 3 ทำให้ไม่สามารถคำนวณพฤติกรรมของโครงข่ายได้โดยตรงโดยใช้คาบของการมาถึงเป็นค่าการให้บริการสำหรับช่องสัญญาณดาวเทียม ในตัวอย่างนี้จะทำการทดลองโดยใช้โปรแกรม MG_C โดยกำหนดให้ ค่าความจุสูงสุดของช่องสัญญาณดาวเทียม (ตัวแปร : CAP) เท่ากับ 2.857 min(0.04 แพคเกจต่อเฟรม)ซึ่งเป็นค่าที่พิจารณาจากจุดทำงานก่อนที่จะเข้าสู่ภาวะโอเวอร์โหลดของระบบดาวเทียม(แสดงในบทที่ 3) ในขณะที่ค่าเวลาโดยเฉลี่ยของการให้บริการต่อแพคเกจที่ค่าความจุสูงสุดของช่องสัญญาณดาวเทียม(แสดงในบทที่ 4)ดังกล่าว(รวมเวลาที่ต้องสูญเสียภายในคิว : SATD) เท่ากับ 0.028875 min

ภาพที่ 15



แสดง โครงข่ายสำหรับตัวอย่างที่ 5.1

ผลจากการทดลองใช้โปรแกรม MG_C ที่มีอัลกอริทึมตามที่ได้อธิบายไว้ แสดงตามขั้นตอนได้ดังนี้

ขั้นตอนที่ 1

กำหนดค่าพารามิเตอร์ของแต่ละ Link เช่น จำนวนสถานี เส้นทาง ต้นทางและปลายทาง อัตราเร็วและชนิดของแต่ละ Link และคำนวณแนวโน้มของการเพิ่มของค่าหน่วยเวลา(Cost)ในแต่ละเส้นทางขณะที่ยังไม่มีทราฟฟิกในทุกๆเส้นทาง ได้ผลขณะที่ไม่มีทราฟฟิกใดๆเดินทางผ่าน ดังนี้

Setting trunk speed rate.

- Link#1 Traffic=0.0000 : Link Speed=85.7140 : Cost[1][2]=0.012 : Delay=0.012
- Link#2 Traffic=0.0000 : Link Speed=64.2860 : Cost[2][1]=0.016 : Delay=0.016
- Link#3 Traffic=0.0000 : Link Speed=64.2860 : Cost[1][3]=0.016 : Delay=0.016
- Link#4 Traffic=0.0000 : Link Speed=85.7140 : Cost[3][1]=0.012 : Delay=0.012
- Link#5 Traffic=0.0000 : Link Speed=21.4286 : Cost[2][3]=0.047 : Delay=0.047
- Link#6 Traffic=0.0000 : Link Speed=85.7140 : Cost[3][2]=0.012 : Delay=0.012
- Link#7 Traffic=0.0000 : Link Speed=64.2860 : Cost[2][4]=0.016 : Delay=0.016
- Link#8 Traffic=0.0000 : Link Speed=85.7140 : Cost[4][2]=0.012 : Delay=0.012
- Link#9 Traffic=0.0000 : Link Speed=2.8570 : Cost[3][4]=0.350 : Delay=0.029
- Link#10 Traffic=0.0000 : Link Speed=21.4286 : Cost[4][3]=0.047 : Delay=0.047

Total Delay=0.2155 min

แสดง Cost เริ่มต้นของเส้นทางต่างๆ ในรูปของเมทริกซ์(Matrix) ได้ดังนี้

ตารางที่ 1

Node	1	2	3	4
1	0	0.012	0.016	999*
2	0.016	0	0.012	0.016
3	0.012	0.047	0	0.047
4	999*	0.012	0.035	0

999* หมายถึง โหนดทั้งสอง ไม่มีเส้นทางที่เชื่อมต่อกัน

แสดง เมทริกซ์ของ Cost ของช่องสัญญาณต่างๆ

ขั้นตอนที่ 2

ใส่ค่าทราฟฟิกที่ต้องการสำหรับในแต่ละเส้นทางที่กำหนด ซึ่งในตัวอย่างนี้ กำหนดให้

เส้นทางจาก Node1 ไปยัง Node4 มี Traffic เท่ากับ 64 packets/min

เส้นทางจาก Node3 ไปยัง Node2 มี Traffic เท่ากับ 75 packets/min

เส้นทางจาก Node2 ไปยัง Node4 มี Traffic เท่ากับ 1 packets/min

ขั้นตอนที่ 3

ทำการหาเส้นทางที่ดีที่สุด โดยใช้ Cost ของแต่ละเส้นทางเป็นเงื่อนไขในการพิจารณา ได้ผลดังนี้

Find shortest path.

Shortest Path[1] : 1- 2- 4 Total cost[1] =0.027

Shortest Path[2] : 3- 2 Total cost[2] =0.012

Shortest Path[3] : 2- 4 Total cost[3] =0.016

และตรวจสอบทราฟฟิกที่เกิดขึ้นในแต่ละเส้นทางเพื่อนำมารวมกันทำให้ทราฟฟิกที่วิ่งผ่านในแต่ละ Link เท่ากับ

Route [1]-[2]=1 : Assigned traffic=64.000 : Taffic flow=64.000

Route [2]-[4]=7 : Assigned traffic=64.000 : Taffic flow=64.000

Route [3]-[2]=6 : Assigned traffic=75.000 : Taffic flow=75.000

Route [2]-[4]=7 : Assigned traffic=1.000 : Taffic flow=65.000

พารามิเตอร์และ Cost ของแต่ละ Link ณ จุดนี้ เท่ากับ

Link#1 Traffic=64.0000 : Link Speed=85.7140 : Cost[1][2]=0.012

Link#2 Traffic=0.0000 : Link Speed=64.2860 : Cost[2][1]=0.016

Link#3 Traffic=0.0000 : Link Speed=64.2860 : Cost[1][3]=0.016

Link#4 Traffic=0.0000 : Link Speed=85.7140 : Cost[3][1]=0.012

Link#5 Traffic=0.0000 : Link Speed=21.4286 : Cost[2][3]=0.047

Link#6 Traffic=75.0000 : Link Speed=85.714 0 : Cost[3][2]=0.012

Link#7 Traffic=65.0000 : Link Speed=64.2860 : Cost[2][4]=0.016

Link#8 Traffic=0.0000 : Link Speed=85.7140 : Cost[4][2]=0.012

Link#9 Traffic=0.0000 : Link Speed=2.8570 : Cost[3][4]=0.350

Link#10 Traffic=0.0000 : Link Speed=21.4286 : Cost[4][3]=0.047

ขั้นตอนที่ 4

กำหนด(สมมติ) Link Speed ชั่วโมงของแต่ละช่องสัญญาณให้เหมาะสมกับอัตราการเกิดของข้อมูล โดยเฉพาะสถานีที่ถูกกำหนดให้ใช้ตัวกลางดาวเทียม

Link#1 Traffic=64.0000 : Link Speed=88.3993 : Cost[1][2]=0.012

Link#2 Traffic=0.0000 : Link Speed=66.3000 : Cost[2][1]=0.016

Link#3 Traffic=0.0000 : Link Speed=66.3000 : Cost[1][3]=0.016

Link#4 Traffic=0.0000 : Link Speed=88.3993 : Cost[3][1]=0.012

Link#5 Traffic=0.0000 : Link Speed=22.0999 : Cost[2][3]=0.047

Link#6 Traffic=75.0000 : Link Speed=88.3993 : Cost[3][2]=0.012

Link#7 Traffic=65.0000 : Link Speed=66.3000 : Cost[2][4]=0.016

Link#8 Traffic=0.0000 : Link Speed=88.3993 : Cost[4][2]=0.012

Link#9 Traffic=0.0000 : Link Speed=2.9465 : Cost[3][4]=0.350

Link#10 Traffic=0.0000 : Link Speed=22.0999 : Cost[4][3]=0.047

ขั้นตอนที่ 5

Start Delay Time=9999.0 min : Total Delay=1.27255126 min

START LOOPขั้นตอนที่ 6 และ 7

คำนวณแนวโน้มของการเพิ่มของค่าหน่วยเวลา(Cost)ในแต่ละเส้นทางคำนวณหาเส้นทางที่ดีที่สุดสำหรับค่าหน่วยเวลา โดยใช้ Cost ของแต่ละเส้นทางที่ได้จากขั้นตอนที่ 6 ซึ่งผลที่ได้เป็นดังนี้

Shortest Path[1] : 1- 3- 4 Total cost[1] =0.365

Shortest Path[2] : 3- 1- 2 Total cost[2] =0.160

Shortest Path[3] : 2- 1-3- 4 Total cost[3] =0.380

ขั้นตอนที่ 8 และ 9

ตรวจสอบกราฟฟิคที่เกิดขึ้นในแต่ละเส้นทางเพื่อนำมารวมกัน เคลื่อนย้ายข้อมูลบางส่วน of เส้นทาง ที่มีค่าหน่วยเวลาที่ต่ำกว่าไปยังเส้นทางที่มีค่าหน่วยเวลาที่สูงกว่า กำหนด(สมมติ) Link Speed ของแต่ละช่องสัญญาณให้เหมาะสม กับอัตราการเกิดของแพคเกจค่าใหม่ในแต่ละ Link ได้ผลลัพธ์ ดังนี้

Link#1 Traffic = 64.273 : Link Speed = 86.202 : Cost[1][2]=0.14849 : Delay=0.046

Link#2 Traffic = 0.025 : Link Speed = 64.652 : Cost[2][1]=0.01508 : Delay=0.015

Link#3 Traffic = 1.615 : Link Speed = 64.652 : Cost[1][3]=0.01508 : Delay=0.016

Link#4 Traffic = 1.864 : Link Speed = 86.202 : Cost[3][1]=0.01131 : Delay=0.012

Link#5 Traffic = 0.000 : Link Speed = 21.551 : Cost[2][3]=0.04525 : Delay=0.046

Link#6 Traffic = 73.136 : Link Speed = 86.202 : Cost[3][2]=0.49236 : Delay=0.077

Link#7 Traffic = 63.385 : Link Speed = 64.652 : Cost[2][4]=39.23077 : Delay=0.789

Link#8 Traffic = 0.000 : Link Speed = 86.202 : Cost[4][2]=0.01131 : Delay=0.012

Link#9 Traffic = 1.615 : Link Speed = 2.873 : Cost[3][4]=0.35002 : Delay=0.226

Link#10 Traffic = 0.000 : Link Speed = 21.551 : Cost[4][3]=0.04525 : Delay=0.046

ขั้นตอนที่ 10

คำนวณค่าหน่วยเวลาของระบบทั้งหมดและทำการเปรียบเทียบกับค่าหน่วยเวลาเดิม

Previous Delay=1.27255126 min : Current Delay=0.009172340 min

ค่าหน่วยเวลาใหม่มีค่าน้อยกว่าหน่วยเวลาเก่า โดยน้อยกว่าเป็นจำนวนที่อยู่ภายนอกช่วงของค่าความมั่นใจ(Confident Interval) ให้อนกลับไปที่ข้อ 6) อีกครั้ง

ขั้นตอนที่ 6 และ 7

คำนวณแนวโน้มของการเพิ่มของค่าหน่วยเวลา(Cost)ในแต่ละเส้นทางคำนวณหาเส้นทางที่ดีที่สุดสำหรับค่าหน่วยเวลา โดยใช้ Cost ของแต่ละเส้นทางที่ได้จากขั้นตอนที่ 6 ซึ่งผลที่ได้เป็นดังนี้.

Shortest Path[1] : 1- 3- 4	Total cost[1] =1.870
Shortest Path[2] : 3- 1- 2	Total cost[2] =0.191
Shortest Path[3] : 2- 1-3- 4	Total cost[3] =1.885

ขั้นตอนที่ 8 และ 9

ตรวจสอบกราฟฟิคที่เกิดขึ้นในแต่ละเส้นทางเพื่อนำมารวมกัน เคลื่อนย้ายข้อมูลบางส่วน of เส้นทาง ที่มีค่าหน่วยเวลาที่ต่ำกว่าไปยังเส้นทางที่มีค่าหน่วยเวลาที่สูงกว่า กำหนด(สมมติ) Link Speed ของแต่ละช่องสัญญาณให้เหมาะสม กับอัตราการเกิดของแพคเกจค่าใหม่ในแต่ละ Link ได้ผลลัพธ์ ดังนี้

Link#1 Traffic = 64.483	: Link Speed = 87.428	: Cost[1][2]=0.17926	: Delay=0.044
Link#2 Traffic = 0.044	: Link Speed = 65.571	: Cost[2][1]=0.01548	: Delay=0.015
Link#3 Traffic = 2.857	: Link Speed = 65.571	: Cost[1][3]=0.01627	: Delay=0.016
Link#4 Traffic = 3.297	: Link Speed = 87.428	: Cost[3][1]=0.01212	: Delay=0.012
Link#5 Traffic = 0.000	: Link Speed = 21.857	: Cost[2][3]=0.04640	: Delay=0.046
Link#6 Traffic = 71.703	: Link Speed = 87.428	: Cost[3][2]=0.50491	: Delay=0.064
Link#7 Traffic = 62.143	: Link Speed = 65.571	: Cost[2][4]=40.23061	: Delay=0.292
Link#8 Traffic = 0.000	: Link Speed = 87.428	: Cost[4][2]=0.01160	: Delay=0.011
Link#9 Traffic = 2.857	: Link Speed = 2.914	: Cost[3][4]=1.85339	: Delay=0.377
Link#10 Traffic = 0.000	: Link Speed = 21.857	: Cost[4][3]=0.04640	: Delay=0.046

ขั้นตอนที่ 10

คำนวณค่าหน่วยเวลาของระบบทั้งหมดและทำการเปรียบเทียบกับค่าหน่วยเวลาเดิม

Previous Delay=0.009172340 min : Current Delay=0.006585538 min

กำหนดวงเวลาใหม่มีค่าน้อยกว่าวงเวลาก่า โดยน้อยกว่าเป็นจำนวนที่อยู่ภายนอกช่วงของค่าความมั่นใจ(Confident Interval) ให้นำกลับไปทำข้อ 6) อีกครั้ง

ขั้นตอนที่ 6 และ 7

คำนวณแนวโน้มของการเพิ่มของกำหนดวงเวลา(Cost)ในแต่ละเส้นทางคำนวณหาเส้นทางที่ดีที่สุดสำหรับกำหนดวงเวลา โดยใช้ Cost ของแต่ละเส้นทางที่ได้จากขั้นตอนที่ 6 ซึ่งผลที่ได้เป็นดังนี้

Shortest Path[1] : 1- 2- 4	Total cost[1] =5.745
Shortest Path[2] : 3- 1- 2	Total cost[2] =0.178
Shortest Path[3] : 2- 4	Total cost[3] =5.579

ขั้นตอนที่ 8 และ 9

ตรวจสอบกราฟฟิคที่เกิดขึ้นในแต่ละเส้นทางเพื่อนำมารวมกัน เคลื่อนย้ายข้อมูลบางส่วนของเส้นทาง ที่มีกำหนดวงเวลาที่ต่ำกว่าไปยังเส้นทางที่มีกำหนดวงเวลาที่สูงกว่า กำหนด(สมมติ) Link Speed ของแต่ละช่องสัญญาณให้เหมาะสมกับอัตราการเกิดของแพคเกจค่าใหม่ในแต่ละ Link ได้ผลลัพธ์ ดังนี้

Link#1 Traffic = 69.569	: Link Speed = 85.714	: Cost[1][2]=0.16607	: Delay=0.062
Link#2 Traffic = 0.041	: Link Speed = 64.286	: Cost[2][1]=0.01527	: Delay=0.016
Link#3 Traffic = 2.662	: Link Speed = 64.286	: Cost[1][3]=0.01667	: Delay=0.016
Link#4 Traffic = 8.190	: Link Speed = 85.714	: Cost[3][1]=0.01235	: Delay=0.013
Link#5 Traffic = 0.000	: Link Speed = 21.429	: Cost[2][3]=0.04575	: Delay=0.047
Link#6 Traffic = 66.810	: Link Speed = 85.714	: Cost[3][2]=0.35359	: Delay=0.053
Link#7 Traffic = 62.338	: Link Speed = 64.286	: Cost[2][4]=5.57863	: Delay=0.513
Link#8 Traffic = 0.000	: Link Speed = 85.714	: Cost[4][2]=0.01144	: Delay=0.012
Link#9 Traffic = 2.662	: Link Speed = 2.857	: Cost[3][4]=999.0	: Delay=0.353
Link#10 Traffic = 0.000	: Link Speed = 21.429	: Cost[4][3]=0.04575	: Delay=0.047

ขั้นตอนที่ 10

คำนวณกำหนดวงเวลาของระบบทั้งหมดและทำการเปรียบเทียบกับกำหนดวงเวลาเดิม

Previous Delay=0.006585538 min : Current Delay=0.008079586 min

กำหนดวงเวลาใหม่มีค่ามากกว่ากำหนดวงเวลาเก่า ให้วนกลับไปทำข้อ 6) อีก

ขั้นตอนที่ 6 และ 7

คำนวณแนวโน้มของการเพิ่มของกำหนดวงเวลา(Cost)ในแต่ละเส้นทางคำนวณหาเส้นทางที่ดีที่สุดสำหรับกำหนดวงเวลา โดยใช้ Cost ของแต่ละเส้นทางที่ได้จากขั้นตอนที่ 6 ซึ่งผลที่ได้เป็นดังนี้

Shortest Path[1] : 1- 3- 2-4	Total cost[1] =17.198
Shortest Path[2] : 3- 2	Total cost[2] =0.240
Shortest Path[3] : 2- 4	Total cost[3] =16.941

ขั้นตอนที่ 8 และ 9

ตรวจสอบกราฟฟิคที่เกิดขึ้นในแต่ละเส้นทางเพื่อนำมารวมกัน เคลื่อนย้ายข้อมูลบางส่วน of เส้นทาง ที่มีกำหนดวงเวลาที่ต่ำกว่าไปยังเส้นทางที่มีกำหนดวงเวลาที่สูงกว่า กำหนด(สมมติ) Link Speed ของแต่ละช่องสัญญาณให้เหมาะสม กับอัตราการเกิดของแพคเกจค่าใหม่ในแต่ละ Link ได้ผลลัพธ์ ดังนี้

Link#1 Traffic = 69.569	: Link Speed = 85.714	: Cost[1][2]=0.32884	: Delay=0.062
Link#2 Traffic = 0.041	: Link Speed = 64.286	: Cost[2][1]=0.01558	: Delay=0.016
Link#3 Traffic = 2.662	: Link Speed = 64.286	: Cost[1][3]=0.01693	: Delay=0.016
Link#4 Traffic = 8.190	: Link Speed = 85.714	: Cost[3][1]=0.01426	: Delay=0.013
Link#5 Traffic = 0.000	: Link Speed = 21.429	: Cost[2][3]=0.04667	: Delay=0.013
Link#6 Traffic = 66.810	: Link Speed = 85.714	: Cost[3][2]=0.23985	: Delay=0.053
Link#7 Traffic = 62.338	: Link Speed = 64.286	: Cost[2][4]=16.94101	: Delay=0.513
Link#8 Traffic = 0.000	: Link Speed = 85.714	: Cost[4][2]=0.01167	: Delay=0.012
Link#9 Traffic = 2.662	: Link Speed = 2.857	: Cost[3][4]=75.13346	: Delay=0.353
Link#10 Traffic = 0.000	: Link Speed = 21.429	: Cost[4][3]=0.04667	: Delay=0.047

ขั้นตอนที่ 10

คำนวณกำหนดวงเวลาของระบบทั้งหมดและทำการเปรียบเทียบกับกำหนดวงเวลาเดิม

Previous Delay=0.008079586 min : Current Delay=0.008079586 min

ค่าหน่วงเวลาใหม่มีค่าน้อยกว่าหน่วงเวลาเก่า โดยน้อยกว่าเป็นจำนวนที่อยู่ในช่วงของ ค่าความมั่นใจ(Confident Interval) แสดงว่าได้ค่าหน่วงเวลาโดยเฉลี่ยที่ต่ำสุดของระบบตามที่ต้องการ (Previous delay - Current delay > err)

พิจารณาเปรียบเทียบตัวอย่างที่ 1 โดยเปลี่ยนโครงข่ายในภาพที่ 14 ที่แต่เดิม โครงข่ายมีช่องสัญญาณที่ใช้ตัวกลางความถี่ขมอยู่หนึ่งช่องสัญญาณคือ Link#9 ให้ช่องสัญญาณดังกล่าวเป็น สถานีที่ใช้ตัวกลางภาคพื้นดิน ที่มีอัตราเร็วของแพคเกจที่มาถึงแต่ละสถานีเท่าเดิม คือ 2.857 แพคเกจ ต่อ นาที และสถานีอื่นๆยังคงค่าพารามิเตอร์ดั้งเดิม เราจะได้ผลลัพธ์สุดท้ายจากการทดลองใช้โปรแกรม MG_C ที่เปลี่ยนแปลงค่าตามที่ได้กำหนดไว้ ดังนี้

ขั้นตอนที่ 6 และ 7

Shortest Path[1] : 1- 3- 4 Total cost[1]=1.821

Shortest Path[2] : 3- 1- 2 Total cost[2]=0.191

Shortest Path[3] : 2- 1- 3- 4 Total cost[3]=1.837

ขั้นตอนที่ 8 และ 9

Link#1 Traffic = 64.273 : Link Speed = 86.207 : Cost[1][2]=0.17918

Link#2 Traffic = 0.025 : Link Speed = 64.656 : Cost[2][1]=0.01548

Link#3 Traffic = 1.612 : Link Speed = 64.656 : Cost[1][3]=0.01627

Link#4 Traffic = 1.860 : Link Speed = 86.207 : Cost[3][1]=0.01212

Link#5 Traffic = 0.000 : Link Speed = 21.552 : Cost[2][3]=0.04640

Link#6 Traffic = 73.140 : Link Speed = 86.207 : Cost[3][2]=0.50488

Link#7 Traffic = 63.388 : Link Speed = 64.656 : Cost[2][4]=40.22827

Link#8 Traffic = 0.000 : Link Speed = 86.207 : Cost[4][2]=0.01160

Link#9 Traffic = 1.612 : Link Speed = 2.873 : Cost[3][4]=1.80504

Link#10 Traffic = 0.000 : Link Speed = 21.552 : Cost[4][3]=0.04640

ขั้นตอนที่ 10

คำนวณค่าหน่วงเวลาของระบบทั้งหมดและทำการเปรียบเทียบกับค่าหน่วงเวลาเดิม

Previous Delay=0.013221955 min : Current Delay=0.013221955 min

สิ้นสุดการทำงาน

จากผลการทดลองใช้โปรแกรมที่แก้ไข พบว่าโปรแกรมทำการวนซ้ำเป็นจำนวน 17 ครั้ง ก่อนที่จะสิ้นสุดการทำงาน พารามิเตอร์ต่างๆ ของแต่ละ Link ที่ได้ มีค่าที่แตกต่างจากการทดลอง โดยใช้โปรแกรมเดิม โดยเฉพาะอย่างยิ่งค่าหน่วยเวลาโดยเฉลี่ยของโครงข่ายทั้งหมด(ผลจากโปรแกรมเดิม Delay=0.008079586 min : ผลจากโปรแกรมที่แก้ไข Delay = 0.0132219 55 min) เมื่อพิจารณาการกระจายของกราฟฟิคพบว่า โปรแกรมเดิมจะทำได้ดีกว่า เนื่องจากค่าเวลาของการให้บริการของสถานีที่ใช้ตัวกลางดาวเทียม(ในกรณีนี้แตกต่างจากค่าหน่วยเวลาของแพคเกจที่กล่าวถึง โดยทั่วไปสำหรับสถานีที่ใช้ตัวกลางภาคพื้นดิน เนื่องจากเวลาที่แพคเกจต้องสูญเสียไปภายในคิวและเวลาการให้บริการมีค่าน้อยมากในสภาวะปรกติเมื่อเทียบกับเวลาที่ต้องสูญเสียไปภายในคิวและเวลาการให้บริการของสถานีดาวเทียม) ไม่เป็นตัวแปรตามกับค่าอัตราการมาถึงของแพคเกจจนกว่าจะเข้าสู่อัตราเฉลี่ยที่ได้กำหนดไว้เป็นค่าสูงสุด ตรงกันข้ามกับค่าแนวโน้มของการเพิ่มของค่าหน่วยเวลา(Cost) ซึ่งถูกกำหนดให้แปรตามอัตราการมาถึงและความจุของช่องสัญญาณ(Link Speed) ทำให้การกระจายกราฟฟิคไปยังสถานีที่ใช้ตัวกลางดาวเทียม ใกล้เคียงความเป็นจริงมากขึ้น

ข้อที่น่าสังเกตอีกประการหนึ่งคือเมื่อถึงการวนรอบของการทำงานครั้งสุดท้าย Link Speed ไม่สามารถปรับเข้าสู่สภาวะเดิม ซึ่งหมายถึงช่องสัญญาณบางเส้นทางมีกราฟฟิคมากเกินไป Link Speed ของช่องสัญญาณดังกล่าวจะรองรับได้ ทั้งนี้โดยสภาพที่เป็นจริง ช่องสัญญาณที่เป็นดาวเทียมสามารถรองรับ Traffic ได้มากกว่าค่าที่ปรากฏจากเหตุผลที่ได้กล่าวมาแล้วข้างต้น

ตัวอย่างที่ 5.2

ในตัวอย่างนี้ จำนวนโหนดเพิ่มขึ้นเป็น 10 โหนดและแต่ละเส้นทางของช่องสัญญาณ(Link) ถูกกำหนด(Directed Route)ให้มีทิศทางเดียว โดยสมมติให้โครงข่ายมีลักษณะดังแสดงในภาพที่ 16 และกำหนดให้ภายในโครงข่ายมีจำนวนช่องสัญญาณดาวเทียมอยู่สองช่องสัญญาณคือช่องสัญญาณที่เชื่อมต่อกับ โหนด4ไปยังโหนด3(Link#8)และช่องสัญญาณที่เชื่อมต่อกับ โหนด6ไปยัง โหนด4(Link#9) พารามิเตอร์ของระบบดาวเทียมที่สถานีดังกล่าวเชื่อมต่ออยู่กำหนดให้ใช้ค่าเดียวกับตัวอย่างที่ 5.1 ที่ช่องสัญญาณดาวเทียมที่มีผู้ให้บริการเท่ากับ 75 users พารามิเตอร์ดังกล่าวเท่ากับ

- ค่าเวลาการให้บริการ(Service time) โดยเฉลี่ย ต่อหนึ่งแพคเกจ 0.028875 นาที
- ค่า Variance ของเวลาการให้บริการเท่ากับ 0.23633 นาที
- อัตราเร็วของแพคเกจที่มาถึงแต่ละสถานีดาวเทียมเท่ากับ 2.857 แพคเกจ ต่อ นาที

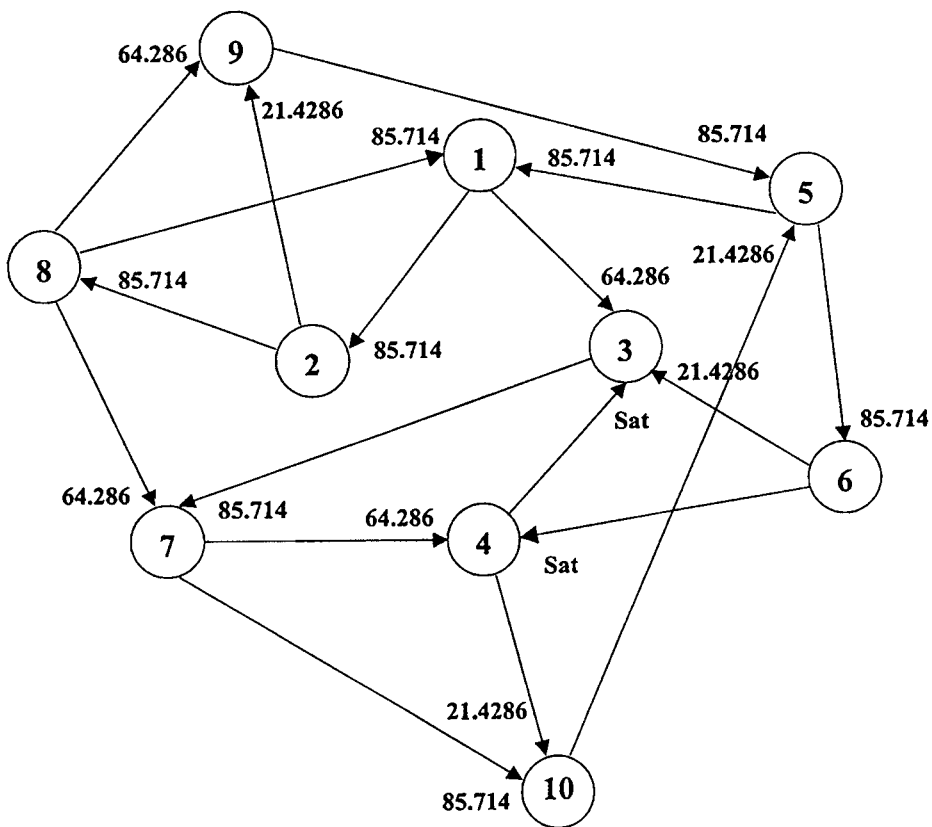
และกราฟฟิคที่ต้องการสำหรับในแต่ละเส้นทางที่กำหนดให้ มีค่าดังนี้

เส้นทางจาก Node1 ไปยัง Node4 มี Traffic เท่ากับ 51.0 packets/min

เส้นทางจาก Node3 ไปยัง Node10 มี Traffic เท่ากับ 47.0 packets/min

เส้นทางจาก Node2 ไปยัง Node6 มี Traffic เท่ากับ 43.0 packets/min
 เส้นทางจาก Node5 ไปยัง Node8 มี Traffic เท่ากับ 44.0 packets/min
 เส้นทางจาก Node6 ไปยัง Node4 มี Traffic เท่ากับ 1.5 packets/min
 เส้นทางจาก Node4 ไปยัง Node2 มี Traffic เท่ากับ 5.0 packets/min
 เส้นทางจาก Node5 ไปยัง Node3 มี Traffic เท่ากับ 28.0 packets/min

ภาพที่ 16



แสดง โครงข่ายสำหรับตัวอย่างที่ 5.2

ผลจากการทดลองใช้โปรแกรม MG_C เฉพาะที่เกิดขึ้นกับช่องสัญญาณที่เป็นตัวกลางดาวเทียมแสดงในตารางที่ 2 และแสดงในรูปของกราฟเปรียบเทียบค่าการเปลี่ยนแปลงของทราฟฟิกสำหรับ Link# 8 และ 9 ซึ่งเป็นช่องสัญญาณดาวเทียม กับ ความจุของช่องสัญญาณ(Link Speed) ในภาพที่ 17 (ผลการทดลองทั้งหมดจากการทดลองรัน โปรแกรม MG_C แสดงตามขั้นตอนในภาคผนวก ค.)

ตารางที่ 2

Loop	Traffic		Link Speed		Avr Delay		Cost	
	No.	Link#8	Link#9	Link#8	Link#9	Link#8	Link#9	Link#8
0	0	0	2.857	2.857	0.029	0.029	0.35	0.35
1	0	0.282	3.093	3.093	0.029	0.063	0.32335	0.39141
2	0	2.395	2.989	2.989	0.029	0.321	0.33457	8.4836
3	0	0.2857	2.976	2.976	0.029	0.377	0.33601	209.73
4	0.422	2.616	2.959	2.959	0.08	0.348	0.4598	25.0631
5	0.409	2.53	2.912	2.912	0.079	0.337	0.46473	20.0415
6	0.532	2.462	2.894	2.894	0.094	0.329	0.51878	15.4947
7	0.645	2.438	2.882	2.882	0.107	0.326	0.57569	14.5953
8	0.71	2.424	2.869	2.869	0.115	0.324	0.61575	14.4883
9	0.721	2.547	2.866	2.866	0.117	0.339	0.62292	28.1268
10	0.755	2.538	2.861	2.861	0.121	0.338	0.64523	27.4138
11	0.77	2.529	2.857	2.857	0.123	0.337	0.65601	26.5243
12	0.783	2.526	2.857	2.857	0.124	0.337	0.66385	26.0320
13	0.783	2.524	2.857	2.857	0.124	0.337	0.66287	25.7277
14	0.783	2.523	2.857	2.857	0.124	0.337	0.66432	25.6427
15	0.783	2.523	2.857	2.857	0.124	0.337	0.66432	25.6427

แสดง ผลการทำงานของโปรแกรม MG_C สำหรับตัวอย่างที่ 5.2

เมื่อพิจารณาผลการทดลองที่ปรากฏในตารางที่ 2 ปรากฏว่า ปริมาณทราฟฟิกและความจุของช่องสัญญาณดาวเทียม(Link Speed) มีการเปลี่ยนแปลงในลักษณะของการสวิง(Damping) เพื่อปรับค่าเข้าสู่จุดที่ทำให้ค่าหน่วงเวลาโดยรวมของระบบมีค่าที่ต่ำสุดในช่วงความมั่นใจที่ได้กำหนดไว้ รูปแบบของการปรับค่า มีลักษณะเดียวกับช่องสัญญาณที่ใช้ตัวกลางภาคพื้นดินอื่นๆ ดังแสดงในภาพที่ 18 ซึ่งเป็นกราฟแสดงปริมาณทราฟฟิกที่เปลี่ยนแปลงของ Link#1 และ Link#18 ที่ Link Speed เท่ากับ 85.714 และ ภาพที่ 19 ซึ่งเป็นกราฟแสดงปริมาณทราฟฟิกที่เปลี่ยนแปลงของ Link#5 และ Link#6 ที่ Link Speed เท่ากับ 21.429 ขนาดของการสวิงจะขึ้นอยู่กับปริมาณทราฟฟิก

ของแต่ละเส้นทาง ที่มี Link ดังกล่าวเป็นส่วนหนึ่งของเส้นทางที่สั้นที่สุดที่เป็นไปได้ของเส้นทางนั้นๆ ตัวอย่างเช่น Link#9 เป็นส่วนหนึ่งของเส้นทางที่สั้นที่สุดของเส้นทางจาก Node 1 ไปยัง Node 4 ในขณะที่ตนเองมีเส้นทางหลักคือเส้นทางจาก Node 6 ไปยัง Node 4 ทำให้ทราฟฟิคที่วิ่งอยู่ใน Link#9 มาจากทราฟฟิค ของเส้นทางจาก Node 1 ไปยัง Node 4 รวมกับทราฟฟิคของเส้นทางจาก Node 6 ไปยัง Node 4 ส่งผลให้ปริมาณทราฟฟิคและการสวิงเข้าสู่จุดที่ต้องการมีขนาดมากกว่าการสวิงของ Link#8 ที่มีการจัดสรรเฉพาะทราฟฟิคของตนเองเท่านั้น ดังนั้นเราจะได้ผลลัพธ์ในแต่ละช่องสัญญาณเพื่อนำไปใช้งานดังนี้

Link#1 Ch[1][2]	Traffic = 60.712	Link Speed = 85.714
Link#2 Ch[2][8]	Traffic = 83.959	Link Speed = 85.714
Link#3 Ch[1][3]	Traffic = 54.958	Link Speed = 64.286
Link#4 Ch[1][3]	Traffic = 83.950	Link Speed = 85.714
Link#5 Ch[2][9]	Traffic = 19.752	Link Speed = 21.429
Link#6 Ch[6][3]	Traffic = 14.209	Link Speed = 21.429
Link#7 Ch[7][4]	Traffic = 49.977	Link Speed = 64.286
Link#8 Ch[4][3]	Traffic = 0.783	Link Speed = 2.857
Link#9 Ch[6][4]	Traffic = 2.523	Link Speed = 2.857
Link#10 Ch[4][10]	Traffic = 4.217	Link Speed = 21.429
Link#11 Ch[7][10]	Traffic = 47.000	Link Speed = 85.714
Link#12 Ch[8][7]	Traffic = 13.027	Link Speed = 64.286
Link#13 Ch[10][5]	Traffic = 4.217	Link Speed = 21.429
Link#14 Ch[5][1]	Traffic = 64.669	Link Speed = 85.714
Link#15 Ch[8][9]	Traffic = 26.933	Link Speed = 64.286
Link#16 Ch[8][1]	Traffic = 0.000	Link Speed = 85.714
Link#17 Ch[9][5]	Traffic = 46.685	Link Speed = 85.714
Link#18 Ch[5][6]	Traffic = 58.232	Link Speed = 85.714

โดยที่คำนวณเวลารวมโดยเฉลี่ยของโครงข่ายเท่ากับ 0.013024349 min

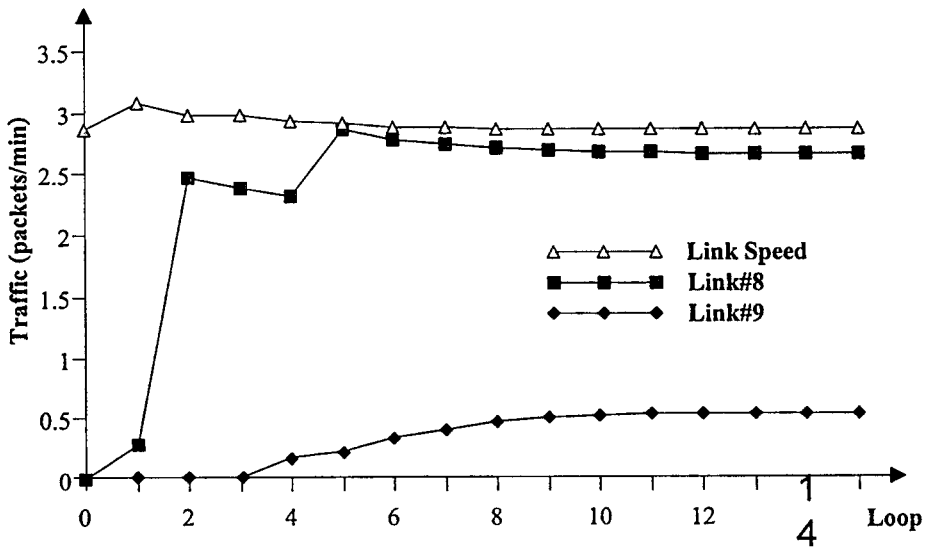
ผลที่ได้จากโปรแกรมที่แสดงโดยกราฟในภาพต่างๆ แสดงการเปลี่ยนแปลงของทราฟฟิค และคำนวณเวลาที่เข้าสู่จุดที่ทำให้คำนวณเวลาโดยรวมของระบบต่ำที่สุด เป็นไปในลักษณะเดียว

กับการเปลี่ยนแปลงของระบบที่มีแต่สถานีที่ใช้ตัวกลางภาคพื้นดินและใช้โมเดลเดิมที่มีอยู่ในปัจจุบันในการออกแบบ ดังนั้นเมื่อใช้โมเดลคำนวณค่าหน่วยเวลาสำหรับสถานีดาวเทียม (สมการที่ 4.37) แทนโมเดลสำหรับคำนวณค่าหน่วยเวลาของตัวกลางภาคพื้นดินในอัลกอริทึมเดิม การเปลี่ยนแปลงยังคงเป็นไปในลักษณะที่สอดคล้องกับข้อกำหนด และรูปแบบเดิมของการทำงานที่ผู้ออกแบบอัลกอริทึมพื้นฐานได้กำหนดไว้

ในส่วนของค่าหน่วยเวลา โดยปกติค่าหน่วยเวลาและ Cost ของช่องสัญญาณที่ใช้ตัวกลางดาวเทียมจะมีค่าที่สูงกว่าตัวกลางภาคพื้นดิน ทำให้เส้นทางที่ใช้ตัวกลางดาวเทียมเป็นจุดที่วิกฤต จุดหนึ่งของระบบ สามารถพิสูจน์และเปรียบเทียบโดยการเปลี่ยนโครงข่ายให้มีแต่สถานีที่ใช้ตัวกลางภาคพื้นดินแทนสถานีดาวเทียม โดยสมมติให้สถานีที่ใช้ตัวกลางภาคพื้นดินมีอัตราเร็วของแพคเกจที่มากถึงแต่ละสถานีเท่ากับสถานีที่ใช้ตัวกลางดาวเทียม คือ 2.857 แพคเกจต่ออนาที และสถานีอื่นยังคงค่าพารามิเตอร์ดั้งเดิม พบว่าเกิดการวนซ้ำเป็นจำนวน 12 ครั้งก่อนที่จะสิ้นสุดการทำงานดังแสดงในตารางที่ 3 แต่ระบบไม่สามารถปรับตัวเข้าสู่สภาวะปกติได้(พิจารณาจาก Link Speed ที่กลายเป็น 2.975 แพคเกจ ต่อ นาที แต่ในสภาวะปกติเท่ากับ 2.857 แพคเกจต่ออนาที) ในขณะที่ค่าหน่วยเวลาและ Cost ใกล้เคียงกับช่องสัญญาณอื่นๆ ในสภาวะของกราฟฟิกเดียวกัน ซึ่งผิดจากสภาพที่เป็นจริง การปรับเพื่อให้ระบบเข้าสู่สภาวะปกติทำได้โดยการลดกราฟฟิกในบางเส้นทางลง หรือเพิ่มขนาดของ Link Speed ให้เหมาะสม

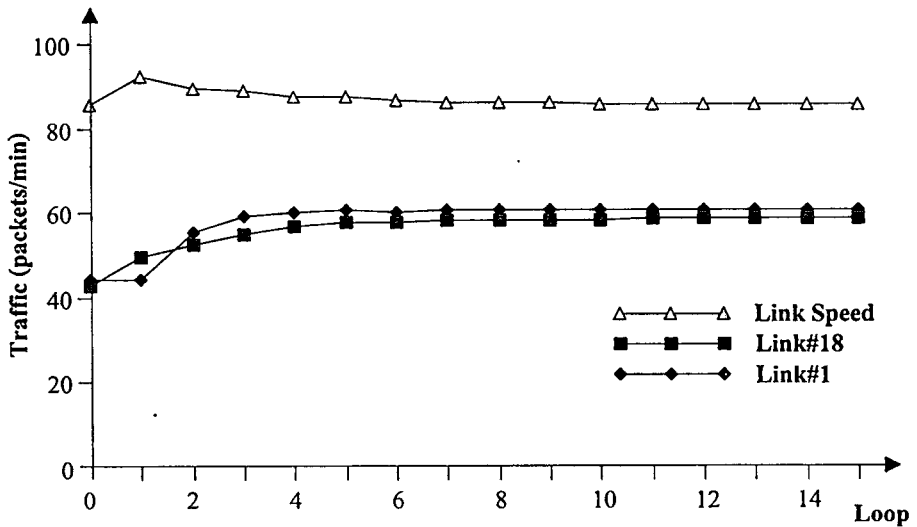
เมื่อพิจารณาค่าหน่วยเวลาที่เกิดขึ้นในแต่ละสถานี ปรากฏว่า ช่องสัญญาณเดียวกันที่ใช้การคำนวณในแบบของดาวเทียมจะมีค่าหน่วยเวลาน้อยกว่าการคำนวณในแบบของเคเบิลตามปกติที่ค่าความจุของช่องสัญญาณเดียวกัน ทั้งนี้เนื่องจากค่าหน่วยเวลาของแพคเกจก่อนที่ส่งสำเร็จของสถานีดาวเทียม มีค่าน้อยกว่าคาบเวลาดำสุดที่ถูกกำหนดของแต่ละแพคเกจที่มาถึง คาบเวลาดำสุดดังกล่าวได้จากอัตราการมาถึงของแพคเกจสูงสุดของแต่ละสถานีดาวเทียม ซึ่งต้องกำหนดให้อยู่ภายในจุดทำงานที่เหมาะสมกับประสิทธิภาพโดยรวมของระบบดาวเทียม (อธิบายโดยละเอียดในบทที่

ภาพที่ 17



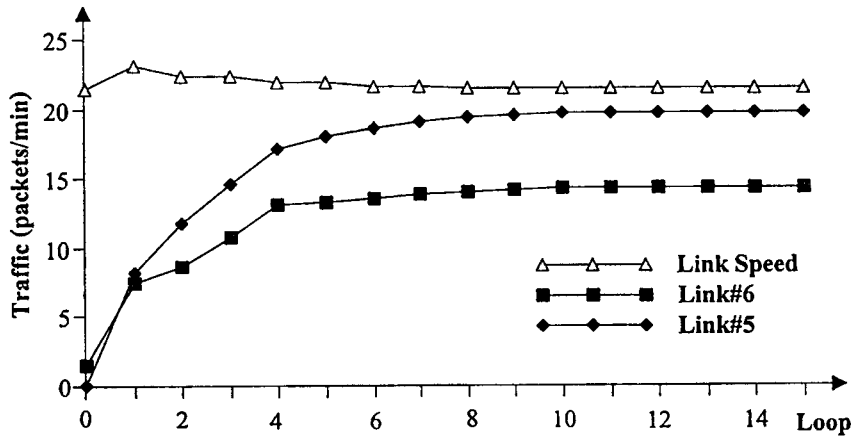
แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 8 และ 9 เทียบกับ Link Speed

ภาพที่ 18



แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 1 และ 18 เทียบกับ Link Speed

ภาพที่ 19



แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 5 และ 6 เทียบกับ Link Speed

สำหรับการพิจารณาค่าหน่วยเวลาโดยรวมของโครงข่ายทั้งสองแบบ พบว่า ค่าหน่วยเวลาของโครงข่ายแบบผสม(0.012865995 min หรือ 0.772 sec) น้อยกว่าค่าหน่วยเวลาของโครงข่ายที่ถูกเปลี่ยนให้มีแต่ตัวกลางภาคพื้นดิน(0.015231511 min) ผลที่ได้ในลักษณะนี้จะเปลี่ยน ไปในทิศทางที่เลวร้ายลงอย่างรวดเร็วเมื่ออัตราการมาถึงเพิ่มสูงขึ้นจนเกินกว่าค่าที่กำหนด ซึ่งส่งผลกระทบต่อระบบดาวเทียมทั้งหมดด้วยในเวลาเดียวกัน

ตารางที่ 3

Loop No.	Traffic		Link Speed		Avr Delay		Cost	
	Link#8	Link#9	Link#8	Link#9	Link#8	Link#9	Link#8	Link#9
0	0	0	2.857	2.857	0.35	0.35	0.35	0.35
1	0	0.308	3.027	3.027	0.33	0.368	0.33	0.4093
2	0	1.348	2.989	2.989	0.335	0.61	0.335	1.11103
3	0	1.704	2.979	2.979	0.336	0.787	0.336	1.83165
4	0	1.796	2.976	2.976	0.336	0.847	0.336	2.1375
5	0	1.821	2.976	2.976	0.336	0.866	0.336	2.2375
6	0	1.827	2.976	2.976	0.336	0.871	0.336	2.23143

ตารางที่ 3(ต่อ)

7	0	1.829	2.976	2.975	0.336	0.872	0.336	2.2539
8	0	1.83	2.975	2.975	0.336	0.873	0.336	2.2574
9	0	1.83	2.975	2.975	0.336	0.873	0.336	2.2644
10	0	1.83	2.975	2.975	0.336	0.873	0.336	2.6627
11	0	1.83	2.975	2.975	0.336	0.873	0.336	2.6682
12	0	1.83	2.975	2.975	0.336	0.873	0.336	2.6682

แสดง ผลการทำงานของโปรแกรม MG_C สำหรับตัวอย่างที่ 5.2 ในกรณีที่โครงข่ายมีเฉพาะแต่สถานีที่ใช้ตัวกลางภาคพื้นดิน

ตัวอย่างที่ 5.3

ตัวอย่างนี้ เป็นการทดลองเปรียบเทียบพฤติกรรมของอัลกอริทึมที่เป็นผลเนื่องจากการเปลี่ยนค่าพารามิเตอร์ของช่องสัญญาณดาวเทียม โดยสมมติให้โครงข่ายมีโครงสร้างเดียวกับตัวอย่างที่ 5.2 และพารามิเตอร์ของระบบดาวเทียมที่ใช้คือ

- ค่าเวลาการให้บริการ(Service time) โดยเฉลี่ย ต่อหนึ่งแพคเกจเท่ากับ 1.366 และ 2.1945 วินาที หรือ 0.022766 และ 0.0366 นาที
- ค่า Variance ของเวลาการให้บริการเท่ากับ 11.99 และ 21.74 วินาที หรือ 0.1998 และ 0.3623 นาที
- อัตราเร็วของแพคเกจที่มาถึงแต่ละสถานีดาวเทียมเท่ากับ 0.0357 และ 0.05923 แพคเกจ ต่อ วินาที หรือ 2.1428 และ 3.57 แพคเกจ ต่อ นาที

สำหรับกรณีที่โครงข่ายมีอัตราเร็วของช่องสัญญาณดาวเทียมที่ 2.1428 แพคเกจต่อนาที ปรากฏว่าเส้นทางที่สั้นที่สุดมีการเปลี่ยนแปลงจากตัวอย่างที่ 5.2 และเป็นผลให้กราฟฟิคที่กำหนดในตัวอย่างที่ 5.2 มีค่าสูงเกินไปสำหรับบางช่องสัญญาณ ช่องสัญญาณที่มีผลต่อขนาดของกราฟฟิคที่สูงเกินไปคือ Link#9(ช่องสัญญาณดาวเทียม) และ Link#2 ส่งผลให้ระบบไม่สามารถกลับสู่สถานะเดิมได้(ความจุของช่องสัญญาณไม่กลับสู่ค่าเดิม) เพื่อให้ระบบสามารถปรับค่าเข้าสู่จุดต่ำสุดได้โดยที่ความจุของช่องสัญญาณหรือ Link Speed อยู่ในสถานะปกติ สามารถทำได้สองวิธีคือ ลดขนาดของกราฟฟิคลงในเส้นทางที่มีค่าสูงเกินไป หรือเพิ่มขนาดความจุของช่องสัญญาณดังกล่าว

วิธีที่สองเป็นวิธีที่สะดวกที่สุดแต่ต้องเสียค่าใช้จ่ายเพิ่มขึ้น มีผลต่อสถานีอื่นที่อยู่ในโครงข่ายและอาจเกิดข้อจำกัดในการเปลี่ยนขนาดของช่องสัญญาณได้ รวมทั้งส่งผลกระทบต่อเส้นทางที่สั้นที่สุดที่อาจเปลี่ยนเป็นเส้นทางอื่น(อธิบายในตัวอย่างที่ 5.4) ในตัวอย่างนี้เลือกใช้วิธีแรก โดยเปลี่ยนขนาดของทราฟฟิกจาก

เส้นทางจาก Node1 ไปยัง Node4 มี Traffic เท่ากับ **51.0 packets/min**

ไปเป็น

เส้นทางจาก Node1 ไปยัง Node4 มี Traffic เท่ากับ **50.0 packets/min**

ผลจากการทดลองใช้โปรแกรมหลังจากลดขนาดทราฟฟิกลงพบว่า เส้นทางที่เกิดขึ้นจากการเลือกเส้นทางที่ดีที่สุดในแต่ละครั้งของการเปลี่ยนแปลงทราฟฟิกและ Cost ในการวนซ้ำแต่ละครั้ง มีเส้นทางที่เกิดขึ้นเหมือนกันเมื่อ run โปรแกรมด้วยทุกค่าของพารามิเตอร์ควาเทียม คือ

เส้นทางจาก Node1 ไปยัง Node4 มี เส้นทางที่ดีที่สุดที่เกิดขึ้น คือ

- 1) node 1- node 3- node 7- node 4
- 2) node 1- node 2- node 8- node 7- node 4
- 3) node 1- node 2- node 9- node 5- node 6- node 4
- 4) node 1- node 2- node 9- node 5- node 6- node 3- node 7- node 4

เส้นทางจาก Node3 ไปยัง Node10 มี เส้นทางที่ดีที่สุดที่เกิดขึ้น คือ

- 1) node 3- node 7- node 10

เส้นทางจาก Node2 ไปยัง Node6 มี เส้นทางที่ดีที่สุดที่เกิดขึ้น คือ

- 1) node 2- node 8- node 9- node 5- node 6
- 2) node 2- node 9- node 5- node 6

เส้นทางจาก Node5 ไปยัง Node8 มี เส้นทางที่ดีที่สุดที่เกิดขึ้น คือ

- 1) node 5- node 1- node 2- node 8

เส้นทางจาก Node6 ไปยัง Node4 มี เส้นทางที่ดีที่สุดที่เกิดขึ้น คือ

- 1) node 6- node 3- node 7- node 4
- 2) node 6- node 4

เส้นทางจาก Node4 ไปยัง Node3 มี เส้นทางที่ดีที่สุดที่เกิดขึ้น คือ

- 1) node 4- node 10- node 5- node 1- node 3
- 2) node 4- node 10- node 5- node 6- node 3
- 3) node 4- node 3

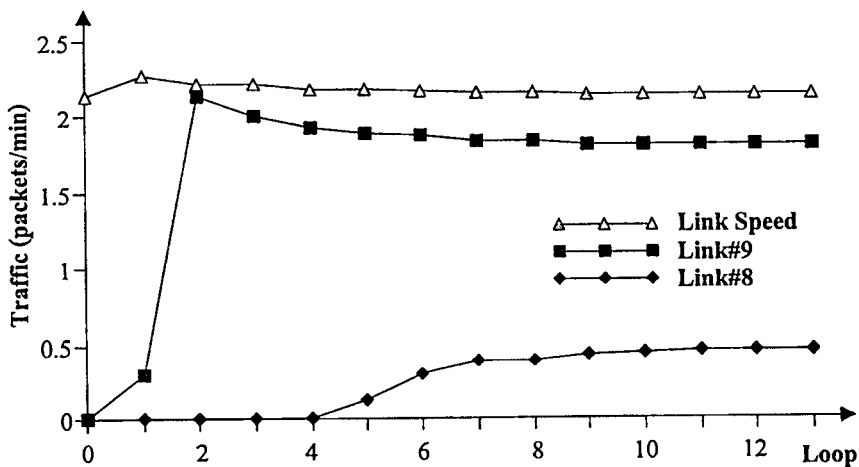
เส้นทางจาก Node5 ไปยัง Node3 มีเส้นทางที่ดีที่สุดเกิดขึ้น คือ

- 1) node 5- node 6- node 3
- 2) node 5- node 1- node 3

เมื่อพิจารณาผลที่ได้จากการคำนวณโดยใช้โปรแกรม MG_C ที่ค่าพารามิเตอร์ของช่องสัญญาณ ความเทียมนเท่ากับ : Link Speed 2.1428 แพกเกตต่อนาทิต(0.03 แพกเกตต่อเฟรม) ค่าเวลาการให้บริการ 0.022766 นาทิตและVariance 0.1998 นาทิต แสดงโดยกราฟใน ภาพที่ 18-20 ที่ช่องสัญญาณที่มี Link Speed ต่างๆกันเช่นเดียวกับตัวอย่างที่ 2 พบว่า แนวโน้มของการเปลี่ยนแปลงของกราฟฟิคสำหรับแต่ละ Link มีลักษณะเดียวกับตัวอย่างที่ 2 เนื่องจากเส้นทางที่สั้นที่สุดของแต่ละวงจรยังคงเป็นเส้นทางเดิม ทำให้การเปลี่ยนแปลงของกราฟฟิคเกิดขึ้นในช่องสัญญาณเดียวกันกับตัวอย่างที่ผ่านมาแต่ขนาดของการเปลี่ยนแปลงมีความแตกต่างเล็กน้อยเนื่องจากผลของพารามิเตอร์ที่เปลี่ยนไปของช่องสัญญาณความเทียมน

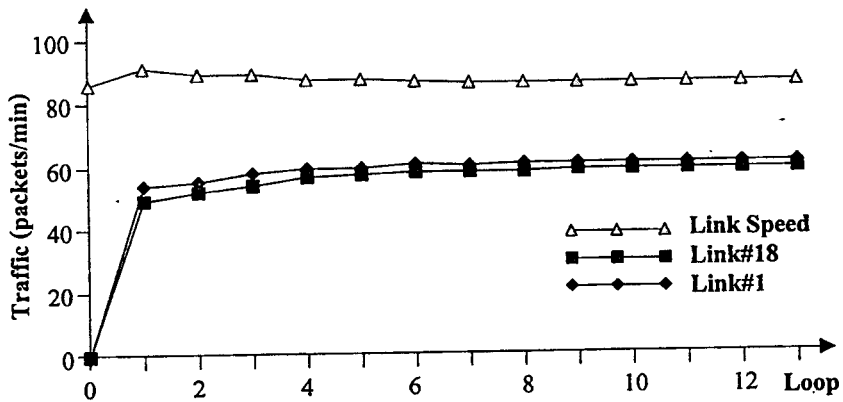
สำหรับผลลัพธ์ที่ได้จากตัวอย่างนี้ เราสามารถสรุปได้ว่าการเปลี่ยนแปลงของกราฟฟิคและค่าหน่วยเวลาเพื่อเข้าสู่จุดที่ค่าหน่วยเวลารวมต่ำสุด ยังคงสอดคล้องกับข้อกำหนดและรูปแบบเดิมที่ได้กำหนดไว้ ผลของโมเดลสำหรับช่องสัญญาณความเทียมนไม่ก่อให้เกิดผลข้างเคียงต่อช่องสัญญาณอื่นๆที่ใช้ตั้งกลางภาคพื้นดิน

ภาพที่ 20



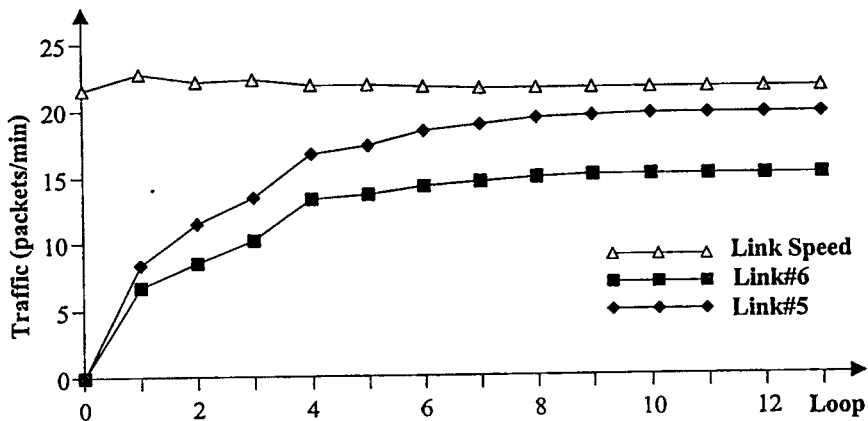
แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 8 และ 9 เทียบกับ Link Speed

ภาพที่ 21



แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 1 และ 18 เทียบกับ Link Speed

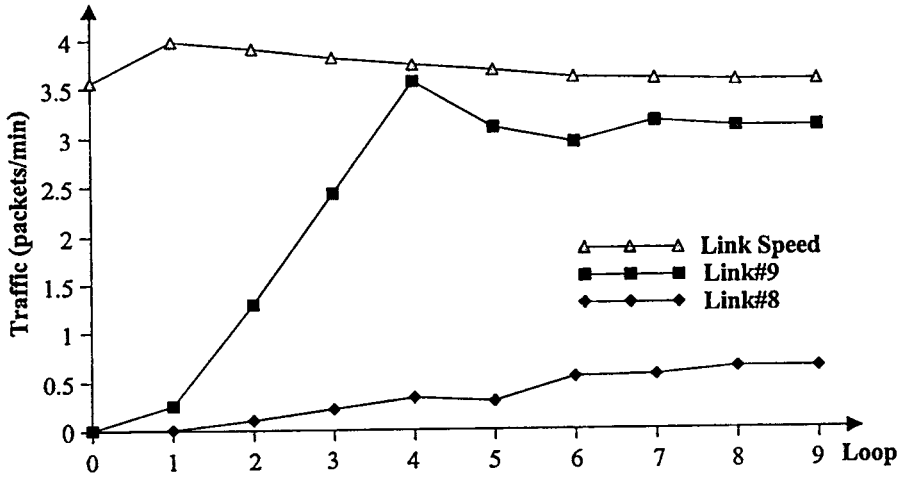
ภาพที่ 22



แสดง การเปลี่ยนแปลงของกราฟฟิคสำหรับ Link# 5 และ 6 เทียบกับ Link Speed

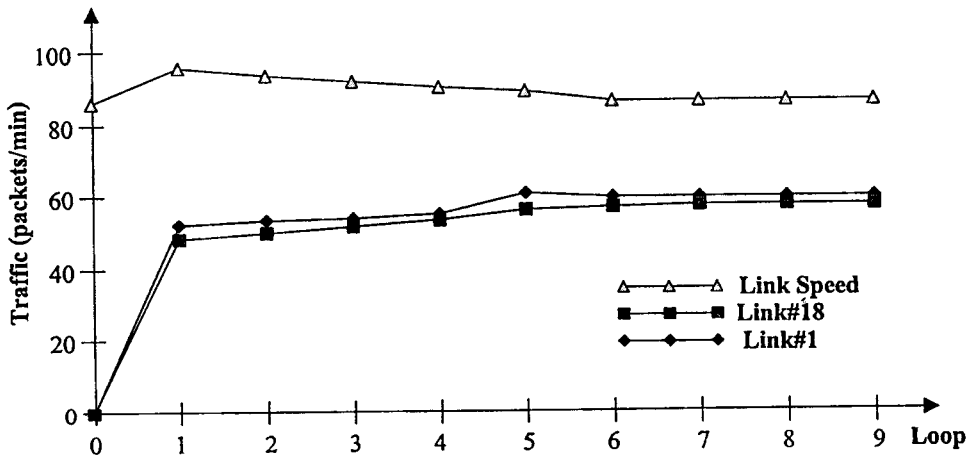
เมื่อพิจารณาโครงข่ายที่มีพารามิเตอร์ของช่องสัญญาณดาวเทียมที่ อัตราเร็วของแพคเกจ 3.57 แพคเกจ ต่อวินาที (0.05 แพคเกจต่อวินาที) เวลาการให้บริการ 0.0366 วินาที Variance 0.3623 วินาที ปรากฏว่าเมื่อใส่ค่ากราฟฟิคเท่ากับในตัวอย่างที่ 1 และ 2 ระบบสามารถปรับค่าเข้าสู่จุดต่ำสุดได้โดยที่ Link Speed อยู่ในสถานะปกติ ผลที่ได้จากโปรแกรมสามารถแสดงในรูปของกราฟในภาพที่ 21-23

ภาพที่ 23



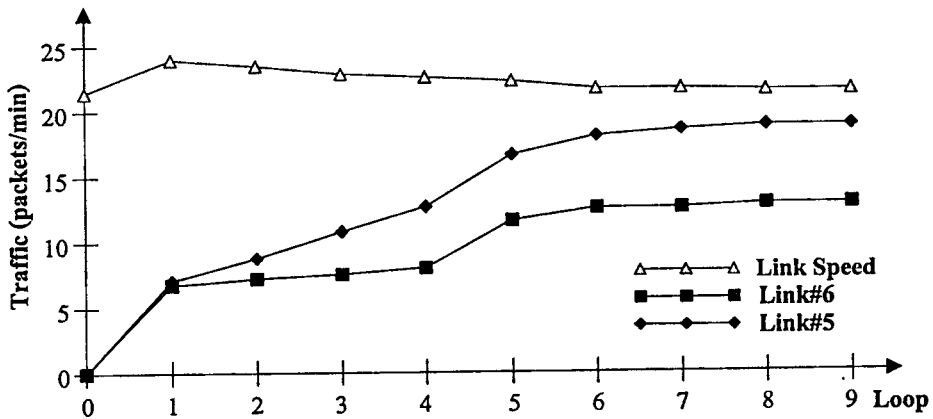
แสดง การเปลี่ยนแปลงของทราฟฟิกสำหรับ Link# 8 และ 9 เทียบกับ Link Speed

ภาพที่ 24



แสดง การเปลี่ยนแปลงของทราฟฟิกสำหรับ Link# 1 และ 18 เทียบกับ Link Speed

ภาพที่ 25



แสดง การเปลี่ยนแปลงของกราฟฟิกสำหรับ Link# 5 และ 6 เทียบกับ Link Speed

ตัวอย่างที่ 5.4

ในตัวอย่างนี้ เป็นการเปรียบเทียบพฤติกรรมของอัลกอริทึมที่เปลี่ยนจำนวนของผู้ใช้ช่องสัญญาณดาวเทียม จากเดิมเท่ากับ 75 users ในตัวอย่างที่ 5.1 และ 5.2 เป็น 100 users โดยใช้พารามิเตอร์ของระบบดาวเทียมที่จุดทำงานเดียวกันคือ ที่อัตราเร็วของแพคเกจที่มาถึงแต่ละสถานีดาวเทียมเท่ากับ 0.03, 0.04, 0.05 แพคเกจ ต่อ เฟรม ทำให้ได้ค่าใหม่ดังนี้

- ค่าเวลาการให้บริการ(Service time) โดยเฉลี่ย ต่อหนึ่งแพคเกจเท่ากับ 1.8913, 2.59 และ 3.4437 วินาที หรือ 0.03152, 0.0432 และ 0.0574 นาที
- ค่า Variance ของเวลาการให้บริการเท่ากับ 19.74, 30 และ 42.778 วินาที หรือ 0.329, 0.5 และ 0.713 นาที
- อัตราเร็วของแพคเกจที่มาถึงแต่ละสถานีดาวเทียมเท่ากับ 0.0357, 0.04762 และ 0.05923 แพคเกจ ต่อ วินาที หรือ 2.1428, 2.857 และ 3.57 แพคเกจ ต่อ นาที

ผลจากการ run โปรแกรม พบว่า เมื่อใช้ปริมาณกราฟฟิกในตัวอย่างที่ 5.1 และ 5.2 โครจข่ายที่ใช้พารามิเตอร์ของดาวเทียมในกลุ่มของอัตราเร็ว 2.1428 แพคเกจต่อนาที ที่ค่าเวลาการให้บริการเท่ากับ 0.03152 นาทีและเวเลี่ยนท์เท่ากับ 0.329 นาทีรูปแบบและแนวโน้มของการเปลี่ยนแปลงยังคงเป็นไปในลักษณะเดิมรวมทั้งเส้นทางที่สั้นที่สุดยังคงเป็นเส้นทางเดียวกับเส้นทางในตัวอย่างที่ 5.3 แต่เมื่อทดลองเปลี่ยนเป็นพารามิเตอร์ของดาวเทียมในกลุ่มของอัตราเร็ว 2.857 แพคเกจ ต่อ นาทีหลังจากที่ทดลองโปรแกรม MG_C โดยใช้ปริมาณกราฟฟิกเท่ากับในตัวอย่างที่ 5.2

โปรแกรมเกิดการวนซ้ำจำนวน 49 ครั้ง แต่ระบบไม่สามารถกลับคืนสู่สภาวะปรกติได้ ทั้งนี้พบว่ากราฟฟิคใน Link#2 มีขนาดมากเกินไป ซึ่งตามปรกติช่องสัญญาณนี้มี Link Speed เท่ากับ 85.714 แพกเกต ต่อนาที

ผลของการวนซ้ำครั้งสุดท้าย

โครงข่ายที่ใช้พารามิเตอร์ในกลุ่มของอัตราเร็ว 2.1428 แพกเกตต่อนาที

Link#2 Ch[2][8] Traffic = 84.05 Link Speed = 85.714

Link#8 Ch[4][3] Traffic = 0.389 Link Speed = 2.143

Link#9 Ch[6][4] Traffic = 1.891 Link Speed = 2.143

โครงข่ายที่ใช้พารามิเตอร์ในกลุ่มของอัตราเร็ว 2.857 แพกเกตต่อนาที

Link#2 Ch[2][8] Traffic = 85.077 Link Speed = 86.778

Link#8 Ch[4][3] Traffic = 0.308 Link Speed = 2.892

Link#9 Ch[6][4] Traffic = 2.226 Link Speed = 2.892

ปริมาณกราฟฟิคที่เพิ่มขึ้นเมื่อเปรียบเทียบกับปริมาณที่เกิดในโครงข่ายที่ใช้พารามิเตอร์ของดาวเทียมในกลุ่มของอัตราเร็ว 2.1428 แพกเกตต่อนาที มีลักษณะที่ไม่เป็นอัตราส่วนต่อกันและไม่สามารถเปรียบเทียบกันได้เนื่องจากสองสาเหตุคือ

1. ค่าหน่วยเวลาของช่องสัญญาณดาวเทียมเป็นอิสระต่อความจุของช่องสัญญาณ(Link Speed) และ Cost ทำให้ค่าหน่วยเวลาของช่องสัญญาณดาวเทียมที่มีอยู่ 2 ช่อง มีผลอย่างยิ่งต่อการกระจายกราฟฟิคจากช่องสัญญาณที่มีกราฟฟิคสูงไปยังช่องสัญญาณอื่นๆที่มีกราฟฟิคต่ำกว่าในเส้นทางที่สั้นที่สุดที่ช่องสัญญาณดาวเทียมนั้นเกี่ยวข้องกับอยู่

โครงข่ายที่ใช้พารามิเตอร์ในกลุ่มของอัตราเร็ว 2.1428 แพกเกตต่อนาที

1st Setting traffic in each link and Adjusting trunk speed

Link#8 Traffic = 0.000 Link Speed = 2.369 Delay=0.032 Cost[4][3]=0.42205

Link#9 Traffic = 0.256 Link Speed = 2.369 Delay=0.075 Cost[6][4]=0.53025

โครงข่ายที่ใช้พารามิเตอร์ในกลุ่มของอัตราเร็ว 2.857 แพกเกตต่อนาที

1st Setting traffic in each link and Adjusting trunk speed

Link#8 Traffic = 0.000 Link Speed = 3.256 Delay=0.043 Cost[4][3]=0.30714

Link#9 Traffic = 0.217 Link Speed = 3.256 Delay=0.100 Cost[6][4]=0.35268

2. เส้นทางที่สั้นที่สุดเปลี่ยนแปลงไปทำให้ปริมาณกราฟฟิคเปลี่ยนแปลงด้วย การเลือกเส้นทางที่สั้นที่สุดจะพิจารณาจาก Cost ของแต่ละเส้นทาง

โครงข่ายที่ใช้พารามิเตอร์ในกลุ่มของอัตราเร็ว 2.1428 แพคเกจต่อนาที

START LOOP(3)

Shortest Path[1] :-1-2-8-7-4	Cost of Shortest past No.1 = 3.784
Shortest Path[2] :-3-7-10	Cost of Shortest past No.2 = 10.842
Shortest Path[3] :-2-9-5-6	Cost of Shortest past No.3 = 0.218
Shortest Path[4] :-5-1-2-8	Cost of Shortest past No.4 = 3.756
Shortest Path[5] :-6-4	Cost of Shortest past No.5 = 4.317
Shortest Path[6] :-4-10-5-6-3	Cost of Shortest past No.6 = 0.289
Shortest Path[7] :-5-6-3	Cost of Shortest past No.7 = 0.148

โครงข่ายที่ใช้พารามิเตอร์ในกลุ่มของอัตราเร็ว 2.857 แพคเกจต่อนาที

START LOOP(3)

Shortest Path[1] :-1-2-9-5-6-4	Cost of Shortest past No.1 = 0.825
Shortest Path[2] :-3-7-10	Cost of Shortest past No.2 = 3.776
Shortest Path[3] :-2-9-5-6	Cost of Shortest past No.3 = 0.164
Shortest Path[4] :-5-1-2-8	Cost of Shortest past No.4 = 1.454
Shortest Path[5] :-6-4	Cost of Shortest past No.5 = 0.612
Shortest Path[6] :-4-10-5-6-3	Cost of Shortest past No.6 = 0.252
Shortest Path[7] :-5-6-3	Cost of Shortest past No.7 = 0.120

เพื่อให้ระบบกลับเข้าสู่สภาวะปกติ ทำการปรับทราฟฟิก จาก

เส้นทางจาก Node1 ไปยัง Node4 มี Traffic เท่ากับ 51.0 packets/min

เส้นทางจาก Node5 ไปยัง Node8 มี Traffic เท่ากับ 44.0 packets/min

ไปเป็น

เส้นทางจาก Node1 ไปยัง Node4 มี Traffic เท่ากับ 52.0 packets/min

เส้นทางจาก Node5 ไปยัง Node8 มี Traffic เท่ากับ 43.0 packets/min

เมื่อทดลองใช้โปรแกรมทำการคำนวณ ปรากฏว่า หลังจากวนซ้ำเป็นจำนวน 87 ครั้ง ได้ผลลัพธ์สุดท้ายดังนี้

87th Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 59.510	Link Speed = 85.714	Delay=0.038	Cost[1][2]=0.12483
Link#2 Traffic = 83.481	Link Speed = 85.714	Delay=0.448	Cost[2][8]=17.18865
Link#3 Traffic = 55.315	Link Speed = 64.286	Delay=0.111	Cost[1][3]=0.79886

Link#4 Traffic = 83.421 Link Speed = 85.714 Delay=0.436 Cost[3][7]=16.30279
 Link#5 Traffic = 19.029 Link Speed = 21.429 Delay=0.417 Cost[2][9]=3.72071
 Link#6 Traffic = 13.478 Link Speed = 21.429 Delay=0.126 Cost[6][3]=0.33900
 Link#7 Traffic = 51.069 Link Speed = 64.286 Delay=0.076 Cost[7][4]=0.36802
 Link#8 Traffic = 0.628 Link Speed = 2.857 Delay=0.207 Cost[4][3]=0.57483
 Link#9 Traffic = 2.431 Link Speed = 2.857 Delay=0.680 Cost[6][4]=15.71666
 Link#10 Traffic = 4.372 Link Speed = 21.429 Delay=0.059 Cost[4][10]=0.07366
 Link#11 Traffic = 47.000 Link Speed = 85.714 Delay=0.026 Cost[7][10]=0.05719
 Link#12 Traffic = 14.648 Link Speed = 64.286 Delay=0.020 Cost[8][7]=0.02609
 Link#13 Traffic = 4.372 Link Speed = 21.429 Delay=0.059 Cost[10][5]=0.07366
 Link#14 Traffic = 62.825 Link Speed = 85.714 Delay=0.044 Cost[5][1]=0.16361
 Link#15 Traffic = 25.833 Link Speed = 64.286 Delay=0.026 Cost[8][9]=0.04348
 Link#16 Traffic = 0.000 Link Speed = 85.714 Delay=0.012 Cost[8][1]=0.01167
 Link#17 Traffic = 44.861 Link Speed = 85.714 Delay=0.024 Cost[9][5]=0.05136
 Link#18 Traffic = 57.409 Link Speed = 85.714 Delay=0.035 Cost[5][6]=0.10698
 โดยที่ค่าหน่วยเวลารวมโดยเฉลี่ยของโครงข่ายเท่ากับ 0.012953533 min

สำหรับโครงข่ายที่มีพารามิเตอร์ของช่องสัญญาณดาวเทียมที่ อัตราเร็วของแพคเกจ 3.57
 แพคเกจ ต่อนาที่(0.05 แพคเกจต่อวินาที) เวลาการให้บริการ 0.0574 นาที่ Variance 0.714 นาที่
 ปรากฏว่าเมื่อใส่ค่าทราฟฟิกในตัวอย่างที่ 5.2 โปรแกรมเกิดการวนซ้ำจำนวน 20 ครั้ง และไม่สา
 มารถกลับสู่สถานะปรกติได้เช่นเดียวกัน เนื่องจากเหตุผลดังที่ได้กล่าวมา เพื่อให้ระบบกลับเข้าสู่
 สถานะปรกติ ทำการปรับทราฟฟิก จาก

เส้นทางจาก Node1 ไปยัง Node4 มี Traffic เท่ากับ 51.0 packets/min

เส้นทางจาก Node5 ไปยัง Node8 มี Traffic เท่ากับ 44.0 packets/min

ไปเป็น

เส้นทางจาก Node1 ไปยัง Node4 มี Traffic เท่ากับ 50.0 packets/min

เส้นทางจาก Node5 ไปยัง Node8 มี Traffic เท่ากับ 43.0 packets/min

เมื่อทดลองใช้โปรแกรมทำการคำนวณ ปรากฏว่า หลังจากวนซ้ำเป็นจำนวน 13 ครั้งได้ผลลัพธ์สุด
 ท้ายดังนี้

13th Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 58.435 Link Speed = 85.714 Delay=0.037 Cost[1][2]=0.11519

Link#2 Traffic = 83.561	Link Speed = 85.714	Delay=0.465	Cost[2][8]=18.49596
Link#3 Traffic = 55.844	Link Speed = 64.286	Delay=0.118	Cost[1][3]=0.90203
Link#4 Traffic = 82.750	Link Speed = 85.714	Delay=0.337	Cost[3][7]=9.75466
Link#5 Traffic = 17.874	Link Speed = 21.429	Delay=0.281	Cost[2][9]=1.69577
Link#6 Traffic = 12.309	Link Speed = 21.429	Delay=0.110	Cost[6][3]=0.25766
Link#7 Traffic = 48.584	Link Speed = 64.286	Delay=0.064	Cost[7][4]=0.26073
Link#8 Traffic = 0.597	Link Speed = 3.570	Delay=0.285	Cost[4][3]=0.40380
Link#9 Traffic = 2.916	Link Speed = 3.570	Delay=1.173	Cost[6][4]=8.35329
Link#10 Traffic = 4.403	Link Speed = 21.429	Delay=0.059	Cost[4][10]=0.07393
Link#11 Traffic = 47.000	Link Speed = 85.714	Delay=0.026	Cost[7][10]=0.05719
Link#12 Traffic = 12.834	Link Speed = 64.286	Delay=0.019	Cost[8][7]=0.02428
Link#13 Traffic = 4.403	Link Speed = 21.429	Delay=0.059	Cost[10][5]=0.07393
Link#14 Traffic = 64.279	Link Speed = 85.714	Delay=0.047	Cost[5][1]=0.18656
Link#15 Traffic = 27.727	Link Speed = 64.286	Delay=0.027	Cost[8][9]=0.04810
Link#16 Traffic = 0.000	Link Speed = 85.714	Delay=0.012	Cost[8][1]=0.01167
Link#17 Traffic = 45.601	Link Speed = 85.714	Delay=0.025	Cost[9][5]=0.05327
Link#18 Traffic = 56.725	Link Speed = 85.714	Delay=0.034	Cost[5][6]=0.10200

โดยที่ค่าหน่วยเวลารวมโดยเฉลี่ยของโครงข่ายเท่ากับ 0.014610336 min

บทที่ 6

บทสรุป

บทนำ

วิทยานิพนธ์นี้ นำเสนอแนวทางในการวิเคราะห์และออกแบบ โครงข่ายที่มีการใช้ตัวกลางที่เป็นเคเบิลภาคพื้นดินร่วมกับตัวกลางที่เป็นช่องสัญญาณดาวเทียมที่มีโครงสร้างที่ใกล้เคียงกับระบบที่มีอยู่ในประเทศไทย

วัตถุประสงค์ของงานวิจัยนี้คือ สามารถใช้เป็นแนวทางสำหรับผู้บริหารโครงข่ายในการเลือกใช้ตัวกลาง ความจุของตัวกลางและจำนวนสถานีที่ใช้ตัวกลางแต่ละแบบที่เหมาะสม ในกรณีของสถานีที่ใช้ตัวกลางดาวเทียม(VSAT)ที่ผู้บริหารโครงข่ายพิจารณาโครงสร้างเบื้องต้นของระบบดาวเทียม(บริหารโดย วิเซท โอเปอเรเตอร์)และพบว่าเหมาะสมกับระบบของตน ผู้บริหารโครงข่ายควรจะต้องสามารถวิเคราะห์ต่อไปได้ว่า จุดทำงานของช่องสัญญาณดาวเทียมที่สถานีดาวเทียมเชื่อมต่ออยู่ ส่งผลต่อระบบของคนในลักษณะใดบ้าง จำนวนของสถานีดาวเทียมที่ใช้เป็นโนด (Access Point)ในโครงข่าย รวมทั้งขนาดของกราฟฟิคที่เดินทางในแต่ละเส้นทาง

ในบทที่ 3 ได้เสนอแนวทางในการสร้างโมเดลเพื่อใช้ในการวิเคราะห์โครงสร้างของระบบดาวเทียม โดยใช้ตัวแปรหลัก(Performance Parameter) 2 ตัวแปรคือกำหนดวงเวลาและทรูพุท จากผลการทดลองในรูปของกราฟที่ได้แสดงไว้ สามารถใช้เป็นตัวอย่างในการวิเคราะห์และเลือกจุดทำงาน ซึ่งจุดทำงานแต่ละจุดของระบบดาวเทียมจะส่งผลต่อระบบโครงข่ายโดยรวมมากขึ้น เมื่อจำนวนสถานีดาวเทียมในโครงข่ายผสมเพิ่มจำนวนขึ้น

จากการศึกษาความสัมพันธ์ระหว่างสถานะของช่องสัญญาณดาวเทียมที่สถานีดาวเทียมในโครงข่ายผสม(กำลังพิจารณา)เชื่อมต่ออยู่ และ สมรรถนะของโครงข่ายผสมทั้งหมดในบทที่ 5 พบว่า พารามิเตอร์ที่เป็นปัจจัยสำคัญต่อความสัมพันธ์ดังกล่าว คือ อัตราการมาถึงของข้อมูลที่สถานีดาวเทียม ค่าเวลาการให้บริการ สถานะของช่องสัญญาณดาวเทียมและขนาดของกราฟฟิคในแต่ละเส้นทาง

ในบทที่ 4 ได้แสดงแนวทางในการสร้างโมเดลสำหรับสถานีดาวเทียมที่อยู่ในโครงข่ายผสม เพื่อนำไปใช้ในการวิเคราะห์โครงข่ายแบบผสมในบทที่ 5 ซึ่งผลที่ได้แสดงตามขั้นตอนการทำงานของโปรแกรมในภาคผนวก ข. และ ค. รวมทั้งกราฟแสดงแนวโน้มการเปลี่ยนแปลงของกราฟฟิคในสถานีดาวเทียมที่กำลังพิจารณา และเปรียบเทียบกับสถานีที่ใช้ตัวกลางภาคพื้นดิน

จากผลการทดลอง สามารถสรุปได้ว่า ในการออกแบบแต่ละครั้ง ผู้บริหารโครงข่ายผสมควรจะต้องมีข้อมูลเบื้องต้นสำหรับสถานีที่ใช้ตัวกลางดาวเทียมเช่น ค่าเวลาตอบรับของข้อมูลโดยเฉลี่ย อัตราการมาถึงของข้อมูลที่เหมาะสม เป็นต้น ข้อมูลดังกล่าวจำเป็นสำหรับการประเมินรูปแบบและปริมาณของข้อมูลต่อสถานีเพื่อกำหนดโครงสร้างของโครงข่ายและจำนวนของสถานีดาวเทียมที่สามารถใช้เป็นโหนดในโครงข่าย รวมทั้งผลกระทบที่อาจจะเกิดจากการเชื่อมต่อ ความเหมาะสมในการติดตั้งและการกำหนดขนาดความจุของตัวกลางและค่าใช้จ่าย

การนำไปประยุกต์ใช้

ในปัจจุบัน ระบบดาวเทียมโดยส่วนใหญ่ที่มีโครงสร้างพื้นฐานในลักษณะที่มีการเข้าใช้ช่องสัญญาณแบบสุ่ม(Random Aloha)โดยผู้ใช้งานหลายคน มีการพัฒนาและใช้งานอย่างกว้างขวางรวมทั้งระบบดาวเทียมที่วีแซทโอเปอเรเตอร์ในประเทศไทยใช้งานอยู่ในปัจจุบัน ดังนั้นโมเดลในงานวิจัยนี้ได้สร้างขึ้นโดยมีโครงสร้างการทำงานตามรูปแบบของการเข้าใช้ช่องสัญญาณแบบสุ่ม ทำให้สามารถนำไปประยุกต์ใช้กับระบบดาวเทียมที่มีโครงสร้างพื้นฐานเดียวกันในหลายๆลักษณะ และในกรณีของการวิเคราะห์เพื่อลงทุนสำหรับธุรกิจการให้บริการวีแซท วิศวกรผู้ออกแบบสามารถใช้เป็นเครื่องมือในการประเมินค่าใช้จ่ายเบื้องต้นสำหรับขนาดอัตราการมาถึงของข้อมูลเพื่อเลือกรูปแบบการให้บริการที่เหมาะสมสำหรับผู้ใช้งานแต่ละราย

แนวทางการศึกษาวิจัยต่อไป

โครงข่ายแบบแพคเกจ สวิตชิง ในปัจจุบันมีอายุการใช้งานค่อนข้างยาวนาน และมีการพัฒนารูปแบบในการคำนวณคุณสมบัติต่างๆของระบบในหลายรูปแบบ ปัจจุบันตัวกลางภาคพื้นดินสามารถให้บริการด้วยอัตราเร็วที่สูงขึ้นมากเมื่อเทียบกับสายโทรศัพท์แบบเดิม ตัวกลางดังกล่าว เช่น สายไฟเบอร์ ออปติก(Fibre Optic Cable) สำหรับระยะทางไกล และสายโคแอกเซียล (Coaxial Cable)สำหรับระยะทางใกล้ โพลโตคอลระดับแคเรียเจอร์ที่ใช้เช่น เฟรมรีเลย์(Frame Relay) เอทีเอ็ม(ATM) และ โพรโตคอลระดับโครงข่ายที่ใช้คือ TCP/IP เป็นต้น แต่มีการใช้งานร่วมกับช่องสัญญาณ

ความถี่ที่น้อยมากเนื่องจากขนาดของแต่ละเซลล์(cell) หรือ เฟรม ที่ใช้มีขนาดเล็กเมื่อเทียบกับค่า
หน่วยเวลาที่เกิดบนช่องสัญญาณความถี่ และขนาดของตัวกลางที่ใหญ่ขึ้น ถ้าเราสามารถชดเชย
หรือลดขนาดของปัญหาดังกล่าวรวมทั้งออกแบบอัลกอริทึมของโปรโตคอล และสร้างการจำลอง
ด้วยโมเดลที่เหมาะสม การศึกษาและวิเคราะห์โครงสร้างแบบผสมสำหรับตัวกลางความเร็วสูงจะทำให้
ได้สะดวกขึ้นและมีความหลากหลายมากขึ้นกว่าในปัจจุบัน

บรรณานุกรม

1. Aaron Kershenbaum Telecommunication Network Design Algorithm Mc graw Hill, 1993
2. A. Fujii, Y. Teshigawara, S. Tejima, and Y. Matsumoto AA/TDMA Adaptive Satellite Access Method for Mini-Earth Station Network in Conf. Rec. Proc. GLOBECOM'86, pp. 1464-1499, Texas, December 1986.
3. Chu. W. W. and Alan G. Konheim On The Analysis And Modeling of A Class of Computer Communication System IEEE Trans. On Comm. NO.3, June 1972
4. D. Raychaudhuri Dynamic Performance of Aloha type VSAT channel: A Simulation Study IEEE Trans. Commun., vol.COM-38, NO. 2, February 1990.
5. Dimitri Bertsekas and Robert Gallager Data Network: Second Edition Prentice Hall, 1992.
6. Geoferey Grimmett and Dominic Welsh Probability : An Introduction Oxford Science Publications.
7. Hisashi Kobayashi Modeling and Analysis, An Introduction to System Performance Evaluation Methodology Addison Wesley Publishing Company, 1978.
8. Jack R. Benjamin and C. Allin Cornell Probability, Statistic, and Decision for Civil Engineer McGraw-Hill, 1981
9. Leonard Klienrock Queuing Systems Volume I: Theory John Wiley and Sons, 1975
10. Leonard Kleinrock and Simon S. Lam Packet-Switching in a Slotted Satellite Channel NCC'73 p703-710, 1973.
11. Leonard Klienrock Queuing Systems Volume I: Theory John Wiley and Sons, 1975
12. X. Wang, J. Kaniyil, Y. Onozato, J. Liu, S. Shimamoto and S.Noguchi Performance Analysis of a Combined Random Reservation Access Scheme IEEE Trans. Commun., vol.COM-39, NO. 4, April 1991.
13. Tri T. Ha Digital Satellite Communication McGraw-Hill, 1990.
14. ผศ. ดร. สุวิพล สัทธีชีวะภาค กระบวนการโพลคอนโทรลในระบบการสื่อสารแบบวีเสทของ ETS-5 การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 18, 1995.

บรรณานุกรม(ต่อ)

14. นรฤทธิ์ สุนทรสารทูล “การกำหนดเส้นทางที่เหมาะสมในการส่งข้อมูลตามสภาพแวดล้อมของเครือข่าย” วิทยานิพนธ์ วิศวกรรมศาสตรมหาบัณฑิต สถาบันเทคโนโลยีฯ ลาดกระบัง พ.ศ. 2539
15. สุวิพล สิริชีวะภาค และ เสรี อัสวารักษ์ “การวิเคราะห์ระบบดาวเทียมเพื่อการออกแบบโครงข่ายวีแสท” การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 19 กลุ่มที่ 2 หน้า CM56-CM61 2539

ภาคผนวก

ภาคผนวก ก.

ซอฟต์แวร์ระบบ

FILE “ SATSIM.C ”

```

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <alloc.h>
#include <time.h>

#define asub    12.0      /** จำนวนสล็อตที่ประกอบเป็น Aloha Subframe **/
#define tslot   0.035    /** เวลาเป็นวินาทีของแต่ละสล็อต ซึ่งรวมทั้งช่วงที่เชื่อมต่อ
                           ระหว่างสล็อต(Gab) **/
#define user    50       /** จำนวนของสถานี **/
#define pdelay  0.25     /** ค่าหน่วงเวลาที่เกิดจากการเดินทางของคลื่นในหน่วยของ
                           สล็อต **/
#define lamdar  0.6      /** อัตราของการส่งซ้ำ(Retransmission rate) **/
#define proc    0.01     /** เวลาของโปรเซสเซอร์ที่สถานีแม่ข่ายและสถานีย่อยที่ใช้
                           สำหรับการทำงานแต่ละแพคเกจ **/
#define frm     24.0     /** จำนวนของสล็อตในแต่ละเฟรม **/
#define lpr     0.5437   /** อัตราส่วนของแพคเกจแรกๆของข้อมูลที่มีขนาดใหญ่
                           กับแพคเกจของข้อมูลขนาดเล็ก **/
#define arr     3.0      /** Packet/Queue/Frame **/
#define sl      20000    /** จำนวนของสล็อตที่ทำการจองทั้งหมด **/
static float mg1q[40];

```

```

static float mg1q[40];

struct node far *array[user];
struct node
{
    float    gent;
    long    int slottime;
    int     pstate;
    long    int nowsl;
    float   tlast;
    int     coll;
    int     cutlpr;
};

float arrtime(int stat,float lamd);
int poisson(void);
float randm();
float rand1();
float rand2();
float percent(int slott);
long int aloha(long int data);
long int backlg(long int dat,float lam1);
float arrt(int stat,float lamd);
long int mkslot(float gentime);
int mg1(float qrate);

main(void)
{
    FILE *rsv1;
    int   i,j,thuput,slot,*pnt,s,q;
    int   *tmp,tmp1;

```

```

int    cnt0,cnt1,cnt2,cnt4,cnt3;
int    lrct=0,lrss=0,queue=0;
long  int slc;
float  a,tcnt,tv,tvar,chk1,chk2,lam=0.0;
float  sttn10[40];          /** สถานที่10 สำหรับการพิจารณา **/
printf("Input rate :\n");
scanf("%f", &lam);  fflush(stdin);
pnt=NULL;  tmp=NULL;
thuput=0;
cnt1=0;
cnt2=0;          /** แพคเกจที่ส่งสำเร็จ **/
cnt3=0;
cnt4=0;          /** เก็บค่าจำนวนครั้งที่เกิดการชน **/
tcnt=0.0;       /** เวลาโดยรวมของแพคเกจทั้งหมดที่ส่งสำเร็จ **/
tv=0.0;
tvar=0.0;       /** Variance **/
/* แสดงผลในไฟล์ชื่อ RSV1.SER */
if ((rsv1 = fopen("d:rsv1.ser", "wt"))== NULL)
{
    fprintf(stderr, "Cannot open input file.\n");
    exit(1);
}
/** การจำลองการทำงานของระบบคิวในระบบในแต่ละสล็อตที่เกิดขึ้นต่อเนื่องกัน ต้องการตัวแปร
แบบ Array ตามจำนวนของผู้ใช้(User)ที่มีอยู่ในระบบ สำหรับบันทึกเหตุการณ์(Event)ที่เกิดขึ้นใน
ขณะนั้น **/
for (i=0;i<user;i++)
    array[i] = farcalloc(1, sizeof(struct node));
for (i=0;i<40;i++)
    sttn10[i]=0.0;
for (i=0;i<user;i++)
{

```

```

array[i]->gent=0.0;          /** เวลาที่เกิดแพคเกจขึ้นใหม่ **/
array[i]->slottime=0;      /** ช่วงเวลาที่ถูกจัดให้อยู่ในช่วงของสล็อต **/
array[i]->pstate=0;       /** สถานะในสถานะเบื่คล็อก = 1 **/
array[i]->tlast=0.0;      /** เวลาที่แพคเกจออกจากระบบ **/
array[i]->nowsl=0;        /** เวลา(สล็อต) ในขณะนั้น **/
array[i]->coll=0;        /** จำนวนครั้งของการชนของแพคเกจ i. **/
array[i]->cutlpr=0;

/** จำลองเหตุการณ์ที่เกิดแพคเกจในแต่ละสถานี โดยเก็บค่าเวลาจำลองที่เกิดแพคเกจขึ้น **/
a=arrtime(array[i]->pstate,lam);
if (a==0.0)
{
    tmp=random(3);
    array[i]->nowsl=array[i]->nowsl+tmp;
    a=arrtime(array[i]->pstate,lam);
}
array[i]->gent=array[i]->nowsl+a;
slc=mkslot(array[i]->gent);
/** จัดให้การส่งแพคเกจใหม่ของแต่ละสถานีอยู่ในเฟรมย่อยอะโลฮาเท่านั้น **/
array[i]->slottime=aloha(slc);
}

tmp1=0;
/** เริ่มต้นรูปของการจำลอง(สล็อตที่ 1)**/
randomize();
s=0;
fprintf(rsv1,"000.000 ");
for (slot=1;slot<=sl;slot++)          /** Slot "0",1,2,3,... **/
{
    tmp=pnt;
    cnt0=0;                            /** นับจำนวนสถานีที่เกิดขึ้นในแต่ละ slot **/
    for (i=0;i<=user-1;i++)
    {

```

```

if (array[i]->slottime==slot)          /** สล็อตที่เกิดในสถานีที่ i **/
{
    *pnt=i; pnt++;                      /** นับและกำหนดสถานีที่เกิดแพคเกจในสล็อต i. **/
    cnt0++;
}
}
*pnt=user+1;
pnt=tmp;
tmp=NULL;
if (cnt0==0) goto stop_slot;
else if (cnt0==1)                      /** มีแพคเกจที่ส่งขึ้นดาวเทียมเพียงแพคเกจเดียวในสล็อต i. **/
{
    while (*pnt!=user+1)
    {
        if (array[*pnt]->pstate==0)    /* cnt0=1 และ state=0 */
        {
            thuput++;
            cnt2=cnt2+cnt0;
            array[*pnt]->tlast=tslot+(array[*pnt]->slottime*tslot)+(2*pdelay)+(2*proc);
            if (randm()>lpr)
            {
                lrcnt++;                /** แพคเกจแรกของข้อมูลที่มีขนาดใหญ่ **/
                queue=queue+poisson(); /** ขนาดโดยสุ่ม **/
                if (queue<=(frm-asub))  /** แพคเกจในคิวมีจำนวนน้อยกว่า Aloha subframe **/
                {
                    array[*pnt]->cutlpr=0;
                }else if (queue>(frm-asub)) /** แพคเกจในคิวมีจำนวนมากกว่า Aloha subframe **/
                {
                    array[*pnt]->pstate=0;
                    array[*pnt]->cutlpr=1;
                    array[*pnt]->nowsl=mkslot(array[*pnt]->tlast);
                }
            }
        }
    }
}

```

```

a=arrtime(array[*pnt]->pstate,lam);          /** แพคเกจที่ถูกส่งใหม่ **/
if (a==0.0)
{
    array[*pnt]->nowsl=array[*pnt]->nowsl+1;
    a=arrtime(array[*pnt]->pstate,lam);
}
slc=array[*pnt]->nowsl+mkslot(a);
array[*pnt]->slottime=aloha(slc);
array[*pnt]->tlast=0.0;
lrlss++;
goto exitlpr;
}
}

array[*pnt]->pstate=0;
tcnt=tcnt+array[*pnt]->tlast-array[*pnt]->gent;          /** เวลาทั้งหมด **/
if (*pnt==40)
{
    s++;
    sttn10[s]=array[*pnt]->tlast-array[*pnt]->gent;
    fprintf(rsv1,"\n%.4f ",sttn10[s]);
}

/** คำนวณค่า Variance **/

tv=tv+((array[*pnt]->tlast-array[*pnt]->gent)*(array[*pnt]->tlast-array[*pnt]->gent));
tmp1++;
if (tmp1==30) tmp1=0;
    cnt4=cnt4+array[*pnt]->coll;
cnt3++;
array[*pnt]->pstate=0;
array[*pnt]->coll=0;
array[*pnt]->nowsl=mkslot(array[*pnt]->tlast);
a=arrtime(array[*pnt]->pstate,lam);

```

```

if (a==0.0)
{
    array[*pnt]->nowsl=array[*pnt]->nowsl+random(3);
    a=arrtime(array[*pnt]->pstate,lam);
}
array[*pnt]->gent=(array[*pnt]->nowsl*tslot)+a;
slc=mkslot(array[*pnt]->gent);
array[*pnt]->slottime=aloha(slc);
array[*pnt]->tlast=0.0;
exitlpr;;
} else
if (array[*pnt]->pstate==1)          /** cnt0=1 และ state=1 **/
{
    cnt2=cnt2+cnt0;
    array[*pnt]->tlast=tslot+(array[*pnt]->slottime*tslot)+(2*pdelay)+(2*proc);
    if (randm(>lpr)
    {
        lrct++;                          /** แพคเกจแรกของข้อมูลที่มีขนาดใหญ่ **/
        queue=queue+poisson();           /** ขนาด โดยสุ่ม **/
        if (queue<=(frm-asub))           /** แพคเกจในคิวมีจำนวนน้อยกว่า Aloha subframe **/
        {
            array[*pnt]->cutlpr=0;
        }
        else if (queue>(frm-asub))
        {
            array[*pnt]->pstate=0;
            array[*pnt]->cutlpr=1;
            array[*pnt]->nowsl=mkslot(array[*pnt]->tlast);
            a=arrtime(array[*pnt]->pstate,lam);          /** แพคเกจที่ถูกส่งใหม่ **/
            if (a==0.0)
            {

```

```

        array[*pnt]->nowsl=array[*pnt]->nowsl+1;
        a=arrtime(array[*pnt]->pstate,lam);
    }
    slc=array[*pnt]->nowsl+mkslot(a);
    array[*pnt]->slottime=aloha(slc);
    array[*pnt]->tlast=0.0;
    lrss++;
    goto exitlpr1;
}
}
    tcnt=tcnt+array[*pnt]->tlast-array[*pnt]->gent;           /** เวลาทั้งหมด **/
if (*pnt==40)
{
    s++;
    sttn10[s]=array[*pnt]->tlast-array[*pnt]->gent;
    fprintf(rsv1,"\n%.4f",sttn10[s]);
}
/** คำนวณค่า Variance **/
tv=tv+((array[*pnt]->tlast-array[*pnt]->gent)*(array[*pnt]->tlast-array[*pnt]->gent));
array[*pnt]->pstate=0;
tmp1++;
if (tmp1==30)
{
    fprintf(rsv1,"\n");
    tmp1:=0;
}
cnt4=cnt4+array[*pnt]->coll;           /** จำนวนครั้งที่เกิดการชน **/
cnt3++;
array[*pnt]->coll=0;
array[*pnt]->nowsl=mkslot(array[*pnt]->tlast);
a=arrtime(array[*pnt]->pstate,lam);

```

```

if (a==0.0)
{
    array[*pnt]->nowsl=array[*pnt]->nowsl+random(3);
    a=arrtime(array[*pnt]->pstate,lam);
}
array[*pnt]->gent=(array[*pnt]->nowsl*tslot)+a;
slc=mkslot(array[*pnt]->gent);
array[*pnt]->slottime=aloha(slc);
array[*pnt]->tlast=0.0;
exitlpr1;;
}

pnt++;
}    /** END WHILE **/
} /** END else if (cnt0==1) **/
else if (cnt0>1)    /** มีจำนวนมากกว่าหนึ่งแพคเกจถูกส่งขึ้นในสล็อตนี้ **/
{
while (*pnt!=user+1)
{
if (array[*pnt]->pstate==0)    /** cnt0>1 และ State = 0 **/
{
    array[*pnt]->coll=array[*pnt]->coll+1;
    array[*pnt]->pstate=1;
    array[*pnt]->tlast=tslot+(array[*pnt]->slottime*tslot)+(2*pdelay)+(2*proc);
    array[*pnt]->nowsl=mkslot(array[*pnt]->tlast);
    a=arrtime(array[*pnt]->pstate,lam);
    if (a==0.0)
    {
        array[*pnt]->nowsl=array[*pnt]->nowsl+1;
        a=arrtime(array[*pnt]->pstate,lam);
    }
    slc=array[*pnt]->nowsl+mkslot(a);

```

```

    array[*pnt]->slottime=backlg(slc,lam);
}
else if (array[*pnt]->pstate==1)                                /**cnt0>1 และ State = 1 **/
{
    array[*pnt]->coll=array[*pnt]->coll+1;
    array[*pnt]->pstate=1;
    array[*pnt]->tlast=tslot+(array[*pnt]->slottime*tslot)+(2*pdelay)+(2*proc);
    array[*pnt]->nowsl=mkslot(array[*pnt]->tlast);
    a=arrtime(array[*pnt]->pstate,lam);
    if (a==0.0)
    {
        array[*pnt]->nowsl=array[*pnt]->nowsl+1;
        a=arrtime(array[*pnt]->pstate,lam);
    }
    slc=array[*pnt]->nowsl+mkslot(a);
    array[*pnt]->slottime=backlg(slc,lam);
/** จัดการส่งของสถานีแบ็คคี่อกในช่วง Aloha Subframe **/
}
pnt++;
} /** END WHILE **/
} /** END else if (cnt0>1) **/
stop_slot ;;
pnt=NULL; tmp=NULL;
cnt0=0;
if (percent(slot)==0.0)
queue=0;
}
/** สิ้นสุดการวนลูป (สล๊อตที่ 20,000)**/**/
chk1=tv/cnt2;
printf("q=%0.3f,tv=%0.2f,ct=%0d,t/c=%0.2f:tdl=%0.2f:thp=%0.3f:dly=%0.3fn"
, lam,tv,cnt2,chk1,tcnt,thuput/(sl/2.0),tcnt/cnt2);

```

```

tvar=(tv/cnt2)-((tcnt/cnt2)*(tcnt/cnt2));
printf("lprl=%d:ct-lprl=%d:tvar1=%0.3lf",lrlss,cnt2-lrlss,tvar);
        tvar=(tv/cnt2)-((tcnt/(cnt2-lrlss))*(tcnt/(cnt2-lrlss)));
printf(":dly=%0.3lf:tvar2=%0.3lf:thp1=%0.3f\n",tcnt/(cnt2-lrlss),
        tvar,(thuput-lrlss)/(sl/2.0));
for (i=0;i<user;i++)
    if (array[i]->pstate==1) cnt1++;
/***** CLOSED POINT *****/
for( i = 0; i <user; i ++ )
    farfree( array[ i ] );
/***** START SIMULATION FOR 10th STATION *****/
fprintf(rsv1,"\n");
tmp1=mg1(lam);
fprintf(rsv1,"\n");
for (i=1;i<=tmp1;i++)
    fprintf(rsv1,"%0.3lf\n",mg1q[i]);
/***** STOP SIMULATION *****/
fclose(rsv1);
} /** END MAIN **/

/** ฟังก์ชันสำหรับจำลองการเกิดแพคเกจในแต่ละสถานี โดยให้เป็นไปตามรูปแบบของปัวซอง **/
float arrtime(int stat,float lamd)
{
float data,gent;
float lamda_a,lamda_r;
lamda_a=lamd/(tslot*frm);          /* New message rate per slot*/
lamda_r=lamdar/(tslot*frm);       /* Backlogged message rate per slot*/
data = randm();
if (data<=0.0) data=0.00001;
if (stat==0)
    gent=(-1/lamda_a)*log(1-data);

```

```

else if (stat==1)
    gent=(-1/lamda_r)*log(1-data);
return(gent);
}

```

*/** ฟังก์ชันสำหรับจัดให้การส่งแพคเกจอยู่ในช่วงของ Aloha Subframe **/*

```

long int aloha(long int dat)
{
float frac,num,num1;
int a,inger1;
long int inger,newslot;
double integer,integer1;
num=dat/frm;
frac = modf(num,&integer); inger=(int) integer;
if ((frac*frm)>asub)
{
    num1=randm()*10;
    inger1=(int) integer1;
    newslot=(inger*frm)+frm+inger1;
}
else if ((frac*frm)<=asub)
    newslot=dat;
return (newslot);
}

```

*/** ฟังก์ชันสำหรับจำลองเหตุการณ์ตามรูปแบบของปัวซอง **/*

```

int poisson(void)
{
int n;
float z,ran;
n=0;

```

```

ran=randm();
while (ran<=0.0) ran=randm();
z=(-1.0/arr)*log(ran);
while (z<1.0)
{
    ran=randm();
    while (ran<=0.0) ran=randm();
    n=n+1; z=z+((-1.0/arr)*log(ran));
}
return(n);
}

```

/** ฟังก์ชันสำหรับจำลองการเกิดแบบ M/G/1 สมมติให้การเกิดเป็นแบบปัวซองของทดลองที่20000 timeslot ได้จำนวนที่ตัวใส่ใน mg1q[] **/

```
int mg1(float qrate)
```

```

{
int n;
float z,ran;
n=0;
randomize();
ran=randm();
while (ran<=0.0) ran=randm();
z=(-1.0*frm/(qrate*sl))*log(ran);    /* "qrate/frm" : packet per timeslot */
mg1q[n+1]=z*sl*0.035;
while (z<1.0)
{
    ran=randm();
    while (ran<=0.0) ran=randm();
    n=n+1; z=z+((-1.0*frm/(qrate*sl))*log(ran));
    mg1q[n+1]=z*sl*0.035;
}
}

```

```
printf("mg1=%d:",n);
return(n);
}
```

```
/** ฟังก์ชันสำหรับการจัดการส่งของสถานีแบ็คล็อกในช่วง Aloha Subframe **/
```

```
long int backlg(long int dat,float lam1)
{
float frac,num;
int tem;
long int inger,newslot,dat1;
double integer;
dat1=dat;
num=dat1/frm;
frac = modf(num,&integer); inger=(int) integer;
if ((frac*frm)>asub)
{
while ((frac*frm)>asub)
{
tem=mkslot(arrt(1,lam1));
dat1=dat1+tem;
num=dat1/frm;
frac = modf(num,&integer); inger=(int) integer;
}
newslot=dat1;
} else if ((frac*frm)<=asub)
newslot=dat1;
return(newslot);
}
```

```
/** ฟังก์ชันสำหรับการจำลองการเกิดแพคเกจในแต่ละสถานี โดยให้การเกิดในแต่ละสถานีเป็นไป
ตามรูปแบบของปัวซอง **/
```

```
float arrt(int stat,float lamd)
```

```
{
float data,gent,cal;
float lamda_a,lamda_r;
lamda_a=lamd/(tslot*frm);
lamda_r=lamdar/(tslot*frm);
data=randm();
if (data<=0.0) data=0.00001;
if (stat==0)
gent=(-1/lamda_a)*log(1-data);
else if (stat==1)
gent=(-1/lamda_r)*log(1-data);
return(gent);
}
```

```
/** ฟังก์ชันสำหรับการสร้างช่วงเวลาตามรูปแบบของสล็อต **/
```

```
long int mkslot(float gentime)
```

```
{
float frac,num;
long int inger;
double integer;
num=gentime/tslot;
frac = modf(num,&integer); inger=(long int) integer;
if (frac>0.0) inger=inger+1;
return (inger);
}
```

```
float percent(int slott)
```

```
{
float frac,num;
double integer;
```

```
num=slott/frm;
frac = modf(num,&integer);
return (frac);
}
```

```
float randm()
{
float f;
int i,j;
j=rand() % 10;
for(i=0; i<=j; i++)
{
rand1(); rand2();
}
f=(float) rand1();
if (f<=0.55) return rand2();
else if (f>0.55) return rand1();
}
```

```
float rand1()
{
static long int a=100001;
static char once=1;
int z;
int i=0,inger;
float frac,num;
double integer;
z=rand() % 100;
while (z<=0)
{
i++;
```

```

num=(rand2()*10.0);
frac = modf(num,&integer); inger=(int) integer;
z=random (100)+inger-10;
}
if (once) {
a=(long) z*1000;
once=0;
}
a=(a*125)%2796203;
return (float) a/2796203;
}

```

```

float rand2()
{
static long int a=1;
static char once=1;
int z;
int i=0;
z=rand() % 100;
while (z<=0)
{
i++;
z=rand() % 100;
}
if (once) {
a=(long) z*1;
once=0;
}
a=(a*32719+3)%32749;
return (float) a/32749;
}

```

File “MG_C”

```

#include <stdio.h>
#include <math.h>
#define t5 21
#define inf 999.0
#define satd 0.028875      /* คำนวณเวลาโดยเฉลี่ยของระบบดาวเทียม */
#define satcp 2.857        /* อัตราการมาถึงที่กำหนดให้สำหรับแต่ละสถานี */
#define satv 0.23633      /* ความแปรปรวนของคำนวณเวลาของระบบดาวเทียม */
static double cp1[t5],prevcp[t5];
static double d[t5][t5],rq[t5][t5];
static double calf[t5],sumf[t5];
struct def1 {
    int num;
    int sce[t5];
    int dest[t5];
    int path[t5][t5];
    double lngth[t5];
};
struct def2{
    int start;
    int stop;
    int sat;
};
static struct def1 spath;
static struct def2 link[t5];

double avdlay(double x,int nl);
void dijkst(int n);
double min(double a,double b);
double lngth(int mssg,int sat,double flow,double cpa);

```

```

int adjcp(int n);
void adjf(int nl);
double chdlay(int sat,double f,double c);

main()
{
FILE *fdtxt;
double r,prevd,currd,t7,sumrq,temp_db;
float packet;
int t0,t6,i,j,node,len,s,t,msg,nlink,cflag,lstlink[t5][t5],temp,cnt;
if ((fdtxt = fopen("d:fdtxt.txt", "wt"))== NULL)
{
fprintf(stderr, "Cannot open input file.\n");
exit(1);
}
clrscr();
prevd=0.0; currd=0.0;
msg=1;
link[0].sat=999.0;
/*****กำหนดโครงสร้างของเครือข่าย*****/
printf("How many nodes ? : ");
do {
scanf("%d",&node); fflush(stdin);
} while(node<=0);
if (node>t5)
printf("%d > %d\n",node,t5);
else
for (i=1;i<=node;i++)
{
for (j=1;j<=node;j++)
{

```

```

    d[i][j]=inf;                /** Clear Link's cost **/
    lstlink[i][j]=0;           /** Clear Link's capacity **/
}
d[i][i]=0; rq[i][j]=0;       /** Clear Link's parameter **/
}
printf("Enter link in the form: Start  Finish \n");
printf("Enter: 0  0 to finish\n");
t6=0;
do {
    t6++;
    do {
        printf("Link Number %d :",t6 );
        scanf("%d %d",&i,&j); fflush(stdin);
        if ((i>node)||(j>node))
            printf("Node number is too large. try again\n");
        if ((i<0)||(j<0))
            printf("Node number is too small. try again\n");
    } while((i<0)||(i>node)||(j<0)||(j>node));
}
/*****ใส่ค่าพารามิเตอร์*****/
if (i!=0)
{
    link[t6].start=i; link[t6].stop=j;
    lstlink[i][j]=t6;
    do {
        printf("Is this sattellite link ?(1=yes,0=no) : ");
        scanf("%d",&temp); fflush(stdin);
        link[t6].sat=temp;
        if ((link[t6].sat!=0)&&(link[t6].sat!=1))
            printf("Incorrect number! Try again :");
    } while ((link[t6].sat!=0)&&(link[t6].sat!=1));
    if (link[t6].sat==0)

```

```

{
    do {
        printf("Capacity of this landline : ");
        scanf("%lf",&t7); fflush(stdin);
    } while (t7==0);
    cp1[t6]=t7;    prevcp[t6]=t7;
}
else if (link[t6].sat==1)
{
    cp1[t6]=satcp;    prevcp[t6]=satcp;
}
t7=0.0;
calf[t6]=0.0; sumf[t6]=0.0;
/** กำหนดค่าแวนวโน้มในแต่ละ Link **/
d[i][j]=lngth(msg,link[t6].sat,calf[t6],cp1[t6]);
}
} while((i!=0)||(j!=0));
temp=0;
nlink=t6-1;
printf("The network matrix has pattern as follows :\n");
for (i=1;i<=node;i++)
{
    for (j=1;j<=node;j++)
    {
        printf("%.3f",d[i][j]);
        if (link[lstlink[i][j]].sat==1)
            printf("(S) ");
        else if (link[lstlink[i][j]].sat==0)
            printf("(L) ");
        else printf("(-) ");
    }
}

```

```

printf("\n");
}
getch();
printf("\n");
for (i=1;i<=nlink;i++)
{
printf("link#%d : s=%d , t=%d :",i,link[i].start,link[i].stop);
printf(" length=%0.4f\n",d[link[i].start][link[i].stop]);
}
fprintf(fdtxt, "\nBefore operating the system :");
cflag=adjcp(nlink);          /** ปรับความจุของช่องสัญญาณในแต่ละ Link **/
fprintf(fdtxt, "\n1st Setting trunk speed rate.\n");
printf("\ncp1[8]=%0.3f\n",cp1[8]);
for (i=1;i<=nlink;i++)
{
temp_db=chdelay(link[i].sat,calf[i],cp1[i]);
fprintf(fdtxt, "Link#%d Traffic=%0.4f:Link Speed=%0.4f:Cost[%d][%d]=%0.3f
:Delay=%0.3f\n",i,calf[i],cp1[i],link[i].start,link[i].stop,d[link[i].start][link[i].stop]);
currd+=temp_db;          /** กำหนดค่าหน่วงเวลาโดยเฉลี่ยของระบบ**/
}
fprintf(fdtxt, ":Total Delay=%0.4f\n",currd);
printf("\n*****\n");
dijkstra(node);          /** พิจารณาเส้นทางที่ดีที่สุด **/
/** กำหนดและใส่ค่าพารามิเตอร์ของเส้นทางที่ต้องการพิจารณา **/
printf("How many shortest path do you want ? : ");
scanf("%d",&t6); fflush(stdin);
spath.num=t6;
sumrq=0.0;
fprintf(fdtxt, "\nInput traffic and Find shortest path.\n");
for (i=1;i<=spath.num;i++)
{

```

```

do {
    printf("\nType: Source Destination Traffic :");
    scanf("%d %d %lf",&s,&t,&r); fflush(stdin);
    if ((s>node)|| (t>node))
        printf("Node number is too large. try again\n");
    if ((s<1)|| (t<1))
        printf("Node number is too small. try again\n");
    if (s==t)
        printf("Source and destination are the same. try again\n");
} while((s==t)&&(s<1)&&(t<1)&&(s>node)&&(t>node));
printf("s=%d : t=%d : rq=%.3f",s,t,r);
spath.sce[i]=s; spath.dest[i]=t; rq[s][t]=r;
sumrq+=r;
}
fprintf(fdtxt,"\n");
dijkst(node);                                /** พิจารณาเส้นทางที่ดีที่สุด **/
for (i=1;i<=spath.num;i++)
{
    fprintf(fdtxt,"spath.path[%d] :",i);
    j=0;
    while (spath.path[i][j]!=512)            /** ตรวจสอบแต่ละ Shortest Path **/
    {
        fprintf(fdtxt,"-%2d",spath.path[i][j]);
        j++;
    }
    fprintf(fdtxt," Total cost[%d] =%.3f\n",i,spath.lngth[i]);
}
for (i=1;i<=spath.num;i++)
if (rq[spath.sce[i]][spath.dest[i]]>0)
{
    j=0;

```

```

/** ปรับค่า calf[] ในแต่ละ Link **/
while (spath.path[i][j]!=512)
{
temp=lstlink[spath.path[i][j]][spath.path[i][j+1]];
if (spath.path[i][j]!=512)
    calf[temp]+=rq[spath.sce[i]][spath.dest[i]];
if (spath.path[i][j+1]!=512)
    fprintf(fdtxt,"link[%d][%d]=%d:Assigned traffic[%d][%d]=%.3lf:Taffic flow[%d]=
%.3lf\n",spath.path[i][j],spath.path[i][j+1],temp,spath.sce[i],spath.dest[i],rq[spath.sce [i]]
[spath.dest[i]],temp,calf[temp]);
    j++;
}
}
fprintf(fdtxt,"BEFORE LOOP\n");
for (i=1;i<=nlink;i++)
{
    fprintf(fdtxt," link#%d Traffic=%.4lf:Link Speed=%.4lf",i,calf[i],cp1[i],i);
    fprintf(fdtxt,":Cost[%d][%d]=%.3f\n",link[i].start,link[i].stop,d[link[i].start][link[i].stop]);
}
cflag=adjcp(nlink);                /** ปรับ Service rate ในแต่ละ Link **/
fprintf(fdtxt,"\nAdjust Link Speed\n");
for (i=1;i<=nlink;i++)
{
    fprintf(fdtxt," link#%d Traffic=%.4lf:Link Speed=%.4lf",i,calf[i],cp1[i],i);
    fprintf(fdtxt,":Cost[%d][%d]=%.3f\n",link[i].start,link[i].stop,d[link[i].start][link[i].stop]);
}
prevd=9999.0;
for (i=1;i<=nlink;i++)
    currd+=chdlay(link[i].sat,calf[i],cp1[i]);
fprintf(fdtxt,"Cflag=%d:Start Delay=%.1f min:Total Delay=%.8f min\n\n",cflag,prevd,currd);
/** เริ่มต้นการวนลูป **/

```

```

cnt=0;
while ((cflag==1)&&((prevd<currd)||((prevd-currd)>0.0000001))
{
    cnt++;
    fprintf(fdtxt,"\nSTART LOOP(%d)\n",cnt);
    for (i=1;i<=nlink;i++)          /** ใส่ค่า Cost ของแต่ละเส้นทาง **/
    {
        d[link[i].start][link[i].stop]=lngth(msg,link[i].sat,calf[i],cp1[i]);
        sumf[i]=0.0;
    }
    dijkst(node);                  /** พิจารณาเส้นทางที่ดีที่สุด **/
    for (i=1;i<=spath.num;i++)
    {
        fprintf(fdtxt,"spath.path[%d] :",i);
        j=0;
        while (spath.path[i][j]!=512)      /** ตรวจสอบ Shortest Path **/
        {
            fprintf(fdtxt,"-%d",spath.path[i][j]);
            j++;
        }
        if (rq[spath.sce[i]][spath.dest[i]>0)
        {
            j=0;
            while (spath.path[i][j]!=512)      /** ตรวจสอบแต่ละ Shortest Path **/
            {
                temp=lstlink[spath.path[i][j]][spath.path[i][j+1]];
                if (spath.path[i][j]!=512)
                    sumf[temp]+=rq[spath.sce[i]][spath.dest[i]];
                j++;
            }
            fprintf(fdtxt," Cost of Shortest Path No.%d = %.3f\n",i,spath.lngth[i]);

```

```

}
}
fprintf(fdtxt,"%dst Setting traffic in each link and Adjusting trunk speed\n",cnt);
adjf(nlink);                                /** ปรับอัตราการใช้งานถึงในแต่ละ Link **/
if (cflag==1)
    cflag=adjcp(nlink);                      /** ปรับ Service rate ในแต่ละ Link **/
prevd=currd;
currd=0.0;
for (i=1;i<=nlink;i++)                      /** ใส่ค่า Cost ของแต่ละเส้นทาง **/
{
    d[link[i].start][link[i].stop]=length(msg,link[i].sat,calf[i],cp1[i]);
    sumf[i]=0.0;
}

for (i=1;i<=nlink;i++)
{
    temp_db=chdelay(link[i].sat,calf[i],cp1[i]);
    fprintf(fdtxt,"Link#%d Traffic = %.3lf : Link Speed = %.3lf :Delay=%.3lf"
            ,i,calf[i],cp1[i],temp_db);
    currd+=temp_db;                          /** คำนวณกำหนดช่วงเวลาโดยเฉลี่ยของระบบ**/
    fprintf(fdtxt,":Cost[%d][%d]=%.5fn"
            ,link[i].start,link[i].stop,d[link[i].start][link[i].stop]);
}
currd=currd/sumrq;
fprintf(fdtxt,"Average Dlay=%.9fn",currd);
}
/** สิ้นสุดการวนลูป **/
cflag=adjcp(nlink);                          /** ปรับ Service rate ในแต่ละ Link **/
fclose(fdtxt);
}

```

```

void adjf(int nl)
{
double x0,x1,x2,x3,x4,x5,f0,f1,f2,f3,f4,f5,factor,temp;

int i,j;

x0=0.0; x2=0.2; x4=0.4;

f0=avdlay(x0,nl);

f2=avdlay(x2,nl);

f4=avdlay(x4,nl);

fprintf(fd,"SP1x0=%.3lf:f0=%.5lf:x2=%.3lf:f2=%.5lf:x4=%.3lf:f4=%.5lf\n"
        ,x0,f0,x2,f2,x4,f4);

while ((x4-x0)>0.000001)
{
x1=(x0+x2)/2; f1=avdlay(x1,nl);
x3=(x2+x4)/2; f3=avdlay(x3,nl);
if ((f0<=f1)||(f1<=f2))
{
x4=x2; f4=f2; x2=x1; f2=f1;
} else
if (f2<=f3)
{
x0=x1; x4=x3; f0=f1; f4=f3;
} else
{
x0=x2; x2=x3; f0=f2; f2=f3;
}
}

if ((f0<=f2)&&(f0<=f4))
factor=x0;

else if (f2<=f4)
factor=x2;

else

```

```

factor=x4;
for (i=1;i<=nl;i++)
{
temp=calf[i];
calf[i]=(factor*sumf[i])+((1-factor)*temp);
}
fprintf(fd,"\n");
}

```

```

double avdlay(double x,int nl)
{
int i;
double flow[t5],dlay[t5],serv;
double avrt,t,tot;
for (i=1;i<=nl;i++)
{
flow[i]=(x*sumf[i])+((1-x)*calf[i]);
if (cp1[i]<=flow[i])
{
dlay[i]=999.0;
}
else if (cp1[i]>flow[i])
{
if (link[i].sat==0)
{
dlay[i]=1/(cp1[i]-flow[i]);
} else if (link[i].sat==1)
{
if (flow[i]>satcp)
{
dlay[i]=999.0;
}
}
}
}
}

```

```

    } else if (flow[i]<=satcp)
    {
        t=2*(1-(satd*satd*flow[i]));
        tot=satd+(flow[i]*satv*(1+satd));
        dlay[i]=(satd/2)+(tot/t);
    }
}
}
}
avrt=0.0;
for (i=1;i<=n1;i++)
    avrt=avrt+dlay[i];
return(avrt);
}

```

```

void dijkst(int n)
{
    int t8,t0,i,j,p,s,t,t9,count;
    int q[t5],o[t5],r[t5];
    double t4,l[t5],temp;
    for (i=1;i<=n;i++) o[i]=0;
    for (count=1;count<=spath.num;count++)
    {
        s=spath.sce[count];
        t=spath.dest[count];
        for (i=1;i<=n;i++)
        {
            l[i]=inf;
            q[i]=0;
        }
        q[s]=1; l[s]=0;
    }
}

```

```

p=s;
do {
    for (i=1;i<=n;i++)
        if (q[i]!=1)
            l[i]=min(l[i],l[p]+d[p][i]);
    p=0; t4=inf;
    for (i=1;i<=n;i++)
        if ((q[i]!=1)&&(t4>=l[i]))
            {
                p=i; t4=l[i];
            }
    q[p]=1;
} while(q[t]!=1);
for (j=1;j<=n;j++)
    if (q[j]==1)
        for (i=1;i<=n;i++)
            {
                temp=l[i]+d[i][j];
                if ((i!=j)&&(l[j]==temp))
                    {
                        r[j]=i;
                        if (d[i][j]>=inf)
                            printf("No path connection between %d and %d\n",i,j);
                    }
            }
t8=t5; o[t8]=t; i=r[t];
while (i!=s)
    {
        t8=t8-1; o[t8]=i; i=r[i];
    }
spath.path[count][0]=s;

```

```

j=1;
for (i=t8;i<=t5;i++)
{
    spath.path[count][j]=o[i];
    j++;
}
spath.path[count][j]=512;
spath.lngth[count]=l[t];
}
}

```

```

double min(double a,double b)
{
    double z;
    if (a<b) z=a; else z=b;
    return(z);
}

```

```

double lngth(int mssg,int sat,double flow,double cpa)
{
    double u;
    if (sat==0)
    {
        u=1-(flow/cpa);
        return((mssg/cpa)/(u*u));
    }
    /***** SATELLITE LENGTH *****/
    else if (sat==1)
    {
        u=1-(flow/cpa);
        return((mssg/cpa)/(u*u));
    }
}

```

```

}
}

int adjcp(int n)
{
    double fact,temp;
    int i;
    fact=1;
    for (i=1;i<=n;i++)
    {
        temp=1.02*calf[i]/prevcp[i];
        if (fact<temp)
            fact=temp;
    }
    for (i=1;i<=n;i++)
    {
        if (link[i].sat==0)
            cp1[i]=fact*prevcp[i];
        else if (link[i].sat==1)
        {
            cp1[i]=fact*prevcp[i];
        }
    }
    if (fact>=1)
        return(1);
    else
        return(0);
}

double chdlay(int sat,double f,double c)
{

```

```
double t,tot,tot1;
if (sat==0)
{
tot=1/(c-f);
return(tot);
}
else if (sat==1)      /***** SAT DLAY *****/
{
    if (f>satcp)
        return(999.0);
    else if (f<=satcp)
    {
        tot1=satd;
        t=2*(1-(tot1*tot1*f));
        tot=tot1+(f*satv*(1+tot1));
        return((tot1/2)+(tot/t));
    }
}
}
```

ภาคผนวก ข.

ตารางแสดงผลการทดลองจากบทที่ 3 และ บทที่ 4

ตารางที่ 4

อัตราเร็ว (Packet/frame)	Simulation			Numerical		
	Avg delay	Thuput	Variance	Avg delay	Thuput	Variance
0.001	0.713	0.003	0.04	0.579	0.006	1.27
0.002	0.916	0.007	1.194	0.582	0.012	1.31
0.003	0.734	0.012	0.057	0.584	0.017	1.354
0.004	0.801	0.015	0.321	0.587	0.023	1.398
0.005	0.822	0.02	0.431	0.589	0.029	1.447
0.006	0.853	0.024	1.133	0.592	0.034	1.497
0.007	1.011	0.028	0.921	0.596	0.04	1.547
0.008	0.779	0.03	0.572	0.599	0.045	1.601
0.009	0.874	0.036	1.326	0.603	0.051	1.655
0.01	0.989	0.039	1.615	0.607	0.056	1.714
0.02	1.1	0.074	2.712	0.659	0.105	2.418
0.03	1.193	0.104	4.481	0.735	0.139	3.379
0.04	1.456	0.128	8.033	0.86	0.163	4.981
0.05	1.611	0.151	8.019	1.046	0.176	7.46
0.06	1.864	0.167	9.24	1.229	0.182	10.02
0.07	2.095	0.178	12.14	1.47	0.179	13.56
0.08	2.268	0.189	13.6	1.729	0.177	17.58
0.09	2.543	0.194	16.24	1.999	0.172	22.01

ตารางที่ 4 (ต่อ)

อัตราเร็ว (Packet/frame)	Simulation			Numerical		
	Avg delay	Thuput	Variance	Avg delay	Thuput	Variance
0.1	2.783	0.196	18.85	2.275	0.165	26.74
0.2	4.762	0.17	40.89	4.422	0.132	70.04
0.3	5.919	0.147	58.32	5.53	0.115	96.35
0.4	6.639	0.137	69.49	6.229	0.114	114.2
0.5	7.035	0.127	78.55	6.312	0.103	116.4
0.6	7.378	0.123	85.92	6.67	0.104	125.9
0.7	7.561	0.118	89.03	6.812	0.102	129.8
0.8	7.704	0.116	90.87	6.747	0.098	128
0.9	7.719	0.117	90.65	6.747	0.103	128
1	7.719	0.117	90.65	6.747	0.103	128

แสดง การเปรียบเทียบผลของช่องสัญญาณควมเทียมที่ได้จากสมการ โมเดลและการจำลองระบบ โดยใช้โปรแกรมคอมพิวเตอร์ที่อัตราการมาถึงของข้อมูลต่างๆกันจากบทที่ 3 ที่ค่า $\text{Max User} = 50$

ตารางที่ 5

อัตราเร็ว (Packet/frame)	Simulation			Numerical		
	Avg delay	Thuput	Variance	Avg delay	Thuput	Variance
0.001	0.702	0.005	0.046	0.581	0.006	1.29
0.002	0.772	0.011	0.18	0.584	0.012	1.356
0.003	0.76	0.016	0.125	0.588	0.017	1.426
0.004	0.821	0.022	0.992	0.593	0.023	1.499
0.005	0.81	0.029	0.447	0.598	0.029	1.579
0.006	0.882	0.034	0.814	0.603	0.034	1.664
0.007	0.94	0.039	1.147	0.61	0.04	1.752
0.008	0.916	0.046	1.318	0.616	0.045	1.848

ตารางที่ 5 (ต่อ)

อัตราเร็ว (Packet/frame)	Simulation			Numerical		
	Avg delay	Thuput	Variance	Avg delay	Thuput	Variance
0.009	0.956	0.051	1.47	0.624	0.051	1.946
0.01	0.988	0.059	1.179	0.631	0.056	2.052
0.02	1.293	0.105	9.598	0.764	0.102	3.757
0.03	1.569	0.143	14.91	1.163	0.124	9.073
0.04	1.951	0.173	14.13	1.514	0.149	14.23
0.05	2.406	0.187	21.07	1.983	0.164	21.73
0.06	2.947	0.194	26.2	2.511	0.173	30.95
0.07	3.46	0.194	28.77	3.083	0.179	41.75
0.08	3.941	0.186	33.32	3.871	0.168	57.92
0.09	4.609	0.175	40.59	4.599	0.158	74.07
0.1	5.183	0.163	48.71	5.303	0.147	90.75
0.2	9.556	0.1	130.7	8.913	0.11	191.1
0.3	11.57	0.073	181.2	10.33	0.102	236.8
0.4	12.54	0.073	211.7	10.72	0.09	250.1
0.5	12.99	0.067	243.4	11.43	0.094	274.6
0.6	13.62	0.063	248.2	11.82	0.094	288.6
0.7	13.97	0.063	256.6	11.9	0.091	291.6
0.8	14.16	0.063	261.8	12.59	0.098	316.9
0.9	14.34	0.063	265.5	12.22	0.09	303.1
1	14.34	0.063	265.6	12.22	0.103	303.1

แสดง การเปรียบเทียบผลของช่องสัญญาณความถี่ที่ได้จากสมการโมเดลและการจำลองระบบ โดยใช้โปรแกรมคอมพิวเตอร์ที่อัตราการมาถึงของข้อมูลต่างๆกันจากบทที่ 3 ที่ค่า $\text{Max User} = 75$

ตารางที่ 6

อัตราเร็ว (Packet/frame)	Simulation			Numerical		
	Avg delay	Thuput	Variance	Avg delay	Thuput	Variance
0.001	0.8439	0.0062	1.0368	0.582	0.008	1.3123
0.002	0.8324	0.0144	0.4241	0.587	0.016	1.4034
0.003	0.7784	0.0224	0.6031	0.593	0.023	1.5024
0.004	0.8434	0.0304	0.7287	0.6	0.031	1.61
0.005	0.8478	0.038	0.9112	0.608	0.038	1.7283
0.006	0.8771	0.0454	1.0327	0.617	0.046	1.8523
0.007	0.9603	0.0537	1.9998	0.627	0.053	1.9879
0.008	1.0707	0.0591	2.1636	0.638	0.06	2.133
0.009	1.1197	0.0655	4.5083	0.649	0.067	2.2908
0.01	1.1925	0.0736	2.8999	0.662	0.074	2.4568
0.02	1.5314	0.1328	17.775	0.877	0.132	5.2059
0.03	2.1586	0.1709	23.555	1.624	0.151	15.926
0.04	2.8329	0.1914	31.995	2.349	0.172	28.034
0.05	3.5964	0.1904	39.676	3.291	0.18	45.88
0.06	4.7907	0.173	52.208	5.081	0.153	85.598
0.07	6.0999	0.1522	67.177	6.659	0.137	125.63
0.08	7.4568	0.1325	93.018	8.153	0.122	168.03
0.09	9.1505	0.1066	123.03	9.308	0.116	203.5
0.1	10.24	0.0998	151.95	10.4	0.107	239.24
0.2	18.088	0.0463	407.86	15.79	0.08	444.57
0.3	20.591	0.0456	539.59	17.846	0.075	535.22
0.4	21.91	0.0348	609.08	18.412	0.066	561.44
0.5	22.827	0.0318	655.85	18.716	0.061	575.64
0.6	23.388	0.0302	667.65	19.082	0.061	593.04
0.7	23.763	0.0303	691.31	19.029	0.057	590.47
0.8	23.785	0.0293	735.87	19.816	0.062	628.41

ตารางที่ 6 (ต่อ)

อัตราเร็ว (Packet/frame)	Simulation			Numerical		
	Avg delay	Thuput	Variance	Avg delay	Thuput	Variance
0.9	24.047	0.0288	722.74	20.56	0.062	665.18
1	24.047	0.0288	677.72	20.56	0.062	665.18

แสดง การเปรียบเทียบผลของช่องสัญญาณความถี่ที่ได้จากสมการ โมเดลและการจำลองระบบ
โดยใช้โปรแกรมคอมพิวเตอร์ที่อัตราการมาถึงของข้อมูลต่างๆกันจากบทที่ 3 ที่ค่า

$$\text{Max User} = 100$$

ตารางที่ 7

อัตราเร็ว (Packet/frame)	Simulation			Numerical		
	Avg delay	Thuput	Variance	Avg delay	Thuput	Variance
0.001	0.797	0.01	0.314	0.583	0.01	1.335
0.002	0.781	0.019	0.47	0.59	0.02	1.453
0.003	0.826	0.029	0.669	0.598	0.029	1.584
0.004	0.843	0.04	1.56	0.608	0.039	1.729
0.005	0.995	0.049	0.909	0.619	0.048	1.89
0.006	1.06	0.056	1.313	0.632	0.057	2.066
0.007	1.007	0.065	1.962	0.647	0.065	2.256
0.008	1.1	0.074	2.387	0.663	0.074	2.464
0.009	1.185	0.082	6.887	0.68	0.082	2.691
0.01	1.181	0.09	8.341	0.7	0.09	2.9345
0.02	1.854	0.155	23.03	1.344	0.139	11.67
0.03	2.873	0.187	45.49	2.298	0.169	27.14
0.04	3.887	0.188	49.99	3.94	0.17	59.42

ตารางที่ 7 (ต่อ)

อัตราเร็ว (Packet/frame)	Simulation			Numerical		
	Avg delay	Thuput	Variance	Avg delay	Thuput	Variance
0.05	5.892	0.157	70.76	7.21	0.132	140.8
0.06	8.858	0.114	127.9	10.07	0.112	228.1
0.07	12.6	0.08	224.3	12.46	0.1	312
0.08	16.22	0.057	365.1	14.68	0.089	398.3
0.09	19.52	0.049	471.1	16.37	0.084	469.3
0.1	21.59	0.04	606.9	17.95	0.078	540.1
0.2	32.34	0.039	1267	25.63	0.059	938.7
0.3	35.08	0.027	1382	28.52	0.054	1112
0.4	37.01	0.016	1556	29.31	0.048	1162
0.5	37.11	0.015	1727	29.74	0.045	1189
0.6	38.37	0.015	1704	30.25	0.044	1222
0.7	39.09	0.015	1848	30.17	0.042	1218
0.8	39.29	0.014	1868	29.57	0.038	1178
0.9	39.39	0.014	1921	30.36	0.04	1230
1	39.71	0.014	1896	32.31	0.04	1360

แสดง การเปรียบเทียบผลของช่องสัญญาณควมเทียมที่ได้จากสมการ โมเดลและการจำลองระบบ
โดยใช้โปรแกรมคอมพิวเตอร์ที่อัตราการมาถึงของข้อมูลต่างๆกัน ที่ค่าจากบทที่ 3
ที่ Max User = 125

ผลการจำลองในบทที่ 4 โดยใช้โปรแกรมเพื่อหาค่าหน่วยเวลาโดยเฉลี่ยของข้อมูลที่เกิดขึ้น
ในสถานีดาวเทียมที่ใช้ระบบดาวเทียมที่ $User = 50$ และที่อัตราการมาถึงของข้อมูลต่างๆกัน ดังนี้

ตารางที่ 8

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	2	1.14	0.94	1.04
2	2	0.59	0.56	0.57
3	2	0.99	0.57	0.78
4	2	0.57	0.56	0.57
5	2	0.87	0.58	0.73
6	5	1.14	0.58	0.76
7	2	0.92	0.57	0.74
8	4	1.05	0.57	0.74
9	3	0.59	0.57	0.58
10	3	1.14	0.56	0.95

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.005 packets/frame

ตารางที่ 9

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	5	0.9	0.58	0.65
2	1	0.56	0.56	0.56
3	1	0.56	0.56	0.56
4	3	0.7	0.56	0.61
5	4	0.58	0.56	0.57
6	6	1.08	0.56	0.79
7	3	0.57	0.56	0.56
8	2	0.58	0.56	0.57

ตารางที่ 9 (ต่อ)

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
8	2	0.58	0.56	0.57
9	2	1.12	0.56	0.84
10	2	0.57	0.56	0.56
11	5	0.99	0.56	0.73

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.006 packets/frame

ตารางที่ 10

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	5	1.14	0.58	0.76
2	5	1.04	0.56	0.75
3	8	1.08	0.56	0.68
4	8	1.02	0.56	0.77
5	6	1.13	0.56	0.76
6	3	0.57	0.56	0.56
7	5	0.94	0.69	0.82
8	6	1.22	0.56	0.7
9	6	1.17	0.57	0.81
10	4	0.9	0.56	0.72
11	9	1	0.56	0.67

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.007 packets/frame

ตารางที่ 11

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	4	0.9	0.56	0.66
2	5	1	0.56	0.66
3	5	0.89	0.58	0.73
4	5	1.11	0.56	0.74
5	5	0.95	0.57	0.65
6	5	0.97	0.58	0.75
7	5	0.89	0.56	0.66
8	3	1.07	0.56	0.74
9	6	0.89	0.56	0.63
10	6	0.99	0.56	0.71

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.008 packets/frame

ตารางที่ 12

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	4	0.56	0.56	0.57
2	2	0.87	0.58	0.72
3	4	1.18	0.58	0.94
4	5	1.09	0.58	0.75
5	6	1.05	0.57	0.81
6	4	0.91	0.58	0.67
7	8	1.12	0.56	0.7
8	4	0.78	0.57	0.64
9	6	2.01	0.57	0.99
10	7	1.05	0.56	0.75

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.009 packets/frame

ตารางที่ 13

ครั้งที่	จำนวน packet.	Maximum	Minimum	Avg Delay
1	11	1.28	0.56	0.73
2	9	0.89	0.56	0.67
3	9	1.16	0.56	0.88
4	6	0.87	0.58	0.7
5	10	1.03	0.56	0.73
6	11	1.03	0.56	0.7
7	8	1.07	0.56	0.79
8	6	1.11	0.56	0.86
9	6	1.13	0.56	0.72
10	8	0.93	0.56	0.7

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.01 packets/frame

ตารางที่ 14

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	11	1.55	0.56	0.87
2	10	2.85	0.57	1.04
3	12	1.22	0.57	0.79
4	17	1.05	0.56	0.7
5	18	1.23	0.56	0.76
6	13	2.48	0.56	0.91
7	10	4.07	0.56	0.96
8	11	1.04	0.57	0.75
9	16	2.33	0.56	0.9
10	12	1.11	0.57	0.79

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.02 packets/frame

ตารางที่ 15

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	20	2.68	0.56	1.07
2	20	1.38	0.56	0.81
3	18	1.47	0.56	0.85
4	24	4.23	0.56	0.88
5	13	1.13	0.56	0.81
6	24	3.71	0.56	0.89
7	20	1.13	0.56	0.7
8	20	4.54	0.56	0.85
9	16	3.07	0.56	0.93
10	19	5.29	0.57	1.15

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.03 packets/frame

ตารางที่ 16

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	30	9.04	0.57	1.54
2	27	5.35	0.56	1.17
3	22	8.47	0.56	1.52
4	26	5.54	0.57	1.35
5	27	5.83	0.56	1.56
6	21	9.2	0.56	1.39
7	20	6.72	0.56	1.36
8	29	7.94	0.56	1.47
9	34	12.4	0.56	1.38
10	23	1.51	0.56	0.88

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.04 packets/frame

ตารางที่ 17

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	39	15.4	0.56	1.89
2	35	9.95	0.56	1.62
3	39	8.88	0.56	1.07
4	27	8.09	0.56	2.57
5	37	7.46	0.56	1.54
6	33	7.38	0.56	1.81
7	33	5.67	0.56	1.02
8	29	9.03	0.56	1.65
9	33	9.25	0.56	1.58
10	36	7.48	0.56	1.72
11	40	9.44	0.56	2.06
12	38	5.76	0.56	1.57
13	34	5.85	0.56	1.45

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.05 packets/frame

ตารางที่ 18

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	38	10.8	0.56	2.03
2	49	16.8	0.57	5.47
3	52	25.8	0.56	8.81
4	35	11.3	0.56	1.66
5	33	21	0.56	3.66
6	40	9.66	0.56	2.02
7	52	45	0.56	4.63

ตารางที่ 18 (ต่อ)

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
8	47	11.5	0.56	4.4
9	47	44.3	0.56	5.42
10	37	20.5	0.56	4.41
11	36	8.29	0.56	2.44
12	38	14.4	0.56	1.75
13	47	44.3	0.56	5.42
14	33	5.18	0.56	1.25
15	50	26.3	0.56	5.52
16	50	26.6	0.56	5.58

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.06 packets/frame

ตารางที่ 19

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	55	53.6	0.57	14.2
2	46	97.5	0.56	31.6
3	53	84.4	0.56	14.9
4	52	152	0.57	40.6
5	35	9.84	0.56	2.09
6	37	59.2	0.56	12.3
7	55	106	0.57	13.2
8	58	26.9	0.57	6.2
9	55	61.6	0.56	8.08
10	58	26.9	0.57	6.2

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.07 packets/frame

ตารางที่ 20

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	55	53.6	0.57	14.2
2	54	141	0.57	22.7
3	54	121	0.57	58
4	52	45.5	0.56	10.5
5	52	37.8	0.56	5.77
6	57	33.1	0.57	6.67
7	55	51.6	0.56	9.74
8	59	50	0.57	7.24
9	52	32.3	0.57	4.31
10	55	56.1	0.57	12.3

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.08 packets/frame

ผลการจำลองในบทที่ 4 โดยใช้โปรแกรมเพื่อหาค่าหน่วยเวลาโดยเฉลี่ยของข้อมูลที่เกิดขึ้น
ในสถานีดาวเทียมที่ใช้ระบบดาวเทียมที่ $User = 75$ และที่อัตราการมาถึงของข้อมูลต่างๆกัน ดังนี้

ตารางที่ 21

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	3	0.87	0.56	0.72
2	3	0.62	0.59	0.63
3	2	0.59	0.56	0.56
4	1	0.85	0.56	0.68
5	1	0.93	0.56	0.71
6	3	0.84	0.56	0.73
7	3	0.94	0.56	0.67

ตารางที่ 21 (ต่อ)

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
8	2	1.59	0.56	0.80
9	4	1.03	0.56	0.72
10	2	0.88	0.56	0.56

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.005 packets/frame

ตารางที่ 22

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	2	0.94	0.56	0.75
2	3	0.59	0.59	0.57
3	1	0.56	0.56	0.56
4	4	0.89	0.56	0.72
5	1	0.56	0.56	0.56
6	3	0.97	0.56	0.70
7	3	0.87	0.56	0.67
8	4	1.59	0.56	0.91
9	2	1.02	0.56	0.80
10	2	0.56	0.56	0.56

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.006 packets/frame

ตารางที่ 23

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	3	0.59	0.56	0.57
2	3	1.10	0.56	0.74

ตารางที่ 23 (ต่อ)

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
3	4	0.96	0.56	0.66
4	5	1.25	0.57	0.86
5	5	0.85	0.56	0.63
6	9	1.26	0.56	0.72
7	6	0.58	0.56	0.57
8	4	0.98	0.57	0.68
9	7	1.57	0.57	0.83
10	5	1.57	0.57	1.01

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.007

ตารางที่ 24

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	6	1.00	0.56	0.66
2	7	1.19	0.57	0.79
3	8	1.12	0.56	0.72
4	9	3.88	0.57	1.04
5	4	1.19	0.56	0.75
6	4	1.32	0.56	0.97
7	6	3.95	0.57	1.28
8	8	1.50	0.57	0.89
9	9	1.14	0.56	0.80
10	5	1.5	0.56	0.82

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.008 packets/frame

ตารางที่ 25

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	7	0.99	0.58	0.69
2	7	1.16	0.58	0.67
3	4	0.96	0.57	0.70
4	5	1.16	0.57	0.75
5	5	0.86	0.56	0.68
6	6	1.14	0.57	0.79
7	4	1.10	0.58	0.83
8	6	0.96	0.57	0.68
9	9	0.95	0.57	0.75
10	7	0.99	0.58	0.69

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.009 packets/frame

ตารางที่ 26

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	8	1.02	0.58	0.77
2	8	1.07	0.56	0.73
3	7	1.12	0.58	0.80
4	7	0.87	0.56	0.68
5	3	0.87	0.57	0.77
6	5	0.87	0.58	0.75
7	10	1.18	0.57	0.82
8	6	0.87	0.56	0.67
9	6	1.22	0.58	0.87
10	8	1.87	0.56	0.95

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.01 packets/frame

ตารางที่ 27

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	15	1.99	0.56	0.80
2	12	3.52	0.57	0.92
3	12	1.07	0.56	0.76
4	13	1.27	0.56	0.85
5	11	1.27	0.56	1.03
6	15	1.27	0.56	0.74
7	21	1.15	0.56	1.03
8	8	1.24	0.57	0.89
9	15	1.3	0.57	0.75
10	15	3.24	0.57	0.95

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.02 packets/frame

ตารางที่ 28

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	19	9.29	0.56	2.10
2	25	4.53	0.56	0.99
3	18	4.18	0.56	1.37
4	17	9.29	0.56	1.22
5	14	2.62	0.56	1.12
6	25	9.60	0.56	2.11
7	21	4.87	0.56	1.49
8	10	8.53	0.57	1.65
9	18	6.10	0.56	1.51
10	20	8.80	0.56	1.65

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.03 packets/frame

ตารางที่ 29

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	35	8.28	0.56	2.55
2	33	15.50	0.56	2.86
3	29	14.21	0.56	2.12
4	26	7.40	0.56	1.53
5	20	8.95	0.56	1.47
6	36	14.96	0.56	2.32
7	32	10.67	0.57	3.15
8	27	14.35	0.56	1.90
9	17	8.23	0.56	2.18
10	29	14.02	0.56	3.57

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.04 packets/frame

ตารางที่ 30

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	38	8.05	0.57	3.54
2	29	13.1	0.56	2.61
3	27	38.8	0.57	5.05
4	30	8.79	0.56	3.03
5	37	13.9	0.56	2.36
6	31	13	0.56	2.50
7	35	30.05	0.57	3.41
8	29	12.21	0.56	3.32
9	33	16.72	0.56	3.29
10	32	19.95	0.56	3.02

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.05 packets/frame

ตารางที่ 31

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	22	56	0.56	29.1
2	34	29.7	0.56	8.8
3	32	42.2	0.56	14.6
4	30	31.5	0.56	17.4
5	25	34.5	0.56	12.1
6	27	51.3	0.56	13.2
7	41	28.6	0.57	7.2
8	32	38.2	0.56	29.6
9	27	29.1	0.57	36.2
10	29	63.2	0.56	20.3

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.06 packets/frame

ตารางที่ 32

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	32	48.2	0.57	58.8
2	21	69.9	0.57	95.3
3	24	38.7	0.57	52.2
4	26	40.7	0.56	75.3
5	32	36.9	0.56	45.7
6	30	59.7	0.56	29
7	34	54.4	0.57	46.8
8	27	42.2	0.56	36.5
9	25	39.6	0.57	57.2
10	26	38.4	0.56	47.9

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.07 packets/frame

ตารางที่ 33

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	25	39.6	0.57	57.2
2	25	70.6	0.57	104
3	28	65.5	0.57	54.4
4	21	99.7	0.56	119
5	25	78.8	0.56	106
6	25	65.1	0.56	93.1
7	19	13.1	0.57	136
8	22	77.6	0.56	193
9	30	85.1	0.57	156
10	20	66.9	0.56	101

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.08 packets/frame

ผลการจำลองโดยใช้โปรแกรมเพื่อหาค่าหน่วยเวลาโดยเฉลี่ยของแพคเกจที่เกิดขึ้นในสถานีควาเทียมที่ใช้ระบบควาเทียมที่มีจำนวน Max User = 100 และที่อัตราการมาถึงของข้อมูลต่างๆกัน

ตารางที่ 34

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	2	0.773	0.56	0.683
2	3	0.971	0.56	0.732
3	4	1.069	0.56	0.896
4	5	0.945	0.57	0.762
5	2	1.034	0.57	0.785
6	3	1.030	0.56	0.698
7	2	0.677	0.57	0.569

ตารางที่ 34 (ต่อ)

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
8	2	0.673	0.56	0.581
9	5	0.938	0.56	0.667
10	3	0.890	0.56	0.699

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.005 packets/frame

ตารางที่ 35

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	3	0.588	0.57	0.576
2	3	0.883	0.56	0.669
3	3	1.069	0.56	0.896
4	5	1.062	0.56	0.808
5	2	0.927	0.57	0.707
6	3	1.030	0.56	0.724
7	3	0.589	0.58	0.577
8	2	0.782	0.57	0.672
9	5	1.261	0.56	0.716
10	3	0.890	0.56	0.723

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.006 packets/frame

ตารางที่ 36

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	6	1.13	0.57	0.66
2	5	1.501	0.56	1.004
3	5	1.606	0.56	0.85
4	4	1.214	0.56	0.968
5	4	1.31	0.57	1.151
6	3	1.746	0.56	1.064
7	5	1.479	0.57	1.067
8	5	1.34	0.57	0.866
9	5	1.11	0.56	0.817
10	3	0.98	0.56	0.838

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.007 packets/frame

ตารางที่ 37

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	6	0.59	0.57	0.58
2	8	1.13	0.56	0.78
3	7	1.60	0.56	0.84
4	7	0.58	0.56	0.57
5	7	1.21	0.57	0.73
6	6	1.6	0.56	0.78
7	4	1.64	0.57	1.05
8	4	2.34	0.86	1.25
9	6	1.083	0.56	1.09
10	6	1.6	0.56	0.78

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.008 packets/frame

ตารางที่ 38

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	4	0.99	0.56	0.70
2	3	0.58	0.56	0.57
3	10	0.96	0.56	0.61
4	4	0.95	0.57	0.67
5	2	0.91	0.58	0.74
6	4	1.02	0.57	0.69
7	5	1.09	0.58	0.70
8	12	1.10	0.56	0.69
9	3	0.58	0.56	0.57
10	8	0.99	0.56	0.72

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.009 packets/frame

ตารางที่ 39

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	9	0.98	0.56	0.65
2	4	1.24	0.87	1.05
3	8	1.01	0.56	0.75
4	7	1.12	0.56	0.74
5	6	1.08	0.56	0.80
6	10	0.87	0.56	0.65
7	4	0.87	0.56	0.65
8	6	1.14	0.56	0.80
9	9	1.79	0.56	0.92
10	7	1.04	0.56	0.73

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.01 packets/frame

ตารางที่ 40

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	16	3.22	0.56	1.15
2	12	1.83	0.56	0.79
3	9	2.11	0.57	1.02
4	14	11.68	0.56	1.64
5	15	1.21	0.57	0.83
6	7	1.00	0.57	0.67
7	15	3.59	0.56	1.15
8	18	5.68	0.56	1.20
9	14	1.68	0.56	0.85
10	14	6.16	0.56	1.67

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.02 packets/frame

ตารางที่ 41

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	22	13.38	0.56	1.50
2	24	24.53	0.57	2.93
3	27	7.96	0.56	2.44
4	22	13.41	0.56	2.32
5	22	14.28	0.56	3.71
6	22	11.24	0.56	2.49
7	22	5.94	0.56	1.12
8	19	7.51	0.56	1.55
9	11	2.81	0.56	1.00
10	26	9.22	0.56	1.75
11	16	17.86	0.56	4.03

ตารางที่ 41 (ต่อ)

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
12	21	37.72	0.56	4.88

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.03 packets/frame

ตารางที่ 42

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	23	32.14	0.56	7.27
2	34	24.79	0.56	3.06
3	28	17.97	0.56	3.41
4	34	32.36	0.56	4.60
5	25	47.59	0.56	10.94
6	28	29.13	0.56	5.70
7	21	9.00	0.58	2.49
8	24	7.54	0.58	3.00
9	30	26.68	0.56	5.79
11	22	34.61	0.56	7.97
12	23	42.00	0.56	5.36

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.04 packets/frame

ตารางที่ 43

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	35	16.60	0.56	6.05
2	26	26.37	0.57	11.49
3	30	34.17	0.56	12.44

ตารางที่ 43 (ต่อ)

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
4	30	37.62	0.56	6.52
5	30	36.78	0.56	6.22
6	39	32.19	0.56	9.71
7	34	60.76	0.56	17.46
8	32	47.60	0.56	11.07
9	30	70.9	0.56	15.1
10	21	48.7	0.56	14

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.05 packets/frame

ตารางที่ 44

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	35	47.82	0.56	12.53
2	38	38.18	0.56	8.57
3	28	33.90	0.57	8.53
4	29	78.22	0.56	14.20
5	30	44.77	0.56	16.56
6	31	42.5	0.58	9.93
7	23	60.8	0.56	19.8
8	28	41	0.58	16
9	28	59.9	0.57	11.7
10	31	42.5	0.58	9.93

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.06 packets/frame

ตารางที่ 45

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	35	28.12	0.56	6.31
2	41	18.69	0.57	7.19
3	39	39.95	0.56	13.59
4	37	62.09	0.56	13.09
5	38	96.10	0.56	21.97
6	34	29.23	0.56	8.36
7	33	70.56	0.57	20.10
8	38	55.25	0.80	20.48
9	35	38.21	0.57	15.50
10	40	35.3	0.56	10.8

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.07 packets/frame

ตารางที่ 46

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	40	37.78	0.56	10.74
2	37	31.35	0.57	9.90
3	39	108.32	0.56	54.39
4	38	62.16	0.57	26.05
5	33	52.32	0.57	15.18
6	32	108.31	2.14	60.06
7	40	82.67	0.56	22.09
8	37	57.19	0.57	18.53
9	33	80.82	0.57	29.22
10	39	48.59	0.57	18.45
11	32	52.50	0.59	16.62

ตารางที่ 46 (ต่อ)

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
12	39	59.84	0.86	16.10
13	37	39.07	0.56	11.59
14	28	64.38	0.57	33.57
15	38	60.82	0.56	19.43

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.08 packets/frame

ตารางที่ 47

ครั้งที่	จำนวน packet	Maximum	Minimum	Avg Delay
1	34	140.90	22.25	87.81
2	33	125.18	20.18	63.99
3	32	103.7	0.58	42.09
4	32	92.09	3.55	44.64
5	35	193.53	31.24	140.62
6	30	170.27	1.57	78.87
7	29	227.45	2.92	146.95
8	21	147	39.2	72
9	26	128	0.57	51.4
10	21	184	1	39.4

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.09 packets/frame

ผลการจำลองโดยใช้โปรแกรมเพื่อหาค่าหน่วยเวลาโดยเฉลี่ยของแพคเกจที่เกิดขึ้นในสถานี ดาวเทียมที่ใช้ระบบดาวเทียมที่มีจำนวน $\text{Max User} = 125$ และที่อัตราการมาถึงของข้อมูลต่างๆกัน

ตารางที่ 48

ครั้งที่	จำนวน packet	Maximum	Minimum	average
1	3	1.14	0.56	0.8
2	6	1.03	0.57	0.73
3	2	0.896	0.56	0.828
4	3	1.08	0.56	0.855
5	3	0.937	0.56	0.837
6	3	0.58	0.57	0.821
7	4	1.14	0.57	0.729
8	3	1.097	0.56	0.739
9	3	0.92	0.57	0.75
10	4	0.91	0.57	0.721

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.005 packets/frame

ตารางที่ 49

ครั้งที่	จำนวน packet	Maximum	Minimum	average
1	3	0.91	0.56	0.751
2	4	0.57	0.57	0.608
3	4	1.065	0.56	0.687
4	4	1.223	0.56	0.851
5	2	0.826	0.56	0.694
6	2	1.098	0.57	0.83
7	3	0.818	0.57	0.645
8	4	1.187	0.56	0.872

ตารางที่ 49 (ต่อ)

ครั้งที่	จำนวน packet	Maximum	Minimum	average
9	4	0.87	0.57	0.71
10	3	0.91	0.57	0.672

แสดง ผลการจำลองที่อัตราการมาถึงของข้อมูล : 0.006 packets/frame

ตารางที่ 50

ครั้งที่	จำนวน packet	Maximum	Minimum	Average
1	7	1.19	0.56	0.9
2	5	1.05	0.57	0.86
3	4	1.12	0.56	0.71
4	4	1.54	0.56	0.92
5	5	1.68	0.56	0.92
6	6	0.78	0.57	0.63
7	3	1.25	0.57	0.80
8	8	1.09	0.58	0.72
9	5	1.08	0.57	0.68
10	6	1.43	0.57	0.64

แสดง ผลการจำลองที่อัตราการมาถึงของข้อมูล : 0.007 packets/frame

ตารางที่ 51

ครั้งที่	จำนวน packet	Maximum	Minimum	Average
1	5	1.08	0.58	0.68
2	7	1.32	0.56	0.72
3	5	0.86	0.57	0.64

ตารางที่ 51

ครั้งที่	จำนวน packet	Maximum	Minimum	Average
4	4	1.43	0.56	0.86
5	4	0.95	0.56	0.76
6	4	1.15	0.57	0.74
7	5	1.25	0.57	0.79
8	5	0.97	0.58	0.75
9	6	1.21	0.57	0.83
10	6	9.93	0.57	2.86

แสดง ผลการจำลองที่อัตราการมาถึงของข้อมูล : 0.008 packets/frame

ตารางที่ 52

ครั้งที่	จำนวน packet	Maximum	Minimum	Average
1	4	0.87	0.56	0.71
2	6	1.10	0.56	0.72
3	13	1.16	0.57	0.85
4	4	1.06	0.58	0.70
5	5	0.99	0.57	0.67
6	7	1.00	0.56	0.81
7	7	1.01	0.57	0.84
8	5	0.98	0.56	0.78
9	6	0.97	0.57	0.79
10	7	0.70	0.56	0.60

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.009 packets/frame

ตารางที่ 53

ครั้งที่	จำนวน packet	Maximum	Minimum	Average
1	4	0.87	0.56	0.71
2	7	1.22	0.56	0.76
3	6	0.93	0.56	0.72
4	11	1.18	0.56	0.71
5	5	1.32	0.58	0.89
6	5	0.87	0.58	0.69
7	9	1.10	0.56	0.80
8	9	1.09	0.57	0.81
9	8	4.31	0.56	1.3
10	7	1.11	0.58	0.89

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.01 packets/frame

ตารางที่ 54

ครั้งที่	จำนวน packet	Maximum	Minimum	Average
1	6	0.9319	0.54	0.72
2	15	1.47	0.57	1.24
3	13	6.91	0.56	2.53
4	10	5.31	0.57	1.93
5	13	11.05	0.56	1.56
6	15	1.01	0.56	0.69
7	14	7.64	0.56	1.51
8	15	1.39	0.56	0.72
9	13	7.09	0.56	1.78
10	16	5.02	0.57	1.71

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.02 packets/frame

ตารางที่ 55

ครั้งที่	จำนวน packet	Maximum	Minimum	Average
1	20	25.78	0.56	6.22
2	12	36.49	0.56	5.31
3	16	9.81	0.56	1.90
4	11	6.83	0.56	1.67
5	11	5.60	0.58	1.41
6	18	5.41	0.56	1.42
7	14	38.52	0.56	5.10
8	20	25.78	0.56	6.22
9	19	13.16	0.56	3.17
10	20	20.10	0.56	4.12

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.03 packets/frame

ตารางที่ 56

ครั้งที่	จำนวน packet	Maximum	Minimum	Average
1	29	47.57	0.57	13.90
2	19	9.09	0.56	1.26
3	24	9.76	0.57	2.38
4	20	8.93	0.56	2.09
5	32	14.93	0.56	2.63
6	23	7.13	0.56	2.63
7	24	22.90	0.56	2.67
8	30	14	0.57	4.49
9	35	31	0.56	4.66
10	30	14	0.57	4.49

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.04 packets/frame

ตารางที่ 57

ครั้งที่	จำนวน packet	Maximum	Minimum	Average
1	28	45.17	0.57	9.69
2	27	37.92	0.56	8.51
3	31	12.37	0.56	4.90
4	22	20.13	0.57	8.83
5	32	25.26	0.56	8.05
6	32	46.8	0.56	9.66
7	32	35.5	0.56	7.8
8	32	28.3	0.56	8.92
9	32	24.6	0.56	7.21
10	32	46.8	0.56	9.66

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.05 packets/frame

ตารางที่ 58

ครั้งที่	จำนวน packet	Maximum	Minimum	average
1	32	55.38	0.57	16.17
2	29	57.54	0.57	20.76
3	29	48.43	0.56	15.12
4	32	55.38	0.57	16.17
5	31	44.44	0.56	7.68
6	38	40.4	0.57	7.16
7	38	24.01	0.56	7.15
8	34	50.8	0.57	13.2
9	37	40.2	0.56	9.13
10	37	38.9	0.57	8.32

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.06 packets/frame

ตารางที่ 59

ครั้งที่	จำนวน packet	Maximum	Minimum	average
1	35	64.43	0.57	10.68
2	27	37.65	0.56	17.97
3	29	111.91	0.57	64.06
4	33	78.59	0.58	29.19
5	19	100	0.98	30.3
6	22	76.1	12.5	37.5
7	39	70.9	12.5	37.5
8	23	97.1	2.55	38.2
9	26	121	0.57	32.5
10	19	100	0.98	30.3

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.07 packets/frame

ตารางที่ 60

ครั้งที่	จำนวน packet	Maximum	Minimum	average
1	27	168.85	0.57	86.47
2	29	167.99	0.57	105.37
3	29	168.82	0.57	97.31
4	27	168.85	0.57	86.47
5	21	179	9.18	59.4
6	27	126	0.57	40.4
7	32	95.5	1.14	28.2
8	21	119	1.97	54.1
9	24	112	0.77	38.5
10	21	116	5.12	64.1

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.08 packets/frame

ตารางที่ 61

ครั้งที่	จำนวน packet	Maximum	Minimum	average
1	26	327.24	10.97	201.8
2	26	245.96	0.58	106.96
3	20	290.15	38.20	146.58
4	23	272.88	1.98	171.96
5	20	176	13.6	68.8
6	19	109	4.51	58.1
7	23	141	0.58	55.8
8	25	150	0.85	58.4
9	25	122	1.28	49.3
10	20	176	13.6	68.8

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.09 packets/frame

ตารางที่ 62

ครั้งที่	จำนวน packet	Maximum	Minimum	average
1	23	288.18	6.37	177.41
2	25	336.68	40.36	211.53
3	17	349.70	5.00	185.26
4	23	288.18	6.27	177.41
5	25	119	0.82	49.8
6	18	172	17.3	78.9
7	28	79.9	1.47	40.1
8	28	116	0.57	33.2
9	18	137	4.03	78.8
10	25	114	0.21	52.2

แสดง ผลการจำลองที่ อัตราการมาถึงของข้อมูล : 0.1 packets/frame

ภาคผนวก ค.

ผลการของทำงานของโปรแกรม MG_C (บทที่ 5 ตัวอย่างที่ 2)

Before Operating Traunk :

Link#1 Traffic=0.0000 :Link Speed=85.7140 :Delay=0.012 :Cost[1][2]=0.012
 Link#2 Traffic=0.0000 :Link Speed=85.7140 :Delay=0.012 :Cost[2][8]=0.012
 Link#3 Traffic=0.0000 :Link Speed=64.2860 :Delay=0.016 :Cost[1][3]=0.016
 Link#4 Traffic=0.0000 :Link Speed=85.7140 :Delay=0.012 :Cost[3][7]=0.012
 Link#5 Traffic=0.0000 :Link Speed=21.4286 :Delay=0.047 :Cost[2][9]=0.047
 Link#6 Traffic=0.0000 :Link Speed=21.4286 :Delay=0.047 :Cost[6][3]=0.047
 Link#7 Traffic=0.0000 :Link Speed=64.2860 :Delay=0.016 :Cost[7][4]=0.016
 Link#8 Traffic=0.0000 :Link Speed=2.8570 :Delay=0.029 :Cost[4][3]=0.350
 Link#9 Traffic=0.0000 :Link Speed=2.8570 :Delay=0.029 :Cost[6][4]=0.350
 Link#10 Traffic=0.0000:Link Speed=21.4286 :Delay=0.047 :Cost[4][10]=0.047
 Link#11 Traffic=0.0000:Link Speed=85.7140 :Delay=0.012 :Cost[7][10]=0.012
 Link#12 Traffic=0.0000:Link Speed=64.2860 :Delay=0.016 :Cost[8][7]=0.016
 Link#13 Traffic=0.0000:Link Speed=21.4286 :Delay=0.047 :Cost[10][5]=0.047
 Link#14 Traffic=0.0000:Link Speed=85.7140 :Delay=0.012 :Cost[5][1]=0.012
 Link#15 Traffic=0.0000:Link Speed=64.2860 :Delay=0.016 :Cost[8][9]=0.016
 Link#16 Traffic=0.0000:Link Speed=85.7140 :Delay=0.012 :Cost[8][1]=0.012
 Link#17 Traffic=0.0000:Link Speed=85.7140 :Delay=0.012 :Cost[9][5]=0.012
 Link#18 Traffic=0.0000:Link Speed=85.7140 :Delay=0.012 :Cost[5][6]=0.012
 :Total Delay=0.4000

Input traffic and find shortest path.

Shortest Path[1] :- 1- 3- 7- 4 Total cost[1] =0.043
 Shortest Path[2] :- 3- 7-10 Total cost[2] =0.023
 Shortest Path[3] :- 2- 8- 9- 5- 6 Total cost[3] =0.051
 Shortest Path[4] :- 5- 1- 2- 8 Total cost[4] =0.035
 Shortest Path[5] :- 6- 3- 7- 4 Total cost[5] =0.074
 Shortest Path[6] :- 4-10- 5- 1- 3 Total cost[6] =0.121

Shortest Path[7] :- 5- 1- 3 Total cost[7] =0.027

BEFORE LOOP

link#1 Traffic=44.0000	:Link Speed=85.7140	:Cost[1][2]=0.012
link#2 Traffic=87.0000	:Link Speed=85.7140	:Cost[2][8]=0.012
link#3 Traffic=84.0000	:Link Speed=64.2860	:Cost[1][3]=0.016
link#4 Traffic=99.5000	:Link Speed=85.7140	:Cost[3][7]=0.012
link#5 Traffic=0.0000	:Link Speed=21.4286	:Cost[2][9]=0.047
link#6 Traffic=1.5000	:Link Speed=21.4286	:Cost[6][3]=0.047
link#7 Traffic=52.5000	:Link Speed=64.2860	:Cost[7][4]=0.016
link#8 Traffic=0.0000	:Link Speed=2.8570	:Cost[4][3]=0.350
link#9 Traffic=0.0000	:Link Speed=2.8570	:Cost[6][4]=0.350
link#10 Traffic=5.0000	:Link Speed=21.4286	:Cost[4][10]=0.047
link#11 Traffic=47.0000	:Link Speed=85.7140	:Cost[7][10]=0.012
link#12 Traffic=0.0000	:Link Speed=64.2860	:Cost[8][7]=0.016
link#13 Traffic=5.0000	:Link Speed=21.4286	:Cost[10][5]=0.047
link#14 Traffic=77.0000	:Link Speed=85.7140	:Cost[5][1]=0.012
link#15 Traffic=43.0000	:Link Speed=64.2860	:Cost[8][9]=0.016
link#16 Traffic=0.0000	:Link Speed=85.7140	:Cost[8][1]=0.012
link#17 Traffic=43.0000	:Link Speed=85.7140	:Cost[9][5]=0.012
link#18 Traffic=43.0000	:Link Speed=85.7140	:Cost[5][6]=0.012

Adjust Link Speed

link#1 Traffic=44.0000	:Link Speed=114.2391	:Cost[1][2]=0.012
link#2 Traffic=87.0000	:Link Speed=114.2391	:Cost[2][8]=0.012
link#3 Traffic=84.0000	:Link Speed=85.6800	:Cost[1][3]=0.016
link#4 Traffic=99.5000	:Link Speed=114.2391	:Cost[3][7]=0.012
link#5 Traffic=0.0000	:Link Speed=28.5599	:Cost[2][9]=0.047
link#6 Traffic=1.5000	:Link Speed=28.5599	:Cost[6][3]=0.047
link#7 Traffic=52.5000	:Link Speed=85.6800	:Cost[7][4]=0.016
link#8 Traffic=0.0000	:Link Speed=3.8078	:Cost[4][3]=0.350
link#9 Traffic=0.0000	:Link Speed=3.8078	:Cost[6][4]=0.350
link#10 Traffic=5.0000	:Link Speed=28.5599	:Cost[4][10]=0.047
link#11 Traffic=47.0000	:Link Speed=114.2391	:Cost[7][10]=0.012
link#12 Traffic=0.0000	:Link Speed=85.6800	:Cost[8][7]=0.016
link#13 Traffic=5.0000	:Link Speed=28.5599	:Cost[10][5]=0.047
link#14 Traffic=77.0000	:Link Speed=114.2391	:Cost[5][1]=0.012

link#15 Traffic=43.0000 :Link Speed=85.6800 :Cost[8][9]=0.016
link#16 Traffic=0.0000 :Link Speed=114.2391 :Cost[8][1]=0.012
link#17 Traffic=43.0000 :Link Speed=114.2391 :Cost[9][5]=0.012
link#18 Traffic=43.0000 :Link Speed=114.2391 :Cost[5][6]=0.012
Start Delay=9999.0 min: Total Delay=1.47240888 min

START LOOP(1)

Shortest Path[1] :-1-2-8-7-4 Cost of shortest past No.1 = 0.267
Shortest Path[2] :-3-7-10 Cost of shortest past No.2 = 0.551
Shortest Path[3] :-2-9-5-6 Cost of shortest past No.3 = 0.080
Shortest Path[4] :-5-1-2-8 Cost of shortest past No.4 = 0.260
Shortest Path[5] :-6-4 Cost of shortest past No.5 = 0.263
Shortest Path[6] :-4-10-5-6-3 Cost of shortest past No.6 = 0.164
Shortest Path[7] :-5-6-3 Cost of shortest past No.7 = 0.062

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 53.578 :Link Speed = 92.784 :Delay=0.026 :Cost[1][2]=0.06036
Link#2 Traffic = 88.502 :Link Speed = 92.784 :Delay=0.234 :Cost[2][8]=5.06161
Link#3 Traffic = 68.224 :Link Speed = 69.588 :Delay=0.733 :Cost[1][3]=37.37687
Link#4 Traffic = 89.640 :Link Speed = 92.784 :Delay=0.318 :Cost[3][7]=9.38702
Link#5 Traffic = 8.076 :Link Speed = 23.196 :Delay=0.066 :Cost[2][9]=0.10146
Link#6 Traffic = 7.416 :Link Speed = 23.196 :Delay=0.063 :Cost[6][3]=0.09315
Link#7 Traffic = 52.218 :Link Speed = 69.588 :Delay=0.058 :Cost[7][4]=0.23064
Link#8 Traffic = 0.000 :Link Speed = 3.093 :Delay=0.029 :Cost[4][3]=0.32335
Link#9 Traffic = 0.282 :Link Speed = 3.093 :Delay=0.063 :Cost[6][4]=0.39141
Link#10 Traffic = 5.000 :Link Speed = 23.196 :Delay=0.055 :Cost[4][10]=0.07006
Link#11 Traffic = 47.000 :Link Speed = 92.784 :Delay=0.022 :Cost[7][10]=0.04426
Link#12 Traffic = 9.578 :Link Speed = 69.588 :Delay=0.020 :Cost[9][5]=0.03744
Link#18 Traffic = 49.198 :Link Speed = 92.784 :Delay=0.023 :Cost[5][6]=0.04884

Average Delay=0.008499685

START LOOP(2)

Shortest Path[1] :-1-2-9-5-6-4 Cost of shortest past No.1 = 0.640
Shortest Path[2] :-3-7-10 Cost of shortest past No.2 = 9.431
Shortest Path[3] :-2-9-5-6 Cost of shortest past No.3 = 0.188
Shortest Path[4] :-5-1-2-8 Cost of shortest past No.4 = 5.314
Shortest Path[5] :-6-4 Cost of shortest past No.5 = 0.391
Shortest Path[6] :-4-10-5-6-3 Cost of shortest past No.6 = 0.282

Shortest Path[7] :-5-6-3 Cost of shortest past No.7 = 0.142

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 55.255	:Link Speed = 89.672	:Delay=0.029	:Cost[1][2]=0.07570
Link#2 Traffic = 86.701	:Link Speed = 89.672	:Delay=0.337	:Cost[2][8]=10.15767
Link#3 Traffic = 65.462	:Link Speed = 67.255	:Delay=0.558	:Cost[1][3]=20.93664
Link#4 Traffic = 87.914	:Link Speed = 89.672	:Delay=0.569	:Cost[3][7]=29.00560
Link#5 Traffic = 11.554	:Link Speed = 22.418	:Delay=0.092	:Cost[2][9]=0.18993
Link#6 Traffic = 8.452	:Link Speed = 22.418	:Delay=0.072	:Cost[6][3]=0.11493
Link#7 Traffic = 50.105	:Link Speed = 67.255	:Delay=0.058	:Cost[7][4]=0.22866
Link#8 Traffic = 0.000	:Link Speed = 2.989	:Delay=0.029	:Cost[4][3]=0.33457
Link#9 Traffic = 2.395	:Link Speed = 2.989	:Delay=0.321	:Cost[6][4]=8.48360
Link#10 Traffic = 5.000	:Link Speed = 22.418	:Delay=0.057	:Cost[4][10]=0.07389
Link#11 Traffic = 47.000	:Link Speed = 89.672	:Delay=0.023	:Cost[7][10]=0.04925
Link#12 Traffic = 9.191	:Link Speed = 67.255	:Delay=0.017	:Cost[8][7]=0.01995
Link#13 Traffic = 5.000	:Link Speed = 22.418	:Delay=0.057	:Cost[10][5]=0.07389
Link#14 Traffic = 69.717	:Link Speed = 89.672	:Delay=0.050	:Cost[5][1]=0.22519
Link#15 Traffic = 33.511	:Link Speed = 67.255	:Delay=0.030	:Cost[8][9]=0.05906
Link#16 Traffic = 0.000	:Link Speed = 89.672	:Delay=0.011	:Cost[8][1]=0.01115
Link#17 Traffic = 45.064	:Link Speed = 89.672	:Delay=0.022	:Cost[9][5]=0.04506
Link#18 Traffic = 52.347	:Link Speed = 89.672	:Delay=0.027	:Cost[5][6]=0.06436
Average Delay=0.010749090			

START LOOP(3)

Shotest Path[1] :-1-2-9-5-6-4	Cost of shortest past No.1 = 8.859
Shotest Path[2] :-3-7-10	Cost of shortest past No.2 = 29.055
Shotest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 0.299
Shotest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 10.459
Shotest Path[5] :-6-4	Cost of shortest past No.5 = 8.484
Shotest Path[6] :-4-10-5-6-3	Cost of shortest past No.6 = 0.327
Shotest Path[7] :-5-6-3	Cost of shortest past No.7 = 0.179

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic= 55.621	:Link Speed = 89.288	:Delay=0.030	:Cost[1][2]=0.07878
Link#2 Traffic = 86.308	:Link Speed = 89.288	:Delay=0.336	:Cost[2][8]=10.05362
Link#3 Traffic = 64.859	:Link Speed = 66.966	:Delay=0.475	:Cost[1][3]=15.08380
Link#4 Traffic = 87.537	:Link Speed = 89.288	:Delay=0.571	:Cost[3][7]=29.13051
Link#5 Traffic = 12.313	:Link Speed = 22.322	:Delay=0.100	:Cost[2][9]=0.22283

Link#6 Traffic = 8.678	:Link Speed = 22.322	:Delay=0.073	:Cost[6][3]=0.11990
Link#7 Traffic = 49.643	:Link Speed = 66.966	:Delay=0.058	:Cost[7][4]=0.22315
Link#8 Traffic = 0.000	:Link Speed = 2.976	:Delay=0.029	:Cost[4][3]=0.33601
Link#9 Traffic = 2.857	:Link Speed = 2.976	:Delay=0.377	:Cost[6][4]=209.73045
Link#10 Traffic = 5.000	:Link Speed = 22.322	:Delay=0.058	:Cost[4][10]=0.07439
Link#11 Traffic = 47.000	:Link Speed = 89.288	:Delay=0.024	:Cost[7][10]=0.04993
Link#12 Traffic = 9.106	:Link Speed = 66.966	:Delay=0.017	:Cost[8][7]=0.02000
Link#13 Traffic = 5.000	:Link Speed = 22.322	:Delay=0.058	:Cost[10][5]=0.07439
Link#14 Traffic = 69.480	:Link Speed = 89.288	:Delay=0.050	:Cost[5][1]=0.22758
Link#15 Traffic = 33.202	:Link Speed = 66.966	:Delay=0.030	:Cost[8][9]=0.05874
Link#16 Traffic = 0.000	:Link Speed = 89.288	:Delay=0.011	:Cost[8][1]=0.01120
Link#17 Traffic = 45.515	:Link Speed = 89.288	:Delay=0.023	:Cost[9][5]=0.04660
Link#18 Traffic = 53.035	:Link Speed = 89.288	:Delay=0.028	:Cost[5][6]=0.06794

Average Delay=0.010688178

START LOOP(4)

Shortest Path[1] :-1-2-8-7-4	Cost of shortest past No.1 = 10.376
Shortest Path[2] :-3-7-10	Cost of shortest past No.2 = 29.180
Shortest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 0.337
Shortest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 10.360
Shortest Path[5] :-6-3-7-4	Cost of shortest past No.5 = 29.474
Shortest Path[6] :-4-3	Cost of shortest past No.6 = 0.336
Shortest Path[7] :-5-6-3	Cost of shortest past No.7 = 0.188

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 58.947	:Link Speed = 88.783	:Delay=0.034	:Cost[1][2]=0.09974
Link#2 Traffic = 87.042	:Link Speed = 88.783	:Delay=0.574	:Cost[2][8]=29.29622
Link#3 Traffic = 59.381	:Link Speed = 66.588	:Delay=0.139	:Cost[1][3]=1.28203
Link#4 Traffic = 84.240	:Link Speed = 88.783	:Delay=0.220	:Cost[3][7]=4.30147
Link#5 Traffic = 14.905	:Link Speed = 22.196	:Delay=0.137	:Cost[2][9]=0.41762
Link#6 Traffic = 10.437	:Link Speed = 22.196	:Delay=0.085	:Cost[6][3]=0.16051
Link#7 Traffic = 49.884	:Link Speed = 66.588	:Delay=0.060	:Cost[7][4]=0.23867
Link#8 Traffic = 0.422	:Link Speed = 2.959	:Delay=0.080	:Cost[4][3]=0.45980
Link#9 Traffic = 2.616	:Link Speed = 2.959	:Delay=0.348	:Cost[6][4]=25.06314
Link#10 Traffic = 4.578	:Link Speed = 22.196	:Delay=0.057	:Cost[4][10]=0.07151
Link#11 Traffic = 47.000	:Link Speed = 88.783	:Delay=0.024	:Cost[7][10]=0.05086
Link#12 Traffic = 12.645	:Link Speed = 66.588	:Delay=0.019	:Cost[8][7]=0.02288

Link#13 Traffic = 4.578 :Link Speed = 22.196 :Delay=0.057 :Cost[10][5]=0.07151
 Link#14 Traffic = 67.328 :Link Speed = 88.783 :Delay=0.047 :Cost[5][1]=0.19288
 Link#15 Traffic = 30.397 :Link Speed = 66.588 :Delay=0.028 :Cost[8][9]=0.05084
 Link#16 Traffic = 0.000 :Link Speed = 88.783 :Delay=0.011 :Cost[8][1]=0.01126
 Link#17 Traffic = 45.303 :Link Speed = 88.783 :Delay=0.023 :Cost[9][5]=0.04696
 Link#18 Traffic = 54.552 :Link Speed = 88.783 :Delay=0.029 :Cost[5][6]=0.07577
 Average Delay=0.008977034

START LOOP(5)

Shortest Path[1] :-1-2-9-5-6-3-7-4 Cost of shortest past No.1 = 5.341
 Shortest Path[2] :-3-7-10 Cost of shortest past No.2 = 4.352
 Shortest Path[3] :-2-9-5-6 Cost of shortest past No.3 = 0.540
 Shortest Path[4] :-5-1-2-8 Cost of shortest past No.4 = 29.589
 Shortest Path[5] :-6-3-7-4 Cost of shortest past No.5 = 4.701
 Shortest Path[6] :-4-10-5-6-3 Cost of shortest past No.6 = 0.379
 Shortest Path[7] :-5-6-3 Cost of shortest past No.7 = 0.236

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.122 :Link Speed = 87.353 :Delay=0.037 :Cost[1][2]=0.11780
 Link#2 Traffic = 85.640 :Link Speed = 87.353 :Delay=0.584 :Cost[2][8]=29.77586
 Link#3 Traffic = 57.446 :Link Speed = 65.515 :Delay=0.124 :Cost[1][3]=1.00633
 Link#4 Traffic = 84.737 :Link Speed = 87.353 :Delay=0.382 :Cost[3][7]=12.76554
 Link#5 Traffic = 17.482 :Link Speed = 21.838 :Delay=0.230 :Cost[2][9]=1.15077
 Link#6 Traffic = 12.882 :Link Speed = 21.838 :Delay=0.112 :Cost[6][3]=0.27224
 Link#7 Traffic = 49.970 :Link Speed = 65.515 :Delay=0.064 :Cost[7][4]=0.27110
 Link#8 Traffic = 0.409 :Link Speed = 2.912 :Delay=0.079 :Cost[4][3]=0.46473
 Link#9 Traffic = 2.530 :Link Speed = 2.912 :Delay=0.337 :Cost[6][4]=20.04157
 Link#10 Traffic = 4.591 :Link Speed = 21.838 :Delay=0.058 :Cost[4][10]=0.07342
 Link#11 Traffic = 47.000 :Link Speed = 87.353 :Delay=0.025 :Cost[7][10]=0.05365
 Link#12 Traffic = 12.233 :Link Speed = 65.515 :Delay=0.019 :Cost[8][7]=0.02308
 Link#13 Traffic = 4.591 :Link Speed = 21.838 :Delay=0.058 :Cost[10][5]=0.07342
 Link#14 Traffic = 66.568 :Link Speed = 87.353 :Delay=0.048 :Cost[5][1]=0.20221
 Link#15 Traffic = 29.407 :Link Speed = 65.515 :Delay=0.028 :Cost[8][9]=0.05025
 Link#16 Traffic = 0.000 :Link Speed = 87.353 :Delay=0.011 :Cost[8][1]=0.01145
 Link#17 Traffic = 46.889 :Link Speed = 87.353 :Delay=0.025 :Cost[9][5]=0.05335
 Link#18 Traffic = 56.912 :Link Speed = 87.353 :Delay=0.033 :Cost[5][6]=0.09427
 Average Delay=0.01026157

START LOOP(6)

Shortest Path[1] :-1-3-7-4	Cost of Short past No.1 = 14.043
Shortest Path[2] :-3-7-10	Cost of Short past No.2 = 12.819
Shortest Path[3] :-2-9-5-6	Cost of Short past No.3 = 1.298
Shortest Path[4] :-5-1-2-8	Cost of Short past No.4 = 30.096
Shortest Path[5] :-6-3-7-4	Cost of Short past No.5 = 13.309
Shortest Path[6] :-4-3	Cost of Short past No.6 = 0.465
Shortest Path[7] :-5-6-3	Cost of Short past No.7 = 0.367

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 59.687	:Link Speed = 86.837	:Delay=0.037	:Cost[1][2]=0.11780
Link#2 Traffic = 84.517	:Link Speed = 86.837	:Delay=0.431	:Cost[2][8]=16.13002
Link#3 Traffic = 57.273	:Link Speed = 65.129	:Delay=0.127	:Cost[1][3]=1.05527
Link#4 Traffic = 85.135	:Link Speed = 86.837	:Delay=0.587	:Cost[3][7]=29.95250
Link#5 Traffic = 18.170	:Link Speed = 21.709	:Delay=0.283	:Cost[2][9]=1.73287
Link#6 Traffic = 13.330	:Link Speed = 21.709	:Delay=0.119	:Cost[6][3]=0.30917
Link#7 Traffic = 50.038	:Link Speed = 65.129	:Delay=0.066	:Cost[7][4]=0.28599
Link#8 Traffic = 0.532	:Link Speed = 2.894	:Delay=0.094	:Cost[4][3]=0.51878
Link#9 Traffic = 2.462	:Link Speed = 2.894	:Delay=0.329	:Cost[6][4]=15.49475
Link#10 Traffic = 4.468	:Link Speed = 21.709	:Delay=0.058	:Cost[4][10]=0.07303
Link#11 Traffic = 47.000	:Link Speed = 86.837	:Delay=0.025	:Cost[7][10]=0.05472
Link#12 Traffic = 11.903	:Link Speed = 65.129	:Delay=0.019	:Cost[8][7]=0.02299
Link#13 Traffic = 4.468	:Link Speed = 21.709	:Delay=0.058	:Cost[10][5]=0.07303
Link#14 Traffic = 65.960	:Link Speed = 86.837	:Delay=0.048	:Cost[5][1]=0.19922
Link#15 Traffic = 28.614	:Link Speed = 65.129	:Delay=0.027	:Cost[8][9]=0.04885
Link#16 Traffic = 0.000	:Link Speed = 86.837	:Delay=0.012	:Cost[8][1]=0.01152
Link#17 Traffic = 46.784	:Link Speed = 86.837	:Delay=0.025	:Cost[9][5]=0.05413
Link#18 Traffic = 57.292	:Link Speed = 86.837	:Delay=0.034	:Cost[5][6]=0.09948

Average Delay=0.010836211

START LOOP(7)

Shortest Path[1] :-1-2-8-7-4	Cost of shortest past No.1 = 16.557
Shortest Path[2] :-3-7-10	Cost of shortest past No.2 = 30.007
Shortest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 1.886
Shortest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 16.447
Shortest Path[5] :-6-4	Cost of shortest past No.5 = 15.495
Shortest Path[6] :-4-3	Cost of shortest past No.6 = 0.519

Shortest Path[7] :-5-6-3 Cost of shortest past No.7 = 0.409

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.576	:Link Speed = 86.477	:Delay=0.039	:Cost[1][2]=0.12891
Link#2 Traffic = 84.781	:Link Speed = 86.477	:Delay=0.590	:Cost[2][8]=30.07747
Link#3 Traffic = 55.831	:Link Speed = 64.858	:Delay=0.111	:Cost[1][3]=0.79595
Link#4 Traffic = 84.175	:Link Speed = 86.477	:Delay=0.434	:Cost[3][7]=16.32339
Link#5 Traffic = 18.795	:Link Speed = 21.619	:Delay=0.354	:Cost[2][9]=2.71017
Link#6 Traffic = 13.699	:Link Speed = 21.619	:Delay=0.126	:Cost[6][3]=0.34464
Link#7 Traffic = 50.062	:Link Speed = 64.858	:Delay=0.068	:Cost[7][4]=0.29626
Link#8 Traffic = 0.645	:Link Speed = 2.882	:Delay=0.107	:Cost[4][3]=0.57569
Link#9 Traffic = 2.438	:Link Speed = 2.882	:Delay=0.326	:Cost[6][4]=14.59535
Link#10 Traffic = 4.355	:Link Speed = 21.619	:Delay=0.058	:Cost[4][10]=0.07254
Link#11 Traffic = 47.000	:Link Speed = 86.477	:Delay=0.025	:Cost[7][10]=0.05549
Link#12 Traffic = 12.887	:Link Speed = 64.858	:Delay=0.019	:Cost[8][7]=0.02401
Link#13 Traffic = 4.355	:Link Speed = 21.619	:Delay=0.058	:Cost[10][5]=0.07254
Link#14 Traffic = 65.407	:Link Speed = 86.477	:Delay=0.047	:Cost[5][1]=0.19480
Link#15 Traffic = 27.894	:Link Speed = 64.858	:Delay=0.027	:Cost[8][9]=0.04747
Link#16 Traffic = 0.000	:Link Speed = 86.477	:Delay=0.012	:Cost[8][1]=0.01156
Link#17 Traffic = 46.689	:Link Speed = 86.477	:Delay=0.025	:Cost[9][5]=0.05463
Link#18 Traffic = 57.637	:Link Speed = 86.477	:Delay=0.035	:Cost[5][6]=0.10397

Average Delay=0.011212111

START LOOP(8)

Shortest Path[1] :-1-3-7-4	Cost of shortest past No.1 = 17.416
Shortest Path[2] :-3-7-10	Cost of shortest past No.2 = 16.379
Shortest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 2.869
Shortest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 30.401
Shortest Path[5] :-6-4	Cost of shortest past No.5 = 14.595
Shortest Path[6] :-4-3	Cost of shortest past No.6 = 0.576
Shortest Path[7] :-5-6-3	Cost of shortest past No.7 = 0.449

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.326	:Link Speed = 86.071	:Delay=0.039	:Cost[1][2]=0.12987
Link#2 Traffic = 84.167	:Link Speed = 86.071	:Delay=0.525	:Cost[2][8]=23.75220
Link#3 Traffic = 55.758	:Link Speed = 64.554	:Delay=0.114	:Cost[1][3]=0.83451
Link#4 Traffic = 84.383	:Link Speed = 86.071	:Delay=0.593	:Cost[3][7]=30.21932
Link#5 Traffic = 19.159	:Link Speed = 21.518	:Delay=0.424	:Cost[2][9]=3.86825

Link#6 Traffic = 13.914	:Link Speed = 21.518	:Delay=0.132	:Cost[6][3]=0.37220
Link#7 Traffic = 50.076	:Link Speed = 64.554	:Delay=0.069	:Cost[7][4]=0.30799
Link#8 Traffic = 0.710	:Link Speed = 2.869	:Delay=0.115	:Cost[4][3]=0.61575
Link#9 Traffic = 2.424	:Link Speed = 2.869	:Delay=0.324	:Cost[6][4]=14.48830
Link#10 Traffic = 4.290	:Link Speed = 21.518	:Delay=0.058	:Cost[4][10]=0.07250
Link#11 Traffic = 47.000	:Link Speed = 86.071	:Delay=0.026	:Cost[7][10]=0.05638
Link#12 Traffic = 12.693	:Link Speed = 64.554	:Delay=0.019	:Cost[8][7]=0.02400
Link#13 Traffic = 4.290	:Link Speed = 21.518	:Delay=0.058	:Cost[10][5]=0.07250
Link#14 Traffic = 65.085	:Link Speed = 86.071	:Delay=0.048	:Cost[5][1]=0.19543
Link#15 Traffic = 27.474	:Link Speed = 64.554	:Delay=0.027	:Cost[8][9]=0.04695
Link#16 Traffic = 0.000	:Link Speed = 86.071	:Delay=0.012	:Cost[8][1]=0.01162
Link#17 Traffic = 46.633:	Link Speed = 86.071	:Delay=0.025	:Cost[9][5]=0.05534
Link#18 Traffic = 57.838:	Link Speed = 86.071	:Delay=0.035	:Cost[5][6]=0.10798

Average Delay=0.012038523

START LOOP(9)

Shortest Path[1] :-1-2-9-5-6-4	Cost of shortest past No.1 = 18.650
Shortest Path[2] :-3-7-10	Cost of shortest past No.2 = 30.276
Shortest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 4.032
Shortest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 24.078
Shortest Path[5] :-6-4	Cost of shortest past No.5 = 14.488
Shortest Path[6] :-4-3	Cost of shortest past No.6 = 0.616
Shortest Path[7] :-5-6-3	Cost of shortest past No.7 = 0.480

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.411	:Link Speed = 85.977	:Delay=0.039	:Cost[1][2]=0.13154
Link#2 Traffic = 84.069	:Link Speed = 85.977	:Delay=0.524	:Cost[2][8]=23.60255
Link#3 Traffic = 55.622	:Link Speed = 64.484	:Delay=0.113	:Cost[1][3]=0.82113
Link#4 Traffic = 84.292	:Link Speed = 85.977	:Delay=0.593	:Cost[3][7]=30.25215
Link#5 Traffic = 19.343	:Link Speed = 21.494	:Delay=0.465	:Cost[2][9]=4.64204
Link#6 Traffic = 13.949	:Link Speed = 21.494	:Delay=0.133	:Cost[6][3]=0.37752
Link#7 Traffic = 49.953	:Link Speed = 64.484	:Delay=0.069	:Cost[7][4]=0.30543
Link#8 Traffic = 0.721	:Link Speed = 2.866	:Delay=0.117	:Cost[4][3]=0.62292
Link#9 Traffic = 2.547	:Link Speed = 2.866	:Delay=0.339	:Cost[6][4]=28.12689
Link#10 Traffic = 4.279	:Link Speed = 21.494	:Delay=0.058	:Cost[4][10]=0.07253
Link#11 Traffic = 47.000	:Link Speed = 85.977	:Delay=0.026	:Cost[7][10]=0.05659
Link#12 Traffic = 12.662	:Link Speed = 64.484	:Delay=0.019	:Cost[8][7]=0.02401

Link#13 Traffic = 4.279	:Link Speed = 21.494	:Delay=0.058	:Cost[10][5]=0.07253
Link#14 Traffic = 65.033	:Link Speed = 85.977	:Delay=0.048	:Cost[5][1]=0.19600
Link#15 Traffic = 27.407	:Link Speed = 64.484	:Delay=0.027	:Cost[8][9]=0.04691
Link#16 Traffic = 0.000	:Link Speed = 85.977	:Delay=0.012	:Cost[8][1]=0.01163
Link#17 Traffic = 46.749	:Link Speed = 85.977	:Delay=0.025	:Cost[9][5]=0.05587
Link#18 Traffic = 57.995	:Link Speed = 85.977	:Delay=0.036	:Cost[5][6]=0.10981

Average Delay=0.012298914

START LOOP(10)

Shortest Path[1] :-1-2-8-7-4 Cost of shortest past No.1 = 24.064

Shortest Path[2] :-3-7-10 Cost of shortest past No.2 = 30.309

Shortest Path[3] :-2-9-5-6 Cost of shortest past No.3 = 4.808

Shortest Path[4] :-5-1-2-8 Cost of shortest past No.4 = 23.930

Shortest Path[5] :-6-4 Cost of shortest past No.5 = 28.127

Shortest Path[6] :-4-3 Cost of shortest past No.6 = 0.623

Shortest Path[7] :-5-6-3 Cost of shortest past No.7 = 0.487

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.690	:Link Speed = 85.840	:Delay=0.040	:Cost[1][2]=0.13571
Link#2 Traffic = 84.157	:Link Speed = 85.840	:Delay=0.594	:Cost[2][8]=30.30053
Link#3 Traffic = 55.173	:Link Speed = 64.381	:Delay=0.109	:Cost[1][3]=0.75941
Link#4 Traffic = 83.991	:Link Speed = 85.840	:Delay=0.541	:Cost[3][7]=25.09815
Link#5 Traffic = 19.533	:Link Speed = 21.460	:Delay=0.519	:Cost[2][9]=5.78112
Link#6 Traffic = 14.062	:Link Speed = 21.460	:Delay=0.135	:Cost[6][3]=0.39211
Link#7 Traffic = 49.962	:Link Speed = 64.381	:Delay=0.069	:Cost[7][4]=0.30967
Link#8 Traffic = 0.755	:Link Speed = 2.861	:Delay=0.121	:Cost[4][3]=0.64523
Link#9 Traffic = 2.538	:Link Speed = 2.861	:Delay=0.338	:Cost[6][4]=27.41379
Link#10 Traffic = 4.245	:Link Speed = 21.460	:Delay=0.058	:Cost[4][10]=0.07241
Link#11 Traffic = 47.000	:Link Speed = 85.840	:Delay=0.026	:Cost[7][10]=0.05690
Link#12 Traffic = 12.971	:Link Speed = 64.381	:Delay=0.019	:Cost[8][7]=0.02436
Link#13 Traffic = 4.245	:Link Speed = 21.460	:Delay=0.058	:Cost[10][5]=0.07241
Link#14 Traffic = 64.864	:Link Speed = 85.840	:Delay=0.048	:Cost[5][1]=0.19508
Link#15 Traffic = 27.186	:Link Speed = 64.381	:Delay=0.027	:Cost[8][9]=0.04654
Link#16 Traffic = 0.000	:Link Speed = 85.840	:Delay=0.012	:Cost[8][1]=0.01165
Link#17 Traffic = 46.719	:Link Speed = 85.840	:Delay=0.026	:Cost[9][5]=0.05609
Link#18 Traffic = 58.100	:Link Speed = 85.840	:Delay=0.036	:Cost[5][6]=0.11155

Average Delay=0.012641859

START LOOP(11)

Shortest Path[1] :-1-3-7-4	Cost of shortest past No.1 = 26.167
Shortest Path[2] :-3-7-10	Cost of shortest past No.2 = 25.155
Shortest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 5.949
Shortest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 30.631
Shortest Path[5] :-6-3-7-4	Cost of shortest past No.5 = 25.800
Shortest Path[6] :-4-3	Cost of shortest past No.6 = 0.645
Shortest Path[7] :-5-6-3	Cost of shortest past No.7 = 0.504

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.632	:Link Speed = 85.726	:Delay=0.040	:Cost[1][2]=0.13613
Link#2 Traffic = 84.015	:Link Speed = 85.726	:Delay=0.584	:Cost[2][8]=29.28606
Link#3 Traffic = 55.158	:Link Speed = 64.295	:Delay=0.109	:Cost[1][3]=0.77018
Link#4 Traffic = 84.045	:Link Speed = 85.726	:Delay=0.595	:Cost[3][7]=30.34075
Link#5 Traffic = 19.616	:Link Speed = 21.432	:Delay=0.551	:Cost[2][9]=6.50217
Link#6 Traffic = 14.117	:Link Speed = 21.432	:Delay=0.137	:Cost[6][3]=0.40051
Link#7 Traffic = 49.971	:Link Speed = 64.295	:Delay=0.070	:Cost[7][4]=0.31335
Link#8 Traffic = 0.770	:Link Speed = 2.857	:Delay=0.123	:Cost[4][3]=0.65601
Link#9 Traffic = 2.529	:Link Speed = 2.857	:Delay=0.337	:Cost[6][4]=26.52431
Link#10 Traffic = 4.230	:Link Speed = 21.432	:Delay=0.058	:Cost[4][10]=0.07243
Link#11 Traffic = 47.000	:Link Speed = 85.726	:Delay=0.026	:Cost[7][10]=0.05716
Link#12 Traffic = 12.925	:Link Speed = 64.295	:Delay=0.019	:Cost[8][7]=0.02436
Link#13 Traffic = 4.230	:Link Speed = 21.432	:Delay=0.058	:Cost[10][5]=0.07243
Link#14 Traffic = 64.790	:Link Speed = 85.726	:Delay=0.048	:Cost[5][1]=0.19557
Link#15 Traffic = 27.090	:Link Speed = 64.295	:Delay=0.027	:Cost[8][9]=0.04645
Link#16 Traffic = 0.000	:Link Speed = 85.726	:Delay=0.012	:Cost[8][1]=0.01167
Link#17 Traffic = 46.706	:Link Speed = 85.726	:Delay=0.026	:Cost[9][5]=0.05630
Link#18 Traffic = 58.146	:Link Speed = 85.726	:Delay=0.036	:Cost[5][6]=0.11270

Average Delay=0.01300872

START LOOP(12)

Shortest Path[1] :-1-2-8-7-4	Cost of shortest past No.1 = 29.760
Shortest Path[2] :-3-7-10	Cost of shortest past No.2 = 30.398
Shortest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 6.671
Shortest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 29.618
Shortest Path[5] :-6-4	Cost of shortest past No.5 = 26.524
Shortest Path[6] :-4-3	Cost of shortest past No.6 = 0.656

Shortest Path[7] :-5-6-3 Cost of shortest past No.7 = 0.513

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.732	:Link Speed = 85.729	:Delay=0.040	:Cost[1][2]=0.13721
Link#2 Traffic = 84.048	:Link Speed = 85.729	:Delay=0.595	:Cost[2][8]=30.33995
Link#3 Traffic = 54.997	:Link Speed = 64.297	:Delay=0.108	:Cost[1][3]=0.74337
Link#4 Traffic = 83.937	:Link Speed = 85.729	:Delay=0.558	:Cost[3][7]=26.70236
Link#5 Traffic = 19.685	:Link Speed = 21.432	:Delay=0.572	:Cost[2][9]=7.01828
Link#6 Traffic = 14.157	:Link Speed = 21.432	:Delay=0.137	:Cost[6][3]=0.40496
Link#7 Traffic = 49.974	:Link Speed = 64.297	:Delay=0.070	:Cost[7][4]=0.31341
Link#8 Traffic = 0.783	:Link Speed = 2.857	:Delay=0.124	:Cost[4][3]=0.66385
Link#9 Traffic = 2.526	:Link Speed = 2.857	:Delay=0.337	:Cost[6][4]=26.03204
Link#10 Traffic = 4.217	:Link Speed = 21.432	:Delay=0.058	:Cost[4][10]=0.07232
Link#11 Traffic = 47.000	:Link Speed = 85.729	:Delay=0.026	:Cost[7][10]=0.05716
Link#12 Traffic = 13.037	:Link Speed = 64.297	:Delay=0.020	:Cost[8][7]=0.02447
Link#13 Traffic = 4.217	:Link Speed = 21.432	:Delay=0.058	:Cost[10][5]=0.07232
Link#14 Traffic = 64.729	:Link Speed = 85.729	:Delay=0.048	:Cost[5][1]=0.19440
Link#15 Traffic = 27.011	:Link Speed = 64.297	:Delay=0.027	:Cost[8][9]=0.04625
Link#16 Traffic = 0.000	:Link Speed = 85.729	:Delay=0.012	:Cost[8][1]=0.01166
Link#17 Traffic = 46.695	:Link Speed = 85.729	:Delay=0.026	:Cost[9][5]=0.05627
Link#18 Traffic = 58.183	:Link Speed = 85.729	:Delay=0.036	:Cost[5][6]=0.11299

Average Delay=0.012985776

START LOOP(13)

Shortest Path[1] :-1-3-7-4	Cost of shortest past No.1 = 27.759
Shortest Path[2] :-3-7-10	Cost of shortest past No.2 = 26.760
Shortest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 7.188
Shortest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 30.672
Shortest Path[5] :-6-4	Cost of shortest past No.5 = 26.032
Shortest Path[6] :-4-10-5-6-3	Cost of shortest past No.6 = 0.663
Shortest Path[7] :-5-6-3	Cost of shortest past No.7 = 0.518

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.693	:Link Speed = 85.714	:Delay=0.040	:Cost[1][2]=0.13691
Link#2 Traffic = 83.954	:Link Speed = 85.714	:Delay=0.568	:Cost[2][8]=27.65608
Link#3 Traffic = 54.987	:Link Speed = 64.286	:Delay=0.108	:Cost[1][3]=0.74348
Link#4 Traffic = 83.970	:Link Speed = 85.714	:Delay=0.573	:Cost[3][7]=28.17444
Link#5 Traffic = 19.740	:Link Speed = 21.429	:Delay=0.592	:Cost[2][9]=7.51077

Link#6 Traffic = 14.202	:Link Speed = 21.429	:Delay=0.138	:Cost[6][3]=0.41027
Link#7 Traffic = 49.976	:Link Speed = 64.286	:Delay=0.070	:Cost[7][4]=0.31394
Link#8 Traffic = 0.781	:Link Speed = 2.857	:Delay=0.124	:Cost[4][3]=0.66287
Link#9 Traffic = 2.524	:Link Speed = 2.857	:Delay=0.336	:Cost[6][4]=25.72769
Link#10 Traffic = 4.219	:Link Speed = 21.429	:Delay=0.058	:Cost[4][10]=0.07235
Link#11 Traffic = 47.000	:Link Speed = 85.714	:Delay=0.026	:Cost[7][10]=0.05719
Link#12 Traffic = 13.006	:Link Speed = 64.286	:Delay=0.020	:Cost[8][7]=0.02445
Link#13 Traffic = 4.219	:Link Speed = 21.429	:Delay=0.058	:Cost[10][5]=0.07235
Link#14 Traffic = 64.680	:Link Speed = 85.714	:Delay=0.048	:Cost[5][1]=0.19374
Link#15 Traffic = 26.947	:Link Speed = 64.286	:Delay=0.027	:Cost[8][9]=0.04611
Link#16 Traffic = 0.000	:Link Speed = 85.714	:Delay=0.012	:Cost[8][1]=0.01167
Link#17 Traffic = 46.687	:Link Speed = 85.714	:Delay=0.026	:Cost[9][5]=0.05627
Link#18 Traffic = 58.225	:Link Speed = 85.714	:Delay=0.036	:Cost[5][6]=0.11343
Average Delay=0.013024867			

START LOOP(14)

Shortest Path[1] :-1-2-8-7-4	Cost of shortest past No.1 = 28.131
Shortest Path[2] :-3-7-10	Cost of shortest past No.2 = 28.232
Shortest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 7.680
Shortest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 27.987
Shortest Path[5] :-6-4	Cost of shortest past No.5 = 25.728
Shortest Path[6] :-4-3	Cost of shortest past No.6 = 0.663
Shortest Path[7] :-5-6-3	Cost of shortest past No.7 = 0.524

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.712	:Link Speed = 85.714	:Delay=0.040	:Cost[1][2]=0.13712
Link#2 Traffic = 83.959	:Link Speed = 85.714	:Delay=0.570	:Cost[2][8]=27.84398
Link#3 Traffic = 54.958	:Link Speed = 64.286	:Delay=0.107	:Cost[1][3]=0.73877
Link#4 Traffic = 83.950	:Link Speed = 85.714	:Delay=0.567	:Cost[3][7]=27.54197
Link#5 Traffic = 19.752	:Link Speed = 21.429	:Delay=0.596	:Cost[2][9]=7.62345
Link#6 Traffic = 14.209	:Link Speed = 21.429	:Delay=0.139	:Cost[6][3]=0.41112
Link#7 Traffic = 49.977	:Link Speed = 64.286	:Delay=0.070	:Cost[7][4]=0.31397
Link#8 Traffic = 0.783	:Link Speed = 2.857	:Delay=0.124	:Cost[4][3]=0.66432
Link#9 Traffic = 2.523	:Link Speed = 2.857	:Delay=0.336	:Cost[6][4]=25.64276
Link#10 Traffic = 4.217	:Link Speed = 21.429	:Delay=0.058	:Cost[4][10]=0.07233
Link#11 Traffic = 47.000	:Link Speed = 85.714	:Delay=0.026	:Cost[7][10]=0.05719
Link#12 Traffic = 13.027	:Link Speed = 64.286	:Delay=0.020	:Cost[8][7]=0.02447

Link#13 Traffic = 4.217	:Link Speed = 21.429	:Delay=0.058	:Cost[10][5]=0.07233
Link#14 Traffic = 64.669	:Link Speed = 85.714	:Delay=0.048	:Cost[5][1]=0.19354
Link#15 Traffic = 26.933	:Link Speed = 64.286	:Delay=0.027	:Cost[8][9]=0.04607
Link#16 Traffic = 0.000	:Link Speed = 85.714	:Delay=0.012	:Cost[8][1]=0.01167
Link#17 Traffic = 46.685	:Link Speed = 85.714	:Delay=0.026	:Cost[9][5]=0.05627
Link#18 Traffic = 58.232	:Link Speed = 85.714	:Delay=0.036	:Cost[5][6]=0.11349

Average Delay=0.013024349

START LOOP(15)

Shortest Path[1] :-1-2-8-7-4	Cost of shortest past No.1 = 28.320
Shortest Path[2] :-3-7-10	Cost of shortest past No.2 = 27.599
Shortest Path[3] :-2-9-5-6	Cost of shortest past No.3 = 7.793
Shortest Path[4] :-5-1-2-8	Cost of shortest past No.4 = 28.175
Shortest Path[5] :-6-4	Cost of shortest past No.5 = 25.643
Shortest Path[6] :-4-3	Cost of shortest past No.6 = 0.664
Shortest Path[7] :-5-6-3	Cost of shortest past No.7 = 0.525

Setting traffic in each link and adjusting trunk speed :

Link#1 Traffic = 60.712	:Link Speed = 85.714	:Delay=0.040	:Cost[1][2]=0.13712
Link#2 Traffic = 83.959	:Link Speed = 85.714	:Delay=0.570	:Cost[2][8]=27.84398
Link#3 Traffic = 54.958	:Link Speed = 64.286	:Delay=0.107	:Cost[1][3]=0.73877
Link#4 Traffic = 83.950	:Link Speed = 85.714	:Delay=0.567	:Cost[3][7]=27.54197
Link#5 Traffic = 19.752	:Link Speed = 21.429	:Delay=0.596	:Cost[2][9]=7.62345
Link#6 Traffic = 14.209	:Link Speed = 21.429	:Delay=0.139	:Cost[6][3]=0.41112
Link#7 Traffic = 49.977	:Link Speed = 64.286	:Delay=0.070	:Cost[7][4]=0.31397
Link#8 Traffic = 0.783	:Link Speed = 2.857	:Delay=0.124	:Cost[4][3]=0.66432
Link#9 Traffic = 2.523	:Link Speed = 2.857	:Delay=0.336	:Cost[6][4]=25.64276
Link#10 Traffic = 4.217	:Link Speed = 21.429	:Delay=0.058	:Cost[4][10]=0.07233
Link#11 Traffic = 47.000	:Link Speed = 85.714	:Delay=0.026	:Cost[7][10]=0.05719
Link#12 Traffic = 13.027	:Link Speed = 64.286	:Delay=0.020	:Cost[8][7]=0.02447
Link#13 Traffic = 4.217	:Link Speed = 21.429	:Delay=0.058	:Cost[10][5]=0.07233
Link#14 Traffic = 64.669	:Link Speed = 85.714	:Delay=0.048	:Cost[5][1]=0.19354
Link#15 Traffic = 26.933	:Link Speed = 64.286	:Delay=0.027	:Cost[8][9]=0.04607
Link#16 Traffic = 0.000	:Link Speed = 85.714	:Delay=0.012	:Cost[8][1]=0.01167
Link#17 Traffic = 46.685	:Link Speed = 85.714	:Delay=0.026	:Cost[9][5]=0.05627
Link#18 Traffic = 58.232	:Link Speed = 85.714	:Delay=0.036	:Cost[5][6]=0.11349

Average Delay=0.013024349

ประวัติผู้เขียน

ชื่อผู้เขียน	นายเสรี อัสวารักษ์
วันเดือนปีเกิด	15 สิงหาคม 2510
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตรบัณฑิต สาขาโทรคมนาคม
สถานที่สำเร็จการศึกษา	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2533
ผลงานทางวิชาการที่ได้รับการตีพิมพ์	สุวิพล สิทธิชีวะภาค และ เสรี อัสวารักษ์ <u>“การวิเคราะห์ระบบควาเทียมเพื่อการออกแบบโครงข่าย วิสเสท”</u> การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 19 (พฤษภาคม 2539 กลุ่มที่ 2 หน้า CM56-CM61)